

AIM:

To write a SQL Queries using windows function.

ALGORITHM:

1. Create a tables Employees and Students then, insert the values.
2. Use count() command to find out the number of times column values occurs in selected column.
3. Use avg(),min(),max() commands for find out the average value,minimum value,maximum value for selected column.
4. Use where command to find out the values for specified column.
5. Use sum() command for find out the total for the selected column.
6. Use rank(),dense rank,row number() commands to find out the rank for the list of students in Students table.
7. Using with cte as() command we find out the maximum,average,minimum values list in all the rows in the table.

CODING:

```
create table employees1(id int,name varchar(30),salary int,age int,location
varchar(30),mobile int);

insert into employees1 values(101,'Suka',50000,22,'Palayamkottai',8667897130);
insert into employees1 values(102,'Saran',40000,25,'Tuticorin',9807654321);
insert into employees1 values(103,'Priya',30000,29,'Tirunelveli',8765432190);
insert into employees1 values(104,'Hari',20000,24,'Chennai',9678543210);
insert into employees1 values(105,'Pavi',36000,23,'Tenkasi',9390876543);
insert into employees1 values(106,'Ram',55000,25,'Kanyakumari',9345678902);

select * from employees1;

select count(age) from employees1;

select AVG(salary) from employees1;

select SUM(salary) from employees1;

select salary from employees1 where id=104;

select min(salary) from employees1;

select max(salary) from employees1;


create table students11(id int,name varchar(20),subject_name
varchar(20),Marks_scored int,Total_marks int);

insert into students11 values(1,'Joseph','DIP',99,100);
insert into students11 values(2,'Ravi','IOT',90,100);
insert into students11 values(3,'Jack','DS',87,100);
insert into students11 values(4,'Steve','SC',85,100);
```

```
insert into students11 values(5,'Karthi','Python',95,100);
```

```
insert into students11 values(6,'Devi','JAVA',80,100);
```

```
select * from students11;
```

```
select Marks_scored,id,name,rank() over(order by Marks_scored  
desc),dense_rank() over(order by Marks_scored desc),row_number() over(order by  
Marks_scored desc)from students11;
```

```
with cte as(select AVG(Marks_scored) as average_marks from students11)
```

```
select students11.*,cte.average_marks from students11,cte
```

```
with cte as(select MIN(Marks_scored) as Min_marks from students11)
```

```
select students11.*,cte.Min_marks from students11,cte
```

```
with cte as(select MAX(Marks_scored) as Max_marks from students11)
```

```
select students11.*,cte.Max_marks from students11,cte
```

OUTPUT:

ID	NAME	SALARY	AGE	LOCATION	MOBILE
101	Suka	50000	22	Palayamkottai	8667897130
102	Saran	40000	25	Tuticorin	9807654321
103	Priya	30000	29	Tirunelveli	8765432190
104	Hari	20000	24	Chennai	9678543210
105	Pavi	36000	23	Tenkasi	9390876543
106	Ram	55000	25	Kanyakumari	9345678902

COUNT(AGE)	SALARY
6	20000

AVG(SALARY)	MIN(SALARY)
38500	20000

SUM(SALARY)	MAX(SALARY)
231000	55000

ID	NAME	SUBJECT_NAME	MARKS_SCORED	TOTAL_MARKS
1	Joseph	DIP	99	100
2	Ravi	IOT	90	100
3	Jack	DS	87	100
4	Steve	SC	87	100
5	Karthi	Python	95	100
6	Devi	JAVA	80	100

MARKS_SCORED	ID	NAME	RANK() OVER(ORDERBYMARKS_SCOREDEDESC)	DENSE_RANK() OVER(ORDERBYMARKS_SCOREDEDESC)	ROW_NUMBER() OVER(ORDERBYMARKS_SCOREDEDESC)
99	1	Joseph	1	1	1
95	5	Karthi	2	2	2
90	2	Ravi	3	3	3
87	4	Steve	4	4	4
87	3	Jack	4	4	5
80	6	Devi	6	5	6

ID	NAME	SUBJECT_NAME	MARKS_SCORED	TOTAL_MARKS	AVERAGE_MARKS
1	Joseph	DIP	99	100	89.6666666666666666666666666667
2	Ravi	IOT	90	100	89.6666666666666666666666666667
3	Jack	DS	87	100	89.6666666666666666666666666667
4	Steve	SC	87	100	89.6666666666666666666666666667
5	Karthi	Python	95	100	89.6666666666666666666666666667
6	Devi	JAVA	80	100	89.6666666666666666666666666667

ID	NAME	SUBJECT_NAME	MARKS_SCORED	TOTAL_MARKS	MIN_MARKS
1	Joseph	DIP	99	100	80
2	Ravi	IOT	90	100	80
3	Jack	DS	87	100	80
4	Steve	SC	85	100	80
5	Karthi	Python	95	100	80
6	Devi	JAVA	80	100	80

ID	NAME	SUBJECT_NAME	MARKS_SCORED	TOTAL_MARKS	MAX_MARKS
1	Joseph	DIP	99	100	99
2	Ravi	IOT	90	100	99
3	Jack	DS	87	100	99
4	Steve	SC	85	100	99
5	Karthi	Python	95	100	99
6	Devi	JAVA	80	100	99

RESULT:

Thus the above SQL Queries are created and the output is verified successfully.