Name: Chandrahasa B

Student code: AF0336567

Batch Code: ANP-C6315

Lab Assignment – 18

DBMS

Question 1: Create a new MySQL database named University.

-> Create a table named Student with the following attributes:

StudentID (Primary Key)

FirstName

LastName

DateOfBirth

Gender

Email

Phone

-> Create a table named Trainer with the following attributes:

TrainerID (Primary Key)

Email

-> add a column TrainerName to Trainer.

-> Display all data of Student table.

-> Display all data of Trainer table.

-> Display only females data from Student table.

-> update phone number set it to Null.

-> Truncate table Student.

-> Drop table Trainer.

Answer:

```
-- Create the University database

CREATE DATABASE IF NOT EXISTS University;

-- Switch to the University database

USE University;

-- Create the Student table

CREATE TABLE IF NOT EXISTS Student (

   StudentID INT PRIMARY KEY,

   FirstName VARCHAR(255),

   LastName VARCHAR(255),

   DateOfBirth DATE,

   Gender VARCHAR(10),

   Email VARCHAR(255),

   Phone VARCHAR(20)

);


-- Create the Trainer table

CREATE TABLE IF NOT EXISTS Trainer (

   TrainerID INT PRIMARY KEY,

   Email VARCHAR(255)

);


-- Add a column TrainerName to Trainer table

ALTER TABLE Trainer ADD COLUMN TrainerName VARCHAR(255);


-- Display all data of Student table
```

```
SELECT * FROM Student;


-- Display all data of Trainer table

SELECT * FROM Trainer;


-- Display only females data from Student table

SELECT * FROM Student WHERE Gender = 'Female';


-- Update phone number and set it to Null

UPDATE Student SET Phone = NULL;


-- Truncate table Student

TRUNCATE TABLE Student;


-- Drop table Trainer

DROP TABLE IF EXISTS Trainer;
```

Question 2: Use the University database.

-> Create a table named Course with the following attributes:

CourseID (Primary Key)

CourseTitle

Stud_id

-> Write a query to retrieve the names of students along with the courses they are enrolled in. Only include records where there is a match in both the Student and Course tables student id. (Inner Join)

-> Write a query to display the names of all courses and, if available, the names of students who are enrolled in each course. Include courses with no students (Left Join)

-> Write a query to retrieve a list of all students, including those who are not enrolled in any courses, along with the names of the courses they are enrolled in (if any) (Right join)

Answer:

```
CREATE TABLE IF NOT EXISTS Course (
   CourseID INT PRIMARY KEY,
   CourseTitle VARCHAR(255),
   Stud_id INT,
   FOREIGN KEY (Stud_id) REFERENCES Student(StudentID)
);
```

1. **Write a query to retrieve the names of students along with the courses they are enrolled in. Only include records where there is a match in both the Student and Course tables student id. (Inner Join):**

   ```
   SELECT Student.FirstName, Student.LastName, Course.CourseTitle
   FROM Student
   INNER JOIN Course ON Student.StudentID = Course.Stud_id;
   ```

2. **Write a query to display the names of all courses and, if available, the names of students who are enrolled in each course. Include courses with no students (Left Join):**

   ```
   SELECT Course.CourseTitle, Student.FirstName, Student.LastName
   FROM Course
   LEFT JOIN Student ON Course.Stud_id = Student.StudentID;
   ```

3. **Write a query to retrieve a list of all students, including those who are not enrolled in any courses, along with the names of the courses they are enrolled in (if any) (Right join):**

   ```
   SELECT Student.FirstName, Student.LastName, Course.CourseTitle
   FROM Student
   RIGHT JOIN Course ON Student.StudentID = Course.Stud_id;
   ```