EXPERIMENT – 8

AIM: Summary Queries: Queries using Aggregate functions, Group By Clause, Having Clause ROLLUP Operator.

Aggregate Functions:

In database management an aggregate function is a function where the values of multiple rows are grouped together as input on certain criteria to form a single value of more significant meaning.

1. COUNT Function:

   COUNT function is used to Count the number of rows in a database table. It can work on both numeric and non-numeric data types.

   Syntax:

   ```
   SELECT COUNT (*) FROM table;
   SELECT COL_NAME, COUNT (*) FROM table WHERE condition;
   SELECT COUNT (DISTINCT COL_NAME) FROM table;
   ```

   Example:

   ```
   SELECT count(ssn) as numberofemployees from EMPLOYEE_91;
   ```

   Output:

	numberofemployees
▶	7

   Example:

   ```
   SELECT (count( distinct dno)) as numberofdepartments from
   EMPLOYEE_91;
   ```

   Output:

	numberofdepartments
▶	3

2. SUM Function:

   Sum function is used to calculate the sum of all selected columns. It works on numeric fields only.

   Syntax:

   ```
   SELECT SUM (COL_NAME) FROM table;
   SELECT SUM (COL_NAME) FROM table WHERE condion;
   ```

   Example :

```
SELECT sum(salary) as sumofsalary FROM EMPLOYEE_91;
```

Output :

	sumofsalary
▶	251000.00

Example:

```
SELECT dno,sum(salary) as sumofsalarybydepartment FROM
EMPLOYEE_91 group by dno;
```

Output:

	dno	sumofsalarybydepartment
▶	1	55000.00
	4	93000.00
	5	103000.00

3. AVG Function:

The AVG function is used to calculate the average value of the numeric type. AVG function returns the average of all non-Null values.

Syntax:

```
SELECT AVG (COL_NAME) FROM table;
```

Example:

```
SELECT avg(salary) avgsalaryofemployees FROM EMPLOYEE_91;
```

Output:

	avgsalaryofemployees
▶	35857.142857

Example:

```
SELECT dno,avg(Salary) as avgsalaryofdepartment FROM
EMPLOYEE_91 group by dno;
```

Output:

	dno	avgsalaryofdepartment
▶	1	55000.000000
	4	31000.000000
	5	34333.333333

4. MAX Function:

MAX function is used to find the maximum value of a certain column. This function determines the largest value of all selected values of a column.

Syntax:

SELECT MAX (COL_NAME) FROM table;

Example:

SELECT fname,lname,max(salary) as maxsalaryofemployee FROM EMPLOYEE_91;

Output:

	fname	lname	maxsalaryofemployee
▶	Franklin	Wong	55000.00

Example:

SELECT dno,fname,lname,max(salary) as maxsalaryofdepartment FROM EMPLOYEE_91 group by dno;

Output:

	dno	fname	lname	maxsalaryofdepartment
▶	1	James	Bong	55000.00
	4	Jennifer	Wallace	43000.00
	5	Franklin	Wong	40000.00

5. MIN Function:

MIN function is used to find the minimum value of a certain column. This function determines the smallest value of all selected values of a column.

Syntax:

SELECT MIN (COL_NAME) FROM table;

Example:

SELECT fname,lname,min(salary) minsalaryofemployee FROM EMPLOYEE_91;

Output:

	fname	lname	minsalaryofemployee
▶	Franklin	Wong	25000.00

Example:

```
SELECT dno,fname,lname,min(salary) minsalaryofdepartment
FROM EMPLOYEE_91 group by dno;
```

Output:

	dno	fname	lname	minsalaryofdepartment
▶	1	James	Bong	55000.00
	4	Jennifer	Wallace	25000.00
	5	Franklin	Wong	25000.00

GROUP BY CLAUSE:

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.

Syntax:

```
SELECT COL_NAME FROM table WHERE condition GROUP BY
COL_NAME;
```

Example:

```
SELECT sex,count(sex) as noofmaleandfemaleemployee FROM
EMPLOYEE_91 group by Sex;
```

Output:

	sex	noofmaleandfemaleemployee
▶	M	4
	F	3

Example:

```
SELECT dno,fname,lname,min(salary) minsalaryofdepartment
FROM EMPLOYEE_91 group by dno;
```

Output:

	dno	fname	lname	minsalaryofdepartment
▶	1	James	Bong	55000.00
	4	Jennifer	Wallace	25000.00
	5	Franklin	Wong	25000.00

HAVING CLAUSE:

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

Syntax:

SELECT COL_NAME FROM table WHERE condition GROUP BY COL_NAME HAVING condition;

Example:

SELECT dname as departmentname,COUNT(ssn) as numberofemployee FROM EMPLOYEE_91,DEPARTMENT_91 WHERE Dno=Dnumber GROUP BY Dno HAVING count(ssn)>=1;

Output:

	departmentname	numberofemployee
▶	Headquarters	1
	Adminstration	3
	Research	3

Example:

SELECT dname as departmentname,AVG(salary) as maxavgsalarydepartment FROM EMPLOYEE_91, DEPARTMENT_91 WHERE Dno=Dnumber GROUP BY Dno HAVING AVG(Salary)> (Select AVG(salary) from employee_91);

Output:

	departmentname	maxavgsalarydepartment
▶	Headquarters	55000.000000

ROLL UP Operator:

The ROLLUP is an extension of the GROUP BY clause. The ROLLUP option allows you to include extra rows that represent the subtotals, which are commonly referred to as super-aggregate rows, along with the grand total row. By using the ROLLUP option, you can use a single query to generate multiple grouping sets.

Syntax:

SELECT COL_NAME, AGGREGATE FUNCTON FROM table GROUP BY ROLLUP (COL_NAME);
SELECT COL_NAME, AGGREGATE FUNCTION FROM table GROUP BY COL_NAME WITH ROLLUP;

Example:

SELECT dno,COUNT(*) as noofemployeeindepartment FROM EMPLOYEE_91 GROUP BY dno WITH ROLLUP;

Output:

dno	noofemployeeindepartment
1	1
4	3
5	3
NULL	7

Example:

SELECT dno,COUNT(distinct dno) as noofdepartment FROM EMPLOYEE_91 GROUP BY dno WITH ROLLUP;

Output:

dno	noofdepartment
1	1
4	1
5	1
NULL	3