Logical Database Design

Lecture 4
Instructor Anna Sidorova

Agenda
- Announcements, Q&A
- Quiz
- Review of HW 1
- Logical database design (lecture)
- Move to the lab – Introduction to Oracle Designer

Next few classes
- Feb 14 – Introduction to relational DB, Oracle Designer
  - Hands on practice in the lab
  - HW 1 is due in class
- Feb 21 – Relational DB, Normalization
  - Project Proposal is due
- Feb 28 – Review for the midterm exam
  - HW 2 is due
- March 7 – Midterm exam

Based on Hoffer, Prescott and Topi Modern Database Management, (c) Prentice Hall 2009

Project Proposal
- Title page and the table of contents
- Executive summary
- Problem description including your understanding of the functional area
- The goal and benefits of the project
- The scope of the project
- Your approach to the project
- Team composition and relevant experiences
- Project Plan

Based on Hoffer, Prescott and Topi Modern Database Management, (c) Prentice Hall 2009

HW1
- Draw ERD models for the following descriptions:
  - Ch. 2, problems 20, 23
  - Ch. 3, problems 12, 15

Based on Hoffer, Prescott and Topi Modern Database Management, (c) Prentice Hall 2009
Objectives

- Definition of terms
- List five properties of relations
- State two properties of candidate keys
- Transform E-R and EER diagrams to relations
- Create tables with entity and relational integrity constraints

Relation

- A relation is a named, two-dimensional table of data
- Table is made up of rows (records), and columns (attributes or field)
- Not all tables qualify as relations

Requirements:

- Every relation has a unique name
- Every attribute value is atomic (not multivalued, not composite)
- Every row is unique (can’t have two rows with exactly the same values for all their fields)
- Attributes (columns) in tables have unique names
- The order of the columns is irrelevant

Correspondence with ER Model

- Relations (tables) correspond to entity types
- Rows correspond to entity instances
- Columns correspond to attributes

NOTE: The word relation (in relational database) is NOT the same the word relationship (in ER model)

Key Fields

- Keys are special fields that serve two main purposes:
  - Primary keys are unique identifiers of the relation in question. Examples include employee numbers, social security numbers, etc. This is how we can guarantee that all rows are unique
  - Foreign keys are identifiers that enable a dependent relation (on the many side of a relationship) to refer to its parent relation (on the one side of the relationship)
- Keys can be simple (a single field) or composite (more than one field)
- Keys usually are used as indexes to speed up the response to user queries (More on this in Ch. 6)
Integrity Constraints

- **Domain Constraints**
  - Allowable values for an attribute.
- **Entity Integrity**
  - No primary key attribute may be null. All primary key fields **MUST** have data.
- **Referential Integrity** – rule that states that any foreign key value (on the relation of the many side) **MUST** match a primary key value in the relation of the one side. (Or the foreign key can be null)

Enforcing referential integrity

- **Delete Rules**
  - Restrict – don’t allow delete of “parent” side if related rows exist in “dependent” side
  - Cascade – automatically delete “dependent” side rows that correspond with the “parent” side row to be deleted
  - Set-to-Null – set the foreign key in the dependent side to null if deleting from the parent side → not allowed for weak entities

Transforming an ER diagram into a relational DB

Transforming EER Diagrams into Relations

Mapping Regular Entities to Relations

1. Simple attributes: E-R attributes map directly onto the relation
2. Composite attributes: Use only their simple, component attributes
3. Multi-valued Attribute - Becomes a separate relation with a foreign key taken from the superior entity
Figure 5-8 Mapping a regular entity
(a) CUSTOMER entity type with simple attributes

(b) CUSTOMER relation

Figure 5-9 Mapping a composite attribute
(a) CUSTOMER entity type with composite attribute

(b) CUSTOMER relation with address detail

Figure 5-10 Mapping an entity with a multivalued attribute
(a) Multivalued attribute becomes a separate relation with foreign key

(b) One-to-many relationship between original entity and new relation

Figure 5-11 Example of mapping a weak entity
a) Weak entity DEPENDENT

b) Relations resulting from weak entity

NOTE: the domain constraint for the foreign key should NOT allow null value if DEPENDENT is a weak entity

Composite primary key
Transforming EER Diagrams into Relations

Mapping Binary Relationships

- One-to-Many - Primary key on the one side becomes a foreign key on the many side
- Many-to-Many - Create a new relation with the primary keys of the two entities as its primary key
- One-to-One - Primary key on the mandatory side becomes a foreign key on the optional side

Figure 5-12 Example of mapping a 1:M relationship

a) Relationship between customers and orders

Note the mandatory one

b) Mapping the relationship

Again, no null value in the

Figure 5-13 Example of mapping an M:N relationship

a) Completes relationship (M:N)

The Completes relationship will need to become a separate relation

b) Three resulting relations

New intersection relation

Figure 5-14 Example of mapping a binary 1:1 relationship

a) In_charge relationship (1:1)

Often in 1:1 relationships, one direction is optional

b) Resulting relations

Foreign key goes in the relation on the optional side, matching the primary key on the mandatory side
Transforming EER Diagrams into Relations

Mapping Associative Entities
- Identifier Not Assigned
  - Default primary key for the association relation is composed of the primary keys of the two entities (as in M:N relationship)
- Identifier Assigned
  - It is natural and familiar to end-users
  - Default identifier may not be unique

Figure 5-15 Example of mapping an associative entity
(a) An associative entity

Figure 5-16 Example of mapping an associative entity with an identifier
(a) SHIPMENT associative entity

Figure 5-15 Example of mapping an associative entity (cont.)
(b) Three resulting relations

Figure 5-16 Example of mapping an associative entity with an identifier (cont.)
(b) Three resulting relations

Transforming EER Diagrams into Relations

Mapping Unary Relationships
- One-to-Many - Recursive foreign key in the same relation
- Many-to-Many - Two relations:
  - One for the entity type
  - One for an associative relation in which the primary key has two attributes, both taken from the primary key of the entity
Figure 5-17 Mapping a unary 1:N relationship

(a) EMPLOYEE entity with unary relationship

(b) EMPLOYEE relation with recursive foreign key

Figure 5-18 Mapping a unary M:N relationship

(a) Bill-of-materials relationships (M:N)

(b) ITEM and COMPONENT relations

Figure 5-19 Mapping a ternary relationship

(a) PATIENT TREATMENT Ternary relationship with associative entity

(b) Mapping the ternary relationship PATIENT TREATMENT

Remember that the primary key MUST be unique 
This is why treatment date and time are included in the composite primary key 
But this makes a very cumbersome key... 
It would be better to create a surrogate key like Treatment#

Transforming EER Diagrams into Relations
Mapping Ternary (and n-ary) Relationships
- One relation for each entity and one for the associative entity
- Associative entity has foreign keys to each entity in the relationship

Transforming EER Diagrams into Relations
Mapping Supertype/Subtype Relationships
- One relation for supertype and for each subtype
- Supertype attributes (including identifier and subtype discriminator) go into supertype relation
- Subtype attributes go into each subtype; primary key of supertype relation also becomes primary key of subtype relation
- 1:1 relationship established between supertype and each subtype, with supertype as primary table
Let us consider a simple example: translate it to a relational DB model

Next
- Move to the lab (BA 333)
- Introduction to Oracle Designer
- Follow the tutorial.