Engineering Databases

Lecture 1

October 19, 2022

M. Saeed Mafipour & Mansour Mehranfar

Course Information

- 3 ETCS
- On Wednesdays, 15:00 - 16:30, in presence at TUM:
  - Computer room: N0199
  - https://campus.tum.de/tumonline/co_loc_roomfinder.doRedirect?raumKey=7566
- Lectures will not be recorded
- Theory & Exercises
- Midterm assignment
- Register for the lecture and exam
- Exam
  - In Early February 2023
  - Closed book
  - 60 minutes

Course Topics

- Database basics
  - Terms / Motivation / Characteristics
  - History
  - Types of databases
- Database design
  - Entity-relationship model (ERM)
- The relational model
  - Transforming conceptual schema to relational schema
  - Relational algebra

Course Topics

- Relational Query Language (SQL)
  - Data types
  - Definition of schema
  - Elementary manipulations of data
- Relational design
  - Keys
  - Functional dependencies
  - Normal forms
- Transactions
- Database in software architecture
- Indexing

What is a Database?

- “A set of information held in a computer”

(Oxford English Dictionary)

- “One or more large structured sets of persistent data, usually associated with software to update and query the data”

(Free On-Line Dictionary of Computing)

- “A collection of data arranged for ease and speed of search and retrieval”

(Dictionary.com)

Introduction

- Much data everywhere
  - More & more → need to store it in a structured way
  - Easy & fast access → Database Management Systems (DBMS)
- Examples
  - bank accounts
  - client databases
  - Web 2.0 web pages, e.g. Facebook, GoogleDocs, TUMonline
  - Google
  - German railway
  - Product Data Management (PDM)
  - Enterprise Resource Planning (ERP)

Introduction

- We distinguish the terms
  - **Database (DB)** = all the data that is stored
  - **Database Management System (DBMS)** = program running on the computer
    - access to database
    - controls consistency, concurrency, integrity, and security
    - ensures fast access through proper organization
  - Sometimes these two terms are mixed

Motivation

- What problems occur if we don't use a DBMS but instead one or many files?
- Redundancy lead to inconsistency
  - If data is kept in isolated files, often the same information is stored multiple times
  - Problem: if the information is changed at one place it has to be updated at all other places
- Missing restricted access
  - It is not possible to link data in a file on a logical basis with data in another file
    - DB: use a query language
- Missing multi-user access
  - Problems occur if multiple users access the same data (modifications!)
  - Word-File → last modifier “wins“
  - Update during read access → no notification → old data

Motivation

- Loss of data
  - Recovery of a consistent state of the database
  - DB: actions to recover / data is always safe
  - File systems: only periodical storing

- Loss of integrity
  - Integrity conditions / rules, e.g. students can do an exam 2 times at maximum
  - Certain modification actions should be refused by the system
  - Database systems: transactions are realized only if they result in a consistent state

Motivation

- Security issues
  - Not all users should have read / write access to all data
  - DBMS: fine-grained user access control = rights management
  - Example roles:
    - Administrator: read / write access to all data
    - Lecturers: read / write access to grades
    - Students: read access to own grades

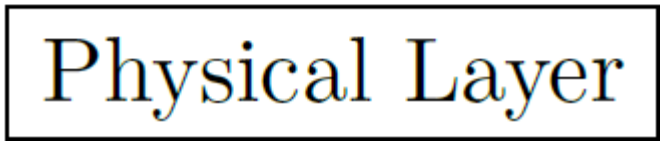
When do we use a database management system?

- Criteria
  - Large amount of data (> 50 MB)
  - Many users (> 5) with
    - different rights
    - simultaneous access
  - Filter data by means of logical conditions
  - Consistency rules → transactions
  - Loss of data → critical

Abstraction layers/levels

- Partial sets of entire data set
- Tailored to the needs of specific users

- how data is logically organized → schema



- how data is stored on disk

Data models

- A **schema** defines the structure of the data that can be stored
  - contains no data
  - e.g. relational model:
    - which “tables" exist
    - what “columns" does it have

- An **instance** is one set of data that complies to the schema, e.g.
  - all students of TUM
  - all students of LMU

Data models

- A **data model** describes how we can model the schema in the database
  - Network model (historical)
  - Hierarchical model (historical)
  - relational model (98%)
  - object-oriented model
    - New as native model for databases
    - powerful, but seldom used
    - Performance issues
  - mixture: object-relational model

Database design

- Do a proper design→ then you'll have an efficient database
- Easy maintenance, future extensibility
- Use a formal approach (methodology)
- Use pen and paper first

Entity Relationship Model

- A graphical model, which includes the following elements
  - Entities
  - Relationships
  - Attributes

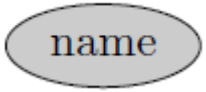
- Entities
  “Well-defined physical objects or mental concepts of the world to be modeled“

  - similar objects are abstracted to object types (= entity types / entity sets)
  - often stated as nouns
  - drawn as rectangle

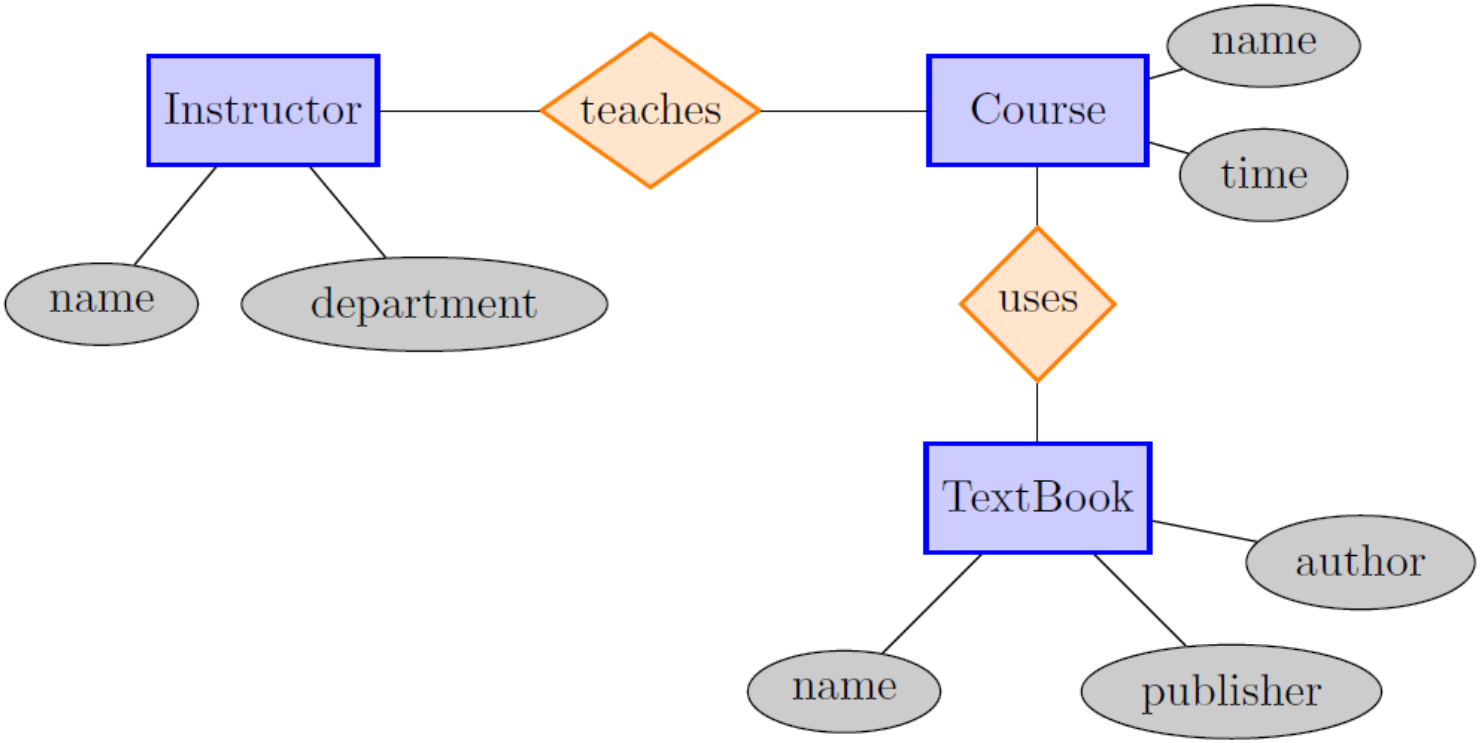
Relationships / Attributes

- Relationships
  - model relationships between entity types
  - drawn as diamond
  - often stated as verbs

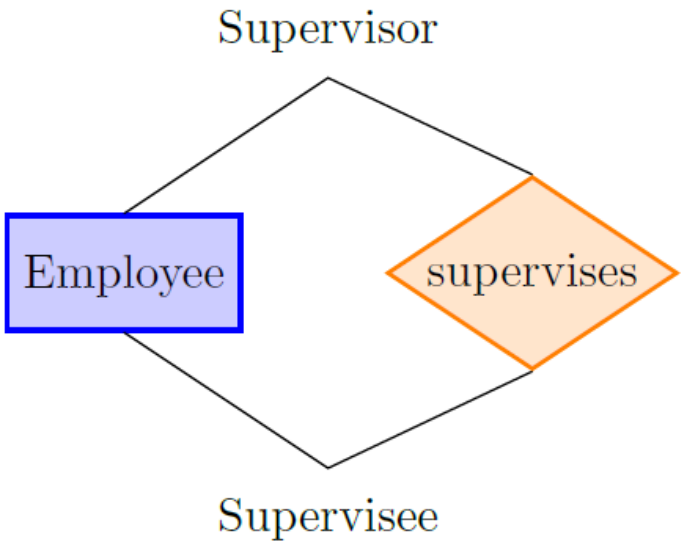
- Attributes
  - characterize entities and relationships
  - drawn as ellipse



Example: An Entity Relationship Model

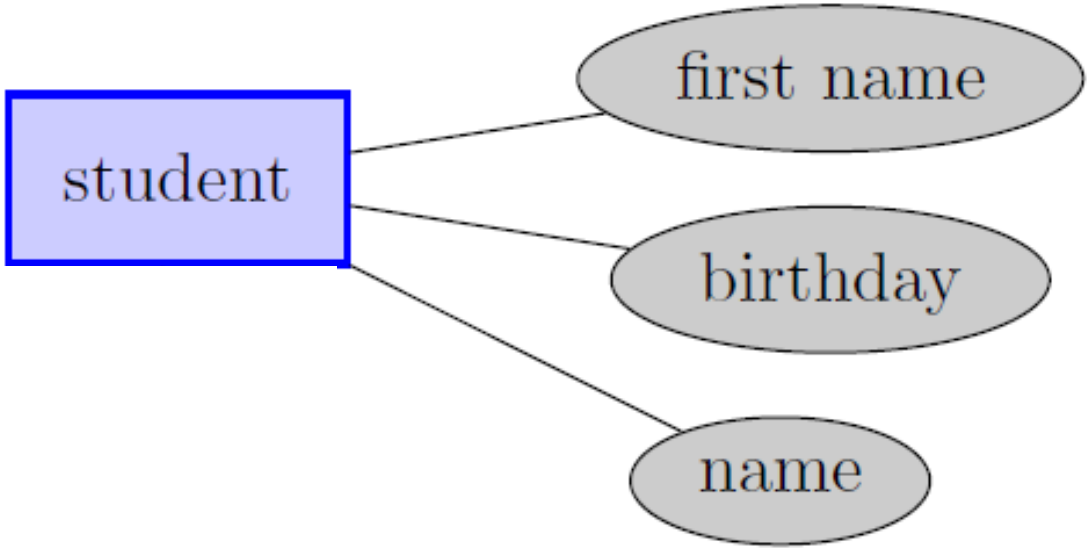


Recursive Relationships

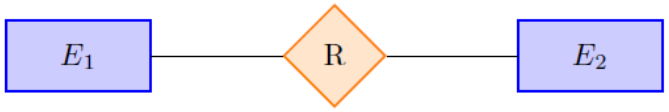


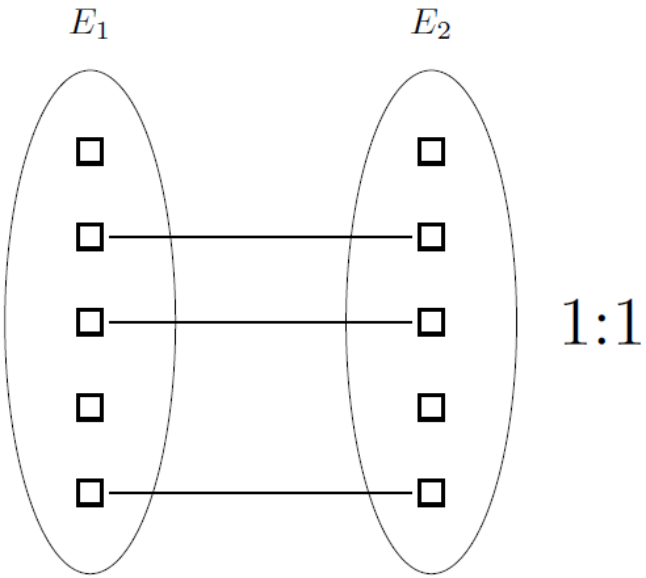
Keys

Primary key candidate = minimal set of attributes that unambiguously identifies one entity among all entities of the same type



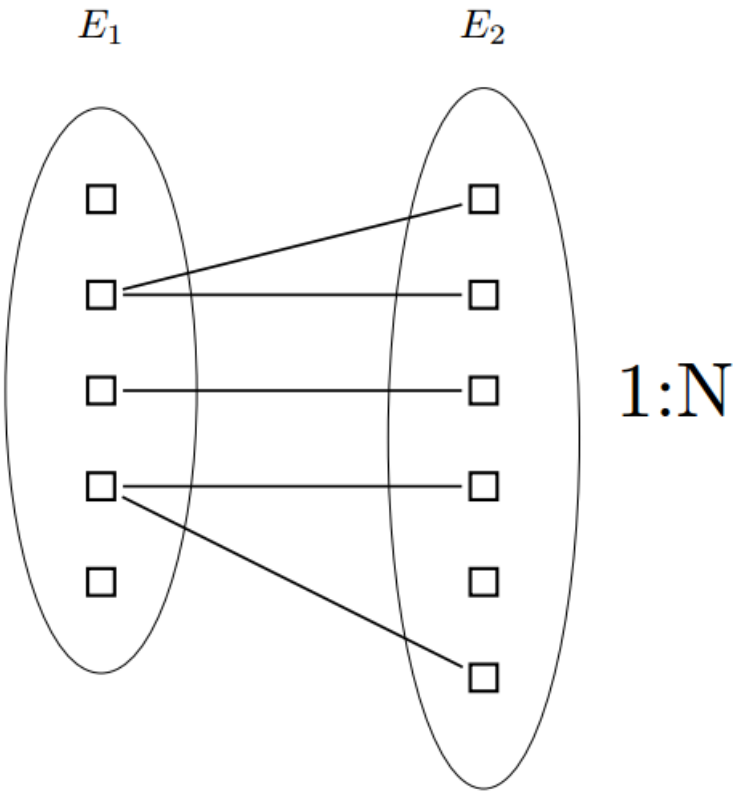
Cardinalities



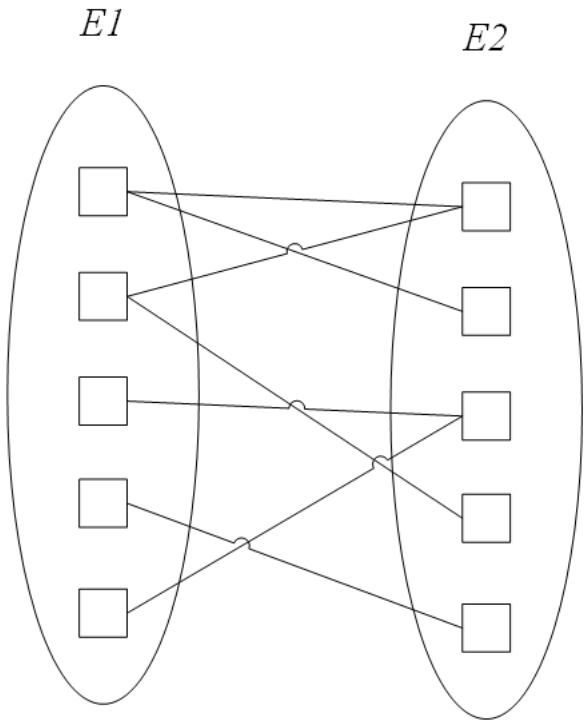


1:1

Cardinalities



1:N



M:N

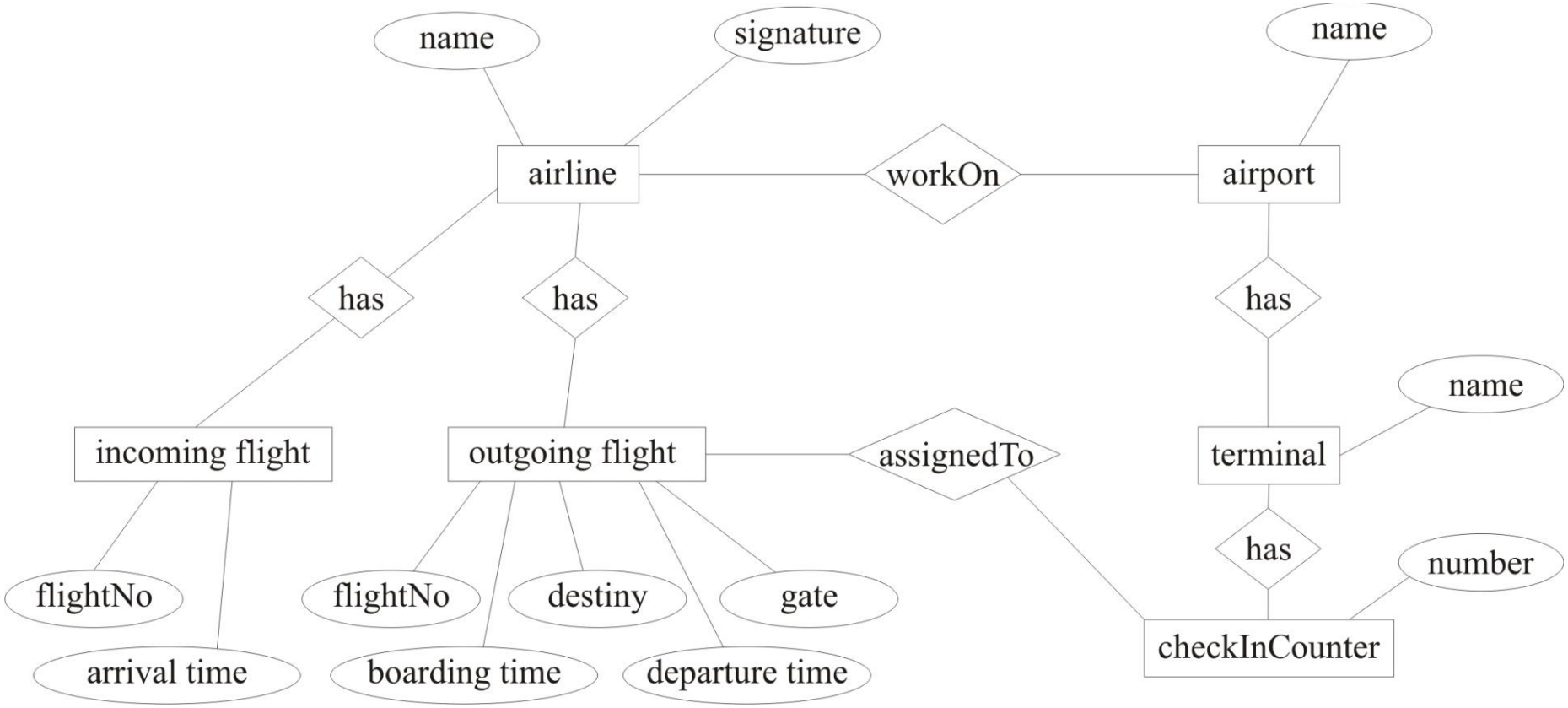
Exercise: Entity-Relationship-Diagram

- Your task is to define the database structure for an airport. To capture the “universe-of-discourse” you use the entity-relationship-model. From the interviews with your clients you know the following:

- A number of airline companies work at the airport. The airline companies have a name and a signature (two letters) to identify their flights. Each airline company has a number of outgoing and incoming flights. Each outgoing flight has a flight number, a destiny, a gate, a boarding time and a departure time. Each incoming flight has a flight number, an origin, and an arrival time. The airport has different terminals. Each terminal has a name and a number of check-in counters. Each check-in-counter has a number. For each outgoing flight there is a specific check-in counter.

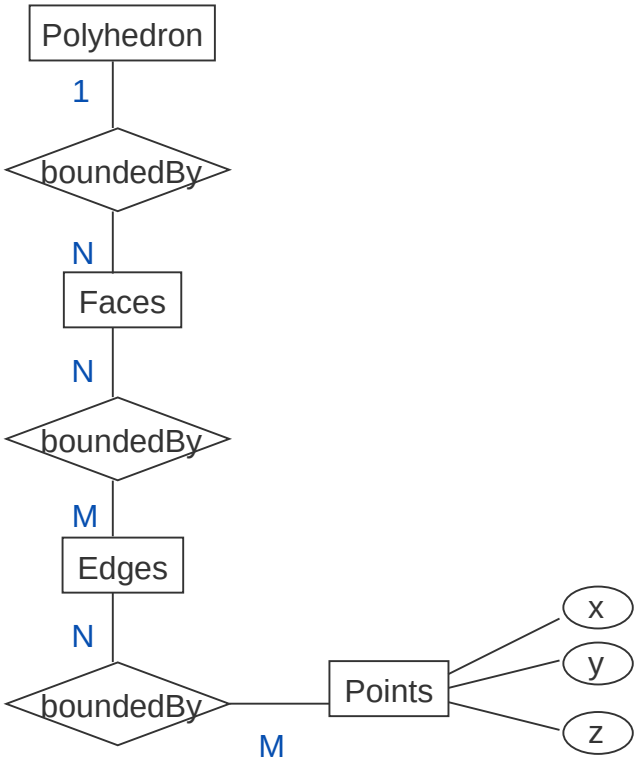
- Draw the entity-relationship-diagram.

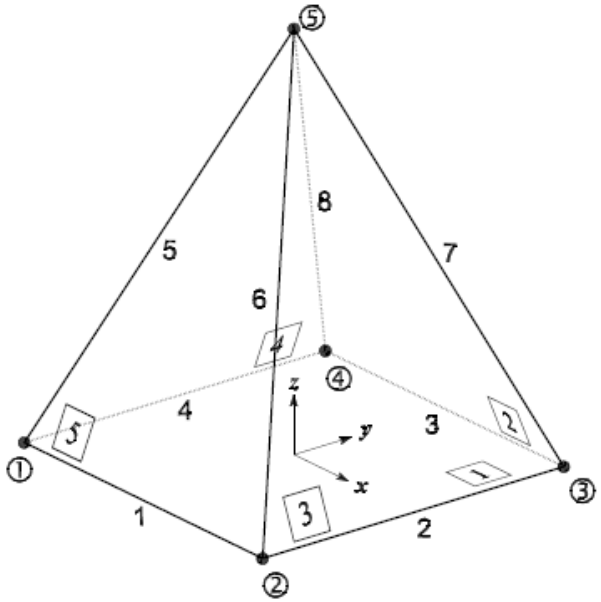
Airport example: Cardinalities



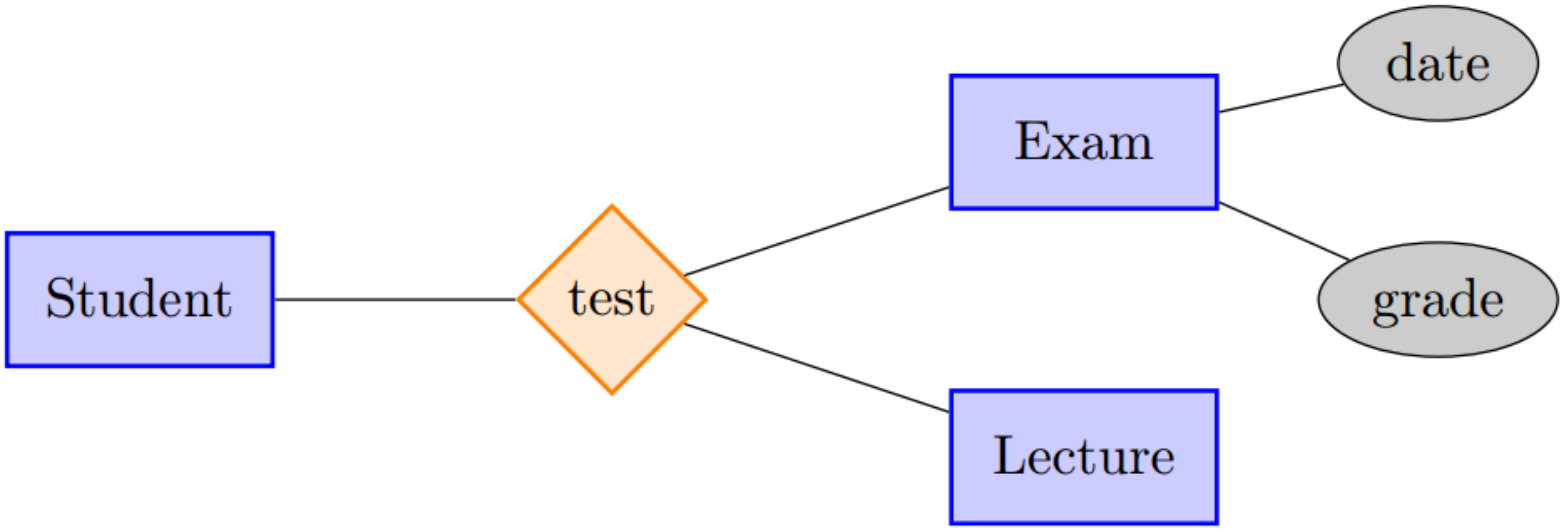
(Min, Max) notation - cardinality

- used to define minimum and maximum number of associated elements
- Example: Polyhedron





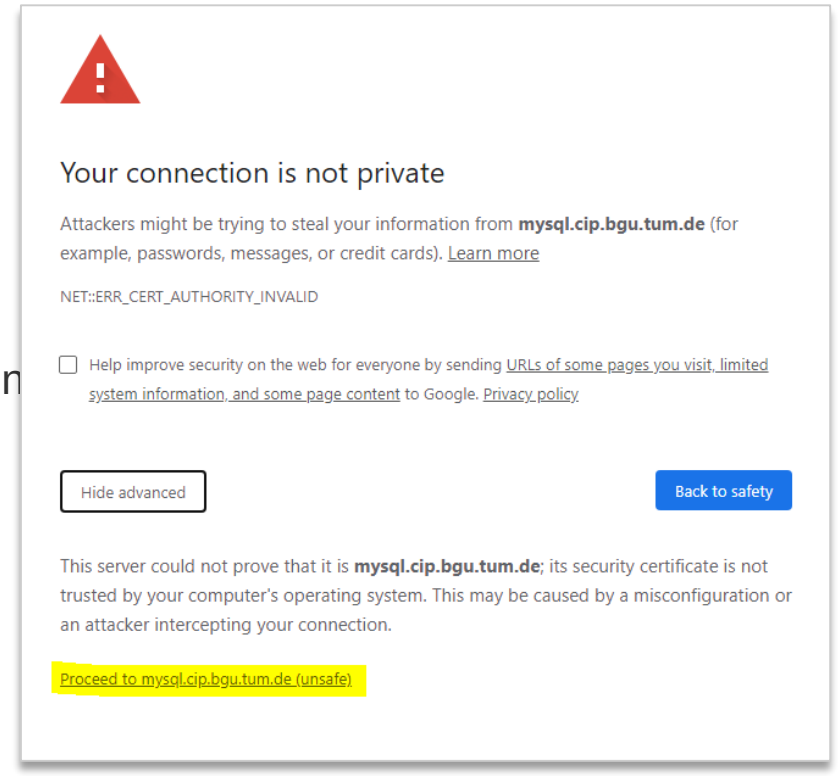
Another relationship type: N-ary



Test your Database Access

- Go to: https://mysql.cip.bgu.tum.de/DBadmin/
- Login: <Your Matriculation-Number> (with leadin
- Password: EngDB_WS2022

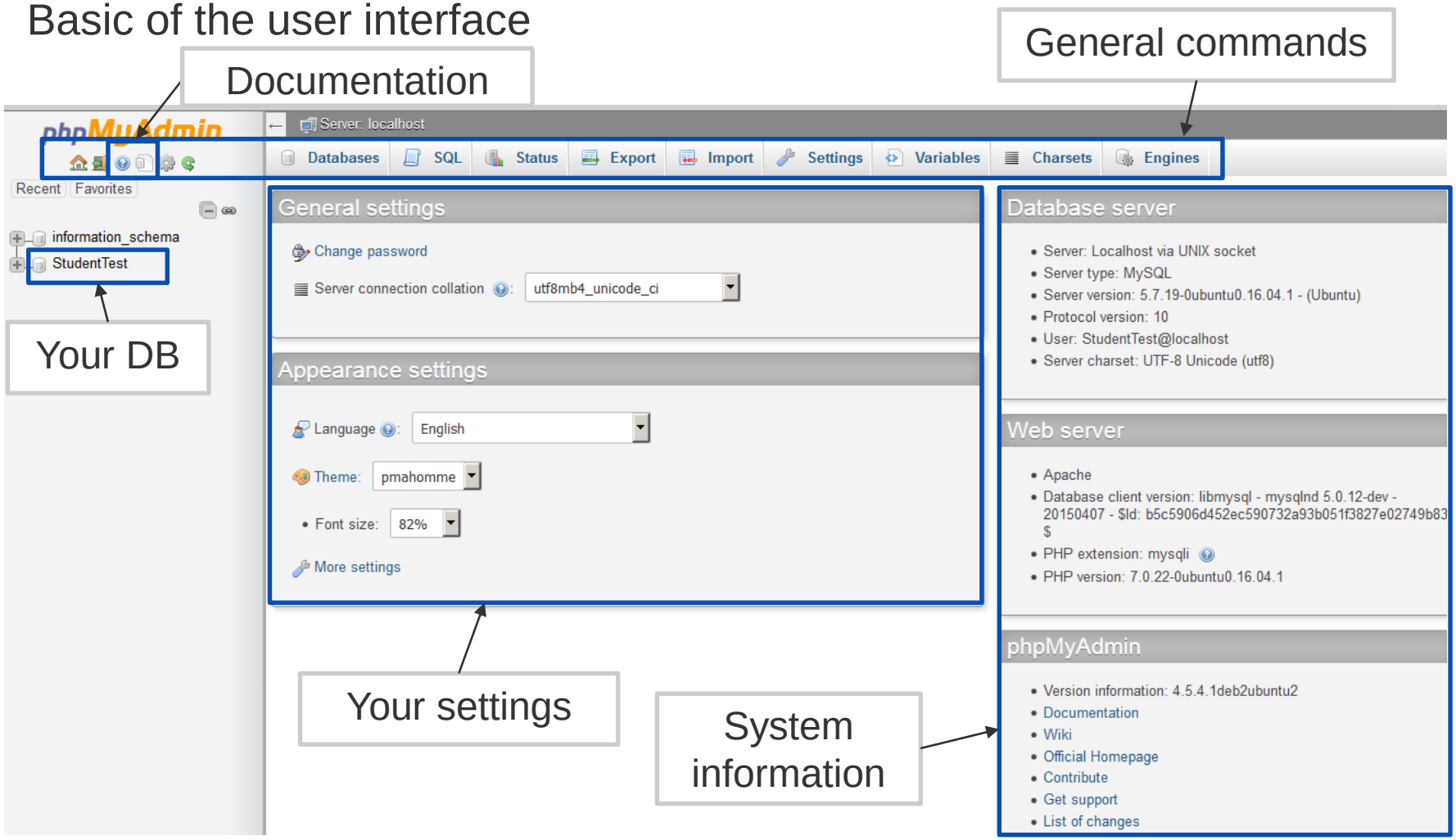
- phpMyAdmin
  - Provides access to a mySql-Database
  - MySQL Documentation https://dev.mysql.com/doc/refman/5.7/en/



Basic of the user interface

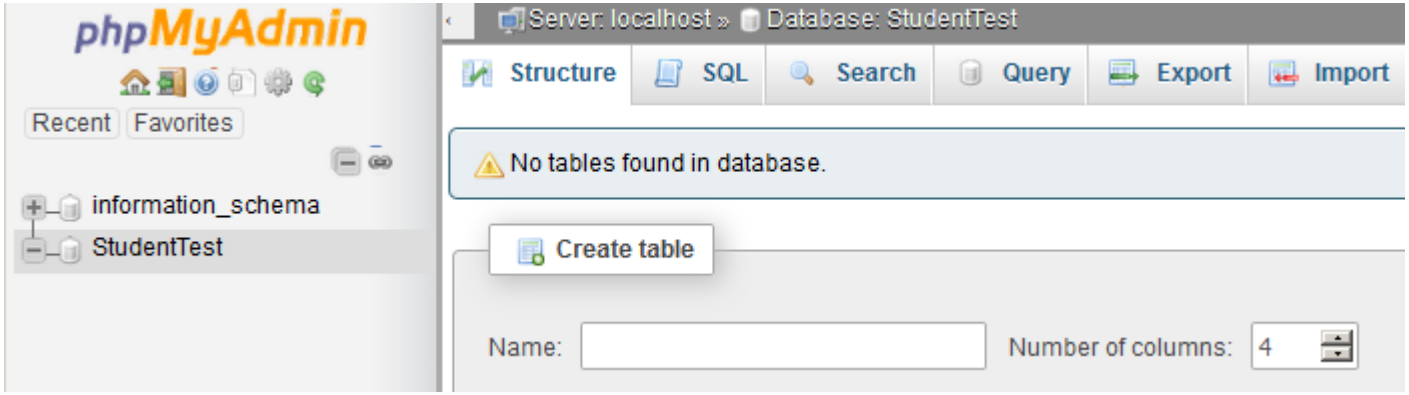
- Change language
- Log-In

Basic of the user interface



Basic of the user interface

- User interface provides all functions we can execute via SQL



- Try to create the Airport example (slide 24) tables using the interface
  - Rectangles are tables, ellipses are columns (attributes)
  - Ideas about the diamond symbol?
  - Be aware, we still do not implement consistency rules

End of Lecture

Thank you for your attention