Web Programming

Databases part 1 basics

Leander Jehl | University of Stavanger

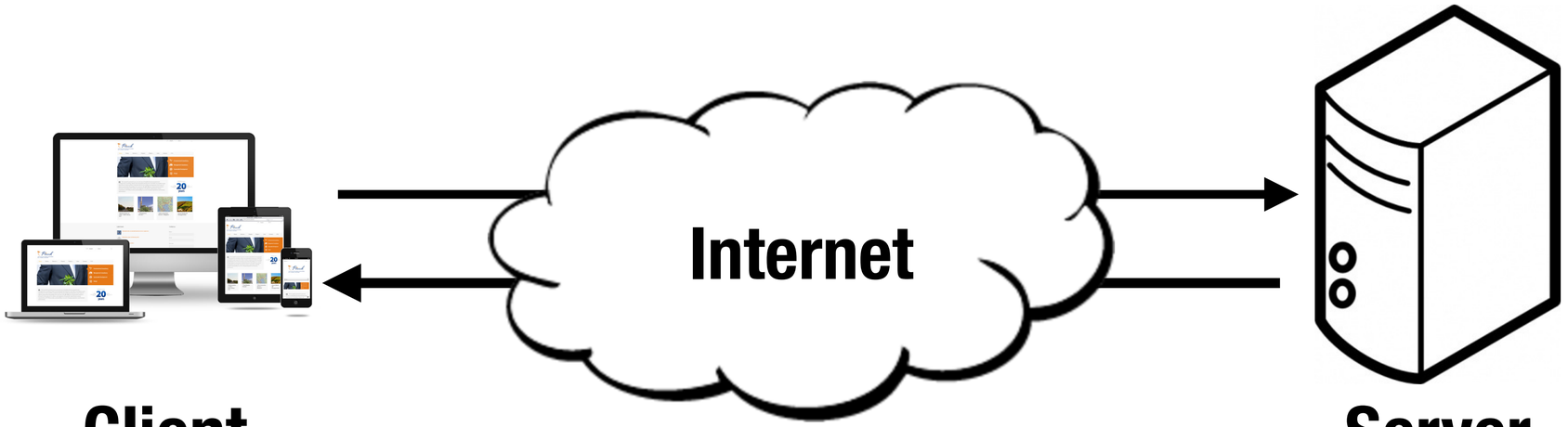
Learning goal:

- Advantages of using a database
- Query a database (using SQL)
- Create a database schema (using SQL)

Storing data

Files

Database



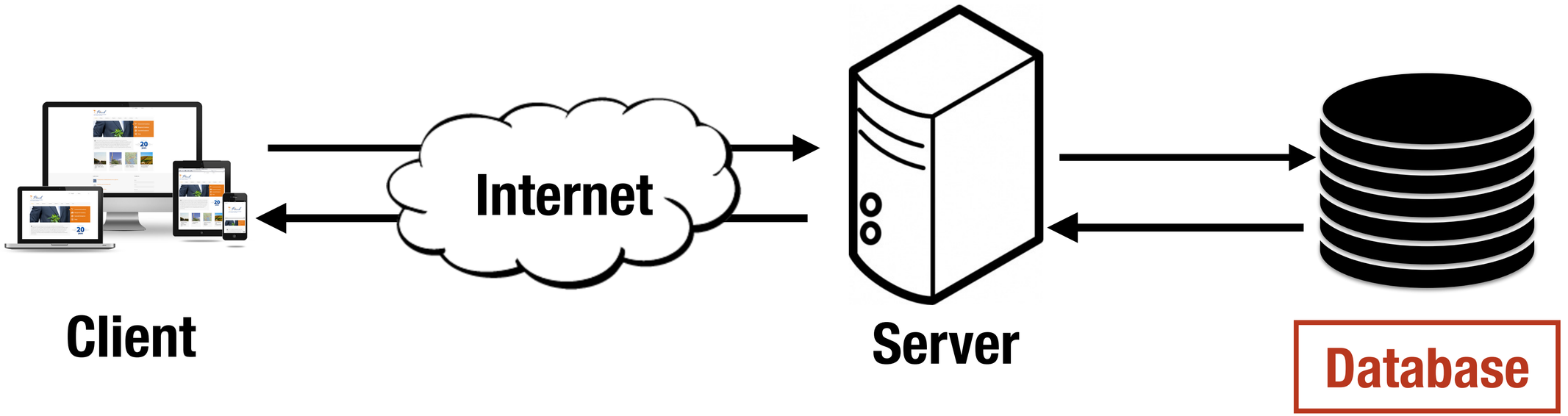
Client

Server

Cookie

Session

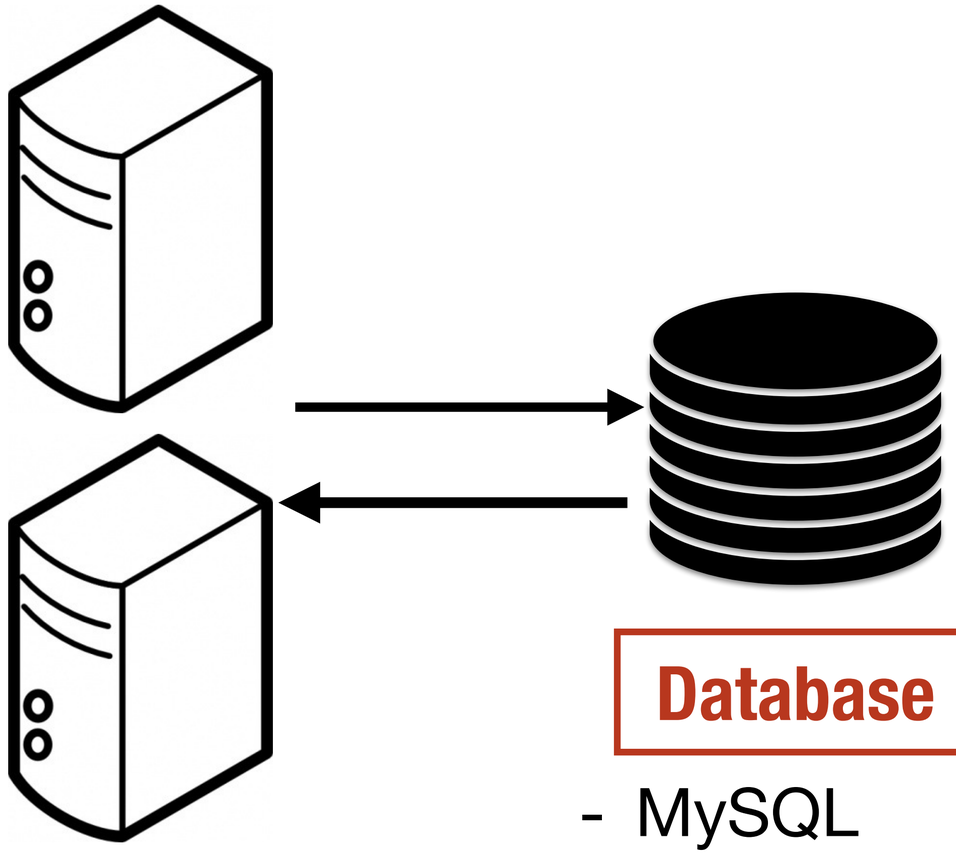
Architecture



Architecture

- Database server:
  - maintains state
  - stays consistent

- Web servers:
  - process client request
  - access state from database servers
  - may cache state but otherwise stateless



Web Server

Database

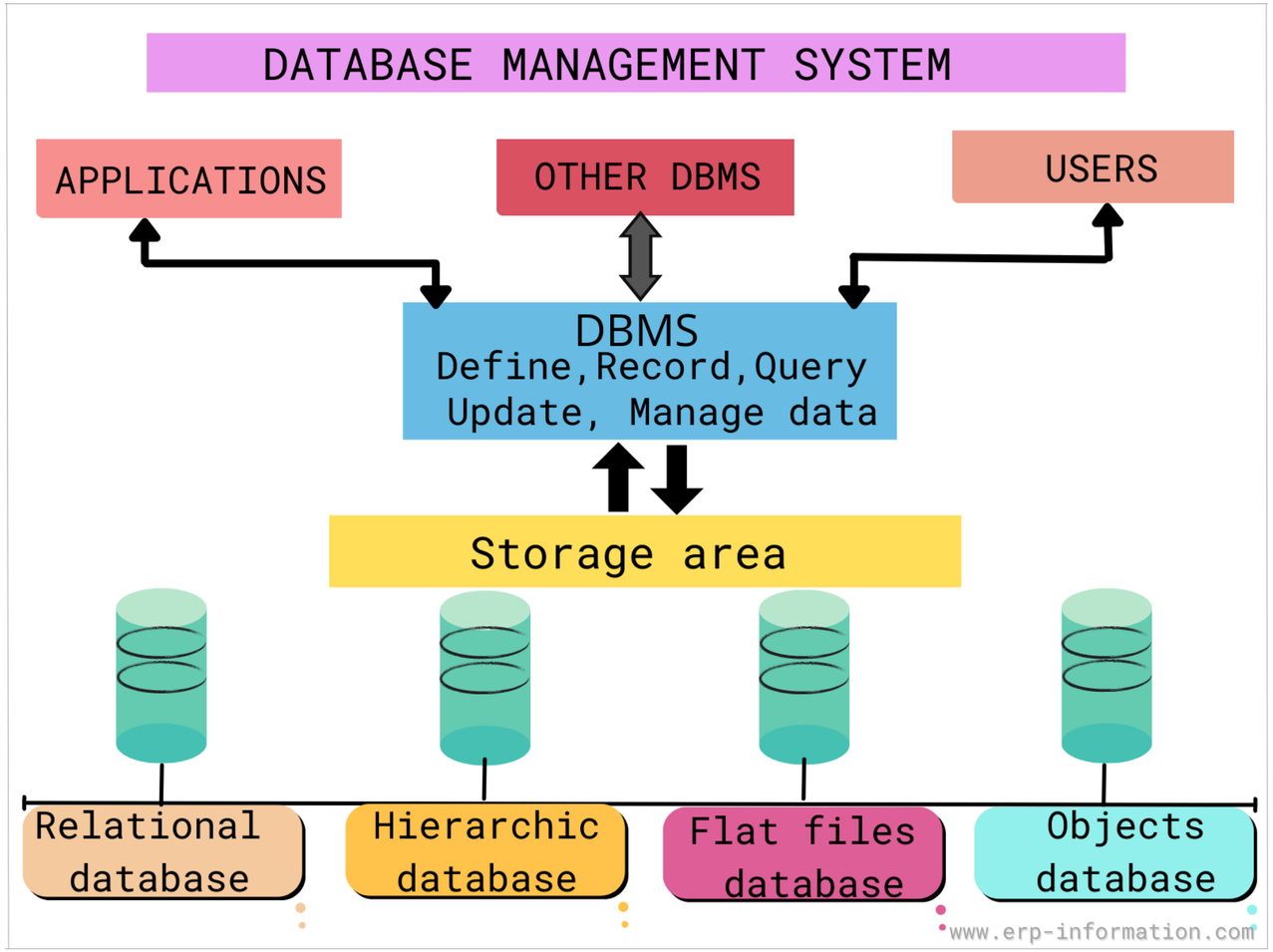
- MySQL
- PostgreSQL
- SQL Server

Database

- Provides a standardized way to store and access data
- User does not care about
  - Files
  - Data format
  - Execution paths
  - Concurrent access
  - Crash recovery

Database management system (DBMS)

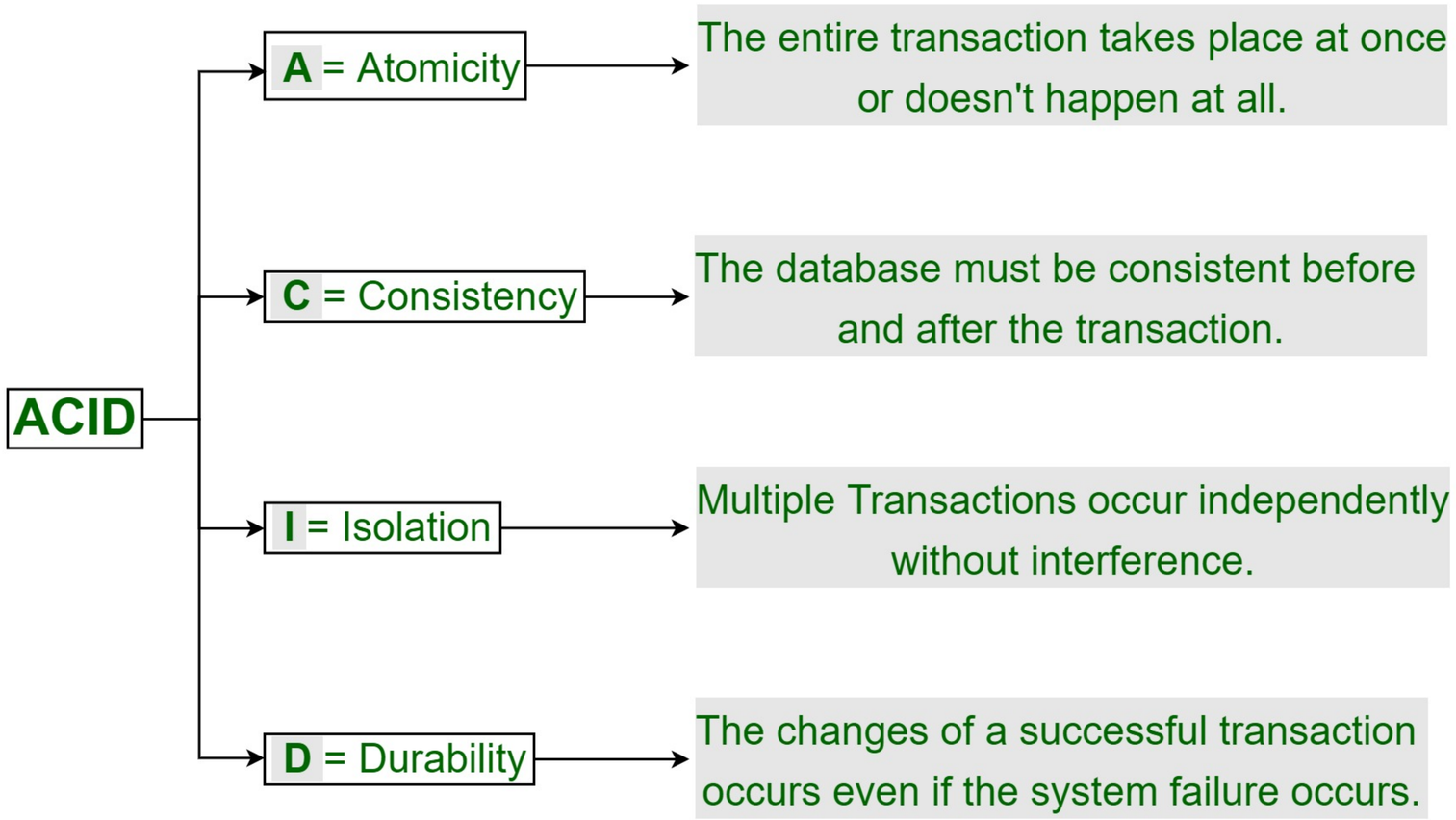
- Software designed to assist in maintaining and utilizing large collections of data.



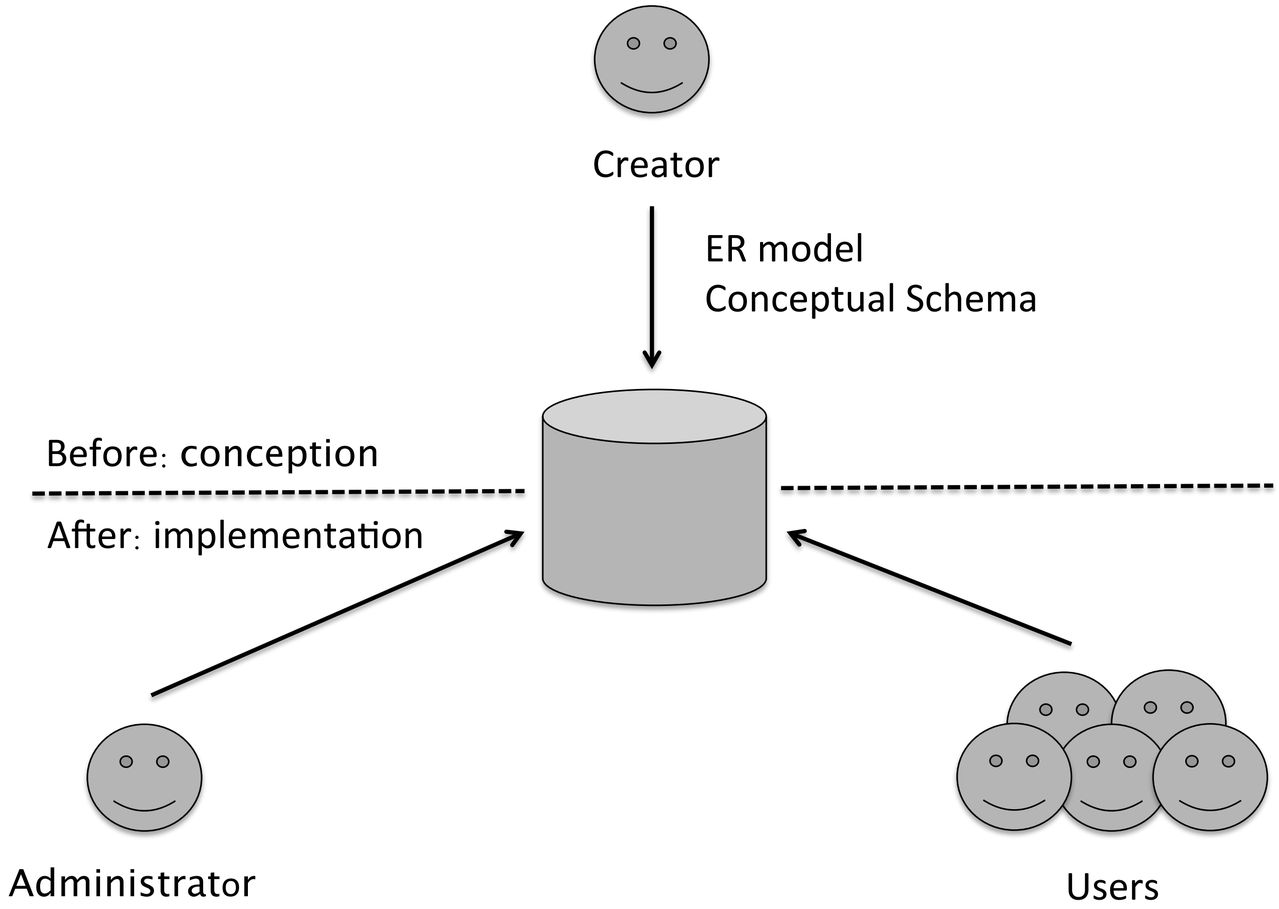
Database management system (DBMS)

- DBMS allow to store and access data:
  - in a shared context: concurrent access by multiple users
  - in an instable context: able to deal with system failures
- To DBMS users, managing a database seems:
  - Private: it is not affected by other users
  - Stable: is is not affected by system problems and failures

Database management system (DBMS) - ACID Properties



Database professions



Database professions

Creator:

- Identify needs
- Identify data to be stored
- Choose a structure
- Create tables

Administrator:

- Server management
- Access rights management
- Maintenance

User

- Query the data
- Insert data
- Update data

Designing a database

Description ≠ Data

The structure of the database (schema) should be distinguished from the data (content).

A DB's schema:

- Describes the structure
- Specified at design stage

Designing a database

A DB’s schema describes:

- What are the entities?
- What are properties of the entities?
- What are relations between entities?

Designing a database

Employee database example

- Employee
- Salary
- Department
- Manager
- Room

A DB’s schema describes:

- What are the entities?
- What are properties of the entities?
- What are relations between entities?

Designing a database

Employee database example

- Employee
- Salary
- Department
- Manager
- Room

Entities

Designing a database

Employee database example

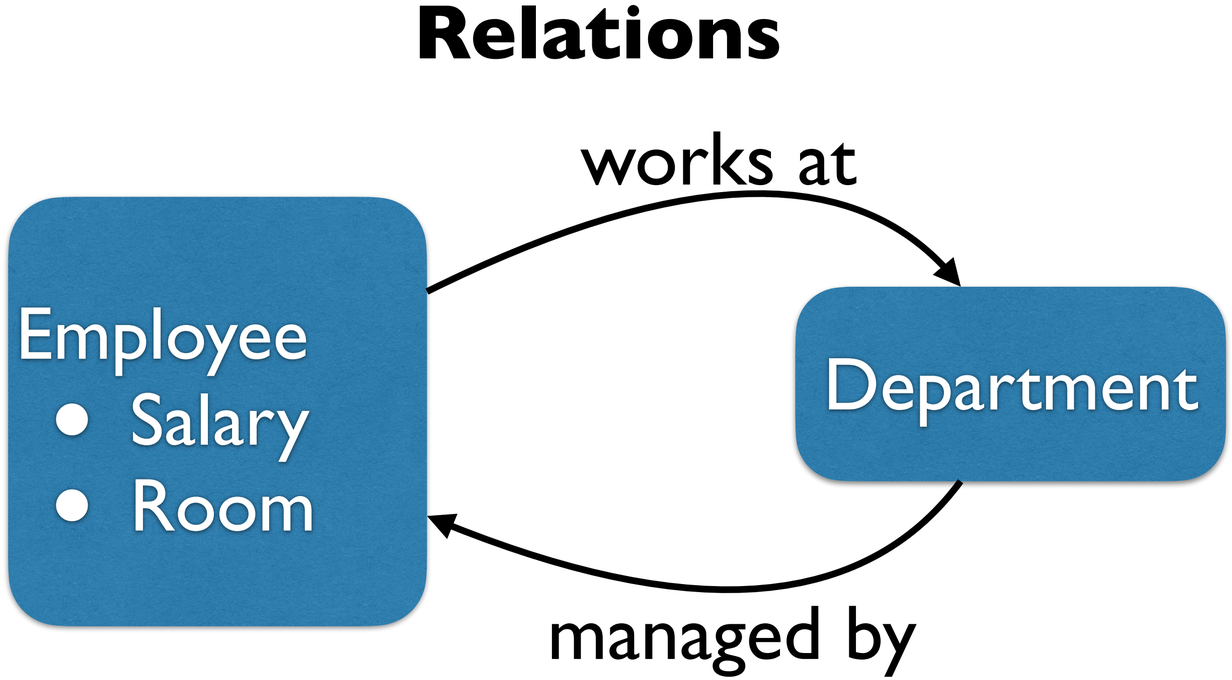
- Employee
- Salary
- Department
- Manager
- Room

Properties

Designing a database

Employee database example

- Employee
- Salary
- Department
- Manager
- Room

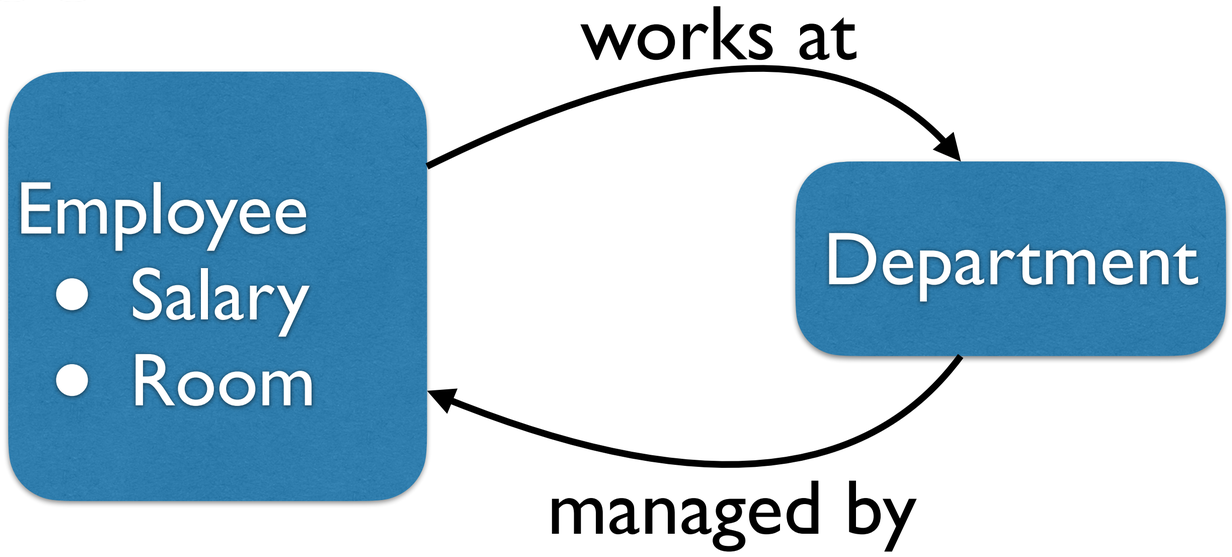


Designing a database

Employee database example

- Employee
- Salary
- Department
- Manager
- Room

Relations



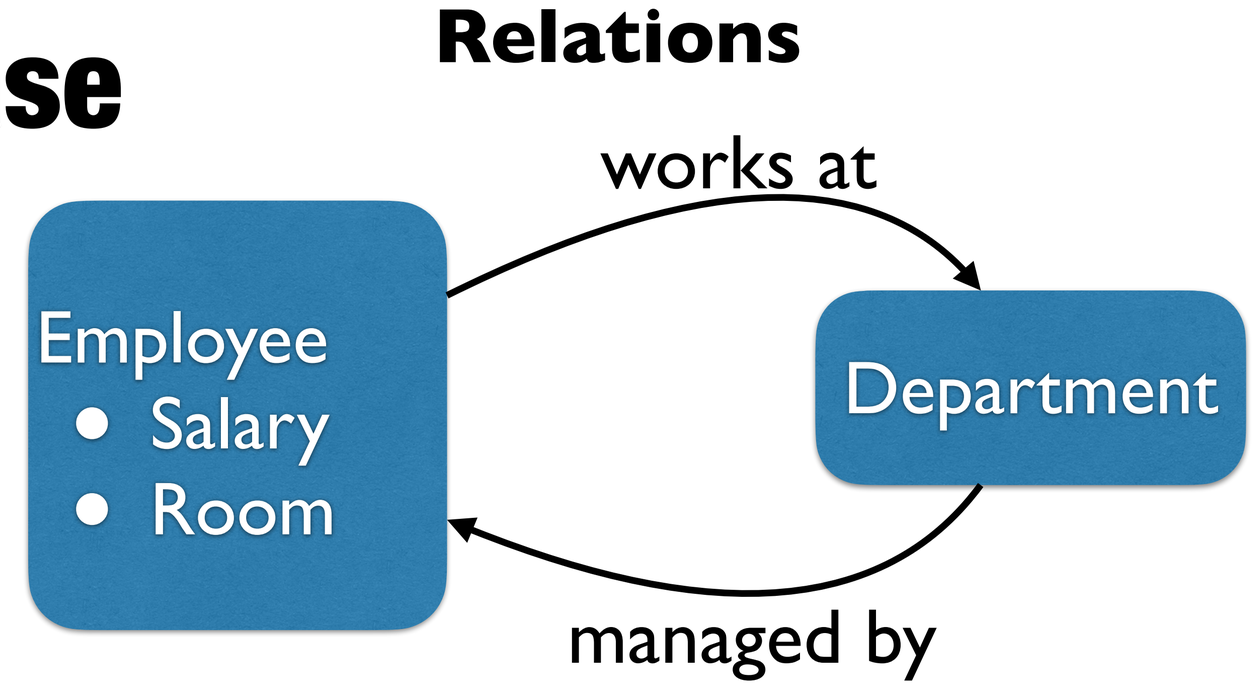
Type of the relation?

- **Can one employee only work at one department?**
- **Can one employee only manage one department?**
- **Can one department be managed by two employees?**

Designing a database

Employee database example

- Employee
- Salary
- Department
- Manager
- Room



Type of the relation?

- **One to one**
- **One to many**
- **Many to many**

Database tables

Databases store data in Tables:

- Defined column names
- Same columns on every row

Can add information and constraints to column, e.g. type, length, non-empty

Postcodes

number	city
"0001"	"Oslo"
“4036"	“Stavanger”
"4041"	“Hafrsfjord"
"7491"	"Trondheim"
"9019"	"Tromsø"

SQL Databases store tables

- Entities -> Tables
- Properties -> Columns

Each row in the table is one instance of the entity

How do we model relation?

Employee

ID	Name	Salary	Office
1234	“Tom"	50k	E123
1235	”Bjørn”	?	E245
1345	“Ida”	60k	

SQL Databases store tables

Department

Name
“Engineering”
“HR”
“Engineering 2”

Employee

ID	Name	Salary	Office	Deptment
1234	“Tom"	50k	E123	
1235	”Bjørn”	?	E245	
1345	“Ida”	60k		

Need unique identifier for relation.

Identifier should not change.

SQL Databases store tables

Department

ID	Name
1	“Engineering”
2	“HR”
3	“Engineering 2”

Employee

ID	Name	Salary	Office	Deptment
1234	“Tom"	50k	E123	1
1235	”Bjørn”	?	E245	2
1345	“Ida”	60k		1

SQL Databases store tables

Department

ID	Name
1	“Engineering”
2	“HR”
3	“Engineering 2”

Employee

ID	Name	Salary	Office	Deptment
1234	“Tom"	50k	E123	1
1235	”Bjørn”	?	E245	2
1345	“Ida”	60k		1

Many to many relations should have their own table!

SQL syntax

CREATE TABLE

- Create a table with row names:

```
CREATE TABLE department (ID, name);
```

```
CREATE TABLE <tablename>
(<rowname>,<rowname>, ...);
```

Department

ID	Name
1	“Engineering”
2	“HR”
3	“Engineering 2”

INSERT

- Insert a row into a table

```
INSERT INTO department (ID, name) VALUES (0, "Engineering");
```

```
INSERT INTO <tablename> (<rowname>,<rowname>) VALUES (value, value);
```

- Insert multiple values at once

```
INSERT INTO department (ID, name) VALUES (0, “Engineering”), (1, "HR");
```

SELECT

- Select named columns

```
SELECT ID, name FROM department;
```

- Select all columns

```
SELECT * FROM department;
```

- Select rows with specific values

```
SELECT name FROM department WHERE ID = 0;
```

DELETE & UPATE

- DELETE rows with specific values

```
DELETE FROM department WHERE name = 'Engineering' AND ID = 0;
```

- UPDATE rows with specific values

```
UPDATE department SET name = 'Subsea' WHERE ID = 0;
```

DELETE & UPATE

- DELETE rows with specific values

```
DELETE FROM department WHERE name = 'Engineering' AND ID = 0;
```

- UPDATE rows with specific values

```
UPDATE department SET name = 'Subsea' WHERE ID = 0;
```

Careful! DELETE and UPDATE can change multiple rows at a time.

```
UPDATE employee SET salary = salary + 10.000 WHERE departmentID = 0;
```

SQLite

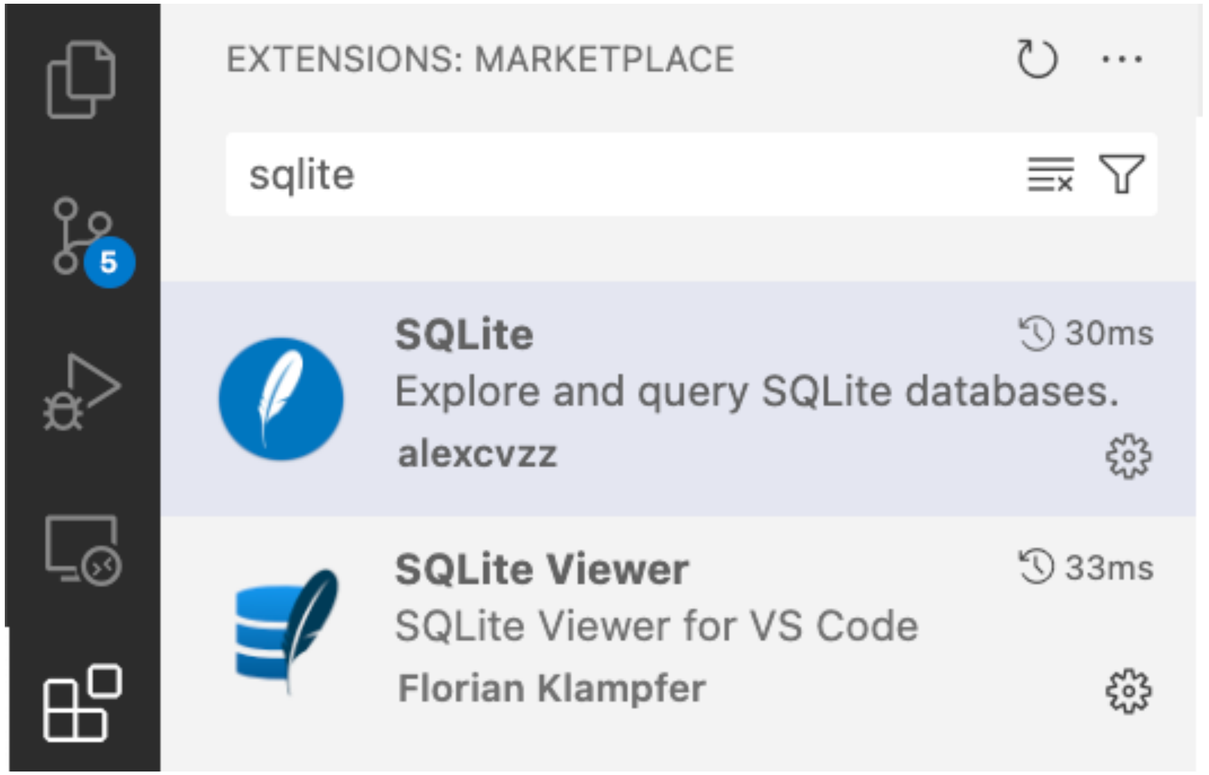
- Lightweight database:
  - Store database in a file
  - Good for prototyping and examples
  - Only for single webserver
  - Tutorial: https://www.sqlitetutorial.net/
  - Try it editor: https://www.sqlitetutorial.net/tryit/

Exercises #1, #2

github.com/dat310-2025/info/tree/master/
exercises/sql/basics

SQLite in VSCode

- Install VSCode extensions:
  - SQLite: run sql scripts and query database
  - SQLite Viewer: see tables in your database



Exercises #3

github.com/dat310-2025/info/tree/master/
exercises/sql/basics