**Experiment-3**

**Aim**

Queries using Aggregatefunctions(COUNT,SUM,AVG,MAXandMIN),GROUPBY, HAVING and Creation and dropping of Views.

**Procedure**

**create table : create table for employee with following column names.**

**Emp\_id, E\_name, deptno, age, and salary.**

**Example :** SQL>create table emp26(emp\_id number(3) primary key,E\_namevarchar(12),deptno number(2),age number(2),salary number(10,2));

**Output :** Table Created.

**Insert Records**

**query to insert the records for employee table.**

**Example**:SQL> insert into emp26 values(&emp\_id,'&e\_name',&deptno,&age,&salary);

Enter value for emp\_id: 102

Enter value for e\_name: Santhosh

Enter value for deptno: 2

Enter value for age: 29

Enter value for salary: 58000

old 1: insert into emp26 values(&emp\_id,'&e\_name',&deptno,&age,&salary)

new 1: insert into emp26 values(102,'Santhosh',2,29,58000)

1 row created.

SQL> /

Enter value for emp\_id: 103

Enter value for e\_name: Amala

Enter value for deptno: 2

Enter value for age: 34

Enter value for salary: 66000

old 1: insert into emp26 values(&emp\_id,'&e\_name',&deptno,&age,&salary)

new 1: insert into emp26 values(103,'Amala',2,34,66000)

1 row created.

SQL> select \* from emp26;

EMP\_ID E\_NAME DEPTNO AGE SALARY

---------- ------------ ---------- ---------- ----------

101 abhilash 1 26 59000

102 Santhosh 2 29 58000

103 Amala 2 34 66000

104 kalyan 1 44 100000

105 Gowri 3 35 80000

106 durga 3 27 70000

107 Arun 1 28 67000

108 Varun 2 33 35000

8 rows selected.

**1. COUNT function:** The COUNT() function returns the number of rows that matches a specified criterion.

|  |
| --- |
| **Syntax:** SELECT COUNT(column name) FROM Table\_name WHERE condition(optional) ; |

**Example1:** SELECT COUNT(\*) FROM emp26;

**Example2:** SELECT COUNT(DEPTNO) FROM emp26 WHERE (deptno=2);

**2. SUM() function:** The SUM() function returns the total sum of a numeric column.

|  |
| --- |
| **Syntax:** SELECT SUM(column name) FROM Table\_nameWHERE condition(optional); |

**Example** : SQL>select sum(salary) from emp26;

3. **AVG() function:** The AVG() function returns the Average value of a column.

|  |
| --- |
| **Syntax:** SELECT AVG(column name) FROM Table\_nameWHERE condition(optional) ; |
| **Example:** SQL> select sum(salary),avg(age) from emp26;  4. **MAX() function**: The MAX() function returns the Maximum value of a column.   |  | | --- | | **Syntax:** SELECT MAX(column name) FROM Table\_nameWHERE condition(optional); |   **Example :** SQL> select max(salary),max(age) from emp26;  **Example2:** SQL> select max(salary),max(age) from emp26 where salary>50000;  5.**MIN() function**: The MIN() function returns the Maximum value of a column.   |  | | --- | | **Syntax:** SELECT MIN(column name) FROM Table\_nameWHERE condition(optional); |   **Example :** SQL> select min(salary), min(age) from emp26;  **Example2** : SQL> select min(salary), min(age) from emp26 where salary>50000;  **6.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".  **Syntax:** SELECT column\_name(s) FROM table\_name WHERE condition GROUP BY column\_name(s).  **Example:**  SQL> SELECT deptno,SUM(salary) as Sal FROM emp26 GROUP BY deptno;  **7.HAVING:** HAVING clause is used to filter the result of [group by](https://www.geeksforgeeks.org/sql-group-by/) based on the specified condition. The conditions are Boolean type i.e*.*use of logical operators  (AND, OR).  **Syntax:** SELECT column1, column2, COUNT(\*) FROM table\_name GROUP BY column1, column2 HAVING condition;  **Example**  SQL> SELECT deptno,SUM(salary) as Sal FROM emp26 GROUP BY deptno HAVING sum(salary) >= 155000;  **View :**  Views in SQL are a type of **virtual table** that simplifies how users interact with data across one or more tables. Unlike**traditional tables**, a view in **SQL** does not store data on disk; instead, it dynamically retrieves data based on a pre-defined query each time it’s accessed. A view in SQL is a saved SQL query that acts as a virtual table. It can fetch data from one or more tables and present it in a customized format.  **Creation of View**  **Syntax:**  CREATE VIEW view\_name AS SELECT column1, column2, . . . . FROM table name WHERE condition(optional);  **Example:**  SQL> create view empview as select emp\_id,e\_name,deptno from emp26;  **Output :** View created.  SQL> select \* from empview;  **Output** : EMP\_ID E\_NAME DEPTNO  -------------- --------------------- ----------  101 abhilash 1  102 Santhosh 2  103 Amala 2  104 kalyan 1  105 Gowri 3  106 durga 3  107 Arun 1  108 Varun 2  8 rows selected.  **Dropping of views**  SQL allows us to delete an existing View. We can [delete](https://www.geeksforgeeks.org/sql-delete-statement/) or drop View using the **DROP statement**.  **Syntax:** DROP VIEW view\_name;  **Example :** SQL> DROP VIEW empview;  **Output** : View Dropped. |
|  |