

Im Auftrag des  
Bundesministeriums für Verkehr,  
Bau- und Wohnungswesen

Forschungsbericht FE-Nr. 09.092 G95D

Standardisierung graphischer Daten  
im Straßen- und Verkehrswesen

Teil 2 - Realisierung

Teilbericht D: Das SQL-Datenschema

interactive instruments  
Gesellschaft für Softwareentwicklung mbH  
Trierer Straße 70-72  
53115 Bonn

10.01.2000



Im Auftrag des  
Bundesministeriums für Verkehr,  
Bau- und Wohnungswesen

Forschungsbericht FE-Nr. 09.092 G95D

Standardisierung graphischer Daten  
im Straßen- und Verkehrswesen

Teil 2 - Realisierung

Teilbericht D: Das SQL-Datenschema

von

Dipl. Phys. Clemens Portele

Dipl. Math. Dietmar König

interactive instruments  
Gesellschaft für Softwareentwicklung mbH  
Trierer Straße 70-72  
53115 Bonn

10.01.2000



## Inhaltsverzeichnis

<b>1</b>	<b>Methoden zur Ableitung des SQL-Schemas für den OKSTRA aus dem EXPRESS-Schema des OKSTRA</b>	<b>8</b>
1.1.1	Grundsätzliches Vorgehen	8
1.1.2	Abbildung von Attributen und EXPRESS-Datentypen	9
1.1.3	Abbildung von Relationen	10
1.1.4	Abbildung von Vererbungen	13
1.1.5	Abbildung von WHERE- und UNIQUE-Clauses	16
1.1.6	Verwendung konzeptueller ENTITYs	17
1.1.7	Beispiele für Hinweise im EXPRESS-Code	17
1.1.8	Abkürzungen für Bezeichner im SQL-Schema	22
<b>2</b>	<b>Fachbereich „Netzdaten“</b>	<b>28</b>
2.1	Teilschema „Straßennetz“	28
2.2	Teilschema „Administration“	41
2.3	Teilschema „Verkehr“	48
<b>3</b>	<b>Fachbereich „Bestandsdaten“</b>	<b>53</b>
3.1	Teilschema „Bauliche Straßeneigenschaften“	53
3.2	Teilschema „Straßenausstattungen“	60
<b>4</b>	<b>Fachbereich „Straßenzustandsdaten“</b>	<b>65</b>
4.1	Teilschema „Straßenzustandsdaten“	65
<b>5</b>	<b>Fachbereich „Bauwerksdaten“</b>	<b>72</b>
5.1	Teilschema „Bauwerke“	72
<b>6</b>	<b>Fachbereich „Vermessung“</b>	<b>97</b>
6.1	Teilschema „Vermessung“	97
<b>7</b>	<b>Fachbereich „Entwurf“</b>	<b>98</b>
7.1	Teilschema „Entwurf“	98
<b>8</b>	<b>Fachbereich „Ökologie“</b>	<b>113</b>
8.1	Teilschema „Ökologie“	113
<b>9</b>	<b>Fachbereich „Ingenieurbauwerke“</b>	<b>114</b>
9.1	Teilschema „Ingenieurbauwerke“	114
<b>10</b>	<b>Fachbereich „Projektressourcen“</b>	<b>115</b>
10.1	Teilschema „Projektressourcen“	115
<b>11</b>	<b>Fachbereich „Ausstattung“</b>	<b>116</b>
11.1	Teilschema „Ausstattung“	116
<b>12</b>	<b>Fachbereich „Verkehrsdaten“</b>	<b>117</b>

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

12.1	Teilschema „Zählstellen / dynamische Verkehrsdaten“	117
12.2	Teilschema „Umfelddaten“	139
12.3	Teilschema „Unfalldaten“	143
<b>13</b>	<b>Fachbereich „Beschilderung“</b>	<b>150</b>
13.1	Teilschema „Dynamische verkehrsregelnde Beschilderung“	150
13.2	Teilschema „Statische wegweisende / verkehrsregelnde Beschilderung“	165
13.3	Teilschema „Lichtsignalanlagen“	172
<b>14</b>	<b>Allgemein verwendbare Objekte</b>	<b>182</b>
14.1	Teilschema „Geometrie“	182
14.2	Teilschema „Historisierung“	186
14.3	Teilschema „Allgemeine Objekte“	190

#### **Inhalt dieses Dokuments:**

- Das Datenschema des OKSTRA in der Abbildung nach SQL.
- Um erste Implementierungen des OKSTRA im Rahmen von marktüblichen (relationalen) Datenbanksystemen zu ermöglichen wurde das EXPRESS-Datenschema des OKSTRA nach SQL abgebildet. Dieser Teilbericht enthält grundlegende Erläuterungen zur Ableitung eines SQL-Datenschemas aus einem EXPRESS-Datenschema sowie das resultierende SQL-Datenschema des OKSTRA.

#### **Zielgruppe:**

- Vor allem Software-Entwickler

#### **Voraussetzungen zum Verständnis:**

- OKSTRA-Modellierung, Teilberichte B+C
- SQL 92 (ISO 9075:1992)

## 1 Methoden zur Ableitung des SQL-Schemas für den OKSTRA aus dem EXPRESS-Schema des OKSTRA

Für die Modellierung des abgeleiteten **SQL**-Datenschemas des OKSTRA wird weitestgehend der Entry Level von SQL92 (ISO 9075:1992) verwendet<sup>1</sup>. Darüber hinaus werden lediglich die Verwendung von Kleinbuchstaben in Bezeichnern und der Datentyp 'VARCHAR' aus höheren SQL-Leveln zugelassen.

In diesem Kapitel soll die Umsetzung des EXPRESS-Schemas nach SQL erläutert werden. Die Beispiele sind an vielen Stellen vereinfacht im Vergleich zum tatsächlichen Datenschema des OKSTRA (z.B. wird größtenteils auf die Historisierung verzichtet). Sie sollen nur zur Verdeutlichung des Prinzips der Abbildung dienen und haben keine inhaltliche Relevanz. Der Straßenbezug der Beispiele dient nur dem besseren Verständlichkeit.

Im ersten Abschnitt wird ein kurzer, vereinfachender Überblick über die Methodik der Ableitung gegeben, um das grundsätzliche Vorgehen zu erläutern. In den folgenden Abschnitten werden dann Details dieser Ableitung genauer erläutert. Die Ableitung ist grundsätzlich automatisch möglich, jedoch ist eine abschließende manuelle Nachbearbeitung speziell zur Optimierung des SQL-Schemas punktuell sinnvoll.

Insgesamt ist anzumerken, daß das fehlende Konstrukt der Vererbung in SQL zu einer relativ komplexen Umsetzung in einem RDBMS führt. Die Umsetzung nach SQL ist eine Gratwanderung zwischen der konzeptuellen Nähe zum eigentlichen OKSTRA und der Anpassung des OKSTRA an das relationale Modell. Wir hoffen, mit dem eingeschlagenen Weg einen sinnvollen Kompromiß zwischen diesen beiden Polen gefunden zu haben, sind uns aber gleichzeitig der vorhandenen Probleme dieser Modellierung bewußt. Die Modellierung wurde daher frühzeitig den Experten – auch aus den betroffenen Softwarehäusern – zur Verfügung gestellt; Kommentare wurden berücksichtigt.

### 1.1.1 Grundsätzliches Vorgehen

Als Grundprinzip wird ein EXPRESS-ENTITY in eine SQL-TABLE umgewandelt. Die Attribute und Relationen des ENTITYs werden dabei zu Spalten der TABLE. Der Objektsicht von EXPRESS folgend erhält jede solche TABLE eine Spalte mit einer (nicht fachlich bestimmten) Objekt-ID zur Verwendung als PRIMARY KEY.

Attribute werden in eine Spalte der TABLE mit einem passenden SQL-Datentyp umgewandelt. Hat ein EXPRESS-Datentyp keine geeignete Entsprechung in SQL, so wird er durch eine passende SQL-TABLE dargestellt, die dann von der Spalte referenziert wird.

Für die Abbildung von Relationen werden grundsätzlich die oben beschriebenen Objekt-IDs in den TABLEs verwendet, indem diese in die darstellenden TABLEs von referenzierenden

---

<sup>1</sup> Hierzu ist anzumerken, daß die Festlegung auf einen bestimmten SQL-Sprachumfang trotz der Normung in der ISO problematisch ist, da in der Realität die Implementierungen stets von der Norm abweichen. Für den produktunabhängigen OKSTRA kann als Referenz aber nur die ISO-Norm verwendet werden. Um eine möglichst große Kompatibilität mit den auf dem Markt verfügbaren RDBMS zu erzielen, wurde versucht, sich prinzipiell auf die allgemein verfügbaren Konstrukte von SQL92 zu beschränken, d.h. vor allem auf den Entry Level. Auf Elemente des Intermediate oder gar des Full Level (die im übrigen noch niemand vollständig implementiert hat) wurde daher weitestgehend verzichtet. Die oben genannten Ausnahmen von dieser Regel (deren Funktionalität i.a. in den gängigen RDBMS verfügbar ist) wurden zugelassen, da sonst die Möglichkeiten des OKSTRA/SQL zu stark limitiert gewesen wären.

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

ENTITYs eingetragen werden. Da in SQL-TABLEs nur einfache Datentypen zugelassen sind, kann es bei mehrdeutigen Relationen notwendig sein, eine gesonderte TABLE zu definieren, die die Relation darstellt.

An einem vereinfachten Beispiel soll der grundsätzliche Abbildungsprozeß verdeutlicht werden:

```
EXPRESS:
ENTITY BAB_Knotennummer_reduziert;
--- Attribute :
    Knotennummer      : INTEGER;
    Buchstabe         : OPTIONAL STRING(1);
--- Relationen :
INVERSE
    von_Netzknoten    : SET [1:?] OF Netzknoten
                        FOR hat_BAB_Knotennummer; (* D *)
    auf_Strasse       : SET [1:?] OF Strasse
                        FOR hat_BAB_Knotennummer; (* D *)
END_ENTITY;
```

```
SQL:
CREATE TABLE BAB_Knotennummer_reduziert (
    BAB_Knotennummer_ID
        INTEGER NOT NULL PRIMARY KEY,
    Knotennummer
        INTEGER NOT NULL,
    Buchstabe
        CHAR(1)
)
```

Es erscheint sinnvoll, Hinweise für die Umsetzung im EXPRESS-Code innerhalb von Kommentaren zu verankern. Die verwendeten Hinweise sind in 1.1.7 erläutert.

#### 1.1.2 Abbildung von Attributen und EXPRESS-Datentypen

Der logische EXPRESS-Datentyp *Boolean* wird als sogenannter KEY-TABLE modelliert. In einer solchen KEY-TABLE enthält die erste Spalte einen Schlüssel, hier eine ganze Zahl, zur Referenzierung. Die zweite Spalte enthält eine explizite Darstellung der aufzunehmenden Werte, im allgemeinen als Text. In diese TABLEs werden schon bei Generierung des SQL-Schemas durch SQL-Statements die entsprechenden Werte eingetragen. Attribute, denen ein entsprechender Typ zugrundeliegt, werden dann wie Relationen behandelt.

An diesem Beispiel soll der Aufbau von KEY-TABLEs verdeutlicht werden:

```
SQL:
CREATE TABLE Boolean (
    Boolean_ID
        INTEGER NOT NULL PRIMARY KEY,
    Boolean
        VARCHAR (5)
)
INSERT INTO Boolean VALUES ( 0 , 'FALSE' )
INSERT INTO Boolean VALUES ( 1 , 'TRUE' )
```

Enumerierte Datentypen werden im EXPRESS-Schema des OKSTRA nicht verwendet, da es in den relevanten Fällen (den Schlüsselstabellen) auf die exakte textliche Repräsentierung der Inhalte ankommt. Dieses ist durch ENUMERATIONS nicht zu erreichen, so daß auf die Verwendung dieses Datentyps ganz verzichtet wurde. Schlüsselstabellen werden in der oben beschriebenen Weise in EXPRESS ebenfalls als KEY-TABLEs modelliert.

Wird für ein Attribut lediglich eine (endliche) Wertemenge definiert, ohne entsprechende Schlüssel festzulegen, so werden in der SQL-Abbildung die Werte aus Gründen der Speicherplatzoptimierung in der gegebenen Reihenfolge mit natürlichen Zahlen als Schlüssel versehen und als KEY-TABLE modelliert.

Für alle übrigen selbstdefinierten EXPRESS-TYPES gibt es für die zugrundeliegenden Datentypen Entsprechungen in SQL. Attribute zu diesen TYPES oder elementaren EXPRESS-TYPES werden in Spalten mit Einträgen korrespondierender SQL-Datentypen in der darstellenden TABLE des ENTITYs abgebildet.

Optionale Attribute werden als optionale Spalten in die darstellende TABLE eingetragen. Man hat jedoch auch die Möglichkeit, eine separate TABLE für die Aufnahme des Attributs einzurichten und diese von der darstellenden TABLE des ENTITYs zu referenzieren. Diese Möglichkeit ist jedoch nur in sehr seltenen Fällen (etwa bei extrem speicherintensiven Einträgen) sinnvoll.

Eine Sonderstellung nehmen abgeleitete Attribute (DERIVE) in EXPRESS-ENTITYs ein. Sie werden nicht in das SQL-Schema übernommen, da sie jederzeit aus den vorhandenen Informationen abgeleitet werden können.

### 1.1.3 Abbildung von Relationen

Der einfachste Fall ist, daß ein ENTITY A eine beidseitig eindeutige Relation zu einem ENTITY B hat. In die TABLE für das ENTITY A wird dann eine Spalte eingetragen, die die Objekt-ID in der TABLE für das ENTITY B referenziert. Auf die explizite Modellierung der inversen Relation wird verzichtet, da die Information bereits vollständig durch die explizite Relation gegeben ist.

Etwas komplizierter ist es, wenn genau eine Richtung der Relation mehrdeutig ist. In diesem Fall wird die mehrdeutige Richtung nicht direkt in die TABLEs abgebildet, sondern nur die eindeutige Richtung wird analog zum ersten Fall als referenzierende Spalte in die entsprechende TABLE eingetragen. Ist die mehrdeutige Richtung als Liste modelliert, so wird zusätzlich ein Attribut zur Aufnahme der *Position* in der Liste in die TABLE eingetragen. Alternativ kann die Relation im Fall des Aggregationstyps Liste wie der weiter unten behandelte Fall beidseitig mehrdeutiger Relationen behandelt werden. Eine weitere Möglichkeit ist die Darstellung der Reihenfolge als „Vorgänger-Nachfolger“-Relation über Objekt-IDs, analog zur Modellierung einer gewöhnlichen Relation.

Die Darstellung von 1:n-Relationen soll an einem Beispiel erläutert werden:

```
EXPRESS: 1:n-Relation zwischen Nullpunktsort und Nullpunkt

ENTITY Nullpunktsort_reduziert;
--- Attribute :
--- Relationen :
    bei_Nullpunkt          : Nullpunkt_reduziert;
END_ENTITY;

ENTITY Nullpunkt_reduziert;
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

```
(* KEY_TYP CHAR(8) *)
--- Attribute :
  Zusatz                : Nullpunktkennung;
  zentraler_Nullpunkt   : OPTIONAL BOOLEAN;
--- Relationen :
  in_Netzknoten         : Netzknoten_reduziert;
INVERSE
  hat_Nullpunktsort     : SET [0:?] OF Nullpunktsort_reduziert
                        FOR bei_Nullpunkt;
END_ENTITY;

TYPE Nullpunktkennung = STRING(1);
WHERE
  Buchstabe             : {'A' <= SELF <= 'Z'} OR SELF = ' ';
  (* BEDINGUNG ( SELF >= 'A' AND SELF <= 'Z' ) OR ( SELF = ' ' ) *)
END_TYPE;
```

SQL: Eintrag für die eindeutige Richtung, kein Eintrag für die mehrdeutige Richtung

```
CREATE TABLE Nullpunktsort_reduziert (
  Nullpunktsort_ID
    INTEGER NOT NULL PRIMARY KEY,
  bei_Nullpunkt
    INTEGER NOT NULL REFERENCES Nullpunkt_reduziert
)

CREATE TABLE Nullpunkt_reduziert (
  Nullpunkt_ID
    INTEGER NOT NULL PRIMARY KEY,
  Nullpunkt_ext
    CHAR(8) NOT NULL,
  Zusatz
    CHAR (1) NOT NULL, -- Nullpunktkennung
  CHECK ( ( Zusatz >= 'A' AND Zusatz <= 'Z' ) OR ( Zusatz = ' ' ) ),
  zentraler_Nullpunkt
    INTEGER REFERENCES Logical_Types.Boolean,
  in_Netzknoten
    INTEGER NOT NULL REFERENCES Netzknoten_reduziert,
)
```

Eine beidseitig mehrdeutige Relation wird in zwei jeweils nur einseitig mehrdeutige Relationen aufgespalten. Eine gesonderte TABLE wird definiert, die je eine eindeutige Relation zu den beteiligten ENTITYs besitzt und so die gesamte Relation darstellt.

Anmerkung: Bei Relationen zwischen historisierenden Objekten ist diese Art der Relation der Standardfall, da solche Relationen stets beidseitig multipel sind.

Ist die explizite Richtung der Relation als Liste modelliert, so wird eine zusätzliche Spalte zur Aufnahme der *Position* in der Liste in diese TABLE eingefügt oder eine weitere Referenz auf die darstellende TABLE des Entities in der LIST-Relation ergänzt.

Zur Darstellung beidseitig mehrdeutiger Relationen wieder ein modifiziertes Beispiel aus dem Straßennetz:

```
EXPRESS: m:n-Relation zwischen Nullpunkt und Netzknoten

ENTITY Nullpunkt_reduziert
SUBTYPE OF (Punktobjekt_Modell,historisches_Objekt);
--- Attribute :
  Zusatz                : Nullpunktkennung;
  zentraler_Nullpunkt   : OPTIONAL BOOLEAN;
--- Relationen :
  in_Netzknoten         : SET [1:?] OF Netzknoten; (* D *)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

```
INVERSE
  Beginn_von_Abschnitt_oder_Ast : SET [0:?] OF Abschnitt_oder_Ast
                                  FOR beginnt_bei_NP;
  Ende_von_Abschnitt_oder_Ast   : SET [0:?] OF Abschnitt_oder_Ast
                                  FOR endet_bei_NP;
  verbindet_Verkehr1_Verknuepfg : SET [0:?] OF Verkehrliche_Verknuepfung
                                  FOR ueber_Nullpunkt;
  hat_Nullpunktsort             : SET [1:?] OF Nullpunktsort
                                  FOR bei_Nullpunkt;
END_ENTITY;

ENTITY Netzknoten_reduziert
SUBTYPE OF (Punktobjekt_Modell,historisches_Objekt);
--- Attribute :
  Numerierungsbezirk           : TK25_Blattnummer;
  Nummer                       : lfd_NK_Nummer;
  Knotenart                    : OPTIONAL Knotenart;
  Knotenname                   : OPTIONAL STRING(30);
  Knotenpunktsform            : OPTIONAL Knotenpunktsform;
  Knotenpunktsystem           : OPTIONAL Knotenpunktsystem;
--- Relationen :
INVERSE
  in_Strassenknoten           : SET [1:?] OF Strasse_Netzknoten
                                FOR hat_Netzknoten;
  hat_Nullpunkt               : SET [1:?] OF Nullpunkt FOR in_Netzknoten;
END_ENTITY;
```

SQL: Referenzen nur von der verbindenden TABLE zu den beiden darstellenden TABLES

```
CREATE TABLE Nullpunkt_reduziert (
  Nullpunkt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Nullpunkt_ID) REFERENCES Geometrieschema.Punktobjekt_Modell,
  FOREIGN KEY (Nullpunkt_ID) REFERENCES Historisierung.historisches_Objekt,
  Zusatz
    CHAR(1) NOT NULL, -- Nullpunktkennung
  CHECK ( ( Zusatz >= 'A' AND Zusatz <= 'Z' ) OR ( Zusatz = ' ' ) ),
  zentraler_Nullpunkt
    INTEGER REFERENCES Logical_Types.Boolean
)

CREATE TABLE Nullpunkt__in_Netzknoten (
  Nullpunkt
    INTEGER NOT NULL REFERENCES Nullpunkt,
  Netzknoten
    INTEGER NOT NULL REFERENCES Netzknoten
)

CREATE TABLE Netzknoten_reduziert (
  Netzknoten_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Nullpunkt_ID) REFERENCES Geometrieschema.Punktobjekt_Modell,
  Numerierungsbezirk
    INTEGER NOT NULL, -- TK25_Blattnummer
  CHECK ( ( Numerierungsbezirk >= 1 ) AND ( Numerierungsbezirk <= 9999 ) ),
  Nummer
    INTEGER NOT NULL, -- lfd_NK_Nummer
  CHECK ( ( Nummer >= 1 ) AND ( Nummer <= 999 ) ),
  Knotenart
    INTEGER REFERENCES Knotenart,
  Knotenname
    VARCHAR(30),
  Knotenpunktsform
    INTEGER REFERENCES Knotenpunktsform,
  Knotenpunktsystem
    INTEGER REFERENCES Knotenpunktsystem
)
```

### 1.1.4 Abbildung von Vererbungen

Für das OKSTRA-Schema wurde festgelegt, daß jede Instanz eines SUPERTYPE-ENTITYs in maximal einem direkten SUBTYPE instantiiert wird, d.h. in der EXPRESS-Definition werden sämtliche SUBTYPEs eines ENTITYs innerhalb einer einzigen ONEOF-Clause aufgeführt. Diese Beschränkung entspricht der Modellierung in NIAM und erleichtert die Abbildung des EXPRESS-Schemas nach SQL oder in gängige OO-Programmiersprachen wie C++.

Ein Konzept, das der Vererbung in EXPRESS direkt entspricht, ist in (der hier verwendeten Version von) SQL nicht vorhanden. Daher muß man Vererbung durch andere Konstruktionen simulieren. Hierin liegt die Hauptursache, daß die Abbildung nach SQL problematisch ist.

Die einfachste Variante hierfür ist, daß man Vererbung durch eine gemeinsame Objekt-ID darstellt. In der darstellenden TABLE des SUBTYPEs wird für den PRIMARY KEY ein weiterer FOREIGN KEY Constraint auf den PRIMARY KEY der darstellenden TABLE des SUPERTYPEs eingerichtet. Durch dieses Vorgehen tragen alle zugehörigen Komponenten eines komplexen Objekts die gleiche Objekt-ID. Diese Methode läßt die Struktur der darstellenden TABLEs weitgehend unverändert und wird als Standardvariante zur Modellierung von Vererbung im SQL-Schema verwendet. Allerdings wird nicht deutlich, daß der SUBTYPE die Eigenschaften des SUPERTYPEs erbt.

Folgendes Beispiel illustriert diese Vererbungsvariante:

```
EXPRESS: Ein Abschnitt_oder_Ast erbt vom Linienobjekt_Modell durch Referenz.

ENTITY Abschnitt_oder_Ast
ABSTRACT SUPERTYPE OF (ONEOF(Ast,Abschnitt))
SUBTYPE OF (Linienobjekt_Modell,historisches_Objekt);
  (* KEY_TYP CHAR(16) *)
  (* ARTEN J *)
  (* SAMMELN J *)
--- Attribute :
  Laenge                : OPTIONAL Meter;
  Stadium               : OPTIONAL Stadium;
  Streckenart           : OPTIONAL INTEGER;
  Verkehrsfreigabe     : OPTIONAL Datum;
--- Relationen :
  beginnt_bei_NP        : SET [1:?] OF Nullpunkt; (* D *)
  endet_bei_NP          : SET [1:?] OF Nullpunkt; (* D *)
INVERSE
  gehoert_zu_Strasse    : SET [1:?] OF Strasse
                        FOR hat_Abschnitt_oder_Ast; (* D *)
  hat_Teilabschnitt     : SET [0:?] OF Teilabschnitt
                        FOR auf_Abschnitt_oder_Ast;
  hat_Strassenpunkt     : SET [0:?] OF Strassenpunkt
                        FOR auf_Abschnitt_oder_Ast;
END_ENTITY;

ENTITY Linienobjekt_Modell
ABSTRACT SUPERTYPE OF (ONEOF(Abschnitt_oder_Ast,Strassenelement,
                             Streifenbegrenzung,GDF_Verbindungselement));
--- Attribute :
--- Relationen :
  dargestellt_von_Kante : OPTIONAL SET [1:?] OF Kante;
  dargestellt_von_Linie : OPTIONAL SET [1:?] OF Linie;
END_ENTITY;
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

SQL: Der Abschnitt\_oder\_Ast erhält eine Referenz auf das Linienobjekt\_Modell.

```
CREATE TABLE Abschnitt_oder_Ast_reduziert (
  Abschnitt_oder_Ast_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Abschnitt_oder_Ast_ID) REFERENCES
Geometrieschema.Linienobjekt_Modell,
  Laenge
    REAL, -- Meter
  CHECK ( Laenge >= 0 ),
  Stadium
    INTEGER REFERENCES Stadium,
  Streckenart
    INTEGER,
  Verkehrsfreigabe
    CHAR(10), -- Datum
  CHECK ( Verkehrsfreigabe LIKE '___.__.____' ),
)

CREATE TABLE Linienobjekt_Modell (
  Linienobjekt_Modell_ID
    INTEGER NOT NULL PRIMARY KEY
)
```

Anmerkung: Zur Optimierung des SQL-Datenschemas kann es bei dieser Vererbungsvariante sinnvoll sein, Attribute des SUPERTYPES in die direkten SUBTYPES hinunterzuschieben um direkte Zugriffe darauf zu ermöglichen. Man erhält dann quasi eine Mischung aus der eben beschriebenen Vererbungsvariante und der im folgenden beschriebenen Vererbungsvariante.

Als weitere Variante können die Spalten der SUPERTYPE-TABLE in die SUBTYPE-TABLE eingefügt werden (bis auf die Objekt-ID der SUPERTYPE-TABLE). Hierdurch hat man vom SUBTYPE aus direkten Zugriff auf die Eigenschaften des SUPERTYPES. Allerdings wird die Beziehung zwischen SUBTYPE und SUPERTYPE im SQL-Schema nicht mehr deutlich.

Hierzu wieder ein Beispiel:

EXPRESS: Ein Nullpunktsort erbt vom Straßenpunkt durch Herunterziehen (DOWN).

```
ENTITY Nullpunktsort_erbt
SUBTYPE OF (Strassenpunkt);
--- Attribute :
--- Relationen :
  bei_Nullpunkt          : SET [1:?] OF Nullpunkt; (* D *)
INVERSE
  Beginn_von_SE          : SET [0:?] OF Strassenelement
  Ende_von_SE            : SET [0:?] OF Strassenelement
                        FOR beginnt_bei_NPO; (* S 2 *)
                        FOR endet_bei_NPO; (* S 2 *)
END_ENTITY;

ENTITY Strassenpunkt
ABSTRACT SUPERTYPE OF (ONEOF(Strassenpunkt_TA, Strassenpunkt_PO, Nullpunktsort));
(* VERERBEN DOWN *)
(* KONZEPTUELL J *)
--- Attribute :
  Station                : Meter;
--- Relationen :
  auf_Abschnitt_oder_Ast : Abschnitt_oder_Ast;
END_ENTITY;
```

SQL: Informationen vom Straßenpunkt werden in den Nullpunktsort eingetragen.

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Nullpunktsort (  
  Nullpunktsort_ID  
    INTEGER NOT NULL PRIMARY KEY,  
  Station  
    REAL NOT NULL, -- Meter  
  CHECK ( Station >= 0 ),  
  auf_Abschnitt_oder_Ast  
    INTEGER NOT NULL REFERENCES Abschnitt_oder_Ast  
)
```

Je nach Bedeutung des SUBTYPEs im Datenschema kann es sinnvoll sein, auf die darstellende TABLE zu verzichten. Dazu fügt man die Spalten der (normalerweise für dieses ENTITY definierten) darstellenden TABLE in die SUPERTYPE-TABLE ein. Dies führt zur dritten verwendeten Variante zur Darstellung von Vererbung im SQL-Schema.

Betrachte folgendes Beispiel:

```
EXPRESS: Modellierung eines Zeitraums  
  
ENTITY Zeitraum_reduziert  
ABSTRACT SUPERTYPE OF (ONEOF(einfacher_Zeitraum, komplexer_Zeitraum));  
--- Attribute :  
--- Relationen :  
INVERSE  
  erster_in          : SET [0:?] OF komplexer_Zeitraum  
                      FOR hat_ersten_Zeitraum;  
  zweiter_in         : SET [0:?] OF komplexer_Zeitraum  
                      FOR hat_zweiten_Zeitraum;  
END_ENTITY;  
  
ENTITY einfacher_Zeitraum  
SUBTYPE OF (Zeitraum);  
(* ERBEN UP *)  
--- Attribute :  
--- Relationen :  
  hat_Startdatum    : Startdatum;  
  hat_Dauer          : Dauer;  
END_ENTITY;  
  
ENTITY komplexer_Zeitraum  
SUBTYPE OF (Zeitraum);  
(* ERBEN UP *)  
--- Attribute :  
--- Relationen :  
  hat_ersten_Zeitraum : Zeitraum;  
  hat_zweiten_Zeitraum : Zeitraum;  
  hat_Operator        : Operator;  
END_ENTITY;
```

SQL: darstellende TABLE des Zeitraums nimmt Spalten für einfachen und komplexen Zeitraum auf

```
CREATE TABLE Zeitraum (  
  Zeitraum_ID  
    INTEGER NOT NULL PRIMARY KEY,  
  hat_ersten_Zeitraum  
    INTEGER REFERENCES Zeitraum,  
  hat_zweiten_Zeitraum  
    INTEGER REFERENCES Zeitraum,  
  hat_Operator  
    CHAR(1) REFERENCES Operator,  
  hat_Startdatum  
    INTEGER REFERENCES Startdatum,  
  hat_Dauer  
    INTEGER REFERENCES Dauer,
```

)

Welche der drei Varianten in welchen Kombinationen gewählt werden sollten, hängt von den beteiligten ENTITYs und deren Rollen im Datenschema ab. Hier ist a priori keine eindeutige Festlegung möglich. In diesem Bereich entscheidet sich maßgeblich, wie weit man sich vom objektorientierten Ansatz des EXPRESS-Datenschemas entfernt, hin zu einem relationalen Ansatz, den SQL repräsentiert.

#### 1.1.5 Abbildung von WHERE- und UNIQUE-Clauses

Beide Konzepte sind sowohl in EXPRESS als auch in SQL vorhanden. Die Verwendung von UNIQUE-Clauses, die die Eindeutigkeit der Werte gewisser Kombinationen von Einträgen erzwingen, ist für EXPRESS-ENTITYs und SQL-TABLEs völlig analog, so daß bei deren Umsetzung keine Schwierigkeiten auftauchen. Problematischer sind aufgrund der unterschiedlichen Ausdrucksfähigkeit von EXPRESS und (speziell Entry-Level-)SQL die WHERE-Clauses, die die zugelassenen Wertebereiche von Einträgen beschränken. In manchen Fällen können die definierenden Statements direkt aus EXPRESS in SQL übernommen werden, in den meisten Fällen ist dies jedoch nicht möglich. Dann besteht die Möglichkeit, eine Übersetzung nach SQL direkt als Hinweis im EXPRESS-Schema mitzugeben, was aber oft aufgrund der Beschränkung auf den Entry-Level von SQL ebenfalls nicht möglich ist.

Ein Beispiel einer WHERE-Clause, bei der man eine relativ elementare Übersetzung angeben kann, ist das Startdatum eines (einfachen) Zeitraums:

EXPRESS: ENTITY Startdatum mit Bedingungen an die Attribute

```
ENTITY Startdatum_reduziert;
--- Attribute :
    Jahr                : OPTIONAL INTEGER;
    Monat_im_Jahr       : OPTIONAL INTEGER;
    Woche_im_Jahr       : OPTIONAL INTEGER;
--- Relationen :
WHERE
    nur_Monat_oder_Woche : NOT ( EXISTS(Monat_im_Jahr)
                                AND EXISTS(Woche_im_Jahr));
    (* BEDINGUNG ( Monat_im_Jahr IS NULL ) OR ( Woche_im_Jahr IS NULL ) *)
END_ENTITY;
```

SQL: Bedingungen werden in SQL umgesetzt

```
CREATE TABLE Startdatum (
    Startdatum_ID
        INTEGER NOT NULL PRIMARY KEY,
    Jahr
        INTEGER,
    Monat_im_Jahr
        INTEGER,
    Woche_im_Jahr
        INTEGER,
    CHECK ( ( Monat_im_Jahr IS NULL ) OR ( Woche_im_Jahr IS NULL ) )
)
```

#### 1.1.6 Verwendung konzeptueller ENTITYs

Konzeptuelle ENTITYs verhalten sich bei der Umsetzung so, daß sie nicht als eigenständige Objekte erscheinen, sondern bei jeder Referenzierung von einem anderen ENTITY ihre Informationen direkt in die Darstellung dieses ENTITYs eintragen. Dieser Weg ist vor allem dann sinnvoll, wenn durch ein ENTITY Informationen transportiert werden, die einen sehr schwachen Objekt-Charakter haben (z.B. Verortungsinformationen), oder wenn es sich bei dem ENTITY um ein Hilfskonstrukt handelt, das nur der größeren konzeptuellen Klarheit im Schema dient.

Dies soll wieder an einem Beispiel verdeutlicht werden:

EXPRESS: Ein Punktobjekt hat eine Relation zu dem konzeptuellen Straßenpunkt\_PO.

```
ENTITY Strassenpunkt
ABSTRACT SUPERTYPE OF (ONEOF(Strassenpunkt_TA,Strassenpunkt_PO,Nullpunktsort))
SUBTYPE OF (Punktobjekt_Modell);
  (* VERERBEN DOWN *)
  (* KONZEPTUELL J *)
--- Attribute :
  Station                : Meter;
--- Relationen :
  auf_Abschnitt_oder_Ast : Abschnitt_oder_Ast_reduziert;
END_ENTITY;

ENTITY Strassenpunkt_PO
SUBTYPE OF (Strassenpunkt);
  (* ERBEN DOWN *)
  (* KONZEPTUELL J *)
--- Attribute :
--- Relationen :
INVERSE
  gehoert_zu_PO          : Punktobjekt FOR bei_Strassenpunkt;
END_ENTITY;

ENTITY Punktobjekt
SUPERTYPE OF (ONEOF(Schichtbegrenzung,Streifenbegrenzung,Durchlass,Kreuzung,
                    Betriebskilometer));
--- Attribute :
--- Relationen :
  bei_Strassenpunkt     : Strassenpunkt_PO;
END_ENTITY;
```

SQL: Das Punktobjekt erhält die Eigenschaften vom Straßenpunkt(\_PO).

```
CREATE TABLE Punktobjekt (
  Punktobjekt_ID
    INTEGER NOT NULL PRIMARY KEY,
  Station
    REAL NOT NULL, -- Meter
  CHECK ( Station >= 0 ),
  auf_Abschnitt_oder_Ast
    INTEGER NOT NULL REFERENCES Abschnitt_oder_Ast_reduziert
)
```

#### 1.1.7 Beispiele für Hinweise im EXPRESS-Code

Mit Hilfe von Hinweisen im EXPRESS-Code können die oben beschriebenen Optionen und weitere Feineinstellungen der SQL-Ableitung festgelegt werden. Diese Hinweise werden innerhalb von Kommentaren gehalten (beispielsweise "( \* ERBEN DOWN \* )"), so daß sie keinerlei Auswirkung auf das OKSTRA-Schema selbst haben. Wird ein solcher Hinweis in

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

einem (mindestens) zweifach verschachtelten Kommentar gegeben, bleibt er wirkungslos. Im folgenden sollen die Hinweise erläutert werden, die derzeit für das EXPRESS-Schema des OKSTRA erklärt sind:

Schlüssel	Argument	Beschreibung
BEZ_LAENGE	<Zahl>	Die angegebene Zahl legt die maximal mögliche Länge von Bezeichnern fest. Bei Überschreitung der Grenze wird der Bezeichner entweder auf die Maximallänge gekürzt oder durch einen Platzhalter ersetzt. Die Grenze gilt sowohl für TABLE- und COLUMN-Namen als auch für Bezeichner von CONSTRAINTs. Entry-Level-SQL erlaubt Bezeichner mit einer Länge bis zu 18 Zeichen. Diese Länge wurde als Voreinstellung <sup>2</sup> gewählt.
BEGRENZER	<Zeichen>	Das angegebene Zeichen wird verwendet, um SQL-Statements zu begrenzen. Voreingestellt ist der Leerstring, d.h. das Ende eines SQL-Statements wird nicht durch ein gesondertes Zeichen vermerkt.
SQL_SCHEMA	J N	Diese Option legt fest, ob die gegebene Unterteilung in Schemas in das SQL-Schema übernommen wird. Die Voreinstellung ist J (Ja).
VERERBUNG	DOWN UP	Diese Option legt global eine von der Darstellung als FOREIGN KEY Constraint abweichende Variante für die Vererbung fest. Für die Umsetzung der Vererbung in EXPRESS sind mehrere Vorgehensweisen möglich: <ul style="list-style-type: none"><li>• Eine TABLE, die einen SUBTYPE darstellt, erhält eine Referenz auf eine TABLE, die einen SUPERTYPE darstellt (Standardfall, muß nicht gesondert eingestellt werden).</li><li>• Eine TABLE, die einen SUBTYPE darstellt, erhält sämtliche relevanten Felder aus der TABLE, die einen SUPERTYPE darstellt, als COLUMNS (Variante DOWN). Der SUPERTYPE würde dann selbst nicht mehr dargestellt.</li><li>• Eine TABLE, die einen SUPERTYPE darstellt, erhält sämtliche relevanten Felder aus der TABLE, die einen SUBTYPE</li></ul>

---

<sup>2</sup> Zur besseren Lesbarkeit wurde für das nachfolgend dargestellte SQL-Schema eine Beschränkung der Bezeichnerlänge auf 30 Zeichen gewählt. Einige Bezeichner mußten daher ersetzt werden. Eine Tabelle mit der Zuordnung der Abkürzungen zu den ursprünglichen Bezeichnern findet sich in 1.1.8.

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

		darstellt, als COLUMNS (Variante UP). Mit dieser Variante hat man die Umkehr der Vererbungsvariante DOWN zur Verfügung. Der SUBTYPE würde entsprechend nicht mehr selbst dargestellt.
HIERARCHIE	J N	Zur Vermeidung von Vorwärtsreferenzen im SQL-Datenschema kann versucht werden, SUPERTYPES stets vor den SUBTYPES und referenzierte TABLEs stets vor referenzierenden TABLEs zu definieren. Zyklische Referenzen werden dann in einem gesonderten Bereich vordefiniert. Voreingestellt ist N (Nein).
CONSTR_BEZ	J N	Diese Option legt fest, ob benannte CONSTRAINTS verwendet werden dürfen oder nicht. Entry-Level-SQL unterstützt dies nicht. Die Voreinstellung ist daher N (Nein).
CONSTR_NUM	J N	Bezeichner von CONSTRAINTs können entweder vom EXPRESS-Code übernommen werden oder in der Reihenfolge der Definition aufsteigend numeriert werden, wobei die Verbindung zwischen diesem Platzhalter und der tatsächlichen Bedeutung vermerkt werden muß. Bei komplexen Vererbungsbäumen empfiehlt sich die automatische Generierung der Bezeichner. Voreingestellt ist N (Nein).
CHAR_LAENGE	<Zahl>	SQL bietet die Möglichkeit, Texte mit fester oder variabler Speicherlänge zu definieren. Diese Option legt fest, ab welcher Länge variable Speicherung erzwungen wird, d.h. auf jeden Fall VARCHAR definiert wird und nicht CHAR. Voreingestellt ist eine maximale Länge von 12 Zeichen.
BEZ_NAMEN	J N	Das SQL-Schema ist in Anlehnung seiner Herkunft vom EXPRESS-Schema relativ objekthaft. In diesem Sinne tragen TABLEs Objekt-IDs 'ID', die eine eindeutige Identifikation ermöglichen. Die COLUMN-Namen dieser Objekt-IDs kann man durch diese Option mit dem Namen des Objekts als Präfix versehen, d.h. etwa 'Netznoten_ID'. Voreingestellt ist J (Ja).
QUELLCODE	J N	Der EXPRESS-Quellcode von ENTITYs oder TYPEs kann vor den darstellenden TABLEs als Kommentar in den SQL-Code eingefügt werden. Voreingestellt ist N (Nein).
FUNCTIONS	J N	Der EXPRESS-Quellcode von FUNCTIONs kann jeweils nach Bearbeitung des SCHEMAS,

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

		in dem sie definiert wurden, in den SQL-Code als Kommentar eingefügt werden. Voreingestellt ist N (Nein).
SQL	:	Dieser Schlüssel bewirkt, daß die folgenden Zeilen (bis zum END_SQL-Schlüssel oder dem Ende des Kommentars) direkt im Anschluß an die Darstellung des aktuellen ENTITYs in den SQL-Code eingefügt werden.
END_SQL		Ende der SQL-Statements
ERBEN	DOWN UP NONE	Innerhalb der Definition eines ENTITYs kann man mit dieser Option festlegen, mit welcher Vererbungsvariante dieses ENTITY von seinen SUPERTYPEs erbt. Wird eine solche lokale Vererbungsvariante festgelegt, so wird sie der global festgelegten vorgezogen. Als Sonderfall gibt es hier die Variante NONE, die eventuelle Festlegungen der SUPERTYPEs außer Kraft setzt und die globale Variante einsetzt.
VERERBEN	DOWN	Mit dieser Option kann eine Vererbungsvariante vom SUPERTYPE aus erzwungen werden. Eine solche Festlegung überstimmt sowohl die globale als auch die lokale Vererbungsvariante, die eventuell vom SUBTYPE aus festgelegt worden ist. Sinnvoll erscheint in diesem Zusammenhang nur die Variante DOWN.
KONZEPTUELL	J N	Mit dieser Option kann ein ENTITY als konzeptuell definiert werden. Das bedeutet, daß die COLUMNS dieses Objekts im SQL-Code direkt in ein referenzierendes Objekt eingefügt werden und die entsprechende Referenz ersetzen.
KEY_TYP	SQL-Datentyp	Mit dieser Option kann eine zusätzliche Spalte für einen externen Bezeichner dieser Objekte mit dem angegebenen Datentyp in die darstellende TABLE des ENTITYs eingefügt werden. Die Spalte erhält den Namen '<ENTITY-Name>_ext'.
KEY_NAME	<Text>	Mit dieser Option kann ein beliebiges Attribut eines ENTITYs zur Verwendung als Objekt-Identifizier bestimmt werden. Die entsprechende COLUMN übernimmt dann die Rolle der sonst extra eingefügten Objekt-ID.
ERSETZEN	$[E_1:] \{E_i\}$	Diese Option kann innerhalb eines ENTITYs genutzt werden. Sie bewirkt, daß das ENTITY durch das ENTITY $E_1$ ersetzt, d.h. umbenannt, wird, und Relationen zu diesem ENTITY durch je eine Relation zu den $\{E_i\}$ ersetzt wird. (Diese

		Option ist insbesondere für Konstruktionen mit Hilfs-ENTITYs gedacht.)
OPT_SEPARAT	J N	Für ein optionales Attribut kann mit dieser Option festgelegt werden, ob eine eigene TABLE für dieses Attribut erzeugt werden soll (J), oder ob es als optionale COLUMN aufgenommen werden soll (N). Voreingestellt ist N (Nein).
BEDINGUNG	SQL-Statement	Im Anschluß an ein WHERE-Statement kann eine Übersetzung in SQL direkt angegeben werden. Der gesamte restliche Text bis zum Ende des Kommentars wird hierbei als Bedingung in den SQL-Code übernommen.
EINMAL	$E_1\{, E_i\}$	Mit dieser Option kann für ein ENTITY festgelegt werden, daß die ENTITYs $E_i$ in der darstellenden TABLE maximal einmal vorkommen. Diese Eigenschaft überträgt sich auch auf Relations-TABLEs, die dieses ENTITY referenzieren.
KEINMAL	$E_1\{, E_i\}$	Mit dieser Option kann für ein ENTITY festgelegt werden, daß die ENTITYs $E_i$ in der darstellenden TABLE gar nicht vorkommen. Diese Eigenschaft überträgt sich auch auf Relations-TABLEs, die dieses ENTITY referenzieren.
ARTEN	J N	Durch diesen Hinweis kann für ein ENTITY automatisch eine Schlüsseltabelle zur Aufnahme der Ausprägungen (SUBTYPEs) des ENTITYs erzeugt und referenziert werden. Voreingestellt ist N (Nein).
SAMMELN	J N	Dieser Hinweis führt dazu, daß die Attribute des ENTITYs in einem SUPERTYPE gesammelt werden (sofern die Option ARTEN nicht gesetzt ist). Für den betreffenden SUPERTYPE müssen die Optionen ARTEN und SAMMELN gesetzt sein. Das ENTITY selbst erscheint dann nicht mehr in der Übersetzung. Referenzen werden auf den entsprechenden SUPERTYPE umgelenkt. Voreingestellt ist N (Nein).
BUENDELN	J N	Mit diesem Hinweis kann man festlegen, daß multiple Relationen von diesem ENTITY in einer einzigen Relations-TABLE gebündelt werden. Gedacht ist bei dieser Option an den Fall von Relationen, die durch die Option ERSETZEN aufgespalten werden. Voreingestellt ist N (Nein).
AUFZAEHLUNG	J N	Eine Tabelle, die nur zur Aufzählung möglicher

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

		Werte eines Attributs dient, d.h. ein einziges Attribut des entsprechenden Datentyps (im allgemeinen 'STRING') hat, wird bei der Umsetzung nach SQL zu einer Schlüsseltable mit einem INTEGER-Schlüssel erweitert.
MIX_IN	$A_i \{, A_i\}$	Die angegebenen Attribute werden nicht in die darstellende TABLE des ENTITYs selbst, sondern in alle direkten SUBTYPEs eingefügt.
BEMERKUNG	<Text>	Die Bemerkung wird nach der darstellenden TABLE für das ENTITY, in dem sie gegeben wird, in den SQL-Code eingefügt.
D	(kein Argument)	Die Relation ist konzeptuell eindeutig und wurde durch die Integration der Historisierung multipel.
S	[h]	Die Relation ist auch konzeptuell multipel und zwar als SET-Relation mit Obergrenze h. Wird keine Obergrenze angegeben, so ist sie auch konzeptuell unbestimmt.
L	[h]	Die Relation ist auch konzeptuell multipel und zwar als LIST-Relation mit Obergrenze h. Wird keine Obergrenze angegeben, so ist sie auch konzeptuell unbestimmt.

#### 1.1.8 Abkürzungen für Bezeichner im SQL-Schema

Für das SQL-Schema des OKSTRA wurde aufgrund der Beschränkungen marktgängiger RDBMS eine maximale Bezeichnerlänge von 30 Zeichen festgelegt. Bezeichner, deren Länge diese Grenze übertraf, wurden durch Abkürzungen ersetzt. Die folgende Tabelle enthält die Zuordnung der alten (überlangen) Bezeichner zu den neuen (gekürzten) Bezeichnern.

ALT	NEU
Strasse__hat__Strassenbezeichnung	Strasse__hat__Bezeichnung
Strasse__hat__Abschnitt_oder_Ast	Strasse__hat__AoA
Strasse_Netzknoten__hat__Netzknoten	Strasse_NK__hat__Netzknoten
Strasse_Netzknoten__hat__Vorgaenger	Strasse_NK__hat__Vorgaenger
Netzknoten__hat__BAB_Knotennummer	NK__hat__BAB_Knotennummer
Abschnitt_oder_Ast__beginnt_bei_NP	AoA__beginnt_bei_Nullpunkt
Abschnitt_oder_Ast__endet_bei_NP	AoA__endet_bei_Nullpunkt
Verkehrliche_Verknuepfung__von__Strassenelement	VV__von__Strassenelement
Verkehrliche_Verknuepfung__nach__Strassenelement	VV__nach__Strassenelement
Verkehrliche_Verknuepfung__ueber__Nullpunkt	VV__ueber__Nullpunkt
Netzbereich__enthaelt__Netzbereichskomp	Netzbrc__enthaelt__Komponenten
Bereichsobjekt__hat__Netzbereich	Bereichsobj__hat__Netzbereich
Teilnetz__enthaelt__Teilnetzkomponente	Teilnetz__enthaelt__Komponente
Routenkomponente__hat__Vorgaenger	Routenkomp__hat__Vorgaenger
Route__entlang__Routenkomponente	Route__entlang__Komponente
Strassenelement__beginnt_bei_NPO	SE__beginnt_bei_Nullpunktsort
Regierungsbezirk__ist_in_Bundesland	RBZ__ist_in_Bundesland
Kreis_kreisfreie_Stadt__ist_in_Regierungsbezirk	Kreis__ist_in_RBZ
Kreis_kreisfreie_Stadt__hat__Kreisverwaltung	Kreis__hat__Verwaltung

# OKSTRA – Teilprojekt 4 – Schlußbericht

## Teilbericht D - Das SQL-Datenschema

ALT	NEU
Gemeindebezirk__ist_in_Kreis_kreisfreie_Stadt	Gemeinde__ist_in_Kreis
Gemeindebezirk__hat_Gemeindeverwaltung	Gemeinde__hat_Verwaltung
Ortsteil__ist_in_Gemeindebezirk	Ortsteil__ist_in_Gemeinde
Regierungspraesidium_Landesamt__untersteht_Land_Ministerium	Landesamt__untersteht_Land
Amt__untersteht_Regierungspr_Ldsamt	Amt__untersteht_Landesamt
UI_Vereinbarung__mit_UI_Partner	UI_Vereinbarung__mit_Partner
Punktobjekt_Modell__dargestellt_von_Knoten	Punktobjekt_M__hat_Knoten
Punktobjekt_Modell__dargestellt_von_Punkt	Punktobjekt_M__hat_Punkt
Linienobjekt_Modell__dargestellt_von_Kante	Linienobjekt_M__hat_Kante
Linienobjekt_Modell__dargestellt_von_Linie	Linienobjekt_M__hat_Linie
Flaechenobjekt_Modell__dargestellt_von_Masche	Flaechenobjekt_M__hat_Masche
Flaechenobjekt_Modell__dargestellt_von_Flaeche	Flaechenobjekt_M__hat_Flaeche
Volumenobjekt_Modell__dargestellt_von_Koerper	Volumenobjekt_M__hat_Koerper
Volumenobjekt_Modell__dargestellt_von_Volumen	Volumenobjekt_M__hat_Volumen
Linie__besteht_aus_Linienelementen	Linie__aus_Linienelementen
Linienelement_Spline__hat_Stuetzpunkte	Spline__hat_Stuetzpunkte
Teilbauwerk__hat_Strassenbaudienststelle	Teilbw__hat_Strbaudienststelle
Teilbauwerk__hat_Bauwerksbeschilderung	Teilbw__hat_Bauwbeschilderung
Teilbauwerk__hat_Bauwerksbelaege	Teilbw__hat_Bauwerksbelaege
Teilbauwerk__hat_Hauptbaustoff_Beton	Teilbw__hat_Hauptbaust_Beton
Teilbauwerk__hat_Hauptbaustoff_Stahl	Teilbw__hat_Hauptbaust_Stahl
Teilbauwerk__hat_Hauptbaustoff_Verbundwst	Teilbw__hat_Hauptbaust_Verbwst
Teilbauwerk__hat_Schadensdaten_abg_Prfg	Teilbw__hat_Schadensd_abg_Prfg
Teilbauwerk__hat_Pruefungsdaten_abg_Prfg	Teilbw__hat_Pruefgsd_abg_Prfg
Teilbauwerk__hat_gegenw_dok_Schadensstand	Teilbw__hat_gegw_dok_Schadstd
Teilbauwerk__hat_gegenw_dok_Pruefungsstand	Teilbw__hat_gegw_dok_Pruefstnd
Teilbauwerk__hat_durchgef_Pruefungen_Messgn	Teilbw__hat_durchgef_Prf_Mess
Teilbauwerk__hat_Prueffahrzeuge_Pruefger	Teilbw__hat_Prueffz_Pruefger
Teilbauwerk__hat_Pruefanweisungen	Teilbw__hat_Pruefanweisungen
Teilbauwerk__hat_Verwaltungsmassnahme	Teilbw__hat_Verwaltungsmassn
Teilbauwerk__hat_Bau_und_Erhaltungsmassn	Teilbw__hat_Bau_u_Erhaltungsmassn
Teilbauwerk__hat_Anlagen_Bauwerksbuch	Teilbw__hat_Anlagen_Bauwrbuch
Teilbauwerk__hat_Entwuerfe_und_Berechnungen	Teilbw__hat_Entwue_und_Berechn
Teilbauwerk__hat_Leitungen_an_Bauwerken	Teilbw__hat_Leitungen_an_Bauw
Teilbauwerk__hat_statisches_System_Tragefgkt	Teilbw__hat_stat_System_Tragef
Teilbauwerk__hat_Erd_und_Felsanker	Teilbw__hat_Erd_und_Felsanker
Teilbauwerk__hat_Brueckenseile_und_Kabel	Teilbw__hat_Brueckseil_u_Kabel
Teilbauwerk__hat_Fahrbahnuebergangskonstr	Teilbw__hat_Fahrbuebergkonstr
Teilbauwerk__hat_Schutzeinrichtungen	Teilbw__hat_Schutzeinrichtgen
Teilbauwerk__hat_Betonersatz_nach_ZTV_SIB	Teilbw__hat_Betonersatz
Teilbauwerk__hat_Oberflschutz_nach_ZTV_SIB	Teilbw__hat_Oberflaechenschutz
Teilbauwerk__hat_Reaktionsharzgeb_Duennbel	Teilbw__hat_Reaktharzgeb_Duennb
Bruecke__hat_Brueckenfeld_Stuetzung	Teilbw__hat_Brueckenf_Stuetz
Tunnel_Trogbauwerk__hat_Segmente_Tunnel_Trogbw	Tunnel_Trogbw__hat_Segmente
Tunnel_Trogbauwerk__hat_Tunnelbeleuchtung	Tunnel_Trogbw__hat_Tunbeleucht
Tunnel_Trogbauwerk__hat_Tunnellueftung	Tunnel_Trogbw__hat_Tunnellueft
Tunnel_Trogbauwerk__hat_Tunnel_Zentrale_Anlagen	Tunnel_Trogbw__hat_Zentr_Anlg
Tunnel_Trogbauwerk__hat_Tunnelsicherheit	Tunnel_Trogbw__hat_Tunsicherh
Tunnel_Trogbauwerk__hat_Tunnel_Verkehrseinrichtgn	Tunnel_Trogbw__hat_Tunverkeinr
Laermschutzbauwerk__hat_Segmente_Laermschutzbw	Laermschbw__hat_Segmente
Stuetzbauwerk__hat_Segmente_Stuetzbw	Stuetzbauwerk__hat_Segmente
Bau_und_Erhaltungsmassnahme__hat_Kosten_fuer_Bau_Erh_u_Betr	Bau_u_Erhaltungsmassn__hat_Kosten
Hauptbaustoff_Stahl_Holz_Kunst__hat_Korrosionsschutzbeschichtg	Hauptbst_SHK__hat_Korroschutz
ZEB_Objekt__gilt_fuer_Querschnittstreifen	ZEB_Obj__gilt_f_Querschnittstr
ZEB_Objekt__gilt_fuer_Fahrstreifen	ZEB_Obj__gilt_f_Fahrstreifen
ZEB_Objekt__hat_ZEB_Parameterwert	ZEB_Obj__hat_ZEB_Parameterwert
Art_des_ZEB_Objektes__hat_Zustandsmerkmal	Art_d_ZEB_Obj__hat_Zustmerkmal
ZEB_Parameterliste__hat_ZEB_Parameter	ZEB_Paramliste__hat_ZEB_Param
Zustandsgroesse_o_Zustandswert__bestimmbar_durch_ZEB_Methode	Zustgr_o_Zustw__bestbar_d_Meth
Zustandsindikator__hat_Zustgroesse_o_Zustwert	Zustindik__hat_Zustgr_o_Zustw

# OKSTRA – Teilprojekt 4 – Schlußbericht

## Teilbericht D - Das SQL-Datenschema

ALT	NEU
Zustandsmerkmal__hat_Zustandsindikator	Zustmerkmal__hat_Zustindikator
Kombinationsmethode__hat_Zustandsteilwerte	Kombmethode__hat_Zustteilwerte
Strassenentwaesserung__hat_entwaess_Fahrbahnflaeche	Strassenentw__hat_entw_Fahrbfl
Strassenentwaesserung__hat_Lage_Entwaesserung	Strassenentw__hat_Lage_Entw
Strassenentwaesserung__hat_Art_der_Entwaesserung	Strassenentw__hat_Art_d_Entw
Strassenentwaesserung__hat_Dokument	Strassenentw__hat_Dokument
Art_der_Entwaesserung__hat_Leitung	Art_der_Entwaess__hat_Leitung
Fahrstreifen_Nummer__auf_Querschnittstreifen	Fahrstr_Nr__auf_Querschnittstr
Beschraenkung_verkehrlich__gilt_fuer_Verkehrsrichtung	Beschr_verk__gilt_f_Verkrichtg
Beschraenkung_verkehrlich__gilt_fuer_Fahrzeugart	Beschr_verk__gilt_f_Fahrzgart
Beschraenkung_verkehrlich__im_Zeitraum	Beschr_verk__im_Zeitraum
Beschraenkung_verkehrlich__gilt_fuer_Fahrstreifen	Beschr_verk__gilt_f_Fahrstreif
Strassenbeschreibung_verkehrl__gilt_fuer_Verkehrsrichtung	Strbeschr_verk__gilt_f_Vrichtg
Strassenbeschreibung_verkehrl__gilt_fuer_Fahrzeugart	Strbeschr_verk__gilt_f_Fzart
Strassenbeschreibung_verkehrl__im_Zeitraum	Strbeschr_verk__im_Zeitraum
Strassenbeschreibung_verkehrl__gilt_fuer_Fahrstreifen	Strbeschr_verk__gilt_f_Fahrstr
Schicht__beginnt_bei_Schichtbegrenzung	Schicht__beg_bei_Schichtbegr
Schicht__endet_bei_Schichtbegrenzung	Schicht__endet_bei_Schichtbegr
Querschnittstreifen__beginnt_bei_Streifenbegrenzung	Querschnstr__beg_bei_Streifbeg
Querschnittstreifen__endet_bei_Streifenbegrenzung	Querschnstr__end_bei_Streifbeg
Querschnittstreifen__hat_Streifenart	Querschnstr__hat_Streifenart
Kreuzung_Strasse_Weg__ist_Bauwerk	Kreuz_Str_Weg__ist_Bauwerk
Kreuzungs_o_Einmuendungsplang__hat_Achse	Kreuz_o_Einmplang__hat_Achse
Laengsschnitt__hat_kreuzende_Bau_o_baul_An1	Laengsschn__hat_kreuz_Bauwerke
Laengsschnitt__hat_Gelaendehorizonte	Laengsschn__hat_Gelaendehoriz
kreuzende_Bauwerke_o_baul_An1__hat_Schnittgeometrie	kreuz_Bauwerk__hat_Schnittgeom
Schnittgeometrie__hat_Schnittpolygone	Schnittgeom__hat_Schnpolygone
Schnittgeometrie__hat_Mindestabstandspolygone	Schnittgeom__hat_Minabstpolyg
Schnittgeometrie__hat_Maximalabstandspolygone	Schnittgeom__hat_Maxabstpolyg
Laengsschnittlinie__hat_LS_Koor	Laengsschnittl__hat_LS_Koor
Trassenkoerper__hat_Oberflaeche	Trassenkoerper__hat_Oberfl
Trassenkoerper__hat_Ausgangsdaten	Trassenkoerper__hat_Ausgdaten
Trassenkoerper__hat_Bildungsgesetze	Trassenkoerper__hat_Bildgsges
Querprofil_von_Ausgangsdaten_Bildungsges	Querpr__von_Ausgdaten_Bildges
Ausgangsdaten__hat_Achsstationswerte	Ausgangsdat__hat_Achsstationsw
Deckenbuch__hat_Spur_aus_Ausgangsdaten	Deckenb__hat_Spur_a_Ausgangsdat
Deckenbuch__hat_Spur_aus_Querprofilen	Deckenb__hat_Spur_aus_Querprof
Spur_aus_Ausgangsdaten__hat_Hoehenzuege	Spur_a_Ausgdaten__hat_Hoehenz
BR_Punktfolge__hat_Aufweitg_Verbreit_Verbind	BR_Punktfl__hat_Aufw_Verbr_Verb
Entwurfparameter__hat_Geschwindigkeitsband	Entwurfsparam__hat_Geschwband
Geschwindigkeitsband__hat_V_Koor	Geschwindigkeitsb__hat_V_Koor
Verkehrsstaerkekennwerte__sind_Verkehrsstaerken	Verkstaerkekew__sind_Verkstaerk
Verkehrsstaerkekennwerte__hat_zeitliche_Zuordnung	Verkstaerkekew__hat_zeitl_Zuord
Verkehrsstaerken__hat_Verkehrsstaerkeverteilung	Verkstaerk__hat_Verkstarkevert
Verkehrsstaerkeverteilung__zugehoerige_FZ_Klasse_VS	Verkstaerkevert__zug_Fz_Kl_VS
Verkehrsstaerkeverteilung__Verkehrsstaerkekewert	Verkstaerkevert__Verkstaerkekew
Achslastdatenerfassung__hat_GG_Verteilung_e_FZ_Klasse	Achsldaterf__hat_GG_Vert_Fz_Kl
Achslastdatenerfassung__hat_Achslastvertlg_e_Achstyps	Achsldaterf__hat_Achslvrt_Atyp
Achslastdatenerfassung__hat_abgel_Werte_und_Vertlgen	Achsldaterf__hat_abgel_W_u_V
GG_Verteilung_einer_FZ_Klasse__hat_GG_Klasse	GG_Vert_Fz_Kl__hat_GG_Klasse
Einzelfahrzeugdatenerfassung__hat_achsbezogene_Daten	Einzfzdaterf__hat_achsbez_Dat
stat_wegweisende_Beschilderung__hat_Wegweiserbild	stat_wegw_Besch__hat_Wegwbild
stat_wegweisende_Beschilderung__hat_Wegweisertafel_m_Epunkten	stat_wegw_Besch__hat_Wegwtafel
Standort_Wegweiser__technische_Ausfuehrung	Standort_Wegw__techn_Ausfuehrg
Wegweisertafel_mit_Epunkten__hat_Beschriftungselem_Symbole	Wegwtafel__hat_Beschrelem_Symb
stat_verkehrsreg_Beschilderung__hat_Gueltigkeitsbereich_VRB	stat_verkreg_Besch__hat_Gueltb
Standort_Beschilderung__technische_Ausfuehrung	Standort_Besch__techn_Ausfuehr
Standort_Beschilderung__hat_Numerierung	Standort_Besch__hat_Numerierg
Standort_Beschilderung__hat_verkehrsregelnde_Funktion	Standort_Besch__hat_verkreg_F
Gueltigkeitsbereich_VRB__verkehrsartabaengiger_GB	Gueltkber_VRB__verkartabh_GB

# OKSTRA – Teilprojekt 4 – Schlußbericht

## Teilbericht D - Das SQL-Datenschema

ALT	NEU
Gueltigkeitsbereich_VRB__ist_raeumlicher_GB	Gueltkber_VRB__ist_rauempl_GB
Gueltigkeitsbereich_VRB__ist_zeitlicher_GB	Gueltkber_VRB__ist_zeitl_GB
Umfeldmessstelle__Messeinricht_Schadstoffimmiss	Umfmesst__Messeinr_Schadstimm
Umfeldmessstelle__hat_Messdaten	Umfmesst__hat_Messdaten
Umfeldmessstelle__hat_Detektoren_Umfelddatenerfg	Umfmesst__hat_Det_Umfdatenerf
Messdaten__hat_Umfang_Umfelddaten	Messdaten__hat_Umfang_Umfdaten
Messdaten__hat_Zaehlintervallaenge	Messdaten__hat_Zaehlintervlge
Detektoren_Umfelddatenerfassg__zugehoerige_EAKs	Detekt_Umfdaterf__zugeh_EAKs
Detektoren_Umfelddatenerfassg__Einsatzzeit	Detekt_Umfdaterf__Einsatzzeit
Detektoren_Umfelddatenerfassg__Betriebszeit	Detekt_Umfdaterf__Betriebszeit
Umfang_Umfelddaten__erfasste_Daten	Umf_Umfelddat__erfasste_Daten
Umfang_Umfelddaten__Ermittlung_Schadstoffimmiss	Umf_Umfelddat__Erm_Schadstimm
Umfang_Umfelddaten__hat_Witterungsdaten	Umf_Umfelddat__hat_Wittergsdat
Umfang_Umfelddaten__hat_Umfelddaten	Umf_Umfelddat__hat_Umfelddaten
Umfang_Umfelddaten__hat_witterungsbed_Strassenzust	Umf_Umfelddat__hat_witb_Strzst
dyn_verkehrsreg_Beschilderung__in_Kombination_v_Anlagentypen	dyn_vreg_Besch__in_Komb_v_Anlt
dyn_verkehrsreg_Beschilderung__hat_verwaltungstechn_Zuordnung	dyn_vreg_Besch__hat_verw_Zuord
dyn_verkehrsreg_Beschilderung__hat_Anlagenausstattung	dyn_vreg_Besch__hat_Anlagausst
verwaltungstechn_Zuordnung_dyn__hat_Verwaltungsbezirk	verwtechn_Zuord_dyn__hat_Vwbez
verwaltungstechn_Zuordnung_dyn__hat_Strassenbaudienststelle	verwtechn_Zuord_dyn__hat_Strbd
Streckenbeeinflussung__hat_Anlagentyp	Streckbeeinfl__hat_Anlagentyp
Streckenbeeinflussung__hat_Anzeigezustaende	Streckbeeinfl__hat_Anzeigzust
Streckenbeeinflussung__hat_Wirkungsbereiche	Streckbeeinfl__hat_Wirkbereich
SB_Wirkungsbereiche__beginnt_bei_Strassenpunkt	SB_Wirkber__beginnt_b_Strpunkt
SB_Wirkungsbereiche__endet_bei_Strassenpunkt	SB_Wirkber__endet_b_Strpunkt
Wechselwegweisung__hat_Anzeigezustaende	Wechselwegw__hat_Anzzustaende
Wechselwegweisung__hat_Wirkungsbereiche	Wechselwegw__hat_Wirkungsber
WW_Wirkungsbereiche__betroffene_BAB	WW_Wirkungsber__betroffene_BAB
WW_Wirkungsbereiche__beginnt_bei_Netzknoden	WW_Wirkungsber__beginnt_bei_NK
WW_Wirkungsbereiche__endet_bei_Netzknoden	WW_Wirkungsber__endet_bei_NK
Knotenpunktbeeinflussung__hat_Anlagentyp	Knotpktbeeinfl__hat_Anlagentyp
Knotenpunktbeeinflussung__hat_Anzeigezustaende	Knotpktbeeinfl__hat_Anzeigzust
KB_Wirkungsbereich__ist_richtungsbezogener_WB	KB_Wirkber__ist_richtbez_WB
KB_Wirkungsbereich__ist_astbezogener_WB	KB_Wirkber__ist_astbezog_WB
KB_Wirkungsbereich__hat_Angaben_zum_Knotenpunkt	KB_Wirkber__hat_Ang_z_Knotpkt
Richtungswechselbetrieb__hat_Betriebsform	Richtwechsbe__hat_Betrform
Richtungswechselbetrieb__hat_Anzeigezustaende	Richtwechsbe__hat_Anzeigzust
Richtungswechselbetrieb__hat_RW_Wirkungsbereich	Richtwechsbe__hat_RW_Wirkber
RW_Wirkungsbereich__beginnt_bei_Strassenpunkt	RW_Wirkber__beginnt_bei_Strpkt
RW_Wirkungsbereich__endet_bei_Strassenpunkt	RW_Wirkber__endet_bei_Strpkt
Anlagenausstattung__hat_Art_der_Stromversorgung	Anlausst__hat_Art_d_Stromvers
Anlagenausstattung__hat_Steuerung	Anlausst__hat_Steuerung
Anlagenausstattung__hat_Ueberwachung_der_Anlage	Anlausst__hat_Ueberwach_d_Anl
Anlagenausstattung__hat_Datenebertragungssystem	Anlausst__hat_Datenebertrrsyst
Anlagenausstattung__hat_Unterzentrale	Anlausst__hat_Unterzentrale
Anlagenausstattung__hat_Verkehrsrechnerzentrale	Anlagenausstattung__hat_VRZ
Anlagenausstattung__hat_Datenerfassungseinrichtgen	Anlausst__hat_Datenerfeinr
Steuerung__automatische_Steuerung	Steuerung__autom_Steuerung
Steuerung__hat_manuelle_Steuerung	Steuerung__hat_manu_Steuerung
Anzeigesystem__hat_Gueltigkeitsbereich	Anzsystem__hat_Gueltigkbereich
Anzeigesystem__hat>Wechselverkehrszeichen	Anzsys__hat>Wechselverkzeich
Anzeigesystem__hat_Aufstellvorrichtung	Anzsys__hat_Aufstellvorrichtg
Anzeigesystem__hat_Anzahl_der_Anzeigequerschn	Anzsys__hat_Anz_d_Anzquerschn
Wechselverkehrszeichen__Bauart_des_Zeichens	Wechselverkz__Bauart_d_Zeich
Aufstellvorrichtung__hat_Art_der_Aufstellung	Aufstellvorr__hat_Art_d_Aufst
Datenerfassungseinrichtungen__hat_Gueltigkeitsbereich	Datenerfeinr__hat_Gueltigkber
Datenerfassungseinrichtungen__hat_Detektionsmethode	Datenerfeinr__hat_Detektmeth
Datenerfassungseinrichtungen__hat_Anordnung_des_Messsystems	Datenerfeinr__hat_Anord_d_Msys
Datenerfassungseinrichtungen__hat_Art_der_erfassten_Daten	Datenerfeinr__hat_Art_d_erf_Dn
Datenerfassungseinrichtungen__hat_Anzahl_d_Messquerschnitte	Datenerfeinr__hat_Anz_d_Messqu

# OKSTRA – Teilprojekt 4 – Schlußbericht

## Teilbericht D - Das SQL-Datenschema

ALT	NEU
Anordnung_des_Messsystems__hat_Art_der_Anordnung	Anordn_d_Msys__hat_Art_d_Anord
Erfassung_Geschwindigkeiten__hat_V_Verteilungen	Erf_Geschw__hat_V_Verteilungen
Erfassung_Geschwindigkeiten__hat_aggregierte_Werte	Erf_Geschw__hat_aggreg_Werte
V_Verteilungen__hat_Anzahl_V_Klassen	V_Verteilung__hat_Anzahl_V_Kl
V_Klassen__hat_abgel_Werte_und_Verteilgen	V_Kl__hat_abgel_Wert_u_Verteil
mittlere_Geschwindigkeit__hat_abgeleitete_Werte	mittl_Geschw__hat_abgel_Werte
automatische_Dauerzaehlstelle__hat_Bezugszaehlstellen	autom_Dauzaehl__Bezugszaehlst
automatische_Dauerzaehlstelle__hat_Erfassungsergebnisse	autom_Dauzaehl__hat_Erfasserg
automatische_Dauerzaehlstelle__hat_Erfassungscharakteristik	autom_Dauzaehl__hat_Erfasschar
automatische_Dauerzaehlstelle__hat_Detektoren	autom_Dauzaehl__hat_Detektoren
automatische_Dauerzaehlstelle__gehoeert_zu_UZ_VRZ	autom_Dauzaehl__geh_zu_ZU_VRZ
Erfassungscharakteristik__Zaehlintervallaenge	Erfcharakt__Zaehlintervlaenge
Erfassungscharakteristik__hat_Art_d_registrierten_Daten	Erfcharakt__hat_Art_d_reg_Dat
Fahrzeugartenunterscheidung__hat_Klassenunterscheidung	Fzartuntersch__hat_Kluntersch
Differenzierung_der_Erfassung__hat_Fahrstreifenzuordnung	Diff_d_Erf__hat_Fahrstrzuordn
Detektoren__zugehoerige_Auswerteschaltung	Detekt__zugeh_Auswerteschalt
Detektoren__hat_Art_der_Registrierung	Detekt__hat_Art_der_Registrung
Moegl_Einzelfahrzeuergfassung__Einzelfahrzeugdaten	Moegl_Einzfzferf__Einzfzdaten
oertliche_Zuordnung_AD__hat_Richtungen	oertl_Zuord_AD__hat_Richtungen
oertliche_Zuordnung_AD__hat_Fahrstreifenzuteilung	oertl_Zuord_AD__hat_Fahrstrzut
Unterzentrale_Verkehrsrechnerz__Name	Unterzentr_Verkrechnzent__Name
Angaben_zum_Unfallort__hat_Angaben_z_einmuend_Strasse	Ang_z_Unfort__hat_Ang_z_einm_S
Unfallumstaende__Witterungsverhaeltnisse	Unfumstaende__Witterverhaeltn
Unfallumstaende__Strassenzustand	Unfumstaende__Strassenzustand
Unfallumstaende__hat_Charakterist_der_Unfallst	Unfumstaende__hat_Char_d_Unfst
Unfallumstaende__hat_Besonderheiten_d_Unfallst	Unfumstde__hat_Besond_d_Unfst
Beteiligte__hat_weitergehende_Angaben	Beteil__hat_weitergeh_Angaben
Mitfahrerangaben__hat_Angaben_z_Alter_und_Geschl	Mitfang__hat_Ang_z_Alt_u_Gesch
manuelle_Zaehlstelle__hat_Erfassung_Gesamtverkehr	manu_Zaehlst__hat_Erf_Gesverk
manuelle_Zaehlstelle__hat_Hochrechnungsergebnisse	manu_Zaehlst__hat_Hochrechnerg
manuelle_Zaehlstelle__hat_Bezugszaehlstelle_Auslverk	manu_Zaehlst__hat_Bezzt_Auslv
manuelle_Zaehlstelle__hat_Bezugszaehlstelle_Gesamtvk	manu_Zaehlst__hat_Bezzt_Gesvk
Erfassung_Auslaenderverkehr__hat_Gueltingkeitsbereich	Erfass_Auslverk__hat_Gueltbber
verwaltungstechn_Zuordnung_MZ__hat_Verwaltungskennziffern	verwtechn_Zuord_MZ__hat_Verwkwz
Hochrechnungsergebnisse__Fahrzeuggruppen	Hochrechnerg__Fahrzeuggruppen
Hochrechnungsergebnisse__Datenbeschreibung	Hochrechnerg__Datenbeschreib
Charakteristik__hat_Angaben_z_Strassenquerschn	Charakt__hat_Ang_z_Strquerschn
Charakteristik__hat_Richtungskennung	Charakt__hat_Richtungskennung
Charakteristik__hat_Bemerkungskennziffern	Charakt__hat_Bemerkgskennziff
oertliche_Zuordnung_MZ__hat_benachbarte_Anchlstellen	oertl_Zuord_Mz__hat_ben_Anchl
Lichtsignalanlage__hat_Rotlichtueberwachung	Lichtsignanl__hat_Rotlueberw
Lichtsignalanlage__hat_Angaben_zum_Knotenpunkt	Lichtsignanl__hat_Ang_z_Knotpkt
Lichtsignalanlage__hat_Grundl_u_Inbetriebn_daten	Lsiganl__hat_Grundl_u_Inbn_Dat
Lichtsignalanlage__hat_Angaben_zu_Verkehrsdaten	Lichtsignanl__hat_Ang_z_Verkdat
Lichtsignalanlage__hat_Softwarekomponenten	Lichtsignanl__hat_Softwarekomp
Lichtsignalanlage__hat_Hardwarekomponenten	Lichtsignanl__hat_Hardwarekomp
Lichtsignalanlage__hat_uebergeordnete_Zentrale	Lichtsignanl__hat_ueberg_Zentr
Angaben_zum_Knotenpunkt_LSA__Markierungspfeile	Ang_z_Knotpkt_LSA__Markpfeile
Grundlage_und_Inbetriebn_daten__Nummer_Nachbar_LSA	Grndl_u_Inbn_dat__Nr_Nachb_LSA
Angaben_zu_Verkehrsdaten__Verkehrsstroeme	Ang_z_Verkdat__Verkehrsstroeme
Softwarekomponenten__hat_Signalprogrammparameter	Softwkomp__hat_Signprogrparam
Softwarekomponenten__hat_hinterlegte_Programme	Softwkomp__hat_hinterl_Progr
Hardwarekomponenten__hat_akust_o_taktile_Signalgeb	Hardwkomp__hat_ak_o_takt_Signg
Hardwarekomponenten__hat_Angaben_zum_Steuergeraet	Hardwkomp__hat_Ang_z_Steuerger
Hardwarekomponenten__hat_Angaben_zur_Verkabelung	Hardwkomp__hat_Ang_z_Verkabelg
Hardwarekomponenten__hat_Angab_z_Aufstellvorrictg	Hardwkomp__hat_Ang_z_Aufstvvorr
Hardwarekomponenten__hat_optische_Signalgeber	Hardwkomp__hat_opt_Signalgeber
Hardwarekomponenten__hat_Erfassungseinrichtungen_IV	Hardwkomp__hat_Erfasseinr_IV
Hardwarekomponenten__hat_Erfassungseinrichtungn_OeV	Hardwkomp__hat_Erfasseinr_OeV



## 2 Fachbereich „Netzdaten“

### 2.1 Teilschema „Straßennetz“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Straßennetz“ inklusive der „GDF-Objekte“ und „Entwurfselemente“.

```
CREATE SCHEMA Strassennetz

CREATE TABLE Strassenklasse (
    Kennung
        CHAR(1) NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Strassenklasse VALUES ('A','Bundesautobahn')
INSERT INTO Strassenklasse VALUES ('B','Bundesstraße')
INSERT INTO Strassenklasse VALUES ('L','Landesstraße')
INSERT INTO Strassenklasse VALUES ('S','Staatsstraße')
INSERT INTO Strassenklasse VALUES ('K','Kreisstraße')
INSERT INTO Strassenklasse VALUES ('G','Gemeindestraße')

CREATE TABLE Strasse (
    Strasse_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Strasse_ID) REFERENCES Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Strassenklasse
        CHAR(1) NOT NULL REFERENCES Strassenklasse,
    Strassennummer
        INTEGER NOT NULL, -- Strassennummer
    CHECK ( ( Strassennummer > 0 ) AND ( Strassennummer <= 9999 ) ),
    Zusatzbuchstabe
        CHAR(1), -- Zusatzbuchstabe
    CHECK ( ( Zusatzbuchstabe >= 'A' ) AND ( Zusatzbuchstabe <= 'Z' ) )
)

CREATE TABLE Strasse__hat_Strassenknoten (
    Strasse
        INTEGER NOT NULL REFERENCES Strasse,
    Strasse_Netzknoten
        INTEGER NOT NULL REFERENCES Strasse_Netzknoten
)

CREATE TABLE Strasse__hat_AoA (
    Strasse
        INTEGER NOT NULL REFERENCES Strasse,
    Abschnitt_oder_Ast
        INTEGER NOT NULL REFERENCES Abschnitt_oder_Ast
)

CREATE TABLE Strasse__hat_BAB_Knotennummer (
    Strasse
        INTEGER NOT NULL REFERENCES Strasse,
    BAB_Knotennummer
        INTEGER NOT NULL REFERENCES BAB_Knotennummer
)

CREATE TABLE Strasse__in_Verwaltungsbezirk (
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Strasse
    INTEGER NOT NULL REFERENCES Strasse,
Verwaltungsbezirk
    INTEGER NOT NULL REFERENCES Administration.Verwaltungsbezirk
)

CREATE TABLE Verlaufskennzeichen (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Verlaufskennzeichen VALUES (1,'Anfang Straße')
INSERT INTO Verlaufskennzeichen VALUES (2,'Ende Straße')
INSERT INTO Verlaufskennzeichen VALUES (3,'Anfang Seitenarm')
INSERT INTO Verlaufskennzeichen VALUES (4,'Ende Seitenarm')
INSERT INTO Verlaufskennzeichen VALUES (5,'Anfang Unterbrechung')
INSERT INTO Verlaufskennzeichen VALUES (6,'Ende Unterbrechung')
INSERT INTO Verlaufskennzeichen VALUES (7,'Anfang getr. verl. Fahrbahn')
INSERT INTO Verlaufskennzeichen VALUES (8,'Ende getr. verl. Fahrbahn')
INSERT INTO Verlaufskennzeichen VALUES (9,'Wiederholung')

CREATE TABLE Strasse_Netzknoten (
    Strasse_Netzknoten_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Strasse_Netzknoten_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Verlaufskennzeichen
        INTEGER REFERENCES Verlaufskennzeichen
)

CREATE TABLE Strasse_NK__hat_Netzknoten (
    Strasse_Netzknoten
        INTEGER NOT NULL REFERENCES Strasse_Netzknoten,
    Netzknoten
        INTEGER NOT NULL REFERENCES Netzknoten
)

CREATE TABLE Strasse_NK__hat_Vorgaenger (
    Strasse_Netzknoten
        INTEGER NOT NULL REFERENCES Strasse_Netzknoten,
    Strasse_Netzknoten_2
        INTEGER NOT NULL REFERENCES Strasse_Netzknoten
)

CREATE TABLE Knotenart (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Knotenart VALUES (1,'plangleich (höhengleich)')
INSERT INTO Knotenart VALUES (2,'planfrei (höhenungleich)')
INSERT INTO Knotenart VALUES (3,'teilplanfrei')
INSERT INTO Knotenart VALUES (0,'fiktiver Netzknoten')

CREATE TABLE Knotenpunktsform (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Knotenpunktsform VALUES (01,'Einmündung oder Kreuzung von ' ||
'2-streifigen Straßen')
INSERT INTO Knotenpunktsform VALUES (02,'Einmündung oder Kreuzung von ' ||
'2-bahnigen mit 2-streifigen Straßen in der Regel mit Lichtsignalanlage')
INSERT INTO Knotenpunktsform VALUES (03,'Einmündung oder Kreuzung von zwei ' ||
'2-bahnigen Straßen mit Lichtsignalanlage')
INSERT INTO Knotenpunktsform VALUES (04,'planfreie Kreuzung von ' ||
'2-streifigen oder 2-bahnigen Straßen')
INSERT INTO Knotenpunktsform VALUES (05,'Kreuzung 2-streifiger Straßen ' ||
'als Versatz')
INSERT INTO Knotenpunktsform VALUES (06,'Aufgeweitete Einmündung oder ' ||
'Kreuzung mit mindestens einer 2-bahnigen Straße')
INSERT INTO Knotenpunktsform VALUES (07,'Kreisverkehrsplatz, d < 30 m')
INSERT INTO Knotenpunktsform VALUES (08,'Kreisverkehr, d > 30 m')
```

```
CREATE TABLE Knotenpunktsystem (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)
```

```
INSERT INTO Knotenpunktsystem VALUES (01,'Trompete')
INSERT INTO Knotenpunktsystem VALUES (02,'Birne')
INSERT INTO Knotenpunktsystem VALUES (03,'Dreieck mit 3 zweigeschossigen ' ||
'Bauwerken')
INSERT INTO Knotenpunktsystem VALUES (04,'Dreieck mit 1 dreigeschossigen ' ||
'Bauwerk')
INSERT INTO Knotenpunktsystem VALUES (05,'Kleeblatt')
INSERT INTO Knotenpunktsystem VALUES (06,'Kleeblatt, abgewandelt mit ' ||
'halbdirekter Führung eines Linksabbiegestromes')
INSERT INTO Knotenpunktsystem VALUES (07,'Kleeblatt, abgewandelt mit ' ||
'zügiger halbdirekter Führung eines Linksabbiegestromes')
INSERT INTO Knotenpunktsystem VALUES (08,'halbes Kleeblatt')
INSERT INTO Knotenpunktsystem VALUES (09,'halbes Kleeblatt, asymmetrisch ' ||
'mit außenliegenden Linksabbiegespuren')
INSERT INTO Knotenpunktsystem VALUES (10,'halbes Kleeblatt, asymmetrisch ' ||
'mit innen- und hintereinanderliegenden Linksabbiegespuren')
INSERT INTO Knotenpunktsystem VALUES (11,'halbes Kleeblatt, asymmetrisch ' ||
'mit innen- und nebeneinanderliegenden Linksabbiegespuren')
INSERT INTO Knotenpunktsystem VALUES (12,'halbes Kleeblatt, symmetrisch')
INSERT INTO Knotenpunktsystem VALUES (13,'Raute mit innen- und ' ||
'hintereinanderliegenden Linksabbiegespuren')
INSERT INTO Knotenpunktsystem VALUES (14,'Raute mit innen- und ' ||
'nebeneinanderliegenden Linksabbiegespuren')
INSERT INTO Knotenpunktsystem VALUES (15,'Raute mit aufgeweiteter Kreuzung')
INSERT INTO Knotenpunktsystem VALUES (16,'Raute mit außenliegenden ' ||
'Linksabbiegespuren')
INSERT INTO Knotenpunktsystem VALUES (17,'Sondersystem')
```

```
CREATE TABLE Netzknoten (
  Netzknoten_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Netzknoten_ID) REFERENCES Geometrieschema.Punktobjekt_Modell,
  FOREIGN KEY (Netzknoten_ID) REFERENCES Historisierung.historisches_Objekt,
  Netzknoten_ext
    CHAR(7),
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Numerierungsbezirk
    INTEGER NOT NULL, -- TK25_Blattnummer
  CHECK ( ( Numerierungsbezirk >= 1 ) AND ( Numerierungsbezirk <= 9999 ) ),
  Nummer
    INTEGER NOT NULL, -- lfd_NK_Nummer
  CHECK ( ( Nummer >= 1 ) AND ( Nummer <= 999 ) ),
  Knotenart
    INTEGER REFERENCES Knotenart,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Knotenname
    VARCHAR(30),
Knotenpunktsform
    INTEGER REFERENCES Knotenpunktsform,
Knotenpunktsystem
    INTEGER REFERENCES Knotenpunktsystem
)

CREATE TABLE NK__hat_BAB_Knotennummer (
    Netzknoten
        INTEGER NOT NULL REFERENCES Netzknoten,
    BAB_Knotennummer
        INTEGER NOT NULL REFERENCES BAB_Knotennummer
)

CREATE TABLE BAB_Knotennummer (
    BAB_Knotennummer_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (BAB_Knotennummer_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Knotennummer
        INTEGER NOT NULL,
    Buchstabe
        CHAR(1)
)

CREATE TABLE Stadium (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Stadium VALUES (00,'unter Verkehr')
INSERT INTO Stadium VALUES (11,'Neubau in Planung')
INSERT INTO Stadium VALUES (12,'Neubau in Ausführung')
INSERT INTO Stadium VALUES (20,'Erneuerung/Ersatzneubau in Planung')
INSERT INTO Stadium VALUES (21,'Erneuerung/Ersatzneubau in Ausführung')
INSERT INTO Stadium VALUES (30,'Instandsetzung in Planung')
INSERT INTO Stadium VALUES (31,'Instandsetzung in Ausführung')

CREATE TABLE Abschnitt_oder_Ast (
    Abschnitt_oder_Ast_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Abschnitt_oder_Ast_ID) REFERENCES
Geometrieschema.Linienobjekt_Modell,
    FOREIGN KEY (Abschnitt_oder_Ast_ID) REFERENCES
Historisierung.historisches_Objekt,
    Abschnitt_oder_Ast_ext
        CHAR(16),
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Art_Abschnitt_oder_Ast
        INTEGER NOT NULL REFERENCES Art_Abschnitt_oder_Ast,
    Laenge
        REAL, -- Meter
    CHECK ( Laenge>=0 ),
    Stadium
        INTEGER REFERENCES Stadium,
    Streckenart
        INTEGER,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Verkehrsfreigabe
  CHAR(10), -- Datum
CHECK ( Verkehrsfreigabe LIKE '___.__.____' ),
Abschnittsnummer
  INTEGER -- Abschnitt
)

CREATE TABLE Art_Abschnitt_oder_Ast (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(9) NOT NULL
)

INSERT INTO Art_Abschnitt_oder_Ast VALUES (1,'Abschnitt')
INSERT INTO Art_Abschnitt_oder_Ast VALUES (2,'Ast')

CREATE TABLE AoA__beginnt_bei_Nullpunkt (
  Abschnitt_oder_Ast
    INTEGER NOT NULL REFERENCES Abschnitt_oder_Ast,
  Nullpunkt
    INTEGER NOT NULL REFERENCES Nullpunkt
)

CREATE TABLE AoA__endet_bei_Nullpunkt (
  Abschnitt_oder_Ast
    INTEGER NOT NULL REFERENCES Abschnitt_oder_Ast,
  Nullpunkt
    INTEGER NOT NULL REFERENCES Nullpunkt
)

CREATE TABLE Verkehrliche_Verknuepfung (
  Verkehrliche_Verknuepfung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Verkehrliche_Verknuepfung_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_bis LIKE '___.__.____' )
)

CREATE TABLE VV__von_Strassenelement (
  Verkehrliche_Verknuepfung
    INTEGER NOT NULL REFERENCES Verkehrliche_Verknuepfung,
  Strassenelement
    INTEGER NOT NULL REFERENCES Strassenelement
)

CREATE TABLE VV__nach_Strassenelement (
  Verkehrliche_Verknuepfung
    INTEGER NOT NULL REFERENCES Verkehrliche_Verknuepfung,
  Strassenelement
    INTEGER NOT NULL REFERENCES Strassenelement
)

CREATE TABLE VV__ueber_Nullpunkt (
  Verkehrliche_Verknuepfung
    INTEGER NOT NULL REFERENCES Verkehrliche_Verknuepfung,
  Nullpunkt
    INTEGER NOT NULL REFERENCES Nullpunkt
)

CREATE TABLE Nullpunkt (
  Nullpunkt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Nullpunkt_ID) REFERENCES Geometrieschema.Punktobjekt_Modell,
  FOREIGN KEY (Nullpunkt_ID) REFERENCES Historisierung.historisches_Objekt,
  FOREIGN KEY (Nullpunkt_ID) REFERENCES verallgemeinerter_Nullpunkt,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Nullpunkt_ext
  CHAR(8),
gueltig_von
  CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_von LIKE '___.__.____' ),
gueltig_bis
  CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_bis LIKE '___.__.____' ),
Zusatz
  CHAR(1) NOT NULL, -- Nullpunktkenung
CHECK ( ( ( Zusatz >= 'A' AND Zusatz <= 'Z' ) OR ( Zusatz = ' ' ) ) ),
zentraler_Nullpunkt
  INTEGER REFERENCES Logical_Types.Boolean
)

CREATE TABLE Nullpunkt__in_Netzknoten (
  Nullpunkt
    INTEGER NOT NULL REFERENCES Nullpunkt,
  Netzknoten
    INTEGER NOT NULL REFERENCES Netzknoten
)

CREATE TABLE Punktobjekt (
  Punktobjekt_ID
    INTEGER NOT NULL PRIMARY KEY,
  Station
    REAL NOT NULL, -- Meter
CHECK ( Station >= 0 ),
auf_Abschnitt_oder_Ast
  INTEGER NOT NULL REFERENCES Abschnitt_oder_Ast
)

CREATE TABLE Punktobjekt_stat (
  Punktobjekt_stat_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Punktobjekt_stat_ID) REFERENCES Punktobjekt,
  Art_Punktobjekt_stat
    INTEGER NOT NULL REFERENCES Art_Punktobjekt_stat
)

CREATE TABLE Art_Punktobjekt_stat (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(30) NOT NULL
)

INSERT INTO Art_Punktobjekt_stat VALUES (1, 'GDF_Verbindungspunkt')
INSERT INTO Art_Punktobjekt_stat VALUES (2, 'Achselementpunkt')
INSERT INTO Art_Punktobjekt_stat VALUES (3, 'Querprofilpunkt')
INSERT INTO Art_Punktobjekt_stat VALUES (4, 'Wirkungsbereich_SP')
INSERT INTO Art_Punktobjekt_stat VALUES (5, 'Anordnung_des_Messsystems')
INSERT INTO Art_Punktobjekt_stat VALUES (6, 'stat_wegweisende_Beschilderung')
INSERT INTO Art_Punktobjekt_stat VALUES (7, 'Position')
INSERT INTO Art_Punktobjekt_stat VALUES (8, 'stat_verkehrsreg_Beschilderung')
INSERT INTO Art_Punktobjekt_stat VALUES (9, 'oertliche_Zuordnung_SP')

CREATE TABLE Punktobjekt_hist (
  Punktobjekt_hist_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Punktobjekt_hist_ID) REFERENCES Punktobjekt,
  FOREIGN KEY (Punktobjekt_hist_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_bis LIKE '___.__.____' ),
  Art_Punktobjekt_hist
    INTEGER NOT NULL REFERENCES Art_Punktobjekt_hist
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

CREATE TABLE Art_Punktobjekt_hist (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(22) NOT NULL
)

INSERT INTO Art_Punktobjekt_hist VALUES (1,'Kilometrierung')
INSERT INTO Art_Punktobjekt_hist VALUES (2,'Teilbauwerk')
INSERT INTO Art_Punktobjekt_hist VALUES (3,'Bruecke')
INSERT INTO Art_Punktobjekt_hist VALUES (4,'Verkehrszeichenbruecke')
INSERT INTO Art_Punktobjekt_hist VALUES (5,'Tunnel_Trogbauwerk')
INSERT INTO Art_Punktobjekt_hist VALUES (6,'Laermschutzbauwerk')
INSERT INTO Art_Punktobjekt_hist VALUES (7,'Stuetzbauwerk')
INSERT INTO Art_Punktobjekt_hist VALUES (8,'sonstiges_Bauwerk')
INSERT INTO Art_Punktobjekt_hist VALUES (9,'Schichtbegrenzung')
INSERT INTO Art_Punktobjekt_hist VALUES (10,'Streifenbegrenzung')
INSERT INTO Art_Punktobjekt_hist VALUES (11,'Gewaesserkreuzung')
INSERT INTO Art_Punktobjekt_hist VALUES (12,'Bahnkreuzung')
INSERT INTO Art_Punktobjekt_hist VALUES (13,'Kreuzung_Strasse_Weg')
INSERT INTO Art_Punktobjekt_hist VALUES (14,'Fussgaengerueberweg')
INSERT INTO Art_Punktobjekt_hist VALUES (15,'Durchlass')
INSERT INTO Art_Punktobjekt_hist VALUES (16,'Aufstellvorrichtung')

CREATE TABLE Nullpunktsort (
  Nullpunktsort_ID
    INTEGER NOT NULL PRIMARY KEY,
  Station
    REAL NOT NULL, -- Meter
  CHECK ( Station>=0 ),
  auf_Abschnitt_oder_Ast
    INTEGER NOT NULL REFERENCES Abschnitt_oder_Ast
)

CREATE TABLE Nullpunktsort__bei_Nullpunkt (
  Nullpunktsort
    INTEGER NOT NULL REFERENCES Nullpunktsort,
  Nullpunkt
    INTEGER NOT NULL REFERENCES Nullpunkt
)

CREATE TABLE Strassenpunkt_PoB (
  Strassenpunkt_PoB_ID
    INTEGER NOT NULL PRIMARY KEY,
  Station
    REAL NOT NULL, -- Meter
  CHECK ( Station>=0 ),
  auf_Abschnitt_oder_Ast
    INTEGER NOT NULL REFERENCES Abschnitt_oder_Ast
)

CREATE TABLE Punktobjekt_o_Bereichsobjekt (
  Punktobjekt_o_Bereichsobjekt_ID
    INTEGER NOT NULL PRIMARY KEY,
  ist_Punktobjekt
    INTEGER REFERENCES Strassenpunkt_PoB,
  ist_Bereichsobjekt
    INTEGER REFERENCES Netzbereich
)

CREATE TABLE Netzbereichskomponente (
  Netzbereichskomponente_ID
    INTEGER NOT NULL PRIMARY KEY,
  Art_Netzbereichskomponente
    INTEGER NOT NULL REFERENCES Art_Netzbereichskomponente
)

CREATE TABLE Art_Netzbereichskomponente (
  Kennung
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
    VARCHAR(16) NOT NULL
)

INSERT INTO Art_Netzbereichskomponente VALUES (1,'Netzbereich')
INSERT INTO Art_Netzbereichskomponente VALUES (2,'Teilabschnitt')
INSERT INTO Art_Netzbereichskomponente VALUES (3,'Strecke')

CREATE TABLE Netzbereich (
    Netzbereich_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Netzbereich_ID) REFERENCES Netzbereichskomponente,
    FOREIGN KEY (Netzbereich_ID) REFERENCES Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' )
)

CREATE TABLE Netzbrch__enthaelt_Komponenten (
    Netzbereich
        INTEGER NOT NULL REFERENCES Netzbereich,
    Netzbereichskomponente
        INTEGER REFERENCES Netzbereichskomponente,
    Station_1
        REAL, -- Meter
    CHECK ( Station_1>=0 ),
    auf_Abschnitt_oder_Ast
        INTEGER REFERENCES Abschnitt_oder_Ast,
    Station_2
        REAL, -- Meter
    CHECK ( Station_2>=0 )
)

CREATE TABLE Bereichsobjekt (
    Bereichsobjekt_ID
        INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Bereichsobj__hat_Netzbereich (
    Bereichsobjekt
        INTEGER NOT NULL REFERENCES Bereichsobjekt,
    Netzbereich
        INTEGER NOT NULL REFERENCES Netzbereich
)

CREATE TABLE Bereichsobjekt_stat (
    Bereichsobjekt_stat_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Bereichsobjekt_stat_ID) REFERENCES Bereichsobjekt,
    Art_Bereichsobjekt_stat
        INTEGER NOT NULL REFERENCES Art_Bereichsobjekt_stat
)

CREATE TABLE Art_Bereichsobjekt_stat (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(22) NOT NULL
)

INSERT INTO Art_Bereichsobjekt_stat VALUES (1,'Teilbauwerk')
INSERT INTO Art_Bereichsobjekt_stat VALUES (2,'Bruecke')
INSERT INTO Art_Bereichsobjekt_stat VALUES (3,'Verkehrszeichenbruecke')
INSERT INTO Art_Bereichsobjekt_stat VALUES (4,'Tunnel_Trogbauwerk')
INSERT INTO Art_Bereichsobjekt_stat VALUES (5,'Laermschutzbauwerk')
INSERT INTO Art_Bereichsobjekt_stat VALUES (6,'Stuetzbauwerk')
INSERT INTO Art_Bereichsobjekt_stat VALUES (7,'sonstiges_Bauwerk')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Bereichsobjekt_hist (
  Bereichsobjekt_hist_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Bereichsobjekt_hist_ID) REFERENCES Bereichsobjekt,
  FOREIGN KEY (Bereichsobjekt_hist_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Art_Bereichsobjekt_hist
    INTEGER NOT NULL REFERENCES Art_Bereichsobjekt_hist
)

CREATE TABLE Art_Bereichsobjekt_hist (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(30) NOT NULL
)

INSERT INTO Art_Bereichsobjekt_hist VALUES (1,'Bundesland')
INSERT INTO Art_Bereichsobjekt_hist VALUES (2,'Regierungsbezirk')
INSERT INTO Art_Bereichsobjekt_hist VALUES (3,'Kreis_kreisfreie_Stadt')
INSERT INTO Art_Bereichsobjekt_hist VALUES (4,'Gemeindebezirk')
INSERT INTO Art_Bereichsobjekt_hist VALUES (5,'Ortsteil')
INSERT INTO Art_Bereichsobjekt_hist VALUES (6,'Land_Ministerium')
INSERT INTO Art_Bereichsobjekt_hist VALUES (7,'Regierungspraesidium_Landesamt')
INSERT INTO Art_Bereichsobjekt_hist VALUES (8,'Amt')
INSERT INTO Art_Bereichsobjekt_hist VALUES (9,'Meisterei')
INSERT INTO Art_Bereichsobjekt_hist VALUES (10,'Kreisverwaltung')
INSERT INTO Art_Bereichsobjekt_hist VALUES (11,'Gemeindeverwaltung')
INSERT INTO Art_Bereichsobjekt_hist VALUES (12,'UI_Vereinbarung')
INSERT INTO Art_Bereichsobjekt_hist VALUES (13,'Baulast')
INSERT INTO Art_Bereichsobjekt_hist VALUES (14,'Schutzzone')

CREATE TABLE Streckenobjekt (
  Streckenobjekt_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Streckenobjekt__hat_Strecke (
  Streckenobjekt
    INTEGER NOT NULL REFERENCES Streckenobjekt,
  Strecke
    INTEGER NOT NULL REFERENCES Strecke
)

CREATE TABLE Streckenobjekt_stat (
  Streckenobjekt_stat_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Streckenobjekt_stat_ID) REFERENCES Streckenobjekt,
  Art_Streckenobjekt_stat
    INTEGER NOT NULL REFERENCES Art_Streckenobjekt_stat
)

CREATE TABLE Art_Streckenobjekt_stat (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(28) NOT NULL
)

INSERT INTO Art_Streckenobjekt_stat VALUES (1,'Entwurfselement_Kreisbogen')
INSERT INTO Art_Streckenobjekt_stat VALUES (2,'Entwurfselement_Klothoide')
INSERT INTO Art_Streckenobjekt_stat VALUES (3,'Entwurfselement_Gerade')
INSERT INTO Art_Streckenobjekt_stat VALUES (4,'ZEB_Objekt')
INSERT INTO Art_Streckenobjekt_stat VALUES (5,'Strassenentwaesserung')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Art_Streckenobjekt_stat VALUES (6, 'Anzahl_Fahrstreifen')
INSERT INTO Art_Streckenobjekt_stat VALUES (7, 'Insassen_min_Anzahl')
INSERT INTO Art_Streckenobjekt_stat VALUES (8, 'Ueberholverbot')
INSERT INTO Art_Streckenobjekt_stat VALUES (9, 'max_Hoehoe')
INSERT INTO Art_Streckenobjekt_stat VALUES (10, 'max_Breite')
INSERT INTO Art_Streckenobjekt_stat VALUES (11, 'max_Laenge')
INSERT INTO Art_Streckenobjekt_stat VALUES (12, 'max_Gewicht')
INSERT INTO Art_Streckenobjekt_stat VALUES (13, 'max_Achsgewicht')
INSERT INTO Art_Streckenobjekt_stat VALUES (14, 'max_Geschwindigkeit')
INSERT INTO Art_Streckenobjekt_stat VALUES (15, 'Zugangsbeschraenkung')
INSERT INTO Art_Streckenobjekt_stat VALUES (16, 'Gesperrt')
INSERT INTO Art_Streckenobjekt_stat VALUES (17, 'Fkt_d_Verb_im_Knotenpktber')
INSERT INTO Art_Streckenobjekt_stat VALUES (18, 'Bergpass')
INSERT INTO Art_Streckenobjekt_stat VALUES (19, 'Verkehrsbedeutung')
INSERT INTO Art_Streckenobjekt_stat VALUES (20, 'Spur_fuer_Rettungsfahrzeuge')
INSERT INTO Art_Streckenobjekt_stat VALUES (21, 'Strassenfunktion')
INSERT INTO Art_Streckenobjekt_stat VALUES (22, 'Aussichtswert')
INSERT INTO Art_Streckenobjekt_stat VALUES (23, 'Staugefahr')
INSERT INTO Art_Streckenobjekt_stat VALUES (24, 'gebuehrenpflichtig')
INSERT INTO Art_Streckenobjekt_stat VALUES (25, 'Durchschnittsgeschwindigkeit')
INSERT INTO Art_Streckenobjekt_stat VALUES (26, 'Fahrbahntrennung')
INSERT INTO Art_Streckenobjekt_stat VALUES (27, 'Trassenbreite')
INSERT INTO Art_Streckenobjekt_stat VALUES (28, 'Fahrbahnbreite')
INSERT INTO Art_Streckenobjekt_stat VALUES (29, 'Fussweg')
INSERT INTO Art_Streckenobjekt_stat VALUES (30, 'Radweg')
INSERT INTO Art_Streckenobjekt_stat VALUES (31, 'Erfassung_Gesamtverkehr')

CREATE TABLE Streckenobjekt_hist (
  Streckenobjekt_hist_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Streckenobjekt_hist_ID) REFERENCES Streckenobjekt,
  FOREIGN KEY (Streckenobjekt_hist_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Art_Streckenobjekt_hist
    INTEGER NOT NULL REFERENCES Art_Streckenobjekt_hist
)

CREATE TABLE Art_Streckenobjekt_hist (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(27) NOT NULL
)

INSERT INTO Art_Streckenobjekt_hist VALUES (1, 'Block')
INSERT INTO Art_Streckenobjekt_hist VALUES (2, 'OD_FS')
INSERT INTO Art_Streckenobjekt_hist VALUES (3, 'Widmung')
INSERT INTO Art_Streckenobjekt_hist VALUES (4, 'Leitung')
INSERT INTO Art_Streckenobjekt_hist VALUES (5, 'Markierung')
INSERT INTO Art_Streckenobjekt_hist VALUES (6, 'Schutzplanke')
INSERT INTO Art_Streckenobjekt_hist VALUES (7, 'Nebenanlage')
INSERT INTO Art_Streckenobjekt_hist VALUES (8, 'Rastanlage_nicht_bew')
INSERT INTO Art_Streckenobjekt_hist VALUES (9, 'Rastanlage_bew')
INSERT INTO Art_Streckenobjekt_hist VALUES (10, 'Nebenbetrieb')
INSERT INTO Art_Streckenobjekt_hist VALUES (11, 'Schicht')
INSERT INTO Art_Streckenobjekt_hist VALUES (12, 'Querschnittstreifen')
INSERT INTO Art_Streckenobjekt_hist VALUES (13, 'Bauklasse_Oberbau')
INSERT INTO Art_Streckenobjekt_hist VALUES (14, 'Bauweise_Oberbau')
INSERT INTO Art_Streckenobjekt_hist VALUES (15, 'Regelquerschnitt')
INSERT INTO Art_Streckenobjekt_hist VALUES (16, 'Bahnigkeit')
INSERT INTO Art_Streckenobjekt_hist VALUES (17, 'Fahrbahnlaengsneigung')
INSERT INTO Art_Streckenobjekt_hist VALUES (18, 'Fahrbahnquerneigung')
INSERT INTO Art_Streckenobjekt_hist VALUES (19, 'Hindernis')
INSERT INTO Art_Streckenobjekt_hist VALUES (20, 'SB_Wirkungsbereiche')
INSERT INTO Art_Streckenobjekt_hist VALUES (21, 'raeumlicher_Gueltigkeitsber')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Strecke (
  Strecke_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Strecke_ID) REFERENCES Netzbereichskomponente,
  FOREIGN KEY (Strecke_ID) REFERENCES Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' )
)

CREATE TABLE Strecke__entlang_Teilabschnitt (
  Strecke
    INTEGER NOT NULL REFERENCES Strecke,
  Position
    INTEGER NOT NULL,
  PRIMARY KEY ( Strecke , Position ),
  Station_1
    REAL NOT NULL, -- Meter
  CHECK ( Station_1 >= 0 ),
  auf_Abschnitt_oder_Ast
    INTEGER NOT NULL REFERENCES Abschnitt_oder_Ast,
  Station_2
    REAL NOT NULL, -- Meter
  CHECK ( Station_2 >= 0 )
)

CREATE TABLE Teilnetzkomponente (
  Teilnetzkomponente_ID
    INTEGER NOT NULL PRIMARY KEY,
  Art_Teilnetzkomponente
    INTEGER NOT NULL REFERENCES Art_Teilnetzkomponente
)

CREATE TABLE Art_Teilnetzkomponente (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(15) NOT NULL
)

INSERT INTO Art_Teilnetzkomponente VALUES (1,'Teilnetz')
INSERT INTO Art_Teilnetzkomponente VALUES (2,'Route')
INSERT INTO Art_Teilnetzkomponente VALUES (3,'Strassenelement')

CREATE TABLE Teilnetz (
  Teilnetz_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Teilnetz_ID) REFERENCES Teilnetzkomponente,
  FOREIGN KEY (Teilnetz_ID) REFERENCES Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Kennzeichen_Teilnetz
    VARCHAR(6),
  Beschreibung_Teilnetz
    VARCHAR(60)
)

CREATE TABLE Teilnetz__enthaelt_Komponente (
  Teilnetz
    INTEGER NOT NULL REFERENCES Teilnetz,
  Teilnetzkomponente
    INTEGER NOT NULL REFERENCES Teilnetzkomponente
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Routenkomponente (
  Routenkomponente_ID
    INTEGER NOT NULL PRIMARY KEY,
  Art_Routenkomponente
    INTEGER NOT NULL REFERENCES Art_Routenkomponente
)

CREATE TABLE Art_Routenkomponente (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(15) NOT NULL
)

INSERT INTO Art_Routenkomponente VALUES (1,'Route')
INSERT INTO Art_Routenkomponente VALUES (2,'Strassenelement')

CREATE TABLE Routenkomp__hat_Vorgaenger (
  Routenkomponente
    INTEGER NOT NULL REFERENCES Routenkomponente,
  Routenkomponente_2
    INTEGER NOT NULL REFERENCES Routenkomponente
)

CREATE TABLE Route (
  Route_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Route_ID) REFERENCES Teilnetzkomponente,
  FOREIGN KEY (Route_ID) REFERENCES Routenkomponente,
  FOREIGN KEY (Route_ID) REFERENCES Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' )
)

CREATE TABLE Route__entlang_Komponente (
  Route
    INTEGER NOT NULL REFERENCES Route,
  Routenkomponente
    INTEGER NOT NULL REFERENCES Routenkomponente
)

CREATE TABLE Strassenelement (
  Strassenelement_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Strassenelement_ID) REFERENCES Teilnetzkomponente,
  FOREIGN KEY (Strassenelement_ID) REFERENCES Routenkomponente,
  FOREIGN KEY (Strassenelement_ID) REFERENCES Historisierung.historisches_Objekt,
  FOREIGN KEY (Strassenelement_ID) REFERENCES
  Geometrieschema.Linienobjekt_Modell,
  Strassenelement_ext
    CHAR(16),
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  RDS_TMC_Code
    VARCHAR(12)
)

CREATE TABLE SE__beginnt_bei_Nullpunktsort (
  Strassenelement
    INTEGER NOT NULL REFERENCES Strassenelement,
  Nullpunktsort
    INTEGER NOT NULL REFERENCES Nullpunktsort
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

CREATE TABLE Strassenelement__endet_bei_NPO (
  Strassenelement
    INTEGER NOT NULL REFERENCES Strassenelement,
  Nullpunktsort
    INTEGER NOT NULL REFERENCES Nullpunktsort
)

CREATE TABLE Kilometrierung (
  Kilometrierung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Kilometrierung_ID) REFERENCES Punktobjekt_hist,
  Kilometrierung
    REAL NOT NULL, -- Kilometer
  CHECK ( Kilometrierung>=0 )
)

CREATE TABLE Betriebskilometer__in_Block (
  Betriebskilometer
    INTEGER NOT NULL REFERENCES Kilometrierung,
  Block
    INTEGER NOT NULL REFERENCES Block
)

CREATE TABLE Block (
  Block_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Block_ID) REFERENCES Streckenobjekt_hist,
  Blocknummer
    INTEGER NOT NULL,
  km_Anfang
    REAL NOT NULL,
  km_Ende
    REAL NOT NULL
)

CREATE TABLE Block__auf_Strasse (
  Block
    INTEGER NOT NULL REFERENCES Block,
  Strasse
    INTEGER NOT NULL REFERENCES Strasse
)

CREATE TABLE GDF_Verbindungspunkt (
  GDF_Verbindungspunkt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (GDF_Verbindungspunkt_ID) REFERENCES Punktobjekt_stat,
  FOREIGN KEY (GDF_Verbindungspunkt_ID) REFERENCES
Geometrieschema.Punktobjekt_Modell,
  FOREIGN KEY (GDF_Verbindungspunkt_ID) REFERENCES verallgemeinerter_Nullpunkt,
  entspricht_Nullpunkt
    INTEGER NOT NULL REFERENCES Nullpunkt
)

CREATE TABLE GDF_Verbindungselement (
  GDF_Verbindungselement_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (GDF_Verbindungselement_ID) REFERENCES
Geometrieschema.Linienobjekt_Modell,
  beginnt_bei_verallg_Nullpunkt
    INTEGER NOT NULL REFERENCES verallgemeinerter_Nullpunkt,
  endet_bei_verallg_Nullpunkt
    INTEGER NOT NULL REFERENCES verallgemeinerter_Nullpunkt,
  gehoert_zu_Nullpunkt
    INTEGER REFERENCES Nullpunkt,
  entspricht_Strassenelement
    INTEGER REFERENCES Strassenelement
)

CREATE TABLE verallgemeinerter_Nullpunkt (
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        verallgemeinerter_Nullpunkt_ID
        INTEGER NOT NULL PRIMARY KEY
    )

CREATE TABLE Entwurfselement (
    Entwurfselement_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Entwurfselement_ID) REFERENCES Streckenobjekt_stat,
    Richtungsw_Tangente_am_Anfang
        REAL, -- Winkel
    beginnt_bei_Vermessungspunkt
        INTEGER NOT NULL,
    endet_bei_Vermessungspunkt
        INTEGER NOT NULL
)

CREATE TABLE Entwurfselement_Kreisbogen (
    Entwurfselement_Kreisbogen_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Entwurfselement_Kreisbogen_ID) REFERENCES Entwurfselement,
    Radius
        REAL NOT NULL, -- Meter
    CHECK ( Radius >= 0 )
)

CREATE TABLE Entwurfselement_Klothoide (
    Entwurfselement_Klothoide_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Entwurfselement_Klothoide_ID) REFERENCES Entwurfselement,
    Radius_am_Anfang
        REAL NOT NULL, -- Meter
    CHECK ( Radius_am_Anfang >= 0 ),
    Radius_am_Ende
        REAL, -- Meter
    CHECK ( Radius_am_Ende >= 0 ),
    Parameter
        REAL NOT NULL, -- Meter
    CHECK ( Parameter >= 0 )
)

CREATE TABLE Entwurfselement_Gerade (
    Entwurfselement_Gerade_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Entwurfselement_Gerade_ID) REFERENCES Entwurfselement
)

```

## 2.2 Teilschema „Administration“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Administration“.

```
CREATE SCHEMA Administration

CREATE TABLE Verwaltungsbezirk (
    Verwaltungsbezirk_ID
        INTEGER NOT NULL PRIMARY KEY,
    Bezeichnung_Lang
        VARCHAR(29),
    Bezeichnung_Mittel
        VARCHAR(19),
    Bezeichnung_Kurz
        VARCHAR(11),
    FOREIGN KEY (Verwaltungsbezirk_ID) REFERENCES Strassennetz.Bereichsobjekt_hist,
    FOREIGN KEY (Verwaltungsbezirk_ID) REFERENCES
    Geometrieschema.Flaechenobjekt_Modell,
    Art_Verwaltungsbezirk
        INTEGER NOT NULL REFERENCES Art_Verwaltungsbezirk
)

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

CREATE TABLE Art_Verwaltungsbezirk (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(22) NOT NULL
)

INSERT INTO Art_Verwaltungsbezirk VALUES (1,'Bundesland')
INSERT INTO Art_Verwaltungsbezirk VALUES (2,'Regierungsbezirk')
INSERT INTO Art_Verwaltungsbezirk VALUES (3,'Kreis_kreisfreie_Stadt')
INSERT INTO Art_Verwaltungsbezirk VALUES (4,'Gemeindebezirk')
INSERT INTO Art_Verwaltungsbezirk VALUES (5,'Ortsteil')

CREATE TABLE Bundesland (
  Bundesland_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Bundesland_ID) REFERENCES Verwaltungsbezirk,
  Kennung_Bundesland
    CHAR(2) NOT NULL
)

CREATE TABLE Regierungsbezirk (
  Regierungsbezirk_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Regierungsbezirk_ID) REFERENCES Verwaltungsbezirk,
  Kennung_Regierungsbezirk
    CHAR(3) NOT NULL
)

CREATE TABLE RBZ__ist_in_Bundesland (
  Regierungsbezirk
    INTEGER NOT NULL REFERENCES Regierungsbezirk,
  Bundesland
    INTEGER NOT NULL REFERENCES Bundesland
)

CREATE TABLE Kreis_kreisfreie_Stadt (
  Kreis_kreisfreie_Stadt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Kreis_kreisfreie_Stadt_ID) REFERENCES Verwaltungsbezirk,
  Kennung_Kreis
    CHAR(5) NOT NULL,
  Kreisart
    CHAR(1) REFERENCES Kreisart
)

CREATE TABLE Kreis__ist_in_RBZ (
  Kreis_kreisfreie_Stadt
    INTEGER NOT NULL REFERENCES Kreis_kreisfreie_Stadt,
  Regierungsbezirk
    INTEGER NOT NULL REFERENCES Regierungsbezirk
)

CREATE TABLE Kreis__hat_Verwaltung (
  Kreis_kreisfreie_Stadt
    INTEGER NOT NULL REFERENCES Kreis_kreisfreie_Stadt,
  Kreisverwaltung
    INTEGER NOT NULL REFERENCES Kreisverwaltung
)

CREATE TABLE Kreisart (
  Kennung
    CHAR(1) NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Kreisart VALUES ('S','Stadtkreis')
INSERT INTO Kreisart VALUES ('L','Landkreis')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Gemeindebezirk (
  Gemeindebezirk_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Gemeindebezirk_ID) REFERENCES Verwaltungsbezirk,
  Kennung_Gemeinde
    CHAR(8) NOT NULL,
  Baulast_Strassenklasse
    CHAR(1) REFERENCES Baulast_Strassenklasse,
  Gemeinde_Funktion
    CHAR(3) REFERENCES Gemeinde_Funktion
)

CREATE TABLE Gemeinde__ist_in_Kreis (
  Gemeindebezirk
    INTEGER NOT NULL REFERENCES Gemeindebezirk,
  Kreis_kreisfreie_Stadt
    INTEGER NOT NULL REFERENCES Kreis_kreisfreie_Stadt
)

CREATE TABLE Gemeinde__hat_Verwaltung (
  Gemeindebezirk
    INTEGER NOT NULL REFERENCES Gemeindebezirk,
  Gemeindeverwaltung
    INTEGER NOT NULL REFERENCES Gemeindeverwaltung
)

CREATE TABLE Baulast_Strassenklasse (
  Kennung
    CHAR(1) NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Baulast_Strassenklasse VALUES ('B','in OD für B-, L- und ' ||
'Kreisstraßen')
INSERT INTO Baulast_Strassenklasse VALUES ('L','in OD für L- und Kreisstraßen')
INSERT INTO Baulast_Strassenklasse VALUES ('K','für Kreisstraßen')

CREATE TABLE Gemeinde_Funktion (
  Kennung
    CHAR(3) NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Gemeinde_Funktion VALUES ('O','Ober-Zentrum')
INSERT INTO Gemeinde_Funktion VALUES ('M','Mittel-Zentrum')
INSERT INTO Gemeinde_Funktion VALUES ('U/G','Unter-/Grund-Zentrum')

CREATE TABLE Ortsteil (
  Ortsteil_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Ortsteil_ID) REFERENCES Verwaltungsbezirk,
  Kennung_Ortsteil
    CHAR(10) NOT NULL
)

CREATE TABLE Ortsteil__ist_in_Gemeinde (
  Ortsteil
    INTEGER NOT NULL REFERENCES Ortsteil,
  Gemeindebezirk
    INTEGER NOT NULL REFERENCES Gemeindebezirk
)

CREATE TABLE Strassenbaudienststelle (
  Strassenbaudienststelle_ID
    INTEGER NOT NULL PRIMARY KEY,
  Bezeichnung_Lang
    VARCHAR(29),
  Bezeichnung_Mittel
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        VARCHAR(19),
        Bezeichnung_Kurz
        VARCHAR(11),
        FOREIGN KEY (Strassenbaudienststelle_ID) REFERENCES
Strassennetz.Bereichsobjekt_hist,
        FOREIGN KEY (Strassenbaudienststelle_ID) REFERENCES Baulasttraeger,
        Art_Strassenbaudienststelle
        INTEGER NOT NULL REFERENCES Art_Strassenbaudienststelle,
        Strasse
        VARCHAR(29),
        Postleitzahl
        VARCHAR(6),
        Ort
        VARCHAR(29)
    )

CREATE TABLE Art_Strassenbaudienststelle (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(30) NOT NULL
)

INSERT INTO Art_Strassenbaudienststelle VALUES (1,'Land_Ministerium')
INSERT INTO Art_Strassenbaudienststelle VALUES
(2,'Regierungspraesidium_Landesamt')
INSERT INTO Art_Strassenbaudienststelle VALUES (3,'Amt')
INSERT INTO Art_Strassenbaudienststelle VALUES (4,'Meisterei')
INSERT INTO Art_Strassenbaudienststelle VALUES (5,'Kreisverwaltung')
INSERT INTO Art_Strassenbaudienststelle VALUES (6,'Gemeindeverwaltung')

CREATE TABLE Land_Ministerium (
    Land_Ministerium_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Land_Ministerium_ID) REFERENCES Strassenbaudienststelle,
    Schluessel_Land_Ministerium
        CHAR(2) NOT NULL
)

CREATE TABLE Regierungspraesidium_Landesamt (
    Regierungspraesidium_Landesamt_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Regierungspraesidium_Landesamt_ID) REFERENCES
Strassenbaudienststelle,
    Schluessel_Regpraes_Landesamt
        CHAR(3) NOT NULL
)

CREATE TABLE Landesamt__untersteht_Land (
    Regierungspraesidium_Landesamt
        INTEGER NOT NULL REFERENCES Regierungspraesidium_Landesamt,
    Land_Ministerium
        INTEGER NOT NULL REFERENCES Land_Ministerium
)

CREATE TABLE Amt (
    Amt_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Amt_ID) REFERENCES Strassenbaudienststelle,
    Schluessel_Amt
        CHAR(4) NOT NULL
)

CREATE TABLE Amt__untersteht_Landesamt (
    Amt
        INTEGER NOT NULL REFERENCES Amt,
    Regierungspraesidium_Landesamt
        INTEGER NOT NULL REFERENCES Regierungspraesidium_Landesamt
)

CREATE TABLE Meisterei (
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Meisterei_ID
    INTEGER NOT NULL PRIMARY KEY,
FOREIGN KEY (Meisterei_ID) REFERENCES Strassenbaudienststelle,
Schluessel_Meisterei
    CHAR(6) NOT NULL
)

CREATE TABLE Meisterei__untersteht_Amt (
    Meisterei
        INTEGER NOT NULL REFERENCES Meisterei,
    Amt
        INTEGER NOT NULL REFERENCES Amt
)

CREATE TABLE Kreisverwaltung (
    Kreisverwaltung_ID
        INTEGER NOT NULL PRIMARY KEY,
FOREIGN KEY (Kreisverwaltung_ID) REFERENCES Strassenbaudienststelle,
FOREIGN KEY (Kreisverwaltung_ID) REFERENCES UI_Partner
)

CREATE TABLE Gemeindeverwaltung (
    Gemeindeverwaltung_ID
        INTEGER NOT NULL PRIMARY KEY,
FOREIGN KEY (Gemeindeverwaltung_ID) REFERENCES Strassenbaudienststelle,
FOREIGN KEY (Gemeindeverwaltung_ID) REFERENCES UI_Partner,
Baulast_Strassenklasse
    CHAR(1),
Gemeindefunktion
    CHAR(1)
)

CREATE TABLE sonstige_UI_Partner (
    sonstige_UI_Partner_ID
        INTEGER NOT NULL PRIMARY KEY,
    Bezeichnung_Lang
        VARCHAR(29),
    Bezeichnung_Mittel
        VARCHAR(19),
    Bezeichnung_Kurz
        VARCHAR(11),
FOREIGN KEY (sonstige_UI_Partner_ID) REFERENCES UI_Partner,
Kennzeichen
    VARCHAR(6)
)

CREATE TABLE UI_Vereinbarung (
    UI_Vereinbarung_ID
        INTEGER NOT NULL PRIMARY KEY,
FOREIGN KEY (UI_Vereinbarung_ID) REFERENCES Strassennetz.Bereichsobjekt_hist,
Vertragsdatum
    CHAR(10), -- Datum
CHECK ( Vertragsdatum LIKE '___.__.____' )
)

CREATE TABLE UI_Vereinbarung__mit_Partner (
    UI_Vereinbarung
        INTEGER NOT NULL REFERENCES UI_Vereinbarung,
    UI_Partner
        INTEGER NOT NULL REFERENCES UI_Partner
)

CREATE TABLE UI_Partner (
    UI_Partner_ID
        INTEGER NOT NULL PRIMARY KEY,
FOREIGN KEY (UI_Partner_ID) REFERENCES Historisierung.historisches_Objekt,
gueltig_von
    CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_von LIKE '___.__.____' ),
gueltig_bis
    CHAR(10) NOT NULL, -- Datum
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CHECK ( gueltig_bis LIKE '___.__.____' ),
Art_UI_Partner
    INTEGER NOT NULL REFERENCES Art_UI_Partner
)

CREATE TABLE Art_UI_Partner (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(19) NOT NULL
)

INSERT INTO Art_UI_Partner VALUES (1, 'Kreisverwaltung')
INSERT INTO Art_UI_Partner VALUES (2, 'Gemeindeverwaltung')
INSERT INTO Art_UI_Partner VALUES (3, 'sonstige_UI_Partner')

CREATE TABLE Baulast (
    Baulast_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Baulast_ID) REFERENCES Strassennetz.Bereichsobjekt_hist,
    Art_Baulast
        CHAR(1) NOT NULL REFERENCES Art_Baulast,
    von_Baulasttraeger
        INTEGER NOT NULL REFERENCES Baulasttraeger
)

CREATE TABLE Art_Baulast (
    Kennung
        CHAR(1) NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Art_Baulast VALUES ('H', 'Hauptbaulast')
INSERT INTO Art_Baulast VALUES ('G', 'Gemeindebaulast')
INSERT INTO Art_Baulast VALUES ('D', 'Baulast Dritter')

CREATE TABLE Baulasttraeger (
    Baulasttraeger_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_Baulasttraeger
        INTEGER NOT NULL REFERENCES Art_Baulasttraeger
)

CREATE TABLE Art_Baulasttraeger (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(30) NOT NULL
)

INSERT INTO Art_Baulasttraeger VALUES (1, 'Land_Ministerium')
INSERT INTO Art_Baulasttraeger VALUES (2, 'Regierungspraesidium_Landesamt')
INSERT INTO Art_Baulasttraeger VALUES (3, 'Amt')
INSERT INTO Art_Baulasttraeger VALUES (4, 'Meisterei')
INSERT INTO Art_Baulasttraeger VALUES (5, 'Kreisverwaltung')
INSERT INTO Art_Baulasttraeger VALUES (6, 'Gemeindeverwaltung')
INSERT INTO Art_Baulasttraeger VALUES (7, 'Baulast_Dritter')

CREATE TABLE Baulast_Dritter (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Bezeichnung_Lang
        VARCHAR(29),
    Bezeichnung_Mittel
        VARCHAR(19),
    Bezeichnung_Kurz
        VARCHAR(11),
    FOREIGN KEY (Kennung) REFERENCES Baulasttraeger,
    Langtext
        VARCHAR(255) NOT NULL
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

INSERT INTO Baulast_Dritter (Kennung,Langtext) VALUES (1,'Deutsche Bahn AG')
INSERT INTO Baulast_Dritter (Kennung,Langtext)
VALUES (2,'Bundeswasserstraßenverwaltung')
INSERT INTO Baulast_Dritter (Kennung,Langtext)
VALUES (3,'Bundesfinanzverwaltung')

CREATE TABLE Dokument (
    Dokument_ID
        INTEGER NOT NULL PRIMARY KEY,
    Kennzeichen_Aktenzeichen
        VARCHAR(30),
    Seite
        INTEGER,
    verfuegt
        INTEGER REFERENCES Logical_Types.Boolean,
    Datum
        CHAR(10), -- Datum
    CHECK ( Datum LIKE '___.__.____' ),
    von_Benutzer
        VARCHAR(12),
    Wirksamkeitsdatum_Dokument
        CHAR(10), -- Datum
    CHECK ( Wirksamkeitsdatum_Dokument LIKE '___.__.____' ),
    Text
        VARCHAR(64)
)

CREATE TABLE OD_FS (
    OD_FS_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (OD_FS_ID) REFERENCES Strassennetz.Streckenobjekt_hist,
    OD_FS
        CHAR(1) NOT NULL REFERENCES Tab_OD_FS
)

CREATE TABLE Tab_OD_FS (
    Kennung
        CHAR(1) NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Tab_OD_FS VALUES ('O','Ortsdurchfahrt')
INSERT INTO Tab_OD_FS VALUES ('V','Verknüpfungsbereich')
INSERT INTO Tab_OD_FS VALUES ('E','Erschließungsbereich')
INSERT INTO Tab_OD_FS VALUES ('F','Freie Strecke')

CREATE TABLE Widmung (
    Widmung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Widmung_ID) REFERENCES Strassennetz.Streckenobjekt_hist,
    rechtsgueltig_ab
        CHAR(10) NOT NULL, -- Datum
    CHECK ( rechtsgueltig_ab LIKE '___.__.____' ),
    Widmung
        INTEGER NOT NULL REFERENCES Tab_Widmung
)

CREATE TABLE Tab_Widmung (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Tab_Widmung VALUES (0,'nicht gewidmet')
INSERT INTO Tab_Widmung VALUES (1,'gewidmet')
INSERT INTO Tab_Widmung VALUES (2,'gewidmet, wird umgestuft')
```

## 2.3 Teilschema „Verkehr“

Dieses Teilschema enthält die SQL-Definitionen für die Teilbereiche „Verkehrliche Beschränkungen“ und „Verkehrlichen Straßeneigenschaften“.

```
CREATE SCHEMA Verkehr
```

```
CREATE TABLE Fahrstreifen_Nummer (
    Fahrstreifen_Nummer_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Fahrstreifen_Nummer_ID) REFERENCES
    Geometrieschema.Flaechenobjekt_Modell,
    FOREIGN KEY (Fahrstreifen_Nummer_ID) REFERENCES
    Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '__.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '__.__.____' ),
    Fahrstreifennummer
        INTEGER,
    hat_Verkehrsrichtung
        CHAR(1) NOT NULL REFERENCES Verkehrsrichtung
)

CREATE TABLE Fahrstr_Nr__auf_Querschnittstr (
    Fahrstreifen_Nummer
        INTEGER NOT NULL REFERENCES Fahrstreifen_Nummer,
    Querschnittstreifen
        INTEGER NOT NULL REFERENCES
    Bauliche_Strasseneigenschaften.Querschnittstreifen
)

CREATE TABLE Verkehrsrichtung (
    Kennung
        CHAR(1) NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Verkehrsrichtung VALUES ('B','Verkehr in beiden Richtungen')
INSERT INTO Verkehrsrichtung VALUES ('R','Einbahnverkehr in ' ||
'Stationierungsrichtung')
INSERT INTO Verkehrsrichtung VALUES ('G','Einbahnverkehr gegen ' ||
'Stationierungsrichtung')

CREATE TABLE Anzahl_Fahrstreifen (
    Anzahl_Fahrstreifen_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Anzahl_Fahrstreifen_ID) REFERENCES
    Strassennetz.Streckenobjekt_stat,
    Fahrstreifen_Gegenrichtung
        INTEGER,
    Fahrstreifen_beide_Richtungen
        INTEGER,
    Fahrstreifen_Richtung
        INTEGER,
    hat_Verkehrsrichtung
        CHAR(1) NOT NULL REFERENCES Verkehrsrichtung
)

CREATE TABLE Fahrzeugart (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    Langtext
    VARCHAR(12) NOT NULL
)

INSERT INTO Fahrzeugart VALUES (1,'LKW')
INSERT INTO Fahrzeugart VALUES (2,'BUS')
INSERT INTO Fahrzeugart VALUES (3,'PKW')
INSERT INTO Fahrzeugart VALUES (4,'Krad')
INSERT INTO Fahrzeugart VALUES (5,'Fahrrad')
INSERT INTO Fahrzeugart VALUES (6,'Fußgänger')

CREATE TABLE Beschraenkung_verkehrlich (
    Beschraenkung_verkehrlich_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Beschraenkung_verkehrlich_ID) REFERENCES
Strassennetz.Streckenobjekt_stat
)

CREATE TABLE Beschr_verk__gilt_f_Verkrichtg (
    Beschraenkung_verkehrlich
    INTEGER NOT NULL REFERENCES Beschraenkung_verkehrlich,
    Verkehrsrichtung
    CHAR(1) NOT NULL REFERENCES Verkehrsrichtung
)

CREATE TABLE Beschr_verk__gilt_f_Fahrzgart (
    Beschraenkung_verkehrlich
    INTEGER NOT NULL REFERENCES Beschraenkung_verkehrlich,
    Fahrzeugart
    INTEGER NOT NULL REFERENCES Fahrzeugart
)

CREATE TABLE Beschr_verk__im_Zeitraum (
    Beschraenkung_verkehrlich
    INTEGER NOT NULL REFERENCES Beschraenkung_verkehrlich,
    Zeitraum
    INTEGER NOT NULL REFERENCES Allgemeine_Objekte.Zeitraum
)

CREATE TABLE Beschr_verk__gilt_f_Fahrstreif (
    Beschraenkung_verkehrlich
    INTEGER NOT NULL REFERENCES Beschraenkung_verkehrlich,
    Fahrstreifen_Nummer
    INTEGER NOT NULL REFERENCES Fahrstreifen_Nummer
)

CREATE TABLE Insassen_min_Anzahl (
    Insassen_min_Anzahl_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Insassen_min_Anzahl_ID) REFERENCES Beschraenkung_verkehrlich,
    Anzahl_Insassen
    INTEGER NOT NULL
)

CREATE TABLE Ueberholverbot (
    Ueberholverbot_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Ueberholverbot_ID) REFERENCES Beschraenkung_verkehrlich,
    Ueberholverbot
    INTEGER NOT NULL REFERENCES Logical_Types.Boolean
)

CREATE TABLE max_Hoehe (
    max_Hoehe_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (max_Hoehe_ID) REFERENCES Beschraenkung_verkehrlich,
    FOREIGN KEY (max_Hoehe_ID) REFERENCES Historisierung.historisches_Objekt,
    gueltig_von
    CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        CHAR(10) NOT NULL, -- Datum
        CHECK ( gueltig_bis LIKE '___.__.____' ),
        Fahrzeughoehe
        REAL NOT NULL, -- Dezimeter
        CHECK ( Fahrzeughoehe>=0 )
    )

CREATE TABLE max_Breite (
    max_Breite_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (max_Breite_ID) REFERENCES Beschraenkung_verkehrlich,
    Fahrzeugbreite
        REAL NOT NULL, -- Dezimeter
        CHECK ( Fahrzeugbreite>=0 )
)

CREATE TABLE max_Laenge (
    max_Laenge_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (max_Laenge_ID) REFERENCES Beschraenkung_verkehrlich,
    max_Laenge
        REAL NOT NULL, -- Meter
        CHECK ( max_Laenge>=0 )
)

CREATE TABLE max_Gewicht (
    max_Gewicht_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (max_Gewicht_ID) REFERENCES Beschraenkung_verkehrlich,
    max_Fahrzeuggewicht
        REAL NOT NULL, -- Tonnen
        CHECK ( max_Fahrzeuggewicht>=0 )
)

CREATE TABLE max_Achsgewicht (
    max_Achsgewicht_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (max_Achsgewicht_ID) REFERENCES Beschraenkung_verkehrlich,
    max_Achsgewicht
        REAL NOT NULL, -- Tonnen
        CHECK ( max_Achsgewicht>=0 )
)

CREATE TABLE max_Geschwindigkeit (
    max_Geschwindigkeit_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (max_Geschwindigkeit_ID) REFERENCES Beschraenkung_verkehrlich,
    max_kmh
        INTEGER NOT NULL, -- Stundenkilometer
        CHECK ( max_kmh>=0 )
)

CREATE TABLE Zugangsbeschraenkung (
    Zugangsbeschraenkung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Zugangsbeschraenkung_ID) REFERENCES Beschraenkung_verkehrlich,
    Zugangsbeschraenkung
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean
)

CREATE TABLE Gesperrt (
    Gesperrt_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Gesperrt_ID) REFERENCES Beschraenkung_verkehrlich,
    Sperrung
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean
)

CREATE TABLE Strassenbeschreibung_verkehrl (
    Strassenbeschreibung_verkehrl_ID
        INTEGER NOT NULL PRIMARY KEY,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
FOREIGN KEY (Strassenbeschreibung_verkehrl_ID) REFERENCES
Strassennetz.Streckenobjekt_stat
)

CREATE TABLE Strbeschr_verk__gilt_f_Vrichtg (
  Strassenbeschreibung_verkehrl
  INTEGER NOT NULL REFERENCES Strassenbeschreibung_verkehrl,
  Verkehrsrichtung
  CHAR(1) NOT NULL REFERENCES Verkehrsrichtung
)

CREATE TABLE Strbeschr_verk__gilt_f_Fzart (
  Strassenbeschreibung_verkehrl
  INTEGER NOT NULL REFERENCES Strassenbeschreibung_verkehrl,
  Fahrzeugart
  INTEGER NOT NULL REFERENCES Fahrzeugart
)

CREATE TABLE Strbeschr_verk__im_Zeitraum (
  Strassenbeschreibung_verkehrl
  INTEGER NOT NULL REFERENCES Strassenbeschreibung_verkehrl,
  Zeitraum
  INTEGER NOT NULL REFERENCES Allgemeine_Objekte.Zeitraum
)

CREATE TABLE Strbeschr_verk__gilt_f_Fahrstr (
  Strassenbeschreibung_verkehrl
  INTEGER NOT NULL REFERENCES Strassenbeschreibung_verkehrl,
  Fahrstreifen_Nummer
  INTEGER NOT NULL REFERENCES Fahrstreifen_Nummer
)

CREATE TABLE Fkt_d_Verb_im_Knotenpktber (
  Fkt_d_Verb_im_Knotenpktber_ID
  INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Fkt_d_Verb_im_Knotenpktber_ID) REFERENCES
Strassenbeschreibung_verkehrl,
  FOREIGN KEY (Fkt_d_Verb_im_Knotenpktber_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
  CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
  CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Funktion
  CHAR(2) NOT NULL REFERENCES Tab_Funktion
)

CREATE TABLE Tab_Funktion (
  Kennung
  CHAR(2) NOT NULL PRIMARY KEY,
  Langtext
  VARCHAR(255) NOT NULL
)

INSERT INTO Tab_Funktion VALUES ('01','Ausfahrt')
INSERT INTO Tab_Funktion VALUES ('02','Einfahrt')
INSERT INTO Tab_Funktion VALUES ('03','Parallelfahrbahn (baulich getrennt)')
INSERT INTO Tab_Funktion VALUES ('04','Verflechtungsspur')
INSERT INTO Tab_Funktion VALUES ('05','Verzögerungsspur')
INSERT INTO Tab_Funktion VALUES ('06','Beschleunigungsspur')

CREATE TABLE Bergpass (
  Bergpass_ID
  INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Bergpass_ID) REFERENCES Strassenbeschreibung_verkehrl,
  Hoehe_ueber_NN
  REAL, -- Meter_ueber_NN
  Anhaenger
  INTEGER REFERENCES Logical_Types.Boolean
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

CREATE TABLE Verkehrsbedeutung (
    Verkehrsbedeutung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Verkehrsbedeutung_ID) REFERENCES Strassenbeschreibung_verkehrl,
    Verkehrsbedeutung
        VARCHAR(30) NOT NULL
)

CREATE TABLE Spur_fuer_Rettungsfahrzeuge (
    Spur_fuer_Rettungsfahrzeuge_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Spur_fuer_Rettungsfahrzeuge_ID) REFERENCES
Strassenbeschreibung_verkehrl,
    Spur_fuer_Rettungsfahrzeuge
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean
)

CREATE TABLE Strassenfunktion (
    Strassenfunktion_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Strassenfunktion_ID) REFERENCES Strassenbeschreibung_verkehrl,
    Strassenfunktion
        VARCHAR(80) NOT NULL
)

CREATE TABLE Aussichtswert (
    Aussichtswert_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Aussichtswert_ID) REFERENCES Strassenbeschreibung_verkehrl,
    Aussichtswert
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean
)

CREATE TABLE Staugefahr (
    Staugefahr_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Staugefahr_ID) REFERENCES Strassenbeschreibung_verkehrl,
    Staugefahr
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean
)

CREATE TABLE gebuehrenpflichtig (
    gebuehrenpflichtig_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (gebuehrenpflichtig_ID) REFERENCES Strassenbeschreibung_verkehrl,
    Gebuehrenpflicht
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean
)

CREATE TABLE Durchschnittsgeschwindigkeit (
    Durchschnittsgeschwindigkeit_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Durchschnittsgeschwindigkeit_ID) REFERENCES
Strassenbeschreibung_verkehrl,
    km_h
        INTEGER NOT NULL, -- Stundenkilometer
    CHECK ( km_h >= 0 )
)
```

### 3 Fachbereich „Bestandsdaten“

#### 3.1 Teilschema „Bauliche Straßeneigenschaften“

Dieses Teilschema enthält die SQL-Definitionen für die Teilbereiche „Bauliche Straßeneigenschaften“, „Durchfahrt/Hindernis“ und „Kreuzung“.

```
CREATE SCHEMA Bauliche_Strasseneigenschaften

CREATE TABLE Schicht (
  Schicht_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Schicht_ID) REFERENCES Strassennetz.Streckenobjekt_hist,
  FOREIGN KEY (Schicht_ID) REFERENCES Geometrieschema.Volumenobjekt_Modell,
  maximale_Korngroesse
    REAL, -- Millimeter
  CHECK ( maximale_Korngroesse >= 0 ),
  Einbaudatum
    CHAR(10), -- Datum
  CHECK ( Einbaudatum LIKE '__.__.____' )
)

CREATE TABLE Schicht__beg_bei_Schichtbegr (
  Schicht
    INTEGER NOT NULL REFERENCES Schicht,
  Schichtbegrenzung
    INTEGER NOT NULL REFERENCES Schichtbegrenzung
)

CREATE TABLE Schicht__endet_bei_Schichtbegr (
  Schicht
    INTEGER NOT NULL REFERENCES Schicht,
  Schichtbegrenzung
    INTEGER NOT NULL REFERENCES Schichtbegrenzung
)

CREATE TABLE Schicht__hat_Schichtart (
  Schicht
    INTEGER NOT NULL REFERENCES Schicht,
  Schichtart
    INTEGER NOT NULL REFERENCES Schichtart
)

CREATE TABLE Schicht__hat_Schichtmaterial (
  Schicht
    INTEGER NOT NULL REFERENCES Schicht,
  Schichtmaterial
    INTEGER NOT NULL REFERENCES Schichtmaterial
)

CREATE TABLE Schicht__unter_Schicht (
  Schicht
    INTEGER NOT NULL REFERENCES Schicht,
  Schicht_2
    INTEGER NOT NULL REFERENCES Schicht
)

CREATE TABLE Schichtbegrenzung (
  Schichtbegrenzung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Schichtbegrenzung_ID) REFERENCES Strassennetz.Punktobjekt_hist,
  Breite
    REAL, -- Meter
  CHECK ( Breite >= 0 ),
  Achsabstand_Anfang
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        REAL, -- Meter
    CHECK ( Achsabstand_Anfang>=0 ),
    Dicke
        REAL, -- Zentimeter
    CHECK ( Dicke>=0 ),
    Abstand_Planum
        REAL, -- Millimeter
    CHECK ( Abstand_Planum>=0 )
)

CREATE TABLE Schichtart (
    Schichtart_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Schichtart_ID) REFERENCES Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Kennung_Schichtart
        INTEGER,
    Kurztext
        VARCHAR(18),
    Schichtbeschreibung
        VARCHAR(29)
)

CREATE TABLE Schichtmaterial (
    Schichtmaterial_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Schichtmaterial_ID) REFERENCES Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Schichtmaterial_Kennung
        INTEGER,
    Schichtmaterial_Kurzbezeichng
        VARCHAR(18),
    Schichtmaterial_Beschreibung
        VARCHAR(29)
)

CREATE TABLE Querschnittstreifen (
    Querschnittstreifen_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Querschnittstreifen_ID) REFERENCES
    Strassennetz.Streckenobjekt_hist,
    FOREIGN KEY (Querschnittstreifen_ID) REFERENCES
    Geometrieschema.Flaechenobjekt_Modell
)

CREATE TABLE Querschnstr__beg_bei_Streifbeg (
    Querschnittstreifen
        INTEGER NOT NULL REFERENCES Querschnittstreifen,
    Streifenbegrenzung
        INTEGER NOT NULL REFERENCES Streifenbegrenzung
)

CREATE TABLE Querschnstr__end_bei_Streifbeg (
    Querschnittstreifen
        INTEGER NOT NULL REFERENCES Querschnittstreifen,
    Streifenbegrenzung
        INTEGER NOT NULL REFERENCES Streifenbegrenzung
)

CREATE TABLE Querschnstr__hat_Streifenart (
    Querschnittstreifen
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER NOT NULL REFERENCES Querschnittstreifen,
    Streifenart
        INTEGER NOT NULL REFERENCES Streifenart
)

CREATE TABLE Streifenart (
    Streifenart_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Streifenart_ID) REFERENCES Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Kennung
        INTEGER,
    Streifenart
        VARCHAR(250)
)

CREATE TABLE Streifenbegrenzung (
    Streifenbegrenzung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Streifenbegrenzung_ID) REFERENCES Strassennetz.Punktobjekt_hist,
    FOREIGN KEY (Streifenbegrenzung_ID) REFERENCES
Geometrieschema.Linienobjekt_Modell,
    Streifenbreite
        REAL, -- Meter
    CHECK ( Streifenbreite >= 0 )
)

CREATE TABLE Bauklasse_Oberbau (
    Bauklasse_Oberbau_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Bauklasse_Oberbau_ID) REFERENCES Strassennetz.Streckenobjekt_hist,
    Strassenoberbau
        VARCHAR(29)
)

CREATE TABLE Bauweise_Oberbau (
    Bauweise_Oberbau_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Bauweise_Oberbau_ID) REFERENCES Strassennetz.Streckenobjekt_hist,
    Oberbauweise
        VARCHAR(29)
)

CREATE TABLE Regelquerschnitt (
    Regelquerschnitt_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Regelquerschnitt_ID) REFERENCES Strassennetz.Streckenobjekt_hist,
    Kennzeichen_Regelquerschnitt
        VARCHAR(10)
)

CREATE TABLE Bahnigkeit (
    Bahnigkeit_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Bahnigkeit_ID) REFERENCES Strassennetz.Streckenobjekt_hist,
    Kennzeichen_Bahnigkeit
        INTEGER REFERENCES Kennzeichen_Bahnigkeit
)

CREATE TABLE Kennzeichen_Bahnigkeit (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Kennzeichen_Bahnigkeit VALUES (1,'einbahnig, Straße mit/ohne ' ||
'Gegenverkehr, ohne Trennstreifen')
INSERT INTO Kennzeichen_Bahnigkeit VALUES (2,'zweibahnig, Straße mit ' ||
'konstruktiv getrennten Richtungsfahrbahnen')
```

```
CREATE TABLE Fahrbahnlaengsneigung (
  Fahrbahnlaengsneigung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Fahrbahnlaengsneigung_ID) REFERENCES
Strassennetz.Streckenobjekt_hist,
  FOREIGN KEY (Fahrbahnlaengsneigung_ID) REFERENCES
Geometrieschema.Flaechenobjekt_Modell,
  Steigung_Gefaelle
    REAL -- Prozent
)

CREATE TABLE Fahrbahnquerneigung (
  Fahrbahnquerneigung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Fahrbahnquerneigung_ID) REFERENCES
Strassennetz.Streckenobjekt_hist,
  FOREIGN KEY (Fahrbahnquerneigung_ID) REFERENCES
Geometrieschema.Flaechenobjekt_Modell,
  Querneigung
    REAL -- Prozent
)

CREATE TABLE Fahrbahntrennung (
  Fahrbahntrennung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Fahrbahntrennung_ID) REFERENCES Strassennetz.Streckenobjekt_stat,
  Art_der_Trennung
    CHAR(1)
)

CREATE TABLE Trassenbreite (
  Trassenbreite_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Trassenbreite_ID) REFERENCES Strassennetz.Streckenobjekt_stat,
  Trassenbreite
    REAL, -- Meter
  CHECK ( Trassenbreite>=0 )
)

CREATE TABLE Fahrbahnbreite (
  Fahrbahnbreite_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Fahrbahnbreite_ID) REFERENCES Strassennetz.Streckenobjekt_stat,
  Fahrbahnbreite
    REAL, -- Meter
  CHECK ( Fahrbahnbreite>=0 )
)

CREATE TABLE Fussweg (
  Fussweg_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Fussweg_ID) REFERENCES Strassennetz.Streckenobjekt_stat
)

CREATE TABLE Radweg (
  Radweg_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Radweg_ID) REFERENCES Strassennetz.Streckenobjekt_stat
)

CREATE TABLE Schutzzone (
  Schutzzone_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Schutzzone_ID) REFERENCES Strassennetz.Bereichsobjekt_hist,
  Kennung_Schutzzone
    CHAR(1),
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Text_Schutzzone
  VARCHAR(29)
)

CREATE TABLE Hindernis (
  Hindernis_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Hindernis_ID) REFERENCES Strassennetz.Streckenobjekt_hist,
  passierbare_Hoehe
    REAL, -- Millimeter
  CHECK ( passierbare_Hoehe>=0 ),
  passierbare_Breite
    REAL, -- Millimeter
  CHECK ( passierbare_Breite>=0 ),
  Belastungsgrenzwert_Hindernis
    REAL, -- Tonnen
  CHECK ( Belastungsgrenzwert_Hindernis>=0 ),
  Laenge_des_Hindernisses
    REAL, -- Meter
  CHECK ( Laenge_des_Hindernisses>=0 ),
  Hindernisart
    INTEGER REFERENCES Hindernis_Art
)

CREATE TABLE Hindernis__ist_Bauwerk (
  Hindernis
    INTEGER NOT NULL REFERENCES Hindernis,
  Teilbauwerk
    INTEGER NOT NULL REFERENCES Bauwerke.Teilbauwerk
)

CREATE TABLE Hindernis_Art (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Hindernis_Art VALUES (1,'Tunnel')
INSERT INTO Hindernis_Art VALUES (2,'Brücken oder ähnliche Konstruktionen ' ||
'(z.B. Stadttor)')
INSERT INTO Hindernis_Art VALUES (3,'größere Konstruktionen neben der ' ||
'Fahrbahn (z.B. Häuser) sowie Masten und Mastenreihen')
INSERT INTO Hindernis_Art VALUES (4,'Fels, Steilböschung')
INSERT INTO Hindernis_Art VALUES (5,'Mauern, Zäune, Geländer')
INSERT INTO Hindernis_Art VALUES (6,'Bäume, Alleen')
INSERT INTO Hindernis_Art VALUES (7,'(kreuzende) Überspannungen (bei ' ||
'Elektrizitätsleitungen Achtung Lebensgefahr! Nicht messen!')')
INSERT INTO Hindernis_Art VALUES (8,'Rückbau wegen ' ||
'Geschwindigkeitsreduzierung')
INSERT INTO Hindernis_Art VALUES (9,'sonstige Hindernisse (z.B. ' ||
'Verkehrszeichen auf 2 Pfosten, Hydranten, Laderampen)')

CREATE TABLE Kreuzung (
  Kreuzung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Kreuzung_ID) REFERENCES Strassennetz.Punktobjekt_hist
)

CREATE TABLE Gewaesserkreuzung (
  Gewaesserkreuzung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Gewaesserkreuzung_ID) REFERENCES Kreuzung,
  Art
    CHAR(1),
  Bezeichnung
    VARCHAR(29),
  Gewaessername
    VARCHAR(29),
  Gewaesserkilometrierung
    FLOAT (7)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

CREATE TABLE Bahnkreuzung (
    Bahnkreuzung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Bahnkreuzung_ID) REFERENCES Kreuzung,
    Bahnkoerperlage
        CHAR(1) REFERENCES Bahnkoerperlage,
    Bahnkoerper
        INTEGER REFERENCES Bahnkoerper,
    Bahnkilometer
        FLOAT (7),
    Streckenbezeichnung
        VARCHAR(11),
    Gleisanzahl
        INTEGER
)

CREATE TABLE Bahnkoerperlage (
    Kennung
        CHAR(1) NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Bahnkoerperlage VALUES ('B','kreuzt gesamte Fahrbahn')
INSERT INTO Bahnkoerperlage VALUES ('L','kreuzt linke Fahrbahn')
INSERT INTO Bahnkoerperlage VALUES ('R','kreuzt rechte Fahrbahn')

CREATE TABLE Bahnkoerper (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Bahnkoerper VALUES (0,'ohne eigenen Bahnkörper')
INSERT INTO Bahnkoerper VALUES (1,'mit eigenem Bahnkörper auf einer Seite')
INSERT INTO Bahnkoerper VALUES (2,'mit eigenem Bahnkörper auf beiden ' ||
'Seiten der Straße')

CREATE TABLE Kreuzung_Strasse_Weg (
    Kreuzung_Strasse_Weg_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Kreuzung_Strasse_Weg_ID) REFERENCES Kreuzung,
    Kreuzungsart
        INTEGER REFERENCES Kreuzungsart,
    klassifiziertes_Netz
        INTEGER REFERENCES Logical_Types.Boolean,
    Bezeichnung_Strasse_Weg
        VARCHAR(29),
    Lage_plangleicher_Kreuzungen
        CHAR(1) REFERENCES Kreuzungslage,
    Vorrangregelung
        INTEGER REFERENCES Vorrangregelung
)

CREATE TABLE Kreuz_Str_Weg__ist_Bauwerk (
    Kreuzung_Strasse_Weg
        INTEGER NOT NULL REFERENCES Kreuzung_Strasse_Weg,
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Bauwerke.Teilbauwerk
)

CREATE TABLE Kreuzungsart (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Kreuzungsart VALUES (1,'Gemeindestraße')
INSERT INTO Kreuzungsart VALUES (2,'sonstige nicht klassifizierte Straße ' ||
'(z.B. Privatstraße)')
INSERT INTO Kreuzungsart VALUES (3,'Feldweg bzw. Forstweg')
INSERT INTO Kreuzungsart VALUES (4,'Radweg, Radwegseitenwechsel')
INSERT INTO Kreuzungsart VALUES (5,'Zufahrt zu bzw. Abfahrt von Rast-, ' ||
'Park- oder Tankanlagen')
INSERT INTO Kreuzungsart VALUES (6,'Grundstückszufahrt')

CREATE TABLE Kreuzungslage (
  Kennung
    CHAR(1) NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Kreuzungslage VALUES ('L','linke Straßenseite')
INSERT INTO Kreuzungslage VALUES ('R','rechte Straßenseite')
INSERT INTO Kreuzungslage VALUES ('B','beide Straßenseiten')

CREATE TABLE Vorrangregelung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(29) NOT NULL
)

INSERT INTO Vorrangregelung VALUES (1,'Signal')
INSERT INTO Vorrangregelung VALUES (2,'Beschilderung')
INSERT INTO Vorrangregelung VALUES (3,'keine Einrichtung')

CREATE TABLE Fussgaengerueberweg (
  Fussgaengerueberweg_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Fussgaengerueberweg_ID) REFERENCES Kreuzung,
  Lage_Fussgaengerueberweg
    CHAR(1) REFERENCES Lage_Ueberweg,
  Regelung_Fussgaengerueberweg
    INTEGER REFERENCES Vorrangregelung
)

CREATE TABLE Lage_Ueberweg (
  Kennung
    CHAR(1) NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Lage_Ueberweg VALUES ('L','linke Fahrbahnhälfte')
INSERT INTO Lage_Ueberweg VALUES ('B','beide Fahrbahnhälften')
INSERT INTO Lage_Ueberweg VALUES ('R','rechte Fahrbahnhälfte')

CREATE TABLE Durchlass (
  Durchlass_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Durchlass_ID) REFERENCES Strassennetz.Punktobjekt_hist,
  Ueberdeckung
    REAL, -- Meter
  CHECK ( Ueberdeckung>=0 ),
  Durchlasslaenge
    REAL, -- Meter
  CHECK ( Durchlasslaenge>=0 ),
  Durchlassart
    VARCHAR(30)
)

CREATE TABLE Durchlass__hat_Baustoff (
  Durchlass
    INTEGER NOT NULL REFERENCES Durchlass,
  Baustoff
    INTEGER NOT NULL REFERENCES Baustoff
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

CREATE TABLE Durchlass__hat_Profil (
  Durchlass
    INTEGER NOT NULL REFERENCES Durchlass,
  Profil
    INTEGER NOT NULL REFERENCES Profil
)

CREATE TABLE Durchlass__hat_Leitung (
  Durchlass
    INTEGER NOT NULL REFERENCES Durchlass,
  Leitung
    INTEGER NOT NULL REFERENCES Strassenausstattungen.Leitung
)

CREATE TABLE Baustoff (
  Baustoff_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Baustoff_ID) REFERENCES Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Material
    VARCHAR(30)
)

CREATE TABLE Profil (
  Profil_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Profil_ID) REFERENCES Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Profilform
    INTEGER REFERENCES Profilform,
  Profilhoehe
    REAL, -- Zentimeter
  CHECK ( Profilhoehe >= 0 ),
  Profilbreite
    REAL, -- Zentimeter
  CHECK ( Profilbreite >= 0 )
)

CREATE TABLE Profilform (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(30) NOT NULL
)

INSERT INTO Profilform VALUES (1,'Kreisprofil')
INSERT INTO Profilform VALUES (2,'Eiprofil')
INSERT INTO Profilform VALUES (3,'Kastenprofil')
INSERT INTO Profilform VALUES (4,'Maulprofil')
INSERT INTO Profilform VALUES (5,'Haubenprofil')
```

### 3.2 Teilschema „Straßenausstattungen“

Dieses Teilschema enthält die SQL-Definitionen für die Teilbereiche „Straßenausstattung“ und „Straßenentwässerung“.

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE SCHEMA Strassenausstattungen
```

```
CREATE TABLE Ausstattung_und_Anlagen (  
  Ausstattung_und_Anlagen_ID  
    INTEGER NOT NULL PRIMARY KEY,  
  FOREIGN KEY (Ausstattung_und_Anlagen_ID) REFERENCES  
Strassennetz.Streckenobjekt_hist,  
  Art_Ausstattung_und_Anlagen  
    INTEGER NOT NULL REFERENCES Art_Ausstattung_und_Anlagen,  
  Kennzeichen  
    CHAR(1),  
  Beschreibungstext  
    VARCHAR(30),  
  Ort_am_Fahrstreifen  
    CHAR(1) REFERENCES Tab_Markierung, -- Markierung  
  Lage_zum_Fahrstreifen  
    CHAR(1), -- Schutzplanke  
  Plankenart  
    VARCHAR(30) -- Schutzplanke  
)
```

```
CREATE TABLE Art_Ausstattung_und_Anlagen (  
  Kennung  
    INTEGER NOT NULL PRIMARY KEY,  
  Langtext  
    VARCHAR(20) NOT NULL  
)
```

```
INSERT INTO Art_Ausstattung_und_Anlagen VALUES (1,'Leitung')  
INSERT INTO Art_Ausstattung_und_Anlagen VALUES (2,'Markierung')  
INSERT INTO Art_Ausstattung_und_Anlagen VALUES (3,'Schutzplanke')  
INSERT INTO Art_Ausstattung_und_Anlagen VALUES (4,'Nebenanlage')  
INSERT INTO Art_Ausstattung_und_Anlagen VALUES (5,'Rastanlage_nicht_bew')  
INSERT INTO Art_Ausstattung_und_Anlagen VALUES (6,'Rastanlage_bew')  
INSERT INTO Art_Ausstattung_und_Anlagen VALUES (7,'Nebenbetrieb')
```

```
CREATE TABLE Leitung (  
  Leitung_ID  
    INTEGER NOT NULL PRIMARY KEY,  
  FOREIGN KEY (Leitung_ID) REFERENCES Strassenausstattung  
)
```

```
CREATE TABLE Leitung__hat_Leitungsverlauf (  
  Leitung  
    INTEGER NOT NULL REFERENCES Leitung,  
  Leitungsverlauf  
    INTEGER NOT NULL REFERENCES Leitungsverlauf  
)
```

```
CREATE TABLE Leitung__hat_Profil (  
  Leitung  
    INTEGER NOT NULL REFERENCES Leitung,  
  Profil  
    INTEGER NOT NULL REFERENCES Bauliche_Strasseneigenschaften.Profil  
)
```

```
CREATE TABLE Leitung__hat_Leitungstraeger (  
  Leitung  
    INTEGER NOT NULL REFERENCES Leitung,  
  Leitungstraeger  
    INTEGER NOT NULL REFERENCES Leitungstraeger  
)
```

```
CREATE TABLE Leitung__hat_Leistungsart (  
  Leitung  
    INTEGER NOT NULL REFERENCES Leitung,  
  Leistungsart  
    INTEGER NOT NULL REFERENCES Leistungsart  
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

CREATE TABLE Leitung__hat_Baustoff (
  Leitung
    INTEGER NOT NULL REFERENCES Leitung,
  Baustoff
    INTEGER NOT NULL REFERENCES Bauliche_Strasseneigenschaften.Baustoff
)

CREATE TABLE Leitung__hat_Dokument (
  Leitung
    INTEGER NOT NULL REFERENCES Leitung,
  Dokument
    INTEGER NOT NULL REFERENCES Administration.Dokument
)

CREATE TABLE Leitungsverlauf (
  Leitungsverlauf_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Leitungsverlauf_ID) REFERENCES Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Leitungslage
    REAL, -- Meter
  CHECK ( Leitungslage >= 0 )
)

CREATE TABLE Leitungstraeger (
  Leitungstraeger_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Leitungstraeger_ID) REFERENCES Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Leitungstraeger_Kurzform
    CHAR(1),
  Leitungstraeger_Langform
    VARCHAR(29)
)

CREATE TABLE Leitungsart (
  Leitungsart_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Leitungsart_ID) REFERENCES Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Kennung_Leitungsart
    CHAR(1),
  Leitungsart_Langtext
    VARCHAR(29),
  UNIQUE ( Kennung_Leitungsart )
)

CREATE TABLE Tab_Markierung (
  Kennung
    CHAR(1) NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Tab_Markierung VALUES ('R','rechts')
INSERT INTO Tab_Markierung VALUES ('L','links')
INSERT INTO Tab_Markierung VALUES ('B','beidseitig')

CREATE TABLE Strassenentwaesserung (
  Strassenentwaesserung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Strassenentwaesserung_ID) REFERENCES
  Strassennetz.Streckenobjekt_stat
)

CREATE TABLE Strassenentw__hat_entw_Fahrblf1 (
  Strassenentwaesserung
    INTEGER NOT NULL REFERENCES Strassenentwaesserung,
  entwaesserte_Fahrbahnflaeche
    INTEGER NOT NULL REFERENCES entwaesserte_Fahrbahnflaeche
)

CREATE TABLE Strassenentw__hat_Lage_Entw (
  Strassenentwaesserung
    INTEGER NOT NULL REFERENCES Strassenentwaesserung,
  Lage_Entwaesserung
    INTEGER NOT NULL REFERENCES Lage_Entwaesserung
)

CREATE TABLE Strassenentw__hat_Art_d_Entw (
  Strassenentwaesserung
    INTEGER NOT NULL REFERENCES Strassenentwaesserung,
  Art_der_Entwaesserung
    INTEGER NOT NULL REFERENCES Art_der_Entwaesserung
)

CREATE TABLE Strassenentw__hat_Dokument (
  Strassenentwaesserung
    INTEGER NOT NULL REFERENCES Strassenentwaesserung,
  Dokument
    INTEGER NOT NULL REFERENCES Administration.Dokument
)

CREATE TABLE entwaesserte_Fahrbahnflaeche (
  entwaesserte_Fahrbahnflaeche_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (entwaesserte_Fahrbahnflaeche_ID) REFERENCES
  Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  entwaesserter_Fahrbahnteil
    CHAR(1),
  Flaeche
    REAL, -- Quadratmeter
  CHECK ( Flaeche>=0 )
)

CREATE TABLE Lage_Entwaesserung (
  Lage_Entwaesserung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Lage_Entwaesserung_ID) REFERENCES
  Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Lage
    CHAR(1) REFERENCES Lage,
  Einleitung

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        CHAR(1) REFERENCES Einleitung
    )

CREATE TABLE Lage (
    Kennung
        CHAR(1) NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Lage VALUES ('R','rechts der Straße')
INSERT INTO Lage VALUES ('L','links der Straße')
INSERT INTO Lage VALUES ('F','auf der Fahrbahn')

CREATE TABLE Einleitung (
    Kennung
        CHAR(1) NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Einleitung VALUES ('P','punktförmige Einleitung')
INSERT INTO Einleitung VALUES ('S','Streckeneinleitung')

CREATE TABLE Art_der_Entwaesserung (
    Art_der_Entwaesserung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Art_der_Entwaesserung_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Transporteinrichtung
        VARCHAR(40),
    Behandlung
        VARCHAR(40),
    Ableitung
        VARCHAR(40)
)

CREATE TABLE Art_der_Entwaess__hat_Leitung (
    Art_der_Entwaesserung
        INTEGER NOT NULL REFERENCES Art_der_Entwaesserung,
    Leitung
        INTEGER NOT NULL REFERENCES Leitung
)
```

## 4 Fachbereich „Straßenzustandsdaten“

### 4.1 Teilschema „Straßenzustandsdaten“

Dieses Teilschema enthält die SQL-Definitionen für die Teilbereiche „Zustand, Erfassung und Bewertung - Organisation -“ und „Zustand, Erfassung und Bewertung - Allgemein -“.

```
CREATE SCHEMA Strassenzustandsdaten

CREATE TABLE ZEB_Objekt (
  ZEB_Objekt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (ZEB_Objekt_ID) REFERENCES Strassennetz.Streckenobjekt_stat,
  Kurzbezeichnung_ZEB_Objekt
    VARCHAR(8),
  ZEB_Objekt
    VARCHAR(30),
  Objektnummer
    INTEGER,
  hat_Verkehrsrichtung
    CHAR(1) NOT NULL REFERENCES Verkehr.Verkehrsrichtung,
  hat_ZEB_Projekt
    INTEGER NOT NULL REFERENCES ZEB_Projekt,
  hat_Art_des_ZEB_Objektes
    INTEGER NOT NULL REFERENCES Art_des_ZEB_Objektes
)

CREATE TABLE ZEB_Obj__gilt_f_Querschnittstr (
  ZEB_Objekt
    INTEGER NOT NULL REFERENCES ZEB_Objekt,
  Querschnittstreifen
    INTEGER NOT NULL REFERENCES
  Bauliche_Strasseneigenschaften.Querschnittstreifen
)

CREATE TABLE ZEB_Obj__gilt_f_Fahrstreifen (
  ZEB_Objekt
    INTEGER NOT NULL REFERENCES ZEB_Objekt,
  Fahrstreifen_Nummer
    INTEGER NOT NULL REFERENCES Verkehr.Fahrstreifen_Nummer
)

CREATE TABLE ZEB_Obj__hat_ZEB_Parameterwert (
  ZEB_Objekt
    INTEGER NOT NULL REFERENCES ZEB_Objekt,
  ZEB_Parameterwert
    INTEGER NOT NULL REFERENCES ZEB_Parameterwert
)

CREATE TABLE ZEB_Projekt (
  ZEB_Projekt_ID
    INTEGER NOT NULL PRIMARY KEY,
  Kurzbezeichnung_ZEB_Kamp
    VARCHAR(8),
  ZEB_Projekt
    VARCHAR(20),
  Hauptmessverfahren
    VARCHAR(20),
  Zweck
    VARCHAR(100),
  Umfang
    VARCHAR(200),
  hat_ZEB_Parameterliste
    INTEGER NOT NULL REFERENCES ZEB_Parameterliste
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE ZEB_Projekt__hat_Zeitraum (
  ZEB_Projekt
    INTEGER NOT NULL REFERENCES ZEB_Projekt,
  Zeitraum
    INTEGER NOT NULL REFERENCES Allgemeine_Objekte.Zeitraum
)

CREATE TABLE ZEB_Projekt__hat_ZEB_Ereignis (
  ZEB_Projekt
    INTEGER NOT NULL REFERENCES ZEB_Projekt,
  ZEB_Ereignis
    INTEGER NOT NULL REFERENCES ZEB_Ereignis
)

CREATE TABLE ZEB_Projekt__hat_ZEB_Subjekt (
  ZEB_Projekt
    INTEGER NOT NULL REFERENCES ZEB_Projekt,
  ZEB_Subjekt
    INTEGER NOT NULL REFERENCES ZEB_Subjekt
)

CREATE TABLE ZEB_Parameterwert (
  ZEB_Parameterwert_ID
    INTEGER NOT NULL PRIMARY KEY,
  Art_ZEB_Parameterwert
    INTEGER NOT NULL REFERENCES Art_ZEB_Parameterwert,
  Kurzbezeichnung
    VARCHAR(8),
  Wert
    VARCHAR(8),
  Einheit
    VARCHAR(8),
  von_ZEB_Parameter
    INTEGER NOT NULL REFERENCES ZEB_Parameter,
  von_ZEB_Ereignis
    INTEGER NOT NULL REFERENCES ZEB_Ereignis,
  Text_ZEB_Parameter
    VARCHAR(80), -- Eintrag_in_Beschreibungstext
  Kurzbez_Mess_Beschreibgroesse
    VARCHAR(8) -- Eintrag_in_Beschreibungstext
)

CREATE TABLE Art_ZEB_Parameterwert (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(29) NOT NULL
)

INSERT INTO Art_ZEB_Parameterwert VALUES (1, 'Wert_der_Zustandsgroesse')
INSERT INTO Art_ZEB_Parameterwert VALUES (2, 'Wert_eines_Messwertes')
INSERT INTO Art_ZEB_Parameterwert VALUES (3, 'Wert_des_Zustandswertes')
INSERT INTO Art_ZEB_Parameterwert VALUES (4, 'Eintrag_in_Beschreibungstext')
INSERT INTO Art_ZEB_Parameterwert VALUES (5, 'Wert_des_Zustandsteilwertes')
INSERT INTO Art_ZEB_Parameterwert VALUES (6, 'Wert_des_Zustandsgesamtwertes')
INSERT INTO Art_ZEB_Parameterwert VALUES (7, 'zugewiesene_Zustandsklasse')

CREATE TABLE Art_des_ZEB_Objektes (
  Art_des_ZEB_Objektes_ID
    INTEGER NOT NULL PRIMARY KEY,
  Kurzbezeichnung_Objektart
    VARCHAR(8),
  ZEB_Objektart
    VARCHAR(30)
)

CREATE TABLE Art_d_ZEB_Obj__hat_Zustmerkmal (
  Art_des_ZEB_Objektes
    INTEGER NOT NULL REFERENCES Art_des_ZEB_Objektes,
  Zustandsmerkmal
```

# OKSTRA – Teilprojekt 4 – Schlußbericht

## Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER NOT NULL REFERENCES Zustandsmerkmal
    )

CREATE TABLE ZEB_Parameter (
    ZEB_Parameter_ID
        INTEGER NOT NULL PRIMARY KEY,
    Kurzbezeichnung_Paramgroesse
        VARCHAR(8),
    Parametergroesse
        VARCHAR(30),
    Parameternummer
        INTEGER,
    von_Zustgroesse_o_Zustwert
        INTEGER NOT NULL REFERENCES Zustandsgroesse_o_Zustandswert
)

CREATE TABLE ZEB_Dokument (
    ZEB_Dokument_ID
        INTEGER NOT NULL PRIMARY KEY,
    Kurzbezeichnung_ZEB_Dokument
        VARCHAR(8),
    ZEB_Dokument
        VARCHAR(20),
    Datum_der_Erstellung
        CHAR(10), -- Datum
    CHECK ( Datum_der_Erstellung LIKE '___.__.____' ),
    Ersteller
        VARCHAR(30),
    Art_des_Dokumentes
        VARCHAR(20)
)

CREATE TABLE ZEB_Ereignis (
    ZEB_Ereignis_ID
        INTEGER NOT NULL PRIMARY KEY,
    Kurzbezeichnung_ZEB_Ereignis
        VARCHAR(8),
    ZEB_Ereignis
        VARCHAR(20),
    hat_ZEB_Dokument
        INTEGER REFERENCES ZEB_Dokument
)

CREATE TABLE ZEB_Ereignis__hat_Zeitraum (
    ZEB_Ereignis
        INTEGER NOT NULL REFERENCES ZEB_Ereignis,
    Zeitraum
        INTEGER NOT NULL REFERENCES Allgemeine_Objekte.Zeitraum
)

CREATE TABLE ZEB_Ereignis__hat_ZEB_Subjekt (
    ZEB_Ereignis
        INTEGER NOT NULL REFERENCES ZEB_Ereignis,
    ZEB_Subjekt
        INTEGER NOT NULL REFERENCES ZEB_Subjekt
)

CREATE TABLE ZEB_Subjekt (
    ZEB_Subjekt_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_ZEB_Subjekt
        INTEGER NOT NULL REFERENCES Art_ZEB_Subjekt,
    Kurzbezeichnung_Subjekt
        VARCHAR(8),
    Subjekt
        VARCHAR(20),
    Adresse_Subjekt
        VARCHAR(100)
)

CREATE TABLE Art_ZEB_Subjekt (
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Kennung
    INTEGER NOT NULL PRIMARY KEY,
Langtext
    VARCHAR(15) NOT NULL
)

INSERT INTO Art_ZEB_Subjekt VALUES (1,'ZEB_Erfasser')
INSERT INTO Art_ZEB_Subjekt VALUES (2,'ZEB_Bewerter')
INSERT INTO Art_ZEB_Subjekt VALUES (3,'ZEB_Veranlasser')

CREATE TABLE ZEB_Parameterliste (
    ZEB_Parameterliste_ID
        INTEGER NOT NULL PRIMARY KEY,
    Kurzbezeichnung_ZEB_Paramliste
        VARCHAR(8),
    Bezeichnung_ZEB_Parameterliste
        VARCHAR(20)
)

CREATE TABLE ZEB_Paramliste__hat_ZEB_Param (
    ZEB_Parameterliste
        INTEGER NOT NULL REFERENCES ZEB_Parameterliste,
    ZEB_Parameter
        INTEGER NOT NULL REFERENCES ZEB_Parameter
)

CREATE TABLE ZEB_Methode (
    ZEB_Methode_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_ZEB_Methode
        INTEGER NOT NULL REFERENCES Art_ZEB_Methode,
    Nummer
        INTEGER,
    Kurzbezeichnung
        CHAR(8),
    Methode
        VARCHAR(30),
    Beschreibung
        VARCHAR(120),
    hat_ZEB_Parameter
        INTEGER NOT NULL REFERENCES ZEB_Parameter
)

CREATE TABLE Art_ZEB_Methode (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(30) NOT NULL
)

INSERT INTO Art_ZEB_Methode VALUES (1,'Zustandsmessverfahren')
INSERT INTO Art_ZEB_Methode VALUES (2,'ZEB_visuell_sensit_Wahrnehmung')
INSERT INTO Art_ZEB_Methode VALUES (3,'Aggregationsmethode')
INSERT INTO Art_ZEB_Methode VALUES (4,'Normierungsmethode')
INSERT INTO Art_ZEB_Methode VALUES (5,'Kombinationsmethode')
INSERT INTO Art_ZEB_Methode VALUES (6,'Klassifizierungsmethode')

CREATE TABLE Zustandserfassungsmethode (
    Zustandserfassungsmethode_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Zustandserfassungsmethode_ID) REFERENCES ZEB_Methode
)

CREATE TABLE Zustandsbewertungsmethode (
    Zustandsbewertungsmethode_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Zustandsbewertungsmethode_ID) REFERENCES ZEB_Methode
)

CREATE TABLE Zustandsgroesse_o_Zustandswert (
    Zustandsgroesse_o_Zustandswert_ID
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER NOT NULL PRIMARY KEY,
    Art_der_Groesse
        VARCHAR(4),
    ID_ZEB_Parameter
        INTEGER
)

CREATE TABLE Zustgr_o_Zustw__bestbar_d_Meth (
    Zustandsgroesse_o_Zustandswert
        INTEGER NOT NULL REFERENCES Zustandsgroesse_o_Zustandswert,
    ZEB_Methode
        INTEGER NOT NULL REFERENCES ZEB_Methode
)

CREATE TABLE Zustandsindikator (
    Zustandsindikator_ID
        INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Zustindik__hat_Zustgr_o_Zustw (
    Zustandsindikator
        INTEGER NOT NULL REFERENCES Zustandsindikator,
    Zustandsgroesse_o_Zustandswert
        INTEGER NOT NULL REFERENCES Zustandsgroesse_o_Zustandswert
)

CREATE TABLE Zustandsmerkmal (
    Zustandsmerkmal_ID
        INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Zustmerkmal__hat_Zustindik (
    Zustandsmerkmal
        INTEGER NOT NULL REFERENCES Zustandsmerkmal,
    Zustandsindikator
        INTEGER NOT NULL REFERENCES Zustandsindikator
)

CREATE TABLE Zustandsmessverfahren (
    Zustandsmessverfahren_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Zustandsmessverfahren_ID) REFERENCES Zustandserfassungsmethode,
    hat_Messwert_o_Beschrgroesse
        INTEGER NOT NULL REFERENCES Messwert_oder_Beschrgroesse
)

CREATE TABLE ZEB_visuell_sensit_Wahrnehmung (
    ZEB_visuell_sensit_Wahrnehmung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (ZEB_visuell_sensit_Wahrnehmung_ID) REFERENCES
    Zustandserfassungsmethode,
    hat_Messwert_o_Beschrgroesse
        INTEGER NOT NULL REFERENCES Messwert_oder_Beschrgroesse
)

CREATE TABLE Aggregationsmethode (
    Aggregationsmethode_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Aggregationsmethode_ID) REFERENCES Zustandserfassungsmethode,
    hat_Messwert_o_Beschrgroesse
        INTEGER NOT NULL REFERENCES Messwert_oder_Beschrgroesse,
    hat_Zustandsgroesse
        INTEGER NOT NULL REFERENCES Zustandsgroesse
)

CREATE TABLE Normierungsmethode (
    Normierungsmethode_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Normierungsmethode_ID) REFERENCES Zustandsbewertungsmethode,
    hat_Zustandsgroesse
        INTEGER NOT NULL REFERENCES Zustandsgroesse,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    hat_Zustandswert
      INTEGER NOT NULL REFERENCES Zustandswert
  )

CREATE TABLE Kombinationsmethode (
  Kombinationsmethode_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Kombinationsmethode_ID) REFERENCES Zustandsbewertungsmethode,
  hat_Zustandswert
    INTEGER NOT NULL REFERENCES Zustandswert,
  hat_Zustandsteilwert
    INTEGER REFERENCES Zustandsteilwert,
  hat_Zustandsgesamtwert
    INTEGER REFERENCES Zustandsgesamtwert,
  CHECK ( ( hat_Zustandsteilwert IS NOT NULL ) OR ( hat_Zustandsgesamtwert IS
NOT NULL ) ) AND ( ( hat_Zustandsteilwert IS NULL ) OR ( hat_Zustandsgesamtwert IS
NULL ) ) )
)

CREATE TABLE Kombmethode__hat_Zustteilwerte (
  Kombinationsmethode
    INTEGER NOT NULL REFERENCES Kombinationsmethode,
  Zustandsteilwert
    INTEGER NOT NULL REFERENCES Zustandsteilwert
)

CREATE TABLE Klassifizierungsmethode (
  Klassifizierungsmethode_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Klassifizierungsmethode_ID) REFERENCES Zustandsbewertungsmethode,
  hat_Zustandsgesamtwert
    INTEGER NOT NULL REFERENCES Zustandsgesamtwert,
  hat_Zustandsklasse
    INTEGER NOT NULL REFERENCES Zustandsklasse
)

CREATE TABLE Messwert_oder_Beschrgroesse (
  Messwert_oder_Beschrgroesse_ID
    INTEGER NOT NULL PRIMARY KEY,
  Kurzbezeichnung_Messw_o_Beschr
    VARCHAR(8),
  Messwert_Beschreibung
    VARCHAR(30)
)

CREATE TABLE Zustandsgroesse (
  Zustandsgroesse_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Zustandsgroesse_ID) REFERENCES Zustandsgroesse_o_Zustandswert
)

CREATE TABLE Zustandswert (
  Zustandswert_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Zustandswert_ID) REFERENCES Zustandsgroesse_o_Zustandswert
)

CREATE TABLE Zustandsteilwert (
  Zustandsteilwert_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Zustandsteilwert_ID) REFERENCES Zustandsgroesse_o_Zustandswert
)

CREATE TABLE Zustandsgesamtwert (
  Zustandsgesamtwert_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Zustandsgesamtwert_ID) REFERENCES Zustandsgroesse_o_Zustandswert
)

CREATE TABLE Zustandsklasse (
  Zustandsklasse_ID
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    INTEGER NOT NULL PRIMARY KEY,  
    FOREIGN KEY (Zustandsklasse_ID) REFERENCES Zustandsgroesse_o_Zustandswert  
)
```

## 5 Fachbereich „Bauwerksdaten“

### 5.1 Teilschema „Bauwerke“

Dieses Teilschema enthält die SQL-Definitionen für die Teilbereiche „Sachverhalt, Teilbauwerke“, „Konstruktionsdetails“, „Maßnahmen“, „Instandsetzungsdetails“, „Hauptbaustoffe“, „Prüfungen“ und „Ausstattungs- und Betriebsdaten von Tunnels“.

```
CREATE SCHEMA Bauwerke
```

```
CREATE TABLE Bauwerk (  
    Bauwerk_ID  
        INTEGER NOT NULL PRIMARY KEY,  
    Bauwerksnummer  
        VARCHAR(7) NOT NULL,  
    Interne_Bauwerksnummer  
        VARCHAR(12),  
    Interner_Sortierschlüssel  
        VARCHAR(15),  
    Bauwerksname  
        VARCHAR(50),  
    naechstgelegener_Ort  
        VARCHAR(50),  
    Bemerkungen  
        VARCHAR(255),  
    Anzahl_Teilbauwerke  
        INTEGER  
)  
  
CREATE TABLE Teilbauwerk (  
    Teilbauwerk_ID  
        INTEGER NOT NULL PRIMARY KEY,  
    FOREIGN KEY (Teilbauwerk_ID) REFERENCES Strassennetz.Punktobjekt_hist,  
    FOREIGN KEY (Teilbauwerk_ID) REFERENCES Strassennetz.Bereichsobjekt_stat,  
    Teilbauwerksnummer  
        VARCHAR(2) NOT NULL,  
    Name_des_Teilbauwerks  
        VARCHAR(50),  
    Interner_Sortierschlüssel  
        VARCHAR(15),  
    Amt  
        VARCHAR(6),  
    Meistere  
        VARCHAR(6),  
    Unterhaltung_Instandsetzung  
        VARCHAR(15),  
    Bauwerksart  
        VARCHAR(15),  
    Stadium_Teilbauwerk  
        VARCHAR(15),  
    Bauwerksakte_Nummer  
        VARCHAR(50),  
    Baulast_Konstruktion  
        VARCHAR(15),  
    Baujahr  
        INTEGER, -- Jahr  
    CHECK ( ( Baujahr >= 1900 ) AND ( Baujahr <= 2099 ) ),  
    Denkmalschutz  
        VARCHAR(50),  
    Unterlagen  
        VARCHAR(255),  
    Datenerfassung_abgeschlossen  
        VARCHAR(15),  
    Konstruktion  
        VARCHAR(50),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Bemerkungen
  VARCHAR(255),
hat_Bauwerkseinzelheiten
  INTEGER NOT NULL REFERENCES Ingenieurbauwerke.Bauwerkseinzelheiten,
gehört_zu_Bauwerk
  INTEGER NOT NULL REFERENCES Bauwerk
)

CREATE TABLE Teilbw__hat_Strbaudienststelle (
  Teilbauwerk
    INTEGER NOT NULL REFERENCES Teilbauwerk,
  Strassenbaudienststelle
    INTEGER NOT NULL REFERENCES Administration.Strassenbaudienststelle
)

CREATE TABLE Teilbauwerk__hat_Route (
  Teilbauwerk
    INTEGER NOT NULL REFERENCES Teilbauwerk,
  Route
    INTEGER NOT NULL REFERENCES Strassennetz.Route
)

CREATE TABLE Teilbauwerk__hat_Sachverhalt (
  Teilbauwerk
    INTEGER NOT NULL REFERENCES Teilbauwerk,
  Sachverhalt
    INTEGER NOT NULL REFERENCES Sachverhalt
)

CREATE TABLE Teilbw__hat_Bauwbeschilderung (
  Teilbauwerk
    INTEGER NOT NULL REFERENCES Teilbauwerk,
  Bauwerksbeschilderung
    INTEGER NOT NULL REFERENCES Bauwerksbeschilderung
)

CREATE TABLE Teilbw__hat_Bauwerksbelaege (
  Teilbauwerk
    INTEGER NOT NULL REFERENCES Teilbauwerk,
  Bauwerksbelaege
    INTEGER NOT NULL REFERENCES Bauwerksbelaege
)

CREATE TABLE Teilbw__hat_Hauptbaust_Beton (
  Teilbauwerk
    INTEGER NOT NULL REFERENCES Teilbauwerk,
  Hauptbaustoff_Beton
    INTEGER NOT NULL REFERENCES Hauptbaustoff_Beton
)

CREATE TABLE Teilbw__hat_Hauptbaust_Stahl (
  Teilbauwerk
    INTEGER NOT NULL REFERENCES Teilbauwerk,
  Hauptbaustoff_Stahl_Holz_Kunst
    INTEGER NOT NULL REFERENCES Hauptbaustoff_Stahl_Holz_Kunst
)

CREATE TABLE Teilbw__hat_Hauptbaust_Verbwst (
  Teilbauwerk
    INTEGER NOT NULL REFERENCES Teilbauwerk,
  Hauptbaustoff_Verbundwerkst
    INTEGER NOT NULL REFERENCES Hauptbaustoff_Verbundwerkst
)

CREATE TABLE Teilbw__hat_Schadensd_abg_Prfg (
  Teilbauwerk
    INTEGER NOT NULL REFERENCES Teilbauwerk,
  Schadensdaten_abgeschl_Prfg
    INTEGER NOT NULL REFERENCES Schadensdaten_abgeschl_Prfg
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Teilbw__hat_Pruefgsd_abg_Prfg (
  Teilbauwerk
  INTEGER NOT NULL REFERENCES Teilbauwerk,
  Pruefungsdaten_abgeschl_Prfg
  INTEGER NOT NULL REFERENCES Pruefungsdaten_abgeschl_Prfg
)

CREATE TABLE Teilbw__hat_gegw_dok_Schadstd (
  Teilbauwerk
  INTEGER NOT NULL REFERENCES Teilbauwerk,
  gegenw_dokum_Schadensstand
  INTEGER NOT NULL REFERENCES gegenw_dokum_Schadensstand
)

CREATE TABLE Teilbw__hat_gegw_dok_Pruefstd (
  Teilbauwerk
  INTEGER NOT NULL REFERENCES Teilbauwerk,
  gegenw_dokum_Pruefungsstand
  INTEGER NOT NULL REFERENCES gegenw_dokum_Pruefungsstand
)

CREATE TABLE Teilbw__hat_durchgef_Prf_Mess (
  Teilbauwerk
  INTEGER NOT NULL REFERENCES Teilbauwerk,
  durchgef_Pruefungen_Messungen
  INTEGER NOT NULL REFERENCES durchgef_Pruefungen_Messungen
)

CREATE TABLE Teilbw__hat_Prueffz_Pruefger (
  Teilbauwerk
  INTEGER NOT NULL REFERENCES Teilbauwerk,
  Pruefffahrzeuge_Pruefgeraete
  INTEGER NOT NULL REFERENCES Pruefffahrzeuge_Pruefgeraete
)

CREATE TABLE Teilbw__hat_Pruefanweisungen (
  Teilbauwerk
  INTEGER NOT NULL REFERENCES Teilbauwerk,
  Pruefanweisungen
  INTEGER NOT NULL REFERENCES Pruefanweisungen
)

CREATE TABLE Teilbw__hat_Verwaltungsmassn (
  Teilbauwerk
  INTEGER NOT NULL REFERENCES Teilbauwerk,
  Verwaltungsmassn_Sondervereinb
  INTEGER NOT NULL REFERENCES Verwaltungsmassn_Sondervereinb
)

CREATE TABLE Teilbw__hat_Bau_u_Erhaltmassn (
  Teilbauwerk
  INTEGER NOT NULL REFERENCES Teilbauwerk,
  Bau_und_Erhaltungsmassnahme
  INTEGER NOT NULL REFERENCES Bau_und_Erhaltungsmassnahme
)

CREATE TABLE Teilbw__hat_Anlagen_Bauwbuch (
  Teilbauwerk
  INTEGER NOT NULL REFERENCES Teilbauwerk,
  Anlagen_Bauwerksbuch
  INTEGER NOT NULL REFERENCES Anlagen_Bauwerksbuch
)

CREATE TABLE Teilbw__hat_Entwue_und_Berechn (
  Teilbauwerk
  INTEGER NOT NULL REFERENCES Teilbauwerk,
  Entwuerfe_und_Berechnungen
  INTEGER NOT NULL REFERENCES Entwuerfe_und_Berechnungen
)

CREATE TABLE Teilbauwerk__hat_Gestaltungen (
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Teilbauwerk
    INTEGER NOT NULL REFERENCES Teilbauwerk,
Gestaltungen
    INTEGER NOT NULL REFERENCES Gestaltungen
)

CREATE TABLE Teilbw__hat_Leitungen_an_Bauw (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Leitungen_an_Bauwerken
        INTEGER NOT NULL REFERENCES Leitungen_an_Bauwerken
)

CREATE TABLE Teilbw__hat_stat_System_Tragf (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    statisches_System_Tragfaehigkt
        INTEGER NOT NULL REFERENCES statisches_System_Tragfaehigkt
)

CREATE TABLE Teilbauwerk__hat_Vorspannungen (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Vorspannungen
        INTEGER NOT NULL REFERENCES Vorspannungen
)

CREATE TABLE Teilbauwerk__hat_Gruendungen (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Gruendungen
        INTEGER NOT NULL REFERENCES Gruendungen
)

CREATE TABLE Teilbw__hat_Erd_und_Felsanker (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Erd_und_Felsanker
        INTEGER NOT NULL REFERENCES Erd_und_Felsanker
)

CREATE TABLE Teilbw__hat_Brueckseil_u_Kabel (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Brueckenseile_und_kabel
        INTEGER NOT NULL REFERENCES Brueckenseile_und_kabel
)

CREATE TABLE Teilbauwerk__hat_Lager (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Lager
        INTEGER NOT NULL REFERENCES Lager
)

CREATE TABLE Teilbw__hat_Fahrbuebergkonstr (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Fahrbahnuebergangskonstruktion
        INTEGER NOT NULL REFERENCES Fahrbahnuebergangskonstruktion
)

CREATE TABLE Teilbauwerk__hat_Abdichtungen (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Abdichtungen
        INTEGER NOT NULL REFERENCES Abdichtungen
)

CREATE TABLE Teilbauwerk__hat_Kappe (
    Teilbauwerk
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Kappe
        INTEGER NOT NULL REFERENCES Kappe
)

CREATE TABLE Teilbw__hat_Schutzeinrichtgen (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Schutzeinrichtungen
        INTEGER NOT NULL REFERENCES Schutzeinrichtungen
)

CREATE TABLE Teilbauwerk__hat_Ausstattung (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Ausstattung
        INTEGER NOT NULL REFERENCES Ausstattung
)

CREATE TABLE Teilbauwerk__hat_Verfuellungen (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Verfuellungen
        INTEGER NOT NULL REFERENCES Verfuellungen
)

CREATE TABLE Teilbw__hat_Betonersatz (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Betonersatz_nach_ZTV_SIB
        INTEGER NOT NULL REFERENCES Betonersatz_nach_ZTV_SIB
)

CREATE TABLE Teilbw__hat_Oberflaechenschutz (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Oberflschutz_nach_ZTV_SIB
        INTEGER NOT NULL REFERENCES Oberflschutz_nach_ZTV_SIB
)

CREATE TABLE Teilbw__hat_Reaktharzgeb_Duennb (
    Teilbauwerk
        INTEGER NOT NULL REFERENCES Teilbauwerk,
    Reaktionsharzgeb_Duennbelaege
        INTEGER NOT NULL REFERENCES Reaktionsharzgeb_Duennbelaege
)

CREATE TABLE Sachverhalt (
    Sachverhalt_ID
        INTEGER NOT NULL PRIMARY KEY,
    Lage_Sachverhalt
        VARCHAR(15),
    Art_des_Sachverhaltes
        VARCHAR(15),
    Name_des_Sachverhaltes
        VARCHAR(255),
    Strassenklasse_des_Sachverh
        CHAR(6),
    Strassennummer_des_Sachverh
        INTEGER,
    Strassenzusatz
        CHAR(1),
    Unterhaltungszuordnung
        INTEGER REFERENCES Logical_Types.Boolean
)

CREATE TABLE Bauwerksbeschilderung (
    Bauwerksbeschilderung_ID
        INTEGER NOT NULL PRIMARY KEY,
    Schild_StVO_Zeichennummer
        VARCHAR(15),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Schild_Bezeichnung
    VARCHAR(50),
Schild_Mengenangabe
    VARCHAR(50),
Schild_Angabe_Zusatzschild
    VARCHAR(50),
Bauwerksbeschild_Bemerkung
    VARCHAR(255)
)

CREATE TABLE Bauwerksbelaege (
    Bauwerksbelaege_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_der_Belaege
        VARCHAR(15),
    Deckschichtkennzeichnung
        INTEGER REFERENCES Logical_Types.Boolean,
    Belaege_Einbauort
        VARCHAR(255),
    Schichtnummer
        INTEGER,
    Schichtdicke
        REAL, -- Millimeter
    CHECK ( Schichtdicke>=0 ),
    Einbaudatum
        CHAR(10), -- Datum
    CHECK ( Einbaudatum LIKE '___.__.____' ),
    Einbauflaeche
        REAL, -- Quadratmeter
    CHECK ( Einbauflaeche>=0 ),
    ausfuehrende_Firma
        VARCHAR(80),
    Bemerkungen
        VARCHAR(255)
)

CREATE TABLE Entwuerfe_und_Berechnungen (
    Entwuerfe_und_Berechnungen_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_Entw_und_Ber
        VARCHAR(15),
    Aufsteller_Entw_und_Ber
        VARCHAR(80),
    Bearbeiter_Entw_und_Ber
        VARCHAR(80),
    Aufstellungsjahr_Entw_und_Ber
        INTEGER, -- Jahr
    CHECK ( ( Aufstellungsjahr_Entw_und_Ber >= 1900 ) AND (
Aufstellungsjahr_Entw_und_Ber <= 2099 ) ),
    Bemerkungen_zu_Entw_und_Ber
        VARCHAR(255)
)

CREATE TABLE Bruecke (
    Bruecke_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Bruecke_ID) REFERENCES Teilbauwerk,
    Gesamtlaenge_Bruecke
        REAL, -- Meter
    CHECK ( Gesamtlaenge_Bruecke>=0 ),
    Breite_Bruecke
        REAL, -- Meter
    CHECK ( Breite_Bruecke>=0 ),
    Gesamtbreite_Bruecke
        REAL, -- Meter
    CHECK ( Gesamtbreite_Bruecke>=0 ),
    Brueckenflaeche
        REAL, -- Quadratmeter
    CHECK ( Brueckenflaeche>=0 ),
    Zwischenraum_Ueberbauten
        REAL, -- Meter
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CHECK ( Zwischenraum_Ueberbauten>=0 ),
Konstruktionshoehe_min
    REAL, -- Meter
CHECK ( Konstruktionshoehe_min>=0 ),
Konstruktionshoehe_max
    REAL, -- Meter
CHECK ( Konstruktionshoehe_max>=0 ),
Anzahl_Felder_Bruecke
    INTEGER,
Anzahl_Ueberbauten_Bruecke
    INTEGER,
Laengsneigung_max
    REAL, -- Prozent
Querneigung_max
    REAL, -- Prozent
Kruemmung
    VARCHAR(15),
Bauwerkswinkel
    REAL, -- Gon
CHECK ( Bauwerkswinkel>=0 ),
Winkelrichtung
    VARCHAR(15),
Querschnitt_Ueberbau
    VARCHAR(15),
Querschnitt_Haupttragwerk
    VARCHAR(15),
Bauverfahren_Ueberbau
    VARCHAR(15),
Konstr_Massn_nachtr_Verstaerk
    VARCHAR(15),
maximale_Ueberschuettungshoehe
    REAL, -- Meter
CHECK ( maximale_Ueberschuettungshoehe>=0 ),
minimale_Ueberschuettungshoehe
    REAL, -- Meter
CHECK ( minimale_Ueberschuettungshoehe>=0 ),
lichte_Hoehe
    REAL, -- Meter
CHECK ( lichte_Hoehe>=0 ),
lichte_Weite_b_Einfeldbruecke
    REAL, -- Meter
CHECK ( lichte_Weite_b_Einfeldbruecke>=0 ),
Bemerkungen_zum_Baugrund
    VARCHAR(255),
Bemerkungen_zur_Bruecke
    VARCHAR(255)
)

CREATE TABLE Teilbw__hat_Brueckenf_Stuetz (
    Bruecke
        INTEGER NOT NULL REFERENCES Bruecke,
    Brueckenfeld_Stuetzung
        INTEGER NOT NULL REFERENCES Brueckenfeld_Stuetzung
)

CREATE TABLE Verkehrszeichenbruecke (
    Verkehrszeichenbruecke_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Verkehrszeichenbruecke_ID) REFERENCES Teilbauwerk,
    Querschnitt_Stiel
        VARCHAR(15),
    Querschnitt_Riegel
        VARCHAR(15),
    Gesamtlaenge_des_Riegels
        REAL, -- Meter
    CHECK ( Gesamtlaenge_des_Riegels>=0 ),
    Besichtigungs_Wartungsoeffnung
        VARCHAR(15),
    Befest_Konstr_Schilder_Signalg
        VARCHAR(15),
    Bemerkung_Verkehrsz_Bruecke
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        VARCHAR(255)
    )

CREATE TABLE Tunnel_Trogbauwerk (
    Tunnel_Trogbauwerk_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Tunnel_Trogbauwerk_ID) REFERENCES Teilbauwerk,
    Gradiente
        VARCHAR(15),
    Rundungshalbmesser
        REAL, -- Meter
    CHECK ( Rundungshalbmesser>=0 ),
    minimale_Laengsneigung
        REAL, -- Prozent
    maximale_Laengsneigung
        REAL, -- Prozent
    minimaler_Radius_im_Grundriss
        REAL, -- Meter
    CHECK ( minimaler_Radius_im_Grundriss>=0 ),
    minimale_Ueberdeckungshoehe
        REAL, -- Meter
    CHECK ( minimale_Ueberdeckungshoehe>=0 ),
    maximale_Ueberdeckungshoehe
        REAL, -- Meter
    CHECK ( maximale_Ueberdeckungshoehe>=0 ),
    Hoehe_rel_NN_in_Tunnelmitte
        REAL, -- Meter
    CHECK ( Hoehe_rel_NN_in_Tunnelmitte>=0 ),
    Bauwerkslaenge
        REAL, -- Meter
    CHECK ( Bauwerkslaenge>=0 ),
    geschl_Laenge_der_Tunnelroehre
        REAL, -- Meter
    CHECK ( geschl_Laenge_der_Tunnelroehre>=0 ),
    Tunnelflaeche
        REAL, -- Quadratmeter
    CHECK ( Tunnelflaeche>=0 ),
    Bauweise
        VARCHAR(15),
    Querschnitt
        VARCHAR(15),
    Ausbruchflaeche
        REAL, -- Quadratmeter
    CHECK ( Ausbruchflaeche>=0 ),
    Sicherung
        VARCHAR(255),
    Geologie
        VARCHAR(15),
    Grund_und_Gebirgswasser verh
        VARCHAR(255),
    Bauverfahren
        VARCHAR(15),
    Vortriebsverfahren
        VARCHAR(15),
    Bemerkungen_zum_Tunnel_Trogbau
        VARCHAR(255)
    )

CREATE TABLE Tunnel_Trogbw__hat_Segmente (
    Tunnel_Trogbauwerk
        INTEGER NOT NULL REFERENCES Tunnel_Trogbauwerk,
    Segment_Tunnel_Trogbauwerk
        INTEGER NOT NULL REFERENCES Segment_Tunnel_Trogbauwerk
    )

CREATE TABLE Tunnel_Trogbw__hat_Tunbeleucht (
    Tunnel_Trogbauwerk
        INTEGER NOT NULL REFERENCES Tunnel_Trogbauwerk,
    Tunnelbeleuchtung
        INTEGER NOT NULL REFERENCES Tunnelbeleuchtung
    )
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Tunnel_Trogbw__hat_Tunnellueft (
  Tunnel_Trogbauwerk
    INTEGER NOT NULL REFERENCES Tunnel_Trogbauwerk,
  Tunnellueftung
    INTEGER NOT NULL REFERENCES Tunnellueftung
)

CREATE TABLE Tunnel_Trogbw__hat_Zentr_An1 (
  Tunnel_Trogbauwerk
    INTEGER NOT NULL REFERENCES Tunnel_Trogbauwerk,
  Tunnel_Zentrale_Anlagen
    INTEGER NOT NULL REFERENCES Tunnel_Zentrale_Anlagen
)

CREATE TABLE Tunnel_Trogbw__hat_Tunsicherh (
  Tunnel_Trogbauwerk
    INTEGER NOT NULL REFERENCES Tunnel_Trogbauwerk,
  Tunnelsicherheit
    INTEGER NOT NULL REFERENCES Tunnelsicherheit
)

CREATE TABLE Tunnel_Trogbw__hat_Tunverkeinr (
  Tunnel_Trogbauwerk
    INTEGER NOT NULL REFERENCES Tunnel_Trogbauwerk,
  Tunnel_Verkehrseinrichtungen
    INTEGER NOT NULL REFERENCES Tunnel_Verkehrseinrichtungen
)

CREATE TABLE Laermschutzbauwerk (
  Laermschutzbauwerk_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Laermschutzbauwerk_ID) REFERENCES Teilbauwerk,
  Gesamtlaenge
    REAL, -- Meter
  CHECK ( Gesamtlaenge>=0 ),
  Flaeche_des_Laermschirmes
    REAL, -- Quadratmeter
  CHECK ( Flaeche_des_Laermschirmes>=0 ),
  Bemerkungen_zum_Laermschutzbw
    VARCHAR(255)
)

CREATE TABLE Laermschbw__hat_Segmente (
  Laermschutzbauwerk
    INTEGER NOT NULL REFERENCES Laermschutzbauwerk,
  Segment_Laermschutzbauwerk
    INTEGER NOT NULL REFERENCES Segment_Laermschutzbauwerk
)

CREATE TABLE Stuetzbauwerk (
  Stuetzbauwerk_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Stuetzbauwerk_ID) REFERENCES Teilbauwerk,
  Gesamtlaenge_Stuetzbauwerk
    REAL, -- Meter
  CHECK ( Gesamtlaenge_Stuetzbauwerk>=0 ),
  Flaeche_Stuetzbauwerk
    REAL, -- Quadratmeter
  CHECK ( Flaeche_Stuetzbauwerk>=0 ),
  Bemerkungen_zum_Stuetzbauwerk
    VARCHAR(255)
)

CREATE TABLE Stuetzbauwerk__hat_Segmente (
  Stuetzbauwerk
    INTEGER NOT NULL REFERENCES Stuetzbauwerk,
  Segment_Stuetzbauwerk
    INTEGER NOT NULL REFERENCES Segment_Stuetzbauwerk
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE sonstiges_Bauwerk (
  sonstiges_Bauwerk_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (sonstiges_Bauwerk_ID) REFERENCES Teilbauwerk,
  Stuetzweite_sonstiges_Bauwerk
    REAL, -- Meter
  CHECK ( Stuetzweite_sonstiges_Bauwerk >= 0 ),
  Beschreibung_sonstiges_Bauwerk
    VARCHAR(255)
)
```

```
CREATE TABLE Brueckenfeld_Stuetzung (
  Brueckenfeld_Stuetzung_ID
    INTEGER NOT NULL PRIMARY KEY,
  Art_der_Stuetzung
    VARCHAR(15),
  Feldnummer
    INTEGER,
  Stuetzweite
    REAL, -- Meter
  CHECK ( Stuetzweite >= 0 ),
  Stuetzungshoehe
    REAL, -- Meter
  CHECK ( Stuetzungshoehe >= 0 ),
  Anzahl_d_Stuetzen_einer_Reihe
    INTEGER,
  Bemerkungen
    VARCHAR(255)
)
```

```
CREATE TABLE Segment_Tunnel_Trogbauwerk (
  Segment_Tunnel_Trogbauwerk_ID
    INTEGER NOT NULL PRIMARY KEY,
  Segmentlaenge
    REAL, -- Meter
  CHECK ( Segmentlaenge >= 0 ),
  Oberflaeche_Tunneldecke
    VARCHAR(15),
  Oberflaeche_Tunnelwaende
    VARCHAR(15),
  Hoehe_Segmentanfang_rechts
    REAL, -- Meter
  CHECK ( Hoehe_Segmentanfang_rechts >= 0 ),
  Hoehe_Segmentanfang_links
    REAL, -- Meter
  CHECK ( Hoehe_Segmentanfang_links >= 0 ),
  Hoehe_Segmentende_rechts
    REAL, -- Meter
  CHECK ( Hoehe_Segmentende_rechts >= 0 ),
  Hoehe_Segmentende_links
    REAL, -- Meter
  CHECK ( Hoehe_Segmentende_links >= 0 ),
  Bemerkungen_Segm_Tunnel_Trogbw
    VARCHAR(255)
)
```

```
CREATE TABLE Segment_Laermschutzbauwerk (
  Segment_Laermschutzbauwerk_ID
    INTEGER NOT NULL PRIMARY KEY,
  Segmentlaenge_Laermschutz
    REAL, -- Meter
  CHECK ( Segmentlaenge_Laermschutz >= 0 ),
  Hoehe_Segmentanfang
    REAL, -- Meter
  CHECK ( Hoehe_Segmentanfang >= 0 ),
  Hoehe_Segmentende
    REAL, -- Meter
  CHECK ( Hoehe_Segmentende >= 0 ),
  Pfostenabstand
    REAL, -- Meter
  CHECK ( Pfostenabstand >= 0 ),
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Standort_des_Pfostens
  VARCHAR(15),
Untergrund
  VARCHAR(15),
Fangvorrichtung
  VARCHAR(15),
Vogelschutz
  VARCHAR(15),
akustische_Eig_Vorsatzschale
  VARCHAR(15),
Pflanzungen_Verkehrsseite
  VARCHAR(15),
Pflanzungen_Anliegerseite
  VARCHAR(15),
Besichtigungsweg_Anliegerseite
  VARCHAR(15),
Bemerkungen_Laermschutz
  VARCHAR(255)
)

CREATE TABLE Segment_Stuetzbauwerk (
  Segment_Stuetzbauwerk_ID
    INTEGER NOT NULL PRIMARY KEY,
  Segmentlaenge_Stuetzbauwerk
    REAL, -- Meter
  CHECK ( Segmentlaenge_Stuetzbauwerk >= 0 ),
  Hoehe_Segmentanfang_Stuetzbw
    REAL, -- Meter
  CHECK ( Hoehe_Segmentanfang_Stuetzbw >= 0 ),
  Hoehe_Segmentende_Stuetzbw
    REAL, -- Meter
  CHECK ( Hoehe_Segmentende_Stuetzbw >= 0 ),
  Besichtigungsweg_Anliegerseite
    VARCHAR(15),
  Bemerkungen_zum_Stuetzbauwerk
    VARCHAR(255)
)

CREATE TABLE Pruefanweisungen (
  Pruefanweisungen_ID
    INTEGER NOT NULL PRIMARY KEY,
  abweichende_Pruefpflicht
    VARCHAR(15),
  Pruefanweisungen
    VARCHAR(255),
  Tauchereinsatz
    VARCHAR(15),
  Pruefung_elektr_Einrichtungen
    VARCHAR(15),
  Pruefung_maschn_Einrichtungen
    VARCHAR(15),
  Bemerkungen_zur_Pruefanweisung
    VARCHAR(255)
)

CREATE TABLE durchgef_Pruefungen_Messungen (
  durchgef_Pruefungen_Messungen_ID
    INTEGER NOT NULL PRIMARY KEY,
  Art_der_Pruefung
    VARCHAR(80),
  Pruefungs_Messungsdatum
    CHAR(10), -- Datum
  CHECK ( Pruefungs_Messungsdatum LIKE '___.__.____' ),
  Pruefungs_Messungszyklus
    INTEGER,
  Zustandsnote
    FLOAT (3),
  Bemerkungen_zur_durchgef_Prfg
    VARCHAR(255)
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE gegenw_dokum_Schadensstand (  
  gegenw_dokum_Schadensstand_ID  
    INTEGER NOT NULL PRIMARY KEY,  
  Menge_allgemein  
    VARCHAR(15),  
  Menge_mit_Dimension_Schluessel  
    VARCHAR(15),  
  Menge_mit_Dimension  
    FLOAT (7),  
  Ortsbestimmungen_Ueberbau  
    VARCHAR(15),  
  Ortsbest_Feld_Pfeiler_Block  
    VARCHAR(15),  
  Ortsbestimmung_Ueberbau  
    VARCHAR(15),  
  Ortsbest_laengs_Schluessel  
    VARCHAR(15),  
  Ortsbestimmung_laengs_Werte  
    INTEGER,  
  Ortsbestimmung_quer  
    VARCHAR(15),  
  Ortsbestimmung_hoch_Schluessel  
    VARCHAR(15),  
  Ortsbestimmung_hoch_Werte  
    INTEGER,  
  Schadensbewertung  
    VARCHAR(15),  
  Schadensveraenderungen  
    VARCHAR(15),  
  Bemerkungen_Schluessel  
    VARCHAR(15),  
  Bemerkungen_Text  
    VARCHAR(255),  
  Hauptbauteil  
    VARCHAR(15),  
  Konstruktionsteil  
    VARCHAR(15),  
  Bauteilergaenzung  
    VARCHAR(15),  
  Schaden  
    VARCHAR(15)  
)
```

```
CREATE TABLE gegenw_dokum_Pruefungsstand (  
  gegenw_dokum_Pruefungsstand_ID  
    INTEGER NOT NULL PRIMARY KEY,  
  Menge_allgemein  
    VARCHAR(15),  
  Menge_mit_Dimension_Schluessel  
    VARCHAR(15),  
  Menge_mit_Dimension  
    FLOAT (7),  
  Ortsbestimmungen_Ueberbau  
    VARCHAR(15),  
  Ortsbest_Feld_Pfeiler_Block  
    VARCHAR(15),  
  Ortsbestimmung_Ueberbau  
    VARCHAR(15),  
  Ortsbest_laengs_Schluessel  
    VARCHAR(15),  
  Ortsbestimmung_laengs_Werte  
    INTEGER,  
  Ortsbestimmung_quer  
    VARCHAR(15),  
  Ortsbestimmung_hoch_Schluessel  
    VARCHAR(15),  
  Ortsbestimmung_hoch_Werte  
    INTEGER,  
  Schadensbewertung  
    VARCHAR(15),  
  Schadensveraenderungen
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        VARCHAR(15),
    Bemerkungen_Schluessel
        VARCHAR(15),
    Bemerkungen_Text
        VARCHAR(255),
    Hauptbauteil
        VARCHAR(15),
    Konstruktionsteil
        VARCHAR(15),
    Bauteilergaenzung
        VARCHAR(15),
    Schaden
        VARCHAR(15)
)

CREATE TABLE Schadensdaten_abgeschl_Prfg (
    Schadensdaten_abgeschl_Prfg_ID
        INTEGER NOT NULL PRIMARY KEY,
    Menge_allgemein
        VARCHAR(15),
    Menge_mit_Dimension_Schluessel
        VARCHAR(15),
    Menge_mit_Dimension
        FLOAT (7),
    Ortsbestimmungen_Ueberbau
        VARCHAR(15),
    Ortsbest_Feld_Pfeiler_Block
        VARCHAR(15),
    Ortsbestimmung_Ueberbau
        VARCHAR(15),
    Ortsbest_laengs_Schluessel
        VARCHAR(15),
    Ortsbestimmung_laengs_Werte
        INTEGER,
    Ortsbestimmung_quer
        VARCHAR(15),
    Ortsbestimmung_hoch_Schluessel
        VARCHAR(15),
    Ortsbestimmung_hoch_Werte
        INTEGER,
    Schadensbewertung
        VARCHAR(15),
    Schadensveraenderungen
        VARCHAR(15),
    Bemerkungen_Schluessel
        VARCHAR(15),
    Bemerkungen_Text
        VARCHAR(255),
    Hauptbauteil
        VARCHAR(15),
    Konstruktionsteil
        VARCHAR(15),
    Bauteilergaenzung
        VARCHAR(15),
    Schaden
        VARCHAR(15)
)

CREATE TABLE Pruefungsdaten_abgeschl_Prfg (
    Pruefungsdaten_abgeschl_Prfg_ID
        INTEGER NOT NULL PRIMARY KEY,
    Menge_allgemein
        VARCHAR(15),
    Menge_mit_Dimension_Schluessel
        VARCHAR(15),
    Menge_mit_Dimension
        FLOAT (7),
    Ortsbestimmungen_Ueberbau
        VARCHAR(15),
    Ortsbest_Feld_Pfeiler_Block
        VARCHAR(15),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Ortsbestimmung_Ueberbau
    VARCHAR(15),
Ortsbest_laengs_Schluessel
    VARCHAR(15),
Ortsbestimmung_laengs_Werte
    INTEGER,
Ortsbestimmung_quer
    VARCHAR(15),
Ortsbestimmung_hoch_Schluessel
    VARCHAR(15),
Ortsbestimmung_hoch_Werte
    INTEGER,
Schadensbewertung
    VARCHAR(15),
Schadensveraenderungen
    VARCHAR(15),
Bemerkungen_Schluessel
    VARCHAR(15),
Bemerkungen_Text
    VARCHAR(255),
Hauptbauteil
    VARCHAR(15),
Konstruktionsteil
    VARCHAR(15),
Bauteilergaenzung
    VARCHAR(15),
Schaden
    VARCHAR(15)
)

CREATE TABLE Prueffahrzeuge_Pruefgeraete (
    Prueffahrzeuge_Pruefgeraete_ID
        INTEGER NOT NULL PRIMARY KEY,
    Geraeteart
        VARCHAR(15),
    voraussichtl_Einsatzdauer
        INTEGER, -- Tage
    CHECK ( voraussichtl_Einsatzdauer >= 0 ),
    Bemerkungen_zu_Prueffahrzeugen
        VARCHAR(255)
)

CREATE TABLE Anlagen_Bauwerksbuch (
    Anlagen_Bauwerksbuch_ID
        INTEGER NOT NULL PRIMARY KEY,
    Beschreibung_sonst_Anlagen
        VARCHAR(255)
)

CREATE TABLE Verwaltungsmassn_Sondervereinb (
    Verwaltungsmassn_Sondervereinb_ID
        INTEGER NOT NULL PRIMARY KEY,
    Aktenzeichen_des_Verw_Akt
        VARCHAR(30),
    Veranlasser
        VARCHAR(80),
    Wirksamkeitsdatum
        CHAR(10), -- Datum
    CHECK ( Wirksamkeitsdatum LIKE '__.__.____' ),
    Beschreibung_der_Massnahme
        VARCHAR(255)
)

CREATE TABLE Bau_und_Erhaltungsmassnahme (
    Bau_und_Erhaltungsmassnahme_ID
        INTEGER NOT NULL PRIMARY KEY,
    Baumassnahmenummer
        VARCHAR(80),
    Bezeichnung_der_Baumassnahme
        VARCHAR(80),
    Jahr_der_Inbetriebnahme
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER, -- Jahr
        CHECK ( ( Jahr_der_Inbetriebnahme >= 1900 ) AND ( Jahr_der_Inbetriebnahme <=
2099 ) ),
        Auftragssumme
            REAL, -- DM
        CHECK ( Auftragssumme >= 0 ),
        Abrechnungssumme
            REAL, -- DM
        CHECK ( Abrechnungssumme >= 0 ),
        Auftraggeber_Bau_Erh_Massn
            VARCHAR(80),
        Auftragnehmer_Bau_Erh_Massn
            VARCHAR(80),
        Bauueberwachung
            VARCHAR(80),
        Baubeginn_Bau_Erh_Massn
            CHAR(10), -- Datum
        CHECK ( Baubeginn_Bau_Erh_Massn LIKE '___.__.____' ),
        Bauende_Bau_Erh_Massn
            CHAR(10), -- Datum
        CHECK ( Bauende_Bau_Erh_Massn LIKE '___.__.____' ),
        Ablauf_Gewaehrleistung
            CHAR(10), -- Datum
        CHECK ( Ablauf_Gewaehrleistung LIKE '___.__.____' ),
        Bemerkungen_Bau_Erh_Massn
            VARCHAR(255)
    )

CREATE TABLE Bau_u_Erhaltungsmassn__hat_Kosten (
    Bau_und_Erhaltungsmassnahme
        INTEGER NOT NULL REFERENCES Bau_und_Erhaltungsmassnahme,
    Kosten_fuer_Bau_Erh_und_Betr
        INTEGER NOT NULL REFERENCES Kosten_fuer_Bau_Erh_und_Betr
)

CREATE TABLE Kosten_fuer_Bau_Erh_und_Betr (
    Kosten_fuer_Bau_Erh_und_Betr_ID
        INTEGER NOT NULL PRIMARY KEY,
    Kostenart_Bau_Erh_Kosten
        VARCHAR(10),
    Menge_Bau_Erh_Kosten
        FLOAT (12),
    Haushaltsjahr_Bau_Erh_Kosten
        INTEGER, -- Jahr
    CHECK ( ( Haushaltsjahr_Bau_Erh_Kosten >= 1900 ) AND (
Haushaltsjahr_Bau_Erh_Kosten <= 2099 ) ),
    Kostentraeger_Bau_Erh_Kosten
        VARCHAR(120),
    Titel_Bau_Erh_Kosten
        VARCHAR(20),
    Ausgabe_Bau_Erh_Kosten
        REAL, -- DM
    CHECK ( Ausgabe_Bau_Erh_Kosten >= 0 ),
    Bemerkungen
        VARCHAR(255)
)

CREATE TABLE Ausstattung (
    Ausstattung_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art
        VARCHAR(10),
    Einbauort
        VARCHAR(255),
    TUEV_Datum
        CHAR(10), -- Datum
    CHECK ( TUEV_Datum LIKE '___.__.____' ),
    Bemerkungen_zu_Ausstattungen
        VARCHAR(255)
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Schutzeinrichtungen (
  Schutzeinrichtungen_ID
    INTEGER NOT NULL PRIMARY KEY,
  Art_Schutzeinrichtung
    VARCHAR(15),
  Einbauort_Schutzeinrichtung
    VARCHAR(255),
  Einbaujahr_Schutzeinrichtung
    INTEGER, -- Jahr
  CHECK ( ( Einbaujahr_Schutzeinrichtung >= 1900 ) AND (
Einbaujahr_Schutzeinrichtung <= 2099 ) ),
  Laenge_Schutzeinrichtung
    REAL, -- Meter
  CHECK ( Laenge_Schutzeinrichtung>=0 ),
  Hoehe_Schutzeinrichtung
    REAL, -- Meter
  CHECK ( Hoehe_Schutzeinrichtung>=0 ),
  Bemerkungen
    VARCHAR(255)
)

CREATE TABLE Vorspannungen (
  Vorspannungen_ID
    INTEGER NOT NULL PRIMARY KEY,
  Richtung
    VARCHAR(15),
  intern_extern
    VARCHAR(15),
  Vorspannungsgrad
    VARCHAR(15),
  Spannverfahren
    VARCHAR(15),
  Spannkraft
    REAL, -- Kilonewton
  CHECK ( Spannkraft>=0 ),
  Spannstahl_Streckgrenze
    INTEGER, -- Newton_pro_Quadratmillimeter
  CHECK ( Spannstahl_Streckgrenze>=0 ),
  Spannstahl_Bruchgrenze
    INTEGER, -- Newton_pro_Quadratmillimeter
  CHECK ( Spannstahl_Bruchgrenze>=0 ),
  Hersteller
    VARCHAR(15),
  Art_der_Verpressung
    VARCHAR(15),
  Typenbezeichnung
    VARCHAR(80),
  Einbaujahr_Vorspannung
    INTEGER, -- Jahr
  CHECK ( ( Einbaujahr_Vorspannung >= 1900 ) AND ( Einbaujahr_Vorspannung <= 2099
) ),
  Einbauort_Vorspannung
    VARCHAR(255),
  Bemerkungen
    VARCHAR(255)
)

CREATE TABLE Gruendungen (
  Gruendungen_ID
    INTEGER NOT NULL PRIMARY KEY,
  Art_der_Gruendungen
    VARCHAR(15),
  Typenbezeichnung_Gruendung
    VARCHAR(80),
  Einbauort_Gruendung
    VARCHAR(255),
  Einbaujahr_Gruendung
    INTEGER, -- Jahr
  CHECK ( ( Einbaujahr_Gruendung >= 1900 ) AND ( Einbaujahr_Gruendung <= 2099 )
),
  Bemerkungen_zur_Gruendung
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        VARCHAR(255)
    )

CREATE TABLE Erd_und_Felsanker (
    Erd_und_Felsanker_ID
        INTEGER NOT NULL PRIMARY KEY,
    Verfahren_Erd_und_Felsanker
        VARCHAR(15),
    Hersteller_Erd_und_Felsanker
        VARCHAR(15),
    Zulassungsnummer
        VARCHAR(50),
    Anzahl_Erd_und_Felsanker
        INTEGER,
    Art_Erd_und_Felsanker
        VARCHAR(255),
    Typenbez_Erd_und_Felsanker
        VARCHAR(80),
    Einbauort_Erd_und_Felsanker
        VARCHAR(255),
    Einbaujahr_Erd_und_Felsanker
        INTEGER, -- Jahr
    CHECK ( ( Einbaujahr_Erd_und_Felsanker >= 1900 ) AND (
Einbaujahr_Erd_und_Felsanker <= 2099 ) ),
    Bemerkungen_Erd_und_Felsanker
        VARCHAR(255)
    )

CREATE TABLE Lager (
    Lager_ID
        INTEGER NOT NULL PRIMARY KEY,
    Anzahl_der_Lager
        INTEGER,
    Art_der_Lager
        VARCHAR(15),
    Typenbezeichnung
        VARCHAR(80),
    Einbauort_der_Lager
        VARCHAR(255),
    Einbaujahr_der_Lager
        INTEGER, -- Jahr
    CHECK ( ( Einbaujahr_der_Lager >= 1900 ) AND ( Einbaujahr_der_Lager <= 2099 )
),
    Hersteller
        VARCHAR(15),
    Bemerkungen
        VARCHAR(255)
    )

CREATE TABLE Fahrbahnuebergangskonstruktion (
    Fahrbahnuebergangskonstruktion_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_Fahrbahnuebergang
        VARCHAR(15),
    Anzahl_Fahrbahnuebergaenge
        INTEGER,
    Anzahl_der_Lamellen
        INTEGER,
    Gesamtdehnweg
        REAL, -- Millimeter
    CHECK ( Gesamtdehnweg>=0 ),
    Hersteller_Fahrbahnuebergaenge
        VARCHAR(15),
    Typenbez_Fahrbahnuebergaenge
        VARCHAR(80),
    regelgepr_Fahrbahnuebergaenge
        VARCHAR(15),
    Einbauort_Fahrbahnuebergaenge
        VARCHAR(255),
    Einbaujahr_Fahrbahnuebergaenge
        INTEGER, -- Jahr
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    CHECK ( ( Einbaujahr_Fahrbahnuebergaenge >= 1900 ) AND (
Einbaujahr_Fahrbahnuebergaenge <= 2099 ) ),
    Konstruktionslaenge_Uebergang
        REAL, -- Meter
    CHECK ( Konstruktionslaenge_Uebergang>=0 ),
    Wartungsg_Fahrbahnuebergaenge
        VARCHAR(15),
    Bemerkungen
        VARCHAR(255)
)

CREATE TABLE Kappe (
    Kappe_ID
        INTEGER NOT NULL PRIMARY KEY,
    Einbauort_Kappe
        VARCHAR(255),
    Konstruktion_Kappe
        VARCHAR(15),
    Verankerung_Kappe
        VARCHAR(15),
    Ankerabstand
        REAL, -- Meter
    CHECK ( Ankerabstand>=0 ),
    groesste_Blocklaenge
        REAL, -- Meter
    CHECK ( groesste_Blocklaenge>=0 ),
    Breite_Kappe
        REAL, -- Meter
    CHECK ( Breite_Kappe>=0 ),
    Einbaujahr_Kappe
        INTEGER, -- Jahr
    CHECK ( ( Einbaujahr_Kappe >= 1900 ) AND ( Einbaujahr_Kappe <= 2099 ) ),
    Bemerkungen_zur_Kappe
        VARCHAR(255)
)

CREATE TABLE Brueckenseile_und_kabel (
    Brueckenseile_und_kabel_ID
        INTEGER NOT NULL PRIMARY KEY,
    Seilart
        VARCHAR(15),
    Hersteller
        VARCHAR(15),
    Tragfunktion
        VARCHAR(15),
    Seillaenge
        REAL, -- Meter
    CHECK ( Seillaenge>=0 ),
    Seildurchmesser
        REAL, -- Millimeter
    CHECK ( Seildurchmesser>=0 ),
    Seilquerschnitt_metallisch
        REAL, -- Quadratmillimeter
    CHECK ( Seilquerschnitt_metallisch>=0 ),
    Drahtfestigkeit
        INTEGER, -- Newton_pro_Quadratmillimeter
    CHECK ( Drahtfestigkeit>=0 ),
    Seilkraft
        REAL, -- Kilonewton
    CHECK ( Seilkraft>=0 ),
    Verformungsmodul
        INTEGER, -- Newton_pro_Quadratmillimeter
    CHECK ( Verformungsmodul>=0 ),
    Schlaglaenge
        REAL, -- Millimeter
    CHECK ( Schlaglaenge>=0 ),
    Gesamtgewicht
        REAL, -- Tonnen
    CHECK ( Gesamtgewicht>=0 ),
    Seilverfuellmittel
        VARCHAR(15),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Seilaufbau
  VARCHAR(15),
Oberflaechenbeh_des_Drahtes
  VARCHAR(15),
Oberflaechenbeh_Seilaufbau
  VARCHAR(15),
Vergusshuelse_Seilkopf
  VARCHAR(15),
Werkstoff_Vergusshuelse
  VARCHAR(15),
Seilverguss
  VARCHAR(15),
Armaturen
  VARCHAR(15),
schwingungsdaempfung_Massnahmen
  VARCHAR(15),
Bemerkungen_zu_Brueckenseile
  VARCHAR(255)
)

CREATE TABLE Abdichtungen (
  Abdichtungen_ID
    INTEGER NOT NULL PRIMARY KEY,
  Einbauort_Abdichtungen
    VARCHAR(255),
  Unterlage
    VARCHAR(15),
  Vorbereitung_der_Unterlage
    VARCHAR(15),
  Behandlung_der_Unterlage
    VARCHAR(15),
  Abdichtung
    VARCHAR(15),
  Schutzschicht
    VARCHAR(15),
  Dicke_Schutzschicht
    REAL, -- Zentimeter
  CHECK ( Dicke_Schutzschicht >= 0 ),
  Einbaujahr_Abdichtungen
    INTEGER, -- Jahr
  CHECK ( ( Einbaujahr_Abdichtungen >= 1900 ) AND ( Einbaujahr_Abdichtungen <=
2099 ) ),
  Flaechen_Abdichtungen
    REAL, -- Quadratmeter
  CHECK ( Flaechen_Abdichtungen >= 0 ),
  Hersteller_Abdichtungen
    VARCHAR(80),
  Ausfuehr_Firma_Abdichtungen
    VARCHAR(80),
  Bemerkungen_zu_Abdichtungen
    VARCHAR(255)
)

CREATE TABLE statisches_System_Tragefaehigkeit (
  statisches_System_Tragefaehigkeit_ID
    INTEGER NOT NULL PRIMARY KEY,
  stat_System_in_Bauwerksachse
    VARCHAR(15),
  stat_System_quer_zu_Bau_achse
    VARCHAR(15),
  Sonderabtragungsrichtung
    VARCHAR(15),
  Ersatzstuetzweite
    FLOAT (7),
  Tragefaehigkeit
    VARCHAR(15),
  Bemessungsfahrzeug_Hauptspur
    REAL, -- Tonnen
  CHECK ( Bemessungsfahrzeug_Hauptspur >= 0 ),
  Bemessungsfahrzeug_Nebenspur
    REAL, -- Tonnen
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CHECK ( Bemessungsfahrzeug_Nebenspur >= 0 ),
    statischer_Auslastungsgrad
        REAL, -- Prozent
    Bemessungslast
        REAL, -- Kilonewton
    CHECK ( Bemessungslast >= 0 ),
    Einstufungsjahr
        INTEGER, -- Jahr
    CHECK ( ( Einstufungsjahr >= 1900 ) AND ( Einstufungsjahr <= 2099 ) ),
    Bemerkungen
        VARCHAR(255)
)

CREATE TABLE Hauptbaustoff_Beton (
    Hauptbaustoff_Beton_ID
        INTEGER NOT NULL PRIMARY KEY,
    Hauptbaustoff
        VARCHAR(15),
    Zement
        VARCHAR(15),
    Zuschlagstoff
        VARCHAR(255),
    Betonzusatz
        VARCHAR(255),
    Betondruckfestigkeit
        INTEGER, -- Newton_pro_Quadratmillimeter
    CHECK ( Betondruckfestigkeit >= 0 ),
    Oberflaeche_Hauptbaust_Beton
        VARCHAR(15),
    Betonstahlguede
        VARCHAR(15),
    Fertigteile
        VARCHAR(15),
    Lieferfirma_Hauptbaust_Beton
        VARCHAR(80),
    Bemerkungen_Hauptbaust_Beton
        VARCHAR(255)
)

CREATE TABLE Hauptbaustoff_Stahl_Holz_Kunst (
    Hauptbaustoff_Stahl_Holz_Kunst_ID
        INTEGER NOT NULL PRIMARY KEY,
    Hauptbaustoff_Stahl
        VARCHAR(80),
    Konstruktionsstahlguede
        VARCHAR(80),
    Verbindungsmittel_zum_Stahl
        VARCHAR(80),
    Lieferfirma_Stahl
        VARCHAR(80),
    Bemerkungen_zum_Stahl
        VARCHAR(255)
)

CREATE TABLE Hauptbst_SHK__hat_Korrosschutz (
    Hauptbaustoff_Stahl_Holz_Kunst
        INTEGER NOT NULL REFERENCES Hauptbaustoff_Stahl_Holz_Kunst,
    Korrosionsschutzbeschichtungen
        INTEGER NOT NULL REFERENCES Korrosionsschutzbeschichtungen
)

CREATE TABLE Hauptbaustoff_Verbundwerkst (
    Hauptbaustoff_Verbundwerkst_ID
        INTEGER NOT NULL PRIMARY KEY,
    Hauptbaustoff_Verbundwerkst
        VARCHAR(15),
    Verbindungsmittel
        VARCHAR(15),
    Lieferfirma_Verbundwerkstoff
        VARCHAR(80),
    Bemerkungen_zum_Verbundwerkst

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        VARCHAR(255)
    )

CREATE TABLE Gestaltungen (
    Gestaltungen_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_Gestaltung
        VARCHAR(255),
    Einbauort_Gestaltung
        VARCHAR(255),
    Bemerkungen_zur_Gestaltung
        VARCHAR(255)
)

CREATE TABLE Leitungen_an_Bauwerken (
    Leitungen_an_Bauwerken_ID
        INTEGER NOT NULL PRIMARY KEY,
    Leitungsarten_an_Bauwerken
        VARCHAR(15),
    Einbauort_Leitungen
        VARCHAR(255),
    Durchmesser
        REAL, -- Millimeter
    CHECK ( Durchmesser >= 0 ),
    Betreiber_der_Leitung
        VARCHAR(255),
    Buendelung
        VARCHAR(15),
    Anzahl_der_Leitungen_an_Bw
        INTEGER,
    Nutzungsvereinbarung
        VARCHAR(15),
    Bemerkungen_zu_Leitungen_an_Bw
        VARCHAR(255)
)

CREATE TABLE Verfuellungen (
    Verfuellungen_ID
        INTEGER NOT NULL PRIMARY KEY,
    Einbauort_Verfuellung
        VARCHAR(255),
    Schaden_Schadenursache
        VARCHAR(255),
    Fuellgut_Fuellmaterial
        VARCHAR(15),
    Hersteller_Produktbezeichnung
        VARCHAR(15),
    Einbaujahr_Verfuellung
        INTEGER, -- Jahr
    CHECK ( ( Einbaujahr_Verfuellung >= 1900 ) AND ( Einbaujahr_Verfuellung <= 2099 ) ),
    Ausfuehrende_Firma_Verfuellung
        VARCHAR(80),
    Bemerkungen_zur_Verfuellung
        VARCHAR(255)
)

CREATE TABLE Betonersatz_nach_ZTV_SIB (
    Betonersatz_nach_ZTV_SIB_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_des_Betonersatzsystems
        VARCHAR(15),
    Einbauort_Betonersatzsystem
        VARCHAR(255),
    Lieferfirma_Produktbezeichnung
        VARCHAR(15),
    Ausfuehrende_Firma_Betonersatz
        VARCHAR(80),
    Bezeichnung_zum_Betonersatz
        VARCHAR(80),
    Einbaujahr_Betonersatzsystem
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER, -- Jahr
        CHECK ( ( Einbaujahr_Betonersatzsystem >= 1900 ) AND (
Einbaujahr_Betonersatzsystem <= 2099 ) ),
        Flaechе_Betonersatzsystem
        REAL, -- Quadratmeter
        CHECK ( Flaechе_Betonersatzsystem>=0 ),
        Bemerkungen_zum_Betonersatz
        VARCHAR(255)
)

CREATE TABLE Oberflschutz_nach_ZTV_SIB (
    Oberflschutz_nach_ZTV_SIB_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_des_Systems
        VARCHAR(15),
    Art_der_schuetzenden_Oberfl
        VARCHAR(15),
    Hersteller_Produktbezeichnung
        VARCHAR(15),
    Bezeichnung
        VARCHAR(80),
    Einbauort
        VARCHAR(255),
    Verbrauch
        REAL, -- Kilogramm_pro_Quadratmeter
    CHECK ( Verbrauch>=0 ),
    Ausfuehrende_Firma
        VARCHAR(80),
    Einbaujahr
        INTEGER, -- Jahr
    CHECK ( ( Einbaujahr >= 1900 ) AND ( Einbaujahr <= 2099 ) ),
    Flaechе
        REAL, -- Quadratmeter
    CHECK ( Flaechе>=0 ),
    Bemerkungen
        VARCHAR(255)
)

CREATE TABLE Korrosionsschutzbeschichtungen (
    Korrosionsschutzbeschichtungen_ID
        INTEGER NOT NULL PRIMARY KEY,
    Bauteiloberflaechе
        VARCHAR(15),
    Oberflaechenvorbereitung
        VARCHAR(15),
    Hauptbindemittel_d_Grundbesch
        VARCHAR(15),
    Hauptpigment_der_Grundbesch
        VARCHAR(15),
    Hauptbindemittel_der_Deckbesch
        VARCHAR(15),
    Hauptpigment_der_Deckbesch
        VARCHAR(15),
    Anzahl_der_Grundbeschichtungen
        INTEGER,
    Anzahl_der_Deckbeschichtungen
        INTEGER,
    Gesamtschichtdicke
        REAL, -- Mikrometer
    CHECK ( Gesamtschichtdicke>=0 ),
    Applikation
        VARCHAR(15),
    Einbauort_Schutzbeschichtung
        VARCHAR(255),
    Dichtstoffe_fuer_Fugen_o_ae
        VARCHAR(15),
    Injizierstoffe
        VARCHAR(15),
    Ausfuehrende_Firma_Besch
        VARCHAR(80),
    Bezeichnung_Schutzbeschichtung
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        VARCHAR(80),
        Einbaujahr_Korrosionsschutzb
            INTEGER, -- Jahr
        CHECK ( ( Einbaujahr_Korrosionsschutzb >= 1900 ) AND (
Einbaujahr_Korrosionsschutzb <= 2099 ) ),
        Beschichtete_Flaeche
            REAL, -- Quadratmeter
        CHECK ( Beschichtete_Flaeche>=0 ),
        Bemerkungen
            VARCHAR(255)
    )

CREATE TABLE Reaktionsharzgeb_Duennbelaege (
    Reaktionsharzgeb_Duennbelaege_ID
        INTEGER NOT NULL PRIMARY KEY,
    Einbauort
        VARCHAR(255),
    Vorbereitung_der_Unterlage
        VARCHAR(15),
    Bindemittel_der_Grundierung
        VARCHAR(15),
    Bindemittel_1_u_2_Deckschicht
        VARCHAR(15),
    Abstreumittel_ob_Deckschicht
        VARCHAR(15),
    Gesamtdicke
        REAL, -- Millimeter
    CHECK ( Gesamtdicke>=0 ),
    Flaeche
        REAL, -- Quadratmeter
    CHECK ( Flaeche>=0 ),
    Einbaujahr
        INTEGER, -- Jahr
    CHECK ( ( Einbaujahr >= 1900 ) AND ( Einbaujahr <= 2099 ) ),
    Ausfuehrende_Firma
        VARCHAR(80),
    Bemerkungen
        VARCHAR(255)
)

CREATE TABLE Tunnelbeleuchtung (
    Tunnelbeleuchtung_ID
        INTEGER NOT NULL PRIMARY KEY,
    Lichtraster
        VARCHAR(15),
    Leuchtdichte
        INTEGER, -- cd_pro_Quadratmeter
    CHECK ( Leuchtdichte>=0 ),
    Beleuchtungsstaerke
        INTEGER, -- Lux
    CHECK ( Beleuchtungsstaerke>=0 ),
    Beleuchtungsart
        VARCHAR(15),
    Lampenart
        VARCHAR(15),
    Bemerkungen_zur_Tunnelbeleucht
        VARCHAR(255)
)

CREATE TABLE Tunnellueftung (
    Tunnellueftung_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_der_Tunnellueftung
        VARCHAR(15),
    Abluftkanaele
        VARCHAR(15),
    Zuluftkanaele
        VARCHAR(15),
    Abluftkamin
        VARCHAR(15),
    Luefterdecke
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        VARCHAR(15),
        Bemerkungen_zur_Tunnellueftung
        VARCHAR(255)
    )

CREATE TABLE Tunnelsicherheit (
    Tunnelsicherheit_ID
        INTEGER NOT NULL PRIMARY KEY,
    Gefahrenqueter_zugelassen
        VARCHAR(15),
    Pannenbucht
        VARCHAR(15),
    Wendebucht
        VARCHAR(15),
    Ueberfahrt
        VARCHAR(15),
    Notrufstation
        VARCHAR(15),
    Automatische_Brandmeldeanlagen
        VARCHAR(15),
    Hydranten
        VARCHAR(15),
    Fluchtwege_Querstollen
        VARCHAR(15),
    Fluchtwege_Fluchtstollen
        VARCHAR(15),
    Fernsehen
        VARCHAR(15),
    Funk
        VARCHAR(15),
    Bemerkungen_zur_Tunnelsicherh
        VARCHAR(255)
    )

CREATE TABLE Tunnel_Verkehrseinrichtungen (
    Tunnel_Verkehrseinrichtungen_ID
        INTEGER NOT NULL PRIMARY KEY,
    Verkehrseinrichtung_vorhanden
        VARCHAR(15),
    Verkehrserfassung
        VARCHAR(15),
    LZA_vor_dem_Tunnel
        VARCHAR(15),
    Wechselverk_Zeich_v_d_Tunnel
        VARCHAR(15),
    Wechselverk_Zeichen_im_Tunnel
        VARCHAR(15),
    Fahrstreifensignale
        VARCHAR(15),
    Bemerkungen_zu_Verkehrseinr
        VARCHAR(255)
    )

CREATE TABLE Tunnel_Zentrale_Anlagen (
    Tunnel_Zentrale_Anlagen_ID
        INTEGER NOT NULL PRIMARY KEY,
    gesamte_elektrische_Leistung
        INTEGER, -- Kilowatt
    CHECK ( gesamte_elektrische_Leistung>=0 ),
    Ersatzstromversorgung
        VARCHAR(15),
    Ersatzstromversorgung_Leistung
        INTEGER, -- Kilowatt
    CHECK ( Ersatzstromversorgung_Leistung>=0 ),
    Betriebsgebaeude
        VARCHAR(15),
    Betriebsgebaeude_Anzahl
        INTEGER,
    Lueftergebaeude
        VARCHAR(15),
    Lueftergebaeude_Anzahl
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    INTEGER,  
    Hebeanlage  
    VARCHAR(15),  
    Bemerkungen_zu_zentr_Anlagen  
    VARCHAR(255)  
)
```

## **6 Fachbereich „Vermessung“**

### **6.1 Teilschema „Vermessung“**

In diesem Fachbereich wurde die Modellierung des TP2 auf Wunsch der BG OKSTRA nicht in das Datenschema übernommen. Die Modellierung der Vermessung wird im Rahmen der OKSTRA-Pflege überarbeitet.

## 7 Fachbereich „Entwurf“

### 7.1 Teilschema „Entwurf“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Entwurf“.

```
CREATE SCHEMA Entwurf

CREATE TABLE Trasse (
    Trasse_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_Netzknoten
        INTEGER REFERENCES Strassennetz.Netzknoten,
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Trasse__hat_Achse (
    Trasse
        INTEGER NOT NULL REFERENCES Trasse,
    Achse
        INTEGER NOT NULL REFERENCES Achse
)

CREATE TABLE Trasse__hat_Entwurfparameter (
    Trasse
        INTEGER NOT NULL REFERENCES Trasse,
    Entwurfparameter
        INTEGER NOT NULL REFERENCES Entwurfparameter
)

CREATE TABLE Trasse__hat_Laengsschnitt (
    Trasse
        INTEGER NOT NULL REFERENCES Trasse,
    Laengsschnitt
        INTEGER NOT NULL REFERENCES Laengsschnitt
)

CREATE TABLE Trasse__hat_Trassenkoerper (
    Trasse
        INTEGER NOT NULL REFERENCES Trasse,
    Trassenkoerper
        INTEGER NOT NULL REFERENCES Trassenkoerper
)

CREATE TABLE Trasse__hat_Sichtweiten (
    Trasse
        INTEGER NOT NULL REFERENCES Trasse,
    Sichtweiten
        INTEGER NOT NULL REFERENCES Sichtweiten
)

CREATE TABLE Achse (
    Achse_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Achselement (
    Achselement_ID
        INTEGER NOT NULL PRIMARY KEY,
    folgt_auf_Achselement
        INTEGER REFERENCES Achselement,
    Elementname

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    VARCHAR(255) NOT NULL,
    Elementtyp
        INTEGER NOT NULL REFERENCES Achselementtyp,
    Anfangsstation_rechnerisch
        REAL NOT NULL, -- Meter
    CHECK ( Anfangsstation_rechnerisch >= 0 ),
    Verwaltungsstation_Betriebskm
        REAL NOT NULL, -- Groesse
    CHECK ( Verwaltungsstation_Betriebskm >= 0 ),
    Laenge
        REAL NOT NULL, -- Meter
    CHECK ( Laenge >= 0 ),
    Richtung
        REAL NOT NULL, -- Winkel
    Parameter
        FLOAT (16) NOT NULL,
    Radius_zu_Beginn
        REAL NOT NULL, -- Meter
    CHECK ( Radius_zu_Beginn >= 0 ),
    Radius_am_Ende
        REAL NOT NULL, -- Meter
    CHECK ( Radius_am_Ende >= 0 ),
    beginnt_bei_Achselementpunkt
        INTEGER NOT NULL REFERENCES Achselementpunkt,
    endet_bei_Achselementpunkt
        INTEGER NOT NULL REFERENCES Achselementpunkt,
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften,
    gehoert_zu_Achse
        INTEGER REFERENCES Achse,
    CHECK ( Elementtyp=0 OR Elementtyp=1 OR Elementtyp=2 OR Elementtyp=12 ),
    UNIQUE ( Pos_Achse , gehoert_zu_Achse )
)

CREATE TABLE Linienelementtyp (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Linienelementtyp VALUES (0,'reserviert')
INSERT INTO Linienelementtyp VALUES (1,'Gerade')
INSERT INTO Linienelementtyp VALUES (2,'Kreisbogen, tangentialer Abschluß')
INSERT INTO Linienelementtyp VALUES (3,'Kreisbogen, 3 Punkte')
INSERT INTO Linienelementtyp VALUES (4,'Kreisbogen aus Mittelpunkt und Radius')
INSERT INTO Linienelementtyp VALUES (5,'Spline, tangentialer Abschluß')
INSERT INTO Linienelementtyp VALUES (6,'Spline')
INSERT INTO Linienelementtyp VALUES (7,'Spline-Ende, tangentialer Abschluß')
INSERT INTO Linienelementtyp VALUES (8,'Spline-Ende')
INSERT INTO Linienelementtyp VALUES (9,'Spline-Ende und -Anfang mit '+'
'Knickpunkt Spline')
INSERT INTO Linienelementtyp VALUES (10,'reserviert')
INSERT INTO Linienelementtyp VALUES (11,'reserviert')
INSERT INTO Linienelementtyp VALUES (12,'Klothoide')
INSERT INTO Linienelementtyp VALUES (13,'quadratische Parabel')
INSERT INTO Linienelementtyp VALUES (14,'kubische Parabel, langer Ast links')
INSERT INTO Linienelementtyp VALUES (15,'kubische Parabel, langer Ast rechts')

CREATE TABLE Achselementtyp (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Achselementtyp VALUES (0,'reserviert')
INSERT INTO Achselementtyp VALUES (1,'Gerade')
INSERT INTO Achselementtyp VALUES (2,'Kreisbogen, tangential')
INSERT INTO Achselementtyp VALUES (11,'Klothoide')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Achselementpunkt (
  Achselementpunkt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Achselementpunkt_ID) REFERENCES Strassennetz.Punktobjekt_stat
)

CREATE TABLE Kreuzungs_o_Einmuendungsplang (
  Kreuzungs_o_Einmuendungsplang_ID
    INTEGER NOT NULL PRIMARY KEY,
  Netzknotennummer
    VARCHAR(7) NOT NULL,
  Knotenpunktsform
    INTEGER NOT NULL REFERENCES Strassennetz.Knotenpunktsform,
  hat_DGM
    INTEGER REFERENCES DGM,
  hat_Achselementpunkt
    INTEGER NOT NULL REFERENCES Achselementpunkt,
  hat_allgemeine_Eigenschaften
    INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Kreuz_o_Einmplang__hat_Achse (
  Kreuzungs_o_Einmuendungsplang
    INTEGER NOT NULL REFERENCES Kreuzungs_o_Einmuendungsplang,
  Achse
    INTEGER NOT NULL REFERENCES Achse
)

CREATE TABLE DGM (
  DGM_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Laengsschnitt (
  Laengsschnitt_ID
    INTEGER NOT NULL PRIMARY KEY,
  abs_Abstand
    FLOAT (16),
  gehoert_zu_Achse
    INTEGER NOT NULL REFERENCES Achse,
  hat_allgemeine_Eigenschaften
    INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Laengsschnitt__hat_Breite (
  Laengsschnitt
    INTEGER NOT NULL REFERENCES Laengsschnitt,
  Breite
    INTEGER NOT NULL REFERENCES Breite
)

CREATE TABLE Laengsschn__hat_kreuz_Bauwerke (
  Laengsschnitt
    INTEGER NOT NULL REFERENCES Laengsschnitt,
  kreuzende_Bauwerke_o_baul_An1
    INTEGER NOT NULL REFERENCES kreuzende_Bauwerke_o_baul_An1
)

CREATE TABLE Laengsschn__hat_Gelaendehoriz (
  Laengsschnitt
    INTEGER NOT NULL REFERENCES Laengsschnitt,
  Gelaendehorizonte
    INTEGER NOT NULL REFERENCES Gelaendehorizonte
)

CREATE TABLE Laengsschnitt__hat_Gradiente (
  Laengsschnitt
    INTEGER NOT NULL REFERENCES Laengsschnitt,
  Gradiente
    INTEGER NOT NULL REFERENCES Gradiente
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Gelaendehorizonte (
  Gelaendehorizonte_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Gelaendehorizonte_ID) REFERENCES Laengsschnittlinie
)

CREATE TABLE Gradiente (
  Gradiente_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Gradiente_ID) REFERENCES Laengsschnittlinie
)

CREATE TABLE kreuzende_Bauwerke_o_baul_An1 (
  kreuzende_Bauwerke_o_baul_An1_ID
    INTEGER NOT NULL PRIMARY KEY,
  hat_Tunnel_Kanal_Strasse_etc
    INTEGER REFERENCES Tunnel_Kanal_Strasse_Durchlass,
  hat_allgemeine_Eigenschaften
    INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE kreuz_Bauwerk__hat_Schnittgeom (
  kreuzende_Bauwerke_o_baul_An1
    INTEGER NOT NULL REFERENCES kreuzende_Bauwerke_o_baul_An1,
  Schnittgeometrie
    INTEGER NOT NULL REFERENCES Schnittgeometrie
)

CREATE TABLE Tunnel_Kanal_Strasse_Durchlass (
  Tunnel_Kanal_Strasse_Durchlass_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Schnittgeometrie (
  Schnittgeometrie_ID
    INTEGER NOT NULL PRIMARY KEY,
  Schnittstation
    REAL NOT NULL, -- Meter
  CHECK ( Schnittstation>=0 ),
  Schnitthoehe
    REAL NOT NULL, -- Meter
  CHECK ( Schnitthoehe>=0 ),
  Schnittwinkel_horizontal
    REAL NOT NULL, -- Winkel
  Laengsneigung_kreuzend_Bauwerk
    REAL NOT NULL, -- Winkel
  abgel_von_Tunnel_Kanal_Str_etc
    INTEGER NOT NULL REFERENCES Tunnel_Kanal_Strasse_Durchlass
)

CREATE TABLE Schnittgeom__hat_Schnpolygone (
  Schnittgeometrie
    INTEGER NOT NULL REFERENCES Schnittgeometrie,
  Schnittpolygone
    INTEGER NOT NULL REFERENCES Schnittpolygone
)

CREATE TABLE Schnittgeom__hat_Mindabstpolyg (
  Schnittgeometrie
    INTEGER NOT NULL REFERENCES Schnittgeometrie,
  Mindestabstandspolygon
    INTEGER NOT NULL REFERENCES Mindestabstandspolygon
)

CREATE TABLE Schnittgeom__hat_Maxabstpolyg (
  Schnittgeometrie
    INTEGER NOT NULL REFERENCES Schnittgeometrie,
  Maximalabstandspolygon
    INTEGER NOT NULL REFERENCES Maximalabstandspolygon
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Schnittpolygone (
  Schnittpolygone_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Schnittpolygone_ID) REFERENCES Polygon
)

CREATE TABLE Mindestabstandspolygon (
  Mindestabstandspolygon_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Mindestabstandspolygon_ID) REFERENCES Polygon
)

CREATE TABLE Maximalabstandspolygon (
  Maximalabstandspolygon_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Maximalabstandspolygon_ID) REFERENCES Polygon
)

CREATE TABLE Polygon (
  Polygon_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Polygon__hat_Polygonpunkte (
  Polygon
    INTEGER NOT NULL REFERENCES Polygon,
  Polygonpunkt
    INTEGER NOT NULL REFERENCES Polygonpunkt,
  folgt_auf_Polygonpunkt
    INTEGER REFERENCES Polygonpunkt
)

CREATE TABLE Polygonpunkt (
  Polygonpunkt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Polygonpunkt_ID) REFERENCES Geometrieschema.Punkt,
  Abstand_zum_Bezugspunkt
    REAL NOT NULL, -- Meter
  CHECK ( Abstand_zum_Bezugspunkt >= 0 ),
  Hoehe_zum_Bezugspunkt
    REAL NOT NULL, -- Meter
  CHECK ( Hoehe_zum_Bezugspunkt >= 0 )
)

CREATE TABLE Laengsschnittlinie (
  Laengsschnittlinie_ID
    INTEGER NOT NULL PRIMARY KEY,
  hat_Spurbezeichnung
    INTEGER REFERENCES Spurbezeichnung,
  hat_DGM
    INTEGER REFERENCES DGM,
  hat_allgemeine_Eigenschaften
    INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Laengsschnittl__hat_LS_Koor (
  Laengsschnittlinie
    INTEGER NOT NULL REFERENCES Laengsschnittlinie,
  LS_Koor
    INTEGER NOT NULL REFERENCES LS_Koor,
  folgt_auf_LS_Koor
    INTEGER REFERENCES LS_Koor
)

CREATE TABLE LS_Koor (
  LS_Koor_ID
    INTEGER NOT NULL PRIMARY KEY,
  Station
    REAL NOT NULL, -- Meter
  CHECK ( Station >= 0 ),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Hoehe
    REAL NOT NULL, -- Meter
CHECK ( Hoehe>=0 ),
folgt_auf_LS_Koor
    INTEGER REFERENCES Punktfolge,
hat_allgemeine_Eigenschaften
    INTEGER REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Punktfolge (
    Punktfolge_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_Tangente_Gerade
        INTEGER NOT NULL REFERENCES Tangente_Gerade
)

CREATE TABLE Tangentenfolge (
    Tangentenfolge_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_Ausrundung
        INTEGER REFERENCES Ausrundung
)

CREATE TABLE Ausrundung (
    Ausrundung_ID
        INTEGER NOT NULL PRIMARY KEY,
    Ausrundungstyp
        INTEGER REFERENCES Ausrundungstyp,
    Ausrundungsparameter
        FLOAT (16)
)

CREATE TABLE Ausrundungstyp (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Ausrundungstyp VALUES (12,'Parabel 2. Grades')
INSERT INTO Ausrundungstyp VALUES (13,'Parabel 3. Grades mit langem Teil '+'
'vor TS')
INSERT INTO Ausrundungstyp VALUES (14,'Parabel 3. Grades mit langem Teil '+'
'hinter TS')

CREATE TABLE Tangente_Gerade (
    Tangente_Gerade_ID
        INTEGER NOT NULL PRIMARY KEY,
    folgt_auf_Tangente_Gerade
        INTEGER REFERENCES Tangentenfolge
)

CREATE TABLE Volumen_Massenermittlung (
    Volumen_Massenermittlung_ID
        INTEGER NOT NULL PRIMARY KEY,
    Station_1
        REAL NOT NULL, -- Meter
CHECK ( Station_1>=0 ),
    Station_2
        REAL NOT NULL, -- Meter
CHECK ( Station_2>=0 ),
    wird_begrenzt_von_Profillinie
        INTEGER NOT NULL REFERENCES Profillinien,
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Oberflaeche (
    Oberflaeche_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_der_Oberflaeche
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER NOT NULL,
    Station_1
        REAL NOT NULL, -- Meter
    CHECK ( Station_1 >= 0 ),
    Station_2
        REAL NOT NULL, -- Meter
    CHECK ( Station_2 >= 0 ),
    liegt_auf_Profillinie
        INTEGER NOT NULL REFERENCES Profillinien,
    beginnt_bei_QP_Punkt
        INTEGER NOT NULL REFERENCES QP_Punkte,
    endet_bei_QP_Punkt
        INTEGER NOT NULL REFERENCES QP_Punkte,
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Trassenkoerper (
    Trassenkoerper_ID
        INTEGER NOT NULL PRIMARY KEY,
    gehoert_zu_Achse
        INTEGER NOT NULL REFERENCES Achse
)

CREATE TABLE Trassenkoerper__hat_Volumen (
    Trassenkoerper
        INTEGER NOT NULL REFERENCES Trassenkoerper,
    Volumen_Massenermittlung
        INTEGER NOT NULL REFERENCES Volumen_Massenermittlung
)

CREATE TABLE Trassenkoerper__hat_Oberfl (
    Trassenkoerper
        INTEGER NOT NULL REFERENCES Trassenkoerper,
    Oberflaeche
        INTEGER NOT NULL REFERENCES Oberflaeche
)

CREATE TABLE Trassenkoerper__hat_Deckenbuch (
    Trassenkoerper
        INTEGER NOT NULL REFERENCES Trassenkoerper,
    Deckenbuch
        INTEGER NOT NULL REFERENCES Deckenbuch
)

CREATE TABLE Trassenkoerper__hat_Querprofil (
    Trassenkoerper
        INTEGER NOT NULL REFERENCES Trassenkoerper,
    Querprofil
        INTEGER NOT NULL REFERENCES Querprofil
)

CREATE TABLE Trassenkoerper__hat_Ausgdaten (
    Trassenkoerper
        INTEGER NOT NULL REFERENCES Trassenkoerper,
    Ausgangsdaten
        INTEGER NOT NULL REFERENCES Ausgangsdaten
)

CREATE TABLE Trassenkoerper__hat_Bildgsges (
    Trassenkoerper
        INTEGER NOT NULL REFERENCES Trassenkoerper,
    Bildungsgesetze
        INTEGER NOT NULL REFERENCES Bildungsgesetze
)

CREATE TABLE Querprofil (
    Querprofil_ID
        INTEGER NOT NULL PRIMARY KEY,
    Station
        REAL NOT NULL, -- Meter
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    CHECK ( Station>=0 ),
    hat_Schnittebene
        INTEGER REFERENCES Schnittebene,
    folgt_auf_Querprofil
        INTEGER REFERENCES Querprofil
)

CREATE TABLE Querpr__von_Ausgdaten_Bildges (
    Querprofil
        INTEGER NOT NULL REFERENCES Querprofil,
    Ausgangsdaten_Bildungsgesetze
        INTEGER NOT NULL REFERENCES Ausgangsdaten_Bildungsgesetze
)

CREATE TABLE Querprofil__hat_Profillinien (
    Querprofil
        INTEGER NOT NULL REFERENCES Querprofil,
    Profillinien
        INTEGER NOT NULL REFERENCES Profillinien
)

CREATE TABLE Querprofil__hat_QP_Punkte (
    Querprofil
        INTEGER NOT NULL REFERENCES Querprofil,
    QP_Punkte
        INTEGER NOT NULL REFERENCES QP_Punkte
)

CREATE TABLE Bildungsgesetze (
    Bildungsgesetze_ID
        INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Ausgangsdaten_Bildungsgesetze (
    Ausgangsdaten_Bildungsgesetze_ID
        INTEGER NOT NULL PRIMARY KEY,
    Bildungsgesetze
        INTEGER NOT NULL REFERENCES Bildungsgesetze,
    Ausgangsdaten
        INTEGER NOT NULL REFERENCES Ausgangsdaten
)

CREATE TABLE Ausgangsdaten (
    Ausgangsdaten_ID
        INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Ausgangsdaten__hat_Querprofil (
    Ausgangsdaten
        INTEGER NOT NULL REFERENCES Ausgangsdaten,
    Querprofil
        INTEGER NOT NULL REFERENCES Querprofil
)

CREATE TABLE Ausgangsdaten__hat_Breite (
    Ausgangsdaten
        INTEGER NOT NULL REFERENCES Ausgangsdaten,
    Breite
        INTEGER NOT NULL REFERENCES Breite
)

CREATE TABLE Ausgangsdaten__hat_Querneigung (
    Ausgangsdaten
        INTEGER NOT NULL REFERENCES Ausgangsdaten,
    Querneigung
        INTEGER NOT NULL REFERENCES Querneigung
)

CREATE TABLE Ausgangsdaten__hat_Hoehenzuege (
    Ausgangsdaten
        INTEGER NOT NULL REFERENCES Ausgangsdaten,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    Hoehenzuege
    INTEGER NOT NULL REFERENCES Hoehenzuege
)

CREATE TABLE Ausgangsd__hat_Achsstationsw (
    Ausgangsdaten
    INTEGER NOT NULL REFERENCES Ausgangsdaten,
    Achsstationswerte
    INTEGER NOT NULL REFERENCES Achsstationswerte
)

CREATE TABLE Hoehenzuege (
    Hoehenzuege_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE QP_Punkte (
    QP_Punkte_ID
    INTEGER NOT NULL PRIMARY KEY,
    Abstand_zur_Achse
    FLOAT (16) NOT NULL,
    Hoehe_NN
    FLOAT (16) NOT NULL,
    hat_Querprofilpunkt
    INTEGER REFERENCES Querprofilpunkt,
    hat_allgemeine_Eigenschaften
    INTEGER REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Querprofilpunkt (
    Querprofilpunkt_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Querprofilpunkt_ID) REFERENCES Strassennetz.Punktobjekt_stat
)

CREATE TABLE Profillinien (
    Profillinien_ID
    INTEGER NOT NULL PRIMARY KEY,
    hat_allgemeine_Eigenschaften
    INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Profillinien__hat_QP_Punkte (
    Profillinien
    INTEGER NOT NULL REFERENCES Profillinien,
    QP_Punkte
    INTEGER NOT NULL REFERENCES QP_Punkte,
    folgt_auf_QP_Punkt
    INTEGER REFERENCES QP_Punkte
)

CREATE TABLE Schnittebene (
    Schnittebene_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Schnittebene__hat_SNT_Punkte (
    Schnittebene
    INTEGER NOT NULL REFERENCES Schnittebene,
    SNT_Punkt
    INTEGER NOT NULL REFERENCES SNT_Punkt,
    folgt_auf_SNT_Punkt
    INTEGER REFERENCES SNT_Punkt
)

CREATE TABLE SNT_Punkt (
    SNT_Punkt_ID
    INTEGER NOT NULL PRIMARY KEY,
    Abstand
    REAL NOT NULL, -- Meter
    CHECK ( Abstand >= 0 ),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Richtungsaenderung
    FLOAT (16) NOT NULL,
korrespondierende_Achse
    VARCHAR(255),
Naehungsstation
    REAL, -- Meter
CHECK ( Naehungsstation>=0 )
)

CREATE TABLE Deckenbuch (
    Deckenbuch_ID
        INTEGER NOT NULL PRIMARY KEY,
    Station_1
        REAL NOT NULL, -- Meter
CHECK ( Station_1>=0 ),
    Station_2
        REAL NOT NULL, -- Meter
CHECK ( Station_2>=0 ),
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Deckenb__hat_Spur_a_Ausgangsdat (
    Deckenbuch
        INTEGER NOT NULL REFERENCES Deckenbuch,
    Spur_aus_Ausgangsdaten
        INTEGER NOT NULL REFERENCES Spur_aus_Ausgangsdaten,
    folgt_auf_Spur_aus_Ausgangsdaten
        INTEGER REFERENCES Spur_aus_Ausgangsdaten
)

CREATE TABLE Deckenb__hat_Spur_aus_Querprofil (
    Deckenbuch
        INTEGER NOT NULL REFERENCES Deckenbuch,
    Spur_aus_Querprofilen
        INTEGER NOT NULL REFERENCES Spur_aus_Querprofilen,
    folgt_auf_Spur_aus_Querprofil
        INTEGER REFERENCES Spur_aus_Querprofilen
)

CREATE TABLE Spurfolge_Ausgangsdaten (
    Spurfolge_Ausgangsdaten_ID
        INTEGER NOT NULL PRIMARY KEY,
    Hoehendifferenz
        REAL NOT NULL, -- Meter
CHECK ( Hoehendifferenz>=0 )
)

CREATE TABLE Spur_aus_Ausgangsdaten (
    Spur_aus_Ausgangsdaten_ID
        INTEGER NOT NULL PRIMARY KEY,
    folgt_auf_Spur_aus_Ausgangsdat
        INTEGER REFERENCES Spurfolge_Ausgangsdaten,
    hat_Querneigung
        INTEGER REFERENCES Querneigung,
    hat_Breite
        INTEGER NOT NULL REFERENCES Breite,
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Spur_a_Ausgangsdaten__hat_Hoehenz (
    Spur_aus_Ausgangsdaten
        INTEGER NOT NULL REFERENCES Spur_aus_Ausgangsdaten,
    Hoehenzuege
        INTEGER NOT NULL REFERENCES Hoehenzuege
)

CREATE TABLE Spur_aus_Querprofilen (
    Spur_aus_Querprofilen_ID
        INTEGER NOT NULL PRIMARY KEY,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
liegt_auf_Profillinie
    INTEGER NOT NULL REFERENCES Profillinien,
beginnt_bei_QP_Punkt
    INTEGER NOT NULL REFERENCES QP_Punkte,
endet_bei_QP_Punkt
    INTEGER NOT NULL REFERENCES QP_Punkte,
hat_allgemeine_Eigenschaften
    INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Breite (
    Breite_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_Spurbezeichnung
        INTEGER REFERENCES Spurbezeichnung,
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Breite__hat_BR_Punkt (
    Breite
        INTEGER NOT NULL REFERENCES Breite,
    BR_Punkt
        INTEGER NOT NULL REFERENCES BR_Punkt,
    folgt_auf_BR_Punkt
        INTEGER REFERENCES BR_Punkt
)

CREATE TABLE BR_Punktfolge (
    BR_Punktfolge_ID
        INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE BR_Punkt__hat_Aufw_Verbr_Verb (
    BR_Punktfolge
        INTEGER NOT NULL REFERENCES BR_Punktfolge,
    Aufweitung_Verbreit_Verbind
        INTEGER NOT NULL REFERENCES Aufweitung_Verbreit_Verbind
)

CREATE TABLE BR_Punkt (
    BR_Punkt_ID
        INTEGER NOT NULL PRIMARY KEY,
    Breite
        REAL, -- Meter
    CHECK ( Breite>=0 ),
    Station
        REAL NOT NULL, -- Meter
    CHECK ( Station>=0 ),
    hat_Abstand_Achse_Achse
        INTEGER REFERENCES Abstand_Achse_Achse,
    hat_Abstand_Achse_Linie
        INTEGER REFERENCES Abstand_Achse_Linie,
    folgt_auf_BR_Punkt
        INTEGER REFERENCES BR_Punktfolge
)

CREATE TABLE Aufweitung_Verbreit_Verbind (
    Aufweitung_Verbreit_Verbind_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_Parabelfolge_zweit_Grades
        INTEGER REFERENCES Parabelfolge_zweiten_Grades,
    hat_Bogenfolge
        INTEGER REFERENCES Bogenfolge,
    hat_Parabelf_zw_Grad_zw_Gerade
        INTEGER REFERENCES Parabelf_zw_Grades_zw_Gerade,
    hat_Gerade
        INTEGER REFERENCES Gerade
)

CREATE TABLE Parabelfolge_zweiten_Grades (
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    Parabelfolge_zweiten_Grades_ID
    INTEGER NOT NULL PRIMARY KEY
)
CREATE TABLE Bogenfolge (
    Bogenfolge_ID
    INTEGER NOT NULL PRIMARY KEY
)
CREATE TABLE Parabelf_zw_Grades_zw_Gerade (
    Parabelf_zw_Grades_zw_Gerade_ID
    INTEGER NOT NULL PRIMARY KEY,
    Tangentenlaenge
    REAL NOT NULL, -- Meter
    CHECK ( Tangentenlaenge>=0 )
)
CREATE TABLE Gerade (
    Gerade_ID
    INTEGER NOT NULL PRIMARY KEY
)
CREATE TABLE Abstand_Achse_Achse (
    Abstand_Achse_Achse_ID
    INTEGER NOT NULL PRIMARY KEY,
    hat_Lage_der_Knicklinie
    INTEGER NOT NULL REFERENCES Lage_der_Knicklinie,
    hat_zweite_Achse
    INTEGER NOT NULL REFERENCES zweite_Achse,
    hat_Naehergsstat_auf_zw_Achse
    INTEGER NOT NULL REFERENCES Naehungsstation_auf_zw_Achse
)
CREATE TABLE Abstand_Achse_Linie (
    Abstand_Achse_Linie_ID
    INTEGER NOT NULL PRIMARY KEY,
    hat_Linie
    INTEGER NOT NULL REFERENCES Geometrieschema.Linie
)
CREATE TABLE Lage_der_Knicklinie (
    Lage_der_Knicklinie_ID
    INTEGER NOT NULL PRIMARY KEY,
    hat_Parall_Breite_z_Achse_zwei
    INTEGER REFERENCES Parallele_Breite_z_Achse_zwei,
    hat_Parall_Breite_z_Achse_eins
    INTEGER REFERENCES Parallele_Breite_z_Achse_eins,
    ist_mittig_m_Abst_z_Knicklinie
    INTEGER REFERENCES mittig_m_Abstand_z_Knicklinie
)
CREATE TABLE zweite_Achse (
    zweite_Achse_ID
    INTEGER NOT NULL PRIMARY KEY
)
CREATE TABLE Naehungsstation_auf_zw_Achse (
    Naehungsstation_auf_zw_Achse_ID
    INTEGER NOT NULL PRIMARY KEY
)
CREATE TABLE Parallele_Breite_z_Achse_zwei (
    Parallele_Breite_z_Achse_zwei_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Parallele_Breite_z_Achse_zwei_ID) REFERENCES Breite
)
CREATE TABLE Parallele_Breite_z_Achse_eins (
    Parallele_Breite_z_Achse_eins_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Parallele_Breite_z_Achse_eins_ID) REFERENCES Breite
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

CREATE TABLE mittig_m_Abstand_z_Knicklinie (
    mittig_m_Abstand_z_Knicklinie_ID
        INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Querneigung (
    Querneigung_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_Spurbezeichnung
        INTEGER REFERENCES Spurbezeichnung,
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Querneigung__hat_QN_Punkte (
    Querneigung
        INTEGER NOT NULL REFERENCES Querneigung,
    QN_Punkt
        INTEGER NOT NULL REFERENCES QN_Punkt,
    folgt_auf_QN_Punkt
        INTEGER REFERENCES QN_Punkt
)

CREATE TABLE Querneigungswechsel (
    Querneigungswechsel_ID
        INTEGER NOT NULL PRIMARY KEY,
    Verziehungsform
        INTEGER NOT NULL REFERENCES Verziehungsform
)

CREATE TABLE Verziehungsform (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Verziehungsform VALUES (0,'reserviert')
INSERT INTO Verziehungsform VALUES (1,'normale Verziehung')
INSERT INTO Verziehungsform VALUES (2,'Verziehung mit Gratlinie '+
    '(Schrägverwindung)')

CREATE TABLE QN_Punkt (
    QN_Punkt_ID
        INTEGER NOT NULL PRIMARY KEY,
    Pos_Querneigung
        INTEGER NOT NULL,
    Station
        REAL NOT NULL, -- Meter
    CHECK ( Station>=0 ),
    hat_allgemeine_Eigenschaften
        INTEGER REFERENCES allgemeine_Eigenschaften,
    folgt_auf_QN_Punkt
        INTEGER REFERENCES Querneigungswechsel,
    gehoert_zu_Querneigung
        INTEGER NOT NULL REFERENCES Querneigung,
    UNIQUE ( Pos_Querneigung , gehoert_zu_Querneigung )
)

CREATE TABLE Spurbezeichnung (
    Spurbezeichnung_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Achsstationswerte (
    Achsstationswerte_ID
        INTEGER NOT NULL PRIMARY KEY,
```

# OKSTRA – Teilprojekt 4 – Schlußbericht

## Teilbericht D - Das SQL-Datenschema

---

```
    hat_allgemeine_Eigenschaften
        INTEGER REFERENCES allgemeine_Eigenschaften
    )

CREATE TABLE Achsstationswerte__hat_AW_Koor (
    Achsstationswerte
        INTEGER NOT NULL REFERENCES Achsstationswerte,
    AW_Koor
        INTEGER NOT NULL REFERENCES AW_Koor,
    folgt_auf_AW_Koor
        INTEGER REFERENCES AW_Koor
    )

CREATE TABLE AW_Koor (
    AW_Koor_ID
        INTEGER NOT NULL PRIMARY KEY,
    Station
        REAL NOT NULL, -- Meter
    CHECK ( Station>=0 ),
    Character_Kennwert
        VARCHAR(255) NOT NULL,
    Integer_Kennwert
        INTEGER NOT NULL,
    Real_Kennwert
        REAL NOT NULL
    )

CREATE TABLE Entwurfparameter (
    Entwurfparameter_ID
        INTEGER NOT NULL PRIMARY KEY,
    Strassenkategorie
        VARCHAR(255) NOT NULL,
    Verkehrsbelastung
        INTEGER NOT NULL,
    Dokumentenverweis
        VARCHAR(255) NOT NULL,
    Regelquerschnitt
        VARCHAR(255) NOT NULL
    )

CREATE TABLE Entwurfsparam__hat_Geschwband (
    Entwurfparameter
        INTEGER NOT NULL REFERENCES Entwurfparameter,
    Geschwindigkeitsband
        INTEGER NOT NULL REFERENCES Geschwindigkeitsband
    )

CREATE TABLE Geschwindigkeitsband (
    Geschwindigkeitsband_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
    )

CREATE TABLE Geschwindigkeitsb__hat_V_Koor (
    Geschwindigkeitsband
        INTEGER NOT NULL REFERENCES Geschwindigkeitsband,
    V_Koor
        INTEGER NOT NULL REFERENCES V_Koor,
    folgt_auf_V_Koor
        INTEGER REFERENCES V_Koor
    )

CREATE TABLE V_Koor (
    V_Koor_ID
        INTEGER NOT NULL PRIMARY KEY,
    Station
        REAL NOT NULL, -- Meter
    CHECK ( Station>=0 ),
    Geschwindigkeit
        INTEGER NOT NULL, -- Stundenkilometer
    )
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    CHECK ( Geschwindigkeit >= 0 )
)

CREATE TABLE Sichtweiten (
    Sichtweiten_ID
        INTEGER NOT NULL PRIMARY KEY,
    Sichtweite
        FLOAT (16) NOT NULL,
    gehoert_zu_Achse
        INTEGER NOT NULL REFERENCES Achse,
    hat_Spurbezeichnung
        INTEGER REFERENCES Spurbezeichnung,
    hat_allgemeine_Eigenschaften
        INTEGER NOT NULL REFERENCES allgemeine_Eigenschaften
)

CREATE TABLE Sichtweiten__hat_SW_Koor (
    Sichtweiten
        INTEGER NOT NULL REFERENCES Sichtweiten,
    SW_Koor
        INTEGER NOT NULL REFERENCES SW_Koor,
    folgt_auf_SW_Koor
        INTEGER REFERENCES SW_Koor
)

CREATE TABLE SW_Koor (
    SW_Koor_ID
        INTEGER NOT NULL PRIMARY KEY,
    Station
        REAL NOT NULL, -- Meter
    CHECK ( Station >= 0 ),
    Sichtweite
        REAL NOT NULL, -- Meter
    CHECK ( Sichtweite >= 0 )
)

CREATE TABLE allgemeine_Eigenschaften (
    allgemeine_Eigenschaften_ID
        INTEGER NOT NULL PRIMARY KEY,
    Bezeichnung
        VARCHAR(255) NOT NULL,
    fachliche_Bedeutung
        VARCHAR(255) NOT NULL,
    Informationstext
        VARCHAR(255)
)
)
```

## **8 Fachbereich „Ökologie“**

### **8.1 Teilschema „Ökologie“**

In diesem Fachbereich wurde vom Teilprojekt 2 keine NIAM-Modellierung durchgeführt, daher konnte auch keine SQL-Modellierung erfolgen.

## 9 Fachbereich „Ingenieurbauwerke“

### 9.1 Teilschema „Ingenieurbauwerke“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Ingenieurbauwerke“.

```
CREATE SCHEMA Ingenieurbauwerke
```

```
CREATE TABLE Bauwerkseinzelheiten (
  Bauwerkseinzelheiten_ID
    INTEGER NOT NULL PRIMARY KEY,
  Bauwerksnummer
    INTEGER NOT NULL,
  Bauwerksname
    VARCHAR(255) NOT NULL,
  Bau_km
    REAL NOT NULL, -- Kilometer
  CHECK ( Bau_km >= 0 ),
  Brueckenklasse
    VARCHAR(255),
  Kreuzungswinkel
    REAL, -- Winkel
  lichte_Weite
    REAL, -- Meter
  CHECK ( lichte_Weite >= 0 ),
  lichte_Hoehe
    REAL, -- Meter
  CHECK ( lichte_Hoehe >= 0 ),
  Konstruktionshoehe
    REAL, -- Meter
  CHECK ( Konstruktionshoehe >= 0 ),
  Nutzbreite
    REAL, -- Meter
  CHECK ( Nutzbreite >= 0 ),
  hat_allgemeine_Eigenschaften
    INTEGER NOT NULL REFERENCES Entwurf.allgemeine_Eigenschaften
)
```

## 10 Fachbereich „Projektressourcen“

### 10.1 Teilschema „Projektressourcen“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Projektressourcen“.

```
CREATE SCHEMA Projektressourcen
```

```
CREATE TABLE Projektressourcen (
  Projektressourcen_ID
    INTEGER NOT NULL PRIMARY KEY,
  Bauanfang_Bau_km
    REAL NOT NULL, -- Kilometer
  CHECK ( Bauanfang_Bau_km>=0 ),
  endet_bei_Bauende
    REAL NOT NULL, -- Kilometer
  CHECK ( endet_bei_Bauende>=0 ),
  Ausbaulaenge
    REAL NOT NULL, -- Kilometer
  CHECK ( Ausbaulaenge>=0 ),
  ist_Kostenberechnung_nach_AKS
    INTEGER REFERENCES Kostenberechnung_nach_AKS,
  ist_Ausschreibung_und_Vergabe
    INTEGER REFERENCES Ausschreibung_und_Vergabe,
  ist_Abrechnung
    INTEGER REFERENCES Abrechnung
)

CREATE TABLE Kostenberechnung_nach_AKS (
  Kostenberechnung_nach_AKS_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Kostenberechnung_nach_AKS_ID) REFERENCES Dokumentenverweis
)

CREATE TABLE Ausschreibung_und_Vergabe (
  Ausschreibung_und_Vergabe_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Ausschreibung_und_Vergabe_ID) REFERENCES Dokumentenverweis
)

CREATE TABLE Abrechnung (
  Abrechnung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Abrechnung_ID) REFERENCES Dokumentenverweis
)

CREATE TABLE Dokumentenverweis (
  Dokumentenverweis_ID
    INTEGER NOT NULL PRIMARY KEY
)
```

## 11 Fachbereich „Ausstattung“

### 11.1 Teilschema „Ausstattung“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Ausstattung“.

```
CREATE SCHEMA Ausstattung
```

```
CREATE TABLE Beschilderung_Lichtsignalanl (
  Beschilderung_Lichtsignalanl_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Beschilderung_Lichtsignalanl_ID) REFERENCES
  Strassennetz.Punktobjekt_o_Bereichsobjekt,
  hat_allgemeine_Eigenschaften
    INTEGER NOT NULL REFERENCES Entwurf.allgemeine_Eigenschaften
)
```

```
CREATE TABLE Fahrbahnmarkierungen (
  Fahrbahnmarkierungen_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Fahrbahnmarkierungen_ID) REFERENCES
  Strassennetz.Punktobjekt_o_Bereichsobjekt,
  hat_allgemeine_Eigenschaften
    INTEGER NOT NULL REFERENCES Entwurf.allgemeine_Eigenschaften
)
```

```
CREATE TABLE Schutz_und_Leiteinrichtungen (
  Schutz_und_Leiteinrichtungen_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Schutz_und_Leiteinrichtungen_ID) REFERENCES
  Strassennetz.Punktobjekt_o_Bereichsobjekt,
  hat_allgemeine_Eigenschaften
    INTEGER NOT NULL REFERENCES Entwurf.allgemeine_Eigenschaften
)
```

## 12 Fachbereich „Verkehrsdaten“

### 12.1 Teilschema „Zählstellen / dynamische Verkehrsdaten“

Dieses Teilschema enthält die SQL-Definitionen für die Teilbereiche „Automatische Dauerzählstelle“, „Verkehrsstärke“, „Erfassung Geschwindigkeiten“, „Achslastdaten“, „Einzelfahrzeugdaten“ und „Manuelle Zählstelle“.

```
CREATE SCHEMA Automatische_Dauerzaehlstelle

CREATE TABLE automatische_Dauerzaehlstelle (
  automatische_Dauerzaehlstelle_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (automatische_Dauerzaehlstelle_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  TK25_Nummer
    INTEGER NOT NULL, -- TK25_Blattnummer
  CHECK ( ( TK25_Nummer >= 1 ) AND ( TK25_Nummer <= 9999 ) ),
  BAST_Nummer
    INTEGER NOT NULL,
  Name
    VARCHAR(50) NOT NULL,
  hat_Landesnummer
    INTEGER REFERENCES Landesnummer,
  hat_verwaltungstechn_Zuordnung
    INTEGER NOT NULL REFERENCES verwaltungstechn_Zuordnung_AD,
  hat_Moeglichk_z_Einzelfahrzerf
    INTEGER REFERENCES Moegl_Einzelfahrzeugetfassung,
  hat_oertliche_Zuordnung
    INTEGER NOT NULL REFERENCES oertliche_Zuordnung_AD
)

CREATE TABLE autom_Dauzaehl__Bezugszaehlst (
  automatische_Dauerzaehlstelle
    INTEGER NOT NULL REFERENCES automatische_Dauerzaehlstelle,
  Bezugszaehlstellen
    INTEGER NOT NULL
)

CREATE TABLE autom_Dauzaehl__hat_Erfasserg (
  automatische_Dauerzaehlstelle
    INTEGER NOT NULL REFERENCES automatische_Dauerzaehlstelle,
  Erfassungsergebnisse
    INTEGER NOT NULL REFERENCES Erfassungsergebnisse
)

CREATE TABLE autom_Dauzaehl__hat_Erfasschar (
  automatische_Dauerzaehlstelle
    INTEGER NOT NULL REFERENCES automatische_Dauerzaehlstelle,
  Erfassungscharakteristik
    INTEGER NOT NULL REFERENCES Erfassungscharakteristik
)

CREATE TABLE autom_Dauzaehl__hat_Detektoren (
  automatische_Dauerzaehlstelle
    INTEGER NOT NULL REFERENCES automatische_Dauerzaehlstelle,
  Detektoren
    INTEGER NOT NULL REFERENCES Detektoren
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE autom_Dauzaehl__geh_zu_ZU_VRZ (
    automatische_Dauerzaehlstelle
        INTEGER NOT NULL REFERENCES automatische_Dauerzaehlstelle,
    Unterzentrale_Verkehrsrechnerz
        INTEGER NOT NULL REFERENCES Unterzentrale_Verkehrsrechnerz
)

CREATE TABLE Erfassungsergebnisse (
    Erfassungsergebnisse_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Erfassungsergebnisse_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Ergebnisse
        VARCHAR(99) NOT NULL
)

CREATE TABLE Erfassungscharakteristik (
    Erfassungscharakteristik_ID
        INTEGER NOT NULL PRIMARY KEY,
    Beginn_der_Zaehlung
        CHAR(10) NOT NULL, -- Datum
    CHECK ( Beginn_der_Zaehlung LIKE '___.__.____' ),
    hat_Fahrzeugartenunterscheidg
        INTEGER NOT NULL REFERENCES Fahrzeugartenunterscheidung,
    hat_Differenzierg_d_Erfassung
        INTEGER NOT NULL REFERENCES Differenzierung_der_Erfassung
)

CREATE TABLE Erfcharakt__Zaehlintervlaenge (
    Erfassungscharakteristik
        INTEGER NOT NULL REFERENCES Erfassungscharakteristik,
    Zaehlintervallaenge
        INTEGER NOT NULL REFERENCES Zaehlintervallaenge
)

CREATE TABLE Erfcharakt__hat_Art_d_reg_Dat (
    Erfassungscharakteristik
        INTEGER NOT NULL REFERENCES Erfassungscharakteristik,
    Art_der_registrierten_Daten
        INTEGER NOT NULL REFERENCES Art_der_registrierten_Daten
)

CREATE TABLE Zaehlintervallaenge (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Zaehlintervallaenge VALUES (1,'60 min')
INSERT INTO Zaehlintervallaenge VALUES (2,'30 min')
INSERT INTO Zaehlintervallaenge VALUES (3,'15 min')
INSERT INTO Zaehlintervallaenge VALUES (4,'1 min')
INSERT INTO Zaehlintervallaenge VALUES (5,'5 min')

CREATE TABLE Fahrzeugartenunterscheidung (
    Fahrzeugartenunterscheidung_ID
        INTEGER NOT NULL PRIMARY KEY,
    Unterscheidungsmerkmal
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean,
    Methode_der_Unterscheidung
        INTEGER NOT NULL REFERENCES Methode_der_Unterscheidung
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Fzartuntersch__hat_Kluntersch (
  Fahrzeugartenunterscheidung
    INTEGER NOT NULL REFERENCES Fahrzeugartenunterscheidung,
  Klassenunterscheidung
    INTEGER NOT NULL REFERENCES Klassenunterscheidung
)

CREATE TABLE Klassenunterscheidung (
  Klassenunterscheidung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Klassenunterscheidung_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Fahrzeugklasse
    INTEGER NOT NULL REFERENCES Fahrzeugklasse
)

CREATE TABLE Fahrzeugklasse (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Fahrzeugklasse VALUES (0,'keine ' ||
  'Fahrzeugklassenunterscheidung (nur Kfz)')
INSERT INTO Fahrzeugklasse VALUES (1,'Unterscheidung Pkw/Lkw')
INSERT INTO Fahrzeugklasse VALUES (2,'5+1 Fahrzeugklassen')
INSERT INTO Fahrzeugklasse VALUES (3,'8+1 Fahrzeugklassen')
INSERT INTO Fahrzeugklasse VALUES (4,'sonstige')

CREATE TABLE Methode_der_Unterscheidung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Methode_der_Unterscheidung VALUES (0,'Unterscheidung nach ' ||
  'Fahrzeuglängen')
INSERT INTO Methode_der_Unterscheidung VALUES (1,'Unterscheidung nach ' ||
  'Hüllkurven')

CREATE TABLE Differenzierung_der_Erfassung (
  Differenzierung_der_Erfassung_ID
    INTEGER NOT NULL PRIMARY KEY,
  hat_raelumliche_Differenzierung
    INTEGER NOT NULL REFERENCES raelumliche_Differenzierung
)

CREATE TABLE Diff_d_Erf__hat_Fahrstrzuordn (
  Differenzierung_der_Erfassung
    INTEGER NOT NULL REFERENCES Differenzierung_der_Erfassung,
  Fahrstreifenzuordnung
    INTEGER NOT NULL REFERENCES Fahrstreifenzuordnung
)

CREATE TABLE Fahrstreifenzuordnung (
  Fahrstreifenzuordnung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Fahrstreifenzuordnung_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_bis LIKE '___.__.____' ),
Fahrstreifen
        INTEGER NOT NULL, -- Anzahl
CHECK ( Fahrstreifen>=0 ),
Fahrtrichtung
        INTEGER NOT NULL REFERENCES Fahrtrichtung_DE,
CHECK ( Fahrstreifen<=8 )
)

CREATE TABLE Fahrtrichtung_DE (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Fahrtrichtung_DE VALUES (1,'IN-Richtung')
INSERT INTO Fahrtrichtung_DE VALUES (2,'GEGEN-Richtung')
INSERT INTO Fahrtrichtung_DE VALUES (3,'Gesamtquerschnitt')

CREATE TABLE raeumliche_Differenzierung (
    raeumliche_Differenzierung_ID
        INTEGER NOT NULL PRIMARY KEY,
    Gesamtspuren
        INTEGER NOT NULL, -- Anzahl
CHECK ( Gesamtspuren>=0 ),
Spuren_IN_Richtung
        INTEGER NOT NULL, -- Anzahl
CHECK ( Spuren_IN_Richtung>=0 ),
Registrierung
        INTEGER NOT NULL REFERENCES Registrierung,
CHECK ( Spuren_IN_Richtung<=8 ),
CHECK ( ( Gesamtspuren >= 1 ) AND ( Gesamtspuren <= 8 ) )
)

CREATE TABLE Registrierung (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Registrierung VALUES (1,'Gesamtquerschnitt')
INSERT INTO Registrierung VALUES (2,'Trennung nach Fahrtrichtung')
INSERT INTO Registrierung VALUES (3,'Trennung nach Fahrstreifen')

CREATE TABLE Art_der_registrierten_Daten (
    Art_der_registrierten_Daten_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Art_der_registrierten_Daten_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_bis LIKE '___.__.____' ),
    Datentyp
        INTEGER NOT NULL REFERENCES Datentyp
)

CREATE TABLE Datentyp (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Datentyp VALUES (1,'Kfz-Einheiten')
INSERT INTO Datentyp VALUES (2,'Achszahl')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Datentyp VALUES (3,'Achslasten')
INSERT INTO Datentyp VALUES (4,'Geschwindigkeiten')
INSERT INTO Datentyp VALUES (5,'Sonstiges')

CREATE TABLE Detektoren (
    Detektoren_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_Gueltingkeitsbereich
        INTEGER NOT NULL REFERENCES Gueltingkeitsbereich
)

CREATE TABLE Detekt__zugeh_Auswerteschalt (
    Detektoren
        INTEGER NOT NULL REFERENCES Detektoren,
    zugehoerige_Auswerteschaltung
        VARCHAR(20) NOT NULL
)

CREATE TABLE Detektoren__hat_Detektorart (
    Detektoren
        INTEGER NOT NULL REFERENCES Detektoren,
    Detektorart
        INTEGER NOT NULL REFERENCES Detektorart
)

CREATE TABLE Detekt__hat_Art_der_Registrung (
    Detektoren
        INTEGER NOT NULL REFERENCES Detektoren,
    Art_der_Registrierung
        INTEGER NOT NULL REFERENCES Art_der_Registrierung
)

CREATE TABLE Detektorart (
    Detektorart_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Detektorart_ID) REFERENCES Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Art_des_Detektors
        INTEGER NOT NULL REFERENCES Art_des_Detektors
)

CREATE TABLE Art_des_Detektors (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Art_des_Detektors VALUES (1,'Induktionsschleife')
INSERT INTO Art_des_Detektors VALUES (2,'Schlauch')
INSERT INTO Art_des_Detektors VALUES (3,'Lichtschranke')
INSERT INTO Art_des_Detektors VALUES (4,'Infrarotlicht')
INSERT INTO Art_des_Detektors VALUES (5,'Ultraschall')
INSERT INTO Art_des_Detektors VALUES (6,'Koaxialkabel')
INSERT INTO Art_des_Detektors VALUES (7,'Wiegeplatte')
INSERT INTO Art_des_Detektors VALUES (8,'Wiegeschwelle')
INSERT INTO Art_des_Detektors VALUES (9,'Radar')
INSERT INTO Art_des_Detektors VALUES (10,'Sonstige')
INSERT INTO Art_des_Detektors VALUES (11,'Witterungssensor')
INSERT INTO Art_des_Detektors VALUES (12,'sonstige')

CREATE TABLE Gueltingkeitsbereich (
    Gueltingkeitsbereich_ID
        INTEGER NOT NULL PRIMARY KEY,
    Aufstelldatum
        CHAR(10) NOT NULL, -- Datum
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CHECK ( Aufstelldatum LIKE '___.__.____' ),
Hersteller
  VARCHAR(50) NOT NULL,
Abbaudatum
  CHAR(10), -- Datum
CHECK ( Abbaudatum LIKE '___.__.____' )
)

CREATE TABLE Art_der_Registrierung (
  Art_der_Registrierung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Art_der_Registrierung_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Registriermedium
    INTEGER NOT NULL REFERENCES Registriermedium
)

CREATE TABLE Registriermedium (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Registriermedium VALUES (1,'in ablesbaren Zählwerten')
INSERT INTO Registriermedium VALUES (2,'graphisch')
INSERT INTO Registriermedium VALUES (3,'normale Schriftzeichen')
INSERT INTO Registriermedium VALUES (4,'maschinenlesbare Schriftzeichen')
INSERT INTO Registriermedium VALUES (5,'5-Kanal-CCITT2(Telex)-Code')
INSERT INTO Registriermedium VALUES (6,'8-Kanal-USASCI-Code')
INSERT INTO Registriermedium VALUES (7,'andere')
INSERT INTO Registriermedium VALUES (8,'7-Kanal-Band')
INSERT INTO Registriermedium VALUES (9,'9-Kanal-Band')
INSERT INTO Registriermedium VALUES (10,'Kassettenband')
INSERT INTO Registriermedium VALUES (11,'sonst. Registrierung')

CREATE TABLE Landesnummer (
  Landesnummer_ID
    INTEGER NOT NULL PRIMARY KEY,
  Nummer
    INTEGER NOT NULL,
  Schluessel
    INTEGER NOT NULL,
  CHECK ( (Schluessel=0)OR(Schluessel=1) )
)

CREATE TABLE Moegl_Einzelfahrzeugetfassung (
  Moegl_Einzelfahrzeugetfassung_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Moegl_Einzfzerf__Einzfzdaten (
  Moegl_Einzelfahrzeugetfassung
    INTEGER NOT NULL REFERENCES Moegl_Einzelfahrzeugetfassung,
  Einzelfahrzeugetdaten
    VARCHAR(99) NOT NULL
)

CREATE TABLE verwaltungstechn_Zuordnung_AD (
  verwaltungstechn_Zuordnung_AD_ID
    INTEGER NOT NULL PRIMARY KEY,
  Regionskennziffer
    INTEGER NOT NULL,
  hat_verwaltungstechn_Zuordnung
    INTEGER NOT NULL REFERENCES Allgemeine_Objekte.verwaltungstechn_Zuordnung
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

CREATE TABLE oertliche_Zuordnung_AD (
  oertliche_Zuordnung_AD_ID
    INTEGER NOT NULL PRIMARY KEY,
  Abschnittslaenge
    REAL, -- Kilometer
  CHECK ( Abschnittslaenge >= 0 ),
  hat_oertliche_Zuordnung
    INTEGER NOT NULL REFERENCES Allgemeine_Objekte.oertliche_Zuordnung
)

CREATE TABLE oertl_Zuord_AD__hat_Richtungen (
  oertliche_Zuordnung_AD
    INTEGER NOT NULL REFERENCES oertliche_Zuordnung_AD,
  Richtungen
    INTEGER NOT NULL REFERENCES Richtungen
)

CREATE TABLE oertl_Zuord_AD__hat_Fahrstrzut (
  oertliche_Zuordnung_AD
    INTEGER NOT NULL REFERENCES oertliche_Zuordnung_AD,
  Fahrstreifenzuteilung
    INTEGER NOT NULL REFERENCES Fahrstreifenzuteilung
)

CREATE TABLE Richtungen (
  Richtungen_ID
    INTEGER NOT NULL PRIMARY KEY,
  Nahziel_IN_Richtung
    VARCHAR(50) NOT NULL,
  Nahziel_GEGEN_Richtung
    VARCHAR(50) NOT NULL,
  Fernziel_IN_Richtung
    VARCHAR(50) NOT NULL,
  Fernziel_GEGEN_Richtung
    VARCHAR(50) NOT NULL,
  Richtung_IN
    INTEGER NOT NULL REFERENCES Himmelsrichtung,
  Richtung_GEGEN
    INTEGER NOT NULL REFERENCES Himmelsrichtung
)

CREATE TABLE Himmelsrichtung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Himmelsrichtung VALUES (1,'Norden')
INSERT INTO Himmelsrichtung VALUES (2,'Osten')
INSERT INTO Himmelsrichtung VALUES (3,'Süden')
INSERT INTO Himmelsrichtung VALUES (4,'Westen')

CREATE TABLE Fahrstreifenzuteilung (
  Fahrstreifenzuteilung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Fahrstreifenzuteilung_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Zahl_der_Fahrstreifen
    INTEGER NOT NULL, -- Anzahl
  CHECK ( Zahl_der_Fahrstreifen >= 0 ),
  Zahl_der_Fahrstr_IN_Richtung
    INTEGER NOT NULL, -- Anzahl
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    CHECK ( Zahl_der_Fahrstr_IN_Richtung >= 0 )
)

CREATE TABLE Unterzentrale_Verkehrsrechnerz (
    Unterzentrale_Verkehrsrechnerz_ID
        INTEGER NOT NULL PRIMARY KEY,
    Zuordnung
        INTEGER REFERENCES Zuordnung
)

CREATE TABLE Unterzentr_Verkrechnzent__Name (
    Unterzentrale_Verkehrsrechnerz
        INTEGER NOT NULL REFERENCES Unterzentrale_Verkehrsrechnerz,
    Name
        VARCHAR(99) NOT NULL
)

CREATE TABLE Zuordnung (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Zuordnung VALUES (0, 'keine Zuordnung')
INSERT INTO Zuordnung VALUES (1, 'Zuordnung zu einer oder mehreren ' ||
    'Unterzentralen')
INSERT INTO Zuordnung VALUES (2, 'Zuordnung zu einer oder mehreren ' ||
    'Verkehrsrechnerzentralen')

CREATE SCHEMA Verkehrsstaerke

CREATE TABLE Verkehrsstaerkekennwerte (
    Verkehrsstaerkekennwerte_ID
        INTEGER NOT NULL PRIMARY KEY,
    werden_addiert_zu_Tageswert_q
        INTEGER REFERENCES Tageswert_q,
    abbildbar_als_Wtganglinien
        INTEGER REFERENCES Wochentagsganglinien_q,
    abbildbar_als_Tganglinien
        INTEGER REFERENCES Tagesganglinien_q
)

CREATE TABLE Verkstaerkek__sind_Verkstaerk (
    Verkehrsstaerkekennwerte
        INTEGER NOT NULL REFERENCES Verkehrsstaerkekennwerte,
    Verkehrsstaerken
        INTEGER NOT NULL REFERENCES Verkehrsstaerken
)

CREATE TABLE Verkstaerkek__hat_zeitl_Zuord (
    Verkehrsstaerkekennwerte
        INTEGER NOT NULL REFERENCES Verkehrsstaerkekennwerte,
    zeitliches_Zuordnungskriterium
        INTEGER NOT NULL REFERENCES zeitliches_Zuordnungskriterium
)

CREATE TABLE Verkehrsstaerken (
    Verkehrsstaerken_ID
        INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Verkstaerk__hat_Verkstarkevert (
    Verkehrsstaerken
        INTEGER NOT NULL REFERENCES Verkehrsstaerken,
    Verkehrsstaerkeverteilung
        INTEGER NOT NULL REFERENCES Verkehrsstaerkeverteilung
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Verkehrsstaerkeverteilung (
  Verkehrsstaerkeverteilung_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Verkstaerkevert__zug_Fz_Kl_VS (
  Verkehrsstaerkeverteilung
    INTEGER NOT NULL REFERENCES Verkehrsstaerkeverteilung,
  zugehoerige_FZ_Klasse_VS
    INTEGER NOT NULL
)

CREATE TABLE Verkstaerkevert__Verkstaerkew (
  Verkehrsstaerkeverteilung
    INTEGER NOT NULL REFERENCES Verkehrsstaerkeverteilung,
  Verkehrsstaerkewert
    INTEGER NOT NULL
)

CREATE TABLE Tageswert_q (
  Tageswert_q_ID
    INTEGER NOT NULL PRIMARY KEY,
  Wert
    INTEGER NOT NULL, -- Kfz_pro_Stunde
  CHECK ( Wert >= 0 ),
  Datum
    CHAR(10) NOT NULL, -- Datum
  CHECK ( Datum LIKE '___.__.____' ),
  werden_aufbereitet_zu_Wgangl
    INTEGER REFERENCES Wochenganglinie_q,
  werden_aufbereitet_zu_Jgangl
    INTEGER REFERENCES Jahresganglinie_q
)

CREATE TABLE Wochenganglinie_q (
  Wochenganglinie_q_ID
    INTEGER NOT NULL PRIMARY KEY,
  Woche
    INTEGER NOT NULL, -- Woche
  CHECK ( ( Woche >= 1 ) AND ( Woche <= 53 ) ),
  Jahr
    INTEGER NOT NULL, -- Jahr
  CHECK ( ( Jahr >= 1900 ) AND ( Jahr <= 2099 ) ),
  Ganglinientyp
    CHAR(1), -- Ganglinientyp
  CHECK ( ( Ganglinientyp >= 'A' ) AND ( Ganglinientyp <= 'F' ) )
)

CREATE TABLE Jahresganglinie_q (
  Jahresganglinie_q_ID
    INTEGER NOT NULL PRIMARY KEY,
  Jahr
    INTEGER NOT NULL, -- Jahr
  CHECK ( ( Jahr >= 1900 ) AND ( Jahr <= 2099 ) ),
  Ganglinientyp
    CHAR(1), -- Ganglinientyp
  CHECK ( ( Ganglinientyp >= 'A' ) AND ( Ganglinientyp <= 'F' ) )
)

CREATE TABLE Tagesganglinien_q (
  Tagesganglinien_q_ID
    INTEGER NOT NULL PRIMARY KEY,
  Datum
    CHAR(10) NOT NULL, -- Datum
  CHECK ( Datum LIKE '___.__.____' )
)

CREATE TABLE Wochentagskennung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Langtext
    VARCHAR(28) NOT NULL
)

INSERT INTO Wochentagskennung VALUES (1,'montags')
INSERT INTO Wochentagskennung VALUES (2,'dienstags bis donnerstags')
INSERT INTO Wochentagskennung VALUES (3,'feiertags')
INSERT INTO Wochentagskennung VALUES (4,'samstags und sonntags')

CREATE TABLE Wochentagsganglinien_q (
    Wochentagsganglinien_q_ID
        INTEGER NOT NULL PRIMARY KEY,
    Wochentag
        INTEGER NOT NULL REFERENCES Allgemeine_Objekte.Wochentag,
    Jahr
        INTEGER NOT NULL, -- Jahr
    CHECK ( ( Jahr >= 1900 ) AND ( Jahr <= 2099 ) ),
    Wochentagskennung
        INTEGER REFERENCES Wochentagskennung,
    Ganglinientyp
        CHAR(1), -- Ganglinientyp
    CHECK ( ( Ganglinientyp >= 'A' ) AND ( Ganglinientyp <= 'F' ) )
)

CREATE TABLE zeitliches_Zuordnungskriterium (
    zeitliches_Zuordnungskriterium_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_zeitliches_Zuordnungskriterium
        INTEGER NOT NULL REFERENCES Art_zeitliches_Zuordnungskriterium
)

CREATE TABLE Art_zeitliches_Zuordnungskriterium (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(18) NOT NULL
)

INSERT INTO Art_zeitliches_Zuordnungskriterium VALUES (1,'Tageszeitwert')
INSERT INTO Art_zeitliches_Zuordnungskriterium VALUES (2,'Nachtzeitwert')
INSERT INTO Art_zeitliches_Zuordnungskriterium VALUES (3,'alle_Tage')
INSERT INTO Art_zeitliches_Zuordnungskriterium VALUES (4,'werktags')
INSERT INTO Art_zeitliches_Zuordnungskriterium VALUES (5,'sonn_und_feiertags')
INSERT INTO Art_zeitliches_Zuordnungskriterium VALUES (6,'ferienwerktags')
INSERT INTO Art_zeitliches_Zuordnungskriterium VALUES (7,'Di_bis_Do')

CREATE TABLE Tageszeitwert (
    Tageszeitwert_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Tageszeitwert_ID) REFERENCES zeitliches_Zuordnungskriterium,
    Zeitbereich
        INTEGER NOT NULL,
    massgebender_Wert_M_T
        INTEGER, -- Kfz_pro_Tag
    CHECK ( massgebender_Wert_M_T >= 0 )
)

CREATE TABLE Nachtzeitwert (
    Nachtzeitwert_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Nachtzeitwert_ID) REFERENCES zeitliches_Zuordnungskriterium,
    Zeitbereich
        INTEGER NOT NULL,
    massgebender_Wert_M_N
        INTEGER, -- Kfz_pro_Tag
    CHECK ( massgebender_Wert_M_N >= 0 )
)

CREATE TABLE alle_Tage (
    alle_Tage_ID
        INTEGER NOT NULL PRIMARY KEY,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
FOREIGN KEY (alle_Tage_ID) REFERENCES zeitliches_Zuordnungskriterium,
Zeitbereich
    VARCHAR(30) NOT NULL,
DTV
    INTEGER NOT NULL, -- Kfz_pro_Tag
CHECK ( DTV>=0 ),
DTV_Lkw
    INTEGER, -- Lkw_pro_Tag
CHECK ( DTV_Lkw>=0 ),
Lkw_Anteil
    REAL, -- Prozent
MSV
    INTEGER NOT NULL, -- Kfz_pro_Stunde
CHECK ( MSV>=0 ),
MSV_Lkw
    INTEGER, -- Lkw_pro_Stunde
CHECK ( MSV_Lkw>=0 )
)

CREATE TABLE werktags (
    werktags_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (werktags_ID) REFERENCES zeitliches_Zuordnungskriterium,
    DTV_W
        INTEGER NOT NULL, -- Kfz_pro_Tag
    CHECK ( DTV_W>=0 ),
    DTV_W_Lkw
        INTEGER, -- Lkw_pro_Tag
    CHECK ( DTV_W_Lkw>=0 ),
    MSV_W
        INTEGER NOT NULL, -- Kfz_pro_Stunde
    CHECK ( MSV_W>=0 ),
    MSV_W_Lkw
        INTEGER, -- Lkw_pro_Stunde
    CHECK ( MSV_W_Lkw>=0 )
)

CREATE TABLE werktags__Zeitbereich (
    werktags
        INTEGER NOT NULL REFERENCES werktags,
    Zeitbereich
        VARCHAR(28) NOT NULL
)

CREATE TABLE sonn_und_feiertags (
    sonn_und_feiertags_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (sonn_und_feiertags_ID) REFERENCES zeitliches_Zuordnungskriterium,
    Anzahl
        INTEGER NOT NULL,
    Jahr
        INTEGER NOT NULL, -- Jahr
    CHECK ( ( Jahr >= 1900 ) AND ( Jahr <= 2099 ) ),
    DTV_S
        INTEGER NOT NULL, -- Kfz_pro_Tag
    CHECK ( DTV_S>=0 ),
    MSV_S
        INTEGER NOT NULL, -- Kfz_pro_Stunde
    CHECK ( MSV_S>=0 )
)

CREATE TABLE ferienwerktags (
    ferienwerktags_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (ferienwerktags_ID) REFERENCES zeitliches_Zuordnungskriterium,
    DTV_F
        INTEGER NOT NULL, -- Kfz_pro_Tag
    CHECK ( DTV_F>=0 ),
    DTV_F_Lkw
        INTEGER, -- Lkw_pro_Tag
    CHECK ( DTV_F_Lkw>=0 ),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
MSV_F
    INTEGER NOT NULL, -- Kfz_pro_Stunde
CHECK ( MSV_F>=0 ),
MSV_F_Lkw
    INTEGER, -- Lkw_pro_Stunde
CHECK ( MSV_F_Lkw>=0 )
)

CREATE TABLE ferienwerktags__Zeitbereich (
    ferienwerktags
        INTEGER NOT NULL REFERENCES ferienwerktags,
    Zeitbereich
        VARCHAR(30) NOT NULL
)

CREATE TABLE Di_bis_Do (
    Di_bis_Do_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Di_bis_Do_ID) REFERENCES zeitliches_Zuordnungskriterium,
    Zeitbereich
        VARCHAR(30) NOT NULL,
    DTV_Di_bis_Do
        INTEGER NOT NULL, -- Kfz_pro_Tag
CHECK ( DTV_Di_bis_Do>=0 )
)

CREATE SCHEMA Erfassung_Geschwindigkeiten

CREATE TABLE Erfassung_Geschwindigkeiten (
    Erfassung_Geschwindigkeiten_ID
        INTEGER NOT NULL PRIMARY KEY,
    zugehoerige_Fahrzeugklasse
        INTEGER NOT NULL
)

CREATE TABLE Erf_Geschw__hat_V_Verteilungen (
    Erfassung_Geschwindigkeiten
        INTEGER NOT NULL REFERENCES Erfassung_Geschwindigkeiten,
    V_Verteilungen
        INTEGER NOT NULL REFERENCES V_Verteilungen
)

CREATE TABLE Erf_Geschw__hat_aggreg_Werte (
    Erfassung_Geschwindigkeiten
        INTEGER NOT NULL REFERENCES Erfassung_Geschwindigkeiten,
    aggregierte_Werte
        INTEGER NOT NULL REFERENCES aggregierte_Werte
)

CREATE TABLE V_Verteilungen (
    V_Verteilungen_ID
        INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE V_Verteilung__hat_Anzahl_V_Kl (
    V_Verteilungen
        INTEGER NOT NULL REFERENCES V_Verteilungen,
    Anzahl_V_Klassen
        INTEGER NOT NULL REFERENCES Anzahl_V_Klassen
)

CREATE TABLE V_Verteilungen__hat_V_Klassen (
    V_Verteilungen
        INTEGER NOT NULL REFERENCES V_Verteilungen,
    V_Klassen
        INTEGER NOT NULL REFERENCES V_Klassen
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Anzahl_V_Klassen (
  Anzahl_V_Klassen_ID
    INTEGER NOT NULL PRIMARY KEY,
  Fahrzeugklasse
    INTEGER NOT NULL,
  Anzahl_V_Klassen
    INTEGER NOT NULL, -- Anzahl
  CHECK ( Anzahl_V_Klassen >= 0 ),
  CHECK ( Anzahl_V_Klassen <= 16 ),
  CHECK ( Fahrzeugklasse >= 1 )
)

CREATE TABLE V_Klassen (
  V_Klassen_ID
    INTEGER NOT NULL PRIMARY KEY,
  Klassennummer
    INTEGER NOT NULL, -- Anzahl_zweistellig
  CHECK ( ( Klassennummer >= 0 ) AND ( Klassennummer <= 99 ) ),
  Klassengrenze
    INTEGER NOT NULL,
  Klassenwert
    INTEGER NOT NULL
)

CREATE TABLE V_Kl__hat_abgel_Wert_u_Verteil (
  V_Klassen
    INTEGER NOT NULL REFERENCES V_Klassen,
  abgel_Werte_u_Verteilgen_VK
    INTEGER NOT NULL REFERENCES abgel_Werte_u_Verteilgen_VK
)

CREATE TABLE abgel_Werte_u_Verteilgen_VK (
  abgel_Werte_u_Verteilgen_VK_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE aggregierte_Werte (
  aggregierte_Werte_ID
    INTEGER NOT NULL PRIMARY KEY,
  Standardabweichung
    INTEGER, -- Stundenkilometer
  CHECK ( Standardabweichung >= 0 ),
  v85
    INTEGER, -- Stundenkilometer
  CHECK ( v85 >= 0 ),
  hat_mittlere_Geschwindigkeit
    INTEGER NOT NULL REFERENCES mittlere_Geschwindigkeit,
  CHECK ( v85 <= 254 ),
  CHECK ( Standardabweichung <= 255 )
)

CREATE TABLE mittlere_Geschwindigkeit (
  mittlere_Geschwindigkeit_ID
    INTEGER NOT NULL PRIMARY KEY,
  Geschwindigkeit
    INTEGER NOT NULL, -- Stundenkilometer
  CHECK ( Geschwindigkeit >= 0 ),
  CHECK ( Geschwindigkeit <= 255 )
)

CREATE TABLE mittl_Geschw__hat_abgel_Werte (
  mittlere_Geschwindigkeit
    INTEGER NOT NULL REFERENCES mittlere_Geschwindigkeit,
  abgeleitete_Werte_VM
    INTEGER NOT NULL REFERENCES abgeleitete_Werte_VM
)

CREATE TABLE abgeleitete_Werte_VM (
  abgeleitete_Werte_VM_ID
    INTEGER NOT NULL PRIMARY KEY
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE SCHEMA Achslastdaten

CREATE TABLE Achslastdatenerfassung (
  Achslastdatenerfassung_ID
  INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Achslasterf__hat_GG_Vert_Fz_Kl (
  Achslastdatenerfassung
  INTEGER NOT NULL REFERENCES Achslastdatenerfassung,
  GG_Verteilung_einer_FZ_Klasse
  INTEGER NOT NULL REFERENCES GG_Verteilung_einer_FZ_Klasse
)

CREATE TABLE Achslasterf__hat_Achslvrt_Atyp (
  Achslastdatenerfassung
  INTEGER NOT NULL REFERENCES Achslastdatenerfassung,
  Achslastverteilung_e_Achstyps
  INTEGER NOT NULL REFERENCES Achslastverteilung_e_Achstyps
)

CREATE TABLE Achslasterf__hat_abgel_W_u_V (
  Achslastdatenerfassung
  INTEGER NOT NULL REFERENCES Achslastdatenerfassung,
  abgel_Werte_und_Verteilungen
  INTEGER NOT NULL REFERENCES abgel_Werte_und_Verteilungen
)

CREATE TABLE abgel_Werte_und_Verteilungen (
  abgel_Werte_und_Verteilungen_ID
  INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE GG_Verteilung_einer_FZ_Klasse (
  GG_Verteilung_einer_FZ_Klasse_ID
  INTEGER NOT NULL PRIMARY KEY,
  Fahrzeugklasse
  INTEGER NOT NULL,
  Anzahl_der_GG_Klassen_FZ_Kl_3
  INTEGER NOT NULL,
  Anzahl_der_GG_Klassen_FZ_Kl_4
  INTEGER NOT NULL,
  Anzahl_der_GG_Klassen_FZ_Kl_5
  INTEGER NOT NULL,
  Anzahl_Ueberladungen
  INTEGER NOT NULL, -- Anzahl
  CHECK ( Anzahl_Ueberladungen>=0 ),
  CHECK ( Anzahl_Ueberladungen<=255 ),
  CHECK ( ( Fahrzeugklasse >= 3 ) AND ( Fahrzeugklasse <= 5 ) )
)

CREATE TABLE GG_Vert_Fz_Kl__hat_GG_Klasse (
  GG_Verteilung_einer_FZ_Klasse
  INTEGER NOT NULL REFERENCES GG_Verteilung_einer_FZ_Klasse,
  GG_Klasse
  INTEGER NOT NULL REFERENCES GG_Klasse
)

CREATE TABLE GG_Klasse (
  GG_Klasse_ID
  INTEGER NOT NULL PRIMARY KEY,
  Klassennummer
  INTEGER NOT NULL,
  Klassengrenze
  REAL NOT NULL, -- Groesse
  CHECK ( Klassengrenze>=0 ),
  Klassenwert
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER NOT NULL, -- Klassenwert
        CHECK ( Klassenwert<=65535 ),
        CHECK ( Klassenwert>=0 ),
        CHECK ( ( Klassennummer >= 1 ) AND ( Klassennummer <= 8 ) )
    )

CREATE TABLE Achstyp (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Achstyp VALUES (1,'Einfachachse')
INSERT INTO Achstyp VALUES (2,'Doppelachse')
INSERT INTO Achstyp VALUES (3,'Dreifachachse')

CREATE TABLE Achslastverteilung_e_Achstyps (
    Achslastverteilung_e_Achstyps_ID
        INTEGER NOT NULL PRIMARY KEY,
    Achstyp
        INTEGER NOT NULL REFERENCES Achstyp,
    Anz_d_Klassen_f_Einzelachsen
        INTEGER NOT NULL, -- Anzahl_Klassen
        CHECK ( ( Anz_d_Klassen_f_Einzelachsen >= 1 ) AND (
Anz_d_Klassen_f_Einzelachsen <= 99 ) ),
    Anz_d_Klassen_f_Doppelachsen
        INTEGER NOT NULL, -- Anzahl_Klassen
        CHECK ( ( Anz_d_Klassen_f_Doppelachsen >= 1 ) AND (
Anz_d_Klassen_f_Doppelachsen <= 99 ) ),
    Anz_d_Klassen_f_Dreifachachsen
        INTEGER NOT NULL, -- Anzahl_Klassen
        CHECK ( ( Anz_d_Klassen_f_Dreifachachsen >= 1 ) AND (
Anz_d_Klassen_f_Dreifachachsen <= 99 ) ),
    hat_Klasse_AL
        INTEGER NOT NULL REFERENCES Klasse_AL
)

CREATE TABLE Klasse_AL (
    Klasse_AL_ID
        INTEGER NOT NULL PRIMARY KEY,
    Klassennummer
        INTEGER NOT NULL,
    Klassengrenze
        REAL NOT NULL, -- Groesse
        CHECK ( Klassengrenze>=0 ),
    Klassenwert
        INTEGER NOT NULL, -- Klassenwert
        CHECK ( Klassenwert<=65535 ),
        CHECK ( Klassenwert>=0 ),
        CHECK ( ( Klassennummer >= 0 ) AND ( Klassennummer <= 99 ) )
)

CREATE SCHEMA Einzelfahrzeugdaten

CREATE TABLE Einzelfahrzeugdatenerfassung (
    Einzelfahrzeugdatenerfassung_ID
        INTEGER NOT NULL PRIMARY KEY,
    Nettozeitluecke_z_vorherig_FZ
        INTEGER, -- Millisekunden
        CHECK ( Nettozeitluecke_z_vorherig_FZ>=0 ),
    Geschwindigkeit
        INTEGER NOT NULL, -- Stundenkilometer
        CHECK ( Geschwindigkeit>=0 ),
    Fahrzeuglaenge
        REAL, -- Zentimeter
        CHECK ( Fahrzeuglaenge>=0 ),
    Abstand_zum_vorausfahrenden_FZ

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        REAL, -- Zentimeter
CHECK ( Abstand_zum_vorausfahrenden_FZ>=0 ),
hat_zugehoerige_Fahrzeugklasse
    INTEGER NOT NULL REFERENCES zugehoerige_Fahrzeugklasse,
hat_zugehoerigen_Zeitpunkt
    INTEGER NOT NULL REFERENCES zugehoeriger_Zeitpunkt,
hat_fahrzeugbez_Achslastdaten
    INTEGER REFERENCES fahrzeugbezogene_Achslastdaten,
CHECK ( Geschwindigkeit<=255 )
)

CREATE TABLE Einzelfzdatenerf__hat_achsbez_Dat (
    Einzelfahrzeugdatenerfassung
        INTEGER NOT NULL REFERENCES Einzelfahrzeugdatenerfassung,
    achsbezogene_Daten
        INTEGER NOT NULL REFERENCES achsbezogene_Daten
)

CREATE TABLE zugehoerige_Fahrzeugklasse (
    zugehoerige_Fahrzeugklasse_ID
        INTEGER NOT NULL PRIMARY KEY,
    Klasse
        VARCHAR(20) NOT NULL,
    Typ
        VARCHAR(20) NOT NULL
)

CREATE TABLE zugehoeriger_Zeitpunkt (
    zugehoeriger_Zeitpunkt_ID
        INTEGER NOT NULL PRIMARY KEY,
    Datum
        CHAR(10) NOT NULL, -- Datum
CHECK ( Datum LIKE '__.__.____' ),
    Uhrzeit
        VARCHAR(12) NOT NULL, -- Uhrzeit
CHECK ( ( Uhrzeit LIKE '__:__' ) OR ( Uhrzeit LIKE '__:__:__' ) OR ( Uhrzeit
LIKE '__:__:__:____' ) )
)

CREATE TABLE fahrzeugbezogene_Achslastdaten (
    fahrzeugbezogene_Achslastdaten_ID
        INTEGER NOT NULL PRIMARY KEY,
    Anzahl_der_Achsen
        INTEGER NOT NULL, -- Anzahl
CHECK ( Anzahl_der_Achsen>=0 ),
    Gesamtgewicht
        REAL NOT NULL, -- Kilogramm
CHECK ( Gesamtgewicht>=0 ),
    hat_Ueberladung
        INTEGER NOT NULL REFERENCES Ueberladung
)

CREATE TABLE Ueberladung (
    Ueberladung_ID
        INTEGER NOT NULL PRIMARY KEY,
    Ueberladung_vorhanden
        INTEGER REFERENCES Logical_Types.Boolean,
    Ueberladung
        REAL, -- Kilogramm
CHECK ( Ueberladung>=0 )
)

CREATE TABLE achsbezogene_Daten (
    achsbezogene_Daten_ID
        INTEGER NOT NULL PRIMARY KEY,
    Achsabstand_zur_Vorderachse
        REAL NOT NULL, -- Zentimeter
CHECK ( Achsabstand_zur_Vorderachse>=0 ),
    Achslast
        REAL NOT NULL, -- Kilogramm
CHECK ( Achslast>=0 ),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Achstyp
  INTEGER NOT NULL REFERENCES Achslastdaten.Achstyp,
hat_Ueberladung
  INTEGER NOT NULL REFERENCES Ueberladung
)

CREATE SCHEMA Manuelle_Zaehlstelle

CREATE TABLE manuelle_Zaehlstelle (
  manuelle_Zaehlstelle_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (manuelle_Zaehlstelle_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '__.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '__.__.____' ),
  TK25_Blattnummer
    INTEGER NOT NULL, -- TK25_Blattnummer
  CHECK ( ( TK25_Blattnummer >= 1 ) AND ( TK25_Blattnummer <= 9999 ) ),
  Zaehlstellenummer
    INTEGER NOT NULL,
  hat_Erfassung_Auslaenderverk
    INTEGER REFERENCES Erfassung_Auslaenderverkehr,
  hat_keine_Erfassung_AV
    INTEGER REFERENCES keine_Erfassung_AV,
  hat_verwaltungstechn_Zuordnung
    INTEGER NOT NULL REFERENCES verwaltungstechn_Zuordnung_MZ,
  hat_Charakteristik
    INTEGER NOT NULL REFERENCES Charakteristik,
  hat_oertliche_Zuordnung
    INTEGER NOT NULL REFERENCES oertliche_Zuordnung_MZ,
  hat_Angaben_zur_letzten_Zaehlg
    INTEGER NOT NULL REFERENCES Angaben_zur_letzten_Zaehlung,
  CHECK ( ( hat_Erfassung_Auslaenderverk IS NULL ) OR ( hat_keine_Erfassung_AV IS
NULL ) )
)

CREATE TABLE manu_Zaehlst__hat_Erf_Gesverk (
  manuelle_Zaehlstelle
    INTEGER NOT NULL REFERENCES manuelle_Zaehlstelle,
  Erfassung_Gesamtverkehr
    INTEGER NOT NULL REFERENCES Erfassung_Gesamtverkehr
)

CREATE TABLE manu_Zaehlst__hat_Hochrechnerg (
  manuelle_Zaehlstelle
    INTEGER NOT NULL REFERENCES manuelle_Zaehlstelle,
  Hochrechnungsergebnisse
    INTEGER NOT NULL REFERENCES Hochrechnungsergebnisse
)

CREATE TABLE manu_Zaehlst__hat_Bezbst_Auslv (
  manuelle_Zaehlstelle
    INTEGER NOT NULL REFERENCES manuelle_Zaehlstelle,
  Bezugszaehlstelle_Auslverkehr
    INTEGER NOT NULL REFERENCES Bezugszaehlstelle_Auslverkehr
)

CREATE TABLE manu_Zaehlst__hat_Bezbst_Gesvk (
  manuelle_Zaehlstelle
    INTEGER NOT NULL REFERENCES manuelle_Zaehlstelle,
  Bezugszaehlstelle_Gesamtverk
    INTEGER NOT NULL REFERENCES Bezugszaehlstelle_Gesamtverk
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Erfassung_Gesamtverkehr (
  Erfassung_Gesamtverkehr_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Erfassung_Gesamtverkehr_ID) REFERENCES
  Strassennetz.Streckenobjekt_stat,
  Abschnittslaenge
    REAL NOT NULL, -- Kilometer
  CHECK ( Abschnittslaenge>=0 )
)

CREATE TABLE Erfassung_Auslaenderverkehr (
  Erfassung_Auslaenderverkehr_ID
    INTEGER NOT NULL PRIMARY KEY,
  Abschnittslaenge
    REAL NOT NULL, -- Kilometer
  CHECK ( Abschnittslaenge>=0 )
)

CREATE TABLE Erfass_Auslverk__hat_Gueltber (
  Erfassung_Auslaenderverkehr
    INTEGER NOT NULL REFERENCES Erfassung_Auslaenderverkehr,
  Gueltigkeitsbereich_AV
    INTEGER NOT NULL REFERENCES Gueltigkeitsbereich_AV
)

CREATE TABLE Gueltigkeitsbereich_AV (
  Gueltigkeitsbereich_AV_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Gueltigkeitsbereich_AV_ID) REFERENCES
  Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Beginn
    INTEGER NOT NULL,
  Ende
    INTEGER NOT NULL
)

CREATE TABLE keine_Erfassung_AV (
  keine_Erfassung_AV_ID
    INTEGER NOT NULL PRIMARY KEY,
  Erfassungsart
    INTEGER NOT NULL REFERENCES Logical_Types.Boolean
)

CREATE TABLE verwaltungstechn_Zuordnung_MZ (
  verwaltungstechn_Zuordnung_MZ_ID
    INTEGER NOT NULL PRIMARY KEY,
  hat_verwaltungstechn_Zuordnung
    INTEGER NOT NULL REFERENCES Allgemeine_Objekte.verwaltungstechn_Zuordnung
)

CREATE TABLE verwtechn_Zuord_MZ__hat_Verwkz (
  verwaltungstechn_Zuordnung_MZ
    INTEGER NOT NULL REFERENCES verwaltungstechn_Zuordnung_MZ,
  Verwaltungskennziffern
    INTEGER NOT NULL REFERENCES Verwaltungskennziffern
)

CREATE TABLE Verwaltungskennziffern (
  Verwaltungskennziffern_ID
    INTEGER NOT NULL PRIMARY KEY,
  Regionskennziffer_1
    INTEGER NOT NULL,
  Regionskennziffer_2
    INTEGER NOT NULL,
  Kreiskennziffer

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER NOT NULL,
        Kennziffer_SM_AM
        INTEGER NOT NULL
    )

CREATE TABLE Hochrechnungsergebnisse (
    Hochrechnungsergebnisse_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Hochrechnungsergebnisse_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Fahrradzaehlung
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean,
    Kradzaehlung
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean
    )

CREATE TABLE Hochrechnerg__Fahrzeuggruppen (
    Hochrechnungsergebnisse
        INTEGER NOT NULL REFERENCES Hochrechnungsergebnisse,
    Fahrzeuggruppen
        VARCHAR(99) NOT NULL
    )

CREATE TABLE Hochrechnerg__Datenbeschreib (
    Hochrechnungsergebnisse
        INTEGER NOT NULL REFERENCES Hochrechnungsergebnisse,
    Datenbeschreibung
        VARCHAR(99) NOT NULL
    )

CREATE TABLE Charakteristik (
    Charakteristik_ID
        INTEGER NOT NULL PRIMARY KEY,
    Charakteristik_1
        INTEGER NOT NULL REFERENCES Tab_Charakteristik,
    Charakteristik_2
        INTEGER NOT NULL REFERENCES Tab_Charakteristik,
    Charakteristik_3
        INTEGER NOT NULL REFERENCES Tab_Charakteristik,
    Lagekennung
        CHAR(1) NOT NULL REFERENCES Lagekennung,
    Netzknotenangaben
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean,
    CHECK ( Charakteristik_3>=3 ),
    CHECK ( Charakteristik_2>=2 )
    )

CREATE TABLE Charakt__hat_Ang_z_Strquerschn (
    Charakteristik
        INTEGER NOT NULL REFERENCES Charakteristik,
    Angaben_z_Strassenquerschnitt
        INTEGER NOT NULL REFERENCES Angaben_z_Strassenquerschnitt
    )

CREATE TABLE Charakt__hat_Richtungskennung (
    Charakteristik
        INTEGER NOT NULL REFERENCES Charakteristik,
    Richtungskennung
        INTEGER NOT NULL REFERENCES Richtungskennung
    )

CREATE TABLE Charakt__hat_Bemerkgskennziff (
    Charakteristik
        INTEGER NOT NULL REFERENCES Charakteristik,
    Bemerkungskennziffern

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER NOT NULL REFERENCES Bemerkungskennziffern
    )

CREATE TABLE Tab_Charakteristik (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Tab_Charakteristik VALUES (1,'Kreuzung')
INSERT INTO Tab_Charakteristik VALUES (2,'Einmündung')
INSERT INTO Tab_Charakteristik VALUES (3,'Ein-/Ausfahrt Grundstück')
INSERT INTO Tab_Charakteristik VALUES (4,'Steigung/Gefälle')
INSERT INTO Tab_Charakteristik VALUES (5,'Kuppe')
INSERT INTO Tab_Charakteristik VALUES (6,'Kurve')
INSERT INTO Tab_Charakteristik VALUES (7,'frei')
INSERT INTO Tab_Charakteristik VALUES (8,'Steigung')
INSERT INTO Tab_Charakteristik VALUES (9,'Gefälle')

CREATE TABLE Lagekennung (
    Kennung
        CHAR(1) NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Lagekennung VALUES ('F','Freie Strecke')
INSERT INTO Lagekennung VALUES ('O','Ortsdurchfahrt')

CREATE TABLE Angaben_z_Strassenquerschnitt (
    Angaben_z_Strassenquerschnitt_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Angaben_z_Strassenquerschnitt_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Anzahl_Fahrstreifen
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl_Fahrstreifen>=0 ),
    Anzahl_Standstreifen
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl_Standstreifen>=0 ),
    Anzahl_Mehrzweckstreifen
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl_Mehrzweckstreifen>=0 ),
    Anzahl_Radwege
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl_Radwege>=0 ),
    Anzahl_Gleise
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl_Gleise>=0 ),
    Anzahl_Breitenklasse
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl_Breitenklasse>=0 )
)

CREATE TABLE Richtungskennung (
    Richtungskennung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Richtungskennung_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CHECK ( gueltig_bis LIKE '___.__.____' ),
Kennung
    INTEGER NOT NULL REFERENCES Tab_Richtungskennung
)

CREATE TABLE Tab_Richtungskennung (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Tab_Richtungskennung VALUES (0,'GQ')
INSERT INTO Tab_Richtungskennung VALUES (1,'N')
INSERT INTO Tab_Richtungskennung VALUES (2,'O')
INSERT INTO Tab_Richtungskennung VALUES (3,'S')
INSERT INTO Tab_Richtungskennung VALUES (4,'W')

CREATE TABLE Bemerkungskennziffern (
    Bemerkungskennziffern_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Bemerkungskennziffern_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Kennziffer_1
        INTEGER NOT NULL REFERENCES Kennziffer_1,
    Kennziffer_2
        INTEGER NOT NULL REFERENCES Kennziffer_2
)

CREATE TABLE Kennziffer_1 (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Kennziffer_1 VALUES (1,'keine aktuelle Zählung')
INSERT INTO Kennziffer_1 VALUES (2,'manuelle und automatische Zählung')
INSERT INTO Kennziffer_1 VALUES (3,'nur automatische Zählung')
INSERT INTO Kennziffer_1 VALUES (4,'Knotenzählstelle')
INSERT INTO Kennziffer_1 VALUES (5,'rechnerisch')

CREATE TABLE Kennziffer_2 (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Kennziffer_2 VALUES (1,'neue Zählstelle')
INSERT INTO Kennziffer_2 VALUES (2,'Silhouettenzählstelle')
INSERT INTO Kennziffer_2 VALUES (3,'Ergänzungszählstelle')
INSERT INTO Kennziffer_2 VALUES (4,'neue und Ergänzungszählstelle')

CREATE TABLE oertliche_Zuordnung_MZ (
    oertliche_Zuordnung_MZ_ID
        INTEGER NOT NULL PRIMARY KEY,
    Sortiernummer
        INTEGER NOT NULL,
    hat_oertliche_Zuordnung
        INTEGER NOT NULL REFERENCES Allgemeine_Objekte.oertliche_Zuordnung
)

CREATE TABLE oertl_Zuord_Mz__hat_ben_Anschl (
    oertliche_Zuordnung_MZ
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER NOT NULL REFERENCES oertliche_Zuordnung_MZ,
        benachbarte_Anschlussstellen
        INTEGER NOT NULL REFERENCES benachbarte_Anschlussstellen
    )

CREATE TABLE benachbarte_Anschlussstellen (
    benachbarte_Anschlussstellen_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (benachbarte_Anschlussstellen_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    AS_von
        VARCHAR(50) NOT NULL,
    AS_bis
        VARCHAR(50) NOT NULL
)

CREATE TABLE Bezugszaehlstelle_Auslverkehr (
    Bezugszaehlstelle_Auslverkehr_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Bezugszaehlstelle_Auslverkehr_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Land
        INTEGER NOT NULL,
    Bauamt
        INTEGER NOT NULL,
    TK25_Nummer
        INTEGER NOT NULL, -- TK25_Blattnummer
    CHECK ( ( TK25_Nummer >= 1 ) AND ( TK25_Nummer <= 9999 ) ),
    Zaehlstellennummer
        INTEGER NOT NULL
)

CREATE TABLE Bezugszaehlstelle_Gesamtverk (
    Bezugszaehlstelle_Gesamtverk_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Bezugszaehlstelle_Gesamtverk_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Land
        INTEGER NOT NULL,
    Bauamt
        INTEGER NOT NULL,
    TK25_Nummer
        INTEGER NOT NULL, -- TK25_Blattnummer
    CHECK ( ( TK25_Nummer >= 1 ) AND ( TK25_Nummer <= 9999 ) ),
    Zaehlstellennummer
        INTEGER NOT NULL
)

CREATE TABLE Angaben_zur_letzten_Zaehlung (
    Angaben_zur_letzten_Zaehlung_ID
        INTEGER NOT NULL PRIMARY KEY,
    Land
        INTEGER NOT NULL,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Bauamt
    INTEGER NOT NULL,
Zaehlstellennummer
    INTEGER NOT NULL,
TK25_Nummer
    INTEGER NOT NULL, -- TK25_Blattnummer
CHECK ( ( TK25_Nummer >= 1 ) AND ( TK25_Nummer <= 9999 ) ),
Strassenklasse
    CHAR(1) NOT NULL,
Strassennummer
    INTEGER NOT NULL,
Zusatz_Strassennummer
    CHAR(1) NOT NULL
)
```

## 12.2 Teilschema „Umfelddaten“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Umfelddaten“.

```
CREATE SCHEMA Umfeldmessstelle
```

```
CREATE TABLE Umfeldmessstelle (
    Umfeldmessstelle_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Umfeldmessstelle_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Nummer
        INTEGER NOT NULL,
    hat_verwaltungstechn_Zuordnung
        INTEGER NOT NULL REFERENCES Allgemeine_Objekte.verwaltungstechn_Zuordnung,
    hat_oertliche_Zuordnung
        INTEGER NOT NULL REFERENCES Allgemeine_Objekte.oertliche_Zuordnung,
    hat_Erfassungsmodalitaeten
        INTEGER NOT NULL REFERENCES Erfassungsmodalitaeten
)
```

```
CREATE TABLE Umfmessst__Messeinr_Schadstimm (
    Umfeldmessstelle
        INTEGER NOT NULL REFERENCES Umfeldmessstelle,
    Messeinricht_Schadstoffimmiss
        VARCHAR(99) NOT NULL
)
```

```
CREATE TABLE Umfmessst__hat_Messdaten (
    Umfeldmessstelle
        INTEGER NOT NULL REFERENCES Umfeldmessstelle,
    Messdaten
        INTEGER NOT NULL REFERENCES Messdaten
)
```

```
CREATE TABLE Umfmessst__hat_Det_Umfdatenerf (
    Umfeldmessstelle
        INTEGER NOT NULL REFERENCES Umfeldmessstelle,
    Detektoren_Umfelddatenerfassg
        INTEGER NOT NULL REFERENCES Detektoren_Umfelddatenerfassg
)
```

```
CREATE TABLE VBA_Wetterstation (
    VBA_Wetterstation_ID
        INTEGER NOT NULL PRIMARY KEY,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
FOREIGN KEY (VBA_Wetterstation_ID) REFERENCES Umfeldmessstelle,
Nutzungsart
    VARCHAR(50)
)

CREATE TABLE SWIS_GMA_Messstelle (
    SWIS_GMA_Messstelle_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (SWIS_GMA_Messstelle_ID) REFERENCES Umfeldmessstelle,
    Nutzungsart
        VARCHAR(30)
)

CREATE TABLE Messdaten (
    Messdaten_ID
        INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Messdaten__hat_Umfang_Umfdaten (
    Messdaten
        INTEGER NOT NULL REFERENCES Messdaten,
    Umfang_Umfelddaten
        INTEGER NOT NULL REFERENCES Umfang_Umfelddaten
)

CREATE TABLE Messdaten__hat_Zaehlintervlge (
    Messdaten
        INTEGER NOT NULL REFERENCES Messdaten,
    Zaehlintervallaenge
        INTEGER NOT NULL REFERENCES
Automatische_Dauerzaehlstelle.Zaehlintervallaenge
)

CREATE TABLE Erfassungsmodalitaeten (
    Erfassungsmodalitaeten_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_Erfassungsbereich
        INTEGER NOT NULL REFERENCES Erfassungsbereich,
    hat_Zaehlzeitraum
        INTEGER NOT NULL REFERENCES Zaehlzeitraum
)

CREATE TABLE Lage_der_Sensoren (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Lage_der_Sensoren VALUES (1,'in der Fahrbahn')
INSERT INTO Lage_der_Sensoren VALUES (2,'neben der Fahrbahn')
INSERT INTO Lage_der_Sensoren VALUES (3,'in und neben der Fahrbahn')
INSERT INTO Lage_der_Sensoren VALUES (4,'außerhalb des Straßenbereichs')

CREATE TABLE Erfassungsbereich (
    Erfassungsbereich_ID
        INTEGER NOT NULL PRIMARY KEY,
    von_Netzknoten
        INTEGER NOT NULL,
    nach_Netzknoten
        INTEGER NOT NULL,
    Lage_der_Sensoren
        INTEGER NOT NULL REFERENCES Lage_der_Sensoren
)

CREATE TABLE Zaehlzeitraum (
    Zaehlzeitraum_ID
        INTEGER NOT NULL PRIMARY KEY,
    Beginn_Erfassung
        CHAR(10) NOT NULL, -- Datum
    CHECK ( Beginn_Erfassung LIKE '___.__.____' ),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    Ende_Erfassung
      CHAR(10), -- Datum
    CHECK ( Ende_Erfassung LIKE '__.__.____' )
  )

CREATE TABLE Detektoren_Umfelddatenerfassg (
  Detektoren_Umfelddatenerfassg_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Detekt_Umfdaterf__zugeh_EAKs (
  Detektoren_Umfelddatenerfassg
    INTEGER NOT NULL REFERENCES Detektoren_Umfelddatenerfassg,
  zugehoerige_EAKs
    VARCHAR(99) NOT NULL
)

CREATE TABLE Detekt_Umfdaterf__Einsatzzeit (
  Detektoren_Umfelddatenerfassg
    INTEGER NOT NULL REFERENCES Detektoren_Umfelddatenerfassg,
  Einsatzzeit
    VARCHAR(50) NOT NULL
)

CREATE TABLE Detekt_Umfdaterf__Betriebszeit (
  Detektoren_Umfelddatenerfassg
    INTEGER NOT NULL REFERENCES Detektoren_Umfelddatenerfassg,
  Betriebszeit
    VARCHAR(50) NOT NULL
)

CREATE TABLE Umfang_Umfelddaten (
  Umfang_Umfelddaten_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Umf_Umfelddat__erfasste_Daten (
  Umfang_Umfelddaten
    INTEGER NOT NULL REFERENCES Umfang_Umfelddaten,
  erfasste_Daten
    INTEGER NOT NULL
)

CREATE TABLE Umf_Umfelddat__Erm_Schadstimm (
  Umfang_Umfelddaten
    INTEGER NOT NULL REFERENCES Umfang_Umfelddaten,
  Ermittlung_Schadstoffimmiss
    VARCHAR(99) NOT NULL
)

CREATE TABLE Umf_Umfelddat__hat_Wittergsdat (
  Umfang_Umfelddaten
    INTEGER NOT NULL REFERENCES Umfang_Umfelddaten,
  Witterungsdaten
    INTEGER NOT NULL REFERENCES Witterungsdaten
)

CREATE TABLE Umf_Umfelddat__hat_Umfelddaten (
  Umfang_Umfelddaten
    INTEGER NOT NULL REFERENCES Umfang_Umfelddaten,
  Umfelddaten
    INTEGER NOT NULL REFERENCES Umfelddaten
)

CREATE TABLE Umf_Umfelddat__hat_witb_Strzst (
  Umfang_Umfelddaten
    INTEGER NOT NULL REFERENCES Umfang_Umfelddaten,
  witterungsbed_Strassenzustand
    INTEGER NOT NULL REFERENCES witterungsbed_Strassenzustand
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Witterungsdaten (
  Witterungsdaten_ID
    INTEGER NOT NULL PRIMARY KEY,
  Luftdruck
    INTEGER, -- Hektopascal
  CHECK ( ( Luftdruck >= 800 ) AND ( Luftdruck <= 1200 ) ),
  relative_Luftfeuchte
    REAL, -- Prozent
  Lufttemperatur
    REAL, -- Grad_Celsius
  CHECK ( Lufttemperatur>=0 ),
  Windrichtung
    INTEGER, -- Grad
  CHECK ( ( Windrichtung >= 0 ) AND ( Windrichtung <= 359 ) ),
  mittlere_Windgeschwindigkeit
    REAL, -- Meter_pro_Sekunde
  CHECK ( mittlere_Windgeschwindigkeit>=0 ),
  CHECK ( mittlere_Windgeschwindigkeit>=0 ),
  Spitzenwindgeschwindigkeit
    REAL, -- Meter_pro_Sekunde
  CHECK ( Spitzenwindgeschwindigkeit>=0 ),
  CHECK ( Spitzenwindgeschwindigkeit>=0 ),
  Niederschlagsart
    INTEGER,
  Niederschlagsintensitaet
    REAL, -- Millimeter_pro_Stunde
  CHECK ( Niederschlagsintensitaet>=0 ),
  CHECK ( Niederschlagsintensitaet>=0 ),
  Niederschlagsmenge
    REAL, -- Millimeter
  CHECK ( Niederschlagsmenge>=0 ),
  CHECK ( Niederschlagsintensitaet<=200 ),
  CHECK ( Spitzenwindgeschwindigkeit<=60 ),
  CHECK ( mittlere_Windgeschwindigkeit<=60 ),
  CHECK ( ( Lufttemperatur >= -30 ) AND ( Lufttemperatur <= 60 ) ),
  CHECK ( ( relative_Luftfeuchte >= 10 ) AND ( relative_Luftfeuchte <= 100 ) )
)

CREATE TABLE Umfelddaten (
  Umfelddaten_ID
    INTEGER NOT NULL PRIMARY KEY,
  Sichtweite
    REAL, -- Meter
  CHECK ( Sichtweite>=0 ),
  Helligkeit
    INTEGER, -- Lux
  CHECK ( Helligkeit>=0 ),
  CHECK ( ( Helligkeit >= 0 ) AND ( Helligkeit <= 60000 ) ),
  CHECK ( ( Sichtweite >= 10 ) AND ( Sichtweite <= 1000 ) )
)

CREATE TABLE witterungsbed_Strassenzustand (
  witterungsbed_Strassenzustand_ID
    INTEGER NOT NULL PRIMARY KEY,
  Zustand_Fahrbahnoberflaeche
    INTEGER,
  Wasserfilmdicke
    REAL, -- Millimeter
  CHECK ( Wasserfilmdicke>=0 ),
  Fahrbahnoberflaechentemperatur
    REAL, -- Grad_Celsius
  CHECK ( Fahrbahnoberflaechentemperatur>=0 ),
  Taupunkttemperatur
    REAL, -- Grad_Celsius
  CHECK ( Taupunkttemperatur>=0 ),
  Gefrierpunkttemperatur
    REAL, -- Grad_Celsius
  CHECK ( Gefrierpunkttemperatur>=0 ),
  Restsalzgehalt
    REAL, -- Prozent
  Schneehoehe

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    REAL, -- Zentimeter
CHECK ( Schneehoehe>=0 ),
Bodentemperatur_in_Tiefe_1
    REAL, -- Grad_Celsius
CHECK ( Bodentemperatur_in_Tiefe_1>=0 ),
Bodentemperatur_in_Tiefe_2
    REAL, -- Grad_Celsius
CHECK ( Bodentemperatur_in_Tiefe_2>=0 ),
Bodentemperatur_in_Tiefe_3
    REAL, -- Grad_Celsius
CHECK ( Bodentemperatur_in_Tiefe_3>=0 ),
CHECK ( ( Bodentemperatur_in_Tiefe_3 >= -30 ) AND ( Bodentemperatur_in_Tiefe_3
<= 80 ) ),
CHECK ( ( Bodentemperatur_in_Tiefe_2 >= -30 ) AND ( Bodentemperatur_in_Tiefe_2
<= 80 ) ),
CHECK ( ( Bodentemperatur_in_Tiefe_1 >= -30 ) AND ( Bodentemperatur_in_Tiefe_1
<= 80 ) ),
CHECK ( ( Schneehoehe >= 0 ) AND ( Schneehoehe <= 200 ) ),
CHECK ( ( Restsalzgehalt >= 0 ) AND ( Restsalzgehalt <= 100 ) ),
CHECK ( ( Gefrierpunkttemperatur >= -30 ) AND ( Gefrierpunkttemperatur <= 0 )
),
CHECK ( ( Taupunkttemperatur >= -30 ) AND ( Taupunkttemperatur <= 60 ) ),
CHECK ( ( Fahrbahnoberflaechentemperatur >= -30 ) AND (
Fahrbahnoberflaechentemperatur <= 80 ) )
)
```

### 12.3 Teilschema „Unfalldaten“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Unfall“.

```
CREATE SCHEMA Unfall
```

```
CREATE TABLE Unfall (
    Unfall_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_Angaben_zur_Unfallzeit
        INTEGER NOT NULL REFERENCES Angaben_zur_Unfallzeit,
    hat_Unfallnummer
        INTEGER REFERENCES Unfallnummer,
    hat_Angaben_zum_Unfallort
        INTEGER NOT NULL REFERENCES Angaben_zum_Unfallort,
    hat_Angaben_z_Unfalleinteilung
        INTEGER NOT NULL REFERENCES Angaben_zur_Unfalleinteilung
)
```

```
CREATE TABLE Unfall__hat_Angaben_zum_Unfall (
    Unfall
        INTEGER NOT NULL REFERENCES Unfall,
    Angaben_zum_Unfall
        INTEGER NOT NULL REFERENCES Angaben_zum_Unfall
)
```

```
CREATE TABLE Unfall__hat_Freiziffern (
    Unfall
        INTEGER NOT NULL REFERENCES Unfall,
    Freiziffern
        INTEGER NOT NULL REFERENCES Freiziffern
)
```

```
CREATE TABLE Unfall__hat_Unfallumstaende (
    Unfall
        INTEGER NOT NULL REFERENCES Unfall,
    Unfallumstaende
        INTEGER NOT NULL REFERENCES Unfallumstaende
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Angaben_zum_Unfall (
  Angaben_zum_Unfall_ID
    INTEGER NOT NULL PRIMARY KEY,
  Anzahl_der_Beteiligten
    INTEGER NOT NULL, -- Anzahl_zweistellig
  CHECK ( ( Anzahl_der_Beteiligten >= 0 ) AND ( Anzahl_der_Beteiligten <= 99 ) ),
  Anzahl_der_Getoeteten
    INTEGER NOT NULL, -- Anzahl_zweistellig
  CHECK ( ( Anzahl_der_Getoeteten >= 0 ) AND ( Anzahl_der_Getoeteten <= 99 ) ),
  Anzahl_Schwerverletzte
    INTEGER NOT NULL, -- Anzahl_zweistellig
  CHECK ( ( Anzahl_Schwerverletzte >= 0 ) AND ( Anzahl_Schwerverletzte <= 99 ) ),
  Anzahl_Leichtverletzte
    INTEGER NOT NULL, -- Anzahl_zweistellig
  CHECK ( ( Anzahl_Leichtverletzte >= 0 ) AND ( Anzahl_Leichtverletzte <= 99 ) ),
  CHECK ( Anzahl_der_Beteiligten>=1 )
)

CREATE TABLE Angaben_zur_Unfallzeit (
  Angaben_zur_Unfallzeit_ID
    INTEGER NOT NULL PRIMARY KEY,
  Wochentag_Unfall
    INTEGER REFERENCES Allgemeine_Objekte.Wochentag,
  Datum_Unfall
    CHAR(10) NOT NULL, -- Datum
  CHECK ( Datum_Unfall LIKE '___.__.____' ),
  Uhrzeit
    VARCHAR(12) NOT NULL, -- Uhrzeit
  CHECK ( ( Uhrzeit LIKE '__:___' ) OR ( Uhrzeit LIKE '__:__:__' ) OR ( Uhrzeit
  LIKE '__:__:__:____' ) )
)

CREATE TABLE Unfallnummer (
  Unfallnummer_ID
    INTEGER NOT NULL PRIMARY KEY,
  Satzmerkmale
    INTEGER NOT NULL,
  Paginier_Nummer
    INTEGER
)

CREATE TABLE Freiziffern (
  Freiziffern_ID
    INTEGER NOT NULL PRIMARY KEY,
  Ziffer_1
    INTEGER NOT NULL,
  Ziffer_2
    INTEGER NOT NULL
)

CREATE TABLE Angaben_zum_Unfallort (
  Angaben_zum_Unfallort_ID
    INTEGER NOT NULL PRIMARY KEY,
  Fahrtrichtung_WB
    VARCHAR(30) NOT NULL,
  Ortslage
    INTEGER NOT NULL REFERENCES Ortslage,
  hat_verwaltungstechn_Zuordnung
    INTEGER NOT NULL REFERENCES Allgemeine_Objekte.verwaltungstechn_Zuordnung,
  hat_oertliche_Zuordnung
    INTEGER NOT NULL REFERENCES Allgemeine_Objekte.oertliche_Zuordnung
)

CREATE TABLE Ang_z_Unfort__hat_Ang_z_einm_S (
  Angaben_zum_Unfallort
    INTEGER NOT NULL REFERENCES Angaben_zum_Unfallort,
  Angaben_zur_einmuend_Strasse
    INTEGER NOT NULL REFERENCES Angaben_zur_einmuend_Strasse
)

CREATE TABLE Angaben_zur_einmuend_Strasse (
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Angaben_zur_einmuend_Strasse_ID
    INTEGER NOT NULL PRIMARY KEY,
Klasse_Nr_1
    INTEGER NOT NULL,
Klasse_Nr_2
    INTEGER NOT NULL,
Schluessel
    INTEGER
)

CREATE TABLE Ortslage (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Ortslage VALUES (1,'innerorts')
INSERT INTO Ortslage VALUES (2,'außerorts')

CREATE TABLE Unfallumstaende (
    Unfallumstaende_ID
        INTEGER NOT NULL PRIMARY KEY,
    Strassenbefestigung
        INTEGER REFERENCES Strassenbefestigung,
    Geschwindigkeitsbegrenzung
        INTEGER, -- Stundenkilometer
    CHECK ( Geschwindigkeitsbegrenzung >= 0 ),
    Lichtzeichenanlage
        INTEGER REFERENCES Lichtzeichenanlage,
    Lichtverhaeltnisse
        INTEGER REFERENCES Lichtverhaeltnisse,
    CHECK ( ( Geschwindigkeitsbegrenzung >= 5 ) AND ( Geschwindigkeitsbegrenzung <=
130 ) )
)

CREATE TABLE Unfumstaende__Witterverhaeltn (
    Unfallumstaende
        INTEGER NOT NULL REFERENCES Unfallumstaende,
    Witterungsverhaeltnisse
        INTEGER NOT NULL REFERENCES Witterungsverhaeltnisse
)

CREATE TABLE Unfumstaende__Strassenzustand (
    Unfallumstaende
        INTEGER NOT NULL REFERENCES Unfallumstaende,
    Strassenzustand
        INTEGER NOT NULL REFERENCES Strassenzustand
)

CREATE TABLE Unfumstaende__hat_Char_d_Unfst (
    Unfallumstaende
        INTEGER NOT NULL REFERENCES Unfallumstaende,
    Charakterist_der_Unfallstelle
        INTEGER NOT NULL REFERENCES Charakterist_der_Unfallstelle
)

CREATE TABLE Unfumstde__hat_Besond_d_Unfst (
    Unfallumstaende
        INTEGER NOT NULL REFERENCES Unfallumstaende,
    Besonderheiten_d_Unfallstelle
        INTEGER NOT NULL REFERENCES Besonderheiten_d_Unfallstelle
)

CREATE TABLE Witterungsverhaeltnisse (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Witterungsverhaeltnisse VALUES (7,'Regen')
INSERT INTO Witterungsverhaeltnisse VALUES (8,'Schneefall/Hagel')
INSERT INTO Witterungsverhaeltnisse VALUES (9,'Nebel/Dunst')
INSERT INTO Witterungsverhaeltnisse VALUES (0,'Sturm')

CREATE TABLE Strassenzustand (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Strassenzustand VALUES (0,'trocken')
INSERT INTO Strassenzustand VALUES (1,'feucht')
INSERT INTO Strassenzustand VALUES (2,'Glatteis')
INSERT INTO Strassenzustand VALUES (3,'Schneeglätte')
INSERT INTO Strassenzustand VALUES (4,'gestreut')
INSERT INTO Strassenzustand VALUES (5,'Schlüpfrigkeit')
INSERT INTO Strassenzustand VALUES (6,'schadhafte Fahrbahn')
INSERT INTO Strassenzustand VALUES (7,'winterglatt')

CREATE TABLE Strassenbefestigung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Strassenbefestigung VALUES (5,'Beton')
INSERT INTO Strassenbefestigung VALUES (6,'Schwarzdecke')
INSERT INTO Strassenbefestigung VALUES (7,'Pflaster')
INSERT INTO Strassenbefestigung VALUES (8,'sonstige befestigte Straße')
INSERT INTO Strassenbefestigung VALUES (9,'unbefestigte Straße')

CREATE TABLE Lichtzeichenanlage (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Lichtzeichenanlage VALUES (0,'keine LZA')
INSERT INTO Lichtzeichenanlage VALUES (1,'LZA vorhanden')

CREATE TABLE Lichtverhaeltnisse (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Lichtverhaeltnisse VALUES (0,'Tageslicht')
INSERT INTO Lichtverhaeltnisse VALUES (1,'Dämmerung')
INSERT INTO Lichtverhaeltnisse VALUES (2,'Dunkelheit')

CREATE TABLE Charakterist_der_Unfallstelle (
  Charakterist_der_Unfallstelle_ID
    INTEGER NOT NULL PRIMARY KEY,
  Charakteristik_1
    INTEGER NOT NULL REFERENCES Charakteristik_Unfst,
  Charakteristik_2
    INTEGER NOT NULL REFERENCES Charakteristik_Unfst,
  Charakteristik_3
    INTEGER NOT NULL REFERENCES Charakteristik_Unfst,
  CHECK ( Charakteristik_3>=3 ),
  CHECK ( Charakteristik_2>=2 )
)

CREATE TABLE Charakteristik_Unfst (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Charakteristik_Unfst VALUES (1,'Kreuzung')
INSERT INTO Charakteristik_Unfst VALUES (2,'Einmündung')
INSERT INTO Charakteristik_Unfst VALUES (3,'Ein-/Ausfahrt Grundstück')
INSERT INTO Charakteristik_Unfst VALUES (4,'Steigung/Gefälle')
INSERT INTO Charakteristik_Unfst VALUES (5,'Kuppe')
INSERT INTO Charakteristik_Unfst VALUES (6,'Kurve')
INSERT INTO Charakteristik_Unfst VALUES (7,'frei')
INSERT INTO Charakteristik_Unfst VALUES (8,'Steigung')
INSERT INTO Charakteristik_Unfst VALUES (9,'Gefälle')

CREATE TABLE Besonderheiten_d_Unfallstelle (
    Besonderheiten_d_Unfallstelle_ID
        INTEGER NOT NULL PRIMARY KEY,
    Besonderheit_1
        INTEGER NOT NULL REFERENCES Besonderheiten_Unfst,
    Besonderheit_2
        INTEGER NOT NULL REFERENCES Besonderheiten_Unfst,
    Besonderheit_3
        INTEGER NOT NULL REFERENCES Besonderheiten_Unfst,
    CHECK ( Besonderheit_3>=3 ),
    CHECK ( Besonderheit_2>=2 )
)

CREATE TABLE Besonderheiten_Unfst (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Besonderheiten_Unfst VALUES (1,'unübersichtlich')
INSERT INTO Besonderheiten_Unfst VALUES (2,'schiengleicher Wegübergang')
INSERT INTO Besonderheiten_Unfst VALUES (3,'Zebrastreifen')
INSERT INTO Besonderheiten_Unfst VALUES (4,'Fußgängerfurt')
INSERT INTO Besonderheiten_Unfst VALUES (5,'Haltestelle')
INSERT INTO Besonderheiten_Unfst VALUES (6,'Arbeitsstelle')
INSERT INTO Besonderheiten_Unfst VALUES (7,'verkehrsberuhigter Bereich')
INSERT INTO Besonderheiten_Unfst VALUES (8,'Querungshilfe')
INSERT INTO Besonderheiten_Unfst VALUES (9,'Kreisverkehr')

CREATE TABLE Beteiligte (
    Beteiligte_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_der_Verkehrsbeteiligung
        INTEGER, -- Art_der_Verkehrsbeteiligung
    CHECK ( ( Art_der_Verkehrsbeteiligung >= 1 ) AND ( Art_der_Verkehrsbeteiligung
<= 93 ) ),
    Alter_Beteiligter
        INTEGER,
    Geschlecht_Beteiligter
        INTEGER NOT NULL REFERENCES Geschlecht,
    Anzahl_Fahrzeugbenutzer
        INTEGER NOT NULL, -- Anzahl_zweistellig
    CHECK ( ( Anzahl_Fahrzeugbenutzer >= 0 ) AND ( Anzahl_Fahrzeugbenutzer <= 99 )
),
    von_Unfall
        INTEGER REFERENCES Unfall
)

CREATE TABLE Beteiligte__Unfallursachen (
    Beteiligte
        INTEGER NOT NULL REFERENCES Beteiligte,
    Unfallursachen
        INTEGER NOT NULL,
    CHECK ( ( Unfallursache >= 1 ) AND ( Unfallursache <= 69 ) )
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Beteil__hat_weitergeh_Angaben (
  Beteiligte
    INTEGER NOT NULL REFERENCES Beteiligte,
  weitergehende_Angaben
    INTEGER NOT NULL REFERENCES weitergehende_Angaben
)

CREATE TABLE Geschlecht (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Geschlecht VALUES (1,'w')
INSERT INTO Geschlecht VALUES (2,'m')

CREATE TABLE Mitfahrerangaben (
  Mitfahrerangaben_ID
    INTEGER NOT NULL PRIMARY KEY,
  getoetete_Mitfahrer
    INTEGER NOT NULL, -- Anzahl_zweistellig
  CHECK ( ( getoetete_Mitfahrer >= 0 ) AND ( getoetete_Mitfahrer <= 99 ) ),
  schwerverletzte_Mitfahrer
    INTEGER NOT NULL, -- Anzahl_zweistellig
  CHECK ( ( schwerverletzte_Mitfahrer >= 0 ) AND ( schwerverletzte_Mitfahrer <=
99 ) ),
  leichtverletzte_Mitfahrer
    INTEGER NOT NULL, -- Anzahl_zweistellig
  CHECK ( ( leichtverletzte_Mitfahrer >= 0 ) AND ( leichtverletzte_Mitfahrer <=
99 ) ),
  zu_Beteiligtem
    INTEGER REFERENCES Beteiligte
)

CREATE TABLE Mitfang__hat_Ang_z_Alt_u_Gesch (
  Mitfahrerangaben
    INTEGER NOT NULL REFERENCES Mitfahrerangaben,
  Angaben_z_Alter_und_Geschlecht
    INTEGER NOT NULL REFERENCES Angaben_z_Alter_und_Geschlecht
)

CREATE TABLE Angaben_z_Alter_und_Geschlecht (
  Angaben_z_Alter_und_Geschlecht_ID
    INTEGER NOT NULL PRIMARY KEY,
  Alter_Mitfahrer
    INTEGER NOT NULL,
  Geschlecht_Mitfahrer
    INTEGER NOT NULL REFERENCES Geschlecht
)

CREATE TABLE weitergehende_Angaben (
  weitergehende_Angaben_ID
    INTEGER NOT NULL PRIMARY KEY,
  Kennzeichen
    INTEGER NOT NULL,
  Ortskenntnisse
    INTEGER NOT NULL REFERENCES Logical_Types.Boolean,
  Auslaenderkennung
    INTEGER NOT NULL,
  Art_der_Verletzung
    INTEGER NOT NULL REFERENCES Art_der_Verletzung,
  Angaben_zum_Fahrzeug
    INTEGER NOT NULL
)

CREATE TABLE Art_der_Verletzung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

INSERT INTO Art_der_Verletzung VALUES (1,'getötet')
INSERT INTO Art_der_Verletzung VALUES (2,'schwerverletzt')
INSERT INTO Art_der_Verletzung VALUES (3,'leichtverletzt')

CREATE TABLE Angaben_zur_Unfalleinteilung (
  Angaben_zur_Unfalleinteilung_ID
    INTEGER NOT NULL PRIMARY KEY,
  allgemeine_Unfallursache_1
    INTEGER NOT NULL, -- allgemeine_Unfallursache
  CHECK ( ( allgemeine_Unfallursache_1 >= 70 ) AND ( allgemeine_Unfallursache_1
<= 89 ) ),
  allgemeine_Unfallursache_2
    INTEGER, -- allgemeine_Unfallursache
  CHECK ( ( allgemeine_Unfallursache_2 >= 70 ) AND ( allgemeine_Unfallursache_2
<= 89 ) ),
  Unfalltyp
    INTEGER NOT NULL,
  Unfallart
    INTEGER NOT NULL,
  Unfallkategorie
    INTEGER NOT NULL,
  hat_Aufprall_auf_Hindernis
    INTEGER REFERENCES Aufprall_auf_Hindernis,
  CHECK ( ( Unfallkategorie >= 1 ) AND ( Unfallkategorie <= 9 ) ),
  CHECK ( ( Unfallart >= 0 ) AND ( Unfallart <= 9 ) ),
  CHECK ( ( Unfalltyp >= 100 ) AND ( Unfalltyp <= 799 ) ),
  CHECK ( allgemeine_Unfallursache_2>=71 )
)

CREATE TABLE Aufprall_auf_Hindernis (
  Aufprall_auf_Hindernis_ID
    INTEGER NOT NULL PRIMARY KEY,
  Hindernisart
    INTEGER NOT NULL REFERENCES Hindernisart,
  Fahrzustand
    INTEGER REFERENCES Fahrzustand
)

CREATE TABLE Hindernisart (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Hindernisart VALUES (0,'Baum')
INSERT INTO Hindernisart VALUES (1,'Mast')
INSERT INTO Hindernisart VALUES (2,'Widerlager')
INSERT INTO Hindernisart VALUES (3,'Schutzplanke')
INSERT INTO Hindernisart VALUES (4,'sonstiges Hindernis')
INSERT INTO Hindernisart VALUES (5,'kein Aufprall')

CREATE TABLE Fahrzustand (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Fahrzustand VALUES (0,'Fahrzeug fahrbereit')
INSERT INTO Fahrzustand VALUES (1,'nicht fahrbereites Kfz')
```

## 13 Fachbereich „Beschilderung“

### 13.1 Teilschema „Dynamische verkehrsregelnde Beschilderung“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Dynamische Beschilderung“.

```
CREATE SCHEMA Dynamische_Beschilderung

CREATE TABLE dyn_verkehrsreg_Beschilderung (
  dyn_verkehrsreg_Beschilderung_ID
    INTEGER NOT NULL PRIMARY KEY,
  Name
    VARCHAR(50),
  hat_Einsatzdauer
    INTEGER NOT NULL REFERENCES Einsatzdauer
)

CREATE TABLE dyn_vreg_Besch__in_Komb_v_Anlt (
  dyn_verkehrsreg_Beschilderung
    INTEGER NOT NULL REFERENCES dyn_verkehrsreg_Beschilderung,
  Kombination_von_Anlagentypen
    INTEGER NOT NULL REFERENCES Kombination_von_Anlagentypen
)

CREATE TABLE dyn_vreg_Besch__hat_verw_Zuord (
  dyn_verkehrsreg_Beschilderung
    INTEGER NOT NULL REFERENCES dyn_verkehrsreg_Beschilderung,
  verwaltungstechn_Zuordnung_dyn
    INTEGER NOT NULL REFERENCES verwaltungstechn_Zuordnung_dyn
)

CREATE TABLE dyn_vreg_Besch__hat_Anlagausst (
  dyn_verkehrsreg_Beschilderung
    INTEGER NOT NULL REFERENCES dyn_verkehrsreg_Beschilderung,
  Anlagenausstattung
    INTEGER NOT NULL REFERENCES Anlagenausstattung
)

CREATE TABLE Kombination_von_Anlagentypen (
  Kombination_von_Anlagentypen_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Kombination_von_Anlagentypen_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' )
)

CREATE TABLE verwaltungstechn_Zuordnung_dyn (
  verwaltungstechn_Zuordnung_dyn_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE verwtechn_Zuord_dyn__hat_Vwbez (
  verwaltungstechn_Zuordnung_dyn
    INTEGER NOT NULL REFERENCES verwaltungstechn_Zuordnung_dyn,
  Verwaltungsbezirk
    INTEGER NOT NULL REFERENCES Administration.Verwaltungsbezirk
)

CREATE TABLE verwtechn_Zuord_dyn__hat_Strbd (
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
verwaltungstechn_Zuordnung_dyn
    INTEGER NOT NULL REFERENCES verwaltungstechn_Zuordnung_dyn,
Strassenbaudienststelle
    INTEGER NOT NULL REFERENCES Administration.Strassenbaudienststelle
)

CREATE TABLE Einsatzdauer (
    Einsatzdauer_ID
        INTEGER NOT NULL PRIMARY KEY,
    Dauer_Art
        INTEGER NOT NULL REFERENCES Art_der_Dauer,
    Dauer_Laenge
        VARCHAR(30) NOT NULL
)

CREATE TABLE Art_der_Dauer (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Art_der_Dauer VALUES (1,'vorübergehend')
INSERT INTO Art_der_Dauer VALUES (2,'längerfristig')

CREATE TABLE Streckenbeeinflussung (
    Streckenbeeinflussung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Streckenbeeinflussung_ID) REFERENCES dyn_verkehrsreg_Beschilderung
)

CREATE TABLE Streckbeeinfl__hat_Anlagentyp (
    Streckenbeeinflussung
        INTEGER NOT NULL REFERENCES Streckenbeeinflussung,
    SB_Anlagentyp
        INTEGER NOT NULL REFERENCES SB_Anlagentyp
)

CREATE TABLE Streckbeeinfl__hat_Anzeigzust (
    Streckenbeeinflussung
        INTEGER NOT NULL REFERENCES Streckenbeeinflussung,
    SB_Anzeigzustaende
        INTEGER NOT NULL REFERENCES SB_Anzeigzustaende
)

CREATE TABLE Streckbeeinfl__hat_Wirkbereich (
    Streckenbeeinflussung
        INTEGER NOT NULL REFERENCES Streckenbeeinflussung,
    SB_Wirkungsbereiche
        INTEGER NOT NULL REFERENCES SB_Wirkungsbereiche
)

CREATE TABLE SB_Anlagentyp (
    SB_Anlagentyp_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (SB_Anlagentyp_ID) REFERENCES Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Typ
        INTEGER NOT NULL REFERENCES Tab_SB_Anlagentyp
)

CREATE TABLE Tab_SB_Anlagentyp (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

INSERT INTO Tab_SB_Anlagentyp VALUES (1,'Geschwindigkeitsbeeinflussung')
INSERT INTO Tab_SB_Anlagentyp VALUES (2,'Stauwarnanlage fest')
INSERT INTO Tab_SB_Anlagentyp VALUES (3,'Stauwarnanlage mobil')
INSERT INTO Tab_SB_Anlagentyp VALUES (4,'Tunnelsteuerung')
INSERT INTO Tab_SB_Anlagentyp VALUES (5,'Fahrstreifenzuteilung / ' ||
'Fahrstreifensperrung')
INSERT INTO Tab_SB_Anlagentyp VALUES (6,'Warnung vor witterungsbedingten ' ||
'Gefahren')
INSERT INTO Tab_SB_Anlagentyp VALUES (7,'Langsamfahrzeugkennung')
INSERT INTO Tab_SB_Anlagentyp VALUES (8,'sonst')

CREATE TABLE SB_Anzeigezustaende (
  SB_Anzeigezustaende_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (SB_Anzeigezustaende_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  StVO_Zeichen
    VARCHAR(99) NOT NULL,
  Zusatzzeichen
    VARCHAR(50) NOT NULL
)

CREATE TABLE SB_Wirkungsbereiche (
  SB_Wirkungsbereiche_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (SB_Wirkungsbereiche_ID) REFERENCES
Strassennetz.Streckenobjekt_hist,
  Bereich
    INTEGER NOT NULL REFERENCES Tab_Wirkungsbereich,
  Richtungsangabe
    VARCHAR(28)
)

CREATE TABLE SB_Wirkber__beginnt_b_Strpunkt (
  SB_Wirkungsbereiche
    INTEGER NOT NULL REFERENCES SB_Wirkungsbereiche,
  Wirkungsbereich_SP
    INTEGER NOT NULL REFERENCES Wirkungsbereich_SP
)

CREATE TABLE SB_Wirkber__endet_b_Strpunkt (
  SB_Wirkungsbereiche
    INTEGER NOT NULL REFERENCES SB_Wirkungsbereiche,
  Wirkungsbereich_SP
    INTEGER NOT NULL REFERENCES Wirkungsbereich_SP
)

CREATE TABLE Tab_Wirkungsbereich (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Tab_Wirkungsbereich VALUES (1,'Beeinflussungsbereich gilt für ' ||
'beide Richtungen')
INSERT INTO Tab_Wirkungsbereich VALUES (2,'Beeinflussungsbereich Richtung 1')
INSERT INTO Tab_Wirkungsbereich VALUES (3,'Beeinflussungsbereich Richtung 2')

CREATE TABLE Wirkungsbereich_SP (
  Wirkungsbereich_SP_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Wirkungsbereich_SP_ID) REFERENCES Strassennetz.Punktobjekt_stat
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

CREATE TABLE Wechselwegweisung (
    Wechselwegweisung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Wechselwegweisung_ID) REFERENCES dyn_verkehrsreg_Beschilderung,
    Anlagentyp
        INTEGER NOT NULL REFERENCES WW_Anlagentyp
)

CREATE TABLE Wechselwegw__hat_Anzustaende (
    Wechselwegweisung
        INTEGER NOT NULL REFERENCES Wechselwegweisung,
    WW_Anzeigezustaende
        INTEGER NOT NULL REFERENCES WW_Anzeigezustaende
)

CREATE TABLE Wechselwegw__hat_Wirkungsber (
    Wechselwegweisung
        INTEGER NOT NULL REFERENCES Wechselwegweisung,
    WW_Wirkungsbereiche
        INTEGER NOT NULL REFERENCES WW_Wirkungsbereiche
)

CREATE TABLE WW_Anzeigezustaende (
    WW_Anzeigezustaende_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (WW_Anzeigezustaende_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Anzeigezustand
        VARCHAR(99) NOT NULL
)

CREATE TABLE WW_Anlagentyp (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO WW_Anlagentyp VALUES (1,'additive Wechselwegweisung')
INSERT INTO WW_Anlagentyp VALUES (2,'substitutive Wechselwegweisung')

CREATE TABLE WW_Wirkungsbereiche (
    WW_Wirkungsbereiche_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (WW_Wirkungsbereiche_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Bereich
        INTEGER NOT NULL REFERENCES Tab_Wirkungsbereich,
    Richtungsangabe
        VARCHAR(28)
)

CREATE TABLE WW_Wirkungsber__betroffene_BAB (
    WW_Wirkungsbereiche
        INTEGER NOT NULL REFERENCES WW_Wirkungsbereiche,
    betroffene_BAB
        VARCHAR(99) NOT NULL
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)

CREATE TABLE WW_Wirkungsber__beginnt_bei_NK (
    WW_Wirkungsbereiche
        INTEGER NOT NULL REFERENCES WW_Wirkungsbereiche,
    Netzknoten
        INTEGER NOT NULL REFERENCES Strassennetz.Netzknoten
)

CREATE TABLE WW_Wirkungsber__endet_bei_NK (
    WW_Wirkungsbereiche
        INTEGER NOT NULL REFERENCES WW_Wirkungsbereiche,
    Netzknoten
        INTEGER NOT NULL REFERENCES Strassennetz.Netzknoten
)

CREATE TABLE Knotenpunktbeeinflussung (
    Knotenpunktbeeinflussung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Knotenpunktbeeinflussung_ID) REFERENCES
dyn_verkehrsreg_Beschilderung,
    hat_Wirkungsbereich
        INTEGER NOT NULL REFERENCES KB_Wirkungsbereich
)

CREATE TABLE Knotpktbeeinfl__hat_Anlagentyp (
    Knotenpunktbeeinflussung
        INTEGER NOT NULL REFERENCES Knotenpunktbeeinflussung,
    KB_Anlagentyp
        INTEGER NOT NULL REFERENCES KB_Anlagentyp
)

CREATE TABLE Knotpktbeeinfl__hat_Anzeigzust (
    Knotenpunktbeeinflussung
        INTEGER NOT NULL REFERENCES Knotenpunktbeeinflussung,
    KB_Anzeigezustaende
        INTEGER NOT NULL REFERENCES KB_Anzeigezustaende
)

CREATE TABLE KB_Anlagentyp (
    KB_Anlagentyp_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (KB_Anlagentyp_ID) REFERENCES Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '__.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '__.__.____' ),
    Typ
        INTEGER NOT NULL REFERENCES Tab_KB_Anlagentyp
)

CREATE TABLE Tab_KB_Anlagentyp (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Tab_KB_Anlagentyp VALUES (1,'Fahrstreifenzuteilung im Knotenpunkt')
INSERT INTO Tab_KB_Anlagentyp VALUES (2,'Geschwindigkeitsbeeinflussung im ' ||
'Einfädelsbereich' )
INSERT INTO Tab_KB_Anlagentyp VALUES (3,'Geschwindigkeitswarnanlage')
INSERT INTO Tab_KB_Anlagentyp VALUES (4,'Stauwarnanlage')
INSERT INTO Tab_KB_Anlagentyp VALUES (5,'sonst')

CREATE TABLE KB_Anzeigezustaende (
    KB_Anzeigezustaende_ID
        INTEGER NOT NULL PRIMARY KEY,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
FOREIGN KEY (KB_Anzeigezustande_ID) REFERENCES
Historisierung.historisches_Objekt,
gueltig_von
    CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_von LIKE '___.__.____' ),
gueltig_bis
    CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_bis LIKE '___.__.____' ),
Anzeigezustand
    VARCHAR(99) NOT NULL
)

CREATE TABLE KB_Wirkungsbereich (
    KB_Wirkungsbereich_ID
        INTEGER NOT NULL PRIMARY KEY,
    Bereich
        INTEGER NOT NULL REFERENCES Tab_KB_Wirkungsbereich,
    CHECK ( ( ist_richtungsbezogener_WB IS NOT NULL ) OR ( ist_astbezogener_WB IS
NOT NULL ) ) AND ( ( ist_richtungsbezogener_WB IS NULL ) OR ( ist_astbezogener_WB
IS NULL ) ) )
)

CREATE TABLE KB_Wirkber__ist_richtbez_WB (
    KB_Wirkungsbereich
        INTEGER NOT NULL REFERENCES KB_Wirkungsbereich,
    richtungsbezogener_WB
        INTEGER NOT NULL REFERENCES richtungsbezogener_WB
)

CREATE TABLE KB_Wirkber__ist_astbezog_WB (
    KB_Wirkungsbereich
        INTEGER NOT NULL REFERENCES KB_Wirkungsbereich,
    astbezogener_WB
        INTEGER NOT NULL REFERENCES astbezogener_WB
)

CREATE TABLE KB_Wirkber__hat_Ang_z_Knotpkt (
    KB_Wirkungsbereich
        INTEGER NOT NULL REFERENCES KB_Wirkungsbereich,
    Angaben_zum_Knotenpunkt
        INTEGER NOT NULL REFERENCES Angaben_zum_Knotenpunkt
)

CREATE TABLE Tab_KB_Wirkungsbereich (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Tab_KB_Wirkungsbereich VALUES (1,'Beeinflussungsbereich gilt ' ||
'für beide Richtungen')
INSERT INTO Tab_KB_Wirkungsbereich VALUES (2,'Beeinflussungsbereich für ' ||
'eine Richtung')

CREATE TABLE richtungsbezogener_WB (
    richtungsbezogener_WB_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (richtungsbezogener_WB_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Richtung_1
        VARCHAR(30) NOT NULL,
    Richtung_2
        VARCHAR(30)
)
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE astbezogener_WB (
  astbezogener_WB_ID
    INTEGER NOT NULL PRIMARY KEY,
  Rampenbezeichnung
    VARCHAR(50) NOT NULL
)

CREATE TABLE astbezogener_WB__Astkennung (
  astbezogener_WB
    INTEGER NOT NULL REFERENCES astbezogener_WB,
  Astkennung
    VARCHAR(5) NOT NULL
)

CREATE TABLE Angaben_zum_Knotenpunkt (
  Angaben_zum_Knotenpunkt_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Angaben_zum_Knotenpunkt_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Netzknotenangabe_1
    INTEGER NOT NULL,
  Netzknotenangabe_2
    INTEGER NOT NULL,
  Netzknotenangabe_3
    INTEGER NOT NULL
)

CREATE TABLE Richtungswechselbetrieb (
  Richtungswechselbetrieb_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Richtungswechselbetrieb_ID) REFERENCES
dyn_verkehrsreg_Beschilderung
)

CREATE TABLE Richtwechsbetr__hat_Betrform (
  Richtungswechselbetrieb
    INTEGER NOT NULL REFERENCES Richtungswechselbetrieb,
  Betriebsform
    INTEGER NOT NULL REFERENCES Betriebsform
)

CREATE TABLE Richtwechsbetr__hat_Anzeigzust (
  Richtungswechselbetrieb
    INTEGER NOT NULL REFERENCES Richtungswechselbetrieb,
  RW_Anzeigezustaende
    INTEGER NOT NULL REFERENCES RW_Anzeigezustaende
)

CREATE TABLE Richtwechsbetr__hat_RW_Wirkber (
  Richtungswechselbetrieb
    INTEGER NOT NULL REFERENCES Richtungswechselbetrieb,
  RW_Wirkungsbereich
    INTEGER NOT NULL REFERENCES RW_Wirkungsbereich
)

CREATE TABLE Betriebsform (
  Betriebsform_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Betriebsform_ID) REFERENCES Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CHECK ( gueltig_bis LIKE '___.__.____' ),
Betriebsform
    VARCHAR(30) NOT NULL
)

CREATE TABLE RW_Anzeigezustaende (
    RW_Anzeigezustaende_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (RW_Anzeigezustaende_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Anzeigezustand
        VARCHAR(99) NOT NULL
)

CREATE TABLE RW_Wirkungsbereich (
    RW_Wirkungsbereich_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (RW_Wirkungsbereich_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Bereich
        VARCHAR(30)
)

CREATE TABLE RW_Wirkber__beginnt_bei_Strpkt (
    RW_Wirkungsbereich
        INTEGER NOT NULL REFERENCES RW_Wirkungsbereich,
    Wirkungsbereich_SP
        INTEGER NOT NULL REFERENCES Wirkungsbereich_SP
)

CREATE TABLE RW_Wirkber__endet_bei_Strpkt (
    RW_Wirkungsbereich
        INTEGER NOT NULL REFERENCES RW_Wirkungsbereich,
    Wirkungsbereich_SP
        INTEGER NOT NULL REFERENCES Wirkungsbereich_SP
)

CREATE TABLE Anlagenausstattung (
    Anlagenausstattung_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_Anzeigesystem
        INTEGER NOT NULL REFERENCES Anzeigesystem
)

CREATE TABLE Anlausst__hat_Art_d_Stromvers (
    Anlagenausstattung
        INTEGER NOT NULL REFERENCES Anlagenausstattung,
    Art_der_Stromversorgung
        INTEGER NOT NULL REFERENCES Art_der_Stromversorgung
)

CREATE TABLE Anlausst__hat_Steuerung (
    Anlagenausstattung
        INTEGER NOT NULL REFERENCES Anlagenausstattung,
    Steuerung
        INTEGER NOT NULL REFERENCES Steuerung
)

CREATE TABLE Anlausst__hat_Ueberwach_d_Anl (
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Anlagenausstattung
    INTEGER NOT NULL REFERENCES Anlagenausstattung,
Ueberwachung_der_Anlage
    INTEGER NOT NULL REFERENCES Ueberwachung_der_Anlage
)

CREATE TABLE Anlausst__hat_Datenebertrsys (
    Anlagenausstattung
        INTEGER NOT NULL REFERENCES Anlagenausstattung,
    Dateneubertragungssystem
        INTEGER NOT NULL REFERENCES Dateneubertragungssystem
)

CREATE TABLE Anlausstatt__hat_Unterzentrale (
    Anlagenausstattung
        INTEGER NOT NULL REFERENCES Anlagenausstattung,
    Unterzentrale
        INTEGER NOT NULL REFERENCES Unterzentrale
)

CREATE TABLE Anlagenausstattung__hat_VRZ (
    Anlagenausstattung
        INTEGER NOT NULL REFERENCES Anlagenausstattung,
    Verkehrsrechnerzentrale
        INTEGER NOT NULL REFERENCES Verkehrsrechnerzentrale
)

CREATE TABLE Anlausstatt__hat_Datenerfeinr (
    Anlagenausstattung
        INTEGER NOT NULL REFERENCES Anlagenausstattung,
    Datenerfassungseinrichtungen
        INTEGER NOT NULL REFERENCES Datenerfassungseinrichtungen
)

CREATE TABLE Art_der_Stromversorgung (
    Art_der_Stromversorgung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Art_der_Stromversorgung_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Art
        INTEGER NOT NULL REFERENCES Tab_Art_der_Stromversorgung
)

CREATE TABLE Tab_Art_der_Stromversorgung (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Tab_Art_der_Stromversorgung VALUES (0,'keine Angaben')
INSERT INTO Tab_Art_der_Stromversorgung VALUES (1,'Zähleranschlußsäulen ' ||
'(lokale Versorgung)')
INSERT INTO Tab_Art_der_Stromversorgung VALUES (2,'Solarenergie')
INSERT INTO Tab_Art_der_Stromversorgung VALUES (3,'Längsverkabelung ' ||
'(zentrale Versorgung)')
INSERT INTO Tab_Art_der_Stromversorgung VALUES (4,'Kombination aus 1 und 2')
INSERT INTO Tab_Art_der_Stromversorgung VALUES (5,'sonstige Kombination')

CREATE TABLE Steuerung (
    Steuerung_ID
        INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Steuerung__autom_Steuerung (
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Steuerung
    INTEGER NOT NULL REFERENCES Steuerung,
    automatische_Steuerung
    INTEGER NOT NULL REFERENCES Logical_Types.Boolean
)

CREATE TABLE Steuerung__hat_manu_Steuerung (
    Steuerung
    INTEGER NOT NULL REFERENCES Steuerung,
    manuelle_Steuerung
    INTEGER NOT NULL REFERENCES manuelle_Steuerung
)

CREATE TABLE manuelle_Steuerung (
    manuelle_Steuerung_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (manuelle_Steuerung_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
    CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
    CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Steuerung
    INTEGER NOT NULL REFERENCES Tab_Steuerung,
    Bedienung
    VARCHAR(50)
)

CREATE TABLE Tab_Steuerung (
    Kennung
    INTEGER NOT NULL PRIMARY KEY,
    Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Tab_Steuerung VALUES (0,'kein manueller Eingriff möglich')
INSERT INTO Tab_Steuerung VALUES (1,'manueller Eingriff über UZ möglich')
INSERT INTO Tab_Steuerung VALUES (2,'nur manuelle Steuerung')
INSERT INTO Tab_Steuerung VALUES (3,'manueller Eingriff vor Ort und über ' ||
'Ferneingriff möglich')

CREATE TABLE Dateneubertragungssystem (
    Dateneubertragungssystem_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Dateneubertragungssystem_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
    CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
    CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Dateneubertragung_SST_UZ
    INTEGER NOT NULL REFERENCES Dateneubertragung_SST_UZ,
    Dateneubertragung_UZ_VRZ
    INTEGER NOT NULL REFERENCES Dateneubertragung_UZ_VRZ
)

CREATE TABLE Dateneubertragung_SST_UZ (
    Kennung
    INTEGER NOT NULL PRIMARY KEY,
    Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Dateneubertragung_SST_UZ VALUES (0,'keine UZ vorhanden')
INSERT INTO Dateneubertragung_SST_UZ VALUES (1,'AUSA-Netz')
INSERT INTO Dateneubertragung_SST_UZ VALUES (2,'Längsverkabelung')
INSERT INTO Dateneubertragung_SST_UZ VALUES (3,'Funk')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Dateneubertragung_SST_UZ VALUES (4,'lokale Steuerung vor Ort')

CREATE TABLE Dateneubertragung_UZ_VRZ (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Dateneubertragung_UZ_VRZ VALUES (0,'kein Anschluß vorhanden')
INSERT INTO Dateneubertragung_UZ_VRZ VALUES (1,'AUSA-Netz')
INSERT INTO Dateneubertragung_UZ_VRZ VALUES (2,'sonst')

CREATE TABLE Unterzentrale (
  Unterzentrale_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Unterzentrale_ID) REFERENCES Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Existenz
    INTEGER NOT NULL REFERENCES Existenz_Unterzentrale
)

CREATE TABLE Unterzentrale__Lage (
  Unterzentrale
    INTEGER NOT NULL REFERENCES Unterzentrale,
  Lage
    VARCHAR(99) NOT NULL
)

CREATE TABLE Existenz_Unterzentrale (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Existenz_Unterzentrale VALUES (0,'Anlage ohne Unterzentrale')
INSERT INTO Existenz_Unterzentrale VALUES (1,'Anlage mit einer Unterzentrale')
INSERT INTO Existenz_Unterzentrale VALUES (2,'Anlage wird 2 oder mehr ' ||
    'Unterzentralen zugeordnet')

CREATE TABLE Verkehrsrechnerzentrale (
  Verkehrsrechnerzentrale_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Verkehrsrechnerzentrale_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Existenz
    INTEGER NOT NULL REFERENCES Exist_Verkehrsrechnerzentrale
)

CREATE TABLE Verkehrsrechnerzentrale__Lage (
  Verkehrsrechnerzentrale
    INTEGER NOT NULL REFERENCES Verkehrsrechnerzentrale,
  Lage
    VARCHAR(30) NOT NULL
)

CREATE TABLE Exist_Verkehrsrechnerzentrale (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Langtext
  VARCHAR(255) NOT NULL
)

INSERT INTO Exist_Verkehrsrechnerzentrale VALUES (0,'Anlage ohne Anschluß ' ||
'an eine Verkehrsrechnerzentrale')
INSERT INTO Exist_Verkehrsrechnerzentrale VALUES (1,'Anlage mit Anschluß ' ||
'an eine Verkehrsrechnerzentrale')
INSERT INTO Exist_Verkehrsrechnerzentrale VALUES (2,'Anlage mit Anschluß ' ||
'an 2 oder mehr Verkehrsrechnerzentralen')

CREATE TABLE Ueberwachung_der_Anlage (
  Ueberwachung_der_Anlage_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Ueberwachung_der_Anlage_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Art_der_Ueberwachung
    VARCHAR(99) NOT NULL
)

CREATE TABLE Anzeigesystem (
  Anzeigesystem_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Anzsystem__hat_Gueltigkbereich (
  Anzeigesystem
    INTEGER NOT NULL REFERENCES Anzeigesystem,
  Gueltigkeitsbereich
    INTEGER NOT NULL REFERENCES
Automatische_Dauerzaehlstelle.Gueltigkeitsbereich
)

CREATE TABLE Anzsyst__hat_Wechselverkehrszeichen (
  Anzeigesystem
    INTEGER NOT NULL REFERENCES Anzeigesystem,
  Wechselverkehrszeichen
    INTEGER NOT NULL REFERENCES Wechselverkehrszeichen
)

CREATE TABLE Anzsyst__hat_Aufstellvorrichtg (
  Anzeigesystem
    INTEGER NOT NULL REFERENCES Anzeigesystem,
  Aufstellvorrichtung
    INTEGER NOT NULL REFERENCES Aufstellvorrichtung
)

CREATE TABLE Anzsyst__hat_Anz_d_Anzquerschn (
  Anzeigesystem
    INTEGER NOT NULL REFERENCES Anzeigesystem,
  Anzahl_der_Anzeigequerschnitte
    INTEGER NOT NULL REFERENCES Anzahl_der_Anzeigequerschnitte
)

CREATE TABLE Wechselverkehrszeichen (
  Wechselverkehrszeichen_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Wechselverkehrszeichen_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Anzahl
    INTEGER NOT NULL, -- Anzahl
CHECK ( Anzahl >= 0 )
)

CREATE TABLE Wechselverkz__Bauart_d_Zeich (
    Wechselverkehrszeichen
        INTEGER NOT NULL REFERENCES Wechselverkehrszeichen,
    Bauart_des_Zeichens
        INTEGER NOT NULL REFERENCES Bauart_des_Zeichens
)

CREATE TABLE Bauart_des_Zeichens (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Bauart_des_Zeichens VALUES (1, 'Faseroptik')
INSERT INTO Bauart_des_Zeichens VALUES (2, 'Prismen')
INSERT INTO Bauart_des_Zeichens VALUES (3, 'Band / Rollo')
INSERT INTO Bauart_des_Zeichens VALUES (4, 'Klapptafel')
INSERT INTO Bauart_des_Zeichens VALUES (5, 'mit Warnblinker')
INSERT INTO Bauart_des_Zeichens VALUES (6, 'Fließtext (frei ' ||
'programmierbare WVZ')
INSERT INTO Bauart_des_Zeichens VALUES (7, 'sonst')

CREATE TABLE Aufstellvorrichtung (
    Aufstellvorrichtung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Aufstellvorrichtung_ID) REFERENCES Strassennetz.Punktobjekt_hist,
    Vorrichtung
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean
)

CREATE TABLE Aufstellvorr__hat_Art_d_Aufst (
    Aufstellvorrichtung
        INTEGER NOT NULL REFERENCES Aufstellvorrichtung,
    Art_der_Aufstellung
        INTEGER NOT NULL REFERENCES Art_der_Aufstellung
)

CREATE TABLE Art_der_Aufstellung (
    Art_der_Aufstellung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Art_der_Aufstellung_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_bis LIKE '___.__.____' ),
    Art
        INTEGER NOT NULL REFERENCES Tab_Art_der_Aufstellung
)

CREATE TABLE Tab_Art_der_Aufstellung (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Tab_Art_der_Aufstellung VALUES (0, 'keine Angabe')
INSERT INTO Tab_Art_der_Aufstellung VALUES (1, 'an VZB')
INSERT INTO Tab_Art_der_Aufstellung VALUES (2, 'in Seitenaufstellung')
INSERT INTO Tab_Art_der_Aufstellung VALUES (3, 'an vorh. Bauwerk')
INSERT INTO Tab_Art_der_Aufstellung VALUES (4, 'in der Rampe')
INSERT INTO Tab_Art_der_Aufstellung VALUES (5, 'im nachgeordneten Netz')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Tab_Art_der_Aufstellung VALUES (6,'an VZB und in ' ||
'Seitenaufstellung')
INSERT INTO Tab_Art_der_Aufstellung VALUES (7,'an VZB und in der Rampe')
INSERT INTO Tab_Art_der_Aufstellung VALUES (8,'sonstige Kombination')

CREATE TABLE Anzahl_der_Anzeigequerschnitte (
  Anzahl_der_Anzeigequerschnitte_ID
  INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Anzahl_der_Anzeigequerschnitte_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
  CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
  CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Anzahl
  INTEGER NOT NULL, -- Anzahl_zweistellig
  CHECK ( ( Anzahl >= 0 ) AND ( Anzahl <= 99 ) ),
  CHECK ( Anzahl>=1 )
)

CREATE TABLE Datenerfassungseinrichtungen (
  Datenerfassungseinrichtungen_ID
  INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Datenerfeindr__hat_Gueltigkber (
  Datenerfassungseinrichtungen
  INTEGER NOT NULL REFERENCES Datenerfassungseinrichtungen,
  Gueltigkeitsbereich
  INTEGER NOT NULL REFERENCES
Automatische_Dauerzaehlstelle.Gueltigkeitsbereich
)

CREATE TABLE Datenerfeindr__hat_Detektmeth (
  Datenerfassungseinrichtungen
  INTEGER NOT NULL REFERENCES Datenerfassungseinrichtungen,
  Detektionsmethode
  INTEGER NOT NULL REFERENCES Detektionsmethode
)

CREATE TABLE Datenerfeindr__hat_Anord_d_Msys (
  Datenerfassungseinrichtungen
  INTEGER NOT NULL REFERENCES Datenerfassungseinrichtungen,
  Anordnung_des_Messsystems
  INTEGER NOT NULL REFERENCES Anordnung_des_Messsystems
)

CREATE TABLE Datenerfeindr__hat_Art_d_erf_Dn (
  Datenerfassungseinrichtungen
  INTEGER NOT NULL REFERENCES Datenerfassungseinrichtungen,
  Art_der_erfassten_Daten
  INTEGER NOT NULL REFERENCES Art_der_erfassten_Daten
)

CREATE TABLE Datenerfeindr__hat_Anz_d_Messqu (
  Datenerfassungseinrichtungen
  INTEGER NOT NULL REFERENCES Datenerfassungseinrichtungen,
  Anzahl_der_Messquerschnitte
  INTEGER NOT NULL REFERENCES Anzahl_der_Messquerschnitte
)

CREATE TABLE Detektionsmethode (
  Detektionsmethode_ID
  INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Detektionsmethode_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
  CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    gueltig_bis
      CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' )
  )

CREATE TABLE Detektionsmethode__Methode_VDE (
  Detektionsmethode
    INTEGER NOT NULL REFERENCES Detektionsmethode,
  Methode_VDE
    INTEGER NOT NULL REFERENCES Methode_VDE
)

CREATE TABLE Detektionsmethode__Methode_UDE (
  Detektionsmethode
    INTEGER NOT NULL REFERENCES Detektionsmethode,
  Methode_UDE
    INTEGER NOT NULL REFERENCES Methode_UDE
)

CREATE TABLE Methode_VDE (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Methode_VDE VALUES (1,'Doppelschleife')
INSERT INTO Methode_VDE VALUES (2,'Einfachschleife')
INSERT INTO Methode_VDE VALUES (3,'Radardetektor')
INSERT INTO Methode_VDE VALUES (4,'Infrarotdetektor')
INSERT INTO Methode_VDE VALUES (5,'Ultraschalldetektor')
INSERT INTO Methode_VDE VALUES (6,'Videokamera')
INSERT INTO Methode_VDE VALUES (7,'Laser')
INSERT INTO Methode_VDE VALUES (8,'sonst.')
```

```
CREATE TABLE Methode_UDE (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Methode_UDE VALUES (0,'keine Umfelddatenerfassung')
INSERT INTO Methode_UDE VALUES (1,'Sichtdetektor')
INSERT INTO Methode_UDE VALUES (2,'Temperaturmesser')
INSERT INTO Methode_UDE VALUES (3,'Helligkeitsmesser')
INSERT INTO Methode_UDE VALUES (4,'Niederschlagsdetektor')
INSERT INTO Methode_UDE VALUES (5,'Windmesser')
INSERT INTO Methode_UDE VALUES (6,'sonst.')
```

```
CREATE TABLE Anordnung_des_Messsystems (
  Anordnung_des_Messsystems_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Anordnung_des_Messsystems_ID) REFERENCES
Strassennetz.Punktobjekt_stat
)

CREATE TABLE Anordn_d_Msys__hat_Art_d_Anord (
  Anordnung_des_Messsystems
    INTEGER NOT NULL REFERENCES Anordnung_des_Messsystems,
  Art_der_Anordnung
    INTEGER NOT NULL REFERENCES Art_der_Anordnung
)

CREATE TABLE Art_der_Anordnung (
  Art_der_Anordnung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Art_der_Anordnung_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CHECK ( gueltig_von LIKE '___.__.____' ),
gueltig_bis
    CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_bis LIKE '___.__.____' ),
Art
    INTEGER NOT NULL REFERENCES Tab_Art_der_Anordnung
)

CREATE TABLE Tab_Art_der_Anordnung (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Tab_Art_der_Anordnung VALUES (0,'keine Angabe')
INSERT INTO Tab_Art_der_Anordnung VALUES (1,'in der Fahrbahn')
INSERT INTO Tab_Art_der_Anordnung VALUES (2,'an einer VZB')
INSERT INTO Tab_Art_der_Anordnung VALUES (3,'an bestehendem Bauwerk')
INSERT INTO Tab_Art_der_Anordnung VALUES (4,'am Straßenrand')
INSERT INTO Tab_Art_der_Anordnung VALUES (5,'Kombination von 1 und 2')
INSERT INTO Tab_Art_der_Anordnung VALUES (6,'Kombination von 1, 2 und 4')
INSERT INTO Tab_Art_der_Anordnung VALUES (7,'Kombination von 1 und 4')
INSERT INTO Tab_Art_der_Anordnung VALUES (8,'sonstige Kombination')

CREATE TABLE Art_der_erfassten_Daten (
    Art_der_erfassten_Daten_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Art_der_erfassten_Daten_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Verkehrsdaten
        VARCHAR(99) NOT NULL,
    Umfelddaten
        VARCHAR(99) NOT NULL
)

CREATE TABLE Anzahl_der_Messquerschnitte (
    Anzahl_der_Messquerschnitte_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Anzahl_der_Messquerschnitte_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Anzahl_d_Messquerschnitte_VDE
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl_d_Messquerschnitte_VDE>=0 ),
    Anzahl_d_Messquerschnitte_UDE
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl_d_Messquerschnitte_UDE>=0 )
)
```

### 13.2 Teilschema „Statische wegweisende / verkehrsregelnde Beschilderung“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Statische Beschilderung“.

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE SCHEMA Statische_Beschilderung
```

```
CREATE TABLE stat_wegweisende_Beschilderung (  
    stat_wegweisende_Beschilderung_ID  
        INTEGER NOT NULL PRIMARY KEY,  
    FOREIGN KEY (stat_wegweisende_Beschilderung_ID) REFERENCES  
Strassennetz.Punktobjekt_stat,  
    Fahrtrichtung_WB  
        VARCHAR(30) NOT NULL,  
    hat_Standort_Wegweiser  
        INTEGER NOT NULL REFERENCES Standort_Wegweiser,  
    hat_Einzugsbereich  
        INTEGER NOT NULL REFERENCES Einzugsbereich  
)
```

```
CREATE TABLE stat_wegw_Besch__hat_Wegwbild (  
    stat_wegweisende_Beschilderung  
        INTEGER NOT NULL REFERENCES stat_wegweisende_Beschilderung,  
    Wegweiserbild  
        INTEGER NOT NULL REFERENCES Wegweiserbild  
)
```

```
CREATE TABLE stat_wegw_Besch__hat_Wegwtafel (  
    stat_wegweisende_Beschilderung  
        INTEGER NOT NULL REFERENCES stat_wegweisende_Beschilderung,  
    Wegweisertafel_mit_Epunkten  
        INTEGER NOT NULL REFERENCES Wegweisertafel_mit_Epunkten  
)
```

```
CREATE TABLE Wegweiserbild (  
    Wegweiserbild_ID  
        INTEGER NOT NULL PRIMARY KEY,  
    FOREIGN KEY (Wegweiserbild_ID) REFERENCES Historisierung.historisches_Objekt,  
    gueltig_von  
        CHAR(10) NOT NULL, -- Datum  
    CHECK ( gueltig_von LIKE '___.__.____' ),  
    gueltig_bis  
        CHAR(10) NOT NULL, -- Datum  
    CHECK ( gueltig_bis LIKE '___.__.____' )  
)
```

```
CREATE TABLE Einzugsbereich (  
    Einzugsbereich_ID  
        INTEGER NOT NULL PRIMARY KEY,  
    Typ  
        INTEGER NOT NULL REFERENCES Typ_des_Einzugsbereiches,  
    Art  
        INTEGER NOT NULL REFERENCES Art_des_Einzugsbereiches,  
    Laenge  
        REAL NOT NULL, -- Meter  
    CHECK ( Laenge >= 0 )  
)
```

```
CREATE TABLE Typ_des_Einzugsbereiches (  
    Kennung  
        INTEGER NOT NULL PRIMARY KEY,  
    Langtext  
        VARCHAR(255) NOT NULL  
)
```

```
INSERT INTO Typ_des_Einzugsbereiches VALUES (1, 'BAB')  
INSERT INTO Typ_des_Einzugsbereiches VALUES (2, 'Bundesstraße')  
INSERT INTO Typ_des_Einzugsbereiches VALUES (3, 'sonst. Straße')
```

```
CREATE TABLE Art_des_Einzugsbereiches (  
    Kennung  
        INTEGER NOT NULL PRIMARY KEY,  
    Langtext  
        VARCHAR(255) NOT NULL  
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Art_des_Einzugsbereiches VALUES (1,'freie Strecke')
INSERT INTO Art_des_Einzugsbereiches VALUES (2,'Rampe')
INSERT INTO Art_des_Einzugsbereiches VALUES (3,'Knotenpunkt')

CREATE TABLE Standort_Wegweiser (
  Standort_Wegweiser_ID
    INTEGER NOT NULL PRIMARY KEY,
  Aufstellung
    INTEGER NOT NULL REFERENCES Aufstellung,
  richtungsbezogene_Ausfuehrung
    INTEGER NOT NULL REFERENCES richtungsbezogene_Ausfuehrung,
  Funktion
    INTEGER NOT NULL REFERENCES Funktion_wegweisend,
  StVO_Nummer
    INTEGER NOT NULL,
  Wegweiserfarbe
    INTEGER NOT NULL REFERENCES Wegweiserfarbe,
  hat_Numerierung
    INTEGER NOT NULL REFERENCES Numerierung,
  hat_Position
    INTEGER REFERENCES Position,
  hat_Netzzuordnung
    INTEGER NOT NULL REFERENCES Netzzuordnung
)

CREATE TABLE Standort_Wegw__techn_Ausfuehrg (
  Standort_Wegweiser
    INTEGER NOT NULL REFERENCES Standort_Wegweiser,
  technische_Ausfuehrung
    VARCHAR(99) NOT NULL
)

CREATE TABLE Numerierung (
  Numerierung_ID
    INTEGER NOT NULL PRIMARY KEY,
  Plannummer
    INTEGER NOT NULL, -- TK25_Blattnummer
  CHECK ( ( Plannummer >= 1 ) AND ( Plannummer <= 9999 ) ),
  Standort_Nummer
    INTEGER
)

CREATE TABLE Position (
  Position_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Position_ID) REFERENCES Strassennetz.Punktobjekt_stat
)

CREATE TABLE Netzzuordnung (
  Netzzuordnung_ID
    INTEGER NOT NULL PRIMARY KEY,
  erster_Entscheidungspunkt
    INTEGER NOT NULL,
  Von_Knoten
    INTEGER NOT NULL
)

CREATE TABLE Aufstellung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(30) NOT NULL
)

INSERT INTO Aufstellung VALUES (1,'Schilderbrücke')
INSERT INTO Aufstellung VALUES (2,'Ausleger rechts')
INSERT INTO Aufstellung VALUES (3,'Ausleger links')
INSERT INTO Aufstellung VALUES (4,'Ausleger seitlich rechts')
INSERT INTO Aufstellung VALUES (5,'Ausleger seitlich links')
INSERT INTO Aufstellung VALUES (6,'Ausleger seitlich hoch')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE richtungsbezogene_Ausfuehrung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO richtungsbezogene_Ausfuehrung VALUES (1,'richtungsbezogen')
INSERT INTO richtungsbezogene_Ausfuehrung VALUES (2,'spur- bzw. ' ||
'fahrstreifenbezogen')
```

```
CREATE TABLE Funktion_wegweisend (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(30) NOT NULL
)

INSERT INTO Funktion_wegweisend VALUES (1,'Vorwegweiser')
INSERT INTO Funktion_wegweisend VALUES (2,'Ankündigung')
INSERT INTO Funktion_wegweisend VALUES (3,'Wegweiser')
INSERT INTO Funktion_wegweisend VALUES (4,'Entfernungstafel')
INSERT INTO Funktion_wegweisend VALUES (5,'Hinweisschild')
INSERT INTO Funktion_wegweisend VALUES (6,'Planskizze')
INSERT INTO Funktion_wegweisend VALUES (7,'sonst. Funktion')
```

```
CREATE TABLE Wegweiserfarbe (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(5) NOT NULL
)

INSERT INTO Wegweiserfarbe VALUES (1,'blau')
INSERT INTO Wegweiserfarbe VALUES (2,'gelb')
INSERT INTO Wegweiserfarbe VALUES (3,'weiß')
INSERT INTO Wegweiserfarbe VALUES (4,'braun')
INSERT INTO Wegweiserfarbe VALUES (5,'grün')
INSERT INTO Wegweiserfarbe VALUES (6,'sonst')
```

```
CREATE TABLE Wegweisertafel_mit_Epunkten (
  Wegweisertafel_mit_Epunkten_ID
    INTEGER NOT NULL PRIMARY KEY,
  Richtung
    INTEGER NOT NULL REFERENCES Richtung,
  hat_NK_von
    INTEGER NOT NULL,
  hat_NK_nach
    INTEGER NOT NULL
)

CREATE TABLE Wegwtafel__hat_Beschrelem_Symb (
  Wegweisertafel_mit_Epunkten
    INTEGER NOT NULL REFERENCES Wegweisertafel_mit_Epunkten,
  Beschriftungselemente_Symbole
    INTEGER NOT NULL REFERENCES Beschriftungselemente_Symbole
)

CREATE TABLE Richtung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(10) NOT NULL
)

INSERT INTO Richtung VALUES (1,'gerade')
INSERT INTO Richtung VALUES (2,'rechts')
INSERT INTO Richtung VALUES (3,'links')
INSERT INTO Richtung VALUES (4,'halb rechts')
INSERT INTO Richtung VALUES (5,'halb links')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Richtung VALUES (6,'U-Turn')
INSERT INTO Richtung VALUES (7,'sonst.')

CREATE TABLE Beschriftungselemente_Symbole (
  Beschriftungselemente_Symbole_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Beschriftungselemente_Symbole_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Elementtyp
    INTEGER NOT NULL REFERENCES Elementtyp,
  Text
    VARCHAR(20) NOT NULL,
  Farbe
    INTEGER NOT NULL REFERENCES Farbe_Beschriftungselem_Symb,
  Anordnung
    INTEGER NOT NULL REFERENCES Anordnung,
  Angaben_zum_Zielgebiet
    VARCHAR(20) NOT NULL,
  Symbolik
    VARCHAR(30)
)

CREATE TABLE Elementtyp (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(30) NOT NULL
)

INSERT INTO Elementtyp VALUES (1,'Text')
INSERT INTO Elementtyp VALUES (2,'Piktogramm')
INSERT INTO Elementtyp VALUES (3,'Entfernungsangabe')
INSERT INTO Elementtyp VALUES (4,'Anschlußstellenummer')
INSERT INTO Elementtyp VALUES (5,'StVO-Zeichen')
INSERT INTO Elementtyp VALUES (6,'Straßenummer')

CREATE TABLE Farbe_Beschriftungselem_Symb (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(10) NOT NULL
)

INSERT INTO Farbe_Beschriftungselem_Symb VALUES (1,'blau')
INSERT INTO Farbe_Beschriftungselem_Symb VALUES (2,'gelb')
INSERT INTO Farbe_Beschriftungselem_Symb VALUES (3,'weiß')
INSERT INTO Farbe_Beschriftungselem_Symb VALUES (4,'grün')
INSERT INTO Farbe_Beschriftungselem_Symb VALUES (5,'sonst')

CREATE TABLE Anordnung (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(20) NOT NULL
)

INSERT INTO Anordnung VALUES (1,'einzeilig')
INSERT INTO Anordnung VALUES (2,'mehrzeilig (zusammengehörig)')
INSERT INTO Anordnung VALUES (3,'mehrzeilig (nicht zusammengehörig)')

CREATE TABLE stat_verkehrsreg_Beschilderung (
  stat_verkehrsreg_Beschilderung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (stat_verkehrsreg_Beschilderung_ID) REFERENCES
Strassennetz.Punktobjekt_stat,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
Fahrtrichtung_WB
  VARCHAR(30),
hat_Standort_Beschilderung
  INTEGER NOT NULL REFERENCES Standort_Beschilderung
)

CREATE TABLE stat_verkreg_Besch__hat_Gueltb (
  stat_verkehrsreg_Beschilderung
    INTEGER NOT NULL REFERENCES stat_verkehrsreg_Beschilderung,
  Gueltigkeitsbereich_VRB
    INTEGER NOT NULL REFERENCES Gueltigkeitsbereich_VRB
)

CREATE TABLE Standort_Beschilderung (
  Standort_Beschilderung_ID
    INTEGER NOT NULL PRIMARY KEY,
  Aufstellart
    INTEGER NOT NULL REFERENCES Aufstellart,
  richtungsbezogene_Ausfuehrung
    INTEGER NOT NULL REFERENCES richtungsbezogene_Ausfuehrung,
  hat_Position
    INTEGER REFERENCES Position,
  hat_StVO_Zuordnung
    INTEGER NOT NULL REFERENCES StVO_Zuordnung
)

CREATE TABLE Standort_Besch__techn_Ausfuehr (
  Standort_Beschilderung
    INTEGER NOT NULL REFERENCES Standort_Beschilderung,
  technische_Ausfuehrung
    VARCHAR(99) NOT NULL
)

CREATE TABLE Standort_Besch__hat_Numerierg (
  Standort_Beschilderung
    INTEGER NOT NULL REFERENCES Standort_Beschilderung,
  Numerierung
    INTEGER NOT NULL REFERENCES Numerierung
)

CREATE TABLE Standort_Besch__hat_verkreg_F (
  Standort_Beschilderung
    INTEGER NOT NULL REFERENCES Standort_Beschilderung,
  verkehrsregelnde_Funktion
    INTEGER NOT NULL REFERENCES verkehrsregelnde_Funktion
)

CREATE TABLE Aufstellart (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(20) NOT NULL
)

INSERT INTO Aufstellart VALUES (1,'Gabelständer')
INSERT INTO Aufstellart VALUES (2,'Ausleger rechts')
INSERT INTO Aufstellart VALUES (3,'Ausleger links')
INSERT INTO Aufstellart VALUES (4,'Ausleger seitlich rechts')
INSERT INTO Aufstellart VALUES (5,'Ausleger seitlich links')
INSERT INTO Aufstellart VALUES (6,'Ausleger seitlich hoch')
INSERT INTO Aufstellart VALUES (7,'Schilderbrücke')
INSERT INTO Aufstellart VALUES (8,'Klappschild')

CREATE TABLE verkehrsregelnde_Funktion (
  verkehrsregelnde_Funktion_ID
    INTEGER NOT NULL PRIMARY KEY,
  Funktion
    INTEGER NOT NULL REFERENCES Funktion_verkehrsregelnd,
  Paragraph
    INTEGER
)

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Funktion_verkehrsregelnd (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(30) NOT NULL
)

INSERT INTO Funktion_verkehrsregelnd VALUES (1,'Gefahrzeichen')
INSERT INTO Funktion_verkehrsregelnd VALUES (2,'Vorschriftszeichen')
INSERT INTO Funktion_verkehrsregelnd VALUES (3,'Richtzeichen')
INSERT INTO Funktion_verkehrsregelnd VALUES (4,'Zusatzzeichen')
INSERT INTO Funktion_verkehrsregelnd VALUES (5,'Kombinationen aus mehreren ' ||
'Zeichen')

CREATE TABLE StVO_Zuordnung (
  StVO_Zuordnung_ID
    INTEGER NOT NULL PRIMARY KEY,
  StVO_Zeichen
    INTEGER NOT NULL,
  Zusatzzeichen_1
    INTEGER,
  Zusatzzeichen_2
    INTEGER
)

CREATE TABLE Gueltigkeitsbereich_VRB (
  Gueltigkeitsbereich_VRB_ID
    INTEGER NOT NULL PRIMARY KEY,
  verkehrssituationsabh_GB
    VARCHAR(50),
  CHECK ( ( verkehrsartabhaengiger_GB IS NOT NULL ) OR ( verkehrssituationsabh_GB
IS NOT NULL ) OR ( ist_raeumlicher_GB IS NOT NULL ) OR ( ist_zeitlicher_GB IS NOT
NULL ) )
)

CREATE TABLE Gueltkber_VRB__verkartabh_GB (
  Gueltigkeitsbereich_VRB
    INTEGER NOT NULL REFERENCES Gueltigkeitsbereich_VRB,
  verkehrsartabhaengiger_GB
    VARCHAR(50) NOT NULL
)

CREATE TABLE Gueltkber_VRB__ist_raueml_GB (
  Gueltigkeitsbereich_VRB
    INTEGER NOT NULL REFERENCES Gueltigkeitsbereich_VRB,
  raeumlicher_Gueltigkeitsber
    INTEGER NOT NULL REFERENCES raeumlicher_Gueltigkeitsber
)

CREATE TABLE Gueltkber_VRB__ist_zeitl_GB (
  Gueltigkeitsbereich_VRB
    INTEGER NOT NULL REFERENCES Gueltigkeitsbereich_VRB,
  zeitlicher_Gueltigkeitsber
    INTEGER NOT NULL REFERENCES zeitlicher_Gueltigkeitsber
)

CREATE TABLE raeumlicher_Gueltigkeitsber (
  raeumlicher_Gueltigkeitsber_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (raeumlicher_Gueltigkeitsber_ID) REFERENCES
Strassennetz.Streckenobjekt_hist
)

CREATE TABLE zeitlicher_Gueltigkeitsber (
  zeitlicher_Gueltigkeitsber_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (zeitlicher_Gueltigkeitsber_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    zeitlicher_Bereich
        VARCHAR(30) NOT NULL
)
)
```

### 13.3 Teilschema „Lichtsignalanlagen“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Lichtsignalanlage“.

```
CREATE SCHEMA Lichtsignalanlage
```

```
CREATE TABLE Lichtsignalanlage (
    Lichtsignalanlage_ID
        INTEGER NOT NULL PRIMARY KEY,
    hat_oertliche_Zuordnung
        INTEGER NOT NULL REFERENCES Allgemeine_Objekte.oertliche_Zuordnung
)
)
```

```
CREATE TABLE Lichtsiganl__hat_Rotlueberw (
    Lichtsignalanlage
        INTEGER NOT NULL REFERENCES Lichtsignalanlage,
    Rotlichtueberwachung
        INTEGER NOT NULL REFERENCES Rotlichtueberwachung
)
)
```

```
CREATE TABLE Lichtsiganl__hat_Ang_z_Knotpkt (
    Lichtsignalanlage
        INTEGER NOT NULL REFERENCES Lichtsignalanlage,
    Angaben_zum_Knotenpunkt_LSA
        INTEGER NOT NULL REFERENCES Angaben_zum_Knotenpunkt_LSA
)
)
```

```
CREATE TABLE Lsiganl__hat_Grundl_u_Inbn_Dat (
    Lichtsignalanlage
        INTEGER NOT NULL REFERENCES Lichtsignalanlage,
    Grundlage_und_Inbetriebn_daten
        INTEGER NOT NULL REFERENCES Grundlage_und_Inbetriebn_daten
)
)
```

```
CREATE TABLE Lichtsiganl__hat_Ang_z_Verkdat (
    Lichtsignalanlage
        INTEGER NOT NULL REFERENCES Lichtsignalanlage,
    Angaben_zu_Verkehrsdaten
        INTEGER NOT NULL REFERENCES Angaben_zu_Verkehrsdaten
)
)
```

```
CREATE TABLE Lichtsiganl__hat_Softwarekomp (
    Lichtsignalanlage
        INTEGER NOT NULL REFERENCES Lichtsignalanlage,
    Softwarekomponenten
        INTEGER NOT NULL REFERENCES Softwarekomponenten
)
)
```

```
CREATE TABLE Lichtsiganl__hat_Hardwarekomp (
    Lichtsignalanlage
        INTEGER NOT NULL REFERENCES Lichtsignalanlage,
    Hardwarekomponenten
        INTEGER NOT NULL REFERENCES Hardwarekomponenten
)
)
```

```
CREATE TABLE Lichtsiganl__hat_ueberg_Zentr (
    Lichtsignalanlage
        INTEGER NOT NULL REFERENCES Lichtsignalanlage,
)
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    uebergeordnete_Zentrale
    INTEGER NOT NULL REFERENCES uebergeordnete_Zentrale
)

CREATE TABLE Rotlichtueberwachung (
    Rotlichtueberwachung_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Rotlichtueberwachung_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
    CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
    CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Ueberwachung
    INTEGER NOT NULL REFERENCES Logical_Types.Boolean,
    Fahrtrichtung
    VARCHAR(50)
)

CREATE TABLE Angaben_zum_Knotenpunkt_LSA (
    Angaben_zum_Knotenpunkt_LSA_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Angaben_zum_Knotenpunkt_LSA_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
    CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
    CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Name
    VARCHAR(50) NOT NULL,
    Zahl_der_Knotenarme
    INTEGER, -- Anzahl
    CHECK ( Zahl_der_Knotenarme >= 0 ),
    Hauptrichtung
    VARCHAR(50) NOT NULL
)

CREATE TABLE Ang_z_Knotpkt_LSA_Markpfeile (
    Angaben_zum_Knotenpunkt_LSA
    INTEGER NOT NULL REFERENCES Angaben_zum_Knotenpunkt_LSA,
    Markierungspfeile
    INTEGER NOT NULL REFERENCES Markierungspfeile
)

CREATE TABLE Markierungspfeile (
    Kennung
    INTEGER NOT NULL PRIMARY KEY,
    Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Markierungspfeile VALUES (1,'Rechtabbiegepfeil')
INSERT INTO Markierungspfeile VALUES (2,'Linksabbiegepfeil')
INSERT INTO Markierungspfeile VALUES (3,'Geradeauspfeil')
INSERT INTO Markierungspfeile VALUES (4,'Geradeaus und Links')
INSERT INTO Markierungspfeile VALUES (5,'Geradeaus und Rechts')

CREATE TABLE Grundlage_und_Inbetriebn_daten (
    Grundlage_und_Inbetriebn_daten_ID
    INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Grundlage_und_Inbetriebn_daten_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
    CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
    CHAR(10) NOT NULL, -- Datum
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CHECK ( gueltig_bis LIKE '___.__.____' ),
Einsatzzeiten
    INTEGER NOT NULL REFERENCES Einsatzzeiten,
Nummer
    INTEGER NOT NULL,
Baulasttraeger
    VARCHAR(50) NOT NULL,
Stoerungsstelle
    VARCHAR(50),
Datum_Installation
    CHAR(10) NOT NULL, -- Datum
CHECK ( Datum_Installation LIKE '___.__.____' ),
Wartungszyklus
    VARCHAR(99) NOT NULL,
Datum_Zwischenzeitmatrix
    CHAR(10), -- Datum
CHECK ( Datum_Zwischenzeitmatrix LIKE '___.__.____' ),
Datum_VZ_Plan
    CHAR(10), -- Datum
CHECK ( Datum_VZ_Plan LIKE '___.__.____' ),
Datum_Signalprogramme
    CHAR(10), -- Datum
CHECK ( Datum_Signalprogramme LIKE '___.__.____' )
)

CREATE TABLE Grndl_u_Inbn_dat__Nr_Nachb_LSA (
    Grundlage_und_Inbetriebn_daten
        INTEGER NOT NULL REFERENCES Grundlage_und_Inbetriebn_daten,
    Nummer_Nachbar_LSA
        INTEGER NOT NULL
)

CREATE TABLE Einsatzzeiten (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Einsatzzeiten VALUES (1,'24-Stunden-Anzeige')
INSERT INTO Einsatzzeiten VALUES (2,'Nachtabstaltung')

CREATE TABLE Angaben_zu_Verkehrsdaten (
    Angaben_zu_Verkehrsdaten_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Angaben_zu_Verkehrsdaten_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_bis LIKE '___.__.____' ),
    Datum_VZ
        CHAR(10) NOT NULL, -- Datum
CHECK ( Datum_VZ LIKE '___.__.____' ),
    OePNV_Frequenzen
        VARCHAR(99),
    Fussgaengerzaehlung
        VARCHAR(99)
)

CREATE TABLE Ang_z_Verkdat__Verkehrsstroeme (
    Angaben_zu_Verkehrsdaten
        INTEGER NOT NULL REFERENCES Angaben_zu_Verkehrsdaten,
    Verkehrsstroeme
        VARCHAR(99) NOT NULL
)

CREATE TABLE Softwarekomponenten (
    Softwarekomponenten_ID
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        INTEGER NOT NULL PRIMARY KEY,
Hersteller
        VARCHAR(50)
)

CREATE TABLE Softwkomp__hat_Signprogrparam (
    Softwarekomponenten
        INTEGER NOT NULL REFERENCES Softwarekomponenten,
    Signalprogrammparameter
        INTEGER NOT NULL REFERENCES Signalprogrammparameter
)

CREATE TABLE Softwkomp__hat_hinterl_Progr (
    Softwarekomponenten
        INTEGER NOT NULL REFERENCES Softwarekomponenten,
    hinterlegte_Programme
        INTEGER NOT NULL REFERENCES hinterlegte_Programme
)

CREATE TABLE Signalprogrammparameter (
    Signalprogrammparameter_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Signalprogrammparameter_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Zwischenzeit
        VARCHAR(99) NOT NULL,
    maximale_Freigabezeit
        VARCHAR(99),
    Umlaufzeit
        VARCHAR(99) NOT NULL,
    maximale_Wartezeit
        VARCHAR(99),
    Zeitluecke_Belegung
        VARCHAR(99),
    Berechnungsverfahren
        VARCHAR(99),
    Mindestfreigabezeit
        VARCHAR(99),
    Versatzzeit
        VARCHAR(99),
    Radfahrerberuecksichtigung
        INTEGER REFERENCES Logical_Types.Boolean,
    OePNV_Beruecksichtigung
        INTEGER REFERENCES Logical_Types.Boolean
)

CREATE TABLE hinterlegte_Programme (
    hinterlegte_Programme_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (hinterlegte_Programme_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Steuerungsverfahren
        VARCHAR(50),
    Anzahl
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl >= 0 ),
    Auswahlverfahren
        INTEGER NOT NULL REFERENCES Auswahlverfahren,
    Art_der_Programme
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        VARCHAR(50),
OePNV_Berechtigung
        INTEGER NOT NULL REFERENCES OePNV_Berechtigung,
Darstellung
        VARCHAR(50)
)

CREATE TABLE Auswahlverfahren (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Auswahlverfahren VALUES (1,'zeitplanabhängiges Verfahren')
INSERT INTO Auswahlverfahren VALUES (2,'verkehrsabhängiges Verfahren')

CREATE TABLE OePNV_Berechtigung (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO OePNV_Berechtigung VALUES (0,'ohne')
INSERT INTO OePNV_Berechtigung VALUES (1,'bedingt')
INSERT INTO OePNV_Berechtigung VALUES (2,'weitgehend')

CREATE TABLE Hardwarekomponenten (
    Hardwarekomponenten_ID
        INTEGER NOT NULL PRIMARY KEY,
    Hersteller
        VARCHAR(50) NOT NULL,
    Signalgruppen
        VARCHAR(99)
)

CREATE TABLE Hardwkomp__hat_ak_o_takt_Signg (
    Hardwarekomponenten
        INTEGER NOT NULL REFERENCES Hardwarekomponenten,
    akust_oder_taktile_Signalgeber
        INTEGER NOT NULL REFERENCES akust_oder_taktile_Signalgeber
)

CREATE TABLE Hardwkomp__hat_Ang_z_Steuerger (
    Hardwarekomponenten
        INTEGER NOT NULL REFERENCES Hardwarekomponenten,
    Angaben_zum_Steuergeraet
        INTEGER NOT NULL REFERENCES Angaben_zum_Steuergeraet
)

CREATE TABLE Hardwkomp__hat_Ang_z_Verkabelg (
    Hardwarekomponenten
        INTEGER NOT NULL REFERENCES Hardwarekomponenten,
    Angaben_zur_Verkabelung
        INTEGER NOT NULL REFERENCES Angaben_zur_Verkabelung
)

CREATE TABLE Hardwkomp__hat_Ang_z_Aufstvorr (
    Hardwarekomponenten
        INTEGER NOT NULL REFERENCES Hardwarekomponenten,
    Angaben_z_Aufstellvorrichtung
        INTEGER NOT NULL REFERENCES Angaben_z_Aufstellvorrichtung
)

CREATE TABLE Hardwkomp__hat_opt_Signalgeber (
    Hardwarekomponenten
        INTEGER NOT NULL REFERENCES Hardwarekomponenten,
    optische_Signalgeber
        INTEGER NOT NULL REFERENCES optische_Signalgeber
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Hardwkomp__hat_Erfasseinr_IV (
    Hardwarekomponenten
        INTEGER NOT NULL REFERENCES Hardwarekomponenten,
    Erfassungseinrichtungen_IV
        INTEGER NOT NULL REFERENCES Erfassungseinrichtungen_IV
)

CREATE TABLE Hardwkomp__hat_Erfasseinr_OeV (
    Hardwarekomponenten
        INTEGER NOT NULL REFERENCES Hardwarekomponenten,
    Erfassungseinrichtungen_OeV
        INTEGER NOT NULL REFERENCES Erfassungseinrichtungen_OeV
)

CREATE TABLE akust_oder_taktile_Signalgeber (
    akust_oder_taktile_Signalgeber_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (akust_oder_taktile_Signalgeber_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Anzahl_1
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl_1 >= 0 ),
    Anzahl_2
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl_2 >= 0 ),
    Position_1
        INTEGER NOT NULL REFERENCES Position_SG,
    Position_2
        INTEGER NOT NULL REFERENCES Position_SG
)

CREATE TABLE Position_SG (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Position_SG VALUES (1,'seitlich')
INSERT INTO Position_SG VALUES (2,'überkopf')
INSERT INTO Position_SG VALUES (3,'beide Varianten')

CREATE TABLE Angaben_zum_Steuergeraet (
    Angaben_zum_Steuergeraet_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Angaben_zum_Steuergeraet_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Geraetebezeichnung
        VARCHAR(50) NOT NULL,
    max_Programmzahl
        INTEGER NOT NULL, -- Anzahl
    CHECK ( max_Programmzahl >= 0 ),
    max_SG_Anzahl_Kfz
        INTEGER NOT NULL, -- Anzahl
    CHECK ( max_SG_Anzahl_Kfz >= 0 ),
    max_SG_Anzahl_Fussgaenger
        INTEGER NOT NULL, -- Anzahl
    CHECK ( max_SG_Anzahl_Fussgaenger >= 0 ),
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
max_SG_Anzahl_OeV
    INTEGER NOT NULL, -- Anzahl
CHECK ( max_SG_Anzahl_OeV>=0 ),
max_SG_Anzahl_Zusatzsignalgeb
    INTEGER NOT NULL, -- Anzahl
CHECK ( max_SG_Anzahl_Zusatzsignalgeb>=0 ),
max_Anzahl_Kanaele
    INTEGER, -- Anzahl
CHECK ( max_Anzahl_Kanaele>=0 ),
Programmierbarkeit
    INTEGER NOT NULL REFERENCES Logical_Types.Boolean,
Spannungsversorgung
    VARCHAR(50),
Anordnung
    VARCHAR(50),
Verkehrszahlenspeicher
    VARCHAR(50),
Schliesssystem
    VARCHAR(50),
Handbedienteil
    INTEGER REFERENCES Logical_Types.Boolean,
BUeSTRA
    INTEGER NOT NULL REFERENCES Logical_Types.Boolean,
Koordinierung
    INTEGER REFERENCES Koordinierung,
Telefonbuchse
    INTEGER REFERENCES Logical_Types.Boolean,
Messschreiberanschluss
    INTEGER REFERENCES Logical_Types.Boolean,
Heizung_Luefter
    INTEGER REFERENCES Logical_Types.Boolean
)

CREATE TABLE Koordinierung (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Koordinierung VALUES (0,'Funkuhr')
INSERT INTO Koordinierung VALUES (1,'Koordinierungskabel')

CREATE TABLE Angaben_zur_Verkabelung (
    Angaben_zur_Verkabelung_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Angaben_zur_Verkabelung_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
CHECK ( gueltig_bis LIKE '___.__.____' ),
    Rohrbelegung
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean,
    Verbindung_NK
        INTEGER NOT NULL REFERENCES Verbindung_NK
)

CREATE TABLE Verbindung_NK (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Verbindung_NK VALUES (0,'keine Verbindung zum Nachbarknoten')
INSERT INTO Verbindung_NK VALUES (1,'Verbindung zum Nachbarknoten vorhanden')
INSERT INTO Verbindung_NK VALUES (2,'Verbindung zu mehreren Nachbarknoten ' ||
'vorhanden')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Angaben_z_Aufstellvorrichtung (
  Angaben_z_Aufstellvorrichtung_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Angaben_z_Aufstellvorrichtung_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Art
    VARCHAR(50) NOT NULL,
  Anzahl
    INTEGER NOT NULL, -- Anzahl
  CHECK ( Anzahl>=0 ),
  Ausleger_Laenge
    REAL, -- Meter
  CHECK ( Ausleger_Laenge>=0 ),
  Ausleger_Anzahl
    INTEGER, -- Anzahl
  CHECK ( Ausleger_Anzahl>=0 ),
  Gruendung
    VARCHAR(50)
)
```

```
CREATE TABLE optische_Signalgeber (
  optische_Signalgeber_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (optische_Signalgeber_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Anzahl
    INTEGER NOT NULL, -- Anzahl
  CHECK ( Anzahl>=0 ),
  Durchmesser
    VARCHAR(50),
  Blendensymbol
    VARCHAR(99) NOT NULL,
  Position
    INTEGER NOT NULL REFERENCES Position_SG
)
```

```
CREATE TABLE Erfassungseinrichtungen_IV (
  Erfassungseinrichtungen_IV_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Erfassungseinrichtungen_IV_ID) REFERENCES
Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Art_der_Erfassung
    VARCHAR(99) NOT NULL,
  Anzahl_1
    INTEGER NOT NULL, -- Anzahl
  CHECK ( Anzahl_1>=0 ),
  Anzahl_2
    INTEGER, -- Anzahl
  CHECK ( Anzahl_2>=0 ),
  zu_erkennende_Fz
    VARCHAR(50),
  Funktion

```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
        VARCHAR(99),
    Gestaltung
        VARCHAR(99)
)

CREATE TABLE Erfassungseinrichtungen_OeV (
    Erfassungseinrichtungen_OeV_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Erfassungseinrichtungen_OeV_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Art_der_Erfassung
        VARCHAR(50) NOT NULL,
    Anzahl
        INTEGER NOT NULL, -- Anzahl
    CHECK ( Anzahl >= 0 ),
    Position
        VARCHAR(50),
    Funktion
        VARCHAR(50) NOT NULL,
    Verbindung
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean
)

CREATE TABLE uebergeordnete_Zentrale (
    uebergeordnete_Zentrale_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (uebergeordnete_Zentrale_ID) REFERENCES
Historisierung.historisches_Objekt,
    gueltig_von
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_von LIKE '___.__.____' ),
    gueltig_bis
        CHAR(10) NOT NULL, -- Datum
    CHECK ( gueltig_bis LIKE '___.__.____' ),
    Lage
        VARCHAR(50) NOT NULL,
    Schnittstellendef_bekannt
        INTEGER NOT NULL REFERENCES Logical_Types.Boolean,
    Typ
        INTEGER NOT NULL REFERENCES Typ_UeZ,
    Eingriffsmoeglichkeit
        INTEGER NOT NULL REFERENCES Eingriffsmoeglichkeit,
    Hersteller_des_Steuerrechners
        VARCHAR(50),
    Typ_des_Steuerrechners
        VARCHAR(50)
)

CREATE TABLE Typ_UeZ (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
)

INSERT INTO Typ_UeZ VALUES (1, 'RBL')
INSERT INTO Typ_UeZ VALUES (2, 'Verkehrsrechner')
INSERT INTO Typ_UeZ VALUES (3, 'gemeinsame Leitzentrale IV/ÖV')
INSERT INTO Typ_UeZ VALUES (4, 'getrennte Leitzentralen IV/ÖV')

CREATE TABLE Eingriffsmoeglichkeit (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(255) NOT NULL
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

)

```
INSERT INTO Eingriffsmoeglichkeit VALUES (0,'nur Visualisierung')
INSERT INTO Eingriffsmoeglichkeit VALUES (1,'mit Steuereingriff')
```



## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Volumenobjekt_Modell (
  Volumenobjekt_Modell_ID
  INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Volumenobjekt_M__hat_Koerper (
  Volumenobjekt_Modell
  INTEGER NOT NULL REFERENCES Volumenobjekt_Modell,
  Koerper
  INTEGER NOT NULL REFERENCES Koerper
)

CREATE TABLE Volumenobjekt_M__hat_Volumen (
  Volumenobjekt_Modell
  INTEGER NOT NULL REFERENCES Volumenobjekt_Modell,
  Volumen
  INTEGER NOT NULL REFERENCES Volumen
)

CREATE TABLE Raumbezugsart (
  Raumbezugsart_ID
  INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE Knoten (
  Knoten_ID
  INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Knoten_ID) REFERENCES Raumbezugsart
)

CREATE TABLE Knoten__hat_Punkt (
  Knoten
  INTEGER NOT NULL REFERENCES Knoten,
  Punkt
  INTEGER NOT NULL REFERENCES Punkt
)

CREATE TABLE isolierter_Knoten (
  isolierter_Knoten_ID
  INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (isolierter_Knoten_ID) REFERENCES Knoten,
  liegt_in_Masche
  INTEGER REFERENCES Masche
)

CREATE TABLE nicht_isolierter_Knoten (
  nicht_isolierter_Knoten_ID
  INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (nicht_isolierter_Knoten_ID) REFERENCES Knoten
)

CREATE TABLE Kante (
  Kante_ID
  INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Kante_ID) REFERENCES Raumbezugsart,
  beginnt_bei_Knoten
  INTEGER REFERENCES nicht_isolierter_Knoten,
  endet_bei_Knoten
  INTEGER REFERENCES nicht_isolierter_Knoten
)

CREATE TABLE Kante__hat_Linie (
  Kante
  INTEGER NOT NULL REFERENCES Kante,
  Linie
  INTEGER NOT NULL REFERENCES Linie
)

CREATE TABLE Masche (
  Masche_ID
  INTEGER NOT NULL PRIMARY KEY,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    FOREIGN KEY (Masche_ID) REFERENCES Raumbezugsart
)

CREATE TABLE Masche__hat_Flaeche (
    Masche
        INTEGER NOT NULL REFERENCES Masche,
    Flaeche
        INTEGER NOT NULL REFERENCES Flaeche
)

CREATE TABLE Masche__hat_Kante (
    Masche
        INTEGER NOT NULL REFERENCES Masche,
    Kante
        INTEGER NOT NULL REFERENCES Kante
)

CREATE TABLE Koerper (
    Koerper_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Koerper_ID) REFERENCES Raumbezugsart
)

CREATE TABLE Koerper__hat_Volumen (
    Koerper
        INTEGER NOT NULL REFERENCES Koerper,
    Volumen
        INTEGER NOT NULL REFERENCES Volumen
)

CREATE TABLE Koerper__hat_Masche (
    Koerper
        INTEGER NOT NULL REFERENCES Koerper,
    Masche
        INTEGER NOT NULL REFERENCES Masche
)

CREATE TABLE Punkt (
    Punkt_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Punkt_ID) REFERENCES Raumbezugsart,
    hat_Lage
        INTEGER NOT NULL REFERENCES direct_positioning_schema.direct_position
)

CREATE TABLE Linie (
    Linie_ID
        INTEGER NOT NULL PRIMARY KEY,
    FOREIGN KEY (Linie_ID) REFERENCES Raumbezugsart
)

CREATE TABLE Linie__aus_Linienelementen (
    Linie
        INTEGER NOT NULL REFERENCES Linie,
    Linienelement_3D
        INTEGER NOT NULL REFERENCES Linienelement_3D,
    folgt_auf_Linienelement_3D
        INTEGER REFERENCES Linienelement_3D
)

CREATE TABLE Linienelement_3D (
    Linienelement_3D_ID
        INTEGER NOT NULL PRIMARY KEY,
    beginnt_bei_Punkt
        INTEGER NOT NULL REFERENCES Punkt,
    endet_bei_Punkt
        INTEGER NOT NULL REFERENCES Punkt,
    zu_Linie
        INTEGER NOT NULL REFERENCES Linie
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Linienelement_Spline (
  Linienelement_Spline_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Linienelement_Spline_ID) REFERENCES Linienelement_3D,
  Kruemmung_am_Anfang
    REAL,
  Kruemmung_am_Ende
    REAL,
  Tangente_am_Anfang
    REAL, -- Winkel
  Tangente_am_Ende
    REAL -- Winkel
)

CREATE TABLE Spline__hat_Stuetzpunkte (
  Linienelement_Spline
    INTEGER NOT NULL REFERENCES Linienelement_Spline,
  Punkt
    INTEGER NOT NULL REFERENCES Punkt,
  folgt_auf_Punkt
    INTEGER REFERENCES Punkt
)

CREATE TABLE gerades_Linienelement (
  gerades_Linienelement_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (gerades_Linienelement_ID) REFERENCES Linienelement_3D
)

CREATE TABLE Linie_Flaechenelement (
  Linie_Flaechenelement_ID
    INTEGER NOT NULL PRIMARY KEY,
  bei_Linienfunktion
    VARCHAR(255) NOT NULL,
  Linie
    INTEGER NOT NULL REFERENCES Linie
)

CREATE TABLE Flaechе (
  Flaechе_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Flaechе_ID) REFERENCES Raumbezugsart
)

CREATE TABLE Flaechenelement (
  Flaechenelement_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Flaechenelement_ID) REFERENCES Flaechе,
  beschrieben_durch_Linien
    INTEGER NOT NULL REFERENCES Linie_Flaechenelement
)

CREATE TABLE komplexe_Flaechе (
  komplexe_Flaechе_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (komplexe_Flaechе_ID) REFERENCES Flaechе,
  hat_erste_Flaechе
    INTEGER NOT NULL REFERENCES Flaechе,
  hat_zweite_Flaechе
    INTEGER NOT NULL REFERENCES Flaechе
)

CREATE TABLE Volumen (
  Volumen_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (Volumen_ID) REFERENCES Raumbezugsart
)

CREATE TABLE Volumen__hat_Flaechе (
  Volumen
    INTEGER NOT NULL REFERENCES Volumen,
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
    Flaechen
    INTEGER NOT NULL REFERENCES Flaechen
)
```

```
CREATE SCHEMA direct_positioning_schema
```

```
CREATE TABLE direct_position (
    direct_position_ID
        INTEGER NOT NULL PRIMARY KEY,
    in_pos_ref_system
        VARCHAR(30),
    Koordinate_1
        REAL NOT NULL,
    Koordinate_2
        REAL NOT NULL,
    Koordinate_3
        REAL,
    in_coordinate_system
        VARCHAR(30)
)
```

## 14.2 Teilschema „Historisierung“

Dieses Teilschema enthält die SQL-Definitionen für den Teilbereich „Historisierung“.

```
CREATE SCHEMA Historisierung
```

```
CREATE TABLE historisches_Objekt (
    historisches_Objekt_ID
        INTEGER NOT NULL PRIMARY KEY,
    Art_historisches_Objekt
        INTEGER NOT NULL REFERENCES Art_historisches_Objekt,
    erzeugt_von_Ereignis
        INTEGER NOT NULL REFERENCES Ereignis,
    geloescht_von_Ereignis
        INTEGER REFERENCES Ereignis,
    hat_Vorgaenger_hist_Objekt
        INTEGER REFERENCES historisches_Objekt
)
```

```
CREATE TABLE Art_historisches_Objekt (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(30) NOT NULL
)
```

```
INSERT INTO Art_historisches_Objekt VALUES (1, 'Strasse')
INSERT INTO Art_historisches_Objekt VALUES (2, 'Strasse_Netzknoten')
INSERT INTO Art_historisches_Objekt VALUES (3, 'Netzknoten')
INSERT INTO Art_historisches_Objekt VALUES (4, 'BAB_Knotennummer')
INSERT INTO Art_historisches_Objekt VALUES (5, 'Abschnitt')
INSERT INTO Art_historisches_Objekt VALUES (6, 'Ast')
INSERT INTO Art_historisches_Objekt VALUES (7, 'Verkehrliche_Verknuepfung')
INSERT INTO Art_historisches_Objekt VALUES (8, 'Nullpunkt')
INSERT INTO Art_historisches_Objekt VALUES (9, 'Kilometrierung')
INSERT INTO Art_historisches_Objekt VALUES (10, 'Teilbauwerk')
INSERT INTO Art_historisches_Objekt VALUES (11, 'Bruecke')
INSERT INTO Art_historisches_Objekt VALUES (12, 'Verkehrszeichenbruecke')
INSERT INTO Art_historisches_Objekt VALUES (13, 'Tunnel_Trogbauwerk')
INSERT INTO Art_historisches_Objekt VALUES (14, 'Laermschutzbauwerk')
INSERT INTO Art_historisches_Objekt VALUES (15, 'Stuetzbauwerk')
INSERT INTO Art_historisches_Objekt VALUES (16, 'sonstiges_Bauwerk')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Art_historisches_Objekt VALUES (17, 'Schichtbegrenzung')
INSERT INTO Art_historisches_Objekt VALUES (18, 'Streifenbegrenzung')
INSERT INTO Art_historisches_Objekt VALUES (19, 'Gewaesserkreuzung')
INSERT INTO Art_historisches_Objekt VALUES (20, 'Bahnkreuzung')
INSERT INTO Art_historisches_Objekt VALUES (21, 'Kreuzung_Strasse_Weg')
INSERT INTO Art_historisches_Objekt VALUES (22, 'Fussgaengerueberweg')
INSERT INTO Art_historisches_Objekt VALUES (23, 'Durchlass')
INSERT INTO Art_historisches_Objekt VALUES (24, 'Aufstellvorrichtung')
INSERT INTO Art_historisches_Objekt VALUES (25, 'Netzbereich')
INSERT INTO Art_historisches_Objekt VALUES (26, 'Bundesland')
INSERT INTO Art_historisches_Objekt VALUES (27, 'Regierungsbezirk')
INSERT INTO Art_historisches_Objekt VALUES (28, 'Kreis_kreisfreie_Stadt')
INSERT INTO Art_historisches_Objekt VALUES (29, 'Gemeindebezirk')
INSERT INTO Art_historisches_Objekt VALUES (30, 'Ortsteil')
INSERT INTO Art_historisches_Objekt VALUES (31, 'Land_Ministerium')
INSERT INTO Art_historisches_Objekt VALUES (32, 'Regierungspraesidium_Landesamt')
INSERT INTO Art_historisches_Objekt VALUES (33, 'Amt')
INSERT INTO Art_historisches_Objekt VALUES (34, 'Meisterei')
INSERT INTO Art_historisches_Objekt VALUES (35, 'Kreisverwaltung')
INSERT INTO Art_historisches_Objekt VALUES (36, 'Gemeindeverwaltung')
INSERT INTO Art_historisches_Objekt VALUES (37, 'UI_Vereinbarung')
INSERT INTO Art_historisches_Objekt VALUES (38, 'Baulast')
INSERT INTO Art_historisches_Objekt VALUES (39, 'Schutzzone')
INSERT INTO Art_historisches_Objekt VALUES (40, 'Block')
INSERT INTO Art_historisches_Objekt VALUES (41, 'OD_FS')
INSERT INTO Art_historisches_Objekt VALUES (42, 'Widmung')
INSERT INTO Art_historisches_Objekt VALUES (43, 'Leitung')
INSERT INTO Art_historisches_Objekt VALUES (44, 'Markierung')
INSERT INTO Art_historisches_Objekt VALUES (45, 'Schutzplanke')
INSERT INTO Art_historisches_Objekt VALUES (46, 'Nebenanlage')
INSERT INTO Art_historisches_Objekt VALUES (47, 'Rastanlage_nicht_bew')
INSERT INTO Art_historisches_Objekt VALUES (48, 'Rastanlage_bew')
INSERT INTO Art_historisches_Objekt VALUES (49, 'Nebenbetrieb')
INSERT INTO Art_historisches_Objekt VALUES (50, 'Schicht')
INSERT INTO Art_historisches_Objekt VALUES (51, 'Querschnittstreifen')
INSERT INTO Art_historisches_Objekt VALUES (52, 'Bauklasse_Oberbau')
INSERT INTO Art_historisches_Objekt VALUES (53, 'Bauweise_Oberbau')
INSERT INTO Art_historisches_Objekt VALUES (54, 'Regelquerschnitt')
INSERT INTO Art_historisches_Objekt VALUES (55, 'Bahnigkeit')
INSERT INTO Art_historisches_Objekt VALUES (56, 'Fahrbahnlaengsneigung')
INSERT INTO Art_historisches_Objekt VALUES (57, 'Fahrbahnquerneigung')
INSERT INTO Art_historisches_Objekt VALUES (58, 'Hindernis')
INSERT INTO Art_historisches_Objekt VALUES (59, 'SB_Wirkungsbereiche')
INSERT INTO Art_historisches_Objekt VALUES (60, 'raeumlicher_Gueltingkeitsber')
INSERT INTO Art_historisches_Objekt VALUES (61, 'Strecke')
INSERT INTO Art_historisches_Objekt VALUES (62, 'Strecke_SO')
INSERT INTO Art_historisches_Objekt VALUES (63, 'Teilnetz')
INSERT INTO Art_historisches_Objekt VALUES (64, 'Route')
INSERT INTO Art_historisches_Objekt VALUES (65, 'Strassenelement')
INSERT INTO Art_historisches_Objekt VALUES (66, 'sonstige_UI_Partner')
INSERT INTO Art_historisches_Objekt VALUES (67, 'Leitungsverlauf')
INSERT INTO Art_historisches_Objekt VALUES (68, 'Leitungstraeger')
INSERT INTO Art_historisches_Objekt VALUES (69, 'Leitungsart')
INSERT INTO Art_historisches_Objekt VALUES (70, 'entwaesserte_Fahrbahnflaeche')
INSERT INTO Art_historisches_Objekt VALUES (71, 'Lage_Entwaesserung')
INSERT INTO Art_historisches_Objekt VALUES (72, 'Art_der_Entwaesserung')
INSERT INTO Art_historisches_Objekt VALUES (73, 'Fahrstreifen_Nummer')
INSERT INTO Art_historisches_Objekt VALUES (74, 'max_Hoehe')
INSERT INTO Art_historisches_Objekt VALUES (75, 'Fkt_d_Verb_im_Knotenpktber')
INSERT INTO Art_historisches_Objekt VALUES (76, 'Schichtart')
INSERT INTO Art_historisches_Objekt VALUES (77, 'Schichtmaterial')
INSERT INTO Art_historisches_Objekt VALUES (78, 'Streifenart')
INSERT INTO Art_historisches_Objekt VALUES (79, 'Baustoff')
INSERT INTO Art_historisches_Objekt VALUES (80, 'Profil')
INSERT INTO Art_historisches_Objekt VALUES (81, 'automatische_Dauerzaehlstelle')
INSERT INTO Art_historisches_Objekt VALUES (82, 'Erfassungsergebnisse')
INSERT INTO Art_historisches_Objekt VALUES (83, 'Klassenunterscheidung')
INSERT INTO Art_historisches_Objekt VALUES (84, 'Fahrstreifenzuordnung')
INSERT INTO Art_historisches_Objekt VALUES (85, 'Art_der_registrierten_Daten')
INSERT INTO Art_historisches_Objekt VALUES (86, 'Detektorart')
INSERT INTO Art_historisches_Objekt VALUES (87, 'Art_der_Registrierung')
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
INSERT INTO Art_historisches_Objekt VALUES (88, 'Fahrstreifenzuteilung')
INSERT INTO Art_historisches_Objekt VALUES (89, 'manuelle_Zaehlstelle')
INSERT INTO Art_historisches_Objekt VALUES (90, 'Gueltigkeitsbereich_AV')
INSERT INTO Art_historisches_Objekt VALUES (91, 'Hochrechnungsergebnisse')
INSERT INTO Art_historisches_Objekt VALUES (92, 'Angaben_z_Strassenquerschnitt')
INSERT INTO Art_historisches_Objekt VALUES (93, 'Richtungskennung')
INSERT INTO Art_historisches_Objekt VALUES (94, 'Bemerkungskennziffern')
INSERT INTO Art_historisches_Objekt VALUES (95, 'benachbarte_Anschlussstellen')
INSERT INTO Art_historisches_Objekt VALUES (96, 'Bezugszaehlstelle_Auslverkehr')
INSERT INTO Art_historisches_Objekt VALUES (97, 'Bezugszaehlstelle_Gesamtverk')
INSERT INTO Art_historisches_Objekt VALUES (98, 'VBA_Wetterstation')
INSERT INTO Art_historisches_Objekt VALUES (99, 'SWIS_GMA_Messstelle')
INSERT INTO Art_historisches_Objekt VALUES (100, 'Kombination_von_Anlagentypen')
INSERT INTO Art_historisches_Objekt VALUES (101, 'SB_Anlagentyp')
INSERT INTO Art_historisches_Objekt VALUES (102, 'SB_Anzeigezustaende')
INSERT INTO Art_historisches_Objekt VALUES (103, 'WW_Anzeigezustaende')
INSERT INTO Art_historisches_Objekt VALUES (104, 'WW_Wirkungsbereiche')
INSERT INTO Art_historisches_Objekt VALUES (105, 'KB_Anlagentyp')
INSERT INTO Art_historisches_Objekt VALUES (106, 'KB_Anzeigezustaende')
INSERT INTO Art_historisches_Objekt VALUES (107, 'richtungsbezogener_WB')
INSERT INTO Art_historisches_Objekt VALUES (108, 'Angaben_zum_Knotenpunkt')
INSERT INTO Art_historisches_Objekt VALUES (109, 'Betriebsform')
INSERT INTO Art_historisches_Objekt VALUES (110, 'RW_Anzeigezustaende')
INSERT INTO Art_historisches_Objekt VALUES (111, 'RW_Wirkungsbereich')
INSERT INTO Art_historisches_Objekt VALUES (112, 'Art_der_Stromversorgung')
INSERT INTO Art_historisches_Objekt VALUES (113, 'manuelle_Steuerung')
INSERT INTO Art_historisches_Objekt VALUES (114, 'Datenuebertragungssystem')
INSERT INTO Art_historisches_Objekt VALUES (115, 'Unterzentrale')
INSERT INTO Art_historisches_Objekt VALUES (116, 'Verkehrsrechnerzentrale')
INSERT INTO Art_historisches_Objekt VALUES (117, 'Ueberwachung_der_Anlage')
INSERT INTO Art_historisches_Objekt VALUES (118, 'Wechselverkehrszeichen')
INSERT INTO Art_historisches_Objekt VALUES (119, 'Art_der_Aufstellung')
INSERT INTO Art_historisches_Objekt VALUES (120, 'Anzahl_der_Anzeigequerschnitte')
INSERT INTO Art_historisches_Objekt VALUES (121, 'Detektionsmethode')
INSERT INTO Art_historisches_Objekt VALUES (122, 'Art_der_Anordnung')
INSERT INTO Art_historisches_Objekt VALUES (123, 'Art_der_erfassten_Daten')
INSERT INTO Art_historisches_Objekt VALUES (124, 'Anzahl_der_Messquerschnitte')
INSERT INTO Art_historisches_Objekt VALUES (125, 'Wegweiserbild')
INSERT INTO Art_historisches_Objekt VALUES (126, 'Beschriftungselemente_Symbole')
INSERT INTO Art_historisches_Objekt VALUES (127, 'zeitlicher_Gueltigkeitsber')
INSERT INTO Art_historisches_Objekt VALUES (128, 'Rotlichtueberwachung')
INSERT INTO Art_historisches_Objekt VALUES (129, 'Angaben_zum_Knotenpunkt_LSA')
INSERT INTO Art_historisches_Objekt VALUES (130, 'Grundlage_und_Inbetriebn_daten')
INSERT INTO Art_historisches_Objekt VALUES (131, 'Angaben_zu_Verkehrsdaten')
INSERT INTO Art_historisches_Objekt VALUES (132, 'Signalprogrammparameter')
INSERT INTO Art_historisches_Objekt VALUES (133, 'hinterlegte_Programme')
INSERT INTO Art_historisches_Objekt VALUES (134, 'akust_oder_taktile_Signalgeber')
INSERT INTO Art_historisches_Objekt VALUES (135, 'Angaben_zum_Steuergeraet')
INSERT INTO Art_historisches_Objekt VALUES (136, 'Angaben_zur_Verkabelung')
INSERT INTO Art_historisches_Objekt VALUES (137, 'Angaben_z_Aufstellvorrichtung')
INSERT INTO Art_historisches_Objekt VALUES (138, 'optische_Signalgeber')
INSERT INTO Art_historisches_Objekt VALUES (139, 'Erfassungseinrichtungen_IV')
INSERT INTO Art_historisches_Objekt VALUES (140, 'Erfassungseinrichtungen_OeV')
INSERT INTO Art_historisches_Objekt VALUES (141, 'uebergeordnete_Zentrale')
INSERT INTO Art_historisches_Objekt VALUES (142, 'einfacher_Zeitraum')
INSERT INTO Art_historisches_Objekt VALUES (143, 'komplexer_Zeitraum')
```

```
CREATE TABLE Ereignis (
    Ereignis_ID
        INTEGER NOT NULL PRIMARY KEY,
    laufende_Nummer
        INTEGER NOT NULL,
    Wirksamkeitsdatum
        CHAR(10) NOT NULL, -- Datum
    CHECK ( Wirksamkeitsdatum LIKE '____.____.____' ),
    gehoert_zu_Projekt
        INTEGER REFERENCES Projekt_Strassenbau,
    hat_Dokument_zur_Erlaeuterung
        INTEGER REFERENCES Administration.Dokument
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Ereignis__in_Bauamt (
  Ereignis
    INTEGER NOT NULL REFERENCES Ereignis,
  Strassenbaudienststelle
    INTEGER NOT NULL REFERENCES Administration.Strassenbaudienststelle
)

CREATE TABLE Projekt_Strassenbau (
  Projekt_Strassenbau_ID
    INTEGER NOT NULL PRIMARY KEY
)

CREATE TABLE identisches_Netzteil (
  identisches_Netzteil_ID
    INTEGER NOT NULL PRIMARY KEY,
  mit_Veraenderungsart
    CHAR(2) REFERENCES Veraenderungsart,
  war_Teilabschnitt
    INTEGER REFERENCES Teilabschnitt_IdNT,
  wurde_zu_Teilabschnitt
    INTEGER REFERENCES Teilabschnitt_IdNT,
  Teil_von_Ereignis
    INTEGER NOT NULL REFERENCES Ereignis,
  CHECK ( ( war_Teilabschnitt IS NOT NULL ) OR ( wurde_zu_Teilabschnitt IS NOT
NULL ) )
)

CREATE TABLE Veraenderungsart (
  Kennung
    CHAR(2) NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Veraenderungsart VALUES ('NB','Neubau')
INSERT INTO Veraenderungsart VALUES ('VB','Vollausbau')
INSERT INTO Veraenderungsart VALUES ('ZB','Zwischenausbau')
INSERT INTO Veraenderungsart VALUES ('RE','Rekultivierung')
INSERT INTO Veraenderungsart VALUES ('DT','Deckenerneuerung im Tiefeinbau')
INSERT INTO Veraenderungsart VALUES ('DH','Deckenerneuerung im Hocheinbau')
INSERT INTO Veraenderungsart VALUES ('WI','Widmung')
INSERT INTO Veraenderungsart VALUES ('AU','Aufstufung')
INSERT INTO Veraenderungsart VALUES ('AB','Abstufung')
INSERT INTO Veraenderungsart VALUES ('UM','Ummumerierung in der ' ||
'Straßenbezeichnung')
INSERT INTO Veraenderungsart VALUES ('EZ','Einziehung')
INSERT INTO Veraenderungsart VALUES ('OD','Verlegung der OD-Grenze')
INSERT INTO Veraenderungsart VALUES ('UI','UI-Vertrag')
INSERT INTO Veraenderungsart VALUES ('GR','Grenzverlegung')
INSERT INTO Veraenderungsart VALUES ('SB','Änderung der Abschnitts-/ ' ||
'Astbezeichnung')
INSERT INTO Veraenderungsart VALUES ('LN','Längenänderung infolge Neumessung')
INSERT INTO Veraenderungsart VALUES ('KO','Korrektur')

CREATE TABLE Teilabschnitt_IdNT (
  Teilabschnitt_IdNT_ID
    INTEGER NOT NULL PRIMARY KEY,
  Station_1
    REAL NOT NULL, -- Meter
  CHECK ( Station_1>=0 ),
  auf_Abschnitt_oder_Ast
    INTEGER NOT NULL REFERENCES Strassennetz.Abschnitt_oder_Ast,
  Station_2
    REAL NOT NULL, -- Meter
  CHECK ( Station_2>=0 ),
)
```

#### 14.3 Teilschema „Allgemeine Objekte“

Dieses Teilschema enthält die SQL-Definitionen für die Teilbereiche „Allgemeine Objekte“ und „Zeitraum“.

```
CREATE SCHEMA Allgemeine_Objekte

CREATE TABLE Wochentag (
    Kennung
        INTEGER NOT NULL PRIMARY KEY,
    Langtext
        VARCHAR(10) NOT NULL
)

INSERT INTO Wochentag VALUES (1,'Sonntag')
INSERT INTO Wochentag VALUES (2,'Montag')
INSERT INTO Wochentag VALUES (3,'Dienstag')
INSERT INTO Wochentag VALUES (4,'Mittwoch')
INSERT INTO Wochentag VALUES (5,'Donnerstag')
INSERT INTO Wochentag VALUES (6,'Freitag')
INSERT INTO Wochentag VALUES (7,'Samstag')

CREATE TABLE Startdatum (
    Startdatum_ID
        INTEGER NOT NULL PRIMARY KEY,
    Jahr
        INTEGER,
    Monat_im_Jahr
        INTEGER,
    Woche_im_Jahr
        INTEGER,
    Tag_im_Monat
        INTEGER,
    vorwaerts_oder_rueckwaerts
        INTEGER REFERENCES Logical_Types.Boolean,
    Woche_im_Monat
        INTEGER,
    Tag_in_der_Woche
        INTEGER,
    Stunde_am_Tag
        INTEGER,
    Minute_in_der_Stunde
        INTEGER,
    Sekunde_in_der_Minute
        INTEGER,
    CHECK ( ( Monat_im_Jahr IS NULL ) OR ( Woche_im_Jahr IS NULL ) )
)

CREATE TABLE Dauer (
    Dauer_ID
        INTEGER NOT NULL PRIMARY KEY,
    Jahre
        INTEGER,
    Monate
        INTEGER,
    Wochen
        INTEGER,
    Tage
        INTEGER,
    Stunden
        INTEGER,
    Minuten
        INTEGER,
    Sekunden
        INTEGER
)
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
CREATE TABLE Operator (
  Kennung
    CHAR(1) NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(255) NOT NULL
)

INSERT INTO Operator VALUES ('+', 'Vereinigung')
INSERT INTO Operator VALUES ('*', 'Durchschnitt')
INSERT INTO Operator VALUES ('-', 'Differenz')

CREATE TABLE Zeitraum (
  Zeitraum_ID
    INTEGER NOT NULL PRIMARY KEY,
  hat_ersten_Zeitraum
    INTEGER REFERENCES Zeitraum,
  hat_zweiten_Zeitraum
    INTEGER REFERENCES Zeitraum,
  hat_Operator
    CHAR(1) REFERENCES Operator,
  hat_Startdatum
    INTEGER REFERENCES Startdatum,
  hat_Dauer
    INTEGER REFERENCES Dauer,
  FOREIGN KEY (Zeitraum_ID) REFERENCES Historisierung.historisches_Objekt,
  gueltig_von
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_von LIKE '___.__.____' ),
  gueltig_bis
    CHAR(10) NOT NULL, -- Datum
  CHECK ( gueltig_bis LIKE '___.__.____' ),
  Art_Zeitraum
    INTEGER NOT NULL REFERENCES Art_Zeitraum
)

CREATE TABLE Art_Zeitraum (
  Kennung
    INTEGER NOT NULL PRIMARY KEY,
  Langtext
    VARCHAR(18) NOT NULL
)

INSERT INTO Art_Zeitraum VALUES (1, 'einfacher_Zeitraum')
INSERT INTO Art_Zeitraum VALUES (2, 'komplexer_Zeitraum')

CREATE TABLE verwaltungstechn_Zuordnung (
  verwaltungstechn_Zuordnung_ID
    INTEGER NOT NULL PRIMARY KEY,
  hat_Strassenbaudienststelle
    INTEGER NOT NULL REFERENCES Administration.Strassenbaudienststelle,
  hat_Verwaltungsbezirk
    INTEGER NOT NULL REFERENCES Administration.Verwaltungsbezirk
)

CREATE TABLE oertliche_Zuordnung (
  oertliche_Zuordnung_ID
    INTEGER NOT NULL PRIMARY KEY,
  liegt_auf_Strassenpunkt
    INTEGER REFERENCES oertliche_Zuordnung_SP,
  liegt_auf_nicht_klass_Strasse
    INTEGER REFERENCES nicht_nach_ASB_klass_Strasse,
  CHECK ( ( liegt_auf_Strassenpunkt IS NOT NULL ) OR (
  liegt_auf_nicht_klass_Strasse IS NOT NULL ) ) AND ( ( liegt_auf_Strassenpunkt IS
  NULL ) OR ( liegt_auf_nicht_klass_Strasse IS NULL ) ) )
)

CREATE TABLE oertliche_Zuordnung_SP (
  oertliche_Zuordnung_SP_ID
    INTEGER NOT NULL PRIMARY KEY,
  FOREIGN KEY (oertliche_Zuordnung_SP_ID) REFERENCES
  Strassennetz.Punktobjekt_stat
```

## OKSTRA – Teilprojekt 4 – Schlußbericht

### Teilbericht D - Das SQL-Datenschema

---

```
)  
  
CREATE TABLE nicht_nach_ASB_klass_Strasse (  
    nicht_nach_ASB_klass_Strasse_ID  
        INTEGER NOT NULL PRIMARY KEY,  
    Strassenklasse  
        VARCHAR(50) NOT NULL,  
    Nummer  
        INTEGER NOT NULL,  
    Lage  
        VARCHAR(28) NOT NULL  
)  
  
CREATE SCHEMA Logical_Types  
  
CREATE TABLE Boolean (  
    Boolean_ID  
        INTEGER NOT NULL PRIMARY KEY,  
    Boolean  
        VARCHAR (5)  
)  
  
INSERT INTO Boolean VALUES ( 0 , 'FALSE' )  
INSERT INTO Boolean VALUES ( 1 , 'TRUE' )
```