EECS 647: Introduction to Database Systems

Instructor: Luke Huan

Spring 2009
Administrative

- I have communicated with KU Bookstore inquiring about the text book status.
- Take home background survey is due this Wednesday.
- I have sent a request to EECS help desk to create your PostgreSQL account. More details later.
- A good reference book:
Administrative

- The last day to drop a full semester course and not have it appear on the student's transcript is **Thursday, February 5**.
- The last day to enroll/add/increase credits for Spring 2009 for full semester courses is **Thursday, February 12**.
Today’s Outline

- Database design
- ER Model
  - Entities and Attributes
  - Entity Types, Value Sets, and Key Attributes
  - Relationships and Relationship Types
  - Weak Entity Types
  - Roles and Attributes in Relationship Types
- ER Diagrams – Notation
- Case studies
Aristotle’s (384-322 BC) Ontology

- Substance
  - plants, animals, ...
- Quality
- Quantity
- Where
- When
- Relation
- Action

How to describe the world?

http://en.wikipedia.org/wiki/Aristotle
Ontology

- **<philosophy>** A systematic account of *Existence*.
- **<artificial intelligence>** (From philosophy) An explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them.
- **<information science>** The hierarchical structuring of knowledge about things by subcategorising them according to their essential (or at least relevant and/or cognitive) qualities.
Database Design

- Entity/Relationship (E/R) model is the most widely used method in DB to describe a physical world
  - Intuitive and convenient
  - Adopted by historical reasons
  - But not necessarily implemented by DBMS
  - Other models are available
- Translate specification to the data model of DBMS
  - Relational, XML, object-oriented, etc.
- Create DBMS schema
Develop a Database for a Corporation
Develop a Database for a Corporation

- Employee
  - Employee ID
- Department
  - Department ID
- Work in Relationship
  - Works in: an employee works in a department
Entities and Attributes

- **Entity**: A specific object or “thing” in the mini-world that is represented in the database.

  For example, the EMPLOYEE John Smith, the Research DEPARTMENT, the ProductX PROJECT.

- **Attributes**: properties used to describe an entity.

  For example, an EMPLOYEE entity may have a Name, SSN, Address, Sex, BirthDate

- A specific entity will have a *value* for each of its attributes.

  For example, a specific employee entity may have Name='John Smith', SSN='123456789', Address ='731 Fondren, Houston, TX', Sex='M', BirthDate='09-JAN-55'
A *relationship* relates two or more distinct entities with a specific meaning.

For example, EMPLOYEE John Smith works on the ProductX PROJECT or EMPLOYEE Franklin Wong manages the Research DEPARTMENT.

Relationships of the same type are grouped or typed into a *relationship type*.

For example, the WORKS_ON relationship type in which EMPLOYEES and PROJECTs participate, or the MANAGES relationship type in which EMPLOYEES and DEPARTMENTs participate.
ER for a Corporation

- Employee, Department
- Works In relationship

Diagram:

- **E ID**
  - Employees
  - Works in
  - Since
  - D ID
  - Departments
More Details: Attributes

- Entity: Employee
- Name: Huan, Jun
- SSN: XXX-XX-XXXX
- Previous Degree: {B.S., M.S., Ph.D.}
- Teaching Classes in S’09: {EECS 647}
- Teaching Credit Hours in S’09: 3
Types of Attributes

- Simple vs. Composite Attributes
  - **Simple**: Each entity has a single atomic value for the attribute. For example, SSN or Sex.
  - **Composite**: The attribute may be composed of several components. For example, Name (FirstName, MiddleName, LastName).
Types of Attributes (cont.)

- Single-valued vs. Multi-valued.
  - **Single-valued**: an entity may have at most one value for the attribute
  - **Multi-valued**: An entity may have multiple values for that attribute. For example, PreviousDegrees of a STUDENT. {PreviousDegrees}.

- **NULL** values
  - What if the student does not hold a previous degree?
  - What if the student has a previous degree but the information is not provided?
  - Apartment number in an address
Types of Attributes (cont.)

- Stored vs. derived
  - Number of credit hours a student took in a semester
  - GPA of a student in a semester
Key Attributes

- Entities with the same basic attributes are grouped or typed into an entity type.
  - For example, the EMPLOYEE entity type or the PROJECT entity type.

- An attribute of an entity type for which each entity must have a unique value is called a key attribute of the entity type. For example, SSN of EMPLOYEE.
  - A key attribute may be composite.
  - An entity type may have more than one key.
## SUMMARY OF ER-DIAGRAM NOTATION

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ENTITY TYPE</td>
</tr>
<tr>
<td></td>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>KEY ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>MULTIVALEUED ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>COMPOSITE ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>DERIVED ATTRIBUTE</td>
</tr>
</tbody>
</table>
An Exercise

- Design the ER for the entity sets DEPARTMENT
- Attributes: Name, Number, locations, Manager, Manager_start_data
- We could identify a department by either Name or Number
- Each department may have multiple locations
Relationships

- A *relationship* relates two or more distinct entities with a specific meaning.
- Relationships of the same type are grouped or typed into a *relationship type*.
- The *degree of a relationship type* is the number of participating entity types.
  - Both MANAGES and WORKS_ON are binary relationships.
Instances of a relationship

EMPLOYEES  WORKS_FOR  DEPARTMENTS

1 to many
More Examples

- Each student may have exactly one account.
- Each faculty may teach many courses, but any course may be listed under exactly one faculty name.
- Each student may enroll many courses.

Diagram:

- **Students** Own **Ku Accounts**
- **Courses** TaughtBy **Instructors**
- **Students** Enroll **Courses**
Structural Constraints (I)

- Maximum Cardinality
  - One-to-one (1:1)
  - One-to-many (1:N) or Many-to-one (N:1)
  - Many-to-many
One-to-many (1:N) RELATIONSHIP

EMPLOYEE

MANAGES

DEPARTMENT

e_1

e_2
e_3
e_4
e_5
e_6
e_7

r_1

r_2

r_3

r_4

r_5

r_6

r_7

d_1
d_2
d_3

r_1

r_2

r_3

r_4

r_5

r_6

r_7
Many-to-many (M:N) RELATIONSHIP

Employees and Projects
Structural Constraints (II)

- Minimum Cardinality (also called participation constraint or existence dependency constraints)
  - Zero (partial participation)
  - One or more (total participation)
- Faculty members Teach Courses
  - Each faculty member may teach zero, one, or multiple courses.
  - Each course has to be taught by one faculty member.

Denote total participation
Roles in relationships

- An entity set may participate more than once in a relationship set
  - May need to label edges to distinguish roles

- Examples
  - People are married as husband and wife; label needed
  - People are roommates of each other; label not needed
Recursive relationship

- We can also have a *recursive relationship type*.
- Both participations are same entity type in different roles.
- For example, SUPERVISION relationships between EMPLOYEE (in role of supervisor or boss) and (another) EMPLOYEE (in role of subordinate or worker).
- In ER diagram, need to display role names to distinguish participations.
A RECURSIVE RELATIONSHIP
SUPERVISION

EMPLOYEE

SUPERVISION

Weak Entity Types

- A weak entity is an entity that does not have a key attribute.
- A weak entity must participate in an identifying relationship type with an owner or identifying entity type.
- Entities are identified by the combination of:
  - A partial key of the weak entity type.
  - The particular entity they are related to in the identifying entity type.

**Example:**

Suppose that a DEPENDENT entity is identified by the dependent’s first name and birthdate, *and* the specific EMPLOYEE that the dependent is related to. DEPENDENT is a weak entity type with EMPLOYEE as its identifying entity type via the identifying relationship type DEPENDENT_OF.
SUMMARY OF ER-DIAGRAM
NOTATION FOR ER SCHEMAS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>ENTITY TYPE</td>
</tr>
<tr>
<td>E</td>
<td>WEAK ENTITY TYPE</td>
</tr>
<tr>
<td>R</td>
<td>RELATIONSHIP TYPE</td>
</tr>
<tr>
<td></td>
<td>IDENTIFYING RELATIONSHIP TYPE</td>
</tr>
<tr>
<td></td>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>KEY ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>MULTIVALUED ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>COMPOSITE ATTRIBUTE</td>
</tr>
<tr>
<td>E</td>
<td>DERIVED ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R</td>
</tr>
</tbody>
</table>
An Database Design Example

- The company is organized into DEPARTMENTs. Each department has a name, number and an employee who manages the department. We keep track of the start date of the department manager.

- Each department controls a number of PROJECTs. Each project has a name, number and is located at a single location.

- We store each EMPLOYEE’s social security number, address, salary, sex, and birthdate. Each employee works for one department but may work on several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct supervisor of each employee.

- Each employee may have a number of DEPENDENTs. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.