

# INTRODUCTION TO SQL

## SUBQUERIES



By Kanokwan Atcharyachanvanich

Faculty of Information Technology

KMITL

Database System Concepts

# OBJECTIVES

- After completing this lesson, you should be able to do the following:
- Describe the types of problem that subqueries can solve
- Define subqueries
- List the types of subqueries
- Write single-row and multiple-row subqueries

# SUBQUERY

- Inner query หรือ Nested query
- ดึงข้อมูลใน table จาก ผลลัพธ์ของการทำ SQL Select query ก่อนหน้านี้
- สามารถใช้ **subquery** ในส่วน **From** เพื่อสร้างตารางเสมือนจากผลลัพธ์ SQL select query
- สามารถใช้ **subquery** ใน **Where Clause**
  - ถูกใช้ในเงื่อนไขที่ main query ไม่สามารถดึงข้อมูลได้ตามเงื่อนไขปกติที่สามารถทำได้ หรือ
  - ติดข้อจำกัดภายใต้กฎของ SQL
- Subquery ถูกใช้ภายใต้
  - SQL Select, SQL Insert, SQL Update SQL Delete ที่เป็น statement และ SQL Operator ต่างๆ

# USING A SUBQUERY TO SOLVE A PROBLEM

- Who has a salary greater than Abel's?

**Main Query:**

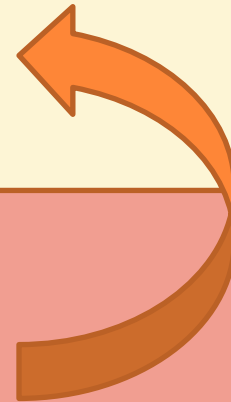


**Which employees have salaries greater than Abel's salary?**

**Subquery**



**What is Abel's salary?**



# SUBQUERY SYNTAX

**SELECT**    **select\_list**

**FROM**      **table**

**WHERE**    **expr operator**

**(SELECT select\_list**

**FROM table) ;**

- The subquery (inner query) executes once before the main query.
- The result of the subquery is used by **the main query** (outer query).

# GUIDELINES FOR USING SUBQUERIES

- **Enclose subqueries in parentheses.**
- **Place subqueries on the right side of the comparison condition.**
- **Operators Usage**
  - **single-row operators (>, =, >=, <, <=, <>) with single-row subqueries**
  - **multiple-row operators (IN, ANY, ALL) with multiple-row subqueries.**

# USING A SUBQUERY

Who has a salary greater than Abel's?

แสดงทุกคนที่มีเงินเดือนมากกว่าเงินเดือนของ Abel

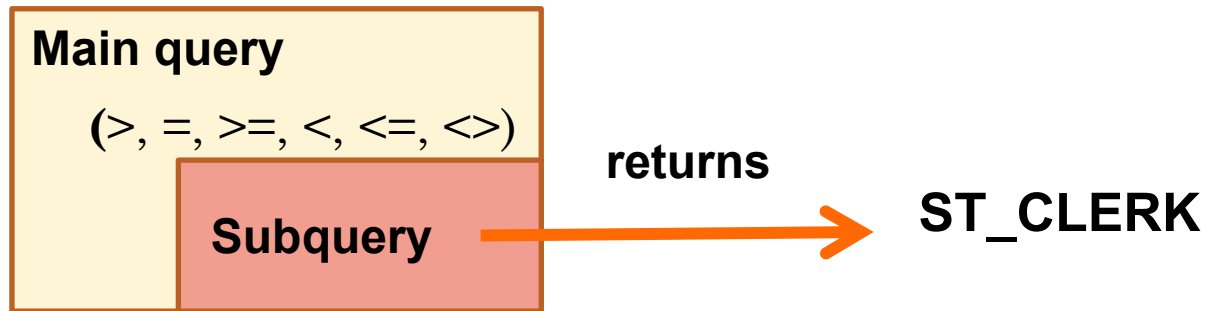
```
SELECT last_name
FROM employees
WHERE salary >
    (SELECT salary
     FROM employees
     WHERE last_name = 'Abel');
```

LAST_NAME
King
Kochhar
De Haan
Greenberg
Russell
Partners
Errazuriz
Ozer
Hartstein
Higgins

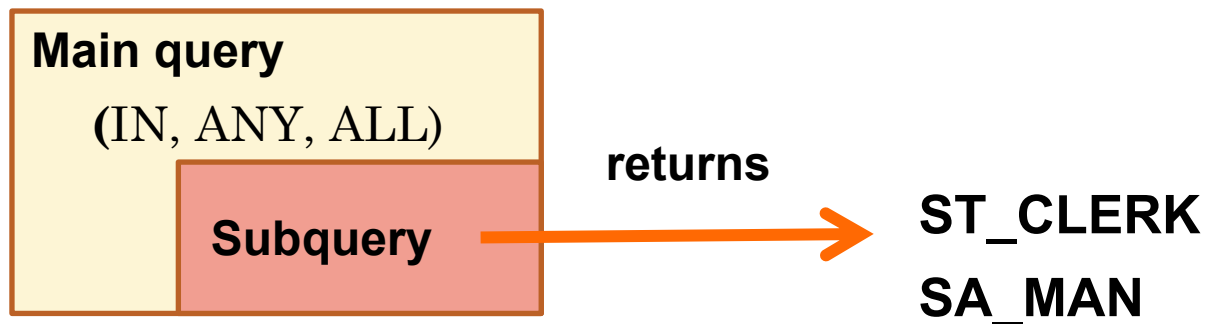
10 rows selected.

# TYPES OF SUBQUERIES

- Single-row subquery



- Multiple-row subquery





# SINGLE-ROW SUBQUERIES

- Return only one row
- 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

# EXECUTING SINGLE-ROW SUBQUERIES

- แสดงนามสกุล รหัสงาน และเงินเดือนของพนักงานทุกคนที่มีรหัสงาน เหมือนกับพนักงานรหัส 141

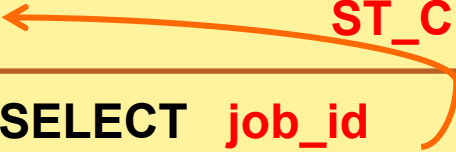

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id = (SELECT job_id
                 FROM employees
                 WHERE employee_id = 141);
```

**ST\_CLERK**

LAST_NAME	JOB_ID	SALARY
Nayer	ST_CLERK	3200
Mikkilineni	ST_CLERK	2700
Bissot	ST_CLERK	3300
Atkinson	ST_CLERK	2800

## EXECUTING SINGLE-ROW SUBQUERIES (CONT.)

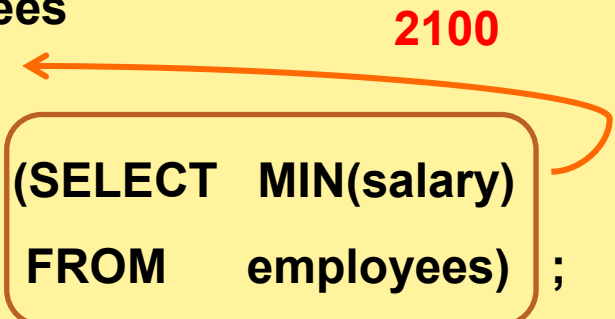
แสดงนามสกุล รหัสงาน และเงินเดือนของพนักงานทุกคนที่มีรหัสงานเหมือนกับ  
พนักงานรหัส 141 และมีเงินเดือนมากกว่าเงินเดือนของพนักงานรหัส 143

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id =  ST_CLERK
AND salary > 
  (SELECT salary
   FROM employees
   WHERE employee_id = 143) ;
```

# USING GROUP FUNCTIONS IN A SUBQUERY

- แสดงนามสกุล รหัสงาน และเงินเดือนของพนักงานทุกคนที่มีเงินเดือนเท่ากับ เงินเดือนต่ำสุด

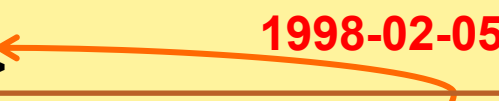
```
SELECT last_name, job_id, salary
FROM employees
WHERE salary = (SELECT MIN(salary)
                FROM employees) ;
```



LAST_NAME	JOB_ID	SALARY
Olson	ST_CLERK	2100

# PRACTICE# 1


- แสดงชื่อ นามสกุล รหัสงานและวันที่เริ่มทำงานของพนักงานที่เริ่มทำงานหลังรหัสพนักงาน 105 เรียงลำดับตามวันที่เริ่มทำงานอดีตไปล่าสุด

```
SELECT first_name, last_name, job_id, hire_date
FROM employees
WHERE hire_date >  1998-02-05
order by hire_date
```

```
(SELECT hire_date
FROM employees
WHERE employee_id = 105 )
```


# THE HAVING CLAUSE WITH SUBQUERIES

- The Oracle server executes subqueries first.
- The Oracle server returns results into the HAVING clause of the main query.
- แสดงรหัสแผนก เงินเดือนต่ำสุดในแผนกเดียวกัน เฉพาะกลุ่มที่มีเงินเดือนต่ำสุดมากกว่าพนักงานในแผนกรหัส 50 ที่ได้เงินเดือนต่ำสุด

```
SELECT    department_id, MIN(salary)
FROM      employees
GROUP BY  department_id
HAVING    MIN(salary) >  2100
          (SELECT MIN(salary)
           FROM      employees
           WHERE     department_id = 50) ;
```

## PRACTICE# 2

- แสดงรหัสแผนก เงินเดือนสูงสุดในแผนกเดียวกัน เฉพาะกลุ่มที่มีเงินเดือนสูงสุดมากกว่าพนักงานในแผนกรหัส 80 ที่ได้เงินเดือนสูงสุด

```
SELECT    department_id, MAX(salary)
FROM      employees
GROUP BY  department_id
HAVING    MAX(salary) >  (SELECT MAX(salary)
                        FROM      employees
                        WHERE    department_id = 80) ;
```

14000

# WHAT IS WRONG WITH THIS STATEMENT?

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

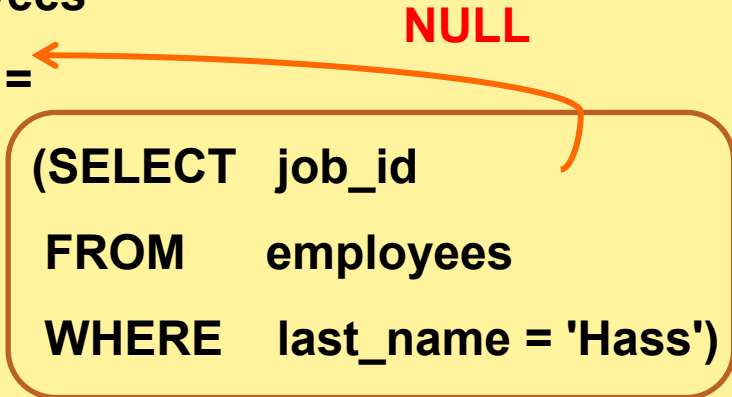
Single-row operator (=) with multiple-row subquery

ERROR at line 4:  
ORA-01427: single-row subquery returns more than one row



# WILL THIS STATEMENT RETURN ROWS?

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




**NULL**

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

# PRACTICE#3

- แสดงนามสกุล รหัสงาน และวันที่รับเข้าทำงานของพนักงานที่อยู่ในแผนกเดียวกันกับแผนกที่ Austin(last\_name) ทำงานอยู่ และไม่ต้องแสดงข้อมูลของ Austin

```
SELECT last_name, job_id, hire_date
FROM employees
WHERE department_id = 60
      (SELECT department_id
       FROM employees
       WHERE last_name = 'Austin')
AND last_name <> 'Austin' ;
```

# MULTIPLE-ROW SUBQUERIES

- Return more than one row
- Use multiple-row comparison operators
- place an =, <>, >, <, <= or >= operator before ANY, ALL in your query.

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

เป็นจริงกับค่าใดค่าหนึ่ง

เป็นจริงกับทุกค่า

## MULTIPLE-ROW SUBQUERIES

- **IN is an alias for = ANY.** Thus, these two statements are the same:

```
SELECT s1 FROM t1 WHERE s1 = ANY (SELECT s1 FROM t2);
```

```
SELECT s1 FROM t1 WHERE s1 IN (SELECT s1 FROM t2);
```

## MULTIPLE-ROW SUBQUERIES

- **NOT IN** is an alias for **<> ALL**. Thus, these two statements are the same:

```
SELECT s1 FROM t1 WHERE s1 <> ALL (SELECT s1 FROM t2);
```

```
SELECT s1 FROM t1 WHERE s1 NOT IN (SELECT s1 FROM t2);
```

## USING THE **IN** OPERATOR IN MULTIPLE-ROW SUBQUERIES

- แสดงพนักงานที่ได้รับเงินเดือนเท่ากับเงินเดือนต่ำสุดของแผนกต่างๆ

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

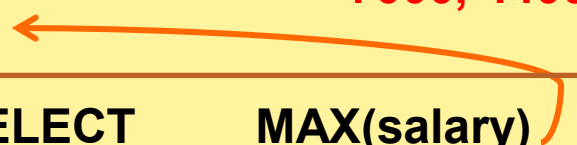
4400, 6000, 2500, .....

## PRACTICE# 4

- แสดงชื่อ นามสกุลพนักงาน เงินเดือน รหัสแผนก ที่ได้รับเงินเดือนเท่ากับเงินเดือนสูงสุดของแผนกใดแผนกหนึ่ง (อนุโลมให้รวมกลุ่มพนักงานที่ไม่มีแผนกด้วย)

```
SELECT first_name, last_name, salary, department_id
FROM employees
WHERE salary IN (SELECT MAX(salary)
                  FROM employees
                  GROUP BY department_id) ;
```

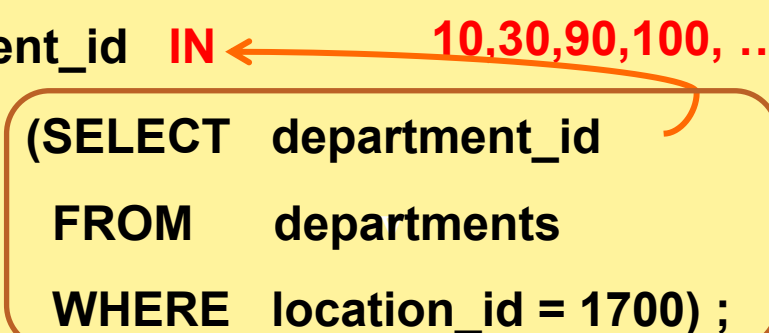
7000, 4400, 13000, .....



## PRACTICE# 5

- แสดงนามสกุล รหัสแผนก และรหัสงานของพนักงานที่แผนกทำงานตั้งอยู่ใน location รหัส 1700

```
SELECT last_name, department_id, job_id
FROM employees
WHERE department_id IN 10,30,90,100, ...
    (SELECT department_id
     FROM departments
     WHERE location_id = 1700) ;
```





## USING THE **ANY** OPERATOR IN MULTIPLE-ROW SUBQUERIES

- แสดงรหัสพนักงาน นามสกุลพนักงาน รหัสงานและเงินเดือน ที่ได้รับเงินเดือนน้อยกว่าเงินเดือนของพนักงานที่มีรหัสงานเป็น IT\_PROG **บางคน** และไม่ต้องแสดงพนักงานที่มีรหัสงาน IT\_PROG

```
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' ;
```

9000, 6000, 4800, 4800, 4200

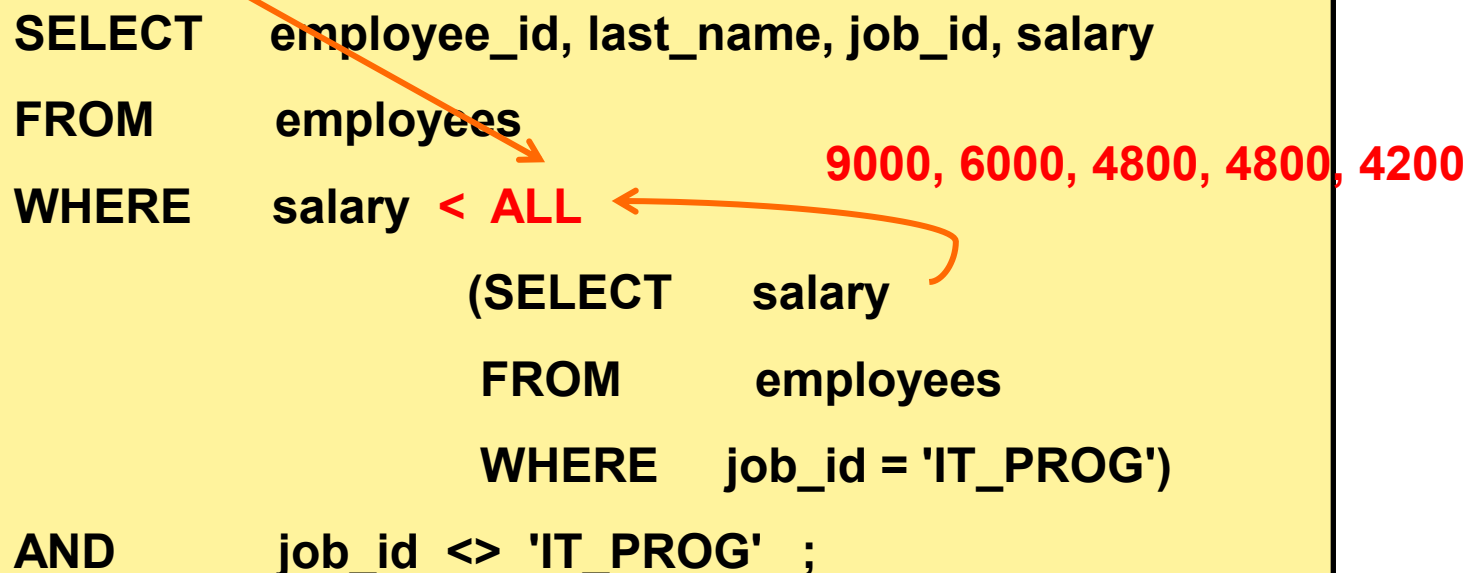
## USING THE **ALL** OPERATOR IN MULTIPLE-ROW SUBQUERIES

- แสดงพนักงานที่ได้รับเงินเดือนน้อยกว่าเงินเดือนของ IT Programmer

ทุกคน และ ที่ไม่ใช่ IT Programmer

```
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' ;
```

9000, 6000, 4800, 4800, 4200



## PRACTICE# 6

- แสดงชื่อ นามสกุลพนักงาน เงินเดือน รหัสแผนก ที่ได้รับเงินเดือนมากกว่าพนักงานทุกคนในแผนก 20

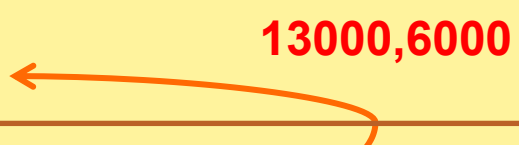
```
SELECT first_name, last_name, salary, department_id
FROM employees
WHERE salary > ALL (SELECT salary
                     FROM employees
                     WHERE department_id=20) ;
```

13000,6000

## PRACTICE# 7

- แสดงชื่อ นามสกุลพนักงาน เงินเดือน รหัสแผนก ที่ได้รับเงินเดือนมากกว่าพนักงานบางคนในแผนก 20 และไม่ต้องแสดงข้อมูลพนักงานในแผนก 20

```
SELECT first_name, last_name, salary, department_id
FROM employees
WHERE salary > ANY
      (SELECT salary
       FROM employees
       WHERE department_id=20)
AND department_id <> 20 ;
```



# NULL VALUES IN A SUBQUERY

- แสดงพนักงานทุกคนที่ไม่มีผู้ใต้บังคับบัญชา (ไม่มีลูกน้อง)/ที่ไม่ใช่ผู้จัดการ

```
SELECT  emp.last_name, emp.employee_id, emp.manager_id
FROM    employees emp
WHERE   emp.employee_id NOT IN (SELECT mgr.manager_id
                                FROM    employees mgr) ;
```

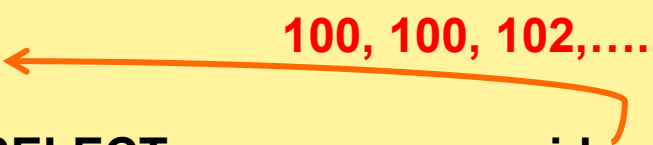
**NULL, 100, 100, 102,.....**

No rows selected.

- All conditions that compare a null value result in a null.
- So whenever null values are likely to be part of the results set of a subquery, do not use the NOT IN operator.
- The NOT IN operator is equivalent to <> ALL.

## USING THE **NOT IN** OPERATOR IN MULTIPLE-ROW SUBQUERIES

- แสดงพนักงานทุกคนที่ไม่มีผู้ใต้บังคับบัญชา (ลูกน้อง) /ที่ไม่ใช่ผู้จัดการ

```
SELECT  emp.last_name, emp.employee_id, emp.manager_id
FROM    employees emp
WHERE   emp.employee_id NOT IN  100, 100, 102,.....
      (SELECT  mgr.manager_id
        FROM    employees mgr
        WHERE   manager_id IS NOT NULL) ;
```

# NESTING GROUP FUNCTIONS USING SUBQUERIES

- Group functions can be nested to a depth of two.
- To display the maximum average salary. แสดงเงินเดือนเฉลี่ยที่สูงสุดจากทุกแผนก โดยคำนวณจากค่าเฉลี่ยเงินเดือนของพนักงานในแผนกเดียวกัน

## ORACLE

```
SELECT      MAX(AVG(salary))  
FROM        employees  
GROUP BY    department_id;
```

## MySQL

```
SELECT      MAX(avs.sal)  
FROM        (select avg(e.salary) sal  
             from employees e  
             group by department_id) avsal ;
```

Column alias

Table alias

MAX(AVG(SALARY))

19333.3333

# SUMMARY

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

```
SELECT    select_list  
FROM      table  
WHERE     expr operator
```

```
(SELECT select_list  
FROM    table) ;
```