Lebanese University **Faculty of Media and Documentation** **- Section 1 -** **Computer Science** **First Year - Semester 1** **Database 1**		**الجامعة اللبنانية** **كلية الإعلام والتوثيق** **- الفرع الأول -** **Chapter 5: Logical Database Design and the Relational data Model**

Chapter1: Introduction

Chapters2-3: ER

Chapter 4: EER

Chapter 5: Logical Database design and mapping steps.

Chapter 6: Normalisation.

Chapter 7: SQL.

Translate the conceptual design into a logical database design (relational data model) that can be implemented on a chosen DBMS.

Input: ER or EER (conceptual model)

Output: relational schema, normalized relations.

Result: database must meet user need for: data sharing, ease of access and flexibility.

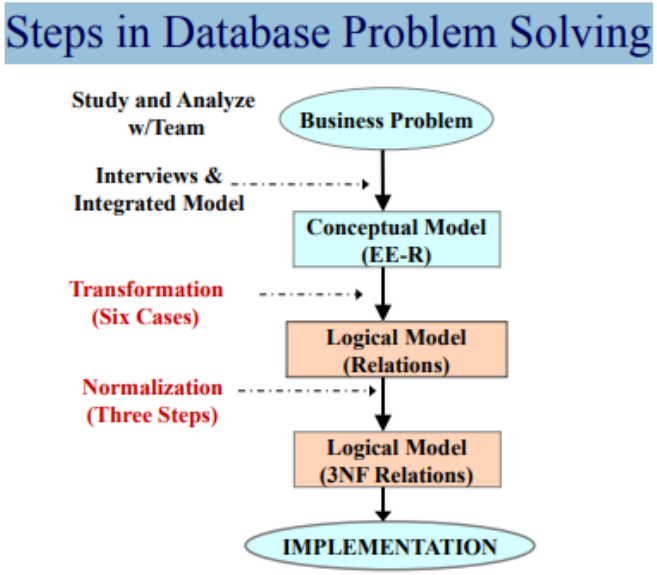
It is a process of converting a high level conceptual data model like ER into a relational data structure.

1. Objectives

- Define terms for the relational data model

- Transform EE-R diagrams to relations
- Create tables with entity and relational integrity constraints

2. Steps in Database Problem Solving:



3. Schemas

Different Schemas are based on different concepts:

- Conceptual schema (high level) schema or ER concepts: Entities, attributes and relationships.
- Logical schema concepts: Tables or relations, columns name and row and tuples.
- Physical schema (Low level): File organization (Files types- Index structures).

Note: relation # relationship

Note 2: **Mapping conceptual models to relational schema is straight-forward**

Straight forward means (Correspondence with ER Model) : Each entity become a table; Each many-to-many relationship become a table; Each attribute become a column in that table and each relationship become a foreign key .

4. Logical Model or Relational data Model advantages:

- Can represent all kinds of information
- Based on Math (relations)
- Natural to people
- Relatively simple
- We know how to implement it fast

5. Components of Relational data model:

- Data structure : Tables (relations), rows, columns (Data organized into tables)
- Data manipulation : Powerful SQL operations for retrieving and modifying data (add, delete,modify and retrieve data using SQL)
- Data integrity : Mechanisms for implementing business rules that maintain integrity of manipulated data (maintained using business rules)

Recall.

1. Formal Definition Schema

- **The Schema (or description) of a Relation**
  - Denoted by **$R(A_1, A_2, ..., A_n)$**
  - **R** is the name of the relation
  - The **attributes** of the relation are **$A_1, A_2, ..., A_n$**

Example:

Customer (Cust_id, Cust_name, Address)

- **Customer** is the relation name
- Defined over the 3 attributes:
  - **Cust_id**, **Cust_name**, **Address**

- **Each Attribute has a domain or a set of valid values** *(مثلاً مجموعة القيم المسموح بها)*
  For example, the domain of **Cust_id** is **number**

- **The domain is defined by a type**
  - A type is the class of atomic values:
    - integers, reals, strings
    - integers between 15 and 80
    - string of (up to) 50 characters.

2. Formalizing: Relations

- Definition: A relation is a named table of data (Table is made up of rows (records or tuples), and columns (attributes or fields))
- Requirements for a table to be a relation:
- 1. Has a unique name.
- 2. Every attribute value is atomic (not multivalued or composite)

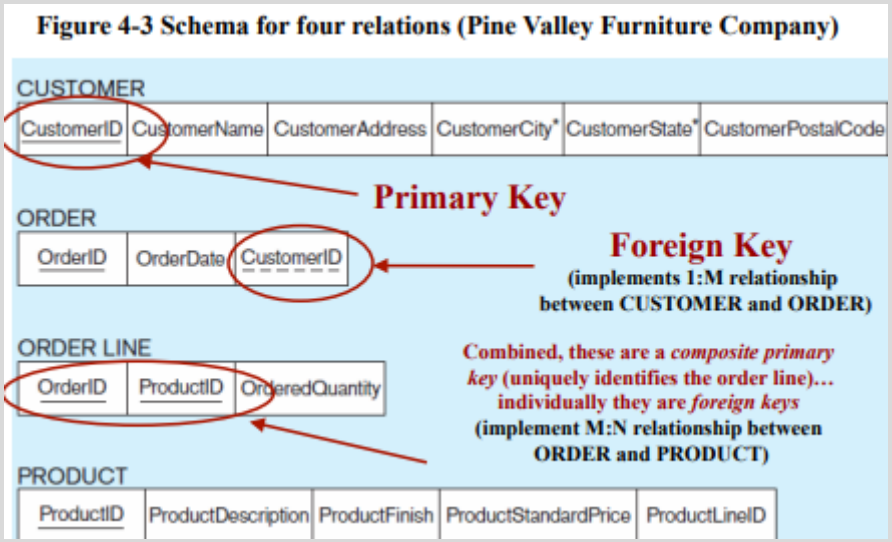
- 3. Every row is unique
- 4. Attributes (columns) in tables have unique names
- 5. The order of the columns is irrelevant
- 6. The order of the rows is irrelevant

By definition, all relations are in 1st Normal Form (1NF)

3. Keys :Formalizing Key Fields

- **Primary key (PK) –** Minimal set of attributes that uniquely identifies a row, chosen for referencing – This is how we can guarantee that all rows are unique
- **Foreign key (FK) –** Set of attributes in a table that serves as a reference to the primary key of another table
- **Keys can be simple or composite**
- **Used as indexes to speed up queries**

Figure 4-3 Schema for four relations (Pine Valley Furniture Company)



4. Constraints

A. Domain constraints

Allowable values for an attribute as defined in the domain

B. Entity integrity constraints

No primary key attribute may be null

C. Operational constraints

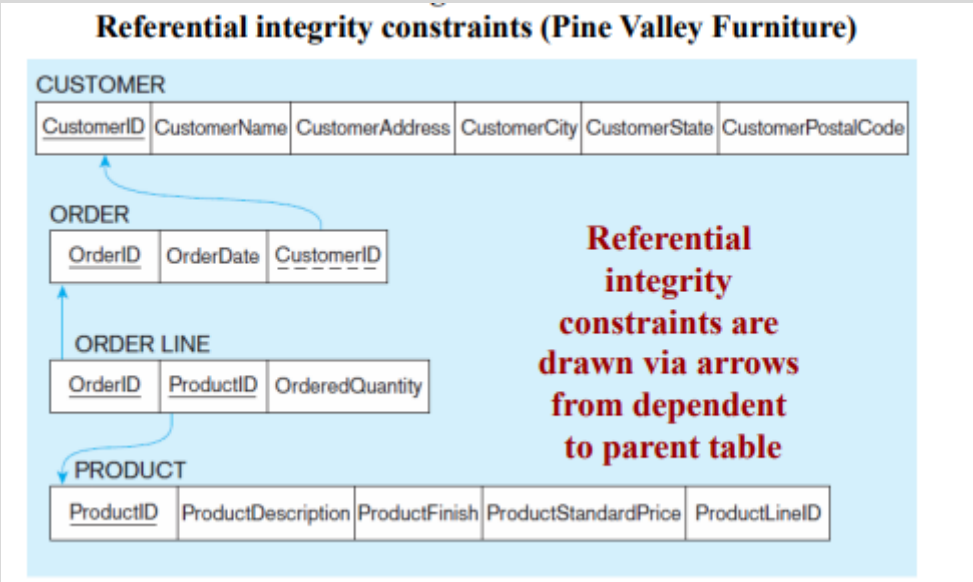
Business rules

D. Referential integrity constraints

- **Maintains consistency among rows of two entities**

  matching of primary and foreign keys.

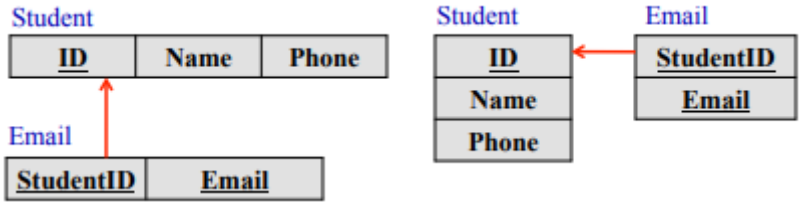
  For a FK, either all attributes are null, or the values appear in the PK of a row of the referred table.



- **Enforcement options for deleting instances**
  - **Restrict:** don't allow delete of "parent" side if related rows exist in "dependent" side.
  - **Cascade:** This ensures that whenever a record from the parent table of a referential integrity constraint is deleted, all dependent records in the child table that have matching primary key values in their foreign key are deleted as well (Automatically delete "dependent" side rows that correspond with the parent side row deleted set).
  - **Set-to-Null:** This ensures that whenever a record from the parent table of a referential integrity constraint is deleted, all the dependent records in the child table that have matching primary key values in their foreign key are set to NULL → not allowed for weak entities.0

6. From ER Diagrams To Relations (Tables).

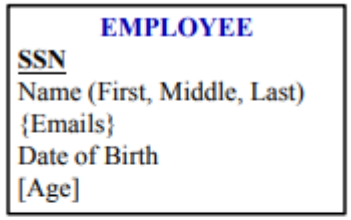
- Use a rectangle for each entity (table), with attributes inside rectangles, too
  - – Can be vertical or horizontal
  - – Primary key is underlined
- Use arrows from Foreign key to Primary key



- Entities are represented by tables
  - – But tables may also represent relationships, or multivalued attributes
- Foreign Keys used to relate table rows
  - – Similar to relationships in E-R, but lower level
- Relational model is more concrete, lower level
  - – Usually many more tables than entities – Harder to understand by non-technical people – Directly implementable

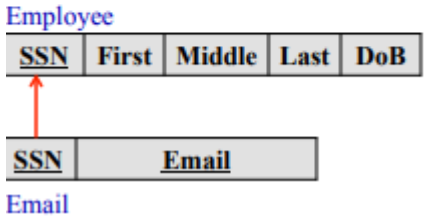
7. Six Cases of Transforming E-R Diagrams into Relations (Mapping

7.1. Map Binary Relationships



Create a new table for each entity

- Remember to underline the identifier
- For composite attributes, map only the basic pieces
- Derived attributes disappear
- For multivalued attributes we need a new table

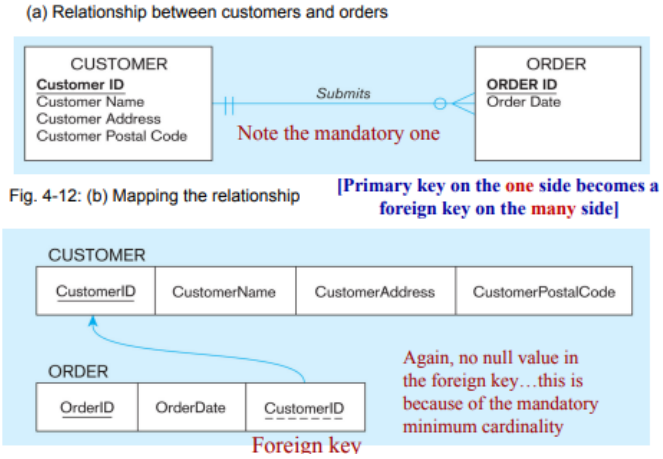


- We may need to create several tables for independent multivalued attributes

7.2. Map Binary Relationships

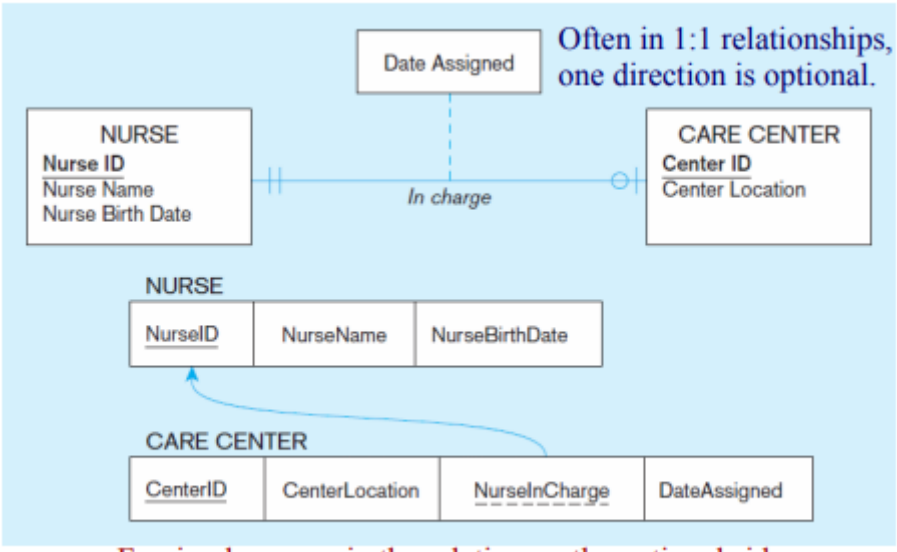
- One-to-Many

– Primary key on the one side becomes a foreign key on the many side



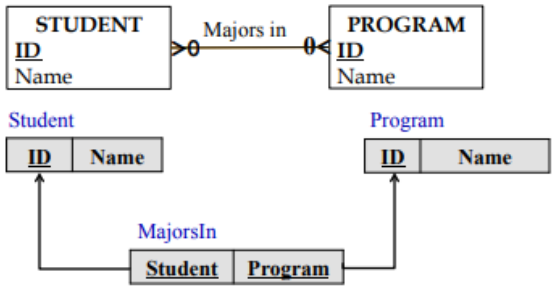
• One-to-One

– Primary key on the mandatory side becomes a foreign key on the optional



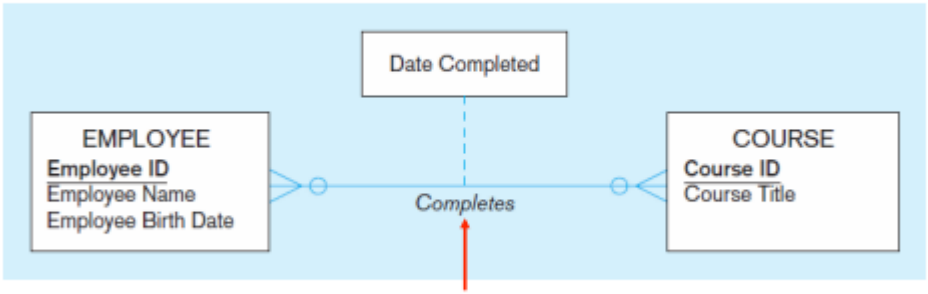
• Many-to-Many

– Create a new relation with the primary keys of the two entities as its primary key



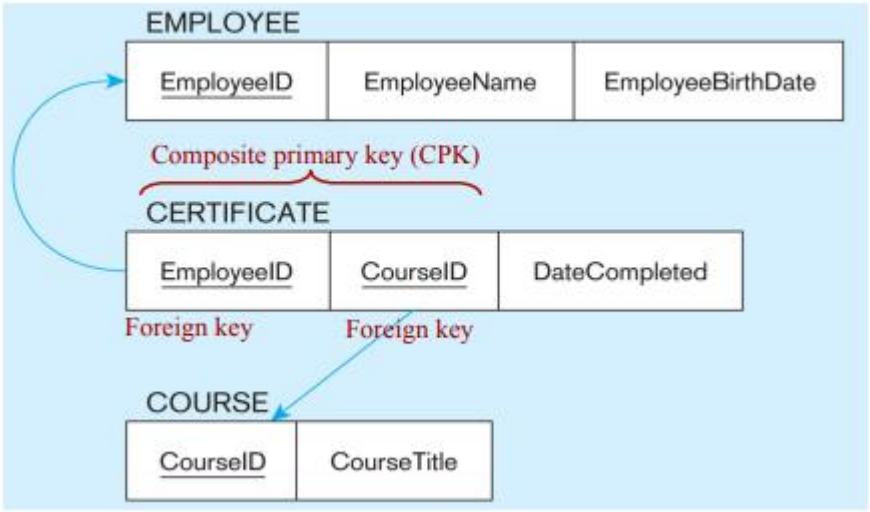
- For a many-to-many, we need a new table representing the relationship.
- This table has Foreign Keys to both entities.

a) Completes relationship (M:N)



The *Completes* relationship will need to become a separate relation

b) Three resulting relations

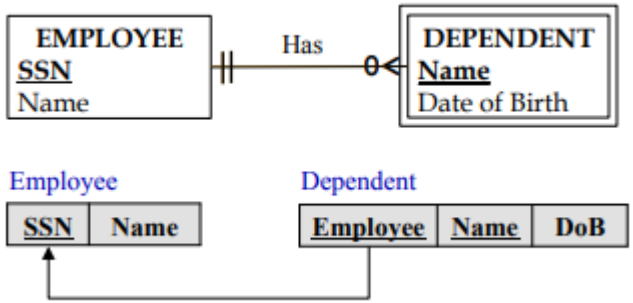


7.3. Map Weak Entities

- A weak entity becomes a separate relation with a foreign key taken from the strong

entity

- Primary key composed of:
  - Partial identifier of weak entity
  - Primary key of identifying relation (strong entity)



- Transform the strong entity normally
- For the weak entity, the PK becomes the identifier, plus the PK of the identifying entity

7.4. Map 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 4-15: Mapping an associative entity

(a) Associative entity (ORDER LINE)

[Default primary key for the association relation is NOT assigned]

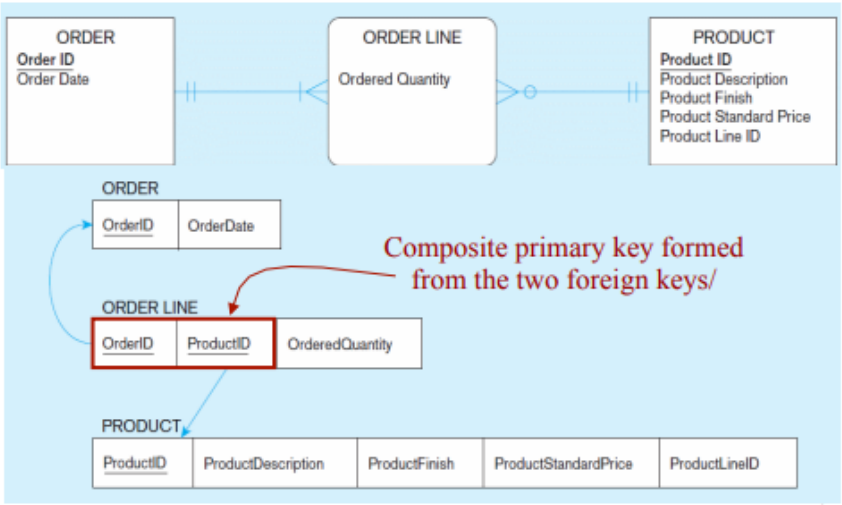
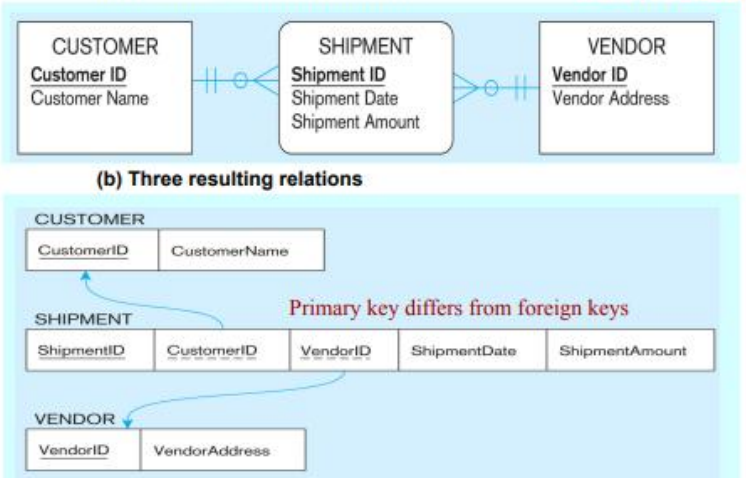


Figure 4-16: Mapping an associative entity

(a) Associative entity (SHIPMENT)

[Default primary key for the association relation is assigned]

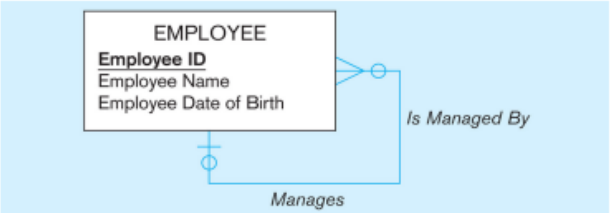


7.5. Map Unary Relationships

- Same as other relationships, except that the FK may go to the same table.
- For one-to-many, the table has a reference to other rows of the same table.
- For many-to-many, an extra table has two FKs, both to the same table

Figure 4-17 Mapping a unary 1:N relationship

(a) EMPLOYEE entity with unary relationship



A recursive FK is a FK in a relation that references the PK values of that same relation.

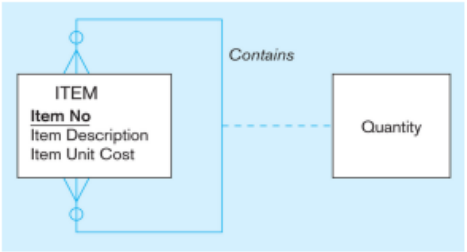
(b) EMPLOYEE relation with recursive foreign key



53

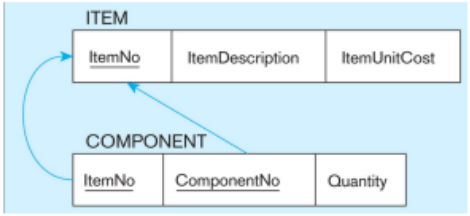
Figure 4-18: Mapping a unary M:N relationship

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



One table for the entity type.

(b) ITEM and COMPONENT relations



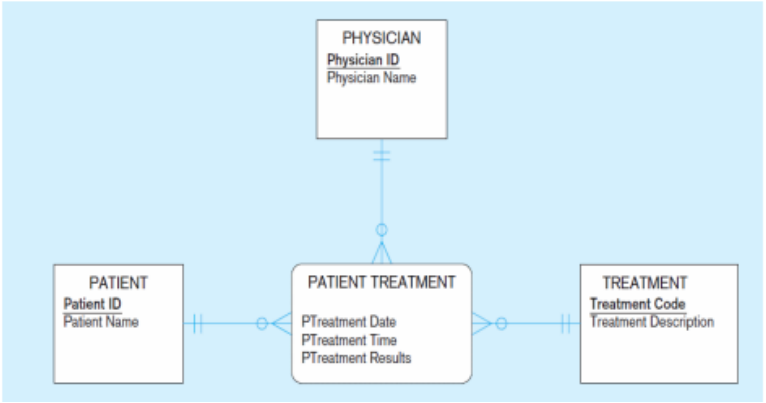
One table for an associative relation in which the primary key has two attributes, both taken from the primary key of the entity.

54

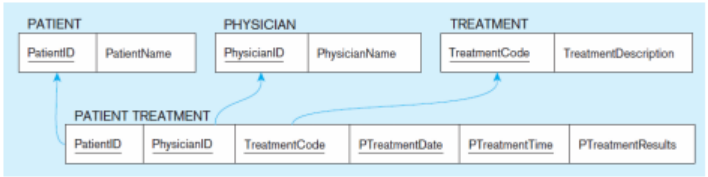
7.6. Map Ternary (and n-ary) Relationships

- One relation for each entity and one for the associative entity.
- Associative entity has foreign keys to each entity in the relationship

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#.

(A patient may receive a treatment once in the morning, then the same treatment in the afternoon.)

62

8. EE-R to 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

Figure 4-20 Supertype/subtype relationship

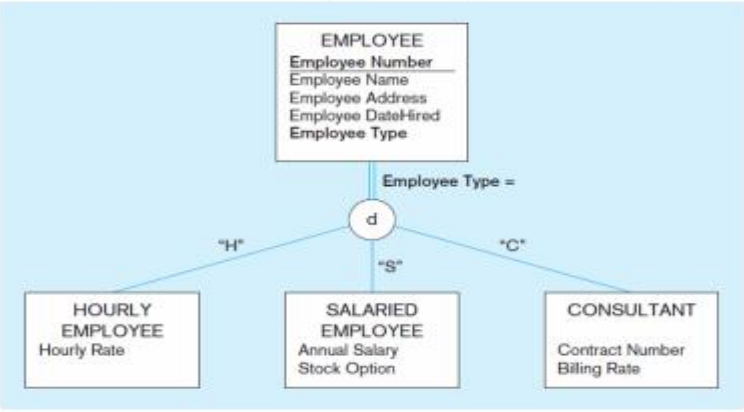
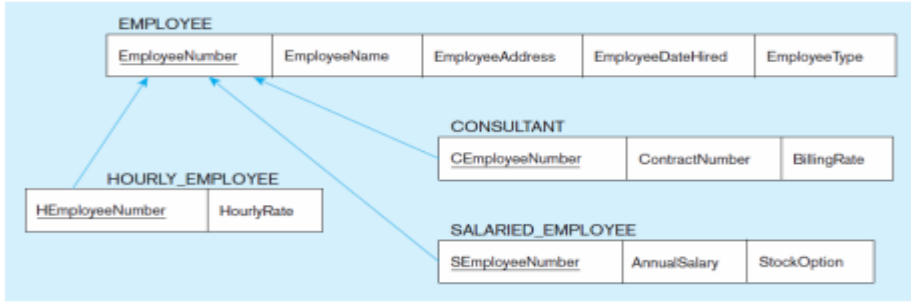


Figure 4-21
Mapping supertype/subtype relationships to relations



These are implemented as one-to-one relationships.

Next Chapter: Normalization