Entity-Relationship (E-R) Diagrams

Outline

- Relational Algebra (6.1)
- E/R Model (7.2 - 7.4)
- E/R Diagrams (7.5)
- Reduction to Schema (7.6)
- Relational Database Design (7.7)
- Functional Dependencies (8.1 – 8.4)
- Normalization (8.5 – 8.7)
- Relational Query Languages
- SQL Basics
- Formal Semantics of SQL
E-R Diagrams

Rectangles represent entity sets.
Diamonds represent relationship sets.
Attributes listed inside entity rectangle
Underline indicates primary key attributes

instructor

<table>
<thead>
<tr>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
</tr>
<tr>
<td>salary</td>
</tr>
</tbody>
</table>

student

<table>
<thead>
<tr>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
</tr>
<tr>
<td>tot_cred</td>
</tr>
</tbody>
</table>

advisor

Entity With Composite, Multivalued, and Derived Attributes

instructor

<table>
<thead>
<tr>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
</tr>
<tr>
<td>first_name</td>
</tr>
<tr>
<td>middle_initial</td>
</tr>
<tr>
<td>last_name</td>
</tr>
<tr>
<td>address</td>
</tr>
<tr>
<td>street</td>
</tr>
<tr>
<td>street_number</td>
</tr>
<tr>
<td>street_name</td>
</tr>
<tr>
<td>apt_number</td>
</tr>
<tr>
<td>city</td>
</tr>
<tr>
<td>state</td>
</tr>
<tr>
<td>zip</td>
</tr>
<tr>
<td>{ phone_number }</td>
</tr>
<tr>
<td>date_of_birth</td>
</tr>
<tr>
<td>age ( )</td>
</tr>
</tbody>
</table>
Entity sets of a relationship need not be distinct
- Each occurrence of an entity set plays a “role” in the relationship
- The labels “course_id” and “prereq_id” are called roles.
Cardinality Constraints

- We express cardinality constraints by drawing either a directed line (→), signifying “one,” or an undirected line (—), signifying “many,” between the relationship set and the entity set.
- One-to-one relationship:
  - A student is associated with at most one instructor via the relationship advisor
  - A student is associated with at most one department via stud_dept

One-to-One Relationship

- one-to-one relationship between an instructor and a student
  - an instructor is associated with at most one student via advisor
  - and a student is associated with at most one instructor via advisor
One-to-Many Relationship

- one-to-many relationship between an instructor and a student
  - an instructor is associated with any number (including 0) of students via advisor
  - a student is associated with at most one instructor via advisor,

```
<table>
<thead>
<tr>
<th>instructor</th>
<th></th>
<th>student</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>name</td>
<td>ID</td>
</tr>
<tr>
<td>name</td>
<td>salary</td>
<td>name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tot_cred</td>
</tr>
</tbody>
</table>

Many-to-One Relationships

- In a many-to-one relationship between an instructor and a student,
  - an instructor is associated with at most one student via advisor,
  - and a student is associated with several (including 0) instructors via advisor

```
<table>
<thead>
<tr>
<th>instructor</th>
<th></th>
<th>student</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>name</td>
<td>ID</td>
</tr>
<tr>
<td>name</td>
<td>salary</td>
<td>name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tot_cred</td>
</tr>
</tbody>
</table>
Many-to-Many Relationship

- An instructor is associated with any number (possibly 0) of students via *advisor*
- A student is associated with any number (possibly 0) of instructors via *advisor*

<table>
<thead>
<tr>
<th>instructor</th>
<th>student</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ID</td>
</tr>
<tr>
<td>name</td>
<td>name</td>
</tr>
<tr>
<td>salary</td>
<td>tot_cred</td>
</tr>
</tbody>
</table>

advisor

---

**Participation of an Entity Set in a Relationship Set**

- Total participation (indicated by **double line**): every entity in the entity set participates in at least one relationship in the relationship set
- E.g., participation of *section* in *sec_course* is total
  - every *section* must have an associated course
- Partial participation: some entities may not participate in any relationship in the relationship set
- Example: participation of *course* in *section* is partial
Alternative Notation for Cardinality Limits

- Cardinality limits can also express participation constraints

E-R Diagram with a Ternary Relationship
Cardinality Constraints on Ternary Relationship

- We allow at most one arrow out of a ternary (or greater degree) relationship to indicate a cardinality constraint.
- E.g., an arrow from proj_guide to instructor indicates each student has at most one guide for a project.
- If there is more than one arrow, there are two ways of defining the meaning.
  1. Each A entity is associated with a unique entity from B and C.
  2. Each pair of entities (A, B) is associated with a unique C entity, and each pair (A, C) is associated with a unique B.
- Each alternative has been used in different formalisms.
- To avoid confusion we outlaw more than one arrow.

Weak Entity Sets

- An entity set that does not have a primary key is referred to as a weak entity set.
- The existence of a weak entity set depends on the existence of a identifying entity set.
  - It must relate to the identifying entity set via a one-to-many relationship set from the identifying to the weak entity set.
  - Weak side of relationship set must be total.
  - Identifying relationship depicted using a double diamond.
- The discriminator (or partial key) of a weak entity set is the set of attributes that help distinguish among all the entities of a weak entity set.
- The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existent-dependent, plus the weak entity set’s discriminator.
Weak Entity Sets (Cont.)

- We underline the discriminator of a weak entity set with a dashed line.
- We put the identifying relationship of a weak entity in a double diamond.
- Primary key for section – (course_id, sec_id, semester, year)

```
course
  course_id
  title
  credits

sec_course

section
  sec_id
  semester
  year
```

Note: the primary key of the strong entity set is not explicitly stored with the weak entity set, since it is implicit in the identifying relationship.

- If course_id were explicitly stored, section could be made a strong entity, but then the relationship between section and course would be duplicated by an implicit relationship defined by the attribute course_id common to course and section
Symbols Used in E-R Notation (Cont.)

- Many-to-many relationship
- Many-to-one relationship
- One-to-one relationship
- Cardinality limits
- Role indicator
- Role name

Redundant Attributes

- Suppose we have entity sets
  - instructor, with attributes including dept_name
  - department
  - and a relationship
    - inst_dept relating instructor and department
- Attribute dept_name in entity instructor is redundant since there is an explicit relationship inst_dept which relates instructors to departments
  - The attribute replicates information present in the relationship, and should be removed from instructor
  - BUT: redundant attributes sometimes get reintroduced when converting back to tables
Reduction to Relational Schemas

Outline

- Relational Algebra (6.1)
- E/R Model (7.2 - 7.4)
- E/R Diagrams (7.5)
- Reduction to Schema (7.6)
- Relational Database Design (7.7)
- Functional Dependencies (8.1 – 8.4)
- Normalization (8.5 – 8.7)
- Relational Query Languages
- SQL Basics
- Formal Semantics of SQL
Reduction to Relation Schemas

- Entity sets and relationship sets can be expressed uniformly as relation schemas that represent the contents of the database.
- A database which conforms to an E-R diagram can be represented by a collection of schemas.
- For each entity set and relationship set there is a unique schema that is assigned the name of the corresponding entity set or relationship set.
- Each schema has a number of columns (generally corresponding to attributes), which have unique names.

Representing Entity Sets With Simple Attributes

- A strong entity set reduces to a schema with the same attributes:
  \[ \text{student(}\text{ID, name, tot_cred)} \]
- A weak entity set becomes a table that includes a foreign key for the primary key of the identifying strong entity set:
  \[ \text{section (course_id, sec_id, sem, year)} \]
Representing Relationship Sets

- A many-to-many relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.
- Example: schema for relationship set **advisor**

\[
\text{advisor} = (s\_id, i\_id)
\]

Redundancy of Schemas

- Many-to-one and one-to-many relationship sets that are total on the many-side can be represented by adding an extra attribute to the “many” side, containing the primary key of the “one” side
- Example: Instead of creating a schema for relationship set **inst_dept**, add an attribute **dept\_name** to the schema arising from entity set **instructor**
Redundancy of Schemas (Cont.)

- For one-to-one relationship sets, either side can be chosen to act as the “many” side
  - That is, extra attribute can be added to either of the tables corresponding to the two entity sets
    - Instructor
      - ID
      - name
      - salary
    - department
      - dept_name
      - building
      - budget
    - Instructor(ID, dept_name, name, salary)
    - or
    - department(dept_name, ID, building, budget)

Redundancy of Schemas (Cont.)

- If participation is partial on the “many” side, replacing a relationship schema by an extra attribute in the schema corresponding to the “many” side could result in null values (generally avoided)
  - i.e. the approach in the previous slides does not work
  - need to represent relationship as a separate table

- The schema corresponding to a relationship set linking a weak entity set to its identifying strong entity set is redundant.
  - Example: The section schema already contains the attributes that would appear in the sec_course schema

- Unless otherwise instructed, assume we wish to avoid NULLs when converting to relations, i.e. remove redundant relationship schemas only when total participation on side where adding attribute.
Composite Attributes

- Composite attributes flattened out
  - Example: given entity set instructor
    - with composite attribute name
    - with component attributes first_name and last_name
    - replace with name_first_name and name_last_name
      - Prefix omitted if there is no ambiguity
- Ignoring multivalued attributes, extended instructor schema is
  - instructor(ID, first_name, middle_initial, last_name, street_number, street_name, apt_number, city, state, zip_code, date_of_birth)

Multivalued Attributes

- Multivalued attribute $M$ of entity $E$ represented by a separate schema $EM$
  - Schema $EM$ includes $E$‘s primary key and attribute corresponding to $M$
  - Example: Multivalued attribute phone_number of instructor:
    - $inst\_phone= (ID, phone\_number)$
  - Each value of the multivalued attribute maps to separate tuple of $EM$:
    - instructor entity with primary key 22222 and numbers 456-7890 and 123-4567 maps to:
      - (22222, 456-7890)
      - (22222, 123-4567)