

Course code : **CSE2007**
Course title : **Database Management System**
Module : **3**
Topic : **5**

SQL - SUBQUERY

Objectives

This session will give the knowledge about

- SQL SUBQUERY

Objectives

```
CREATE TABLE employee (eid number(3), ename varchar(10), age number(2), did  
number(3), exp number(3,1), salary number(6,2), hra number(6,2), PRIMARY  
KEY(eid), CONSTRAINT fk_key FOREIGN KEY (did) REFERENCES department);
```

```
CREATE TABLE department(did NUMBER(3), dname VARCHAR2(10), strength  
number(3), PRIMARY KEY(did));
```

```
CREATE TABLE projects(pid NUMBER(3),pname VARCHAR2(10), dname  
VARCHAR2(10), eid NUMBER(3));
```

```
CREATE SEQUENCE seq_eid START WITH 101 INCREMENT BY 1 MINVALUE  
100 MAXVALUE 1000 NOCACHE;
```

Objectives

INSERT INTO employee VALUES (103, 'Reddy', 30,3,7,1200,120);

INSERT INTO employee VALUES (101, 'Naidu', 28,2,5,1800,120);

INSERT INTO employee VALUES (102,'Mark',32,1,4,1100,70);

INSERT INTO employee VALUES (104,'David',25,2,1,7000,520);

INSERT INTO project VALUES(801,'website','Developers',101);

INSERT INTO project VALUES(802,'salary credit','Accounts',102);

INSERT INTO project VALUES(803,'recruitment','HR',101);

INSERT INTO project VALUES(804,'recruitment','HR',104);

SQL Subquery

Employee

EID	ENAME	AGE	DID	EXP	SALARY	HRA
103	Reddy	30	3	7	1200	120
101	Naidu	28	2	5	1800	120
102	Mark	32	1	4	1100	70
104	David	25	2	1	7000	520

Department

DID	DNAME	STRENGTH
1	Accounts	10
2	HR	5
3	Developers	50
4	Testers	20

Projects

PID	PNAME	DNAME	EID
801	website	Developers	102
802	salary credit	Accounts	102
803	recruitment	HR	101
803	recruitment	HR	104

SQL Subquery

A subquery is a SELECT statement that is embedded in a clause of another SELECT statement.

You can place the subquery in a number of SQL clauses:

- WHERE clause
- HAVING clause
- FROM clause

SQL Subquery

Syntax

SELECT <column, ...>

FROM <table>

WHERE expression operator

(SELECT <column, ...>

FROM <table>

WHERE <condition>);

The portion in **blue** is considered as the "inner query", while the portion in **red** is considered as the "outer query".

SQL Subquery

A subquery is a query within another query. The outer query is called as main query and inner query is called as subquery.

The subquery is often referred to as a nested SELECT, sub-SELECT, or inner SELECT statement.

The **subquery generally executes first**, and its **output is used to complete the query** condition for the main or outer query.

Enclose subqueries in parentheses.

SQL Subquery

Place subqueries on the right side of the comparison operator.

Do not add an ORDER BY clause to a subquery.

Use single-row operators with singlerow subqueries.

Use multiple-row operators with multiple-row subqueries

Types of Subqueries

Subquery	Description
Single-Row Subquery	Returns one row of results that consists of one column to the outer query
Multiple-Row Subquery	Returns more than one row of results to the outer query
Multiple-Column Subquery	Returns more than one column of results to the outer query
Correlated Subquery	References a column in the outer query. Executes the subquery once for every row in the outer query
Uncorrelated Subquery	Executes the subquery first and passes the value to the outer query

Single-Row Subqueries

A single -row subquery is one that returns one row from the inner SELECT statement. This type of subquery uses a single -row operator.

The operators that can be used with single-row subqueries are =, >, >=, <, <=, and <>.

Example:

```
SELECT ename FROM employee WHERE eid=(SELECT eid FROM projects  
WHERE pid=801);
```

```
ENAME  
-----  
Reddy
```

Single row Subquery using group function

SELECT ename,salary FROM employee WHERE salary=(SELECT MIN(salary)
FROM employee);

ENAME	SALARY
Mark	1100

SELECT dname,COUNT(PID) FROM projects GROUP BY dname;

DNAME	COUNT(PID)
Accounts	1
HR	2
Developers	1

Subquery with INSERT

```
INSERT INTO employee1 SELECT * FROM employee;
```

```
INSERT INTO employee1 SELECT * FROM employee WHERE did=3;
```

```
INSERT INTO employee1 SELECT * FROM employee WHERE eid=ANY(  
SELECT eid FROM projects WHERE pid=803);
```

Subquery with UPDATE & DELETE

UPDATE employee1 SET salary= (SELECT AVG(salary) FROM employee1);

UPDATE employee1 SET salary= (SELECT AVG(salary) FROM employee1),hra=(SELECT AVG(salary)/10 FROM employee1);

DELETE FROM employee1 WHERE salary = (SELECT AVG(salary) FROM employee1);

Multiple-Row Subqueries

Returns more than one row

Use multiple row comparison operator

- IN – Equal to ANY number in the result
- ALL – Compare value to every value returned by the sub query
- ANY - Compare value to each value returned by the sub query

Multiple-Row Subqueries

```
SELECT * FROM employee WHERE salary IN (SELECT MIN(salary) FROM  
employee GROUP BY did);
```

```
SELECT * FROM employee WHERE salary > ALL (SELECT AVG(salary) FROM  
employee GROUP BY did);
```

```
SELECT * FROM employee WHERE salary > ANY (SELECT AVG(salary) FROM  
employee GROUP BY did);
```


Nested Subqueries

```
SELECT dname FROM department WHERE did IN  
    (SELECT did FROM employee WHERE eid IN  
        (SELECT eid FROM projects WHERE pid=803));
```

Summary

This session will give the knowledge about

- SQL SUBQUERY