UPTAKE\_SQL\_LEARNING\_DOCUMATION

1. SQL DATABASE

SQL Definition : SQL is a standard language for storing, manipulating and retrieving data in databases.

SQL stands for Structured Query Language

SQL lets you access and manipulate databases

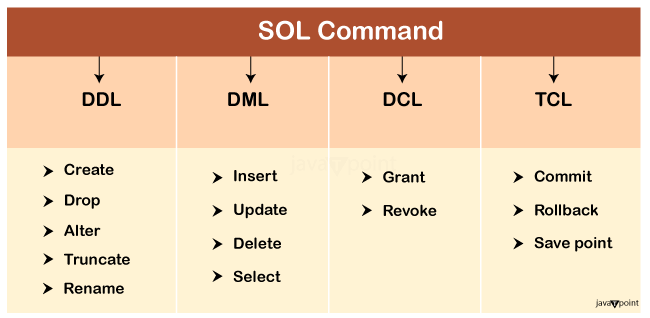
SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

1. NONSQL DATABASE

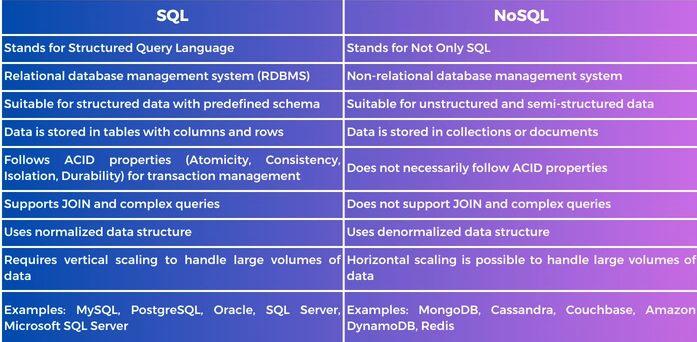
🡪 NoSQL stands for "not only SQL" and refers toa non-relational database that stores data in a non-tabular format. NoSQL databases are known for being flexible, easy to develop, and highly scalable. They are often used for applications that require large scale, high availability, and frequent data changes

1. Types of SQL Commands And query’s

There are four types of SQL commands: DDL, DML, DCL, TCL.



SQL VS NON SQL



----------------------------------------------------------------QUERYS PRACTICAL PRACTICE ----------------------------------

* Create : used to create database

QUERY : create database Bank;

Query OK, 1 row affected (0.01 sec)

create table Department(

-> d\_id int,

-> d\_name varchar(20),

-> d\_Address varchar(20));

Query OK, 0 rows affected (0.03 sec)

* INSERT :used to insert the values into existing tables

QUERY : insert into Department values(1,'Cashier','Ahemdabad');

Query OK, 1 row affected (0.01 sec)

* SELECT : used to select the records from the table

QUERY : select \* from Department;

* UPDATE :used to the update the existing records in table

QUERY : update Employes set Name='janik sir' where E\_id=1;

Query OK, 1 row affected (0.02 sec)

Rows matched: 1 Changed: 1 Warnings: 0

* DELETE : Delete used to delete the records from the tables

QUERY : delete from Employes where E\_id=1;

Query OK, 1 row affected (0.01 sec)

* DROP : used to drop entire data base even table also

QUERY : drop database demodb;

Query OK, 0 rows affected (0.05 sec)

* ALTER : to add new column inside the table even to Modify the column name also

QUERY : alter table Employes add E\_phone int;

Query OK, 0 rows affected (0.07 sec)

Records: 0 Duplicates: 0 Warnings: 0

* TRUNCATE : A truncate SQL statement is used to remove all rows (complete data) from a table. It is similar to the DELETE statement with no WHERE clause.

QUERY : TRUNCATE TABLE Employes;

Query OK, 0 rows affected (0.09 sec)

* GRANT : Managing user access and privileges is a crucial aspect of **database**administration in **MySQL**. After creating a user account with the **CREATE USER**statement, the next step is to define what that user can do within the database.
* REVOKE : The **Revoke**statement is used to revoke some or all of the privileges which have been granted to a user in the past.
* COMMIT :used to save the changes or transaction and even the your quays

QUERY : COMMIT;

Query OK, 0 rows affected (0.03 sec)

* ROLLBACK :used to undo save the changes or transaction and even the your quays From commit

QUERY : ROLLBACK;

Query OK, 0 rows affected (0.01 sec)

* WHERE CALUSE : used to spacify condition With existing tables

QUERY : select \* from Employes where E\_id=1;

1 row in set (0.00 sec)

* ORDER BY : used to store rows in ascending and deceasing Order

QUERY : select \* from Employes ORDER BY E\_id;

4 rows in set (0.01 sec)

* AND : -its used to contain two conditions

all condition needs be true

The AND operator is used to filter records based on more than one condition

QUERY : select \* from Employes where Name like 'deep' And E\_id=1;

1. row in set (0.01 sec)

* OR : its used to contain two conditions

any one condition could be true its valid

QUERY : select \* from Employes where Name like 'janik sir' or E\_id=2;

1. rows in set (0.00 sec)

* NUll Values

we can create columns not assign the values s it would be store the null values

----------------------------------------------AGGERAGATE FUNCTION WITH QURIES--------------------------------------------

MIN() - returns the smallest value within the selected column

MAX() - returns the largest value within the selected column

COUNT() - returns the number of rows in a set

SUM() - returns the total sum of a numerical column

AVG() - returns the average value of a numerical column

1. MIN : QUERY : select min(price) from Employes;

row in set (0.01 sec)

1. MAX : QUERY : select max(price) from Employes;

row in set (0.00 sec)

1. COUNT : QUERY : select Count(E\_id) from Employes;

1 row in set (0.01 sec)

1. SUM :QUERY: select sum(price) from Employes;

1 row in set (0.01 sec)

1. AVG : QUERY : select avg(price) from Employes;

1 row in set (0.00 sec)

* LIKE OPERATOR : its store the reference of special character or number

QUERY : select \* from Employes where Name LIKE 'deep';

7 rows in set (0.00 sec)

* BETWEEN : IT used to store the range data between two given values

QUERY : select \* from Employes where price between 130 and 170;

4 rows in set (0.00 sec)

------------------------------------------------- SQL JOINS---------------------------------------------------------

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

1. INNER JOIN : its use to store matching values from two tables

QUERY : select d\_Name,Name,city,state,price from Employes inner join Department where E\_id = d\_id;

3 rows in set (0.01 sec)

1. RIGHT JOIN : store the values from the left side table and matches the values from right table

QUERY : select d\_Name,Name,city,state,price from Employes right join Department on E\_id = d\_id;

3 rows in set (0.01 sec)

1. LEFT JOIN : RETURNS the values from the right side table and matches the values from left table

QUERY : select d\_Name,Name,city,state,price from Employes left join Department on E\_id = d\_id;

10 rows in set (0.00 sec)

1. FULL JOIN : it used to store matching values and non matching values from bot tables

QUERY : select d\_Name,Name,city,state,price from Employes full join Department on E\_id = d\_id;

1. rows in set, 1 warning (0.02 sec)

-----------------------------------------------------------DAY-2---------------------------------------------------------------

* UNION : is used to store both tables values in single column

QUERY : select Name from Employes union select d\_city from Department;

5 rows in set (0.01 sec)

* CASE : when condition finds true its return values otherwise null just like if else statement

QUERY : SELECT E\_id, price,

-> CASE

-> WHEN price > 130 THEN 'the price is greater'

-> WHEN price = 130 THEN 'the price is 130'

-> ELSE 'the price is under 130'

-> END

-> FROM Employes;

8 rows in set (0.01 sec)

----------------------------------------------OPERATORS IN MYSQL-----------------------------

* Arithmetic Operators

|  |  |  |
| --- | --- | --- |
| + | Add | [Try it](https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_op_add) |
| - | Subtract | [Try it](https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_op_subtract) |
| \* | Multiply | [Try it](https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_op_multiply) |
| / | Divide | [Try it](https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_op_divide) |
| % | Modulo |  |

* Bitwise Operators

|  |  |
| --- | --- |
| & | Bitwise AND |
| | | Bitwise OR |
| ^ | Bitwise exclusive OR |

* Comparison Operators

|  |  |  |
| --- | --- | --- |
| = | Equal to | [Try it](https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_op_equal_to) |
| > | Greater than | [Try it](https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_op_greater_than) |
| < | Less than | [Try it](https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_op_less_than) |
| >= | Greater than or equal to | [Try it](https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_op_greater_than2) |
| <= | Less than or equal to | [Try it](https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_op_less_than2) |
| <> | Not equal to |  |

* ALTER

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

ALTER ADD

QUERY : alter table Employes add Pincode int;

8 rows in set (0.00 sec)

ALTER MODIFY

QUERY : alter table Employes Modify column Pincode varchar(20);

Query OK, 8 rows affected (0.18 sec)

Records: 8 Duplicates: 0 Warnings: 0

DROP ALTER :

QUERY : alter table Employes drop column Pincode;

Query OK, 0 rows affected (0.03 sec)

Records: 0 Duplicates: 0 Warnings: 0

* CONSTRAINS : that set the rules in to the database
* [NOT NULL](https://www.w3schools.com/MySQL/mysql_notnull.asp) - Ensures that a column cannot have a NULL value
* [UNIQUE](https://www.w3schools.com/MySQL/mysql_unique.asp) - Ensures that all values in a column are different
* [PRIMARY KEY](https://www.w3schools.com/MySQL/mysql_primarykey.asp) - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
* [FOREIGN KEY](https://www.w3schools.com/MySQL/mysql_foreignkey.asp) - Prevents actions that would destroy links between tables
* [CHECK](https://www.w3schools.com/MySQL/mysql_check.asp) - Ensures that the values in a column satisfies a specific conditi
* PRIMARY KEY :used to identify rows uniqly and combination of NOT null values

Query : create table Marvel(

-> m\_id int NOT NULL,

-> M\_name varchar(20),

-> PRIMARY KEY (m\_id));

Query OK, 0 rows affected (0.04 sec)

* FOREIGN KEY :matches the more than one tables primary key even can store null values

QUERY : CREATE TABLE ironman (

suit\_id INT,

PRIMARY KEY (suit\_id),

m\_id INT,

FOREIGN KEY (m\_id) REFERENCES Marvel(m\_id)

);

2 rows in set (0.01 sec)

* CHECK : its specify the each of column that matching the condition
* DEFUALT : The DEFAULT constraint is used to set a default value for a column.

The default value will be added to all new records, if no other value is specified.

* INDEXES : Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

mysql> create index idx\_name on Employes(Name,city);

Query OK, 0 rows affected (0.11 sec)

Records: 0 Duplicates: 0 Warnings: 0

* AUTO INCREMENT : that values everytime each of new record added it will be increase by 1

insert into ironman values(3,'sohan');

Query OK, 1 row affected (0.00 sec)

* DATES : this formats are used to add the date inside the tables
* DATE - format YYYY-MM-DD
* DATETIME - format: YYYY-MM-DD HH:MI:SS
* TIMESTAMP - format: YYYY-MM-DD HH:MI:SS
* YEAR - format YYYY or YY

QUERY : insert into Employes values(5,'raj sir','England','uk',94,50,'2024-12-20');

Query OK, 1 row affected (0.01 sec)

-----------------------------------------------VIEWS TEMPARORY TABLE--------------------------------------

* In SQL, a view is a virtual table based on the result-set of an SQL statement.
* Its just like normal table who contains the rows and columns
* Its also called as the virtual based table
* USECASE PURPOSE : when we manage multiple tables and data it would raise query

About data loss sometimes we can create shallow copy of that data at different place

So its again came in to use after the original data loss

QUERY : create view Example\_Data As

-> select Name,city,d\_city,d\_Name from Employes inner join Department where E\_id = d\_id;

Query OK, 0 rows affected (0.03 sec)

select \* from Example\_Data;

3 rows in set (0.01 sec)

-----------------------------------------------------FUNCTION --------------------------------------------------

* Functions in SQL Server are the database objects that contains a **set of SQL statements to perform a specific task And Returns the values as the result**

RULES OF FUNCTION IN SQL SERVER

* A function must have a name, and the name cannot begin with a special character such as @, $, #, or other similar characters.
* SELECT statements are the only ones that operate with functions.
* We can use a function anywhere such as AVG, COUNT, SUM, MIN, DATE, and other functions with the SELECT query in SQL.
* Whenever a function is called, it compiles.
* Functions must return a value or result.
* Functions use only input parameters.
* We cannot use TRY and CATCH statements in functions.

TYPES OF FUNCTION

1. System Functions : system function are also known as the built Function

String Functions (LEN, SUBSTRING, REPLACE, CONCAT, TRIM)

Date and Time Functions (datetime, datetime2, smalldatetime)

Aggregate Functions (COUNT, MAX, MIN, SUM, AVG)

Mathematical Functions (ABS, POWER, PI, EXP, LOG)

Ranking Functions (RANK, DENSE\_RANK, ROW\_NUMBER, NTILE)

1. User-Defined Functions : created by user thats called the user-defined function

TYPES OF USERDEFINED :

1. Scalar Functions : **accepts parameters, either single or multiple and returns a single value**
2. Table-Valued Function : A table-valued function returns a single rowset (unlike stored procedures, which can return multiple result shapes). Because the return type of a table-valued function is Table, you can use a table-valued function anywhere in SQL that you can use a table. You can also treat the table-valued function just as you would a table.

------------------------------------------STORE PROCEDURE IN MY SQL-----------------------------------------------

* Store procedure carry outs the SQL statement more than from one table
* Stored procedures help group one or multiple SQL statements for reuse under a common name, encapsulating common business logic within the database itself
* Its doesn’t return values like function we have to call it that procedure for execution
* SP and View has only difference that view known as the temporary table
* This will print the all the data of Employes table With

DELIMITER $$

mysql>

mysql> CREATE PROCEDURE data()

-> BEGIN

-> SELECT \* FROM Employes;

-> END$$

Query OK, 0 rows affected (0.01 sec)

mysql>

mysql> DELIMITER ;

mysql> call data();

* STORE PROCEDURE WITH JOIN WERE MULTIPLE TABLES DATA

DELIMITER $$

mysql>

mysql> CREATE PROCEDURE ViewAllData()

-> BEGIN

-> select Name,city,d\_id,d\_Name from Employes inner join Department on E\_id=d\_id ;

-> END$$

Query OK, 0 rows affected (0.01 sec)

mysql>

mysql> DELIMITER ;

mysql> call ViewAllData();