Databases I

Etelka Szendrői Dr. (PhD)
associate professor
System and Software Technology Department

szendroi@mik.pte.hu

The Extended Entity Relationship (EER) Model

- Actually, what we will be discussing is an extension of Peter Chen’s proposal (hence “extended” ER).
- These represent logical links between an entity E, known as **parent** entity, and one or more entities E1,...,En called **child** entities, of which E is more general, in the sense that they are a particular case.
- In this situation we say that E is a **generalization** of E1,...,En and that the entities E1,...,En are **specializations** of E.

2

Properties of Generalization

- Every instance of a child entity is also an instance of the parent entity.
- Every property of the parent entity (attribute, identifier, relationship or other generalization) is also a property of a child entity. This property of generalizations is known as **inheritance**.

3

Types of Generalizations

- A generalization is **total** if every instance of the parent entity is also an instance of one of its children, otherwise it is **partial**.
- A generalization is **exclusive** if every instance of the parent entity is at most an instance of one of the children, otherwise it is **overlapping**.
- The generalization Person, of Man and Woman is total (the sets of men and the women constitute ‘all’ the people) and exclusive (a person is either a man or a woman).
- The generalization Vehicle of Automobile and Bicycle is partial and exclusive, because there are other types of vehicle (for example, motor bike) that are neither cars nor bicycle.
- The generalization Person of Student and Employee is partial and overlapping, because there are students who are also employed.

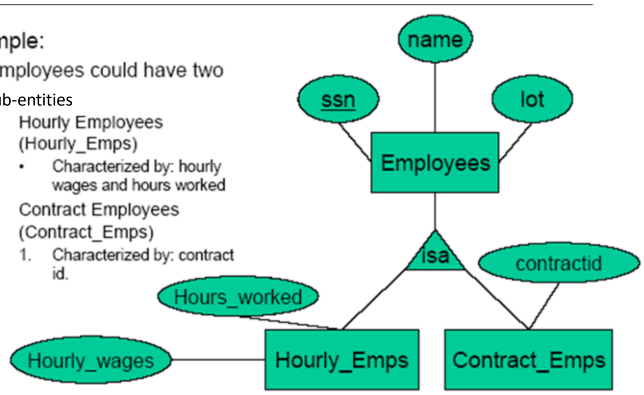
4

Generalization Hierarchies

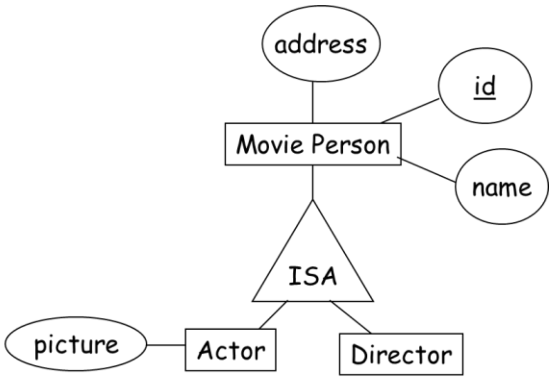
- Total generalization (i.e., every instance of the super entity is an instance of some sub-entity) is represented by a solid arrow.
- One arrow with multiple sub-entities (e.g., arrow from Woman/Man to People) means that sub-entities are mutually exclusive.
- In most applications, modeling the domain involves a hierarchy of generalizations that includes several levels

The EER Model, as an EER Diagram

- Example:
  - Employees could have two Sub-entities
    1. Hourly Employees (Hourly_Emps)
       - Characterized by: hourly wages and hours worked
    2. Contract Employees (Contract_Emps)
       1. Characterized by: contract id.



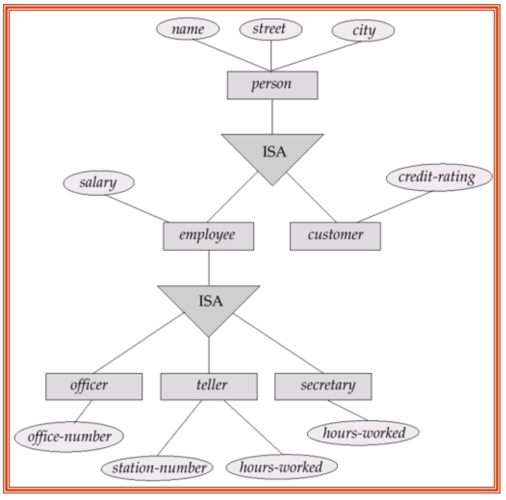
Another Example



Specialization

- Top-down design process; we designate subgroupings within an entity set that are distinctive from other entities in the set.
- These subgroupings become lower-level entity sets that have attributes or participate in relationships that do not apply to the higher-level entity set.
- Depicted by a *triangle* component labeled ISA (E.g. *customer* "is a" *person*).
- **Attribute inheritance** – a lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked.

Specialization Example



Database Model Levels

- A Conceptual model represents reality in an abstracted form that can be used in developing an information system in a wide variety of formats (e.g. relational, object-oriented, flat-file, etc.)
  - It is hardware and software independent
  - It is independent of any logical model type
- A Logical model represents reality in the format required by a particular database model (e.g. relational or object-oriented)
  - Is still hardware and software independent
  - Depends on the chosen logical model type
- A Physical model is created specifically for a particular database software package
  - Is dependent on hardware, software, and on the chosen logical model type

Relational Database Model

- A relational database consists of tables (relations)
- Relational database tables are made up of rows and columns
  - Rows are called the table extension or tuples
    - The ordering of rows in a table does not matter
  - Columns are called the table intension or schema
    - The ordering of columns in a table does not matter
    - All values in a column must conform to the same data format (e.g. date, text, currency, etc.)
  - Each cell in a database table (a row-column intersection) can contain only one value
    - no repeating groups are allowed

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

14

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.

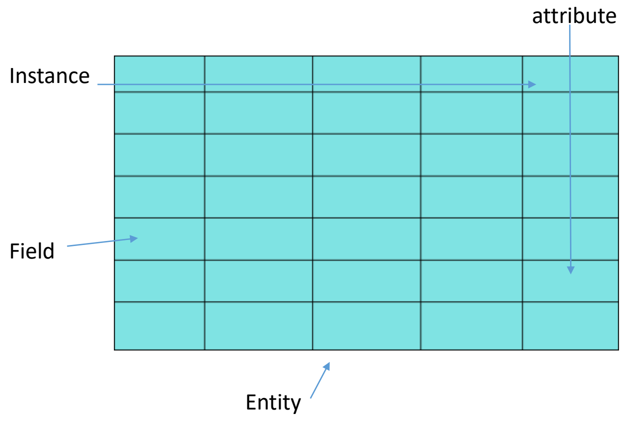
15

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 key is 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 attributes
  - Weak entities
  - Associative entities

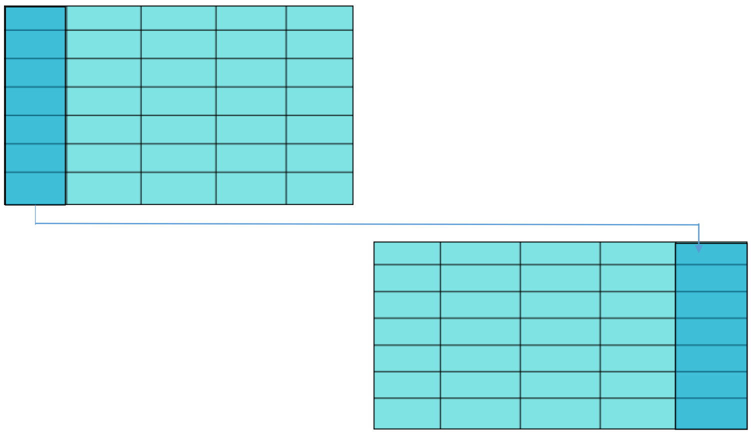
16

Implementing it



17

What about relationships?



Foreign Key Example

SaleID	Date	Amount	Salesperson
061401A	6/14	$4,218	123456
061401B	6/14	$6,437	654321
061501A	6/15	$1,112	654321

SalespersonID	Name
123456	Fred
654321	Francis

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

20

Referential integrity constraint

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

21

Transforming the ER and EER diagram into relations

The steps:
- Map regular entities
- Map weak entities
- Map binary relationships
- Map associative entities
- Map unary relationships
- Map ternary relationships
- Map supertype/subtype relationships

22

Transforming E-R diagrams into relations

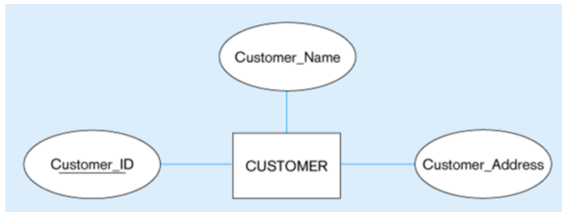
Mapping Regular Entities to Relations

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

23

Mapping a regular entity

(a) CUSTOMER entity type with simple attributes



(b) CUSTOMER relation

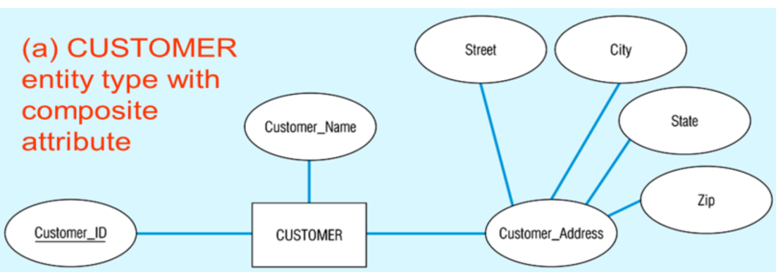
CUSTOMER

Customer_ID	Customer_Name	Customer_Address

24

Mapping a composite attribute

(a) CUSTOMER entity type with composite attribute

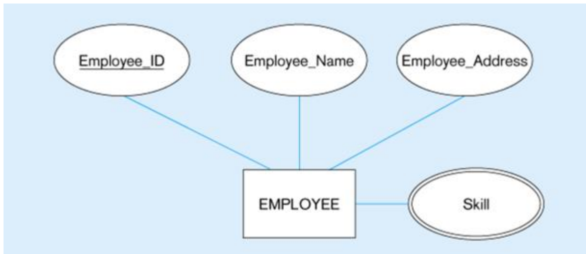


CUSTOMER (b) CUSTOMER relation with address detail

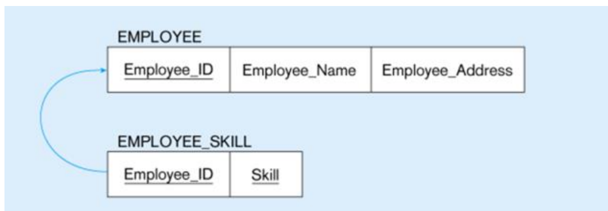
Customer_ID	Customer_Name	Street	City	State	Zip

25

Mapping a multivalued attribute



Multivalued attribute becomes a separate relation with foreign key



1 – to – many relationship between original entity and new relation

26

Transforming E-R diagrams into relations

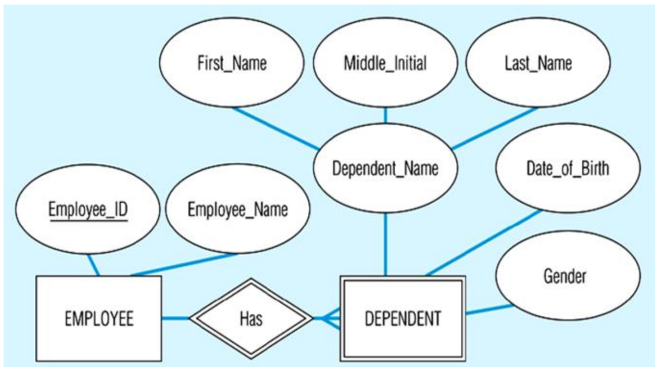
Mapping Weak Entities

- Becomes a separate relation with a foreign key taken from the superior entity
- Primary key composed of:
  - Partial identifier of weak entity
  - Primary key of identifying relation (strong entity)

27

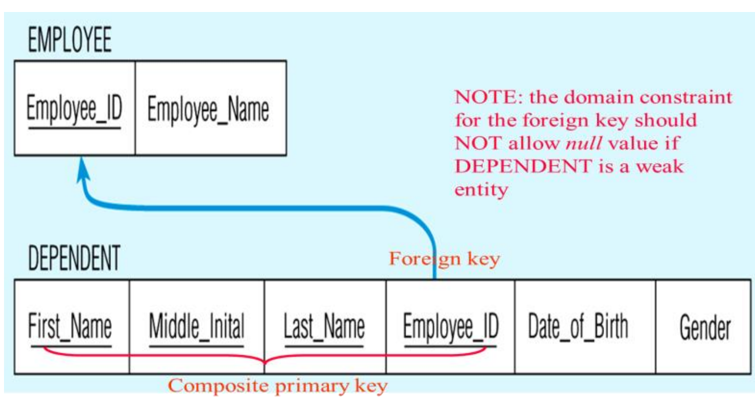
Example of mapping a weak entity

(a) Weak entity DEPENDENT



28

Relations resulting from weak entity



29

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

30

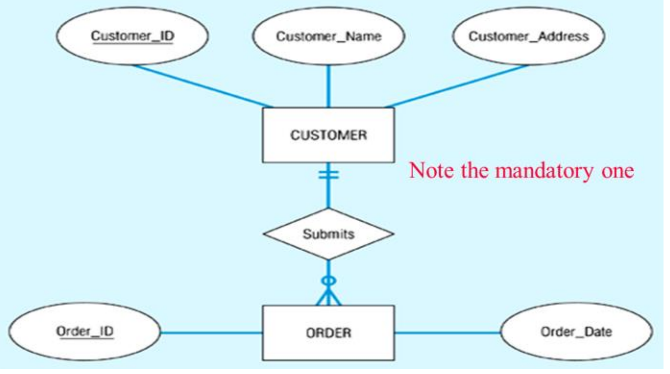
NULL Values in Foreign Keys

- Whether or not a Foreign Key can have NULL values depends on the minimum cardinality of the concerned relationship
- Minimum cardinality of 0 represented as NULL allowed for foreign key columns
- Minimum cardinality of 1 represented as NULL disallowed for foreign key columns

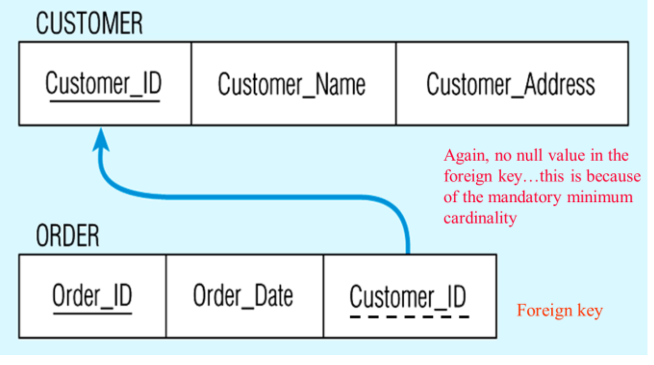
31

Example of mapping a 1:M relationship

(a) Relationship between customers and orders

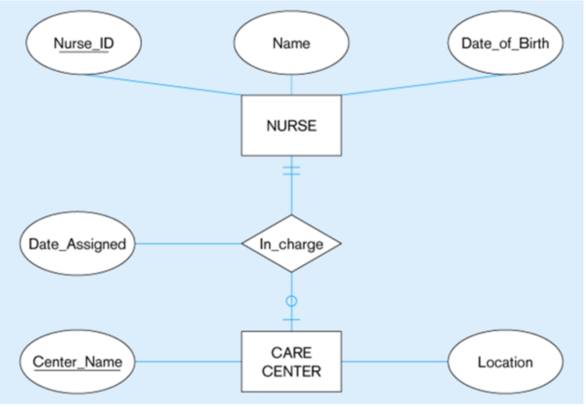


Mapping the relationship

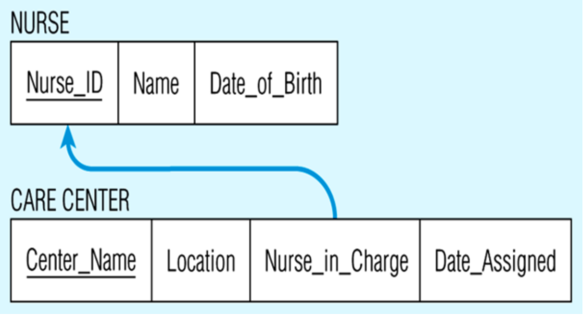


Mapping a binary 1:1 relationship

(a) Binary 1:1 relationship

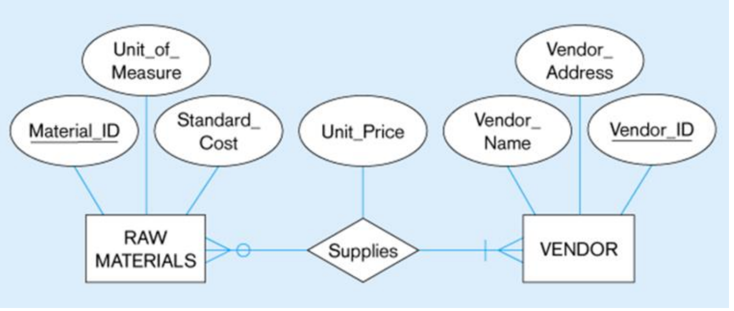


(b) Resulting relations



Example of mapping an M:N relationship

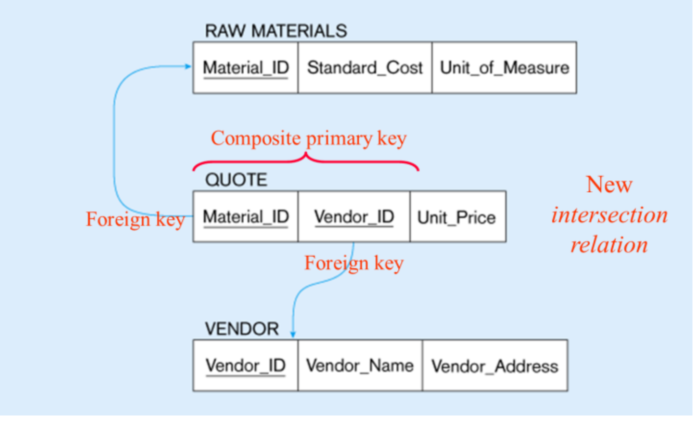
(a) ER diagram (M:N)



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

36

Three resulting relations



37

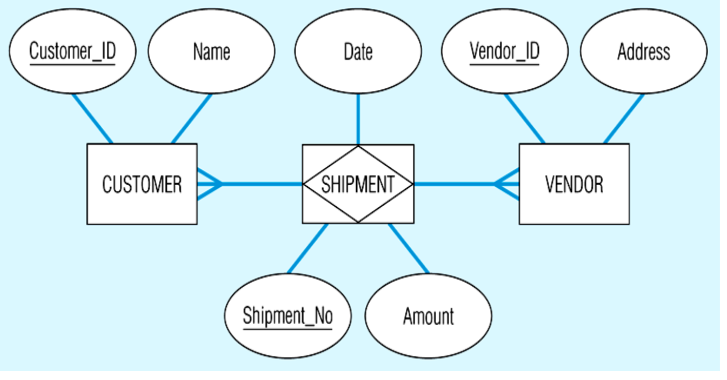
Transforming ER Diagrams into Relations

Mapping Associative Entities

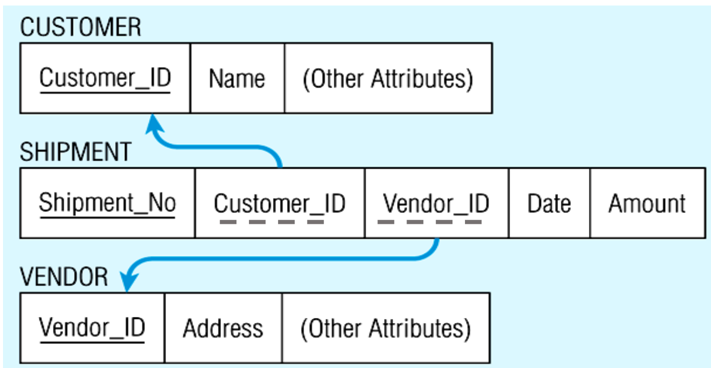
- Identifier Not Assigned
  - Default primary key for the association relation is composed of the primary keys of the two entities (as in M:N relationship)
- Identifier Assigned
  - It is natural and familiar to end-users
  - Default identifier may not be unique

Mapping an associative entity

(a) Associative entity



(b) Three resulting relations



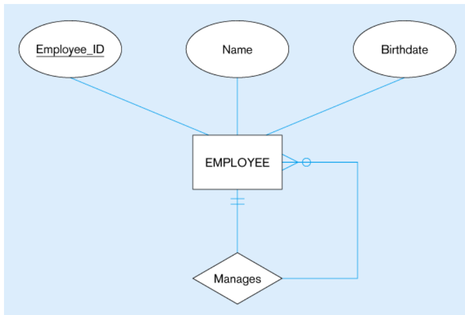
Transforming ER Diagrams into Relations

Mapping Unary Relationships

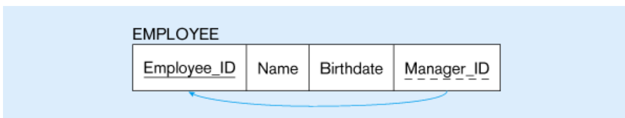
- One-to-Many - Recursive foreign key in the same relation
- Many-to-Many - Two relations:
  - One for the entity type
  - One for an associative relation in which the primary key has two attributes, both taken from the primary key of the entity

Mapping a unary 1:N relationship

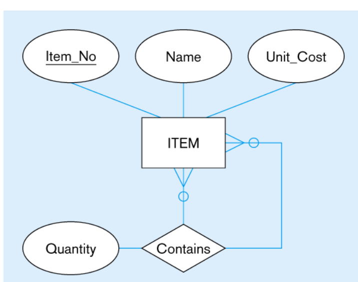
(a) EMPLOYEE entity with Manages relationship



(b) EMPLOYEE relation with recursive foreign key

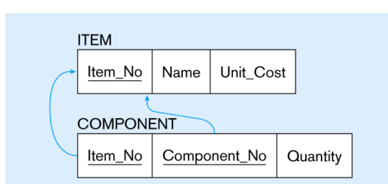


Mapping a unary M:N relationship



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

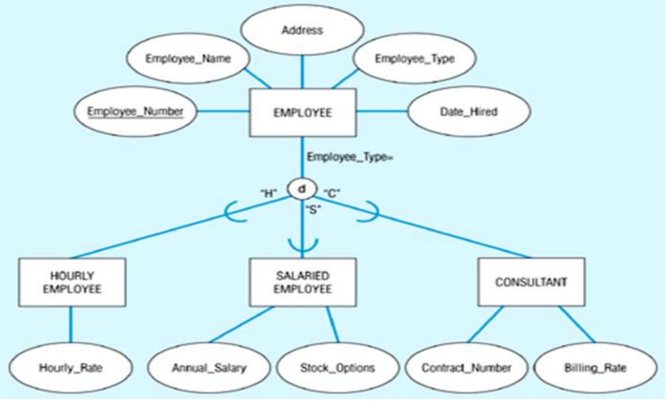
(b) ITEM and COMPONENT relations



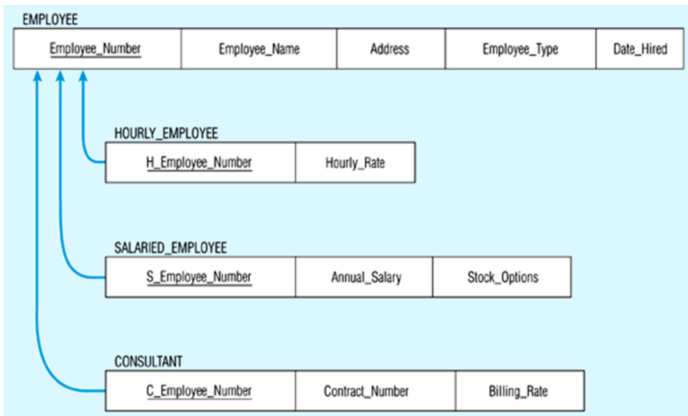
Transforming EER diagrams into relations

- Mapping Supertype/subtype relationships
- Create a separate relation for the supertype and each of the subtypes
- Assign common attributes to supertype
- Assign primary key and unique attributes to each subtype
- Assign an attribute of the supertype to act as subtype discriminator

Mapping Supertype/subtype relationships



Would look like this...



46

Thank you for your attention!