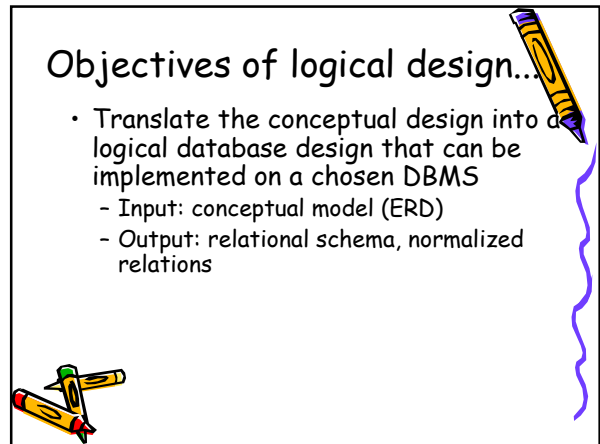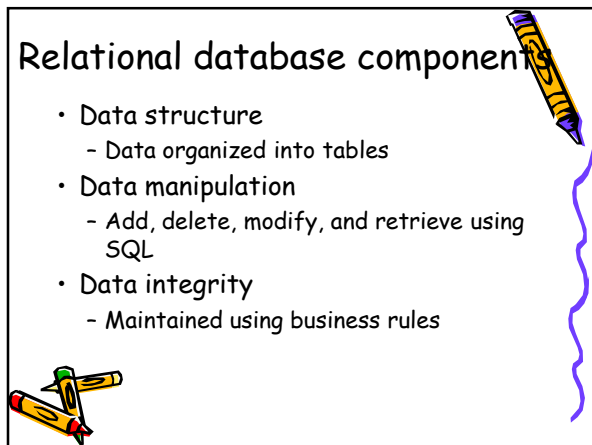# Logical database design and the relational model

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
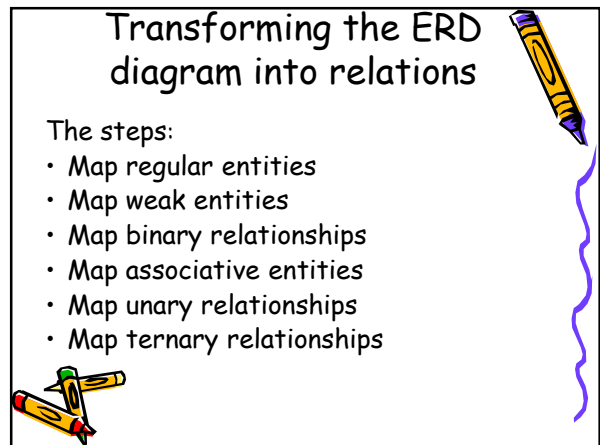
Relational database components

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

Transforming the ERD diagram into relations
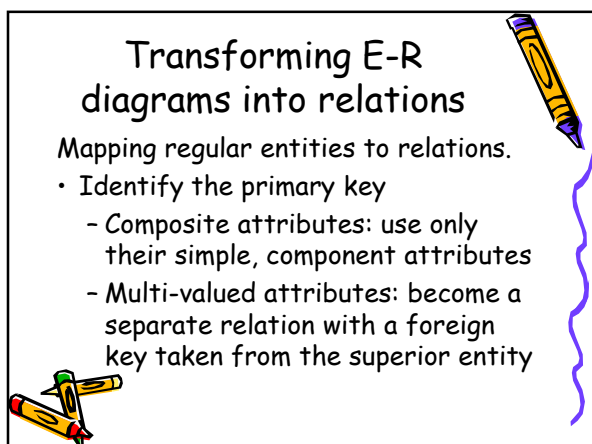
The steps:
- Map regular entities
- Map weak entities
- Map binary relationships
- Map associative entities
- Map unary relationships
- Map ternary relationships

Transforming E-R diagrams into relations

Mapping regular entities to relations.
- Identify the primary key
  - 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
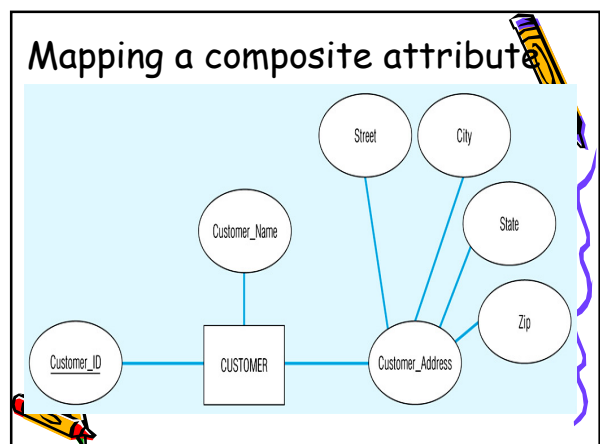
Looks like this using relational schema notation

CUSTOMER

Customer_ID	Customer_Name	Street	City	State	Zip

Transforming E-R diagrams into relations

Mapping weak entities
- Becomes a separate relation with a foreign key taken from the superior entity.
- The primary key is a combination of the parent identifier and the weak entity identifier.

Example of mapping a weak entity
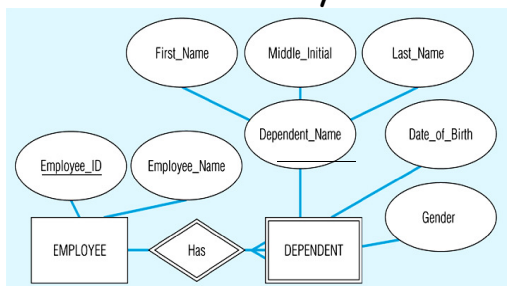
Looks like this using relational schema notation

EMPLOYEE

Employee_ID	Employee_Name

DEPENDENT

| First_Name	Middle_Inital	Last_Name	Employee_ID	Date_of_Birth	Gender

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

Transforming E-R diagrams into relations

- Many-to-many - create a new relation (associative entity) 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
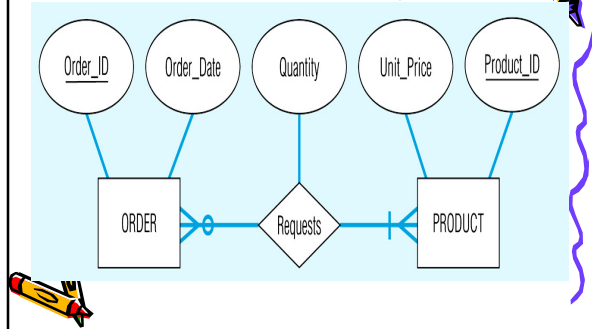
Example of mapping a 1:M relationship
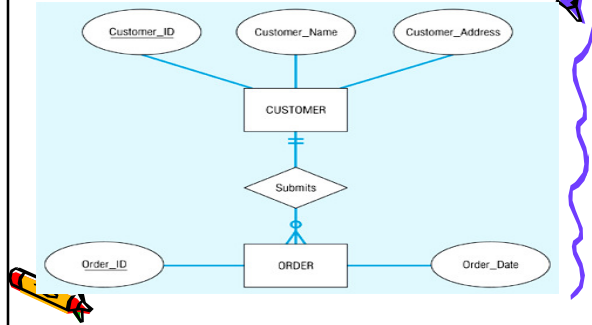
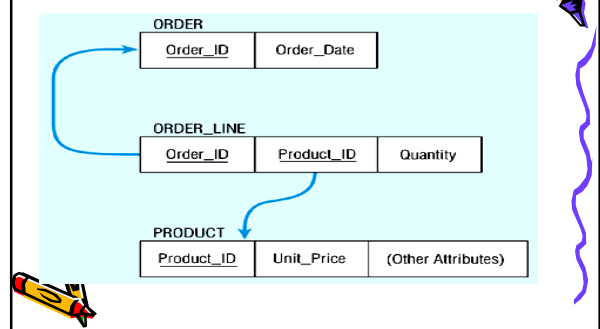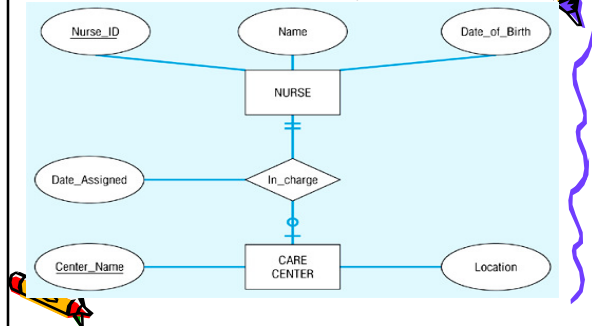
Looks like this using relational schema notation
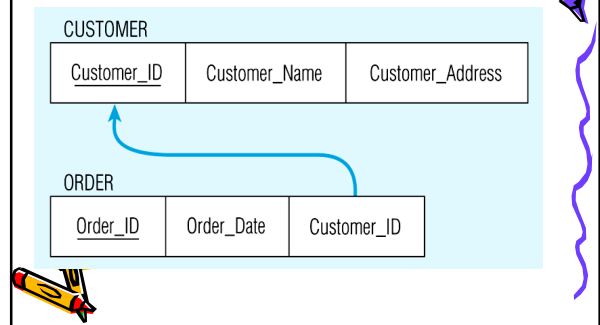
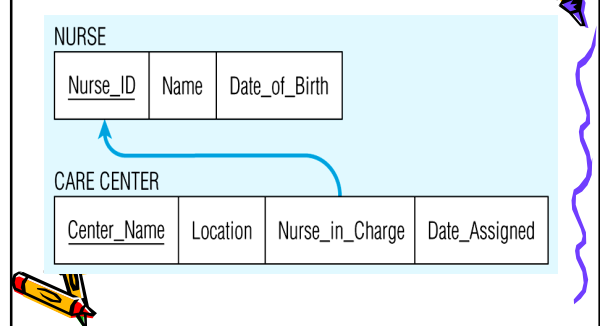## Example of mapping an M:M relationship

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

Transforming E-R diagrams into relations

– 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
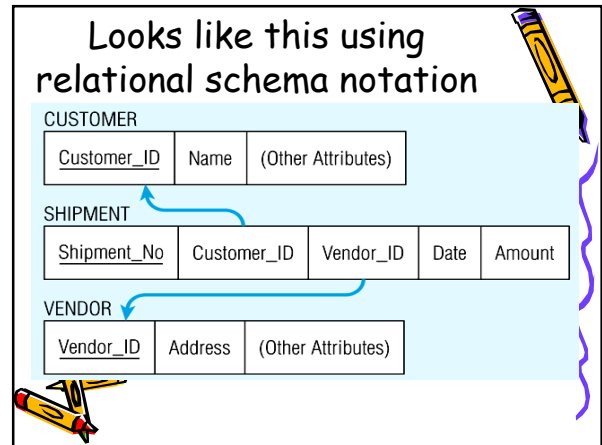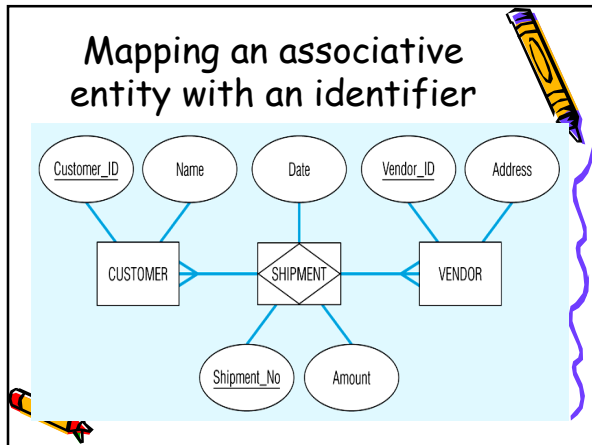
Transforming E-R diagrams into relations

Mapping unary relationships

• One-to-many - A foreign key attribute is added within the same relation that references the primary key values (this foreign key must have the same domain as the primary key)
• A recursive foreign key is a foreign key in a relation that references the primary key values of that same relation

Example

Would look like...

EMPLOYEE

Employee_ID	Name	Birthdate	Manager_ID
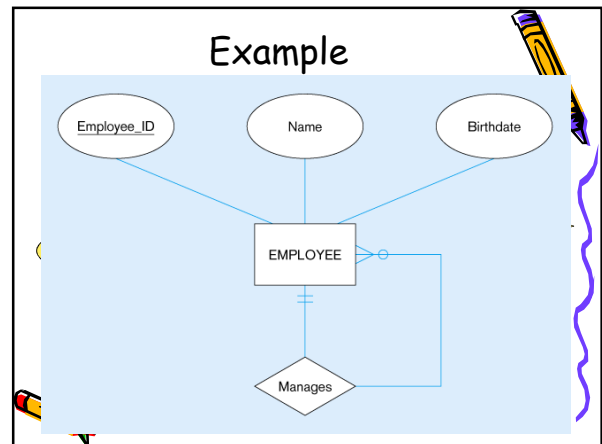
Unary Many-to-many

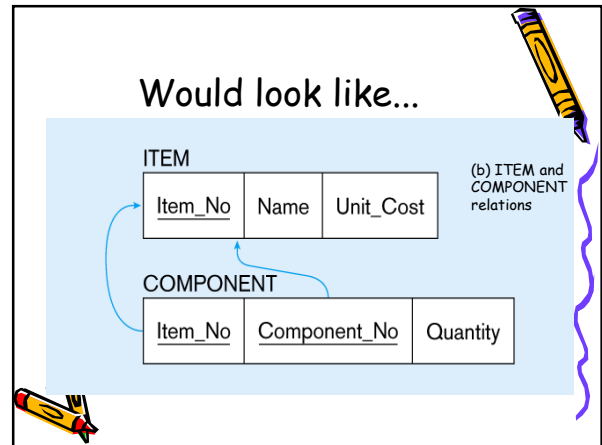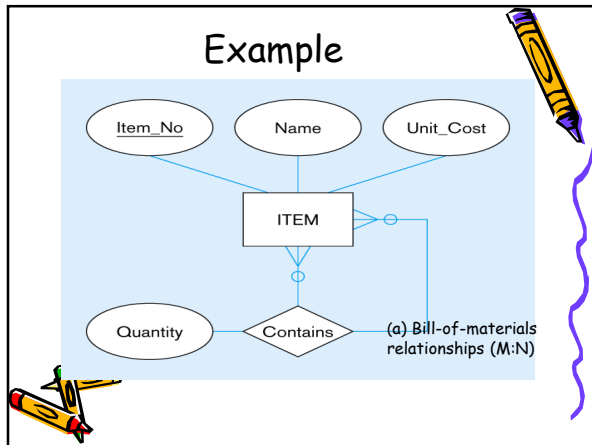- Here two relations are created, one to represent the entity type in the relationship and another representing the M:N relationship itself
- The primary key of the associative relation consists of two attributes, both taking their values from the primary key of the other relation
- Any non-key attribute of the relationship is included in the associative relation

Example

Item_No   Name   Unit_Cost

ITEM

Quantity   Contains

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

Would look like...

ITEM

Item_No	Name	Unit_Cost

COMPONENT

Item_No	Component_No	Quantity

(b) ITEM and COMPONENT relations

Mapping ternary (and n-ary) relationships

- convert a ternary relationship to an associative entity in order to represent participation constraints more accurately.
- Firstly, we create a new associative relation.
- The default primary key of this relation consists of the three primary key attributes for the participating entities

Mapping ternary (and n-ary) relationships

- These attributes then act in the role of foreign keys that reference the individual primary keys of the participating entity types .
- Any attributes of the associative entity type become attributes of the new relation

Mapping a ternary relationship

Looks like this using relational schema notation

Questions