

Module 12

What to learn

Return JSON output from Store Procedure

Use the SET NOCOUNT ON

WITH ENCRYPTION

Exception Handling

Using Try Catch

Practice Exercise

Practice 1

Count the Number of Customers Living in the City where Branch is Located

Practice 2

Practice: Retrieve Specific Columns from a Table

Using the **SELECT** statement, query the sample **Employees** table to get only the **EmployeeID**, **FirstName**, and **LastName** of all employees. Ensure the results display these columns in the mentioned order.

Table Example: Employees (EmployeeID, FirstName, LastName, Department, Salary).

Practice 3

Practice: Insert Data into a Table

Using the **INSERT** statement, write a query to add a new record to the **Products** table. The record should include details such as **ProductID**, **ProductName**, and **Price**. Ensure you specify proper values for these fields.

Table Example: Products (ProductID, ProductName, Price, Quantity).

Practice 4

Practice: Update Specific Data in a Table

Using the **UPDATE** statement, modify the **Salary** field of an employee (identified through **EmployeeID**) in the **Employees** table. Increase their salary by 10%. Use the appropriate **WHERE** clause to ensure only one record is updated.

Table Example: Employees (EmployeeID, FirstName, LastName, Department, Salary).

Practice 5

Practice: Delete Data Conditionally

Using the DELETE statement, remove records from the **Orders** table where the **Status** is set to 'Cancelled'. Use the appropriate WHERE clause to ensure only such records are deleted.

Table Example: Orders (OrderID, CustomerID, OrderDate, Status, TotalAmount).

Practice 6

Practice: Filter Results Using Comparison and Logical Operators

Using the SELECT statement along with a WHERE clause, query the **Products** table to retrieve all products with a **Price** greater than 50 and **Quantity** less than or equal to 20. Combine conditions using logical operators.

Table Example: Products (ProductID, ProductName, Price, Quantity).

Practice 7

Practice: Use the IN and NOT IN Keywords

Using the IN keyword, query the **Orders** table to retrieve all orders made by customers with **CustomerID** either 101, 102, or 103. Next, write another query using NOT IN to exclude these customers.

Table Example: Orders (OrderID, CustomerID, OrderDate, Status, TotalAmount).

Practice 8

Practice: Use LIKE and Wildcard Patterns

Using the LIKE keyword, query the **Customers** table to find all customers whose **FirstName** starts with the letter 'A' and ends with 'n'. Experiment with different wildcard patterns.

Table Example: Customers (CustomerID, FirstName, LastName, Email, ContactNumber).

Practice 9

Practice: Order Data Using ORDER BY

Using the ORDER BY clause, query the **Transactions** table to list all transactions sorted by **TransactionAmount** in descending order. Use a secondary sort on **TransactionDate** in ascending order.

Table Example: Transactions (TransactionID, CustomerID, TransactionAmount, TransactionDate, PaymentMethod).

Practice 10

Practice: Use TOP and DISTINCT

Write two separate queries using the **Products** table:

1. Retrieve the top 5 most expensive products using the **TOP** keyword.
2. Retrieve a list of distinct **Categories** available in the table using **DISTINCT**.

Table Example: Products (ProductID, ProductName, Category, Price, Quantity).

Assignment Exercise

Assignment 1

Step 1: Create the following tables and insert the data as listed above : Step 2: Create the queries listed below: Q1: Create a Store Procedure which will accept name of the customer as input parameter and product the following output, List Names of Customers who are Depositors and have Same Branch City as that of input parameter customer's Name. Q2: Create a Store Procedure which will accept name of the customer as input parameter and produce the following output List in JSON format, All the Depositors Having Depositors Having Deposit in All the Branches where input parameter customer is Having an Account Q3: Create a Store Procedure that will accept city name and returns the number of customers in that city. Q4: Create a Store Procedure which will accept city of the customer as input parameter and product the following output List in JSON format List All the Customers Living in city provided in input parameter and Having the Branch City as MUMBAI or DELHI Q5: Count the Number of Customers Living in the City where Branch is Located Q6: Create a Procedure which will accept input in JSON parameter CustomerName, City, ACTNO, Branch, amount And insert these record in the Deposit table. Before inserting some validation should be done like amount should be greater than 10Rs. and date should always be current date.

Assignment 2

Assignment: University Management System

Design a database for a university system using the following tables. Populate the tables with sample data (at least 5 records per table) and perform the given operations.

Tables:

Students: StudentID (PK), FirstName, LastName, DateOfBirth, DepartmentID (FK to Departments).

Departments: DepartmentID (PK), DepartmentName, Location.

Courses: CourseID (PK), CourseName, DepartmentID (FK to Departments), Credits.

Enrollments: EnrollmentID (PK), StudentID (FK to Students), CourseID (FK to Courses), EnrollDate.

Tasks:

Create these tables with all appropriate relationships and constraints.

Insert sample data into each table.

Write a query to retrieve the full names of students who are enrolled in the 'Computer Science' department.

Write a query to calculate the total number of students in each department.

Write a query to fetch all courses along with their respective departments.

Update the credits of all courses in the 'Mathematics' department to increase by 1.

Delete enrollments older than 5 years from the enrollments table.

Create a view to display students' enrollment details (Student Full Name, Course Name, EnrollDate).

Create another view to display the count of courses per department.

Create a view to fetch students who are enrolled in more than 3 courses.

Ensure the queries cover all provided topics, such

as SELECT, INSERT, UPDATE, DELETE, DQL, WHERE clause, ORDER BY, DISTINCT, etc.

Online Reference

<https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/...>

Introduction to Relational Databases

Introduction to Select Statement

Filtering Results with WHERE Statements

Utilizing Joins

Executing Sub queries and Unions

Aggregating Data

Advanced Data Aggregations

Built in Functions

Query Optimization

Modifying Data

Advanced Data Modification

Stored Procedure

Transaction

Error handling

Designing Tables

triggers