Retrieving Data Using Subqueries

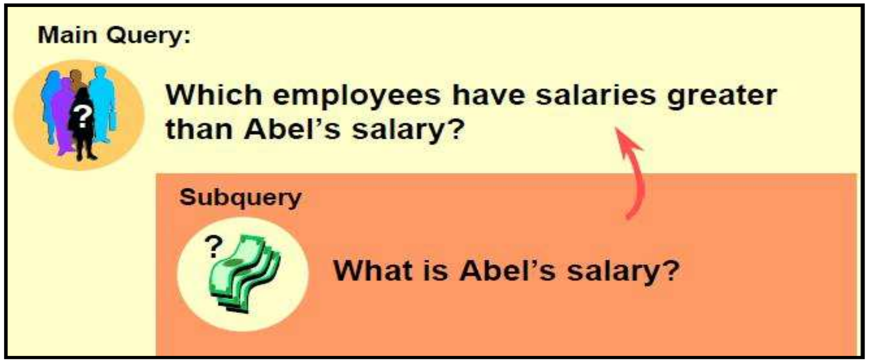
7

Lesson Agenda

- Subquery : Types, syntax and guidelines
- Single-row subqueries :
  - Group functions in a subquery
  - `HAVING` clause with subqueries
- Multiple-row subqueries :
  - Use `ALL` or `ANY` operator

Using a Subquery to Solve a Problem

- Who has a salary greater than Abel's?



Main Query:

Which employees have salaries greater than Abel's salary?

Subquery

What is Abel's salary?

- To solve a problem, you need two queries: one to fine how much Abel earns, and a second query to find who earns more than that amount.

Subquery Syntax

- A subquery is a SELECT statement embedded in a clause of another SQL statement , called the parent statement.
- The subquery (inner query) is executed once, before the main query.
- The result of the subquery is used by the main query (outer query).
- Subqueries can be used for the following purposes:
  - `WHERE` clause
  - `HAVING` clause
  - `FROM` clause

Subquery Syntax

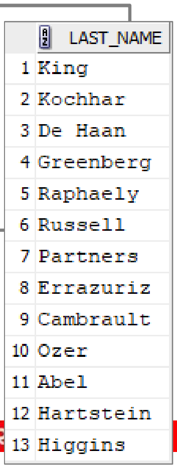
```
SELECT  select_list
FROM    table
WHERE   expr operator (SELECT  select_list
                              FROM  table);
```

- In the syntax :
  - *operator* includes a comparison condition such as >,=,or IN

Using a Subquery

In the slide example, the inner query returns the salary of the employee with employee number 149. The outer query uses the result of the inner query to display the names of all the employees who earn more than this amount.

```
SELECT  last_name
FROM    employees
WHERE   salary > (SELECT  salary
                   FROM    employees
                   WHERE   employee_id = 149);
```

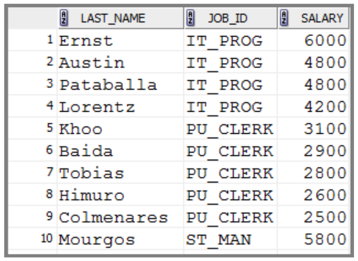


	LAST_NAME
1	King
2	Kochhar
3	De Haan
4	Greenberg
5	Raphaely
6	Russell
7	Partners
8	Errazuriz
9	Cambrault
10	Ozer
11	Abel
12	Hartstein
13	Higgins

Using a Subquery

Practice:

Display the last names, job id and salary of all employees who earn less than the average salary in the company.



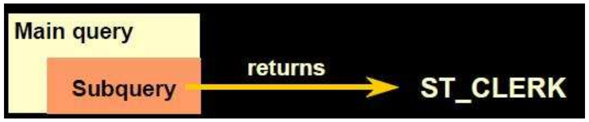
	LAST_NAME	JOB_ID	SALARY
1	Ernst	IT_PROG	6000
2	Austin	IT_PROG	4800
3	Pataballa	IT_PROG	4800
4	Lorentz	IT_PROG	4200
5	Khoo	PU_CLERK	3100
6	Baida	PU_CLERK	2900
7	Tobias	PU_CLERK	2800
8	Himuro	PU_CLERK	2600
9	Colmenares	PU_CLERK	2500
10	Mourgos	ST_MAN	5800

Guidelines for Using Subqueries

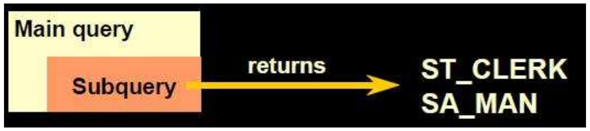
- Enclose subqueries in parentheses.
- Place subqueries on the right side of the comparison condition for readability.
- Use single-row operators with single-row subqueries and multiple-row operators with multiple-row subqueries.

Types of Subqueries

- **Single-row subquery**: Queries that return only one row from the inner `SELECT` statement



- **Multiple-row subquery**: Queries that return more than one row from the inner `SELECT` statement



Single-Row Subqueries

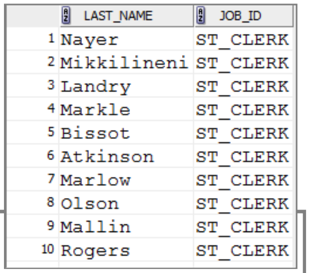
- A single-row subquery is one that return one row from the inner `SELECT` statement.
- Use single-row comparison operators.

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

Single-Row Subqueries

Example:

Display the employees whose job ID is the same as that of employee 141.



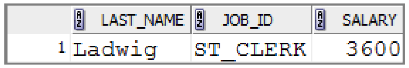
	LAST_NAME	JOB_ID
1	Nayer	ST_CLERK
2	Mikkilineni	ST_CLERK
3	Landry	ST_CLERK
4	Markle	ST_CLERK
5	Bissot	ST_CLERK
6	Atkinson	ST_CLERK
7	Marlow	ST_CLERK
8	Olson	ST_CLERK
9	Mallin	ST_CLERK
10	Rogers	ST_CLERK

```
• SELECT  last_name,job_id
• FROM employees
• WHERE  job_id =
•
•
```

Executing Single-Row Subqueries

Practice:

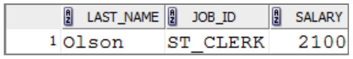
Modify in the previous slide displays employees who do the same job as employee number 141,but earn more salary than him.



	LAST_NAME	JOB_ID	SALARY
1	Ladwig	ST_CLERK	3600

Using Group Functions in a Subquery

- You can display data from a main query by using a group function in a subquery to return a single row.
- The subquery is in parentheses and is placed after the comparison condition.
- **Example:**
  - Displays the employee last name, job ID, and salary of all employees whose salary is equal to the minimum salary. The MIN group function returns a single value to the outer query.

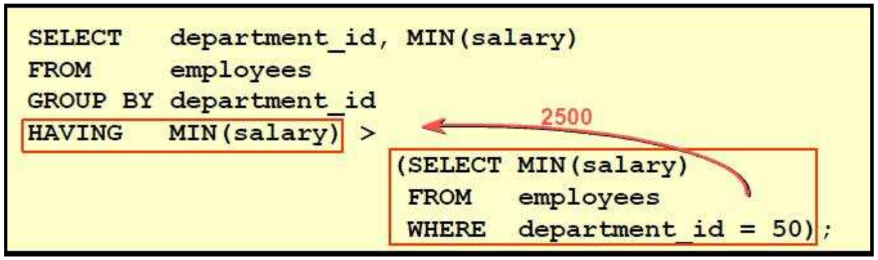


	LAST_NAME	JOB_ID	SALARY
1	Olson	ST_CLERK	2100

```
•  SELECT  last_name,job_id,  salary
•  FROM employees
•  WHERE salary =
•
```

The HAVING Clause with Subqueries

- The Oracle server executes subqueries first.
- The Oracle server returns results into the HAVING clause of the main query.
- The SQL statement in the slide displays all the departments that have a minimum salary greater than that of department 50.

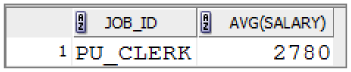


```
SELECT    department_id, MIN(salary)
FROM      employees
GROUP BY department_id
HAVING    MIN(salary) >
                        (SELECT MIN(salary)
                         FROM    employees
                         WHERE   department_id = 50);
```

The HAVING Clause with Subqueries

Example:

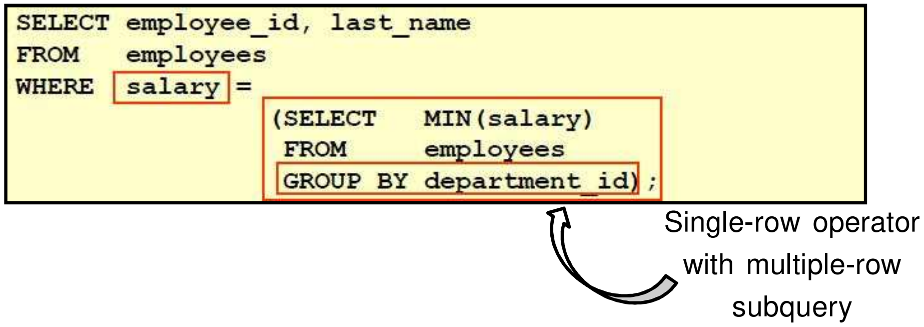
Find the job with lowest average salary.



	JOB_ID	AVG(SALARY)
1	PU_CLERK	2780

```
•  SELECT  job_id,AVG(salary)
•  FROM employees
•  GROUP  BY  job_id
•
•
•
```

What is Wrong with this Statement?



```
SELECT employee_id, last_name
FROM   employees
WHERE  salary =
                  (SELECT    MIN(salary)
                   FROM      employees
                   GROUP BY department_id);
```

Single-row operator with multiple-row subquery

- The outer query takes those results and uses them in its WHERE clause. The WHERE clause contains an equal(=) operator, an single-row comparison operator that expects only one value. The = operator cannot accept more than one value from the subquery and, therefore, generates the error.
- To correct this error, change the = operator to IN.

Will this Statement Return Rows?

```
SELECT last_name, job_id
FROM   employees
WHERE  job_id =
                 (SELECT job_id
                  FROM   employees
                  WHERE  last_name = 'Haas');
```

```
0 rows selected
```

Subquery returns no rows because there is no employee named "Haas"

- There is no employee named Haas. So the subquery returns no rows. The outer query takes the results of the subquery(null) and uses these results in its WHERE clause. The outer query finds no employee with a job ID equal to null, and so returns no rows.

Multiple-Row Subqueries

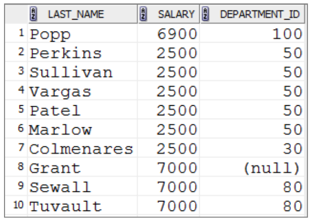
- Return more than one row are called multiple-row subqueries.
- Use multiple-row comparison operators

Operator	Meaning
`IN`	Equal to an member in the list
`ANY`	Must be preceded by =, !=, >, <, <=, >=. Compares a value to each value in a list or returned by a query. Evaluates to `FALSE` if the query returns no rows.
`ALL`	Must be preceded by =, !=, >, <, <=, >=. Compares a value to each value in a list or returned by a query. Evaluates to `TRUE` if the query returns no rows.

Using the IN Operator in Multiple-Row Subqueries

Example:

Find the employee who earn the same salary as the minimum salary for each department.



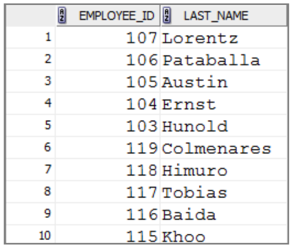
	LAST_NAME	SALARY	DEPARTMENT_ID
1	Popp	6900	100
2	Perkins	2500	50
3	Sullivan	2500	50
4	Vargas	2500	50
5	Patel	2500	50
6	Marlow	2500	50
7	Colmenares	2500	30
8	Grant	7000	(null)
9	Sewall	7000	80
10	Tuvault	7000	80

```
• SELECT  last_name,salary,department_id
• FROM    employees
• WHERE salary IN
•
•
```

Using the IN Operator in Multiple-Row Subqueries

Practice:

Display the employee number and last name of all employees who work in a department with any employee whose last name contains "u."

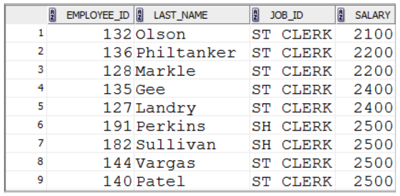


	EMPLOYEE_ID	LAST_NAME
1	107	Lorentz
2	106	Pataballa
3	105	Austin
4	104	Ernst
5	103	Hunold
6	119	Colmenares
7	118	Himuro
8	117	Tobias
9	116	Baida
10	115	Khoo

Using the ANY Operator in Multiple-Row Subqueries

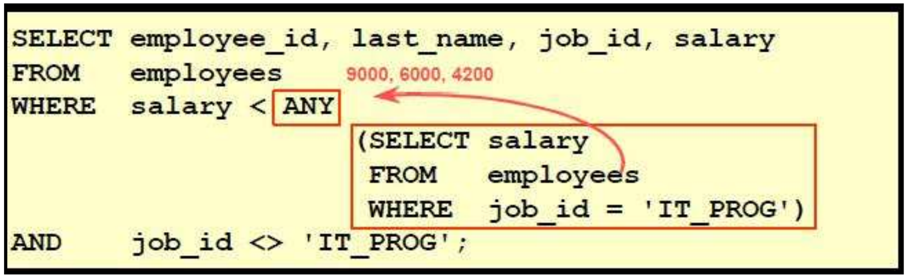
Example:

The slide displays employees who are not IT programmers and whose salary is less than that of any IT programmer.



	EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
1	132	Olson	ST_CLERK	2100
2	136	Philtanker	ST_CLERK	2200
3	128	Markle	ST_CLERK	2200
4	135	Gee	ST_CLERK	2400
5	127	Landry	ST_CLERK	2400
6	191	Perkins	SH_CLERK	2500
7	182	Sullivan	SH_CLERK	2500
8	144	Vargas	ST_CLERK	2500
9	140	Patel	ST_CLERK	2500

Using the ANY Operator in Multiple-Row Subqueries



```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary < ANY
                    (SELECT salary
                     FROM   employees
                     WHERE  job_id = 'IT_PROG')
AND    job_id <> 'IT_PROG';
```

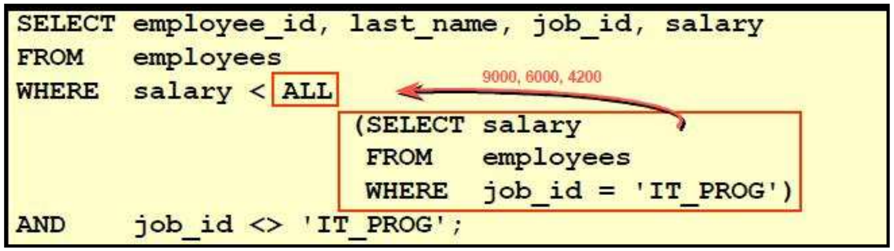
NOTE:

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

Using the ALL Operator in Multiple-Row Subqueries

Example:

The slide displays employees whose salary is less than the salary of all employees with a job ID of IT_PROG and whose job is not IT_PROG.



```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary < ALL
                    (SELECT salary
                     FROM   employees
                     WHERE  job_id = 'IT_PROG')
AND    job_id <> 'IT_PROG';
```

NOTE:

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

Quiz

Using a subquery is equivalent to performing two sequential queries and using the result of the first query as the search value(s) in the second subquery.

- True
- False

Summary

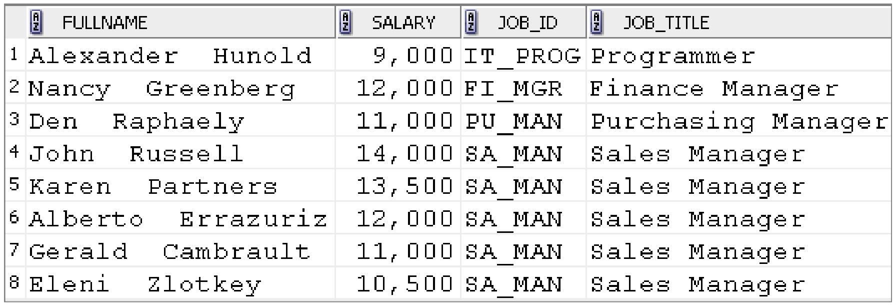
- In this lesson,you should have learned how to:
  - Identify when a subquery can help solve a problem
  - Write subqueries when a query is based on unknown values

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

Practice

จงแสดงข้อมูล ชื่อ-สกุล รหัสงาน เงินเดือน รหัสงาน ชื่องาน โดยแสดงพนักงานที่มีเงินเดือน มากกว่าเงินเดือนเฉลี่ย(ของทั้งหมด) และ job_title ที่ไม่มีตัว t

*** ใช้ sub query ***



	FULLNAME	SALARY	JOB_ID	JOB_TITLE
1	Alexander Hunold	9,000	IT_PROG	Programmer
2	Nancy Greenberg	12,000	FI_MGR	Finance Manager
3	Den Raphaely	11,000	PU_MAN	Purchasing Manager
4	John Russell	14,000	SA_MAN	Sales Manager
5	Karen Partners	13,500	SA_MAN	Sales Manager
6	Alberto Errazuriz	12,000	SA_MAN	Sales Manager
7	Gerald Cambrault	11,000	SA_MAN	Sales Manager
8	Eleni Zlotkey	10,500	SA_MAN	Sales Manager