Intro To Database

(Database Fundamental using SQL SERVER)

Mohamed ELshafei

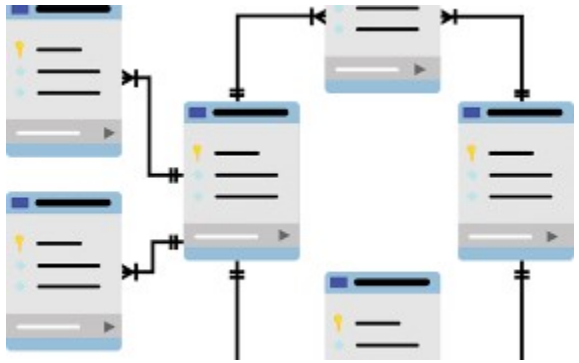
Agenda

- Relational Database.
- ERD Mapping to Tables
- SQL.
- MYSQL.
- DDL.
- MySQL Data Types
- DCL.
- General Query

Intake 39 - DB

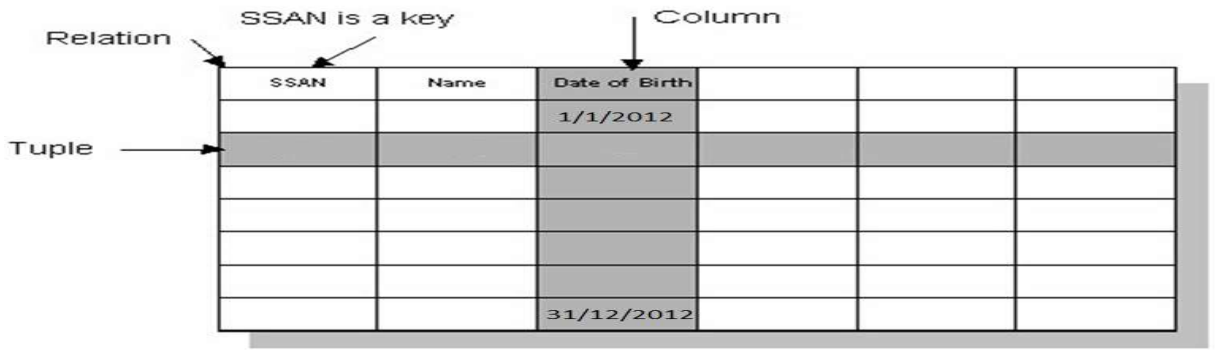
Relational Database

- 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.



Basic Database Structure

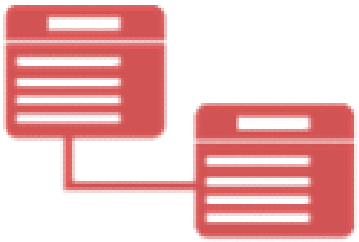
- **Table or entity**: a collection of records
- **Attribute or Column or field**: a Characteristic of an entity
- **Row or Record** : the specific characteristics of one entity
- **Database**: a collection of tables



Mapping

ERD Mapping to Tables

Steps



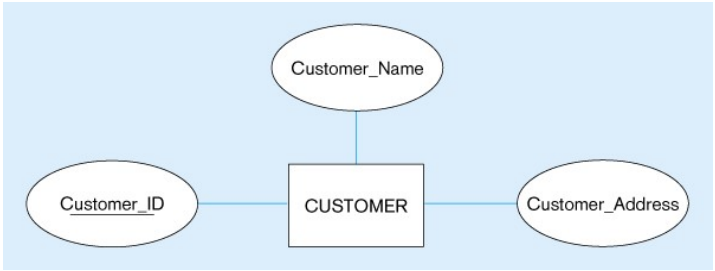
Intake 39 - DB

ER-to-Relational Mapping

- **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 1: Mapping of Regular Entity Types

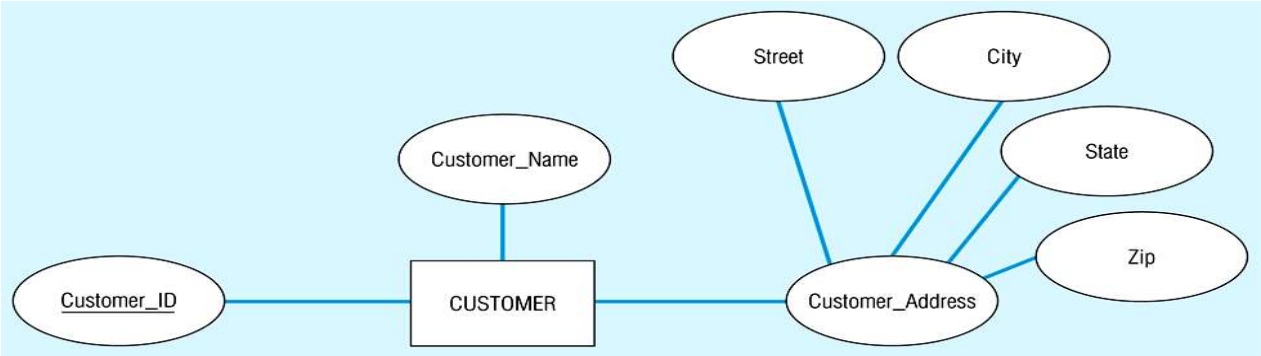
- Create table for each entity type.
- Choose one of key attributes to be the primary key.



CUSTOMER

Customer_ID	Customer_Name	Customer_Address

Mapping Composite attribute

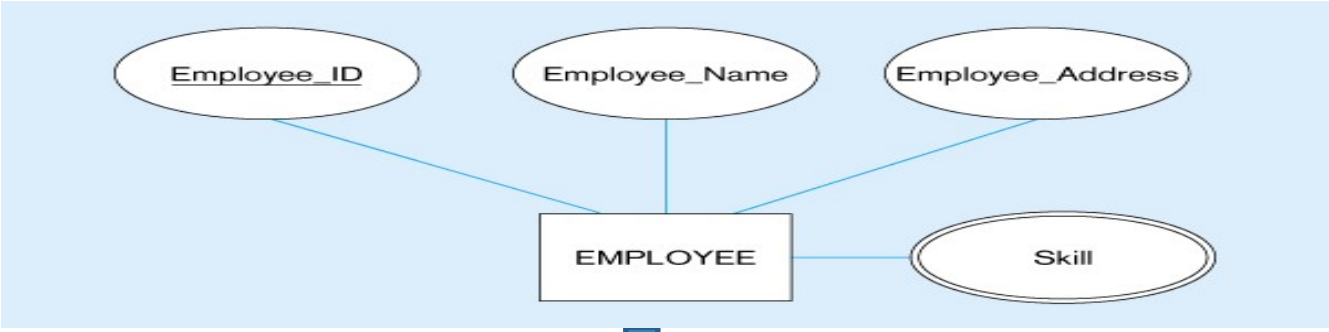


CUSTOMER relation with address detail

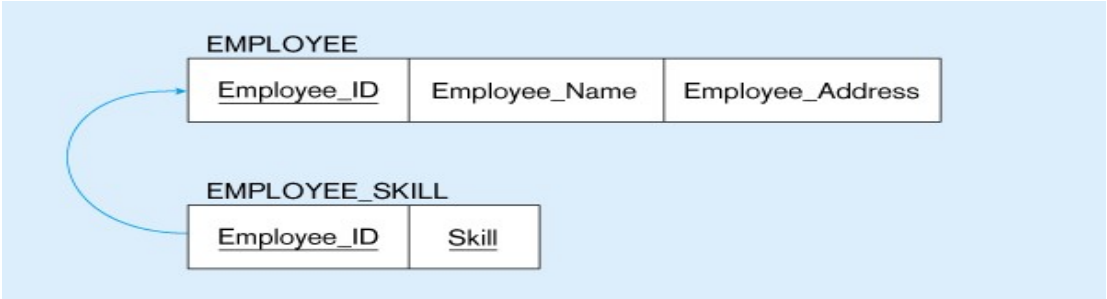
CUSTOMER

Customer_ID	Customer_Name	Street	City	State	Zip

Mapping Multivalued Attribute



Multivalued attribute becomes a separate relation with foreign key



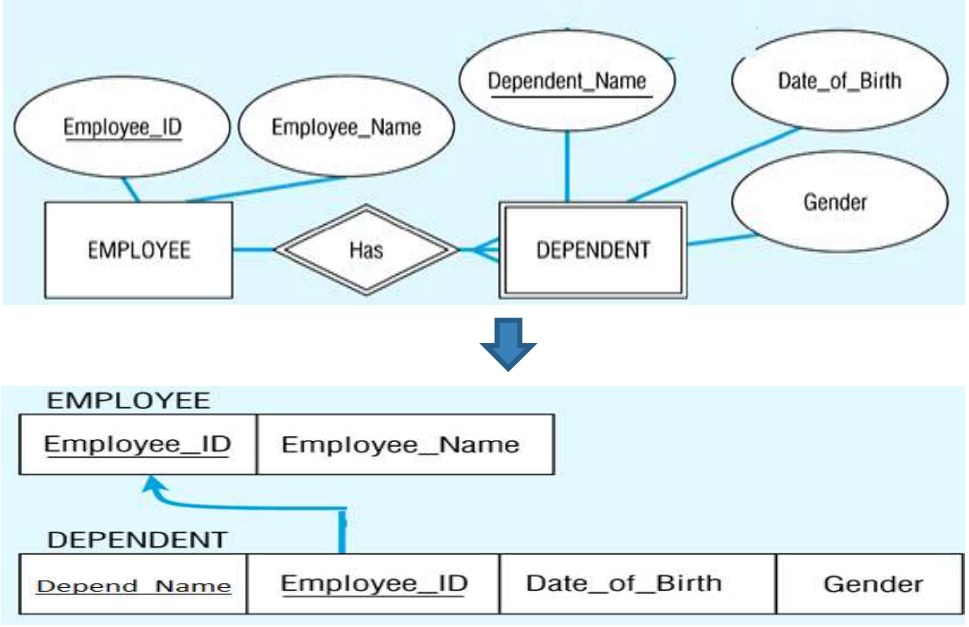
Mapping Derived & Complex

- 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)

Step 3: Mapping of Binary 1:1 Relation Types

- Merged two tables if both sides are Mandatory.

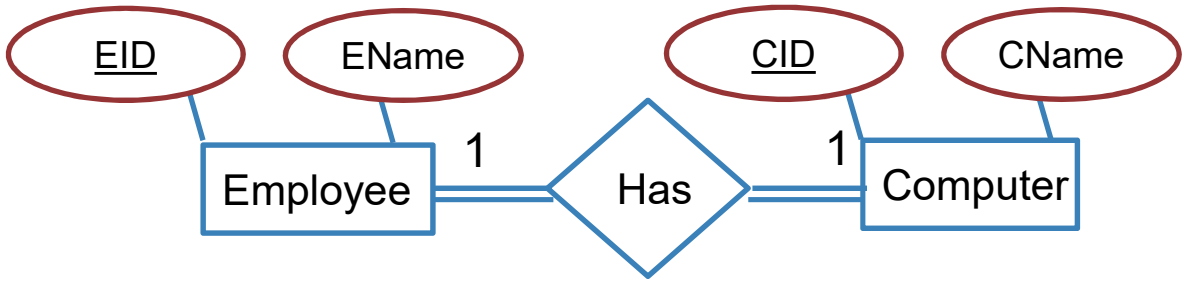
- Add FK into table with the total participation relationship to represent optional side.

- Create third table if both sides are optional.

One To One 2 Mandatory

One-to-One

2 Mandatory



1 table

tbl_xy (PK,….,…,….)

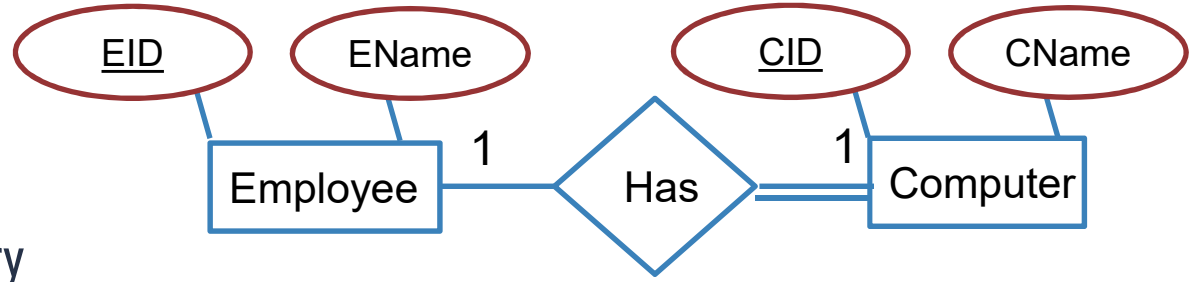
PK = PKx or PKy

Emp(EID, Ename, Cname, CID)

One To One Optional-Mandatory

One-to-One

X optional – Y mandatory



2 tables

tbl_x (PKx,….,…….)

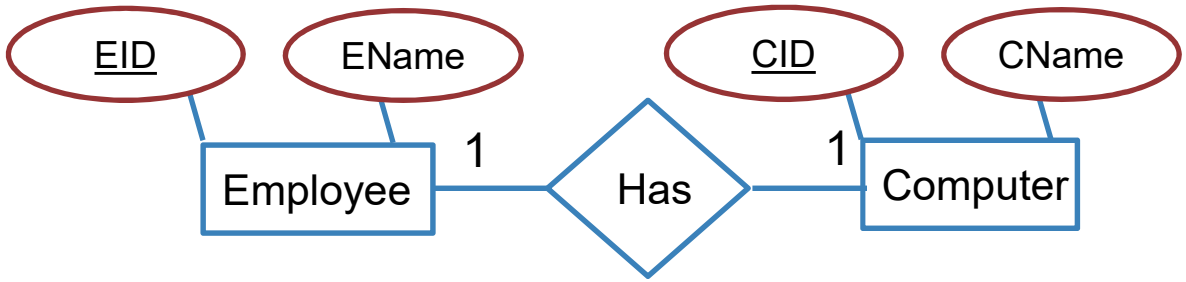
tbl_y (PKy,….,….,PKx….)

Employee(EID, Ename)

Computer(CID, Cname, EID_FK)

One To One 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)

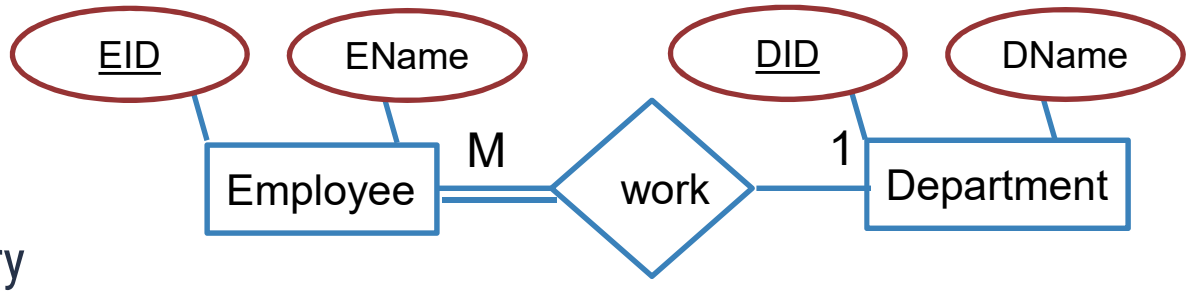
Emp_Car(EID, CID_FK)

Step 4: Mapping of Binary 1: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.

One To Many (Many is Mandatory)

One-to-Many



X whatever– Y mandatory

2 tables

tbl_x (PKx,….,…….)

tbl_y (PKy,….,….,FKy….)

FKy= PKx

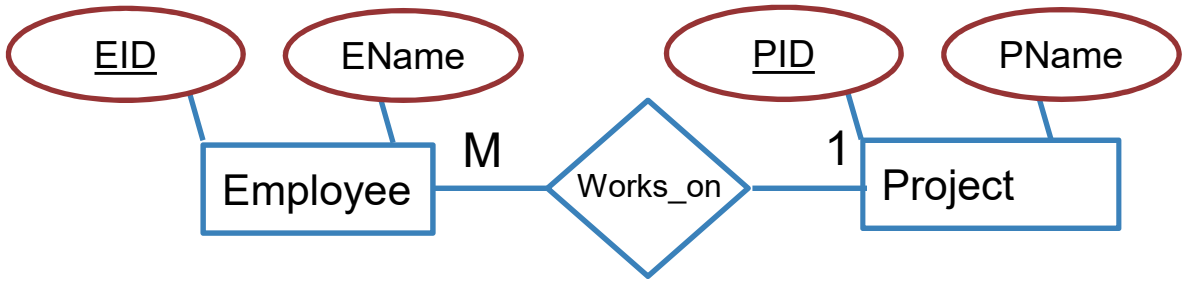
Department(DID, Dname)

Employee(EID, Ename,DID)

One To Many (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)

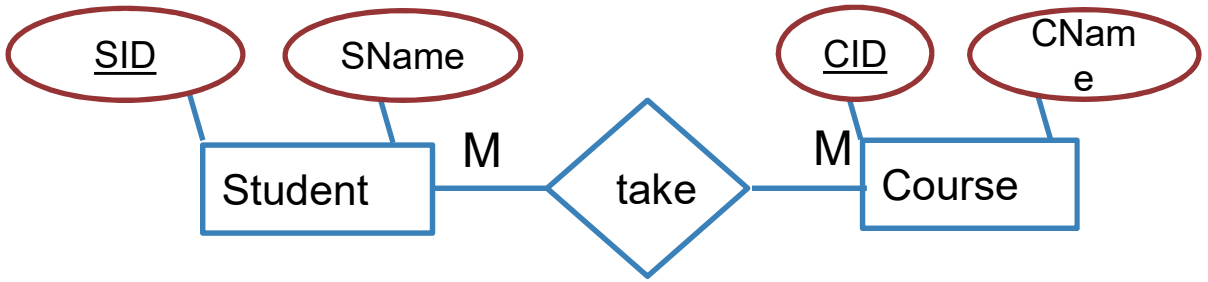
Proj_Emp(EID,PID_FK)

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 .

Many To Many

Many-to-Many



X whatever– Y whatever

3 tables

tbl_x (PKx,….,…….)

tbl_y (PKy,….,…….)

tbl_xy (PKx ,PKy, ….,…….)

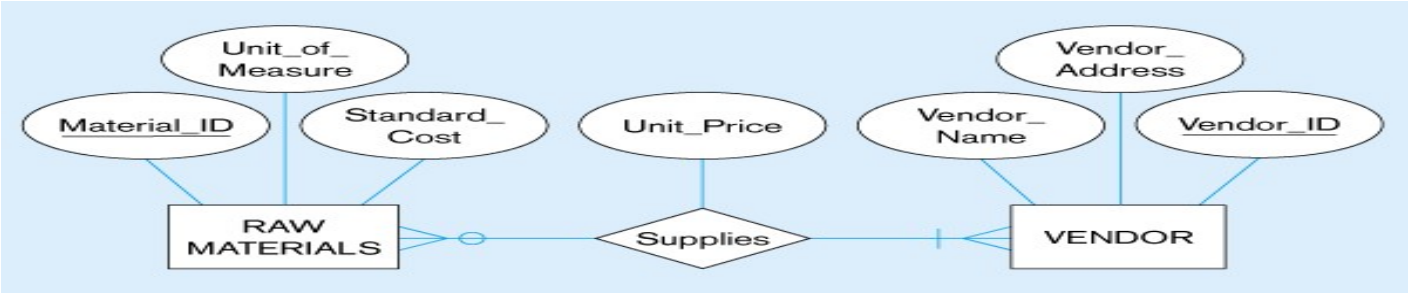
PKxy=_PKx+PKy

Student(SID, Sname)

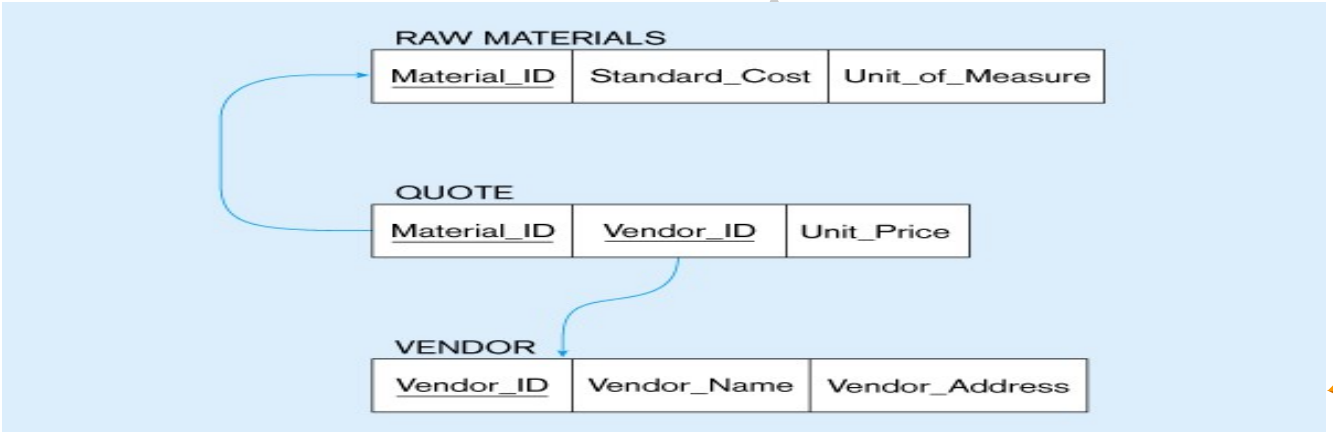
Course(CID, Cname)

Stud_Course(SID, CID)

Many To Many with attribute



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

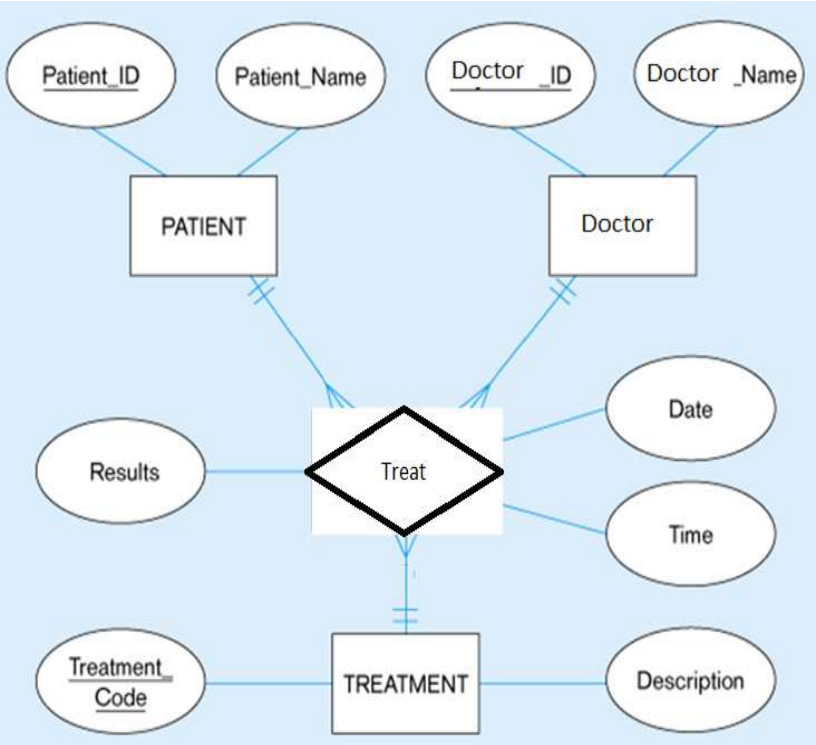


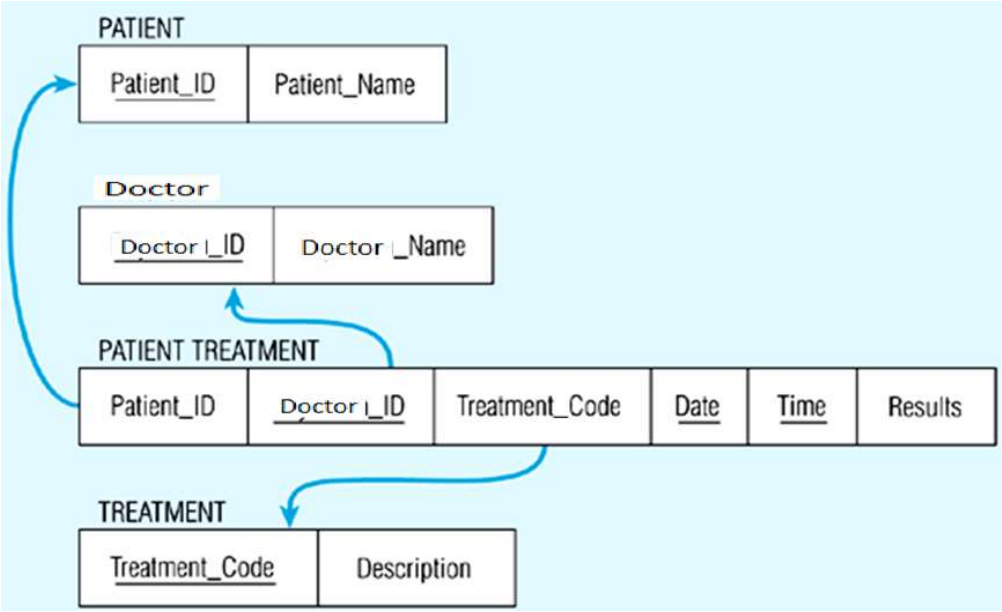
Step 6: Mapping of N-ary Relationship Types.

If n > 2 then :

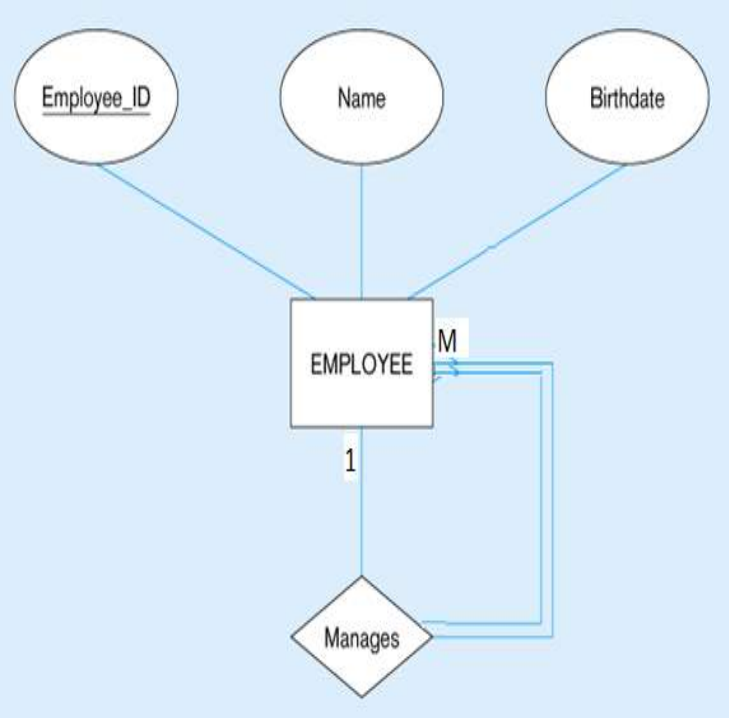
- Create a new third table.
- Add FKs to the new table for all parent tables.
- Add simple attributes of relationship to the new table if any .

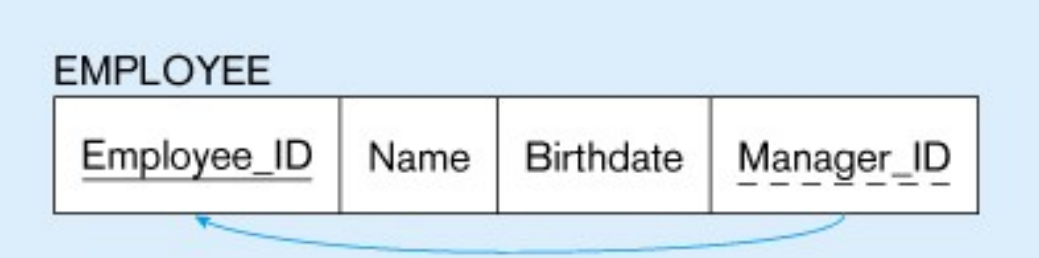
Step 6: Mapping of N-ary Relationship Types.





Step 7:Mapping Unary Relationship





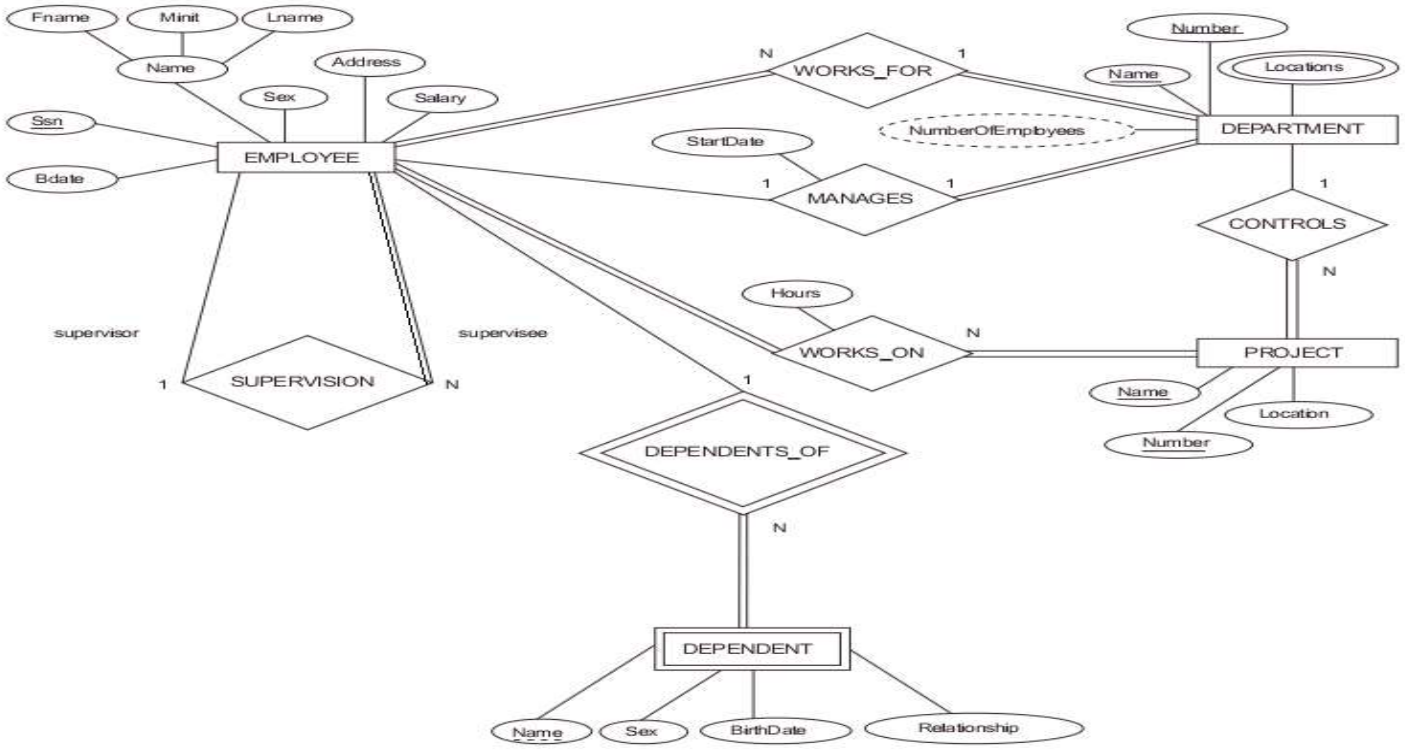
- EMPLOYEE relation with recursive foreign key

Case Study

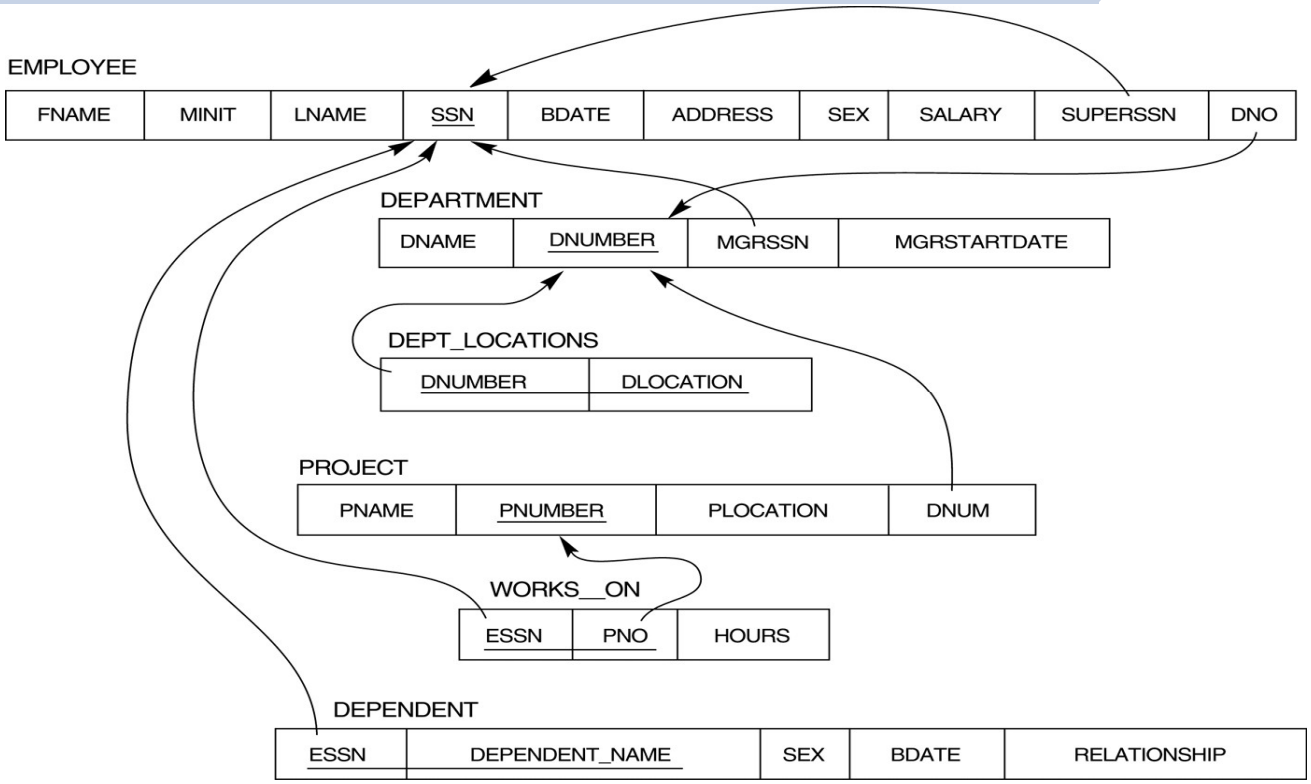
Company

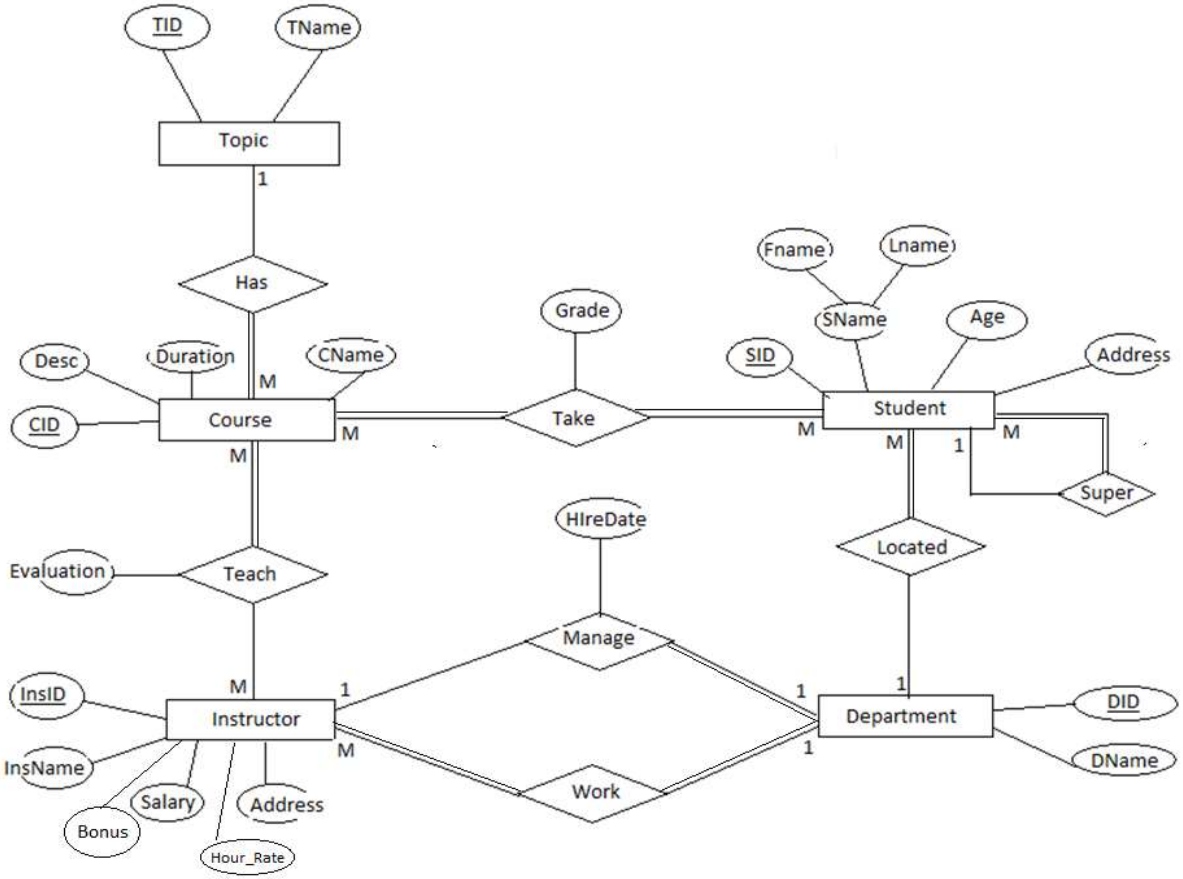
Intake 39 - DB

Case study


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

Result


EMPLOYEE
FNAME
MINIT
LNAME
SSN
BDATE
ADDRESS
SEX
SALARY
SUPERSSN
DNO
DEPARTMENT
DNAME
DNUMBER
MGRSSN
MGRSTARTDATE
DEPT_LOCATIONS
DNUMBER
DLOCATION
PROJECT
PNAME
PNUMBER
PLOCATION
DNUM
WORKS_ON
ESSN
PNO
HOURS
DEPENDENT
ESSN
DEPENDENT_NAME
SEX
BDATE
RELATIONSHIP


TID
TName
Topic
1
Has
M
Desc
Duration
CName
CID
Course
M
M
Grade
Take
M
Fname
Lname
SName
Age
SID
Address
Student
M
1
M
Super
HIreDate
Located
Evaluation
Teach
M
Manage
1
1
InsID
Instructor
M
1
Department
DID
DName
InsName
Salary
Address
Bonus
Hour_Rate
Work

Result

- Student(**St-id**,st-fname,st-Lname,st-age,st-super,Dept-ID)
- Course(**Crs-id**.Crs-Name,Crs-Duration,Top-id)
- Topic(**Top-ID**,Top-Name)
- Stud-Course(**St-ID,Crs-ID**,grade)
- Instructor(**Ins-ID**,ins-Name,Address,Salary,Dept-ID)
- Ins-Course(**Ins-ID,Crs-ID**,Evalution)
- Department(**Dept-ID**,Dept-Name,Manager-ID,HireDate)