

SQL Queries

Section 1. Querying data

This section helps you learn how to query data from the SQL Server database. We will start with a simple query that allows you to retrieve data from a single table.

- **SELECT** – show you how to query data against a single table.

Section 2. Sorting data

- **ORDER BY** – sort the result set based on values in a specified list of columns

Section 3. Limiting rows

- **OFFSET FETCH** – limit the number of rows returned by a query.
- **SELECT TOP** – limit the number of rows or percentage of rows returned in a query's result set.

Section 4. Filtering data

- **DISTINCT** – select distinct values in one or more columns of a table.
- **WHERE** – filter rows in the output of a query based on one or more conditions.
- **AND** – combine two Boolean expressions and return true if all expressions are true.
- **OR** – combine two Boolean expressions and return true if either of conditions is true.
- **IN** – check whether a value matches any value in a list or a subquery.
- **BETWEEN** – test if a value is between a range of values.
- **LIKE** – check if a character string matches a specified pattern.
- **Column & table aliases** – show you how to use column aliases to change the heading of the query output and table alias to improve the readability of a query.

Section 5. Joining tables

- **Joins** – give you a brief overview of joins types in SQL Server including inner join, left join, right join and full outer join.
- **INNER JOIN** – select rows from a table that have matching rows in another table.
- **LEFT JOIN** – return all rows from the left table and matching rows from the right table. In case the right table does not have the matching rows, use null values for the column values from the right table.
- **RIGHT JOIN** – learn a reversed version of the left join.
- **FULL OUTER JOIN** – return matching rows from both left and right tables, and rows from each side if no matching rows exist.

SQL Learning Material

- **CROSS JOIN** – join multiple unrelated tables and create Cartesian products of rows in the joined tables.
- **Self join** – show you how to use the self-join to query hierarchical data and compare rows within the same table.

Section 6. Grouping data

- **GROUP BY**– group the query result based on the values in a specified list of column expressions.
- **HAVING** – specify a search condition for a group or an aggregate.
- **GROUPING SETS** – generates multiple grouping sets.
- **CUBE** – generate grouping sets with all combinations of the dimension columns.
- **ROLLUP** – generate grouping sets with an assumption of the hierarchy between input columns.

Section 7. Subquery

This section deals with the subquery which is a query nested within another statement such as SELECT, INSERT, UPDATE or DELETE statement.

- **Subquery** – explain the subquery concept and show you how to use various subquery type to select data.
- **Correlated subquery** – introduce you to the correlated subquery concept.
- **EXISTS** – test for the existence of rows returned by a subquery.
- **ANY** – compare a value with a single-column set of values returned by a subquery and return TRUE the value matches any value in the set.
- **ALL** – compare a value with a single-column set of values returned by a subquery and return TRUE the value matches all values in the set.

Section 8. Set Operators

This section walks you through of using the set operators including union, intersect, and except to combine multiple result sets from the input queries.

- **UNION** – combine the result sets of two or more queries into a single result set.
- **INTERSECT** – return the intersection of the result sets of two or more queries.
- **EXCEPT** – find the difference between the two result sets of two input queries.

Section 9. Common Table Expression (CTE)

- **CTE** – use common table expressions to make complex queries more readable.
- **Recursive CTE** – query hierarchical data using recursive CTE.

Section 10. Pivot

- PIVOT – convert rows to columns

Section 11. Modifying data

In this section, you will learn how to change the contents of tables in the SQL Server database. The SQL commands for modifying data such as insert, delete, and update are referred to as data manipulation language (DML).

- INSERT – insert a row into a table
- INSERT multiple rows – insert multiple rows into a table using a single INSERT statement
- INSERT INTO SELECT – insert data into a table from the result of a query.
- UPDATE – change the existing values in a table.
- UPDATE JOIN – update values in a table based on values from another table using JOIN clauses.
- DELETE – delete one or more rows of a table.
- MERGE – walk you through the steps of performing a mixture of insertion, update, and deletion using a single statement.
- Transaction – show you how to start a transaction explicitly using the BEGIN TRANSACTION, COMMIT, and ROLLBACK statements

Section 12. Data definition

This section shows you how to manage the most important database objects including databases and tables.

- CREATE DATABASE – show you how to create a new database in a SQL Server instance using the CREATE DATABASE statement and SQL Server Management Studio.
- DROP DATABASE – learn how to delete existing databases.
- CREATE SCHEMA – describe how to create a new schema in a database.
- ALTER SCHEMA – show how to transfer a securable from a schema to another within the same database.
- DROP SCHEMA – learn how to delete a schema from a database.
- CREATE TABLE – walk you through the steps of creating a new table in a specific schema of a database.
- Identity column – learn how to use the IDENTITY property to create the identity column for a table.
- Sequence – describe how to generate a sequence of numeric values based on a specification.
- ALTER TABLE ADD column – show you how to add one or more columns to an existing table

SQL Learning Material

- **ALTER TABLE ALTER COLUMN** – show you how to change the definition of existing columns in a table.
- **ALTER TABLE DROP COLUMN** – learn how to drop one or more columns from a table.
- **Computed columns** – how to use the computed columns to reuse the calculation logic in multiple queries.
- **DROP TABLE** – show you how to delete tables from the database.
- **TRUNCATE TABLE** – delete all data from a table faster and more efficiently.
- **SELECT INTO** – learn how to create a table and insert data from a query into it.
- **Rename a table** – walk you through the process of renaming a table to a new one.
- **Temporary tables** – introduce you to the temporary tables for storing temporarily immediate data in stored procedures or database session.
- **Synonym** – explain you the synonym and show you how to create synonyms for database objects.

Section 13. SQL Server Data Types

- **SQL Server data types** – give you an overview of the built-in SQL Server data types.
- **BIT** – store bit data i.e., 0, 1, or NULL in the database with the BIT data type.
- **INT** – learn about various integer types in SQL server including BIGINT, INT, SMALLINT, and TINYINT.
- **DECIMAL** – show you how to store exact numeric values in the database by using DECIMAL or NUMERIC data type.
- **CHAR** – learn how to store fixed-length, non-Unicode character string in the database.
- **NCHAR** – show you how to store fixed-length, Unicode character strings and explain the differences between CHAR and NCHAR data types
- **VARCHAR** – store variable-length, non-Unicode string data in the database.
- **NVARCHAR** – learn how to store variable-length, Unicode string data in a table and understand the main differences between VARCHAR and NVARCHAR.
- **DATETIME2** – illustrate how to store both date and time data in a database.
- **DATE** – discuss the date data type and how to store the dates in the table.
- **TIME** – show you how to store time data in the database by using the TIME data type.
- **DATETIMEOFFSET** – show you how to manipulate datetime with the time zone.
- **GUID** – learn about the GUID and how to use the NEWID() function to generate GUID values.

Section 14. Constraints

- Primary key – explain you to the primary key concept and show you how to use the primary key constraint to manage a primary key of a table.
- Foreign key – introduce you to the foreign key concept and show you use the FOREIGN KEY constraint to enforce the link of data in two tables.
- NOT NULL constraint – show you how to ensure a column not to accept NULL.
- UNIQUE constraint – ensure that data contained in a column, or a group of columns, is unique among rows in a table.
- CHECK constraint – walk you through the process of adding logic for checking data before storing them in tables.

Section 15. Expressions

- CASE – add if-else logic to SQL queries by using simple and searched CASE expressions.
- COALESCE – handle NULL values effectively using the COALESCE expression.
- NULLIF – return NULL if the two arguments are equal; otherwise, return the first argument.

Section 16. Useful Tips

- Find duplicates – show you how to find duplicate values in one or more columns of a table.
- Delete duplicates – describe how to remove duplicate rows from a table.