Experiment: 3

Aim: Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

COUNT: The COUNT() function returns the number of rows that matches a specified criteria

Syntax: select count(*) from table_name;

Calculate the FNAME of employees whose gender is M.

```
SQL> select count (*) fname from employee where gender='m';

  FNAME
----------
        6
```

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

Syntax: SELECT SUM(column_name) FROM table_name;

Calculate the total salaries for each employee

```
SQL> select fname, sum(salary) from employee group by fname;
FNAME                SUM(SALARY)
-------------------- -----------
ALLEN                      3000
JONES                      2500
SMITH                     7000
MARTIN                     4000
TURNER                      6000
BLAKE                       2500
```

AVG: The AVG() function returns the average value of a numeric column.

Syntax: SELECT AVG(column_name) FROM table_name;

Calculate the average salaries from employee table

```
SQL> select avg(salary) from employee;
```

```
AVG(SALARY)
-----------
 3571.42857
```

MAX: The MAX() function returns the largest value of the selected column.

Syntax: **SELECT MAX(column_name) FROM table_name;**

Calculate the maximum salary for employee table

```
SQL> select max(salary) from employee;
MAX(SALARY)
-----------
      6000
```

MIN: The MIN() function returns the smallest value of the selected column.

Syntax: SELECT MIN(column_name) FROM table_name;

Calculate the minimum salaries from employee table

```
SQL> select min(salary) from employee;
MIN(SALARY)
-----------
      2000
```

GROUP BY

The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

Syntax: SELECT column1, column2 FROM table_name WHERE [conditions] GROUP BY column1

```
SQL> select count (gender), gender from employee GROUP BY gender;
COUNT(GENDER)   G
-------------   -
          6     M
```

Grouping using Multiple Columns

Syntax: SELECT Column1, Column2, AGGREGATE_FUNCTION (Column3) FROM TABLE1 GROUPBY Column1, Column2

```
SQL> select fname,dno from employee group by fname,dno;

FNAME                DNO
-------------------- ----------
SMITH                  10
ALLEN                  20
TURNER                 30
JONES                  10
TURNER                 20
BLAKE                  10
MARTIN                 30
```

```
SQL> select gender,dno from employee group by gender,dno;
G      DNO
-       ----------
M       20
M       30
M       10
```

HAVING

- The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions
- The WHERE clause places conditions on the selected columns, whereas the HAVING clause places conditions on groups created by the GROUP BY clause.
- The HAVING clause must follow the GROUP BY clause in a query and must also precede the ORDER BY clause if usedmk

Syntax

SELECT column1, column2 FROM table1, table2 WHERE [conditions] GROUP BY column1, column2 HAVING [conditions] ORDER BY column1, column2;

```
SQL> select fname, sum(salary) from employee group by fname having
sum(salary)>4000;
FNAME          SUM(SALARY)
-------------------- -----------
TURNER                   6000
```

View

- Views in SQL are considered as a virtual table. A view also contains rows and columns.
- To create the view, we can select the fields from one or more tables present in the database.
- A view can either have specific rows based on certain condition or all the rows of a table.

```
SQL> create view HYD as select *from employee where address='HYD';
View created.

SQL> select * from HYD;

FNAME       LNAME             SSN     B_DATE        ADDRESS       G    SALARY     DNO
--------------- -------------------- ---------- --------- ------------------------------      - ----------      ----------
MARTIN                         3333   03-NOV-16     HYD           M     4000      30

SQL> drop view HYD;
View dropped.
```