Introduction to Databases (Postgres)

Zeyad Ashraf

Agenda

01 Introduction

02 File system

03 Advantages & disadvantages

04 DB & DBMS

05 SQL vs NOSQL

06 DBLS

07 ERD

08 Mapping

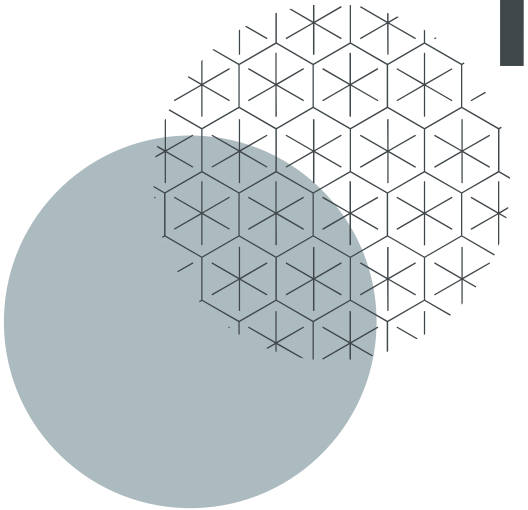
09 Why Postgres

10 History of Postgress

11 SQL

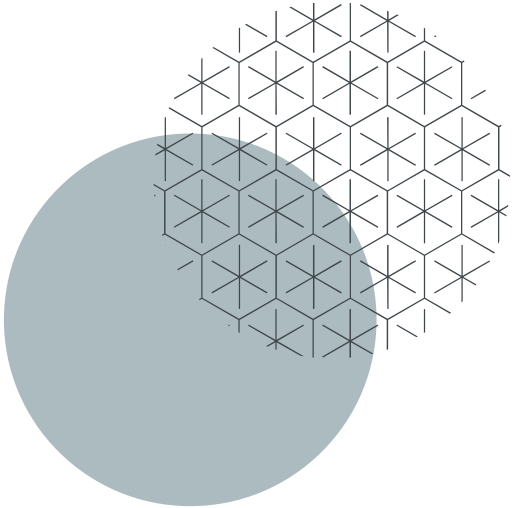
01

Introduction



02

File system



File Based System

- Separation & Isolation Of data (each user has a copy) cause inconsistencies
- Incompatible File Formats
- Program–Data Dependence
  - All programs maintain metadata for each file they use
  - Each application program needs to include code for the metadata of each file
  - Non–standard file formats
- Lengthy Development Times
  - Programmers must design their own file formats (Metadata)
- Data Redundancy (Duplication of data)
  - Different systems/programs have separate copies of the same data
  - When data changes in one file, could cause inconsistencies
  - No Database integrity
- Limited Data Sharing
  - No centralized control of data

What is a Relational Database?

A data structure through which data is stored in tables that are related to one another in some way.

The way the tables are related is described through a relationship.

03

Advantages & disadvantages

DBMS Advantages

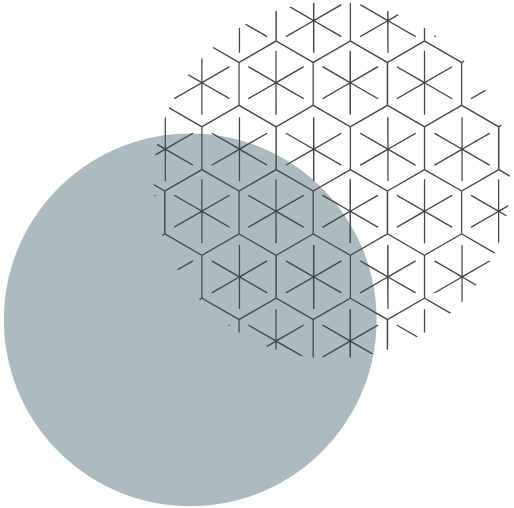
- Standardization and better Data accessibility and response (SQL)
- Sharing data.
- Enforcing Integrity Constraints
- Improved Data Quality
- Inconsistency can be avoided because of data sharing.
- Restricting Unauthorized Access.
- Providing Backup and Recovery.
- Minimal Data Redundancy
- Program–Data Independence

DBMS Disadvantages

- It needs expertise to use
- DBMS itself is expensive
- The DBMS may be incompatible with any other available DBMS

04

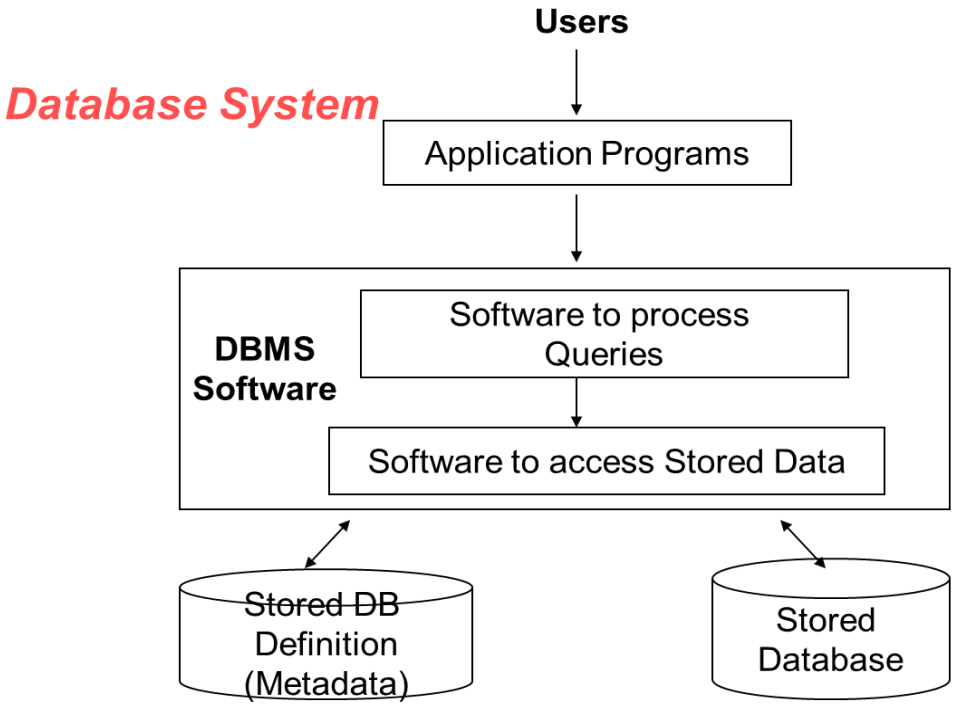
DB & DBMS



Basic Definitions

Database	A collection of related data.
DBMS	A software package/ system to facilitate the creation and maintenance of a computerized database
Database system	The DBMS together with data itself. (Software + Database)

Database System



05

SQL vs NOSQL

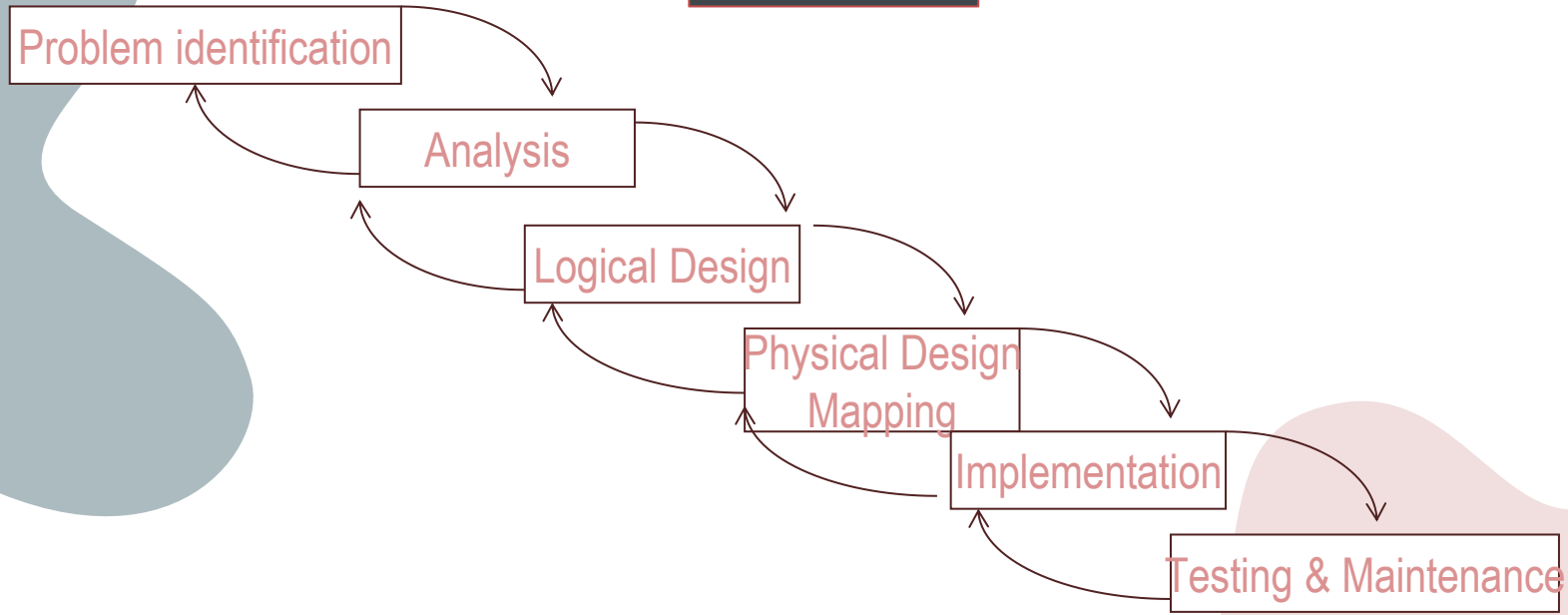
SQL vs NoSQL

	SQL (Relational DB)	NoSQL (Non-Relational DB)
Structure	Tables with rows and columns	Documents (JSON), Key–Value, Column, Graph
Schema	Fixed schema	Flexible schema – dynamic fields
Relationships	Strong support via Foreign Keys	not relational (or limited relationship support)
Read/Write Performance	Optimized for complex queries & structured data	Optimized for high–volume, unstructured, or semi–structured data
Examples	MySQL, PostgreSQL, Oracle, SQL Server	MongoDB, Cassandra, Redis, CouchDB, Neo4j

06

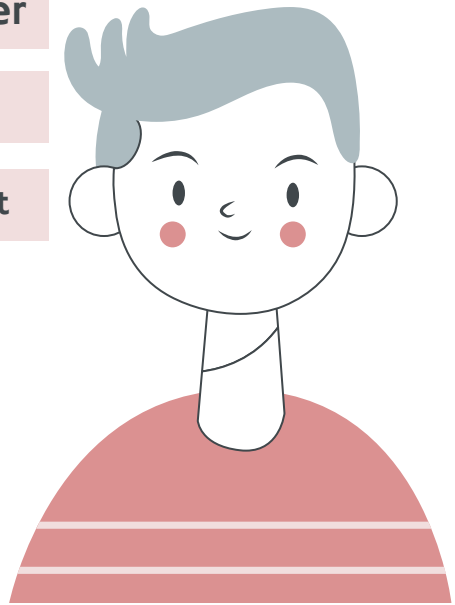
DBLS

DBLS


Problem identification
Analysis
Logical Design
Physical Design
Mapping
Implementation
Testing & Maintenance

Database Users

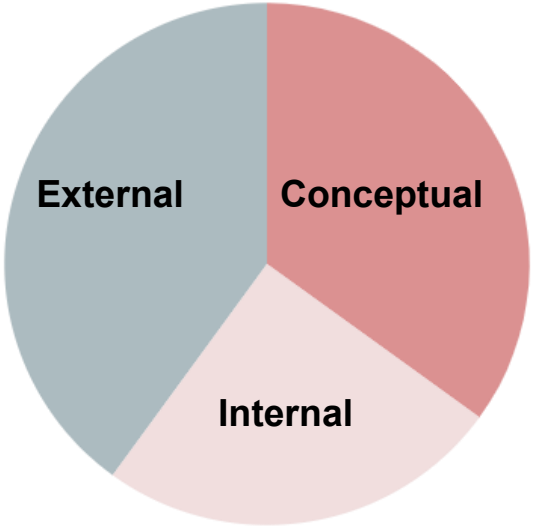
- Database Administrator (DBA)
- System Analysts
- Database Designer
- Database Developer
- App programmers
- BI & BigData Specialist
- End users



Three Level/Schema Architecture

What the user sees

how the data will be presented to the user



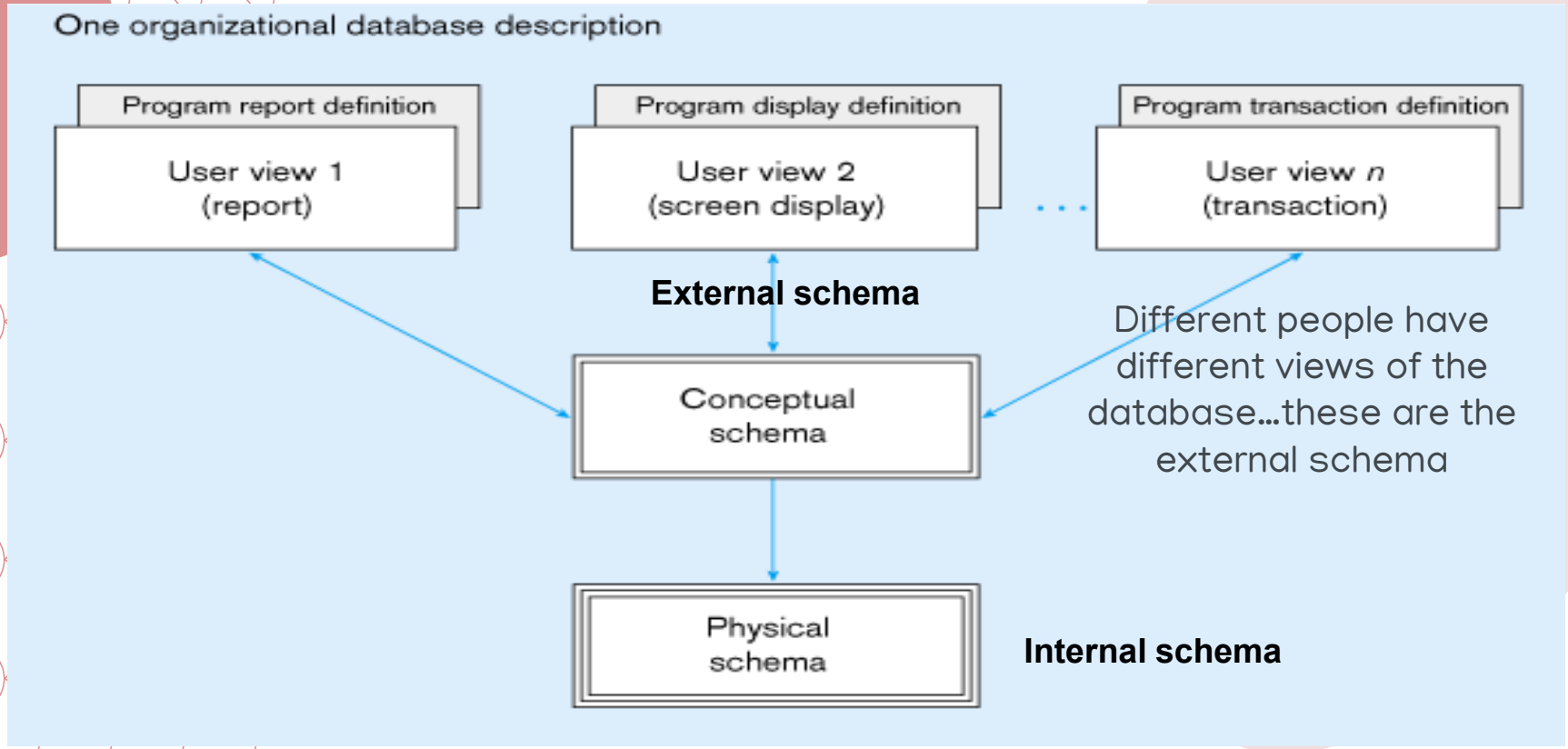
The logical model

what is represented rather than how it is represented

The physical model

how the data are represented

Three Level/Schema Architecture



07

Entity Relationship Diagram Concepts

Entity-Relationship Diagram (ERD)

Identifies information required by the business by displaying the relevant entities and the relationships between them.

The ER Model

Entities

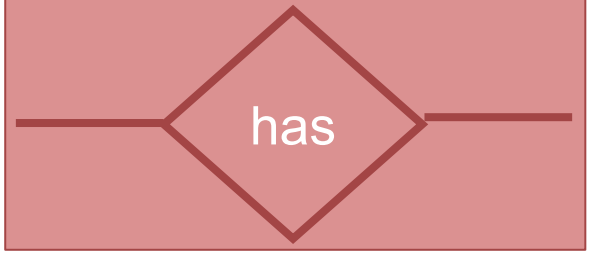
A– person, place, object, event, concept (often corresponds to a real time object that is distinguishable from any other object)

Attributes

Aproperty or characteristic of an entity type (often corresponds to a field in a table)

Relationships

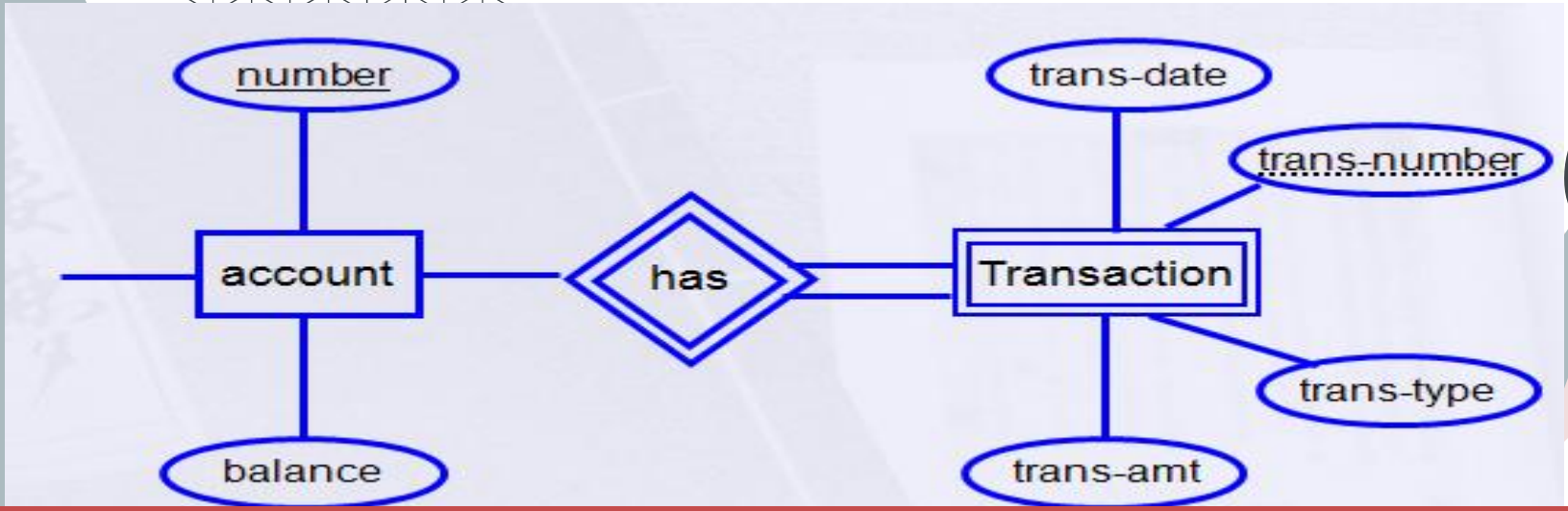
link between entities (corresponds to primary key–foreign key equivalencies in related tables)



Strong Entity Vs Weak Entity

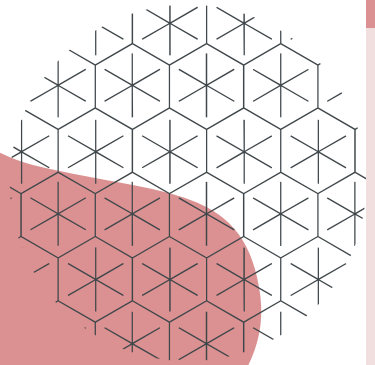
A Strong Entity- An Entity set that has a primary key.

A Weak Entity- An entity set that do not have sufficient attributes to form a primary key.



Partial key: A set of attributes that can be associated with P.K of an owner entity set to distinguish a weak entity.

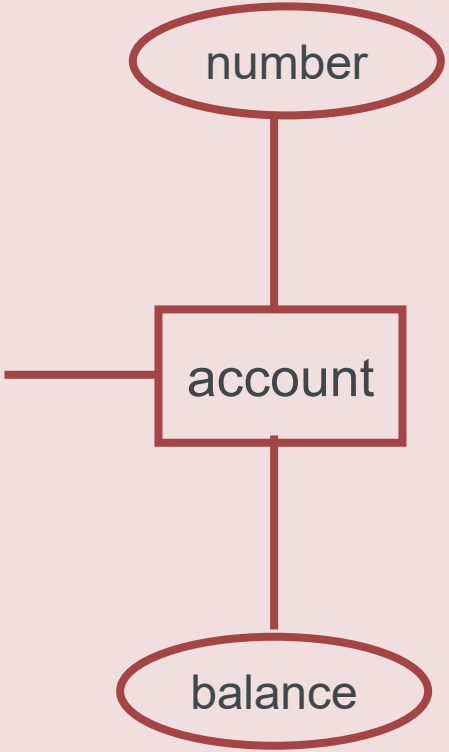
Attributes



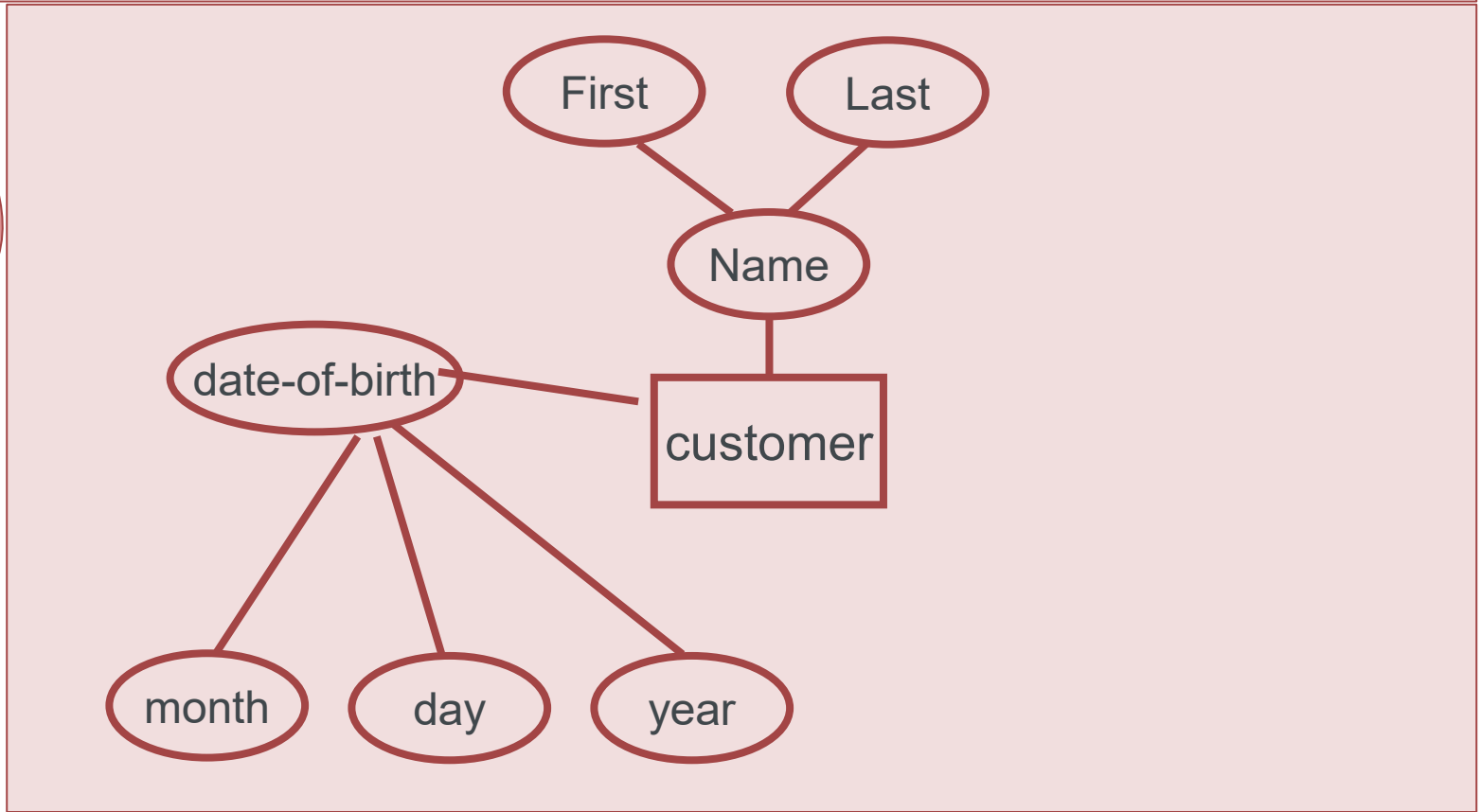
Types of Attributes

- Composite Attribute
- Multi-valued Attribute
- Derived Attribute
- Complex Attribute
- Simple Attribute

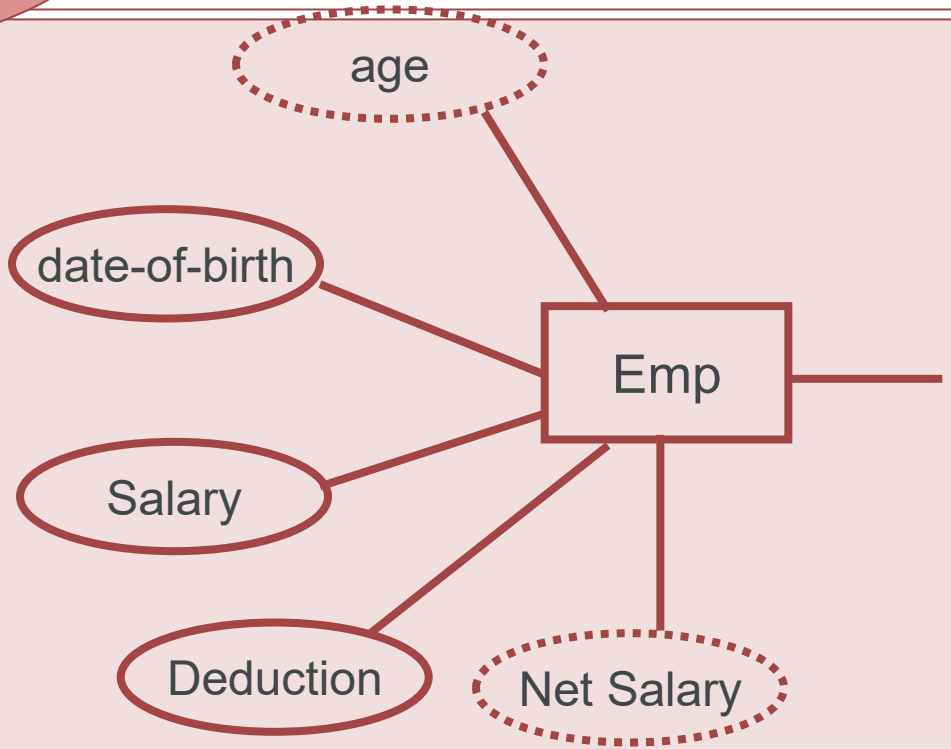
Simple Attribute



Composite Attribute

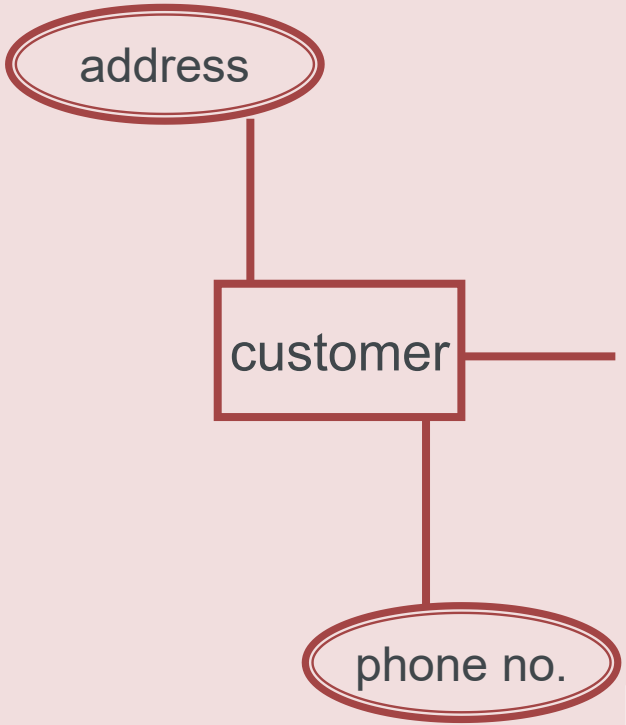


Derived Attribute



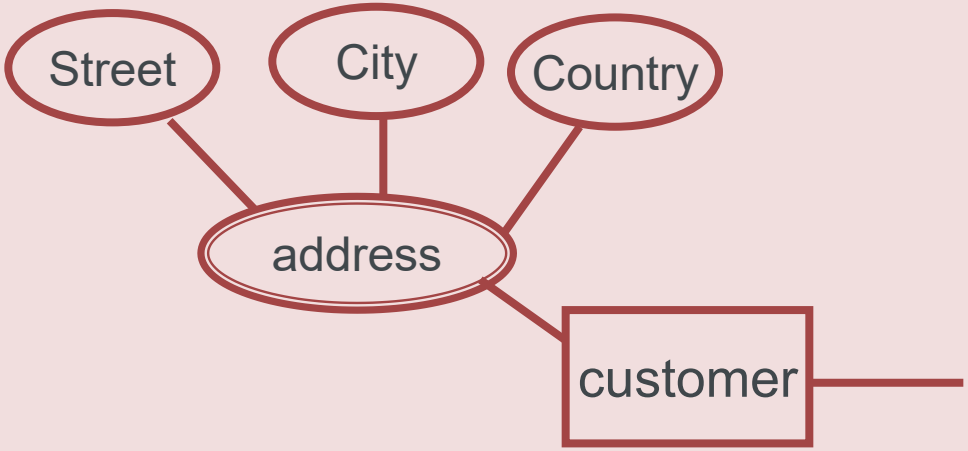
- derived (dashed ellipse)

Multi-valued



- multi-valued (double ellipse)

Complex Attribute



- multi-valued + Composite

Relationship

A Relationship is an association among several entities.

- A relationship may also have attributes

For example, consider the entity sets customer and loan and the relationship set borrower. We could associate the attribute date–issued to that relationship to specify the date when the loan was issued.

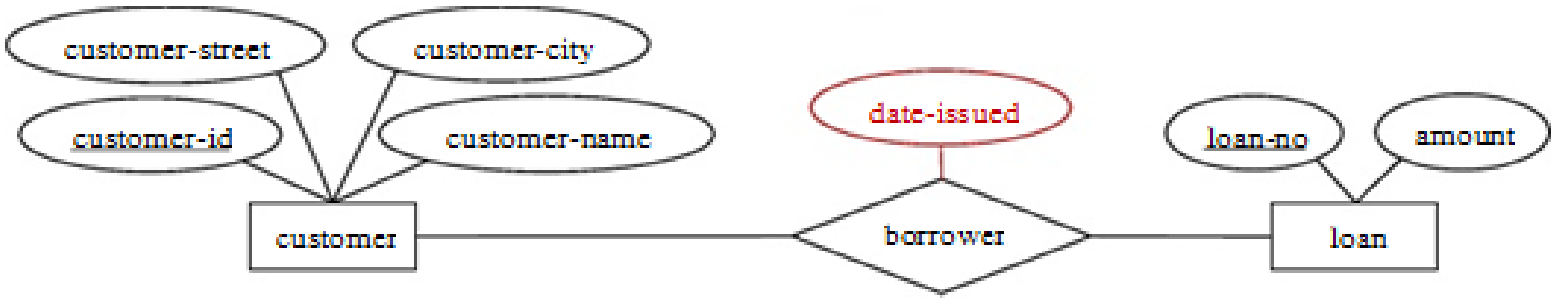
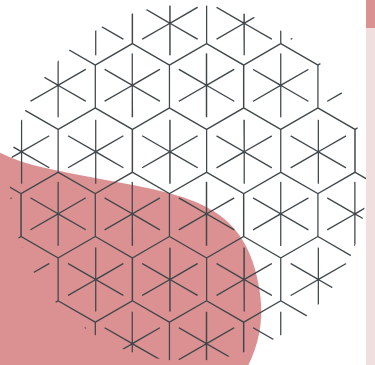


Figure: Descriptive attribute *date-issued*.

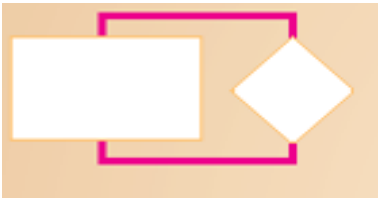
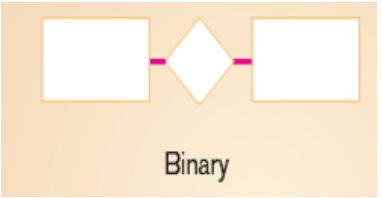
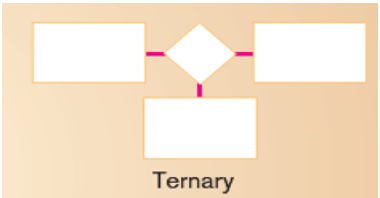
Relation



Relation has three Properties:

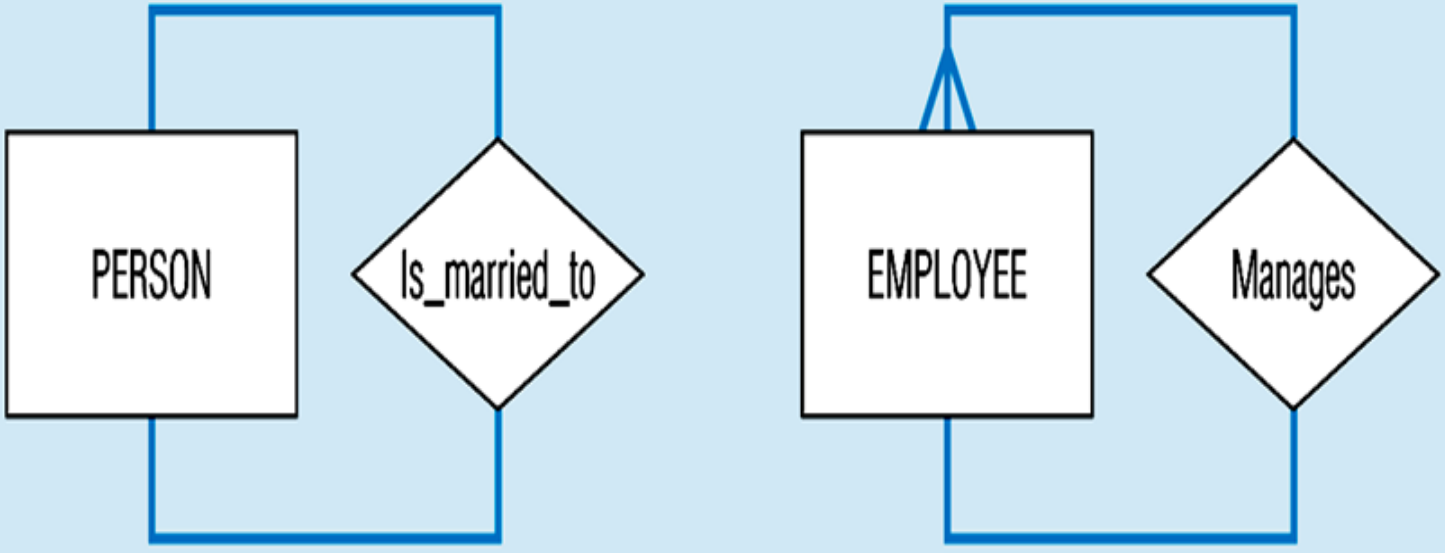
- Degree of Relationships
- Cardinality Constraint
- Participation Constraint

Degree of Relationships

Degree	Define	Shape
Unary	between two instances of one entity type	
Binary	between the instances of two entity types	
Ternary	among the instances of three entity types	

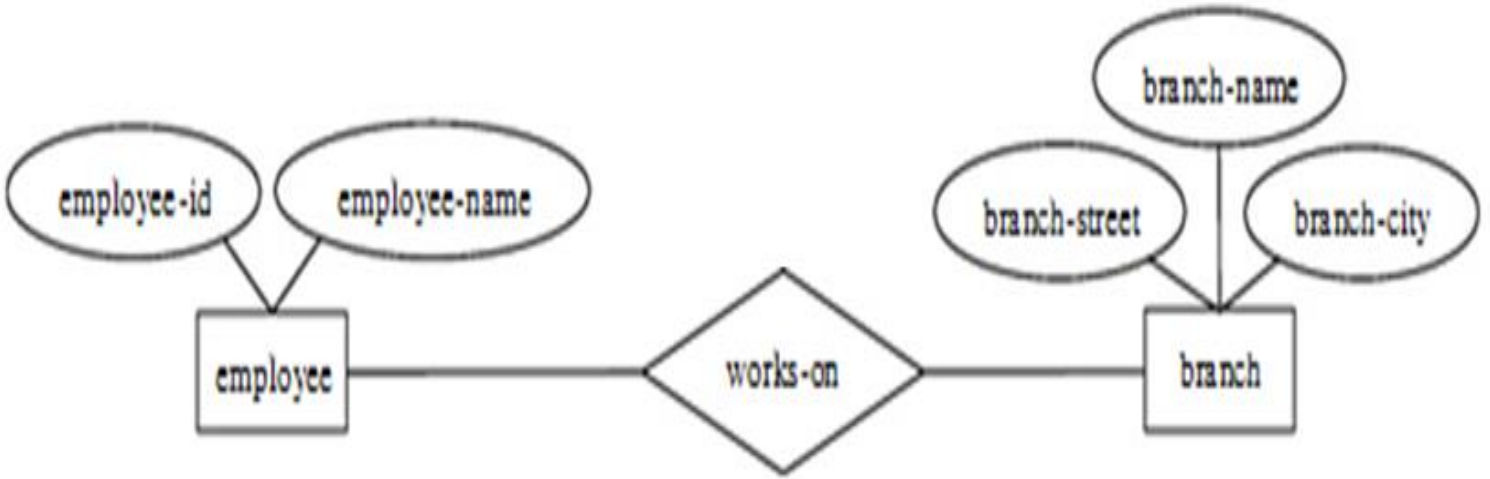
Recursive Relationship (Unary)

A relationship in which the same entity participates more than once.



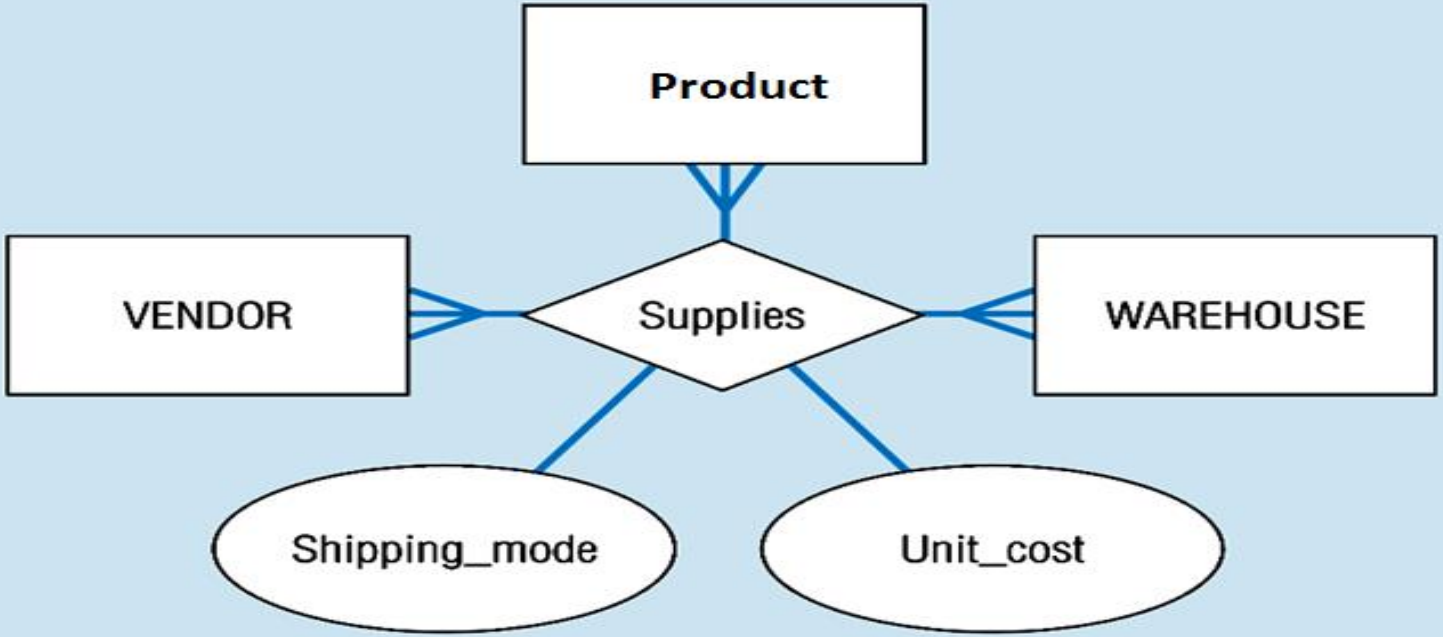
Binary Relationship

A binary relationship set is of degree 2.



Ternary Relationship

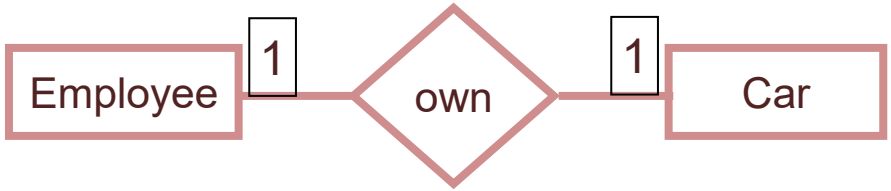
ternary relationship set is of degree 3.



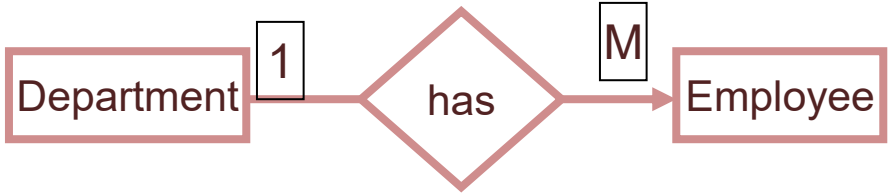
Cardinality

How many instances of one entity will or must be connected to a single instance from the other entities.

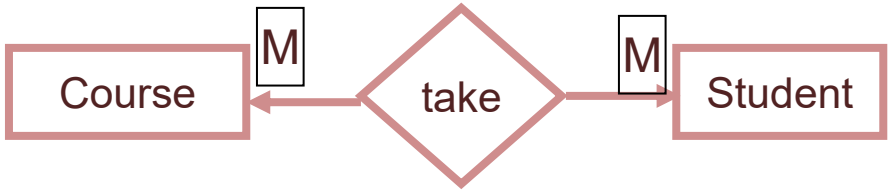
- One-One Relationship



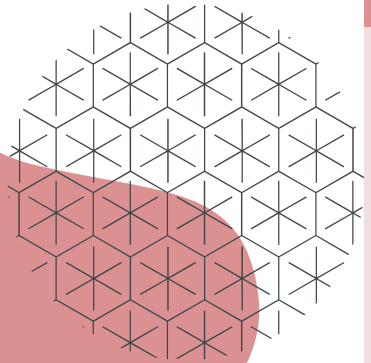
- One-Many Relationship



- Many- Many Relationship



PARTICIPATION



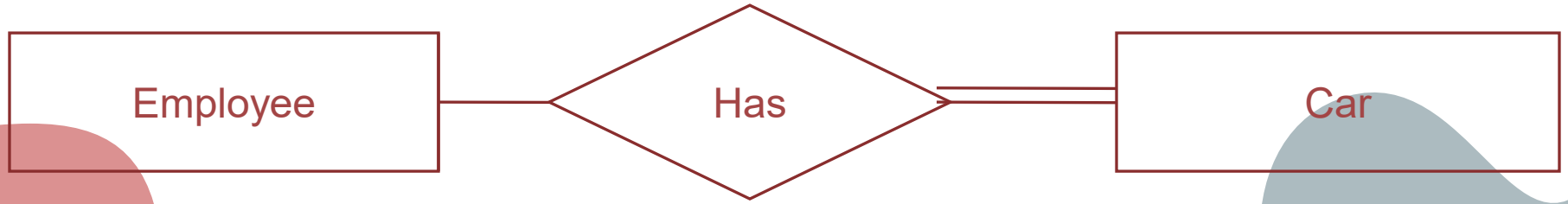
Relation has three Properties:

- **TOTAL**
- **PARTIAL**

PARTICIPATION CONSTRAINT

- An employee MUST work for a department
  An employee entity can exist only if it participates in a WORKS_FOR relationship instance
  So this participation is TOTAL

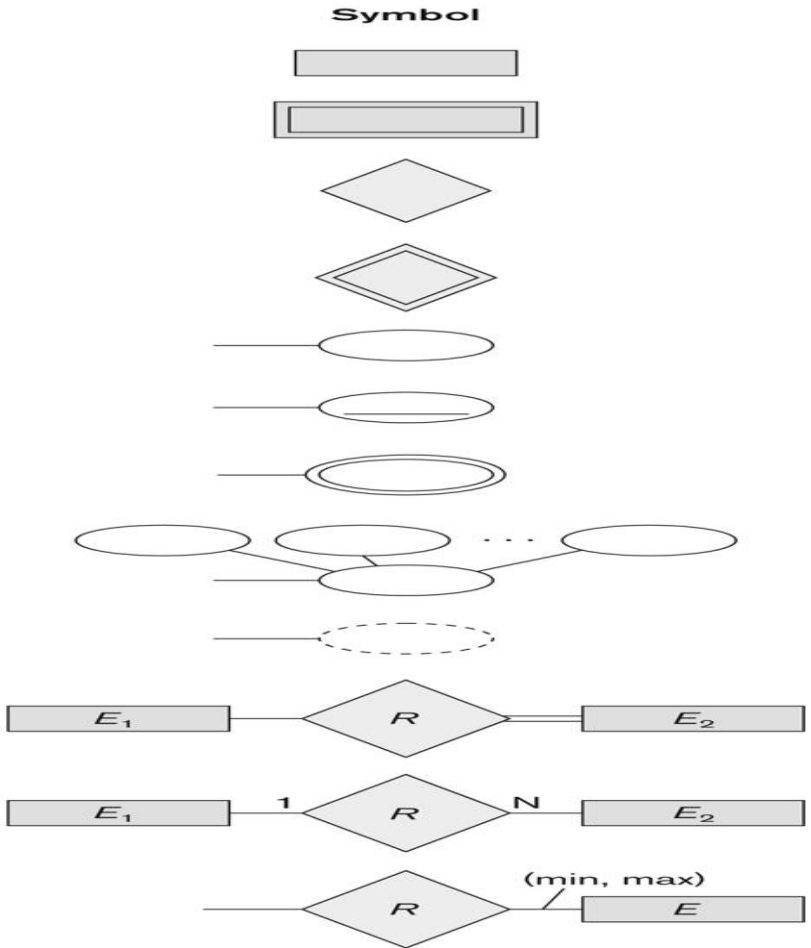
  Only some employees manage departments
  The participation is PARTIAL



An Employee may have a car.
A Car must be assigned to particular employee

Summary of notation for ER diagrams

Figure 3.14
Summary of the notation for ER diagrams.



Meaning
Entity
Weak Entity
Relationship
Indentifying Relationship
Attribute
Key Attribute
Multivalued Attribute
Composite Attribute
Derived Attribute
Total Participation of E_2 in R
Cardinality Ratio 1 : N for E_1:E_2 in R
Structural Constraint (min, max) on Participation of E in R

Case Study

(word)

Keys

Candidate Key

Primary Key

Foreign Key

Composite Key

Partial Key

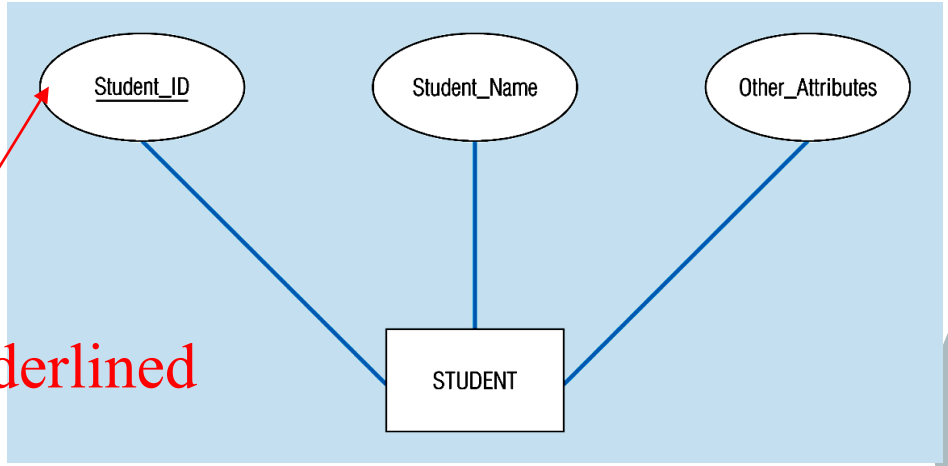
Alternate key

Super Key

Primary Key

Is the candidate key that is chosen by the database designer as the unique identifier of an entity. [Unique & Not Null]

- Primary key May be Composite

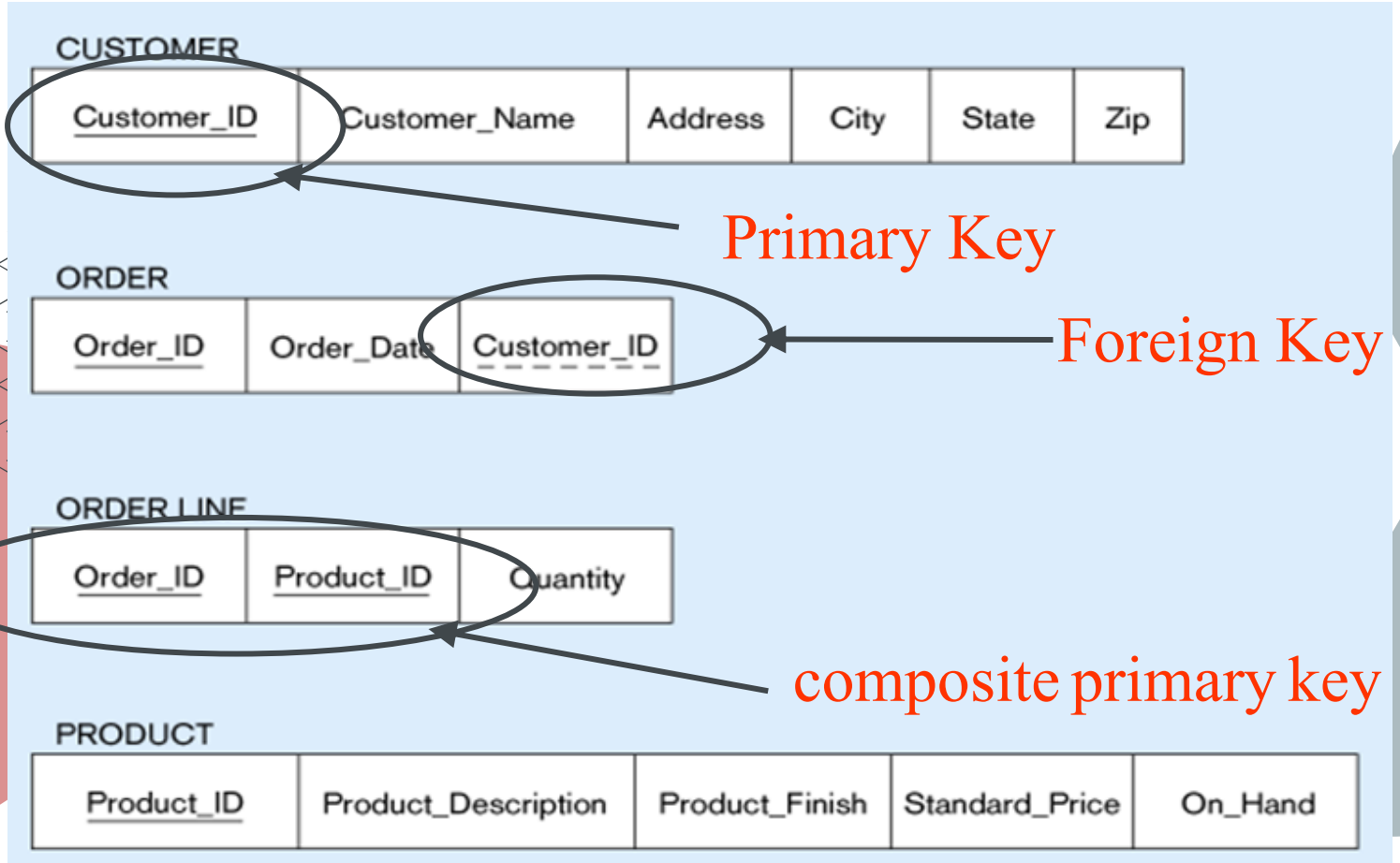


The key is underlined

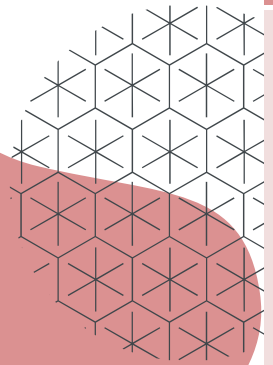
08

Mapping

Mapping -> DB Tables



ER-to-Relational Mapping



steps

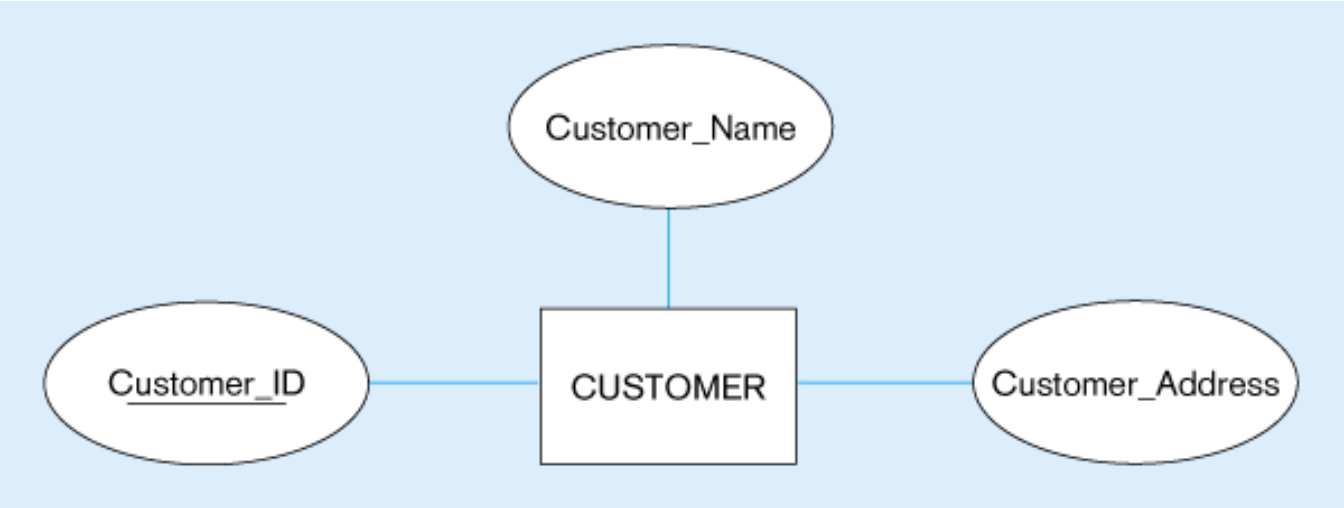
- Step 1: Mapping of Regular Entity Types
- Step 2: Mapping of Weak Entity Types
- Step 3: Mapping of Binary 1:1 Relation Types
- Step 4: Mapping of Binary 1:N Relationship Types.
- Step 5: Mapping of Binary M:N Relationship Types.
- Step 6: Mapping of N-ary Relationship Types.
- Step 7: Mapping of Unary Relationship.

Step I: Mapping of Regular Entity Types

- Create table for each entity type –> if there is no 1–1 relationship mandatory from 2 sides
- Choose one of key attributes to be the primary key

Mapping Regular entity

- (a) CUSTOMER entity type with simple attributes



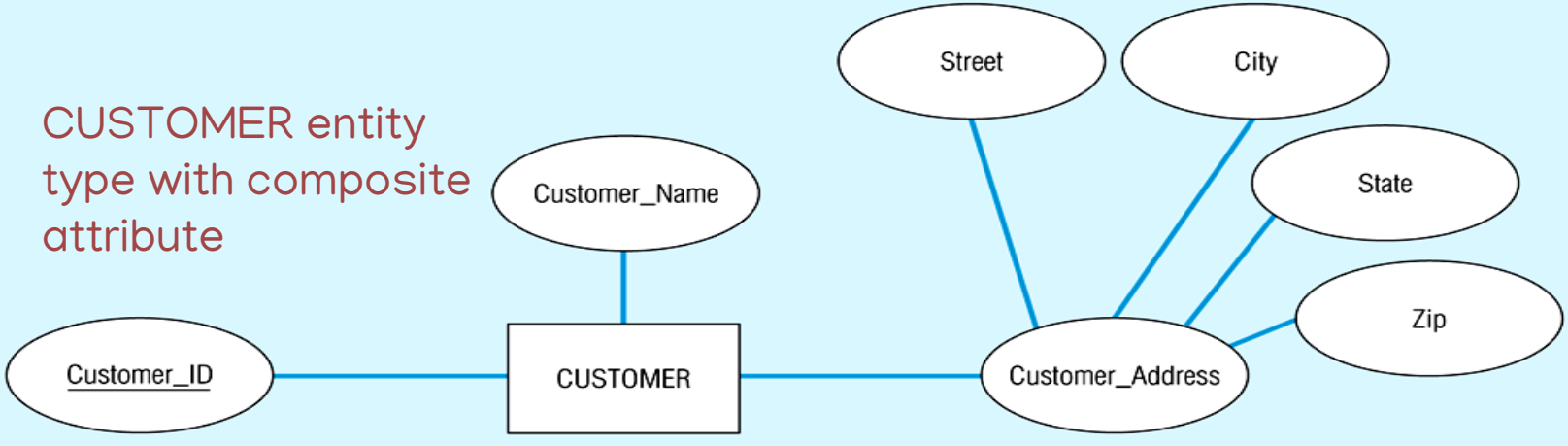
- (b) CUSTOMER relation

CUSTOMER

Customer_ID	Customer_Name	Customer_Address

Mapping Composite attribute

CUSTOMER entity type with composite attribute

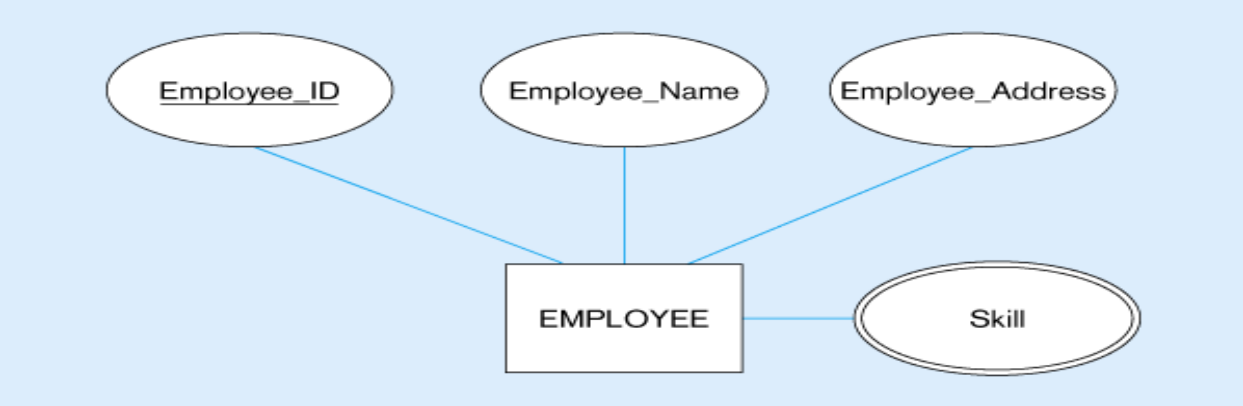


CUSTOMER relation with address detail

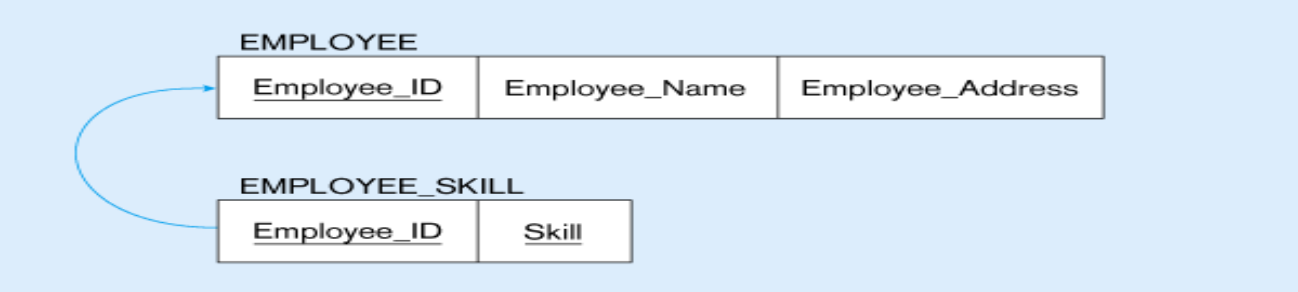
CUSTOMER

Customer_ID	Customer_Name	Street	City	State	Zip

Mapping Multivalued Attribute



Multivalued attribute becomes a separate relation with foreign key



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

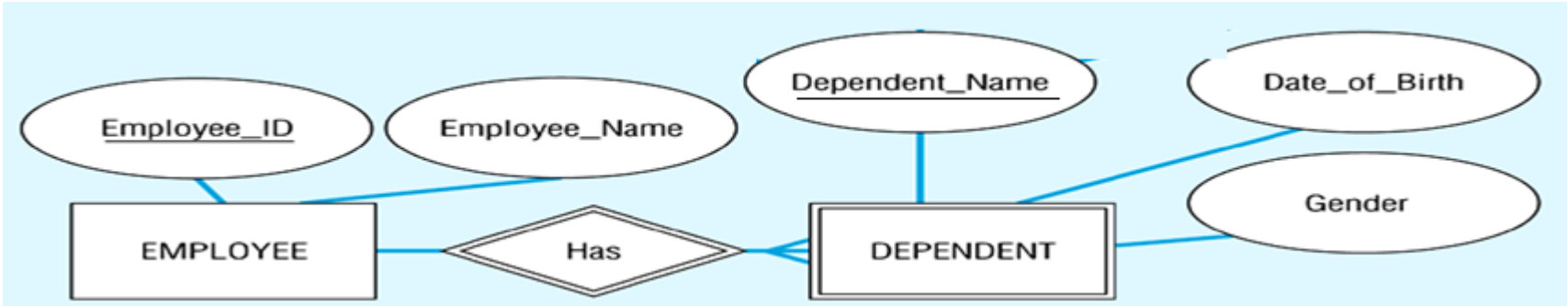
Mapping Derived & Complex

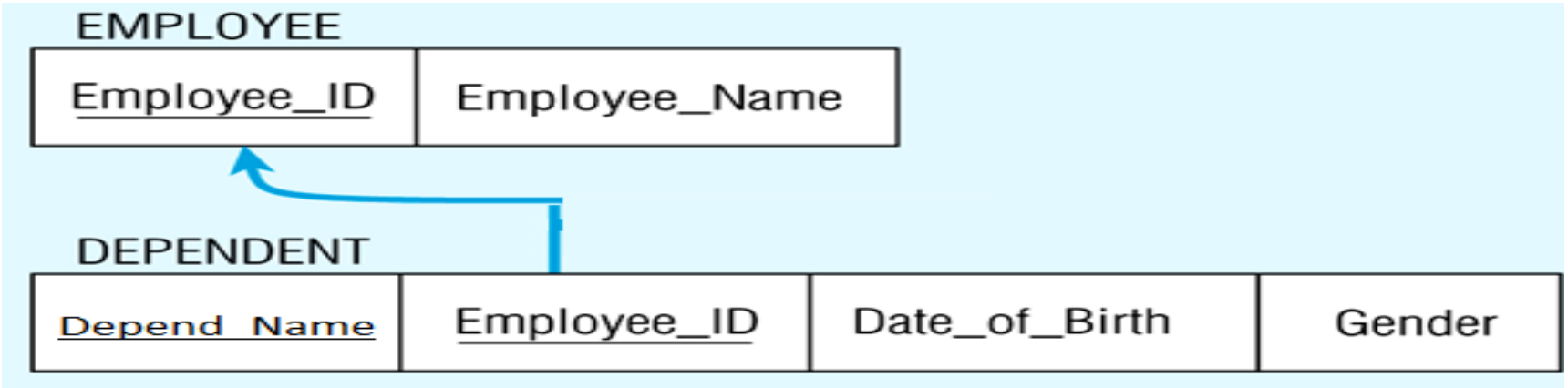
- In the most cases Derived attribute not be stored in DB
- Mapping Complex Like Mapping Multivalued attribute then including parts of the multivalued attributes as columns in DB

Step 2: Mapping of Weak Entity Types

- Create table for each weak entity.
- Add foreign key that correspond to the owner entity type.
- Primary key composed of:
  - Partial identifier of weak entity
  - Primary key of identifying relation (strong entity)

Mapping Weak entity





Composite primary key

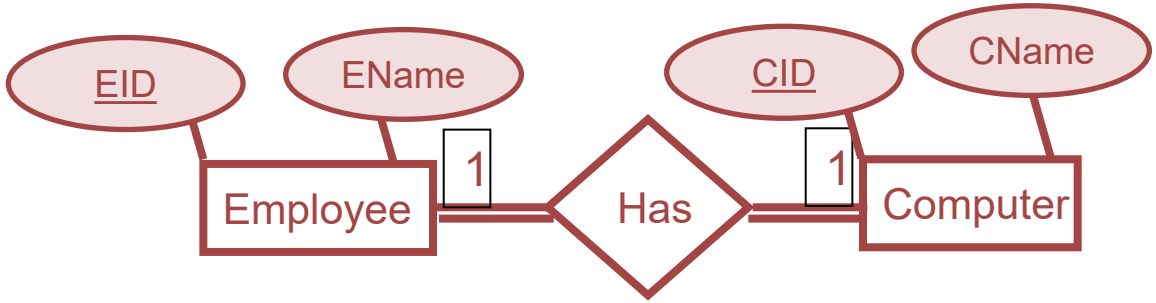
Step 3: Mapping of Binary 1:1 Relation Types

- Merged two tables if both sides are Mandatory.
- Add FK into table with the total participation relationship to represent optional side.
- Create third table if both sides are optional.

2 Mandatory

One-to-One

2 Mandatory



1 table

tbl_xy (PK,….,…,….)

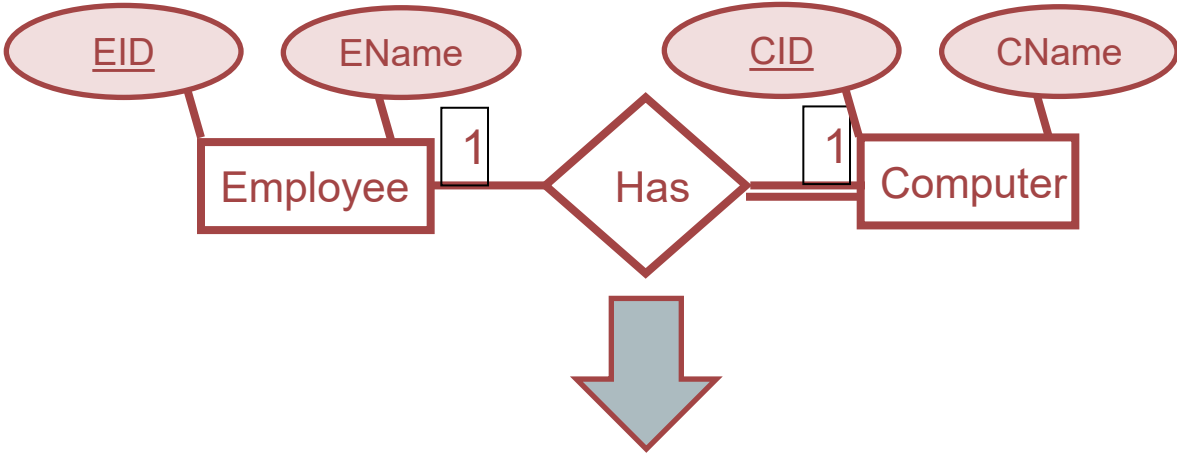
PK = PKx or PKy

Emp(EID, Ename, Cname, CID)

Optional-Mandatory

One-to-One

X optional – Y mandatory



2 tables

tbl_x (PKx,….,…….)

tbl_y (PKy,….,….,PKx….)

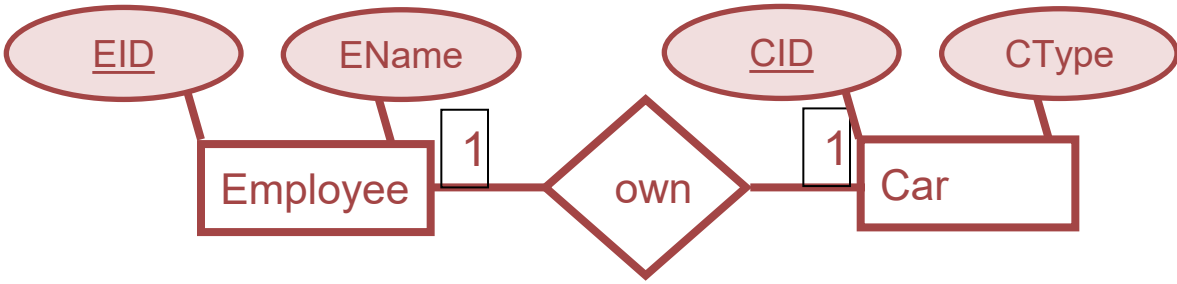
Employee(EID, Ename)

Computer(CID, Cname, EID_FK)

2 Optional

One-to-One

2 Optional



3 tables

tbl_x (PKx,….,…….)

tbl_y (PKy,….,…….)

tbl_xy (PKxy,….,…,FKxy,….)

PKxy = PKx or PKy

Employee(EID, Ename)

Car(CID, CType)

Emp_Car(EID, CID_FK)

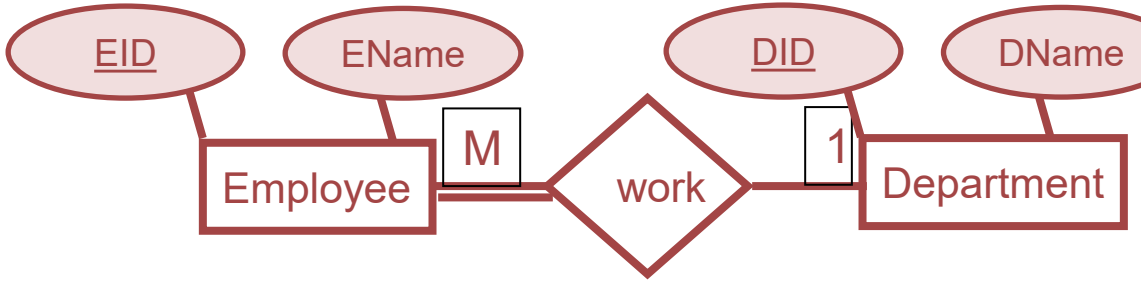
Step 4: Mapping of Binary I:N Relationship Types.

- Add FK to N-side table if N-Side mandatory
- Add any simple attributes of relationship as column to N-side table.

Many is Mandatory

One-to-Many

X whatever– Y mandatory



2 tables

tbl_x (PKx,….,…….)

tbl_y (PKy,….,….,FKy….)

FKy= PKx

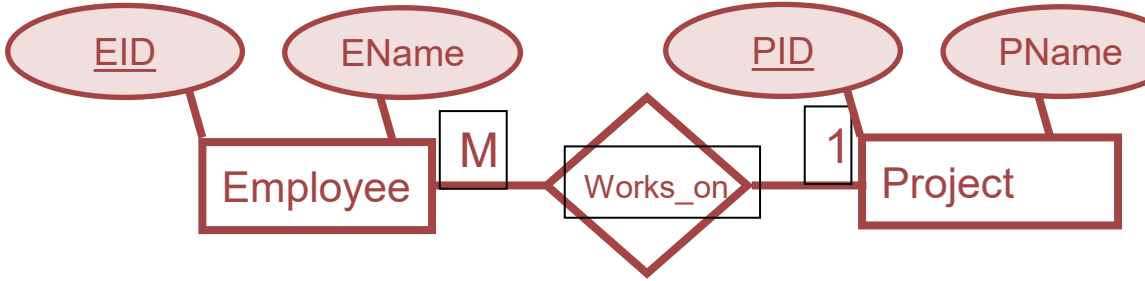
Department(DID, Dname)

Employee(EID, Ename,DID)

Many is Optional

One-to-Many

X whatever– Y Optional



3 tables

tbl_x (PKx,….,…….)

tbl_y (PKy,….,…….)

tbl_xy (PKxy,….,…….)

PKxy = PKy

Project(PID, Pname)

Employee(EID, Ename)

Proj_Emp(EID,PID_FK)

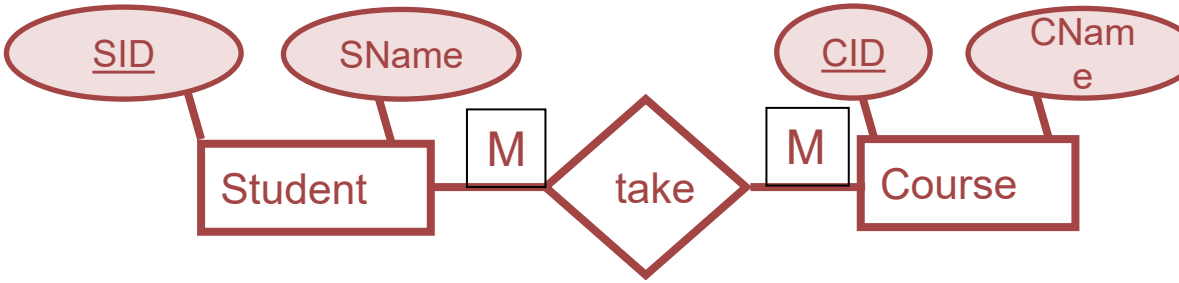
Step 5: Mapping of Binary M:N Relationship Types.

- Create a new third table
- Add FKs to the new table for both parent tables
- Add simple attributes of relationship to the new table if any .

M:N

Many-to-Many

X whatever– Y whatever



3 tables

tbl_x (PKx,….,…….)

tbl_y (PKy,….,…….)

tbl_xy (PKx ,PKy, ….,…….)

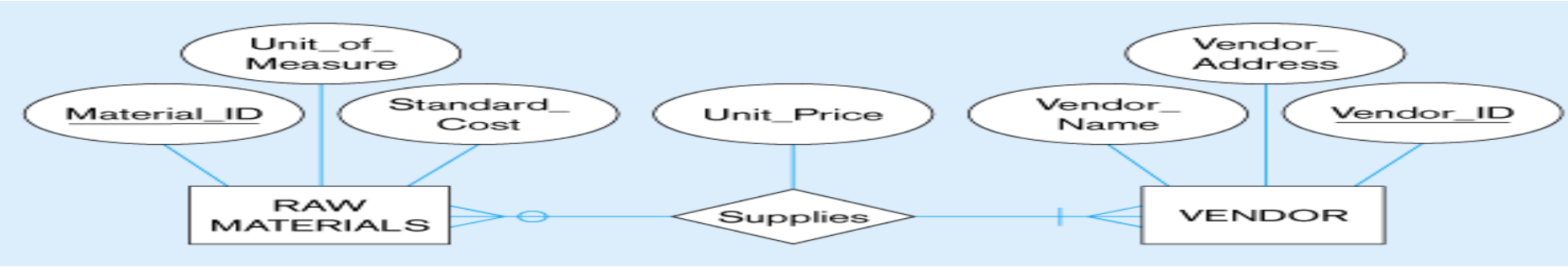
PKxy=_PKx+PKy

Student(SID, Sname)

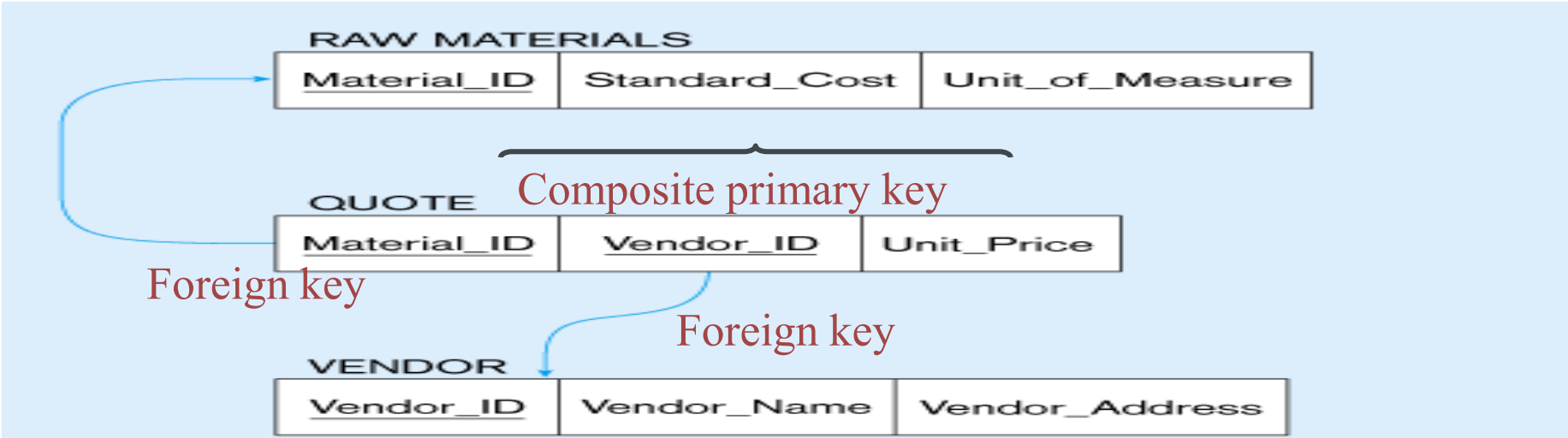
Course(CID, Cname)

Stud_Course(SID, CID)

M:N with attribute



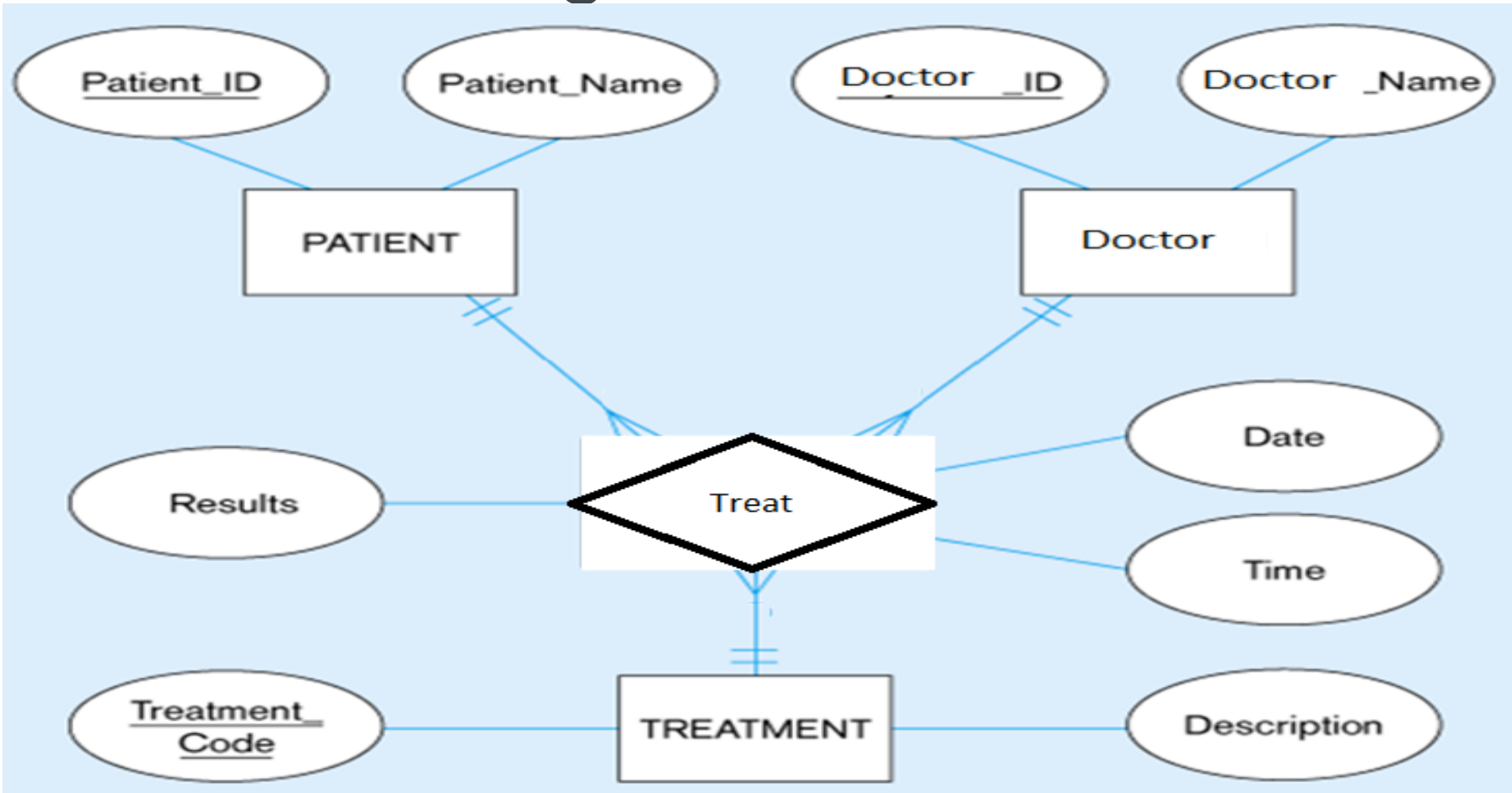
The *Supplies* relationship will need to become a separate relation



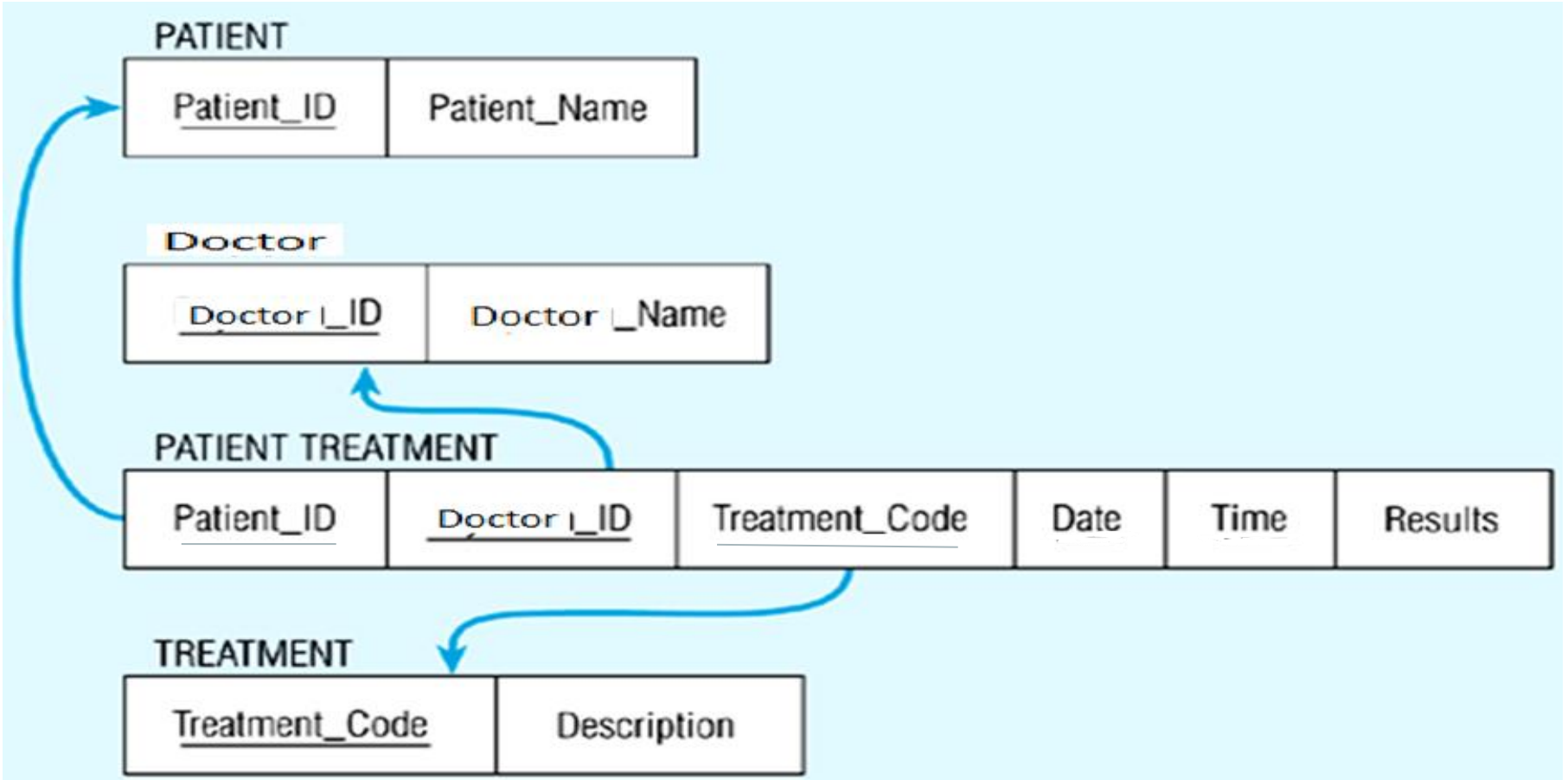
Step 6: Mapping of N-ary Relationship Types.

- If n > 2 then :
- Create a new third table
- Add FKs to the new table for all parent tables
- Add simple attributes of relationship to the new table if any .

Step 6: Mapping of N-ary Relationship Types.

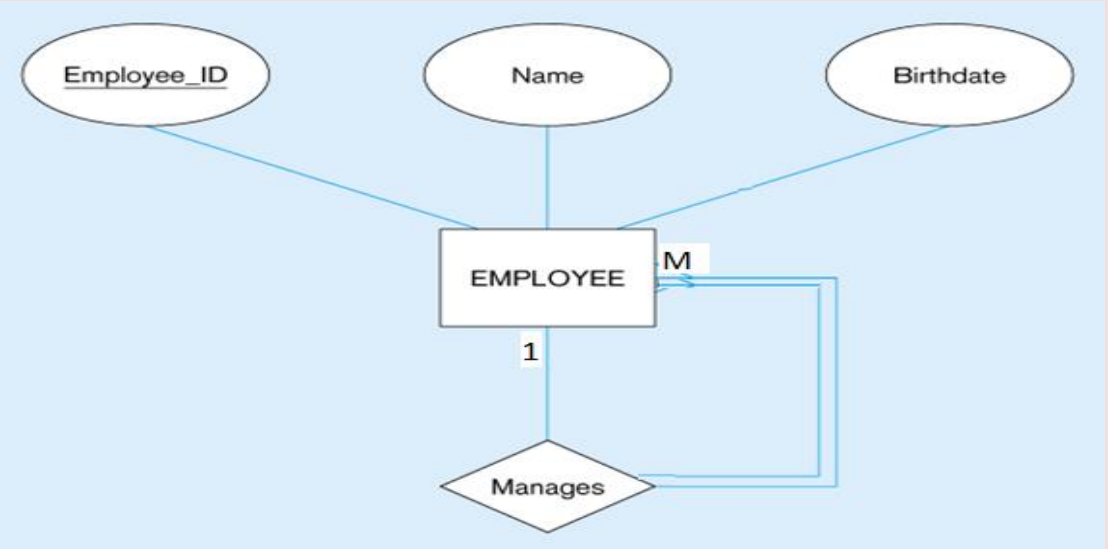


Step 6: Mapping of N-ary Relationship Types.

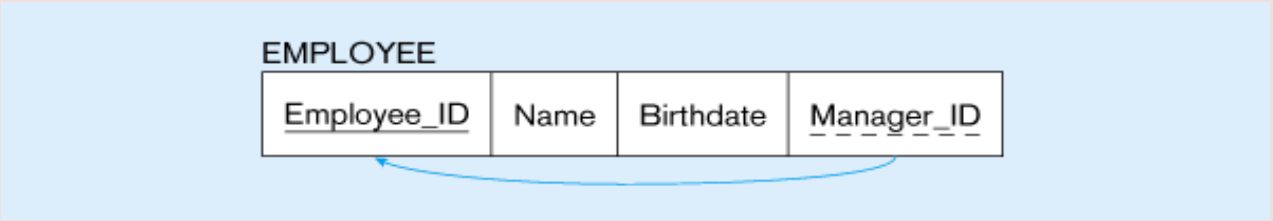


Step 7:Mapping Unary Relationship

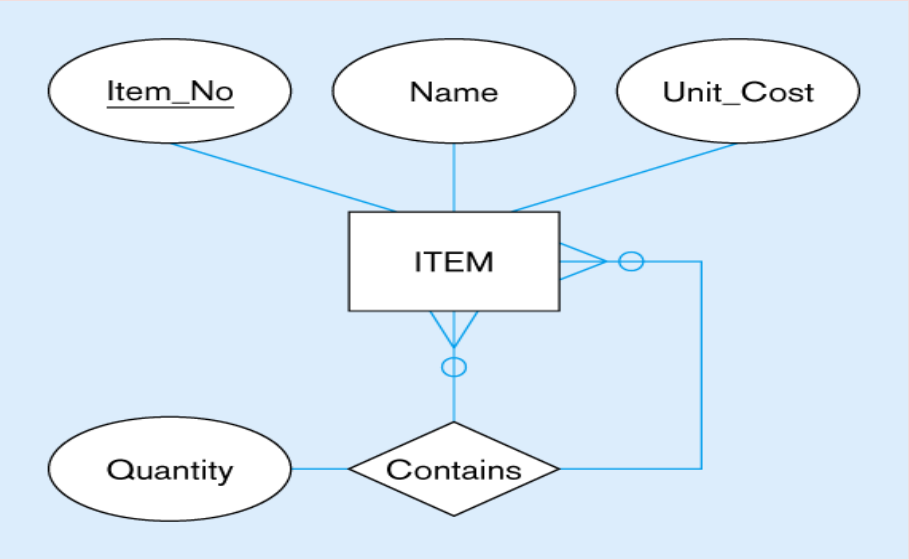
(a) EMPLOYEE entity with Manages relationship



(b) EMPLOYEE relation with recursive foreign key

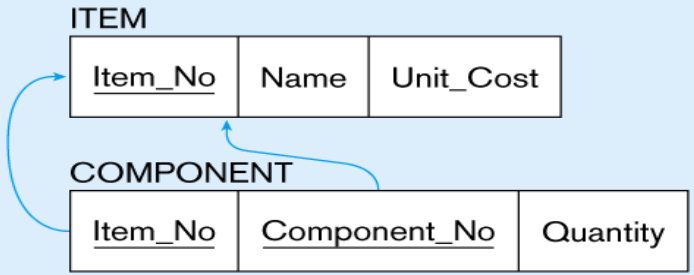


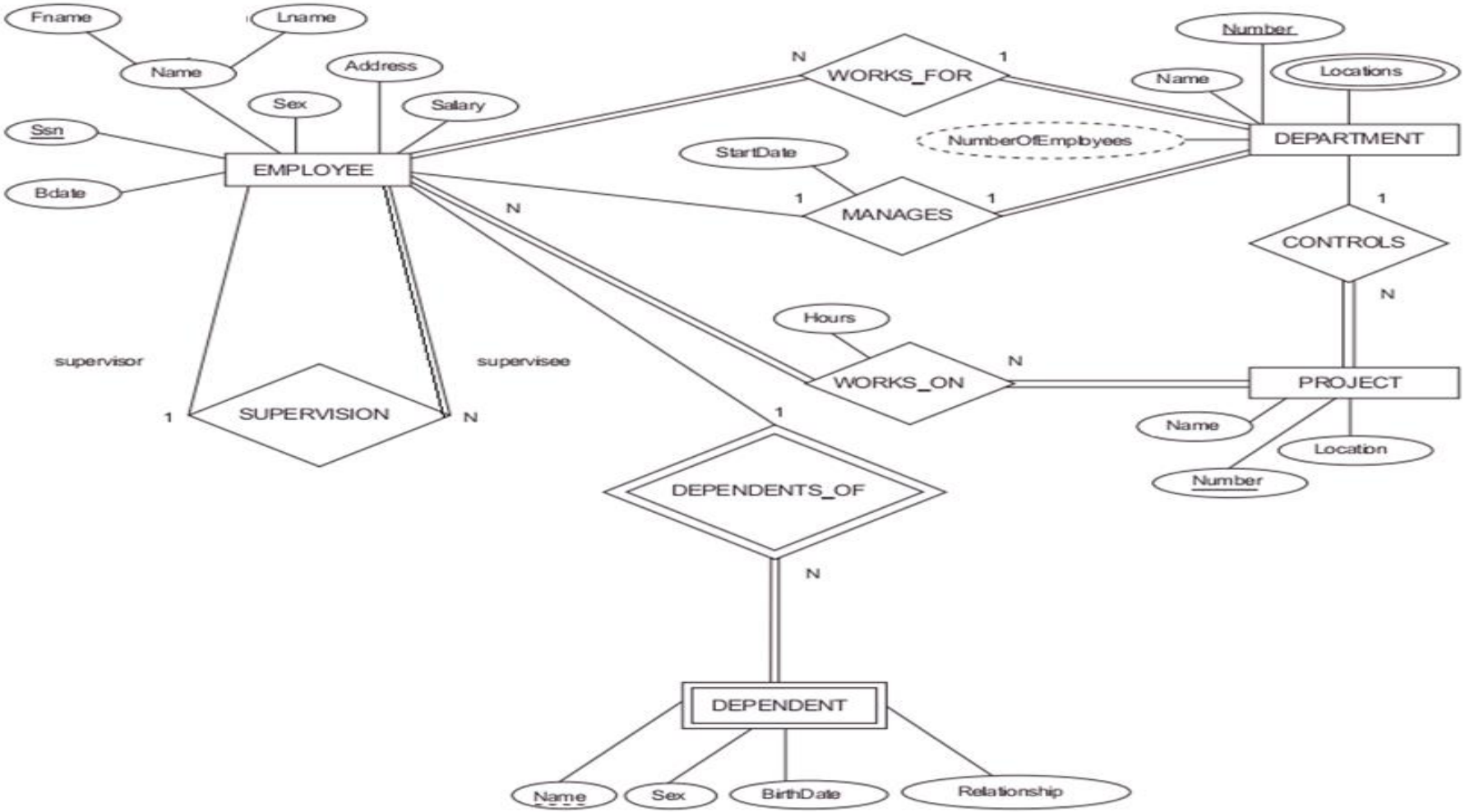
Mapping a unary M:N relationship



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

(b) ITEM and COMPONENT relations




Fname
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
N
1
CONTROLS
N
Hours
WORKS_ON
N
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Thanks

zeyadashraf015@gmail.com

01097143595

Zeyad Elmalky