**Slide 1: Introduction to SQL Table Operations**

**Agenda**

* Creating Tables
* Inserting Data into Tables
* Updating Data in Tables
* Deleting Data from Tables

**Slide 2: Creating Tables**

**What is a Table in SQL?**

* A table is a collection of data organized into rows and columns.
* Each column in a table has a specific data type (e.g., integer, varchar).
* Tables are the core storage structures in a database.

**SQL Syntax: CREATE TABLE**

CREATE TABLE TableName (

Column1 DataType Constraints,

Column2 DataType Constraints,

...

);

**Example: Creating a 'Customers' Table**

CREATE TABLE Customers

(

CustomerID INTEGER PRIMARY KEY IDENTITY(1,1),

CustomerName VARCHAR(50),

ContactName VARCHAR(50),

Address VARCHAR(50),

City VARCHAR(20),

PostalCode VARCHAR(10),

Country VARCHAR(15)

);

**Slide 3: Explanation of CREATE TABLE**

**Key Points:**

* **CustomerID**: This is an INTEGER column that uniquely identifies each customer, using PRIMARY KEY constraint and IDENTITY(1,1) to auto-increment the ID starting from 1.
* **CustomerName, ContactName, Address**: These are VARCHAR columns, meaning they store variable-length character strings.
* **City, PostalCode, Country**: Additional details about the customer’s location.

**Slide 4: Inserting Data into Tables**

**What is INSERT INTO?**

* The INSERT INTO statement is used to add new rows to a table.
* We specify the table name and the values for the columns.

**SQL Syntax: INSERT INTO**

INSERT INTO TableName (Column1, Column2, ...)

VALUES (Value1, Value2, ...);

**Example: Inserting Data into the 'Customers' Table**

INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('John Doe', 'Jane Smith', '123 Elm St', 'Springfield', '12345', 'USA');

**Slide 5: Explanation of INSERT INTO**

**Key Points:**

* We’re not inserting a value for CustomerID because it's auto-generated due to the IDENTITY(1,1) constraint.
* The VALUES clause includes the values for each column in the order specified.

**Slide 6: Updating Data in Tables**

**What is UPDATE?**

* The UPDATE statement modifies existing rows in a table.
* It can be used to change values in one or more columns for a specific row or multiple rows.

**SQL Syntax: UPDATE**

UPDATE TableName

SET Column1 = Value1, Column2 = Value2, ...

WHERE Condition;

**Example: Updating a Record in 'Customers' Table**

UPDATE Customers

SET City = 'New York'

WHERE CustomerID = 1;

**Slide 7: Explanation of UPDATE**

**Key Points:**

* The SET clause specifies the columns and new values to update.
* The WHERE clause identifies the specific rows to update (in this case, CustomerID = 1).

**Slide 8: Deleting Data from Tables**

**What is DELETE?**

* The DELETE statement removes one or more rows from a table.
* Be cautious when using DELETE without a WHERE clause—it will remove all rows from the table!

**SQL Syntax: DELETE**

DELETE FROM TableName

WHERE Condition;

**Example: Deleting a Record from the 'Customers' Table**

DELETE FROM Customers

WHERE CustomerID = 1;

**Slide 9: Explanation of DELETE**

**Key Points:**

* The WHERE clause specifies which row(s) to delete.
* If the WHERE clause is omitted, all rows in the table will be deleted.

**Slide 10: Summary of Today's Lesson**

* **Create Table**: Define the structure of a table with columns and data types.
* **Insert Into**: Add new data into the table.
* **Update**: Modify existing data in the table.
* **Delete**: Remove data from the table.

**Slide 11: Sample Data for Practice**

**INSERT INTO Example Data:**

INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('Alice Brown', 'Tom Hardy', '456 Oak St', 'Chicago', '60610', 'USA');

INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('Emily Clark', 'Chris Evans', '789 Pine St', 'Los Angeles', '90001', 'USA');

**UPDATE Example:**

UPDATE Customers

SET City = 'San Francisco'

WHERE CustomerID = 2;

**DELETE Example:**

DELETE FROM Customers

WHERE CustomerID = 3;