



Beschreibung des SQLite-Datenbankformats

Inhalt

Motivation.....	1
Struktur des Datenbankformats.....	1
Beispiel.....	6

Motivation

Dieses Dokument beschreibt ein auf der Open-Source-Datenbank *SQLite*¹ inklusive der Erweiterung *SpatialLite*² basierendes Datenbankformat, mit dem sich OKSTRA®-Daten in Form einer Datenbankdatei darstellen und austauschen lassen. Die Repräsentation als SQLite-Datenbankdatei bietet dabei folgende Vorteile:

- Die Daten können in dem Datenbankformat nicht nur dargestellt und ausgetauscht, sondern unter Zuhilfenahme der Datenbankanwendung SQLite bzw. entsprechenden Tools wie z.B. dem *DB Browser for SQLite*³ auch in vielfältiger Weise abgefragt und ausgewertet werden. Dazu steht der mächtige Sprachumfang der Datenbank-Abfragesprache SQL zur Verfügung.
- SQLite-Datenbankdateien mit SpatialLite-Geometrien können als Datenquelle für die freie GIS-Software *QGIS*⁴ oder das kommerzielle GIS-Produkt *ArcGIS pro*⁵ verwendet und damit grafisch betrachtet und analysiert werden.

Zusätzliche Information (nicht normativ): Die OKSTRA®-Klassenbibliothek und das darauf basierende OKSTRA®-Werkzeug können SQLite-Datenbankdateien gemäß der vorliegenden Spezifikation einlesen und bei Bedarf in andere Repräsentationsformen wandeln (z.B. in OKSTRA®-XML-Dateien). Mit dem SQLite-Datenbankformat steht somit ein vollwertiges Datenaustauschformat zur Verfügung, das in Datenaustausch-Vorgängen als Alternative zu OKSTRA®-XML verwendet werden kann.

Struktur des Datenbankformats

Allgemeines

Eine SQLite-Datenbankdatei dient zur Darstellung eines OKSTRA®-Datenbestandes.

Sofern in diesem Dokument nicht explizit anders angegeben erhält jede Datenbanktabelle in der Datenbankdatei eine Spalte namens "OID" vom Datentyp *text*, in der tabellenweit eindeutige, aber ansonsten beliebig gestaltbare Identifikatoren für die einzelnen Datensätze eingetragen werden. Diese Spalte wird jeweils als Primärschlüssel festgelegt.

Die Reihenfolge der Spalten in einer Datenbanktabelle ist beliebig.

¹ <https://www.sqlite.org/index.html>

² <https://www.gaia-gis.it/fossil/libspatialite/index>

³ <https://sqlitebrowser.org/>

⁴ <https://www.qgis.org/de/site/>

⁵ <https://www.esri.com/de-de/arcgis/products/arcgis-pro/overview>



Namen aus dem OKSTRA®-UML-Modell, die zur Bezeichnung von Datenbankentitäten (Tabellen, Tabellenspalten etc.) herangezogen werden und Minuszeichen (-) enthalten oder mit einer Ziffer beginnen, werden in doppelte Anführungszeichen gesetzt ("...").

Da im OKSTRA® Geometriedaten eine Rolle spielen, muss zur Erzeugung einer SQLite-Datenbankdatei gemäß dem vorliegenden Dokument zwingend die SpatialLite-Erweiterung geladen werden. Die Berücksichtigung von Fremdschlüsselbedingungen (foreign key constraints) muss über das SQL-Statement

PRAGMA foreign_keys = ON;

aktiviert werden.

Objektarten, komplexe Datentypen, Unionen

Für jede instanzierbare Objektart, jeden komplexen Datentypen und jede Union, die in dem darzustellenden Datenbestand auftritt, wird eine eigene Datenbanktabelle mit dem Namen des jeweiligen Modellkonstrukts angelegt. Ein Datensatz in dieser Tabelle entspricht dabei einer Instanz des betreffenden Konstrukts.

Für jedes Attribut des Konstrukts, dessen Datentyp ein elementarer Datentyp oder eine Schlüsseltable ist, – auch für die geerbten – existiert eine Tabellenspalte, die den Namen des betreffenden Attributs trägt. Für die Datentypen dieser Tabellenspalten siehe den Abschnitt *Attribute mit elementarem oder Schlüsseltable-Datentyp*.

Für Attribute des Konstrukts, deren Datentyp ein komplexer Datentyp oder eine Union ist, existieren hingegen keine Tabellenspalten. Sie werden stattdessen über eine Zwischentabelle angebunden, siehe den Abschnitt *Relationen und Attribute mit komplexem oder Unions-Datentyp*.

Schlüsseltable

Für jede Schlüsseltable, die in dem darzustellenden Datenbestand auftritt, wird eine eigene Datenbanktabelle mit dem Namen der Schlüsseltable angelegt. Ein Datensatz in dieser Tabelle entspricht einem Wertekatalog-Eintrag der betreffenden Schlüsseltable. Für jedes Attribut der Schlüsseltable existiert eine Tabellenspalte, die den Namen des betreffenden Attributs trägt (bei Schlüsseltable sind im OKSTRA®-UML-Modell nur Attribute vom Datentyp *CharacterString* – d.h. nur elementare Attribute – zulässig, keine komplexen Datentypen oder Unionen).

Zusätzlich erhält eine Datenbanktabelle zur Abbildung einer Schlüsseltable eine Spalte namens "SCHEMA" vom Datentyp *bool*. In dieser Spalte wird vermerkt, ob der jeweilige Wertekatalog-Eintrag im OKSTRA®-UML-Modell hinterlegt ist oder nicht. Diese Angabe hat rein dokumentarischen Charakter und erleichtert u.a. Auswertungen der Datenbank im Hinblick auf nicht OKSTRA®-konforme Wertekatalog-Einträge. Beim Datenimport mit der OKSTRA®-Klassenbibliothek wird diese Spalte nicht ausgewertet.

Attribute mit elementarem oder Schlüsseltable-Datentyp

Zur Abbildung eines Attributs, das einen elementaren oder einen Schlüsseltable-Datentyp besitzt, dient eine Tabellenspalte in der Datenbanktabelle, die den "Eigentümer" des Attributs repräsentiert (je nach Kontext eine instanzierbare Objektart, ein komplexer Datentyp, eine Union oder eine Schlüsseltable). Der Datentyp einer solchen Tabellenspalte richtet sich nach dem Datentyp im OKSTRA®-UML-Modell. Bei elementaren Datentypen gilt dabei folgende Zuordnung:



Datentyp im OKSTRA®-UML-Modell	Datentyp in der SQLite-Datenbank
<i>ClockTime</i>	<i>timestamp</i> (Datumsangabe wird nicht berücksichtigt)
<i>Date</i>	<i>timestamp</i> (Zeitangabe wird nicht berücksichtigt)
<i>Sequence<Bit></i>	<i>text</i> (Binärdaten in Base64-Codierung)
<i>Integer</i> (und Ableitungen daraus)	<i>int</i>
<i>Real</i> (und Ableitungen daraus)	<i>double precision</i>
<i>CharacterString</i> (und Ableitungen daraus)	<i>text</i>
<i>Boolean</i>	<i>int</i>
Ableitungen aus <i>Measure</i> (<i>Meter</i> etc.)	<i>double precision</i>
<i>GM_Point</i>	<i>MULTIPOINT</i> (für 2D-Punkte) <i>MULTIPOINTZ</i> (für 3D-Punkte)
<i>GM_Curve</i>	<i>MULTILINESTRING</i> (für 2D-Linien) <i>MULTILINESTRINGZ</i> (für 3D-Linien)
<i>GM_Surface</i>	<i>MULTIPOLYGON</i> (für 2D-Flächen) <i>MULTIPOLYGONZ</i> (für 3D-Flächen)
<i>GM_Solid</i>	<i>MULTIPOLYGONZ</i>
<i>GM_MultiPoint</i>	<i>MULTIPOINT</i> (für 2D-Punkte) <i>MULTIPOINTZ</i> (für 3D-Punkte)
<i>GM_MultiCurve</i>	<i>MULTILINESTRING</i> (für 2D-Linien) <i>MULTILINESTRINGZ</i> (für 3D-Linien)
<i>GM_MultiSurface</i>	<i>MULTIPOLYGON</i> (für 2D-Flächen) <i>MULTIPOLYGONZ</i> (für 3D-Flächen)
<i>GM_MultiSolid</i>	<i>MULTIPOLYGONZ</i> ⁶

Im Hinblick auf die Geometriedatentypen wird davon ausgegangen, dass im gesamten Datenbestand konsequent entweder 2D- oder 3D-Geometrien vorhanden sind; eine Angabe dazu muss in der Tabelle "metadaten" erfolgen, vgl. Abschnitt *Metadaten*. Eine Mischung von 2D- und 3D-Geometrien ist nicht vorgesehen.

Geometriespalten müssen in der Datenbank über die Funktion *AddGeometryColumn()* definiert werden.

Bei der Abbildung des Datentypen *GM_MultiSolid* ist zu berücksichtigen, dass nur einfach zusammenhängende Volumenkörper sinnvoll in das Datenbankformat überführt werden können, weil in dem Datentyp *MULTIPOLYGONZ* lediglich eine Ansammlung von Randflächen gespeichert werden kann und keine Zuordnung der Randflächen zu unterschiedlichen Volumenkörpern möglich ist.

Zusätzliche Information (nicht normativ): Die Implementierung der OKSTRA®-Klassenbibliothek sieht bei einigen Objektarten vor, dass abgeleitete Geometrie ermittelt werden kann (z.B. die geometrische Lage eines Strassenpunktes anhand der Station auf dem Abschnitt). Diese berechneten Geometrien können auch in der Datenbank in zusätzlichen Geometriespalten abgelegt werden, sie haben rein informativen Charakter und erleichtern beispielsweise Visualisierung und Analyse der Daten mit GIS-Werkzeugen. Beim Einlesen von Daten aus einer Datenbank in die OKSTRA®-Klassenbibliothek werden diese künstlich ergänzten Geometriespalten nicht berücksichtigt. Der Name einer hinzugefügten Geometriespalte ergibt sich aus dem jeweiligen Geometriotyp (Punkt: "GeoPunkt", Linie: "GeoLinie", Fläche: "GeoFlaeche", Volumen: "GeoVolumen").

⁶ beschränkt auf einfach zusammenhängende Volumenkörper



Wenn es sich bei dem Datentyp eines Attributs im OKSTRA®-UML-Modell um eine Schlüsseltablette handelt, wird in der SQLite-Datenbank der Datentyp *text* verwendet. Sofern das Attribut belegt ist, wird dort die OID (also der Primärschlüssel) des betreffenden Wertekatalogeintrags in der zur Schlüsseltablette gehörenden Datenbanktabelle angegeben. Um zu garantieren, dass nur OIDs aus dieser Datenbanktabelle eingetragen werden können, wird eine entsprechende Fremdschlüsselbedingung (foreign key constraint) eingeführt.

Die Werte von *CharacterString*-Attributen sind für den Eintrag in das zugehörige *text*-Feld der Datenbank in einfache Hochkommata ('...') zu setzen. Innerhalb des anzugebenden Strings auftretende einfache Hochkommata (') und Backslashes (\) sind dabei entweder durch einen vorgesetzten Backslash oder durch Verdoppelung zu quoten (dies ergibt dann im Fall des Hochkommata \' oder "', im Fall des Backslashes in beiden Varianten \\).

Multiple Attributwerte werden in Form einer Mengennotation angegeben. Dazu werden die einzelnen Attributwerte durch Kommata getrennt und von geschweiften Klammern umgeben.

Beispiel: {Attributwert1, Attributwert2, Attributwert3}

Relationen und Attribute mit komplexem oder Unions-Datentyp

Zur Darstellung der Relationen innerhalb des Datenbestandes dient eine Datenbanktabelle namens "zwischenstab". Ein Datensatz in dieser Tabelle entspricht einer Relation von einem Objekt oder einem komplexen Datentyp zu einem anderen Objekt. Gibt es mehrere Ziel-Objekte (multiple Relation), werden entsprechend viele Datensätze eingetragen.

Die Tabelle "zwischenstab" repräsentiert darüber hinaus diejenigen Attribute von Objekten, komplexen Datentypen und Unionen, deren Datentyp ein komplexer Datentyp oder eine Union ist. Diese – im OKSTRA®-UML-Modell in Form von Attributen modellierten Beziehungen – werden in der Tabelle "zwischenstab" analog zu Relationen dargestellt. Das Verweisziel ist in diesen Fällen allerdings kein Objekt, sondern der entsprechende komplexe Datentyp bzw. die Union.

Relationen zwischen Objekten werden beidseitig eingetragen, d.h. sie werden über je zwei Datensätze (für beide Relationsrichtungen) repräsentiert.

Relationen, die von komplexen Datentypen ausgehen, und die o.g. Attributbeziehungen werden einseitig eingetragen, d.h. sie werden über je einen Datensatz repräsentiert.

Die Tabelle "zwischenstab" besitzt folgende Spalten:

Name	Datentyp	Bedeutung
OID	<i>text</i>	Identifikator des jeweiligen Datensatzes (muss zusammen mit dem Inhalt des Feldes "ROLE" innerhalb der Tabelle eindeutig sein)
ROLE	<i>text</i>	Rollenbezeichnung der Beziehung (der Relation bzw. des Attributs) beim Quellkonstrukt
ID	<i>text</i>	OID des Quellkonstrukts
RID	<i>text</i>	OID des Zielkonstrukts
SEQNR	<i>int</i>	Folgenummer; die Datensätze zu einer Rollenbezeichnung eines bestimmten Quellkonstrukts werden mit 0 beginnend hochgezählt
SOURCE	<i>text</i>	Name der Quellkonstrukts (Objektart, komplexer Datentyp, Union), von dem die Beziehung ausgeht
TARGET	<i>text</i>	Name der Zielkonstrukts (Objektart, komplexer Datentyp, Union), auf das die Beziehung zeigt



Abweichend von der sonstigen Verfahrensweise muss in der Tabelle "zwischenstab" nicht der Inhalt des Feldes "OID" allein, sondern nur zusammen mit dem Inhalt des Feldes "ROLE" innerhalb der Tabelle eindeutig sein. Entsprechend wird die Kombination beider Felder als Primärschlüssel festgelegt.

Für einen performanten Zugriff auf die in der Tabelle "zwischenstab" gespeicherten Beziehungsinformationen wird die Einrichtung von Datenbankindizes auf folgenden Spalten bzw. Spaltenkombinationen empfohlen:

1. Spalten "ROLE", "ID", "RID", "SOURCE", "TARGET"
2. Spalten "ID", "SOURCE"
3. Spalte "ID"
4. Spalte "RID"

Metadaten

Allgemeine Metadaten zum in der SQLite-Datenbankdatei gespeicherten Datenbestand werden in der Datenbanktabelle "metadaten" abgelegt. Die Repräsentation der Metadaten erfolgt dabei in Form von Schlüssel-Wert-Paaren. Entsprechend besitzt die Tabelle "metadaten" zwei Spalten namens "KEY" und "VALUE" vom Datentyp *text*. Die Tabelle "metadaten" verfügt nicht über eine Spalte namens "OID" und es ist kein Primärschlüssel festgelegt.

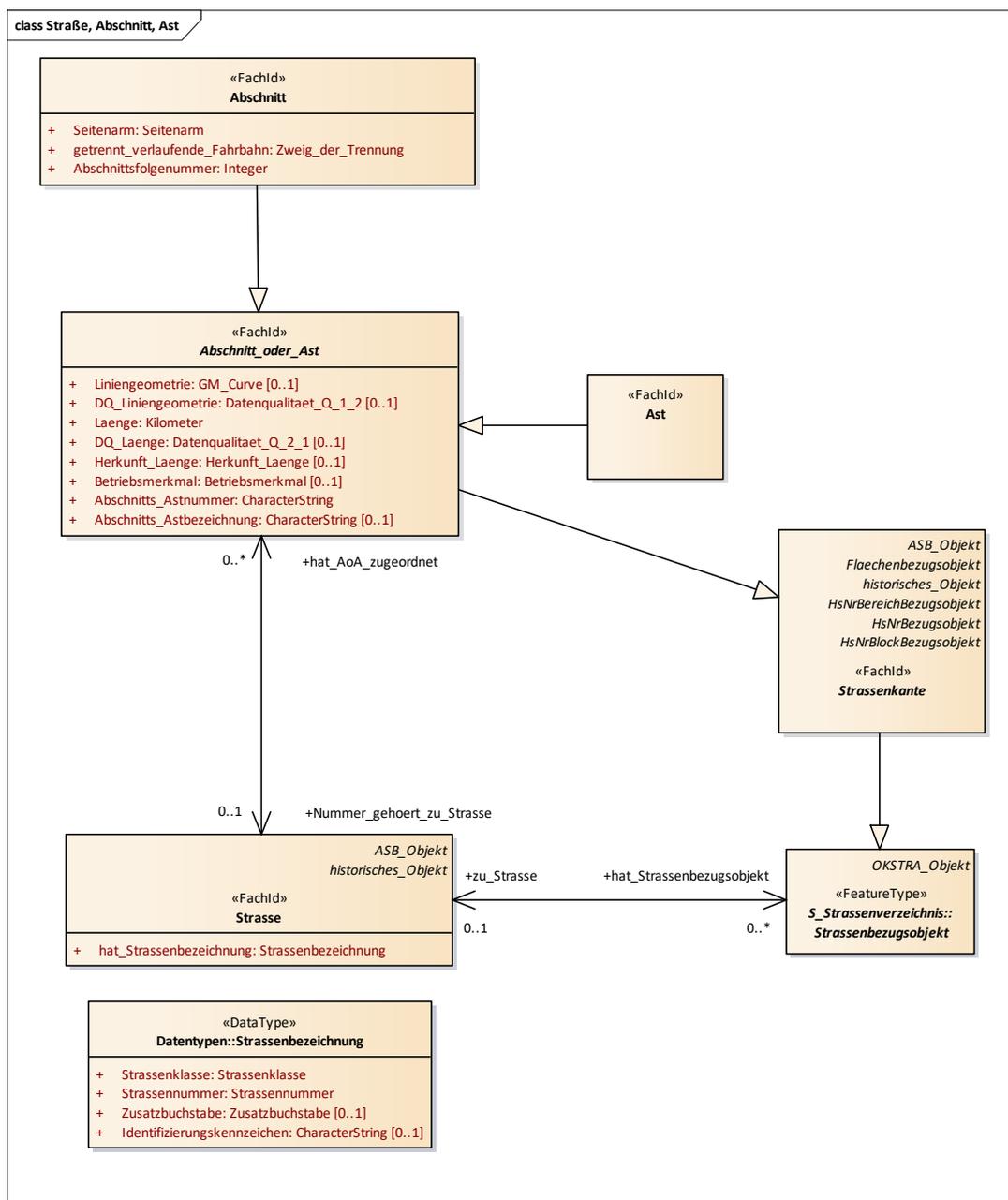
Die Tabelle "metadaten" muss folgende Einträge enthalten:

KEY	mögliche Werte für VALUE	Bedeutung
dimension	'2' oder '3'	Dimension der enthaltenen Koordinatengeometrien
hoehensystem	Bezeichnungen für Höhenreferenzsysteme gemäß OKSTRA®-Dokument N0135, z.B. 'DE_DHHN92_NH'	Koordinatenreferenzsystem für Höhenangaben
kodierung	'windows-1252' oder 'utf-8'	Zeichenkodierung der eingetragenen Strings
version	alle OKSTRA-Versionsnummern, z.B. 'OKSTRA-2.020'	OKSTRA®-Versionsnummer in der Syntax OKSTRA-x.xxx
dbversion	'1.0'	Versionsnummer des SQLite-Datenbankformats



Beispiel

Zur Illustration soll exemplarisch die Abbildung von einer Instanz der Objektart *Strasse* und zwei Instanzen der Objektart *Abschnitt* in das SQLite-Datenbankformat beschrieben werden. Den entsprechenden Ausschnitt aus dem OKSTRA®-UML-Modell (Version 2.020) zeigt das folgende UML-Klassendiagramm. Die beiden *Abschnitt*-Instanzen sollen dabei über die von der abstrakten Objektart *Strassenbezugsobjekt* geerbte Relation "zu_Strasse" mit der *Strasse*-Instanz verknüpft werden. Darüber hinaus tritt in der Objektart *Strasse* mit dem Pflichtattribut "hat_Strassenbezeichnung" ein Attribut auf, das einen komplexen Datentyp besitzt und dessen Abbildung hier ebenfalls dargestellt wird. Außerdem werden einige Schlüsselattributen belegt.





Datenbanktabelle "Strasse"

Die Tabelle "Strasse" dient zur Darstellung von Instanzen der Objektart *Strasse*. Sie enthält Spalten für alle – auch die von den Objektarten *ASB_Objekt* und *historisches_Objekt* geerbten – Attribute. Ausgenommen davon sind nur diejenigen Attribute, die einen komplexen oder Union-Datentyp besitzen und daher über die Zwischentabelle "zwischenstab" mit der jeweiligen *Strasse*-Instanz verknüpft werden. Aus diesem Grund gibt es in der Tabelle "Strasse" für das Attribut "hat_Strassenbezeichnung", das einen komplexen Datentyp besitzt, keine Spalte.

Zur Erhöhung der Übersichtlichkeit werden in der folgenden Darstellung einige Spalten für geerbte Attribute weggelassen, was durch drei Punkte (...) verdeutlicht wird.

Bei der Spalte "GeoLinie" am Ende der Tabelle handelt es sich um eine für die Datenbankdarstellung von der OKSTRA®-Klassenbibliothek ergänzte Spalte, in der eine abgeleitete Liniengeometrie für eine *Strasse* abgelegt werden kann.

Nicht belegte Felder eines Datensatzes besitzen den Wert *NULL*. Zur Angabe von Geometrie wird ein spezielles Binärformat verwendet, das aus dem Erweiterungspaket *SpatialLite* stammt und datentechnisch gesehen als *BLOB* ("Binary Large Object") abgelegt wird.

OID	OKSTRA_ID	Name	Textfeld	RFID	...	gueltig_von	gueltig_bis	GeoLinie
2673	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>	...	<i>NULL</i>	<i>NULL</i>	<i>BLOB</i>

Datenbanktabelle "Strassenbezeichnung"

In die Tabelle "Strassenbezeichnung" werden Instanzen des komplexen Datentyps *Strassenbezeichnung* eingetragen. Dieser Datentyp hat nur Attribute, deren Datentypen Schlüssel Tabellen oder elementare Datentypen sind. Deshalb existiert für jedes Attribut des OKSTRA®-UML-Modells eine Tabellenspalte.

Bei der Spalte "Strassenklasse" handelt es sich um die Abbildung eines Attributs, dessen Datentyp eine Schlüssel Tabelle ist. Dort wird die OID des betreffenden Wertekatalogeintrags aus der Datenbanktabelle zur Abbildung der Schlüssel Tabelle eingetragen (diese Datenbanktabelle wird im Anschluss vorgestellt).

OID	Strassenklasse	Strassennummer	Zusatzbuchstabe	Identifizierungskennzeichen
2675	<i>Strassenklasse.A</i>	2	<i>NULL</i>	<i>NULL</i>



Datenbanktabelle "Strassenklasse"

In die Tabelle "Strassenklasse" werden die Wertekatalogeinträge der gleichnamigen Schlüsseltable als Datensätze eingetragen. Die Spalte "SCHEMA" gibt an, ob der betreffende Eintrag aus dem OKSTRA®-UML-Modell stammt (was hier für alle Einträge der Fall ist).

OID	SCHEMA	Kennung	Langtext
<i>Strassenklasse.A</i>	<i>1</i>	<i>A</i>	<i>Bundesautobahn</i>
<i>Strassenklasse.B</i>	<i>1</i>	<i>B</i>	<i>Bundesstraße</i>
<i>Strassenklasse.L</i>	<i>1</i>	<i>L</i>	<i>Landesstraße</i>
<i>Strassenklasse.S</i>	<i>1</i>	<i>S</i>	<i>Staatsstraße</i>
<i>Strassenklasse.K</i>	<i>1</i>	<i>K</i>	<i>Kreisstraße</i>
<i>Strassenklasse.G</i>	<i>1</i>	<i>G</i>	<i>Gemeindestraße</i>
<i>Strassenklasse.N</i>	<i>1</i>	<i>N</i>	<i>Nicht öffentliche Straße</i>
<i>Strassenklasse.Z</i>	<i>1</i>	<i>Z</i>	<i>Bezirksstraße</i>

Datenbanktabelle "Abschnitt"

Die Tabelle "Abschnitt" dient zur Darstellung von Instanzen der Objektart *Abschnitt*. Neben Spalten für die Attribute der Objektart *Abschnitt* sind auch Spalten für die von der *Objektart Abschnitt_oder_Ast* (und deren Oberklassen) geerbten Attribute vorhanden. Im hier aufgeführten Beispiel sind zwei *Abschnitt*-Instanzen eingetragen.

Wegen der Vielzahl der Attribute erfolgt die Darstellung in drei Teilen. Zur Erhöhung der Übersichtlichkeit werden außerdem einige Spalten für geerbte Attribute weggelassen, was durch drei Punkte (...) verdeutlicht wird.

Bei den Attributen "Betriebsmerkmal", "Seitenarm" und "getrennt_verlaufende_Fahrbahn" handelt es sich um Attribute, deren Datentyp eine Schlüsseltable ist. Hier ist jeweils die OID des entsprechenden Wertekatalogeintrags angegeben.

Dass die Spalte "Liniengeometrie" hier ganz am Ende (und nicht gemäß der Reihenfolge aus dem OKSTRA®-UML-Modell vor der Spalte "DQ_Liniengeometrie") erscheint, hat rein technische Gründe und ist für die Aussage der enthaltenen Daten unerheblich, da die Reihenfolge der Spalten prinzipiell beliebig sein kann.

OID	OKSTRA_ID	...	gueltig_bis	DQ_Liniengeometrie	Laenge	DQ_Laenge	Herkunft_Laenge
<i>2</i>	<i>NULL</i>	<i>...</i>	<i>NULL</i>	<i>NULL</i>	<i>5.918</i>	<i>NULL</i>	<i>NULL</i>
<i>3</i>	<i>NULL</i>	<i>...</i>	<i>NULL</i>	<i>NULL</i>	<i>7.629</i>	<i>NULL</i>	<i>NULL</i>

Betriebsmerkmal	Abschnitts_Astnummer	Abschnitts_Astbezeichnung
<i>Betriebsmerkmal.01</i>	<i>32</i>	<i>Abschnitt 3818042A3918074A, Abs.Nr. 32 auf der A2</i>
<i>Betriebsmerkmal.01</i>	<i>23</i>	<i>Abschnitt 4114036O4214015O, Abs.Nr. 23 auf der A2</i>

Seitenarm	getrennt_verlaufende_Fahrbahn	Abschnittsfolgenummer	Liniengeometrie
<i>Seitenarm.0</i>	<i>Zweig_der_Trennung.0</i>	<i>100009000</i>	<i>BLOB</i>
<i>Seitenarm.0</i>	<i>Zweig_der_Trennung.0</i>	<i>100017000</i>	<i>BLOB</i>



Datenbanktabelle "Betriebsmerkmal"

In die Tabelle "Betriebsmerkmal" werden die Wertekatalogeinträge der gleichnamigen Schlüsseltablelle als Datensätze eingetragen.

OID	SCHEMA	Kennung	Langtext
<i>Betriebsmerkmal.00</i>	<i>1</i>	<i>00</i>	<i>unbekannt</i>
<i>Betriebsmerkmal.01</i>	<i>1</i>	<i>01</i>	<i>durchgehende Strecke</i>
<i>Betriebsmerkmal.02</i>	<i>1</i>	<i>02</i>	<i>Verbindungsrampe</i>

Datenbanktabelle "Seitenarm"

In die Tabelle "Seitenarm" werden die Wertekatalogeinträge der gleichnamigen Schlüsseltablelle als Datensätze eingetragen.

OID	SCHEMA	Kennung	Langtext
<i>Seitenarm.0</i>	<i>1</i>	<i>0</i>	<i>kein Seitenarm</i>
<i>Seitenarm.1</i>	<i>1</i>	<i>1</i>	<i>Seitenarm</i>

Datenbanktabelle "Zweig_der_Trennung"

In die Tabelle "Zweig_der_Trennung" werden die Wertekatalogeinträge der gleichnamigen Schlüsseltablelle als Datensätze eingetragen.

OID	SCHEMA	Kennung	Langtext
<i>Zweig_der_Trennung.0</i>	<i>1</i>	<i>0</i>	<i>keine getrennt verlaufende Fahrbahn</i>
<i>Zweig_der_Trennung.1</i>	<i>1</i>	<i>1</i>	<i>getrennt verlaufende Fahrbahn, Zweig 1 (In Stationierungsrichtung befahren)</i>
<i>Zweig_der_Trennung.2</i>	<i>1</i>	<i>2</i>	<i>getrennt verlaufende Fahrbahn, Zweig 2 (Gegen Stationierungsrichtung befahren)</i>

Datenbanktabelle "zwischenstab"

Die Tabelle "zwischenstab" dient zur Darstellung von Relationen und von Attributbeziehungen, bei denen ein komplexer Datentyp oder eine Union als Datentyp auftritt.

Im vorliegenden Beispiel existieren Relationen zwischen den beiden *Abschnitt*-Instanzen und der *Strasse*-Instanz. Diese Relationen werden beidseitig eingetragen, d.h. zur Darstellung einer Relation dienen je zwei Datensätze, die die beiden Relationsrichtungen repräsentieren. Entsprechend sind im jeweils zweiten Datensatz die Inhalte der Felder "ID" (OID des Quellobjekts) und "RID" (OID des Zielobjekts) sowie die Inhalte der Felder "SOURCE" (Name der Quellobjektart) und "TARGET" (Name der Zielobjektart) gegenüber dem jeweils ersten Datensatz vertauscht.

Im Feld "ROLE" wird der Rollenname der jeweiligen Relation bei der Quellobjektart angegeben. Dies ist im Fall der Quellobjektart *Abschnitt* der Rollenname "zu_Strasse", im Fall der Quellobjektart *Strasse* der Rollenname "hat_Strassenbezugsobjekt" (vgl. UML-Klassendiagramm).



Im Feld "SEQNR" werden die Datensätze zu einer Rollenbezeichnung eines bestimmten Quellobjekts beginnend mit 0 hochgezählt. Da die beiden *Abschnitt*-Instanzen in ihrer Relation "zu_Strasse" nur je eine *Strasse*-Instanz als Relationspartner besitzen, ist in den entsprechenden Datensätzen jeweils eine 0 eingetragen. Umgekehrt hat die *Strasse*-Instanz in ihrer Relation "hat_Strassenbezugsobjekt" zwei Relationspartner, nämlich die beiden *Abschnitt*-Instanzen. Deshalb sind die Einträge im Feld "SEQNR" in diesem Fall 0 und 1. Gäbe es in dieser Relation noch weitere Relationspartner, würden die Einträge im Feld "SEQNR" in den zugehörigen Datensätzen weiter hochgezählt werden und hätten die Werte 2, 3, 4 usw.

Als letzter Datensatz ist in dieser Tabelle noch die Attributbeziehung von der *Strasse*-Instanz zur Instanz des komplexen Datentyps *Strassenbezeichnung* enthalten. Attributbeziehungen werden grundsätzlich nur einseitig eingetragen, und zwar in der Richtung vom Konstrukt, dass das Attribut besitzt, zur Instanz des jeweiligen Datentyps.

Analog dazu werden auch Relationen, die von komplexen Datentypen ausgehen und auf eine Objektart verweisen, nur einseitig eingetragen. Dieser Fall tritt im vorliegenden Beispiel allerdings nicht auf.

OID	ROLE	ID	RID	SEQNR	SOURCE	TARGET
2-2673-0	zu_Strasse	2	2673	0	abschnitt	strasse
3-2673-0	zu_Strasse	3	2673	0	abschnitt	strasse
2673-2-0	hat_Strassenbezugsobjekt	2673	2	0	strasse	abschnitt
2673-3-0	hat_Strassenbezugsobjekt	2673	3	1	strasse	abschnitt
2673-2675-0	hat_Strassenbezeichnung	2673	2675	0	strasse	strassenbezeichnung