

DATABASE LAB

BSIT-F23

In this lab, you will use the SQL statements **CREATE**, **ALTER**, and **DROP** as described in this document. Additionally, you are expected to apply the concepts of **INSERT**, **DELETE**, and **UPDATE** statements, which were covered in the previous lab. Make sure to review those concepts before starting this lab.

After thoroughly understanding the provided explanations and examples, proceed to write and execute the SQL queries for the questions given at the end of this document.

CREATE STATEMENT:

The CREATE TABLE statement is fundamental in SQL and allows you to define a table's structure, including its columns, data types, and constraints.

BASIC SYNTAX:

```
CREATE TABLE table_name (  
    column1 datatype [constraint],  
    column2 datatype [constraint],  
    ...  
    ColumnN datatype [constraint]  
);
```

- **table_name**: The name of the table.
- **column1, column2, ...**: The names of the columns in the table.
- **datatype**: The type of data that the column will store (e.g., INT, VARCHAR, DATE, etc.).
- **constraint**: Optional rules applied to a column (e.g., NOT NULL, PRIMARY KEY, FOREIGN KEY, etc.).

EXAMPLE:

```
CREATE TABLE Employees (
```

```
EmployeeID INT PRIMARY KEY, -- Unique ID for each employee
Name VARCHAR(50) NOT NULL -- Employee's name, cannot be NULL
);
```

ALTER STATEMENT:

The ALTER TABLE statement in SQL allows you to modify the structure of an existing table. It can be used to add, delete, or modify columns, as well as to add or drop constraints.

BASIC SYNTAX:

```
ALTER TABLE table_name
{
    ADD column_name datatype [constraint],          -- Add a new column
    DROP COLUMN column_name,                        -- Delete an existing column
    MODIFY COLUMN column_name datatype [constraint],-- Modify an existing
column
    ADD CONSTRAINT constraint_name constraint,      -- Add a new constraint
    DROP CONSTRAINT constraint_name                -- Drop an existing constraint
};
```

- **table_name:** The name of the table to be altered.
- **column_name:** The name of the column to add, delete, or modify.
- **constraint_name:** The name of the constraint to be added or removed.
- **datatype:** The data type for the new or modified column.

EXAMPLE:

- ALTER TABLE Employees --Adding a new column

ADD Age INT;

- ALTER TABLE Employees --Deleting an existing column
DROP COLUMN Age;
- ALTER TABLE Employees --Modifying a column
MODIFY COLUMN Name VARCHAR(100);
- ALTER TABLE Employees -- Dropping a constraint
DROP CONSTRAINT chk_Age; ETC..
- ALTER TABLE Students ADD CONSTRAINT chk_Age CHECK (Age >= 18); --For check statements

DROP STATEMENT:

The DROP statement in SQL is used to delete objects from the database, such as tables, views, databases, or constraints. Once an object is dropped, it is permanently removed, and all the data stored in it is lost.

EXAMPLE: DROP TABLE Employees;

QUESTIONS:

1. **Create a table** Students with the following columns:
 - StudentID (integer, primary key)
 - Name (text, cannot be null)
 - Age (integer)
2. **Insert** two records into the Students table with the following data:
 - StudentID: 1, Name: 'John', Age: 20
 - StudentID: 2, Name: 'Emma', Age: 22
3. **Select** all records from the Students table.
4. **Delete** a student record with StudentID = 1 from the Students table.
5. **Add** a new column Gender (VARCHAR(10)) to the Students table using the ALTER TABLE statement.

6. **Modify** the Name column in the Students table to allow up to 100 characters using the ALTER TABLE statement.
7. **Insert** a new record into the Students table with the following data:
StudentID: 3, Name: 'Liam', Age: 19, Gender: 'Male'
8. **Select** only the Name and Age columns from the Students table.
9. **Drop** the Gender column from the Students table.
10. **Create** a table Courses with the following columns:
CourseID (integer, primary key)
CourseName (text, cannot be null)
Credits (integer)
11. **Insert** three records into the Courses table with the following data:
(1, 'Mathematics', 4)
(2, 'Physics', 3)
(3, 'Chemistry', 4)
12. **Delete** all records from the Courses table where Credits is less than 4.
13. **Select** the Name of students who are 20 years old from the Students table.
14. **Drop** the Courses table completely from the database.
15. **Add a constraint** to the Students table to ensure that the Age column cannot be less than 18 using the ALTER TABLE statement.