Ex.No.8	SET OPERATIONS AND AGGEREGATE FUNCTIONS

AIM

To perform various set operations, aggregate functions, group by and having clause on the relational database.

CREATE TABLE

```
SQL> CREATE TABLE student2 (
  2      SID VARCHAR(10),
  3      SNAME VARCHAR(20),
  4      SCITY VARCHAR(20)
  5  );
Table created.
SQL> CREATE TABLE student3 (
  2      SID VARCHAR(10),
  3      SNAME VARCHAR(20),
  4      SCITY VARCHAR(20)
  5  );
Table created.
SQL> CREATE TABLE employee1 (
  2      emp_id VARCHAR(10),
  3      emp_name VARCHAR(20),
  4      emp_salary NUMBER(10)
  5  );
Table created.
SQL> -- student2 values
SQL> INSERT INTO student2 (SID, SNAME, SCITY) VALUES ('cse11', 'divya',
'coimbatore');
1 row created.
SQL> INSERT INTO student2 (SID, SNAME, SCITY) VALUES ('it11', 'karthik', 'madurai');
```

```
1 row created.

SQL> INSERT INTO student2 (SID, SNAME, SCITY) VALUES ('ece11', 'priya', 'trichy');

1 row created.

SQL> INSERT INTO student2 (SID, SNAME, SCITY) VALUES ('eee11', 'vishnu', 'vellore');

1 row created.

SQL> INSERT INTO student2 (SID, SNAME, SCITY) VALUES ('cse12', 'sanjay', 'salem');

1 row created.

SQL> -- student3 values

SQL> INSERT INTO student3 (SID, SNAME, SCITY) VALUES ('cs11', 'meera', 'karur');

1 row created.

SQL> INSERT INTO student3 (SID, SNAME, SCITY) VALUES ('it12', 'ram', 'erode');

1 row created.

SQL> INSERT INTO student3 (SID, SNAME, SCITY) VALUES ('ece12', 'lakshmi',
'chennai');

1 row created.

SQL> INSERT INTO student3 (SID, SNAME, SCITY) VALUES ('eee11', 'vishnu', 'vellore');

1 row created.

SQL> INSERT INTO student3 (SID, SNAME, SCITY) VALUES ('eie11', 'anita',
'cuddalore');

1 row created.

SQL> INSERT INTO student3 (SID, SNAME, SCITY) VALUES ('cse11', 'divya',
'coimbatore');

1 row created.

SQL> INSERT INTO student3 (SID, SNAME, SCITY) VALUES ('it11', 'karthik', 'madurai');

1 row created.

SQL> INSERT INTO student3 (SID, SNAME, SCITY) VALUES ('eee12', 'swetha', 'salem');

1 row created.

SQL> COMMIT;

Commit complete.
```

UNION KEYWORD

```
SQL> SELECT * FROM student2
  2  UNION
  3  SELECT * FROM student3;
SID        SNAME                SCITY
---------- -------------------- --------------------
cse11      divya                coimbatore
it11       karthik              madurai
ece11      priya                trichy
eee11      vishnu               vellore
cse12      sanjay               salem
cs11       meera                karur
it12       ram                  erode
ece12      lakshmi              chennai
eie11      anita                cuddalore
eee12      swetha               salem

10 rows selected.
```

UNION ALL KEYWORD

```
SQL> SELECT * FROM student2
  2  UNION ALL
  3  SELECT * FROM student3;

SID        SNAME                SCITY
---------- -------------------- --------------------
cse11      divya                coimbatore
it11       karthik              madurai
ece11      priya                trichy
```

```
eee11      vishnu               vellore

cse12      sanjay               salem

cs11       meera                karur

it12       ram                  erode

ece12      lakshmi              chennai

eee11      vishnu               vellore

eie11      anita                cuddalore

cse11      divya                coimbatore


SID        SNAME                SCITY

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

it11       karthik              madurai

eee12      swetha               salem


13 rows selected.
```

INTERSECT KEYWORD

```
SQL> SELECT * FROM student2

  2  INTERSECT

  3  SELECT * FROM student3;


SID        SNAME                SCITY

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

cse11      divya                coimbatore

it11       karthik              madurai

eee11      vishnu               vellore
```

MINUS KEYWORD

```
SQL> SELECT * FROM student2

  2  MINUS
  3  SELECT * FROM student3;


SID        SNAME                SCITY
---------- -------------------- --------------------
ece11      priya              trichy
cse12      sanjay              salem
```

AGGREGATE FUNCTIONS

```
SQL> -- Insert employee data

SQL> INSERT INTO employee1 (emp_id, emp_name, emp_salary) VALUES ('E001',
'Arun', 50000);

1 row created.

SQL> INSERT INTO employee1 (emp_id, emp_name, emp_salary) VALUES ('E002',
'Divya', 60000);

1 row created.

SQL> INSERT INTO employee1 (emp_id, emp_name, emp_salary) VALUES ('E003', 'Ravi',
45000);

1 row created.

SQL> INSERT INTO employee1 (emp_id, emp_name, emp_salary) VALUES ('E004',
'Swetha', 70000);

1 row created.

SQL> INSERT INTO employee1 (emp_id, emp_name, emp_salary) VALUES ('E005',
'Karthik', 55000);

1 row created.

SQL> INSERT INTO employee1 (emp_id, emp_name, emp_salary) VALUES ('E006',
'Meera', 65000);

1 row created.

SQL> COMMIT;

Commit complete.
```

MAX

```
SQL> SELECT MAX(emp_salary) AS Maximum_Salary FROM employee1;
```

```
MAXIMUM_SALARY
--------------
        70000
```

MIN

```
SQL> SELECT MIN(emp_salary) AS Minimum_Salary FROM employee1;
MINIMUM_SALARY
--------------
        45000
```

AVG

```
SQL> SELECT AVG(emp_salary) AS Average_Salary FROM employee1;
AVERAGE_SALARY
--------------
        57500
```

SUM

```
SQL> SELECT SUM(emp_salary) AS Total_Salary FROM employee1;
TOTAL_SALARY
------------
     345000
```

COUNT

```
SQL> SELECT COUNT(*) AS Total_Employees FROM employee1;
TOTAL_EMPLOYEES
---------------
            6
```

SECOND MAXIMUM SALARY

```
SQL> SELECT MAX(emp_salary)
  2  FROM employee1
  3  WHERE emp_salary NOT IN (SELECT MAX(emp_salary) FROM employee1);
```

```
MAX(EMP_SALARY)
---------------
          65000
```

SECOND MINIMUM SALARY

```
SQL> SELECT MIN(emp_salary)
  2  FROM employee1
  3  WHERE emp_salary NOT IN (SELECT MIN(emp_salary) FROM employee1);


MIN(EMP_SALARY)
---------------
          50000
```

AGGREGATE FUNCTIONS WITH GROUPBY AND HAVING:

```
SQL> CREATE TABLE employee1(
  2  EMP_ID VARCHAR2(10),
  3  EMP_NAME VARCHAR2(20),
  4  EMP_SALARY NUMBER(10),
  5  DEPARTMENT_ID VARCHAR2(10),
  6  DEPARTMENT_NAME VARCHAR2(20)
  7  );
Table created.
SQL> INSERT INTO employee1 (EMP_ID, EMP_NAME, EMP_SALARY,
DEPARTMENT_ID, DEPARTMENT_NAME)
  2  VALUES ('E001', 'Arun', 50000, 'D001', 'HR');
1 row created.
SQL> INSERT INTO employee1 (EMP_ID, EMP_NAME, EMP_SALARY,
DEPARTMENT_ID, DEPARTMENT_NAME)
  2  VALUES ('E002', 'Divya', 60000, 'D002', 'Finance');


1 row created.
```

```
SQL> INSERT INTO employee1 (EMP_ID, EMP_NAME, EMP_SALARY, DEPARTMENT_ID, DEPARTMENT_NAME)
  2  VALUES ('E003', 'Ravi', 45000, 'D001', 'HR');

1 row created.

SQL> INSERT INTO employee1 (EMP_ID, EMP_NAME, EMP_SALARY, DEPARTMENT_ID, DEPARTMENT_NAME)
  2  VALUES ('E004', 'Swetha', 70000, 'D003', 'IT');

1 row created.

SQL> INSERT INTO employee1 (EMP_ID, EMP_NAME, EMP_SALARY, DEPARTMENT_ID, DEPARTMENT_NAME)
  2  VALUES ('E005', 'Karthik', 55000, 'D002', 'Finance');

1 row created.

SQL> INSERT INTO employee1 (EMP_ID, EMP_NAME, EMP_SALARY, DEPARTMENT_ID, DEPARTMENT_NAME)
  2  VALUES ('E006', 'Meera', 65000, 'D003', 'IT');

1 row created.

SQL> select * from employee1;

EMP_ID     EMP_NAME            EMP_SALARY DEPARTMENT DEPARTMENT_NAME
---------- ------------------- ---------- ---------- --------------------
E001       Arun                     50000 D001       HR
E002       Divya                    60000 D002       Finance
E003       Ravi                     45000 D001       HR
E004       Swetha                   70000 D003       IT
E005       Karthik                  55000 D002       Finance
E006       Meera                    65000 D003       IT


6 rows selected.
```

GROUP BY

AVERAGE SALARY OF EACH DEPARTMENT

```
SQL> SELECT DEPARTMENT_NAME, AVG(EMP_SALARY) AS AVERAGE_SALARY
  2  FROM employee1
  3  GROUP BY DEPARTMENT_NAME;

DEPARTMENT_NAME      AVERAGE_SALARY
-------------------- --------------
HR                          47500
Finance                     57500
IT                          67500
SQL> -- Only those departments where avg salary > 43000
SQL> SELECT DEPARTMENT_NAME, AVG(EMP_SALARY) AS AVG_SAL
  2  FROM employee1
  3  GROUP BY DEPARTMENT_NAME
  4  HAVING AVG(EMP_SALARY) > 43000;

DEPARTMENT_NAME         AVG_SAL
-------------------- ----------
HR                        47500
Finance                   57500
IT                        67500
```

DISTINCT

```
SQL> select distinct DEPARTMENT_NAME from employee1;
DEPARTMENT_NAME
--------------------
HR
Finance
IT
```

TO FIND THE EMPLOYEES WHO EARN SALARY HIGHER THAN THE AVG SALARY OF THEIR CITY

```
SQL> SELECT name
  2  FROM employee1
  3  WHERE salary > ALL (
  4  SELECT AVG(salary)
  5   FROM employee1 e
  6   WHERE e.address = employee1.address
  7  );

NAME
--------------------------------------------------
Swati
Raju
```

TO FIND THE NAME OF THE PERSONS WHO HAVE HIGHER SALARY THAN THE AVERAGE SALARY OF THEIR DEPARTMENT

```
SQL> SELECT name
  2  FROM employee1 e
  3  WHERE salary > ALL (
  4      SELECT AVG(salary)
  5      FROM employee1
  6      WHERE dept = e.dept
  7  );

NAME
---------------------------------
Swati
Raju
```

CONTENTS	MARKS ALLOTED	MARKS OBTAINED
Aim,Algorithm,SQL,PL/SQL	30	
Execution and Result	20	
Viva	10	
Total	60	

RESULT

Thus various set opertions, aggregate functions, group by and having clause opertions are performed on the relational database.