Database Design

DBS can take care automatically of many things – but the user has to specify

- Requirements of the application
- Characteristics of the data

Two important concepts during DBS design:
- Data Model: How to describe the data?
- Data Schema: Concrete description of the data (using the chosen data model)
Data modeling

Excerpt of the Real World

Conceptual Schema
(E/R- or UML-Schema)

Relational Schema

XML Schema

Network Schema

Object-oriented Schema

Manual/intellectual Modeling

Semi-automatic Transformation
Logical Data Models

- Network Model
- Hierarchical Model
- **Relational Data Model**
- XML Model
- Object-oriented Data Model
  - Object-relational Schema
- Deductive Data Model
Modeling a small example application: E/R

Real World: University

Conceptual Modeling

Students

Lectures

StudNr
Name

attend

LectureNr
Title

Students

Lectures

24-Oct-17
Modeling a small example application: UML

Students
+StudNr : int
+Name : String
+Semester : int
+GPA() : float
+SumWeeklyHours() : short

Lectures
+LectureNr : int
+Title : String
+WeeklyHours : int
+NumberAttendees() : int
+FailureRate() : float

+Attendee
1..*
attend

+Successor

requires

24-Oct-17
### Relational Data Model

<table>
<thead>
<tr>
<th>Students</th>
<th>attend</th>
<th>Lectures</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudNr</td>
<td>Name</td>
<td>LectureNr</td>
</tr>
<tr>
<td>26120</td>
<td>Fichte</td>
<td>25403</td>
</tr>
<tr>
<td>25403</td>
<td>Jonas</td>
<td>26120</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Select** Name  
**From** Students, attend, Lectures  
**Where** Students.StudNr = attend.StudNr and attend.LectureNr = Lectures.LectureNr and Lectures.Title = `Grundzüge´;
Database Design

Database Abstraction Layers

1. Conceptual Model

2. Logical Model

3. Physical Database Design
Phases of Database Design

- Information Requirements
- Processing Requirements

- Requirements Engineering
  - Book of Duty
  - Conceptual Modeling
    - ER Schema
    - Logical Modeling
      - Logical Schema
      - Logical Design
  - Logical Design
  - Physical Modeling
    - Physical Design

- DBMS-Characteristics
- Hardware/OS-Characteristics
Software Development and Ability to Communicate

How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it

How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed
Schema Design

Approach in principle:

Information Requirements

Semantical Data Modeling

Logical Data Modeling

Database Installation / Tuning

Coarse Grain Data Modeling

Semantical Analysis

- Interview
- Brainstorming
- Document‘s Analy.
- ...

Fine Grain Data Modeling

- ERM
- UML
- ...

Logical Schema Design

- Hierarchical
- network
- relational
- object-oriented
- ....

Physical Schema Design

- IMS
- UDS
- DB2
- Ozone
- ...

Conceptual Schema Design

Time
Object Description

University Employees
- Quantity: 1000
- Attributes
  - Salary
    - Type: decimal
    - Length: (7,2)
    - Unit: Euro per month
    - Defined: 10%
    - Identifying: no
  - EmpNumber
    - Type: Integer
    - Domain: 0...999.999.99
    - Defined: 100%
    - Identifying: yes
    - Example: 007
  - Level
    - Type: String
    - Length: 2
    - Defined: 100%
    - Identifying: no
    - Example: W2

24-Oct-17
Database System Concepts for Non-Computer Scientists WS 2017/2018
Relation Description: test

Involved Objects:
- Professor as Tester
- Student as Testee
- Lecture as Test Subject

Attributes of the Relation:
- Date
- Time
- Grade

Quantity: 100,000 per year
<table>
<thead>
<tr>
<th>Process Description: Issue a Certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Frequency: semiannually</td>
</tr>
<tr>
<td>- Required Data</td>
</tr>
<tr>
<td>* Tests</td>
</tr>
<tr>
<td>* Examination Rules</td>
</tr>
<tr>
<td>* Student‘s Records</td>
</tr>
<tr>
<td>* ...</td>
</tr>
<tr>
<td>- Priority: high</td>
</tr>
<tr>
<td>- Data Volume to be processed</td>
</tr>
<tr>
<td>* 500 Students</td>
</tr>
<tr>
<td>* 3000 Tests</td>
</tr>
<tr>
<td>* 10 Versions of Examination Rules</td>
</tr>
</tbody>
</table>
Phases of Database Design

Information Requirements

Processing Requirements

Requirements Engineering

Conceptual Modeling

Logical Modeling

Physical Modeling

ER Schema

Conceptual Design

Logical Design

Physical Design

Book of Duty

DBMS-Characteristics

Hardware/OS-Characteristics

24-Oct-17
Conceptual Design

The ideal design (the ideal specification) is
• unique
• complete
• comprehensible (for all participants)
• nonredundant
• . . . and not reachable in reality
Creating a Specification

The actual analysis is an iterative process:
• Customer tells developer his/her needs
• Developer notes everything down (s/he understood) in his/her „language“ . . .
• . . . and translates it into the "language" of the customer
• This is shown to the customer who does not agree with everything
• Change requests are agreed on
• Back to step 2
Entity/Relationship-Modeling

Entity

Relationship

Attribute (property)

Key (identification)

Role

Students

Lectures

Attendee

StudNr

Name

Semester

attend

LectureNr

Title

Weekly hours

StudNr

Name

Semester

LectureNr

Title

Weekly hours

attend

Entity: Students, Lectures
Relationship: Attendee
Attribute (property): attend
Key (identification): StudNr, LectureNr
Role: StudNr, Name, Semester, LectureNr, Title, Weekly hours
Functionalities

1:1

1:N

N:1

N:M

$R \subseteq E_1 \times E_2$
Relationship 1:1

Relationship 1:1

$e_1$ out of $E_1$ takes part in 1 relation of type $R$.

$e_2$ out of $E_2$ takes part in 1 relation of type $R$.

Example:

one car has one license plate
one license plate belongs to one car
Relationship 1:N

$e_1$ out of $E_1$ takes part in $N$ relations of type $R$

$e_2$ out of $E_2$ takes part in 1 relation of type $R$

Example:

one mentor advises several students
one student is advised by one mentor
Relationship N:M

e₁ out of E₁ takes part in M relations of type R
e₂ out of E₂ takes part in N relation of type R

Example:

one actor acts in several movies
one movie has several actors
Unary Relationship 1:N

Relation 1:N

$E_1$ role A $1$ $R$ $N$ role B

e$_1$ out of $E_1$ takes part in role A in $N$ relations of type $R$
e$_1$ out of $E_1$ takes part in role B in 1 relation of type $R$

Example:

Persons $1$ rel-ship $N$
mother child

one person is mother of several persons (children)
one person is child of one person (mother)
Functionalities in $n$-ary Relationships

\[ R : E_1 \times \ldots \times E_{k-1} \times E_{k+1} \times \ldots \times E_n \rightarrow E_k \]
supervise : Professors x Students \(\rightarrow\) Topics

supervise : Topics x Students \(\rightarrow\) Professors
Thereby induced Consistency Constraints

1. Students may work on only one topic with the same professor (to cover a broad spectrum)

2. Students may work on the same topic only once – thus they may not work on the same topic again with another professor

3. Professors can reuse the same topic – i.e. give the same topic to different students

4. One topic can be given by different professors – but to different students
Occurrence of the Relationship \textit{supervise}

Dashed lines represent illegal occurrences
3-ary relationship:

One checkup is performed by one expert with several patients

One Patient gets only one checkup from one expert

One checkup is performed at one patient only by one expert
(min, max)-Notation

For every $e_i \in E_i$ there are

- at least $\min_i$ tuples $(..., e_i, ...) \in R$ and
- at most $\max_i$ tuples $(..., e_i, ...) \in R$
Example (min, max)

One mentor advises up to 20 students.
One student is advised by exactly one mentor.
Excercise for next class

Inform yourself about unary – binary – ternary relationships

Discussion / new examples next class!
Weak Entities

- Relationship between "strong" and "weak" type is 1: N (or 1:1 in rare cases) - why not N:M?
- The existence of a room depends on the existence of the associated building
- RoomNr is unique only within the building
- Key of Rooms is: RoomNr and BldNr
Tests as weak entity type

• Several professors design one test
• Several lectures are inquired in one test
Generalization / Specialization:

S is a specialization of G

Example:

module \( \text{Is-a} \) seminar
Conclusion

University schema with generalization and \((\text{min}, \text{max})\)-notation
Aggregation

Bikes

<table>
<thead>
<tr>
<th>Part-of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frames</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

24-Oct-17

Database System Concepts for Non-Computer Scientists WS 2017/2018
Vehicles

Non-mot. Vehicles

Mot. Vehicles

Bikes

Motorbikes

Automobiles

Part-of

Frames

Wheels

Part-of

Part-of

Pipes

Handlebars

Rims

Spokes

Aggregation and Generalization