

Structured Query Language (SQL)

ภาษา SQL ถูกพัฒนาขึ้นมาโดยบริษัท IBM เมื่อประมาณปี 1970 และถูกจัดให้เป็นภาษามาตรฐานเพื่อใช้กับ Relational Database โดย American National Standards Institute (ANSI) เมื่อปี 1986 และ International Organization for Standardization (ISO) เมื่อปี 1987 Version ของ SQL ที่ถูกยอมรับคือ 1989 และ 1992 สำหรับ Version 1992 อาจเรียกว่า SQL-92 หรือ ANSI-92 SQL ปี 1999 SQL:1999 (หรืออาจเรียกว่า SQL3) จะมีแนวคิดของ object-oriented ปี 2008 SQL:2008 ได้เพิ่มคุณสมบัติต่างๆ เข้ามาที่สำคัญคือสนับสนุน Extensible Markup Language (XML)

ปัจจุบัน Relational Database เป็นรูปแบบฐานข้อมูลที่ถูกใช้กันอย่างแพร่หลาย ทำให้ผู้รู้ภาษา SQL สามารถเข้าถึงฐานข้อมูลโดยมิต้องพึ่ง programmer ดังตัวอย่างผู้สอบบัญชีสามารถค้นหาข้อมูลของบริษัทที่ถูกตรวจสอบตามที่ตนต้องการ เจ้าหน้าที่ฝ่ายการตลาดสามารถดูข้อมูลลูกค้าเพื่อวางแผนทางการตลาด เจ้าหน้าที่ฝ่ายบุคคลสามารถทำรายงานสรุปตามที่ผู้บริหารต้องการ ซึ่งทั้งหมดนี้เกิดขึ้นด้วยการเขียนคำสั่งของผู้ใช้เอง

ภาษา SQL จัดเป็นภาษา Very High Level Language ซึ่งหมายถึงเป็นภาษาที่ใกล้เคียงกับภาษามนุษย์มาก เพราะฉะนั้นการเรียนรู้ภาษาดังกล่าวทำได้ค่อนข้างง่าย ประกอบกับภาษาดังกล่าวมีความสามารถสูงหมายถึงสามารถค้นหาข้อมูลได้ตามที่ผู้ใช้ต้องการทำให้ภาษา SQL เป็นที่ยอมรับในปัจจุบัน และเนื่องจากเป็นภาษามาตรฐาน ทำให้ผู้รู้ไม่จำเป็นต้องไปทำงานองค์กรใด ไม่ว่าองค์กรนั้นใช้ Database ของบริษัทอะไร ก็สามารถใช้ภาษาดังกล่าวได้

เนื่องจาก SQL ไม่ใช่ภาษาที่สมบูรณ์ในตัวเอง มักจะต้องใช้ร่วมกับภาษาอื่นจึงเรียกภาษา SQL ว่า data sublanguage ภาษา SQL จะมีหน้าที่ในการสร้างและประมวลข้อมูล เราสามารถใช้ภาษา SQL โดยสั่งตรงกับ DBMS หรือเป็นส่วนหนึ่งของ application program

ภาษา SQL แบ่งเป็น

- Data Definition Language (DDL) เป็นภาษาเพื่อใช้สำหรับสร้างตาราง สร้างความสัมพันธ์ระหว่างตาราง และคำสั่งเกี่ยวกับโครงสร้าง เช่น CREATE DATABASE, CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, DROP INDEX
- Data Manipulation Language (DML) เป็นคำสั่งเพื่อใช้ในการค้นหา (query) เพิ่ม (insert) แก้ไข (modify) และ ลบ (delete) ข้อมูล เช่น SELECT, UPDATE, DELETE, INSERT INTO

การสืบค้นข้อมูล (Query)

การสืบค้นข้อมูล หรือการ Query ข้อมูลจะถูกใช้อย่างมาก และมีความสำคัญในระบบฐานข้อมูลเพราะจุดประสงค์หลักที่เราใช้ฐานข้อมูลก็เพื่อให้ระบบฐานข้อมูลสืบค้นข้อมูลให้เราได้ ในเอกสารฉบับนี้จะกล่าวถึงคำสั่ง SQL ในส่วนของการ query ข้อมูลโดยใช้ table EMP ในหน้าถัดไป

แนวคิดที่สำคัญของการสืบค้นข้อมูล

ผู้ใช้งานต้องเข้าใจว่าการเก็บข้อมูลโดยใช้ Relational Database ข้อมูลจะอยู่ในรูปแบบตารางเสมอไม่ว่าจะเก็บข้อมูลเรื่องอะไร เช่น พนักงาน สินค้า ขาย ข้อมูลจะอยู่ในรูปแบบตาราง และเมื่ออยู่ในรูปแบบตารางการสืบค้นของผู้ใช้จะแบ่งออกเป็นสามลักษณะหลักคือ

1. **แสดงบาง row** เช่น หาพนักงานที่อยู่ Bangkok หาพนักงานที่มีเงินเดือนมากกว่า 20,000 ผลลัพธ์จะออกบาง row ไม่ใช่ทั้งตาราง
2. **ประมวลผล** เช่น ต้องการทราบว่าบริษัทต้องจ่ายเงินเดือนรวมแล้วเดือนละเท่าไร ต้องการหาค่าเฉลี่ยเงินเดือน
3. **จัดกลุ่มข้อมูล** เช่น ต้องจ่ายเงินเดือนแต่ละฝ่ายเป็นเงินเท่าไร (กลุ่มก็คือฝ่าย) หรือ พนักงานที่มาจากแต่ละจังหวัดมีกี่คน (กลุ่มก็คือจังหวัด)

EMP					
ID	Name	City	Deptno	Sal	Boss
101	Tom	Bangkok	Acc	30,000.	
102	Tim	Bangkok	Fin	24,000.	
103	Tony	Chiangmai	Acc	25,000.	101
104	Sam	Bangkok	Mkt	27,000.	
105	Philips	Chiangrai	Fin	18,000.	102
106	Linda	Chiangmai	Acc	24,000.	101
107	Maurice	Bangkok	Mkt	18,000.	104
108	Scott	Chiangrai	Acc	22,000.	103
109	Betty	Bangkok	Fin	15,000.	102
110	Cooker	Bangkok		20,000.	
111	Ben	Chiangmai	Mkt	10,000.	107
112	Bell	Bangkok	Acc	9,000.	103
113	Rose	Chiangmai		8,000.	
114	Frank	Bangkok	Acc	11,000.	106
115	Richard	Bangkok	Mkt	8,000.	104
116	Nick	Bangkok	Fin	9,500.	102
117	Hover	Chiangmai	Mkt	7,500.	107
118	Coco	Chiangmai	Acc	20,000.	106
119	Samson	Bangkok	Acc	12,000.	106
120	Westin	Bangkok	Fin	8,500.	102

รูปแบบคำสั่ง SQL

SELECT	<i>column(s)</i>
FROM	<i>table(s)</i>
WHERE	<i>criteria</i>
GROUP BY	<i>column(s)</i>
HAVING	<i>criteria</i>
ORDER BY	<i>column(s)</i>

SELECT	<i>column(s)</i>	column ที่ต้องการให้แสดง
FROM	<i>table(s)</i>	ตารางข้อมูลเพื่อใช้ค้นหาข้อมูล
WHERE	<i>criteria</i>	เงื่อนไข rows ที่ต้องการให้แสดง (option)
GROUP BY	<i>column(s)</i>	จัดกลุ่มตาม column (option)
HAVING	<i>criteria</i>	เงื่อนไขผลลัพธ์ที่ต้องการให้แสดงหลังจากการจัดกลุ่ม (option)
ORDER BY	<i>column(s)</i>	เรียงลำดับผลลัพธ์ (option)

* option หมายถึงคำสั่งดังกล่าวจะใช้หรือไม่ก็ได้

- การสืบค้นข้อมูล – คำสั่งพื้นฐาน

1. SELECT *
FROM emp;

คำสั่งสั้นที่สุดของ SQL ต้องมี SELECT และ FROM โดยคำสั่ง SELECT เพื่อเลือก column ส่วน FROM เพื่อบอกว่ามาจาก table ไหน
เครื่องหมาย * คือทุก column
2. SELECT id, name, city, deptno, sal, sal * 12
FROM emp;

เราอาจให้มีการคำนวณในส่วน of คำสั่ง select ได้
3. SELECT id AS empid, name, city, deptno, sal, sal
* 12 AS yearly_salary
FROM emp;

เราสามารถตั้ง Alias ให้ column ที่ต้องการได้ โดยใช้ Alias
4. SELECT DISTINCT city
FROM emp;

เมื่อเราต้องการให้แสดงข้อมูลค่าที่ไม่ซ้ำ ออกมาเช่น พนักงานมาจาก city ไหนบ้าง เรา
จะใช้ DISTINCT ที่คำสั่ง SELECT

5. SELECT DISTINCT city, deptno
FROM emp;

การใช้ distinct ดังกล่าวหมายถึง ทั้ง city และ deptno ต้องไม่ซ้ำ

• การสืบค้นข้อมูล – แสดงบาง row

การสืบค้นข้อมูลโดยแสดงบาง row หรือการสืบค้นข้อมูลโดยมีเงื่อนไข จะต้องสั่งไว้ที่ WHERE โดย operators ที่สามารถใช้ได้จะมีดังนี้

Operators	คำอธิบาย
=	เท่ากับ
<>	ไม่เท่ากับ (DBMS บางตัวอาจใช้เครื่องหมาย !=)
>	มากกว่า
<	น้อยกว่า
>=	มากกว่าเท่ากับ
<=	น้อยกว่าเท่ากับ
IN	ระบุค่าที่เป็นไปได้หลายค่า
BETWEEN	ค่าระหว่าง
LIKE	ค้นหารูปแบบ
IS NULL	การค้นหาข้อมูลที่มีค่าเป็นค่าว่าง (NULL)

6. SELECT id, name, city, deptno, sal
FROM emp
WHERE city = 'Bangkok';
7. SELECT id, name, city, deptno, sal
FROM emp
WHERE sal > 15000;
8. SELECT id, name, city, deptno, sal
FROM emp
WHERE city IN ('Bangkok', 'Chiangmai');
9. SELECT id, name, city, deptno, sal
FROM emp
WHERE sal IN (8000, 9000, 10000);
10. SELECT id, name, city, deptno, sal
FROM emp
WHERE sal between 10000 and 18000;

การค้นหาข้อมูลที่เป็นตัวอักษรต้องใส่เครื่องหมาย single quote (')

การค้นหาข้อมูลที่เป็นตัวเลขไม่ต้องใส่เครื่องหมาย single quote และห้ามใส่ comma เช่นห้ามใส่ 8,000

ค่า between ทางฐานข้อมูลจะแสดงค่า 10000 และ 18000 ด้วย ซึ่งทางคณิตศาสตร์ค่าระหว่างจะไม่รวมค่า 10000 และ 18000

11. SELECT id, name, city, deptno, sal
FROM emp
WHERE name LIKE 'T%';

แสดงพนักงานที่ชื่อขึ้นต้นด้วยตัวอักษร t
คำสั่งดังกล่าวต้องใช้ LIKE 't%' ไม่สามารถใช้ =
't%'

12. SELECT id, name, city, deptno, sal
FROM emp
WHERE name LIKE '%t';

แสดงพนักงานที่ชื่อลงท้ายด้วยตัวอักษร t

13. SELECT id, name, city, deptno, sal
FROM emp
WHERE name LIKE '%t__';

แสดงพนักงานที่ชื่อตัวอักษรที่ 3 จากท้าย
เป็นตัวอักษร (ใช้ _ สองครั้ง)

14. SELECT id, name, city, deptno, sal
FROM emp
WHERE deptno IS NULL;

แสดงพนักงานที่ไม่มีฝ่ายอยู่ นั้นหมายถึง
deptno เป็นค่าว่างซึ่งระบบฐานข้อมูลจะ
เรียกว่าค่า null และต้องสั่ง IS NULL ไม่
สามารถสั่งว่า = NULL

การค้นหาโดยมีมากกว่าหนึ่งเงื่อนไข

ในกรณีเงื่อนไขมีมากกว่าหนึ่ง เราจะ **ต้อง** เชื่อมระหว่างเงื่อนไขด้วย operators **AND** หรือ **OR** เสมอ โดย AND หมายถึงเงื่อนไขทั้งคู่ต้องเป็นจริง ในขณะที่ **OR** หมายถึงเงื่อนไขใดเงื่อนไขหนึ่งต้องเป็นจริง

15. SELECT id, name, city, deptno, sal
FROM emp
WHERE deptno = 'Acc' AND sal >= 12000; แสดงผู้ที่อยู่ฝ่าย ACC และมีเงินเดือนตั้งแต่ 12,000 บาท
16. SELECT id, name, city, deptno, sal
FROM emp
WHERE deptno = 'Acc' OR sal >=12000; แสดงผู้ที่อยู่ฝ่าย ACC หรือมีเงินเดือนตั้งแต่ 12,000 บาท

ลำดับการทำงาน (Rules of Precedence)

การสั่ง **AND, OR** อยู่ในคำสั่ง **WHERE** เดียวกันจะต้องคำนึงถึงลำดับการทำงานด้วย โดย SQL จะมีลำดับการทำงานดังนี้

ลำดับการทำงาน	Operator
1	เครื่องหมายเปรียบเทียบ (Comparison operators)
2	NOT
3	AND
4	OR

```
SELECT id, name, city, deptno, sal
FROM emp
WHERE deptno = 'Acc' OR sal >=12000 AND city = 'Bangkok';
```

ลำดับการทำงานจะมีดังนี้

- SQL จะหาพนักงานที่มีเงินเดือน มากกว่าเท่ากับ 12000 และ ต้องอยู่กรุงเทพ
- หรือใครก็ได้ที่อยู่ฝ่าย ACC

- การสืบค้นข้อมูล – ประมวลผล

เราสามารถนำ Functions ในคำสั่ง SQL ได้ โดย functions: SUM, AVG, MAX, MIN, COUNT เพื่อทำการประมวลผล หาผลรวม ค่าเฉลี่ย สูงสุด ต่ำสุด และนับจำนวน row ตามลำดับ

17. SELECT SUM(sal), AVG(sal), MAX(sal),
MIN(sal), COUNT(sal)
FROM emp;

การใช้ function ต้องมีเฉพาะ column ที่
เกี่ยวข้องกับการประมวลผลเท่านั้นหากสั่ง
SELECT id, SUM(sal), AVG(sal)
จะไม่สามารถประมวลผลได้

18. SELECT SUM(sal), AVG(sal), MAX(sal),
MIN(sal), COUNT(sal)
FROM emp
WHERE deptno = 'Acc';

ในการใช้ function สามารถที่จะให้ประมวลผล
เพียงบาง row ได้โดยใช้ WHERE เข้าร่วม เช่น
จากโจทย์ข้อ 16. ให้ประมวลผลเฉพาะ
พนักงานฝ่ายบัญชี

การใช้ Functions กับค่า NULL

CUSTOMERS			
CID	CNAME	CITY	TOTAL
101	White	Paris	1000
102	Brown		2000
103	Green	Rome	3000
104	Rose		4000
105	Cherry	London	

* กำหนดให้ CID เก็บเป็นตัวอักษร TOTAL เก็บเป็นตัวเลข

ค่า NULL จะถูกละเลยใน ทุก functions เสมือนว่าไม่มี row นั้นๆ อยู่ในตาราง หากเราใช้คำสั่ง SQL เพื่อทำการประมวลผลโดยใช้ functions จะได้ผลลัพธ์ดังต่อไปนี้

	CID	CNAME	CITY	TOTAL
COUNT	5	5	3	4
MAX	105	White	Rome	4000
MIN	101	Brown	London	1000
AVERAGE	n/a	n/a	n/a	2500
SUM	n/a	n/a	n/a	10000

- การสืบค้นข้อมูล – จัดกลุ่มข้อมูล

การสืบค้นชนิดจัดกลุ่มข้อมูล จะมีการใช้คำสั่งอยู่สองคำสั่งคือ Group by และ Having คำสั่ง Group by เป็นคำสั่งที่ใช้จัดกลุ่มข้อมูล ส่วน Having คือเงื่อนไขที่ไว้คัดผลลัพธ์หลังจากจัดกลุ่มโดย Group by เพราะฉะนั้นหากใช้ Having ต้องใช้คู่กับ Group by เสมอ

Where และ Having เหมือนกันคือทั้งคู่ต่างเป็นเงื่อนไขซึ่ง Where คือเงื่อนไขเพื่อใช้คัด row จากคำสั่ง FROM เข้ามาเพื่อประมวลผล ในขณะที่ Having คือเงื่อนไขที่คัดผลลัพธ์หลังจากทำคำสั่ง Group by เพื่อแสดงผล เพราะฉะนั้น Where จะทำก่อนส่วน Having จะทำหลัง สรุปลำดับการทำงานได้ดังนี้

FROM → WHERE → GROUP BY → HAVING

ตัวอย่าง 19 – 23 เป็นการแสดงการทำงานของ WHERE, GROUP BY, และ HAVING

19. SELECT deptno
FROM emp;

คำสั่งดังกล่าวแสดง deptno จาก table EMP
จำนวนทั้งสิ้น 20 rows

20. SELECT deptno
FROM emp
GROUP BY deptno;

จากผลลัพธ์ 20 rows จากข้อ 19 จะถูกจัดกลุ่ม
เป็นสี่กลุ่มโดยคำสั่ง group by คือ Acc, Fin, MKT,
และ ค่า null

21. SELECT deptno, SUM(sal)
FROM emp
GROUP BY deptno;

โดยปกติ group by มักจะไม่ใช้เดี่ยวๆ อย่างข้อ
19 แต่มักจะใช้คู่กับ function เช่น
หาว่าต้องจ่ายเงินเดือนแต่ละฝ่ายเป็นเงิน
เท่าใด
หากเรา GROUP BY อะไรเราสามารถ SELECT
อย่างนั้น ดังตัวอย่าง GROUP BY deptno เรา
สามารถ SELECT deptno

22. SELECT deptno, SUM (sal)
FROM emp
WHERE sal >= 12000
GROUP BY deptno;

หาว่าต้องจ่ายเงินเดือนแต่ละฝ่ายเป็นเงิน
เท่าไร โดยให้นำเฉพาะพนักงานที่มี
เงินเดือนตั้งแต่ 12000 บาทมาคำนวณหา
เงินเดือนที่ต้องจ่ายสำหรับแต่ละฝ่าย

23. SELECT deptno, SUM (sal)
FROM emp
WHERE sal >= 12000
GROUP BY deptno
HAVING SUM(sal) >= 50000;

หาว่าต้องจ่ายเงินเดือนแต่ละฝ่ายเป็นเงิน
เท่าไร โดยให้นำเฉพาะพนักงานที่มี
เงินเดือนตั้งแต่ 12000 บาทมาคำนวณหา
เงินเดือนรวม และแสดงผลลัพธ์เฉพาะฝ่าย
ที่ ต้องจ่ายเงินเดือนสูงกว่า 50,000 บาท

ความสำคัญของการสืบค้นข้อมูล – จัดกลุ่มข้อมูล

การสืบค้นข้อมูล – จัดกลุ่มข้อมูลมีความสำคัญมากต่อการใช้งานฐานข้อมูล อาจกล่าวได้เลยว่าผู้ใช้ข้อมูลของฐานข้อมูล
ทุกประเภท (บัญชี สินค้าคงคลัง โรงแรม สถาบันการศึกษา) ต้องการให้ระบบฐานข้อมูลทำการจัดกลุ่มข้อมูลให้ และทุก
column สามารถนำมาจัดกลุ่มได้ โดยตัวอย่างต่อไปนี้จะแสดงให้เห็นถึงข้อมูลที่ได้รับจากการจัดกลุ่มข้อมูลจากตาราง
EMP

EMP					
ID	Name	City	Deptno	Sal	Boss
101	Tom	Bangkok	Acc	30,000.	
102	Tim	Bangkok	Fin	24,000.	
103	Tony	Chiangmai	Acc	25,000.	101
104	Sam	Bangkok	Mkt	27,000.	
105	Philips	Chiangrai	Fin	18,000.	102
106	Linda	Chiangmai	Acc	24,000.	101
107	Maurice	Bangkok	Mkt	18,000.	104
108	Scott	Chiangrai	Acc	22,000.	103
109	Betty	Bangkok	Fin	15,000.	102
110	Cooker	Bangkok		20,000.	
111	Ben	Chiangmai	Mkt	10,000.	107
112	Bell	Bangkok	Acc	9,000.	103
113	Rose	Chiangmai		8,000.	
114	Frank	Bangkok	Acc	11,000.	106
115	Richard	Bangkok	Mkt	8,000.	104
116	Nick	Bangkok	Fin	9,500.	102
117	Hover	Chiangmai	Mkt	7,500.	107
118	Coco	Chiangmai	Acc	20,000.	106
119	Samson	Bangkok	Acc	12,000.	106
120	Westin	Bangkok	Fin	8,500.	102

24. SELECT deptno, COUNT(id), SUM(sal)
FROM emp
GROUP BY deptno;

แต่ละฝ่ายมีกี่คน ต้องจ่ายเงินเดือนเท่าไร

25. SELECT city, COUNT(id), SUM(sal)
FROM emp
GROUP BY city;

มีพนักงานมาจากแต่ละเมืองกี่คน และต้อง
จ่ายเงินเดือนเท่าไร

26. SELECT boss, COUNT(id), SUM(sal)
FROM emp
GROUP BY boss;

หัวหน้าแต่ละคน มีลูกน้องกี่คน และต้องจ่าย
เงินเดือนลูกน้องเท่าไร

27. SELECT sal, COUNT(id)
FROM emp
GROUP BY sal;

เงินเดือนแต่ละค่า มีผู้ครองจำนวนกี่คน

28. SELECT id, SUM(sal)
FROM emp
GROUP BY id;

คำสั่งนี้คือสั่งให้จัดกลุ่มตาม id และแต่ละ
กลุ่มของ id ให้รวมเงินเดือน

เนื่องจากแต่ละ id มีเพียงคนเดียว เมื่อสั่งให้

SUM(sal) ก็จะหาผลรวมของคนคนเดียว

ผลลัพธ์ที่ได้จากคำสั่งดังกล่าวจะเท่ากับ

SELECT id, sal

FROM emp;

29. SELECT name, COUNT(id)
FROM emp
GROUP BY name;

คำสั่งดังกล่าว จะให้ผลลัพธ์ว่าพนักงานใน
บริษัทมีชื่ออะไรกันบ้าง ชื่อละกี่คน เนื่องจาก
ข้อมูลใน table EMP ไม่มีชื่อซ้ำผลลัพธ์ก็ออก
ทุกชื่อ ชื่อละหนึ่งคน

การเรียงลำดับผลลัพธ์

ผลลัพธ์จากการสืบค้นข้อมูลทั้ง 3 ประเภทข้างต้นสามารถให้มีการเรียงลำดับผลลัพธ์โดยใช้คำสั่ง ORDER BY ดังต่อไปนี้

30.

```
SELECT *  
FROM emp  
ORDER BY sal;
```
31.

```
SELECT *  
FROM emp  
ORDER BY deptno DESC, sal;
```
32.

```
SELECT *  
FROM emp  
ORDER BY deptno DESC, sal DESC;
```
33.

```
SELECT TOP 10 *  
FROM emp  
ORDER BY sal;
```
31.

```
SELECT TOP 20 PERCENT id, name, deptno  
FROM emp;
```

Oracle จะไม่สามารถใช้คำสั่ง TOP ได้ หากต้องการผลลัพธ์เหมือนกับการใช้คำสั่ง TOP ผู้ใช้จะต้องเข้าไปในเงื่อนไขของ ROW NUMBER ซึ่งอยู่ในหน้าถัดไป

ROW NUMBER in ORACLE

Oracle ได้กำหนดหมายเลข Row ให้กับแต่ละ row ในแต่ละ table โดยเราสามารถเรียกดูหมายเลขดังกล่าวได้จาก column ROWNUM

```
SELECT rownum, id, name, sal
FROM emp;
```

แสดงการใช้ column ROWNUM

```
SELECT rownum, id, name, sal
FROM emp
WHERE rownum < 6;
```

เราสามารถให้ column ROWNUM มาแสดงผลลัพธ์

Row แรกสุดได้เช่นเดียวกับคำสั่ง TOP

ตัวอย่างให้แสดง 5 rows แรก ของ table emp

```
SELECT rownum, id, name, sal
FROM emp
WHERE rownum > 5 AND
      rownum < 11;
```

อย่างไรก็ดีการใช้ ROWNUM ก็มีปัญหา เช่นเราต้องการให้แสดง row ระหว่างหมายเลข 6 ถึง 10 ดังตัวอย่างคำสั่งด้านซ้ายจะไม่มีทางได้ผลลัพธ์เลย

```
SELECT r, id, name, sal
FROM (
      SELECT rownum r, id, name, sal
      FROM emp
    )
WHERE r > 5 AND
      r < 11 ;
```

การแก้ปัญหาดังกล่าวสามารถแก้ได้ด้วย Nested Select (อาจเรียก inner select)

จากคำสั่งดังกล่าวเราจะได้ row ระหว่างหมายเลข 6 ถึง 10

ข้อสังเกต จะต้องใช้ Alias เข้ามาด้วย

```
SELECT rownum, id, name, sal
FROM emp
WHERE rownum < 6
ORDER BY sal ;
```

หมายเลข rownum เป็นหมายเลขที่กำหนดโดย oracle

ให้กับ row หนึ่งและเป็นค่าคงที่ตลอดเวลา

เพราะฉะนั้นจะสร้างปัญหาให้กับเราได้เช่นในกรณีเรา

ต้องการให้แสดง row ห้าลำดับแรก โดยให้เรียงตาม

เงินเดือน คำสั่งด้านซ้ายมือจะได้ผลลัพธ์ไม่ถูกต้อง

```
SELECT r, id, name, sal
FROM (
      SELECT id, name, sal, row_number()
      OVER (ORDER BY sal) r
      FROM emp
    )
WHERE r BETWEEN 1 AND 5;
```

การแก้ปัญหาดังกล่าวจะต้องใช้ function

ROW_NUMBER() ร่วมกับ Nested Select