

D. Geroe

DER GIP.AT OGD-EXPORT NEUIGKEITEN, TIPPS & TRICKS

- Prozess GIP.AT OGD-Export und Inhalt derzeit
- Neuigkeiten Sommer 2019
- Tipps für den Umgang mit dem Datenformat und Usecases
 - Formatbezogene Beispiele
 - Datenbezogene Beispiele

TEIL 1 – PROZESS GIP OGD- EXPORT UND INHALTE DERZEIT



- Dateneingaben durch GIP-User
- Synchronisation von Landes GIP zu ITS
- Sync-Deadline (alle 2 Monate)

20. April

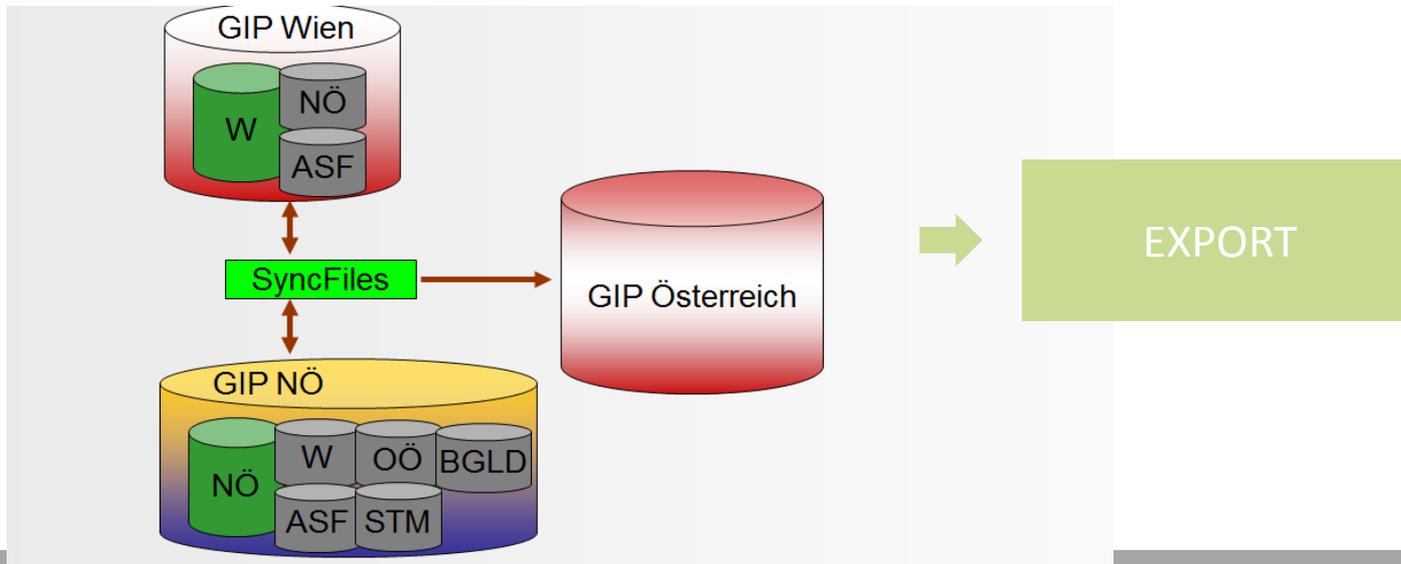
- ITS startet Netzaufbereitung und stellt VAO-Vorabversion bereit
- Basemap-Export
- QM Prüfungsphase (GIP und VAO)
- INSPIRE-Export
- VAO-Finalexport
- Behördenexport
- OGD-Export

12. Juni am Vormittag

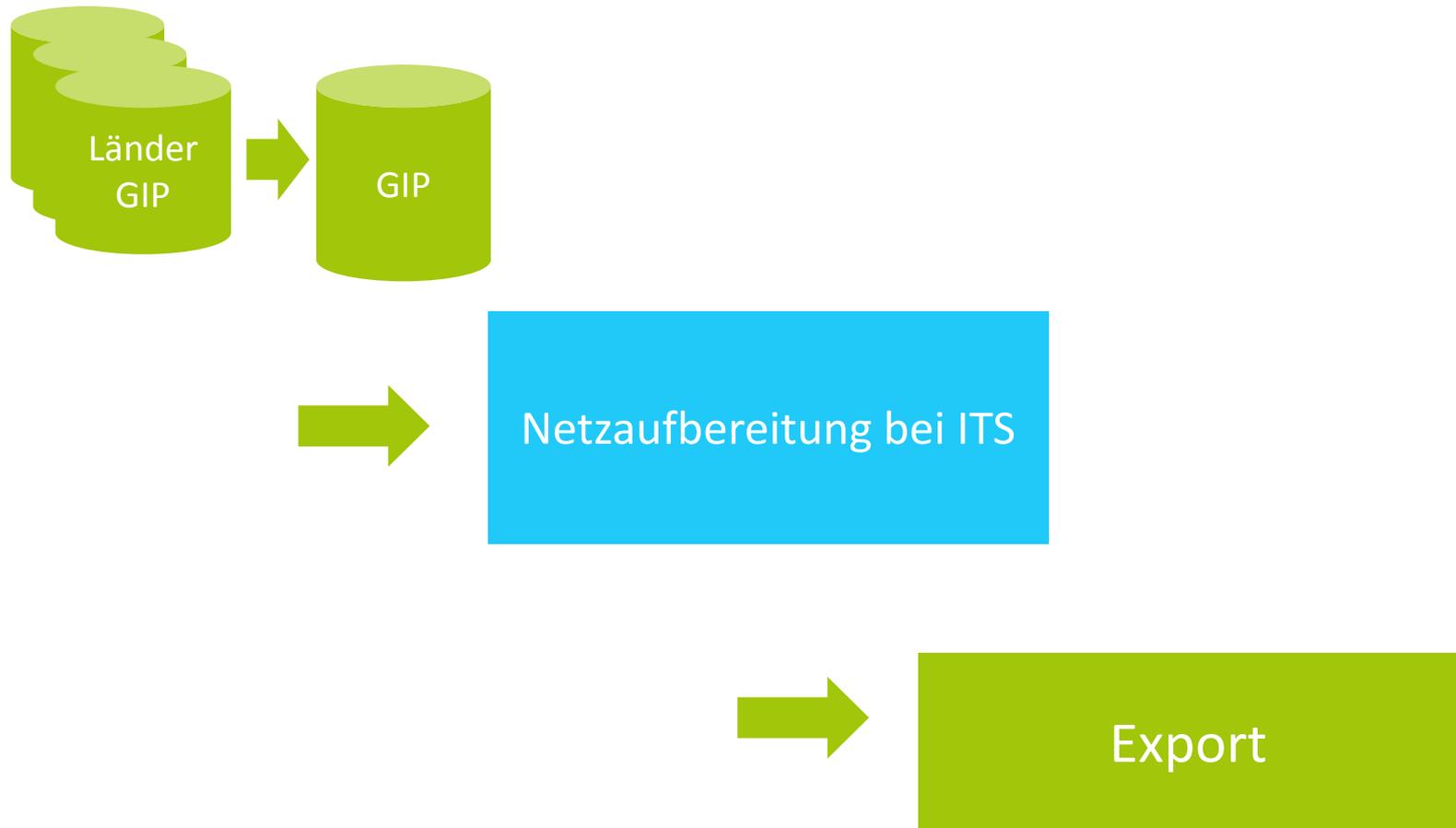
- Sync-Deadline

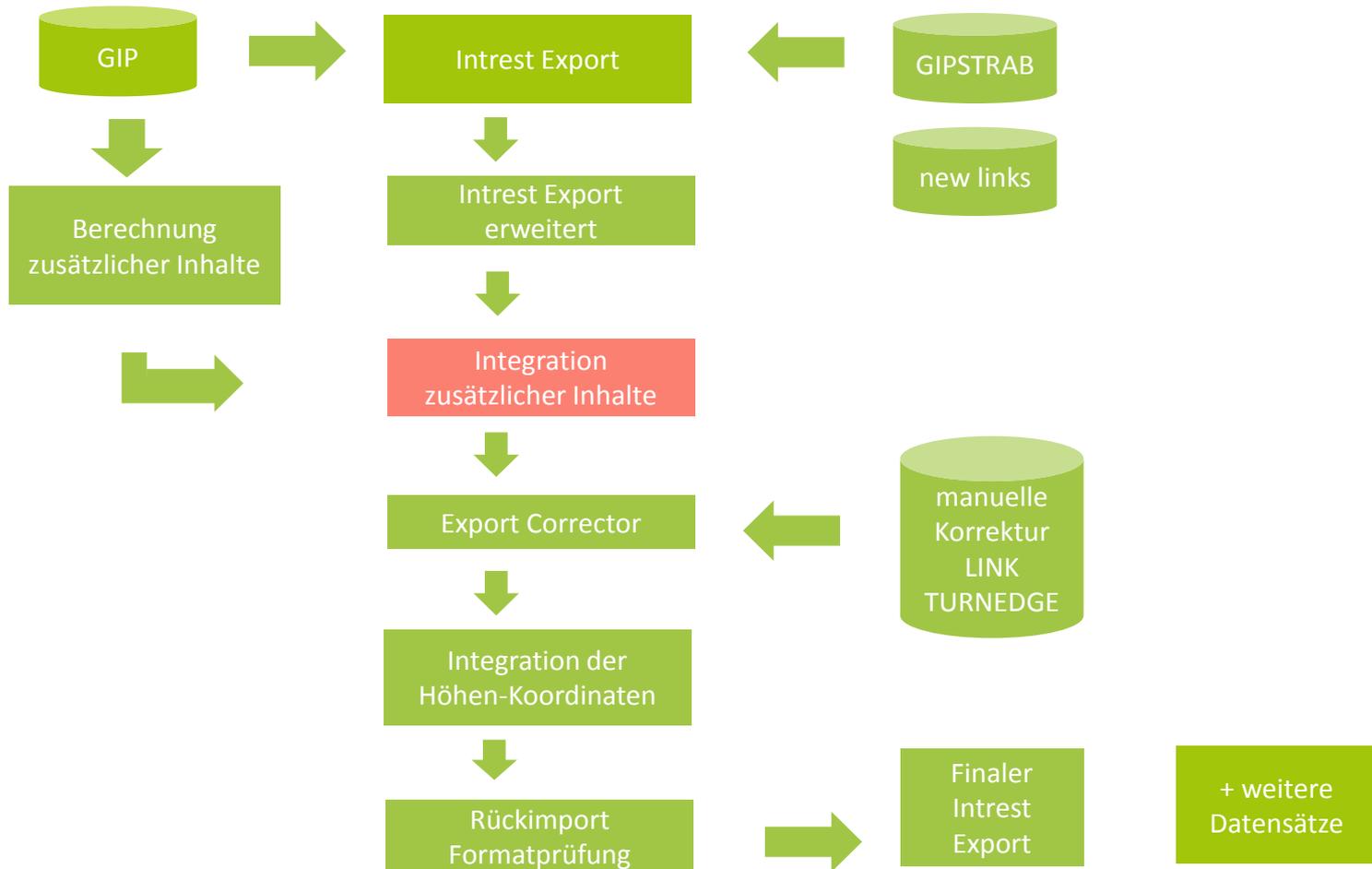
12. Juni am Nachmittag

- 2 Monate lang kann jeder bei sich lokal an den Daten arbeiten, die tagesaktuellen automatisierten Datenlieferungen an den GIP Ö Betrieb werden nur für Visualisierungen und QM Zwecke genutzt, aber nicht exportiert
- Alle 2 Monate gibt es eine Deadline, zu der ein QM geprüfter Stand an den GIP Österreich Betrieb gesendet wird
- Das 2 - Monats - System existiert in 2 Instanzen: GIPAT / GIPOGD

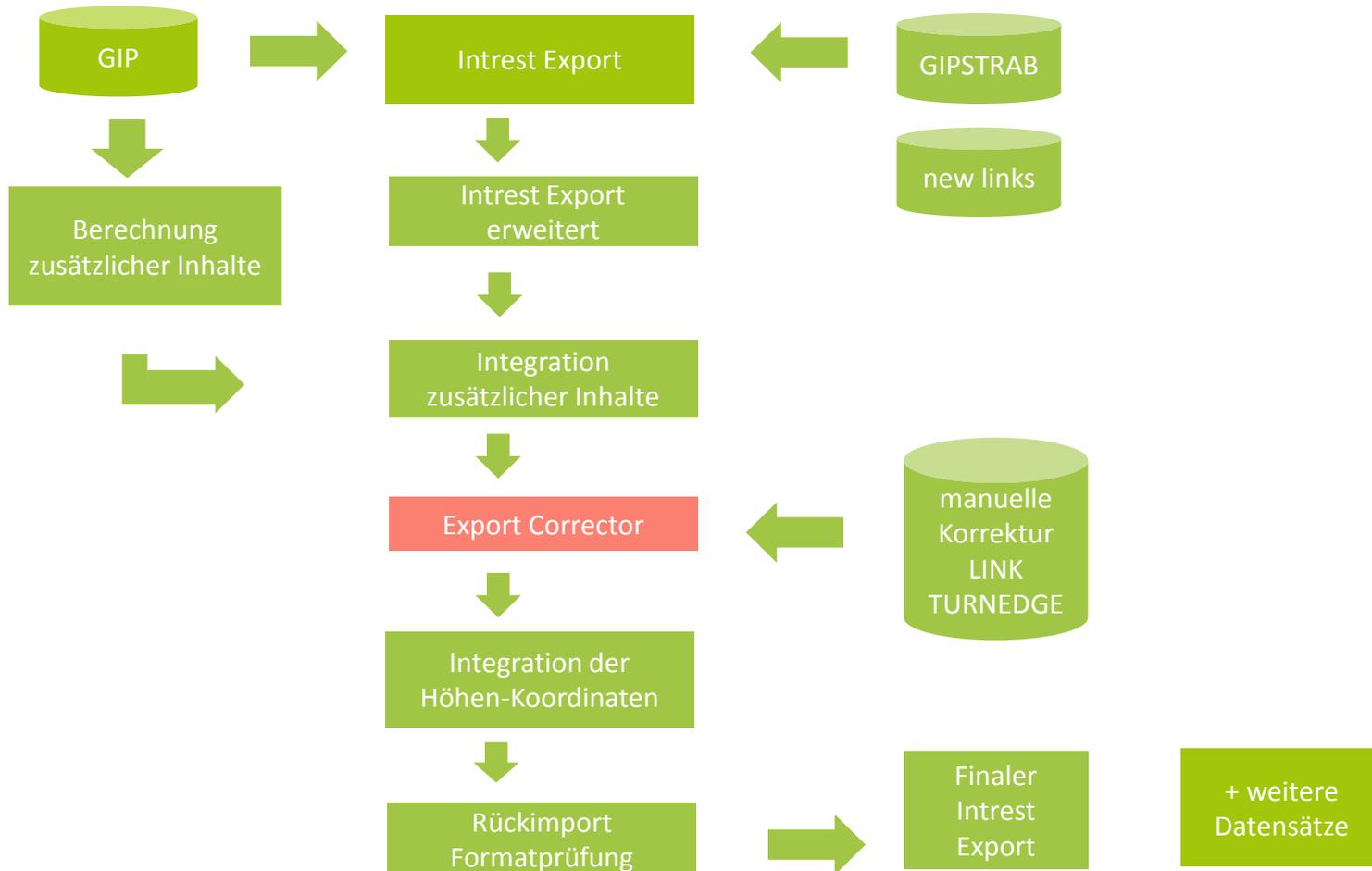


- Die GIP Software enthält ein Exportmodul
- Dessen Output im „IDF“ Format (.txt) wird von ITS ergänzt, vervollständigt und korrigiert
- IDF = **I**NTREST **d**ata **f**ormat
- INTREST: **I**ntermodal **r**eferencing **s**ystem for **t**raffic related data
= Sammlung von CSV Tabellen in einem Textfile mit integrierter Definition des Datenformates
- Zusätzlich werden GIS-Datensätze direkt aus der GIP exportiert



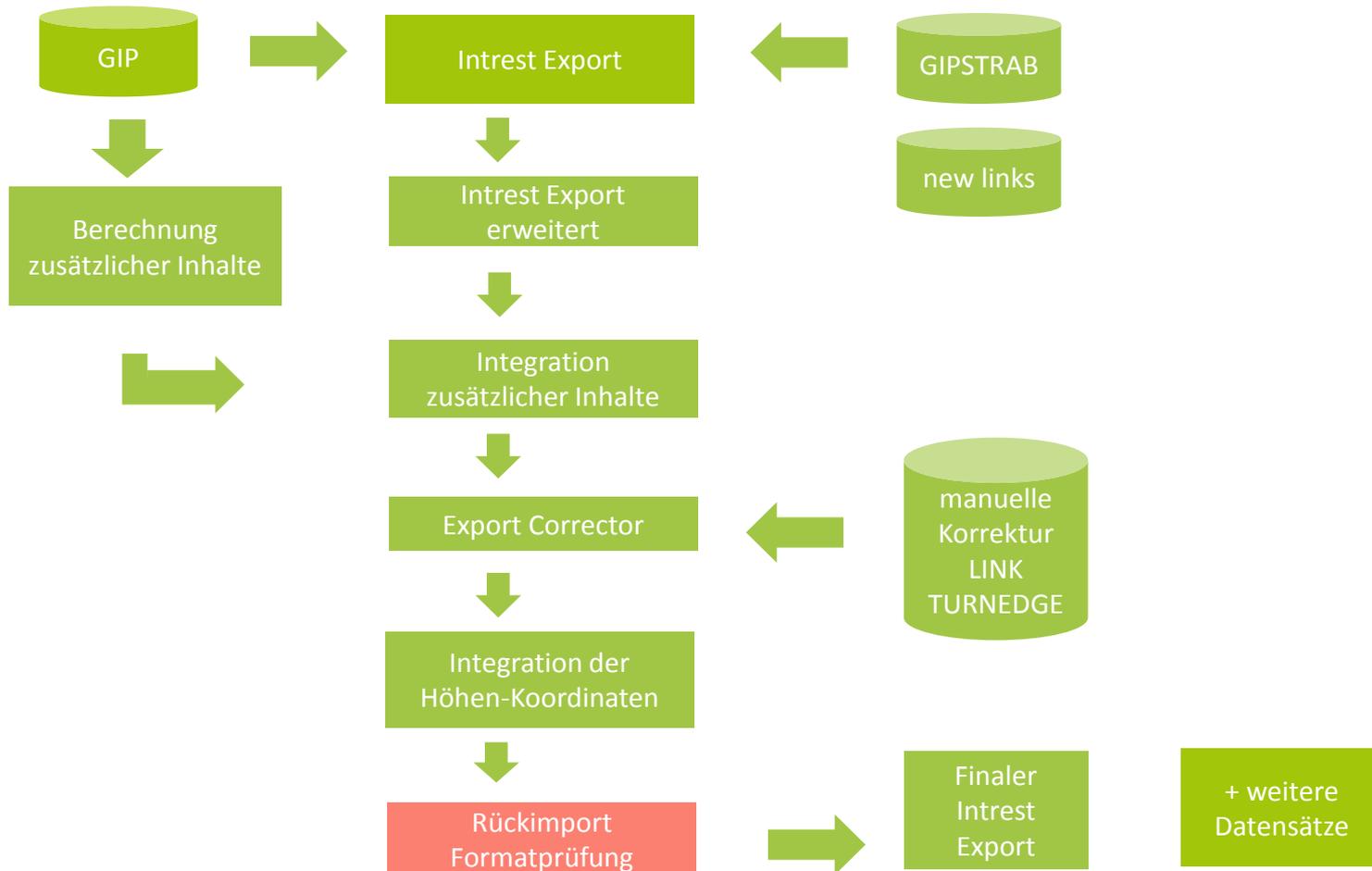


- neue Links, Attributeänderungen und Turnedges
- Ortsgebieteflag (URBAN)
- Belag (SURFACE/BIKEENVIRONMENT)
- Sustainer
- Regionalcode
- Datenbankverbinder
- Elektrifizierung
- Radattribute
- Touristische Raddaten
- Kurzname
- Richtungsfahrbahn
- Beschränkungen aus der Wirkungsableitung (Tonnagen etc.)



Beispiele für Aufbereitungsschritte im Korrekturteil

- Durchschnittsgeschwindigkeit gemäß Regelwerk setzen: 2.203.185 Datensätze bearbeitet
- Spurenanzahl gemäß Regelwerk setzen: 484.965 Datensätze bearbeitet
- Widersprüchliche Attributierung auflösen: 78.585 Datensätze bearbeitet
- Beispiel inhaltlicher Check - Fußgängereinbahnen: 2.267 Datensätze bearbeitet
- isolierte Links für Router beheben: 318 Datensätze bearbeitet

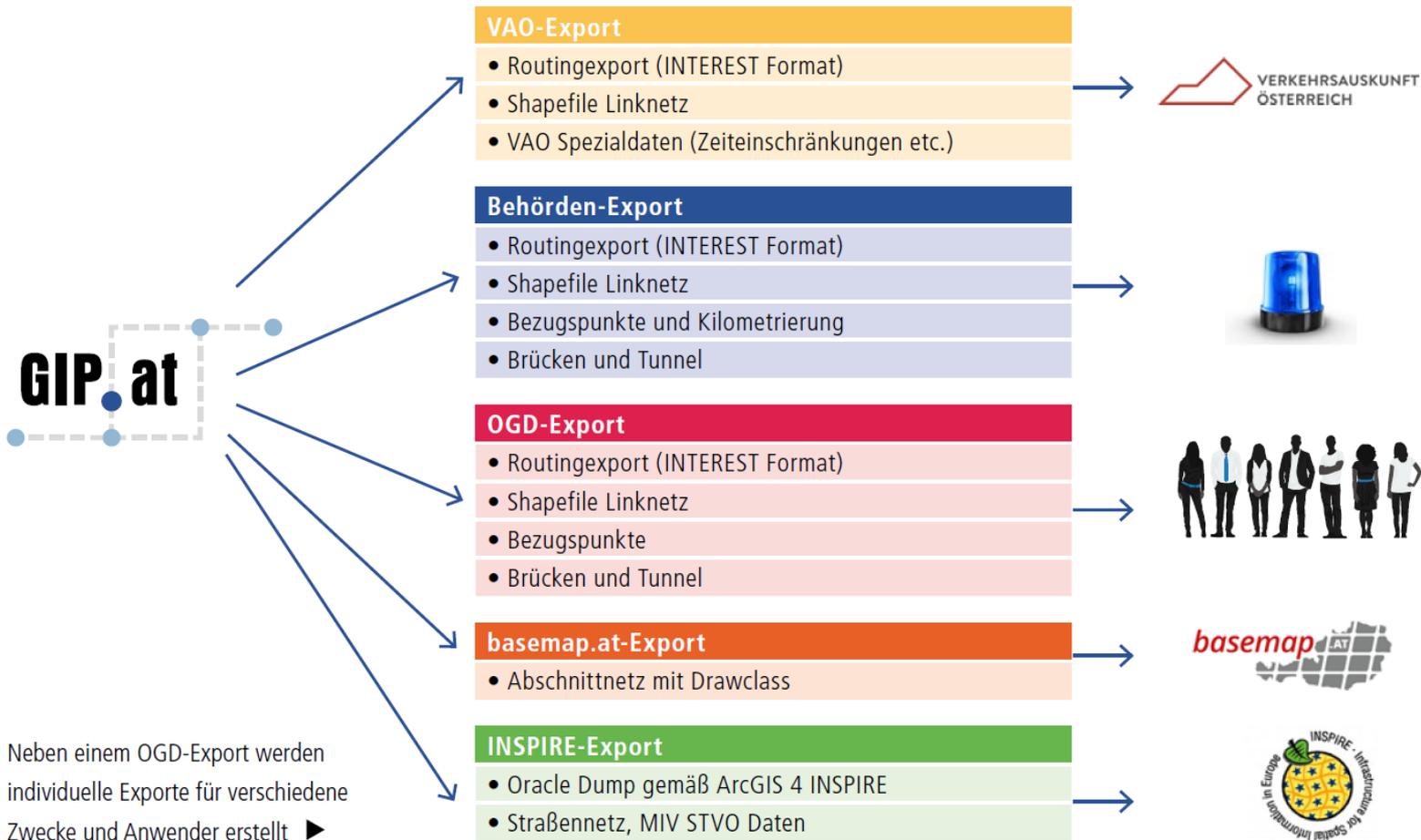


- Sind alle **Tabellen vorhanden** und mit **Daten befüllt**?
- **Korrekte Anzahl an Datensätzen** in jeder Tabelle?
- **Erfüllen** alle Datensätze die **Formatspezifikation**?
- Gibt es **ungültige Zeichen** in den Tabellen?
- Gibt es die **korrekte Anzahl Leerzeilen** zwischen den Tabellen?
- Ist das **Abschlusszeichen korrekt** gesetzt?
- Passt die **Anzahl Anführungszeichen**?
- Sind **Trennzeichen** und **Stringcharacter** korrekt **berücksichtigt**?
- Lässt sich der Datensatz **fehlerfrei in Oracle importieren**?
- Lässt sich der Datensatz **fehlerfrei in das Testroutingsystem importieren**?

Wie sieht das GIP-OGD-Export-Format jetzt aus?

- Die GIP enthält 318 Tabellen in ORACLE
 - Das wäre kein schönes Exportformat
 - Nur wenige davon sind relevant
- Der Export erzeugt aus dem komplexen GIP-Datenmodell konkrete Files in einfach interpretierbaren Formaten, die genutzt werden können
- Je nach Anwendungszweck werden unterschiedliche Inhalte bereitgestellt
- Beispiel: Routing, Kartographie, Blaulichtorganisationen, einfacher Zugang,...

ITS erzeugt als GIP Österreich Betrieb regulär alle 2 Monate folgende Exporte:



- Linknetzshapefile = „einfacher Zugang“ zu den GIP Daten
 - Shapefile
 - Kantengeometrie für Straßen, Schiene, Fähren und Wege
 - Funktionale Einteilung der Straße
 - Befahrbarkeiten und Beschränkungen
 - Routinggeschwindigkeit
 - Baustatus
- Routingexport = komplex, für Routinganwendungen notwendig
 - Namen
 - Brücken/Tunnel und ihre Netzreferenzen
 - Abbiegerelationen
 - Zusatzattribute für Fuß und Rad
- Bezugspunkte Shapefile (= „Kilometertaferl“) = Um Straßen-KM zu finden
- Brunnel Shapefile = **Brücken** und **Tunnel** (+ Netzreferenz-Tabelle Link2Brunnel)

TEIL 2 – NEUERUNGEN IM OGD- EXPORT AB GIP RELEASE JUNI

Technologisches Problem:

Shapefile für GIS Datensätze nicht geeignet, da diese zu groß werden
Spaltennamen >10 Zeichen bei Shapes unmöglich

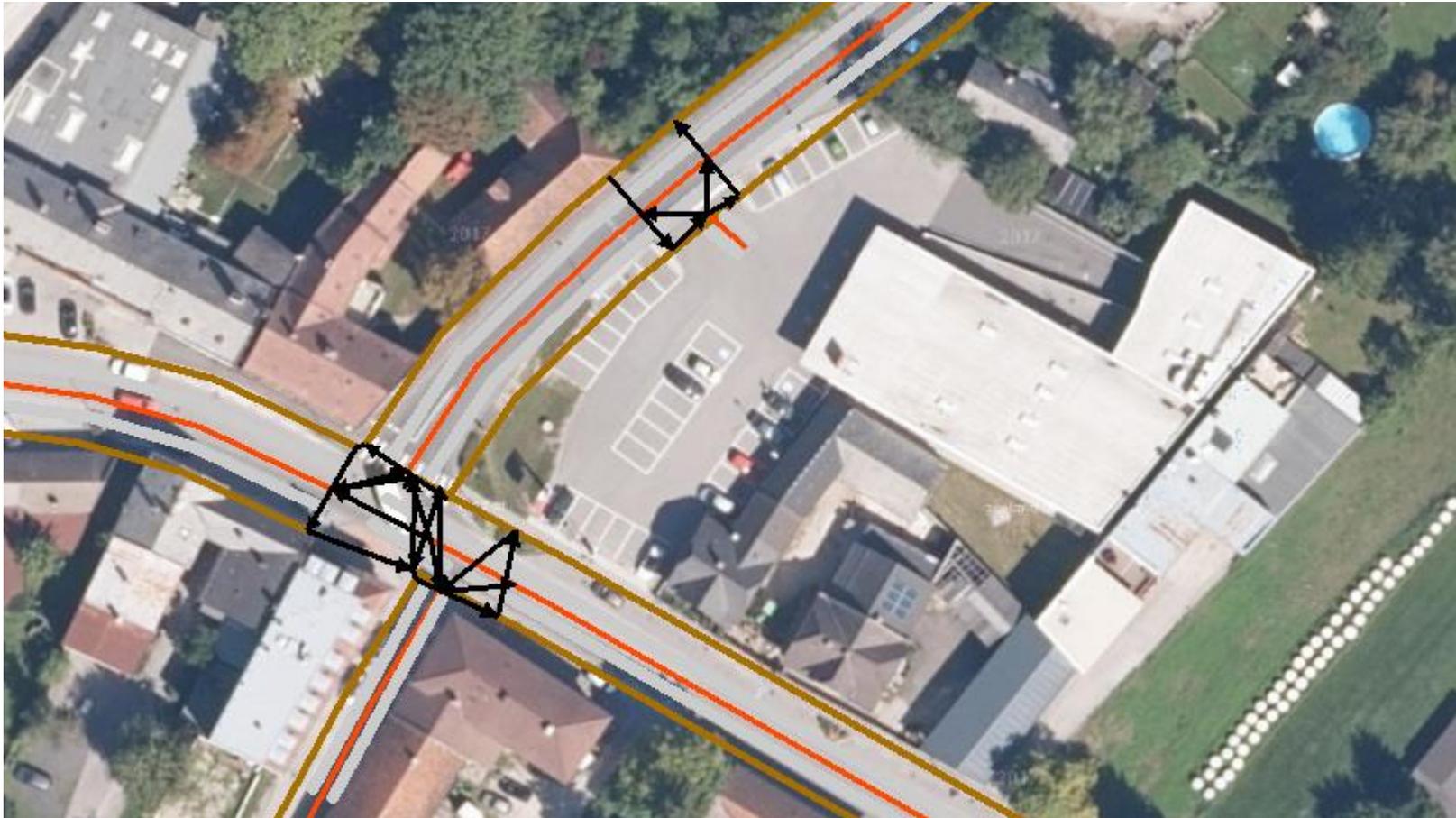
→ Umstieg für die GIS Datensätze auf GEOPACKAGE

Inhaltlich:

- Einige Anforderungen aus diversen Projekten oder von Abnehmern der Exporte
- Erweiterung der Datensätze

- A „Routingexport“ → IDF Files
- B „GIP Network“ → Geopackage mit GIP Basisnetz
- C „GIP Reference“ → Geopackage mit auf die GIP referenzierten Objekten
- D Lookuptabellen
- E alte Datensätze (um Umstellungszeit zu geben)
Bis Dezember

- **Edges** = Abschnittsnetz
→ nicht routingfähig, für Referenzierung ideal
 - **Nodes** = Knoten
→ bisher nur implizit, erleichtert Referenzierungen
 - **Links** = das bestehende Linknetz
 - **LinearUse und TurnUse** = Nutzungstreifen und Abbiegeerlaubnisse mit den tatsächlichen Geometrien
- **Enthält das GIP Netz für (simples) Routing und Referenzierung**
- **Erleichtert die Interpretation der GIP und verhindert systembedingte Fehler**



GIP Referenzen enthalten folgende Datensätze:

- Bezugspunkte (wie bisher)
- Brunnels (wie bisher)
- Radrouten (neu)
- Geonamen (ÖBB Bahnhöfe)

→ Fasst netzreferenzierte Objekte auf der GIP zusammen
→ Ist für die Zukunft leicht erweiterbar

- Punktueller Datensatz
- Input für die Kilometrierung
- „Kilometertafeln“
- Ermöglicht ungefähres Auffinden von Straßen/Bahn-Kilometern
- Verjoinen auf Edge – Datensatz möglich
- Verjoinen auf Link-Edge-Route Datensatz möglich

→ **Bezugssystem zum Auffinden von km-bezogenen Örtlichkeiten auf der GIP**

- BRUNNEL = **Br**ücken und **Tun**nel
- Linearer Datensatz für (relevante) Kunstbauwerke auf dem GIP Netz
- Brücken, Tunnel, Galerien, Unterführungen, Überführungen, Wannen
- Nicht enthalten: Viehsperren, Durchlässe usw.
- Derzeit nicht flächendeckend, werden laufend erweitert

→ **Für kartographische Darstellungen oder weiterführende Berechnungen im Routing (Z Koordinaten,...) wichtig**

- Geoname = **Name** mit Attributen und **Geo**metrie
- Keine direkte GIP – Netzreferenz aber in GIP verortet
- Erste Verwendung für ÖBB-Bahnhöfe
- Künftig erweiterbar für neue Datensätze
- Wichtigste Attribute:
 - Nametext
 - Namecategory
 - Zusatzattribute
 - Geometrie

→ **Datensatz für GIP Objekte im Export, die keinen direkten Netzbezug besitzen**

- Lineare Verortungsobjekte, die den Verlauf von Radrouten kennzeichnen
- Initial 4 Typen
 - Beschilderte Radroute
 - Mountainbikestrecke
 - Singletrail
 - Rennradroute
- Attribute: Geometrie, Name, Netzreferenz, Schwierigkeitsgrad (nur MTB/Sitr), Beschilderung, ...
- In der ersten Version Tirol und Vorarlberg, danach laufend erweitert

→ **Datensatz, der als Erweiterung des Routingexportes nicht nur die Radroutinginfo am Nutzungstreifen, sondern Daten über die Radroute als solche beinhaltet**

Layers

- GIEXPORT.BIKEROUTES_VAO
 - < alle anderen Werte >
 - REFTYPE_N
 - Mountainbike Strecke
 - Radroute**
 - Radroute VBG
 - Rennradroute
 - Singletrail

Identifizieren

Identifizieren aus: <Oberster Layer>

GIEXPORT.BIKEROUTES_VAO
Zillertalradweg

Position: -336 024,612 252 687,311 Meter

Feld	Wert
OBJECTID	800002087177
REFTYPE	5019
REFTYPE_N	Radroute
EXTORGCODE	700
EXTERNALID	13
ROUTETYPE	Beschilderte Radroute
SUSTAINER	<null>
SYSCODE	<null>
OWNER_ID	8
OBJECTNAME	Zillertalradweg
GNAME_ID	11721574695
GNAME_CAT	Radroute
GNAME_TEXT	Zillertalradweg

TEIL 3 – FORMATBEZOGENE TIPPS UND USECASES

Was ist ein Geopackage?

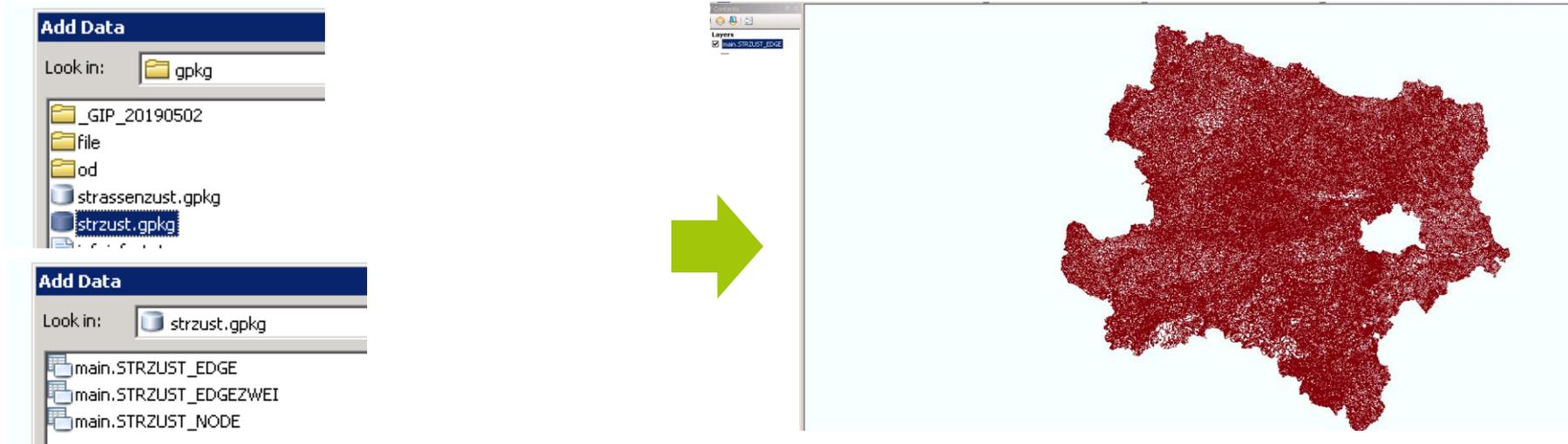
<https://www.geopackage.org/>

<http://www.geopackage.org/guidance/getting-started.html>

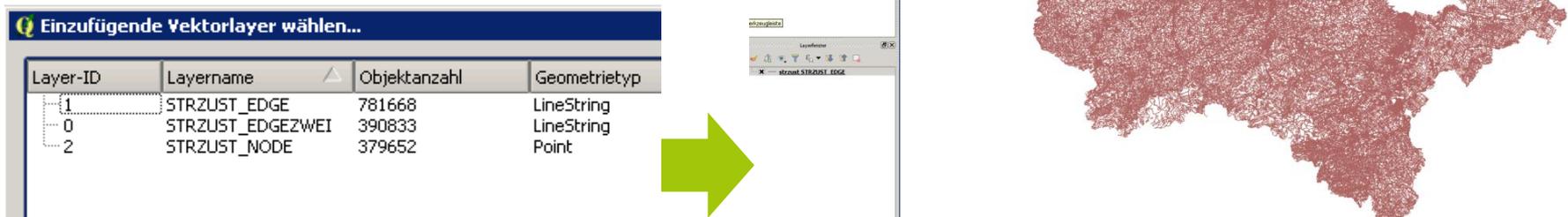
- SQLite Datenbank mit Endung .gpkg
- Offener Standard
- User-defined Tabellen und Metadaten
- Mehrere Tabellen möglich
- Tabellen mit Geometrie und ohne („Attributes“) zulässig
- Mehr als 10 Zeichen für Spaltennamen zulässig
- Mehr als 2 GB Daten zulässig

Geopackage öffnen:

- **ArcGIS: Add Data**

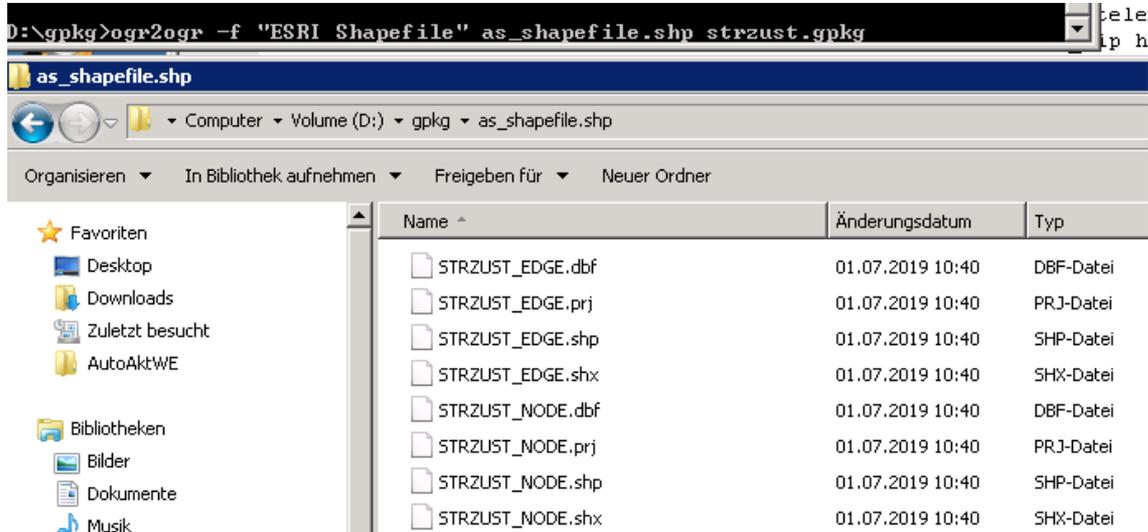


- **QGIS: Drag and Drop**



Skript für Umwandeln in ein Shape – mit ogr2ogr (zum Beispiel)

```
ogr2ogr -f "ESRI Shapefile" shapefilename.shp geopackagefilename.gpkg
```



→ Jeder Layer wird ein Shapefile

Skript für Import in Postgis DB

```
ogr2ogr -overwrite -f PostgreSQL PG:"user=USERNAME dbname=DBNAME  
host=localhost port=5432" D:\folder\filename.gpkg
```

tbl;Mitarbeiter

Jetzt kommt die Tabelle „Mitarbeiter“

atr;ID;NAME

Diese hat 2 Spalten, ID und Name

frm;decimal(5);string(30)

ID ist eine Zahl, Name ist ein Text

num;2

2 Datensätze stehen da drin

rec;1;“Hans Müller“

Das ist der erste Datensatz

rec;2;“Liese Maier“

Das ist der zweite...

end;2

Und schon ist Schluss mit der Tabelle

Leerzeile – eof oder neue Tabelle?

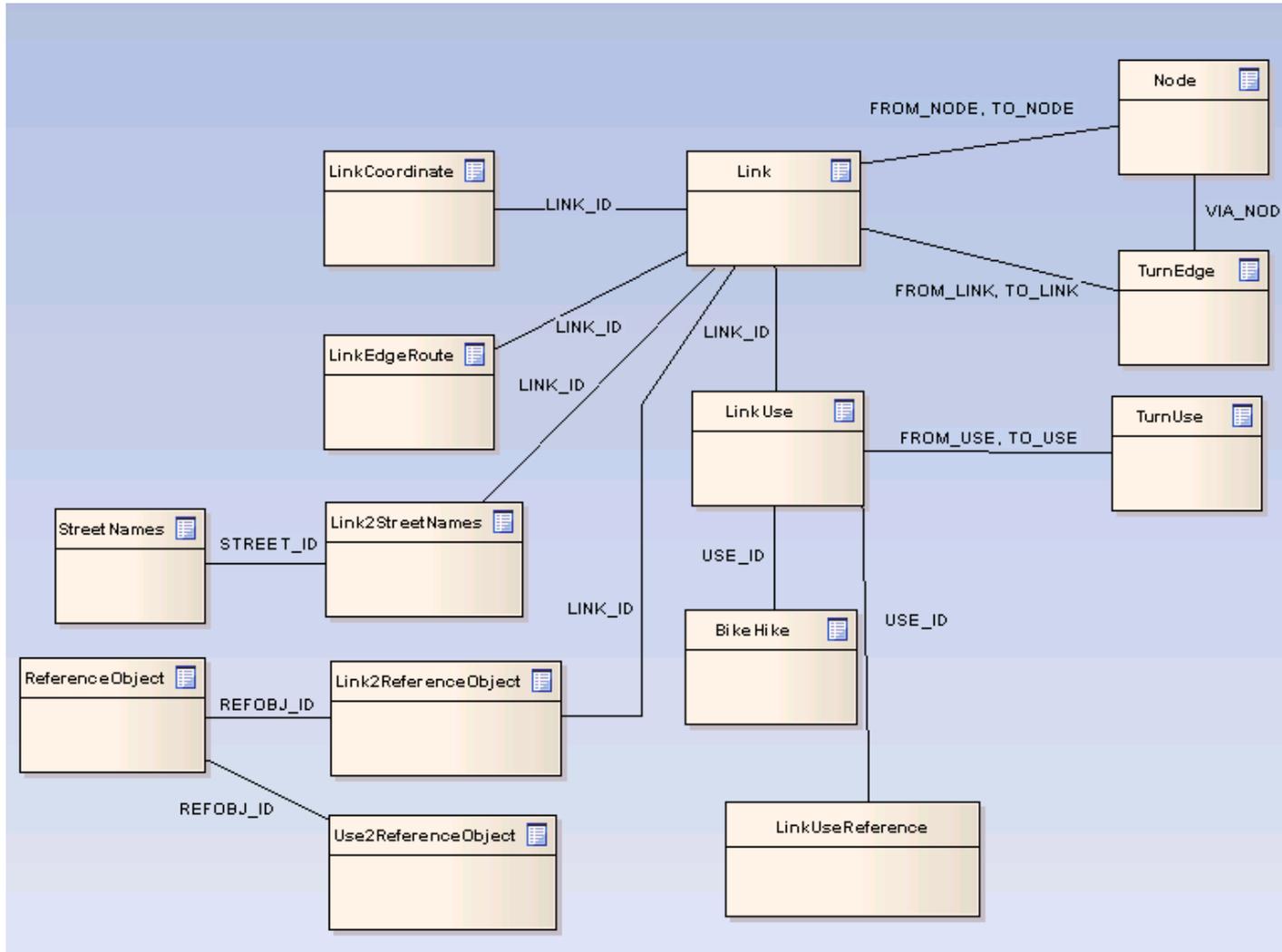
tbl;Besprechungsräume

There it is... neue Tabelle

atr;ID;Name;Stockwerk

...

- NODE = Knoten
- LINK = Links
- LINKCOORDINATE= Zwischenpunkte der Links
- LINKUSE = Nutzungstreifen
- BIKEHIKE = Rad/Fuß Spezialdaten
- STREETNAMES = Straßennamen
- LINK2STREETNAMES = Welcher Link hat welchen Namen
- LINK2BIKEHIKE = Aggregation der Radinfos auf den LINK
- LINKEDGEROUTE = Zusammenhang Link – Edge – Subroute – Route
- REFERENCEOBJECT = Verortungsobjekte
- LINK2REFERENCEOBJECT = Netzreferenz auf Linkebene
- USE2REFERENCEOBJECT = Netzreferenz auf Use-Ebene
- LINKUSEREFERENCE = Detailreferenzen auf Use-Ebene, initial der Belag



Wie mit dem GIP Österreich Export arbeiten?

- **Variante A:** Direkt die Shapefiles und/oder die CSV Files in die Anwendung einlesen (dabei auf die benötigten Inhalte filtern)
Anwendungsfall ITS: Routingtool

```
31 unlimitieren: on
32
33 #Einstellungen auf Subnetze
34 exclude_subnet:408
35 exclude_subnet:409
```

ormal text file

- **Variante B:** Export in eine lokale Datenbank laden und von dort aus die Fremdanwendung bespielen
- Für GIS Datensätze räumlicher oder attributiver Filter einfach
- Wie bekommt man den IDF Export effizient in die lokale Datenbank?

Massenimport

- Datenbanksysteme wie Oracle/Postgres haben optimierte Massentools

Zeilenweiser Import

- Verwendung von Einzelaktionen „Insert Into“ ist auch möglich
- Führt zum selben Ziel
- Dauert aber viel länger
- Beispiel Link: ca. 2 Minuten mit SQL Loader, ca. 2 Stunden mit zeilenweisem Import

→ Falls der zeilenweise Import nicht notwendig ist, wird die Verwendung eines Massentools empfohlen

Import des INTREST Exportes nach ORACLE

- Tool „SQL Loader“ ist Teil des Oracle Standard Clients
- Control File gibt an, was importiert werden soll
 - Wie viele Zeilen sollen übersprungen werden
 - Codierung angeben
 - File, das importiert werden soll
 - Soll bestehende Tabelle getruncated werden
 - Wie heißt die Tabelle in die übernommen werden soll
 - Was ist das Trennzeichen
 - Wie werden Strings abgebildet (darin zählt Trennzeichen NICHT!)
 - Umgang mit Leerspalten
 - Welche Spalten werden wie befüllt

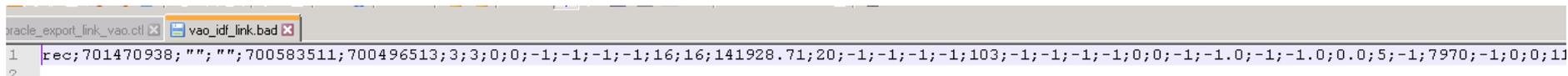
```
oracle_export_link_vao.ctf
1  OPTIONS (SKIP=4)
2  load data
3  characterset WE8MSWIN1252
4  infile '.\oracle_gip_export\vao\Link.txt'
5  truncate
6  into table IDF_LINK_VAO
7  fields terminated by ";" OPTIONALLY ENCLOSED BY '"'
8  TRAILING NULLCOLS
9  (
10     REC,
11     LINK_ID,
12     NAME1,
13     NAME2,
14     FROM_NODE,
15     TO_NODE,
```

Import des INTREST Exportes nach ORACLE

- Beim Aufruf Control File und Pfad für Log und Bad Files geben

```
set NLS_NUMERIC_CHARACTERS=.,
sqlldr DATENBANKSCHEMANAME/PASSWORD@INSTANCE control=.\oracle_export_link_vao.ct1 log=.\log_und_bad\vao_idf_link bad=.\log_und_bad\vao_idf_link
pause
```

- Importer schreibt ein Logfile
- Im BAD File stehen die Datensätze, die nicht importiert werden können
- Wenn zu viele Datensätze nicht passen, wird der Import beendet



```
oracle_export_link_vao.ct1 vao_idf_link.bad x
1 rec;701470938;"";";700583511;700496513;3;3;0;0;-1;-1;-1;-1;16;16;141928.71;20;-1;-1;-1;-1;103;-1;-1;-1;-1;0;0;-1;-1.0;-1;-1.0;0.0;5;-1;7970;-1;0;0;11
2
```

```
Tabelle IDF_LINK_VAO:
```

```
2130426 Zeilen erfolgreich geladen.
```

```
1 Zeile aufgrund von Datenfehlern nicht geladen.
```

```
0 Zeilen nicht geladen, da alle WHEN-Klauseln nicht erfolgreich waren.
```

```
0 Zeilen nicht geladen, da alle Felder null waren.
```

Import des INTREST Exportes nach POSTGRES

- COPY FROM → nicht so flexibel wie der SQL Loader
- aber noch schneller
- Einschränkungen:
 - Spaltennamen können nicht anders genannt werden als im CSV File
 - Es kann nur 1 Zeile übersprungen werden, was bei IDF Files ein Problem darstellt, da 4 Header vorliegen
 - Diese Zeilen müssen mit Hilfe von Betriebssystem-Tools oder mit einem kleinen Skript vorab weggefiltert werden
- Import für ganz Ö ca. 7 Minuten

Tutorial:

„Wie kann ich den Export filtern und einschränken?“

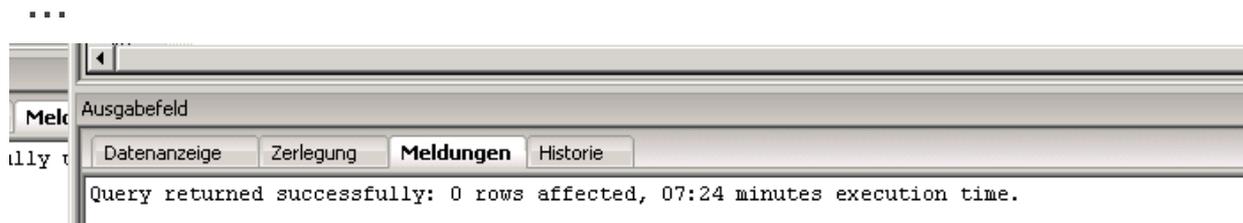
- Schritt 1: Österreich IDF Export in Temp Tabellen einspielen
- Schritt 2: Filterung der Tabelle LINK
- Schritt 3: Abhängige Tabellen gefiltert erzeugen
- Schritt 4: Österrichtabellen wieder leeren
- Schritt 5: Weitere Datensätze auch filtern

Schritt 1: GIP Ö Export einspielen (temp)

Dauer: 7 min Postgres, 36 min Oracle

```
...  
cd D:\Massenimport\pg  
CALL D:\Massenimport\pg\trimtable.py D:\Massenimport\oracle_gip_export\vao\Node.txt D:\Massenimport\pg\oracle_gip_export\vao\Node.txt  
CALL D:\Massenimport\pg\trimtable.py D:\Massenimport\oracle_gip_export\vao\Link.txt D:\Massenimport\pg\oracle_gip_export\vao\Link.txt  
...
```

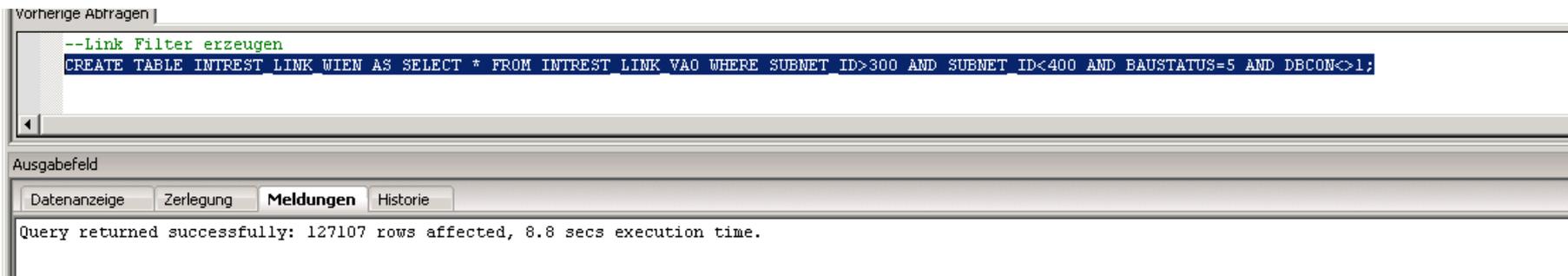
```
...  
truncate table public.intrest_link_vao;  
COPY public.intrest_link_vao FROM 'D:\Massenimport\pg\oracle_gip_export\vao\Link.txt'  
WITH DELIMITER ';' CSV HEADER QUOTE '"' ENCODING 'WIN1252';  
truncate table public.intrest_linkcoordinate_vao;  
COPY public.intrest_linkcoordinate_vao FROM 'D:\Massenimport\pg\oracle_gip_export\vao\LinkCoordinate.txt'  
WITH DELIMITER ';' CSV HEADER QUOTE '"' ENCODING 'WIN1252';  
...
```



Schritt 2: Filter für LINK definieren und lokale Tabelle für LINK erstellen

Beispiel Filter: Subnetz, Regionalcode, FRC, Befahrbarkeiten??

Dauer: 8,8 Sekunden



The screenshot shows a database query execution window. At the top, it says "Vorherige Abfragen". Below that, a SQL command is entered in a text area: `--Link Filter erzeugen` followed by `CREATE TABLE INTREST LINK WIEN AS SELECT * FROM INTREST LINK VAO WHERE SUBNET_ID>300 AND SUBNET_ID<400 AND BAUSTATUS=5 AND DBCON<>1;`. Below the text area is a scroll bar. Underneath, there is a section labeled "Ausgabefeld" with four tabs: "Datenanzeige", "Zerlegung", "Meldungen", and "Historie". The "Meldungen" tab is selected, and it displays the message: "Query returned successfully: 127107 rows affected, 8.8 secs execution time."

```
--Link Filter erzeugen
CREATE TABLE INTREST LINK WIEN AS SELECT * FROM INTREST LINK VAO WHERE SUBNET_ID>300 AND SUBNET_ID<400 AND BAUSTATUS=5 AND DBCON<>1;
```

Ausgabefeld

Datenanzeige Zerlegung **Meldungen** Historie

Query returned successfully: 127107 rows affected, 8.8 secs execution time.

Schritt 3: Für abhängige Tabellen über INNER JOIN eingeschränkte Variante erzeugen

Dauer: 1,5 Minuten

Beispiel LinkUse: (Dauer 5,4 sek), Link Coordinate (33 sek)

```
13 UN idd.use_id=li.use_id;
14
15 --Tabelle LINKCOORDINATE filtern
16 CREATE TABLE INTREST_LINK COORDINATE WIEN AS
17 SELECT DISTINCT v.* FROM
18 (SELECT * FROM INTREST_LINK_WIEN) li
19 INNER JOIN INTREST_LINKCOORDINATE VAO v
20 ON v.link_id=li.link_id;
21
22
23
24
25
26
27
28
29
30
```

gabefeld

atenanzeige Zerlegung **Meldungen** Historie

ery returned successfully: 223523 rows affected, 33.1 secs execution time.

Schritt 4: Temp Tabellen leeren

damit nicht ganz Ö dauerhaft im Speicher
(Dauer: 1,1 sek)

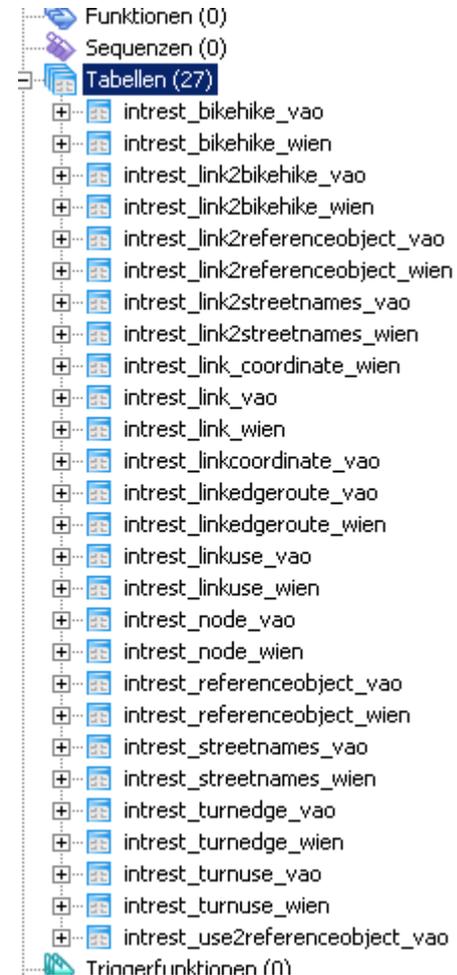
Schritt 5 optional?:

Über JOINS oder über Selektionen in den Datensätzen selbst auch die weiteren Files einschränken

GESAMTER ABLAUF für Import und Filterung (Test mit WIEN)

Durchlaufzeit ca. 8,5 Minuten POSTGRES

Durchlaufzeit ca. 40 Minuten ORACLE



Geometrie aus IDF erzeugen in PostGIS

- Mit ST_MAKELINE können Linien aus Punkten erzeugt werden
- Oracle: Zusätzlich Eintrag in SDO_METADATA sowie Funktion SDO_UTIL.FOM_WKTGEOMETRY statt ST_GEOFROMTEXT

```
1
2  --Tabelle NODEGEOMETRY
3  --Schritt 1: INTREST Tabelle nehmen
4  CREATE TABLE INTREST_NODE_WIEN_GEOM AS
5  SELECT * FROM INTREST_NODE_WIEN;
6
7  --Eindeutige ID als Schlüssel einfügen
8  ALTER TABLE INTREST_NODE_WIEN_GEOM ADD COLUMN gid serial PRIMARY KEY;
9
10 --Geomspalte
11 ALTER TABLE INTREST_NODE_WIEN_GEOM ADD COLUMN geom geometry(POINT,4326);
12
13 --Punkte machen
14 UPDATE INTREST_NODE_WIEN_GEOM SET geom=ST_GEOFROMWKT('SRID=4326;POINT(' || x || ' ' || y || ')');
15
16 --Index machen
17 CREATE INDEX idf_node_wien ON INTREST_NODE_WIEN_GEOM USING GIST(geom);
18
```

→ Output:

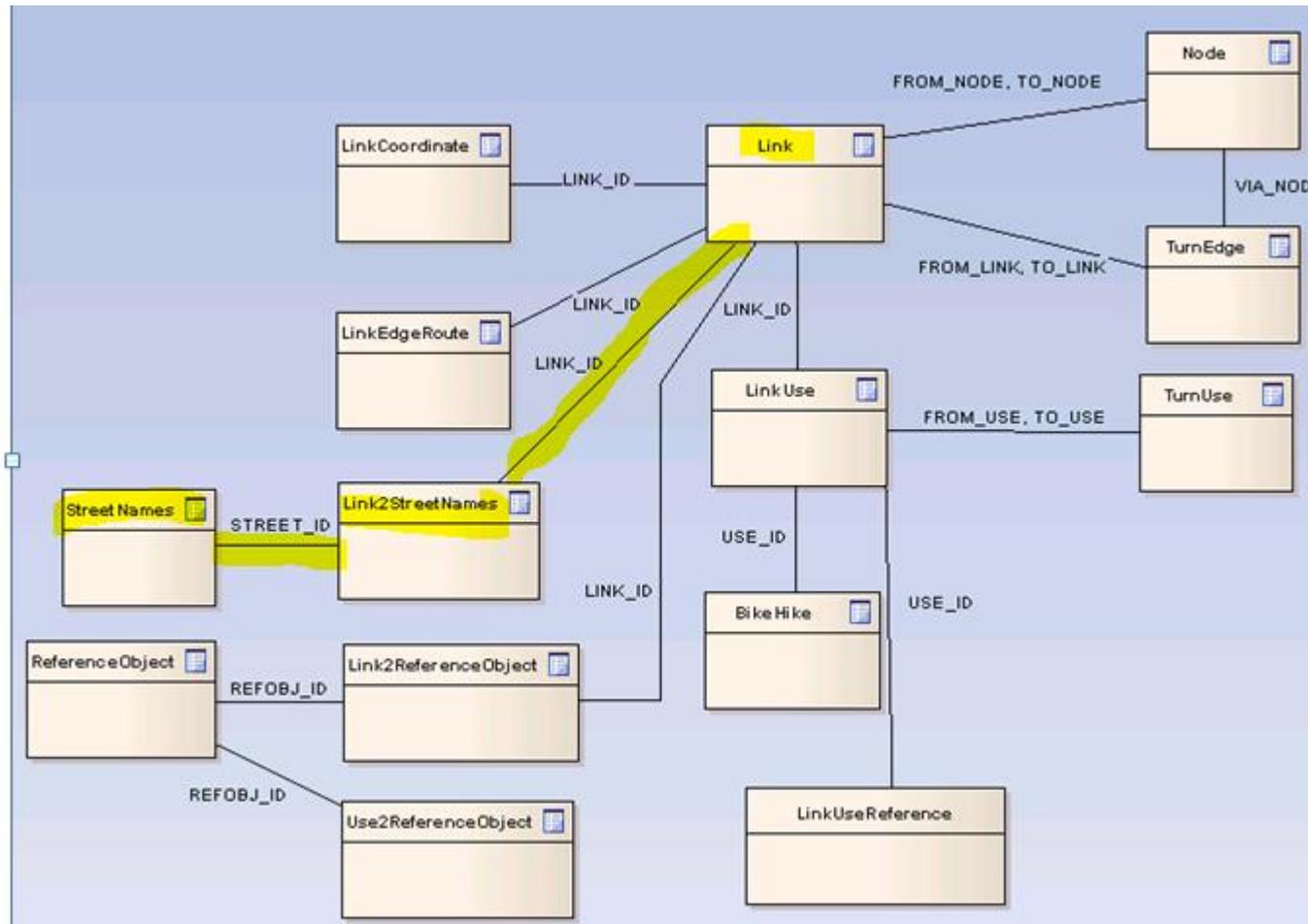
The screenshot displays a GIS application interface. On the left, a 'Table Of Contents' window shows a single layer named 'giparchivpg.public.intrest_node_wien_geom'. The central map area shows a dense network of blue lines representing a city's street network. On the right, a 'Catalog' window lists various data layers, including 'giparchivpg.public.intrest_node_wien_geom', which is highlighted. The catalog also shows a folder named 'giparchivpg.public.intrest_node_wien_geom' containing several sub-layers.

TEIL 4 – DATENBEZOGENE TIPPS UND USECASES

- Welche Namen sind einem Link zugewiesen?
- Welche Abbiegerelationen existieren an einem Knoten für PKW?
- Ich möchte auf die GIP Daten referenzieren
- Welche Eigenschaften hat der Link für Radrouting?
- Welche Links sind eine Brücke oder ein Tunnel?

- **Welche Namen sind einem Link zugewiesen?**
- Welche Abbiegerelationen existieren an einem Knoten für PKW?
- Ich möchte auf die GIP Daten referenzieren
- Welche Eigenschaften hat der Link für Radrouting?
- Welche Links sind eine Brücke oder ein Tunnel?

- 3 Tabellen betroffen



- Für Link auf Landesstraße sind 3 Namen zugewiesen
- Warum?

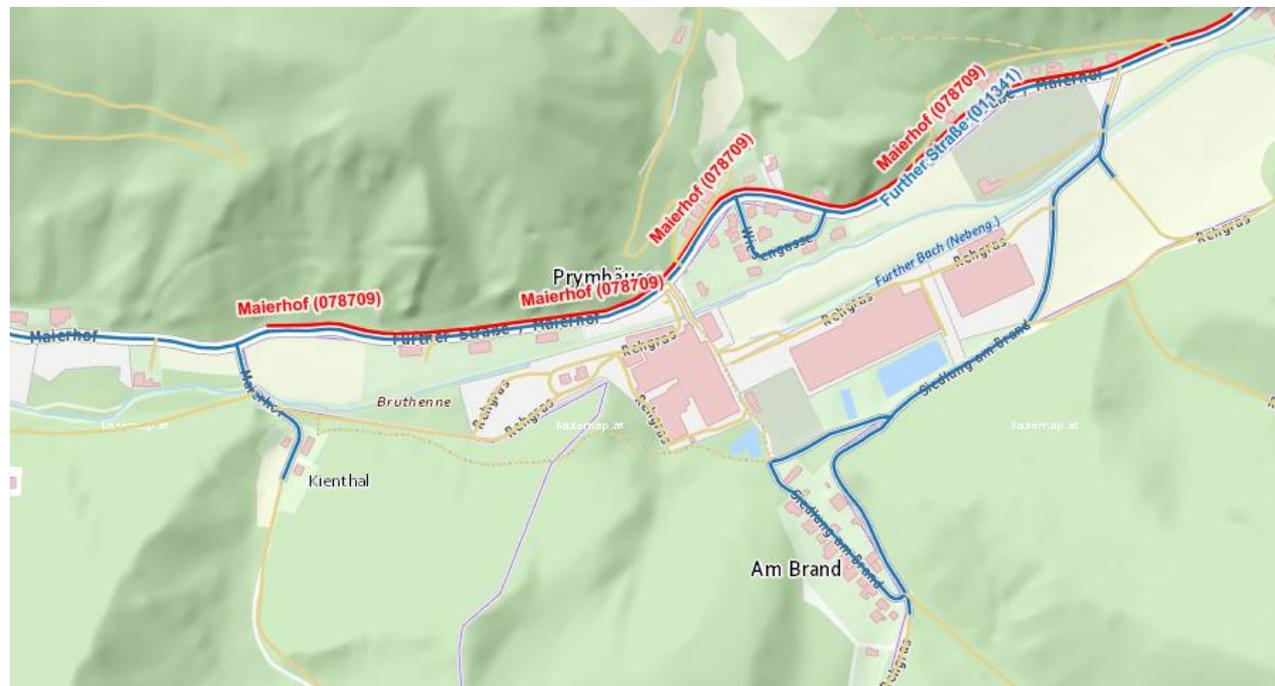
```
SELECT noe_link.link_id, streetn.street_name, r.name, streetn.namecat_val, streetn.shortname FROM
(SELECT * FROM IDF_LINK_VAO WHERE SUBNET_ID=401) noe_link
INNER JOIN
(SELECT * FROM IDF_LINK2STREETNAMES_VAO) link2street
ON link2street.link_id=noe_link.link_id
INNER JOIN
(SELECT * FROM IDF_STREETNAMES_VAO) streetn
ON streetn.street_id=link2street.street_id
INNER JOIN GIPAT.LUT_REGIONALCODE r
ON r.id=streetn.official_mun_code
WHERE noe_link.link_id=2318471;
```

Abfrageergebnis x

SQL | Alle Zeilen abgerufen: 3 in 1,344 Sekunden

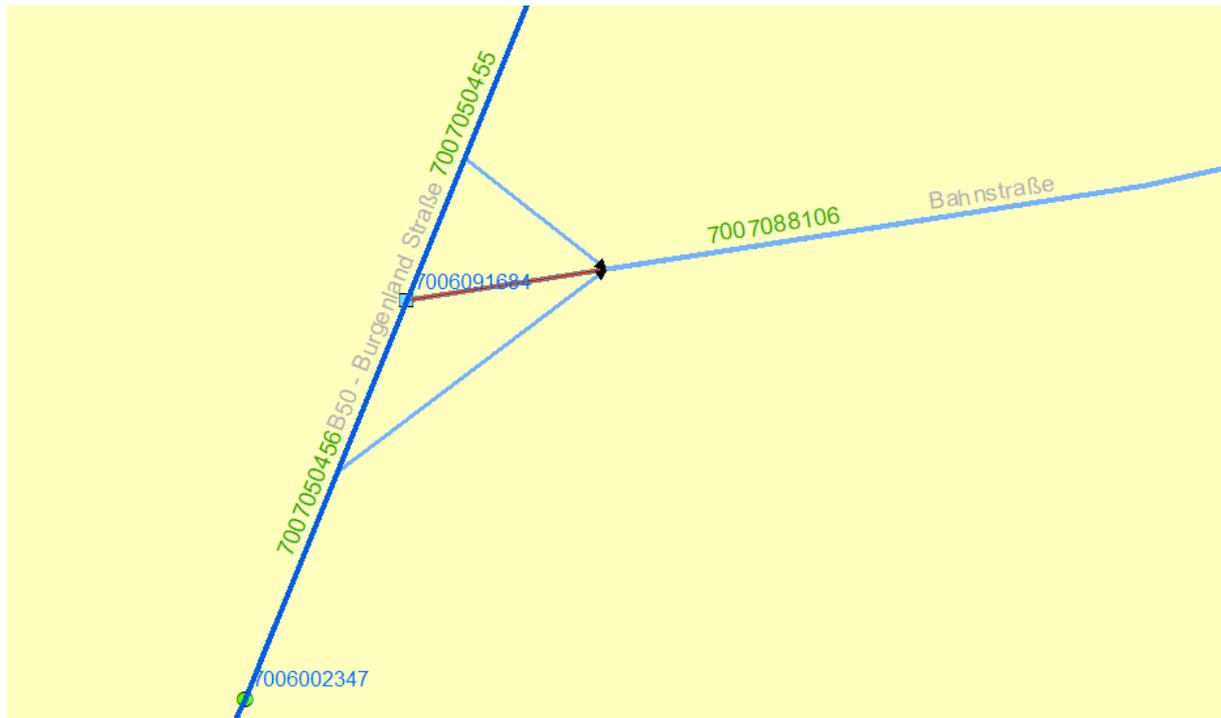
	LINK_ID	STREET_NAME	NAME	NAMECAT_VAL	SHORTNAME
1	2318471	B60 - Leitha Straße	Land Niederösterreich	Landesstraße B	B60
2	2318471	Hauptstraße	Götzendorf an der Leitha	Gemeindestraße	(null)
3	2318471	Vöslauer Straße	Bad Vöslau	Gemeindestraße	(null)

- Landesstraßenname: im Landesstraßengesetz vorgegeben
- Gemeindestraßenname: Straße bildet Gemeindegrenze und trägt 2 Namen
- Zusätzlich möglich: Gebrauchsnamen, Radwegennamen, Güterwegennamen usw.



- Welche Namen sind einem Link zugewiesen?
- **Welche Abbiegerelationen existieren an einem Knoten für PKW?**
- Ich möchte auf die GIP Daten referenzieren
- Welche Eigenschaften hat der Link für Radrouting?
- Welche Links sind eine Brücke oder ein Tunnel?

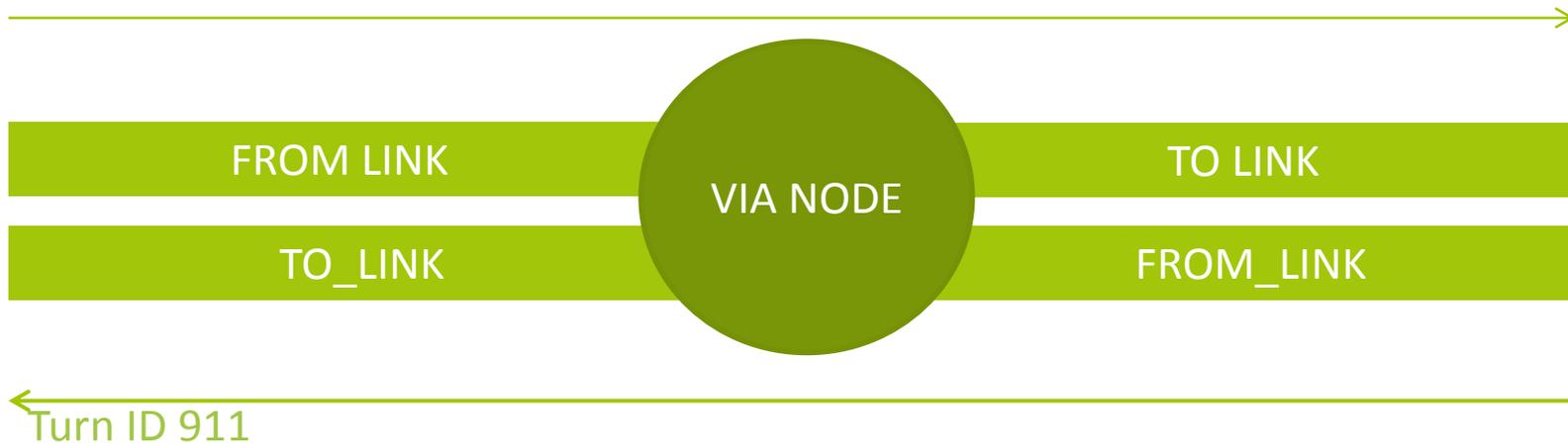
- Was ist im GIP OGD Export eine Abbiegerelation?



Für ein STVO
konformes Routing
von mot. Fahrzeugen
wird empfohlen, die
TURNEDGES zu
verwenden

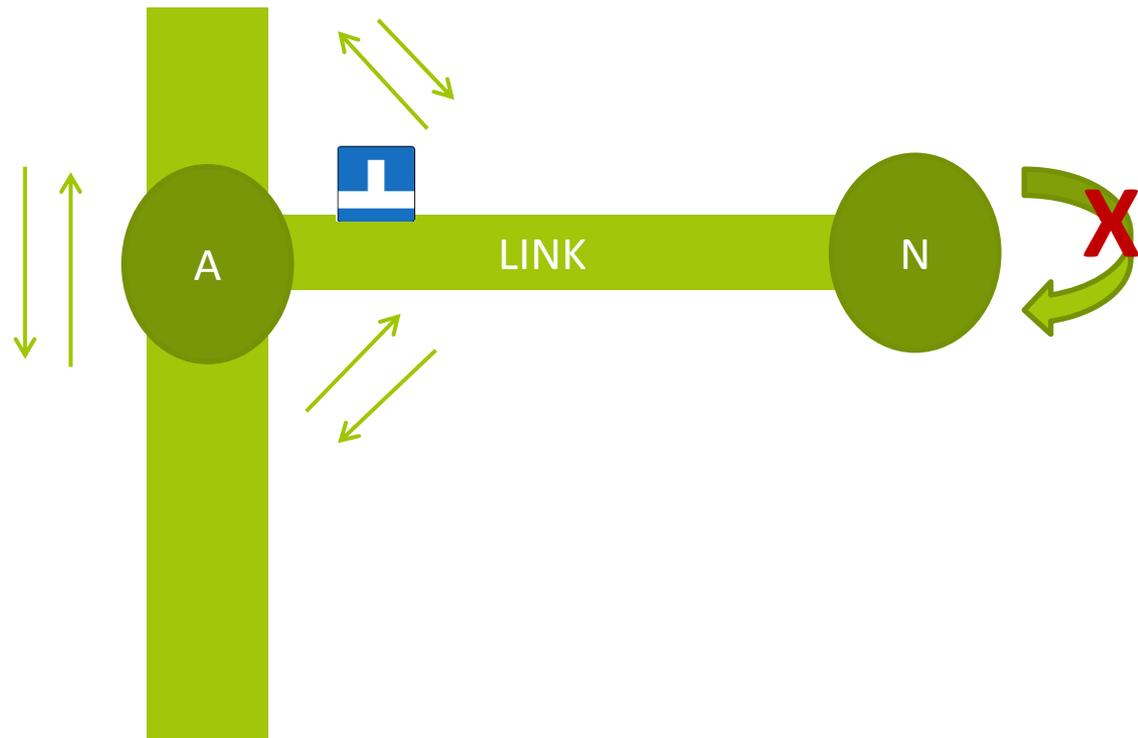
- Was ist im GIP OGD Export eine Abbiegerelation??

Turn ID 11

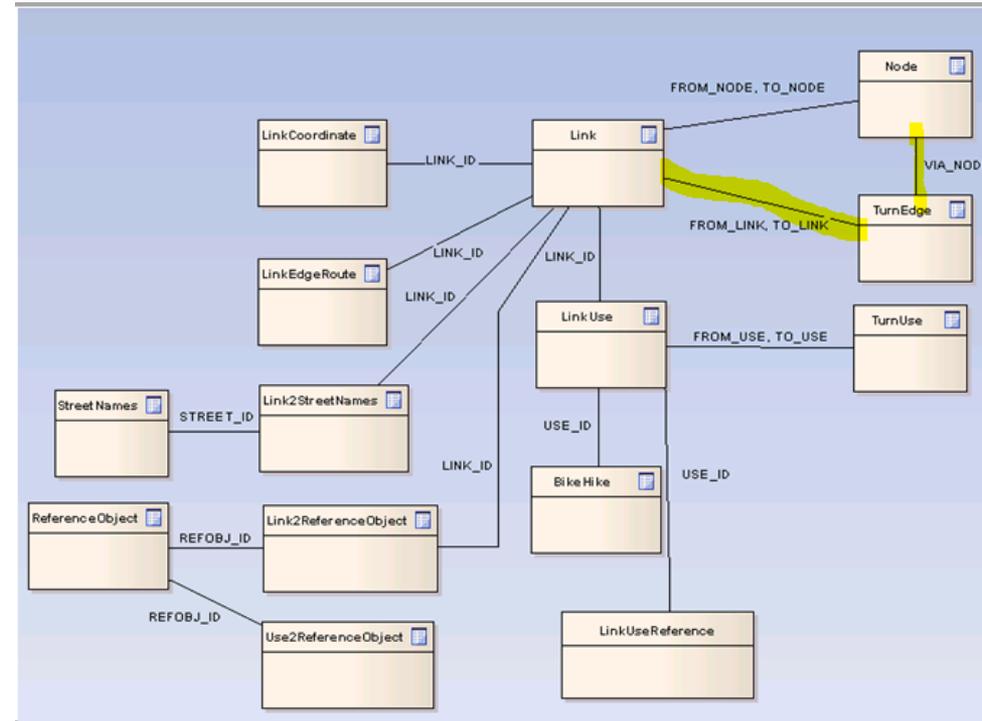


- Achtung → U Turns sind NICHT enthalten!

- MIV Routing muss damit umgehen können
- „Umkehren verboten“ derzeit im GIP Export nicht abgebildet

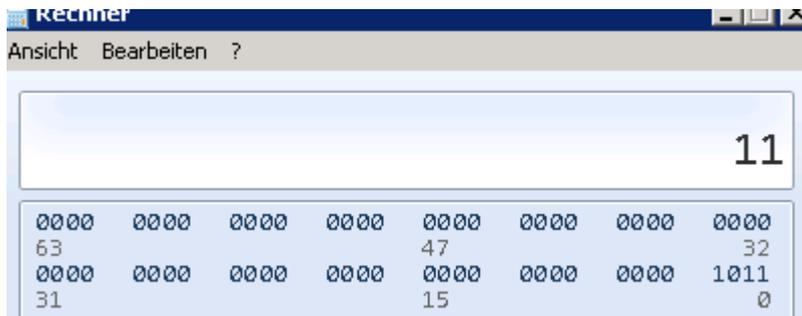


- IDF Tabelle TurnEdge enthält die Abbiegerlaubnisse
 - (nicht die Verbote)auf der Ebene der GIP Links
 - (nicht Nutzungstreifen)
- Alle Abbiegemanöver von Link nach Link über Knoten sind verboten, außer die explizit erlaubten!



- Der VEHICLE_TYPE gibt an, welche Verkehrsmittel abbiegen dürfen
- Das ist eine Bitmaske
- 11 = Bus, Rad, Fuß erlaubt
- $11 = 2^3 + 2^1 + 2^0$

Was sind Bit 3,1,0??



	A	B	C	M
1	ID	NAME	DECIMAL	MO
2	0	Fußgänger	1	
3	1	Radfahrer	2	
4	2	PKW	4	
5	3	Bus	8	
6	4	Eisenbahn	16	
7	5	Straßenbahn	32	
8	6	U-Bahn	64	
9	7	Schiff	128	

- Beispiel: Alle Abbiegerelationen für PKW und Rad ausgeflacht
- (Postgres: Bit Operator & statt BITAND)

Query builder

```
SELECT
  from_link.link_id,
  from_link.subnet_id,
  turnn.vehicle_type,
  CASE WHEN BITAND(turnn.vehicle_type,4)=4 THEN 'Y' ELSE 'N' END AS PKW_ABBIEGER_ERLAUBT,
  CASE WHEN BITAND(turnn.vehicle_type,2)=2 THEN 'Y' ELSE 'N' END AS RAD_ABBIEGER_ERLAUBT,
  to_link.link_id,
  to_link.subnet_id
FROM
  (SELECT * FROM IDF_TURNEDGE_VAO WHERE VIA_NODE=6091684) turnn
INNER JOIN IDF_LINK_VAO from_link
ON from_link.link_id=turnn.from_link
INNER JOIN IDF_LINK_VAO to_link
ON to_link.link_id=turnn.to_link;
```

Abfrageergebnis x

Alle Zeilen abgerufen: 6 in 3,165 Sekunden

	LINK_ID	SUBNET_ID	VEHICLE_TYPE	PKW_ABBIEGER_ERLAUBT	RAD_ABBIEGER_ERLAUBT	LINK_ID_1	SUBNET_ID_1
1	7088106	1130	1	N	N	7050455	1120
2	7050456	1120	2383631	Y	Y	7050455	1120
3	7088106	1130	1	N	N	7050456	1120
4	7050455	1120	2383631	Y	Y	7050456	1120
5	7050456	1120	2383631	Y	Y	7088106	1130
6	7050455	1120	2383631	Y	Y	7088106	1130

- Welche Namen sind einem Link zugewiesen?
- Welche Abbiegerelationen existieren an einem Knoten für PKW?
- **Ich möchte auf die GIP Daten referenzieren**
- Welche Eigenschaften hat der Link für Radrouting?
- Welche Links sind eine Brücke oder ein Tunnel?

- Der INTREST Export kennt nur LINKs, die aus den GIP Abschnitten erzeugt werden
- Link Ids sind instabiler als Abschnitts Ids
- Das Linknetz ist routingfähig, das Edgenetz nicht

→ **Empfehlung GIP Betrieb:**
Routing mit Links,
Referenzierung mit Edges

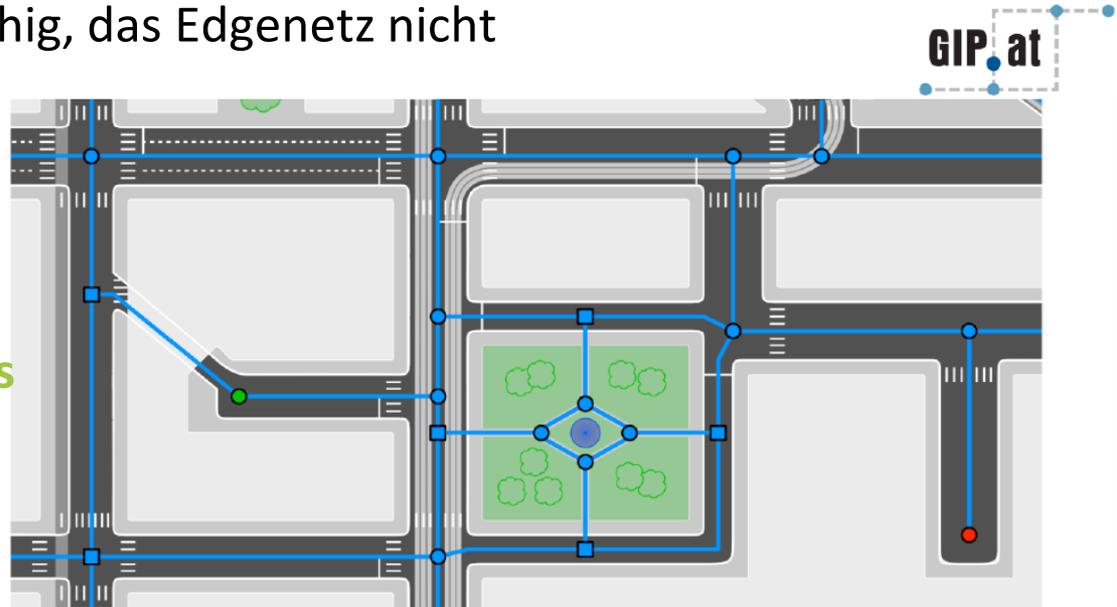
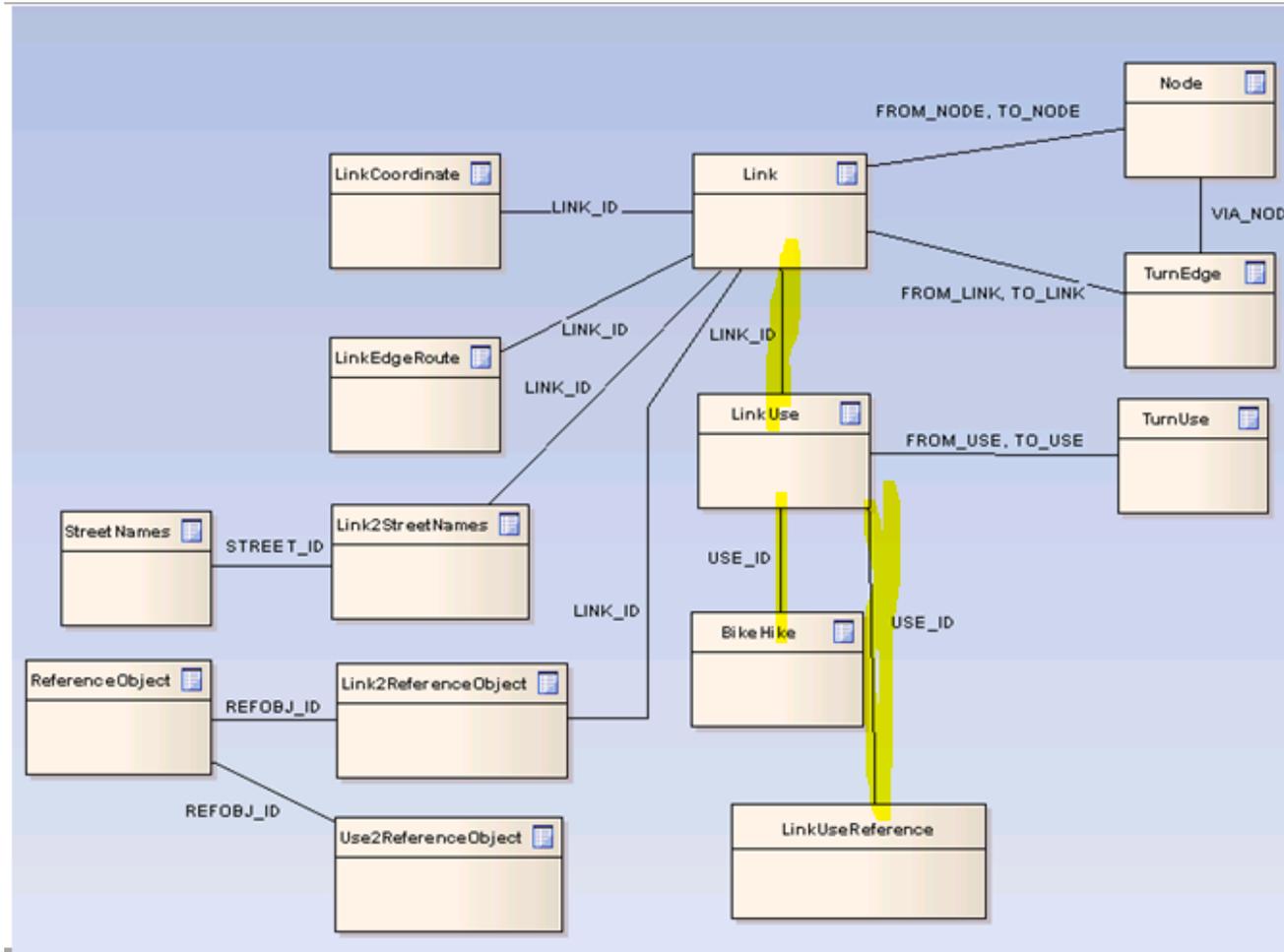


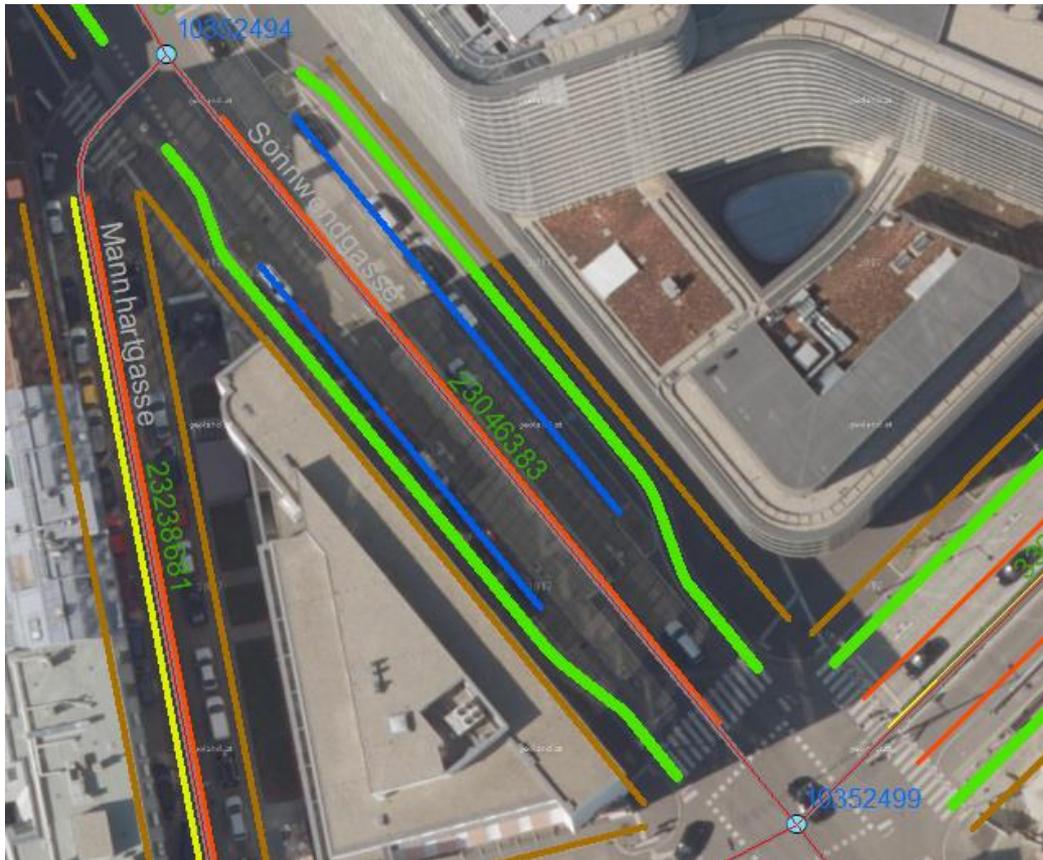
Abbildung: Die Unterscheidung in Knoten und virtuelle Knoten soll anhand dieser Abbildung beschrieben werden. Während Abschnitte eines Subnetzes an Kreuzungen immer aufgesplittet werden (Kreise), wird bei einer Kreuzung mit einem niederrangigen Subnetz ein virtueller Knoten (Quadrat) erzeugt.

- Welche Namen sind einem Link zugewiesen?
- Welche Abbiegerelationen existieren an einem Knoten für PKW?
- Ich möchte auf die GIP Daten referenzieren
- **Welche Eigenschaften hat der Link für Radrouting?**
- Welche Links sind eine Brücke oder ein Tunnel?

- Radattribute hängen am Nutzungstreifen!



- Radattribute hängen am Nutzungstreifen!



- Straßenquerschnitt gebildet

```
SELECT l.link_id,l.name1,y.name AS TYPE,x.offset,x.width,x.use_access_tow,
x.use_access_bkw,br.bikefeaturetow,br.bikefeaturebkw FROM
(SELECT * FROM IDF_LINK_VAO WHERE LINK_ID=23046383) l
INNER JOIN
(SELECT * FROM IDF_LINKUSE_VAO) x
ON x.link_id=l.link_id
LEFT JOIN
(SELECT * FROM IDF_BIKEHIKE_VAO) br
ON br.use_id=x.use_id
LEFT JOIN GIPAT.LUT_BASETYPE y
ON y.id=x.basetype ORDER BY OFFSET;
```

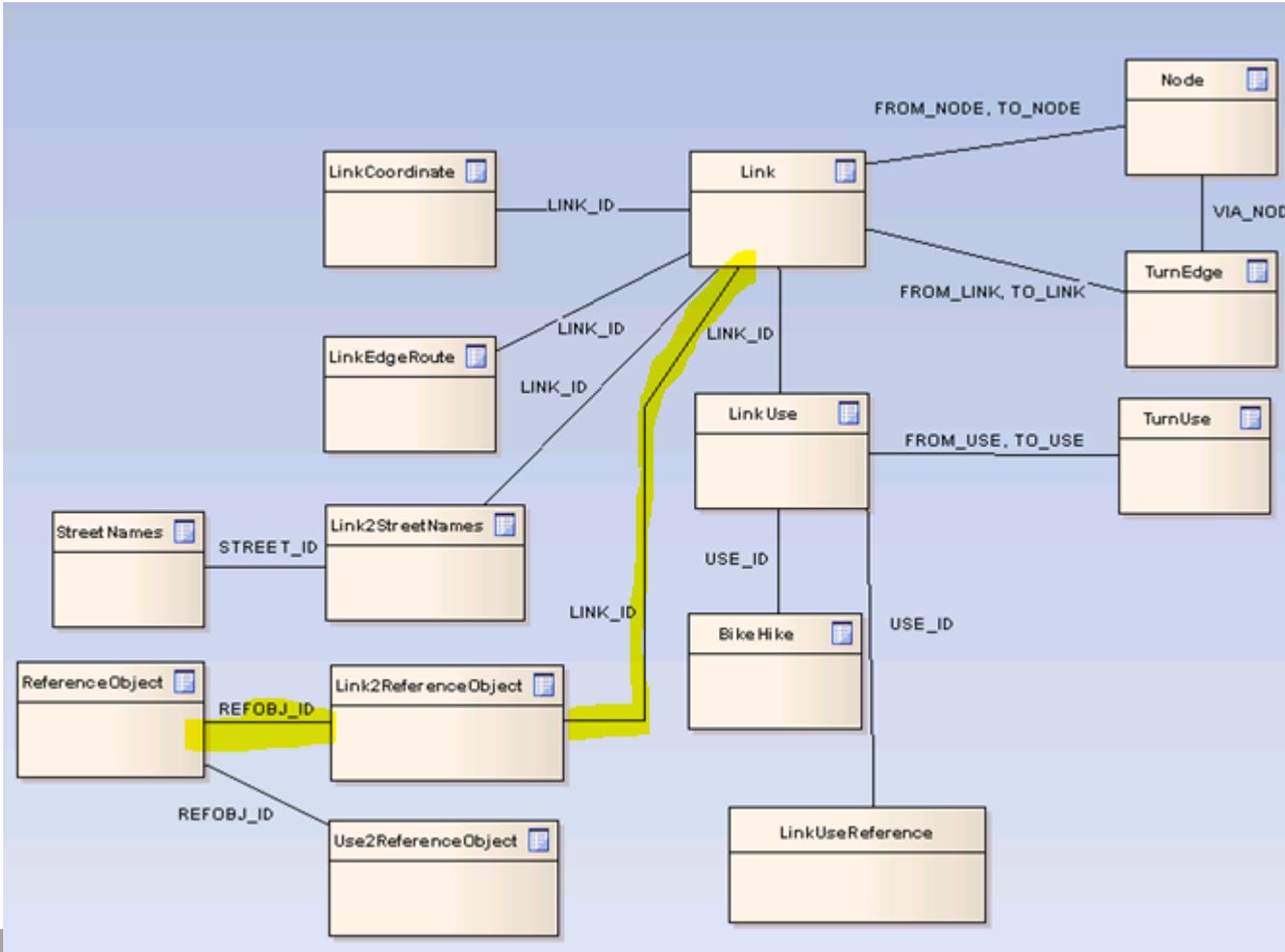
Abfrageergebnis x

SQL | Alle Zeilen abgerufen: 7 in 10,482 Sekunden

	LINK_ID	NAME1	TYPE	OFFSET	WIDTH	USE_ACCESS_TOW	USE_ACCESS_BKW	BIKEFEATURETOW	BIKEFEATUREBKW
1	23046383	Sonnwendgasse	Gehweg	-10,1	2,5	1		1 (null)	(null)
2	23046383	Sonnwendgasse	Radweg	-7,9	2,3	2		0 RW	(null)
3	23046383	Sonnwendgasse	Parkstreifen	-5,8	2	0		0 (null)	(null)
4	23046383	Sonnwendgasse	Fahrbahn	0,5	14,5	2383628	2383628	(null)	(null)
5	23046383	Sonnwendgasse	Parkstreifen	6	2,5	0		0 (null)	(null)
6	23046383	Sonnwendgasse	Radweg	9,8	2	0		2 (null)	RW
7	23046383	Sonnwendgasse	Gehweg	12,1	2,5	0		0 (null)	(null)

- Welche Namen sind einem Link zugewiesen?
- Welche Abbiegerelationen existieren an einem Knoten für PKW?
- Ich möchte auf die GIP Daten referenzieren
- Welche Eigenschaften hat der Link für Radrouting?
- **Welche Links sind eine Brücke oder ein Tunnel?**

- Brücken sind lineare Verortungsobjekte



- Abfrage der Links mit Brücken im Land Vorarlberg
- 1 Brücke kann n Links referenzieren!

```
SELECT p.refobj_id,lx.link_id,p.reftype_l,p.nametext,p.namecat_l,l.from_linkp,l.to_linkp FROM
(SELECT * FROM IDF_REFERENCEOBJECT_VAO WHERE REFTYPE='5002' AND OWNER_ID=13) p
LEFT JOIN IDF_LINK2REFERENCEOBJECT_VAO l
ON l.refobj_id=p.refobj_id
LEFT JOIN IDF_LINK_VAO lx
ON lx.link_id=l.link_id
ORDER BY p.nametext asc;
```

Abfrageergebnis x

SQL | 50 Zeilen abgerufen in 6,28 Sekunden

	REFOBJ_ID	LINK_ID	REFTYPE_L	NAMETEXT	NAMECAT_L	FROM_LINKP	TO_LINKP
1	1300000002349	801145915	Brücken	Alfenzbrücke	Brücke	0	31,4
2	1300000001560	801143416	Brücken	Alfenzbrücke	Brücke	67,4	87,2
3	1300000002349	801047720	Brücken	Alfenzbrücke	Brücke	33,9	100
4	1300000002606	801095983	Brücken	Alfenzbrücke	Brücke	67,4	98,4
5	1300000001429	801095980	Brücken	Alfenzbrücke	Brücke	0	0,6
6	1300000001429	801095225	Brücken	Alfenzbrücke	Brücke	18,6	100
7	1300000002717	801110756	Brücken	Almeintobelbrücke	Brücke	56,7	67
8	1300000003598	801110806	Brücken	Angerlitterbrücke	Brücke	72,9	100
9	1300000002955	801051579	Brücken	Apriltobelbrücke	Brücke	30,6	33,7
10	1300000002090	801139940	Brücken	Argenbachbrücke	Brücke	10,4	89,8
11	1300000001976	801111213	Brücken	Argenwaldbrücke	Brücke	7,9	9,2
12	1300000002190	801051734	Brücken	Atzibachbrücke	Brücke	0	37,3
13	1300000002190	801051570	Brücken	Atzibachbrücke	Brücke	71,2	100

- Abfrage der Links mit Brücken im Land Vorarlberg
- 1 Brücke kann n Links referenzieren!



Suchen

Features: Orte | Lineare Referenzierung

Suchen: 801111213

In: Links

Features suchen, die dem Ausdruck entsprechen oder ihn enthalten

Suchen:

Alle Felder

In Feld:

OBJECTID

Suchen

Anhalten

Neue Suche

Abbrechen

Klicken Sie mit der rechten Maustaste auf eine Zeile, um das Kontextmenü aufzurufen.

Wert	Layer	Feld
12801111213	Links	OBJECTID

FRAGEN?

Jederzeit auch gerne an gip@its-viennaregion.at