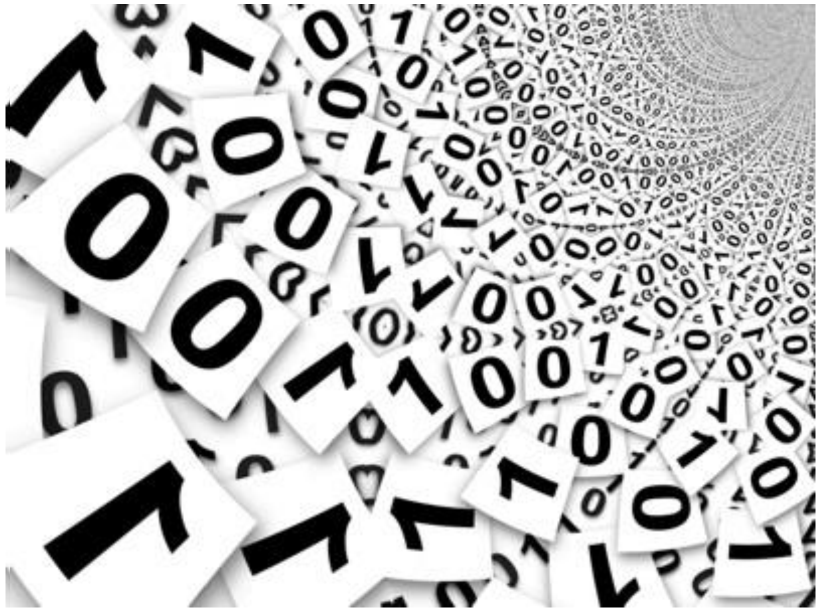
Geo-Databases

Frequently Asked Questions (FAQ)

Institute for Geodesy and Geoinformation Science
Technische Universität Berlin



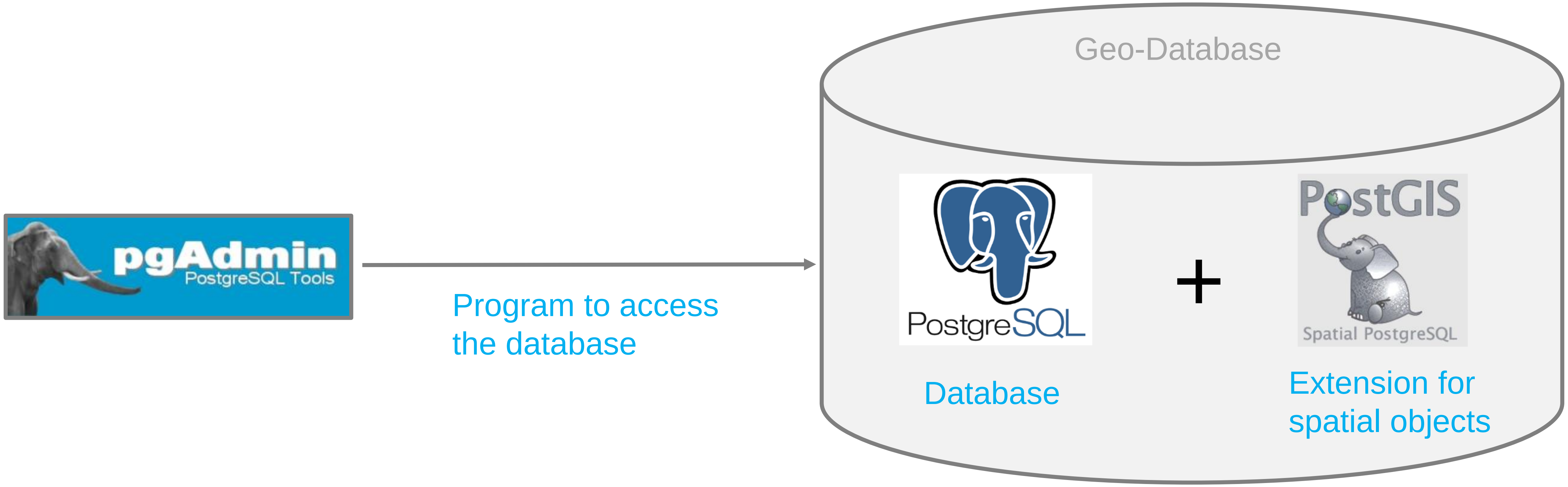
PostgreSQL, PostGIS and pgAdmin

- **PostgreSQL** is a open source object-relational database system. The purpose of a database is to store and retrieve related information.

- **pgAdmin** is a PostgreSQL Tool to get access to the database. Any data querying and manipulation can be done using pgAdmin.

- **PostGIS** is a spatial database extender for **PostgreSQL** database. It adds support for geographic objects allowing location queries to be run in SQL.

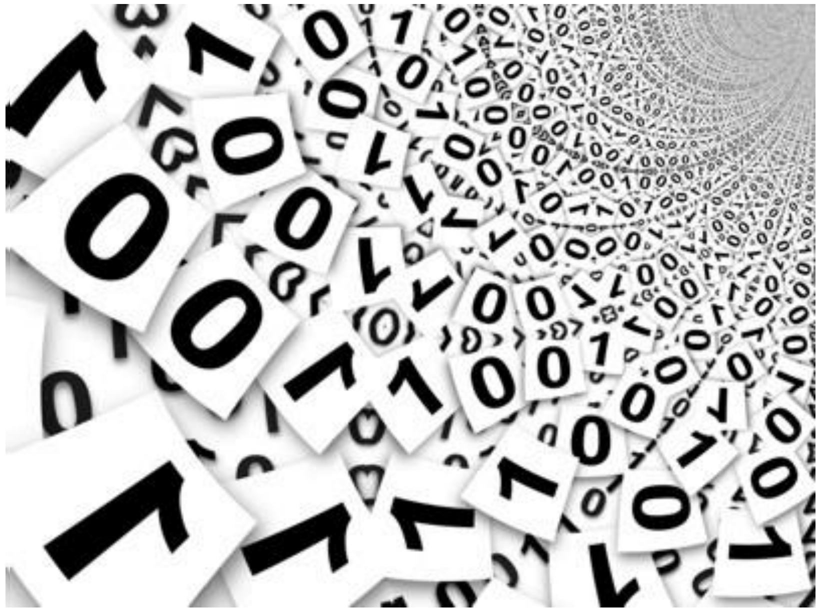
PostgreSQL, PostGIS and pgAdmin



Geo-Databases

Exercise 3: SQL Joins

Institute for Geodesy and Geoinformation Science
Technische Universität Berlin



SQL Joins

A SQL JOIN is used to combine rows from two or more tables, based on a common field between them. The most common type of join is: **SQL INNER JOIN (simple join)**. An SQL INNER JOIN returns all rows from multiple tables where the join condition is met.

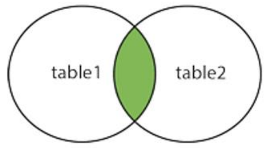
There are different SQL JOINs you can use:

- **INNER JOIN**: Returns all rows when there is at least one match in BOTH tables
- **LEFT JOIN**: Return all rows from the left table, and the matched rows from the right table
- **RIGHT JOIN**: Return all rows from the right table, and the matched rows from the left table
- **FULL JOIN**: Return all rows when there is a match in ONE of the tables

Source: http://www.w3schools.com/sql/sql_join.asp

SQL JOIN TYPES

INNER JOIN

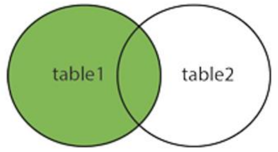


```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name=table2.column_name;
```

or:

```
SELECT column_name(s)
FROM table1
JOIN table2
ON table1.column_name=table2.column_name;
```

LEFT JOIN

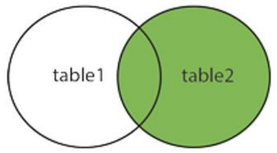


```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name=table2.column_name;
```

or:

```
SELECT column_name(s)
FROM table1
LEFT OUTER JOIN table2
ON table1.column_name=table2.column_name;
```

RIGHT JOIN

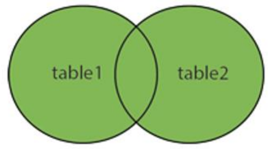


```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name=table2.column_name;
```

or:

```
SELECT column_name(s)
FROM table1
RIGHT OUTER JOIN table2
ON table1.column_name=table2.column_name;
```

FULL OUTER JOIN



```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name=table2.column_name;
```

Source: http://www.w3schools.com/sql/sql_join.asp

Improve your skills in SQL

Create the tables LECTURER (attributes: ID_lecturer, lastname, firstname, ID_supervisor) and CLASSES (attributes: ID_classes, class, lecturer_ID). Fulfill the following statements (you will also find the .sql-file in ISIS):

```
INSERT INTO lecturer  (ID_lecturer, lastname, firstname) VALUES (1, 'Neitzel', 'Frank');
INSERT INTO lecturer  (ID_lecturer, lastname, firstname) VALUES (2, 'Oberst', 'Juergen');
INSERT INTO lecturer  (ID_lecturer, lastname, firstname) VALUES (3, ‘Galas', ‘Roman');
INSERT INTO lecturer  (ID_lecturer, lastname, firstname) VALUES (4, 'Kada', 'Martin');
INSERT INTO lecturer   VALUES (5, 'Weisbrich', 'Sven', 1);
INSERT INTO lecturer   VALUES (6, 'Wujanz', 'Daniel', 1);
INSERT INTO lecturer   VALUES (7, 'Glaeser', 'Philipp', 2);
INSERT INTO lecturer   VALUES (8, 'Becker', 'Thomas', 4);
INSERT INTO lecturer   VALUES (9, 'Koenig', 'Gerhard', 4);
```

Using Copy and Paste is recommended

Improve your skills in SQL

```
INSERT INTO classes  VALUES (19, 'Satellite Geodesy', 2);
INSERT INTO classes  VALUES (21, 'GNSS', 3);
INSERT INTO classes  VALUES (3, 'Adjustment', 1);
INSERT INTO classes  VALUES (24, 'Engineering Surveys', 5);
INSERT INTO classes  VALUES (51, 'Geo Databases', 9);
INSERT INTO classes  VALUES (16, 'Statistic Tests', 1);
INSERT INTO classes  VALUES (72, 'GIT', 4);
INSERT INTO classes  VALUES (73, 'GIT', 8);
INSERT INTO classes  (ID_classes, class) VALUES (8, 'Geostatistics');
```

Improve your skills in SQL

Apply the different JOIN operations:

a: Select the person teaching GIT

b: Select the lecture that isn't given this semester

c: Select the person who has no lectures

d: Focus on the lecturer table and find out which persons are supervised by which professors (--->self join)

2. Homework (Deadline Nov 16, 11.59pm – please upload your homework to ISIS)

H2.1: Create a new table COUNTRY containing the attributes: ID_COUNTRY, COUNTRY, CAPITAL (e.g. 1, ‘Ghana’, ‘Accra’)

H2.2: Transfer the content of the column COUNTRY from STUDENTS_16 into the new table and fill the corresponding columns. Avoid redundant data.

H2.3: To define a unique identifier create a database object (SEQUENCE) for generating unique numbers, and use them as new **ID**-values as follows:

```
CREATE SEQUENCE country_seq INCREMENT BY 1
START WITH 1 MINVALUE 1 CACHE 100;
UPDATE country
SET ID_country = country_seq.nextval;
```

2. Homework (Deadline Nov 16, 11.59pm – please upload your homework to ISIS)

H2.4: Change the schema of table STUDENTS_16 by adding a new attribute named COUNTRY_ID. Establish the relationship of both tables referencing ID_COUNTRY and COUNTRY_ID.

H2.5: Is the column COUNTRY in STUDENTS_16 still necessary? Explain your answer.