

การเลือกดูข้อมูลแบบกำหนดเงื่อนไข

ในการเลือกดูข้อมูลจากฐานข้อมูล เราสามารถกำหนดเงื่อนไขเพื่อเลือกเฉพาะ row ที่เราสนใจ โดยการเพิ่ม WHERE clause ซึ่งเป็นส่วนหนึ่งในคำสั่ง SELECT

รูปแบบคำสั่งการเลือกดูข้อมูลแบบกำหนดเงื่อนไข

```
SELECT [* | col1, col2, col3, ..., coln]  
FROM table-name  
[WHERE conditions]
```

โดยที่ conditions คือเงื่อนไขในส่วนของ WHERE clause จะมีรูปแบบให้เลือกใช้ดังต่อไปนี้

- เงื่อนไขที่ใช้ในการเปรียบเทียบ

- เงื่อนไขที่ใช้ BETWEEN operator
- เงื่อนไขที่ใช้ IN operator
- เงื่อนไขที่ใช้ LIKE operator
- เงื่อนไขที่ใช้ IS NULL operator
- การรวมหลายเงื่อนไขเข้าด้วยกัน

เงื่อนไขที่ใช้ในการเปรียบเทียบ

เงื่อนไขที่ใช้ในการเปรียบเทียบ เป็นเงื่อนไขพื้นฐานทั่วไป เช่น การเปรียบเทียบมากกว่า น้อยกว่า เท่ากับ หรือ ไม่เท่ากับ ตารางต่อไปนี้จะแสดงตัวอย่างและความหมายของการใช้เงื่อนไขเปรียบเทียบ ตารางแสดงเงื่อนไขในการเปรียบเทียบ

สัญลักษณ์	ตัวอย่างการใช้	ความหมาย
=	WHERE name = 'SOMSAK'	จะเลือกเฉพาะข้อมูลที่มีค่าใน column NAME เท่ากับ SOMSAK
=	WHERE salary = 25000	จะเลือกเฉพาะข้อมูลที่มีค่าใน column SALARY เท่ากับ 25000 (สังเกตว่าไม่ต้องใช้เครื่องหมาย '' สำหรับชนิดข้อมูลแบบตัวเลข)
<	WHERE salary > 25000	จะเลือกเฉพาะข้อมูลที่มีค่าใน column SALARY มากกว่า 25000
<=	WHERE salary >= 25000	จะเลือกเฉพาะข้อมูลที่มีค่าใน column SALARY มากกว่าหรือเท่ากับ 25000
>	WHERE salary < 25000	จะเลือกเฉพาะข้อมูลที่มีค่าใน column SALARY น้อยกว่า 25000
>=	WHERE salary <= 25000	จะเลือกเฉพาะข้อมูลที่มีค่าใน column SALARY น้อยกว่าหรือเท่ากับ 25000
<> หรือ != หรือ ^=	WHERE salary <> 25000 หรือ WHERE salary != 25000 หรือ WHERE salary ^= 25000	จะเลือกเฉพาะข้อมูลที่มีค่าใน column SALARY ไม่เท่ากับ 25000 (สังเกตว่ามีสัญลักษณ์ที่แทน ไม่เท่ากับ ได้สามแบบ จะเลือกแบบไหนก็ได้ เพราะว่าจะได้ผลลัพธ์เหมือนกัน)

ปฏิบัติการที่ 6.8 การเลือกดูข้อมูลโดยกำหนดเงื่อนไข เท่ากับ

- เมื่อได้หน้าจอสำหรับเขียน SQL ให้ผู้อ่านเขียนคำสั่ง ดังรูปภาพต่อไปนี้ จากนั้นให้คลิกปุ่ม Run เพื่อสั่งให้คำสั่งดังกล่าวทำงาน

0.52268708 seco

Enter SQL Statement:

```
SELECT department_id, first_name, salary
FROM employees
where department_id = 90;
```

ตัวอย่างนี้แสดงการเลือกดูข้อมูล โดยกำหนดเงื่อนไขเลือกเฉพาะ row ที่มีค่า column ชื่อ department_id เท่ากับ 90

DEPARTMENT_ID	FIRST_NAME	SALARY
90	Steven	24000
90	Neena	17000
90	Lex	17000

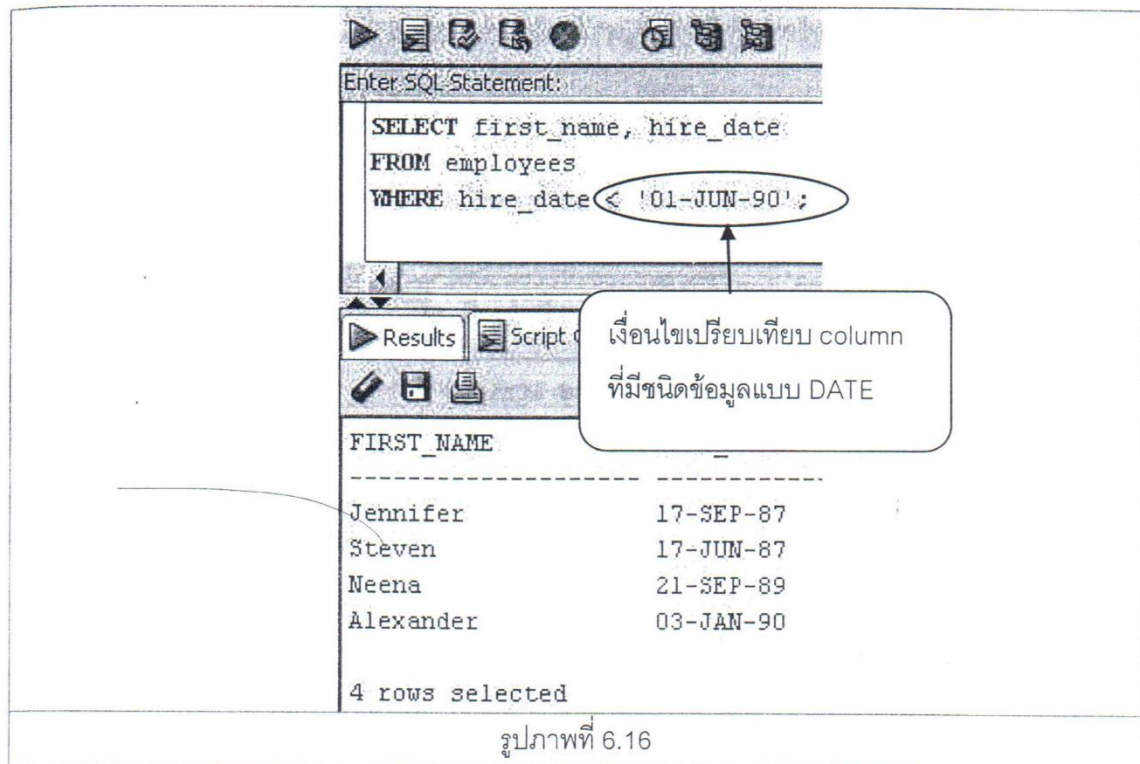
3 rows selected

รูปภาพที่ 6.15

ปฏิบัติการที่ 6.9 การเลือกดูข้อมูลโดยกำหนดเงื่อนไขเปรียบเทียบ column ที่มีชนิดข้อมูลแบบ DATE

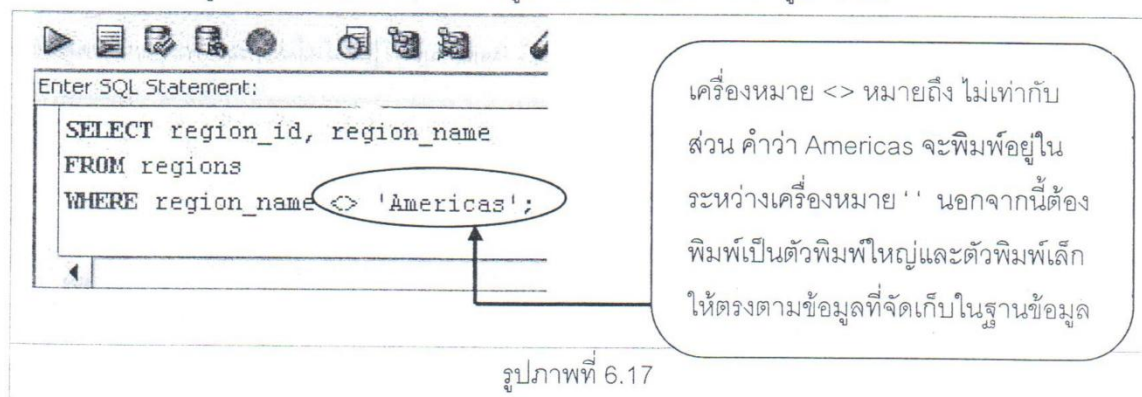
ให้ผู้อ่านเขียนคำสั่ง SQL ดังแสดงในรูปภาพต่อไปนี้ จากนั้นสั่งให้คำสั่งดังกล่าวทำงาน สังเกตการเปรียบเทียบ hire_date < '01-JUN-90' ซึ่งมีความหมายว่าจะเลือกเฉพาะ row ของพนักงานที่มีวันที่เริ่มทำงาน (hire_date) ก่อนวันที่ 1 เดือน June ปี 1990

รูปแบบวันที่ที่ใช้ในการเปรียบเทียบคือ 'DD-MON-YY' โดยที่ DD คือวันที่ MON คืออักษรสามตัวแรกของชื่อเดือน และ YY คือปี ค.ศ. รูปแบบดังกล่าวนี้เป็นรูปแบบ โดย default ของชนิดข้อมูลแบบ DATE ในกรณีที่ต้องการรูปแบบของวันที่ที่ต่างไปจากนี้จะต้องใช้ฟังก์ชันช่วยในการจัดรูปแบบ



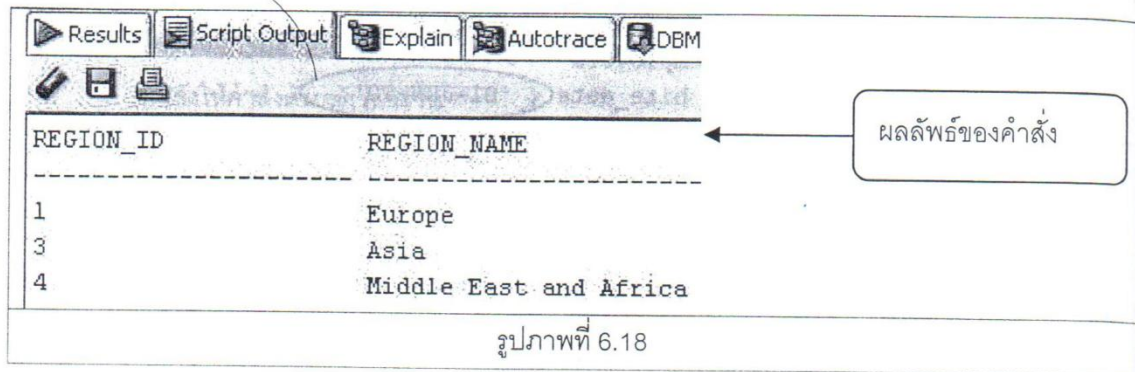
ปฏิบัติการที่ 6.10 การเลือกดูข้อมูลโดยกำหนดเงื่อนไขเปรียบเทียบ column ที่มีชนิดข้อมูลแบบตัวอักษร

ในการเปรียบเทียบเงื่อนไข column ที่มีชนิดข้อมูลแบบตัวอักษร จะกำหนดค่าที่ต้องการจะมาเปรียบเทียบให้อยู่ระหว่างเครื่องหมาย '' ให้ผู้อ่านทดลองพิมพ์คำสั่งดังรูปต่อไปนี้



หน้าจอข้างต้นนี้แสดงการเลือกดูข้อมูลจาก table ชื่อ regions โดยการกำหนดเงื่อนไขเลือกเฉพาะข้อมูลของ row ที่มีค่า region_name ไม่เท่ากับ Americas สังเกตว่าในการใช้ column ที่มีชนิด

ข้อมูลเป็นตัวอักษร หรือ ชนิดข้อมูลที่เป็นวันที่ เราต้องใช้เครื่องหมาย ' ' ครอบค่าของข้อมูลที่เรานำมาเปรียบเทียบ



REGION_ID	REGION_NAME
1	Europe
3	Asia
4	Middle East and Africa

รูปภาพที่ 6.18

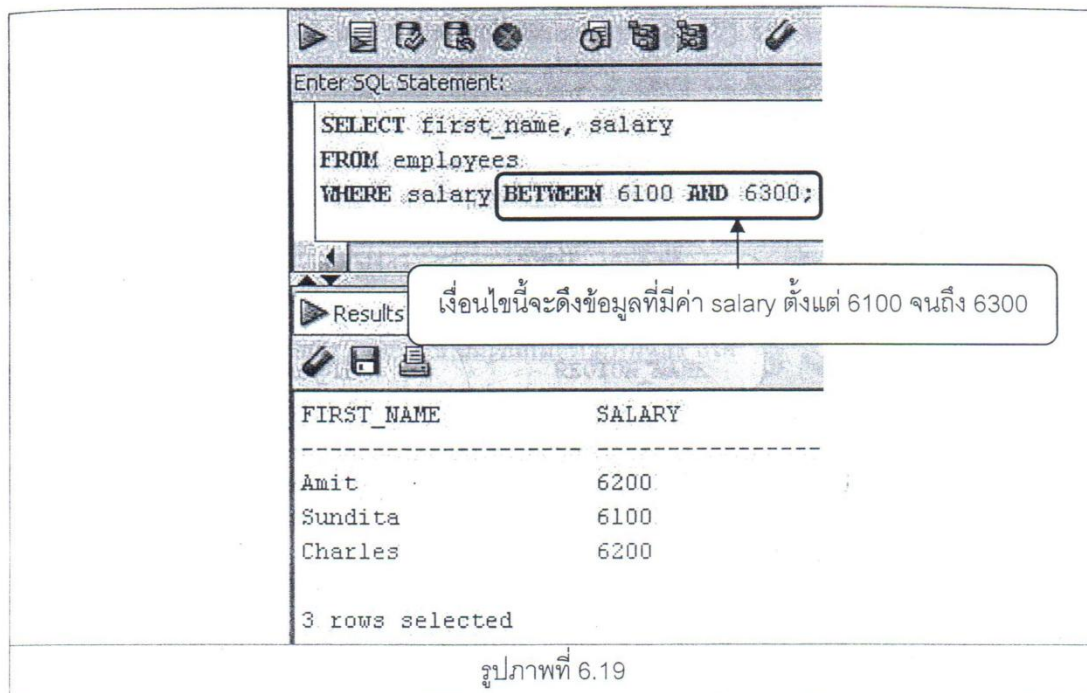
โดยปกติทุกส่วนของคำสั่ง SQL จะพิมพ์เป็นตัวพิมพ์เล็ก ตัวพิมพ์ใหญ่ หรือ ผสมกันระหว่างตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก จะไม่มีความแตกต่างกัน (case insensitive) ดังที่ได้กล่าวมาแล้ว แต่สำหรับค่าของข้อมูลที่เรานำมาเปรียบเทียบที่พิมพ์อยู่ในเครื่องหมาย ' ' ต้องพิมพ์เป็นตัวพิมพ์ใหญ่ ตัวพิมพ์เล็ก หรือ ผสม ขึ้นอยู่กับลักษณะของข้อมูลที่บันทึกในฐานข้อมูล

เงื่อนไขที่ใช้ BETWEEN AND operator

การใช้ BETWEEN...AND operator จะช่วยในการเปรียบเทียบค่าสำหรับเงื่อนไขที่เปรียบเทียบเป็นช่วง มีความกะทัดรัดมากขึ้น ให้ผู้อ่านทดลองเขียนคำสั่งตามสองปฏิบัติการต่อไปนี้ เพื่อทำความเข้าใจการทำงานของ BETWEEN...AND operator

ปฏิบัติการที่ 6.11 การเลือกดูข้อมูลโดยกำหนดเงื่อนไข BETWEEN...AND

- ให้ผู้อ่านทดลองพิมพ์คำสั่งดังรูปภาพที่ 6.19 ต่อไปนี้ จากนั้นให้คลิกปุ่ม Run Script เพื่อดูผลลัพธ์ที่เกิดขึ้น
- รูปภาพ ที่ 6.19 ต่อไปนี้แสดงการใช้เงื่อนไข BETWEEN...AND ในการเลือกดูข้อมูลพนักงานที่มีเงินเดือนตั้งแต่ 6100 จนถึง 6300 สังเกตว่า การใช้เงื่อนไข BETWEEN...AND นั้นจะดึงข้อมูลที่มีค่าใน column ชื่อ salary อยู่ระหว่าง 6100 และ 6300 รวมทั้งจะดึงข้อมูลที่มีค่าใน column ชื่อ salary เท่ากับ 6100 (ถ้ามี) และ ดึงข้อมูลที่มีค่าใน column ชื่อ salary เท่ากับ 6300 (ถ้ามี)



ปฏิบัติการที่ 6.12 การเลือกดูข้อมูลโดยกำหนดเงื่อนไข BETWEEN...AND ร่วมกับ คีย์เวิร์ด NOT

การใช้คีย์เวิร์ด NOT ร่วมกับ BETWEEN...AND จะเป็นเงื่อนไขเลือกข้อมูลที่อยู่นอกช่วงที่ระบุให้ผู้อ่านทดลองพิมพ์คำสั่งดังรูปภาพที่ 6.20 ต่อไปนี้ จากนั้นให้คลิกปุ่ม Run Script เพื่อดูผลลัพธ์ที่เกิดขึ้น สังเกตว่า เงื่อนไข WHERE salary NOT BETWEEN 2200 AND 14000 หมายถึงการเลือกดูข้อมูลพนักงานที่เงินเดือนไม่ได้อยู่ในช่วงตั้งแต่ 2200 จนถึง 14000

Enter SQL Statement:

```
SELECT first_name, salary
FROM employees
WHERE salary NOT BETWEEN 2200 AND 14000;
```

เงื่อนไขนี้จะแสดงข้อมูลพนักงานที่มีเงินเดือนน้อยกว่า 2200 หรือ ข้อมูลพนักงานที่มีเงินเดือนมากกว่า 14000

FIRST_NAME	SALARY
Steven	24000
Neena	17000
Lex	17000
TJ	2100

4 rows selected

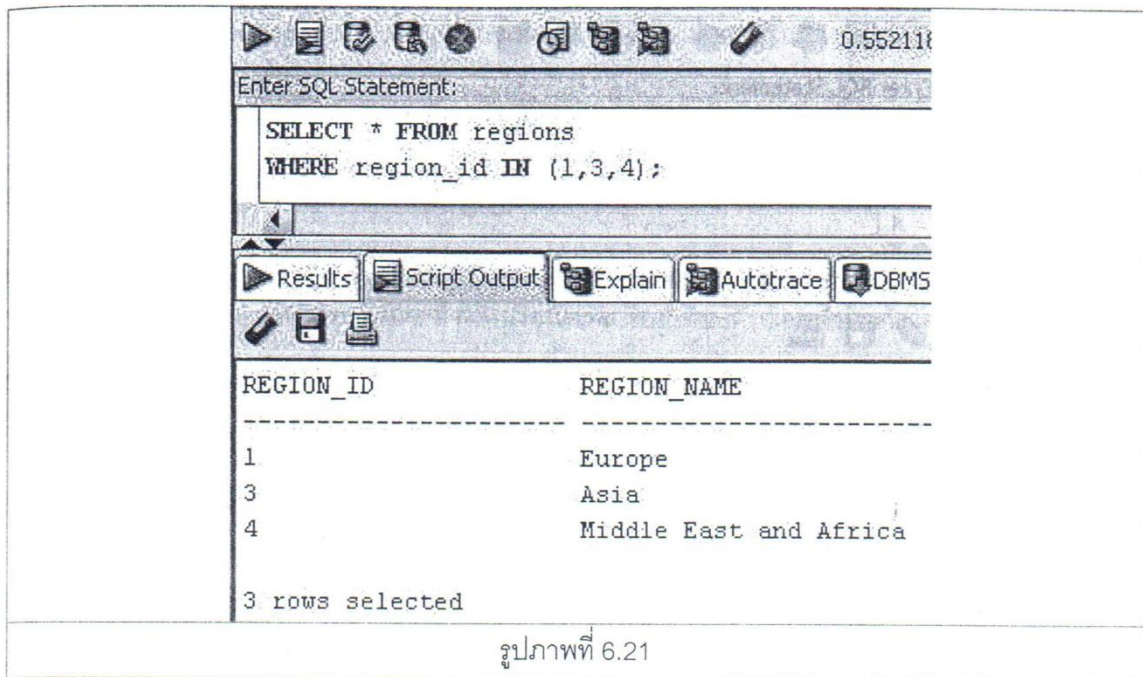
รูปภาพที่ 6.20

เงื่อนไขที่ใช้ IN operator

การใช้ IN operator เหมาะสำหรับเงื่อนไขที่มีการเปรียบเทียบเท่ากัน หรือ การเปรียบเทียบไม่เท่ากัน โดยต้องการเปรียบเทียบทีละหลายๆค่า ปฏิบัติการถัดไปนี้จะแสดงรายละเอียดการใช้เงื่อนไขที่เรียกว่า IN operator

ปฏิบัติการที่ 6.13 การเลือกดูข้อมูลโดยกำหนดเงื่อนไข IN กับ column ที่มีชนิดข้อมูลแบบตัวเลข

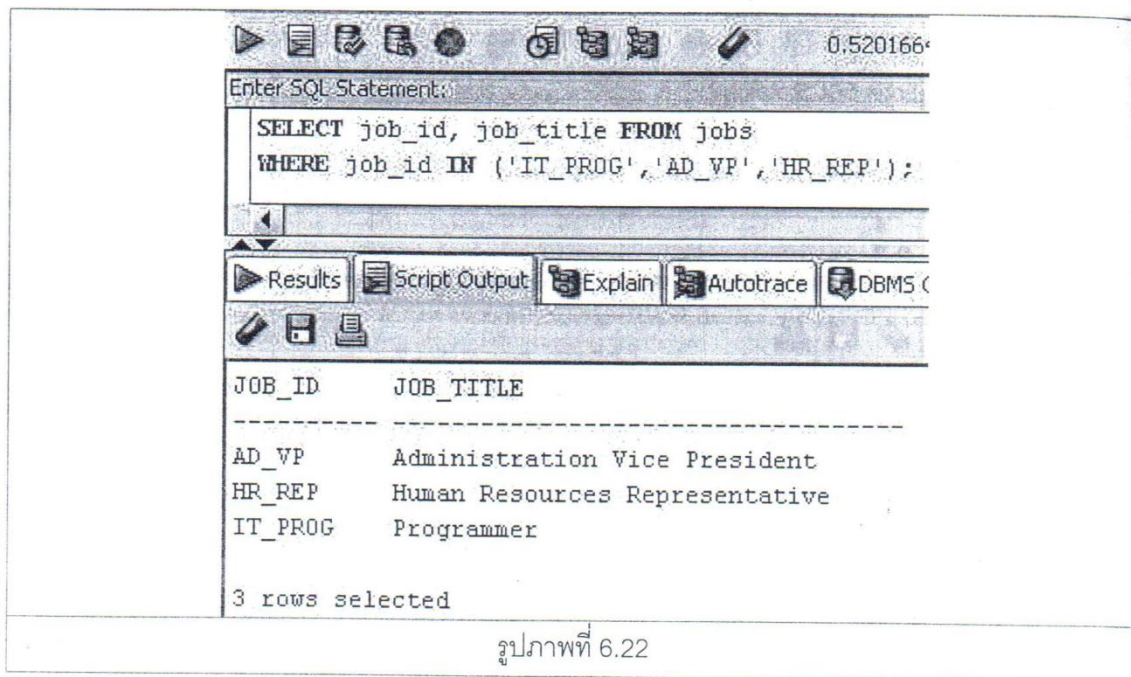
- ให้ผู้อ่านทดลองพิมพ์คำสั่งดังรูปภาพ 6.21 ต่อไปนี้ จากนั้นให้คลิกปุ่ม Run Script เพื่อดูผลลัพธ์ที่เกิดขึ้น
- ผลลัพธ์ของคำสั่งในรูปภาพที่ 6.21 ต่อไปนี้จะแสดงการใช้เงื่อนไข WHERE region_id IN (1,3,4) ซึ่งจะเป็นการเลือกดูข้อมูลเฉพาะ row ที่มีค่า region_id เท่ากับ 1 หรือ 2 หรือ 3 ค่าใดค่าหนึ่ง ตัวเลขที่เขียนในวงเล็บไม่จำเป็นต้องเรียงลำดับ



รูปภาพที่ 6.21

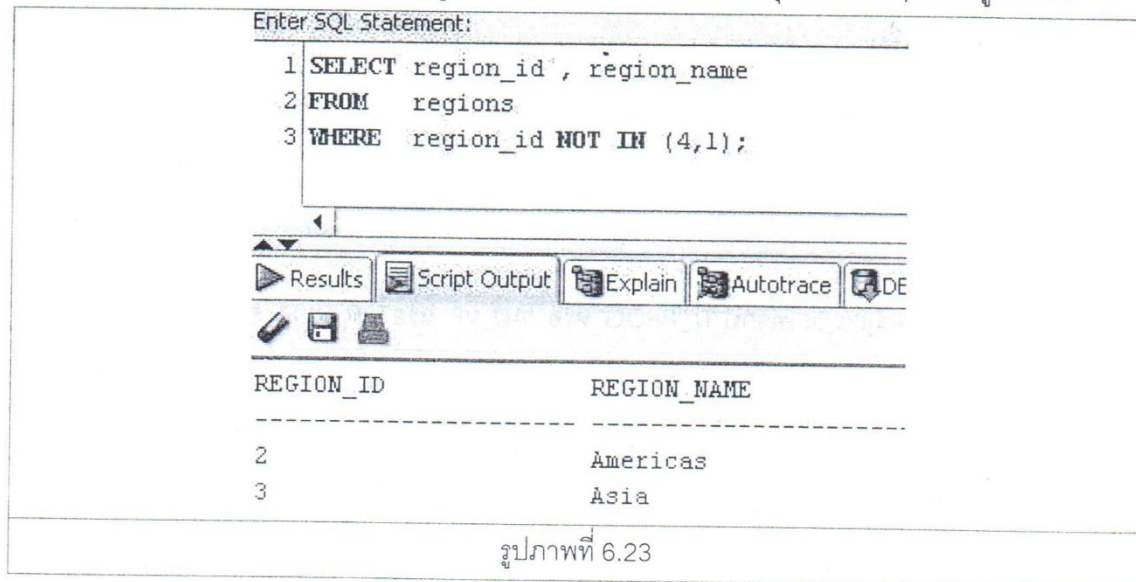
ปฏิบัติการที่ 6.14 การเลือกดูข้อมูลโดยกำหนดเงื่อนไข IN กับ column ที่มีชนิดข้อมูลแบบตัวอักษร

- ให้ผู้อ่านทดลองพิมพ์คำสั่งดังรูปภาพ 6.22 ต่อไปนี้ จากนั้นให้คลิกปุ่ม Run Script เพื่อดูผลลัพธ์ที่เกิดขึ้น
- สังเกตว่าค่าของข้อมูลชนิดตัวอักษรที่นำมาเปรียบเทียบ ต้องเขียนอยู่ในเครื่องหมาย ' ' และจะเขียนเป็นตัวอักษรตัวพิมพ์เล็ก ตัวพิมพ์ใหญ่หรือผสมกัน ขึ้นอยู่กับเนื้อข้อมูลที่บันทึกในฐานข้อมูล
- เงื่อนไข WHERE job_id IN ('IT_PROG', 'AD_VP', 'HR_REP') หมายถึงการเลือกข้อมูลเฉพาะ row ที่มีค่าของ job_id เท่ากับ 'IT_PROG' หรือ 'AD_VP' หรือ 'HR_REP' ค่าใดค่าหนึ่ง



ปฏิบัติการที่ 6.15 การเลือกดูข้อมูลโดยกำหนดเงื่อนไข IN ร่วมกับคีย์เวิร์ด NOT

- ให้ผู้อ่านทดลองพิมพ์คำสั่งดังรูปภาพต่อไปนี้ จากนั้นให้คลิกปุ่ม Run Script เพื่อดูผลลัพธ์



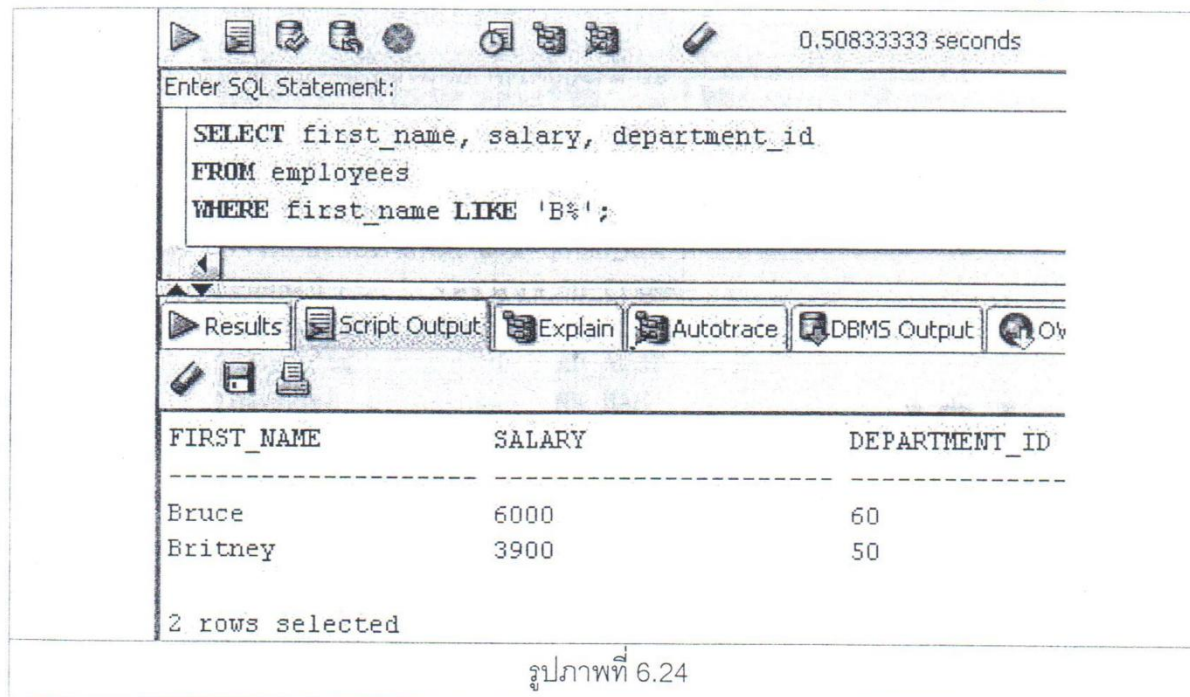
เงื่อนไข WHERE region_id NOT IN (4, 1) หมายถึงให้แสดงข้อมูล row ที่มีค่า region_id เป็นค่าใดก็ได้ที่ไม่ใช่ 4 หรือ 1

เงื่อนไขที่ใช้ LIKE operator

การใช้ LIKE operator เหมาะสำหรับเงื่อนไขที่เปรียบเทียบกับข้อมูลชนิดตัวอักษรหรือข้อมูลชนิดวันที่ โดยระบุค่าสำหรับเปรียบเทียบเพียงบางส่วน

ปฏิบัติการที่ 6.16 การเลือกดูข้อมูลโดยกำหนดเงื่อนไข LIKE operator

ให้ผู้อ่านทดลองพิมพ์คำสั่งดังรูปภาพต่อไปนี้ จากนั้นให้คลิกปุ่ม Run Script เพื่อดูผลลัพธ์



The screenshot shows a SQL IDE interface. At the top, there is a toolbar with various icons and a timer showing 0.50833333 seconds. Below the toolbar, there is a text area for entering SQL statements. The entered statement is:

```
SELECT first_name, salary, department_id
FROM employees
WHERE first_name LIKE 'B%';
```

Below the text area, there is a row of buttons: Results, Script Output, Explain, Autotrace, DBMS Output, and O/v. The Results button is selected, and the results are displayed in a table below. The table has three columns: FIRST_NAME, SALARY, and DEPARTMENT_ID. The results show two rows: Bruce with salary 6000 and department_id 60, and Britney with salary 3900 and department_id 50. At the bottom of the table, it says "2 rows selected".

FIRST_NAME	SALARY	DEPARTMENT_ID
Bruce	6000	60
Britney	3900	50

รูปภาพที่ 6.24

จากรูปภาพ เงื่อนไข WHERE first_name LIKE 'B%' หมายถึงให้แสดงข้อมูลพนักงาน ที่มีชื่อขึ้นต้นด้วยตัวอักษร B ส่วนตัวอักษรที่เหลือจะเป็นตัวอักษรใดก็ได้และจะมีอักษรกี่ตัวก็ได้ (หมายเหตุ สัญลักษณ์ % จะถูกเรียกว่า wildcard หมายถึงสัญลักษณ์ที่ใช้แทนตัวอักษรใดๆจำนวนกี่ตัวก็ได้)

สัญลักษณ์ % ที่ใช้ในเงื่อนไขร่วมกับ LIKE สามารถใช้ได้มากกว่า 1 ตัว ตารางต่อไปนี้แสดงตัวอย่างการใช้ % ร่วมกับ LIKE และความหมายของแต่ละเงื่อนไข

ตารางแสดงตัวอย่างการใช้ % กับ LIKE operator

ตัวอย่างการใช้	ความหมาย
WHERE first_name LIKE '%b%'	หมายถึงเลือกข้อมูลเฉพาะ row ที่ ค่าของ column ชื่อ first_name มีอักษร b อยู่ โดยที่ตำแหน่งของอักษร b นี้จะอยู่ตรงตำแหน่งใดก็ได้
WHERE first_name LIKE '%ab%'	หมายถึงเลือกข้อมูลเฉพาะ row ที่ ค่าของ column ชื่อ first_name มีอักษร ab อยู่ โดยที่ตำแหน่งของอักษร ab นี้จะอยู่ตรงตำแหน่งใดก็ได้ แต่อักษร ab ต้องอยู่ติดกัน
WHERE first_name LIKE '%a%b%'	หมายถึงเลือกข้อมูลเฉพาะ row ที่ ค่าของ column ชื่อ first_name มีอักษร ab อยู่ โดยที่ตำแหน่งของอักษร ab นี้จะอยู่ตรงตำแหน่งใดก็ได้ นอกจากนี้ อักษร ab นี้จะอยู่ติดกัน หรืออยู่แยกกันก็ได้
WHERE first_name NOT LIKE 'A%';	หมายถึงเลือกข้อมูลเฉพาะ row ที่ ค่าของ column ชื่อ first_name ไม่ได้ขึ้นต้นด้วยตัวอักษร A นำหน้า (สังเกตการใช้ คีย์เวิร์ด NOT ร่วมกับ LIKE)

เงื่อนไขที่ใช้ IS NULL operator

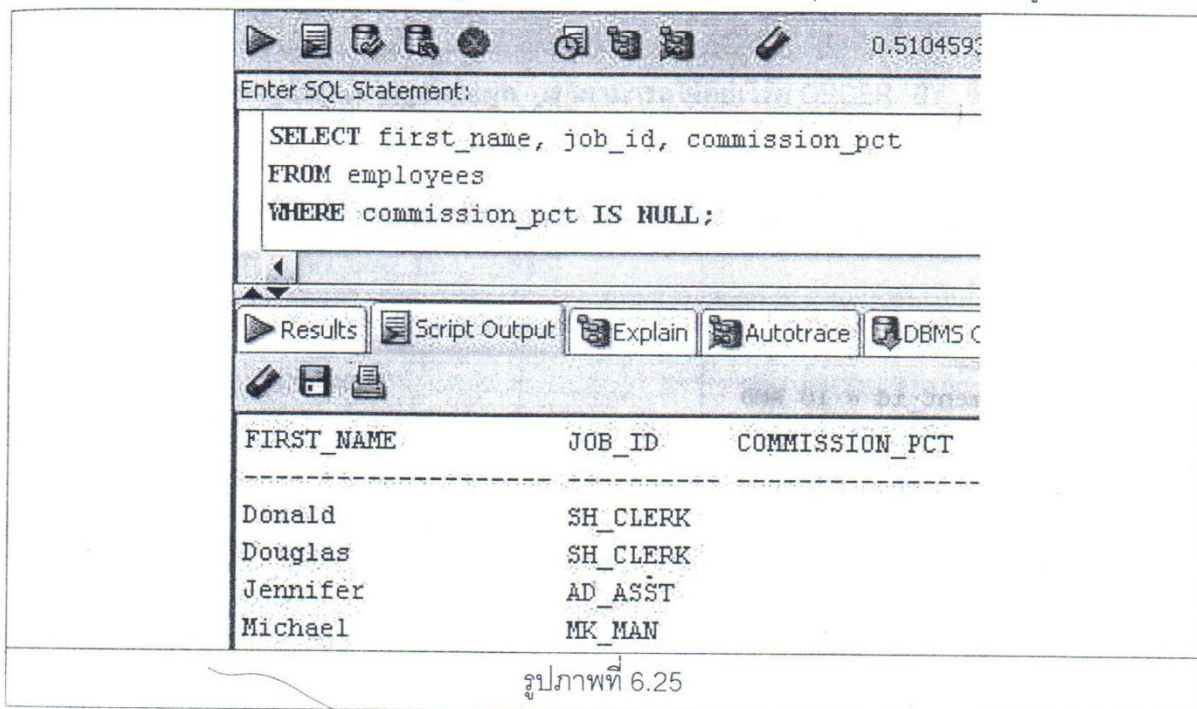
สำหรับ column ใดที่ไม่มีการกำหนดค่าให้ ในระบบฐานข้อมูลจะกำหนดเป็น ค่าว่าง(null) โดยที่ ค่าว่างจะไม่ได้หมายถึง 0 หรือ blank (ตัวอักษรที่เกิดจากการ เคาะ space bar บนแป้นพิมพ์)

ตัวอย่างการกำหนดค่าว่างในระบบฐานข้อมูล ให้ผู้อ่านลองพิจารณาโครงสร้างของ table ชื่อ employees จะพบว่า มี column ชื่อ commission_pct ซึ่งกำหนดไว้เพื่อเก็บค่าเปอร์เซ็นต์ของการคิดคอมมิชชั่น ข้อมูลใน table ชื่อ employees คือข้อมูลของพนักงานทุกคน แต่จะมีพนักงานจำนวนหนึ่งเท่านั้นที่เป็นพนักงานขาย และสำหรับข้อมูลของพนักงานขาย column ชื่อ commission_pct จะเก็บตัวเลขเปอร์เซ็นต์ของการคิดคอมมิชชั่น ส่วนข้อมูลของพนักงานอื่นๆ column ชื่อ commission_pct จะถูกกำหนดเป็นค่าว่าง

ถ้าผู้อ่านต้องการเลือกดูข้อมูลของพนักงานอื่นๆ ที่ไม่ใช่พนักงานขาย เงื่อนไขที่สามารถกำหนดได้คือ เลือกข้อมูลพนักงานที่ค่าของ commission_pct เท่ากับ ค่าว่าง เงื่อนไขดังกล่าวนี้จะใช้ IS NULL operator ดังที่จะแสดงในปฏิบัติการต่อไปนี้

ปฏิบัติการที่ 6.17 การเลือกดูข้อมูลโดยกำหนดเงื่อนไข IS NULL operator

- ให้ผู้อ่านทดลองพิมพ์คำสั่งดังรูปภาพต่อไปนี้ จากนั้นให้คลิกปุ่ม Run Script เพื่อดูผลลัพธ์



The screenshot shows an SQL IDE interface. At the top, there is a toolbar with various icons and a version number '0.5104593'. Below the toolbar is a text area labeled 'Enter SQL Statement:' containing the following SQL query:

```
SELECT first_name, job_id, commission_pct
FROM employees
WHERE commission_pct IS NULL;
```

Below the text area is a row of buttons: 'Results', 'Script Output', 'Explain', 'Autotrace', and 'DBMS C'. Below these buttons is a table displaying the results of the query. The table has three columns: 'FIRST_NAME', 'JOB_ID', and 'COMMISSION_PCT'. The data rows are as follows:

FIRST_NAME	JOB_ID	COMMISSION_PCT
Donald	SH_CLERK	
Douglas	SH_CLERK	
Jennifer	AD_ASST	
Michael	MK_MAN	

Below the table, the text 'รูปภาพที่ 6.25' is written.

รูปภาพที่ 6.25 แสดงการใช้เงื่อนไข WHERE commission_pct IS NULL เพื่อเลือกดูข้อมูลพนักงานอื่นๆ ที่ไม่ใช่พนักงานขาย สังเกตว่าเราไม่สามารถเขียนเงื่อนไข WHERE commission_pct = "" หรือ WHERE commission_pct = 0ทดแทนการใช้ IS NULL operator

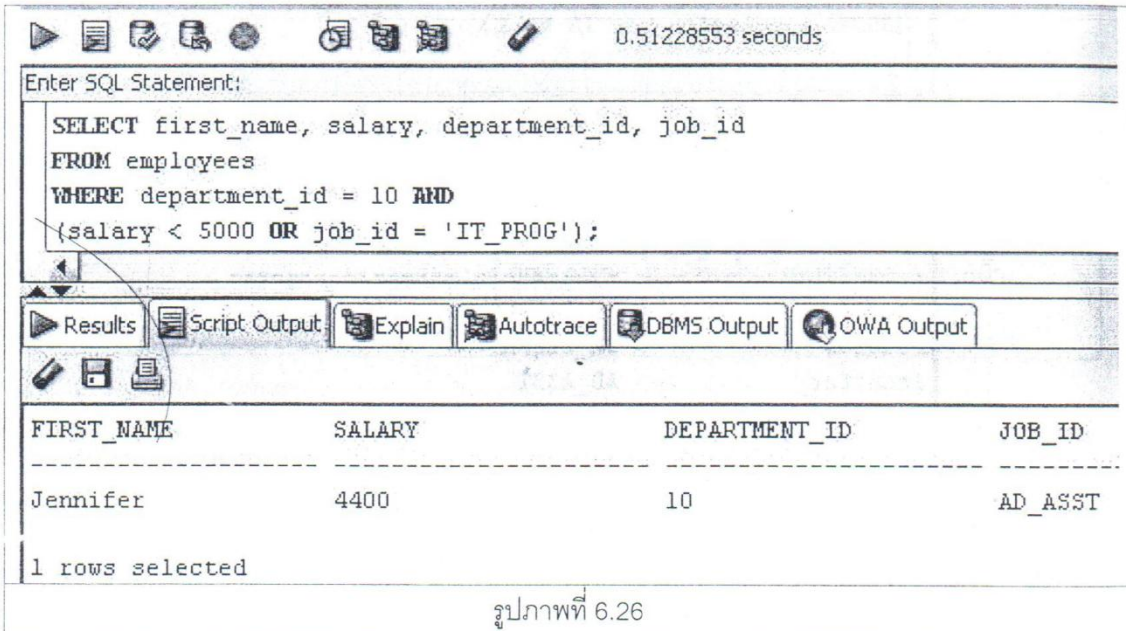
เราสามารถใส่คีย์เวิร์ด NOT ร่วมกับ IS NULL operator เพื่อใช้กำหนดเงื่อนไขในการเลือกดูข้อมูล จากคำสั่งในรูปภาพที่ 6.25 นี้ ถ้าเขียนเงื่อนไขเป็น WHERE commission_pct IS NOT NULL จะหมายถึง เลือกดูเฉพาะข้อมูลของพนักงานขาย (ข้อมูลของพนักงานขาย ค่าของ column ชื่อ commission_pct จะไม่เป็นค่าว่าง)

การรวมหลายเงื่อนไขไว้ด้วยกัน

เราสามารถนำเงื่อนไขรูปแบบต่างๆ ที่ได้กล่าวมาข้างต้น มากำหนดเป็นเงื่อนไขที่ซับซ้อนขึ้น โดยการเพิ่มคีย์เวิร์ด AND หรือ คีย์เวิร์ด OR เพื่อรวมเงื่อนไขเข้าด้วยกัน ในกรณีที่ต้องการกำหนดลำดับของการตรวจเช็คเงื่อนไข จะนำเครื่องหมายวงเล็บมาเป็นตัวกำหนด โดยมีหลักการง่ายๆ คือ การตรวจเช็คเงื่อนไขจะเริ่มจากวงเล็บในสุดก่อน แล้ววงเล็บลำดับถัดมาจะถูกตรวจเช็คเงื่อนไขต่อมา ตามลำดับ

ปฏิบัติการที่ 6.18 การเลือกข้อมูลโดยกำหนดหลายเงื่อนไขรวมกัน

- ให้ผู้อ่านทดลองพิมพ์คำสั่งดังรูปภาพ ต่อไปนี้ และคลิกปุ่ม Run Script เพื่อดูผลลัพธ์



The screenshot shows a SQL query execution interface. At the top, there is a toolbar with various icons and a timer showing 0.51228553 seconds. Below the toolbar, the "Enter SQL Statement:" field contains the following query:

```
SELECT first_name, salary, department_id, job_id
FROM employees
WHERE department_id = 10 AND
(salary < 5000 OR job_id = 'IT_PROG');
```

Below the query field, there are tabs for "Results", "Script Output", "Explain", "Autotrace", "DBMS Output", and "OWA Output". The "Results" tab is selected, and it displays a table with the following data:

FIRST_NAME	SALARY	DEPARTMENT_ID	JOB_ID
Jennifer	4400	10	AD_ASST

Below the table, it says "1 rows selected".

รูปภาพที่ 6.26

พิจารณาเงื่อนไขในรูปภาพ ดังต่อไปนี้

WHERE department_id = 10 AND (salary < 5000 OR job_id = 'IT_PROG')



เงื่อนไขส่วนที่ 1 ซึ่งเป็นเงื่อนไขในวงเล็บจะถูกตรวจเช็คก่อน ในที่นี้ข้อมูลที่ถูกเลือกมาคือข้อมูล
ของพนักงานที่มีเงินเดือนน้อยกว่า 5000 หรือ (OR) ข้อมูลของพนักงานที่มีค่า job_id เท่ากับ IT_PROG

จากนั้น ข้อมูลที่ถูกเลือกด้วยเงื่อนไขที่ 1 จะถูกนำมาตรวจเช็คกับเงื่อนไขที่ 2 ซึ่งเชื่อมด้วยคีย์
เวิร์ด AND หมายความว่า ข้อมูลที่แสดงในผลลัพธ์ จะต้องเป็นข้อมูลที่ได้จากเงื่อนไขที่ 1 และ (AND) มี
ค่า department_id เท่ากับ 10