Lecture 2: Entity-Relation (ER) Diagrams
11.10.2016
Overview of Database Design

- **Conceptual design**: *(ER Model is used at this stage.)*
  - What are the *entities* and *relationships* in the enterprise?
  - What information about these entities and relationships should we store in the database?
  - First step, from an informal description to a more precise description
  - What are the *integrity constraints* or *business rules* that hold?
  - A database `schema` in the ER Model can be represented pictorially *(ER diagrams).*
  - Can map an ER diagram into a relational schema.
**Entity:** Real-world object distinguishable from other objects. An entity is described (in DB) using a set of attributes.
   - e.g. Ali, Ayşe, CS, 371, 201 etc.

**Attribute:** Entities have attributes
   - e.g. Ayşe has an address, Ali has a phone number etc.

**Entity Set:** A collection of similar entities.
   - e.g., all employees, the set of students, the set of courses etc.
   - All entities in an entity set have the same set of attributes. (Until we consider ISA hierarchies, anyway!)
   - Each entity set has a *key*.
   - Each attribute has a *domain*.
Underline the key attributes.

**SIMPLE KEY ATTRIBUTE**

- Rolino
- Name

**STUDENT**

**COMPOSITE KEY ATTRIBUTE**

- Date
- Flightno
- Flightid
- No-of-passengers

**FLIGHT**
A **candidate key** is a minimal set of attributes that uniquely identifies each instance of an entity type.
- For example, the number attribute uniquely identifies an Employee and is a candidate key for the Employee entity type.

A **primary key** is a candidate key that is selected to identify each instance of an entity type.
- The primary key is chosen from a set of candidate keys. For instance, an employee may also have SSN as an attribute. The primary key may be either SSN or number as both are candidate keys.

A **composite key** is a key that consists of two or more attributes.
- For example, a course is uniquely identified only by the department code (22C) and the course number within the department (144).
ER Model Basics (cont.)

<table>
<thead>
<tr>
<th>SID</th>
<th>SName</th>
<th>SAge</th>
<th>SClass</th>
<th>SSection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1101</td>
<td>Alex</td>
<td>14</td>
<td>9</td>
<td>A</td>
</tr>
<tr>
<td>1102</td>
<td>Maria</td>
<td>15</td>
<td>9</td>
<td>A</td>
</tr>
<tr>
<td>1103</td>
<td>Maya</td>
<td>14</td>
<td>10</td>
<td>B</td>
</tr>
<tr>
<td>1104</td>
<td>Bob</td>
<td>14</td>
<td>9</td>
<td>A</td>
</tr>
<tr>
<td>1105</td>
<td>Newton</td>
<td>15</td>
<td>10</td>
<td>B</td>
</tr>
</tbody>
</table>
An entity is represented by a set of attributes
**Relationship:** Association among two or more entities. E.g., Ayşe works in Pharmacy department, Ali takes 371, etc.

**Relationship Set:** Collection of similar relationships.
- An n-ary relationship set $R$ relates $n$ entity sets $E_1 \ldots E_n$
  - Same entity set could participate in different relationship sets, or in different “roles” in same set as in Reports_To relationship
Consider Works_In: An employee can work in many departments; a dept can have many employees.

In contrast, each dept has at most one manager, according to the key constraint on Manages.
Match the constraints

1) 1-to-1
2) 1-to Many
3) Many-to-1
4) Many-to-1

a) E - R - F
b) E - R - F
c) E - R - F
d) E - R - F
Cardinality

1. Each book is written by an author

2. Each book has a publisher

3. Some shopping baskets may contain more than one copy of the same book

4. The warehouse stocks several books
Making ER Models

To make an ER model you need to identify:
- Entities
- Attributes
- Relationships
- Cardinality ratios

- Entities are things or objects. They are often nouns.
- Attributes are facts or properties. They are also often nouns.
- Verbs often describe relationships between entities.
A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enrol in a particular course and take modules towards the completion of that course. Each module is taught by a lecturer from the appropriate department, and each lecturer tutors a group of students.
A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enrol in a particular course and take modules towards the completion of that course. Each module is taught by a lecturer from the appropriate department, and each lecturer tutors a group of students.
A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enrol in a particular course and take modules towards the completion of that course. Each module is taught by a lecturer from the appropriate department, and each lecturer tutors a group of students.
Example – ER Diagram

Entities: Department, Course, Module, Lecturer, Student
Example – ER Diagram

Each department offers several courses

- Course
- Module
- Lecturer
- Student
A number of modules *make up* each course.

- **Course**
  - **Includes** Module
  - **Offers** Department

- **Module**
- **Department**
- **Student**
- **Lecturer**
Example - ER Diagram

Students enrol in a particular course

- Course
  - Includes Module
  - Enrols In Student
  - Offers Department

- Lecturer
Example - ER Diagram

Students … take modules

- Offers
- Includes
- Enrolls In
- Takes
- Takes
- Student
- Module
- Course
- Department
- Lecturer

Entity Relationship Modelling
Each module is taught by a lecturer
a lecturer from the appropriate department

Diagram:
- Department
  - Offers
  - Employs
- Course
  - Includes
- Module
  - Teaches
- Student
  - Takes
- Lecturer
  - Enrolls In
each lecturer tutors a group of students
Example - ER Diagram

- Offers
- Includes
- Employs
- Includes
- Teaches
- Takes
- Enrols In
- Tutors
- Tutors
- Tutors

Entity Relationship Modelling
Can a star be contracted by multiple studios?
Can a movie have multiple contracts with a studio?
Can a star act in multiple movies?
What is constrained?
Participation Constraints

- Does every department have a manager?
  - If so, this is a participation constraint: the participation of Departments in Manages is said to be total (vs. partial).
  - Every Departments entity must appear in an instance of the Manages relationship.
Weak Entities

- A **weak entity** can be identified uniquely only by considering the primary key of another (owner) entity.
  - Owner entity set and weak entity set must participate in a one-to-many relationship set (one owner, many weak entities).
  - Weak entity set must have total participation in this **identifying** relationship set.
As in C++, or other PLs, attributes are inherited.

If we declare A ISA B, every A entity is also considered to be a B entity.

**Overlap constraints:** Can Joe be an Hourly_Emps as well as a Contract_Emps entity? (Allowed/disallowed)

**Covering constraints:** Does every Employees entity also have to be an Hourly_Emps or a Contract_Emps entity? (Yes/no)

**Reasons for using ISA:**
- To add descriptive attributes specific to a subclass.
- To identify entities that participate in a relationship.
Aggregation

- Used when we have to model a relationship involving (entity sets and) a relationship set.
- **Aggregation** allows us to treat a relationship set as an entity set for purposes of participation in (other) relationships.

**Aggregation vs. ternary relationship:**
- Monitors is a distinct relationship, with a descriptive attribute.
- Also, can say that each sponsorship is monitored by at most one employee.
Conceptual Design Using the ER Model

- **Design choices:**
  - Should a concept be modeled as an entity or an attribute?
  - Should a concept be modeled as an entity or a relationship?
  - Identifying relationships: Binary or ternary? Aggregation?

- **Constraints in the ER Model:**
  - A lot of data semantics can (and should) be captured.
  - But some constraints cannot be captured in ER diagrams.
Should *address* be an attribute of Employees or an entity (connected to Employees by a relationship)?

- Depends upon the use we want to make of address information, and the semantics of the data:
  - If we have several addresses per employee, *address* must be an entity (since attributes cannot be set-valued).
  - If the structure (city, street, etc.) is important, e.g., we want to retrieve employees in a given city, *address* must be modeled as an entity (since attribute values are atomic).
Works_In4 does not allow an employee to work in a department for two or more periods.

Similar to the problem of wanting to record several addresses for an employee: We want to record several values of the descriptive attributes for each instance of this relationship. Accomplished by introducing new entity set, Duration.
First ER diagram OK if a manager gets a separate discretionary budget for each dept.

What if a manager gets a discretionary budget that covers all managed depts?

- Redundancy: $dbudget$ stored for each dept managed by manager.
- Misleading: Suggests $dbudget$ associated with department-mgr combination.

This fixes the problem!
If each policy is owned by just 1 employee, and each dependent is tied to the covering policy, first diagram is inaccurate.

What are the additional constraints in the 2nd diagram?
Previous example illustrated a case when two binary relationships were better than one ternary relationship.

An example in the other direction: a ternary relation **Contracts** relates entity sets **Parts, Departments** and **Suppliers**, and has descriptive attribute **qty**. No combination of binary relationships is an adequate substitute:

- S “can-supply” P, D “needs” P, and D “deals-with” S does not imply that D has agreed to buy P from S.
- How do we record **qty**?
Summary of Conceptual Design

► Conceptual design follows requirements analysis,
  ► Yields a high-level description of data to be stored

► ER model popular for conceptual design
  ► Constructs are expressive, close to the way people think about their applications.

► Basic constructs: entities, relationships, and attributes (of entities and relationships).

► Some additional constructs: weak entities, ISA hierarchies, and aggregation.

► Note: There are many variations on ER model.
Several kinds of integrity constraints can be expressed in the ER model: *key constraints*, *participation constraints*, and *overlap/covering constraints* for ISA hierarchies. Some *foreign key constraints* are also implicit in the definition of a relationship set.

Some constraints (notably, *functional dependencies*) cannot be expressed in the ER model.

Constraints play an important role in determining the best database design for an enterprise.
ER design is subjective. There are often many ways to model a given scenario! Analyzing alternatives can be tricky, especially for a large enterprise. Common choices include:

- Entity vs. attribute, entity vs. relationship, binary or n-ary relationship, whether or not to use ISA hierarchies, and whether or not to use aggregation.

Ensuring good database design: resulting relational schema should be analyzed and refined further. FD information and normalization techniques are especially useful.
Homework

“A database will be made to store information about patients in a hospital. On arrival, each patient’s personal details (name, address, and telephone number) are recorded where possible, and they are given an admission number. They are then assigned to a particular ward (Accident and Emergency, Cardiology, Oncology, etc.). In each ward there are a number of doctors and nurses. A patient will be treated by one doctor and several nurses over the course of their stay, and each doctor and nurse may be involved with several patients at any given time.”
Homework – cont.

► Identify the entities, attributes, relationships, and cardinality ratios from the description.

► Draw an entity-relationship diagram showing the items you identified.
End of the second lecture...

If libraries were like relational databases