Module 1: Introduction to the `ALTER` Statement

Definition:
The `ALTER` statement is used to modify existing database structures like tables, columns, constraints, and indexes.

Example (Checking Table Structure Before and After Alteration):

```
-- Check table structure (MS SQL & MySQL)
DESC Customers;
```

Module 2: Altering Tables – Adding and Removing Columns

Adding a Column

MS SQL Server:

```
ALTER TABLE Customers ADD Age INT;
```

MySQL:

```
ALTER TABLE Customers ADD COLUMN Age INT AFTER LastName;
```

Removing a Column

MS SQL Server:

```
ALTER TABLE Customers DROP COLUMN Age;
```

MySQL:

```
ALTER TABLE Customers DROP COLUMN Age;
```

Module 3: Modifying Column Data Types and Constraints

Changing Column Data Type

MS SQL Server:

```
ALTER TABLE Customers ALTER COLUMN Age BIGINT;
```

MySQL:

```
ALTER TABLE Customers MODIFY Age BIGINT;
```

Adding and Removing `NOT NULL` Constraints

MS SQL Server:

```
ALTER TABLE Customers ALTER COLUMN Age INT NOT NULL;
```

MySQL:

```
ALTER TABLE Customers MODIFY Age INT NOT NULL;
```

Module 4: Renaming Columns and Tables

Renaming a Column

MS SQL Server:

```
EXEC sp_rename 'Customers.Age', 'CustomerAge', 'COLUMN';
```

MySQL:

```
ALTER TABLE Customers RENAME COLUMN Age TO CustomerAge;
```

Renaming a Table

MS SQL Server:

```
EXEC sp_rename 'Customers', 'ClientList';
```

MySQL:

```
ALTER TABLE Customers RENAME TO ClientList;
```

Module 5: Adding and Removing Constraints

Adding a Primary Key

MS SQL Server:

```
ALTER TABLE Customers ADD CONSTRAINT PK_Customer PRIMARY KEY (CustomerID);
```

MySQL:

```
ALTER TABLE Customers ADD PRIMARY KEY (CustomerID);
```

Removing a Primary Key

MS SQL Server:

```
ALTER TABLE Customers DROP CONSTRAINT PK_Customer;
```

MySQL:

```
ALTER TABLE Customers DROP PRIMARY KEY;
```

Adding a Foreign Key

MS SQL Server:

```
ALTER TABLE Orders ADD CONSTRAINT FK_Customer FOREIGN KEY (CustomerID) REFERENCES
Customers(CustomerID);
```

MySQL:

```
ALTER TABLE Orders ADD CONSTRAINT FK_Customer FOREIGN KEY (CustomerID) REFERENCES
Customers(CustomerID);
```

Removing a Foreign Key

MS SQL Server:

```
ALTER TABLE Orders DROP CONSTRAINT FK_Customer;
```

MySQL:

```
ALTER TABLE Orders DROP FOREIGN KEY FK_Customer;
```

-->

Sample Table: `Employees`

To start, create a sample table to practice `ALTER` operations:

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Age INT,
    DepartmentID INT,
    HireDate DATE
);
```

** Hands-on Exercises and Case Studies**

Exercise 1: Modify the Schema of an Employee Database

1. Add a column `Salary` to the `Employees` table.
2. Rename the column `Salary` to `MonthlySalary`.
3. Change the data type of `MonthlySalary` to `DECIMAL(10,2)`.
4. Add a `NOT NULL` constraint to `MonthlySalary`.
5. Add a foreign key from `Employees` to a `Departments` table.

Solution (MS SQL Server & MySQL):

```
ALTER TABLE Employees ADD Salary DECIMAL(10,2);
ALTER TABLE Employees RENAME COLUMN Salary TO MonthlySalary;
ALTER TABLE Employees MODIFY MonthlySalary DECIMAL(10,2) NOT NULL;
ALTER TABLE Employees ADD CONSTRAINT FK_Dept FOREIGN KEY (DeptID) REFERENCES
Departments(DeptID);
```

🛠️ Practice Questions on ALTER TABLE

📌 Section 1: Adding and Removing Columns (5 Questions)

Q1: Add a new column `Salary` (DECIMAL(10,2)) to the `Employees` table.

```
ALTER TABLE Employees ADD Salary DECIMAL(10,2);
```

Q2: Remove the `Age` column from the `Employees` table.

```
ALTER TABLE Employees DROP COLUMN Age;
```

Q3: Add a column `Email` (VARCHAR(100)) after `LastName` (only in MySQL).

```
ALTER TABLE Employees ADD COLUMN Email VARCHAR(100) AFTER LastName;
```

Q4: Add a `PhoneNumber` column (VARCHAR(15)) and make it unique.

```
ALTER TABLE Employees ADD PhoneNumber VARCHAR(15) UNIQUE;
```

Q5: Remove the `PhoneNumber` column from the `Employees` table.

```
ALTER TABLE Employees DROP COLUMN PhoneNumber;
```

📌 Section 2: Modifying Column Data Types and Constraints (5 Questions)

Q6: Change the data type of `Salary` to `FLOAT`.

```
ALTER TABLE Employees ALTER COLUMN Salary FLOAT;
```

(For MySQL: `ALTER TABLE Employees MODIFY Salary FLOAT;`)

Q7: Make `LastName` column **NOT NULL**.

```
ALTER TABLE Employees ALTER COLUMN LastName VARCHAR(50) NOT NULL;
```

(For MySQL: `ALTER TABLE Employees MODIFY LastName VARCHAR(50) NOT NULL;`)

Q8: Change `HireDate` to `DATETIME` instead of `DATE`.

```
ALTER TABLE Employees ALTER COLUMN HireDate DATETIME;
```

Q9: Increase the `FirstName` column size from 50 to 100.

```
ALTER TABLE Employees ALTER COLUMN FirstName VARCHAR(100);
```

Q10: Set a default value of `10000` for `Salary`.

```
ALTER TABLE Employees ADD CONSTRAINT DF_Salary DEFAULT 10000 FOR Salary;
```

(MySQL: `ALTER TABLE Employees ALTER Salary SET DEFAULT 10000;`)

📌 Section 3: Renaming Columns and Tables (5 Questions)

Q11: Rename the column `FirstName` to `EmpFirstName`.

```
EXEC sp_rename 'Employees.FirstName', 'EmpFirstName', 'COLUMN';
```

(MySQL: `ALTER TABLE Employees RENAME COLUMN FirstName TO EmpFirstName;`)

Q12: Rename the table `Employees` to `Staff`.

```
EXEC sp_rename 'Employees', 'Staff';
```

(MySQL: `ALTER TABLE Employees RENAME TO Staff;`)

Q13: Rename the column `HireDate` to `JoiningDate`.

```
ALTER TABLE Employees RENAME COLUMN HireDate TO JoiningDate;
```

Q14: Rename the column `LastName` to `Surname`.

```
EXEC sp_rename 'Employees.LastName', 'Surname', 'COLUMN';
```

Q15: Rename `Salary` to `MonthlySalary`.

```
ALTER TABLE Employees RENAME COLUMN Salary TO MonthlySalary;
```

📌 Section 4: Adding and Removing Constraints (5 Questions)

Q16: Add a `PRIMARY KEY` to the `EmployeeID` column.

```
ALTER TABLE Employees ADD CONSTRAINT PK_Employee PRIMARY KEY (EmployeeID);
```

Q17: Remove the `PRIMARY KEY` from `EmployeeID`.

```
ALTER TABLE Employees DROP CONSTRAINT PK_Employee;
```

Q18: Add a `FOREIGN KEY` constraint on `DepartmentID` referencing `Departments(DepartmentID)`.

```
ALTER TABLE Employees ADD CONSTRAINT FK_Department FOREIGN KEY (DepartmentID)
REFERENCES Departments(DepartmentID);
```

Q19: Remove the `FOREIGN KEY` constraint on `DepartmentID`.

```
ALTER TABLE Employees DROP CONSTRAINT FK_Department;
```

Q20: Add a `CHECK` constraint to ensure `Salary` is greater than 5000.

```sql
ALTER TABLE Employees ADD CONSTRAINT CHK_Salary CHECK (Salary > 5000);
```
