**SQL and Data Base Learnings**

**SQL Intro:** SQL is a standard language for accessing and manipulating databases.

## **What is SQL?**

* SQL stands for Structured Query Language
* SQL lets you access and manipulate databases
* SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

**Commands in SQL:**

1. DDL (Data Definition Language)
2. DML (Data Manipulation Language)
3. DQL (Data Query Language)
4. DCL (Data Control Language) and
5. TCL (Transaction Control Language)

**DDL – Data Definition Language:** It consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database. DDL is a set of SQL commands used to create, modify, and delete database structures but not data. These commands are normally not used by a general user, who should be accessing the database via an application.

**List of DDL commands:**

**CREATE**: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).

**DROP**: This command is used to delete objects from the database.

**ALTER**: This is used to alter the structure of the database.

**TRUNCATE**: This is used to remove all records from a table, including all spaces allocated for the records are removed.

**COMMENT**: This is used to add comments to the data dictionary.

**RENAME**: This is used to rename an object existing in the database.

**DQL – Data Query Language:** It is used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it. We can define DQL as follows it is a component of SQL statement that allows getting data from the database and imposing order upon it. It includes the SELECT statement. This command allows getting the data out of the database to perform operations with it. When a **SELECT** is fired against a table or tables the result is compiled into a further temporary table, which is displayed or received by the program i.e., front-end.

**SELECT:** It is used to retrieve data from the database.

**DML –** Data Manipulation Language: It deals with the manipulation of data present in the database belonging to DML or Data Manipulation Language and this includes most of the SQL statements. It is the SQL statement component that controls access to data and the database. DCL statements are grouped with DML statements.

**List of DML commands:**

**INSERT**: It is used to insert data into a table.

**UPDATE**: It is used to update existing data within a table.

**DELETE**: It is used to delete records from a database table.

**DCL – Data Control Language:** It includes commands such as GRANT and REVOKE, which deal with the database system's rights, permissions, and other controls.

**List of DCL commands:**

**GRANT**: This command gives users access privileges to the database.

**REVOKE**: This command withdraws the user’s access privileges given by using the GRANT command.

**TCL – Transaction Control Language:** Transactions group a set of tasks into a single execution unit. Each transaction begins with a specific task and ends when all the tasks in the group are successfully completed. If any of the tasks fail, the transaction fails. Therefore, a transaction has only two results: success or failure.

**List of TCL commands:**

**Commit:** Commits a Transaction.

**Grant:** Rollbacks a transaction in case any error occurs.

**Roll Back:** Sets a save point within a transaction.

**Creating, Drop and Restore a DB:**

**Create:** The CREATE DATABASE statement is used to create a new SQL database.

**Syntax**: CREATE DATABASE databasename;

**Drop**: The DROP DATABASE statement is used to drop an existing SQL database.

**Syntax:** DROP DATABASE databasename;

**Restore**: To restore your database, follow these steps:

* Launch SQL Server Management Studio (SSMS) and connect to your SQL Server instance.
* Right-click the **Databases** node in **Object Explorer** and select **Restore Database...**.
* Select **Device:** and then select the ellipses (...) to locate your backup file.
* Select **Add** and navigate to where your .bak file is located. Select the .bak file and then select **OK**.
* Select **OK** to close the **Select backup devices** dialog box.
* Select **OK** to restore the backup of your database.

**Datatypes in SQL:**

Bigint, numeric, bit, smallint,

Decimal, smallmoney, int, tinyint,

Money, float, real, date,

Datetimeoffset, datetime2, smalldatetime, datetime,

Time, char, varchar, text,

Nchar, nvarchar, ntext, binary,

Varbinary, image,

**Constraints in SQL:**

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

**The following constraints are commonly used in SQL:**

* NOT NULL - Ensures that a column cannot have a NULL value.
* UNIQUE - Ensures that all values in a column are different.
* PRIMARY KEY - A combination of NOT NULL and UNIQUE. Uniquely identifies each row in a table.
* FOREIGN KEY - Prevents actions that would destroy links between tables.
* CHECK - Ensures that the value in a column satisfies a specific condition.
* DEFAULT - Sets a default value for a column if no value is specified.
* CREATE INDEX - Used to create and retrieve data from the database very quickly.

**Identity Column:**

Identity column of a table is a column whose value increases automatically. The value in an identity column is created by the server. A user cannot insert a value into an identity column. The identity column can be used to uniquely identify the rows in the table.

**Syntax:** IDENTITY [(seed, increment)]

Seed: Starting value of a column.

Default value is 1.

Increment: Incremental value that is

added to the identity value of the previous

row that was loaded. The default value is 1.

**WHERE clause:** It is used to filter records. It is used to extract only those records that fulfill a specified condition.

**Example**: SELECT \* FROM Customers

WHERE Country='Mexico';

**Order By Clause:** The ORDER BY keyword is used to sort the result-set in ascending or descending order.

**Example:** SELECT \* FROM Products

ORDER BY Price;

**Distinct Keyword:** The SELECT DISTINCT statement is used to return only distinct (different) values.

**Example:** SELECT DISTINCT Country FROM Customers;

**Isnull() function:** The ISNULL () function returns a specified value if the expression is NULL. If the expression is NOT NULL, this function returns the expression.

**Example:** SELECT ISNULL ('Hello', 'W3Schools.com');

**Column aliases:** SQL aliases are used to give a table, or a column in a table, a temporary name. Aliases are often used to make column names more readable. An alias only exists for the duration of that query. An alias is created with the AS keyword.

**Example:** SELECT CustomerID AS ID

FROM Customers;

**Between, And, In, Like:**

* **Between:** The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

**Example**: SELECT \* FROM Products

WHERE Price BETWEEN 10 AND 20;

* **In:** The IN operator allows you to specify multiple values in a WHERE clause.

**Example**: SELECT \* FROM Customers

WHERE Country IN ('Germany', 'France', 'UK');

* **And:** The AND operator is used to filter records based on more than one condition, like if you want to return all customers from Spain that starts with the letter 'G'.

**Example:** SELECT \*FROM Customers

WHERE Country = 'Spain' AND CustomerName LIKE 'G%';

* **Like:** The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the LIKE operator:

* The percent sign % represents zero, one, or multiple characters
* The underscore sign \_ represents one, single character

**Example:** SELECT \* FROM Customers

WHERE CustomerName LIKE 'a%';

**Order of processing of SQL Queries:** The SQL order of execution defines the order in which the clauses of a query are evaluated. Some of the most common query challenges people run into could be easily avoided with a clearer understanding of the SQL order of execution, sometimes called the SQL order of operations. Understanding SQL query order can help you diagnose why a query will not run, and even more frequently will help you optimize your queries to run faster.

Example: SELECT customer\_ID, SUM (total\_amount) AS “Total”

FROM orders

WHERE order\_date BETWEEN ‘2022-01-01’ AND ‘2022-03-31’

AND customer\_city = ‘New York’

GROUP BY customer\_id

ORDER BY Total DESC;

**Database Name: Nagendra**

**Table data**

**Employees Table:**

Select \* From Employees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EmpID** | **EmployeeName** | **EmployeeSalary** | **EmployeeEmail** | **EmployeeMobile** |
| 101 | Henry | 15230.26 | henry@outlook.com | 9856472515 |
| 102 | Shane | 23485.56 | shane@outlook.com | 8741254123 |
| 103 | Alex | 53864.98 | alex@outlook.com | 74109638521 |
| 104 | Richard | 28654.45 | richard@outlook.com | 8529637412 |
| 105 | Nicholas | 19852.63 | nicholas@outlook.com | 9638527418 |

**Company Table:**

Select \* from Company

|  |  |  |
| --- | --- | --- |
| **CompanyId** | **CompanyName** | **EmployeeId** |
| 1001 | Cognine | 101 |
| 1002 | Cognizant | 102 |
| 1003 | Accenture | 103 |
| 1004 | Infosys | 104 |
| 1005 | Wipro | 105 |

Select \* From Employees Select \* from Company gives the results of the entire table for the Company table

**Database Testing Topics:**

1. **Types of Testing:** There are three types of testing in the Database

* Functional Database Testing
* Non-Functional Database Testing
* Structural Database Testing
* **Functional Database Testing:** It is a process of verifying the database's functionality by testing the data stored in it**.** This type of testing can be used to test the functionality of a database system as a whole or to test the functionality of individual components within the system.

It is of two types -

White Box Testing

Black Box Testing

**White Box Testing:** It is a process of testing the database by looking at its internal structure**.** This type of testing is also known as clear box testing or structural testing. White box testing can be used to verify the data's correctness, integrity, and consistency. White-box testing is a type of testing that uses knowledge of the internal structure of the system under test to design test cases.

**Black Box Testing:** It is a process of testing the database without looking at its internal structure. This type of testing is also known as behavioral testing. Black box testing can be used to verify the functionality of the database. Black-box testing is a type of testing that does not use any knowledge of the system's internal structure. This type of testing is often used to assess the functionality of a system.

* **Non – Functional Database Testing:**

The following are the several types of non-functional database testing:

* + Stress Testing
  + Security Testing
  + Volume Testing
  + Load Testing
  + Performance Testing

1. **Stress Testing:** It is a type of performance testing conducted to evaluate a system's response when exposed to workloads that exceed normal operational capacity or approach the limits of its anticipated workload**.** Stress testing can be used to determine whether a system can handle increased loads, identify bottlenecks, and determine how well the system recovers from failures. It can also help assess the impact of changes to the system’s architecture or configuration. Security testing is performed to ensure a system is secure from unauthorized access and data breaches. This type of testing can be performed manually or usingautomated tools.
2. **Security Testing:** It is a type of testing that is used to assess the security of a software application. The goal of security testing is to identify security vulnerabilities in the application so that they can be fixed before the application is deployed. Security testing can be performed manually or using automated tools. Security testing is the process of identifying vulnerabilities in a software system and assessing the risks posed by those vulnerabilities. It is a critical part of any software development process and should be performed throughout the software lifecycle, from requirement gathering to final deployment. There are many different types of security tests, but they can broadly be classified into two categories:

* **Vulnerability scanning**: Identifying potential security vulnerabilities in a system.
* **Penetration testing**: Attempting to exploit vulnerabilities to gain access to sensitive data or systems.