TATA

TATA CONSULTANCY SERVICES

Sub Queries – Types of Sub Queries

Version 1.0

2 - TYPE OF SUB QUERIES

Sub Queries are classified into different types

2.1 Single row subquery

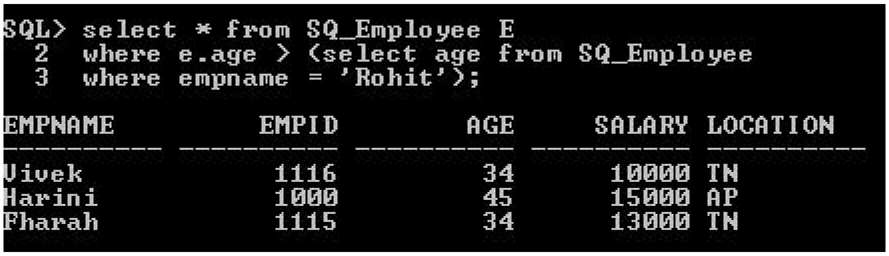
Single row sub queries returns only one row. The operators that can be used with single row subqueries are '=', '!=', '<', '<=', '>', '>=' etc

Example;

Find the Employees who are older than Rohit.

Query

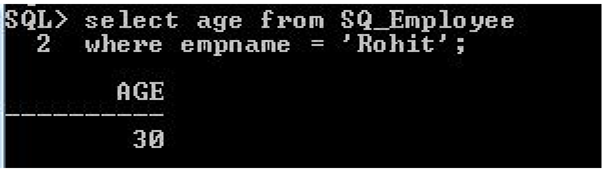
```
select * from SQ_Employee E
where e.age > (select age from SQ_Employee
where empname = 'Rohit');
```



```
SQL> select * from SQ_Employee E
  2  where e.age > (select age from SQ_Employee
  3  where empname = 'Rohit');

EMPNAME          EMPID        AGE     SALARY LOCATION
---------- ---------- ---------- ---------- ----------
Vivek             1116         34      10000 TN
Harini            1000         45      15000 AP
Fharah            1115         34      13000 TN
```

Here the subquery returns only one value i.e. Rohit's age.

```
select age from SQ_Employee
where empname = 'Rohit';
```



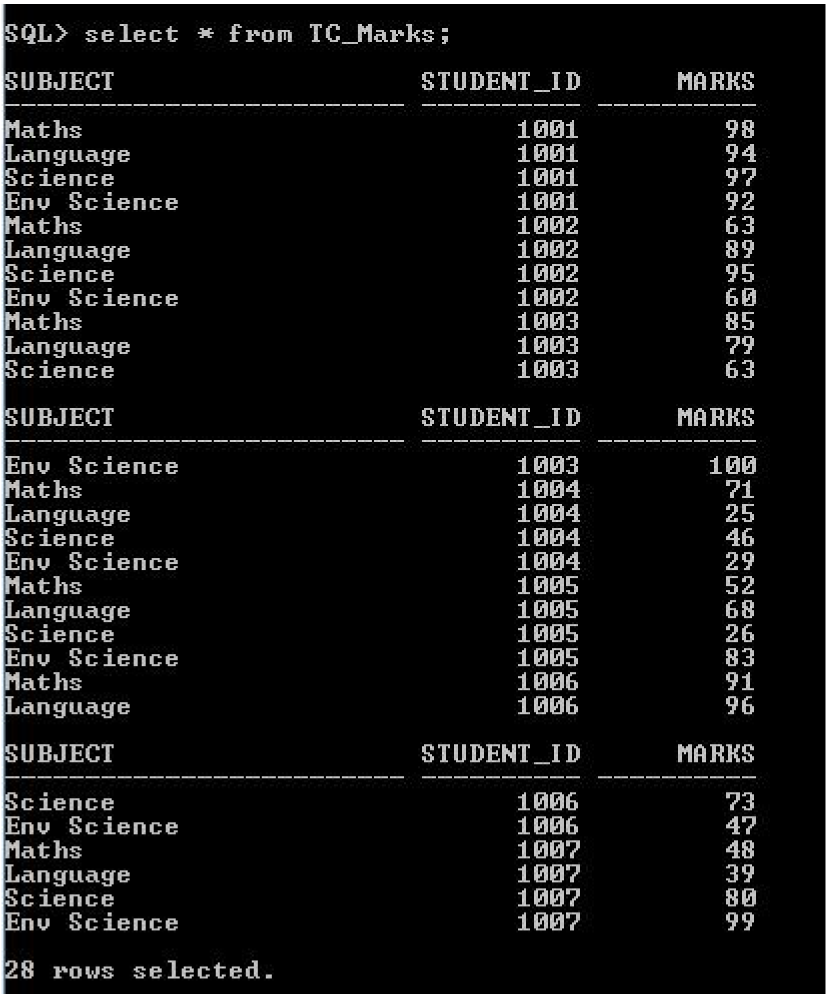
```
SQL> select age from SQ_Employee
  2  where empname = 'Rohit';

       AGE
----------
        30
```

2.2 Multi Row Subquery

Multi row sub queries are the sub queries which returns more than one row

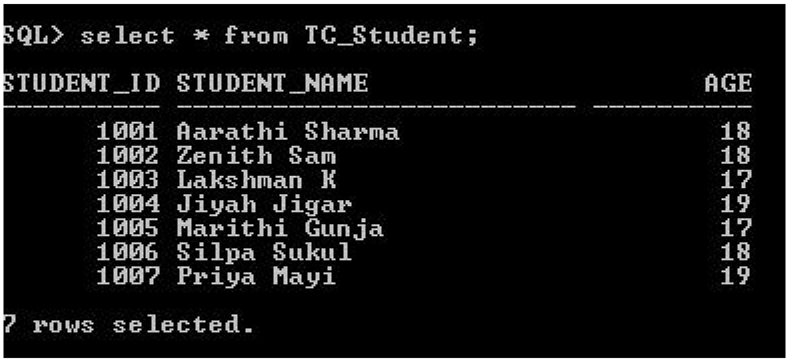
Consider the following Tables

TC_Marks



```
SQL> select * from TC_Marks;

SUBJECT                   STUDENT_ID      MARKS
------------------------- ---------- ----------
Maths                           1001         98
Language                        1001         94
Science                         1001         97
Env Science                     1001         92
Maths                           1002         63
Language                        1002         89
Science                         1002         95
Env Science                     1002         60
Maths                           1003         85
Language                        1003         79
Science                         1003         63

SUBJECT                   STUDENT_ID      MARKS
------------------------- ---------- ----------
Env Science                     1003        100
Maths                           1004         71
Language                        1004         25
Science                         1004         46
Env Science                     1004         29
Maths                           1005         52
Language                        1005         68
Science                         1005         26
Env Science                     1005         83
Maths                           1006         91
Language                        1006         96

SUBJECT                   STUDENT_ID      MARKS
------------------------- ---------- ----------
Science                         1006         73
Env Science                     1006         47
Maths                           1007         48
Language                        1007         39
Science                         1007         80
Env Science                     1007         99

28 rows selected.
```

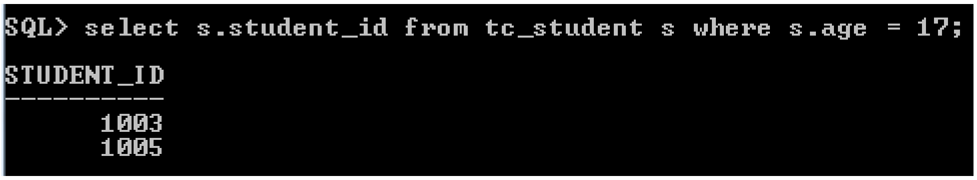
TC_Student,



```
SQL> select * from TC_Student;

STUDENT_ID STUDENT_NAME                              AGE
---------- ------------------------- ----------
      1001 Aarathi Sharma                             18
      1002 Zenith Sam                                 18
      1003 Lakshman K                                 17
      1004 Jiyah Jigar                                19
      1005 Marithi Gunja                              17
      1006 Silpa Sukul                                18
      1007 Priya Mayi                                 19

7 rows selected.
```

Consider the query

select s.student_id from tc_student s where s.age = 17;



```
SQL> select s.student_id from tc_student s where s.age = 17;

STUDENT_ID
----------
      1003
      1005
```

This returns more than one value. If this query is used as a subquery then it is a multi row sub query

The operators that can be used with multi row sub queries are 'IN', 'ANY', 'ALL', EXISTS NOT EXISTS etc.

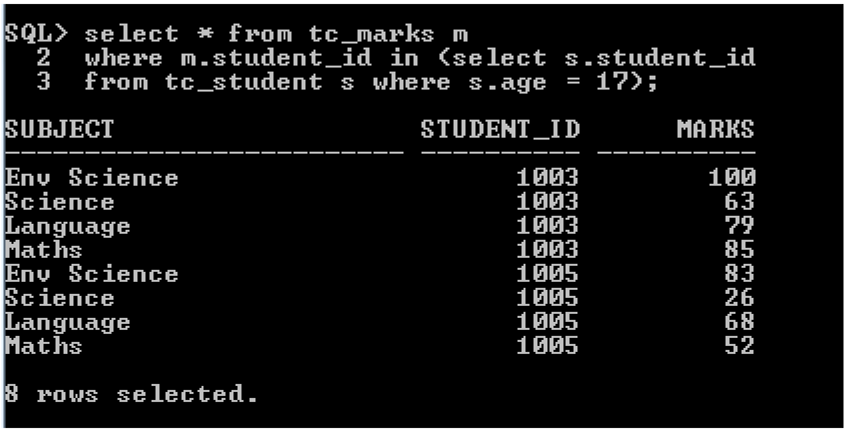
2.2.1 IN Operator

The IN operator compares each and every value from the subquery with the column values on the left side of the operator

Example

Display the marks of all the students whose age is 17.

select * from tc_marks m
where m.student_id IN (select s.student_id
from tc_student s where s.age = 17)



```
SQL> select * from tc_marks m
  2  where m.student_id in (select s.student_id
  3  from tc_student s where s.age = 17);

SUBJECT                          STUDENT_ID      MARKS
-------------------------------- ---------- ----------
Env Science                            1003        100
Science                                1003         63
Language                               1003         79
Maths                                  1003         85
Env Science                            1005         83
Science                                1005         26
Language                               1005         68
Maths                                  1005         52

8 rows selected.
```

Here the subquery will return 1003 and 1005 and the outer query will display the marks of students with ids 1003 and 1005

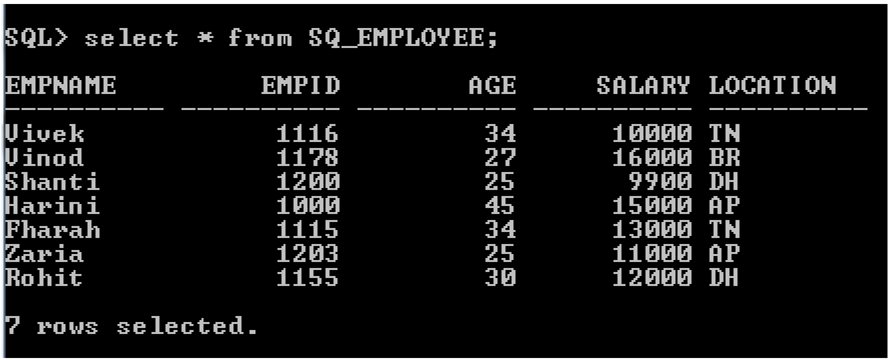
2.2.2 ANY Operator

< ANY → column values on the left side of the operator should be less than the maximum value in the list

> ANY → column values on the left side of the operator should be more than the minimum value in the list

Example

Consider the table SQ_EMPLOYEE

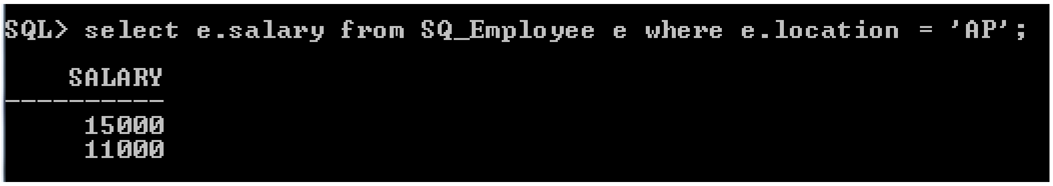


```
SQL> select * from SQ_EMPLOYEE;

EMPNAME         EMPID        AGE     SALARY LOCATION
---------- ---------- ---------- ---------- ----------
Vivek            1116         34      10000 TN
Vinod            1178         27      16000 BR
Shanti           1200         25       9900 DH
Harini           1000         45      15000 AP
Fharah           1115         34      13000 TN
Zaria            1203         25      11000 AP
Rohit            1155         30      12000 DH

7 rows selected.
```

Find the details of employees who get a salary less than any employee in location 'AP'.

Salary list of employees in 'AP'.

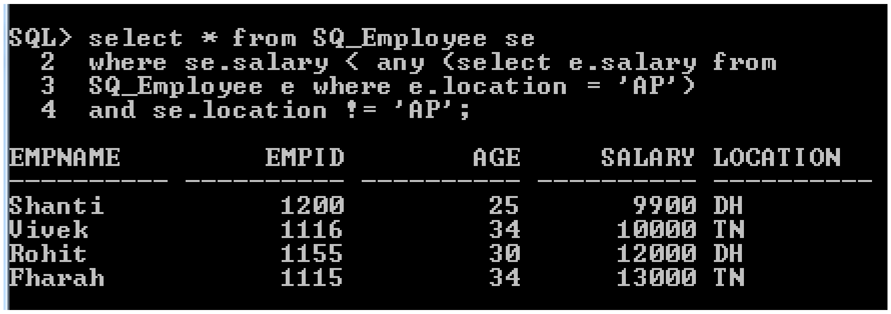
select e.salary from SQ_Employee e where e.location = 'AP';



```
SQL> select e.salary from SQ_Employee e where e.location = 'AP';

    SALARY
----------
     15000
     11000
```

Details of employees who get a salary less than any employee in location 'AP'.

select * from SQ_Employee se
where se.salary < any (select e.salary from
SQ_Employee e where e.location = 'AP')
and se.location != 'AP';

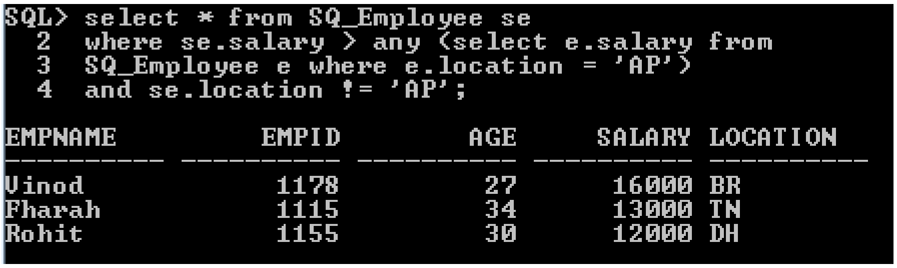


```
SQL> select * from SQ_Employee se
  2  where se.salary < any (select e.salary from
  3  SQ_Employee e where e.location = 'AP')
  4  and se.location != 'AP';

EMPNAME         EMPID        AGE     SALARY LOCATION
---------- ---------- ---------- ---------- ----------
Shanti           1200         25       9900 DH
Vivek            1116         34      10000 TN
Rohit            1155         30      12000 DH
Fharah           1115         34      13000 TN
```

Here the the details of all the employees whose salary is less than 15000 is displayed. 15000 is the highest salary in location 'AP'.

Find the details of employees who get a salary greater than any employee in location 'AP'.

select * from SQ_Employee se
where se.salary > any (select e.salary from
SQ_Employee e where e.location = 'AP')
and se.location != 'AP';



```
SQL> select * from SQ_Employee se
  2  where se.salary > any (select e.salary from
  3  SQ_Employee e where e.location = 'AP')
  4  and se.location != 'AP';

EMPNAME         EMPID        AGE     SALARY LOCATION
---------- ---------- ---------- ---------- ----------
Vinod            1178         27      16000 BR
Fharah           1115         34      13000 TN
Rohit            1155         30      12000 DH
```

Here the the details of all the employees whose salary is greater than 11000 is displayed. 11000 is the lowest salary in location 'AP'.

2.2.3 ALL Operator

< ALL → column values on the left side of the operator should be less than the minimum value in the list

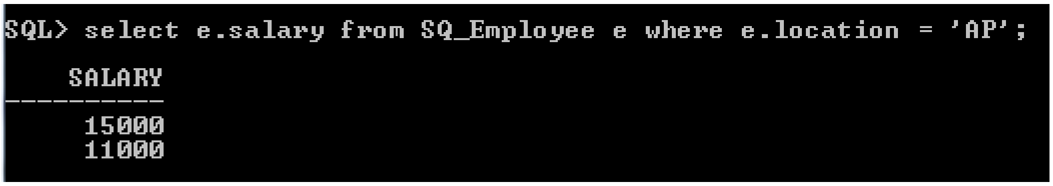
> ALL → column values on the left side of the operator should be more than the maximum value in the list

Example

Find the details of employees who gets a salary less than all employees in location 'AP'.

Salary list of employees in 'AP'.

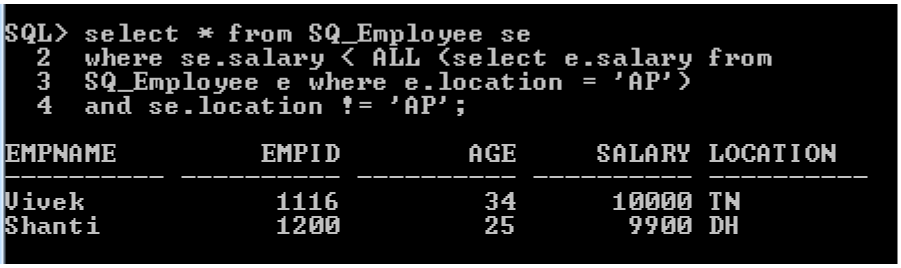
select e.salary from SQ_Employee e where e.location = 'AP';



```
SQL> select e.salary from SQ_Employee e where e.location = 'AP';

    SALARY
----------
     15000
     11000
```

Details of employees who gets a lower salary than all employees in location 'AP'.

select * from SQ_Employee se
where se.salary < ALL (select e.salary from
SQ_Employee e where e.location = 'AP')
and se.location != 'AP';

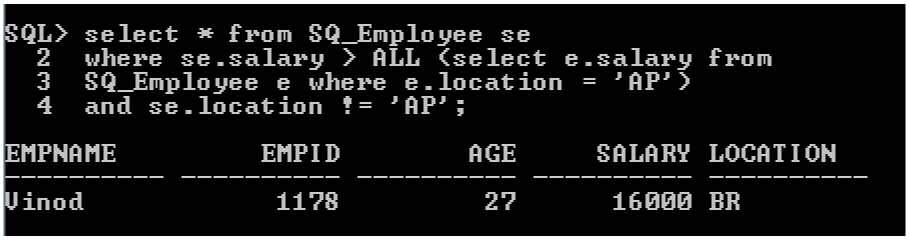


```
SQL> select * from SQ_Employee se
  2  where se.salary < ALL (select e.salary from
  3  SQ_Employee e where e.location = 'AP')
  4  and se.location != 'AP';

EMPNAME         EMPID        AGE     SALARY LOCATION
---------- ---------- ---------- ---------- ----------
Vivek            1116         34      10000 TN
Shanti           1200         25       9900 DH
```

Here the the details of all the employees whose salary is less than 11000 is displayed. 11000 is the lowest salary in location 'AP'.

Find the details of employees who gets a higher salary than all the employees in location 'AP'.

select * from SQ_Employee se
where se.salary > ALL (select e.salary from
SQ_Employee e where e.location = 'AP')
and se.location != 'AP';



```
SQL> select * from SQ_Employee se
  2  where se.salary > ALL (select e.salary from
  3  SQ_Employee e where e.location = 'AP')
  4  and se.location != 'AP';

EMPNAME         EMPID        AGE     SALARY LOCATION
---------- ---------- ---------- ---------- ----------
Vinod            1178         27      16000 BR
```

Here the the details of all the employees whose salary is higher than 15000 is displayed. 15000 is the highest salary in location 'AP'.

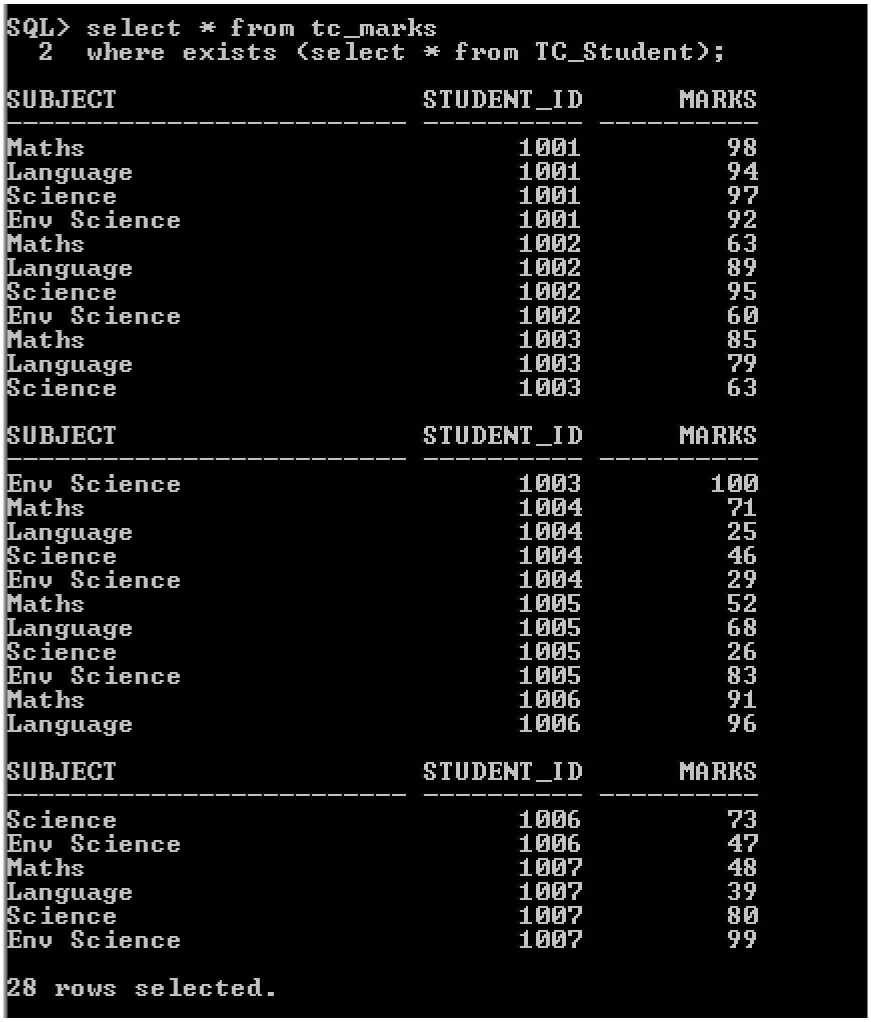
2.2.4 EXISTS Operator

The 'EXISTS' operator checks whether the subqury returns any records or not.
If a subquery returns any rows at all, EXISTS subquery is TRUE, and NOT EXISTS subquery is FALSE.

Example

Display the marks of students, if there are any student records in the student table

select * from tc_marks
where exists (select * from TC_Student);

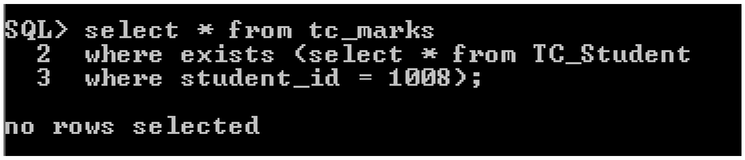


```
SQL> select * from tc_marks
  2  where exists (select * from TC_Student);

SUBJECT                   STUDENT_ID      MARKS
------------------------- ---------- ----------
Maths                           1001         98
Language                        1001         94
Science                         1001         97
Env Science                     1001         92
Maths                           1002         63
Language                        1002         89
Science                         1002         95
Env Science                     1002         60
Maths                           1003         85
Language                        1003         79
Science                         1003         63

SUBJECT                   STUDENT_ID      MARKS
------------------------- ---------- ----------
Env Science                     1003        100
Maths                           1004         71
Language                        1004         25
Science                         1004         46
Env Science                     1004         29
Maths                           1005         52
Language                        1005         68
Science                         1005         26
Env Science                     1005         83
Maths                           1006         91
Language                        1006         96

SUBJECT                   STUDENT_ID      MARKS
------------------------- ---------- ----------
Science                         1006         73
Env Science                     1006         47
Maths                           1007         48
Language                        1007         39
Science                         1007         80
Env Science                     1007         99

28 rows selected.
```

Here the subquery '***select * from TC_Student'*** returns 7 records hence the condition , 'EXITS (subquery)' is true. So the outer query displays results.

Display the marks of all students, if a student with student_id '1008' is present in the student table.

```
select * from tc_marks
where exists (select * from TC_Student
where student_id = 1008);
```



```
SQL> select * from tc_marks
  2  where exists (select * from TC_Student
  3  where student_id = 1008);

no rows selected
```

Here the there is no record in the student table for student_id '1008'. Hence the condition , 'EXITS (subquery)' is false. So the outer query does not display any records.

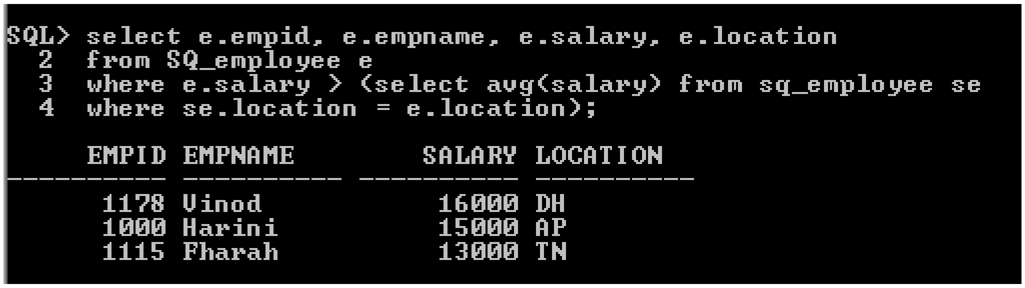
2.3 Corelated Subquery

In a corelated subquery, the subquery will use a column from the outer query. A corelated subquery is evaluated once for each row in the outer query.

Example

Find the list of employees whose salary is higher than the average salary for that location

```
select e.empid, e.empname, e.salary, e.location
from SQ_employee e
where e.salary > (select avg(salary) from sq_employee se
where se.location = e.location);
```



```
SQL> select e.empid, e.empname, e.salary, e.location
     from SQ_employee e
     where e.salary > (select avg(salary) from sq_employee se
     where se.location = e.location);

     EMPID EMPNAME         SALARY LOCATION
---------- ---------- ---------- ----------
      1178 Vinod            16000 DH
      1000 Harini           15000 AP
      1115 Fharah           13000 TN
```

2.3.1 Order of Precedence

1. First the Outer query is executed.
2. Passes the executed column value to the Inner queries WHERE clause.
3. Now the Inner query is executed.
4. The result of the Inner query is passed to the Outer queries WHERE clause.
5. Depending on the provided value the condition is qualified for the specific record.
6. If successful displays the output.