

Sub Queries – Sub Query Usage

Version 1.0

3 - SUB QUERY USAGE

Sub queries can be used almost anywhere in an SQL statement where expressions can be used. They can be used in all DML statements and create table statements

3.1 Where Clause

When the subquery is placed in the where clause, it is called nested sub query.

Consider the table SQ_EMPLOYEE

```
SQL> select * from SQ_EMPLOYEE;
```

EMPNAME	EMPID	AGE	SALARY	LOCATION
Uivek	1116	34	10000	TN
Uinod	1178	27	10009	BR
Shanti	1200	25	9900	DH
Harini	1000	45	15000	AP
Pharah	1115	34	13000	TN
Zaria	1203	25	11000	AP
Rohit	1155	30	12000	DH

```
7 rows selected.
```

Example

Find out the details of all the employees that get a salary that is greater than the average salary from SQ_EMPLOYEE table.

```
select * from SQ_Employee E where
e.salary > (Select avg(salary) from SQ_Employee)
```

```
SQL> select * from SQ_Employee E where
2 e.salary > (select avg(salary) from SQ_Employee);
```

EMPNAME	EMPID	AGE	SALARY	LOCATION
Harini	1000	45	15000	AP
Pharah	1115	34	13000	TN
Rohit	1155	30	12000	DH

3.2 From Clause

When the subquery is placed in the From clause, it is called inline view.

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
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
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.
```

Example

Find the highest total marks.

To find the total marks

```
select m.student_id, sum(m.marks) as Total_Marks
from tc_marks m group by m.student_id;
```

```
SQL> select m.student_id, sum(m.marks) as Total_Marks
2  from tc_marks m group by m.student_id;
```

STUDENT_ID	TOTAL_MARKS
1003	327
1006	307
1001	381
1002	307
1007	266
1004	171
1005	229

```
7 rows selected.
```

We have to find the highest value from this list of total marks.

```
select max(Total_Marks)
from (select sum(m.marks) as Total_Marks
from tc_marks m group by m.student_id);
```

```
SQL> select max(Total_Marks)
2  from (select sum(m.marks) as Total_Marks
3  from tc_marks m group by m.student_id);
```

MAX(TOTAL_MARKS)
381

3.3 Joins

Example

Find the deviation of each employees individual salary from the average salary.
(Individual salary – average slaray)

Average Salary Of Employees

select Round(avg(E.salary)) as Avg_salary from SQ_EMPLOYEE E

```
SQL> select Round(avg(E.salary)) as Avg_salary from SQ_EMPLOYEE E;
AUG_SALARY
-----
12414
```

Deviation of each employees individual salary from the average salary

*select SE.EMPNAME, SE.SALARY,
SE.SALARY - A.Avg_sal Deviation , A.Avg_sal
from SQ_Employee SE cross join
(select Round(avg(e.salary)) as Avg_sal from SQ_EMPLOYEE E) A*

```
SQL> select SE.EMPNAME, SE.SALARY,
2 SE.SALARY - A.Avg_sal Deviation , A.Avg_sal
3 from SQ_Employee SE cross join
4 (select Round(avg(e.salary)) as Avg_sal from SQ_EMPLOYEE E) A;

EMPNAME      SALARY  DEVIATION  AVG_SAL
-----
Uivek         10000     -2414     12414
Uinod         16000     3586      12414
Shanti         9900     -2514      12414
Harini         15000     2586      12414
Pharah         13000      586      12414
Zaria          11000     -1414      12414
Rohit          12000     -414      12414

7 rows selected.
```

3.4 Create table Statements

Syntax:

Create Table table_name as (Sub query);

Example

Create a table which has the student ids and their total marks from TC_Marks.

```
create table Student_Total_Marks as
(select m.student_id, sum(m.marks) as Total_Marks
from tc_marks m group by m.student_id);
```

```
SQL> create table Student_Total_Marks as
  2  (select m.student_id, sum(m.marks) as Total_Marks
  3  from tc_marks m group by m.student_id);

Table created.

SQL> Desc Student_Total_Marks;
+-----+-----+-----+
Name                               Null?   Type
+-----+-----+-----+
STUDENT_ID                         NOT NULL NUMBER
TOTAL_MARKS                        NUMBER

SQL> Select * from Student_Total_Marks;

STUDENT_ID TOTAL_MARKS
+-----+-----+
      1003         327
      1006         307
      1001         381
      1002         307
      1007         266
      1004         171
      1005         229

7 rows selected.
```

When you create a table in this method, keys, indexes, and most of the constraints do not get copied. Not null constraint will get copied.

3.5 Insert statement

Syntax:

Insert into table_name (column list) (sub query);

Example

Insert student ids and their total marks from TC_Marks into the Student_Total_Marks table.

*Insert into Student_Total_Marks
(select m.student_id, sum(m.marks) as Total_Marks
from tc_marks m group by m.student_id);*

```
SQL> select * from Student_Total_Marks;
no rows selected

SQL> Insert into Student_Total_Marks
  2  (select m.student_id, sum(m.marks) as Total_Marks
  3  from tc_marks m group by m.student_id);

7 rows created.

SQL> commit;
Commit complete.

SQL>
SQL> select * from Student_Total_Marks;

STUDENT_ID TOTAL_MARKS
-----
1003          327
1006          307
1001          381
1002          307
1007          266
1004          171
1005          229

7 rows selected.
```

3.6 Update statements

Syntax:

*Update table_name
set column_name = (subquery)
where condition;*

Example

Update the location of employee with emp_id 1178 to the same location as that of employee with empid 1200

*UPDATE sq_employee
SET Location=(SELECT Location
FROM sq_employee WHERE empid=1200)
WHERE empid=1178;*

```
SQL> select * from sq_employee;
```

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

```
7 rows selected.

SQL> UPDATE sq_employee
2 SET Location=(SELECT Location
3 FROM sq_employee WHERE empid=1200)
4 WHERE empid=1178;

1 row updated.

SQL> commit;

Commit complete.

SQL> select * from sq_employee;
```

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

```
7 rows selected.
```

3.7 Delete statements

Syntax:

Delete From table_name
Where column_name operator (subquery);

Example:

Delete the details of the employee who has the lowest salary from SQ_Employee table

Delete From SQ_Employee e
Where e.salary = (select min(salary) from SQ_Employee);


```
SQL> select * from sq_employee;
```

EMPNAME	EMPID	AGE	SALARY	LOCATION
Uivek	1116	34	10000	TN
Uinod	1178	27	16000	DH
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.
```

```
SQL> Delete From SQ_Employee e  
2 Where e.salary = (select min(salary) from SQ_Employee);
```

```
1 row deleted.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> select * from sq_employee;
```

EMPNAME	EMPID	AGE	SALARY	LOCATION
Uivek	1116	34	10000	TN
Uinod	1178	27	16000	DH
Harini	1000	45	15000	AP
Fharah	1115	34	13000	TN
Zaria	1203	25	11000	AP
Rohit	1155	30	12000	DH

```
6 rows selected.
```