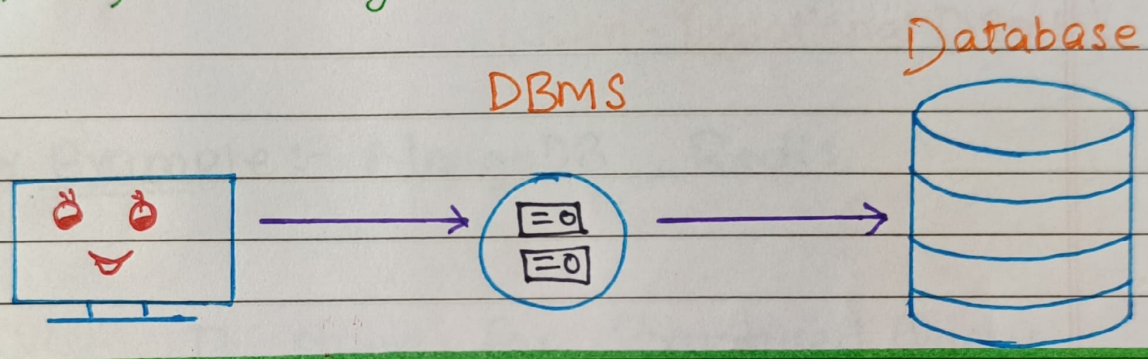
SQL Introduction

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 :-

ATUL KUMAR (LinkedIn / Twitter
NOTES GALLERY / CODING BUCS (Telegram)

- Relational DBMS
- Non-Relational DBMS

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

Roll No	Name	Class
1	Jai	5th
2	Amar	7th
3	Anuj	5th
4	Ram	8th

For Example :- My SQL, Oracle.

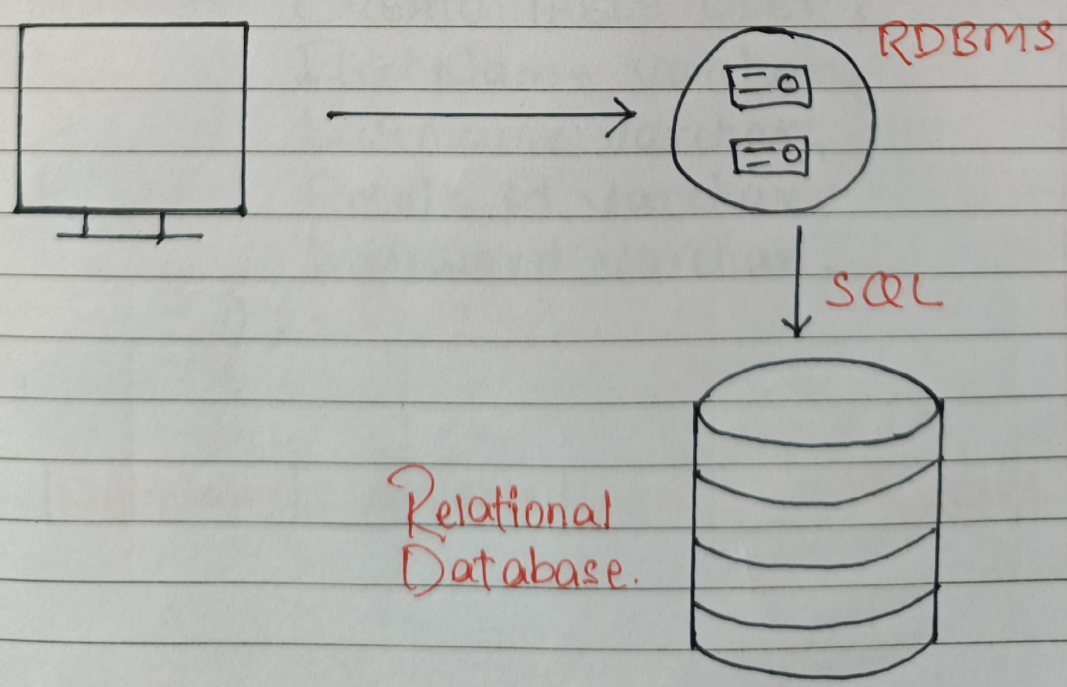
Non-Relational DBMS :- In this DBMS data is stored in Key-value pair.

```
{
  "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.



SQL CREATE Command :-

It is used for create 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.

CLASSTIME Pg. No.
Date / /

SQL INSERT INTO Command :-

It is used to insert data into tables.

Syntax :-

```
INSERT INTO tableName (column 1,
    column 2 ... )
values ( value 1 , value2 ... );
```

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

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	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 Column 1, Column 2
From tableName;
```

- To Select Complete table, use * (star)

```
SELECT *
From tableName;
```

Example:-

FirstName	Last Name	Password
Jai	Kumar	123
Jaya	Singh	abc
Amit	Sharma	xyz

Table :- USER

Command:-

```
SELECT FirstName
From USER;
```

Output:-

ATUL KUMAR (LinkedIn / Twitter)
NOTES GALLERY / CODING BUGS (Telegram)

First Name
Jai
Jaya
Amit

How Does SQL Work

SQL Query
↓
Query Language Processor ← Passer + Optimizer
↓
DBMS Engine ← File Manager + Transaction Manager.
↓
Physical Database

Passing :- In this Process, Query statement is tokenized.

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

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

CLASSTIME Pg. No.
Date / /

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.

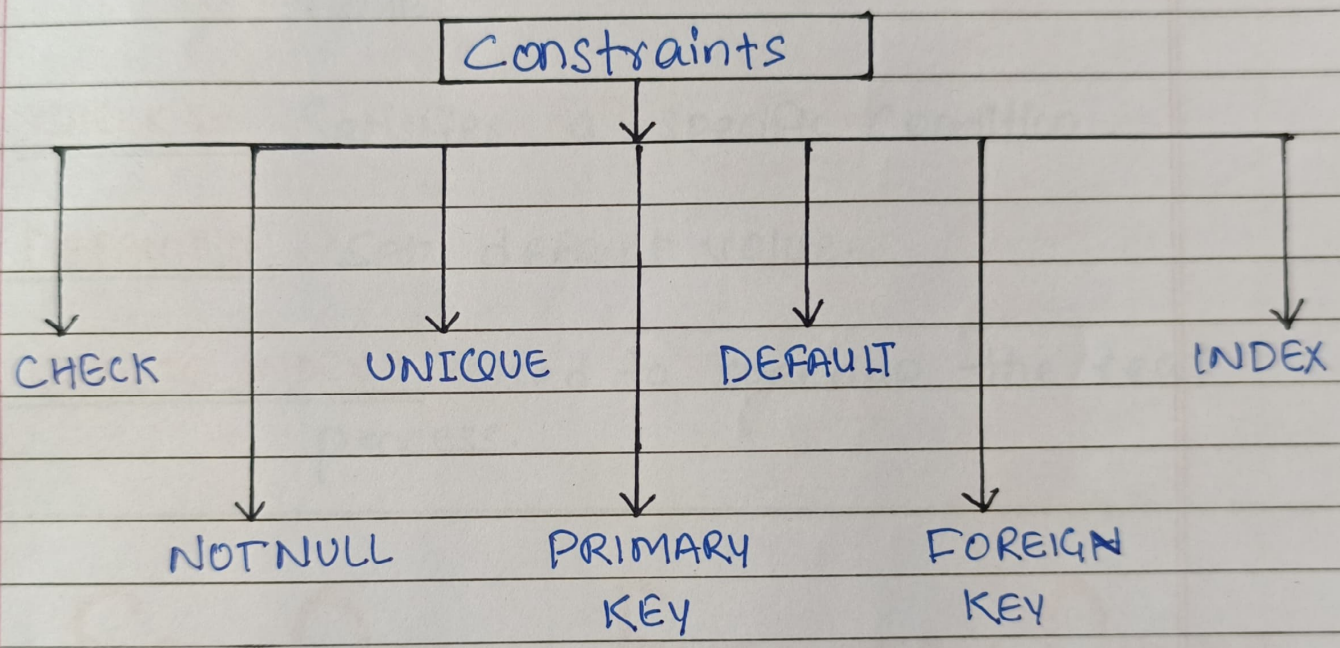
ORDER BY :- It is used to sort the data in ascending or descending Order.

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

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

SQL Constraints

ATUL KUMAR (LinkedIn/Twitter)
NOTES GALLERY/CODING BUGS (Telegram)

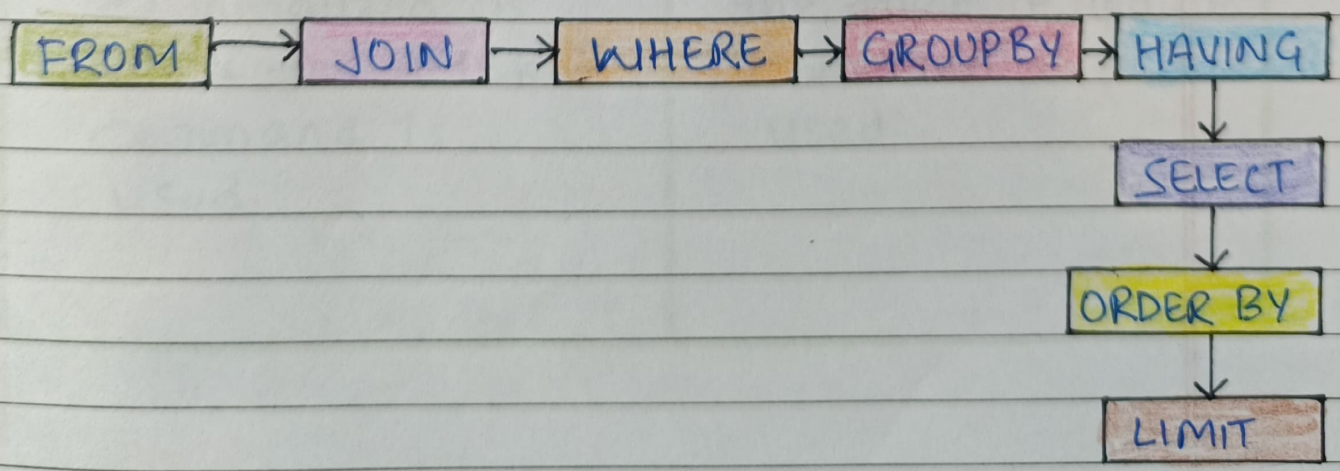


These Constraints also Known as Integrity Constraints.

- **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.
- **PRIMARY KEY :-** Used 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



DIFFERENCE BETWEEN ALTER AND UPDATE

ATUL KUMAR (LinkedIn/Twitter)
NOTES GALLERY/CODING BUGS (Twitter)

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

Command :-

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

Output :-

Password
123
xyz

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	Last Name
Amit	Sharma.

SQL AND :-

The AND Operator returns true if both condition are true, and false otherwise.

Syntax:-

```
WHERE Condition 1 AND Condition2;
```

ATUL KUMAR (LinkedIn/Twitter)
NOTES GALLERY / CODING BUGS (Telegram)

SQL OR :-

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

Syntax:-

```
WHERE Condition 1 OR Condition 2;
```

SQL NOT :-

It returns the opposite of a Condition.

Syntax:-

```
WHERE NOT Condition;
```

AND, OR, NOT Operator 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 Column 1, column 2, ...
FROM table_name
ORDER By Column 1 [ASC][DESC], Column 2
                    [ASC][DESC], ....
```

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

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

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
Jay	Kumar	49

INSERT INFO :-

It is used to insert data into a table.

Syntax :-

Insert into tablename (Column 1, Column 2, ...)
values (value 1, value 2, ..) ;

Note :- These 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);

Output:-

First Name	Last Name	Age.
Jai	Kumar	10
Jaya	Singh	15
Amit	Sharma	20
abc	xyz	25

SQL NULL Values:-

It is used represent missing or unknown data.

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

ATUL KUMAR (Linkedin / Twitter)
NOTES GALLERY / CODING BUGS (Telegram)

Insert NULL value:-

```
INSERT INTO tablename (Column 1, Column 2, ...)
VALUES (Value 1, Null, ..);
```

To check for Null Values:-

IS NULL :-

```
SELECT Column 1, Column 2, ...
From table_name
WHERE column 2 IS NULL;
```

IS NOT NULL :-

```
SELECT Column1, Column2,...
FROM table_name
WHERE column1 IS NOT NULL;
```

SQL UPDATE :-

It is used to modify existing data in table.

Syntax :-

```
UPDATE table_name
SET Column1 = value 1, Column2 = value 2,...
WHERE Some_column = Some_value;
```

~~Syntax~~ :- 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 ;
```

Output :-

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

ATUL KUMAR (LinkedIn / Twitter)
NOTES GALLERY / CODING BUGS (Telegram)

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 !

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 Column 1, Column 2, ...
FROM table_name
WHERE Column_name LIKE pattern;
```

SQL IN :-

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

Syntax :-

```
SELECT Column 1 , Column 2...
FROM table_name
WHERE Column_name IN (value 1, value 2);
```

SQL Between :-

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

Syntax :-

```
SELECT Column 1, Column 2, ...
FROM table_name
WHERE Column_name Between
      value 1 AND value 2;
```

SQL Alias :-

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 sets.

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

1
2
3

Union ↓

SELECT B

3
4
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.

ATUL KUMAR (LinkedIn / Twitter)
NOTES GALLERY / CODING BUGS (Telegram)

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.

Types of Error In SQL

- **Syntax Errors:-** These occur when SQL statements do not follow the correct syntax and structure of the language.

- **Semantic Error :-** These occur when the SQL statement is grammatically correct, But does not produce the desired result due to incorrect Logic.

- **Constraint Voilations :-** These occur when the SQL violates one or more constraints on the database.

- **Datatype Errors :-** These occurs 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 Error :-** These occur when a transaction fails due to problems with locking.

SQL JOINS

SQL JOINS :-

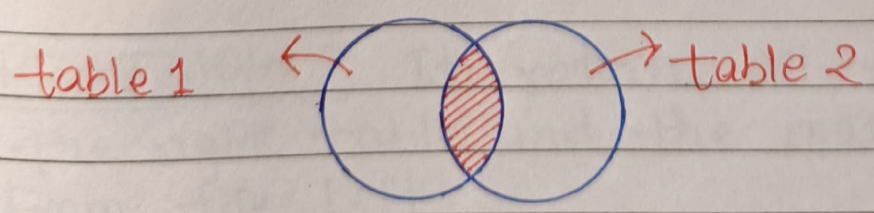
These statements allows us to access information from two or more tables at once. They also keep our database normalized.

Types of Joins :-

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

ATUL KUMAR (LinkedIn / Twitter)
NOTES GALLERY / CODING BUGS (Telegram

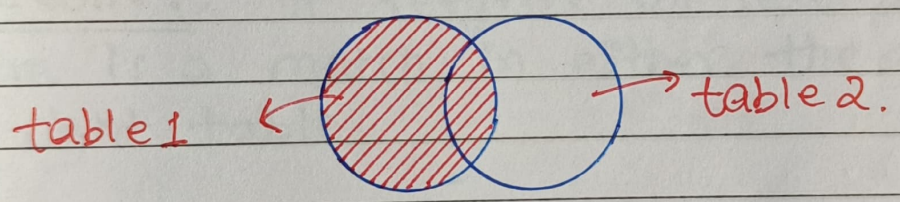
INNER JOIN :- It returns dataset that have matching values in both tables.



```
SELECT Column_name
FROM table 1
INNER JOIN table 2
ON table. Column_name = table 2. Column_name;
```

LEFT JOIN

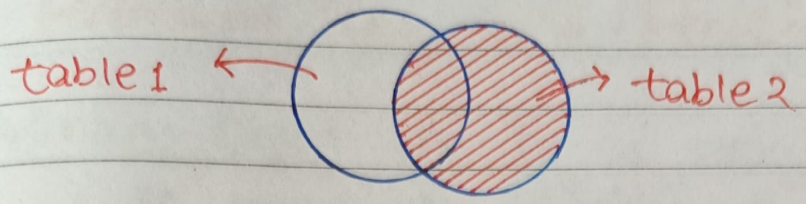
LEFT JOIN :- It returns all records from the left table and matched records from the right.




```
SELECT column_name
FROM table 1
LEFT JOIN table 2.
ON table1. Column_name = table 2. Column_name.
```

RIGHT JOIN

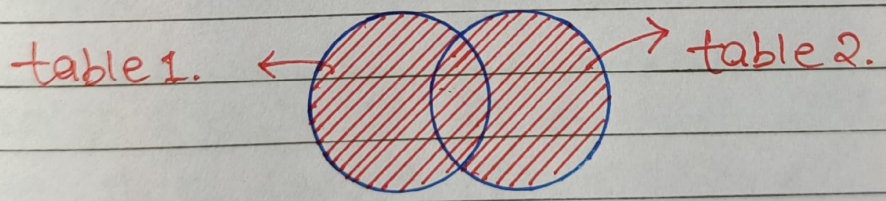
RIGHT JOIN :- It returns all records from the right table and the match records from the Left.




```
SELECT Column_name
FROM table 1
RIGHT JOIN table 2
ON table. Column_name = table 2. Column_name;
```

FULL JOIN

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



ATUL KUMAR (Linkedin/Twitter)
NOTES GALLERY/CODING BUGS(Telegram)

```
SELECT Column_name
FROM table 1
FULL OUTER JOIN table 2
ON table1. Column_name = table2. column_name
WHERE Condition
```

* 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 :-

```
Create table tablename (
    Column 1 datatype,
    Column 2 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 value.

Syntax :-

```
Create table tablename (
    column 1 datatype ,
    column 2 datatype ,
    - - - - - -
    Foreign Key [ Column 1, - - - ]
    References [ Primary Key table name ]
);
```

→ Let's Take Example :-

Roll No.	Name	Address
1	Atul	Delhi
2	Abhi	Mumbai
3	Zayn	Delhi

→ Primary Key.

Name and address are not unique.
Here Roll No is unique.
So we discussed before, Primary Key is unique and not null value.

Result table.

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

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

→ You want to check Roll No. 3 marks in Maths, Foreign Key come into picture student table + Result table.

We have to add Primary Key in Result table which act as Foreign Key.

ATUL KUMAR (Linkedin/Twitter)
NOTES GALLERY & CODING BUGS (Telegram)

Foreign Key

Maths	Science	English	Roll No.
10%	80%	70%	1
90%	60%	20%	2
70%	30%	21%	3

COMMIT AND ROLLBACK

commit and Rollback are transaction statements.

Commit :-

It is used to storing changes Permanantly Performed by a transaction.

Commit Syntax :-

```
COMMIT;
```

Rollback :-

It is used for reverting changes Performed by a transaction.

Rollback Syntax :-

ATUL KUMAR (LinkedIn / Twitter)
NOTES GALLERY / CODING BUGS (Telegram)

```
Rollback;
```