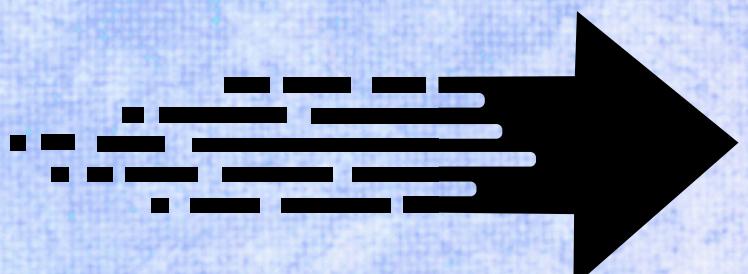
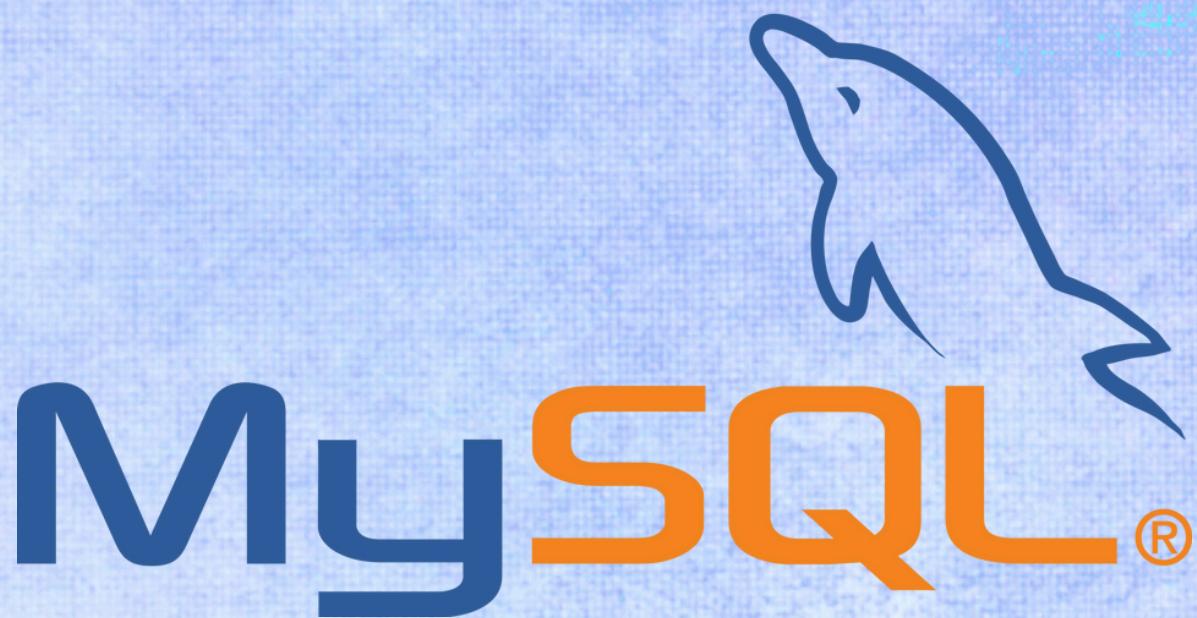


SQL HANDBOOK

"Master SQL: A Journey Through Handwritten Precision"
"Your Essential Guide to Database Mastery, Handcrafted

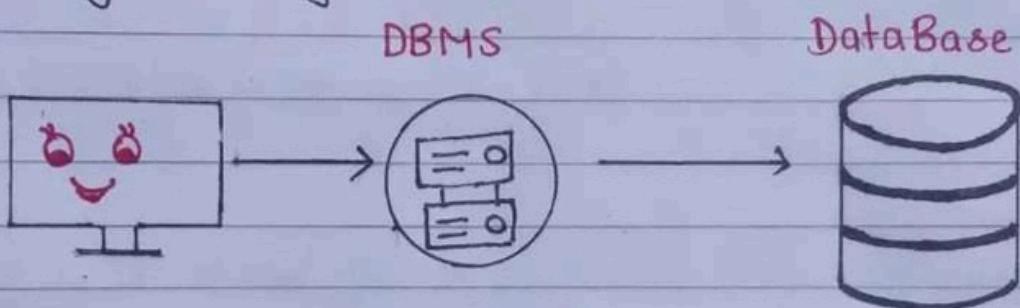


@harshit harsh

SQL Introduction

Database :- It is an organized collection of data so that it can be easily accessed.

To manage these databases, DBMS (Database Management System) are used.



Types of DBMS :-

@programmer-girl--

- Relational DBMS
- Non-Relational DBMS

Relational DBMS :- In this DBMS, data stored in table Format.

RollNo	Name	Class
1	Jai	5th
2	Amar	7th
3	Anuj	5th
4	Ram	8th

(Relational DBMS)

For Example :- MySQL, Oracle.

Non-Relational DBMS :- In this DBMS data is stored in key-value pair.

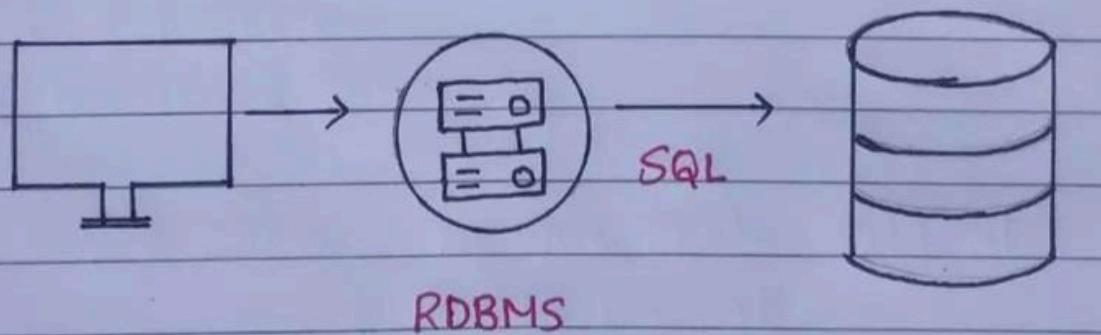
```
{  
  "RollNo": 1,  
  "Class": "5th",  
  "name": "Jai"  
}
```

(Non-Relational DBMS)

For Example :- MongoDB, Redis

SQL :- It stand for Structured Query language.

SQL is used for update, delete, insert data in table or Relational DataBase.



@programmer-girl--

SQL CREATE Command:-

It is used for creating Tables.

Syntax :-

```
CREATE TABLE tableName (  
    column1 datatype,  
    column2 datatype,  
    ...  
)
```

SQL Keywords are **case-insensitive**.

In MySQL, case-insensitive is an option you can turn on and off.

For Example:-

```
CREATE TABLE user (  
    FirstName varchar,  
    LastName varchar,  
    Email_id varchar,  
    password varchar  
)
```

FirstName	LastName	Email_id	password
-----------	----------	----------	----------

@programmer-girl--

SQL INSERT INTO Command :-

It is used to insert ^{data} into tables.

Syntax:-

```
INSERT INTO tableName (column1,  
                      column2 ... )  
VALUES (value1, value2 ... );
```

- A row of database table is known as record or a tuple.
- A column of database table is known as an attribute.

@programmer-girl--

For Example :-

```
INSERT INTO USER (FirstName,  
                  LastName, Email-id, Password)  
VALUES (Jai, Sharma, abc@gmail.com,  
        abc#123);
```

FirstName	LastName	Email-id	Password
Jai	Sharma	abc@gmail.com	abc#123

How to Insert Multiple Record (row, tuple):-

```
VALUES (Jai, Sharma, abc@gmail.com, 123),  
(Jaya, Sharma, xyz@gmail.com, abc);
```

SQL SELECT Command :-

It is used to retrieves data from the table.

Syntax :-

```
SELECT Column1, Column2  
From tableName;
```

- To Select complete table, Use * (Star)

```
SELECT *  
From tableName;
```

Example :-

@programmer-girl--

FirstName	LastName	Password
Jai	Kumar	123
Jaya	Singh	abc
Amit	Sharma	xyz

Table :- USER

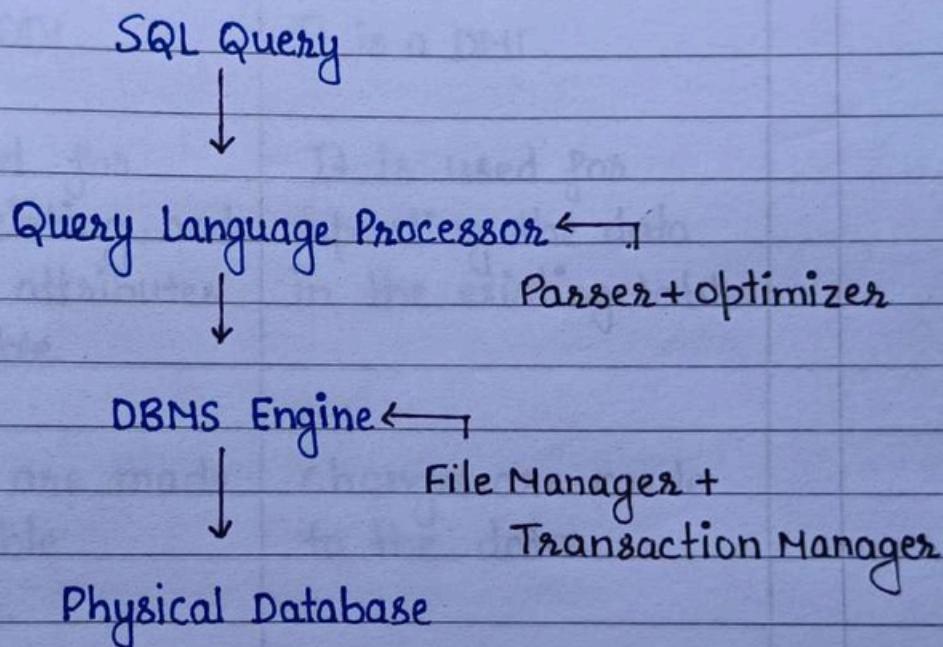
Command :-

```
SELECT FirstName  
From USER;
```

Output :-

FirstName
Jai
Jaya
Amit

How Does SQL Work



Parsing :- In this process, Query statement is tokenized.

Optimising :- It optimize the best algorithm for the byte code.

@programmer-girl--

FROM :- It is used to specify the tables from which data fetched.

WHERE :- It is used to filter records based on the given condition.

JOIN :- It is used to combine data from tables based on a common field.

GROUP BY :- It is used to group records based on our requirement.

HAVING :- It is used to filter groups.

ORDER BY :- It is used to sort the data in ascending or descending order.

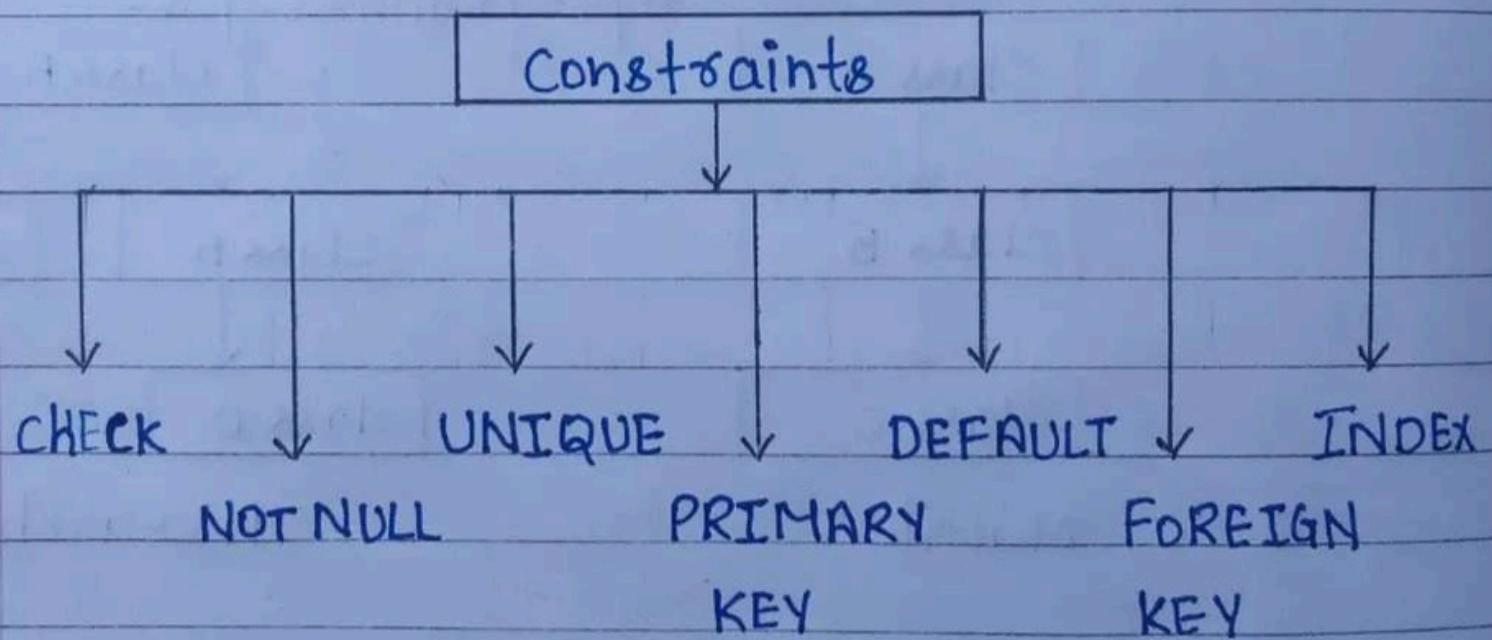
SELECT :- It is used to retrieves data from the table.

LIMIT :- It is used to specify how many rows are returned.

@programmer-girl--

SQL

Constraints



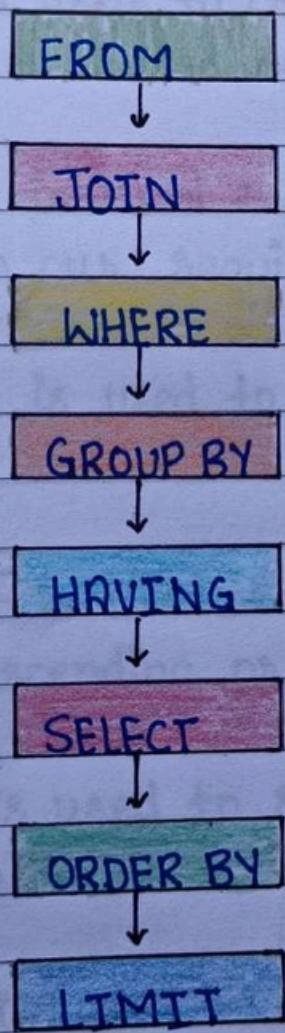
@programmer-girl--

These Constraints also known as Integrity constraints.

- SQL Constraints:- Constraints are the rules and restrictions applied on the data in a table.
- NOT NULL:- Value cannot be Null in a column.
- UNIQUE:- Value cannot be same in a column.
@programmer-girl--
- PRIMARY KEY:- Used to uniquely identify a row.
- FOREIGN KEY:- References a row in another table.
- CHECK:- Satisfies a specific condition
- DEFAULT:- Set default value
- CREATE INDEX:- used to speedup the read process.

SQL Query Execution

Order →



@programmer-girl--

Difference Between ALTER And UPDATE

ALTER	UPDATE
It is a DDL.	It is a DML.
It is used for adding, deleting, and modifying attributes of the table.	It is used for updating the data in the existing table.
Changes are made to the table structure.	Changes are made to the data.
By default, all the values in the tuple are initialized as null if the ALTER command is used.	It sets the specified value to the tuple if Update Command is used.

SQL SELECT DISTINCT :-

It is used to return only unique values from a specified column in a table.

Syntax :-

```
SELECT DISTINCT column-name  
FROM table-name;
```

Example :-

FirstName	LastName	Password
Jai	Kumar	123
Jaya	Singh	123
Amit	Sharma	xyz

Table :- USER

Command :-

```
SELECT Distinct Password  
FROM USER;
```

Output :-

Password
123
xyz

@programmer-girl--

SQL WHERE CLAUSE :-

It is used to filter rows in a table based on a specified condition.

Syntax :-

```
SELECT column-name  
FROM table-name  
WHERE condition;
```

Example :-

FirstName	LastName	Age
Jai	Kumar	19
Jaya	Singh	20
Amit	Sharma	21

Table :- USER

Command :-

```
SELECT FirstName, LastName  
FROM USER  
WHERE Age > 20;
```

Output :-

FirstName	LastName
Amit	Sharma

SQL AND :-

The AND operator returns true if both conditions are true, and false otherwise.

Syntax :- WHERE condition1 AND condition2;

SQL OR :-

It returns true if either condition is true, and false if both conditions are false.

Syntax :- WHERE condition1 OR condition2;

SQL NOT :-

It returns the opposite of a condition.

Syntax :- WHERE NOT condition;

@programmer-girl--

AND, OR, NOT operators are used to combine conditions in a where clause to create more complex filtering conditions.

SQL ORDER BY :-

It is used to sort the result of a query in ascending or descending order.

Syntax :-

```
SELECT column1, column2, ...
FROM table-name
ORDER BY column1 [ASC] [DESC], column2
[ASC] [DESC], ...
```

ASC :- It is used to sort the result in ascending order.

DESC :- It is used to sort the result in descending order.

@programmer_girl--

Example :-

FirstName	LastName	Age
Jai	Kumar	49
Jaya	Singh	20
Amit	Sharma	21

Table :- user

Command :-

```
SELECT * FROM user
ORDER BY Age;
```

Output :-

FirstName	LastName	Age
Jaya	Singh	20
Amit	Sharma	21
Jai	Kumar	49

INSERT INTO :-

It is used to insert data into a table.

Syntax :-

```
Insert into tablename (column1, column2, ..)
values (value1, value2, ..);
```

Note :- There must be the same number of values as the same number of columns specified.

Example :-

FirstName	LastName	Age
Jai	Kumar	10
Jaya	Singh	15
Amit	Sharma	20

Table :- user

Command :-

```
Insert into user (FirstName, LastName, Age)
values (abc, xyz, 25);
```

@programmer-girl--

Output :-

FirstName	LastName	Age
Jai	Kumar	10
Jaya	Singh	15
Amit	Sharma	20
abc	xyz	25

SQL NULL Values :-

It is used to represent missing or unknown data.

Note:- Null is different from zero or empty string.

Insert Null value :-

```
INSERT INTO tablename(column1,column2,...)  
VALUES (value1,NULL,..);
```

To Check for Null Values :-

IS NULL :-

```
SELECT column1,column2,...  
From table-name  
WHERE column2 IS NULL;
```

IS NOT NULL :-

```
SELECT column1,column2,...  
FROM table-name  
WHERE column1 IS NOT NULL;
```

@programmer-girl

SQL UPDATE :-

It is used to modify existing data in table.

Syntax :-

```
UPDATE table-name  
SET column1 = value1, column2 = value2, ...  
WHERE some-column = Some-value;
```

SET :- It is used to specify the column and values to update.

Example :-

FirstName	LastName	Age
Jai	Kumar	10
Jaya	Singh	15
Amit	Sharma	20

Table :- Users

Command :-

```
UPDATE USERS  
SET age = age+1;
```

@programmer.girl--

Output :-

FirstName	LastName	Age
Jai	Kumar	11
Jaya	Singh	16
Amit	Sharma	21

SQL DELETE :-

It is used to remove existing record from a table in a SQL Database.

Syntax :-

DELETE FROM tablename WHERE condition;

Note:- This operation is not reversible, so be careful when using DELETE statements!

@programmer-girl--

SQL Wildcards :-

Wildcards are special characters used in SQL 'LIKE' operator to search for a specific pattern in a column of a table.

- The percent sign (%) represents zero, one or multiple characters.
- The underscore sign (-) represents one, single character.

SQL LIKE :-

It is used to search for a specific pattern in a column of a table.

Syntax :-

```
SELECT column1, column2, ...
FROM table-name
WHERE column-name LIKE pattern;
```

@programmer-girl--

SQL IN :-

It is used to specify multiple values in a WHERE clause for filtering data.

Syntax :-

```
SELECT column1, column2, ...
FROM table-name
WHERE Columnname IN (value1,value2);
```

SQL Between :-

It is used to filter data based on a range of values in a WHERE clause.

Syntax :-

```
SELECT column1, column2, ...
FROM table-name
WHERE column-name Between
value1 AND value 2;
```

SQL Alias:-

@programmer-girl...

It is used to give a temporary name to a table or a column in a query.

Syntax:-

```
SELECT column-name AS alias-name
From table-name;
```

SQL UNION Operator :-

It is used to combine the result sets of two or more SELECT statements into a single result set.

NOTE:- It returns only distinct rows.

Syntax:-

```
SELECT column-name
FROM table-name-1
UNION
SELECT column-name
FROM table-name-2;
```

Example:-

SELECT A		SELECT B	
1		3	
2	Union	4	
3		5	

1	
2	
3	
4	
5	

SQL GROUP BY :- It is used to group rows that have the same values into summary rows, Like "Find the number of customers in each city."

SQL HAVING :- It is used to filter the results of a 'GROUP BY' query Based on the values of an aggregate function.

SQL EXISTS :- It is used to check if a subquery returns any rows.

It returns a boolean value.

SQL ALL :- It is used to compare a value with the result of a subquery.

It returns true if the value is true for all elements.

SQL ANY:- It is also a comparison operator.
It returns true if the value is true for at least one element.

@programmer-girl--

Types Of Error In Sql

- **Syntax Errors:-** These occur when SQL statements do not follow the correct syntax and structure of the language.
- **Semantic Errors:-** These occur when the SQL statement is grammatically correct, but does not produce the desired result due to incorrect logic.
- **Constraint violations:-** These occur when the SQL violates one or more constraints on the database.
- **Datatype Errors:-** These occur when data is inserted in a way that does not match the expected data type.
For Example:- Insert a string into a numeric field.
- **Transaction Errors:-** These occur when a transaction fails due to problems with locking.

@programmer-girl-

SQL JOINS

SQL JOINS:- These statements allow us to access information from two or more tables at once.

They also keep our database normalized.

Types of Joins :-

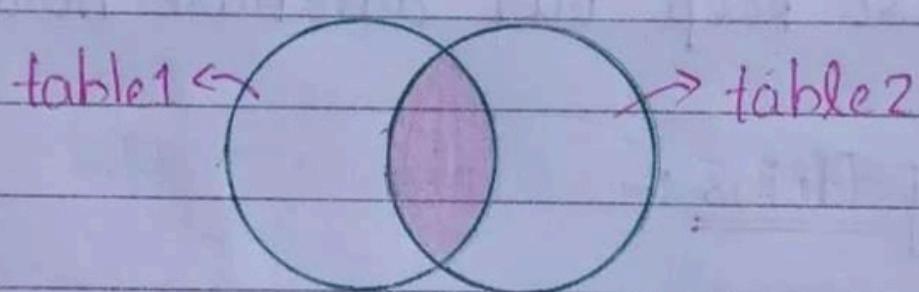
- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

@programmer-girl--

Swipe → ...

INNER JOIN

INNER JOIN :- It returns dataset that have matching values in both tables.



SELECT column-name
FROM table1

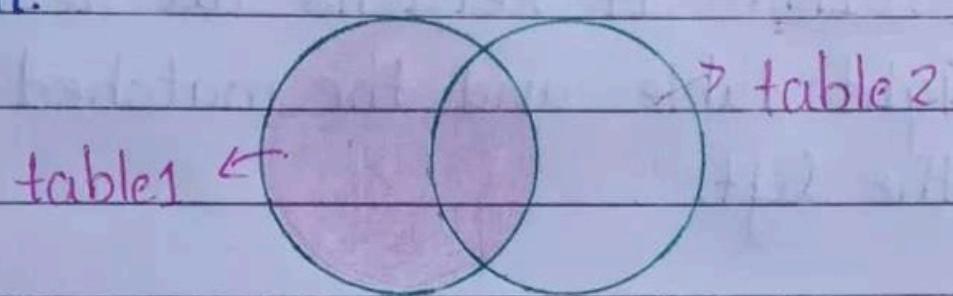
INNER JOIN table2
ON table1.Column-name = table2.Column-name;

@programmer-girl--

Swipe →...

LEFT JOIN

LEFT JOIN:- It returns all records from the left table and matched records from the right.



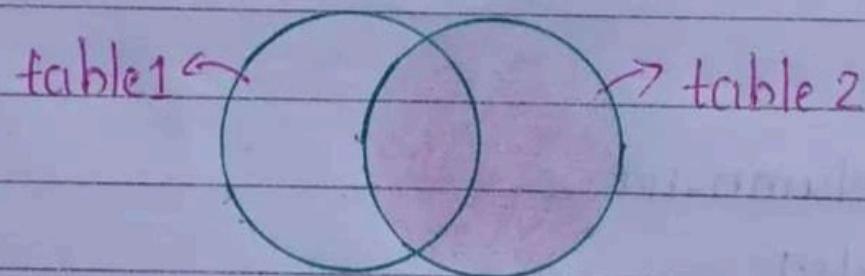
```
SELECT column-name  
FROM table1  
LEFT JOIN table2.  
ON table1.column-name = table2.column-name
```

@programmer-girl--

Swipe → ...

RIGHT JOIN

RIGHT JOIN:- It returns all records from the right table and the matched records from the left.



SELECT Column-name

FROM table1

RIGHT JOIN table2

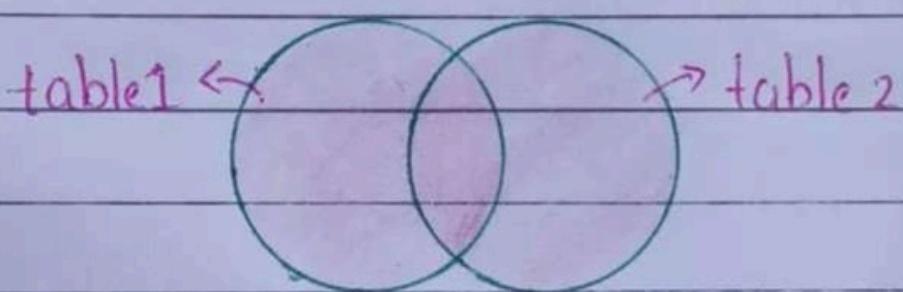
ON table1.column-name = table2.column-name;

@programmer_girl..

Swipe →...

FULL JOIN

FULL JOIN:- It returns all records when there is a match in either the left table or right table.



SELECT column-name

FROM table1

Full OUTER JOIN table2

ON table1.column-name = table2.column-name

WHERE condition.

@programmer-girl--

* Primary Key and Foreign Key :-

→ Primary key :- It is a unique value that is used to identify a row in a table.

If you are thinking about unique constant, then you must to know the difference.

* Unique constant can store null value also.
Primary key cannot store null value.

→ Syntax :-

@programmer-girl--

create table tablename (

column1 datatype,

column2 datatype,

Primary key (column name)

);

→ A table can contain only one primary key.

- Foreign Key:- It is a key which is used to link two table together.
- Foreign key can have multiple null values.
- Syntax:-

@programmer-girl--

```
Create table tablename (  
    column1 datatype,  
    column2 datatype,  
    - - -
```

Foreign key [column1, - -].

References [primary key tablename]

) ;

→ LET'S TAKE EXAMPLE:-

Student			
Primary Key ↪	ROLLNO	Name	Address
1	RAM	Delhi	
2	Lalit	Mumbai	
3	RAM	Delhi	

Name and Address are not unique.

Here RollNo is unique.

So we discussed before, Primary Key is unique and not null value.

Result table

Maths	Science	English
10%	80%	70%
90%	60%	20%
70%	50%	21%

	Maths	Science	English
1	10%	80%	70%
2	90%	60%	20%
3	70%	50%	21%

→ You want to check RollNo 3 marks in maths, Foreign key came into picture
 Student table + Result table.
 As we have to add primary key in Result table which act as Foreign key.

	Maths	Science	English	RollNo	Foreign Key
1	10%	80%	70%	1	
2	90%	60%	20%	2	
3	70%	50%	21%	3	

* Commit and Rollback :-

→ Commit and Rollback are transaction statements.

* Commit :-

It is used to storing changes permanently performed by a transaction.

→ Commit Syntax :-

COMMIT;

@programmer-gisl...

* Rollback :-

It is used for reverting changes performed by a transaction.

→ Rollback Syntax :-

Rollback;