

**Database Management System**

**(DBMS)**

**Lab Manual**

**Name:** Syed Hamza Jaffar Zaidi

**Reg Id:** 52031

**Course:** DBMS (Lab)

**Timing:** Mon/Wed (11:45 – 2:45)

**Program:** BSCS

***CONTENTS***

|  |  |
| --- | --- |
| **Lab No** | **Lab Objective** |
| **1** | **Introduction to Database** |
| **2** | **Database Environment** |
| **3** | **The Relational Data Model** |
| **4** | **Entity Relationship Model** |
| **5** | **SQL - Data Definition Language** |
| **6** | **SQL - Data Manipulation Language** |
| **7** | **SQL – Data Control Language, Transaction Control Language** |
| **8** | **Relational Algebra (Structured Query Language)** |

|  |  |  |
| --- | --- | --- |
| **Lab No** | **Lab Objective** | **Remarks** |
| 1. | **Introduction to Database:**  Objective: Basic Database concepts, Database applications, Database approach vs. file based system, DBMS, Components and Roles in DBMS Environment, History of DB, Advantages and disadvantages |  |
|  |  |  |
|  | **Database Environment:**  Objective: Three level architecture, Data independence, Database languages, Data Models, Network and Hierarchical models, Multi user DBMS Architectures, Teleprocessing, File server and Client server architecture, DBMS Functions | **Assignment#1** |
| 2 |  |  |
| 3 | **The Relational Data Model:**  Objective: Relation, attributes, tuples, domains, Relational schema, Relational database, properties of relational data model, keys of relations, integrity constraints | **Quiz# 1** |
| 4 | **Entity Relationship Model*:***  Objective: Entity and types, Attributes and types, Relationships and types, Keys, attributes on relationships, Entity vs. attributes, Multiplicity, Cardinality, Participation, Entity relationship diagram (ERD) |  |
| 5 | **SQL - Data Definition Language:**  Objective: MySQL data types, Create Table, Alter Table, Add and modify Constraints, Create View, View Resolution, View Materialization, updating  Views |  |
| 6 | **SQL - Data Manipulation Language Advanced:**  Objective: Joins, Types of Join (Inner, Outer, Left, Right, Natural), computing a join, union, intersect and difference, Insert, bulk insert, update,delete | **Assignment #2** |
| 7 | **SQL – Data Control Language, Transaction Control Language*:***  Objective: Create User, Grant, Revoke. | ***Quiz#2*** |
| 8 | The Purpose of this Lab is to get familiar with basic Select Queries which includes Selection & Projection Queries & Some Built-in Functions |  |
|  |  |  |



**LAB # 01**

**The Purpose of this Lab is to get familiar with the File System & File Handling**

**Lab Exercise**

**Question 1:**

**What are the disadvantages of file processing system?**

**Answer:**

Disadvantages of FPS:

1. Slow access time
2. Presence of redundant data
3. Inconsistent Data
4. Data Integrity Problems
5. Difficulty in recovery of corrupt data
6. Lack of Atomicity
7. Problem in Concurrent Access
8. Unauthorized Access



**LAB # 02**

**The Purpose of this Lab is to get familiar with the Relational DBMS Architecture & its Concepts**

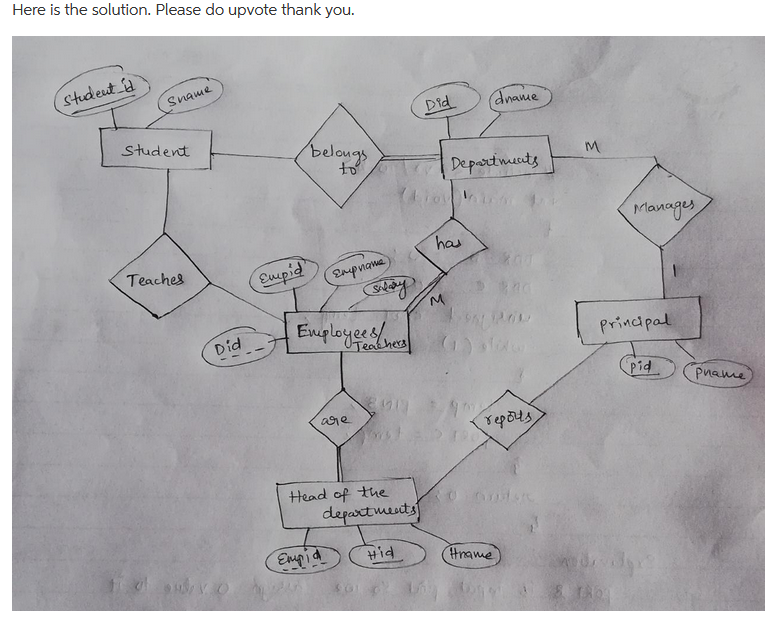
**Lab Exercise**

**Question 1:**

**Modern School of higher education has around 2000 students and three departments. The students belong to various departments in the school. Every department has a Head of Department along with teachers. The head of department manages his departmental teachers and the students belonging to his department. The head of department also teaches the students. All the head of departments report to the Principal of the school. The Principal manages the departments and also teaches the students. Handling the administrative staff of the school is also the responsibility of the Principal. Every individual, except the students, are the employees of the school**

**Problem:**

* **Identify the entity**
* **Identify the entity’s attribute**
* **Identify the primary and foreign Keys**
* **Identify relationship between Entities**
* **Identify the Cardinality constraints**

****

**Question 2:**

1. **a professor teaches zero, one or many classes and a class is taught by one professor**
2. **a course may generate zero, one or many classes and a class comes from one course**
3. **a class is held in one room but a room has many classes**

**Answer:**

**Example-1 Solution:**

The many-to-many relationship is not resolved, therefore the solution is incomplete. In the final solution the many-to-many must always be resolved.



**Second Solution:**

In this example, the many-to-many relationship between student and class is resolved.





**LAB # 03**

**The Purpose of this Lab is to introduce Entity Relationship Diagram & its Notations**

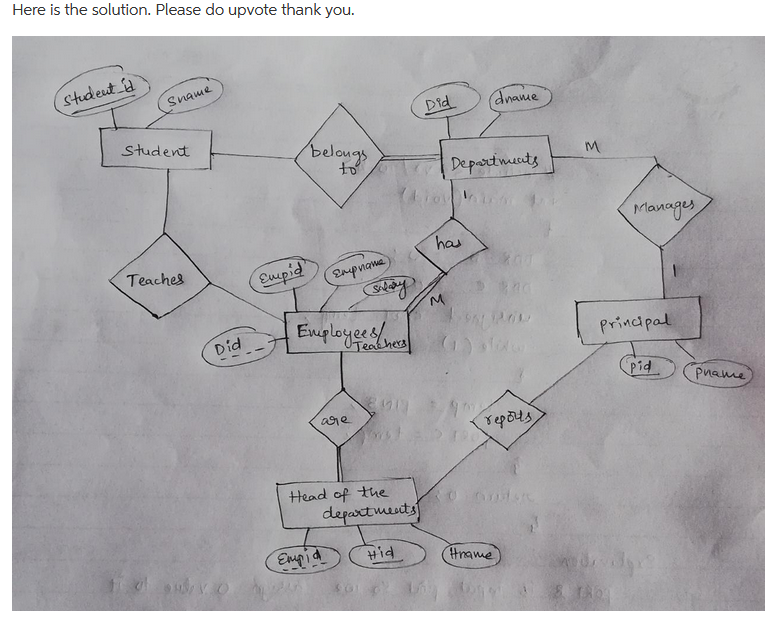
**Exercise**

**Question 1**

**Modern School of higher education has around 2000 students and three departments. The students belong to various departments in the school. Every department has a Head of Department along with teachers. The head of department manages his departmental teachers and the students belonging to his department. The head of department also teaches the students. All the head of departments report to the Principal of the school. The Principal manages the departments and also teaches the students. Handling the administrative staff of the school is also the responsibility of the Principal. Every individual, except the students, are the employees of the school**

* **Identify the entity**
* **Identify the entity’s attribute**
* **Identify the primary and foreign Keys**
* **Identify relationship between Entities**
* **Identify the Cardinality constraints**
* **Draw ERD**

**Answer:**

****

**Question 2**

1. **an invoice is written by one sales rep but a sales rep writes many invoices**
2. **a vendor sells many products but a product is bought from one vendor**
3. **an invoice has one or many products and a product is found on zero, one or many invoices**

* **Identify the entity**
* **Identify the entity’s attribute**
* **Identify the primary and foreign Keys**
* **Identify relationship between Entities**
* **Identify the Cardinality constraints**
* **Draw ERD**

**Answer:**





**LAB # 04**

**The Purpose of this Lab is to make Entity Relationship Diagram by using Case Studies**

**Lab Exercise**

**CASE – STUDY I:**

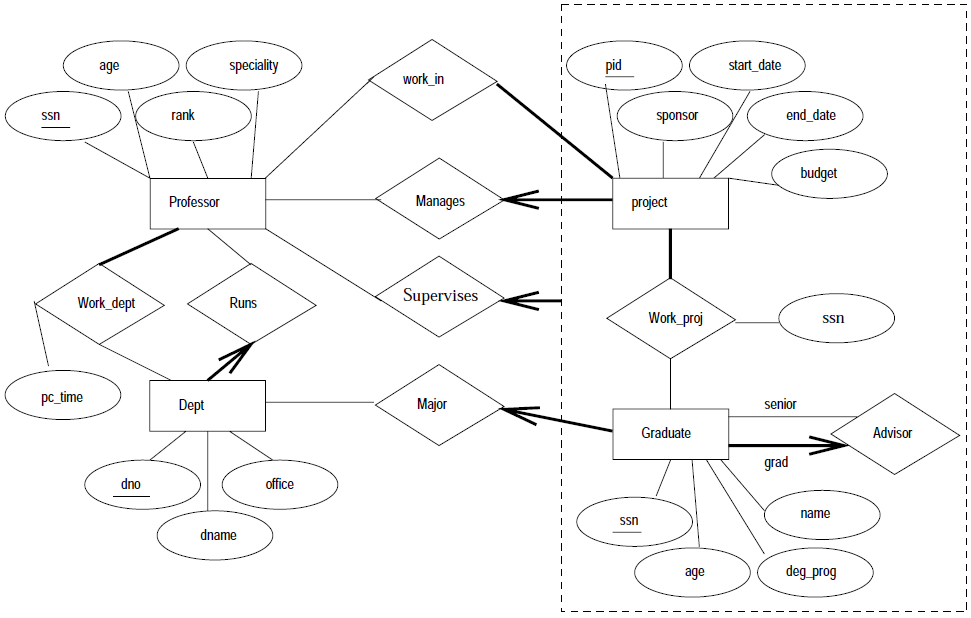
**Consider the following information about a university database:**

* **Professors have an SSN, a name, an age, a rank, and a research specialty.**
* **Projects have a project number, a sponsor name (e.g., NSF), a starting date, an ending date, and a budget.**
* **Graduate students have an SSN, a name, an age, and a degree program (e.g., M.S. or Ph.D.).**
* **Each project is managed by one professor (known as the project’s principal investigator).**
* **Each project is worked on by one or more professors (known as the project’s co-investigators).**
* **Professors can manage and/or work on multiple projects.**
* **Each project is worked on by one or more graduate students (known as the project’s research assistants).**
* **When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one.**
* **Departments have a department number, a department name, and a main office.**
* **Departments have a professor (known as the chairman) who runs the department.**
* **Professor’s work in one or more departments and for each department that they work in, a time percentage is associated with their job.**
* **Graduate students have one major department in which they are working on their degree.**
* **Each graduate student has another, more senior graduate student (known as a student advisor) who advises him or her on what courses to take.**

**Design and draw an ER diagram that captures the information about the university.**

**Use only the basic ER model here; that is, entities, relationships, and attributes. Be**

**Sure to indicate any key and participation constraints.**



**CASE - STUDY –II:**

**The Prescriptions-R-X chain of pharmacies has offered to give you a free lifetime supply of medicine if you design its database. Given the rising cost of health care, you agree. Here’s the information that you gather:**

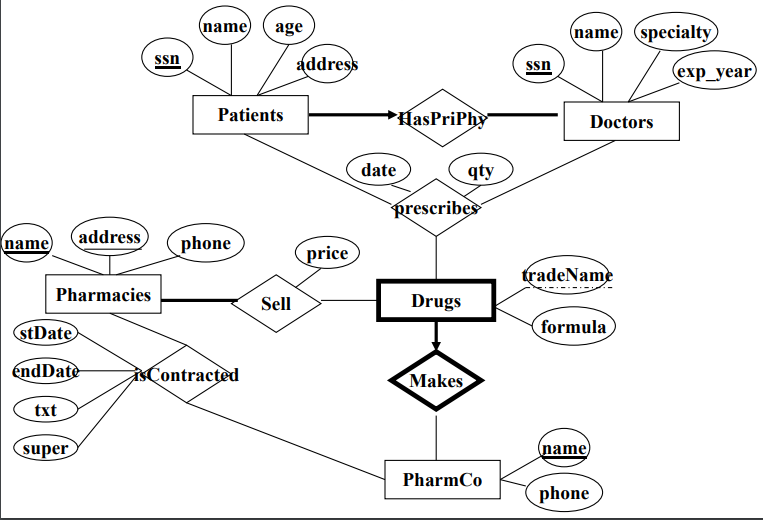
* **Patients are identified by an SSN, and their names, addresses, and ages must be recorded.**
* **Doctors are identified by an SSN. For each doctor, the name, specialty, and years of experience must be recorded.**
* **Each pharmaceutical company is identified by name and has a phone number.**
* **For each drug, the trade name and formula must be recorded. Each drug is sold by a given pharmaceutical company, and the trade name identifies a drug uniquely from among the products of that company. If a pharmaceutical company is deleted, you need not keep track of its products any longer.**
* **Each pharmacy has a name, address, and phone number.**
* **Every patient has a primary physician. Every doctor has at least one patient.**
* **Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.**
* **Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors.**
* **Each prescription has a date and a quantity associated with it. You can assume that, if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored.**
* **Pharmaceutical companies have long-term contracts with pharmacies. A pharmaceutical company can contract with several pharmacies, and a pharmacy can contract with several pharmaceutical companies. For each contract, you have to store a start date, an end date, and the text of the contract.**
* **Pharmacies appoint a supervisor for each contract. There must always be a supervisor for each contract, but the contract supervisor can change over the lifetime of the contract.**

**Draw an ER diagram that captures the preceding information. Identify any constraints not captured by the ER diagram.**

**How would your design change if each drug must be sold at a fixed price by all pharmacies?**

**How would your design change if the design requirements change as follows: If a doctor prescribes the same drug for the same patient more than once, several such prescriptions may have to be stored?**

**Answer:**

****



**LAB # 05**

**The Purpose of this Lab is to introduce the DDL (Data Definition Language) which includes Create, Drop, and Alter & Truncate Statements**

**Lab Exercise**

**Consider the following schema:**

1. **Customer (cust\_id, cust\_name).**
2. **Product (prod\_code ,prod\_name, unit\_price)**
3. **Customer\_Order (order\_code, order\_date, cust\_id)**
4. **Order\_Item (order\_code, prod\_code, num\_of\_units)**
5. **Develop DDL of in SQL**
6. **Add Column contact in Customer Table**
7. **Add Column company\_name in Product Table**

**Code:**

# I. Develop DDL of in SQL

CREATE DATABASE company

CREATE TABLE CUSTOMER (

CUST\_NO VARCHAR (5),

CUST\_NAME VARCHAR (50)

);

ALTER TABLE CUSTOMER

ADD PRIMARY KEY (CUST\_NO)

CREATE TABLE Product (

Prod\_code INT NOT NULL,

Prod\_Name VARCHAR(50),

Prod\_Price INT NOT NULL

);

ALTER TABLE Product

ADD PRIMARY KEY (Prod\_code)

CREATE TABLE Customer\_Order (

Order\_Code INT NOT NULL,

Order\_Date INT NOT NULL,

Cust\_NO INT NOT NULL

);

CREATE TABLE Order\_Item (

Order\_Code INT NOT NULL,

Prod\_code INT NOT NULL,

Num\_of\_units INT NOT NULL

);

# II. Add Column contact in Customer Table

ALTER TABLE Customer

ADD COLUMN Contact INT NOT NULL

ALTER TABLE CUSTOMER

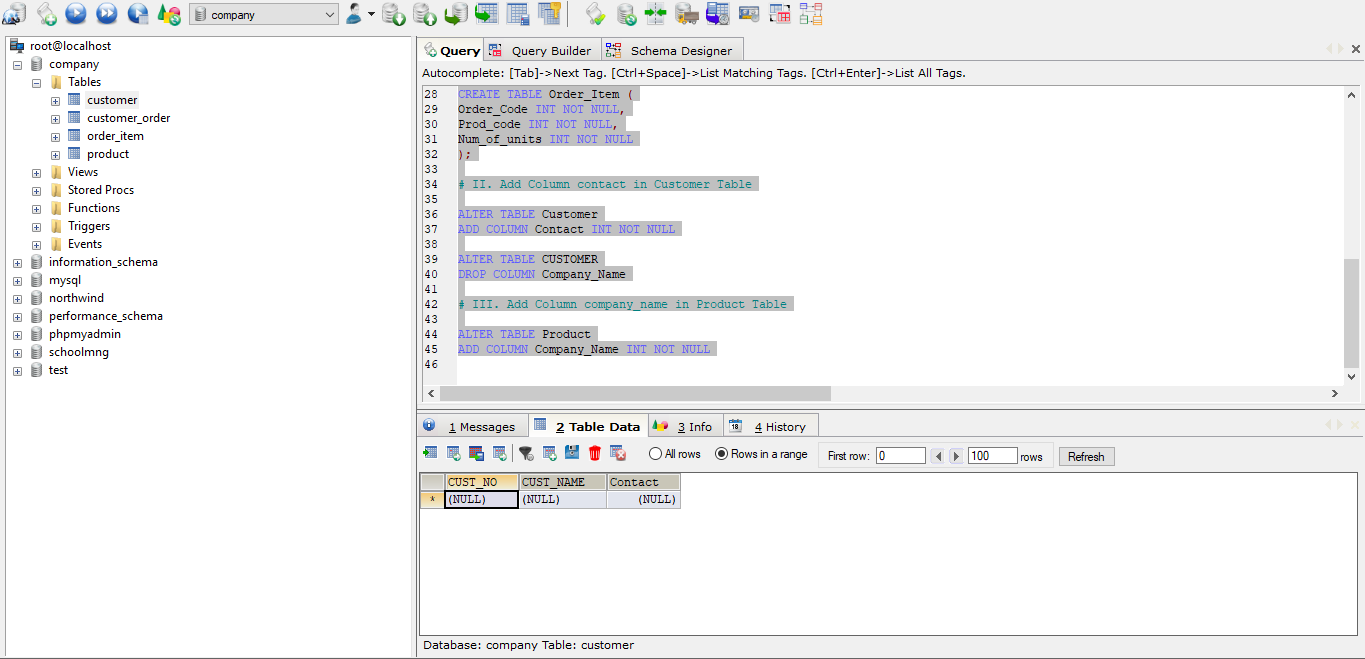
DROP COLUMN Company\_Name

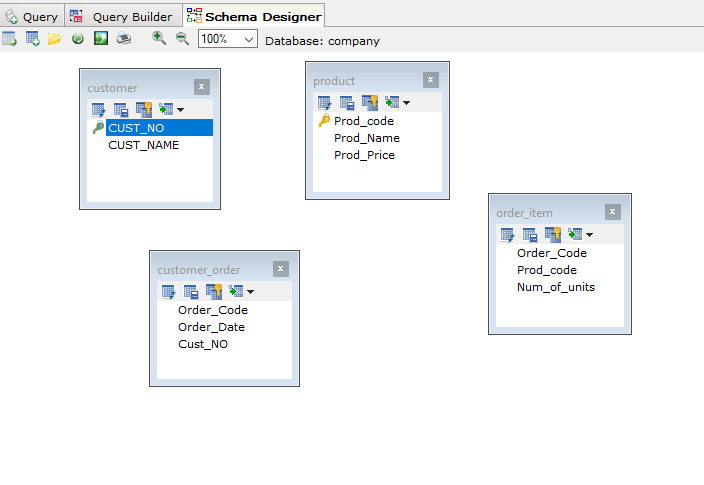
# III. Add Column company\_name in Product Table

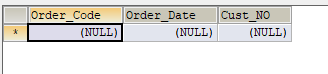
ALTER TABLE Product

ADD COLUMN Company\_Name INT NOT NULL

**Output Screenshots:**

****

****

****



**LAB # 06**

**The Purpose of this Lab is to introduce the DML (Data Manipulation Language) which includes Insert, Update & Delete Statements**

**Lab Exercise**

**Question 1:**

**Insert the following data into the following table**

**Movie (id, title, year, director)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Title** | **Year** | **Director** |
| **1** | **Ben hur** | **2016** | **Thomas** |
| **2** | **Get Smart** | **2012** | **Richard Bell** |
| **3** | **Spider Man** | **2009** | **Tam Morry** |
| **4** | **Batman V/S Super Man** | **2015** | **Gerald Hond** |

**Code:**

# Question # 1:

# Insert the following data into the following table

# Movie (id, title, year, director)

CREATE DATABASE Movies

CREATE TABLE Movie1 (

id INT NOT NULL,

Title VARCHAR (50),

M\_Year INT NOT NULL,

Director VARCHAR (50)

);

INSERT INTO Movie1 (id, Title, M\_Year, Director)

VALUES ('1', 'Ben hur', '2016', 'Thomas');

INSERT INTO Movie1 (id, Title, M\_Year, Director)

VALUES ('2', 'Get Smart', '2012', 'Richard Bell');

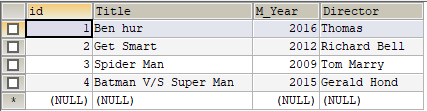
INSERT INTO Movie1 (id, Title, M\_Year, Director)

VALUES ('3', 'Spider Man', '2009', 'Tom Marry');

INSERT INTO Movie1 (id, Title, M\_Year, Director)

VALUES ('4', 'Batman V/S Super Man', '2015', 'Gerald Hond');

**Output Screenshots:**





**LAB # 07**

**The Purpose of this Lab is to introduce the DCL (Data Control Language) which includes Grant & Revoke Statements**

L

**Lab Exercise**

**Question 1: Grant all privilege to user which name Ali.**

**Code:**

SELECT USER,HOST,PASSWORD

FROM mysql.user

CREATE USER 'Ali'@'localhost'

GRANT ALL PRIVILEGES ON schoolmng. \* TO 'Ali'@'localhost';

**Question 2: Grant only creates permission to user which name Fahad.**

**Code:**

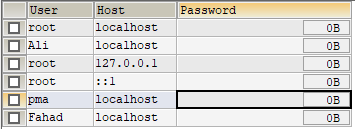
SELECT USER,HOST,PASSWORD

FROM mysql.user

CREATE USER 'Fahad'@'localhost'

GRANT CREATE ANY TABLE TO username

**Output:**

****



**LAB # 08**

**The Purpose of this Lab is to get familiar with basic Select Queries which includes Selection & Projection Queries & Some Built-in Functions**

**Lab Exercise**

|  |  |  |  |
| --- | --- | --- | --- |
| **Emp\_num** | **Emp\_Name** | **Emp\_Job** | **Emp\_Sal** |
| **E101** | **Salman** | **Analyst** | **6000** |
| **E102** | **Bushra** | **Programmer** | **5000** |
| **E103** | **Madiha** | **Web Designer** | **6000** |
| **E104** | **Aiman** | **ERD designer** | **4000** |
| **E105** | **Hameed** | **Web Designer** | **3000** |
| **E106** | **Nini** | **Analyst** | **2500** |
| **E107** | **Imtiaz** | **Web Designer** | **6500** |
| **E108** | **Rashid** | **Programmer** | **4000** |
| **E109** | **Muzzamil** | **ERD designer** | **2000** |

1. **List all employees’ number, employee’s name and jobs from EMP.**
2. **List all employees’ number, employee’s name and jobs from emp whose salaries greater than 5,000.**
3. **List all employees’ number, employee’s name and jobs from emp whose salaries less than 5,000.**
4. **List all employees’ number, employee’s name and jobs from emp whose salaries between 1,000 to 5,000.**
5. **List all employees’ number, employee’s name, jobs and salaries from EMP.**

**Code:**

CREATE DATABASE Employees

CREATE TABLE Employees (

Emp\_No INT NOT NULL AUTO\_INCREMENT,

Emp\_Name VARCHAR (50),

Emp\_Job VARCHAR(50),

Emp\_Salary VARCHAR (50),

PRIMARY KEY(Emp\_No)

);

INSERT INTO Employees (Emp\_No, Emp\_Name, Emp\_Job, Emp\_Salary)

VALUES ('1', 'Salman', 'Analyst', '6000');

INSERT INTO Employees (Emp\_No, Emp\_Name, Emp\_Job, Emp\_Salary)

VALUES ('2', 'Bushra', 'Programmer', '5000');

INSERT INTO Employees (Emp\_No, Emp\_Name, Emp\_Job, Emp\_Salary)

VALUES ('3', 'Madiha', 'Web Designer', '6000');

INSERT INTO Employees (Emp\_No, Emp\_Name, Emp\_Job, Emp\_Salary)

VALUES ('4', 'Aiman', 'ERD Designer', '4000');

INSERT INTO Employees (Emp\_No, Emp\_Name, Emp\_Job, Emp\_Salary)

VALUES ('5', 'Hammed', 'Web Designer', '3000');

INSERT INTO Employees (Emp\_No, Emp\_Name, Emp\_Job, Emp\_Salary)

VALUES ('6', 'Nini', 'Analyst', '2500');

INSERT INTO Employees (Emp\_No, Emp\_Name, Emp\_Job, Emp\_Salary)

VALUES ('7', 'Imtiaz', 'Web Designer', '6500');

INSERT INTO Employees (Emp\_No, Emp\_Name, Emp\_Job, Emp\_Salary)

VALUES ('8', 'Rashid', 'Programmer', '4000');

INSERT INTO Employees (Emp\_No, Emp\_Name, Emp\_Job, Emp\_Salary)

VALUES ('9', 'Muzzamil', 'ERD Designer', '2000');

SELECT Emp\_No, Emp\_Name, Emp\_Job

FROM employees

WHERE emp\_salary > 5000;

SELECT Emp\_No, Emp\_Name, Emp\_Job

FROM employees

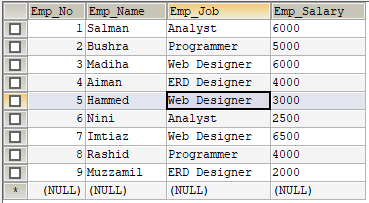
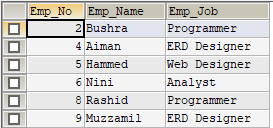
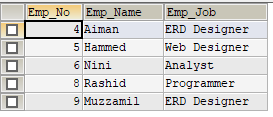
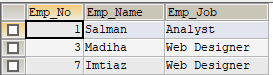
WHERE emp\_salary < 5000;

SELECT Emp\_No, Emp\_Name, Emp\_Job

FROM employees

WHERE emp\_salary BETWEEN 1000 AND 5000;

**Output Screenshots:**

1. **-> List of All the Emp Data**
2. **-> Greater Than 5,000 Salaries Emp Data**
3. **-> Less Than 5,000 Salaries Emp Data**
4. **-> Salaries Between 1,000 to 5,000 Emp Data**