

## Module 6

### What to learn

Subqueries

Views

### Practice Exercise

#### Practice 1

Do the hands on from video and ppt available on tutorial site

#### Practice 2

### Understanding Clustered Indexes

Create a table named `Employees` with the structure: `EmployeeID` (Primary Key, INT), `FirstName` (NVARCHAR(50)), `LastName` (NVARCHAR(50)), `Salary` (DECIMAL(10, 2)). Insert 10 rows of sample data. Query the data sorted by `EmployeeID`. Explain why the Primary Key automatically creates a Clustered Index and how it impacts data retrieval performance.

#### Practice 3

### Working with NonClustered Indexes

Given the same `Employees` table, create a NonClustered Index on the `LastName` column. Write a query that searches for employees with a specific `LastName`. Measure and compare the query performance before and after adding the NonClustered Index. Provide an explanation of the difference.

#### Practice 4

### Combining Clustered and NonClustered Indexes

Using the `Employees` table, write a query to retrieve all employees earning a salary greater than \$8000, ordered by `Salary`. Add a NonClustered Index on the `Salary` column. Explain how this setup leverages both the Clustered (on `EmployeeID`) and NonClustered Index (on `Salary`) for faster query execution.

#### Practice 5

### Covering Index Concepts

Create a table named `Orders` with the structure: `OrderID` (Primary Key, INT), `CustomerID` (INT), `OrderDate` (DATE), `TotalAmount` (DECIMAL(10, 2)). Insert 10 rows of sample data. Create a NonClustered Index on (`CustomerID`, `OrderDate`). Write queries that utilize this covering index and explain how it avoids additional table lookups.

#### Practice 6

## Index Maintenance

Using the **Orders** table, add 10 more rows of data and run queries to search for a specific **CustomerID**. Observe and document how adding too many records affects the performance. Then, rebuild the NonClustered Index using **REBUILD** or **REORGANIZE** commands. Explain how index maintenance improves performance.

### Assignment Exercise

#### Assignment 1

Note: Refer existing employee Table Select employee details from employee table if data exists in incentive table ? Find Salary of the employee whose salary is more than Roy Salary Create a view to select

FirstName,LastName,Salary,JoiningDate,IncentiveDate and IncentiveAmount

Create a view to select Select first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000

#### Assignment 2

(Note refer tables from sql supported files) Create a View to Find the names (first\_name, last\_name), job, department number, and department name of the employees who work in London Create a View to get the department name and number of employees in the department. Find the employee ID, job title, number of days between ending date and starting date for all jobs in department 90 from job history.

#### Assignment 3

Write a View to display the department name, manager name, and city. Create a View to display department name, name (first\_name, last\_name), hire date, salary of the manager for all managers whose experience is more than 15 years.

#### Assignment 4

## Library Management System – Index Optimization

You are working on building a Library Management System. The following database schema is provided:

**Books** table: **BookID** (Primary Key, INT), **Title** (NVARCHAR(255)), **Author** (NVARCHAR(255)), **PublicationYear** (YEAR), **Genre** (NVARCHAR(50)).

**Members** table: **MemberID** (Primary Key, INT), **FirstName** (NVARCHAR(50)), **LastName** (NVARCHAR(50)), **MembershipDate** (DATE).

**BorrowRecords** table: **BorrowID** (Primary Key, INT), **BookID** (Foreign Key), **MemberID** (Foreign Key), **BorrowDate** (DATE), **ReturnDate** (DATE).

Perform the following tasks:

Create the above tables and populate them with at least 20 records for **Books**, 10 records for **Members**, and 30 records for **BorrowRecords**.

Analyze query performance for: searching books by **Title**, searching members by **LastName**, and listing all borrow records for a specific member sorted by **BorrowDate**.

Add a Clustered Index on **BorrowID** and a NonClustered Index on **Title** (in **Books**) and **LastName** (in **Members**). Re-execute the queries and measure the performance improvement.

Create a covering NonClustered Index on **BorrowRecords** for columns (**MemberID**, **BorrowDate**, **ReturnDate**). Write and test a query to retrieve the borrowing history of a specific member using this index. Provide a report detailing the performance improvement observed, explaining how the indexes impacted the query execution.

This assignment incorporates Clustered Indexes, NonClustered Indexes, and their impact on practical query optimization.

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