|  |  |
| --- | --- |
| Exp.No.8  Date: | **SET OPERATIONS AND AGGREGATE FUNCTIONS** |

**Aim**

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

**Tables**

SQL> select \*from student;

ID NAME AGE S DEPT

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

109 jeevan 18 B IT

100 abi 19 B IT

119 mithun 19 B IT

103 naresh 17 B ECE

107 deban 17 B ECE

108 raman 17 B IT

110 raman 19 B IT

SQL> select \*from student1;

ID NAME AGE S DEPT

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

109 jeevan 29 B IT

100 abi 19 B IT

119 mithun 19 B IT

10 vijay 20 B IT

11 vasanth 20 B IT

**Union**

SQL> select id,name from student union select id,name from student1;

ID NAME

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

109 jeevan

100 abi

119 mithun

103 naresh

107 deban

108 raman

110 raman

10 vijay

11 vasanth

9 rows selected.

**Union All**

SQL> select id,name from student union all select id,name from student1;

ID NAME

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

109 jeevan

100 abi

119 mithun

103 naresh

107 deban

108 raman

110 raman

109 jeevan

100 abi

119 mithun

10 vijay

11 vasanth

12 rows selected.

**Intersect**

SQL> select \*from student intersect select \*from student1;

ID NAME AGE S DEPT

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

100 abi 19 B IT

119 mithun 19 B IT

**Minus**

SQL> select \*from student minus select \*from student1;

ID NAME AGE S DEPT

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

109 jeevan 18 B IT

103 naresh 17 B ECE

107 deban 17 B ECE

108 raman 17 B IT

110 raman 19 B IT

**Aggregate function**

**Max**

SQL> select max(age) from student;

MAX(AGE)

----------

19

**Min**

SQL> select min(age) from student;

MIN(AGE)

----------

17

**Avg**

SQL> select avg(age) from student;

AVG(AGE)

----------

18

**Sum**

SQL> select sum(age) from student;

SUM(AGE)

----------

126

**Count**

SQL> select count(name) from student;

COUNT(NAME)

-----------

7

**Second Maximum age**

SQL> select max(age) from student where age not in (select max(age) from student);

MAX(AGE)

----------

18

**Second Minimum age**

SQL> select min(age) from student where age not in (select min(age) from student);

MIN(AGE)

----------

18

Group By

SQL> select dept,avg(age) from student group by dept;

DEPT AVG(AGE)

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

IT 18.4

ECE 18.6

**Group By with having**

SQL> select dept,avg(age) from student group by dept having avg(age)>18;

DEPT AVG(AGE)

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

IT 18.4

ECE 18.6

**Age is greater than average salary**

SQL> select \*from student where age>(select avg(age) from student);

ID NAME AGE S DEPT

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

100 abi 19 B IT

119 mithun 19 B IT

110 raman 19 B IT

20 kavin raj 20 B ECE

180 deepak 20 B ECE

101 mahendran 19 B ECE

6 rows selected.

**Age greater than average age of their department**

SQL> select name,age,dept from student s1 where age>(select avg(age) from student s2 group by dept having s1.dept=s2.dept);

NAME AGE DEPT

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

abi 19 IT

mithun 19 IT

raman 19 IT

karthi 19 ECE

|  |  |  |
| --- | --- | --- |
| CONTENTS | MARKS ALLOTTED | MARKS OBTAINED |
| Aim,Alogorithm,SQL,PL/SQL | 30 |  |
| Execution and Result | 20 |  |
| Viva | 10 |  |
| Total | 60 |  |

**Result:**

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