Kapitel 10

Ausblick

Stefan Keller

Ausblick

- In diesem Kurs nicht abgedeckt:
  - Extension Raster (postgis_raster) => siehe Anhang
  - Extension pgRouting (pgrouting)
  - Extension Topology (postgis_topology)
  - etc.

- Diskussion

- Ask-me-Anything (auf Etherpad)

ANHANG: PostGIS Raster

- Hybride Speicherung/Datenverarbeitung
- Gleiche/ähnliche SQL-Befehle wie bei Vektordaten-Verarbeitung
- Wachsende Anzahl von Analysemöglichkeiten
- Sehr flexible Speichermöglichkeiten
  - Innerhalb/ausserhalb DB
  - Eine Zelle per Record, ein Tile per Record, gesamter
- Rasterdatensatz per Record
  - Jeder Record kann andere Georeferenzierung haben
- Metadaten können direkt vom Rasterdatentyp abgeleitet werden

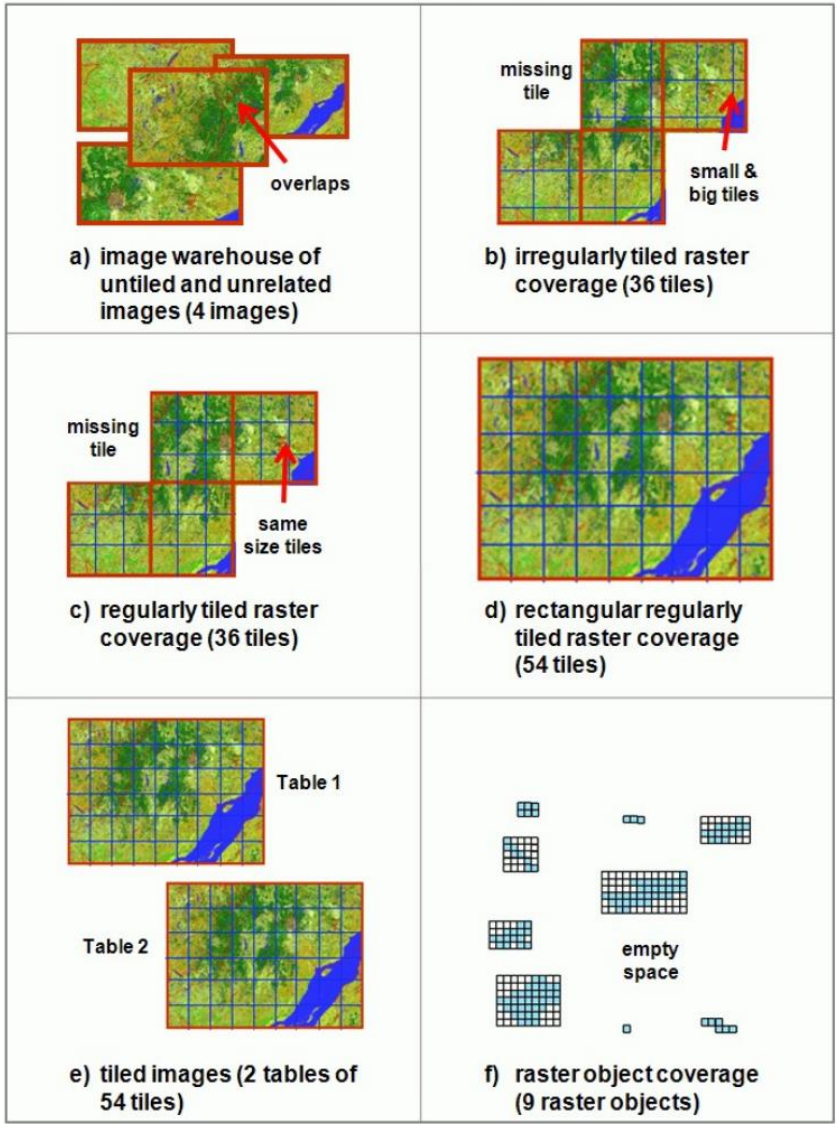
Mögliche Anwendungen

- Backend für Desktop-GIS
- Backend für ModelBuilder
- Backend für Reports/Berichte
- Backend für Web-GIS
- Backend für WPS
- Toolbox für Entwickler

Datenspeicherung in DB

- Vorteile
  - Alles in einer DB/Dump
  - Schnellere Analyse
  - Rasterbearbeitung kann Teil von DB-Transaktion sein
  - Besser getestet als bei Speicherung ausserhalb DB
- Nachteile
  - Bläst Dumpfile auf (Backup)
  - Raster können nicht von beliebigen anderen Tools bearbeitet werden – andere Tools brauchen DB-Treiber für Zugriff

Arten der Speicherung



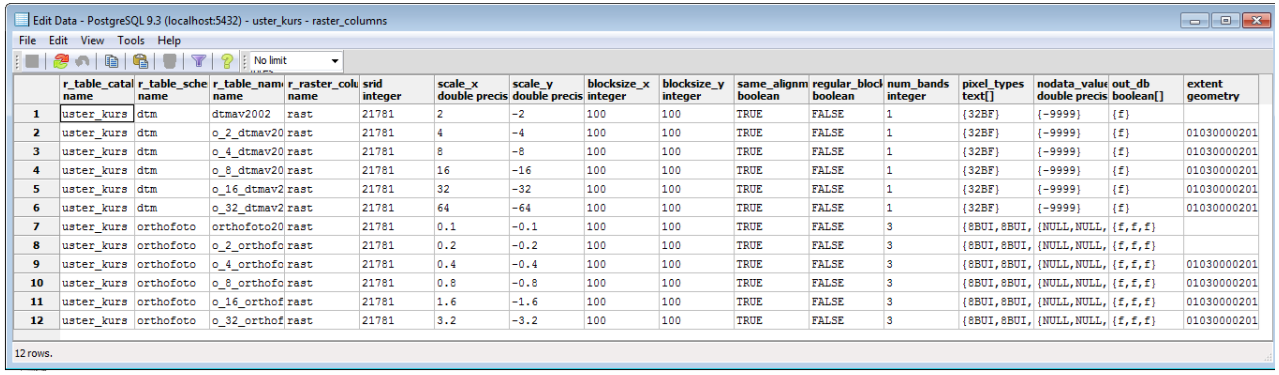
Verschiedene Raster können unterschiedliche…

- **Zellgrössen**
- **Anzahl Bänder**
- **Ursprünge/Grids**
- **Projektionen**

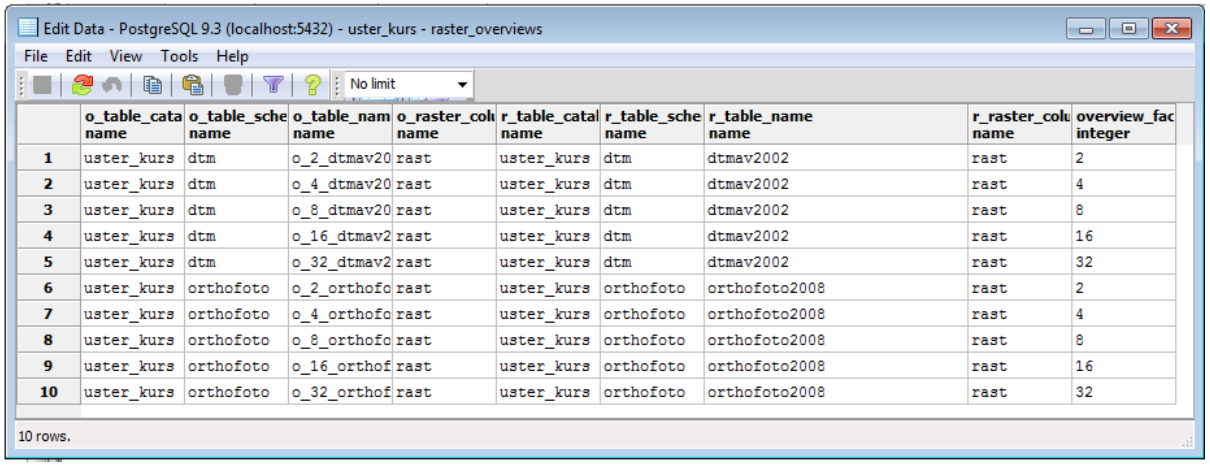
haben… und können trotzdem in einem gemeinsamen SQL-Statement verwendet werden

Metadaten-Views

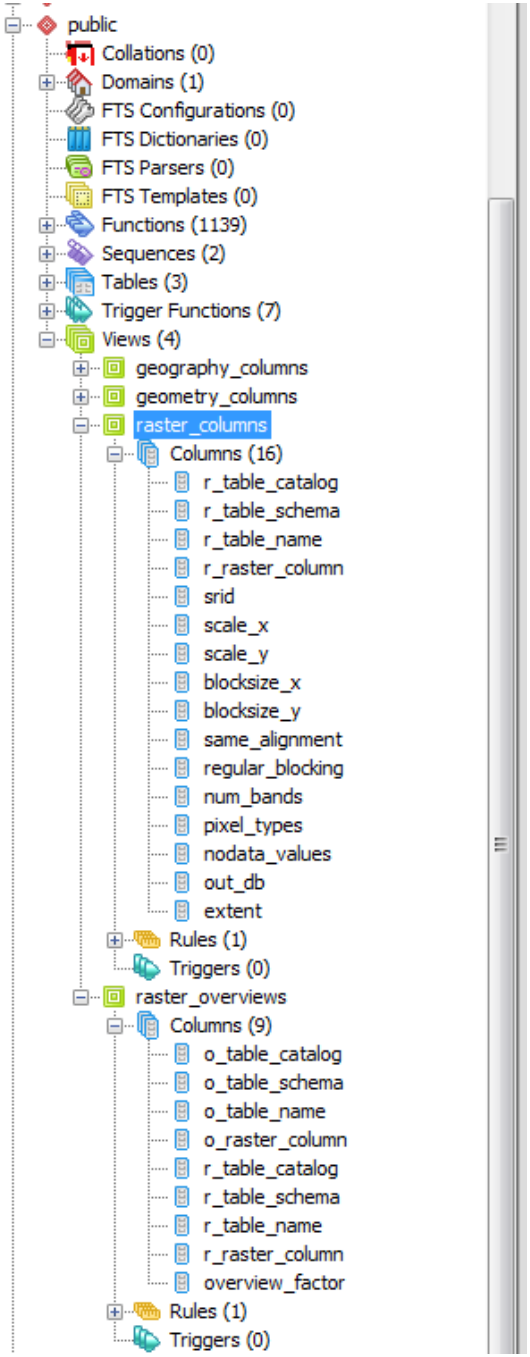
- public.raster_columns
- public.raster_overviews



	r_table_catalog name	r_table_schema name	r_table_name name	r_raster_column name	srid integer	scale_x double precision	scale_y double precision	blocksize_x integer	blocksize_y integer	same_alignment boolean	regular_blocking boolean	num_bands integer	pixel_types text[]	nodata_values double precision	out_db boolean[]	extent geometry
1	uster_kurs	dtm	dtmav2002	rast	21781	2	-2	100	100	TRUE	FALSE	1	{32BF}	{-9999}	{f}	
2	uster_kurs	dtm	o_2_dtmav20	rast	21781	4	-4	100	100	TRUE	FALSE	1	{32BF}	{-9999}	{f}	01030000201
3	uster_kurs	dtm	o_4_dtmav20	rast	21781	8	-8	100	100	TRUE	FALSE	1	{32BF}	{-9999}	{f}	01030000201
4	uster_kurs	dtm	o_8_dtmav20	rast	21781	16	-16	100	100	TRUE	FALSE	1	{32BF}	{-9999}	{f}	01030000201
5	uster_kurs	dtm	o_16_dtmav2	rast	21781	32	-32	100	100	TRUE	FALSE	1	{32BF}	{-9999}	{f}	01030000201
6	uster_kurs	dtm	o_32_dtmav2	rast	21781	64	-64	100	100	TRUE	FALSE	1	{32BF}	{-9999}	{f}	01030000201
7	uster_kurs	orthofoto	orthofoto20	rast	21781	0.1	-0.1	100	100	TRUE	FALSE	3	{8BUI,8BUI,	{NULL,NULL,	{f,f,f}	
8	uster_kurs	orthofoto	o_2_orthofo	rast	21781	0.2	-0.2	100	100	TRUE	FALSE	3	{8BUI,8BUI,	{NULL,NULL,	{f,f,f}	
9	uster_kurs	orthofoto	o_4_orthofo	rast	21781	0.4	-0.4	100	100	TRUE	FALSE	3	{8BUI,8BUI,	{NULL,NULL,	{f,f,f}	01030000201
10	uster_kurs	orthofoto	o_8_orthofo	rast	21781	0.8	-0.8	100	100	TRUE	FALSE	3	{8BUI,8BUI,	{NULL,NULL,	{f,f,f}	01030000201
11	uster_kurs	orthofoto	o_16_orthof	rast	21781	1.6	-1.6	100	100	TRUE	FALSE	3	{8BUI,8BUI,	{NULL,NULL,	{f,f,f}	01030000201
12	uster_kurs	orthofoto	o_32_orthof	rast	21781	3.2	-3.2	100	100	TRUE	FALSE	3	{8BUI,8BUI,	{NULL,NULL,	{f,f,f}	01030000201



	o_table_catalog name	o_table_schema name	o_table_name name	o_raster_column name	r_table_catalog name	r_table_schema name	r_table_name name	r_raster_column name	overview_factor integer
1	uster_kurs	dtm	o_2_dtmav20	rast	uster_kurs	dtm	dtmav2002	rast	2
2	uster_kurs	dtm	o_4_dtmav20	rast	uster_kurs	dtm	dtmav2002	rast	4
3	uster_kurs	dtm	o_8_dtmav20	rast	uster_kurs	dtm	dtmav2002	rast	8
4	uster_kurs	dtm	o_16_dtmav2	rast	uster_kurs	dtm	dtmav2002	rast	16
5	uster_kurs	dtm	o_32_dtmav2	rast	uster_kurs	dtm	dtmav2002	rast	32
6	uster_kurs	orthofoto	o_2_orthofo	rast	uster_kurs	orthofoto	orthofoto2008	rast	2
7	uster_kurs	orthofoto	o_4_orthofo	rast	uster_kurs	orthofoto	orthofoto2008	rast	4
8	uster_kurs	orthofoto	o_8_orthofo	rast	uster_kurs	orthofoto	orthofoto2008	rast	8
9	uster_kurs	orthofoto	o_16_orthof	rast	uster_kurs	orthofoto	orthofoto2008	rast	16
10	uster_kurs	orthofoto	o_32_orthof	rast	uster_kurs	orthofoto	orthofoto2008	rast	32



Format, Bänder und Pyramiden

- Speicherformat: wkb
- Beliebige Anzahl Bänder pro Raster
- Pyramiden werden vor dem Einlesen generiert (gdal), müssen im Vorhinein festgelegt werden
- Pyramiden sind in separaten Tabellen gespeichert

Rasterimport

- Import über raster2pgsql
  - Alle Formate die GDAL unterstützt (>100 Formate)
  - Import in 2 Schritten: Erzeugung eines .sql-Files, dann einlesen des SQL-Files – oder Pipe verwenden
  - SQL kann nach dem Import gelöscht werden
- Beispiel:

```
% raster2pgsql -s 21781 -I -t 100x100
  -l 2,4,8,16,32
  -C dtm_av.txt dtm.dtmav2002
  > dtmav2002.sql
```

- Optional können beim Import auch Pyramiden generiert werden
- Alternativ: direkte Konvertierung mit gdal_translate

Rasterexport

- Über GDAL (gdal_translate, gdal_warp) – wie alle anderen Raster
- ST_AsBinary()
- ST_AsGDALRaster()
- ST_AsJPEG
- ST_AsTiff
- ST_AsPNG
- Vektorobjekte können beim Export auf Wunsch rasterisiert werden
- ST_As<XX>Export() eher umständlich! - besser GDAL verwenden

Befehlsgruppen

- Raster Management Funktionen
- Raster und Raster-Band Zugriffe und Konstruktoren
- Raster Pixel Zugriff und Setters
- Raster Editoren und Band Editoren
- Raster Ausgabe
- Rastern Statistik und Analyse
- Raster Prozessierungs-Funktionen
- Raster Operatoren
- Raster Spatial-Relationship Operatoren

Rastereditoren

- ST_SetGeoreference() - neue Georeferenz
- ST_Transform(), ST_SetScale(), ST_SetRotation(), ST_SetSkew() - diverse Transformationen
- ST_SetUpperLeft() - Verschieben (oben links)
- ST_SetSRID() - neues Koordinatensystem definieren
- ST_Rescale() - verschiedene Interpolationen
- ST_Resample() - verschiedene Resampling-Algorithmen

Statistik

- ST_Count()
- ST_Histogram()
- ST_Quantile
- ST_SummaryStats()
- ST_ValueCount()

Rasterprozessierung

- ST_Clip()
- ST_ConvexHull()
- ST_DumpAsPolygons() - Vektorisierung
- ST_Hillshade() - Schattierungsberechnung
- ST_Intersection() - Langsam! > ST_Clip() verwenden
- ST_Union()
- ST_MapAlgebraExpr()/ST_MapAlgebraFct()

Räumliche Relationen

- ST_Intersects()
- ST_SameAlignment()

Ideen für SQL-Abfragen

- Rasterwerte abfragen für Punkte/Linien/Flächen (z.b. Mittlere Gebäudehöhe, Baumhöhen (DOM/DTMAV))
- Reklassifizierungen
- Map-Algebra für Analysen
- Geländemodellierung direkt in DB
- Kürzeste Wege finden wobei Raster die Gewichte als „Hindernisse“ darstellen
- Raster ausschneiden aufgrund von Vektorgeometrien
- (z.B. Alle Hausdächer aus dem Orthofoto extrahieren)

Mögliche Probleme

- Tilegrösse- und Pyramiden-Parameter können nur beim Import definiert werden
- Out of DB-Raster noch zu wenig getestet
- Noch zu wenige Clients/Applikationen/GUI
- Manche Werkzeuge verwenden Pyramiden nicht automatisch
- Performance manchmal noch nicht optimal

Fazit

- Vektor/Rasteranalyse neu?
- Nein, aber einzigartige Realisierung!
- Existierendes SQL-Knowhow kann auch für Rasteranalyse eingesetzt werden
- Transparente SQL-Funktionen/Operatoren wenn möglich gleich bei Vektor- und Raster
- Berechnungen nur dann wenn nötig und nur für den gerade nötigen geographischen Ausschnitt