

PostGIS

3

A cartoon illustration of a grey elephant sitting on the ground, facing left. Its trunk is raised and curled, holding a small globe of the Earth. The globe shows the Americas in green and blue oceans. The elephant has small blue eyes and a friendly expression.

FELIX KUNDE

slides.com/fxku/postgis-v3

ABOUT ME

Database Engineer @ Zalando

Geoinformatics background

Guest lecturer on spatial databases

Postgres Operator, 3DCityDB and pgMemento

@FlxKu



WHAT IS POSTGIS?



- Extension to [PostgreSQL](#) database
- Comes with it's own datatypes for geodata
- Supports coordinate reference systems
- Spatial indexing for fast geo queries
- Open Source under GPLv2
- More infos under <http://postgis.net/>

WHY IS IT GREAT?

- Faster and more robust than your GIS
- So much geo power with just some SQL
- Great acceptance in the spatial industry
- Follows international OGC/ISO standards
- Build on top of one of the best databases



TERADATA



DW / MPP
/ Hadoop



Cloud



NoSQL



POSTGRESQL

Sharding



FORKS & EXTENSIONS

GPU PG-STORM

Streaming SQL

Time Series



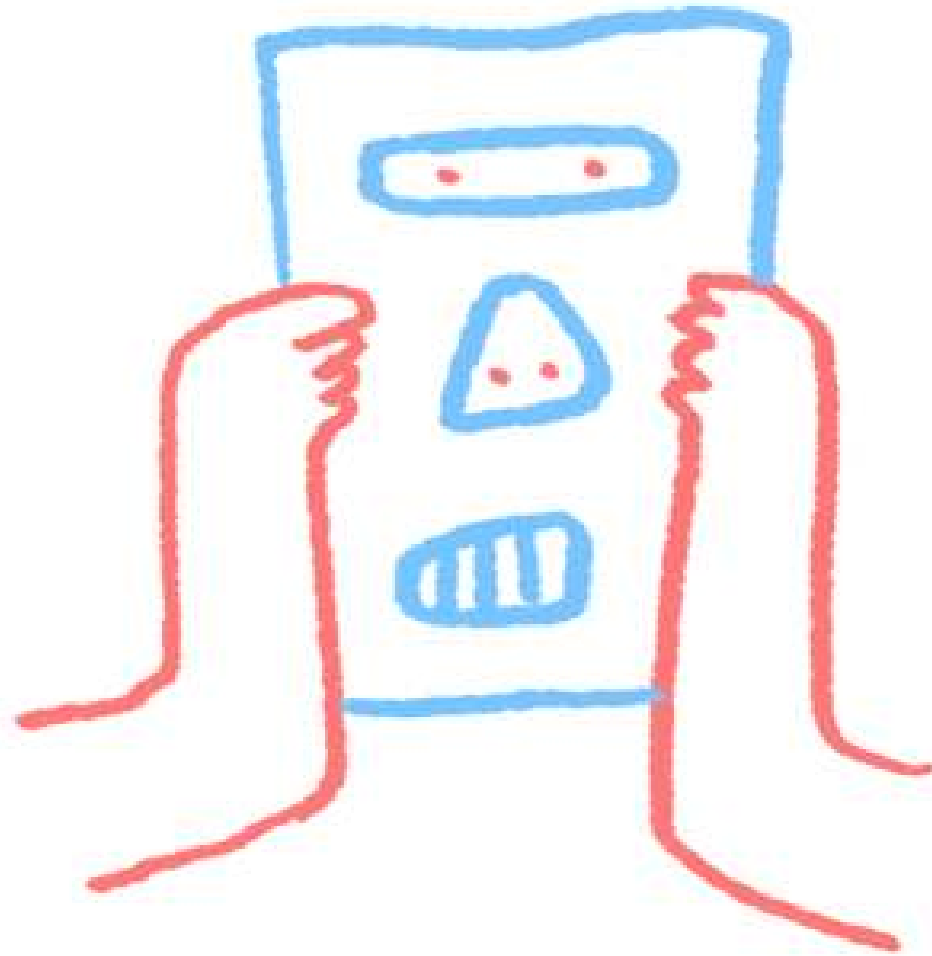
TIMESCALE



@delawen on PostGIS day



*THE
HIDDEN
THINGS*



NEW ON-DISC FORMAT

- More optional bytes for [new things](#)
- Probably: Efficient point type
- Probably: Faster joins
- Maybe: Precision model
- Upgrade support (no dump and restore)

MINIMAL TOAST DECOMPRESSION

- Big geoms are sliced and [compressed](#)
- When read, decompression takes time
- Postgres 12 can "sneak" into first slice
- E.g. read BBox to decide to skip geom

OPTIMIZER

SUPPORT FUNCTIONS

- Problem <v3: Function inlining to trigger index
 - Hard for planner to consider parallel query
- Solution: Give optimizer *insights* about functions ...
 - ... and see more parallel spatial queries

O
L
D

N
E
W

ST_Intersects(geom1, geoms2)

SQL function



SELECT geom1 && geom2 **AND**
_ST_Intersects(geom1, geoms2)

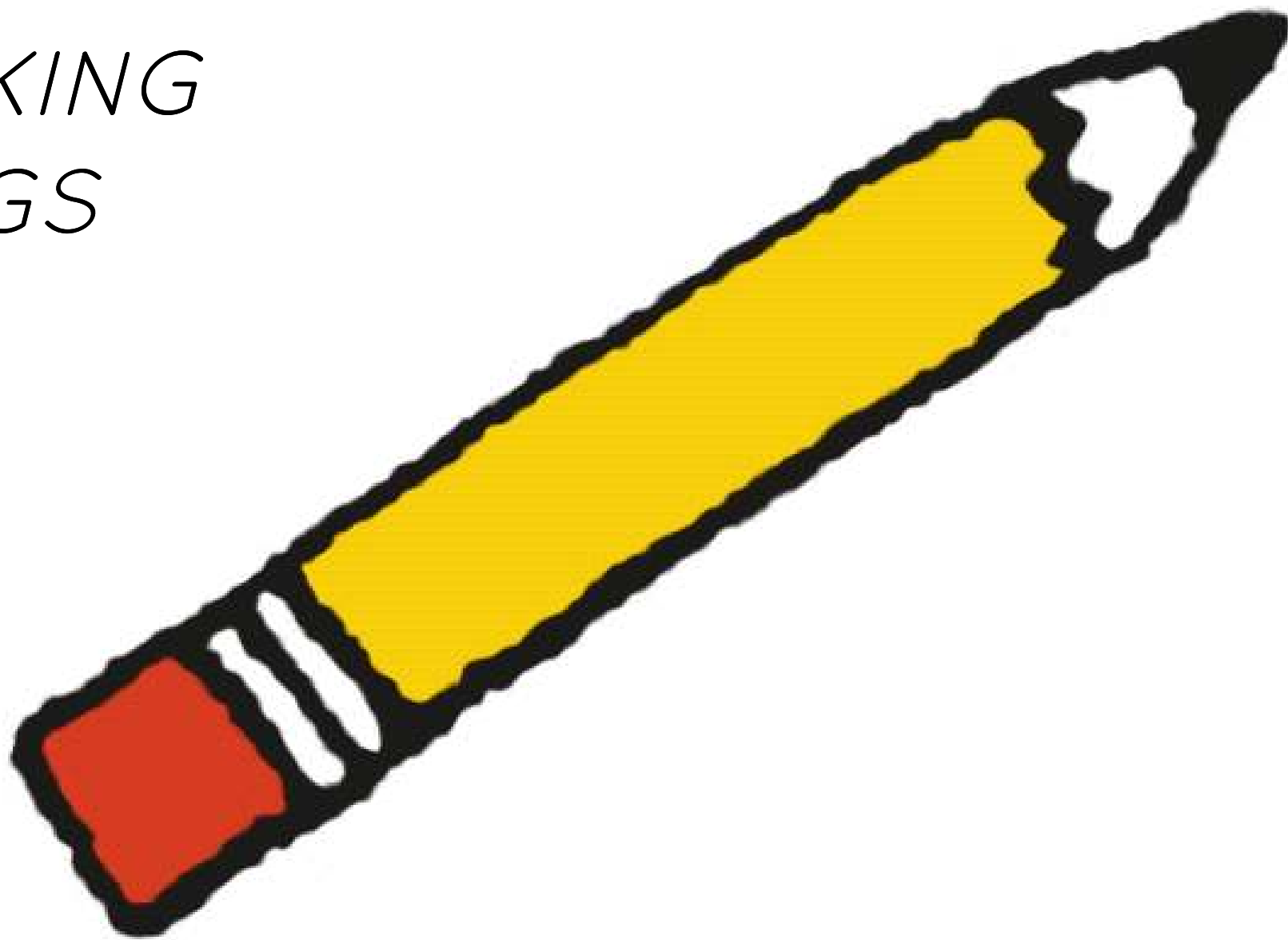
→ *triggers index*

→ *C function*

ST_Intersects(geom1, geoms2)
SUPPORT postgis_index_supportfn

→ *C function triggers index*

*BREAKING
THINGS*



POSTGIS_RASTER

```
my_postgis_db=# ALTER EXTENSION postgis UPDATE TO '3.0.0';  
WARNUNG:  unpackaging raster  
WARNUNG:  PostGIS Raster functionality has been unpackaged  
TIP:  type `SELECT postgis_extensions_upgrade();` to finish  
the upgrade. After upgrading, if you want to drop raster,  
run: DROP EXTENSION postgis_raster;  
ALTER EXTENSION  
  
my_postgis_db=# SELECT postgis_extensions_upgrade();  
HINWEIS:  Packaging extension postgis_raster
```

STRIP MINOR FROM LIB

```
$> pg_upgrade  
ERROR: could not access file "$libdir/postgis-2.5":  
No such file or directory
```

- Before: ALTER EXTENSION postgis UPDATE first to fix it
- Now: Lib is called **postgis-3.so** for all minor releases
- Upgrade your Postgres with pg_upgrade and get the new PostGIS functions when running ALTER EXTENSION etc.

BROKEN INDEXES

- REINDEX your **HASH** indexes
- REINDEX your **BTREE** indexes
- REINDEX your ***nD spatial*** indexes
 - But hey, SP-GiST and GiST now support *nD* box operators for overlaps, contains, within, equals

BYE BYE

- ST_Accum(), use array_agg
- ST_AsGeoJSON(version, geometry)
- ST_AsKML(version, geometry)
- Remove SFCGAL support for functions which are already covered by GEOS
- postgisversion() > postgislibrevision()
- liblwgeom headers, librttopo if you need
- **PostgreSQL 9.5 support** (3.1+)

*THE
SHINY
THINGS*



DEMO TIME

REALLY RANDOM POINTS

O
L
D

```
SELECT ST_GeneratePoints(poly, 3)  
FROM geom, generate_series(1,5);
```

→ *same result*

REALLY RANDOM POINTS

O

```
SELECT ST_GeneratePoints(poly, 3)  
FROM geom, generate_series(1,5);
```

→ *same result*

L

D

N

E

W

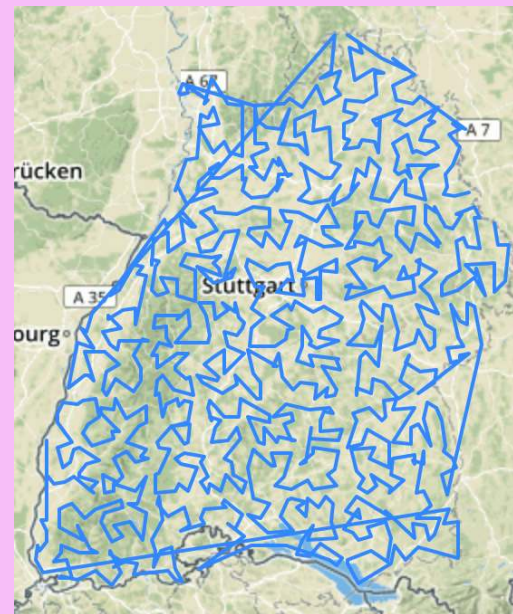
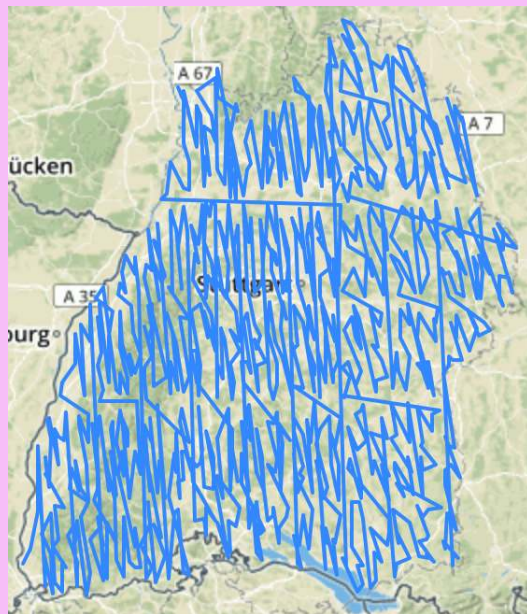
```
SELECT ST_GeneratePoints(poly, 3, seed := 1)  
FROM geom, generate_series(1,5);
```

→ *random result*

```
SELECT ST_GeneratePoints(poly, 3)  
FROM geom, generate_series(1,5);
```

HILBERT CURVE ORDER

- Switch from Z-Curve in 2.x to Hilbert Curve
- More compact spatial alignment
- 30% faster algorithm also boosts GiST creation



PLAYING WITH TINs

- ST_ConstrainedDelaunayTriangles in SFCGAL
- TINs can be passed to GEOS functions
- Now supported by all output functions
- ST_3DIntersects (2D, Solid), ST_3DDistance (Solid)



MORE LRS FOR POLYGONS

- ST_LocateBetween/Elevations now support:
 - GeometryCollection, Polygon, TIN, Triangle



triangles < -2m

TOPOLOGY TESTS FOR GEOMETRYCOLLECTION

- ST_Overlaps, ST_Contains, ST_ContainsProperly, ST_Covers, ST_CoveredBy, ST_Crosses, ST_Touches, ST_Disjoint, ST_Relate, ST_Equals now work with GeometryCollection
- Think about all the queries where you needed to ST_Dump before (like after ST_LocateBetween ;)

GEOM::JSON

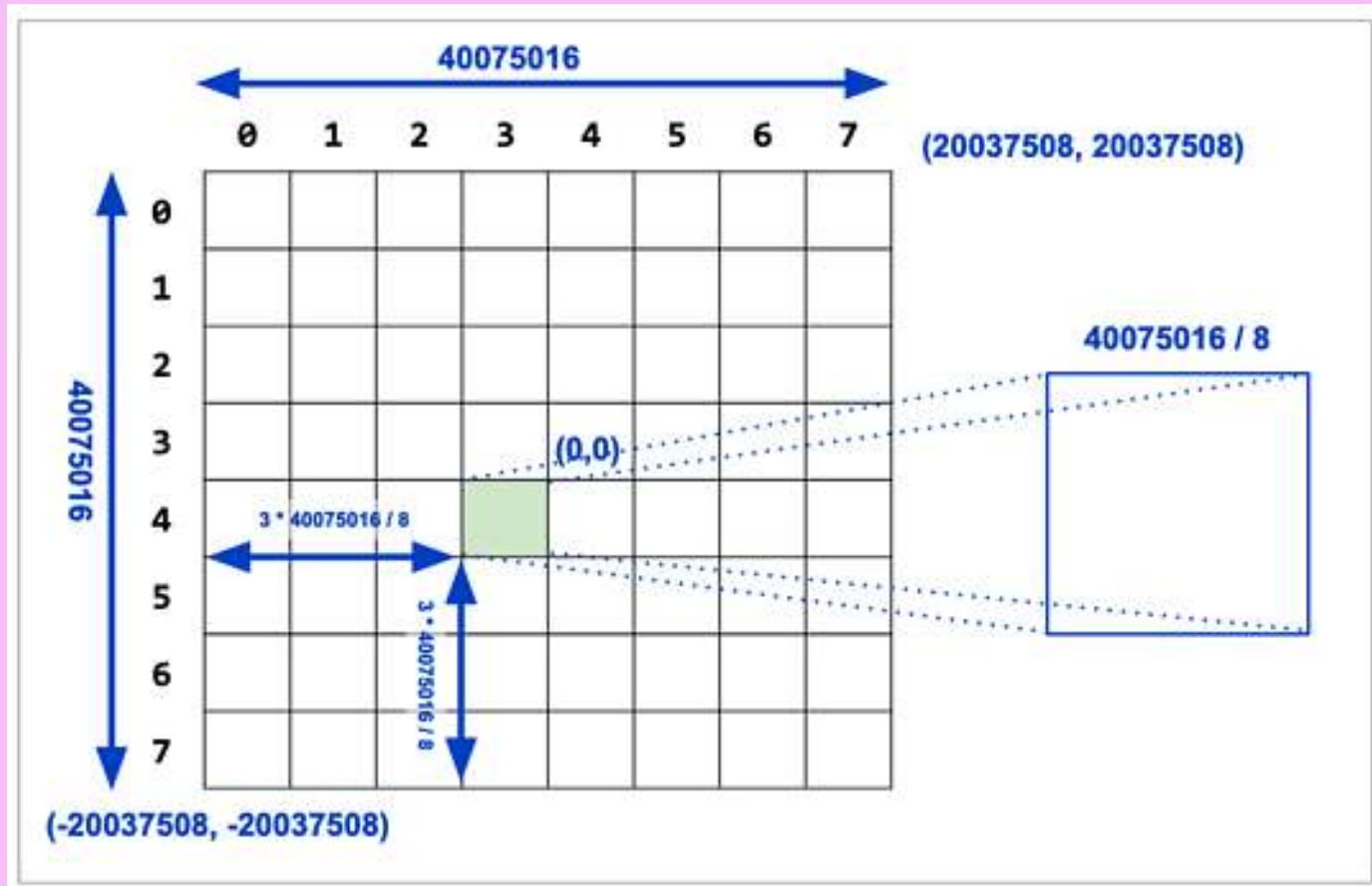
- Geometry can be casted with `::json` / `::jsonb`
- `ST_AsGeoJSON(record) > GeoJSON Feature`
- Geography columns supported when in row

FASTER VECTOR TILES

- ST_AsMVT **boost** with parallel aggregation
- ST_AsMVTGeom more robust output
- **Wagyu** for validation and clipping (GEOS job in the future)
- Feature ID support
- **Serving MVT from PostGIS is easy**



ST_TILEENVELOPE

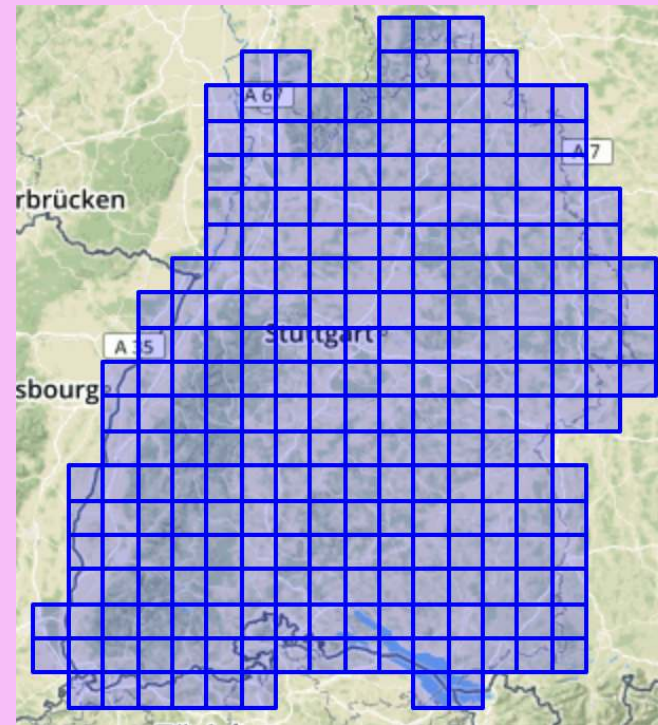


- BBox of Tile per zoom level
- WebMercator (EPSG 3857) bounds by default
- Custom bounds possible
- Margin in PostGIS 3.1

GRIDS (3.1)



ST_HexagonGrid



ST_SquareGrid

DEFAULT 3D/4D (3.1)

- Additional argument for ST_ForceXXX functions
- Define default Z/M, still 0 when left out
- **[open]** Have ST_SetZ, ST_SetM function

HIDDEN HEROES

P R Ø J



GDAL - Geospatial Data
Abstraction Library



GEOS

Geometry
Engine
Open
Source

SF Ø GAL





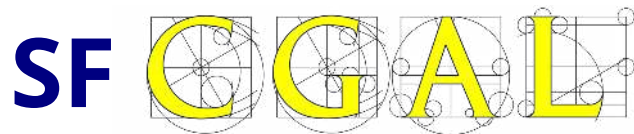
- Many performance improvements
- `postgis_geos_noop` (SQL \leftrightarrow GEOS)



- More precise datum **transformation**
- WKT projections support



- out-db raster support since 2.4
- check out `ogr_fdw` extension!



- Make it totally independent
- New: `ST_ConstrainedDelaunayTriangles`

POSTGIS FUTURE

2020/21

- New efficient geom types? External storage type?
- Point density surfaces (weighted, kriging)
- **Tolerance & Precision** ([#1629](#))?
- Much work in GEOS 3.9
- 3D-aware geography?
- **Index-only scans** with geometry?
- <https://trac.osgeo.org/postgis/wiki/PostGIS3>

THANKS

to

Regina, Paul, Sandro, Darafei, Raúl, Bborie,

Alex, Andrea, Andreas, Anne, Arthur, Barbara, Bas, Ben, Bernhard, Bill, Björn, Brian, Bruce, Bruno, Bryce,
Carl, Charlie, Chris, Christian, Dan, Dane, David, Eduin, Even, Esteban, Frank, George, Gerald, Gino,
Guillaume, Iida, Ingvild, Jason, Jeff, Joe, Jorge, Jose Carlos, Julien, Hugh, Kashif, Kevin, Klaus, Kris, Kristian,
Laurenz, Leo, Loic, Luca, Lucas, Maria, Mark, Markus, Mateusz, Matti, Maxime, Michael, Mike, Nathan,
Nathaniel, Nicklas, Nikita, Norman, Olivier, Pierre, Rafal, Ralph, Rémi, Richard, Silvio, Steffen, Stephen,
Steven, Sunveer, Tom, Vincent, Yuri

Teams behind GEOS/JTS, Proj, GDAL and (SF)CGAL!

The whole PostgreSQL community!

The funding companies, organisations and individuals!