Module 2

What to learn

- DML
  - SELECT
  - INSERT
  - UPDATE
  - DELETE
- DQL
  - Basic Select Statement
  - Literals & Concatenate
  - Calculating Column Values
  - Where clause
  - Comparison & Logical Operator
  - Range Operator
  - In/not in keyword
  - Like, OrderBy, Top, Distinct

Practice Exercise

Practice 1

Do the hands on the whatever you learned from ppt and videos.

Practice 2

Practice Exercise: Using SQL String Functions

Objective:

Write an SQL query to find the position of the first occurrence of the word 'SQL' in the 'Description' column of a table named 'Courses'. If the string is not found, return zero.

Setup:

Table: Courses

- Columns: CourseID (INT), CourseName (VARCHAR(50)), Description (VARCHAR(255))

Specific Functions to Cover:

- CHARINDEX
- LEN

Practice 3

Practice Exercise: Manipulating Case and Trimming Strings

Objective:

Write an SQL query to convert all entries in the 'EmployeeName' column of the 'Employees' table to lowercase, and then trim whitespaces from both ends of each name.

Setup:

Table: Employees

Columns: EmployeeID (INT), EmployeeName (VARCHAR(100)), Department (VARCHAR(50))

Specific Functions to Cover:

LOWER

TRIM

Practice 4

Practice Exercise: Replacing and Replicating Strings

Objective:

Write an SQL query to replace occurrences of the word 'Expired' with 'Inactive' in the 'Status' column of the 'Products' table. Then, replicate the updated status three times in the result set.

Setup:

Table: Products

Columns: ProductID (INT), ProductName (VARCHAR(50)), Status (VARCHAR(20))

Specific Functions to Cover:

REPLACE

REPLICATE

Practice 5

Practice Exercise: Using Date Functions

Objective:

Write an SQL query to retrieve the names of customers whose transactions occurred in the current month using the 'TransactionDate' column.

Setup:

Table: Transactions

Columns: TransactionID (INT), CustomerName (VARCHAR(50)), TransactionDate (DATE)

Specific Functions to Cover:

GETDATE
DATENAME

Practice 6

Practice Exercise: Mathematical Functions

Objective:

Calculate the total amount spent by each customer by summing up the 'AmountSpent' column and round the total to 2 decimal places.

Setup:

Table: CustomerOrders

Columns: OrderID (INT), CustomerID (INT), AmountSpent (DECIMAL(10,2))

Specific Functions to Cover:

SUM
ROUND

Practice 7

Practice Exercise: Combining Functions

Objective:

Write an SQL query to select the first 5 characters from the reversed 'EmployeeCode' column, and concatenate it with a space followed by the department name.

Setup:

Table: EmployeeDetails

Columns: EmployeeID (INT), EmployeeCode (VARCHAR(20)), DepartmentName (VARCHAR(50))

Specific Functions to Cover:

REVERSE
LEFT
CONCAT

Assignment Exercise

Assignment 1

Write a SQL statement to change the Email column of Employees table with ‘not available’ for all employees.

Assignment 2

Write a SQL statement to change the Email and CommissionPct column of employees table with 'not available' and 0.10 for all employees.

Assignment 3

Write a SQL statement to change the Email and CommissionPct column of employees table with 'not available' and 0.10 for those employees whose DepartmentID is 110.

Assignment 4

Write a SQL statement to change the Email column of employees table with 'not available' for those employees whose DepartmentID is 80 and gets a commission is less than 20%

Assignment 5

Write a SQL statement to change the Email column of employees table with 'not available' for those employees who belongs to the 'Accouning' department.

Assignment 6

Write a SQL statement to change salary of employee to 8000 whose ID is 105, if the existing salary is less than 5000.

Assignment 7

Write a SQL statement to change job ID of employee which ID is 118, to SH_CLERK if the employee belongs to department, which ID is 30 and the existing job ID does not start with SH.

Assignment 8

Write a SQL statement to increase the salary of employees under the department 40, 90 and 110 according to the company rules that, salary will be increased by 25% for the department 40, 15% for department 90 and 10% for the department 110 and the rest of the departments will remain same.

Assignment 9

Write a SQL statement to increase the minimum and maximum salary of PU_CLERK by 2000 as well as the salary for those employees by 20% and commission by 10% .

Assignment 10

Basic Select Queries: Get all employee details from the Employee table Get FirstName, LastName from Employees table Get FirstName from Employees table using alias name "Employee Name" Get employee details from Employees table whose Employee Name is "Steven" Get employee details from Employees table whose Employee Name are "Neena" and "Lex" Get employee details from Employees table whose Employee name are not "Neena" and "Neena" Get employee details from Employees table whose Salary between 5000 and 8000 Write a query to get the names (FirstName, LastName), Salary, PF of all the Employees (PF is calculated as 12% of salary). Get employee details from Employees table whose FirstName starts with 'N' Write a query to get unique department ID from Employees table Write a query to get all employee details

from the employee table order by FirstName, descending. Write a query to get the EmployeeID, names (FirstName, LastName), salary in ascending order of salary.
Select TOP 2 salary from employee table

Assignment 11

Assignment: Customer Order and Transaction Management

Objective:

The goal of this assignment is to design and query a database that manages customer orders and transactions for an e-commerce company. This assignment will test your understanding of string, date, mathematical, and system functions along with foundational SQL skills.

Scenario:

You are working for an e-commerce company. The company has multiple customers ordering products. Each customer transaction is logged and is associated with various details such as date, amount, and status. Your task is to manage customer data, their orders, and transactions using SQL queries.

Tables Setup:

CustomerDetails

CustomerID: INT (Primary Key)
FullName: VARCHAR(100)
Email: VARCHAR(100)
Address: VARCHAR(255)
ContactNumber: BIGINT

OrderDetails

OrderID: INT (Primary Key)
CustomerID: INT (Foreign Key)
OrderDate: DATE
TotalAmount: DECIMAL(10, 2)
OrderStatus: VARCHAR(20)

TransactionDetails

TransactionID: INT (Primary Key)
OrderID: INT (Foreign Key)
TransactionDate: DATE
TransactionAmount: DECIMAL(10, 2)
PaymentMethod: VARCHAR(20)

Tasks:

1. Table Creation

Create the three tables (CustomerDetails, OrderDetails, TransactionDetails) as described above with appropriate constraints.

2. Data Insertion

Insert at least 10 records into each table. Ensure the records have meaningful relationships between each other.

3. String Manipulation

Prepare a query to display customer names in uppercase along with the first 30 characters of their address and their email domain (e.g., extract 'gmail.com' from 'user@gmail.com').

4. Date-based Filtering

Write an SQL query to list the names of customers and their order details where the order was placed within the last 90 days.

5. Complex Queries

a. Find customers whose total spending exceeds $1000. Use mathematical functions to calculate the cumulative transaction amounts per customer.

b. Generate a report that lists all transactions with the order date, customer name, and the payment method concatenated into a single field (separated by spaces).

6. System Functions

Create a query to generate a system-based unique identifier (using a system function) for all recent transactions in the TransactionDetails table. Display these IDs along with the transaction records.

Output Requirements:

All queries should output meaningful, formatted results based on the explicit requirements described. The assignment should be robust, covering all provided aspects of the data and specific functions.

Online Reference

No online Reference

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