

Geo875 | FS25  
University of Zürich

## **Introduction to Spatial Databases**

**Rolf Meile**

Eidg. Forschungsanstalt für Wald, Schnee und Landschaft (WSL)  
Swiss Federal Institute for Forest, Snow and Landscape Research

**Esra Suel**

Dept. of Geography, University of Zürich

## The Geo875-Team



**Rolf Meile**, Database- and GIS-Specialist, WSL

**CV:** M. Sc. Biology (UZH), Eidg. Forschungsanstalt für Wald, Schnee und Landschaft (WSL), Monitoring Institute (NGO), Cityline AG (private company), self-employed

**Interests:** spatio-temporal data, data quality, dealing with large datasets, calculating chains, fine cooking and eating, growing plants from seeds

**Contact:** [rolf.meile@wsl.ch](mailto:rolf.meile@wsl.ch)



**Esra Suel**, Professor of Urban Analytics, GIUZ

**CV:** PhD Transportation and Urban Systems, Civil and Environmental Engineering (Imperial College London)

**Interests:** urban analytics, transportation/mobility, housing, built environment

**Contact:** [esra.suel@geo.uzh.ch](mailto:esra.suel@geo.uzh.ch), Y25 L90

## Real world...

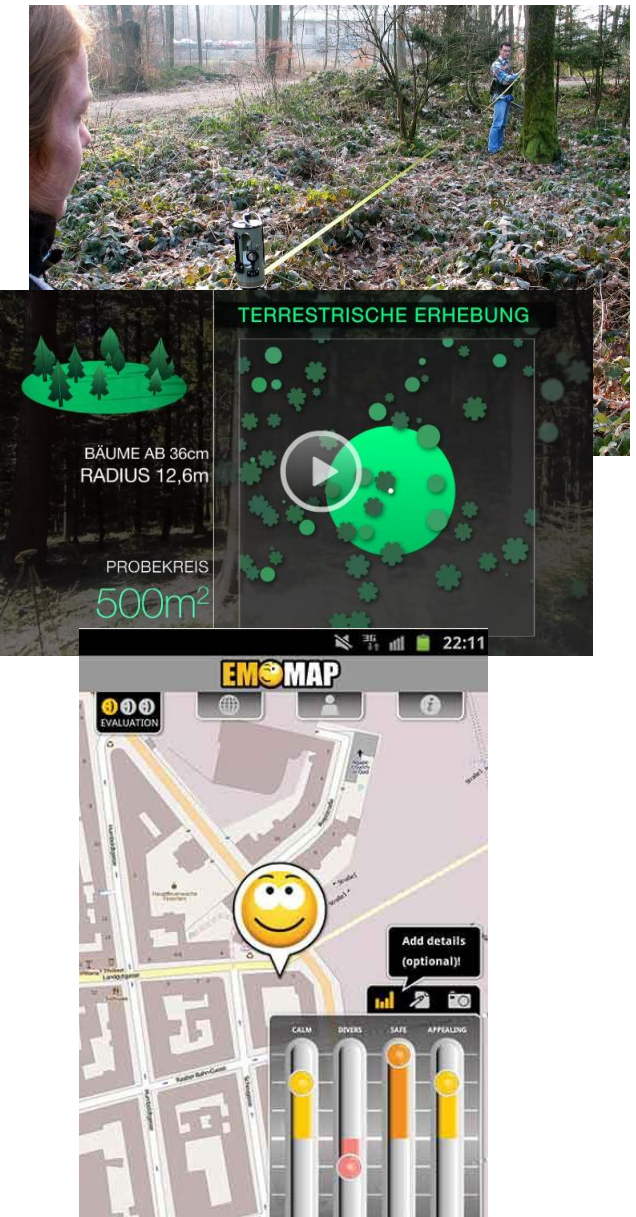
### The Swiss Forest Inventory

- 80'000 trees measured along with 300 attributes
- Stored in a spatially enabled database
- Analysis of the thematic and spatial characteristics of the forest as well as geostatistic;

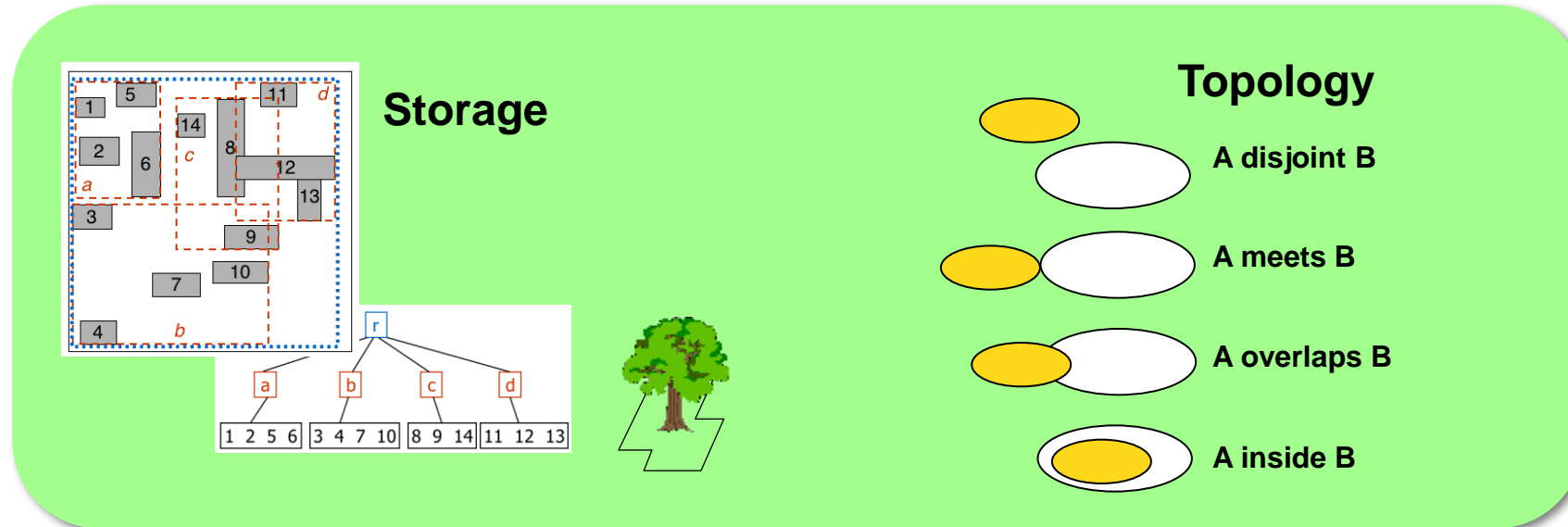
### Affective Route Planning

- Integrating and aggregating crowd-sourced datasets
- User based input drives selection of routes
- Routing algorithms suggest 'best' route considering emotions and mood

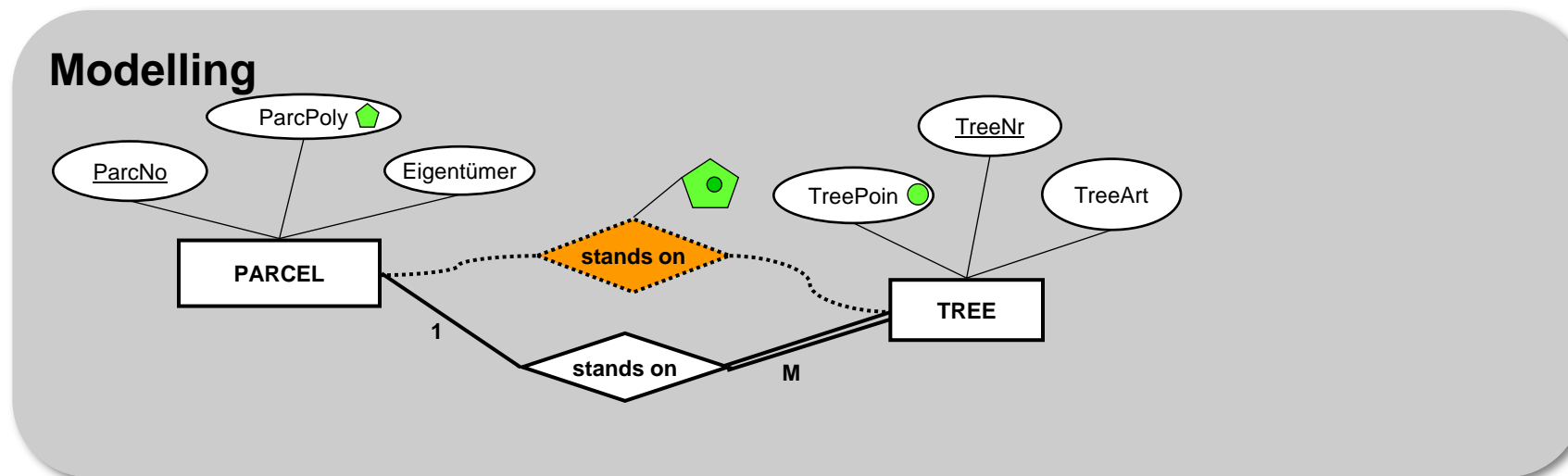
→ [https://www.lfi.ch/lfi/lfi\\_film.php](https://www.lfi.ch/lfi/lfi_film.php)



# Enabling space in databases



**Geo875:**  
Spatial  
databases



**Geo874:**  
Introduction  
to databases

# Learning Objectives Course geo875

- **“Introduction to spatial databases”** – understand how spatially enabled databases help us better manage spatial data in large, complex projects.
- Gain proficiency in the fundamentals of spatial data storage and querying in relational DBMS.
- Master the skills of spatial data modelling and apply them in a project.
- Understand and correctly apply spatial data types and spatial data storage structures (topology) for the efficient storage and querying of spatial data.
- Understand spatial indices in spatial databases to achieve efficient storage, modification and retrieval of geometries and rasters.
- Understand how this principles apply beyond relational DBMS when using web map and geometry servers.



# What now?

L1	<b>Introduction to spatial databases: definitions and concepts</b>	"What are spatial databases and how are they different to non-spatial DBs?"
L2	<b>Spatial data types</b>	"How do we store, query, manipulate, and efficiently manage space in DBs?"
L3	<b>Topology</b>	"How can we ensure spatial integrity of our spatial data?"
L4	<b>Spatial queries</b>	"How can we define, modify and query a DB?"
L5	<b>Spatial indices</b>	"How to speed up answers from a spatial database?"
L6	<b>Raster data Advanced topics Recap</b>	"How can we organise huge or small amount of raster based data?" Recap some interesting bit and pieces on advances in spatial DBs
L7	-	Project delivery

# Housekeeping



# Housekeeping

## Online contents:

- Lecture notes
  - Week by week
  - Usually on Thursday noon
- Exercise handouts
- Project description
- FAQ (Problems, Tips and Tricks, Resources)
  1. FAQ-site, project-site
  2. Colleagues
  3. Finally, email us
- Learning objectives and course deliverables

<https://www.geo.uzh.ch/microsite/geo875/index.html>

Home Geo875

[Program](#)

[Project](#)

[FAQ](#)

[Resources](#)

## Spatial Databases Geo875

Lectures, exercises and practical sessions in Fall Semester 2025

### Registration and booking

The course is fully booked for Fall Semester 2025.

### Content and objectives

In this course we extend the knowledge of data management in relational databases acquired in Geo874 (Introduction to Databases) with a focus on spatial data management, usage and analysis. We will introduce spatial indexes, topological data models and spatial queries. Open source software (PostGIS) will be used as an example.

You will learn about the handling of both geometrical and attribute variables in an efficient manner. We will then extend this knowledge to handling of georeferenced raster data, considerations for the storage of topological data, intricacies of spatial indexing and the optimization of topological analysis through spatial indexes. The course will cover the extensions to the SQL query language that relate to spatial data handling.

In parallel to the theoretical stream, practical skills will be acquired by dealing with real-world data step-by-step from the creation of relational data models, through data import to querying and spatial data analysis. The practical part will consist of exercises relating to selected parts of lectures and an accompanying [group project](#).


# We will use the following colour code

V1 | Einführung räumliche Datenbanken | Geo451 | Räumliche Datenbanken | HS09  
39 | Geodatenbanksysteme / Räumliche Datenbanksysteme | P. Laube, R. Meile, Uni Zürich

### Komponenten eines Räumliche Datenbanksysteme

RDBS bestehen im wesentlichen aus folgenden Komponenten:

1. Schemata, die räumliche Datentypen zur Verfügung stellen z.B. SDO\_GEOMETRY
2. Räumliche Indizierungsmechanismen (R-Baum)
3. Sammlung von Funktionen um räumliche Operationen auszuführen z.B. Spatial-Join- und area-of-interest-Queries
4. Administrative Tools

Main contents (just basics, take notes!) 

V1 | Einführung räumliche Datenbanken | Geo451 | Räumliche Datenbanken | HS09  
34 | Housekeeping | P. Laube, R. Meile, Uni Zürich

### Relationale Algebra

Formale Sprache zur Spezifizierung grundlegender Suchabfragen

- Mengenoperationen
- SELECT-Operation
- Joins

Abbildung 8.11: Die Mengenoperationen UNION, INTERSECTION und DIFFERENCE: (a) Zwei unkompatible Relationen: (b) Student ∪ Instruktor; (c) Student ∩ Instruktor; (d) Student - Instruktor; (e) Instruktor - Student.

STUDENT	FN	LN
1	Susan	Walt
2	Ronald	Shah
3	Johnny	Keller
4	Barbara	Jones
5	Jenny	Ford
6	Ernest	Olson

INSTRUCTOR	FN	LN
1	John	Smith
2	Ronald	Brown
3	Scott	Wheeler
4	Francis	Johnson
5	Nelson	Shaw

(b) UNION: Student ∪ Instruktor

(c) INTERSECTION: Student ∩ Instruktor

(d) DIFFERENCE: Student - Instruktor

(e) DIFFERENCE: Instruktor - Student

On your own...

V1 | Einführung räumliche Datenbanken | Geo451 | Räumliche Datenbanken | HS09  
13 | P. Laube, R. Meile, Uni Zürich

### Lernziele 1.1

- ✓ Sie wissen was Datenbanksystemen (DBS) sind
  - ✓ Sie können verschiedene Anwendungen von DBS aufzählen.
  - ✓ Sie können die Vor- und Nachteile eines DBS gegenüber einer file-basierten Datenhaltung diskutieren.
- ✓ Sie wissen was Datenbankmanagementsysteme (DBMS) sind.
  - ✓ Sie kennen die Grobarchitektur eines DBMS.
  - ✓ Sie können die 5 Schichten eines DBMS aufzählen.
- ✓ Sie wissen was Räumliche Datenbanksysteme (RDBS) sind.
  - ✓ Sie erkennen RDBS als Mittelschicht zwischen DBMS und GIS.
  - ✓ Sie kennen die Produkte Oracle Spatial und ArcSDE und wissen, wo sie in der obigen Begriffswelt einzuordnen sind.
- ✓ Sie können die Unterschiede zwischen einem DBS, RDBS und einem GIS aufzeigen.

Learning objectives

V3 | Datenbankentwurf | Geo451 | Räumliche Datenbanken | HS09  
33 | Entitätstypen, Entitätsmengen, Primärschlüssel | P. Laube, R. Meile, Uni Zürich

### Zusammenfassung 1.1


- Als **Entität** werden in der Informatik unterscheidbare, in der realen Welt eindeutig identifizierbare einzelne Objekte bezeichnet
- Entitätstyp** – Schablone mit eindeutigem Namen, dient zur Zusammenfassung von Entitäten (nur Entitätstyp wird im Entity-Relationship-Modell dargestellt)
- Entitäten als Teil einer Entitätsmenge werden durch ihre Eigenschaften (**Attribute**) charakterisiert
- Eindeutige Unterscheidung von Entitäten eines Typs erfolgt durch zugeordneten **Primärschlüssel** (meist eine Nummer).

Summary


V3 | Datenbankentwurf | Geo451 | Räumliche Datenbanken | HS09  
51 | Beziehungstypen | P. Laube, R. Meile, Uni Zürich

### Aufgabe 3.3

Ergänzen Sie die strukturellen Einschränkungen.

3.1.) 

- Jede Person muss in einem einzigen Ort wohnen
- Jeder Ort kann von einer oder mehreren Personen bewohnt werden

3.2.) 

- Jeder Mitarbeiter kann an einem oder mehreren Projekten arbeiten
- Jedes Projekt kann von einem oder mehreren Mitarbeitern bearbeitet werden

Excercises

V3 | Datenbankentwurf | Geo451 | Räumliche Datenbanken | HS09  
64 | Zusammenfassung | P. Laube, R. Meile, Uni Zürich

### DB-Deutsch Deutsch-DB

<b>Datenbankentwurf</b>	Entwurf der logischen und physischen Struktur einer DB, dient der Erfüllung der Bedürfnisse der Benutzer.
<b>Entität</b>	individuelles, eindeutig identifizierbares Objekt der realen Welt, das durch Eigenschaften charakterisiert wird (Angestellter „P. Muster“)
<b>Entitätstyp</b>	Menge von Entitäten, Schablone von Entität, die im ER-Diagramm mit Namen und Attribut dargestellt wird (das Kästchen im ER-Diagramm) [ANGESTELLTER]
<b>Attribut</b>	Eine Eigenschaft, die die Beschreibung eines Entitätstyps weiter ausführt (z.B. A1ter: eines Angestellten), besteht aus Name und Datentyp.
<b>Beziehung (relationship)</b>	Wenn sich ein Attribut eines Entitätstyps auf einen anderen Entitätstyp bezieht, wird damit eine Beziehung wiedergegeben.
<b>Relation (relation)</b>	Im Kontext Relationaler Datenbanken meint der Begriff <i>Relation</i> eine Tabelle.
<b>rekursiv</b>	durch sich selbst definierend, sich selbst aufrufend.

Terminology, Glossary

# Exercises & Practicals



## Theoretical Exercises

- Modelling with „Pen and Paper“
- e-learning lessons to expand on lectures' content (GITTA)

## Practicals

- Modelling and implementing a spatial project (= group work)
- Application of knowledge in a real-world research project
- Loading and modification of spatial data
- Creating spatial and standard indices
- SQL-Queries (combinations of non-spatial and spatial)
- Apply knowledge from Geo874

# Timeplan

## Course room

- 8:00 – 8:45 Lecture Part I
- 9:00 – 9:45 Lecture Part II / Theoretical Exercises

## Practical room

- 10:15 -12:00 Practicals with demo, exercises, project
- Individual study, work on reports, online tutorials,...

## Timetable

- 19.11. ER-diagram, email pdf proposals to Esra
- 21.11. Group feedback discussions about ER models
- 5.12. Feedback discussion with groups on work progress
- 19.12. Project delivery (pdf) to Esra

## Evaluation criteria



### Marking of Geo875

- Group project is graded
- Topics evaluated: **a)** ER-modelling, **b)** realisation in the database and **c)** project final report
- If grade is under 4.0 the whole project can be revised once.

No exam 😊

**BEWARE OF PLAGIARISM!**

### Practicals / Exercises

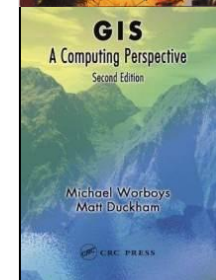
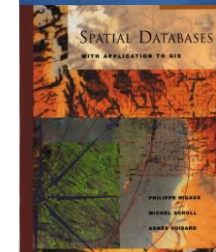
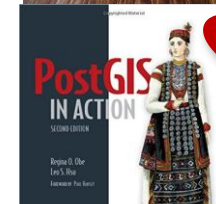
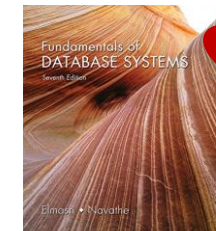
- Not graded. Still possible fail reasons: Regular absences in the exercises; Insufficient participation in group work.

See **Geo 875 — Evaluation criteria for FS 2025**

[https://www.geo.uzh.ch/microsite/geo875/scripte/geo875\\_FS25\\_criteria.pdf](https://www.geo.uzh.ch/microsite/geo875/scripte/geo875_FS25_criteria.pdf)

# Literature

- Ramez A. Elmasri / Shamkant B. Navathe (2016). Fundamentals of Database Systems 7th ed., Pearson, pp. 1242. ISBN: 978-0-13-397077-7
- Regina O. Obe and Leo S. Hsu (2021). PostGIS in Action, Third Edition, Manning Publications pp. 600. ISBN 9781617296697
- Spatial databases a tour (2003) –  
S. Shekhar and S. Chawla, Prentice Hall,  
pp. 262. ISBN 0-13-017480-7
- Spatial databases : with application to GIS (2002) -  
P. Rigaux, M. O. Scholl and A. Voisard, Morgan Kaufmann Publishers, San  
Francisco,  
ISBN 1558605886
- Worboys, M. / Duckham, M. (2004). GIS – A Computing Perspective, 2. Issue,  
CRC Press, Boca Raton FL



## e-Learning online



- <http://www.gitta.info> Swiss virtual campus
- An alternative take on the lectures' content
- DE and EN content available
- Links on geo875 Website
  
- <https://postgis.net/workshops/postgis-intro/index.html> a comprehensive workshop (originally from the Boundless company)
- we use these samples to build up our exercises in geo875
- datasets available in our database = postgresql with postgis installation
- have a look at schema userdemo in the course database

Geo875 | FS25  
University of Zürich

## **1. Lecture Spatial Databases**

### **Definitions and Concepts**

**Rolf Meile**

Eidg. Forschungsanstalt für Wald, Schnee und Landschaft (WSL)  
Swiss Federal Institute for Forest, Snow and Landscape Research

**Esra Suel**

Dept. of Geography, University of Zürich

## Learning Objectives

- Gain a basic understanding of the advantages of managing spatial data in DBMS over file-based spatial data handling;
- Towards object-relational DB modeling: understand how spatial entity types, their attributes and spatial relationship types are modeled in ER models;
- Understand how spatial entities, attributes and relationships are implemented in a spatial database
- Know how to use correct spatial types or subtypes in a spatial database
- Understand what standardisation is good for in the spatial domain
- Get a glimpse of simple features



# Introduction to Spatial Databases

- ▶ **1. Why spatial databases?**
- 2. Extending relational models to handle space
- 3. Storing the spatial world (object-relational data storage)
  - Relations with object attributes (geometries)
  - Spatial query methods
  - Implementing spatial relationship consistency
- 4. Standardisation
  - Why standardise?
  - What to standardise
  - Intro to Simple Features

# Why spatial databases?

## Recall advantages of non-spatial databases

- *Integrity*: Rules and constraints
- *Independence*: Data is separated from the software
- *Security*: Access control

## How does this apply to spatial data?

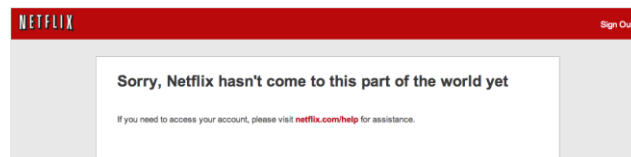
- *Integrity* (conceptual, logical, storage):
  - A specific person unlocks the door of room Y25-H-79; Cars can drive on roads; Parcels do not overlap; Structural: primary keys, foreign keys, unique key, data types, topology, ...
  - Storage integrity: attribute and spatial data handled together, not separately (e.g. in different files, different storage locations)
- *Independence*:
  - Spatial database for generic use with many clients: Desktop GIS, Apps, Web, R, Python, ...
  - Concurrent use and edits of the same data source

# Why spatial databases? cont.

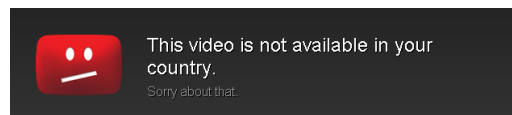
## How does this apply to spatial data? cont.

### – *Security:*

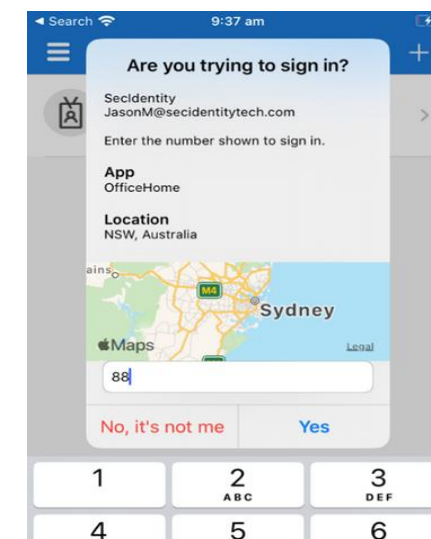
- A user can only find out information about object within 50 m
- Users within a certain space can not have access to certain information (*geofencing*)
- Users approaching other user beyond a certain threshold are immediately warned
- Locations of critical infrastructure (company, government) not visible to all user groups
- Locality of rare Red List species (plants, animals)



Streaming



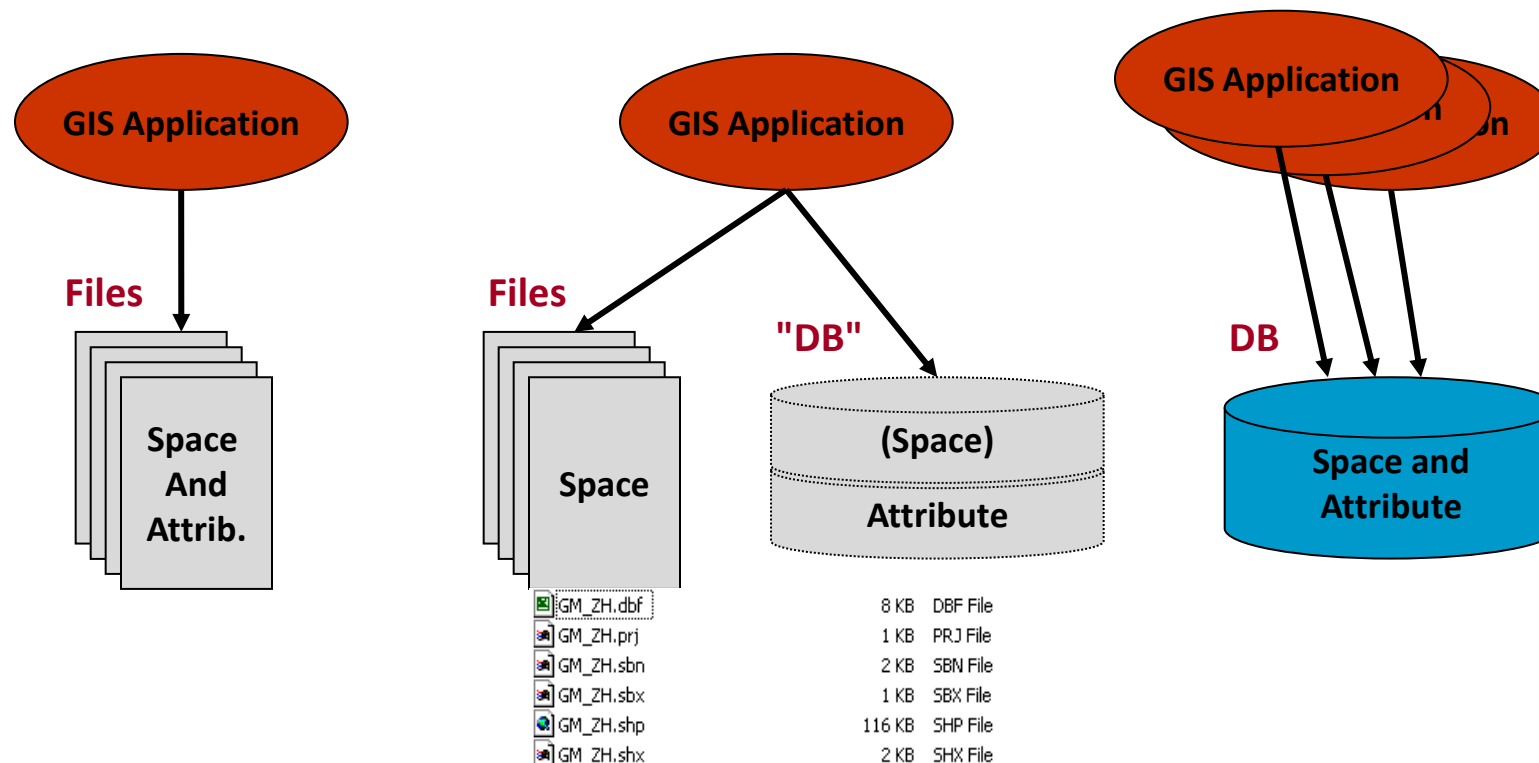
2-Factor authorization incl. location



# Why spatial databases (cont.)?

## Storage of spatial and non-spatial attributes

1. Files: geojson, GML, Interlis (geometers CH), Geopackage...
2. Files with "DB format" for attributes (e.g. Shapefile), and so-called DBs (e.g. FileGeoDatabase)
3. Real DBs (PostgreSQL, Oracle,...), SQLite



## Why spatial databases (cont.)?

### Persistent storage of 'layers' (=spatial datasets)

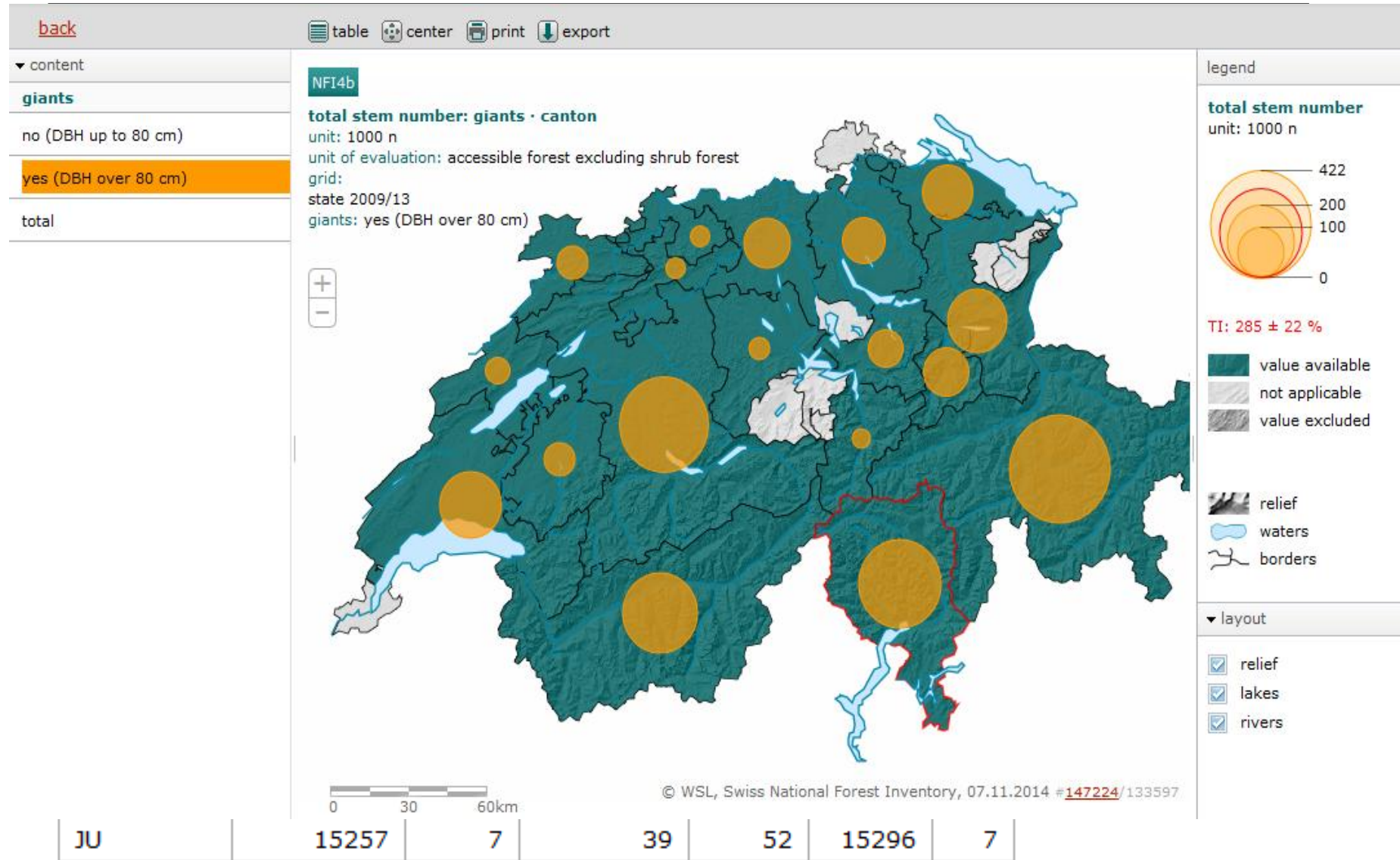
- Altogether with all connected data and metadata chains
- Spatial and non-spatial data is formalized and logically connected
- Reliable data sources; one piece of information at one place;
- On the fly calculating of all kind of relationships

### Multiple applications and users access the same spatial data

- Web applications, R analysis software, SQL analysis, GUIs, GISs - all can query at the same time
- Consistent state for all users through their applications
- Editing (means update, insert, delete) is consistent. Troubles can arise with long ongoing transactions.

# Data analysis Swiss NFI

lfi.ch



# Sharing large data collections [geo.admin.ch](http://geo.admin.ch)

The screenshot displays the geo.admin.ch web application interface. At the top left, the Swiss flag is shown next to the text: "Schweizerische Eidgenossenschaft", "Confédération suisse", "Confederazione Svizzera", and "Confederaziun svizra", with the note "In collaboration with the cantons".

A search bar at the top center contains the text "ZIP and locations".

On the left side, there is a navigation menu with the following items:

- Share
- Print
- Draw & Measure on map
- Advanced tools
- Geocatalog Change topic
- Maps displayed**

Under "Maps displayed", the following layers are listed with checkboxes and gear icons:

- 4G antenna locations (LTE)
- Register of Buildings and Dwellings
- Public transport stops
- ZIP and locations
- National Map 1:10'000 (grey)

Below the list is the text "Looking for more maps?".

At the bottom of the menu is "Configuration".

The main map area shows a street map of Zurich with several data layers overlaid: orange dots representing 4G antenna locations, blue squares representing public transport stops, and grey outlines representing buildings and dwellings. The text "3057 Zürich" is visible on the map.

On the right side of the map, there are navigation controls: a home button, a plus sign for zoom in, a minus sign for zoom out, and a 3D button.

At the bottom right of the map, there is a "Background" button with a thumbnail image.

At the bottom of the interface, there is a scale bar showing "200 m", a coordinate system selector set to "CH1903+ / LV95", and coordinate values: "Coordinates (m): 2'683'350.0, 1'250'393.7".

At the very bottom, there is a copyright notice: "© Data: swisstopo, FOT, FSO, OFCOM" and "geo.admin.ch Copyright & data protection".

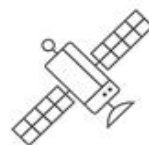
# Massive spatial data analytics [earthengine.google.com](http://earthengine.google.com)

☰ Google Earth Engine

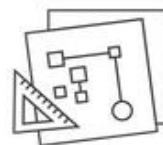


## Meet Earth Engine

Google Earth Engine combines a multi-petabyte catalog of satellite imagery and geospatial datasets with planetary-scale analysis capabilities and makes it available for scientists, researchers, and developers to detect changes, map trends, and quantify differences on the Earth's surface.



SATELLITE IMAGERY



YOUR ALGORITHMS



REAL WORLD APPLICATIONS

## Spatial databases @ job

e.g. [www.wsl.ch](http://www.wsl.ch)

### Specialist in data management (80%)

You will be responsible for the user-friendly recording of forest inventory data as well as their long-term storage within the framework of the projects "Experimental Forest Management" and "Forest Reserve Research". Your tasks will include the development and support of **spatial and relational databases (PostgreSQL/PostGIS)** as well as the development of tools for data management. These are a data collection software used in

### Web-/ Software-Entwickler/-in (80%)

SwissLichens & SwissFungi sind die nationalen Kompetenzzentren für Flechten und Pilze. Die Datenbestände umfassen über 750'000 aktuelle und historische Fundmeldungen von über 10'000 Arten. Die Aufgabe der Datenzentren umfasst das Sammeln, Validieren und Publizieren von Verbreitungsnachweisen und Artinformationen. Sie stellen die bestehenden **Verbreitungskarten** und webbasierten Eingabetools von SwissLichens & SwissFungi auf moderne Technologien um. Sie entwickeln neue Werkzeuge für den Datenimport und programmieren Schnittstellen zur Anbindung von Partnerorganisationen. Für das Datenzentrum Natur & Landschaft entwickeln Sie **webbasierte GIS-Applikationen** für die Umsetzung der Biodiversitätsstrategie der Schweiz und integrieren Daten zu den Natur- und Landschaftsinventaren des Bundes. Unterhalt und Pflege der Applikationen und **Datenbanken** gehören ebenfalls zu Ihren Aufgaben.

# Introduction to Spatial Databases

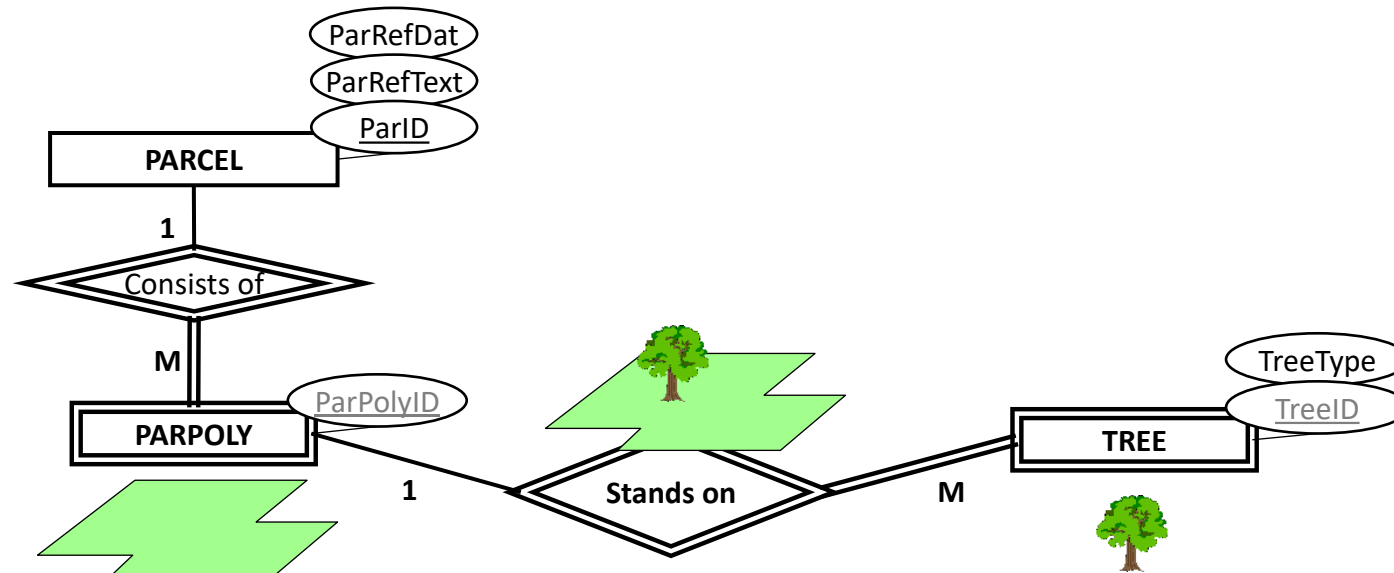
1. Why spatial databases?
- ▶ **2. Extending relational models to handle space**
3. Storing the spatial world (=object-relational data storage)
  - Relations with object attributes (geometries)
  - Spatial query methods
  - Implementing spatial relationship consistency
4. Standardisation
  - Why standardise?
  - What to standardise
  - Intro to Simple Features

## Exercise 1.1



How would you extend **traditional relational** databases with specific capabilities for storing spatial data? Discuss using the ER example below:

- Which things we need for modelling (ER, Attributes, Relations)? focus is ER modeling
- What is missing in these databases? focus is database



# Modeling and storing data about space

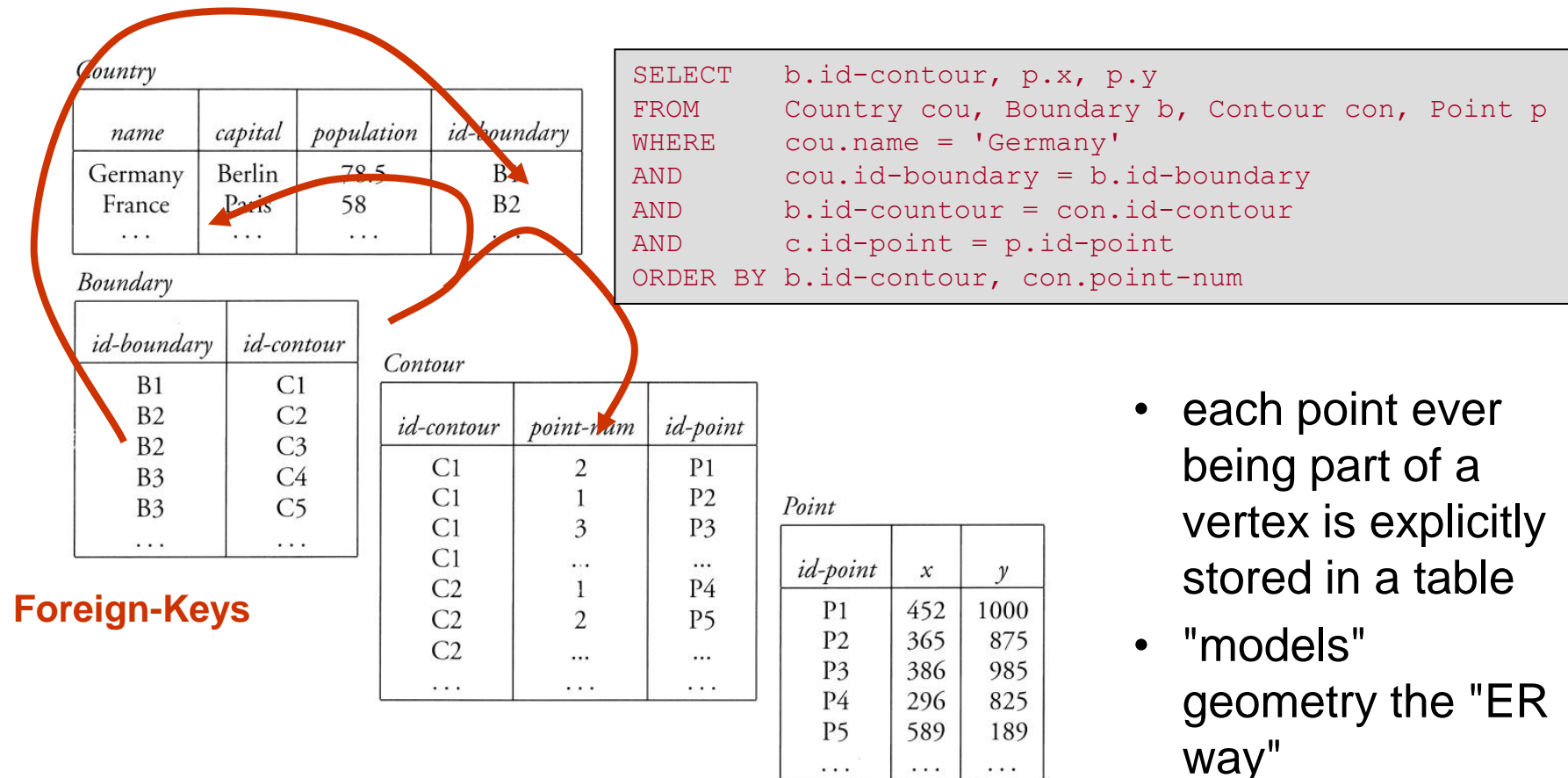
**Problem levels:** ER model -> relations -> DB tables...

1. An entity type has attributes of homogeneous basic values (numbers, strings, dates). **But how do you model (ER) and later store (DB) their location, incl. shape & size?**
2. Relationships between normal entity types are in one or another way deterministic (each company can have one or more managers). **But does a **spatial relationships** behave the same way (each tree can stand on one or more parcels)?**
3. Querying: How to do queries based on spatial attributes? **How do you (efficiently) find entities with a certain spatial relationship?**

# Modeling Geometry Data Type

**Pure relational:** A complex (and complicated) solution; feasible for points only;

Example with country-boundary-contour-point:



# Solutions

## Prerequisites on the DB side for ER modeling

- Introducing an abstract **geometry** type
- Enabling the **computation** and **storage** of spatial relationships
- Support **efficient spatial querying** and storage with data structures (spatial indexes) and spatial methods.
- all this is implemented in a spatial database (no GIS client/application);

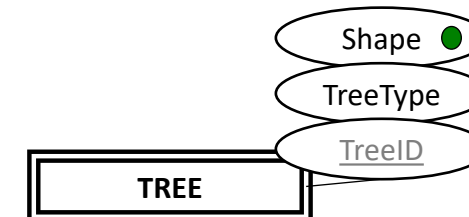
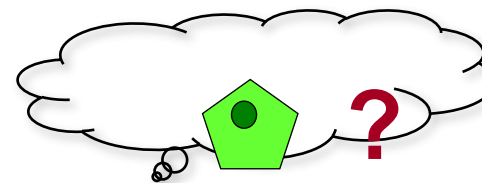
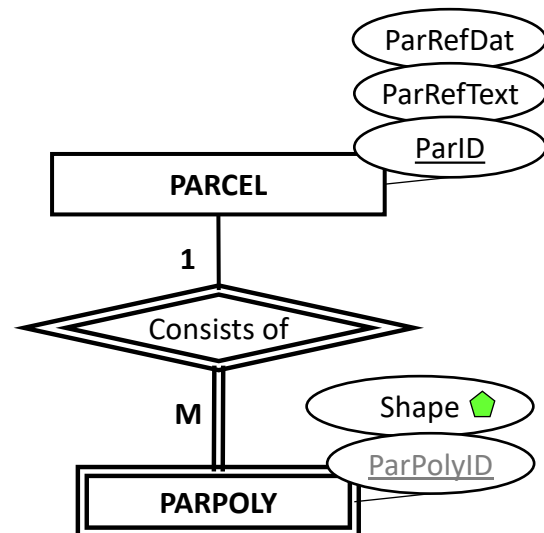
## Now we can on the ER side

- model a **spatial entity type** with its geometry attribute
- model a **spatial relationship type** between two spatial entity types (option C in the following slides; -> hardly ever used this way; leads to separate topology implementation)

# ER models of the spatial world

## Solution in detail:

- **Spatial entity types** are modeled as having a combination of attributes of simple data types and of **one** single geometry data subtype;
- **Geometric attributes** are tagged with an icon of type Geometry;
- **Relationships** – explicit ones or none (**see next slides**)



## Modeling spatial relationships

**If two spatial entity types (with their spatial attribute) have a mutual spatial relationship:**

- A) The nature of the relationship is that of a '**strong participation**' or a '**part of**' or '**weak entity type**'.
- > **model this with standard ER techniques using cardinalities and participation**
  - > later in the DB this leads to foreign keys pointing to primary key attributes; spatial querying possible as well;
- B) The relationship is **loosely spatial** in the topological sense (e.g., lies in, close to, adjacent to,...) ; no topology needs to be enforced while gathering new data
- > **nothing needs to be modeled**; leads to standalone spatial table (this is the 'shapefile' or 'layer' paradigm stored in database tables)
  - > recall: spatial querying fully available;
- C) The relationship is **loosely spatial** in the topological sense; some topology rules, if possible, should be applied to newly inserted data
- > **model a spatial relationship without cardinality (-> rather implement topology)**
  - > could lead to a trigger (little checking program) in the DB

# Modeling spatial relationships

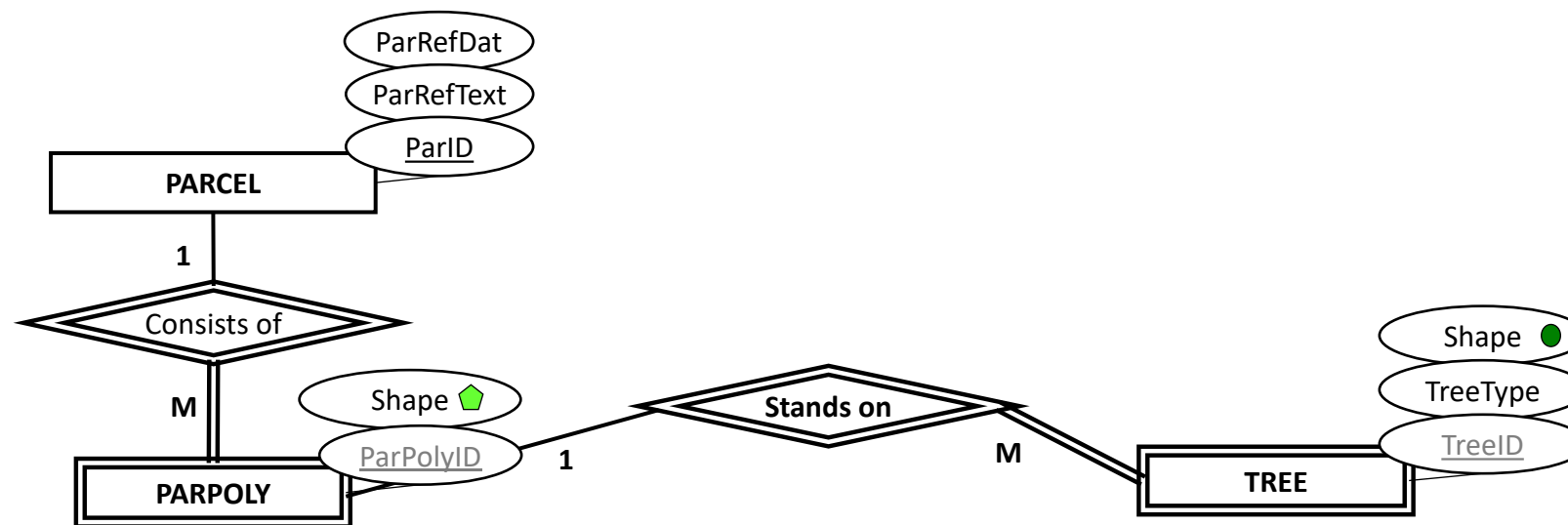
## Option A: Strong participation

Relationship type reading as we know from geo874

- „Each tree must stand on exactly one parcel.“
- „Each parcel can have one or more trees.“

Further descriptions that fit the context

- „A tree can only exist if there is a given parcel polygon on which it can stand.“
- „I record trees in my database which I have seen standing on given parcels.“



## Excercise 1.2



### Option A - modelling strong participation

Which main problem could arise when modelling like this? Think of real world data that would be filled into the resulting empty spatial tables. Or if data has to be changed at some point in time.

- Quick discussion with your colleague(s)

# Modeling spatial relationships

## Option B: Loosely spatial - no relationship

- "Locations of trees were collected independently from parcels spatial and non-spatial attributes"
- "New incoming tree data is not checked regarding to parcels or polygons of parcels"
- "Tree datasets could come from whole Switzerland - parcels could be from the city of Zurich, only"
- "The datasets are loosely spatial because they are spatial! This is implicit and the only common property."
- "There is no explicit relationship between the entity types. But still, we can spatially query them."

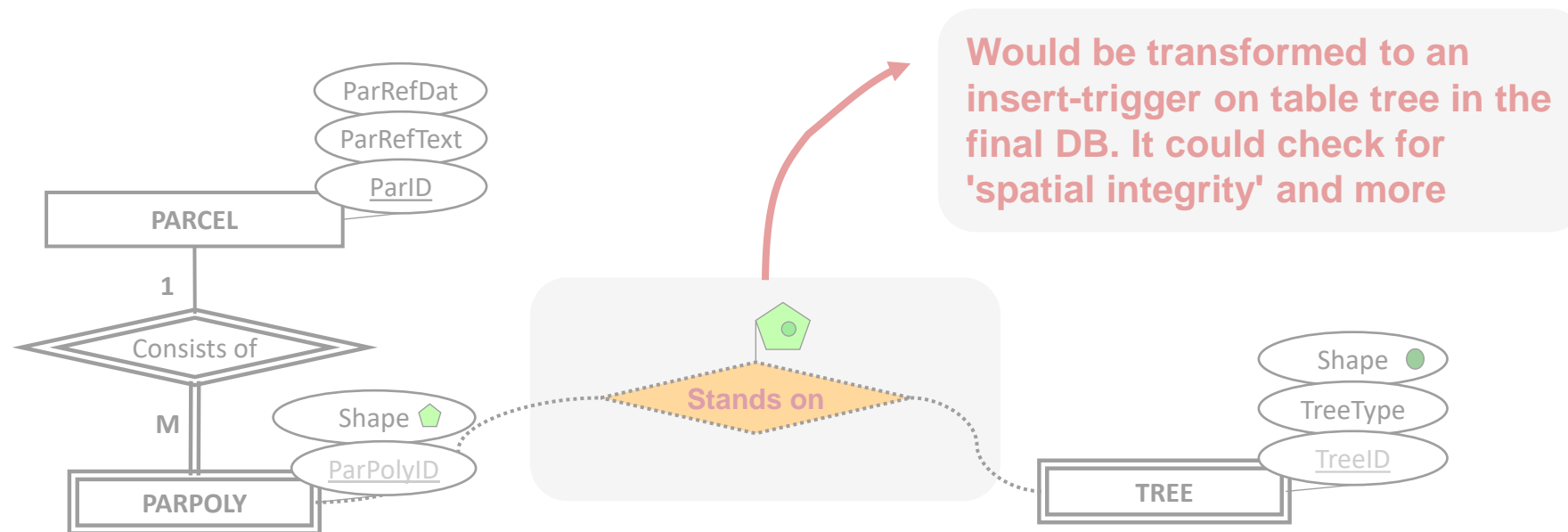


# Modeling spatial relationships

## Option C: Loosely spatial - spatial relationship

- „Each tree must stand on some (existing) parcel.“
- „A tree can only exist when there is some parcel polygon which it can stand on.“
- „I record trees with their coordinates. Tree data is not actively connected with parcel information. Only when entering new tree data into the database we check wheter the trees stand on some parcel. This is done by a spatial comparison with a spatial operator.“

-> **topology implementation** (lecture 3)



# Introduction to Spatial Databases

1. Why spatial databases?
2. Extending relational models to handle space
- ▶ **3. Storing the spatial world (Object-Relational data storage)**
  - Relations with object attributes (geometries)
  - Spatial query methods
  - Implementing spatial relationship consistency
4. Standardisation
  - Why standardise?
  - What to standardise
  - Intro to Simple Features

# Storing the DB spatial world – Abstract Data Types

## The **object-relational** model

- **Spatial tables** (spatial relations) have a combination of columns of **simple data types** and a **single** column of the **geometry data type**
- Displayed in a mapping application a spatial table is called *feature class* or *layer*
- Metadata is specified as part of the geometry type, e.g., spatial reference systems for coordinate storage.
- Attributes of type geometry are filled with **object** values
- These spatial objects are **supported by all spatial DBMS** (e.g PostgreSQL with PostGIS)
- **Query methods** and **spatial indices** are supported, too

### Geometry as Abstract-Datatype ADT

treepoint	
treepointid (PK)	bigint
parzid (PK)	int
parzpolyid (PK)	smallint
<b>geom</b>	<b>geometry</b>
treespecies	varchar(256)

# Storing the spatial world – Abstract Data Types

## More details on geometry data type and subtypes

- Base geometry type can accept different 'geometries'
- points, lines, polygons up to 3D space and/or metrics
- multipoints, multilines, multipolygons with multiple geometries per record
- geometrycollection with possibility to store heterogeneous geometries in the same spatial attribute of a spatial table; try to avoid;
- Cartesian plane as a base for all calculations

## Subtyping, constraining the base geometry type



- Check on imported datasets in a DB! Don't expect anything. Check it.
- Modeling fits to database
- Transparency & consistency
- Search for: casting, typmod, type modifier

**Subtyped geometry attribute  
with point and CH1903+ / LV95**

treepoint	
treepointid (PK)	bigint
parzid (PK)	int
parzpolyid (PK)	smallint
<b>geom</b>	<b>geometry(point,2056)</b>
treespecies	varchar(256)

# Storing the spatial world – Abstract Data Types

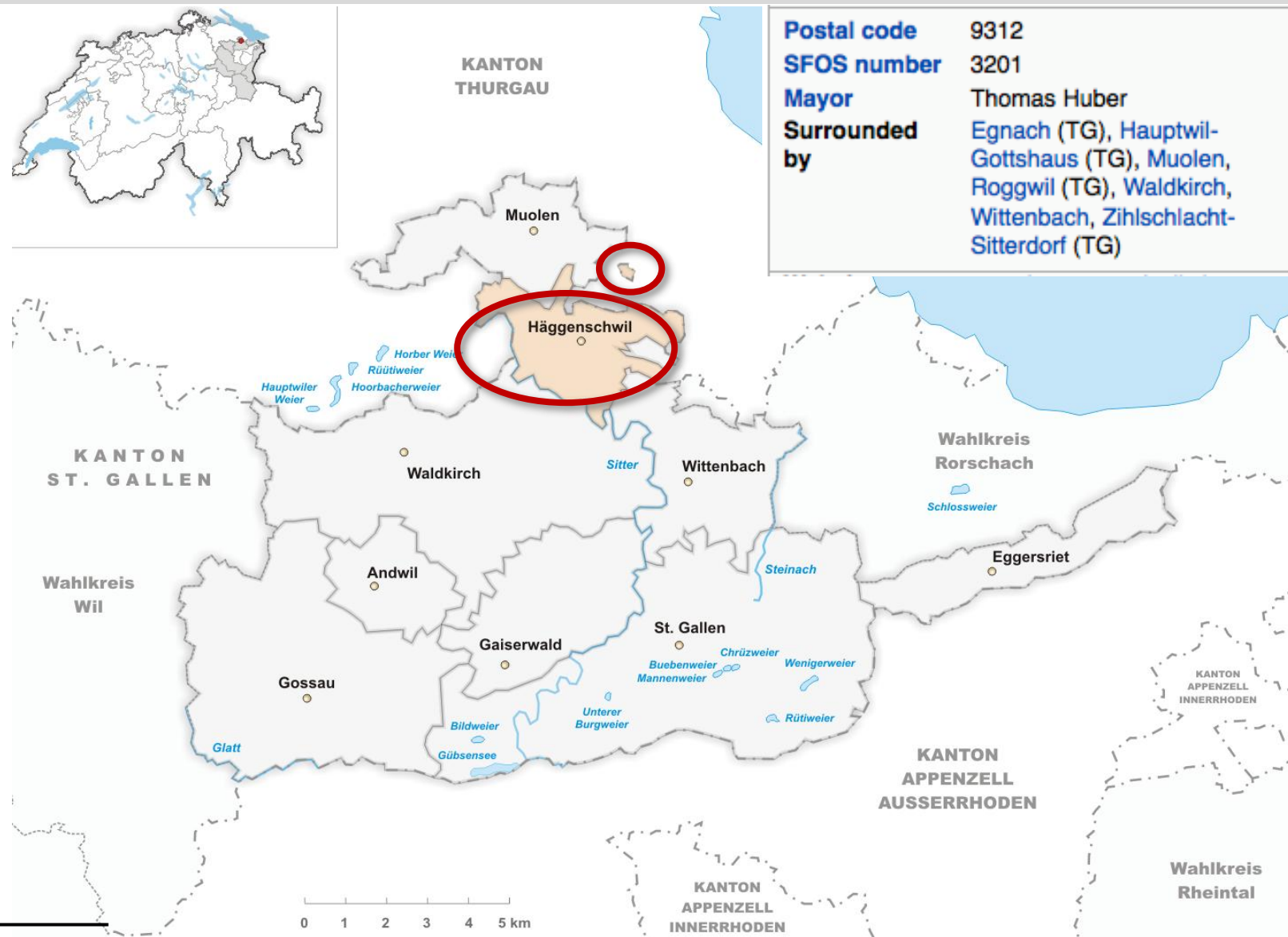
## Keep in mind

- Geometries are queried through functions (hence, *object-relational*):  
ST\_Area(geometry), ST\_Centroid(geometry), ST\_Distance(geometry,geometry)...
- Geometry data type exposes many functions about the contained geometries: ST\_SRID(geometry), ST\_Envelope(geometry)...
- Multi geometries behave just as geometries  
ST\_Area(  ) works as well as ST\_Area()
- Efficient evaluation of some topological queries (ST\_Touches, ST\_Inside,...) requires ordering of geometries → **spatial indices**
- Abstract data types and functions can be used in **pure SQL**:  
`SELECT ST_GeometryType(t.geom) FROM treepoint t;`

## More on implementation details of ADTs

- see later in this and next lecture and practicals
- <https://postgis.net/docs/reference.html>

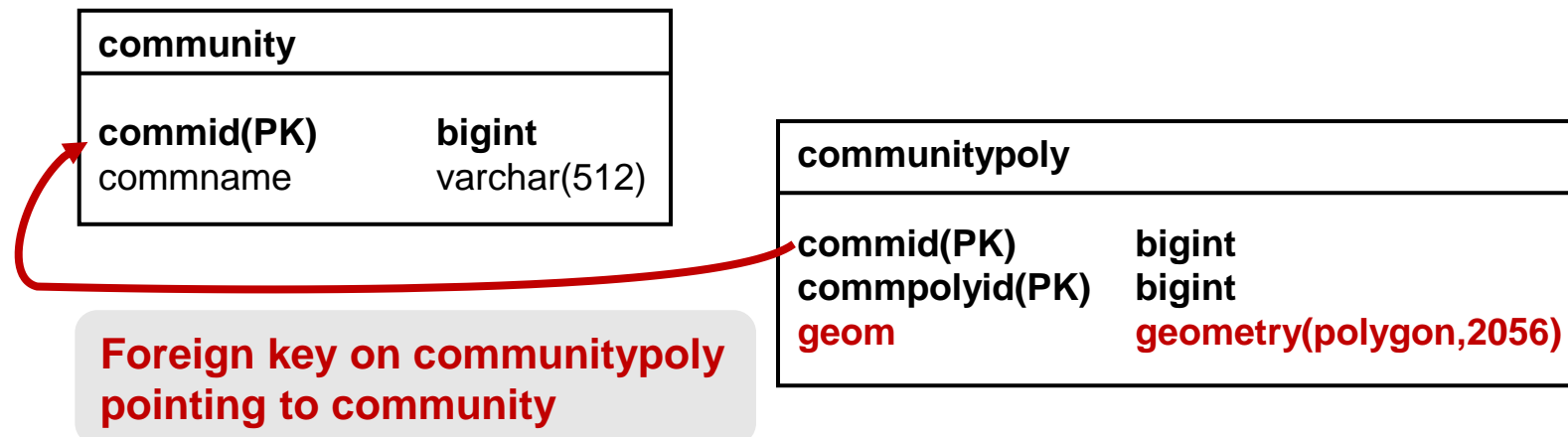
# Community Häggenschwil SG – a MultiPolygon



## Modeling Multi or Single (-Polygons, -Lines, -Points)

### Method A: Use singlepart geometry subtype

- Each geometric polygon is planned to be saved as a single record for the 'multi-case' depicted in the previous slide
- Leads to **two entity types**: Example Häggenschwil with entity type *community*, relationship type *consistsof*, entity type *communitypoly*.
- Add normal attributes to the appropriate entity type
- Advantage: This **separates semantic meaning** for community **from spatial meaning!**



## Modeling Multi (-Polygons, -Lines, -Points)

### Method B: Use multipart geometry subtype

- ER model reflects a 1:m relationship in one single spatial entity type; circumstances somewhat hidden in the geometry subtype; advantage: **intuitive**;
- Simply one entity type for community; one has to 'know' that we have stored more than one polygon in one record (!)
- Not all GIS visualisation and calculations work with multipart:  
E.g. ST\_Split(geometry1, geometry2)
- If attributes only apply to some polygons being part of a multipolygon record -> use method A

community	
commid(PK)	bigint
commname	varchar(512)
<b>geom</b>	<b>geometry(multipolygon,2056)</b>

**Subtyped geometry  
attribute with polygon and  
CH1903+ / LV95**

## Storing the spatial world – rasters

- **An Abstract Data Type model as well:** multiple tables to reflect raster pieces and metadata; the raster itself is stored in a column of (behind the scenes) type BLOB (Binary Large Object)
- Esri raster type: SDE.ST\_RASTER
- Oracle raster type: SDO.SDO\_GEORASTER
- Postgis raster type: raster.
- Rasters not necessarily stored in DBs
- More details later in the semester. Rasters are not easy!

## Excercise 1.3



### ER modeling with spatial data

You get two spatial datasets from the national map agency. One is the community dataset of Switzerland and the second one the cantons of Switzerland. How would you model and finally store these two datasets in a database?

- Discuss with your colleague(s).
- Suggest ER/Relation/Table solution(s)
- What are possible pitfalls or implicit assumptions we have here?

# Introduction to Spatial Databases

1. Why spatial databases?
2. Extending relational models to handle space
3. Storing the spatial world (Object-Relational data storage)
  - Relations with object attributes (geometries)
  - Spatial query methods
  - Implementing spatial relationship consistency
- ▶ **4. Standardisation**
  - Why standardise?
  - What to standardise
  - Intro to Simple Features

## Spatial schemas – the subject of standardisation

- **Spatial capabilities** are often an optional “schema” that can be loaded on top of a pure relational database
- Oracles spatial option is included in the SDO schema
- ESRI's spatial extension (on top of many DBs) goes to the SDE schema
- PostgreSQL with PostGIS (spatial extension) is available from the schema named **public**
- These schemata share their types, subtypes, operators, functions, views, tables with us; these are the spatial capabilities of a spatial DB;

```
SELECT PostGIS_full_version();  --identical to  
SELECT public.PostGIS_full_version();
```

```
POSTGIS="3.2.1 5fae8e5" [EXTENSION] PGSQL="140" GEOS="3.8.0-  
CAPI-1.13.1 " SFCGAL="1.3.7" PROJ="6.3.1" LIBXML="2.9.10"  
LIBJSON="0.13.1" LIBPROTOBUF="1.3.3" WAGYU="0.5.0  
(Internal) "
```

# Standardisation

- Spatial data storage and handling standardised by the Open Geospatial Consortium (<https://www.opengeospatial.org/>)
- ISO group TC211 is very active on geography/geomatics standards (<https://www.isotc211.org/>)
- Standardisation contributes to **interoperability**;
- **Meaning 1**: export your DB into an SQL dump, and load into the RDBMS of a different vendor;
- **Meaning 2**: apply your skills with whichever RDBMS;
- Standardisation covers:
  - Definition of Abstract Data Types (ST\_Geometry...) and their parts (coordinate pairs, coordinate sequences)
  - Definitions of spatial reference systems, units, etc
  - Formats and notation
  - ST\_ method nomenclature as an example. Your scripts should be portable from one RDBMS to another
- ...in reality, differences exist. Beware!

# Standardisation

- Geodata standards:
  - **Open Geospatial Consortium (OGC)**: One of the most important ones -> **Simple Features** Specification. Paralleled by ISO-19107 and 19125;
  - **ISO: SQL/MM Part 3**: Spatial ISO-13249. Multimedia and Spatial Data; Defines abstract data types;
- For DB interfacing (=query language):
  - ISO:SQL-92 up until ISO:**SQL-2016** and beyond

## Published standards [\[edit\]](#)

The list of standards by the ISO/TC 211 committee:<sup>[10]</sup>

[ISO 6709:2008](#) *Standard representation of geographic point location by coordinates*

ISO 19101:2002 *Geographic information -- Reference model*

ISO/DIS 19101-1 *Geographic information -- Reference model - Part 1: Fundamentals*

ISO/TS 19101-2:2008 *Geographic information -- Reference model -- Part 2: Imagery*

ISO/TS 19103:2005 *Geographic information -- Conceptual schema language*

ISO/TS 19104:2008 *Geographic information -- Terminology*

ISO 19105:2000 *Geographic information -- Conformance and testing*

ISO 19106:2004 *Geographic information -- Profiles*

ISO 19107:2003 *Geographic information -- Spatial schema*

ISO 19108:2002 *Geographic information -- Temporal schema*

ISO/CD 19109 *Geographic information -- Rules for application schema*

ISO 19109:2005 *Geographic information -- Rules for application schema*

ISO 19110:2005 *Geographic information -- Methodology for feature cataloguing*

ISO 19111:2007 *Geographic information -- [Spatial referencing by coordinates](#)*

ISO 19111-2:2009 *Geographic information -- Spatial referencing by coordinates -- Part 2: Extension for parametric values*

Incomplete list of standards by the ISO group TC211

## Example: Simple Feature Specification for SQL

- Standardises:
  - Main aspects of spatial data access in a database;
  - Focuses on 0-2D Geometries (not 3D), and only on geometries with linear interpolation between vertices;
  - Defines what are **valid and invalid** geometries
  - Categorises geometry types and **defines their methods and operations**
  - Standardises topology and raster access;
- Thus, enables the development of new methods, exchange of data, sharing of know-how
- Does not prescribe internals – only aspects pertaining to access and exchange!

Very useful: OGC Simple Features Spec 1.2.1:

[https://portal.opengeospatial.org/files/?artifact\\_id=25355](https://portal.opengeospatial.org/files/?artifact_id=25355)

[https://portal.opengeospatial.org/files/?artifact\\_id=25354](https://portal.opengeospatial.org/files/?artifact_id=25354)

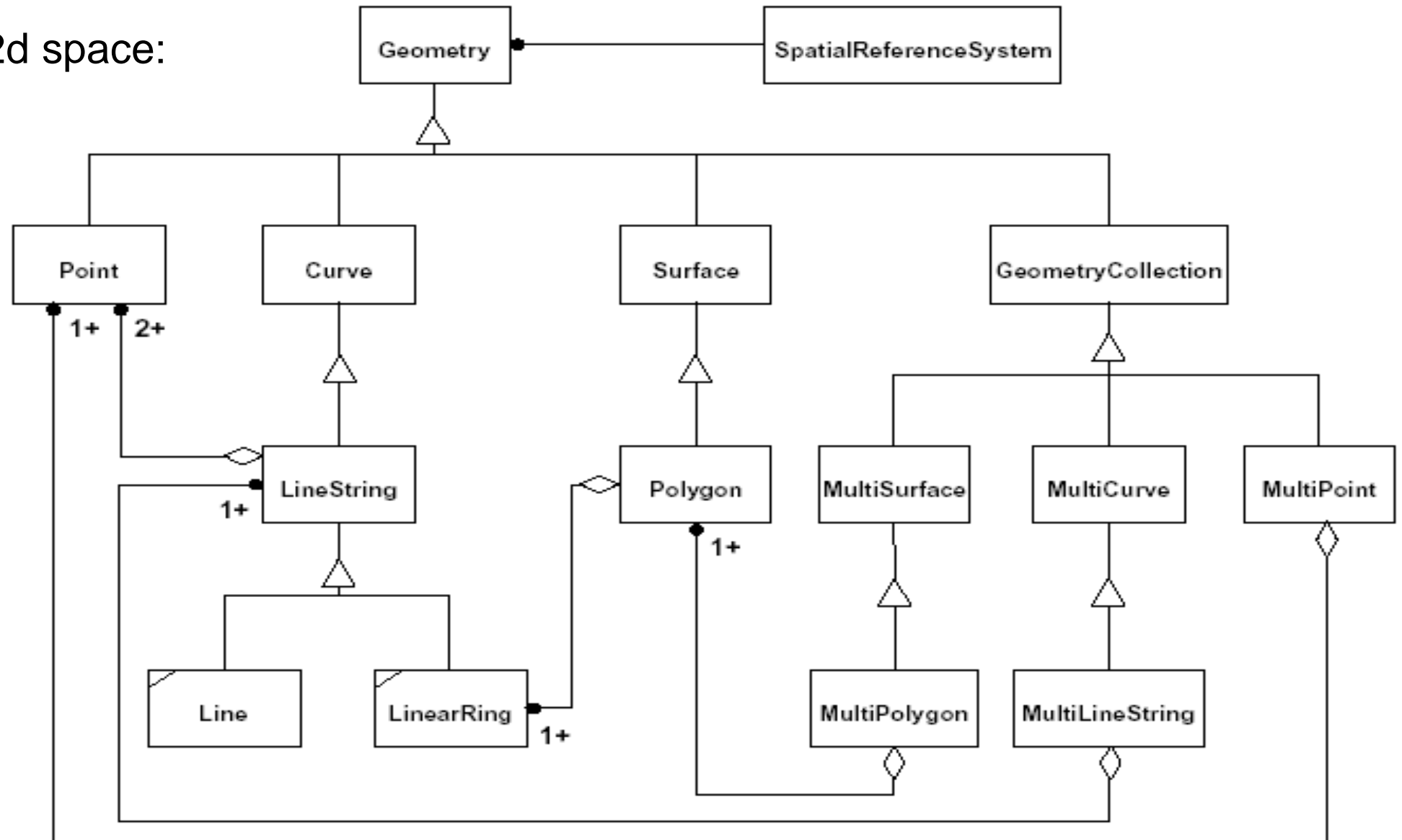
## Example: SQL-MM Part 3 (ISO/IEC 13249-3:2016)

- SQL Multimedia Applications Spatial specification
- Standardises concerning geometries:
  - circularly interpolated curves;
  - 3d and 4d geometries; still no srid - as in simple features....
  - what does 3d and 4d mean... **only coordinates!** not geometries
- most parts of that are integrated in Postgis/Postgresql; additional cool stuff, too!

[http://www.iso.org/iso/home/store/catalogue\\_ics/catalogue\\_detail\\_ics.htm?csnumber=60343](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=60343)

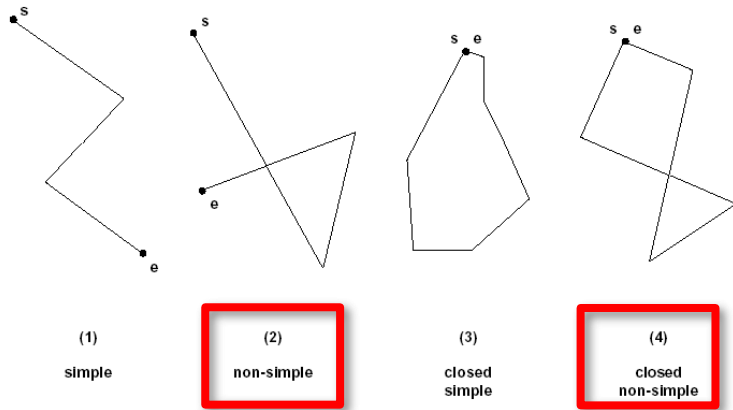
# Simple Features

All geometries in 2d space:

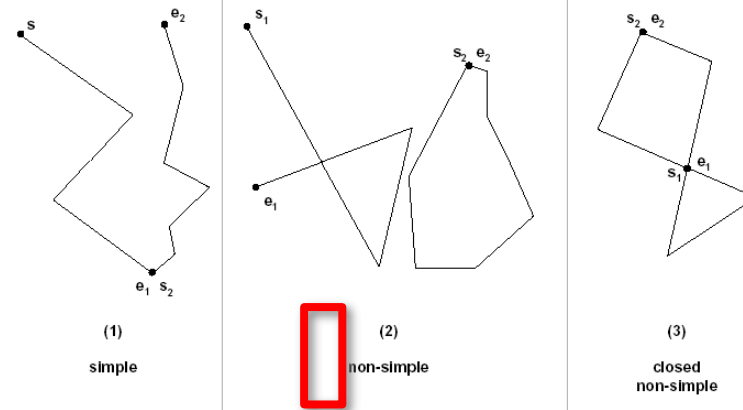


# Valid geometries - examples (also simple vs. non simple)

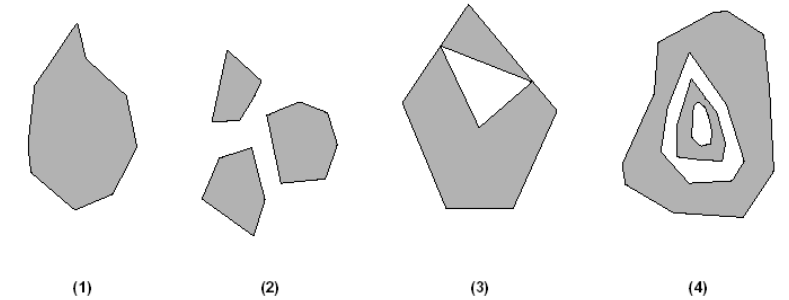
## Valid LineStrings



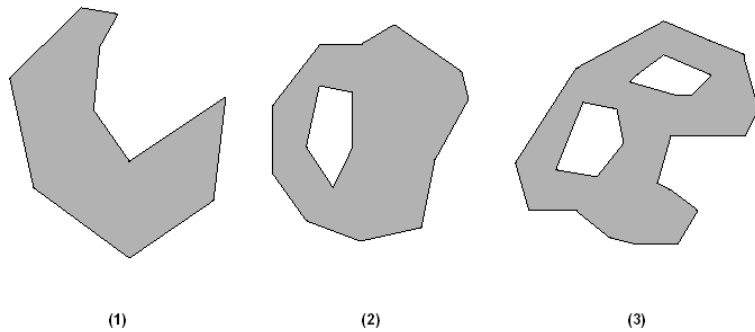
## Valid MultiLineStrings



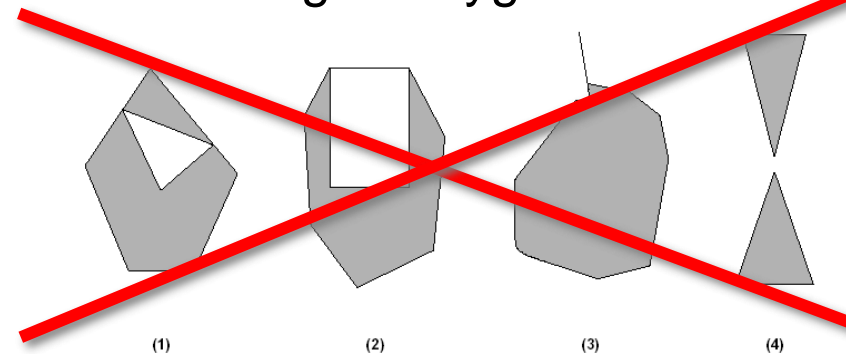
## Valid Multipolygons



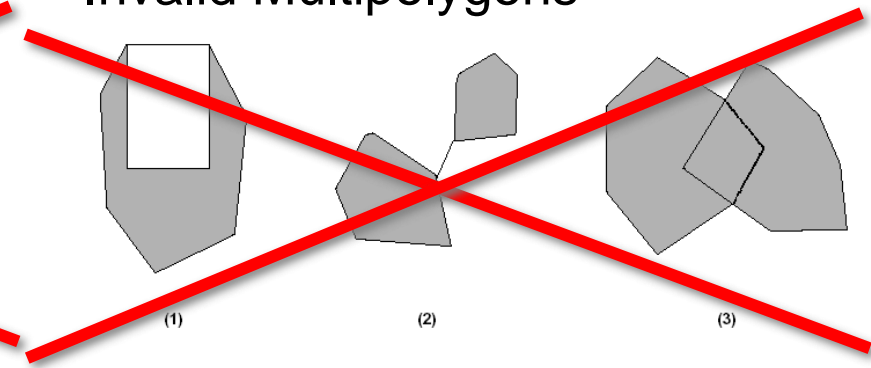
## Valid single Polygons



## Invalid single Polygons



## Invalid Multipolygons



## Standardized Methods (Simple Features)

### Geometry constructors

```
ST_GeometryFromText (geomtxt)
ST_GeomFromGeoJSON (geomjson) .....
```

### Spatial relationships and measurements

```
ST_Contains (geomA, geomB)
ST_Distance (geomA, geomB) .....
```

### Geometry accessors

```
ST_Envelope (geom)
ST_GeometryType (geom) .....
```

### Geometry processing

```
ST_Difference (geomA, geomB)
ST_Union (geomA, geomB) .....
```

### Geometry editors

```
ST_RemovePoint (linestring, offset)
ST_SetSRID (geom) .....
```

### Linear referencing

```
ST_AddMeasure (geom_mline,
measure_start, measure_end)
ST_Line_Substring (a_linestring,
startfraction, endfraction) .....
```

### Validity

```
SELECT
ST_IsValid(ST_GeomFromText('LINESTRING(
0 0, 1 1)')) As good_line,
ST_IsValid(ST_GeomFromText('POLYGON((0
0, 1 1, 1 2, 1 1, 0 0))')) As bad_poly;
```

# Summary

- We need to extend the relational model to handle space
  - Geometry attributes (Geometry objects, reference systems)
  - Spatial relationships (topological, range,...)
  - Spatial methods for (efficient) spatial queries (supported by spatial indexes)
- Spatial relationships can be stored either by:
  - Explicitly storage - strong participation, weak entity (with known cardinality);
  - no relationship
  - spatial relationship
- Spatial data handling in DBs is standardised by OGC or ISO
- Simple Feature specification
  - Valid and invalid geometries
  - 0D,1D,2D (Multi)geometries



# Deadlines

## Timetable and deadlines

- ▶ 13.11.2025 (Thu) - Deadline for cancellation of module booking
- ▶ 19.11.2025 (Wed 11:59 pm) - ER-Diagram proposals to **Esra Suel** (see the above Communication rules)
- ▶ 21.11.2025 (Fri) - Group feedback discussions about ER models
- ▶ 05.12.2025 (Fri) - Short oral report and discussion about current state of the project
- ▶ 19.12.2025 (Fri 11.59 am) - Project report delivery (digital PDF version to **Esra Suel** (see the above Communication rules) and finalization of database modeling