
**www.gisspecialisten.nl** | Ringwade 41 | 3439 LM Nieuwegein | +31 (0)85 7 325 825

GIS Specialisten
consultancy & detachering

# Spatial Databases (1)

# PostgreSQL+ PostGIS

Parya Pasha Zadeh
Utrecht, May 2019

# Introduction and structure

| | |
|---|---|
| Beginner | Introduction to databases and its advantages,<br>Principles of relational data model, installation and data import,<br>Database access through pgAdmin and QGIS<br>Simple SQL queries. |
| Intermediate | OGC simple features standards,<br>Spatial data ingestion,<br>Advanced spatial queries and analysis, views and tables |
| Advanced | Complex spatial queries,<br>Integrity constraints implementation,<br>Database functions and triggers,<br>Server-side programming |

GIS Specialisten
consultancy & detachering

# Today's schedule

| | |
|---|---|
| 09:00 – 10:30 | intro lecture,<br>software installations<br>materials access |
| 10:30 – 10:45 | *Coffee break* |
| 10:45 – 12:30 | non-spatial queries and SQL recap<br>spatial data handling (spatial_ref_sys and geometry columns, OGC SFS) |
| 12:30 – 13:15 | *Lunch break* |
| 13:15 – 15:00 | spatial queries<br>spatial functions<br>invalid geometries |
| 15:00 – 15:15 | *Coffee break* |
| 15:15 – 17:00 | dynamic queries (views vs tables)<br>PK-FK, GiST indexes<br>The way forward |

GIS Specialisten
consultancy & detachering

# Objectives

Upon completion of this training you will be able to:

- Explain the concepts and structure of relational (spatial) databases;
- Understand the client-server architecture
- Recognize the differences between spatial and non-spatial DBMS
- Perform systematic query formulation using SQL
- Extend your SQL knowledge and use spatial functions in your queries.

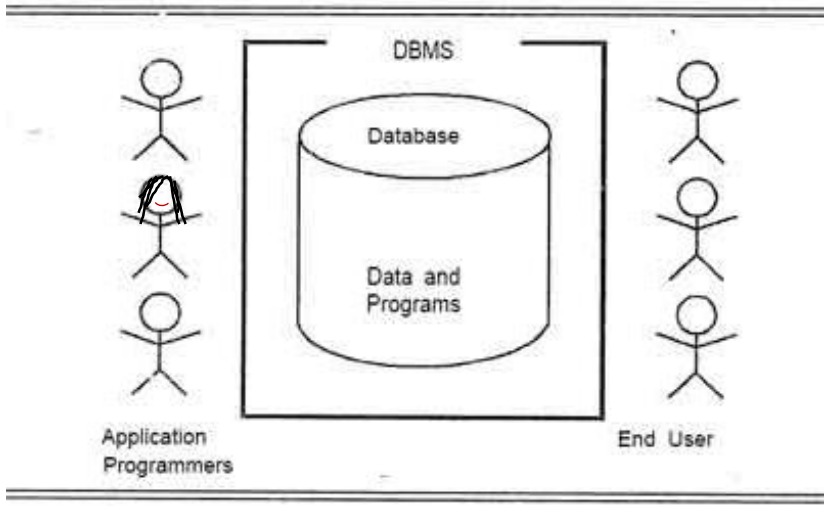
GIS Specialisten
consultancy & detachering

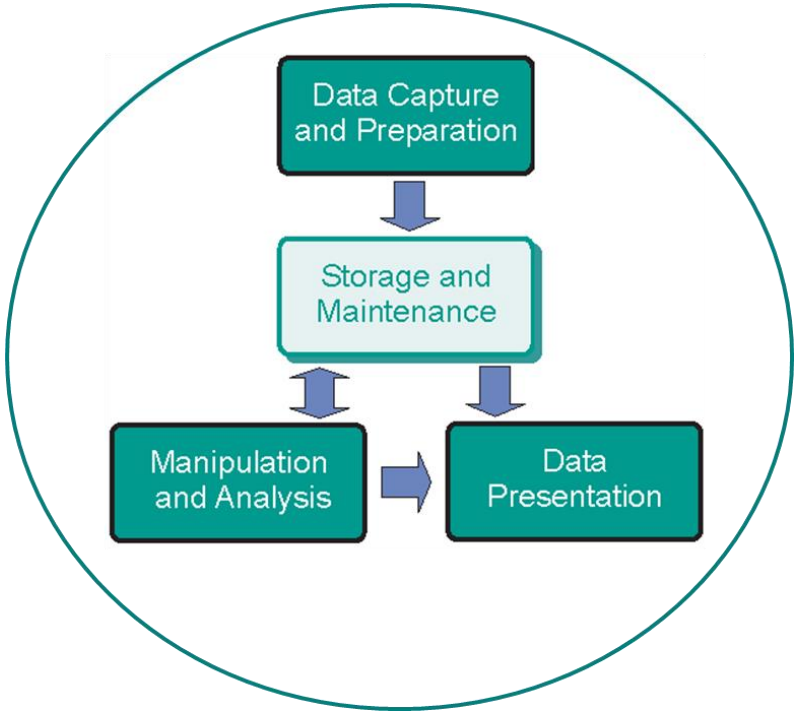
# Data, Dataset, Database, DBMS and DB system

- Data
  - Is a resource held on paper or in digital format that serves to record or administer some facts and description of phenomena of interest.
- Data set (or dataset):
  - A homogeneous collection of data normally describing a single kind of phenomenon
- Database
  - A *collection of interrelated data sets* properly *structured* by means of, and stored through a DBMS.

# Data, Dataset, Database, DBMS and DB system

- Database management system (DBMS)
  - A *software package* that is designed for the purpose of managing databases. This means, DBMS allows to set-up, maintain and exploit one or more databases.
  - DBMS uses a *data model* that is a collection of data structuring primitives, rules of how to structure, and mechanisms to handle the data in a database.




GIS Specialisten
consultancy & detachering

# Database management system (DBMS)

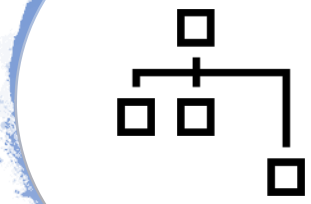

- Supports the storage and manipulation of very large data sets.
- Can be instructed to guard over data correctness.
- Supports the concurrent use of the same data set by many users.
- Provides a high-level, declarative query language.
- Includes data backup and recovery functions to ensure data availability at all times.
- Allows the control of data redundancy.
- Supports the use of data model.

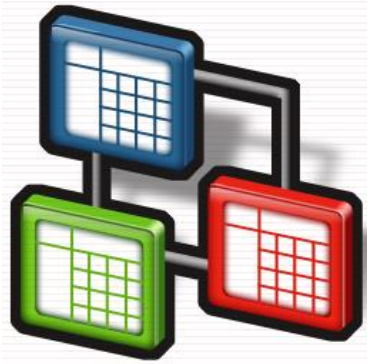
# Data Model

A DBMS supports the use of a *data model*.

- Data model is an integrated collection of:
  - Data structuring primitives,
  - Rules of how to structure, and
  - Mechanisms to handle the data in a database.

In other words, a data model is a *toolbox* that allows us to *create/define a* database structure and *manipulate* the data stored in it.





GIS Specialisten
consultancy & detachering

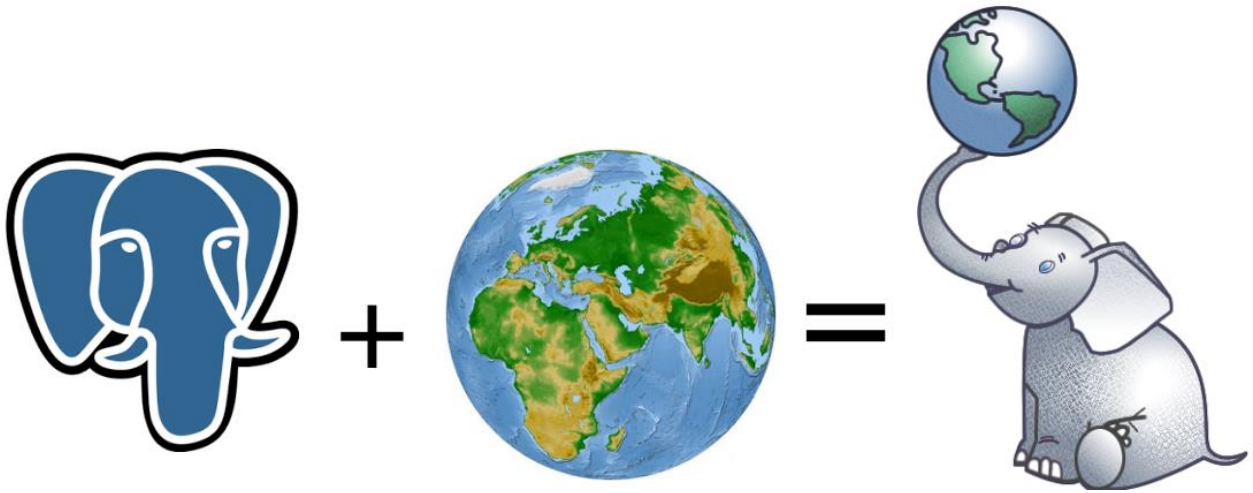
# PostgreSQL and PostGIS


- **PostgreSQL** is a database management system
- It has a Server-Client architecture
- It runs on every main operating system
- It ensures purely **ACID** transactions:
  - **A**tomicity - (a transaction only knows two states – unstarted or finished)
  - **C**onsistency - (a transaction always complies with the constrains of the DB)
  - **I**solation - (how concurrent operations became visible to each other)
  - **D**urability - (after a transactions is terminated its effects remain)

GIS Specialisten
consultancy & detachering

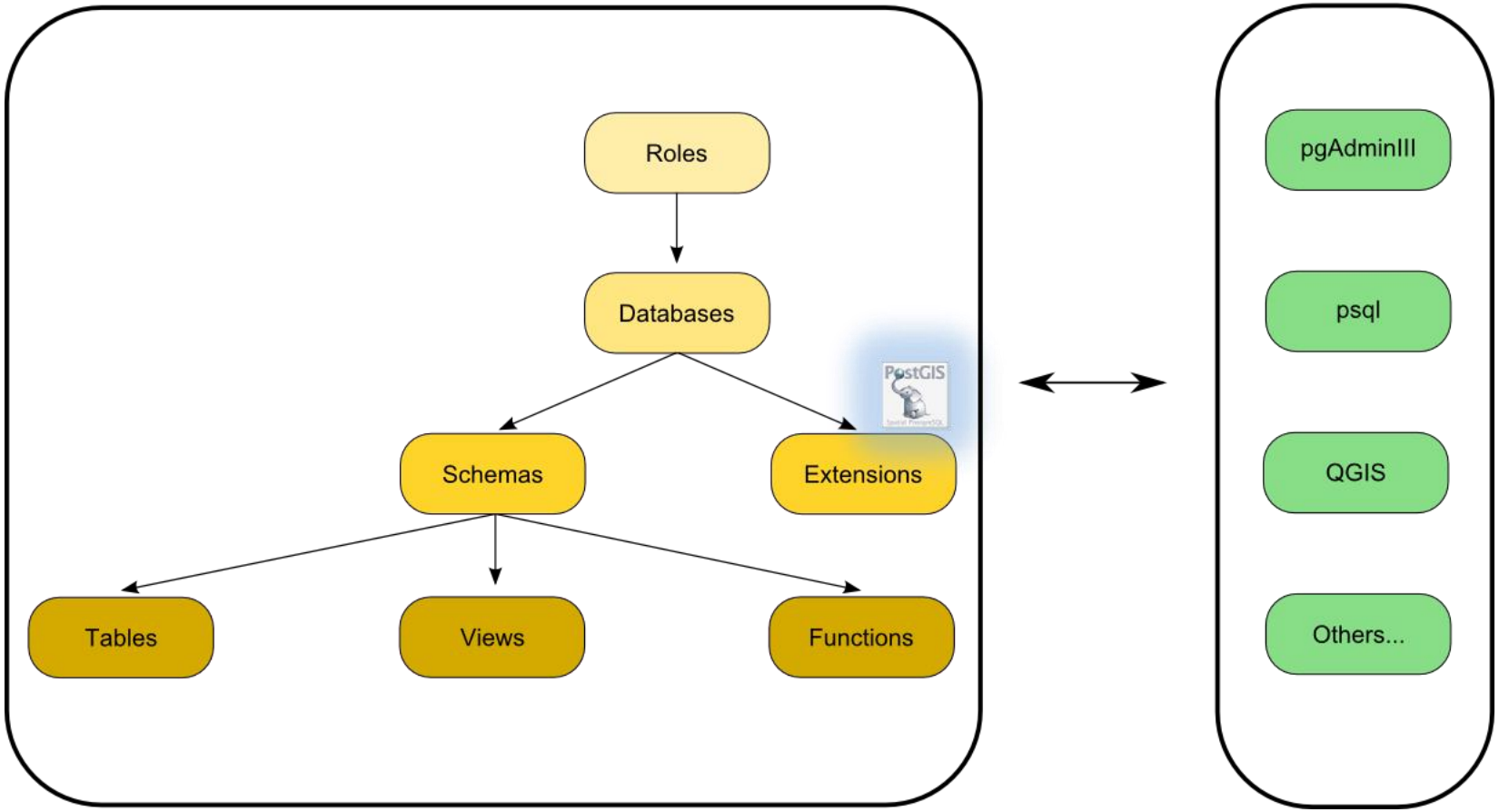
# PostgreSQL and PostGIS

- A spatial database allows users to store, query and manipulate collections of ***spatial data***.
- Spatial data is stored in as a geometry data type which complies with OGC standards.
- Spatial database is not constrained by the need to present data visually
- PostGIS is an extension that enable PostgreSQL with spatial data handling



GIS Specialisten
consultancy & detachering

# PostgreSQL and PostGIS - System Architecture




GIS Specialisten
consultancy & detachering

GIS Specialisten
consultancy & detachering

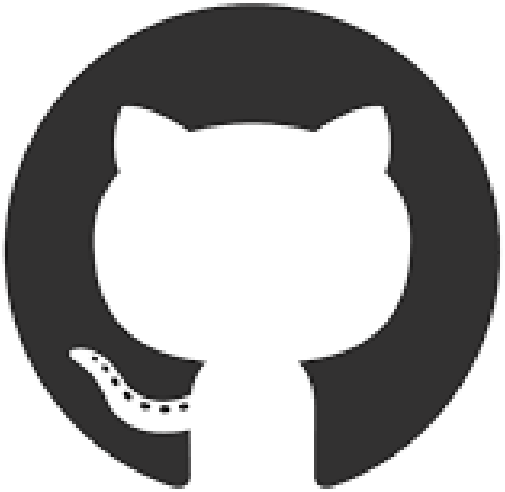
# Let’s get started!

# Materials and software

- Access the course at Github

https://github.com/paryapasha/Cursus_PostGIS_n1

- OpenSource software
  - PosgreSQL + PostGIS + pgAdmin
  - QGIS



GIS Specialisten
consultancy & detachering

# Query formulation

Relation/ Table

| | id [PK] bigint | geom geometry | dimune_code character varying (254) | neighborhood character varying (254) | municipality character varying (254) | district character varying (254) |
|---|---|---|---|---|---|---|
| 1 | 1 | 0106000020B30... | 130103 | Ansiães | AMARANTE | PORTO |
| 2 | 2 | 0106000020B30... | 130107 | Candemil | AMARANTE | PORTO |
| 3 | 3 | 0106000020B30... | 130112 | Fregim | AMARANTE | PORTO |
| 4 | 4 | 0106000020B30... | 130115 | Fridão | AMARANTE | PORTO |
| 5 | 21 | 0106000020B30... | 130143 | União das freguesias de Bu... | AMARANTE | PORTO |
| 6 | 5 | 0106000020B30... | 130117 | Gondar | AMARANTE | PORTO |
| 7 | 6 | 0106000020B30... | 130118 | Jazente | AMARANTE | PORTO |
| 8 | 7 | 0106000020B30... | 130119 | Lomba | AMARANTE | PORTO |
| 9 | 8 | 0106000020B30... | 130120 | Louredo | AMARANTE | PORTO |
| 10 | 9 | 0106000020B30... | 130121 | Lufrei | AMARANTE | PORTO |
| 11 | 10 | 0106000020B30... | 130123 | Mancelos | AMARANTE | PORTO |

Tuples/Records/Rows

Attributes/Columns/Fields

GIS Specialisten
consultancy & detachering

# Query formulation
## Simple Method of query definition

**Steps:**

1. INPUT RELATION
   - On which relation is the query based?
   - What will be the tuple variable?
2. SELECTION CONDITION
   - What is the condition that the selected tuple for the output must fulfill?
3. OUTPUT RELATION
   - What are the attributes in the output?

**In SQL:**

```
SELECT output tuple definition
(step 3)
FROM input tuple declaration
(step 1)
WHERE selection condition
(step 2)
```

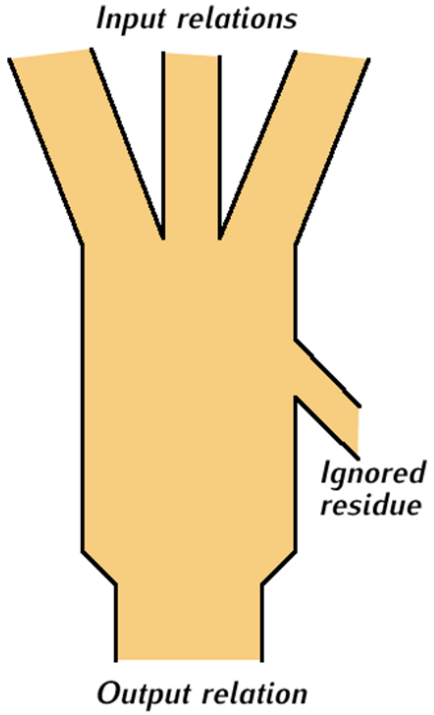
GIS Specialisten
consultancy & detachering

# JSP queries
## Join, Select, Project

Very often, to extract required data from a database, there is a need to combine data from two or more relations in one query. In such query, selection condition depends on a relationship of relations involved in the query.

*Phases of JSP query:*

1. Cartesian product of input relations
   - *Which relations are involved in a query?*
2. Tuple selection
   - *What is the join condition?*
   - *What is the selection condition?*
3. Output tuple creation
   - *What attributes will the output relation have?*




GIS Specialisten
consultancy & detachering

# Summary queries

- A ***summary query*** is a query that infers statistical information from the tuples in the database.

- It involves ***aggregate functions***: functions that operate on a set (bag) of tuples and produce a single value. Examples are:
  - counting tuples: *count*
  - summing up attribute values: *sum*
  - computing averages, minimum, maximum values from existing attribute values: *avg, min, max*
  - computing standard deviations, variances: *sdev*, *var*

```
SELECT attributename, aggregate(*)
FROM tablename
WHERE selection condition
GROUP BY attributename
HAVING groupselection condition
```

GIS Specialisten
consultancy & detachering

# Spatial databases

# Spatial data handling

What if you want to know…

- What is the area of a district?
- Do any of the countries in my table share a boundary?
- Which railroads pass through a city?
- How many schools are within a distance of 5 km to your house?

GIS Specialisten
consultancy & detachering

# Spatially enabling Postgres

What happens when we enable PostGIS in our PostgreSQL database?

- Hundreds of *spatial functions* became available within the database;

- *Metadata views* are created:
  - geography columns
  - geometry columns
  - raster columns
  - raster overviews

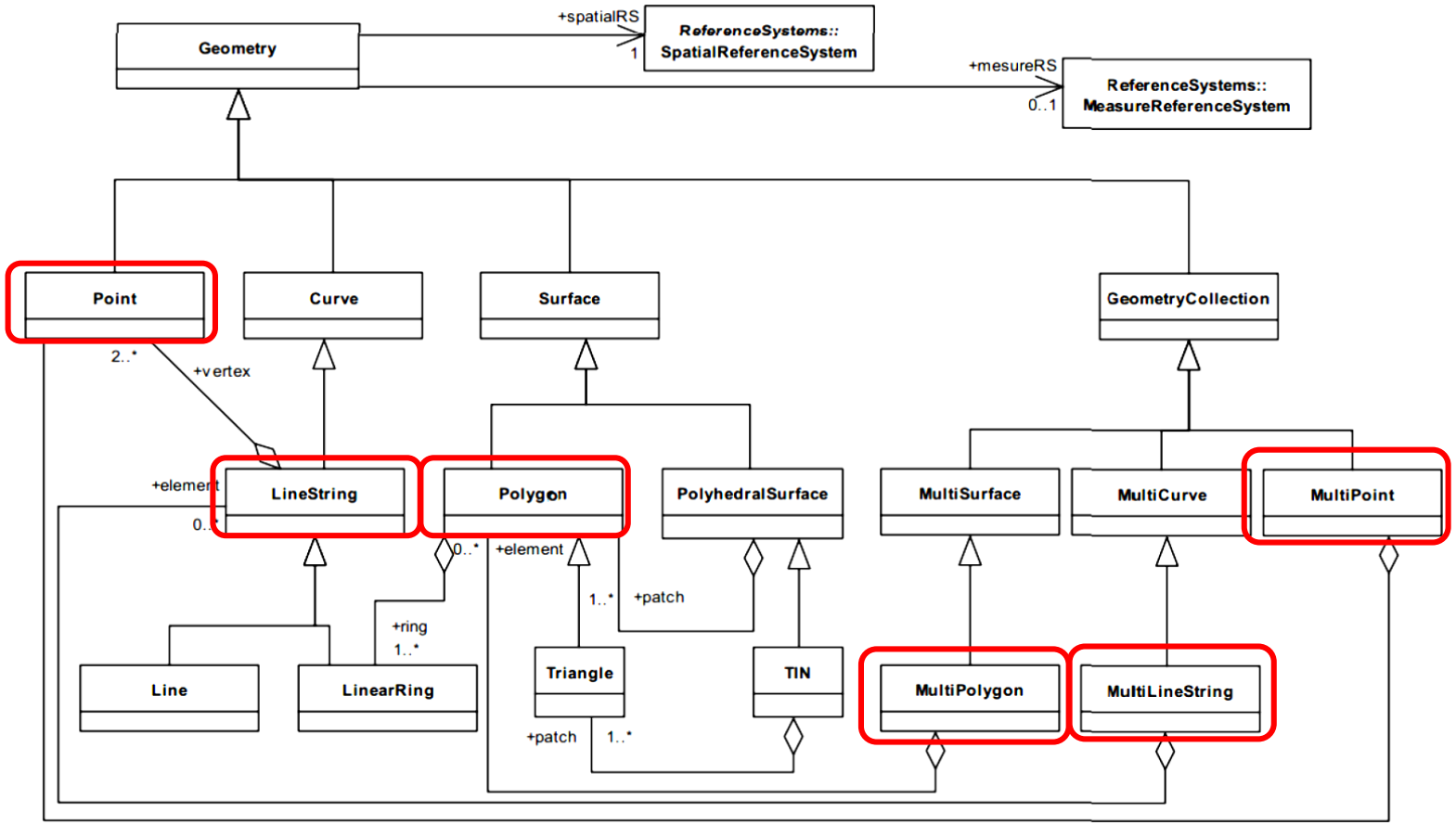
- *Metadata table* is created:
  - spatial_ref_sys (**SRID**s are the primary key to this table)

❖ All these metadata views and table are necessary in order to add support for OGC Simple Feature Model

GIS Specialisten
consultancy & detachering

# OGC Simple Features Specification

- The Simple Feature Specification (SFS) is a standard from OGC on how data should be stored.
- It also  defines functions for accessing, operating, and constructing data.

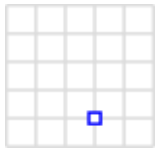



GIS Specialisten
consultancy & detachering

# OGC Simple Features Specification
## Geometry types

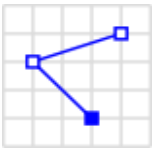
- Point

  POINT (30 10)



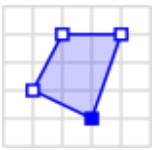
- Linestring

  LINESTRING (30 10, 10 30, 40 40)



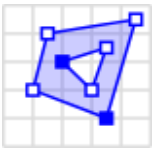
- Polygon

  POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))



- Polygon (with “hole”)

  POLYGON ((35 10, 45 45, 15 40, 10 20, 35 10), (20 30, 35 35, 30 20, 20 30))

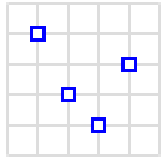


GIS Specialisten
consultancy & detachering

# OGC Simple Features Specification
## Geometry types

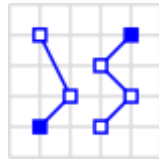
- MultiPoint

  MULTIPOINT ((10 40), (40 30), (20 20), (30 10))



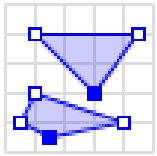
- MultiLinestring

  MULTILINESTRING ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))



- MultiPolygon

  MULTIPOLYGON (((30 20, 45 40, 10 40, 30 20)), ((15 5, 40 10, 10 20, 5 10, 15 5)))

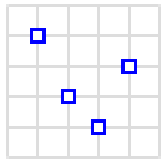


GIS Specialisten
consultancy & detachering

# OGC Simple Features Specification
## Geometry types

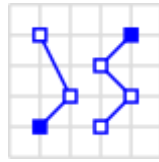
- MultiPoint

  MULTIPOINT ((10 40), (40 30), (20 20), (30 10))



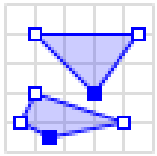
- MultiLinestring

  MULTILINESTRING ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))



- MultiPolygon

  MULTIPOLYGON (((30 20, 45 40, 10 40, 30 20)), ((15 5, 40 10, 10 20, 5 10, 15 5)))



GIS Specialisten
consultancy & detachering

# Some famous (!) Spatial functions

ST_Length

```
SELECT  id, st_length(geom)
FROM    rivers
WHERE   rivername = 'IJssel'
```

output: length of the line/multiline

ST_Area

```
SELECT  st_area(s.geom)
FROM    squares as s
```

output: area of the polygon/multipolygon

ST_Centroid

```
SELECT  b.id, st_centroid (b.geom)
FROM    buildings as b
```

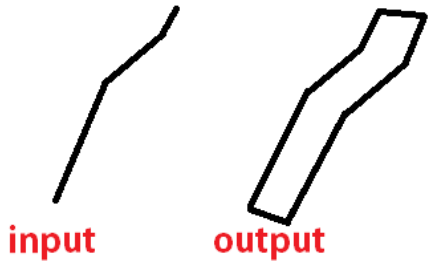
output: a point geometry, centroid of input geometry

GIS Specialisten
consultancy & detachering

# Some famous (!) Spatial functions

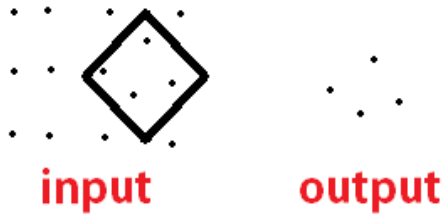
ST_Buffer

```
SELECT  id, st_buffer(geom, 100)
FROM    rivers
WHERE   rivername = 'Tejo'
```




ST_Within (check st_dwithin and st_contains)

```
SELECT  l.id, st_within(l.geom, s.geom)
FROM    lamps as l, squares as s
```




ST_Intersection (check st_intersects)

```
SELECT  b.id, st_intersection(b.geom, s.geom)
FROM    buildings as b, squares as s
```


GIS Specialisten
consultancy & detachering

# More Spatial function to explore

- `ST_transform`
- `ST_distance`
- `ST_dwithin`
- `ST_crosses`
- `ST_overlaps`
- `ST_disjoint`
- `ST_contains`
- …

In addition to the functionality, try to understand the number and type of input features and the type of output.


GIS Specialisten
consultancy & detachering

GIS Specialisten
consultancy & detachering

# On DB design and integrity

# Creating a table

```
CREATE TABLE vectors.porto_neighborhood
(
    id bigint NOT NULL,
    geom geometry(MultiPolygon,3763),
    dimune_code character varying(254) ,
    neighborhood character varying(254) ,
    municipality character varying(254) ,
    district character varying(254),
    area_ha character varying(254) ,
    neighborhood_shortname character varying(254) ,
    CONSTRAINT porto_neighborhood_pkey PRIMARY KEY (id)
)
```

GIS Specialisten
consultancy & detachering

# Creating a table
## Primary Key

- The key of a relation R is a set K of R's attributes such that:

1. It is **unique** – there are no two distinct tuples of K that have the same attribute values.
2. It is **minimal** – there is no proper subset of K that is unique.
3. And it cannot be **NULL**

If there is more than one key for R, the database designer chooses one of them to become the primary key. The other remain candidate keys.

# Referential integrity
## Foreign Key

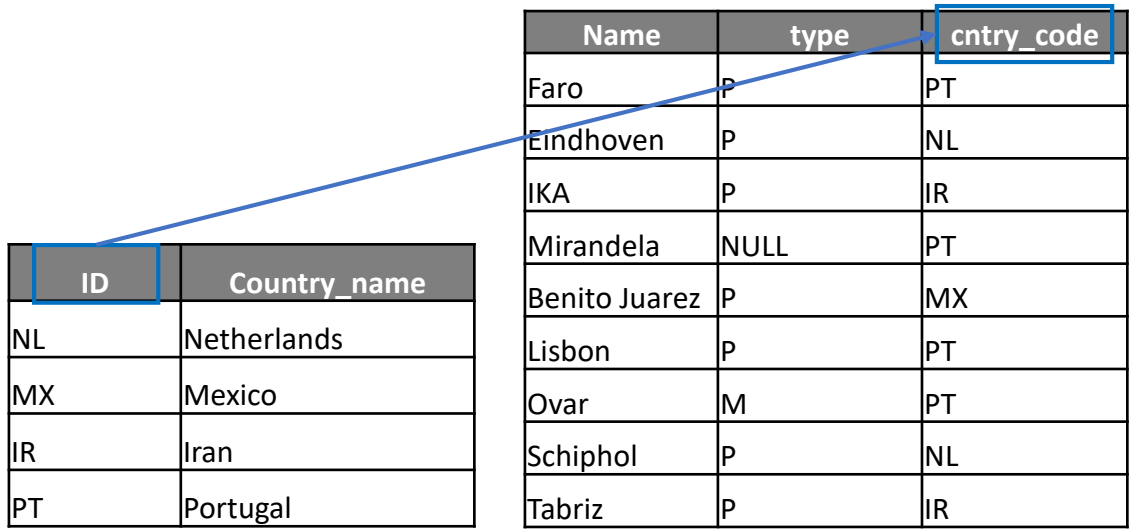
A foreign key is a set of attributes that is used to refer to a tuple in another relation.

- It must correspond with a primary key value in the second relation.
- A foreign key behaves like a 'logical pointer'.

```
CREATE TABLE country_codes
(   ID character varying(254),
    country_name character varying(254),

PRIMARY KEY (ID),
FOREIGN KEY (ID) REFERENCES airports(cntry_code)
ON DELETE CASCADE
ON UPDATE SET DEFAULT

)
```



| ID | Country_name |
|---|---|
| NL | Netherlands |
| MX | Mexico |
| IR | Iran |
| PT | Portugal |

country_codes

| Name | type | cntry_code |
|---|---|---|
| Faro | P | PT |
| Eindhoven | P | NL |
| IKA | P | IR |
| Mirandela | NULL | PT |
| Benito Juarez | P | MX |
| Lisbon | P | PT |
| Ovar | M | PT |
| Schiphol | P | NL |
| Tabriz | P | IR |

airports

GIS Specialisten
consultancy & detachering

# Create Views

**View vs. Table**

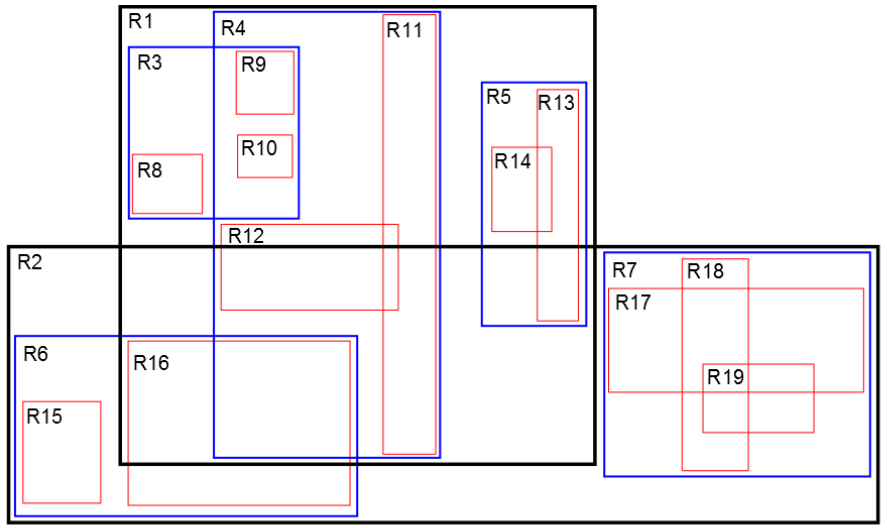
- A view is a stored query in a form of a table which is not physically materialized;
- A view table can be queried but cannot be directly updated;
- A view changes every time the underlying data changes;
- Views are therefore dynamic;

```
CREATE VIEW expansion AS
        SELECT id, area, st_buffer(geom, 10)
        FROM landplot;
```

GIS Specialisten
consultancy & detachering

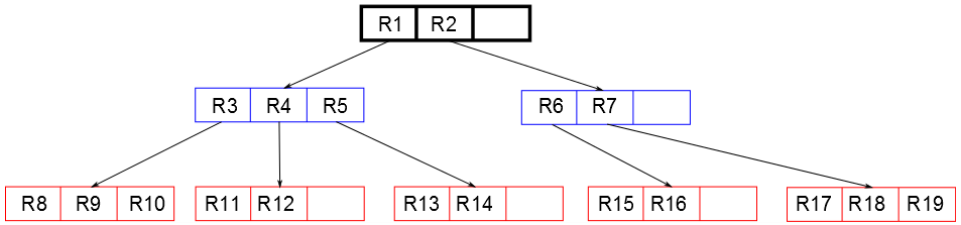
# Spatial indexing

- An important feature of spatial databases is the support of Spatial Indexes.
- PostGIS uses GIST Indexation.




- Can simply be created on a table

```
CREATE INDEX index_name ON tablename
USING GIST (geomcolumnname);
```




GIS Specialisten
consultancy & detachering

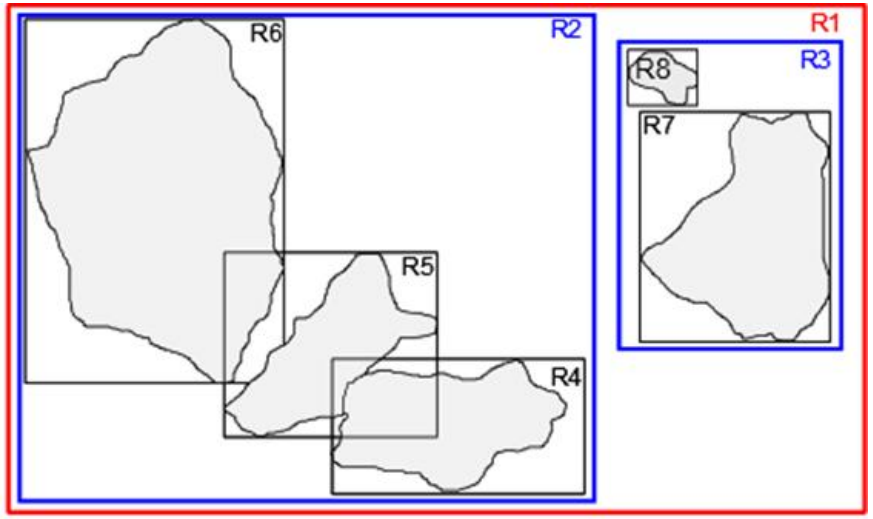
# Spatial indexing

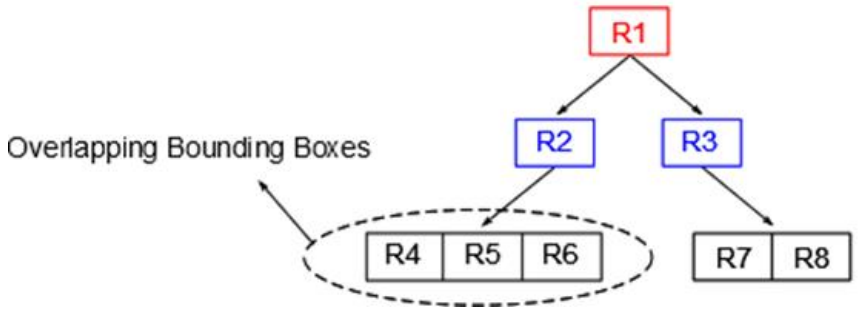
- Which features intersect (st_intersects)?

GIS Specialisten
consultancy & detachering

# Spatial indexing

- First the database starts by identifying the overlapping bounding boxes:




GIS Specialisten
consultancy & detachering

# Spatial indexing

- Then the actual topological operation takes place on the geometries whose bounding boxes overlap, giving us the result

- Most of the commonly used functions in PostGIS (ST_Contains, ST_Intersects, ST_DWithin, etc) include an index filter automatically. But some functions (e.g., ST_Relate) do not include and index filter.
- To do a bounding-box search using the index (and no filtering), make use of the **&&** operator.
- For geometries, the **&&** operator means “bounding boxes overlap or touch” in the same way that for number the **=** operator means “values are the same”.

GIS Specialisten
consultancy & detachering

# How to continue…



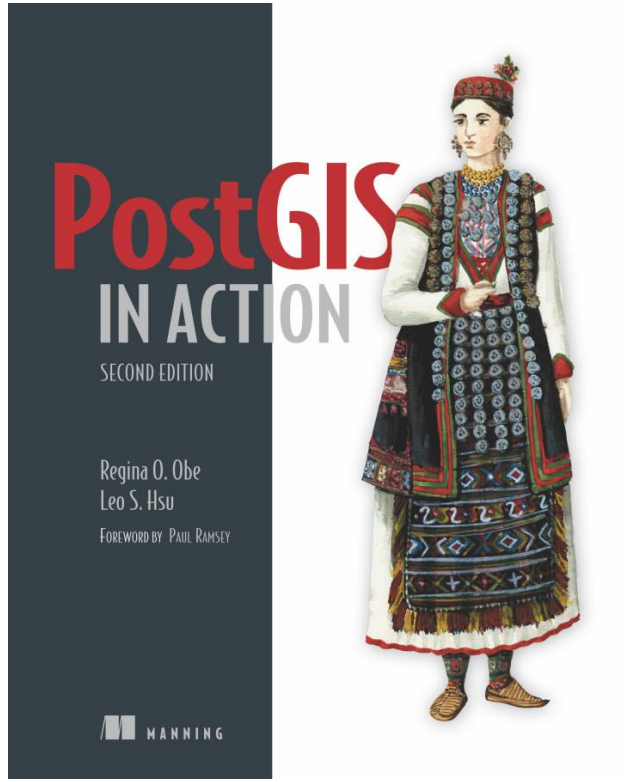
www.gisspecialisten.nl | Ringwade 41 | 3439 LM Nieuwegein | +31 (0)85 7 325 825

# Continue

- PostGIS in Action 2nd edition
  - Find our all about it here
  - Download the code and the data here
  - Purchase here

- Simple Feature Access
  - Part 1: Common Architecture (pg 25-40)
  - Part 2: SQL Option

- Manuals
  - PostgreSQL
  - PostGIS




OGC®
Making location count.

GIS Specialisten
consultancy & detachering

GIS Specialisten
consultancy & detachering

**Lesley Porter**
Office Manager
**M:** +31(0)6 11 12 07 47
**E:** lesley@gisspecialisten.nl

**Parya Pasha Zadeh**
Operations Manager
**M:** +31(0)6 83 84 72 95
**E:** parya@gisspecialisten.nl

# SPATIAL VS NON-SPATIAL QUERIES

JOINS

neigbourhood (
  neigh_name *String*,
  pop *Double*,
  geom *(Polygon,28992)*
)

bus_stop(
  stop_name *String*,
  geom *(Point,28992)*
)

# Spatial Databases

```
SELECT bus_stop.stop_name, neighborhoods.neigh_name,
       neighborhoods.pop
FROM neighborhoods JOIN bus_stop
ON   ST_Intersects(neighborhoods.geom, bus_stop.geom)
```

SELECT *
FROM table1, table2
WHERE  join condition