Chapter 2: ER-Diagrams

Content:

• Learn how to draw ER diagrams
• Useful to model a database
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)

Manual/intellectual Modeling

Semi-automatic Transformation

Relational Schema

XML Schema

Network Schema

Object-oriented Schema
Modeling a small example application: E/R

Real World: University

Conceptual Modeling

Students
- StudNr
- Name

Lectures
- LectureNr
- Title

attend

requires
Logical Data Models

- Network Model
- Hierarchical Model
- **Relational Data Model**
- XML Model
- Object-oriented Data Model
  - Object-relational Schema
- Deductive Data Model

* [Michael Stonebraker: What Goes Around Comes Around]*
### Relational Data Model

#### Students

<table>
<thead>
<tr>
<th>StudNr</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>26120</td>
<td>Fichte</td>
</tr>
<tr>
<td>25403</td>
<td>Jonas</td>
</tr>
</tbody>
</table>

#### attend

<table>
<thead>
<tr>
<th>StudNr</th>
<th>Lecture Nr</th>
</tr>
</thead>
<tbody>
<tr>
<td>25403</td>
<td>5022</td>
</tr>
<tr>
<td>26120</td>
<td>5001</td>
</tr>
</tbody>
</table>

#### Lectures

<table>
<thead>
<tr>
<th>Lecture Nr</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>5001</td>
<td>Grundzüge Glaube und Wissen</td>
</tr>
<tr>
<td>5022</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';
Modeling a small example application: UML

Students

+StudNr : int
+Name : String
+Semester : int

+GPA() : float
+SumWeeklyHours() : short

1..*

Attendee

Lectures

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

attend

requires

*
Database Design

Database Abstraction Layers

1. Conceptual Design
2. Logical Design
3. Physical Database Design
Phases of Database Design

- Information Requirements
- Processing Requirements
- Requirements Engineering
  - Scope Statement
  - Conceptual Modeling
  - ER Schema
  - Conceptual Design
  - Logical Modeling
  - Logical Design
  - Logical Schema
  - Physical Modeling
  - Physical Database Design

Hardware/OS-Characteristics
DBMS-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:**

<table>
<thead>
<tr>
<th>Information Requirements</th>
<th>Semantical Data Modeling</th>
<th>Logical Data Modeling</th>
<th>Database Installation / Tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse Grain Data Modeling</td>
<td>Semantical Analysis</td>
<td>Logical Data Modeling</td>
<td>Database Installation / Tuning</td>
</tr>
</tbody>
</table>

### Approach in principle:

- **Information Requirements**
- **Semantical Data Modeling**
- **Logical Data Modeling**
- **Database Installation / Tuning**

#### Coarse Grain Data Modeling
- 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
- ...

#### Time
- Interview
- Brainstorming
- Document’s Analy.
- ...

#### Conceptual Schema Design
- ERM
- UML
- ...

Database System Concepts for Non-Computer Scientists WS 2018/2019
Requirements Engineering

Entity description
Relation description
Process description

...
Entity Description

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

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

Attributes of the Relation:
- Date
- Time
- Grade

Quantity: 100,000 per year
Process Description: *Issue a Certificate*

- Frequency: semiannually

- Required Data
  * Tests
  * Examination Rules
  * Student’s Records
  * ...

- Priority: high

- Data Volume to be processed
  * 500 Students
  * 3000 Tests
  * 10 Versions of Examination Rules
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
Phases of Database Design

1. Information Requirements
2. Processing Requirements
3. DBMS-Characteristics
4. Hardware/OS-Characteristics

- Requirements Engineering
  - Scope Statement
    - Conceptual Modeling
      - ER Schema
        - Logical Modeling
          - Logical Schema
            - Logical Design
              - Physical Modeling
                - Physical Database Design

<table>
<thead>
<tr>
<th>Relation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Att1</td>
<td></td>
</tr>
<tr>
<td>Att2</td>
<td></td>
</tr>
</tbody>
</table>
Conceptual Design

The ideal design (the ideal specification) is
• unique
• complete
• comprehensible (for all participants)
• nonredundant
• . . . and not reachable in reality
Entity/Relationship-Modeling

Entity

Relationship

Attribute (property)

Key (identification)

Role

```
Entity:
- Students
  - StudNr
  - Name
  - Semester
  - Attendee

Relationship:
- Attend

Attribute (property):
- LectureNr
- Title
- Weekly hours

Key (identification): 
- StudNr

Role:
```
Functionalities

$E_1 \ R \ E_2$

$R \subseteq E_1 \times E_2$

1:1

1:N

N:1

N:M
Relationship 1:1

**Example:**

- One car has one license plate.
- One license plate belongs to one car.

\[ e_1 \text{ out of } E_1 \text{ takes part in at most 1 relation of type } R \]
\[ e_2 \text{ out of } E_2 \text{ takes part in at most 1 relation of type } R \]
Relationship 1:N

**Example:**

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

\[ E_1 \rightarrow N \rightarrow R \rightarrow M \rightarrow E_2 \]

\( e_1 \) out of \( E_1 \) takes part in \( M \) relations of type \( R \)
\( e_2 \) out of \( E_2 \) takes part in \( N \) relation of type \( R \)

Example:

\[ \text{Actors} \rightarrow N \rightarrow \text{stars} \rightarrow M \rightarrow \text{Movies} \]

one actor stars in several movies
one movie has several actors
Recursive Relationship 1:N

Relationship 1:N

\[ e_1 \text{ out of } E_1 \text{ takes part in role A in N relations of type } R \]
\[ e_1 \text{ out of } E_1 \text{ takes part in role B in at most 1 relation of type } R \]

Example:

```
Persons
<table>
<thead>
<tr>
<th>mother</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>child</td>
<td>N</td>
</tr>
</tbody>
</table>
```

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 : Topics × Students → Professors
supervise : Professors × Students → Topics
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 *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
University Schema

Students

- StudNr
- Name
- Semester

Lectures

- LectureNr
- Title
- Weekly hours

Grades

- test

Assistants

- PersNr
- Name
- Area

Professors

- PersNr
- Name
- Level
- Room

Relationships:

- Students attend Lectures
- Lectures require Students
- Students prerequisite Lectures
- Lectures follow-up Professors
- Assistants work-for Professors
- Professors give Lectures
- Student work-for Assistants
- Assistant work-for Student
- Assistants follow-up Professors
- Professors require Lectures
(min, max)-Notation

For every $e_i \in E_i$ there are

- at least $\min_i$ tuples $(..., e_i, ...) \in R$
- 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

Generalization / Specialization:

G \quad \text{Is-a} \quad S

S is a specialization of G

Example:

animal \quad \text{Is-a} \quad cat
Conclusion

University schema with generalization and (min, max)-notation
Aggregation

- Bikes
  - Part-of Frames
    - Part-of Pipes
    - Part-of Handlebars
  - Part-of Wheels
    - Part-of Rims
    - Part-of Spokes
Vehicles

- Non-mot. Vehicles
  - Bikes
    - Frames
      - Pipes
    - Handlebars
  - Mot. Vehicles
    - Motorbikes
      - Wheels
        - Rims
    - Automobiles
      - Spokes

Aggregation and Generalization
Where are we?

- Information Requirements
- Requirements Engineering
  - Scope Statement
  - Conceptual Modeling
    - ER Schema
    - Conceptual Design
    - Logical Modeling
      - Logical Schema
      - Logical Design
      - Physical Modeling
        - Physical Database Design

- Processing Requirements
- DBMS-Characteristics
- Hardware/OS-Characteristics

ER Schema

<table>
<thead>
<tr>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Att1</td>
</tr>
<tr>
<td>Att2</td>
</tr>
</tbody>
</table>
Min, max Notation and Functionalities

**Polygon**
- $N (3, *)$
- borders
- $M (1, 2)$
- Edge
- $N (2, 2)$
- begin/end
- $M (2, *)$
- Point
- $x$
- $y$
- $z$

Min-max:
A Polygon has at least 3 Edges.
An Edge has 1 or 2 Polygons.
Design criteria

• Rules for classification of entities and attributes:
  • Entities should contain descriptive information
  • Multi valued attributes should be classified as entities
  • Attribute should be assigned to that Entity which describes it most directly
  • Redundant relationships should be avoided

  – However, it always depends on the application
Example: Order

As entity:

As relationship:

As attribute: