***---SQL(Structured Query Language)---***

--> **Two types-**

1. Sql (used to store relational data eg.Tables)

2. NoSql (Used to store non-relational data eg.Wikipedia)

--> **Datatypes in SQL-**

1. String

2. Numeric

3. Date and Time

1.String:-

1 CHAR(size)-> (can store 0-255 char)

2 VARCHAR(size)-> 0-65535 char

3 BINARY(size)->binary numbers

4 VARBINARY(size)

5 TINYTEXT -> 255 char only

6 TEXT(size)-> 65535 bytes-It Stores data in bytes

7 MEDIUMTEXT -> 16777215 chars

8 LONGTEXT -> 4294967295 chars

9 BLOB(size) -> 65535 bytes

10 TINYBLOB -> 255 bytes

11 MEDIUMBLOB -> 16777215 bytes

12 LONGBLOB -> 4294967295 bytes

13 ENUM(val1, val2, val3,.....) -> we can select from any values to enter in columns, ->

lists up to 65535 values

14 SET(val1, val2, val3,.....)-> lists up to 64 values.

2.Numeric:-

1 BIT(size) -> 0-64

2 TINYINT(size) -> -128 to 127

3 INT(size) -> -2147483648 to 2147483648

4 INTEGER(size)

5 SMALLINT(size) -> -32768 to 32768

6 MEDIUMINT(size) -> -8388608 to 8388608

7 BIGINT(size) -> -9223372036854775808 to 9223372036854775808

8 BOOL

9 BOOLEAN -> o and 1

10 FLOAT(p)

11 DOUBLE(size, d) -> 255.568

12 DECIMAL(size, d) -> size= 60 and d=30

13 DEC(size, d) ->

3.Date and Time:-

1 DATE -> "1000-01-01 to 9999-12-31"

2 DATETIME(fsp) -> format="YYYY-MM-DD hh:mm:ss"

3 TIMESTAMP(fsp) -> format="YYYY-MM-DD hh:mm:ss"

4 TIME(fsp) -> "hh:mm:ss"

5 YEAR -> "YYYY"->in four digits only

--> **Constrains in SQL-**

- Constrains are the restrictions to the column data that is to be entered.

1. NOT NULL -> (column data can’t be empty)

2. UNIQUE -> (only allows unique data in column)

3. DEFAULT -> (sets a default value to the column, if data is not entered by user

it sets to its default value)

4. CHECK -> (Use to check the conditions such as age (age>=18) that is to be entered in column)

5. FOREIGN KEY

6. PRIMARY KEY

--> **Operators in SQL -**

1. **AND** -> when both conditions need to be true

2. **OR** -> when either one condition needs to be true

3. **NOT** -> It gives all other values except the mention condition...

(NOT is use just after where keyword)

4. **IN** -> It gives multiple conditional data in single column

(syntax-> where <column name> IN(cond1,cond2,cond3,.....); )

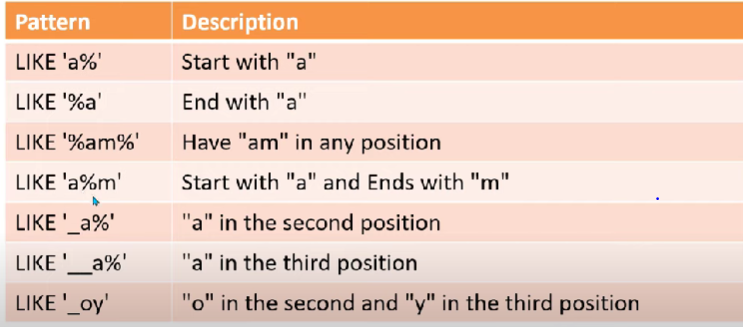
5. **BETWEEN** -> It gives data in the given range. (eg. age between 18 to 20)

(Syntax-> where <clm name> BETWEEN 18 AND 20 ;)

6.**LIKE** ->

wildcard characters:-

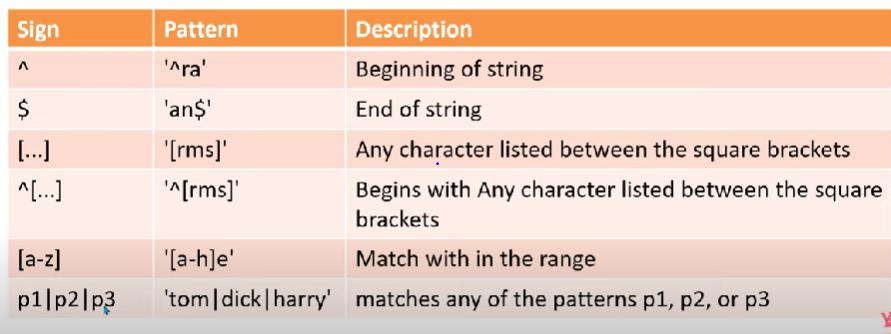
-->Uses of wildcard characters-



--> **Regular Expression command:-** (command - REGEXP)

**-** use to sort table data based on different patterns using different symbols.

Patterns are as follows-

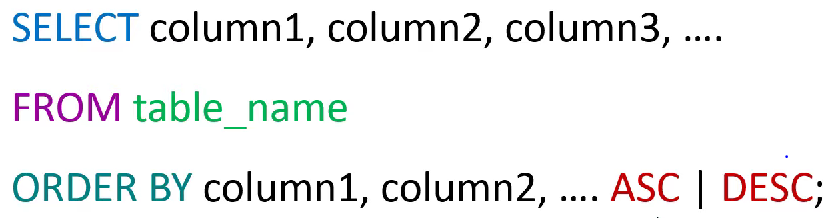


**--> ORDER BY and DISTINCT :-**

ORDER BY = > Use to sort the columns data either by ascending or descending order,

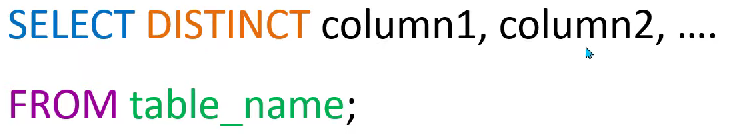
-By default it order in ascending order if we don’t mention any command using ‘order by’.

ORDER BY syntax-



DISTINCT = > Use to give unique elements in column.

DISTINCT syntax-

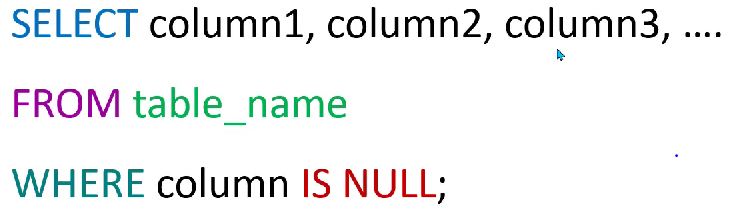


--> **IS NULL and IS NOT NULL :-**

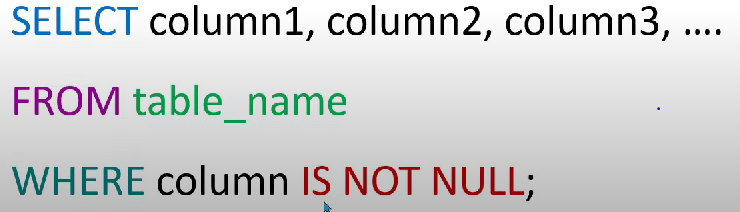
- Used to check if any value in column is null or not.

- and IS NOT NULL is use to know if there are no null elements in a column.

IS NULL syntax-



IS NOT NULL syntax-

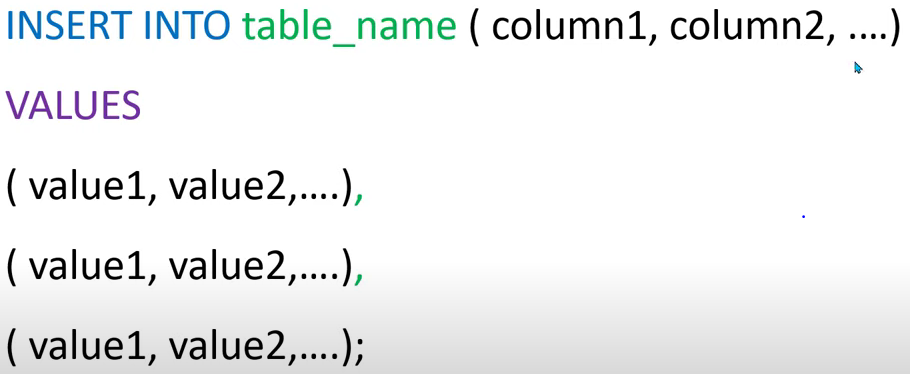


-->**INSERT Command :- (Insert data Into table-)**

-Use to insert data in a table.

- Before using Insert command we need to select database in which data is to be inserted, we can select database by using command (USE <database name> ;).

INSERT syntax-

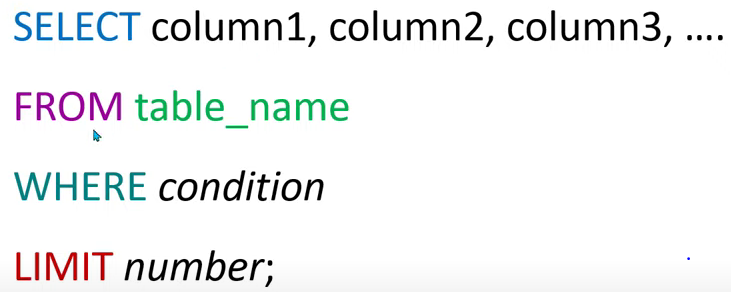


-->**LIMIT and OFFSET Command :-**

**Limit-**

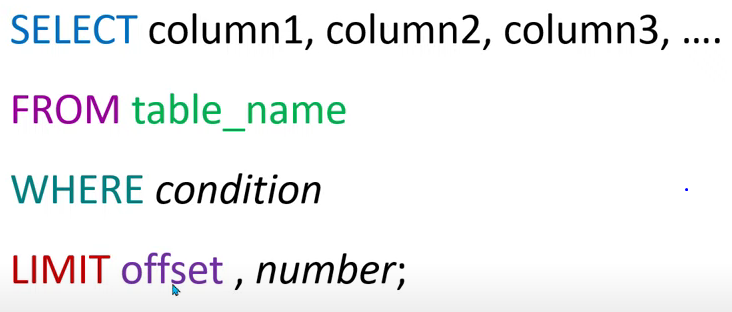
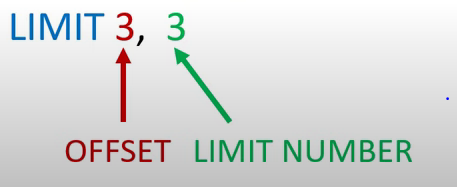
**-**Use to limit the number of table rows to be shown.

LIMIT syntax-



**Offset-** offset number is the number from where the limit is starts, example- if offset is 2 and limit is 4 then it will give the data after row 2 and gives further 4 rows.

OFFSET syntax-

-->**Aggregate functions:-**

- These are the functions which includes the operations on particular column such as

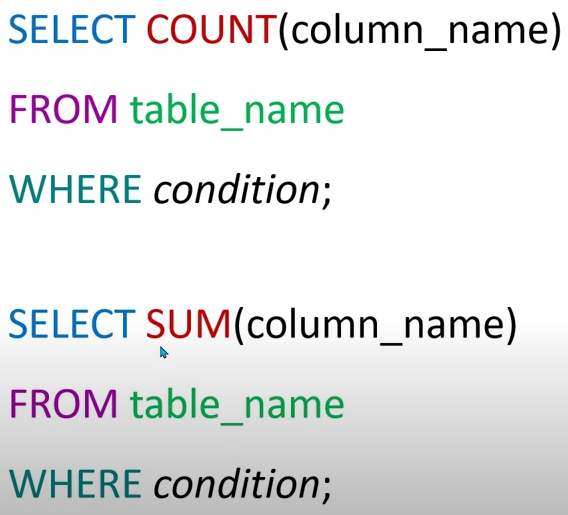
Addition of column values, maximum/minimum among the column values,

Average of column values, total count of elements in column etc.

Few of the aggregate functions are:-

1. Sum
2. Maximum
3. Minimum
4. Count
5. Average

Syntax:-



- Here in place of count or sum we can replace any other aggregate function with same syntax.

- Also we can use different commands in aggregate functions as DISTINCT and so on.

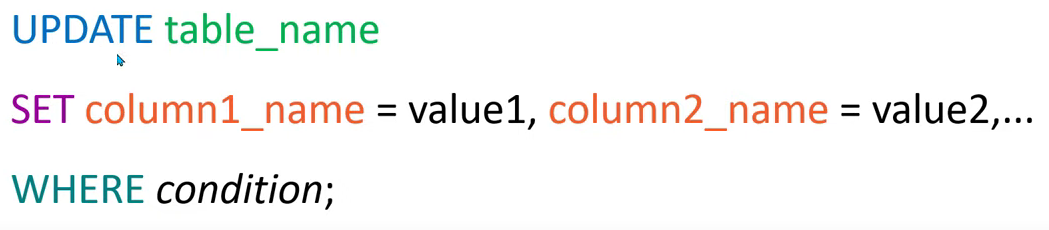


- And we can name the output column of any aggregate function using Alice (AS) as mentioned above.

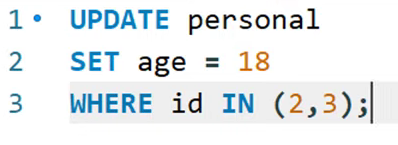
-->**UPDATE Command :- (Update data in table)**

- It is use to update/correct/add data in previously created table.

UPDATE syntax-



-We can change values of multiple rows at a time using “IN” command as follows,



-This will update age in both rows with id no as 2 and 3.

\*-We should use where clause on mandate basis, as if we don’t use where clause changes will be applicable to whole column.

-->**COMMIT and ROLLBACK Command:- (works as undo)**

COMMIT :- when we use commit command then it will save all the changes previously done

Using the commands insert/delete/update.

-In MySQL commit command is executed automatically, we need to uncheck it to use

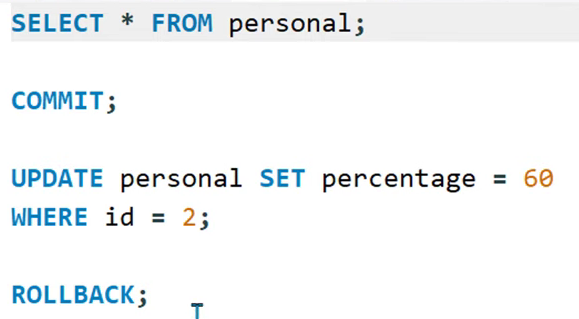
Rollback command.

ROLLBACK :- Rollback is works as undo command, it only works if the previous commands are

Not committed.

\* -We can use rollback command to only three commands as –

INSERT/DELETE/UPDATE.

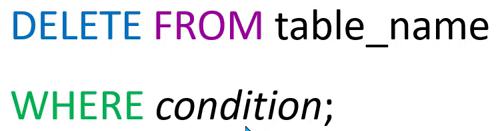


-->**DELETE Command:-**

- Delete is used to delete the data from the table, we can either delete single row or we

Can delete whole data present in table using this command.

DELETE syntax –



-If we need to delete a single row from table then we need to use where clause

Mandatorily, If we don’t pit will delete all the data from table.

-->**Foreign Key and Primary Key:-**

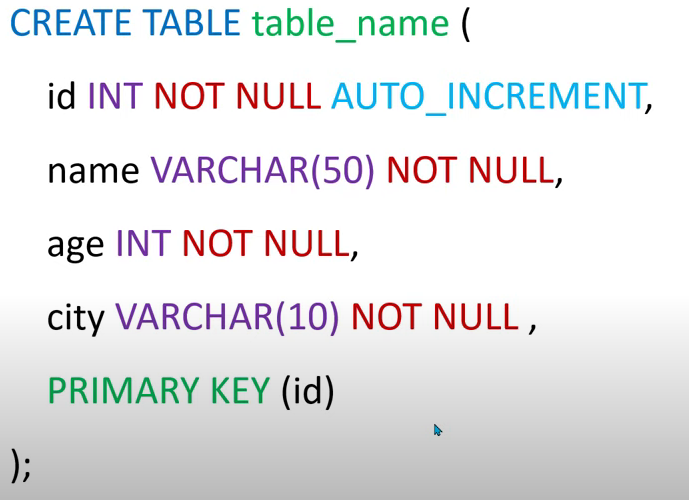
**-**The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

-->PRIMARY KEY:- - It always has unique data.

-It can’t have null value.

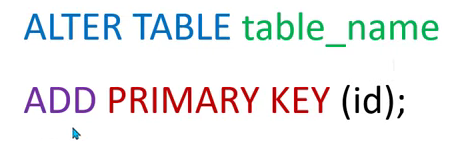
-A table can contain only one primary key constrain.

Primary Key Syntax-



-Here “id” is column name.

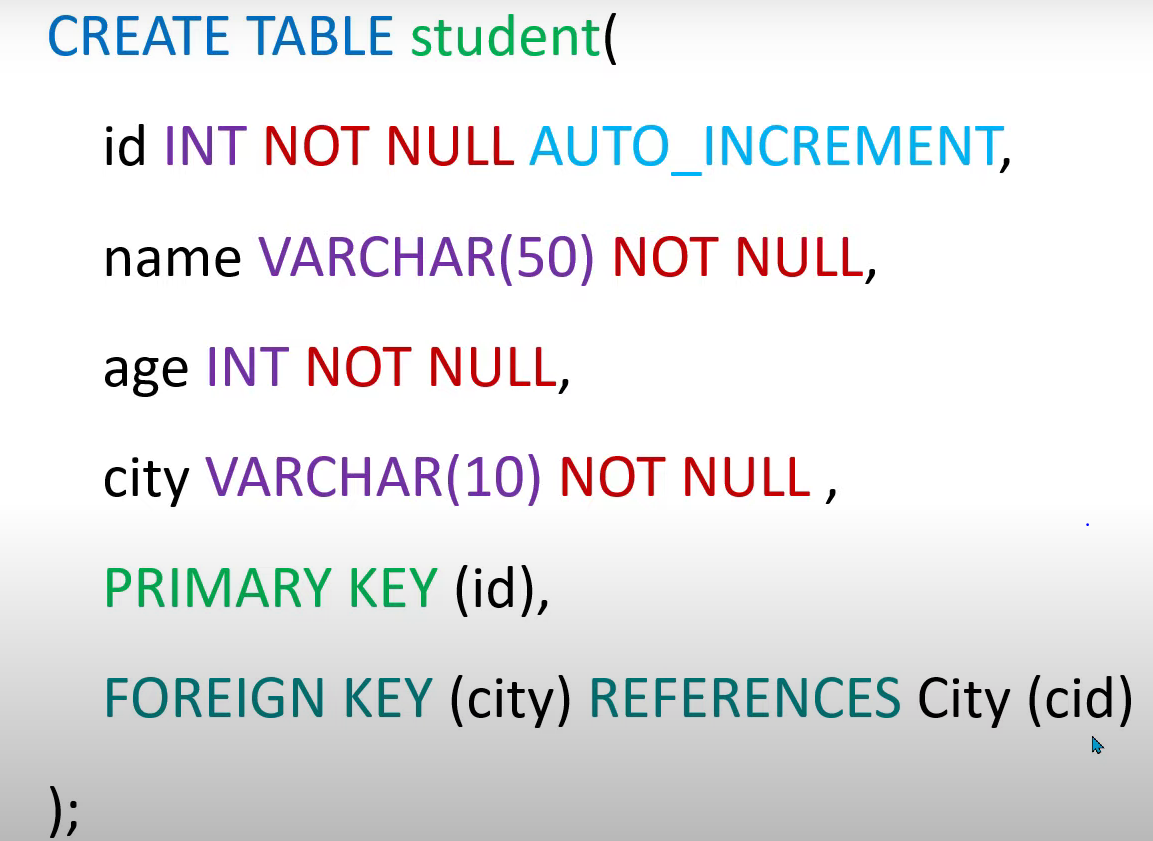
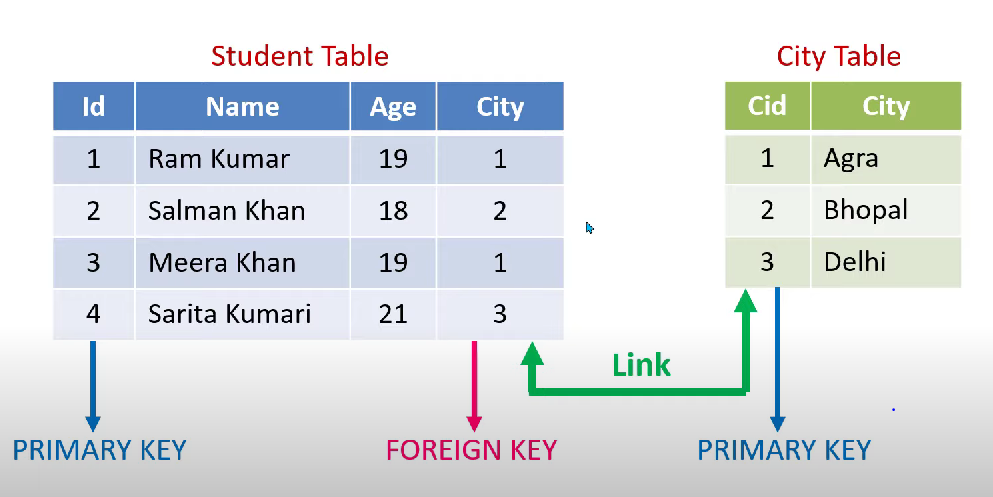
-Or we can declare a primary key to already created table using ALTER command as follws:-



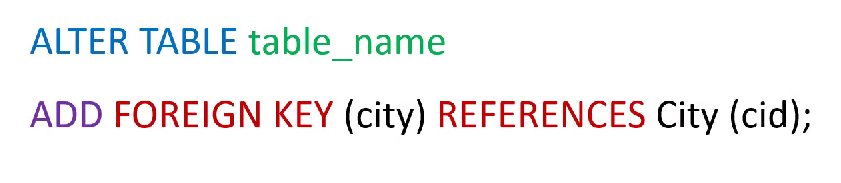
-->FOREIGN KEY:- Foreign key can have data which is primary key in any other table.

- Using foreign key we can link two tables.

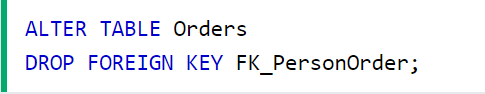
Foreign Key syntax:-



-We can set the foreign key to existing table as.



-We can drop the foreign key and primary key constrain using following syntax-



-->**JOINS:-**

**-**Joins are used to relate similar data in two tables.

-we can use joins to combine data in two or more tables.

-Joins need to have primary key and foreign key in tables except cross join.

-There are four types of joins:-

1. Inner join.

2. Left join.

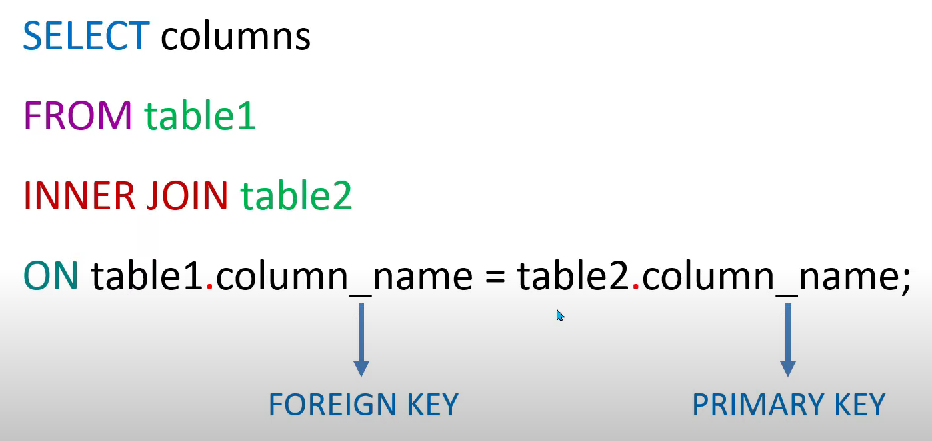
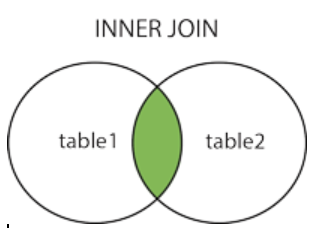
3. Right Join.

4. Cross Join.

1. INNER JOIN:-

-Inner join gives the unique data or similar data in two tables.

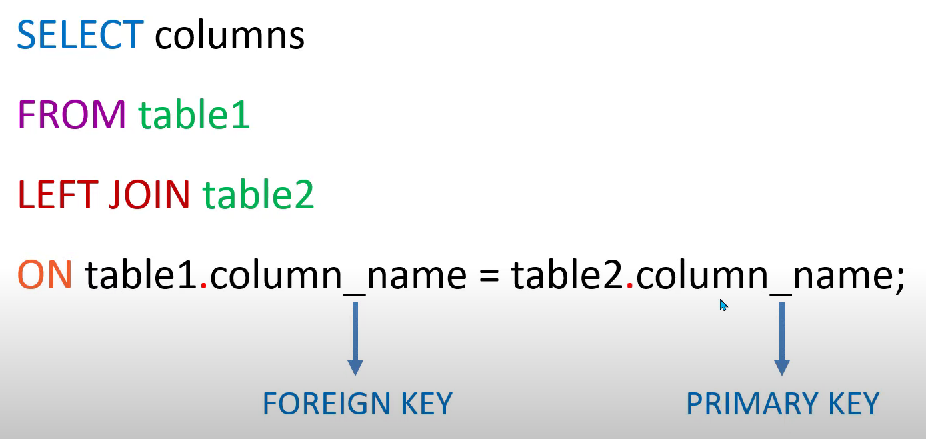
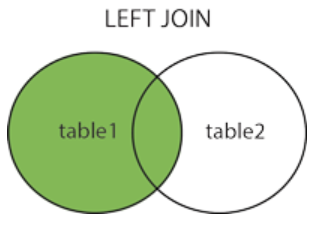
Inner join syntax:-

1. LEFT JOIN:-

-Left join returns the all data from left table and also gives common data in both the tables.

Left join syntax:-

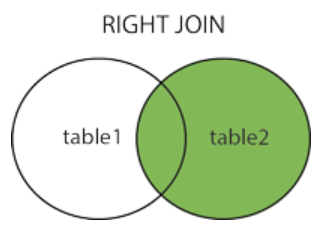
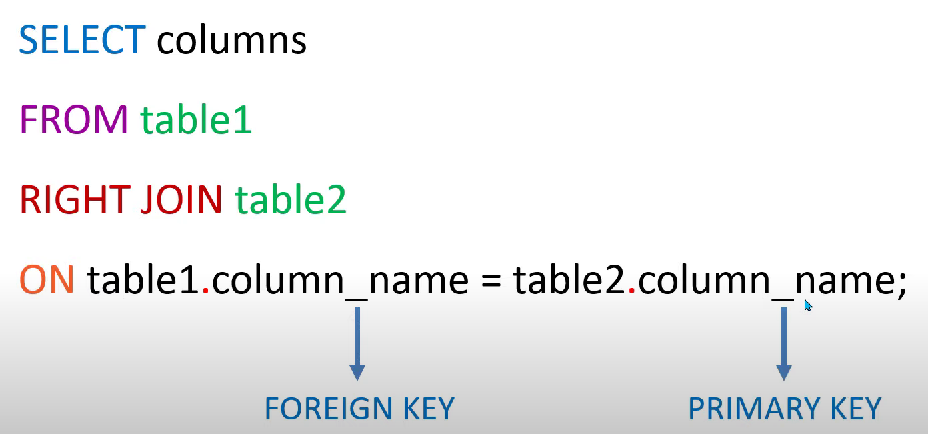
 

-Here table1 is left table and table2 is right table.

1. RIGHT JOIN:-

-Similar to left join right join returns all the data from right table and matched data from left table.

Right join syntax:-



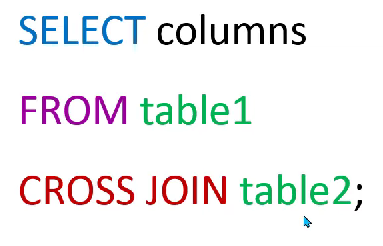
1. CROSS JOIN:-

-Cross joins do not need to have primary key and foreign keys in tables.

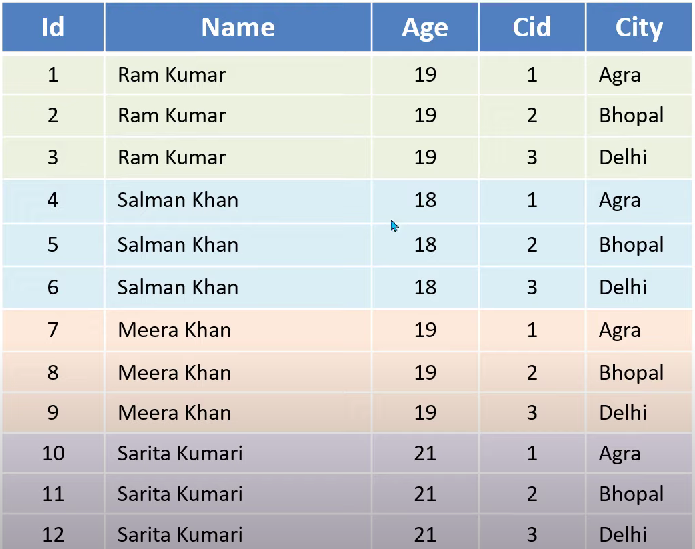
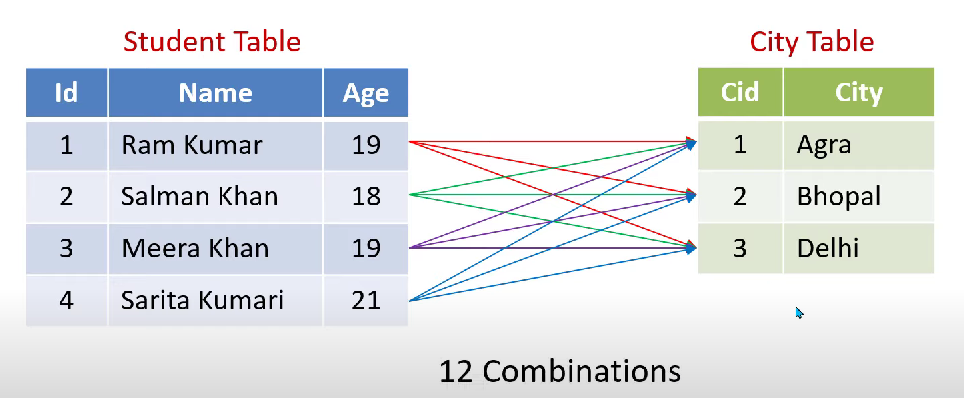
-Cross join returns all the data from both table1 and table2.

-It used less comparatively to the remaining joins.

Cross join syntax:-



-Result of cross join:-

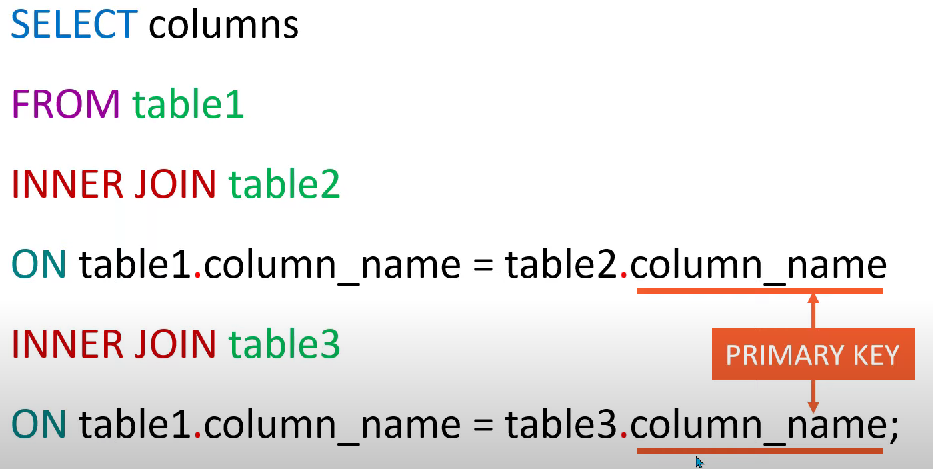


-->**Joining Multiple Tables:-**

**-**We can join multiple tables using joins such as inner,left,right joins.

-To join multiple tables we need to use multiple join commands.

Syntax:-



-Similar to inner joins we can use left join as well as right join using same syntax to join multiple tables.

-Using above syntax we can join many more tables.

-->**Group By And Having Clause:-**

GROUP BY:-

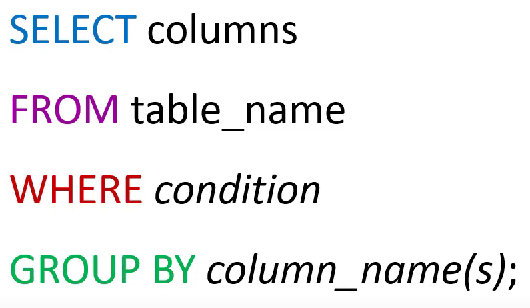
-> The group by statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

-> The group by clause is used in conjunction with the SELECT Statement and

Aggregate function to group rows together by common column values.

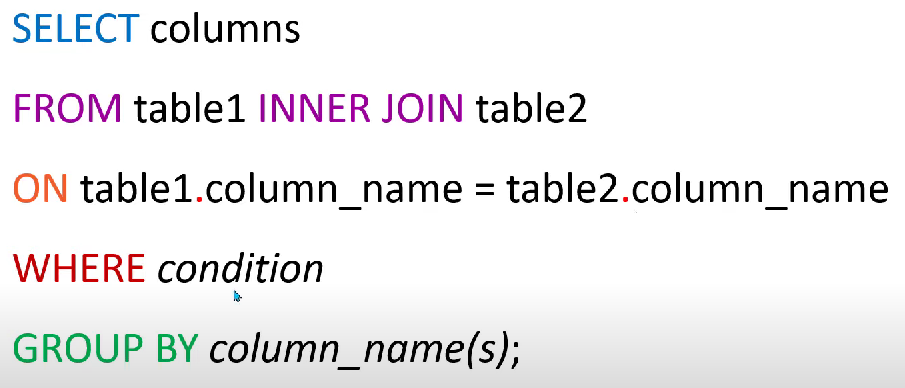
-> The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

-Group by syntax:-



-Also we can use group by clause by joining the two or more tables to get the relatable data from both the tables.

Synatx for multiple tables:-



-We need to use where condition before group by clause and order by after group by clause, similarly we can use having clause after group by clause only.

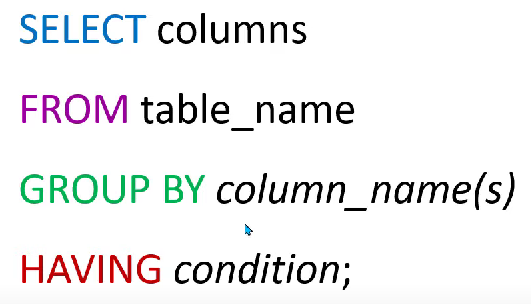
HAVING :-

-We can’t use where keyword with aggregate function that’s why there is need of having clause.

-Having clause is use to add conditions on result of group by clause.

-Having clause condition is applicable only on group by clause result.

Having clause syntax:-



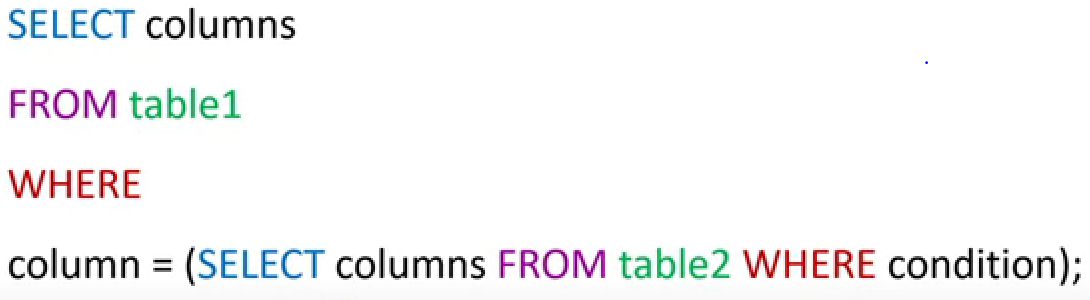
**-->SubQuery Or Nested Query with EXISTS and NOT EXISTS:-**

**-**When we use one Query inside another query It is said to be Nested Or SubQuery.

-Here there are two queries- parent query and subquery, the main starting query is said

to be parent query and another query inside the parent query is said to be SubQuery.

Syntax:-



-Here in subquery we can use any other queries like INSERT/UPDATE/DELETE in place of SELECT.

EXISTS:-

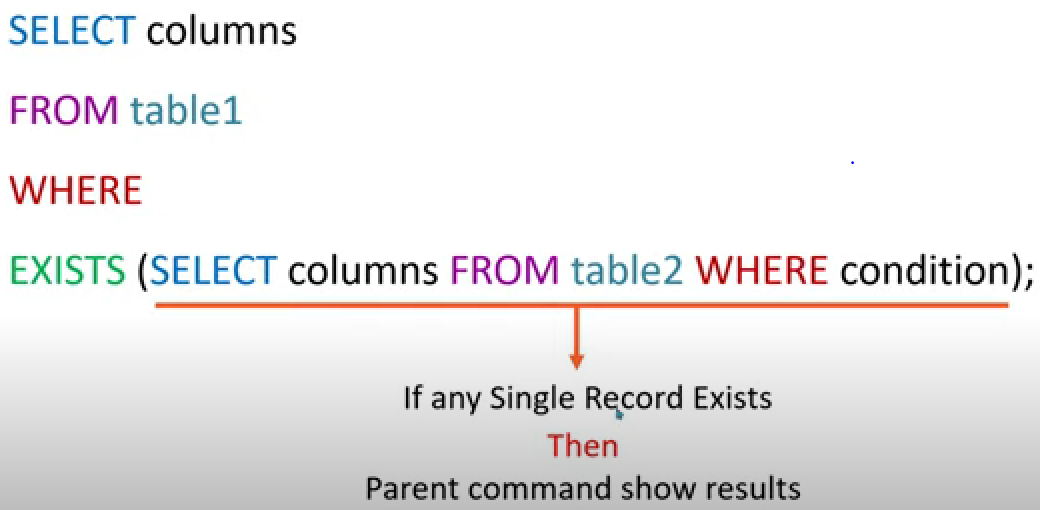
-We can use EXISTS command after Where clause.

-Here if EXISTS command or child command have any single result then parent

Command will return the result, Or if child command have no result then parent

Command will also give no results.

EXISTS Syntax:-

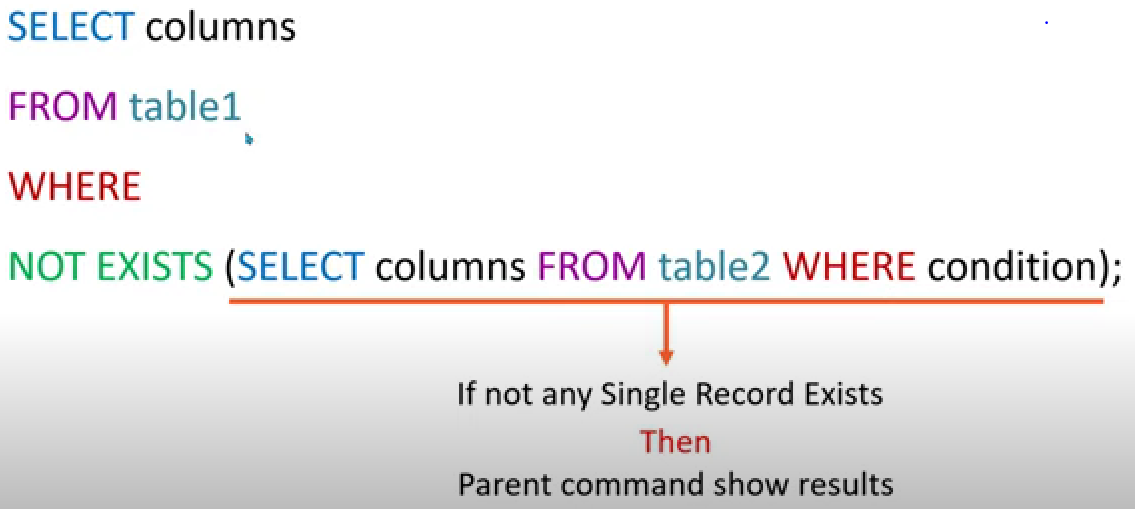


NOT EXISTS:-

-In NOT EXISTS if child command does not have any records to display then only parent

Parent command will give results.

NOT EXISTS Syntax:-



**-->UNION and UNION ALL:-**

--Union query will give us combine records of both tables without any duplicate records,

this means it will remove the duplicate records exists in two tables if any exists.

--and UNION ALL will give us all records of both tables with all records including

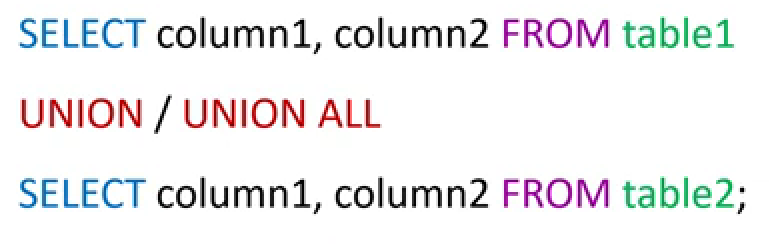
duplicate records if any exists.

--To use the UNION both the tables need to have same number of columns, and also

the Same respective columns datatype.

--And columns in each SELECT statement must also be in the same order.

UNION and UNION ALL Syntax:-



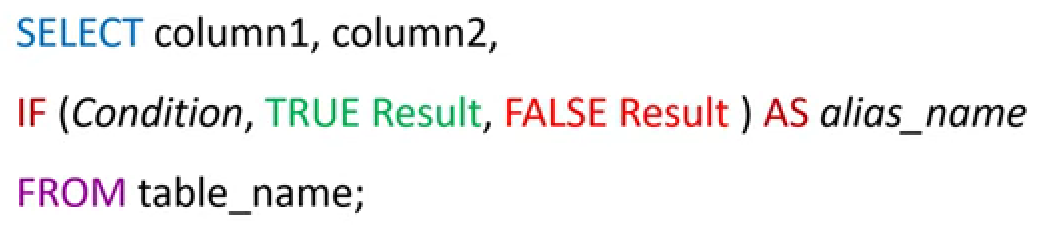
**-->IF and CASE Statements:-**

--When we need to get a condition based result from table then we can use IF clause.

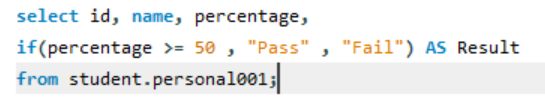
--Example\_ If we want to know the passed and failed students in a class based on their

Marks then we can use IF clause, It will give results as pass or fail.

IF clause Syntax:-



Example:-



CASE Clause:-

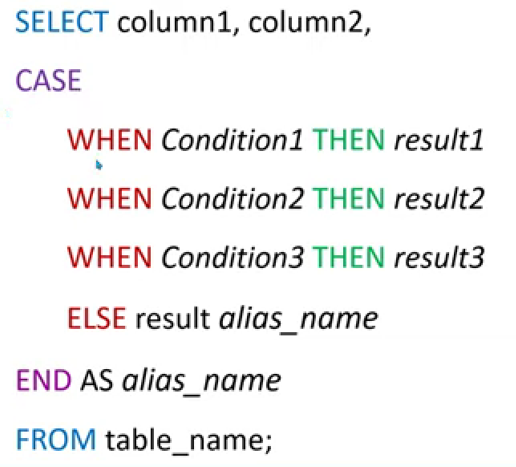
--CASE clause is use to get the multiple results based on multiple conditions.

--Example\_ If we want to get the results as merit based, 1st class, 2nd class, etc with

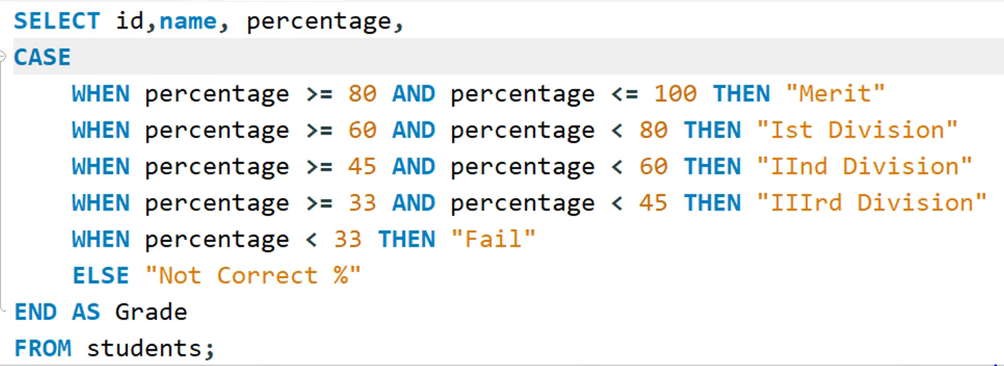
Different conditions then we can use CASE clause.

--We can use CASE statement with UPDATE/DELETE/INSERT commands also.

Case Clause Syntax:-

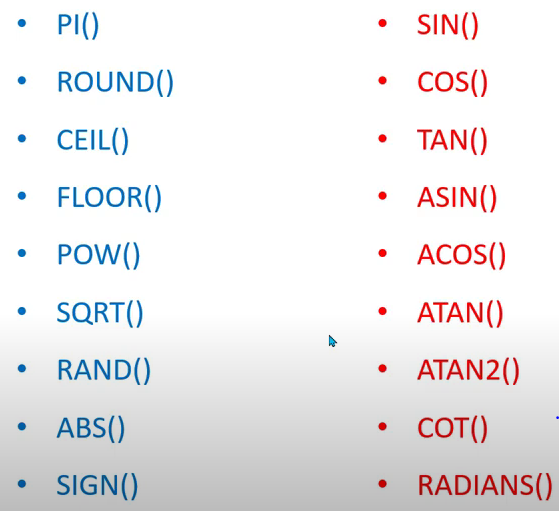


Example:-



**-->Arithmetic Functions:-**

**--**These are the functions used to perform mathematical operations.



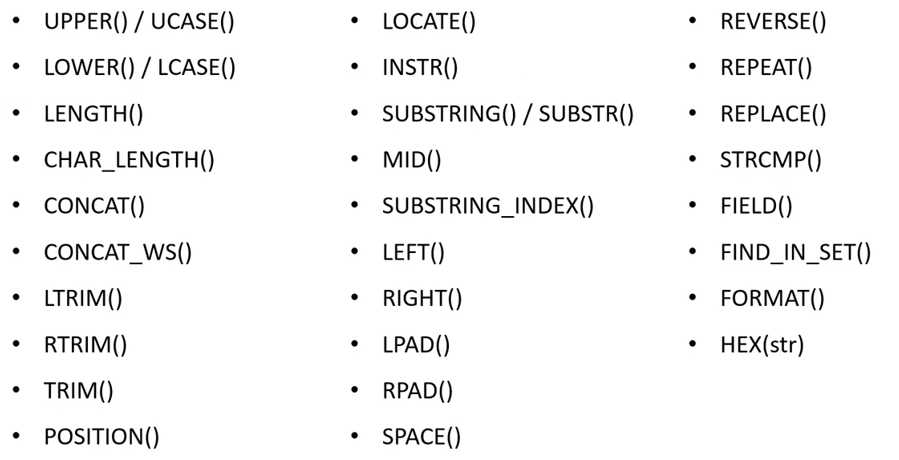
--Below are the few important functions which are mostly used.

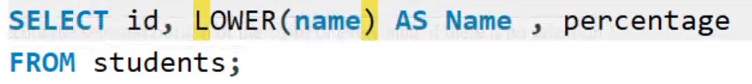
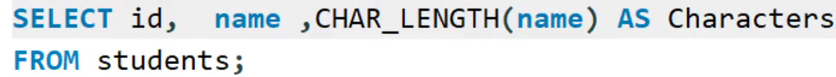
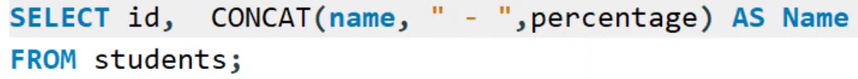
1. **PI() :-** It is pi function used in general mathematics.
2. **ROUND() :-**  Round is use to round up the fractional values.
3. **CEIL() :-** It converts fractional number into its next whole number. Example:- if number is 4.6 or 4.1 it will give 5.
4. **FLOOR() :-** Floor converts the fractional value into its precede whole number. Example :- if the number is 2.2 or 2.9 it will give 2.
5. **ABS() :-**  Absolute function gives us the absolute whole number provided the negative values. It means it converts negative values into positive.
6. **SQRT() :-** Square root is function which gives the square root of given number.
7. **POW(base, exp) :-** Pow gives the power value of base number. Example:- if we write as POW(2,3) then it means 2 to the power 3, and it will give result as 8.
8. **SIGN() :-** In sign function if we provide number greater than zero it gives result 1, if we give number zero it gives result 0, and if we give number less than zero it gives result as -1.
9. **RAND() :-** Random function generates the random value between 0 to 1. And we limit the random value generated within the required range.



This will give us random value between 1 to 5.

**-->String Functions:-**



1. UPPER()/UCASE() :- Converts the given column data into Uppercase. 
2. LOWER()/LCASE() :- Converts the given column data into lowercase.
3. CHAR\_LENGTH() :- It gives the total number of characters row wise, It counts the charecters including spaces.
4. LENGTH() :- It gives the length of string in bytes.
5. CONCAT() :- It is used to concat two or more columns. 
6. CONCAT\_WS() :- It works same as concat but it adds the first string in between remaining words,  
7. LTRIM() :- It removes all the spaces from left side of string. 
8. RTRIM() :- It removes spaces from right side of string.
9. TRIM() :- It removes spaces from both sides of string.
10. POSITION() :- It gives the position of perticular word or character in the string. 
11. INSTR():- It works same as position with syntax difference.  Here it will search ‘a’ inside the string and gives output as 2.
12. LOCATE() :-It also works same as the POSITION or INSTR but we can give the starting point to search from. Here it will search ‘a’ from third position.