Database Management Systems

Database Design (2)

Topics

- Data Base Design
  - Logical Design (Review)
  - Physical Design
- Entity Relationship (ER) Model to Relational Model
  - Entity
  - Relationship
  - Attributes
- Normalization
  - First normal form
  - Second normal form
  - Boyce-Codd normal form

Database Design

- Database design is the process of producing a detailed data model of a database.
- Database design includes:
  - Determining data to be stored
  - Determining main entities and their attributes
  - Determining relationships between entities
  - Designing a suitable model to store them

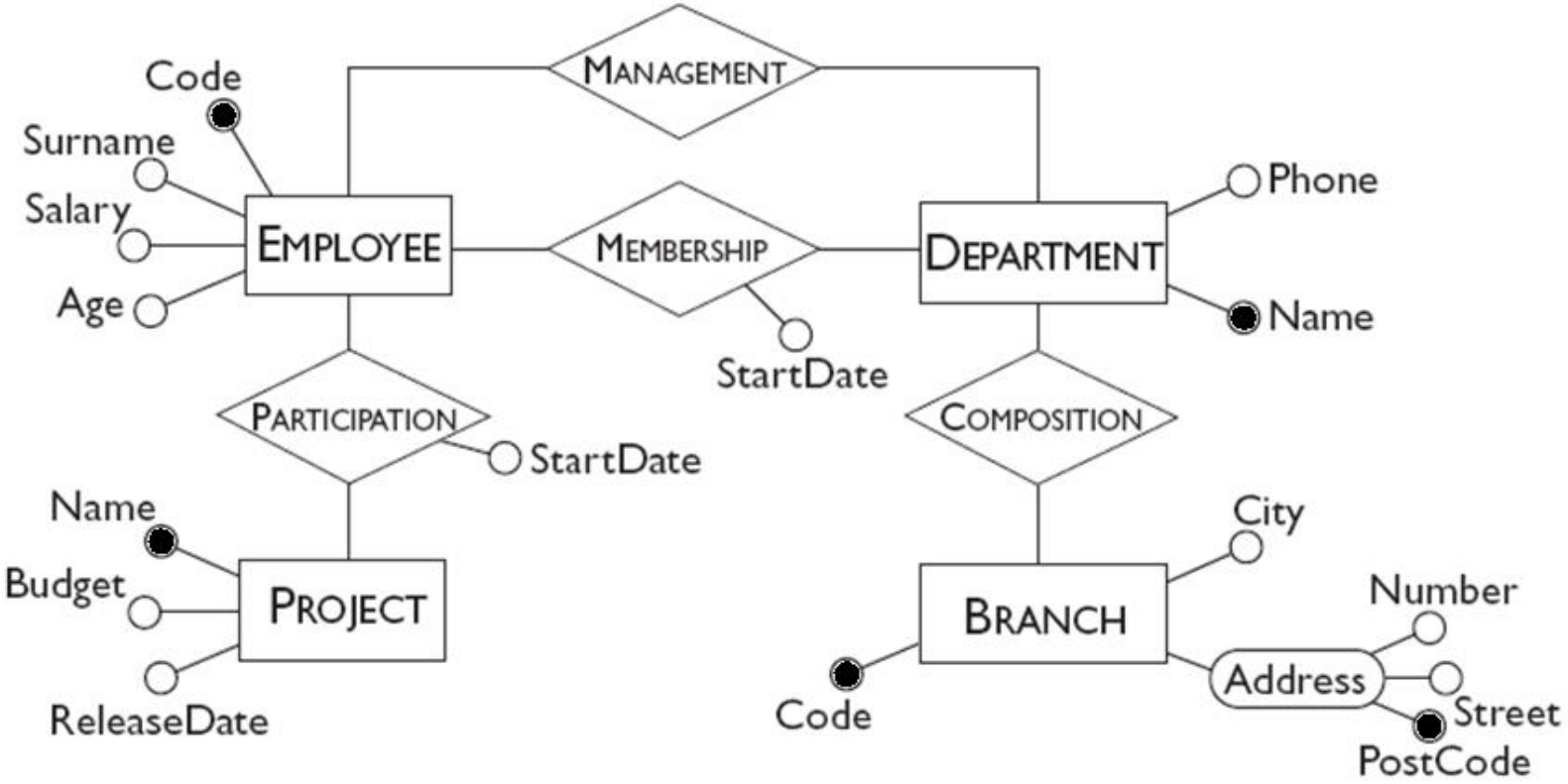
Logical and Physical Design

- Logical design is about gathering requirements and converting those requirements into a model.
- Physical design is the process of converting the logical model into database tables.

Converting E-R Model into Relational Model

- Each entity in an E-R model is converted to a table in relational model.
- The attributes of the entity become fields of the table
- Primary key is given by the identifiers of the entity

Sample E-R Model



Example

```
CREATE Table Employee
(
    Code Integer PRIMARY KEY,
    Surname Char(30),
    Salary Integer,
    Age Integer
)
```

One-to-One Relationships

- One-to-one relationships are defined as attributes.
  - e.g. Book ➔ ISBN : Each book has only one ISBN and each ISBN represents only one book. ISBN should be an attribute for book.

- For restricting access to sensitive data, some attributes can be stored in a second table.
  - User and Password have a one-to-one relationship. Password should be an attribute for the user but we generally create a new table as <UsrID,Password>

One-to-Many Relationships (1)

- One-to-many relationships are defined as foreign keys.
  - e.g. An employee is a member of one department

    A department has many members

    In employee table define attribute ***dept*** to show the department of each employee. ***dept*** is a foreign key referring to Department.

One-to-Many Relationships (2)

- If relation has attributes then we have to define it as a table.
  - e.g. The relationship ***Membership*** between ***Employee*** and ***Department*** has "*start date*" attribute.
  - The relationship which is defined as a table can store the history of a relationship.
  - The primary keys of both entities and the attributes of the relationship are added to the table

Example

```
Create Table Membership
(   EmpCode Number references Employee(code),
    DeptName    char(20) references
                                        department(name),
    startDate   date,
     position    char(30),
     PRIMAR KEY ( EmpCode, DeptName, startDate)
)
```

Many-to-Many Relationships

- All many to many relationships are defined as tables.
  - e.g. Student ➔ Course
  - A student takes many courses and a course is taken by many students
  - The primary key of both entities are added to the table
  - <StudentID, CourseCode, Year-Semester, Grade>

Recursive Relationships

- Recursive relationships are defined as a foreign key attribute.
  - e.g.

  Human < ID, Name, Surname, Birth Date, Sex, Father >

  Father is a foreign key to table Human (recursive)

Ternary Relationships

- Ternary relationships are defined as a table.
- The primary key of three tables are included in the table
- e.g. A client borrows a book from a library

  entities are: Client, Book, Library

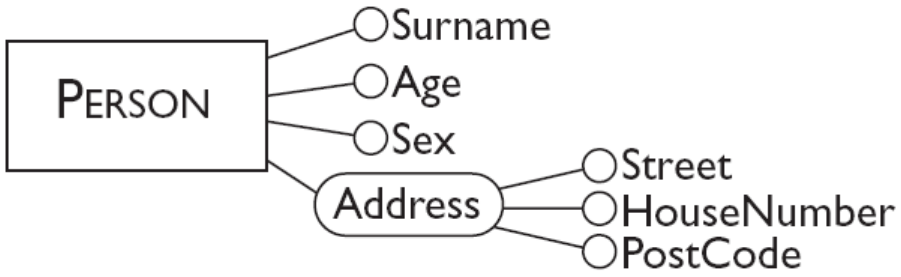
  Borrow < BookID, ClientID, LibID, Date,..>

Composite Attributes

- Composite attributes are defined as a table
- The primary key of the composite attribute table is added to the table using it as foreign key.
  - e.g. Address<Street, HouseNumber, PostCode>

    PostCode is primary key
  - Person <Surname, Age, Sex, PostCode >

    PostCode is foreign key



Normalization

- Normalization is a method for designing databases.
- Normalization starts with a pile of data items then puts them in a single table. Next it breaks the table into more tables.
- Normalization is performed in several stages (normal forms)

First Normal Form

- After gathering data, they are stored in a table. The table is said to be in first normal form if it is flat.

- e.g. A customer in a banking system performs many transactions in a given date.

Converting to First Normal Form

Customer	Transactions		
	Tr. ID	*Date*	*Amt.*
Jones	12890	14-Oct-2003	-87
	12904	15-Oct-2003	-50
Wilkins	12898	14-Oct-2003	-21
Stevens	12907	15-Oct-2003	-18
	14920	20-Nov-2003	-70
	15003	27-Nov-2003	-60

Customer	Tr. ID	Date	Amount
Jones	12890	14-Oct-2003	-87
Jones	12904	15-Oct-2003	-50
Wilkins	12898	14-Oct-2003	-21
Stevens	12907	15-Oct-2003	-18
Stevens	14920	20-Nov-2003	-70
Stevens	15003	27-Nov-2003	-60

Second Normal Form

- A table is in second normal form if all functional dependencies are candidate keys.

- A functional dependency is a relationship between two attributes where the first attribute is determined by the second attribute.
  - e.g. Person < ID, Name, Age, Address, Phone>
  - Address has a functional dependency on ID

Example

- Functional dependencies:
  - Salary depends on employee
  - Budget depends on project
  - Neither employee nor project can be primary key

Employee	Salary	Project	Budget
Brown	20	Mars	2
Green	35	Jupiter	15
Green	35	Venus	15
Hoskins	55	Venus	15
Hoskins	55	Jupiter	15
Hoskins	55	Mars	2
Moore	48	Mars	2
Moore	48	Venus	15
Kemp	48	Venus	15
Kemp	48	Jupiter	15

Converting into Second Normal Form

- Divide the table into two tables. The salary which depends on employee only is in the first table, and project and budget in the second table.

Employee	Salary
Brown	20
Green	35
Hoskins	55
Moore	48
Kemp	48

Project	Budget
Mars	2
Jupiter	15
Venus	15

Boyce-Codd Normal Form

- Some attributes may depend on more than one attribute. In this case the table is divided so that each attribute depends on the primary key of its table only.

- The normal form given above is called Boyce-Codd normal form.

Example

Employee	Salary	Project	Budget	Function
Brown	20	Mars	2	technician
Green	35	Jupiter	15	designer
Green	35	Venus	15	designer
Hoskins	55	Venus	15	manager
Hoskins	55	Jupiter	15	consultant
Hoskins	55	Mars	2	consultant
Moore	48	Mars	2	manager
Moore	48	Venus	15	designer
Kemp	48	Venus	15	designer
Kemp	48	Jupiter	15	manager

Boyce-Codd Normal Form

Employee	Salary
Brown	20
Green	35
Hoskins	55
Moore	48
Kemp	48

Project	Budget
Mars	2
Jupiter	15
Venus	15

Employee	Project	Function
Brown	Mars	technician
Green	Jupiter	designer
Green	Venus	designer
Hoskins	Venus	manager
Hoskins	Jupiter	consultant
Hoskins	Mars	consultant
Moore	Mars	manager
Moore	Venus	designer
Kemp	Venus	designer
Kemp	Jupiter	manager

Summary

- Database design is the process of modeling data in an information system.
- E-R model is converted into relational tables in physical design.
- Entities and many-to-many relationships, and one-to-many relationships with attribute in E-R model are converted to tables.
- Normalization is used a database design method in logical and physical design.

Questions?

Term Projects

- Follow the following steps in your term project:
- Choose your teammate (groups of two at most)
- Choose a title, e.g. Library Database
- Send your team members, project title by email to me.

Term Projects

- Create a list of data items that will be stored in your project. For example, in library project, you may ask the library. Items need not be put in any order.
- Find and list all entities.
- Find the attributes of entities (the attributes are data items that you already found)
- Define restrictions for each attribute (e.g. Number of digits in student ID, range of values for Age attribute, …)

Term Projects

- Define relationships between entities and their types (Explain how you found the type of the relationships)
- Create Entity-Relationship (ER) model of your database.
- Create necessary tables for your database.
- Show in which normal forms your tables are.

Term Projects

- Design necessary queries based on the requirements of the project
- Write necessary SQL commands to create tables and queries
- Write a report for your project (put all steps explained above)
- Implement your project using a DBMS (Oracle, MS-SQL, MySQL, MS-Access, …)

Term Projects

Evaluation

- Gathering data items (10)
- Defining entities, the attributes of the entities, the restrictions on the attributes (15)
- Relationships (5)
- ER model (5)
- Normalization (10)
- Queries (15)
- Implementation (15)
- Report (10)
- Presentation (15) (is necessary)

Presentations

- Will be during the last week of the semester starting from May 10th
- The presentation schedule is posted on the webpage of the course.