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

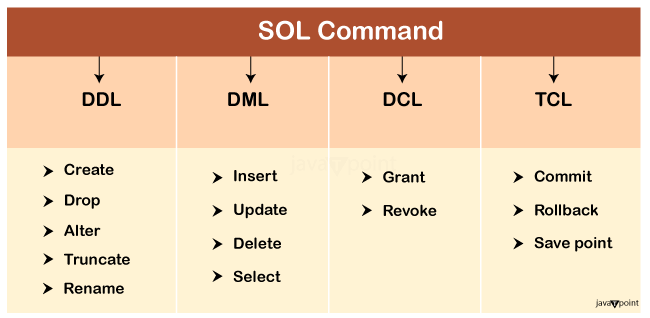
SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

1. NONSQL DATABASE

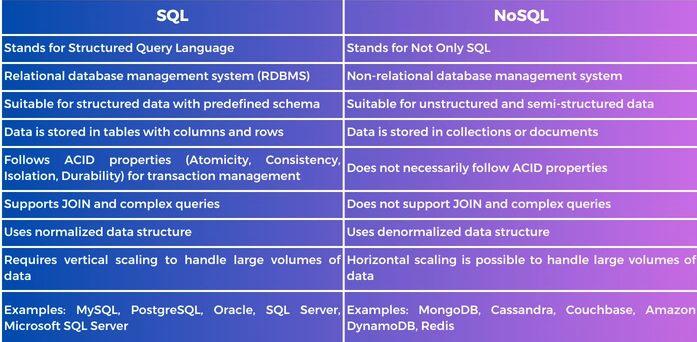
🡪 NoSQL stands for "not only SQL" and refers toa non-relational database that stores data in a non-tabular format. NoSQL databases are known for being flexible, easy to develop, and highly scalable. They are often used for applications that require large scale, high availability, and frequent data changes

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;

3 rows in set, 1 warning (0.02 sec)