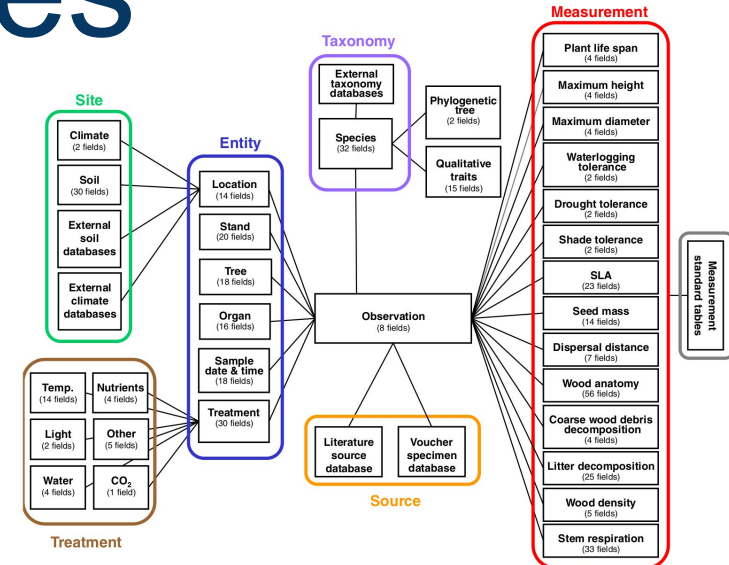


Big Data Ecology Databases

Christian König

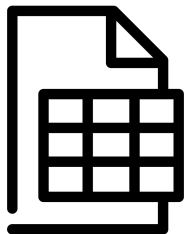
Ecology and Macroecology Lab
Institute for Biochemistry and Biology
University of Potsdam



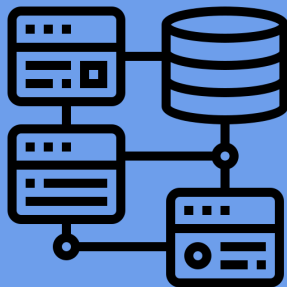
Overview

- A database is a collection of electronic data, usually organized for rapid search and retrieval
- The software that manages the data organization and interacts with the end user is called a database management system (DBMS)
- Different DBMS designs for different applications:

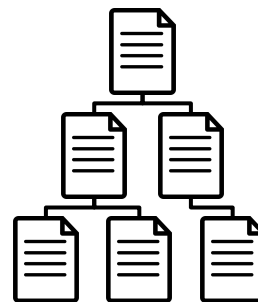
Spreadsheets



Relational DBMS

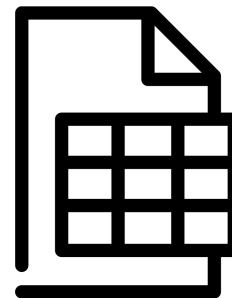


Non-Relational (NoSQL) DBMS



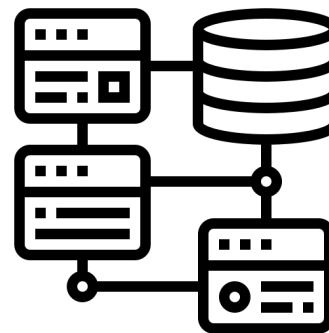
Spreadsheets

- Not a “database” in the traditional sense, but still widely used
- **Advantages:**
 - Easily accessible and usable, human-readable
 - better than your field notebook
- **Drawbacks:**
 - Error-prone
 - redundant data storage
 - no user access management
 - [...]



Relational databases

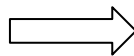
- Data format and relationships are strictly defined
- **Advantages:**
 - No redundant data storage,
 - Data consistency due to [normalization](#)
 - good [vertical scalability](#)
 - Fast data search, modification and extraction via [SQL](#)
- **Drawbacks:**
 - Can be complex and inflexible
 - May incur additional hardware and maintenance costs
 - does not scale to really large data (PB and beyond)



Relational databases - the database schema

- The database schema defines the structure of the data and the relationship among different tables

taxonomy			
	Field name	Data type	[...]
🔑	species_id	<unsigned int(10)>	[...]
	species_name	<varchar>	[...]
	genus	<varchar>	[...]



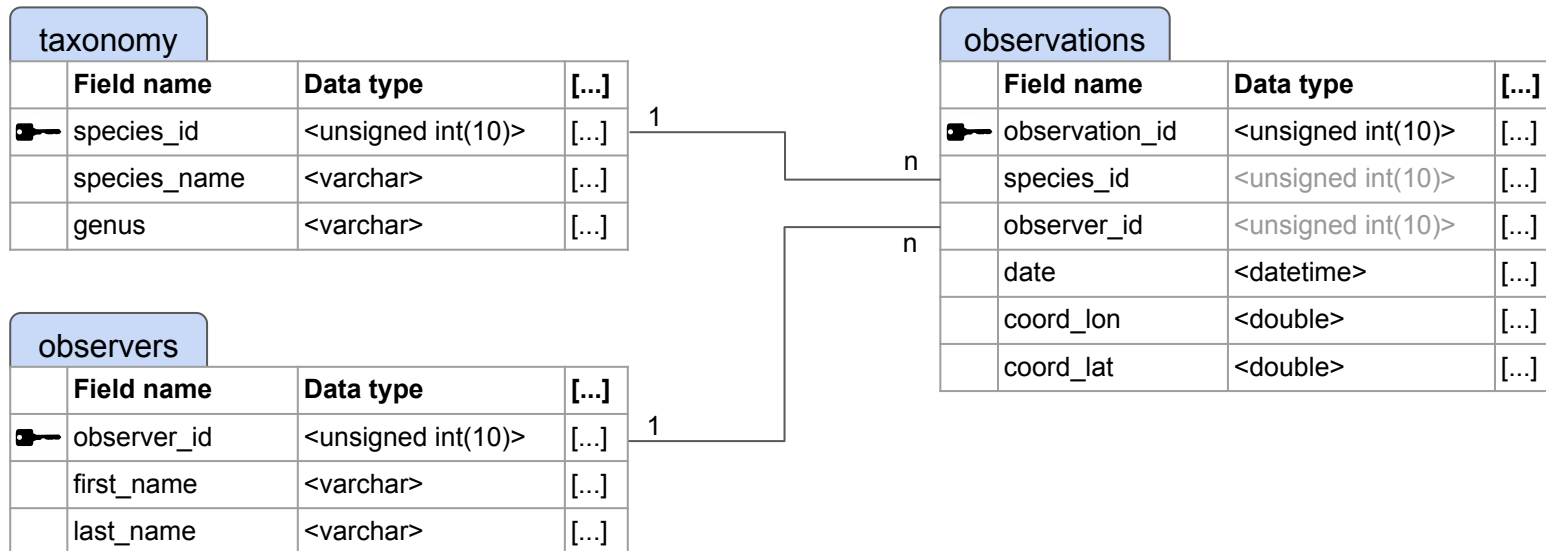
taxonomy		
species_id	species_name	genus
1	Circus aeruginosus	Circus
2	Circus cyaneus	Circus
3	Circus pygargus	Circus
4	Circus macrourus	Circus
5	[...]	[...]

Structure

Data

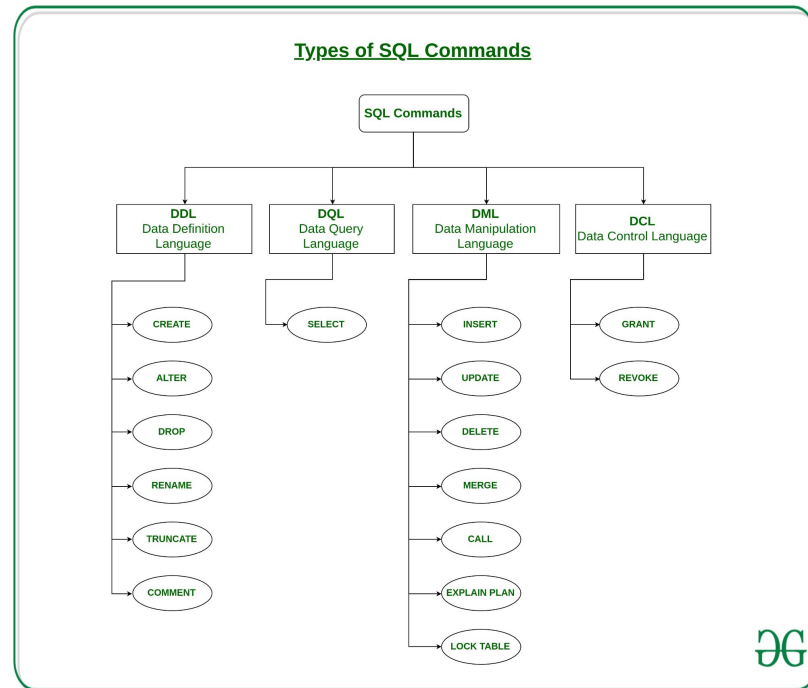
Relational databases - the database schema

- The database schema defines the structure of the data and the relationship among different tables



Relational databases - SQL

- SQL is a standardized language to interact with relational databases
- SQL allows you to:
 - Modify the database structure
 - Query the data
 - Manipulate the data
 - Manage access to the database



Relational databases - SQL Joins

Combining Data Tables – SQL Joins Explained

A JOIN clause in SQL is used to combine rows from two or more tables, based on a **related column** between them.

Table 1

1		
2		

Table 2

1		
3		
4		

Outer Join

1				
2				
3				
4				

Inner Join

1				

Left Join

1				
2				

Union

1		
2		
1		
3		
4		

Cross Join

1			1	
1			3	
1			4	
2			1	
2			3	
2			4	

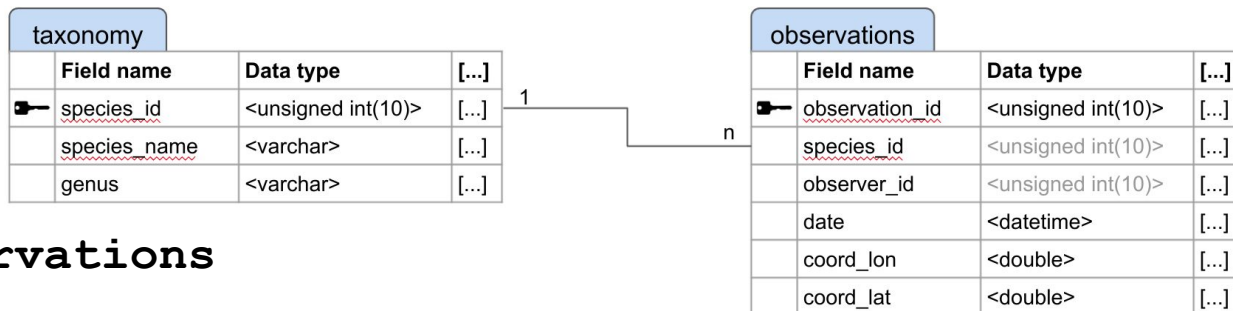
Relational databases - some more SQL keywords

SELECT	Select data (entire tables or specific columns)
WHERE	Filter by rows
ORDER BY	Sort results by column in ASC or DESC order
GROUP BY	Group results by column values and apply summary functions (MIN () , MAX () , AVG () ,...)
AND, OR, NOT	Combine logical statements

see <https://www.w3schools.com/sql/> for more

Relational databases - a query example

Aim: From our example database on slide 6, select all observations of *Circus* above a latitude of 50°



```
SELECT * FROM observations
```

```
LEFT JOIN taxonomy
```

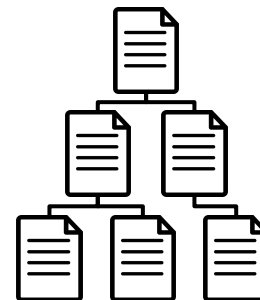
```
ON observations.species_id = taxonomy.species_id
```

```
WHERE taxonomy.genus = "Circus"
```

```
AND observations.latitude > 50;
```

NoSQL databases

- Umbrella term for different DBMS designs with a non-relational approach, e.g. document-based, key-value, graph or wide-column stores
- **Advantages:**
 - Highly flexible data schemas that can cope with unstructured data
 - High [horizontal scalability](#)
 - Simple queries can be very fast
- **Drawbacks:**
 - No standardized querying language
 - Problems with concurrency



Application programming interfaces (API)

- An API allows communication with a database using specified protocols and services
- Main benefits are **abstraction** (APIs offer a stable interface to the database) and **isolation** (APIs do not expose the database itself to the client)
- APIs enable a connected network of database and applications
- An example: the [GBIF web API](https://api.gbif.org/v1/occurrence/search?taxonKey=2480481)

<https://api.gbif.org/v1/occurrence/search?taxonKey=2480481>

https://api.gbif.org/v1/occurrence/search?continent=EUROPE&country=SE&axon_key=2480481&year=1900,2021

Where do these different technologies fit into a global biodiversity data landscape?



Practical

Work through the R practical [databases.md](#)

Further readings

Publications:

Kattge, J. et al. 2011. A generic structure for plant trait databases. - Methods Ecol Evol 2: 202–213.

McIntosh, A. C. et al. 2007. Database design for ecologists: composing core entities with observations. - Ecological informatics 2: 224–236.

Schulman, L. et al. 2021. The Finnish Biodiversity Information Facility as a best-practice model for biodiversity data infrastructures. - Sci Data 8: 137.

Weigelt, P. et al. 2020. GIFT – A Global Inventory of Floras and Traits for macroecology and biogeography. - J Biogeography 47: 16–43.

List of biodiversity databases:

https://en.wikipedia.org/wiki/List_of_biodiversity_databases

<https://ecologicaldata.org/find-data>

Technical background:

<https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5>

<https://www.ibm.com/cloud/blog/sql-vs-nosql>