Subquery

Invalid query

- Find the staff whose salary is more than average staff salary.

```
SELECT *
FROM Staff
Where Salary > AVG(Salary);
```

  - ERROR! invalid use of the group function!
  - Because, you cannot use the group function AVG in WHERE clause!
  - Group functions (AVG, COUNT, MIN, MAX, SUM) can be used in SELECT, HAVING, ORDER BY
- So, what could be a solution?
- Say, the average value is known and it is 46937.506, so you can write

```
SELECT *
FROM Staff
Where Salary > 46937.506;
```

  - Or,

```
SELECT *
FROM Staff
Where Salary > (SELECT 46937.506);
```

Subquery!

  - So, we need to write a query inside other query to find this average value!

Subquery

- A subquery is a query that is embedded (or nested) inside another query
- Also known as a nested query or an inner query

- Syntax:

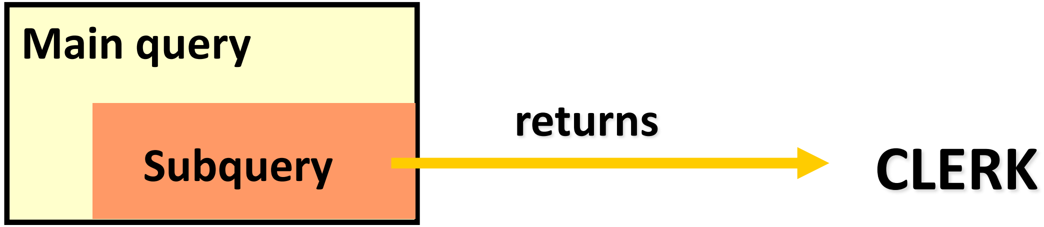
```
SELECT select_list
FROM   table
WHERE expr operator      (SELECT select_list
                          FROM   table);
```

Subquery with WHERE clause!

- The first query in the SQL statement is known as the outer query
- The query inside the SQL statement is known as the inner query
- The inner query is evaluated first and the output from this query is used as the input for the outer query
- The inner query is normally expressed inside parentheses

Subquery types

- **Single-row subquery (a single value)**



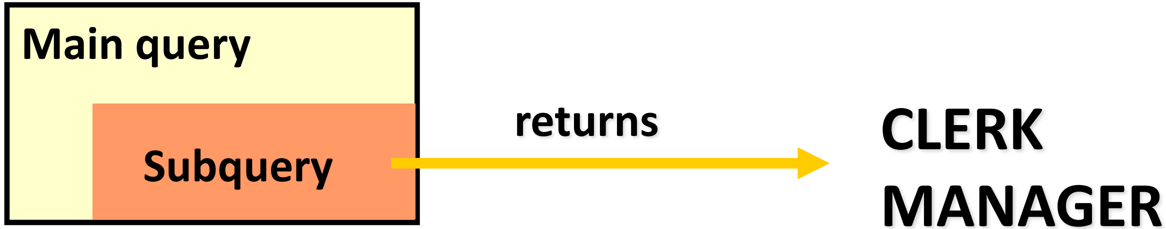
Main query

Subquery

returns

CLERK

- **Multiple-row subquery (a list of values – many rows, one column)**



Main query

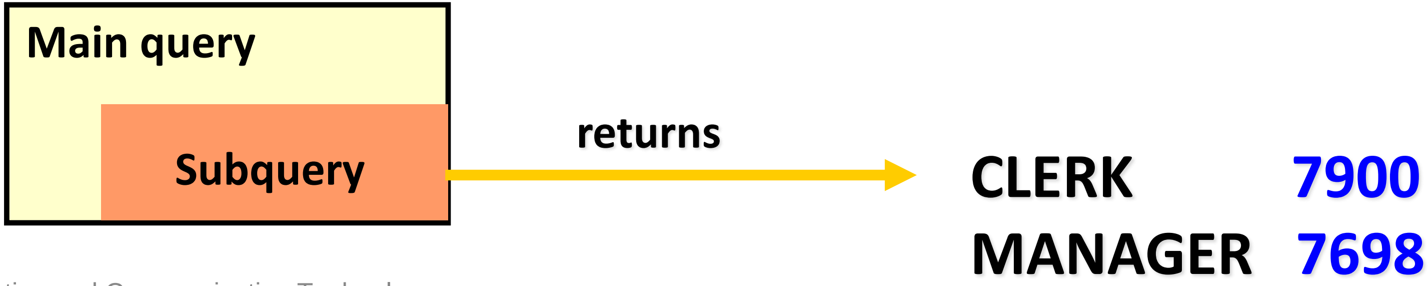
Subquery

returns

CLERK

MANAGER

- **Multiple-column subquery (a virtual table – many rows, many columns)**



Main query

Subquery

returns

CLERK	7900
MANAGER	7698

Single-row subquery

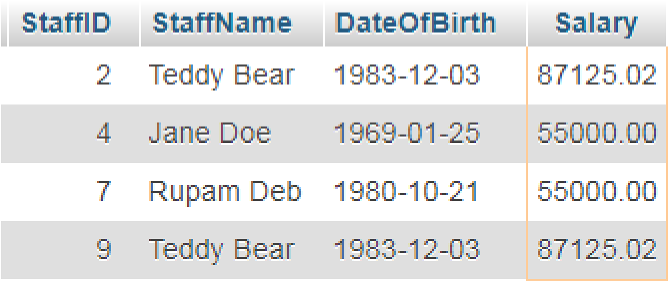
- Return only one row, one column (a single value)
- Use single-row comparison operators in WHERE clause

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

Single-row subquery

- Find the staff whose salary is more than average staff salary.

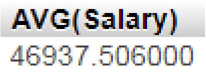
```
SELECT *
FROM Staff
Where Salary > (SELECT AVG(Salary)
                       FROM Staff);
```



StaffID	StaffName	DateOfBirth	Salary
2	Teddy Bear	1983-12-03	87125.02
4	Jane Doe	1969-01-25	55000.00
7	Rupam Deb	1980-10-21	55000.00
9	Teddy Bear	1983-12-03	87125.02

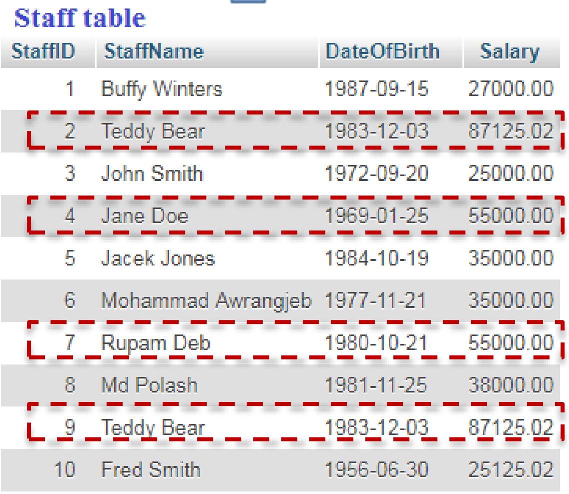
- Subquery (i.e., the inner query) returns:

```
SELECT AVG(Salary)
       FROM Staff
```



AVG(Salary)
46937.506000

- So, the outer query compares this value with Salary in Staff table and produces the output!



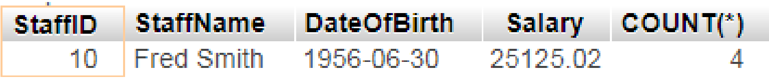
Staff table

StaffID	StaffName	DateOfBirth	Salary
1	Buffy Winters	1987-09-15	27000.00
2	Teddy Bear	1983-12-03	87125.02
3	John Smith	1972-09-20	25000.00
4	Jane Doe	1969-01-25	55000.00
5	Jacek Jones	1984-10-19	35000.00
6	Mohammad Awrangjeb	1977-11-21	35000.00
7	Rupam Deb	1980-10-21	55000.00
8	Md Polash	1981-11-25	38000.00
9	Teddy Bear	1983-12-03	87125.02
10	Fred Smith	1956-06-30	25125.02

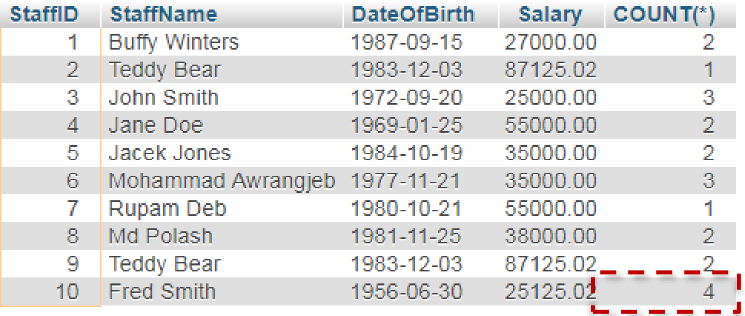
Single-row subquery

- Find staff who works in the most departments? Show all information about this staff.

```
SELECT S.StaffID, S.StaffName, S.DateOfBirth, S.Salary, COUNT(*)
FROM Staff AS S, workallocation AS WA
Where S.StaffID = WA.StaffID
GROUP BY WA.StaffID
HAVING COUNT(*) = (SELECT COUNT(*)
                    FROM workallocation AS W
                    GROUP BY W.StaffID
                    ORDER BY COUNT(*) DESC
                    LIMIT 0,1);
```



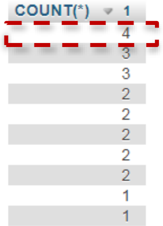
StaffID	StaffName	DateOfBirth	Salary	COUNT(*)
10	Fred Smith	1956-06-30	25125.02	4



StaffID	StaffName	DateOfBirth	Salary	COUNT(*)
1	Buffy Winters	1987-09-15	27000.00	2
2	Teddy Bear	1983-12-03	87125.02	1
3	John Smith	1972-09-20	25000.00	3
4	Jane Doe	1969-01-25	55000.00	2
5	Jacek Jones	1984-10-19	35000.00	2
6	Mohammad Awrangjeb	1977-11-21	35000.00	3
7	Rupam Deb	1980-10-21	55000.00	1
8	Md Polash	1981-11-25	38000.00	2
9	Teddy Bear	1983-12-03	87125.02	2
10	Fred Smith	1956-06-30	25125.02	4

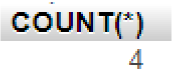
- Let's count number of departments where each staff works in

```
SELECT COUNT(*)
    FROM workallocation AS W
    GROUP BY W.StaffID
    ORDER BY COUNT(*) DESC
```



COUNT(*)
4
3
3
2
2
2
2
2
1
1

- So, the subquery (i.e., the inner query) above finds the staff who works in the most departments and returns:



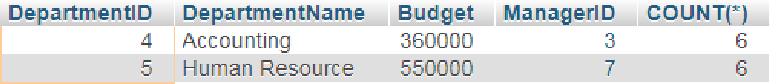
COUNT(*)
4

- Finally, the outer query compares this value!

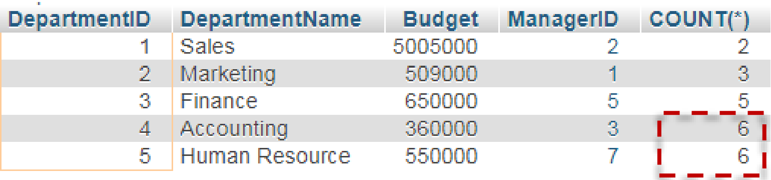
Single-row subquery

- Find the departments with the most staff working in them. Show all information of these departments.

```
SELECT D.DepartmentID, D.DepartmentName, D.Budget, D.ManagerID, COUNT(*)
FROM department AS D, workallocation AS WA
Where D.DepartmentID = WA.DepartmentID
GROUP BY WA.DepartmentID
HAVING COUNT(*) = (SELECT COUNT(*)
                   FROM workallocation AS W
                   GROUP BY W.DepartmentID
                   ORDER BY COUNT(*) DESC
                   LIMIT 0,1);
```



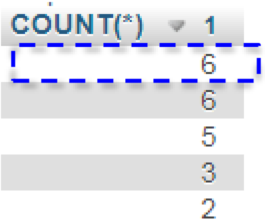
DepartmentID	DepartmentName	Budget	ManagerID	COUNT(*)
4	Accounting	360000	3	6
5	Human Resource	550000	7	6



DepartmentID	DepartmentName	Budget	ManagerID	COUNT(*)
1	Sales	5005000	2	2
2	Marketing	509000	1	3
3	Finance	650000	5	5
4	Accounting	360000	3	6
5	Human Resource	550000	7	6

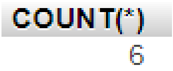
- Let's count staff in each department

```
SELECT COUNT(*)
    FROM workallocation AS W
    GROUP BY W.DepartmentID
    ORDER BY COUNT(*) DESC
```



COUNT(*) ▼ 1
6
6
5
3
2

- So, the subquery (i.e., the inner query) finds the department with maximum staff and returns:



COUNT(*)
6

- Finally, the outer query compares this value with DeparmentID in Department table and produces the output!

Thank you