

บทปฏิบัติการวิชา 674330 ระบบฐานข้อมูลขั้นสูง
เรื่อง การกำหนดเงื่อนไขในการแสดงข้อมูล โดยใช้คำสั่ง SQL II

วัตถุประสงค์: เมื่อจบบทปฏิบัติการนี้แล้ว นักศึกษาสามารถใช้คำสั่งในการเลือกแสดงข้อมูลตามเงื่อนไขที่ต้องการและใช้ฟังก์ชันและคำสั่งพิเศษเพื่อประกอบการเลือกแสดงข้อมูลได้

ให้ทำการพิมพ์คำสั่งเพื่อแสดงข้อมูลตามเงื่อนไขต่างๆ แล้วศึกษาการทำงานของคำสั่ง

1. การแสดงข้อมูล

การแสดงข้อมูลทั้งหมด

```
SELECT *  
FROM EMPLOYEE
```

การแสดงข้อมูลเฉพาะแอททริบิวต์ที่ต้องการ

```
SELECT EMPNAME, POSITION, SALARY  
FROM EMPLOYEE
```

2. การแสดงข้อมูลแบบเรียงลำดับ

การแสดงข้อมูลแบบเรียงลำดับจากน้อยไปหามาก

```
SELECT EMPNAME, POSITION  
FROM EMPLOYEE  
ORDER BY EMPNAME ASC
```

ASC เป็นการกำหนดให้เรียงลำดับจากน้อยไปหามาก เป็นค่า default ไม่ต้องระบุก็ได้
การแสดงข้อมูลแบบเรียงลำดับจากมากไปหาน้อย

```
SELECT EMPNAME, POSITION  
FROM EMPLOYEE  
ORDER BY EMPNAME DESC
```

DESC เป็นการกำหนดให้เรียงลำดับจากมากไปหาน้อย

3. การใช้คำสัง่ย่อ DISTINCT

```
SELECT DISTINCT PROJNO  
FROM PROJWORK
```

เป็นการแสดงค่าของ PROJNO ถ้าแต่ละแถวมีข้อมูลซ้ำกัน ให้แสดงเพียงแถวเดียว

4. การใช้คำสั่งย่อย TOP n

```
SELECT TOP 3 EMPNAME, SALARY  
FROM EMPLOYEE  
ORDER BY SALARY
```

TOP 3 หมายถึงแสดงข้อมูลเพียง 3 แถวแรกเท่านั้น

5. การใช้คำสั่งย่อย INTO สำหรับการสร้างตารางใหม่

```
SELECT EMPNAME, SALARY, POSITION  
INTO emp1  
FROM EMPLOYEE  
ORDER BY EMPNAME
```

เป็นการสร้างตารางใหม่ชื่อ emp1 ที่ประกอบด้วย EMPNAME, SALARY, POSITION โดยข้อมูลจะเรียงลำดับตาม EMPNAME จากน้อยไปหามาก

```
SELECT *  
FROM emp1
```

6. การใช้คำสั่งย่อย AS

การใช้ AS แทนที่ชื่อแอททริบิวต์

```
SELECT EMPNAME AS 'Employee Name', DEPNO AS 'Department'  
FROM EMPLOYEE  
ORDER BY EMPNAME
```

ในการแสดงข้อมูลที่หัวตารางจะมีชื่อตามที่เรากำหนดหลัง AS

การใช้ AS แทนที่ชื่อรีเลชัน

```
SELECT E1.EMPNAME, E1.POSITION  
FROM EMPLOYEE AS E1  
ORDER BY EMPNAME
```

เราสามารถใช้ชื่อตารางที่กำหนดไว้หลัง AS แทนในคำสั่งได้ จะทำให้การพิมพ์คำสั่งสั้นลง

7. การใช้ WHERE

การใช้กับเครื่องหมายเปรียบเทียบ =, >, >=, <, <=, <> ดังตัวอย่างเช่น

```
SELECT EMPNAME, HIREDATE  
FROM EMPLOYEE  
WHERE POSITION = 'MANAGER'
```

แสดงข้อมูลเฉพาะผู้ที่มีตำแหน่งเป็น MANAGER เท่านั้น

```
SELECT EMPNUM, EMPNAME, POSITION, SALARY
FROM EMPLOYEE
WHERE SALARY >= 25000
```

แสดงข้อมูลเฉพาะผู้ที่เงินเดือนมากกว่าหรือเท่ากับ 25000

```
SELECT EMPNUM, EMPNAME, POSITION, SALARY
FROM EMPLOYEE
WHERE SALARY BETWEEN 15000 AND 25000
```

การใช้กับคำสั่งย่อย IN

```
SELECT EMPNUM, EMPNAME, POSITION
FROM EMPLOYEE
WHERE EMPNUM IN ('1002', '3003', '4001')
```

แสดงข้อมูลเฉพาะพนักงานที่มีรหัส 1002, 3003 และ 4001

การใช้กับคำสั่งย่อย LIKE

```
SELECT EMPNAME, POSITION
FROM EMPLOYEE
WHERE EMPNAME LIKE 'A%'
```

แสดงข้อมูลเฉพาะพนักงานที่มีชื่อขึ้นต้นด้วยตัวอักษร A
สำหรับเครื่องหมาย % ใช้แทนตัวอักษรอะไรก็ได้ กี่ตัวก็ได้

```
SELECT EMPNAME
FROM EMPLOYEE
WHERE EMPNAME LIKE '_____'
```

แสดงข้อมูลเฉพาะพนักงานที่มีชื่อ 6 ตัวอักษร
สำหรับเครื่องหมาย _ ใช้แทนตัวอักษรอะไรก็ได้ จำนวน 1 ตัว

```
SELECT EMPNAME
FROM EMPLOYEE
WHERE EMPNAME LIKE '%T%'
```

แสดงข้อมูลของพนักงานทุกคนที่มีตัวอักษร T อยู่ในชื่อ

การใช้กับคำสั่งย่อย NOT

```
SELECT EMPNAME
FROM EMPLOYEE
WHERE EMPNAME NOT LIKE '%A%'
```

แสดงข้อมูลพนักงานทุกคนที่ไม่มีตัวอักษร A อยู่ในชื่อ

การใช้กับคำสั่งย่อย AND

```
SELECT EMPNAME, SALARY, HIREDATE  
FROM EMPLOYEE  
WHERE SALARY > 20000 AND HIREDATE >= '01/01/1994'
```

แสดงข้อมูลพนักงานที่มีเงินเดือนมากกว่า 20000 และมีวันเริ่มทำงานตั้งแต่ 01/01/1994

การใช้กับคำสั่งย่อย OR

```
SELECT EMPNAME, SALARY, HIREDATE  
FROM EMPLOYEE  
WHERE SALARY > 20000 OR HIREDATE >= '01/01/1994'
```

แสดงข้อมูลพนักงานที่มีเงินเดือนมากกว่า 20000 หรือมีวันเริ่มทำงานตั้งแต่ 01/01/1994

8. การใช้ฟังก์ชัน

AVG → การหาค่าเฉลี่ย

```
SELECT AVG (SALARY) AS 'Average of salary'  
FROM EMPLOYEE
```

COUNT → การนับจำนวนแถว

```
SELECT COUNT (SALARY) AS 'Employee as salary > 20000'  
FROM EMPLOYEE  
WHERE SALARY > 20000
```

SUM → หาค่าผลรวม

```
SELECT SUM (SALARY) AS 'SUM of salary'  
FROM EMPLOYEE
```

MIN → หาค่าน้อยที่สุด

```
SELECT MIN (SALARY) AS 'MIN of salary'  
FROM EMPLOYEE
```

MAX → หาค่ามากที่สุด

```
SELECT MAX (SALARY) AS 'MAX of salary'  
FROM EMPLOYEE
```

9. การใช้ฟังก์ชัน GROUP BY และ HAVING

ตัวอย่างการใช้ GROUP BY เพื่อแสดงข้อมูลตามกลุ่ม

```
SELECT DEPNO, SUM (SALARY) AS 'SUM of salary'
FROM EMPLOYEE
GROUP BY DEPNO
```

```
SELECT DEPNO, MAX (SALARY) AS 'MAX of salary', MIN (SALARY) AS 'MIN of salary'
FROM EMPLOYEE
GROUP BY DEPNO
```

การใช้ HAVING เพื่อแสดงข้อมูลตามเงื่อนไข ซึ่ง HAVING จะต้องใช้คู่กับ GROUP BY เสมอ

```
SELECT POSITION, AVG (SALARY) AS 'Average of salary'
FROM EMPLOYEE
GROUP BY POSITION
HAVING AVG (SALARY) > 25000
```

เป็นการแสดงตำแหน่งเฉพาะตำแหน่งที่มีค่าเฉลี่ยของเงินเดือนสูงกว่า 25000

10. การ JOINS

INNER JOINS → ต้องมีทั้งสองตารางจึงนำมาแสดง

```
SELECT E.EMPNAME, D.DEPNAME, E.SALARY
FROM EMPLOYEE AS E, DEP AS D
WHERE E.DEPNO = D.DEPNO
```

LEFT OUTER JOINS → แสดงข้อมูลของตารางทางซ้ายทั้งหมด

```
SELECT E.EMPNAME, D.DEPNAME
FROM EMPLOYEE AS E LEFT OUTER JOIN DEP AS D ON E.DEPNO = D.DEPNO
```

```
SELECT E.EMPNAME, D.DEPNAME
FROM DEP D LEFT OUTER JOIN EMPLOYEE E ON E.DEPNO = D.DEPNO
```

RIGHT OUTER JOINS → แสดงข้อมูลของตารางทางขวาทั้งหมด

```
SELECT E.EMPNAME, D.DEPNAME
FROM EMPLOYEE E RIGHT OUTER JOIN DEP D ON E.DEPNO = D.DEPNO
```

```
SELECT E.EMPNAME, D.DEPNAME
FROM DEP D RIGHT OUTER JOIN EMPLOYEE E ON E.DEPNO = D.DEPNO
```

FULL OUTER JOINS → นำข้อมูลทั้งหมดจากทั้งสองตารางมาแสดง

```
SELECT E.EMPNAME, D.DEPNAME  
FROM EMPLOYEE E FULL OUTER JOIN DEP D ON E.DEPNO = D.DEPNO
```

```
SELECT E.EMPNAME, D.DEPNAME  
FROM DEP D FULL OUTER JOIN EMPLOYEE E ON E.DEPNO = D.DEPNO
```

SELF JOINS

```
SELECT E1.EMPNUM, E1.EMPNAME, E1.POSITION, E2.EMPNAME AS Manager  
FROM EMPLOYEE AS E1, EMPLOYEE AS E2  
WHERE E1.MGRNO = E2.EMPNUM
```

11. การ UNION

```
SELECT EMPNAME, SALARY  
FROM EMPLOYEE  
WHERE SALARY > 30000  
UNION  
SELECT EMPNAME, SALARY  
FROM EMPLOYEE  
WHERE SALARY < 10000
```

12. การใช้ Sub queries ในคำสั่ง SELECT

```
SELECT EMPNAME, SALARY, POSITION  
FROM EMPLOYEE  
WHERE DEPNO = (SELECT DEPHO  
FROM DEP  
WHERE DEPHAME='ACCOUNTING')
```

เป็นการแสดงข้อมูลเฉพาะที่มีชื่อหน่วยงานเป็น ACCOUNTING

```
SELECT EMPNAME, SALARY, POSITION  
FROM EMPLOYEE  
WHERE DEPNO = (SELECT DEPHO  
FROM DEP  
WHERE DEPHAME='ACCOUNTING')
```

เป็นการแสดงข้อมูลพนักงานเฉพาะที่มีเงินเดือนสูงกว่าเงินเดือนเฉลี่ย และเรียงลำดับเงินเดือนจากมากไปหาน้อย

IN → ใช้ในกรณีที่มีค่าต้องการเปรียบเทียบว่าต้องเหมือนค่าใดค่าหนึ่งในหลาย ๆ ค่า

```
SELECT EMPNAME, SALARY, POSITION  
FROM EMPLOYEE  
WHERE DEPNO IN (SELECT DEPHO  
FROM DEP  
WHERE DEPHAME='ACCOUNTING' OR  
DEPHAME='FINANCE')
```

เป็นการแสดงข้อมูลพนักงานเฉพาะที่ทำงานใน ACCOUNTING หรือ FINANCE

NOT IN → ใช้ในกรณีที่ต้องการเปรียบเทียบว่าต้องไม่เหมือนค่าใดค่าหนึ่งในหลาย ๆ ค่า

```
SELECT EMPNAME, SALARY, POSITION
FROM EMPLOYEE
WHERE DEPNO NOT IN (SELECT DEPTO
                     FROM DEPT
                     WHERE DEPTNAME='ACCOUNTING' OR
                     DEPTNAME='FINANCE')
```

เป็นการแสดงข้อมูลพนักงานเฉพาะที่ไม่ทำงานใน *ACCOUNTING* หรือ *FINANCE*

ANY → ใช้ในการเปรียบเทียบกับหลายค่า ถ้าเป็นจริงตามค่าใดค่าหนึ่งก็นำมาแสดง

```
SELECT EMPNAME, SALARY, POSITION, DEPTO
FROM EMPLOYEE
WHERE DEPTO > ANY (SELECT DEPTO FROM DEPT
                   WHERE DEPTNAME='ACCOUNTING' OR DEPTNAME='FINANCE')
```

ALL → ใช้ในการเปรียบเทียบกับหลายค่า เมื่อเปรียบเทียบแล้วต้องเป็นจริงทุกค่าจึงนำมา

แสดง

```
SELECT EMPNAME, SALARY, POSITION, DEPTO
FROM EMPLOYEE
WHERE DEPTO >= ALL (SELECT DEPTO FROM DEPT
                    WHERE DEPTNAME='ACCOUNTING' OR DEPTNAME='FINANCE')
```

EXISTS → เปรียบเทียบค่าว่ามีอยู่ใน sub queries หรือไม่ ถ้ามีจึงจะนำมาแสดง

```
SELECT EMPNUM, EMPNAME, SALARY, POSITION
FROM EMPLOYEE E1
WHERE EXISTS (SELECT EMPNUM
              FROM EMPLOYEE E2
              WHERE E2.MGRNO = E1.EMPNUM)
```

NOT EXISTS → เปรียบเทียบค่าว่าจะต้องไม่ปรากฏใน sub queries จึงจะนำมาแสดง

```
SELECT DEPTO, DEPTNAME
FROM DEPT D
WHERE NOT EXISTS (SELECT DEPTO
                  FROM EMPLOYEE E
                  WHERE E.DEPTO = D.DEPTO)
```

ลองทำดู 1

ข้อมูลที่แสดง

ชื่อพนักงาน เงินเดือน ตำแหน่ง และชื่อแผนก ของพนักงาน

เงื่อนไข

ให้แสดงข้อมูลพนักงานที่มีเงินเดือนมากกว่าค่าเฉลี่ย และพนักงานจะต้องมีแผนกอยู่ที่ sukumvit

.....

.....

.....

.....

ลองทำดู 2

ข้อมูลที่แสดง

ชื่อพนักงาน เงินเดือน ตำแหน่ง และรหัส โครงการที่พนักงานรับผิดชอบ

เงื่อนไข

ให้แสดงข้อมูลพนักงานที่รับผิดชอบในโครงการที่มีชั่วโมงการทำงานมากที่สุด 3 ลำดับแรก

.....

.....

.....

.....

ลองทำดู 3

ข้อมูลที่แสดง

ชื่อพนักงาน ตำแหน่ง เงินเดือน ชื่อแผนก และที่อยู่ของแผนก

เงื่อนไข

ให้แสดงข้อมูลในกรณีที่พนักงานมีเงินเดือนมากกว่าค่าเฉลี่ยของตำแหน่ง Clerk และไม่ได้อยู่แผนก marketing

.....

.....

.....

.....

ลองทำดู 4

ข้อมูลที่แสดง

ชื่อแผนก และสถานที่

เงื่อนไข

ให้แสดงข้อมูลแผนกที่ไม่มีพนักงานอยู่เลย

.....

.....

.....

.....