# Introduction to SQL

# LAB 12
## Subqueries

By Kanokwan Atchariyachanvanich

Faculty of Information Technology

KMITL

Database System Concepts

2/2563

# OUTLINE

| Date | SQL |
|---|---|
| 8~11 MAR-2021 | • **Quiz 2A-SELECT statement (LAB 8) 1 ชั่วโมง**<br>• ฝึกฝนคำสั่ง SELECT ด้วย ScrambleSQL (Drag-and-drop tool) |
| 15~18 MAR-2021 | • LAB 9 - Displaying Data from Multiple Tables<br>• ฝึกฝนคำสั่ง SELECT ด้วย ScrambleSQL (Drag-and-drop tool) |
| 22~25 MAR-2021 | • **Quiz 2B-SELECT statement (LAB 8) 1 ชั่วโมง**<br>• LAB 10 - Displaying Data from Multiple Tables (outer join) |
| 29 MAR ~ 1 APR-2021 | LAB 11 - Group function |
| 5~8 APR-2021 | LAB 12 - Subqueries<br>LAB 13 - Manipulating Data (DML) |
| 19~22 APR-2021 | Review LAB 9-13 |
| 26~29 APR-2021 | **QUIZ #3: GROUP FUNCTION, SUBQUERIES, DML (LAB 9-13)** |
| 3~6-MAY-2021 | **Project Presentation – Application** |
| 13 MAY-2021 | **FINAL examination (9.30-12.30)** |

# 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 ใน Where Clause
  - ถูกใช้ในเงื่อนไขที่ main query ไม่สามารถดึงข้อมูลได้ตามเงื่อนไขปกติที่สามารถทำได้ หรือ
  - ติดข้อจำกัดภายใต้กฎของ SQL
- สามารถใช้ subquery ในส่วน From เพื่อสร้างตารางเสมือนจากผลลัพธ์ SQL select query
- 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?**

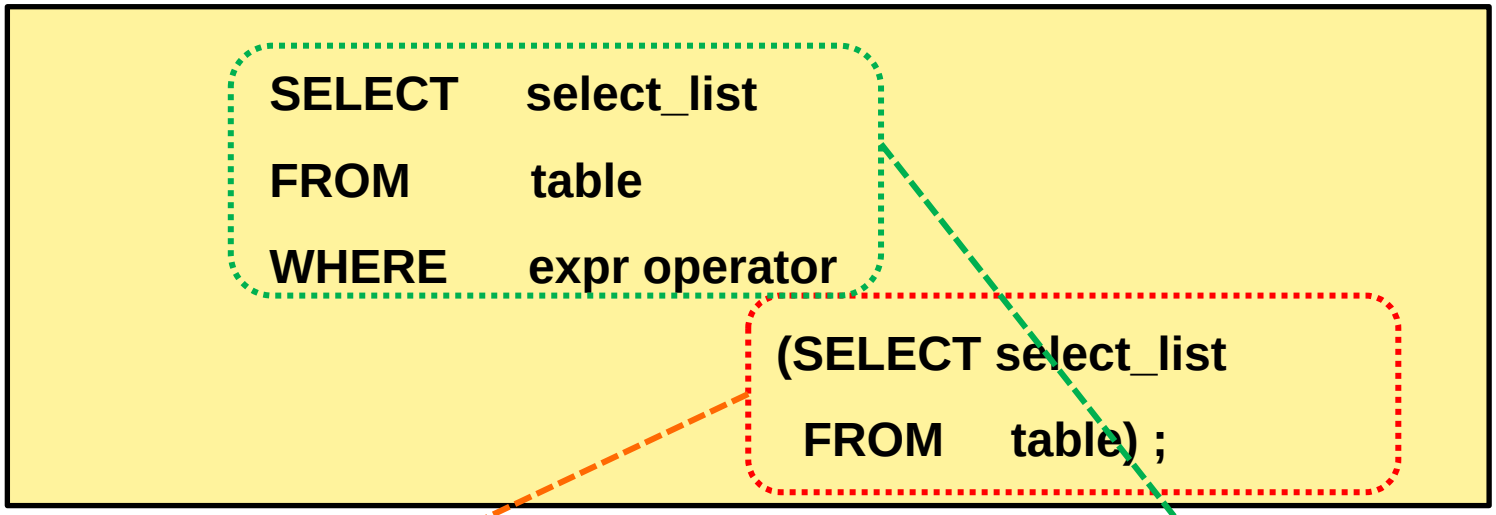
**Main Query:**

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

**Subquery**

**What is Abel's salary?**

# Subquery Syntax




- 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

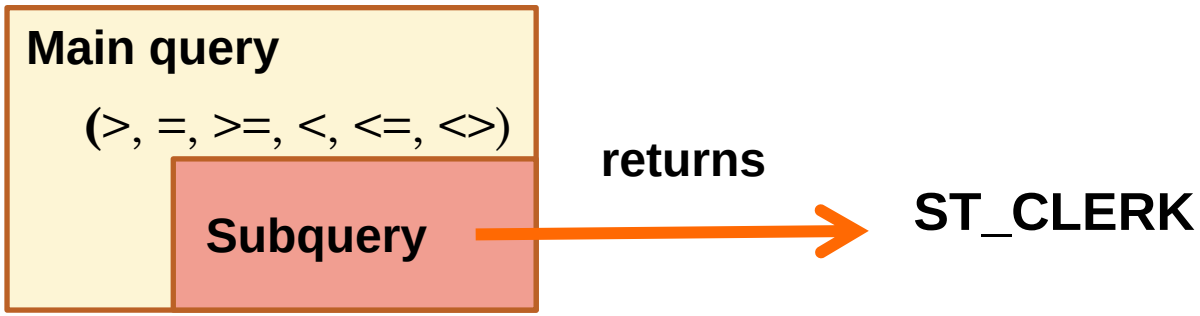
```
SELECT    last_name
FROM      employees
WHERE     salary >
          (SELECT   salary
           FROM     employees
           WHERE    last_name = 'Abel');
```

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

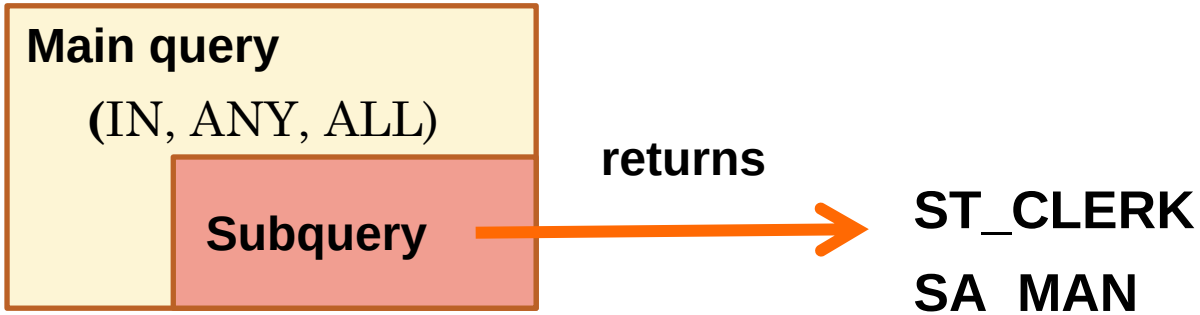
10 rows selected.

# 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 และเงินเดือนของพนักงานทุกคนที่มีรหัสงานเหมือนกับพนักงานรหัส 141 และมีเงินเดือนมากกว่าเงินเดือนของพนักงานรหัส 143

```
SELECT    last_name, job_id, salary
FROM      employees
WHERE     job_id =
                    (SELECT   job_id
                     FROM     employees
                     WHERE    employee_id = 141)
AND       salary >
                    (SELECT   salary
                     FROM     employees
                     WHERE    employee_id = 143) ;
```

ST_CLERK

2600

# Using Group Functions in a Subquery

- แสดงนามสกุล รหัสงาน และเงินเดือนของพนักงานทุกคนที่มีเงินเดือนเท่ากับเงินเดือนต่ำสุด

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

2100

| LAST_NAME | JOB_ID | SALARY |
|---|---|---|
| Olson | ST_CLERK | 2100 |

# PRACTICE# 1

- แสดงชื่อ นามสกุล รหัสงานและวันที่เริ่มทำงานของพนักงานที่เริ่มทำงานหลังรหัสพนักงาน 105 เรียงลำดับตามวันที่เริ่มทำงานอดีตไปล่าสุด

# 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

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

NULL

# PRACTICE#3

- แสดงนามสกุล รหัสงาน และวันที่รับเข้าทำงานของพนักงานที่อยู่ในแผนกเดียวกันกับแผนกที่ Austin(last_name) ทำงานอยู่ และไม่ต้องแสดงข้อมูลของ Austin

# Multiple-Row Subqueries

- Return more than one row
- Use multiple-row comparison 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 |

เป็นจริงกับค่าใดค่าหนึ่ง

เป็นจริงกับทุกค่า

# 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

- แสดงชือ นามสกุลพนักงาน เงินเดือน รหัสแผนก ที่ได้รับเงินเดือนเท่ากับเงินเดือนสูงสุดของแผนกใดแผนกหนึ่ง

# PRACTICE# 5

- แสดงนามสกุล รหัสแผนก และรหัสงานของพนักงานที่แผนกทำงานตั้งอยู่ใน location รหัส 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 และที่ไม่ใช่ 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

- แสดงชือ นามสกุลพนักงาน เงินเดือน รหัสแผนก ที่ได้รับเงินเดือนมากกว่าพนักงานทุกคนในแผนก 20

# PRACTICE# 7

- แสดงชือ นามสกุลพนักงาน เงินเดือน รหัสแผนก ที่ได้รับเงินเดือนมากกว่าพนักงานบางคนในแผนก 20 และไม่ต้องแสดงข้อมูลพนักงานในแผนก 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
                                (SELECT    mgr.manager_id
                                FROM      employees  mgr
                                WHERE     manager_id  IS  NOT   NULL) ;
```

100, 100, 102,…..

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

Column alias

Table alias

| MAX(AVG(SALARY)) |
| --- |
| 19333.3333 |

# Additional Subqueries in DDL

# CREATING A TABLE BY USING A SUBQUERY SYNTAX

- สร้างตารางพร้อมกับเพิ่มข้อมูลลงตาราง จากการใช้คำสั่ง CREATE TABLE statement และ AS subquery

```
CREATE   TABLE     table_name
                   [(column_name, column_name...)]
AS subquery;
```

- จับคู่จำนวนคอลัมน์ของตารางที่ต้องการสร้างกับจำนวนคอลัมน์ใน subquery
- นิยามคอลัมน์ด้วยชื่อคอลัมน์และค่าเบื้องต้น

# CREATING A TABLE BY USING A SUBQUERY

```
CREATE    TABLE    dept80
  AS
    SELECT    employee_id, last_name, salary, hire_date
    FROM        employees
    WHERE      department_id = 80 ;
```

Subquery

**Table created.**

```
DESCRIBE  dept80;
```

# 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) ;
```