Data Modelling/Data Base Systems
VU 184.685/VU 184.686, WS 2020

Introduction

Anela Lolić

Institute of Logic and Computation, TU Wien

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Acknowledgements

The slides are based on the slides (in German) of Sebastian Skritek.

The content is based on Chapter 1 of
(Kemper, Eickler: Datenbanksysteme – Eine Einführung).

For related literature in English see Chapter 1 of
(Ramakrishnan, Gehrke: Database Management Systems).

What do these companies have in common?

Motivation

Today we have

- a large amount of data ...

- ... in various areas

- distributed applications

- critical applications

Motivation

Today we have

- a large amount of data . . .

- . . . in various areas

- distributed applications

- critical applications

$\Rightarrow$ we need software for an efficient processing

Additional Software Requirements

problems that need to be solved:

- redundancy and inconsistency
- loss of data
- multiple users
- security issues
- integrity violations
- restricted access

lead to:

- complex software
- high development costs

source: https://xkcd.com/1906/

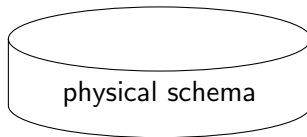# Data Base Management System (DBMS)

Definition (data base management system (DBMS))

A data base management system (DBMS) is a software for accessing data (stored in the DBMS).
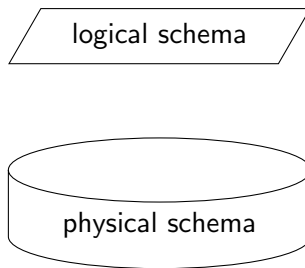
Definition (data base)

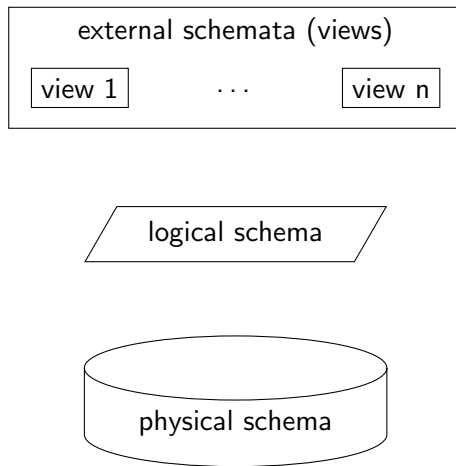The data stored in a DBMS is called data base.

Data Abstraction and Data Independence

physical schema

Data Abstraction and Data Independence

logical schema

physical schema

Data Abstraction and Data Independence

external schemata (views)

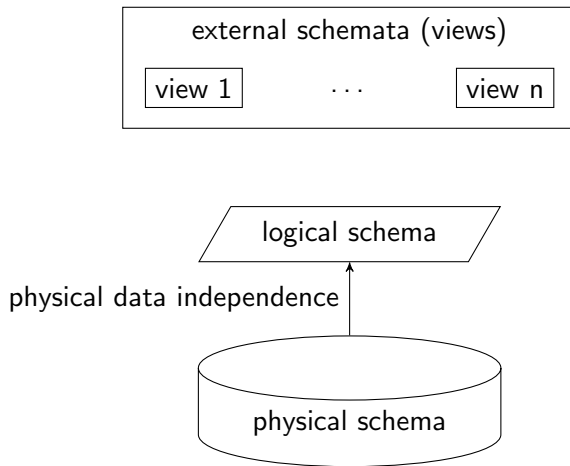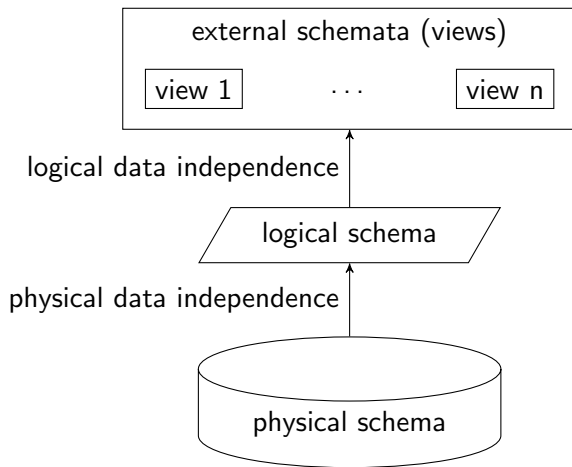view 1 $\cdots$ view n

logical schema

physical schema

Data Abstraction and Data Independence

Data Abstraction and Data Independence
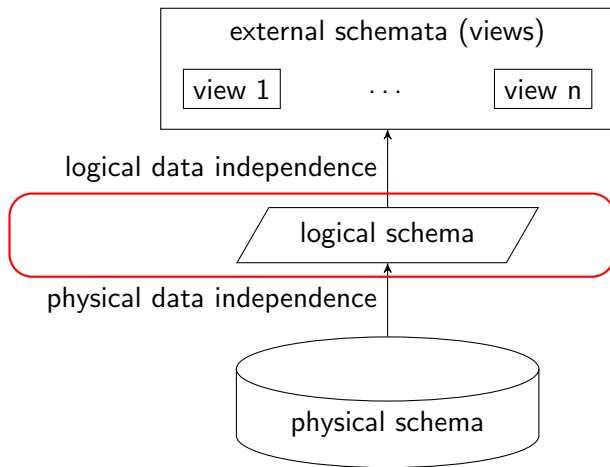
Data Abstraction and Data Independence

```
┌─────────────────────────────────────────────┐
│         external schemata (views)            │
│   ┌────────┐                    ┌────────┐   │
│   │ view 1 │        · · ·       │ view n │   │
│   └────────┘                    └────────┘   │
└─────────────────────────────────────────────┘
                        ↑
        logical data independence

        ╱────────── logical schema ──────────╲
                        ↑
        physical data independence

                physical schema
```
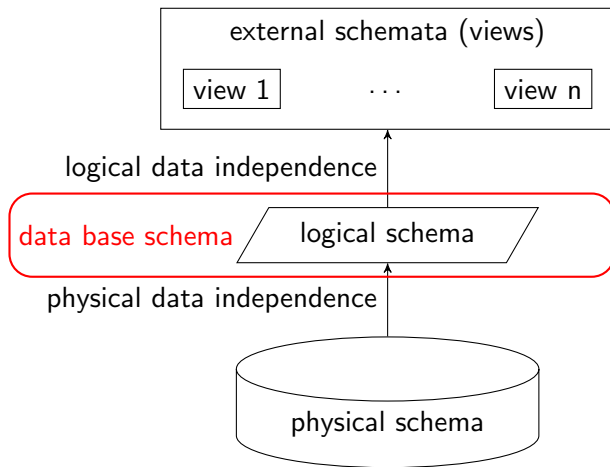
Data Abstraction and Data Independence


```
                external schemata (views)

              view 1        · · ·        view n


              logical data independence


data base schema /   logical schema   /

              physical data independence


                   physical schema
```

Data Base Schema and Instance

Definition (data base schema)

The data base schema determines the structure of the data.
(meta-data: data about the database.)

Definition (data base instance)

A data base instance is a specific data base that obeys the
structure described and satisfies all the conditions determined by
the data base schema.

Data Model

intuitively: A data model defines the language and the tools to describe the data base schema and to manipulate and interact with the data base instance.

Data Model

intuitively: A data model defines the language and the tools to describe the data base schema and to manipulate and interact with the data base instance.
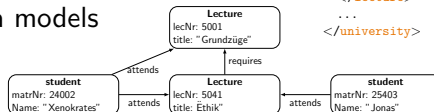
Data models:

- network model
- hierarchical model
- relational model
- object-oriented data models
- XML
- graph data models
- ...

student

matrNr	Name
24002	Xenokrates
25403	Jonas
...	...

attend

matrNr	lecNr
26120	5001
...	...

lecture

lecNr	title
5001	Grundzüge
5041	Ethik
...	...

```xml
<?xml version="1.0">
<university>
  <lecture lecNr="5001" title="Grundzüge">
    <attended_by>
      <student matrNr="24002" name="xenokrates"/>
      <student matrNr="25403" name="jonas"/>
      ...
    </attended_by>
    ...
  </lecture>
  ...
</university>
```

Data Model

Definition (data model)

A data model determines the applicable structures and operators. It consists of:

- a data definition language (DDL) for defining the schema

Data Model

Definition (data model)

A data model determines the applicable structures and operators.
It consists of:

- a data definition language (DDL) for defining the schema
- a data manipulation language (DML) for interacting with and manipulating the data base instance. Usually we distinguish
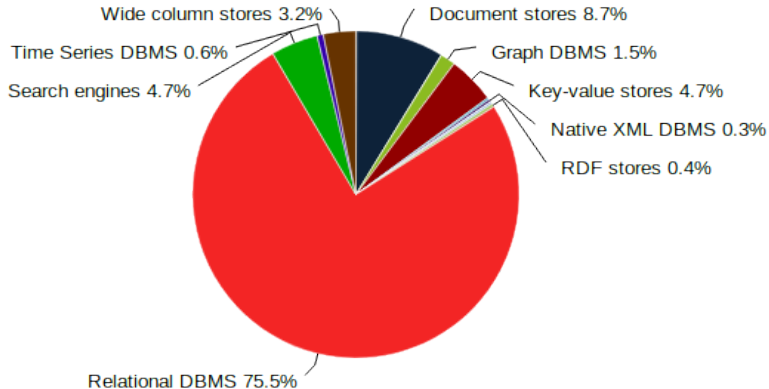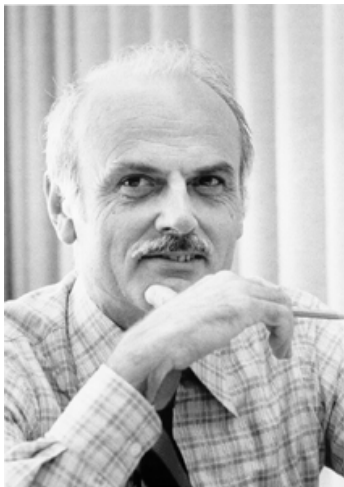
Data Model

Definition (data model)

A data model determines the applicable structures and operators.
It consists of:

- a data definition language (DDL) for defining the schema
- a data manipulation language (DML) for interacting with and
  manipulating the data base instance. Usually we distinguish

  - the query language
  - the actual data manipulation language (commands for
    inserting, deleting, . . .)

Data Models

Pie chart of data models:
- Wide column stores 3.2%
- Time Series DBMS 0.6%
- Search engines 4.7%
- Document stores 8.7%
- Graph DBMS 1.5%
- Key-value stores 4.7%
- Native XML DBMS 0.3%
- RDF stores 0.4%
- Relational DBMS 75.5%

© 2019, DB-Engines.com

The Relational Data Model

Edgar F. Codd (1923-2003)

A Relational Model of Data for Large Shared Data Banks in: Communications of the ACM, Volume 13, Issue 6 (June 1970) Pages: 377 - 387 (1970)

Turing Award (1981) $\approx$ Nobel Prize in Computer Science

The Relational Data Model

requirements according to E.F. Codd:

- data integration (consistent management of data)
- operations (storing, searching, changing)
- data dictionary (access to data description)
- user views (view depending on application)
- consistency surveillance (logical correctness of data)
- access control (data privacy)
- transactions (indivisible blocks of operations)
- synchronisation (multiple users)
- data backup (recovery after a crash)

Modern Relational DBMS

- DDL/DML: SQL (Structured Query Language)
- embedding in programming languages
- tools (e.g.. design, masks, interaction, ...)
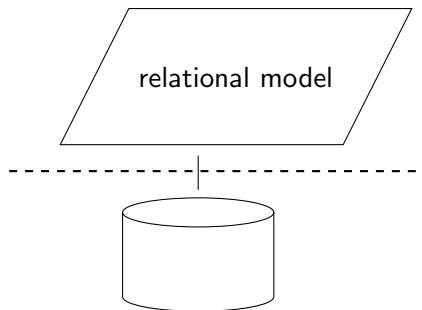- multiple users, security mechanisms

Oracle, IBM DB2, MS SQL Server, PostgreSQL, MySQL, IBM Informix, SAP Sybase, Teradata, SQLite, . . .

Modern Relational DBMS

- DDL/DML: SQL (Structured Query Language)
- embedding in programming languages
- tools (e.g.. design, masks, interaction, ...)
- multiple users, security mechanisms

Oracle, IBM DB2, MS SQL Server, PostgreSQL, MySQL, IBM Informix, SAP Sybase, Teradata, SQLite, . . .

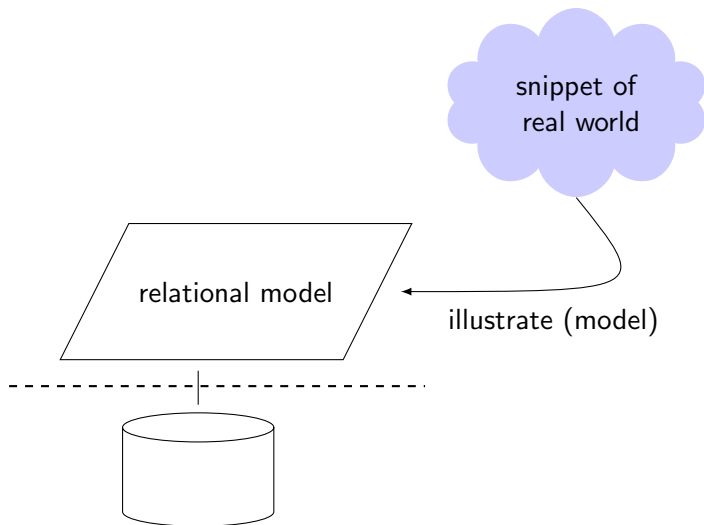355 systems in ranking, October 2019

	Rank		DBMS	Database Model	Score		
Oct 2019	Sep 2019	Oct 2018			Oct 2019	Sep 2019	Oct 2018
1.	1.	1.	Oracle	Relational, Multi-model	1355.88	+9.22	+36.61
2.	2.	2.	MySQL	Relational, Multi-model	1283.06	+3.99	+104.94
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1094.72	+9.66	+36.39
4.	4.	4.	PostgreSQL	Relational, Multi-model	483.91	+1.66	+64.52
5.	5.	5.	MongoDB	Document	412.09	+2.03	+48.90
6.	6.	6.	IBM Db2	Relational, Multi-model	170.77	-0.79	-8.91
7.	7.	↑8.	Elasticsearch	Search engine, Multi-model	150.17	+0.90	+7.85
8.	8.	↓7.	Redis	Key-value, Multi-model	142.91	+1.01	-2.38
9.	9.	9.	Microsoft Access	Relational	131.18	-1.53	-5.62
10.	10.	10.	Cassandra	Wide column	123.22	-0.18	-0.17
11.	11.	11.	SQLite	Relational	122.62	-0.74	+5.88
12.	12.	↑13.	Splunk	Search engine	86.84	-0.17	+9.94
13.	13.	↑14.	MariaDB	Relational, Multi-model	86.77	+0.71	+13.64

source: https://db-engines.com/en/ranking

Content of the Lecture

Content of the Lecture

snippet of real world

relational model

illustrate (model)

Content of the Lecture

query

snippet of
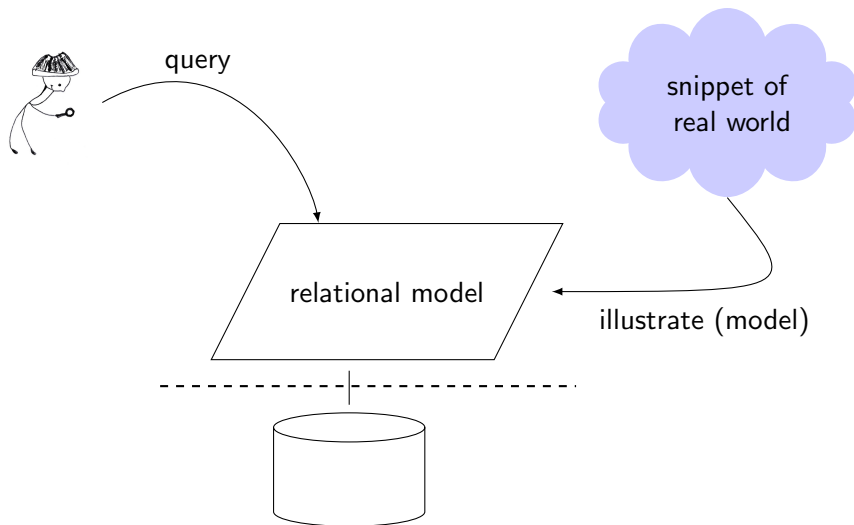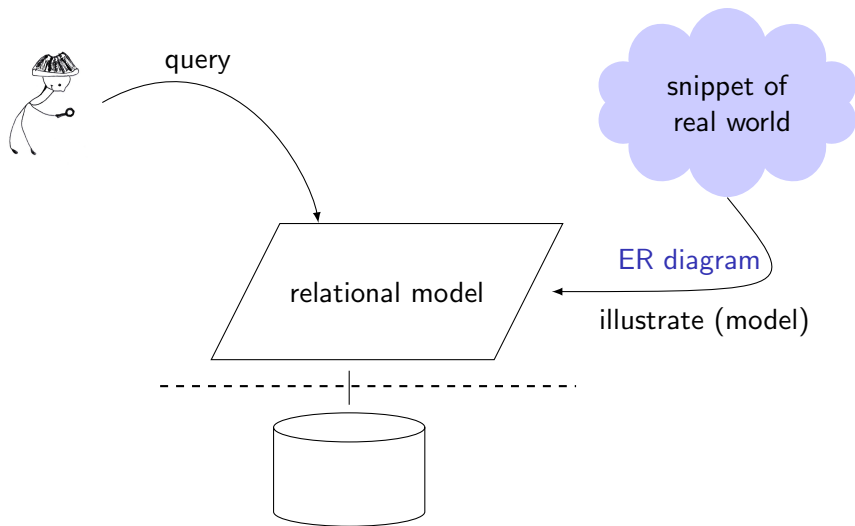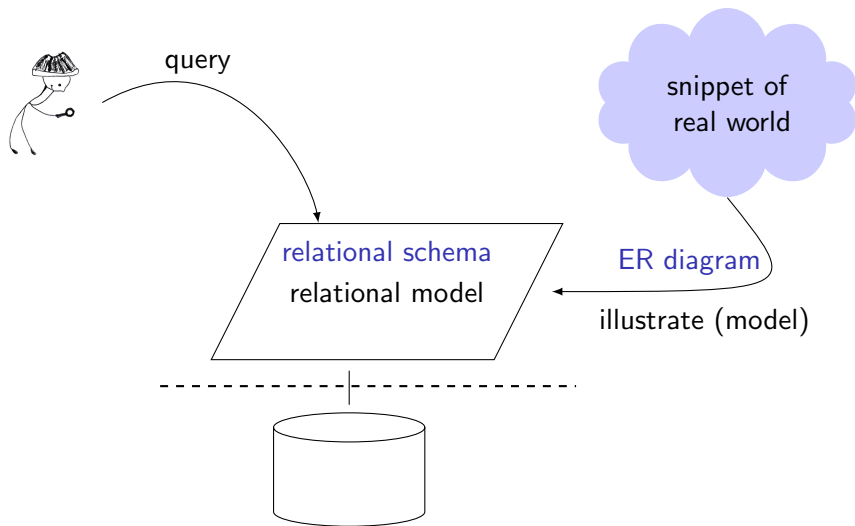real world

relational model

illustrate (model)

Content of the Lecture
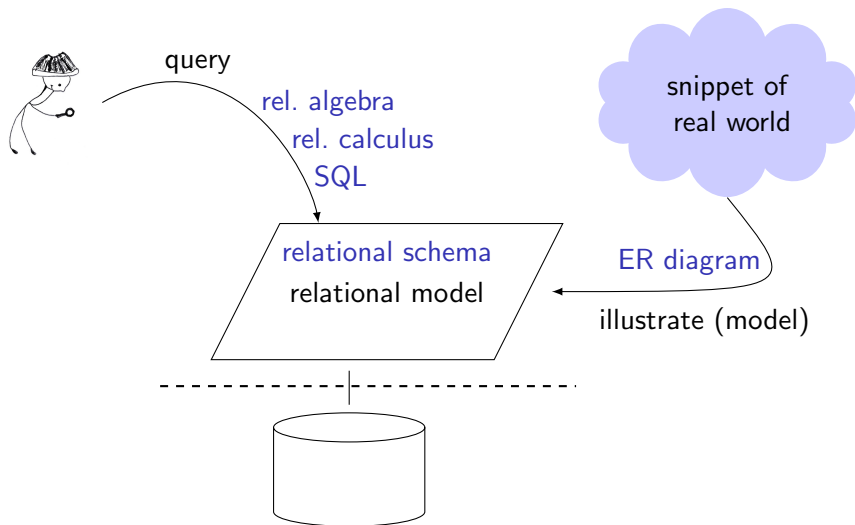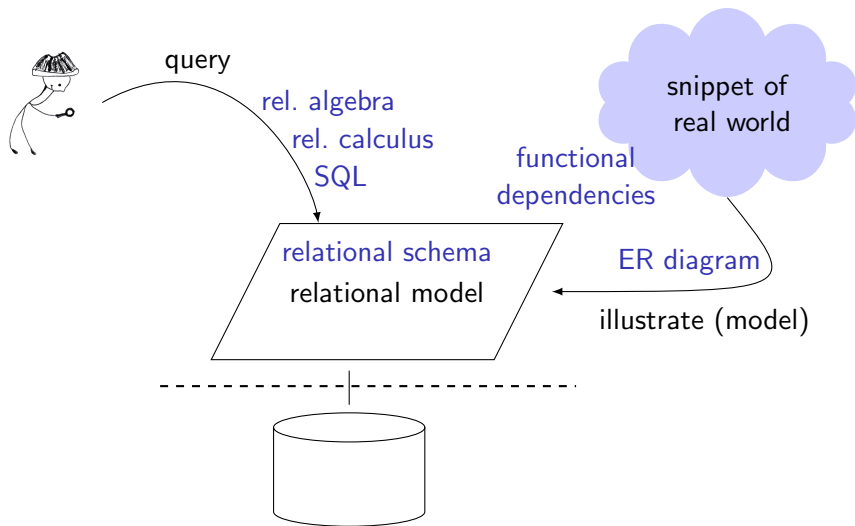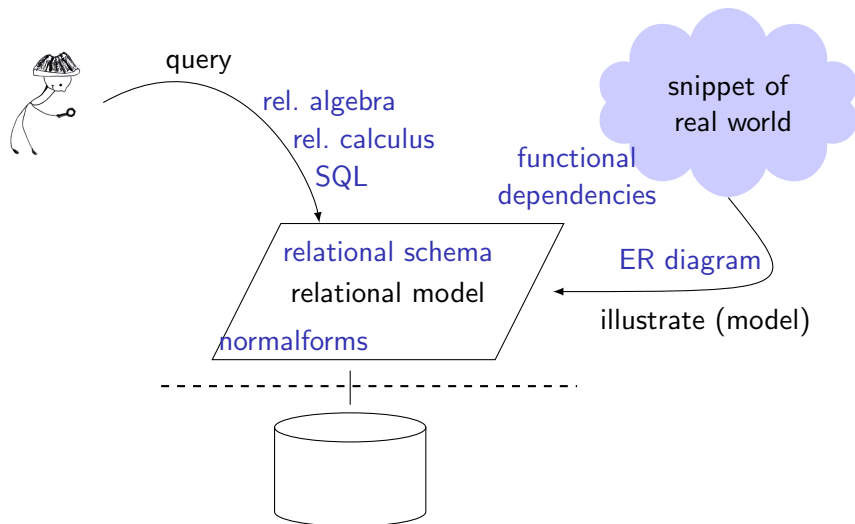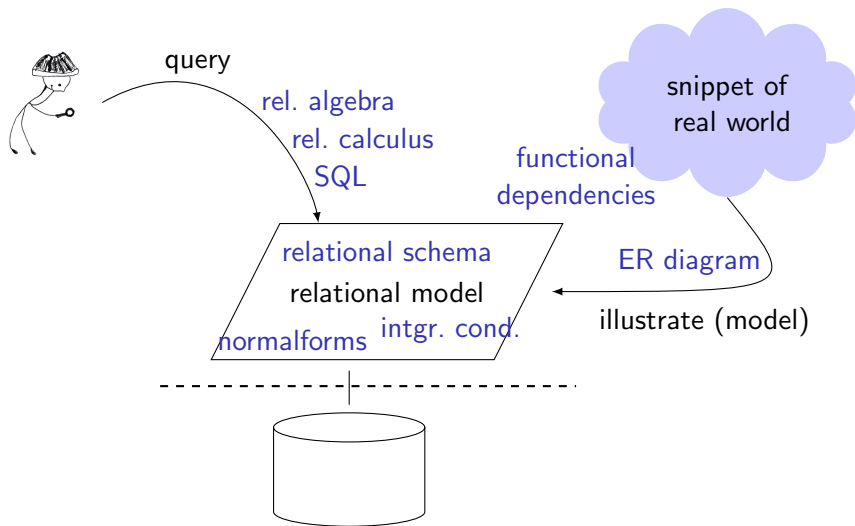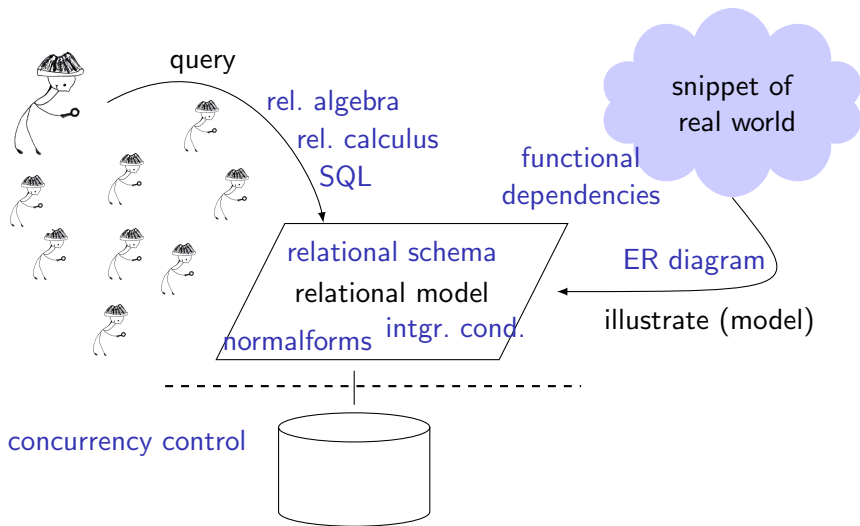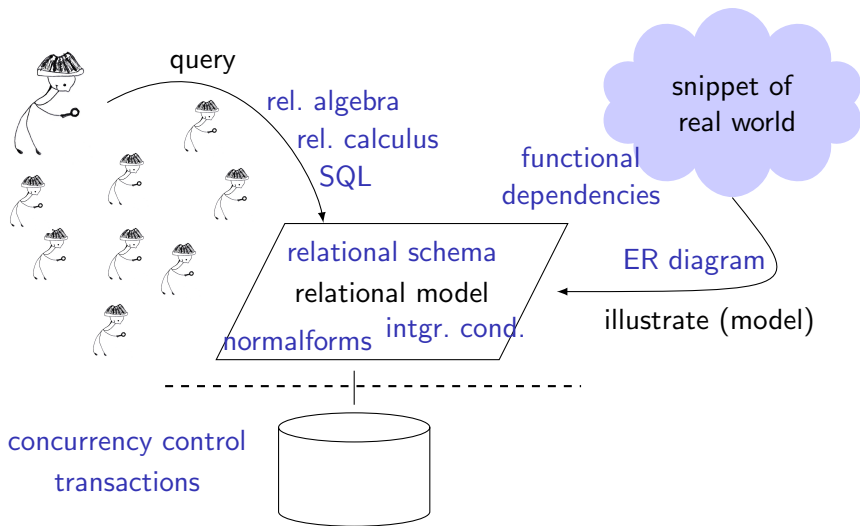
Content of the Lecture

Content of the Lecture

Content of the Lecture

query

rel. algebra
rel. calculus
SQL

snippet of
real world

functional
dependencies

relational schema
relational model

ER diagram

illustrate (model)

Content of the Lecture

query

rel. algebra
rel. calculus
SQL

snippet of
real world

functional
dependencies

relational schema
relational model
normalforms

ER diagram

illustrate (model)

Content of the Lecture

query

rel. algebra
rel. calculus
SQL

snippet of
real world

functional
dependencies

relational schema
relational model

ER diagram

normalforms   intgr. cond.

illustrate (model)

Content of the Lecture

Content of the Lecture

query

rel. algebra
rel. calculus
SQL

snippet of
real world

functional
dependencies

relational schema
relational model
normalforms   intgr. cond.

ER diagram

illustrate (model)

concurrency control
transactions

Content of the Lecture
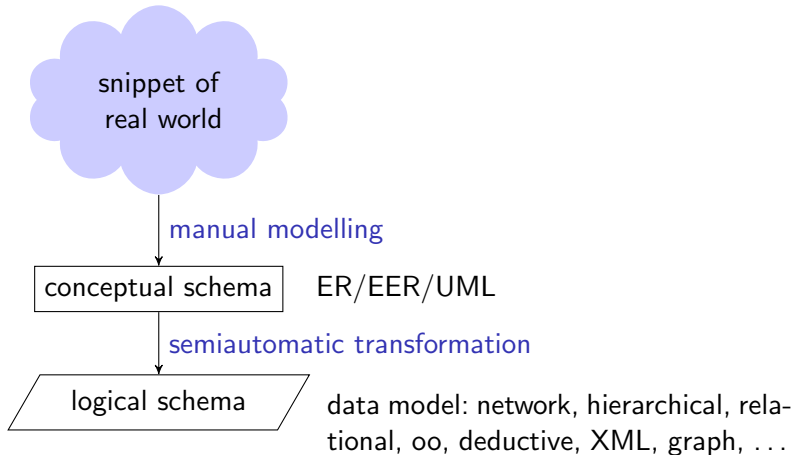
Data Modelling

snippet of
real world

Data Modelling

snippet of
real world

conceptual schema   ER/EER/UML

Data Modelling

snippet of
real world

conceptual schema | ER/EER/UML

logical schema | data model: network, hierarchical, relational, oo, deductive, XML, graph, ...

Data Modelling

snippet of
real world

manual modelling

conceptual schema | ER/EER/UML

logical schema | data model: network, hierarchical, relational, oo, deductive, XML, graph, . . .

Data Modelling

snippet of real world

manual modelling

conceptual schema   ER/EER/UML

semiautomatic transformation

logical schema   data model: network, hierarchical, relational, oo, deductive, XML, graph, . . .
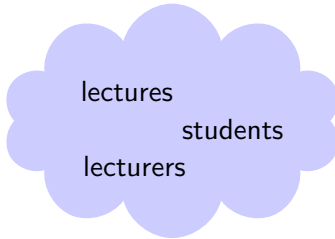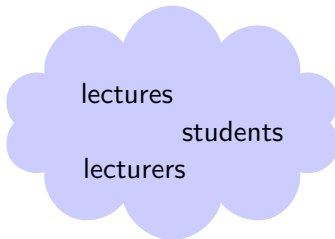
Data Modelling (Ex.)

1. differentiation to the world to be modelled

Data Modelling (Ex.)

1. differentiation to the world to be modelled

real world: university
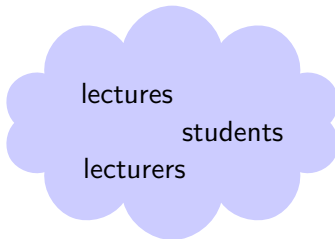
lectures

students

lecturers

Data Modelling (Ex.)

1. differentiation to the world to be modelled

real world: university

lectures

students

lecturers

2. transformation of the world to be modelled in a conceptual schema (in the lecture: EER)

Data Modelling (Ex.)

1 differentiation to the world to be modelled

real world: university

lectures

students

lecturers

2 transformation of the world to be modelled in a conceptual schema (in the lecture: EER)

Data Modelling (Ex.)

3 translation of the conceptual schema in a logical schema (in the lecture: relational schema)

Data Modelling (Ex.)

3 translation of the conceptual schema in a logical schema (in the lecture: relational schema)

student	
matrNr	name
24002	Xenokrates
25403	Jonas
26120	Fichte
26830	Aristoxenos
28106	Carnap
29555	Feuerbach
. . .	. . .

attend	
matrNr	lecNr
26120	5001
24002	5001
24002	4052
. . .	. . .

lecture	
lecNr	title
5001	Grundzüge
5041	Ethik
5049	Mäeutik
4052	Logik
5216	Bioethik
. . .	. . .

Modelling . . .

models:

- simplification of reality (abstraction)
- (better) understanding of complex issues
  (or of parts/aspects thereof)

(formal) modelling languages

- used to describe models
- facilitate communication about models
- have a clearly defined semantics
- make sure that everyone involved assumes the same model