**SQL SYNTAX**

1. **DESCRIBE-**

**DESCRIBE** Table\_name;

1. **CREATE TABLE**

**CREATE TABLE** Table\_name(column1 data\_type, column2 data\_type….);

EX-

CREATE TABLE Sonali(

Std\_id INT PRIMARY KEY AUTO\_INCREMENT,

Age Int,

Name Varchar(40) NOT NULL,

Mail\_id Varchar(40));

1. **SELECT STATEMENT**-

Using for showing the table

**SELECT \* FROM** Table\_name; - for all the data in the table.

**SELECT** COLUMN1, COLUMN2…

**FROM** TABLE\_NAME;

**SELECT** COLUMN1, COLUMN2…

**FROM** TABLE\_NAME **LIMIT** LIMIT\_RANGE;

**SELECT** column1,column2+number as updated\_value

**FROM** Table\_name;

1. **DISTINCT VALUES –**

**SELECT** **DISTINCT** COLUMN1, COLUMN2…

**FROM** TABLE\_NAME;

1. **COUNT-**

Gives the vales of number of rows

**SELECT** **COUNT**(DISTINCT COLUMN1, COLUMN2…)

**FROM** TABLE\_NAME;

**SELECT** **COUNT**(DISTINCT COLUMN1, COLUMN2…)

**AS** counted\_name

**FROM** TABLE\_NAME;

1. **WHERE CLAUSE-**

**SELECT** column1, column2…

**FROM** Table\_name

**WHERE** condition;

* Note- always put coated that may be single or double for writing the conditions (string only not for numeric value).

1. **AND/OR/NOT-**

**SELECT** column1, column2

**FROM** Table\_name

**Where** condition1 **AND** condition2;

For using multiple “and or not clauses” need to use condition like –WHERE first\_name= ‘Sonali’ AND fast\_name=’Sonam’;

**Not –**

**SELECT** column1, column2…

**FROM** Sonali

**WHERE NOT** column1= value

For multiple NOT clause- WHERE NOT 1stName=’Sonali’ AND NOT 1stname=’Sonam’;

**LIKE-**

**SELECT** column1, column2…

**FROM** sonali

**WHERE** column1 **LIKE ‘R%’;**

In this case it returns value which are starts with ‘r’ at first in the particular column

**If ‘%r’ –** then returns the value which is ends with r in that particular **column.**

**If ‘%r%’ –** returns the word start and end with r.

**If ‘\_y’ –** 1st character does not matter but last character should be y.

1. **ORDER BY- ASC, DESC**

**SELECT** \*

**FROM** TABLE\_NAME

**ORDER BY** column1,column2 **DESC;**

In case there is only use order by clause then it sort in ascending order.

1. **INSERT INTO VALUES-**

**INSERT INTO** Table\_name(column1, column2….)

**Values** (‘value1’, ‘value2’..);

In case of id (which is primary key), if I don’t give any value and didn’t mention the id column then it automatically increment the column.

We can even enter the data in specific column.

1. **IS NULL-** This syntax is used to see the null values in the table

**SELECT** column\_names

**FROM** table\_name

**WHERE** column\_name **IS NULL;**

**IS NOT NULL-**

**SELECT** column\_names

**FROM** table\_name

**WHERE** column\_name **IS NOT NULL**;

1. **UPDATE SET VALUES Syntax**

**UPDATE** table\_name

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

**WHERE** condition;

Always put the condition when use this update table syntax or else every value inside the table will get modify.

1. **DELETE**

**DELETE FROM** table\_name **WHERE** condition;

Here also need to specify the where and condition, or else every value will be deleted from the table.

1. **IN operator –**

In case of using multiple ‘OR’ operator we can use ‘IN’ operator.

**SELECT \* FROM** Table\_name

**WHERE** columnName **IN (**values**);**

1. **CONCAT – Adding two column values into one column.**

**SELECT** column1,column2.. , **CONCAT(column1, ’ ’, column2)**

**AS** new\_name

**FROM** table\_name**;**

1. **UPPER/ LOWER-**

**SELECT** column1,column2, **UPPER(column1) AS**

Upper\_name

**FROM** Table\_name**;**

1. **LENGTH-**

**SELECT** column1,column2**, LENGTH(**column1**) AS**

Upper\_name

**FROM** Table\_name**;**

1. **SUBSTR/INSTR-**

**SELECT** Column1,column2, **SUBSTR**(Column1, starting point number, ending point number)

**AS** edited\_text **FROM** Table\_name**;**

Using SUBSTR we can give the string position to get the value at that position. It can start from one number and with different number.

Give the exact number till that you want to print.

And using INSTR which is totally opposite of the SBSTR, means- we can give the string and can know the position of the string. At a time we can give only one string to know the position of that string.

1. **LPAD/RPAD-**

**SELECT column1,column2, LPAD(column1, number,’special character’)**

**FROM Table\_name**

**WHERE Condition;**

**Padding is use to add the special character into the column.**

* Number is use to give the number of special character.
* If you give a particular number then in edited form it will return if the existing column has less number of character. Means if you give the number as 5 and in the column if one data is with already having 6charcter then it will print only first 5 character as a result. And if the existing data has less than 5 then it will return remaining number as Special character as chosen.
* If LPAD is chosen then special character get added to left side & if right then in right.

1. **TRIM-**

This function only TRIM the 1st & last character from the string.

TRIM(‘the 1st /last character you want to trim/delete’ FROM Column\_name)

FROM table\_name)

1. **REPLACE-**

**SELECT column1,column2, REPLACE(column1, character/value need to be replace, replaced value)**

**FROM table\_name;**

Here 1st condition is put the value which you want to change in the table. And the 2nd value is the replaced value which you want to give after replacing a value.

1. **CURRENT DATE-**

**SELECT CURRENT\_DATE();**

1. **Date Format-**

**SELECT DATE\_FORMAT(now(), ‘%d’ ‘%m’ ‘%y’) AS Today;**

1. **CURRENT TIME-**

**SELECT now();**

1. **CURRENT Year-**

**SELECT year(now());**

1. **ROUND-**

Provide the round figure. Increment the number if the value after the point is greater than 5 or equal to 5.

SELECT salary, ROUND(salary) AS rounded\_salary;

EX- ROUND(35.62 , 2)

O/P- 35.62

Round(35.62 , 1)

O/P- 36

1. **TRUNCATE-**

Only delete/truncate the values present after the point, we gave as input

TRUNCATE(value, place till when u want to truncate)

Ex- TRUNCATE(35.62 , 2)

O/P- 35.62

TRUNCATE(35.62 , 1)

O/P- 35.6

1. **MOD-** Provides the remainder

Mod(number1, number 2)

Number1 = dividend, Number 2= divisor

1. **COALESCE-**

SELECT customer\_id, first\_name,phone,

COALESCE (phone,city,"city not provided") AS Phone\_number

FROM customers;

If phone number not provided then show the city data and also city also missed then show as city not provided.

1. **Adding day/year – if the number is negative then it will show the previous value.**

SELECT DATE\_ADD(NOW(), INTERVAL 1 DAY/YEAR)AS Tomorrow/NEXT\_YEAR;

1. **Date\_Sub** – Here if you provide positive number then it gives output as previous and if u provide negative then it will add. (cause you are subtracting, means subs means (-) (-3) it will plus and gives the result accordingly. Syntax is ame as the DAE\_ADD
2. **DATEDIFF** – Gives the difference between two days.

SELECT DATEDIFF(‘2012/05/12’,’2012/05/1’’) AS Days\_difference;

1. **TIME DIFFERENCE-**

SELECT TIME\_TO\_SEC(’09:45’) – TIME\_TO\_SEC(’09:05’) AS Time\_difference;

1. **For calculating age-**

(YEAR(NOW) – YEAR(birth\_date))

1. **For Boolean values-**

SELECT TRUE;

It will return 1 if true or in case of false it returns 0;