INTRODUCTION AND HAND NOTES ON SQL - STRUCTURED QUERY LANGUAGE :

TOPICS TO BE COVERED :

1. WHAT IS A DATABASE
2. DATA TYPES IN SQL
3. INTRODUCTION TO MYSQL
4. SQL COMMANDS
5. CONSTRAINTS
6. PRACTICAL APPROACH TO SQL COMMANDS
7. KEYS
8. WHERE CLAUSE
9. OPERATORS
10. LIMIT CLAUSE
11. ORDER BY CLAUSE
12. AGGREGATE FUNCTIONS
13. GROUP BY CLAUSE
14. GENERAL ORDER OF COMMANDS

- ## WHAT IS A DATABASE ?

A database is a structured collection of data that is organised in a way to easily retrieve, manage, and update information.

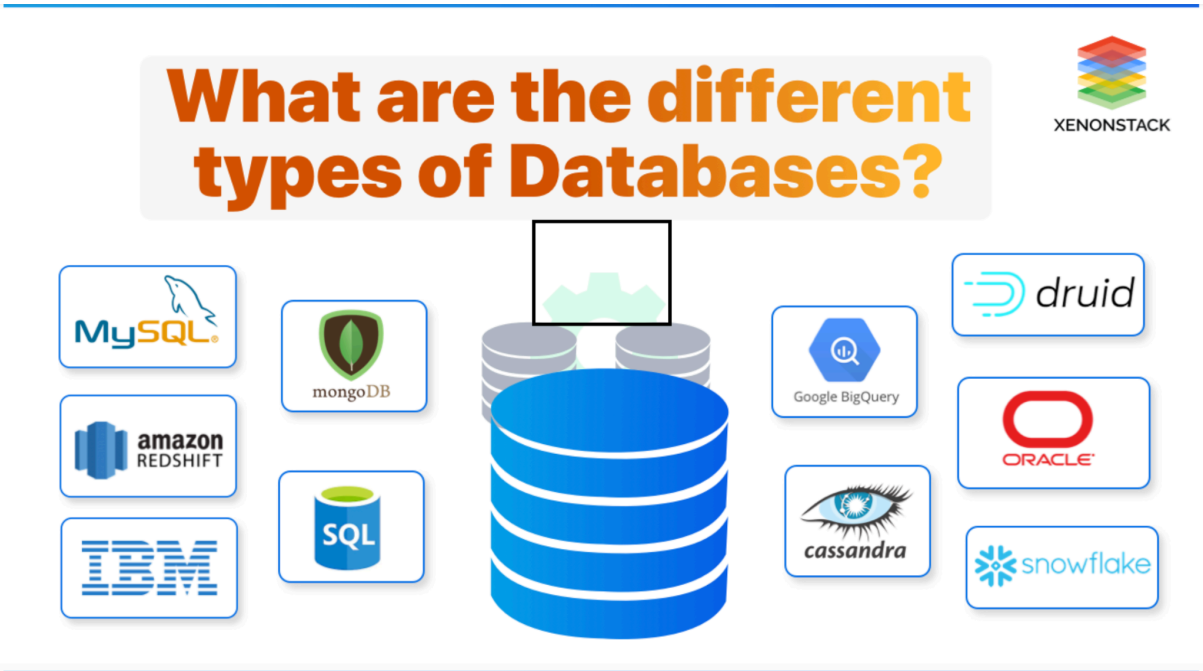
- ## TYPES OF DATABASE :

1. Relational Databases:

In Relational database or RDMS A relational database is the most commonly used database. It contains several tables, and each table has its primary key.
Due to a collection of an organised set of tables, data can be accessed easily in RDBMS.
Example: MySQL, PostgreSQL, SQLite, Microsoft SQL Server, Oracle Database.

2. Non-Relational database or NoSQL Databases:

Suitable for handling large amounts of unstructured or semi-structured data.
Types: Document-oriented (MongoDB),
Key-value stores (Redis),
Column-family stores (Cassandra),
Graph databases (Neo4j).



DATA TYPES IN SQL :

A data type is a type/category or attribute which defines the type of data in the table.
A table is composed of columns, rows,
In MySQL there are three main data types: string, numeric, and date and time.

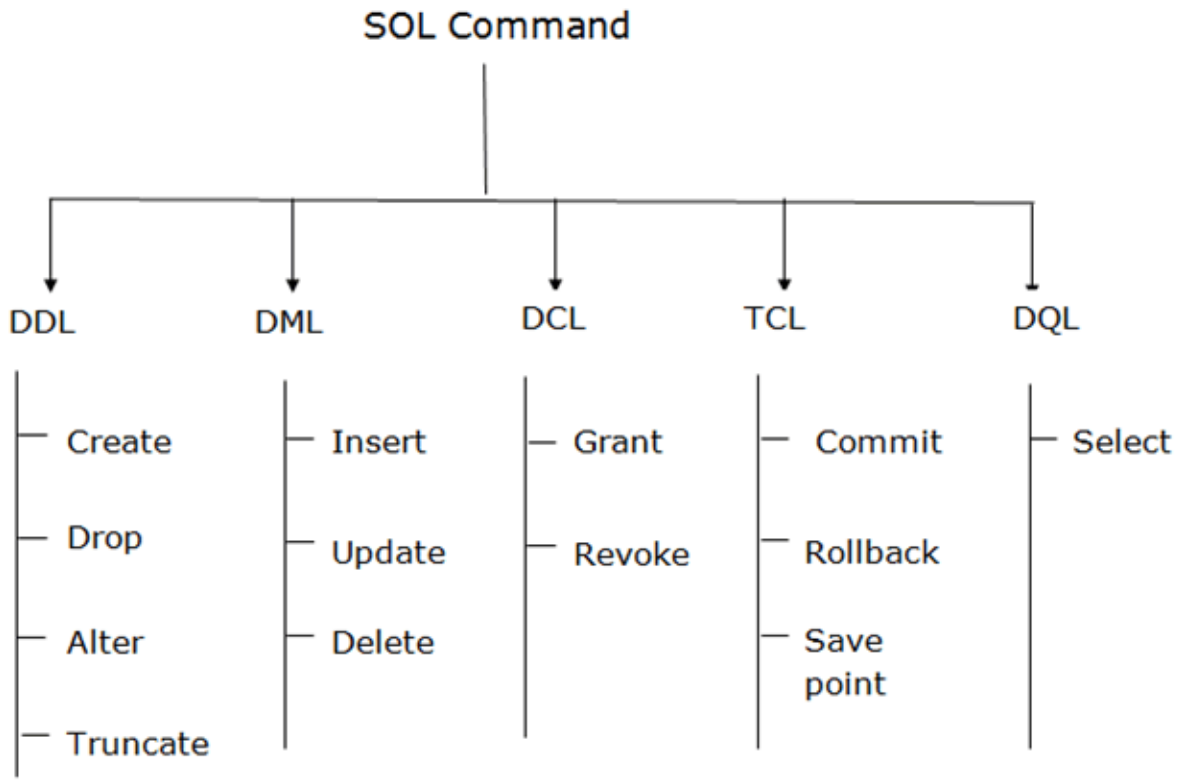
DATATYPE :

CHAR : represents a single fixed character data type but CHAR when definite with size as CHAR(SIZE) hold the data upto the mentioned fixed length .
Eg : CHAR(50)
VARCHAR: character Data type with variable length, where you have to define the length of the character and the upon defining maximum size it stores upto the defined size.
VARCHAR(50), here we can store upto 50.
INT : Keyword INT is used to define integer/numeric value data type.
FLOAT : For storing decimal values
DATE AND TIME: Date and time data type consists of
There are highly used data types, although there are numerous data types.

OPERATIONS/INTERACTIONS ON DATABASE :

SQL or Structured Query Language is used to operate on the data stored in a database. To query or interact with the database to extract information we use SQL commands or statements composed of syntax.

SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.
Types of SQL Commands
There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.



DATA DEFINITION LANGUAGE : Data definition language is used for performing operations like delete, alter/modify the data, create the structure of the table.All the commands of DDL are auto-committed, which means it permanently saves all the changes in the database.

Here are some commands that come under DDL:

1.CREATE COMMAND : SQL CREATE TABLE statement is used to create tables in a database.

If you want to create a table, you should name the table and define its column and each column's data type.

SYNTAX :

```
CREATE TABLE table_name (
      column1 datatype,
column2 datatype,
column3 datatype,
)
```

The column parameters specify the names of the columns of the table.The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

2. ALTER COMMAND: The ALTER table statement is used to add, delete, or modify columns in an existing table.The ALTER table statement is also used to add and drop various constraints on an existing table

SYNTAX :

```
ALTER TABLE table_name

 ADD column_name datatype;
```

3.DROP COMMAND: It is used to delete both the structure and record stored in the table.
SYNTAX :

```
DROP TABLE table_name;
```

4.TRUNCATE COMMAND:
A truncate SQL statement is used to remove all rows (complete data) from a table. It is similar to the DELETE statement without the WHERE clause.The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

SYNTAX:

```
TRUNCATE TABLE table_name;
```

TRUNCATE TABLE Vs DELETE TABLE: The DELETE statement is used when we want to remove some or all of the records from the table, while the TRUNCATE statement will delete entire rows from a table.

TRUNCATE TABLE Vs DROP TABLE: Drop table command can also be used to delete complete table but it deletes table structure too. TRUNCATE TABLE doesn't delete the structure of the table.

DATA MANIPULATION LANGUAGE :

Commands such as Insert, Update, Delete are a part of DML

INSERT COMMAND : It is used to insert data into the table, records can be inserted into the single row, or multiple row. Each value in the records we are inserting in a table using this statement should be of the same datatype as the respective column and satisfy the constraints of the column (if any). The values passed using an insert statement should

match the number of columns in the table or, the number of columns mentioned in the current query. If any of these conditions are not satisfied, this statement generates an error.

SYNTAX:

```
INSERT INTO table_name (column1, column2, column3, ...)

 VALUES (value1, value2, value3, ...);
```

UPDATE COMMAND :Update command is used to modify data in the table in columns or rows.

```
UPDATE table_name

SET column1 = value1, column2 = value2, ...

WHERE condition;
```

DELETE COMMAND: The delete statement is used to delete existing record in the table.

```
DELETE FROM table_name WHERE condition;
```

DCL COMMANDS:

DCL (Data Control Language)
DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

List of DCL commands:

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

Syntax:

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

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

Syntax:

```
REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```

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. You can explore more about transactions here. Hence, the following TCL commands are used to control the execution of a transaction:

BEGIN: Opens a Transaction.

COMMIT: Commits a Transaction.

ROLLBACK: Rollbacks a transaction in case of any error occurs.

SAVEPOINT: Sets a save point within a transaction.

Syntax:

SAVEPOINT SAVEPOINT_NAME;

PRACTICAL APPROACH TO:

1.CREATE DATABASE : Input the desired name for the database
*** SQL is not case sensitive, but generally Upper case should be used for better visibility ***

SYNTAX for creating a database TEMP1(Any name can be chosen)
CREATE DATABASE TEMP1;

***** Using ; is mandatory, it acts as a full stop for the sql command

DELETE :

To delete database : Drop command deletes the entire table along with the design/structure of the table .

```
DROP DATABASE TEMP1;
```

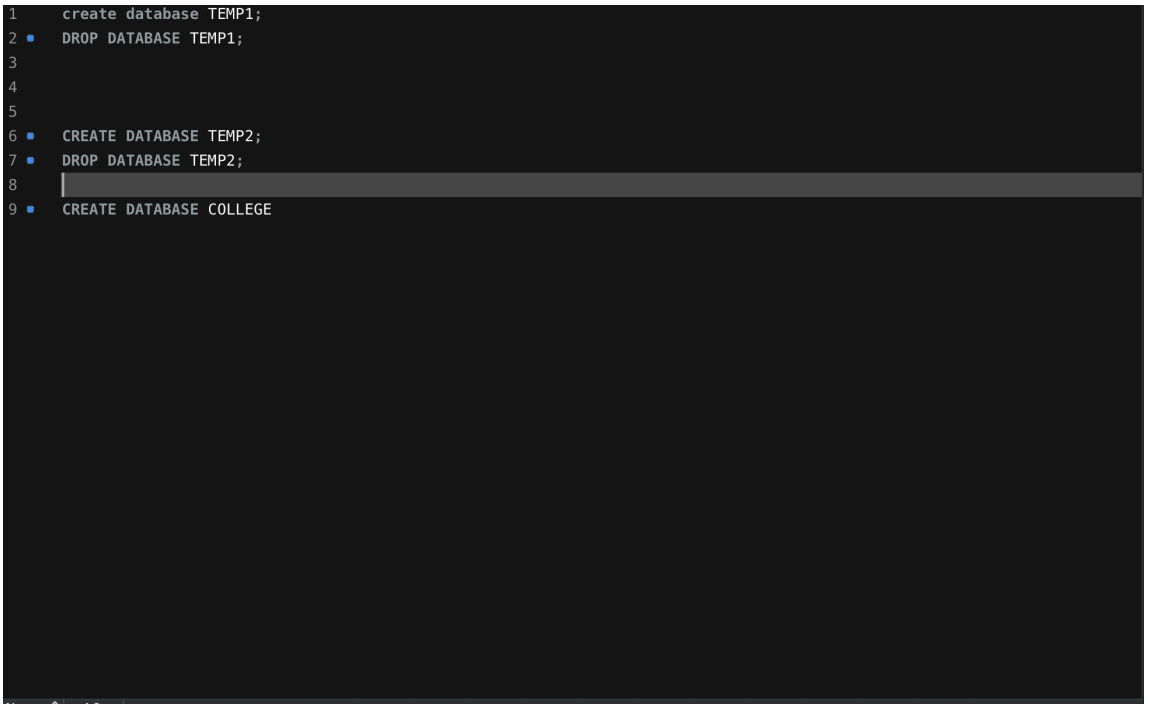


TABLE RELATED QUERIES :

**** Now, before creating table, let's call the database first where we want to create our table by using USE command , followed by create table command :

```
CREATE TABLE table_name(

column1 datatype,

column2 datatype,

column3 datatype
);
```

The column parameters specify the names of the columns of the table.

The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

In simple words, we have defined the schema of the table, or designed the table.

1.CREATING A TABLE :

```
CREATE TABLE student(
rollno INT PRIMARY KEY,
name VARCHAR(50)
);
```

2.SELECT COMMAND :

Select command to view entire data in the table use select command

```
SELECT * from TABLE_NAME;
```

To select data from particular columns, enter data base

Ex : Consider a table Customers,

To extract all the data from CustomerName, City run the command as below :

```
SELECT CustomerName,City FROM Customers;
```

Number of Records: 91

CustomerName	City
Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim
Blondel père et fils	Strasbourg
Bólido Comidas preparadas	Madrid

3. DISTINCT KEYWORD IN SELECT STATEMENT :

Using the distinct keyword gives the unique value for instead of the repeating values
For example, here we will get only distinct , different or unique values .

In the example below we have 21 distinct records,

```
SELECT DISTINCT Country FROM Customers;
```

Here , we get 21 distinct records,
whereas in SELECT COUNTRY FROM Customers, give a total of 91 records.

Number of Records: 21

Country
Argentina
Austria
Belgium
Brazil
Canada
Denmark
Finland
France

Number of Records: 91

Country
Germany
Mexico
Mexico
UK
Sweden
Germany
France
Spain

INSERT INTO COMMAND :

The SQL INSERT INTO Statement is used to add new rows of data into a table in the database. Almost all the RDBMS provide this SQL query to add the records in database tables.

Each value in the records we are inserting in a table using this statement should be of the same datatype as the respective column and satisfy the constraints of the column (if any). The values passed using an insert statement should match the number of columns in the table or, the number of columns mentioned in the current query. If any of these conditions are not satisfied, this statement generates an error.

SYNTAX:

There are two basic syntaxes of the SQL INSERT INTO statement which are shown below :

1. **With Column names**

```
INSERT INTO TABLE_NAME
(column1, column2...columnN)
VALUES
(value1, value2...valueN);
```

EXAMPLE : Consider a table studentsdata with columns rollno and name, marks

```
CREATE TABLE studentsdata;


INSERT INTO studentsdata
    (rollno, name, marks)
 VALUES
    (1, "ADITI", 23),
    (2, "PRIYANKA", 85),
    (3, "ZUNAID",87),
    (4, "ADITI", 90),
    (5, "YAMINI", 78),
    (6, "ANI", 67),
    (7, "EKTA", 55)
;

OUTPUT :
1|ADITI|23
2|PRIYANKA|85
3|ZUNAID|87
4|ADITI|90
5|YAMINI|78
```

```
6|ANI|67
7|EKTA|55
```

Here, column1, column2, column3,...columnN are the names of the columns in the table into which you want to insert the data.

2. **Without column names.**

   Make sure the order of the values is in the same order as the columns in the table.

```
INSERT INTO TABLENAME

VALUES(value1, value2……valueN);
```

UPDATE COMMAND ;

Update command is used to update existing data in the row/rows,
“Set” Keyword is used to modify the old data into new data.

SYNTAX:

```
UPDATE TableName,

Set col1 = value 1, col2 = value2

WHERE Condition;
```

Example,

If the table studentsdata has rollno, name, city, grade, marks and we want to upgrade grade “A” to grade “0” , we will run the following query

```
SELECT studentsdata,

SET Grade = “0”

WHERE Grade = “A”;
```

ALTER COMMAND :

Alter command is used to alter or modify the design of the schema or table , we can perform certain actions like Add column, drop column, rename column,

Let's add a column into the table:

SYNTAX :

```
ALTER Table_name
ADD Column_name datatype constraint;
```

RENAME TABLE :
SYNTAX :

```
ALTER Table_name
RENAME TO new_table_name;
```

Let's drop a column

```
ALTER Table_name
DROP Column_name;
```

CHANGE Column(rename:

```
ALTER TABLE table_name
CHANGE COLUMN old_name new_name new_datatype new_constrain
```

MODIFY COMMAND (column-modify datatype/constraint):

```
ALTER TABLE table_name
MODIFY col_name new_datatype new_constraint
```

TRUNCATE COMMAND :

```
TRUNCATE TABLE table_name;
```

Example:

```
TRUNCATE TABLE table_name studentsdata;
This will give blank values under the columns
```

DELETE COMMAND :

As the name suggests, the delete command is used to delete the data from the existing rows.

```
DELETE FROM table_name WHERE condition;
```

Consider the table studentsdata, where we want to delete the record of the student whose marks are less than 60, the command will be written as follows:
Now, let's write the code for deleting the rows of students whose marks are less than 60

BEFORE :

```
1|ADITI|23
2|PRIYANKA|85
3|ZUNAID|87
4|ADITI|90
5|YAMINI|78
6|ANI|67
7|EKTA|55
```

Now , the command and result will be :
Here the students with less than marks 60 will be deleted.

```
DELETE FROM studentsdata WHERE marks < 60;

OUTPUT :
2|PRIYANKA|85
3|ZUNAID|87
4|ADITI|90
5|YAMINI|78
6|ANI|67
```

KEYS IN SQL:

Keys in SQL refer to the special columns in tables inside the database , there are 2 important keys in the table

1. PRIMARY KEY

2. FOREIGN KEY

1. PRIMARY KEY : In simple terms, PRIMARY KEYS = UNIQUE + NOT NULL
It is a column or set of columns which uniquely identifies each row,
Primary key should contain the unique values and no null values, or in other words, the column which is selected as the primary key should have all the unique values and no null values.

Eg , customer_id, phone number in a table

2.FOREIGN KEY :

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.
The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.
It helps establish the connection between both the tables .
There can be multiple foreign keys
They can be duplicate and null values.

** There are various other keys such as Super key, Candidate keys in SQL , Primary key is a type of Candidate key

CONSTRAINTS:

Constraints in SQL are used to specify rules or conditions in a table, we do that by using Keywords.

There are multiple constraints in SQL but let's look at the most important constraint :

1. NOT NULL : Where we define not null means the columns cannot have the null value.The NOT NULL constraint enforces a column to NOT accept NULL values. This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.
2. UNIQUE : As the name says, all the values in the column should be uniiquer or different, no duplicate values
3. PRIMARY KEY : Where we are keeping the values as unique and not null

```
CREATE TABLE Temp1(
```

```
 Id INT,

 name VARCHAR(50),

 city VARCHAR(50),

 age INT,

PRIMARY KEY(Id, name)

);
```

We can make a combination of 2 columns as the primary key because the values within the columns can be duplicate but the combination can't be.

4. FOREIGN KEY: It is used to define the foreign key for a table and establish the relationship .
   We use "references" keyword to define relationship between the foreign key and table and primary key of another table

```
CREATE TABLE Temp(

 Cust _id INT,

FOREIGN KEY(cust_id) references CUSTOMER(id)
```

Here cust_id = Foreign key of the new table, the one mentioned before reference keyword.
CUSTOMER(id) = Table customer and id is the primary key of the customer table

5. DEFAULT: Is used to set default values within the columns.

```
SALARY INT default 25000
```

6. CHECK : This is used to limit the values inside the column

WHERE CLAUSE IN SQL :

The WHERE clause is used to filter records.
It is used to extract only those records that fulfill a specified condition or an extra condition or define few conditions .
Where clause is used with the select statement

SYNTAX :

```
SELECT column1, column2, ...

FROM table_name

WHERE condition;
```

Here , conditions will be defined by the user.

Example,
Consider a table with rollno, name and marks of the students in the table studentsdata , now to find out students with the marks above 20 use the command as follows:

```
SELECT * FROM studentsdata WHERE marks > 40;
```

In the table studentsdata, the data is as follows :

```
1|ADITI|23
2|PRIYANKA|85
3|ZUNAID|87
4|ADITI|90
5|YAMINI|78
```

Now, executing our select statement with the where clause for students with marks above 40 , it will yield the result as follows:

OUTPUT :

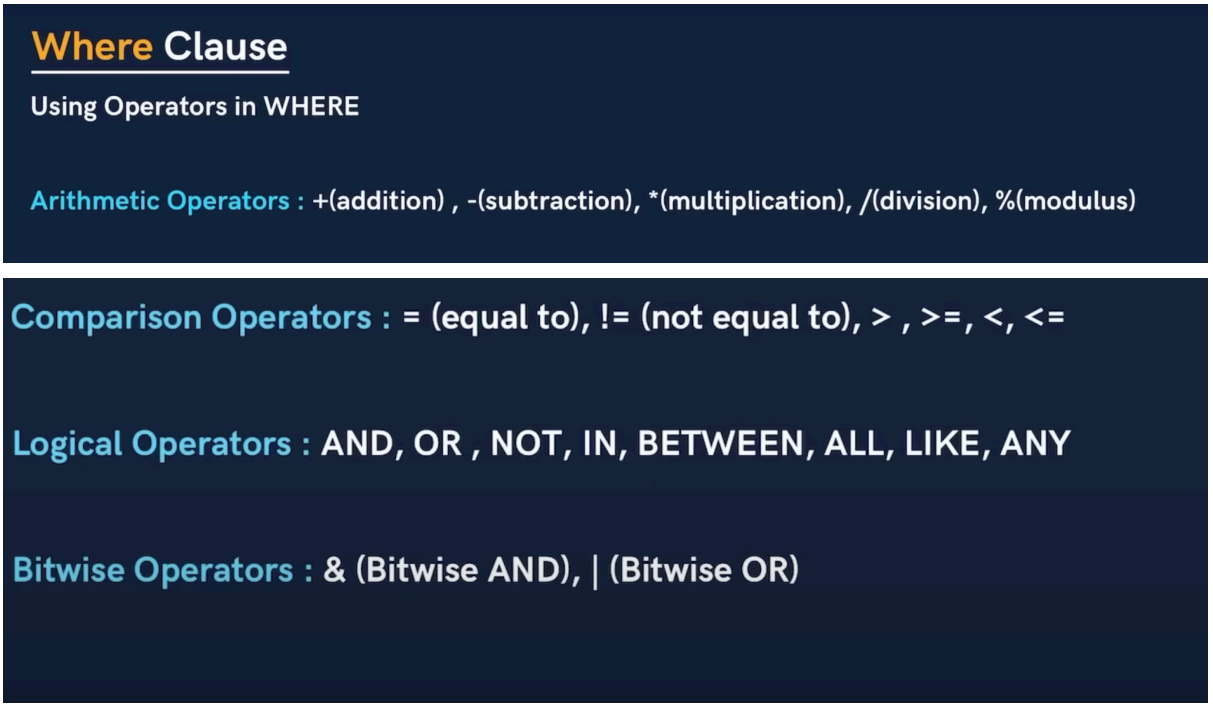
```
2|PRIYANKA|85
3|ZUNAID|87
4|ADITI|90
5|YAMINI|78
```

For marks > greater than 80 , it will give the remaining 3 records instead of 4, removing the data for Yamini as well.

```
2|PRIYANKA|85
3|ZUNAID|87
4|ADITI|90
```

OPERATORS :

Operators used in SQL are similar to the mathematical operators .



Here : % gives the remainder as the result from the division

Example for AND , OR operator

Consider the table studentsdata with rollno, name, marks, city ,

1. AND operator : It is used to execute both the conditions in the statement

```
SELECT *  FROM studentsdata WHERE marks > 80 AND city =”Mumbai”
```

   This gives the data for students with marks greater than 80 and belonging only from mumbai.

2. OR operator : With the usage of OR keyword , either of the conditions, if satisfied yields the result.

```
SELECT *  FROM studentsdata WHERE marks > 90 OR  city =”Mumbai”
```

LIMIT CLAUSE :

Limit clause in SQL is used to set an upper limit on a number of rows to be returned, in other simple words, limit is used to specify the number of records we want to return. It gives and counts the record from the Top .

SYNTAX :

```
SELECT Col1,Col2 FROM table_name
LIMIT number
```

In Table containing multiple rows, if we want to extract top 4 records put LIMIT 4

```
SELECT * FROM studentsdata LIMIT 4;

1|ADITI|23
2|PRIYANKA|85
3|ZUNAID|87
4|ADITI|90
```

For extracting top 3 records of the students where marks of the students are less than 60, The command and output will be as follows:

```
SELECT *
FROM studentsdata
WHERE marks<60
LIMIT 3;
```

ORDER BY CLAUSE:

- The MYSQL ORDER BY Clause is used to sort the records in ascending or descending order at times depending upon the data

  Syntax:

```
SELECT *
FROM tables
WHERE conditions
ORDER BY expression [ ASC | DESC ];
```

1.WHERE conditions: It is optional. It specifies conditions that must be fulfilled for the records to be selected.

2.ASC: It is optional. It sorts the result set in ascending order by expression (default, if no

modifier is provider).

3.DESC: It is also optional. It sorts the result set in descending order by expression.

ORDER BY: WITHOUT USING ASC/DESC ATTRIBUTE

- If you use MySQL ORDER BY clause without specifying the ASC and DESC modifier then by default you will get the result in ascending order.

```
SELECT *
FROM officers
WHERE address = 'Lucknow'
ORDER BY officer_name;
```

AGGREGATE FUNCTIONS :

The aggregate functions perform a set of calculations. They can be classified as follows:

MAX()
MIN()
SUM()
AVG()
COUNT()

GROUP BY CLAUSE:

The GROUP BY Clause is used to collect data from multiple records and group the result by one or more columns .

The main purpose of grouping the records of a table based on particular columns is to perform calculations on these groups. Therefore, the GROUP BY clause is typically used with aggregate functions such as SUM(), COUNT(), AVG(), MAX(), or MIN() etc.

NOTE : Whichever columns we are using in the select statement without the use of aggregate function should be used/mentioned within the GROUP BY statement for it to work.

SYNTAX :

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

Example : To count the number of students in the city, here we will use the COUNT() function, as the GROUPBY is used mostly with aggregate functions,

The approach here is that it will group students by every city and will give the count of number of students in each city based on roll no. and will result in the column of the same.

```
Select City, Count(rollno)
FROM studentsdata
GROUP BY City
```

city	count(rollno)
Delhi	3
Mumbai	2
Pune	1

GENERAL ORDER OF THE CLAUSES