**Lab9**

**1. Perform the following tasks:**

**a. Create Student table with following attributes (STUDENT\_ID , FIRST\_NAME, LAST\_NAME, PHONE\_NUMBER, MARKS, COURSE\_ID).**

**Code-**

CREATE TABLE Course (

COURSE\_ID INT PRIMARY KEY,

COURSE\_NAME VARCHAR(100)

);

CREATE TABLE Student (

STUDENT\_ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(100),

LAST\_NAME VARCHAR(100),

PHONE\_NUMBER VARCHAR(15),

MARKS DECIMAL(5, 2),

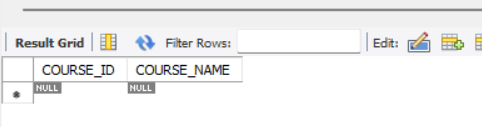
COURSE\_ID INT,

FOREIGN KEY (COURSE\_ID) REFERENCES Course(COURSE\_ID)

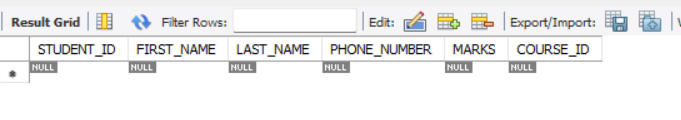
);

**Output-**

**Course table**



**Student table**



**b. Create Course table with following attributes (COURSE\_ID, COURSE\_NAME).**

**CODE-**

INSERT INTO Course (COURSE\_ID, COURSE\_NAME) VALUES

(1, 'Java Programming'),

(2, 'Database Systems'),

(3, 'Web Development'),

(4, 'Data Structures'),

(5, 'Algorithms');

INSERT INTO Student (STUDENT\_ID, FIRST\_NAME, LAST\_NAME, PHONE\_NUMBER, MARKS, COURSE\_ID) VALUES

(1, 'PRATHAM', 'JAGDHANE', '1234567890', 85.5, 1),

(2, 'ATHARVA', 'CHAVAN', '2345678901', 78.0, 2),

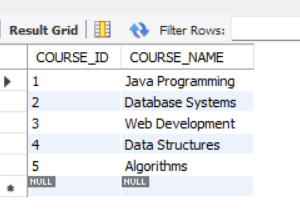
(3, 'PRACHIT', 'GAWAND', '3456789012', 92.5, 1),

(4, 'DEEPAK', 'CHAMOLA', '4567890123', 87.0, 3),

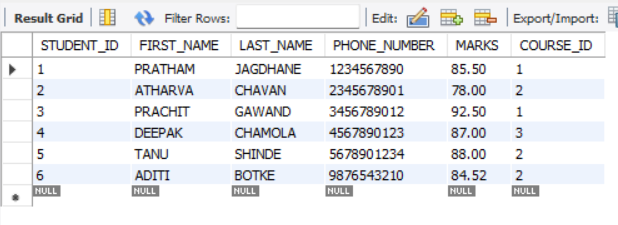
(5, 'TANU', 'SHINDE', '5678901234', 88.0, 2);

**OUTPUT-**

**Course table-**



**Student table-**



**c. Write a SQL statement to insert 8 records with your own value into the tables.**

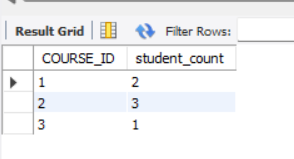
**CODE-**

SELECT COURSE\_ID, COUNT(\*) AS student\_count

FROM Student

GROUP BY COURSE\_ID;

**OUTPUT-**

****

**d. Write a query to get the number of students with the same course.**

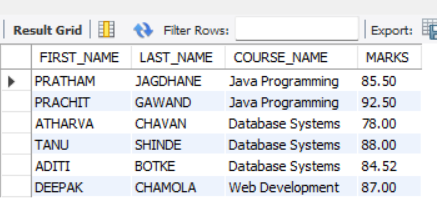
**CODE-**

SELECT s.FIRST\_NAME, s.LAST\_NAME, c.COURSE\_NAME, s.MARKS

FROM Student s

JOIN Course c ON s.COURSE\_ID = c.COURSE\_ID;

**OUTPUT-**



**g. Write a query to get the Average marks of students course wise.**

**Code-**

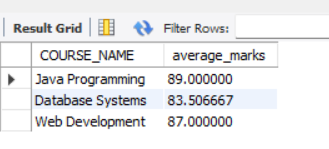
SELECT c.COURSE\_NAME, AVG(s.MARKS) AS average\_marks

FROM Student s

JOIN Course c ON s.COURSE\_ID = c.COURSE\_ID

GROUP BY c.COURSE\_NAME;

**OUTPUT-**



**2.  Create database for hospital management system & Perform the following tasks:**

1. **Create HEALTH CARE WORKERS table with following attributes (EMPLOYEE\_ID , FIRST\_NAME, LAST\_NAME,EMAIL, PHONE\_NUMBER, HIRE\_DATE, SALARY, DESIGNATION).**

**CODE-**

CREATE TABLE HEALTH\_CARE\_WORKERS (

EMPLOYEE\_ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(100),

LAST\_NAME VARCHAR(100),

EMAIL VARCHAR(100),

PHONE\_NUMBER VARCHAR(15),

HIRE\_DATE DATE,

SALARY DECIMAL(10, 2),

DESIGNATION VARCHAR(100)

);

CREATE TABLE PATIENT (

PATIENT\_ID INT PRIMARY KEY,

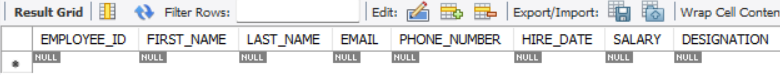
NAME VARCHAR(100),

PHONE\_NUMBER VARCHAR(15)

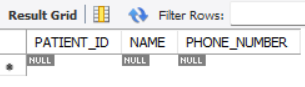
);

**OUTPUT-**

**HEALTH CARE TABLE-**



**PATIENT TABLE-**



1. **Write a SQL statement to insert 10 records with your own value into the tables.**

**CODE-**

INSERT INTO HEALTH\_CARE\_WORKERS (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, SALARY, DESIGNATION) VALUES

(1, 'Alice', 'Johnson', 'alice.johnson@example.com', '1234567890', '2020-01-15', 30000.00, 'Doctor'),

(2, 'Bob', 'Smith', 'bob.smith@example.com', '2345678901', '2019-07-22', 25000.00, 'Nurse'),

(3, 'Carol', 'Williams', 'carol.williams@example.com', '3456789012', '2018-09-30', 27000.00, 'Technician'),

(4, 'David', 'Jones', 'david.jones@example.com', '4567890123', '2021-03-10', 32000.00, 'Doctor'),

(5, 'Eve', 'Brown', 'eve.brown@example.com', '5678901234', '2017-11-25', 28000.00, 'Nurse'),

(6, 'Frank', 'Davis', 'frank.davis@example.com', '6789012345', '2022-01-12', 26000.00, 'Technician'),

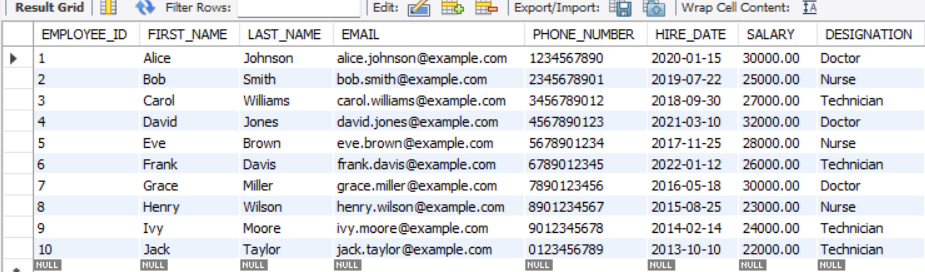
(7, 'Grace', 'Miller', 'grace.miller@example.com', '7890123456', '2016-05-18', 30000.00, 'Doctor'),

(8, 'Henry', 'Wilson', 'henry.wilson@example.com', '8901234567', '2015-08-25', 23000.00, 'Nurse'),

(9, 'Ivy', 'Moore', 'ivy.moore@example.com', '9012345678', '2014-02-14', 24000.00, 'Technician'),

(10, 'Jack', 'Taylor', 'jack.taylor@example.com', '0123456789', '2013-10-10', 22000.00, 'Technician');

**OUTPUT-**



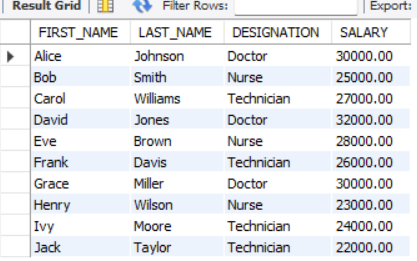
1. **Write a query to get the names (first\_name, last\_name),Designation, salary.**

**CODE-**

SELECT FIRST\_NAME, LAST\_NAME, DESIGNATION, SALARY

FROM HEALTH\_CARE\_WORKERS;

**Output-**



1. **Write a query to get the number of employees with the same Designation**

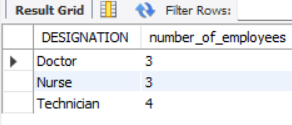
**Code-**

SELECT DESIGNATION, COUNT(\*) AS number\_of\_employees

FROM HEALTH\_CARE\_WORKERS

GROUP BY DESIGNATION;

**Output-**



1. **Write a query to get employee name who are getting salary more than 25000.**

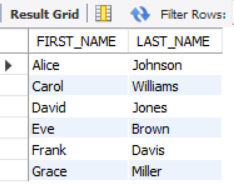
**Code-**

SELECT FIRST\_NAME, LAST\_NAME

FROM HEALTH\_CARE\_WORKERS

WHERE SALARY > 25000;

**Output-**



1. **Fetch HEALTH CARE WORKERS name using their employee id.**

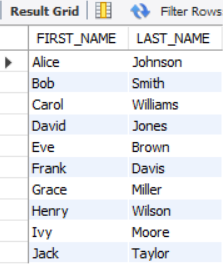
**Code-**

SELECT FIRST\_NAME, LAST\_NAME

FROM HEALTH\_CARE\_WORKERS

WHERE EMPLOYEE\_ID = employee\_id;

**Output-**



**3. Consider two tables, customers and orders, with the following structures:**

**a. Customers Table: customer\_id (Primary Key) first\_name Last\_name**

**code-**

-- Customers Table

CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

first\_name VARCHAR(100),

last\_name VARCHAR(100)

);

-- Orders Table

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_date DATE,

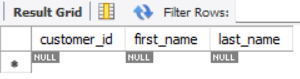
total\_amount DECIMAL(10, 2),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

**Output-**

**Customer table-**



**Order table-**



**b. Orders Table: order\_id (Primary Key) customer\_id (Foreign Key) order\_date Total\_amount**

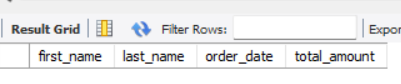
**code-**

SELECT c.first\_name, c.last\_name, o.order\_date, o.total\_amount

FROM Customers c

INNER JOIN Orders o ON c.customer\_id = o.customer\_id;

**Output-**



**4.Consider two tables, departments and employees, with the following structures:**

**Departments Table: department\_id (Primary Key) department\_name**

**Employees Table: employee\_id (Primary Key) first\_name last\_name department\_id (Foreign Key)**

**Write an SQL query to retrieve a list of all departments and the names of employees who belong to each department. Use a LEFT JOIN to include departments that have no employees.**

**Code-**

-- Departments Table

CREATE TABLE Departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(100)

);

-- Employees Table

CREATE TABLE Employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(100),

last\_name VARCHAR(100),

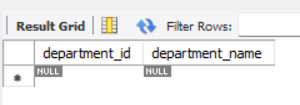
department\_id INT,

FOREIGN KEY (department\_id) REFERENCES Departments(department\_id)

);

**Output-**

**Department table**

****

**Employees table**

