Lecture 8

Database Systems

Instructor: M.Imran Khalil

Imrankhalil3@gmail.com

Resource:Imrankhalil3.wordpress.com

Todays lecture

- logical design and relational model

Objectives of logical design...

- Translate the conceptual design into a logical database design that can be implemented on a chosen DBMS
  - Input: conceptual model (ERD)
  - Output: relational schema, normalized relations
- Resulting database must meet user needs for:
  - Data sharing
  - Ease of access
  - Flexibility

Relational database components

- Data structure
  - Data organized into tables
- Data manipulation
  - Add, delete, modify, and retrieve using SQL
- Data integrity
  - Maintained using business rules

Why do I need to know this?

- Mapping conceptual models to relational schema is straight-forward
- CASE tools can perform many of the steps, but..
  - Often CASE cannot model complexity of data and relationship (e.G., Ternary relationships, supertype/subtypes)
  - There are times when legitimate alternates must be evaluated
  - You must be able to perform a quality check on CASE tool results

Some rules...

- Every table has a unique name.
- Attributes in tables have unique names.
- Every attribute value is atomic.
  - Multi-valued and composite attributes?
- Every row is unique.
- The order of the columns is irrelevant.
- The order of the rows is irrelevant.

The key…

- Relational modeling uses primary keys and foreign keys to maintain relationships
- Primary keys are typically the unique identifier noted on the conceptual model
- Foreign keys are the primary key of another entity to which an entity has a relationship
- Composite keys are primary keys that are made of more than one attribute
  - Weak entities
  - Associative entities

Constraints

- Domain constraints
  - Allowable values for an attribute as defined in the domain
- Entity integrity constraints
  - No primary key attribute may be null
- Operational constraints
  - Business rules
- Referential integrity constraints

Referential integrity constraint

- Maintains consistency among rows of two entities
  - matching of primary and foreign keys
- Enforcement options for deleting records
  - Restrict
  - Cascade
  - Set-to-Null

Transforming the EER diagram into relations

The steps:

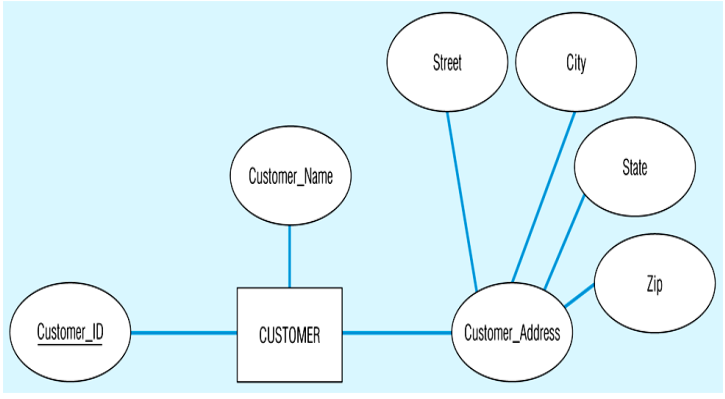
- Map regular/strong entities
- Map weak entities
- Map binary relationships
- Map associative entities
- Map unary relationships
- Map ternary relationships
- Map supertype/subtype relationships

Transforming E-R diagrams into relations

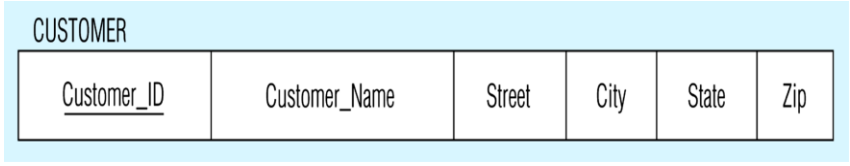
Mapping regular entities to relations

- Composite attributes: use only their simple, component attributes
- Multi-valued attributes: become a separate relation with a foreign key taken from the superior entity

Mapping a composite attribute



Looks like this using relational schema notation

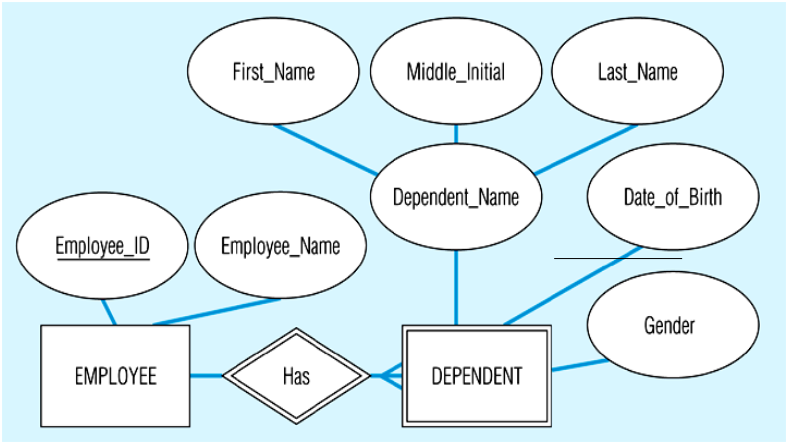


Transforming E-R diagrams into relations

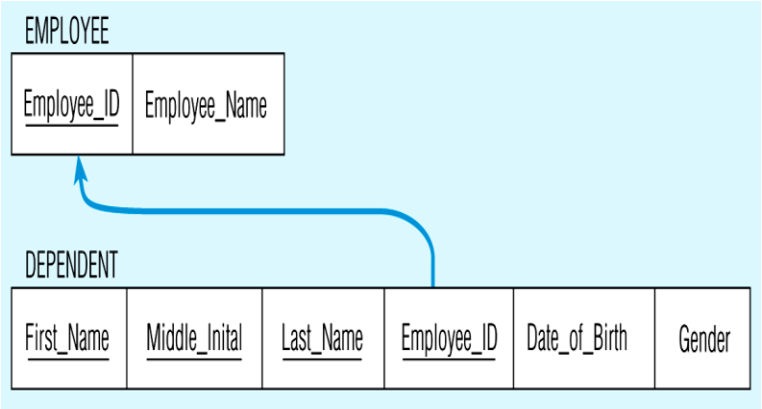
Mapping weak entities

- Becomes a separate relation with a foreign key taken from the superior entity

Example of mapping a weak entity



Looks like this using relational schema notation

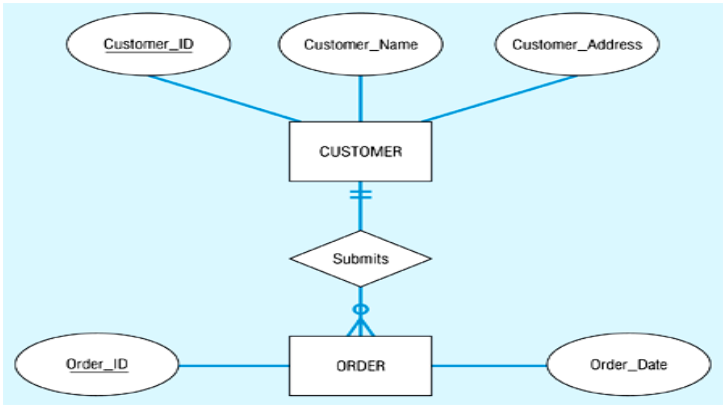


Transforming E-R diagrams into relations

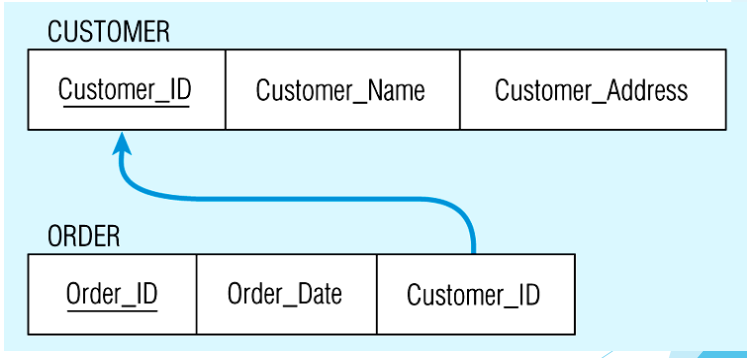
Mapping binary relationships

- One-to-many - primary key on the one side becomes a foreign key on the many side
- Many-to-many - create a new relation (associative entity) with the primary keys of the two entities as its primary key
  - I like to call **intersection entities** these to distinguish them from associative entities created at the conceptual level
- One-to-one - primary key on the mandatory side becomes a foreign key on the optional side

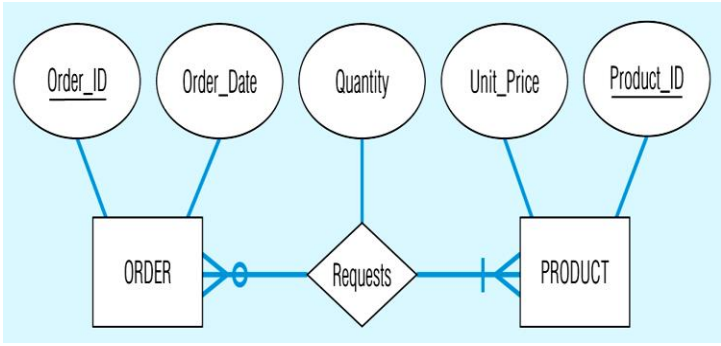
Example of mapping a 1:M relationship



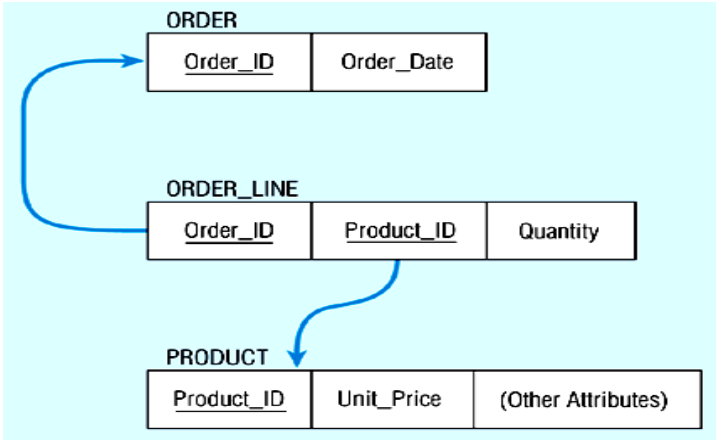
Looks like this using relational schema notation



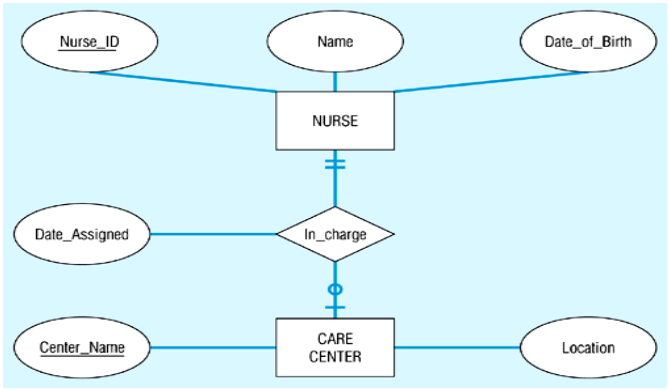
Example of mapping an M:M relationship



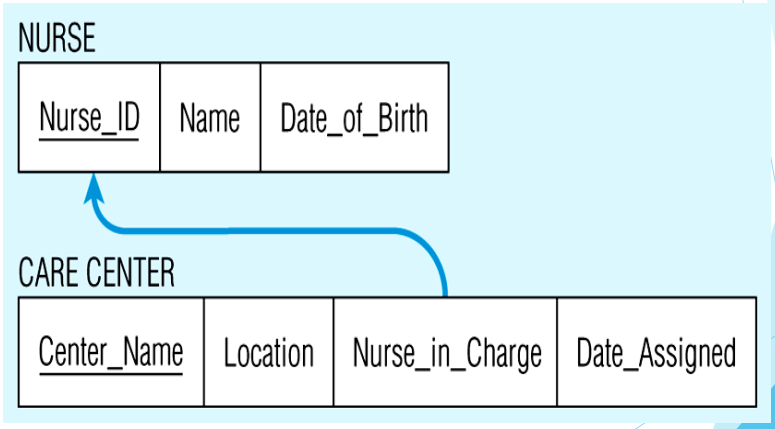
Looks like this using relational schema notation



Mapping a binary 1:1 relationship



Looks like this using relational schema notation

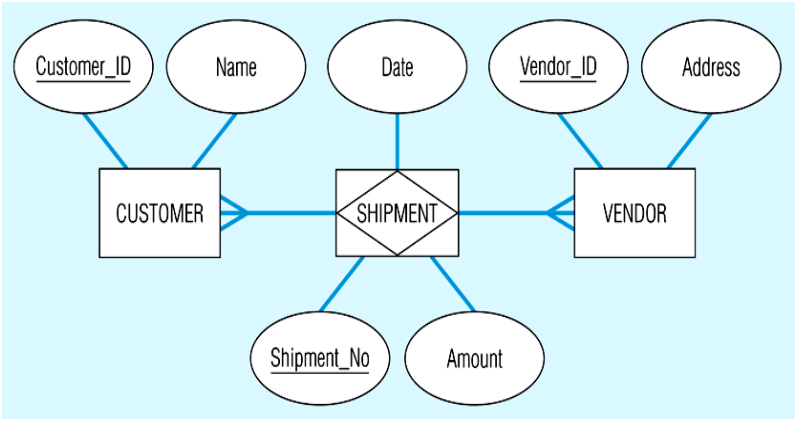


Transforming E-R diagrams into relations

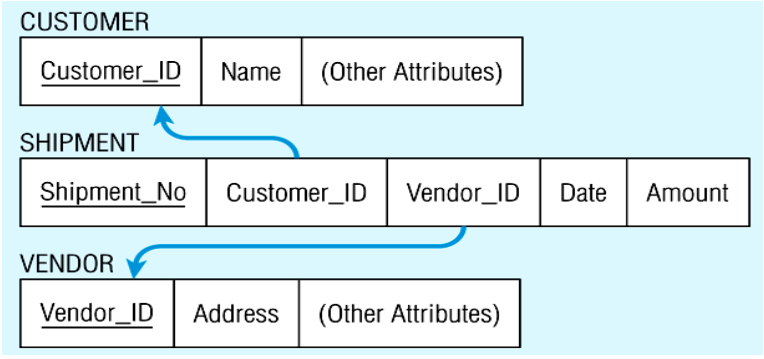
Mapping associative entities

- Identifier not assigned
  - Default primary key for the association relation is the primary keys of the two entities
- Identifier assigned
  - It is natural and familiar to end-users
  - Default identifier may not be unique

Mapping an associative entity with an identifier



Looks like this using relational schema notation



Transforming E-R 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

For example...