MTAT.03.105
Introduction to Databases

Lecture #4

Entity-Relationship Modelling

Ljubov Jaanuska
(ljubov.jaanuska@ut.ee)
Lecture 3. Summary

So far we have been dealing with a relational database \((PCA.db)\)

- **Relation** is physically represented as a table with columns and rows
  - The relation must have a unique name within the database
- Rows correspond to tuples
- Columns correspond to attributes

<table>
<thead>
<tr>
<th>Formal Terms</th>
<th>Alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation</td>
<td>Table</td>
</tr>
<tr>
<td>Tuple</td>
<td>Row</td>
</tr>
<tr>
<td>Attribute</td>
<td>Column</td>
</tr>
</tbody>
</table>
Lecture 3. Summary

• Columns correspond to attributes
  • An attribute is a named column of a relation
  • The values in an attribute (named column) must be of the same data type

• Data types indicate the kind of data for the column (character, numeric, etc.) and permissible operations (numeric operations, string operations) for the column

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric</td>
<td>Numeric data are data on which you can perform arithmetic operations of addition, subtraction, multiplication and division</td>
</tr>
<tr>
<td>Character</td>
<td>For fixed-length text which can contain any character (space included) or symbol not intended for mathematical operation.</td>
</tr>
<tr>
<td>Variable Character</td>
<td>For variable-length text which can contain any character (space included) or symbol not intended for mathematical operation.</td>
</tr>
<tr>
<td>Date</td>
<td>Date is used to store calendar dates using the YEAR, MONTH and DAY fields. For dates the allowable operations includes comparing two dates and generate a date by adding or subtracting a number of days from a given date.</td>
</tr>
<tr>
<td>Logical</td>
<td>For attributes containing data with two values such as True/False or Yes/No</td>
</tr>
</tbody>
</table>

• The domain is the set of allowable values for one or more attributes
  • E.g. CHECK(sex in ('m','f')); NOT NULL; CHAR(5); TINYINT
  • Every attribute is defined on a domain
  • The domain concept is important because it allows the user to define the meaning and source of values that attributes can hold

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>25</td>
<td>3292</td>
</tr>
<tr>
<td>Tom</td>
<td>30</td>
<td>30292</td>
</tr>
<tr>
<td>Ann</td>
<td>28</td>
<td>3392</td>
</tr>
</tbody>
</table>
Lecture 3. Summary

• Rows correspond to tuples
  • A tuple is a row of a relation
• The intersection of column/row represents a single atomic value
Lecture 3. Updates

• Check that all tables in *PCA.db* have primary key
• Rename column *name* to *t_name* in table *Tournament*
• Check data
Lecture 2. What will you learn

• Acquire the basic concepts of the Entity-Relationship (ER) model needed at the first stage of DB design

• Discuss how to represent 1-1, 1-M and M-N relationships

• Attain the skills and knowledge to represent generalisation hierarchies

• Discuss how to represent business rules in an ER Diagram.
Entity-Relationship (E-R) model

The **E-R model** is a means of communication that is non-technical and easily understood. 

- The E-R model allows the nature of the data and how it is used in the business to be specified. 
  - DB designers 
  - programmers 
  - end-users
Entity-Relationship (E-R) model

To design a database and meet user requirements,

- there must be a clear understanding on how the business operates

Begin by identifying the significant data known as entities. Then add information about the entities (attributes). Identify constraints on the entities, relationship and attributes.
Entity

- Entity is a basic building block of the E-R data model
- The entity used in three different meanings or term
  - **Entity**
    - Each tuple is an entity

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>73</td>
<td>Pjotr</td>
<td>Pustota</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

- **Entity set**
  - A set of entities

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>71</td>
<td>Arvo</td>
<td>Mets</td>
<td>(NULL)</td>
</tr>
<tr>
<td>72</td>
<td>Maari</td>
<td>Mustikas</td>
<td>(NULL)</td>
</tr>
<tr>
<td>73</td>
<td>Pjotr</td>
<td>Pustota</td>
<td>(NULL)</td>
</tr>
<tr>
<td>74</td>
<td>Kalle</td>
<td>Kivine</td>
<td>(NULL)</td>
</tr>
</tbody>
</table>

- **Entity type**
  - A table in a DB
  - Has a name and attributes
Entity

- Attribute
  - A descriptive property or characteristics of an entity

<table>
<thead>
<tr>
<th>Id</th>
<th>p_name</th>
<th>surname</th>
<th>player_id</th>
<th>club_id</th>
</tr>
</thead>
</table>


Relations

- A **relationship type** is a set of business association that exists between one or more entities.
- Each relationship type is given a **name** that describes its function.
- Relation names appear on the line connecting the entity types involved in the relation.

A binary relationship (involves two entity types)
Relationship cardinality

Cardinalities contain the number of objects that participate in a relationship

Each player is related to exactly one club;
In other direction, each club has zero or more players.
3 main type of relationship between entities

- one-to-one relationship
- one-to-many relationship
- many-many relationship
Classification of cardinalities

Cardinalities are classified by common values for minimum and maximum cardinality

<table>
<thead>
<tr>
<th>Cardinality Interpretation</th>
<th>Minimum Instances</th>
<th>Maximum Instances</th>
<th>Graphic Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exactly one (one and one only)</td>
<td>1</td>
<td>1</td>
<td><img src="image1.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Zero or one</td>
<td>0</td>
<td>1</td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>One or more</td>
<td>1</td>
<td>Many (&gt;1)</td>
<td><img src="image3.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Zero, one, or more</td>
<td>0</td>
<td>Many (&gt;1)</td>
<td><img src="image4.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

- A minimum cardinality of one or more shows a mandatory relationship
- A minimum cardinality of zero indicates an optional relationship
Classification of cardinalities

- A club has zero, one or more players;
- Each player plays for one club;
- The *Has* relationship is optional to the *Club* entity type because a *Club* entity can be stored without being related to a *Player* entity type.
Types of entity types

A **Strong Entity Type** are known as parent or dominant entities
- Strong entity types have primary keys

A **Weak Entity Type** is an entity type that does not have sufficient attributes to form a primary key
- A weak entity type is identified by a double rectangle
- Weak entities borrow all or part of the primary keys from another strong entity - **discriminator** (one or more attributes) for distinguishing among its entities
- The discriminator is underlined by a dashed line
Types of Entity Types

- Strong Entity Type
- Weak Entity Type

Diagram:

- Employee
- Dependants
- Book
- Book copy
Generalisation

• **Generalisation hierarchy** is a technique where the attributes that are common to several entity types are grouped into their own entity called a **supertype**.

• **Supertype** is an entity type that stores attributes that are common to one or more entity **subtypes**.

• **Subtype** is an entity type that inherits some common attributes from an entity supertype and then adds other attributes that are unique to an instance of the subtype.

• The sharing of attributes between supertypes and subtypes is known as **inheritance**.
Generalisation
Disjointness (D) means that subtypes in a generalisation hierarchy do not have any entities in common.

The generalisation hierarchy is disjoint (non-overlapping) because an Employee cannot be a Pilot and at the same time a Mechanic.
Disjointness

The generalisation hierarchy is **not disjoint** (overlapping) because a member of a Faculty can be an Academic Staff as well as a Student. The absence of the D indicates that the generalisation hierarchy is not disjoint.
Completeness

Completeness (C) means that every entity of a supertype must be an entity in one of the subtypes in the generalisation hierarchy.

The completeness constraint means every Staff must either be employed FullTime or as PartTime Staff.
Benefits of E-R model

- Documents information for the organization in a clear, precise format
- Provide a clear picture of the scope of the information requirement
- Provide easy to understand pictorial map for the database design
- ER models are readily translated to relations
Example

Consider the COMPANY database which keeps track of a company’s employees, departments and projects:

• The company is organised into departments. Each department has a unique name, a unique number and a particular employee that manages the department.

• A department controls a number of projects, each of which has a unique name and unique number.

• We store each employee’s name, a national card ID, address, salary, and birth date. An employee is assigned to one department but may work on several projects, which are not controlled by the same department. We keep track of the number of hours per week that the employee works on each project. A project may involve more than one employee.

• We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent’s first name, birth date and relationship to the employee.

• Answer the following questions about the COMPANY database (state clearly any assumptions that you may make)
  • List all entities with their attributes. Underline the primary key. Identify all weak entities.
  • Draw the entity relationship diagram for the COMPANY database.
Example