--Functions

--1.min()

--2.max()

--3.count()

--4.TOP

--5.sum()

--6.avg()

--7.Distinct()

--1.MIN()

--This function will return the minimum value from a selected column

select min(salary) as minsal from employee --numbers === 0 to 9

select min(FirstName) from employee -- text value === A to Z

--2.MAX()

--This function will return maximum value from selected column

select max(salary) from employee)

--3.Count()

--This function is used to count the number of records from table or column.

--Count function always accepts one argument.

--It won’t count NULL values from the table or column.

select count(\*) as EmpCount from employee

select \* from employee

select count(loc) from employee

--Q In count function NULL value can be considered?

--NO, Null value is not considered in count function.

--4.TOP()

--This function is used to display the top records from table as per specified count.

--This function is very useful when we have large amount of data in table.

select Top 3 \* from employee -- it will display the top 3 records from table.

--5.Sum

--this function add all records from a column.

--it will return the total sum value in numeric expression.

--It will ignore NULL values from column.

select \* from employee

select sum(salary) as totalsary from employee

--6.avg()

--This function is used to find the avg of the column.

--It will ignore the NULL values.

select sum(salary) as totalsary from employee

select count(salary) from employee

select avg(salary) as AVGSAL from employee

--NOTE: In count,sum, and Avg function NULL values are ignored.

--7.Distinct

--This function is used to find the unique records from column.

select \* from employee

select distinct(Dept) from employee

select count(distinct(Dept)) from employee

select count(\*) from employee where Dept ='Account'

select distinct(salary) from employee

--Constraints

--Constraints are used to maintain the accuracy and integrity of the data.

--1.Primary Key

--2.Foreign Key

--3.NOT NULL key

--4.Unique Key

--5.Check Key

--6.Default key

--1.Primary Key --PK

--NOT NULL + UNIQUE

--It will always identify unique record into column of the table.

--PK is used in general with numeric values.

Create table student(S\_ID int primary key,

STUDENT\_NAME varchar(20),

LOC varchar(20))

insert into student values (1,'praveen','pune')

insert into student values (2,'Rohan','mumbai')

insert into student values (3,'Rohan','mumbai')

insert into student values (NULL,'veen','pune')

select \* from student

--2.NOT NULL

--NOT NULL constraint restricts you to insert NULL values into a column.

--If you define NOT NULL constraint on column then you can’t insert the NULL values in it.

--It will allow duplicates.

create table NOTNULL (NID int , FirstName varchar(20) NOT NULL, AGE int NOT NULL)

insert into NOTNULL values (1,'Amrita',27)

insert into NOTNULL values (2,'Amrita',27)

insert into NOTNULL values (3,NULL,27)

select \* from NOTNULL

--3.Unique

--It ensures that all the values in a column should be unique or different value.

--It will accept one NULL value into the column.

create table UNIQUE\_TEST (U\_ID int Unique , FirstName varchar(20) NOT NULL unique, AGE int NOT NULL)

insert into UNIQUE\_TEST values (1,'Amrita',27)

insert into UNIQUE\_TEST values (2,'Sangita',27)

insert into UNIQUE\_TEST values (NULL,'Arpita',23)

insert into UNIQUE\_TEST values (NULL,'mehir',23)

select \* from UNIQUE\_TEST

--4.Check key

--It ensures that all values in a column satisfies a specific condition.

--Check constraints is used to restrict the value of a column.

--It is just like condition checking before inserting the data into column.

Create table CHECK\_KEY(

C\_ID int primary key ,

C\_Name varchar(10) NOT NULL UNIQUE,

C\_AGE int check(C\_AGE >18))

insert into CHECK\_KEY values(1,'Sumit',19)

insert into CHECK\_KEY values(2,'Ronit',17)