DATABASE SYSTEMS

Dr. Noha Nagy

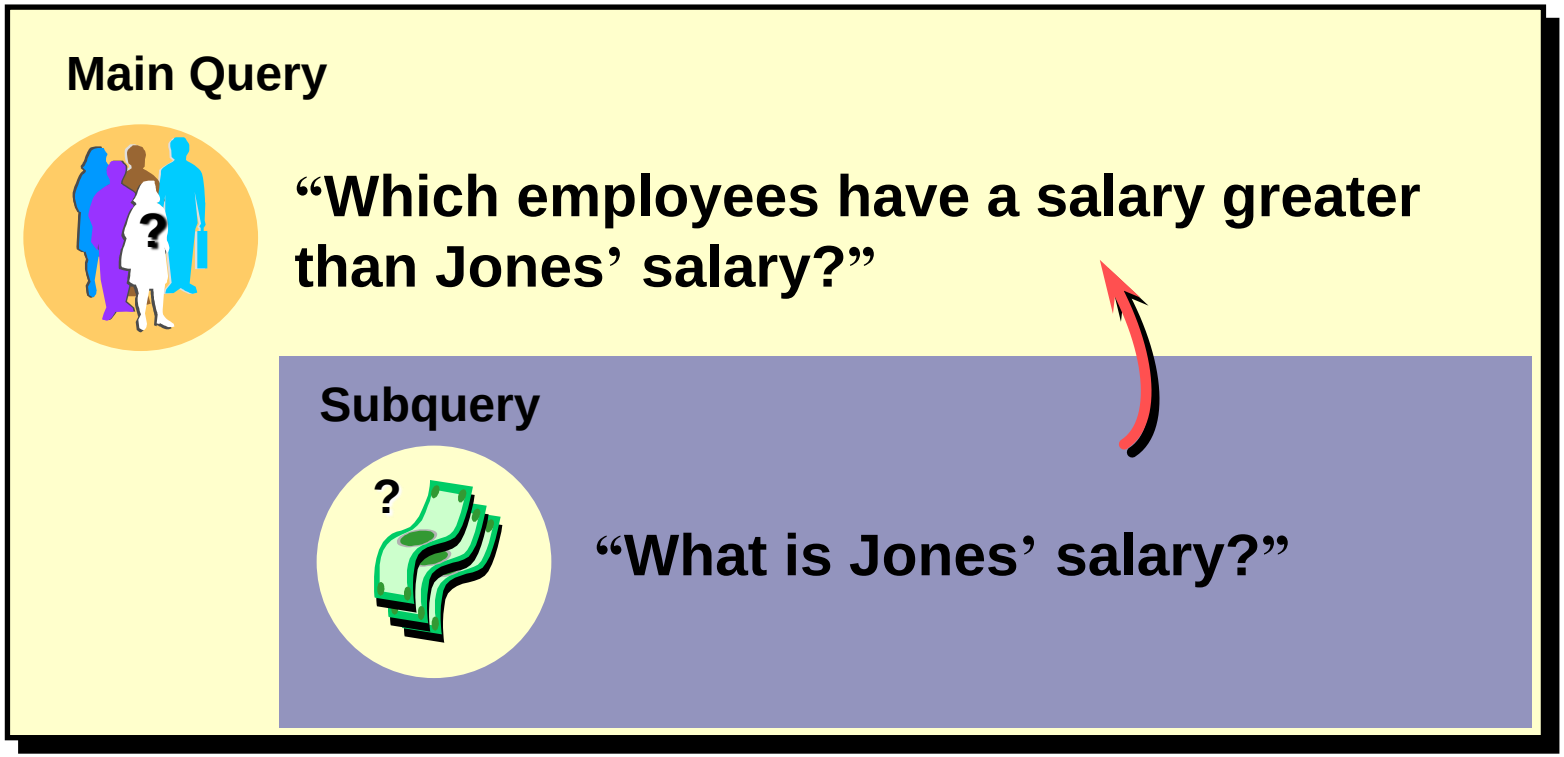
Lecture 10

Subqueries

SUBQUERIES

Using a Subquery to Solve a Problem

"Who has a salary greater than Jones'?"



Guidelines for using Subqueries

- Enclose subqueries in parentheses.
- 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 single-row subqueries.
- Use multiple-row operators with multiple-row subqueries.

Types of Subqueries

- Non Correlated
  - The inner query is first evaluated and used in evaluating the outer query
- Correlated
  - Outer queries provides values for the inner query evaluation

Non Correlated vs Corelated


```
Select name
From publishers
Where id in
        (Select id
        From titles
        Where type= 'business')
```

Inner query executes first and return a list that is used in evaluating outer query

```
Select name
From publishers
Where type in
        (Select type
        From titles
Where titles.pid= publishers.id)
```

inner query executes for every row the outer query returns

Types of subqueries results

1- Subqueries that operate on single value

- **operator (=, <, >, <>)**

2- Subqueries that operate on lists but the values must be from a single column of a table.

- **IN**
- **ANY**
- **ALL**

3- Subqueries that use the **EXISTS** operator to test the *existence* of data rows satisfying specified criteria.

1- Single Row Subqueries operators

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

Single Row Subquery


```
SQL> SELECT ename
  2  FROM   emp
  3  WHERE  sal >
  4                (SELECT sal
  5                 FROM   emp
  6                 WHERE  empno=7566);
```

2975

In the example, the inner query determines the salary of employee 7566. The outer query takes the result of the inner query and uses this result to display all the employees who earn more than this amount.

```
ENAME
----------
KING
FORD
SCOTT
```

Display the employees whose job title is the same as that of employee 7369.

```
SQL> SELECT    ename, job
  2  FROM      emp
  3  WHERE     job =
  4                 (SELECT  job
  5                       FROM      emp
  6                       WHERE     empno = 7369);
```

```
ENAME      JOB
---------- ---------
JAMES      CLERK
SMITH      CLERK
ADAMS      CLERK
MILLER     CLERK
```

Display employees whose job title is the same as that of employee 7369 and whose salary is greater than that of employee 7876.


```
SQL> SELECT    ename, job
  2  FROM      emp
  3  WHERE     job =
  4                    (SELECT      job
  5                    FROM         emp
  6                    WHERE        empno = 7369)
  7  AND       sal >
  8                    (SELECT      sal
  9                    FROM         emp
 10                    WHERE        empno = 7876);
```

CLERK

1100

```
ENAME      JOB
---------- ---------
MILLER     CLERK
```

Using Aggregate Function in Subquery

Display the employee name, job title, and salary of all employees whose salary is equal to the minimum salary.

```
SQL> SELECT   ename, job, sal
  2  FROM     emp
  3  WHERE    sal =
  4                  (SELECT     MIN(sal)
  5                  FROM        emp);
```

800

```
ENAME      JOB              SAL
---------- --------- ---------
SMITH      CLERK            800
```

Having Clause with Subqueries

Display all the departments that have a minimum salary greater than that of department 20.

```
SQL> SELECT        deptno, MIN(sal)
  2  FROM          emp
  3  GROUP BY      deptno
  4  HAVING        MIN(sal) >
  5                (SELECT    MIN(sal)
  6                FROM       emp
  7                WHERE      deptno = 20);
```

800

What is Wrong?


```
SQL> SELECT empno, ename
  2  FROM   emp
  3  WHERE  sal =
  4
  5
  6
```

```
(SELECT   MIN(sal)
FROM      emp
GROUP BY  deptno);
```

Single-row operator with multiple-row subquery

```
ERROR:
ORA-01427: single-row subquery returns more than
one row

no rows selected
```

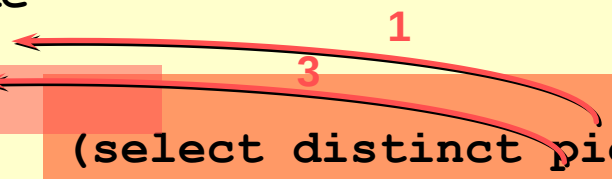
2- Multiple Row Subquery Operators

Operator	Meaning
IN	Equal to any member in the list
ANY	Compare value to each value returned by the subquery
ALL	Compare value to every value returned by the subquery

In Operator

Display all the products that have been sold at least once.

```
Select Pid, Name
From Product
Where Pid in
                (select distinct pid
                 from ProductSales
```




```
     Pid    NAME
--------- ----------
       1      TV
       3     Laptop
```

Any Operator

Display the employees whose salary is less than any clerk and who are not clerks.

```
SQL> SELECT   empno, ename, job
  2  FROM     emp
  3  WHERE    sal < ANY
  4                    (SELECT   sal
  5                    FROM      emp
  6                    WHERE     job = 'CLERK')
  7  AND      job <> 'CLERK';
```

1300
1100
800
950

```
     EMPNO ENAME      JOB
---------- ---------- ---------
      7654 MARTIN     SALESMAN
      7521 WARD       SALESMAN
```

<ANY means less than the maximum. >ANY means more than the minimum. =ANY is equivalent to IN.

All Operator

Display the employees whose salary is greater than the average salaries of all the departments.

```
SQL> SELECT   empno, ename, job
  2  FROM     emp
  3  WHERE    sal > ALL
  4                  (SELECT       avg(sal)
  5                  FROM          emp
  6                  GROUP BY      deptno);
```

1566.6667
2175
2916.6667

```
     EMPNO ENAME      JOB
---------- ---------- ---------
      7839 KING       PRESIDENT
      7566 JONES      MANAGER
      7902 FORD       ANALYST
      7788 SCOTT      ANALYST
```

>ALL means more than the maximum and <ALL means less than the minimum.

Summary

Subqueries are useful when a query is based on unknown values.

Exercise

- Student(sid , Name, age)
- Marks(sid, courseid,mark)
- Get the names of student who didn't pass(mark<40) course DB11

Select Name

From Student, Marks

Where Student.sid=Marks.sid

And mark <40

And coursed='DB11'

Exercise

- Student(sid , Name, age)
- Marks(sid, courseid,mark)
- Get the names of student who didn't pass(mark<40) course DB11

```
Select Name
From Student
Where sid in ( select sid
                    From marks
                     Where courseid = 'DB11'
                     And mark < 40);
```

Exercise

Write a query that will list the names of who is older than the average student.

Student (sid,name, age)

```
select name
from student
where age >
        (select avg(age) from student);
```

Exercise

- Write a query to display the employee number and name for all employees who work in a department with any employee whose name contains a T.

- SELECT empno, ename
- FROM emp
- WHERE deptno IN

```
( SELECT deptno
    FROM emp
   WHERE ename LIKE  '%T%' );
```