**CASE STUDY : Project of Student Management System**

**Submitted By : Khairuzzaman**

**Trainee ID : 1278830**

**Batch : WADA/PNTL-A/56/01**

**Introduction:**

* Microsoft SQL Server is a relational database management system. As a database server that stores and retrieves data as requested by other software applications on the same computer or a remote computer using the client-server model. Microsoft provides APIs to access SQL Server over the internet as a web service.
* The purpose of this project to create a simple and user-friendly system to Manage Students.
* In this Project we use SQL Server to create a Student Management System. Where you can INSERT, UPDATE and DELETE information's.

**About This Project:**

In this Project we can VIEW, INSERT, UPDATE and DELETE following Information's.

* Students Information.
* Instructor Information.
* Department Information.
* Course Information.
* Semester Information.
* Subject Information.
* Attendance Information.
* Results Information.

**Basic Table Name with Columns:**

* Students Information.
* SL
* StudentID
* StudentName
* FathersName
* MothersName
* Gender
* Date of Birth
* Email
* ContactNumber
* Address
* CurrentSemester
* DepartmentID
* CourseID
* AvgResults (OutQuery)
* Instructor Information.
* SL
* InstructorID
* InstructorName
* Gender
* Email
* ContactNumber
* Address
* SubjectCode
* DepartmentID
* SemesterCode
* CourseID
* Department Information.
* SL
* DepartmentID
* DepartmentName
* Course Information.
* SL
* CourseID
* CourseName
* Duration
* InstructorID
* Semester Information.
* SL
* SemesterCode
* SemesterName
* SubjectCode
* Subject Information.
* SL
* SubjectCode
* SubjectName
* InstructorID
* DepartmentID
* Attendance Information.
* SL
* StudentID
* Attandance
* Results.
* SL
* StudentID
* SemesterCode
* SubjectCode
* Result

**About DDL File:**

**=> SECTION 01: CREATE a Database [SMS]**

**Definition:** This section involves the SQL command to create a new database. A database is a structured collection of data that is organized and managed in a way that allows for efficient storage, retrieval, and manipulation of data.

**=> SECTION 02: CREATE Appropriate Tables with column definition, Schema.**

**Definition:** In this section, you define the structure of your database by creating tables. Tables are used to store data in a relational database. Each table consists of rows (records) and columns (attributes). The definition includes specifying column names, data types, constraints (e.g., primary key, foreign key), and optionally, setting default values or applying validation rules.

**=> SECTION 03: ALTER, DROP AND MODIFY TABLES & COLUMNS.**

**Definition:** SQL provides commands to alter the structure of existing tables. You can add, modify, or remove columns, as well as change column data types and constraints using the ALTER TABLE statement. The DROP TABLE statement is used to delete an entire table and its data from the database.

**=> SECTION 04: CREATE CLUSTERED AND NONCLUSTERED INDEX**

**Definition:** Indexes in SQL are used to improve the performance of data retrieval operations. They work like the index of a book, allowing the database engine to find data faster. A clustered index determines the physical order of data rows in a table, while a non-clustered index provides a separate structure that points to the actual data rows.

**=> SECTION 05: CREATE SEQUENCE & ALTER SEQUENCE**

**Definition:** A sequence in SQL is an object that generates a series of numeric values in an incremental manner. It is often used to generate unique primary key values for a table automatically. The CREATE SEQUENCE statement is used to create a new sequence, and the ALTER SEQUENCE statement allows modification of the sequence properties.

**=> SECTION 06: CREATE A VIEW & ALTER VIEW**

**Definition:** A view in SQL is a virtual table that is based on the result of an SQL SELECT query. It allows you to encapsulate complex queries into a single object, making it easier to manage and use them. The CREATE VIEW statement is used to create a view, and the ALTER VIEW statement allows modification of the view's definition.

**=> SECTION 07: CREATE STORED PROCEDURE & ALTER STORED PROCEDURE**

**Definition:** A stored procedure in SQL is a precompiled set of one or more SQL statements that are stored and executed on the database server. They can accept input parameters and return results. The CREATE PROCEDURE statement is used to create a new stored procedure, and the ALTER PROCEDURE statement allows modification of the procedure's code.

**=> SECTION 08: CREATE FUNCTION(SCALAR, SIMPLE TABLE VALUED, MULTISTATEMENT TABLE VALUED) & ALTER FUNCTION**

**Definition:** Functions in SQL are similar to stored procedures but return a single value or a table of values as a result. Scalar functions return a single value, simple table-valued functions return a table with a single result set, and multi-statement table-valued functions return a table with multiple result sets. The CREATE FUNCTION statement is used to create a new function, and the ALTER FUNCTION statement allows modification of the function's code.

**=> SECTION 09: CREATE TRIGGER (FOR/AFTER TRIGGER and INSTEAD OF TRIGGER)**

**Definition:** Triggers in SQL are special types of stored procedures that are automatically executed when certain database events occur, such as INSERT, UPDATE, DELETE, etc. They are used to enforce data integrity rules, audit changes, or perform automated actions. FOR/AFTER triggers execute after the triggering event, and INSTEAD OF triggers replace the triggering event. The CREATE TRIGGER statement is used to create a new trigger.

**About DML File:**

**Section: 01 => Using INSERT INTO KEYWORD Insert data into Current TABLE.**

**Definition:** The INSERT INTO statement in SQL is used to add new rows (records) into a table. It allows you to specify the table name and provide values or a SELECT statement that retrieves the data to be inserted into the table.

**Section: 02 => Using SELECT FROM Keyword View TABLE and their Data**

**Definition:** The SELECT statement in SQL is used to retrieve data from one or more tables. It allows you to specify the columns you want to retrieve and filter the data using various conditions using the WHERE clause. You can also join multiple tables to fetch data from related tables.

**Section: 03 => Using SELECT INTO Keyword Copy all TABLE Data into another TABLE**

**Definition:** The SELECT INTO statement in SQL is used to create a new table and copy data from an existing table into the new table. It allows you to create a copy of the entire table or a subset of the data based on a condition specified in the WHERE clause.

**Section: 04 => IMPLICIT INNER JOIN, OUTER JOIN and CROSS JOIN**

**Definition:** Joins in SQL are used to combine rows from two or more tables based on related columns. An INNER JOIN retrieves rows that have matching values in both tables. OUTER JOINs (LEFT JOIN, RIGHT JOIN, and FULL JOIN) retrieve rows from one table even if there are no matches in the other table. CROSS JOIN produces the Cartesian product of the two tables.

**Section: 05 => IMPLICIT WHERE CLAUSE, Order BY, DISTINCT, TOP ,BETWEEN and Comparison Operator (AND, OR, NOT)**

**Definition:** The WHERE clause in SQL is used to filter rows based on specified conditions. The ORDER BY clause is used to sort the result set. The DISTINCT keyword is used to retrieve unique rows from the result set. The TOP keyword limits the number of rows returned. The BETWEEN operator is used to filter rows within a specific range. Comparison operators (AND, OR, NOT) are used to combine multiple conditions in the WHERE clause.

**Section: 06 => LIKE,IN,IS NULL Operator**

**Definition:** The LIKE operator is used in SQL to perform pattern matching for string values. The IN operator allows you to specify multiple values in a WHERE clause. The IS NULL operator is used to check for null values in a column.

**Section: 07 => OFFSET, FETCH, AGGREGATE FUNCTIONS**

**Definition:** The OFFSET-FETCH clause in SQL is used for pagination, allowing you to skip a specified number of rows and retrieve a specific number of rows from the result set. Aggregate functions (e.g., SUM, AVG, COUNT, MAX, MIN) perform calculations on a set of values and return a single value as the result.

**Section: 08 => ROLLUP, CUBE, GROUP BY, GROUPING SETS AND HAVING**

**Definition:** These are advanced grouping features in SQL. GROUP BY is used to group rows based on one or more columns. ROLLUP and CUBE are used for creating subtotal and grand total rows in the result set. GROUPING SETS allow grouping by multiple sets of columns. The HAVING clause is used to filter the result set after grouping.

**Section: 09 => Union**

**Definition:** The UNION operator in SQL is used to combine the result sets of two or more SELECT queries into a single result set. It allows you to retrieve data from multiple tables or queries and present the combined result as a single result set. The UNION operator removes duplicate rows from the combined result, meaning it returns only distinct rows from all the SELECT queries involved.

**Section: 10 => SubQuerie (subqueries,and co-related subqueries)**

**Definition:** A subquery (also known as a nested query or inner query) is a query inside another query. Subqueries can be used in the SELECT, FROM, or WHERE clause. A correlated subquery is a subquery that depends on the outer query for its values.

**Section: 11 => EXISTS CLAUSE**

**Definition:** The EXISTS clause in SQL is used to check the existence of rows returned by a subquery. It returns true if the subquery returns any rows; otherwise, it returns false.

**Section: 12 => CTE for Combined Data**

**Definition:** CTE stands for Common Table Expression. It is a temporary result set that can be referred to within the context of a single SQL statement. CTEs are often used to simplify complex queries by breaking them into smaller, more manageable parts.

**Section: 13 => MERGE**

**Definition:** The MERGE statement in SQL is used to perform an UPSERT operation (i.e., UPDATE or INSERT) based on a condition. It allows you to insert new rows into a table or update existing rows if they match the specified condition.

**Section: 14 => IIF,CASE,CHOOSE**

**Definition:** These are conditional expressions in SQL. IIF is used for simple if-else conditions. CASE allows you to perform conditional logic with multiple branches. CHOOSE is used to select a value from a list based on the given index.

**Section: 15 => COALESCE & ISNULL**

**Definition:** COALESCE and ISNULL are used to handle NULL values. They return the first non-null expression among the provided arguments. However, COALESCE can handle multiple expressions, whereas ISNULL takes only two arguments.

**Section: 16 => GROPING FUNCTION**

**Definition:** The GROUPING function in SQL is used with the GROUP BY clause and is used to identify aggregated rows (subtotals) produced by ROLLUP and CUBE.

**Section: 17 => RANKING FUNCTION(RANK,DENSE RANK)**

**Definition:** Ranking functions in SQL (e.g., RANK, DENSE\_RANK) assign a unique rank to each row based on the ordering specified in the ORDER BY clause. RANK leaves gaps in ranking for tied values, while DENSE\_RANK does not.

**Section: 18 => BUILD IN FUNCTIONS (GETDATE, CAST, CONVERT, TRY\_CONVERT, DATEDIFF, DATENAME)**

**Definition:**

GETDATE: The GETDATE() function in SQL is used to retrieve the current system date and time from the database server.

CAST and CONVERT: These functions are used to explicitly convert one data type to another. CAST is the ANSI-standard way, while CONVERT is specific to SQL Server.

TRY\_CONVERT: The TRY\_CONVERT function in SQL Server attempts to convert a value to a specified data type. If the conversion fails, it returns NULL instead of raising an error.

DATEDIFF: The DATEDIFF function is used to calculate the difference between two dates or times, such as the number of days, months, or years between two dates.

DATENAME: The DATENAME function is used to retrieve a specific part of a date or time, such as the day, month, year, etc. It returns the name of the specified date part as a character string.

These built-in functions are commonly used for date and time manipulation and data type conversions in SQL queries and expressions**.**

**Section: 19 => IF ELSE & PRINT**

**Definition:** IF ELSE is a conditional control flow statement in SQL that allows you to execute different blocks of code based on a condition. PRINT is used to display messages or variable values during script execution, primarily for debugging purposes.

**Section: 20 => WHILE, TRY CATCH**

**Definition:**

WHILE: The WHILE loop is a control flow statement in SQL that allows you to repeatedly execute a block of SQL statements as long as a specified condition is true. It is often used for iterative operations.

TRY CATCH: The TRY CATCH block is used for error handling in SQL. It allows you to catch and handle exceptions that occur during the execution of a SQL batch. The TRY block contains the code to be executed, and the CATCH block contains the code to handle any errors that occur within the TRY block.

**Section: 21 => WAITFOR**

**Definition:** The WAITFOR statement in SQL is used to introduce a delay in the execution of a script or batch. It is often used for testing or to introduce pauses in script execution.

**Section: 22 => sp\_helptext**

**Definition:** sp\_helptext is a system stored procedure in SQL Server that is used to retrieve the definition (source code) of user-defined objects like stored procedures, functions, and views. It is helpful when you want to view the script used to create these objects.