CSE 532 – Theory of Database Systems
Lecture 05 (Chapter 4)
Conceptual Modeling of Databases with E-R Model and UML

Ilchul Yoon
Assistant Professor
State University of New York, Korea

Representing Participation Constraints in the Relational Model

- Inclusion dependency: Every professor works in at least one dept.
  - in the relational model: (easy)
    - Professor (id) references WorksIn (ProfId) - means simply “references”
  - in SQL:
    - Simple case – ProfId is a key in WorksIn (i.e., if every professor works in exactly one department) then it is easy:
      - Solution 1: FOREIGN KEY Professor (id) REFERENCES WorksIn (ProfId)
      - Solution 2: We can combine the tables Professor and WorksIn
    - General case – ProfId is not a key in WorksIn, so can’t use foreign key constraint. Then must use ASSERTION statement – more complex (shown later)
Participation and Key Constraint in the Relational Model – Simple Case

- Example:

<table>
<thead>
<tr>
<th>Id</th>
<th>ProfId</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxx</td>
<td>1123 CSE</td>
</tr>
<tr>
<td>yyyyy</td>
<td>4100 ECO</td>
</tr>
<tr>
<td>zzzzzz</td>
<td>3216 AMS</td>
</tr>
</tbody>
</table>

Representing Participation and Key Constraint in SQL – Simple Case, Solution 1

- If both participation and key constraints are specified, then we can use foreign key constraint in entity table

```sql
CREATE TABLE Professor (    
  Id INTEGER,  
  PRIMARY KEY (Id),  
  -- Id can’t be null  
  FOREIGN KEY (Id) REFERENCES WorksIn (ProfId)  
  -- all professors participate
) 
```
Participation and Key Constraint in Relational Model – Simple Case, Solution 2

- Alternative solution if both key and participation constraints apply
  - Merge the tables representing the entity and relationship types
    - Can be done since the 1-1 and onto relationship between the rows of the entity set and the rows in the relationship sets allows us to put all attributes in one table

```
CREATE TABLE Prof_WorksIn (  
  Id INTEGER,  
  Name CHAR(40),  
  DeptId CHAR(4),  
  PRIMARY KEY (Id)  
)  
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Id</th>
<th>DeptId</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxxxx</td>
<td>1123</td>
<td>CSE</td>
</tr>
<tr>
<td>yyyyyny</td>
<td>4100</td>
<td>ECO</td>
</tr>
<tr>
<td>zzzzzzz</td>
<td>3216</td>
<td>AMS</td>
</tr>
</tbody>
</table>

- Example

**Merge of Professor and WorksIn**
Representing Participation Constraint in the Relational Model – General Case

- Cannot use foreign key in Professor if ProfId is not a CK in WorksIn – must use assertions

Professor

<table>
<thead>
<tr>
<th>Id</th>
<th>ProfId</th>
</tr>
</thead>
<tbody>
<tr>
<td>1123</td>
<td>1123</td>
</tr>
<tr>
<td>4100</td>
<td>1123</td>
</tr>
<tr>
<td>3216</td>
<td>3216</td>
</tr>
</tbody>
</table>

WorksIn

<table>
<thead>
<tr>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1123</td>
</tr>
<tr>
<td>1123</td>
</tr>
<tr>
<td>4100</td>
</tr>
<tr>
<td>4100</td>
</tr>
<tr>
<td>3216</td>
</tr>
<tr>
<td>3216</td>
</tr>
</tbody>
</table>

CREATE ASSERTION ProfsInDepts
CHECK ( NOT EXISTS ( SELECT * FROM Professor P WHERE NOT EXISTS ( SELECT * FROM WorksIn W WHERE P.Id = W.ProfId ) ) )

Representing Part-Of Constraint in the Relational Model

- Non-exclusive part-of
  - Case 1: Subpart can exist independently but can be part of at most one whole
    - Make one relation from subpart entity and relationship
  - Case 2: Subpart can exist independently and be shared between different wholes
    - Translate into separate relations
Representing Part-Of Constraint in the Relational Model

CREATE TABLE WheelMergedWithPartOf
  SerialNumber    INTEGER,
  Size            CHAR(10),
  Manufacturer    CHAR(20),
  VehicleID       CHAR(20),
PRIMARY KEY (SerialNumber),
FOREIGN KEY (VehicleID) REFERENCES Automobile

Representing Part-Of Constraint in the Relational Model

- Exclusive part-of
  - Subpart is a weak entity (one and only one participation)
    - Use the rule for participation and key constraints
Entity or Attribute?

- Sometimes information can be represented as either an entity or an attribute.

Entity or Relationship?

(Next slide: Semester has attributes)
(Non-) Equivalence of Diagrams

- Transformations between binary and ternary relationships.

Summary of E-R Notations
Crow’s Foot Notation for Cardinality

from Zero to Many
from One to Many
from One to One
i.e., one and only one
from Zero to One

Use of UML to represent E-R Model

- Unified Modeling Language
  - Unified representation for software design
  - Class diagram will be used for E-R model

- Other diagrams in UML
  - Use-case diagram: user-interaction
  - Sequence diagram: dynamic interaction between objects
  - State diagram: state transition of complex objects
  - Activity diagram: flowchart or workflow
  - Communication diagram (formerly, collaboration diagram): message flow between objects
  - ....
Representing Entities in UML

- Modeled as classes
  - Attributes
  - Methods represent transactions associated with entities

- Object Constraint Language (OCL) for specifying constraints
  - Can be used for specifying keys, CHECK and ASSERTION
  - e.g., {Key: Customer, Product; Date}

- Extensibility mechanism
  - Use of stereotype for PK and FK
    - SSN: INT <<PK>>
    - <<FK Professor (Id)>> ProfID: INT

Representing Entities in UML

<table>
<thead>
<tr>
<th>PERSON</th>
<th>STUDENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: CHAR(20)</td>
<td>Name: CHAR(20)</td>
</tr>
<tr>
<td>SSN: INTEGER &lt;&lt;PK&gt;&gt;</td>
<td>Id: INTEGER &lt;&lt;PK&gt;&gt;</td>
</tr>
<tr>
<td>Address: CHAR(50)</td>
<td>Address: CHAR(50)</td>
</tr>
<tr>
<td>Hobbies[0..*]: CHAR(10)</td>
<td>GPA: DEC(2,1)</td>
</tr>
<tr>
<td>ChangeAddr(NewAddr:CHAR(50))</td>
<td>StartDate: DATE</td>
</tr>
<tr>
<td>AddHobby(Hobby:CHAR(10))</td>
<td>ChangeAddr(NewAddr:CHAR(50))</td>
</tr>
<tr>
<td>...</td>
<td>SetStartDate(Date:DATE)</td>
</tr>
<tr>
<td></td>
<td>&lt;&lt;Invariant&gt;&gt; self.GPA &gt; 2.0</td>
</tr>
</tbody>
</table>
Representing Relationship in UML

- Relationships are called **associations**
  - For binary association types, UML uses a line to connect classes
  - For 3+ association types, uses diamond symbol

![Diagram showing relationships between classes](image)

Representing Relationship in UML

- Association class
  - Previously, relationship attributes
  - Connected to an association using a dashed line

- Multiplicity constraints on roles
  - Range specification: \( n..m \)
  - Lower and upper bounds on the number of objects of a class that can be connected by means of an association type to any given set (or combination) of objects attached to the other ends of the association.
UML Multiplicity vs. E-R Cardinality Constraint

- Looks similar but has “opposite meaning”
- In E-R model,

![E-R Model Diagram]

- In UML,

![UML Diagram]

Certainly, opposite meaning and same expression power (including key constraints) for binary associations.

Cardinality constraint in E-R  Equivalent multiplicity constraint in UML
UML Multiplicity vs. E-R Cardinality Constraint

- For ternary and higher-degree, the difference is greater
  - Incomparable expression power
  - Some cardinality constraints cannot be represented in multiplicity in UML

Foreign Keys in UML
IsA Hierarchy in UML

- IsA relationship is called **generalization**
  - Solid arrow with a large hollow head leading from a subclass to a superclass

![Diagram of IsA Hierarchy in UML](image)

Participation Constraint in UML

- **Obvious for binary associations**
  - Use multiplicity lower bound greater than or equal to 1

![Participation Constraint in UML](image)

- **For ternary and higher-degree associations?**
  - Difficult to represent identical semantics
Participation Constraint in UML

- In ER,
  - for every entity \( c \in C \) there are entities \( d \in D \) and \( e \in E \) that participate in a relationship \( a \in A \) with \( c \).

- In UML,
  - for every pair of objects \( c \in C \) and \( e \in E \), there is at least one object \( d \in D \) that participates in a relationship \( a \in A \) with \( c \) and \( e \).

```
Part-Of Relationship in UML

- Non-exclusive part-of relationship is called aggregation
  - Often accompanied by appropriate multiplicity constraints

  AUTOMOBILE 0..1 4 WHEEL
  PROGRAM 0..* 3..* COURSE

- Exclusive part-of relationship is called composition
  - Viewed as a special kind of aggregation

  EMPLOYEE 1 1 DEPENDENT
  UNIVERSITY 1 PROGRAM
```
Summary of UML Notations

- **Class**
  - Attributes
  - Role 1
  - Role 2
  - Ternary Association
    - 2..5
    - Association Class
      - Attributes
    - 1..*

- **Binary Association**
  - 0:1
  - 1..*

- **Role**
  - Role 1
  - Role 2

- **Aggregation**
- **Composition**
  - (Disjoint, complete)

- **Generalization**