LAB-02

DDL and DML Commands

1. Questions 1:

(a) Create a Table

 Create a table named Students with the following fields:

StudentID (Primary Key), Name, Age, Class, Email.

(b) Alter a Table

 Add a column PhoneNumber to the Students table.

 Modify the data type of Age to TINYINT.

 Drop the column Email from the Students table.

(c) Drop a Table

 Drop the Students table from the database.

(d) Rename a Table

 Rename the Students table to StudentRecords.

SQL QUERY:

```sql
CREATE DATABASE SchoolDB;
USE SchoolDB;

CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    Name VARCHAR(100),
    Age INT,
    Class VARCHAR(50),
    Email VARCHAR(100)
);
ALTER TABLE Students
ADD PhoneNumber VARCHAR(15);

ALTER TABLE Students
MODIFY Age TINYINT;

ALTER TABLE Students
DROP COLUMN Email;

DROP TABLE Students;

RENAME TABLE Students TO StudentRecords;
```

✗	1	09:54:23	CREATE DATABASE SchoolDB	Error Code: 1007. Can't create database 'schooldb'; database exists
✓	2	09:54:26	USE SchoolDB	0 row(s) affected
✓	3	09:54:28	CREATE TABLE Students (StudentID INT PRIMARY KEY, Name VARCHAR(100), Age INT, Class VARCHAR(50), Email ...	0 row(s) affected
✓	4	09:54:29	ALTER TABLE Students ADD PhoneNumber VARCHAR(15)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓	5	09:54:31	ALTER TABLE Students MODIFY Age TINYINT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓	6	09:54:32	ALTER TABLE Students DROP COLUMN Email	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓	7	09:54:34	DROP TABLE Students	0 row(s) affected
✗	8	09:54:35	RENAME TABLE Students TO StudentRecords	Error Code: 1146. Table 'schooldb.students' doesn't exist

2. Question 2:

(a) Create a Table with Constraints

o Create a table named Courses with the following fields:

CourseID (Primary Key), CourseName (Unique), Credits (Default value 3),

InstructorID (Foreign Key referencing Instructors table).

(b) Check Constraint
o Create a table Employees with a check constraint to ensure the Salary is greater than 5000.
(c) Unique Constraint
o Add a unique constraint on the Email column of the Employees table.

```sql
CREATE DATABASE IF NOT EXISTS UniversityDB;
USE UniversityDB;

DROP TABLE IF EXISTS Instructors;
CREATE TABLE Instructors (
    InstructorID INT PRIMARY KEY,
    Name VARCHAR(100),
    Department VARCHAR(50)
);
CREATE TABLE Courses (
    CourseID INT PRIMARY KEY,
    CourseName VARCHAR(100) UNIQUE,
    Credits INT DEFAULT 3,
    InstructorID INT,
    FOREIGN KEY (InstructorID) REFERENCES Instructors(InstructorID)
);
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(100),
    Salary DECIMAL(10, 2),
    Email VARCHAR(100) UNIQUE,
    CHECK (Salary > 5000)
);
ALTER TABLE Employees
ADD CONSTRAINT Unique_Email UNIQUE (Email);
```

#	Time	Action	Message
1	09:59:54	CREATE DATABASE IF NOT EXISTS UniversityDB	1 row
2	09:59:56	USE UniversityDB	0 row
3	10:00:01	DROP TABLE IF EXISTS Instructors	0 row
4	10:00:02	CREATE TABLE Instructors (InstructorID INT PRIMARY KEY, Name VARCHAR(100), Department VARCHAR(50))	0 row
5	10:00:04	CREATE TABLE Courses (CourseID INT PRIMARY KEY, CourseName VARCHAR(100) UNIQUE, Credits INT DEFAULT 3, I...	0 row
6	10:00:07	CREATE TABLE Employees (EmployeeID INT PRIMARY KEY, Name VARCHAR(100), Salary DECIMAL(10, 2), Email VARC...	0 row
7	10:00:09	ALTER TABLE Employees ADD CONSTRAINT Unique_Email UNIQUE (Email)	0 row

3. Question 3:
(a) Design and Create Tables for a Library Database
- Create tables Books, Members, and Transactions with appropriate fields and relationships.

(b) Implement Referential Integrity
- Create a Department table and an Employee table.

Ensure referential integrity such that every Employee belongs to a valid Department.
(c) Create a Table with a Composite Primary Key
- Create a Marks table with a composite primary key on StudentID and SubjectCode.

(d) Perform DDL Statements for Migration
- Write DDL commands to copy the structure of a table OldData into a new table NewData without copying its data.

```sql
CREATE DATABASE IF NOT EXISTS LibraryDB ;
USE LibraryDB;
CREATE TABLE Books (
    BookID INT PRIMARY KEY,
    Title VARCHAR(200),
    Author VARCHAR(100),
    Publisher VARCHAR(100),
    YearPublished YEAR,
    ISBN VARCHAR(20) UNIQUE
);
CREATE TABLE Members (
    MemberID INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(200),
    PhoneNumber VARCHAR(15),
    MembershipDate DATE
);
CREATE TABLE Transactions (
    TransactionID INT PRIMARY KEY,
    BookID INT,
    MemberID INT,
    IssueDate DATE,
    ReturnDate DATE,
    FOREIGN KEY (BookID) REFERENCES Books(BookID),
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID)
);

CREATE TABLE Department (
    DepartmentID INT PRIMARY KEY,
    DepartmentName VARCHAR(100) UNIQUE
);
CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(100),
    Salary DECIMAL(10, 2),
    DepartmentID INT,
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
```