

Intro To Database

(Database Fundamental using MySQL)

Mohamed ELshafei





Agenda

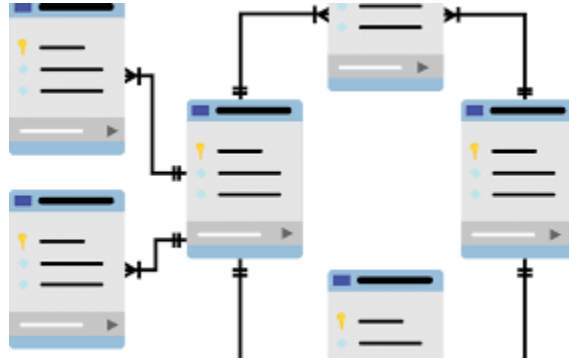
- Relational Database.
- ERD Mapping to Tables
- SQL.
- MYSQL.
- DDL.
- MySQL Data Types
- DCL.
- General Query





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

Diagram illustrating a database table structure with annotations:

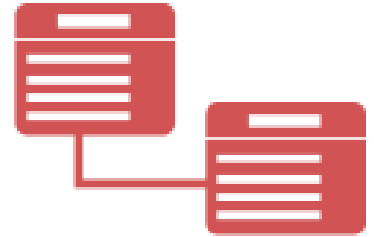
- Relation:** Points to the entire table structure.
- SSAN is a key:** Points to the SSAN column header.
- Column:** Points to the Date of Birth column header.
- Tuple:** Points to a row in the table.

SSAN	Name	Date of Birth			
		1/1/2012			
		31/12/2012			

Mapping

ERD Mapping to Tables

Steps





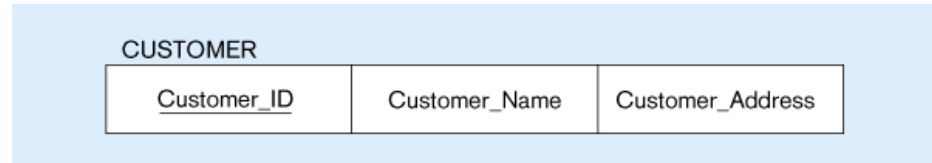
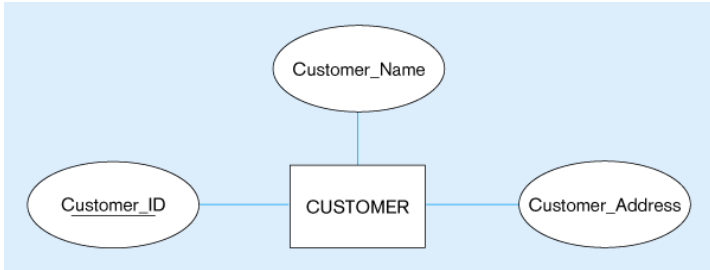
ER-to-Relational Mapping

- **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.



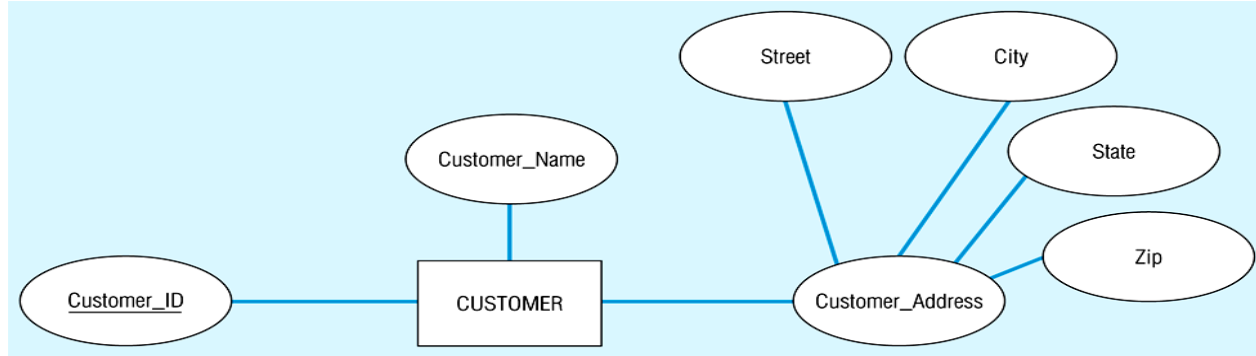
Step 1: Mapping of Regular Entity Types

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





Mapping Composite attribute

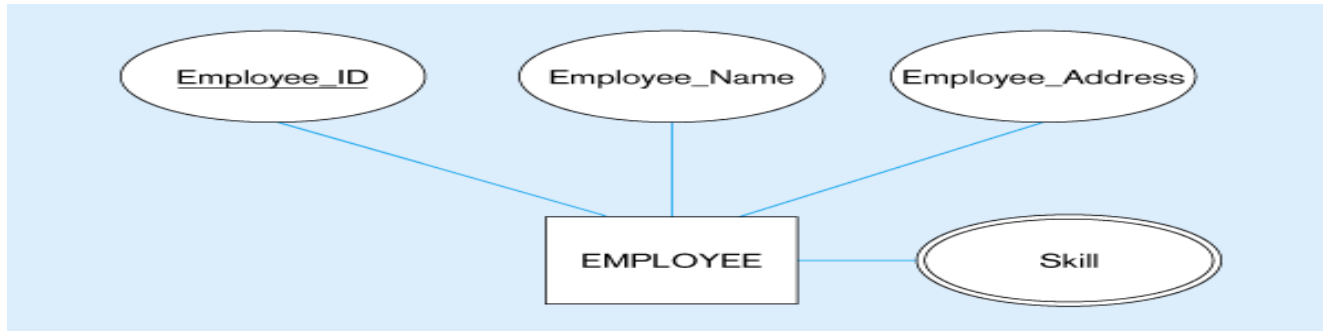


CUSTOMER CUSTOMER relation with address detail

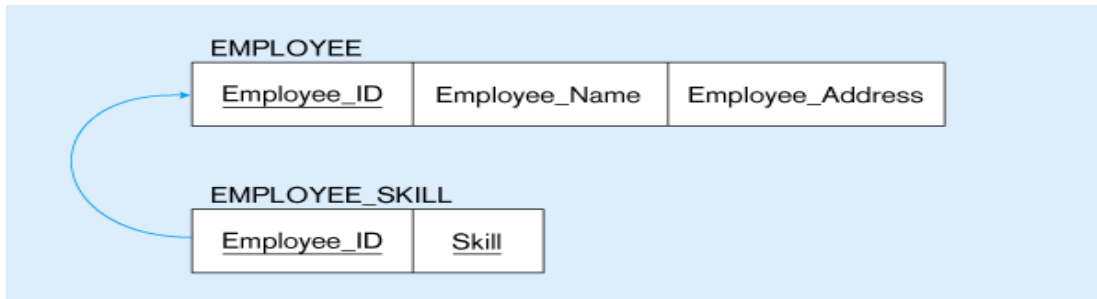
<u>Customer_ID</u>	Customer_Name	Street	City	State	Zip
--------------------	---------------	--------	------	-------	-----



Mapping Multivalued Attribute



Multivalued attribute becomes a separate relation with foreign key



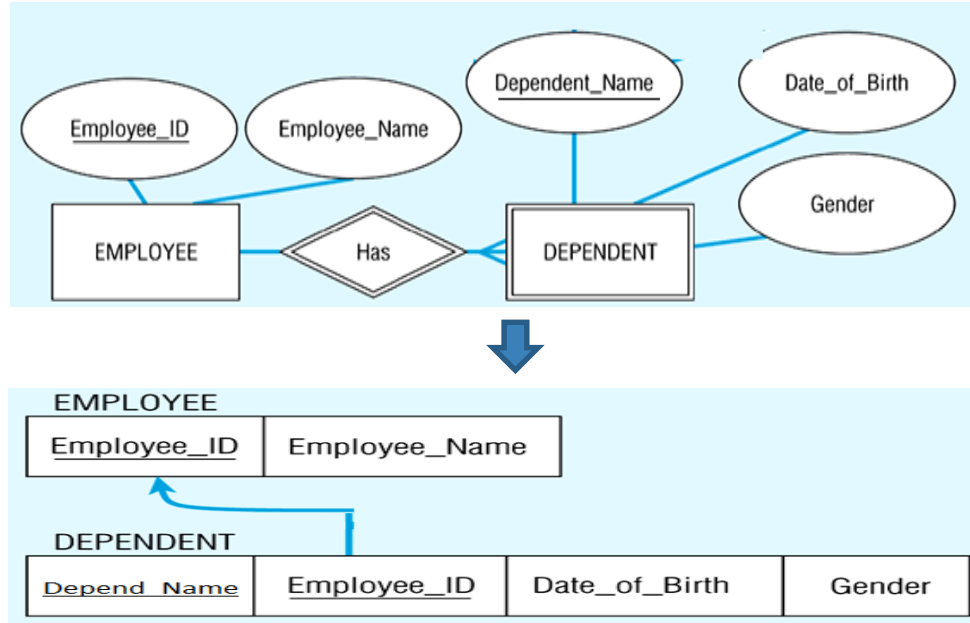


Mapping Derived & Complex

- 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.

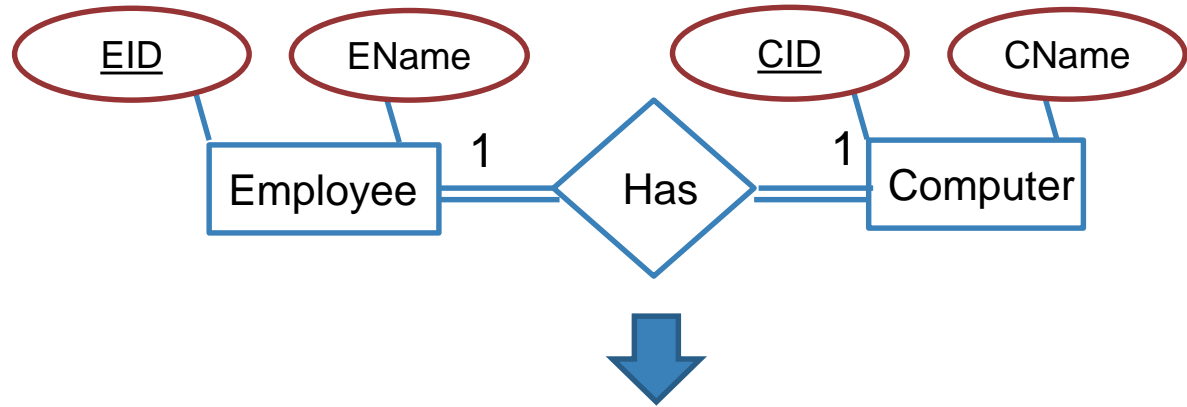


One To One 2 Mandatory

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

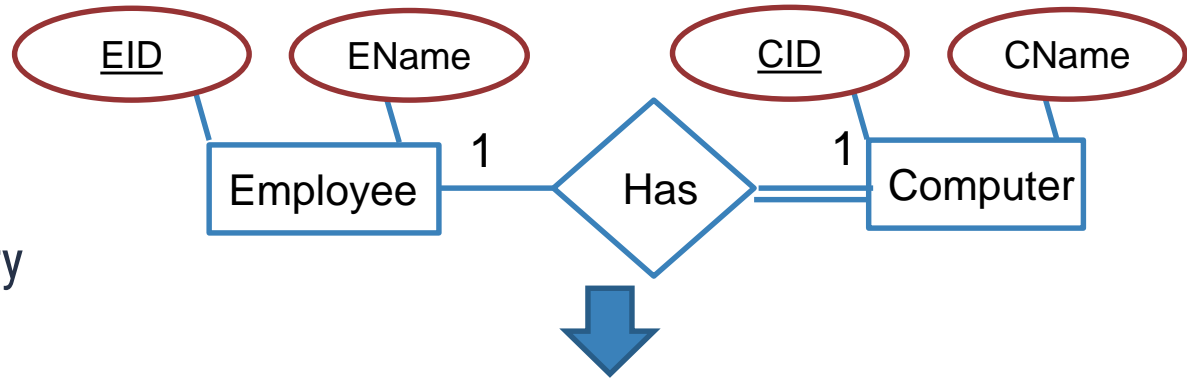
X optional – Y mandatory



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



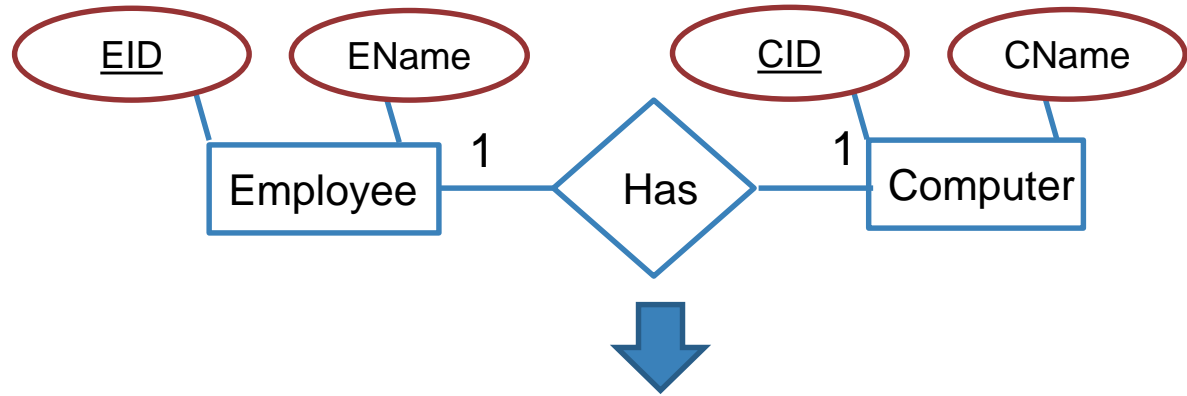
3 tables

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 Optional)

One-to-Many

X whatever– Y Optional



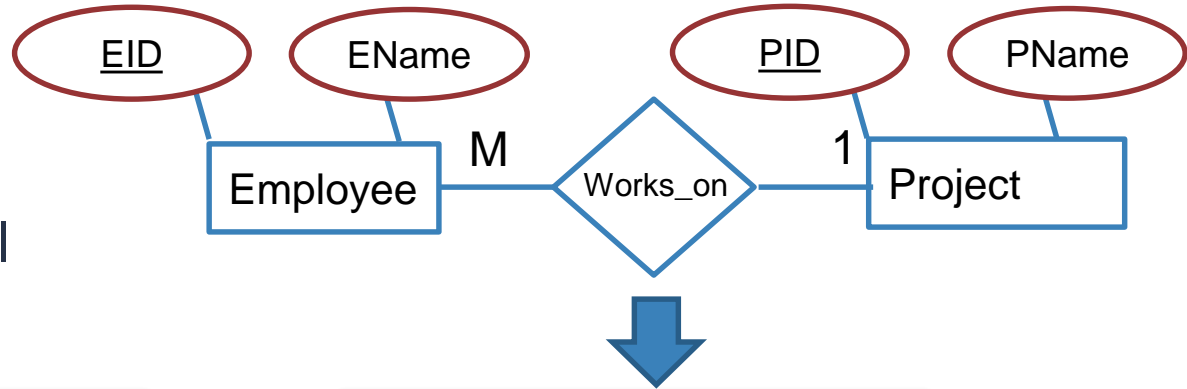
3 tables

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



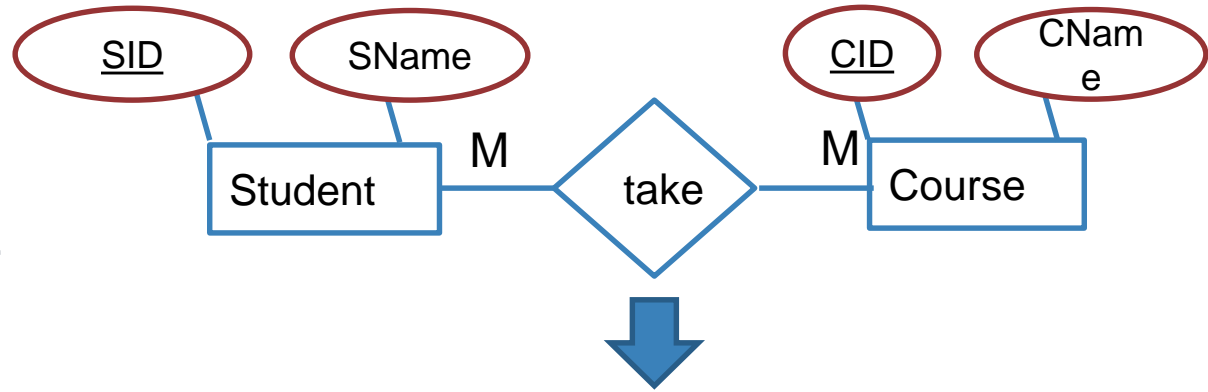
3 tables

tbl_x (PK_x,.....)

tbl_y (PK_y,.....)

tbl_xy (PK_x ,PK_y,,.....)

PK_{xy}=_PK_x+PK_y



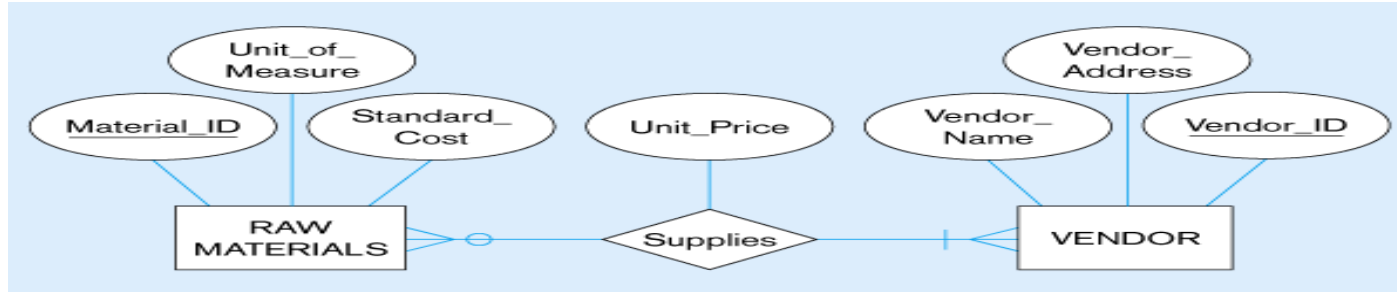
Student(SID, Sname)

Course(CID, Cname)

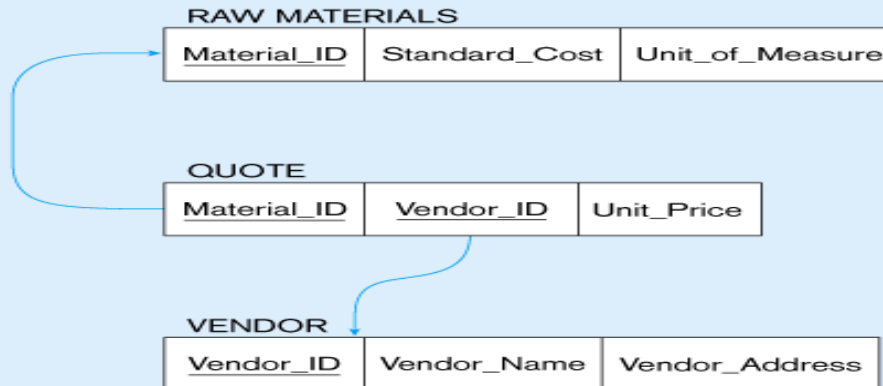
Stud_Course(SID, CID)



Many To Many with attribute



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





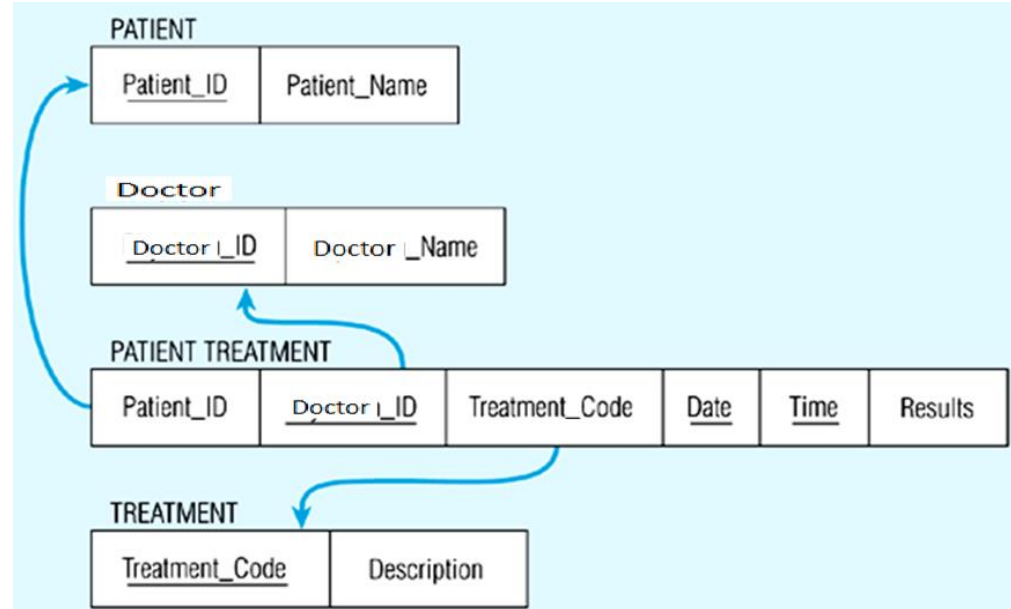
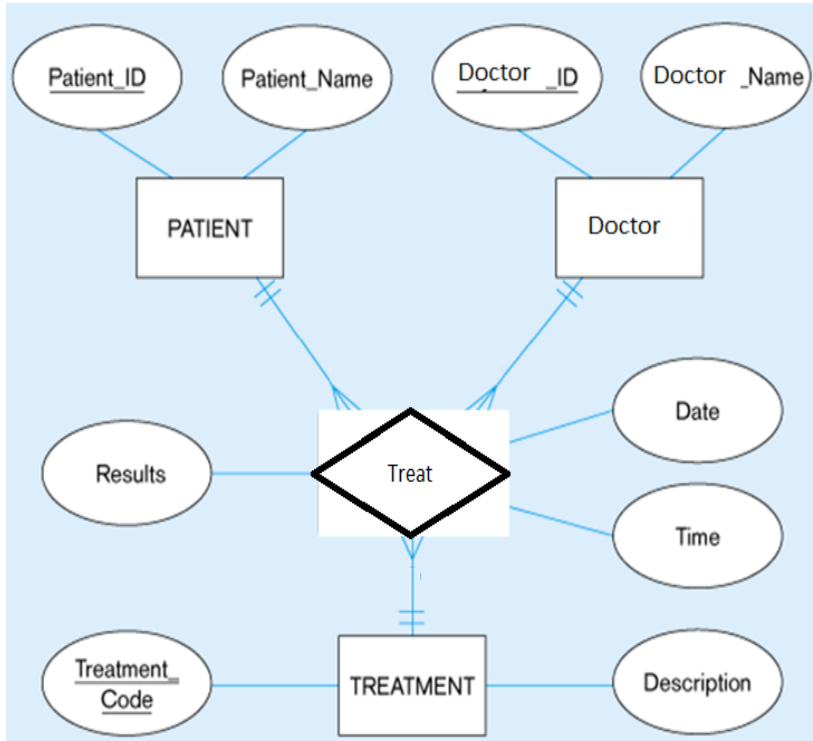
Step 6: Mapping of N-ary Relationship Types.

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 .

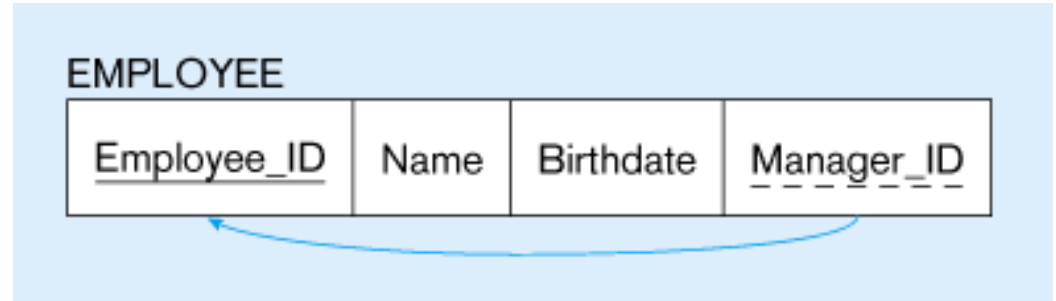
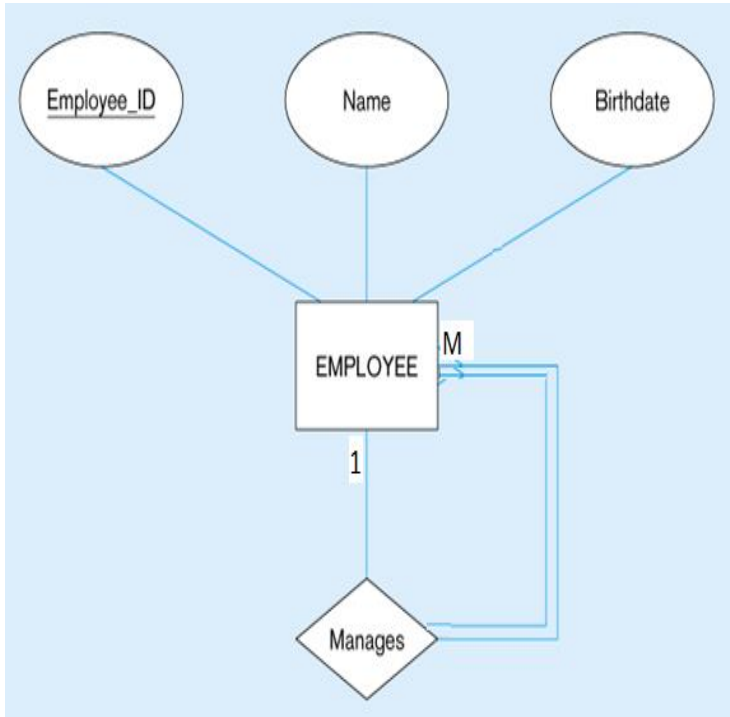


Step 6: Mapping of N-ary Relationship Types.





Step 7: Mapping Unary Relationship



- EMPLOYEE relation with recursive foreign key

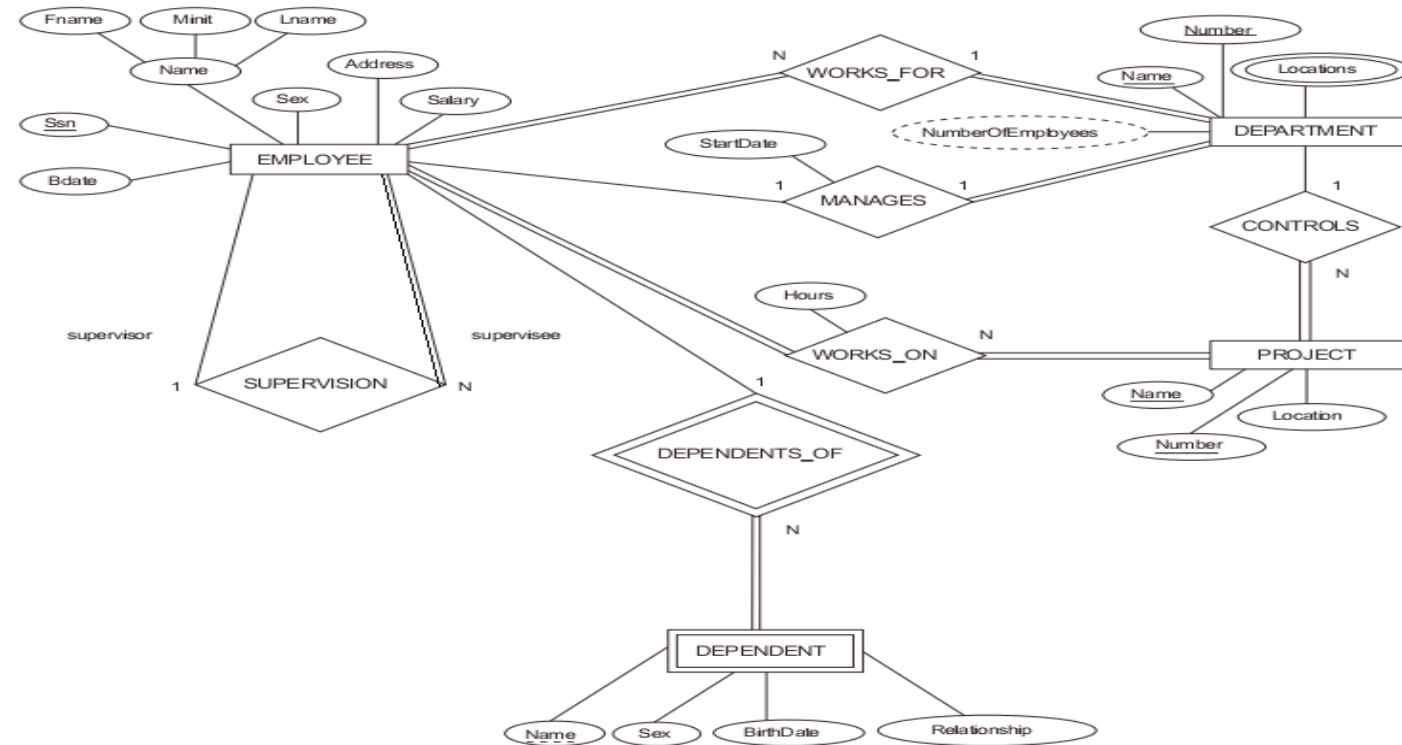


Case Study

Company

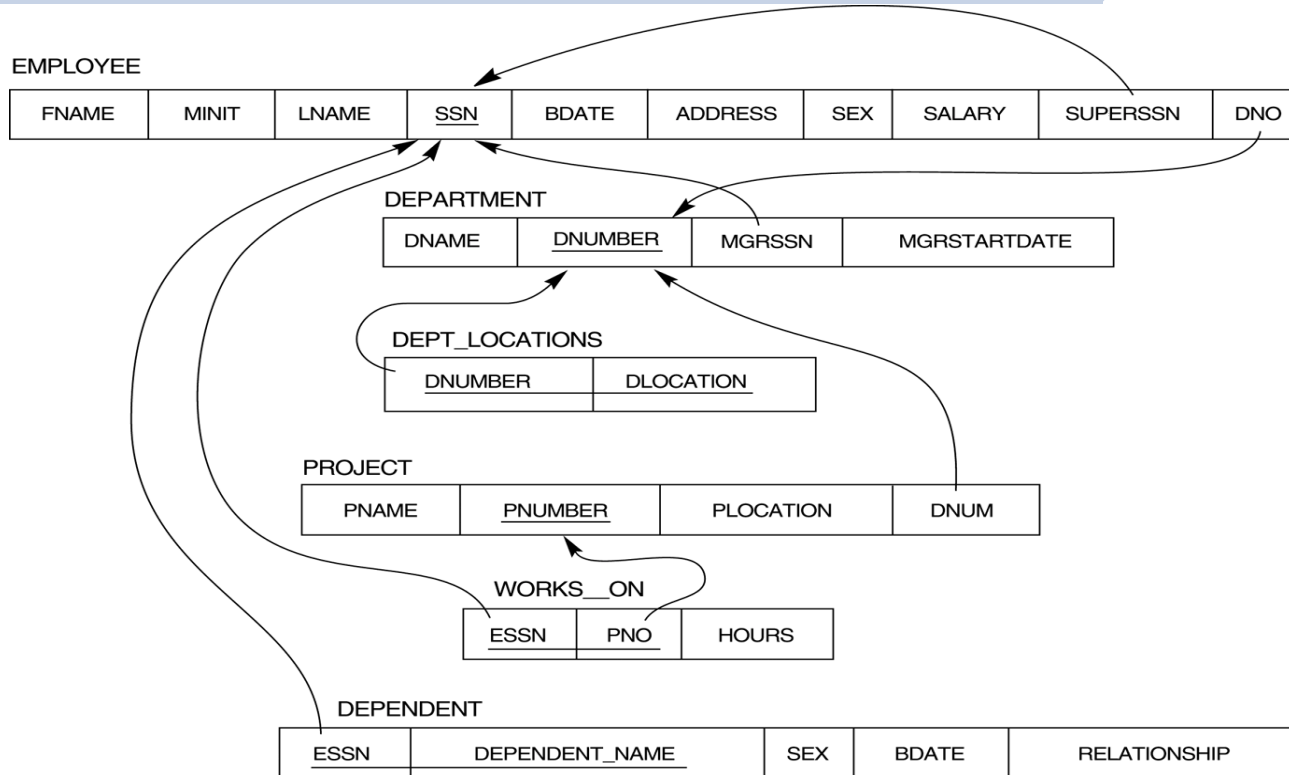


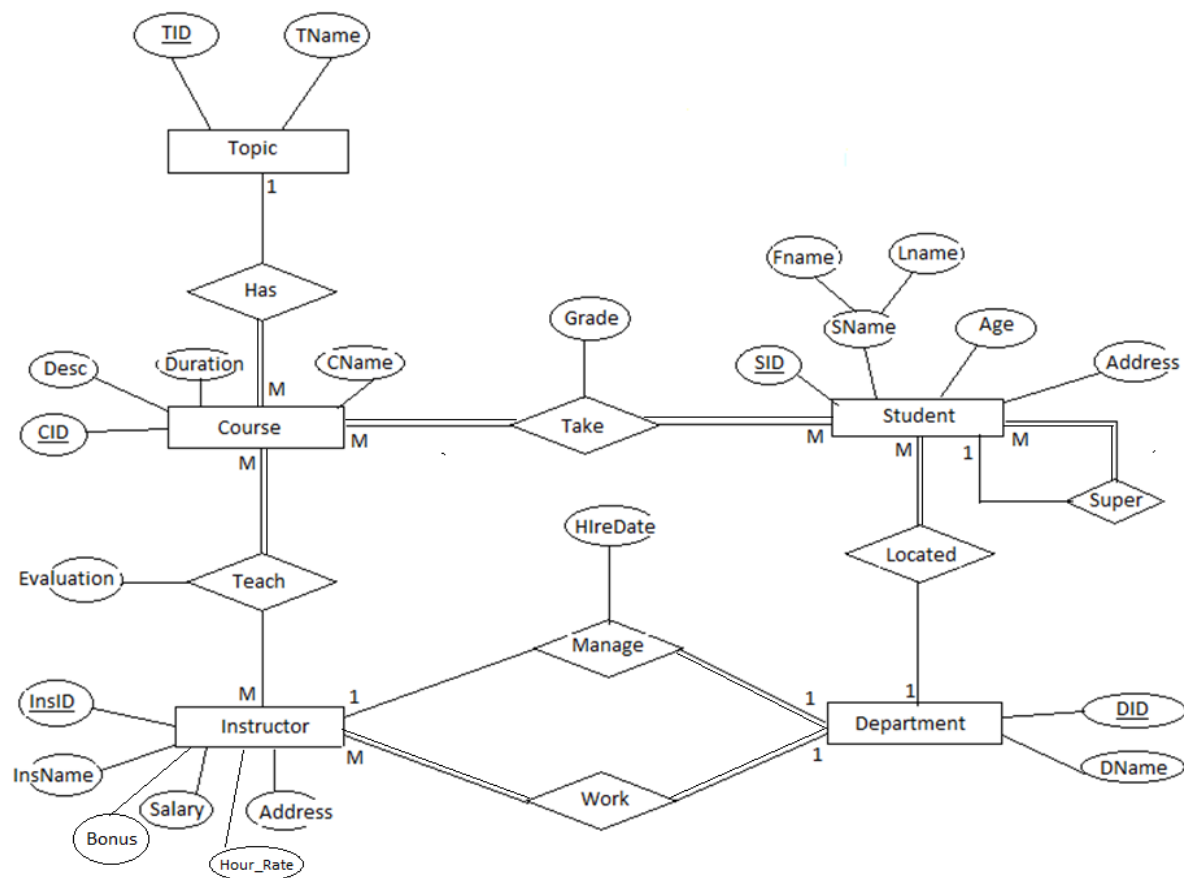
Case study





Result







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**,Evaluation)
- Department(**Dept-ID**,Dept-Name,**Manager-ID**,HireDate)

MySQL

Structured Query Language
SQL





SQL

- ☐ SQL (pronounced "ess-que-el") stands for **Structured Query Language**.
- ☐ SQL is a database computer language designed for the retrieval and management of data in a relational database
- ☐ Developed in the early 1974 (SEQual)
- ☐ ANSI-SQL defined by the American National Standards Institute
- ☐ SQL is a language to operate databases; it includes database creation, deletion, fetching rows, modifying rows, etc.



MySQL

- ❑ **MySQL** is an open source SQL database, which is developed by a Swedish company – MySQL AB. MySQL is pronounced as "my ess-que-ell".
- ❑ MySQL is written in [C](#) and [C++](#). Its SQL parser is written in [yacc](#)
- ❑ MySQL is supporting many different platforms including Microsoft Windows, the major Linux distributions, UNIX, and Mac OS X.
- ❑ MySQL has free and paid versions, depending on its usage (non-commercial/commercial) and features. MySQL comes with a very fast, multi-threaded, multi-user and robust SQL database server.



MySQL History

- Development of MySQL by Michael Widenius & David Axmark beginning in 1994.
- First internal release on 23rd May 1995.
- Windows Version was released on the 8th January 1998 for Windows 95 and NT.
- Sun Microsystems acquired MySQL AB on the 26th February 2008.
- Version 8.0 : production release April 19, 2018



MySQL Features

- High Performance.
- High Availability.
- Robust Transactional Support.
- Strong Data Protection.
- Open Source Freedom and 24 x 7 Support.



Categories of MySQL Statements

- **DML – Data Manipulation Language** :refers to the INSERT, UPDATE and DELETE statements , DML allows to add / modify / delete data itself.
- **DCL – Data Control Language** : refers to the GRANT and REVOKE statements
- **DDL – Data Definition Language** : refers to the CREATE, ALTER and DROP statements , DDL allows to add / modify / delete the logical structures
- **DTL - Data Transaction Language** :refers to the START TRANSACTION, SAVEPOINT, COMMIT and ROLLBACK [TO SAVEPOINT] statements
- **DQL - Data Query Language (Select)** :refers to the SELECT, SHOW and HELP statements (queries)



Data Types

■ A data type determines the type of data that can be stored in a database column. The most commonly used data types are:

1. Alphanumeric: data types used to store characters, numbers, special characters, or nearly any combination.
2. Numeric
3. Date and Time



String Data Types

CHAR(size)	Maximum size of 255 characters.
VARCHAR(size)	Maximum size of 255 characters.
TINYTEXT(size)	Maximum size of 255 characters.
TEXT(size)	Maximum size of 65,535 characters.
MEDIUMTEXT(size)	Maximum size of 16,777,215 characters.
LONGTEXT(size)	Maximum size of 4GB or 4,294,967,295 characters

Varchar vs tinytext !?



Number Data Types

TINYINT	values range from -128 to 127.
SMALLINT	values range from -32768 to 32767
MEDIUMINT	values range from -8388608 to 8388607.
INT	values range from -2147483648 to 2147483647.
BIGINT	values range from -9223372036854775808 to 9223372036854775807
FLOAT(M,D)	7 digits m is the total digits and d is the number of digits after the decimal.
DOUBLE(M,D)	15-16 digits
DECIMAL(m,d)	28-29 significant digits



Date Data Types

DATE	Values range from '1000-01-01' to '9999-12-31'.
DATETIME	Values range from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.
TIME	Values range from '-838:59:59' to '838:59:59'.
YEAR[(2 4)]	Year value as 2 digits or 4 digits.



Database Constraints

- Not Null.
- Primary Key.
- Unique Key.
- Referential Integrity (FK).
- AUTO_INCREMENT



Creating New Database

➤ **PHPMysqlAdmin**

In the database tab write the Database name with no space .

➤ **Console**

```
CREATE DATABASE mydatabasename ;
```

```
CREATE DATABASE IF NOT EXISTS mydatabasename ;
```

```
DROP DATABASE mydatabasename ;
```



CREATING USERS

- Create new user

```
CREATE USER 'username'@'server' IDENTIFIED BY  
'password';
```

- Example

```
CREATE USER 'Admin'@'localhost' IDENTIFIED BY  
'P@ssw0rd';
```



Grant Permission to User

```
GRANT PRIVILEGES ON database.table TO  
'username'@'localhost' IDENTIFIED BY 'password';
```

- PRIVILEGES => create ,drop ,select ,... or ALL PRIVILEGES.

- Example

```
GRANT ALL PRIVILEGES ON *.* TO 'username'@'localhost'  
IDENTIFIED BY 'password';
```

```
GRANT select,update,delete,insert ON iti.* TO 'username'@'localhost'  
IDENTIFIED BY 'password';
```



Revoking Privileges From Users

```
REVOKE privileges ON object FROM user;
```

- PRIVILEGES => create ,drop ,select ,... or ALL , Grant.
- Example

```
REVOKE SELECT, INSERT, DELETE, UPDATE ON *.* from  
'Amit'@'localhost';
```

```
REVOKE ALL ON ITI.* from 'Amit'@'localhost';
```



Create Command

Create table "table_name" ("column name" data type, "column name" data type, ...)

Example (1)

```
CREATE TABLE customer  
(ID int (3) Not Null, First_Name char(50), Last_Name char(50),  
City char(25), Birth_Date date, Primary key (ID)  
FOREIGN KEY (PersonID) REFERENCES Persons(PersonID));
```

Example (2)

```
CREATE TABLE customer  
(ID int (15) Primary key, First_Name char(50), Last_Name  
char(50), City char(25), Birth_Date date);
```



Drop command

Drop table "table name";

✓ Drop table Customer



Alter command

```
ALTER TABLE table_name ADD column_name datatype
```

```
ALTER TABLE table_name DROP COLUMN column_name
```

Example:

- ✓ **ALTER TABLE Customer ADD Address char(40)**
- ✓ **ALTER TABLE Customer DROP COLUMN Address**



General Query

- Show SQL Version :

```
SELECT VERSION();
```

- Show Current Date:

```
SELECT CURRENT_DATE() ;
```

- Use MYSQL as Calculator :

```
SELECT SIN(PI()/4);  
SELECT (4+1)*5;
```

- Show Current Date & Current Time :

```
SELECT NOW();
```

- Show User name :

```
SELECT USER();
```




General Query

- View All Database to use one of them

```
SHOW DATABASES ;
```

- Use Database to work on it

```
USE databasename;
```

- Setting Admin Password

```
SET PASSWORD FOR  
root@localhost=PASSWORD('1234');
```

- To exit the MySQL

```
QUIT , EXIT
```



General Query with Table

- Show table of the database

```
SHOW TABLES;
```

- Show Table Column Information

```
Describe tablename ;  
SHOW COLUMNS FROM tablename;
```

- Show Table Creation Query

```
Show Create Table tablename ;
```