

EP1 : สร้างฐานข้อมูล และตาราง

(Ex)

ข้อมูลพนักงาน (Employee)

- รหัสพนักงาน (Pk) : E001, E002 (String)

เก็บตัวหนังสือ
กับตัวอักษร
→ ชื่อ (String, VARCHAR(255), TEXT) ปัญชื่อยางไร ต้องเกิน 255 ตัว

→ นามสกุล

- อายุ (TEXT)

→ (อายุ) ถูกกำหนด (integer) ต้องเป็นตัวเลข

- เงินเดือน (INT, FLOATING)

บัญชีเงินเดือน

- รหัสเงินเดือน (Pk) (P001) 1, 2, 3, ...

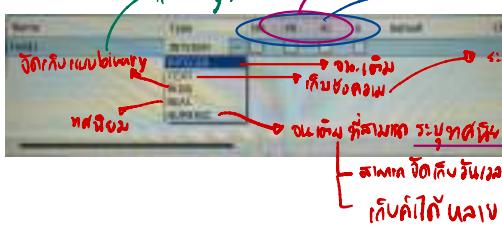
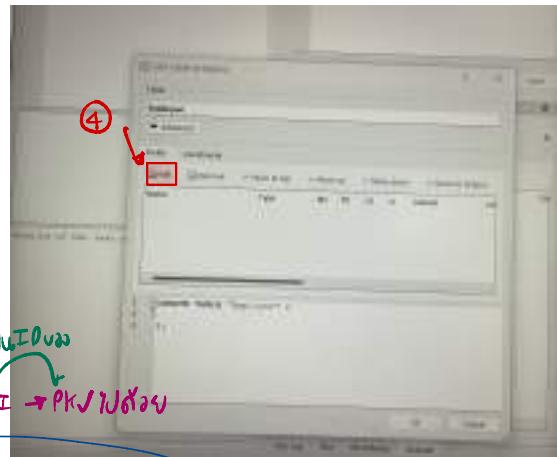
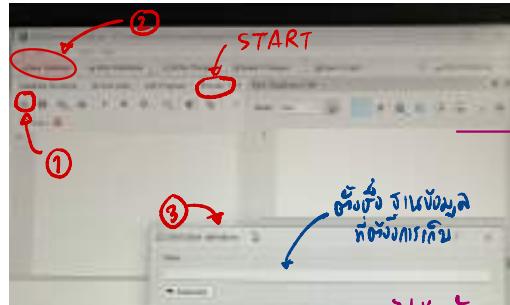
- รุ่นเงินเดือน

- รายวันเงินเดือน

- ยอดคงเหลือ (fk)

- รายการ

STEP สร้างฐานข้อมูล

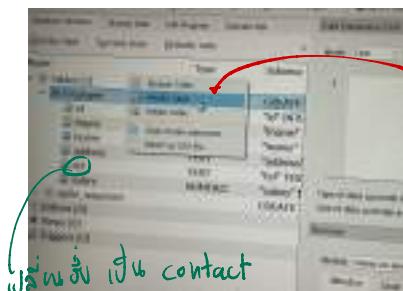
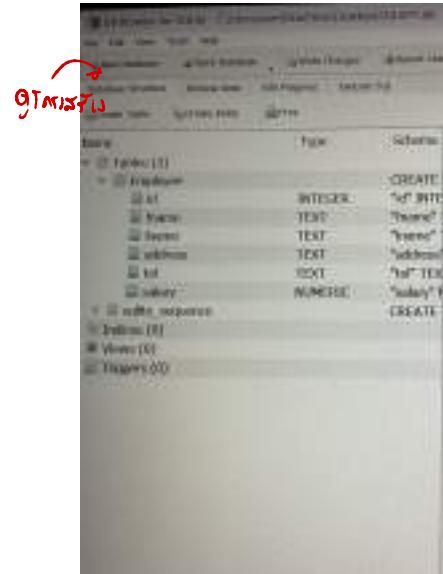
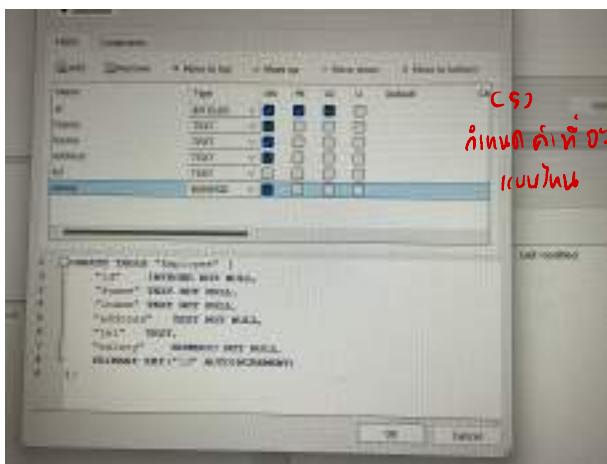


NN : Not Null = กำหนดมา
ให้เป็นตัวร่วมไป
ด้วยเป็นตัวบ่งบอก

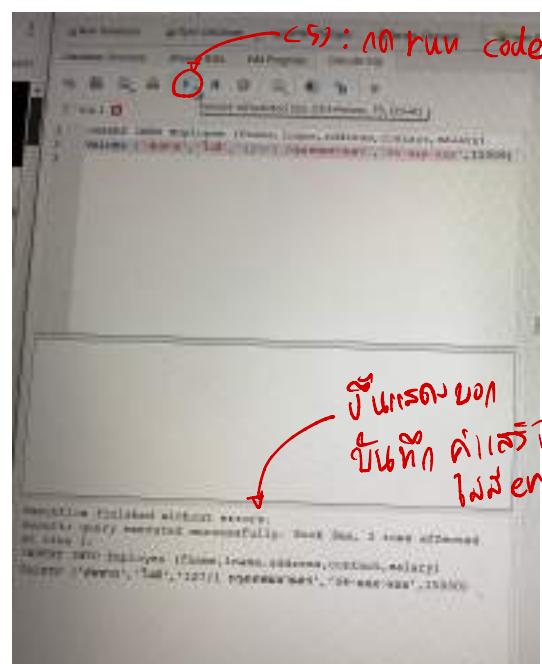
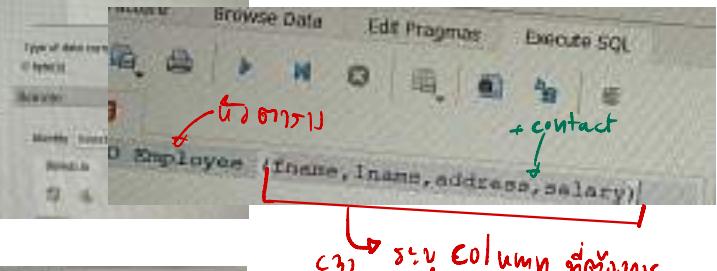
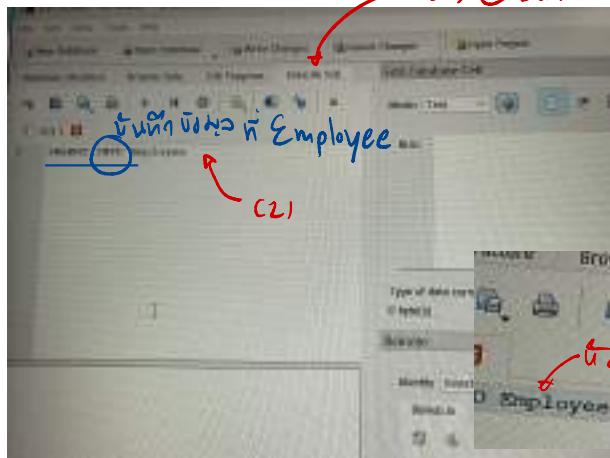
PK : Primary key = หัวตัวเรื่อง เช่น
auto increment หรือ 1,2,3,...

AI : Autoincrement = หัวตัวเรื่อง เช่น
auto increment หรือ 1,2,3,...

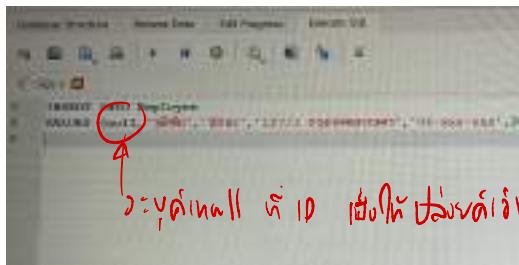
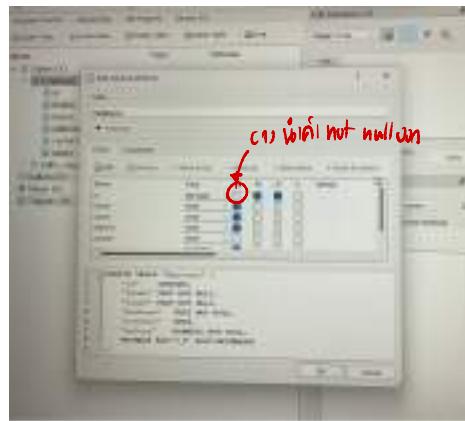
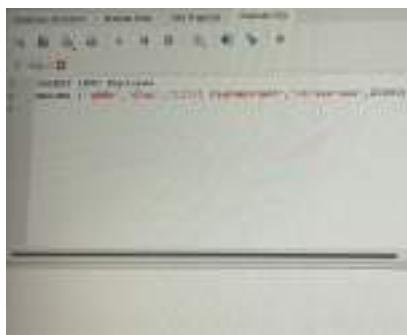
คำอธิบาย: กำหนด primary key และ nullable



EP2 : បិនទៅឱ្យសម្រាប់ (Insert) រូបរាងទាំង 1



EP3 : ប្រើកែសម្រាប់លើកសម្រាប់ column ទី ២

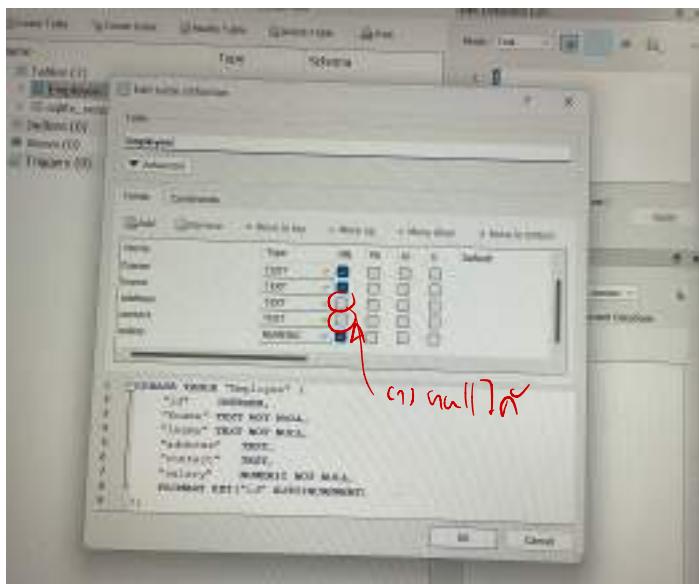


#	Name	Address	Salary
1	John	123 Main St.	25000
2	Jane	123 Main St.	15000
3	Mike	123 Main St.	20000

ជាឧតុលាតាម id, AI

EP4 : ឧបនា សម្រេចនូវ column នៃលីតុលិក

លីតុលិក



(1)

```
SQL> INSERT INTO employee (ename,deptno,sal)
  VALUES ('SALESMAN',10,2000)
```

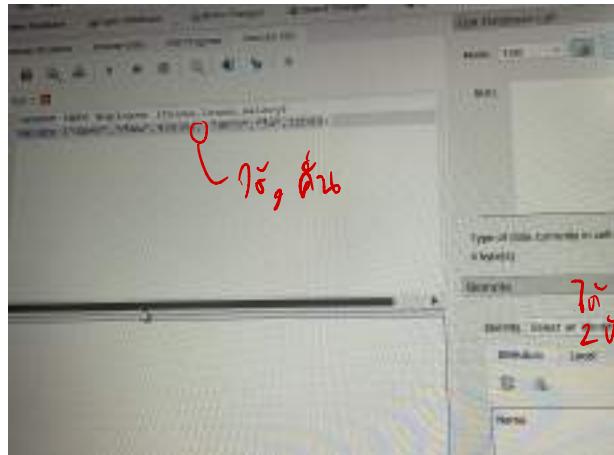
(2)

(3)

#	empno	ename	deptno	sal	comm
1	100	SALESMAN	10	2000	0
2	101	MANAGER	10	3000	0
3	102	TICKET	10	1500	0
4	103	CLERK	10	1200	0
5	104	CLERK	10	1200	0
6	105	CLERK	10	1200	0
7	106	CLERK	10	1200	0
8	107	CLERK	10	1200	0
9	108	CLERK	10	1200	0
10	109	CLERK	10	1200	0
11	110	CLERK	10	1200	0
12	111	CLERK	10	1200	0
13	112	CLERK	10	1200	0
14	113	CLERK	14	1500	0

ត្រូវនា 2នៃ column ទាំង ៤

EP 5 : ផ្លូវកំណើនមុន ឬលើរដូវការនិងទីតាំង និងបេណ្ណ



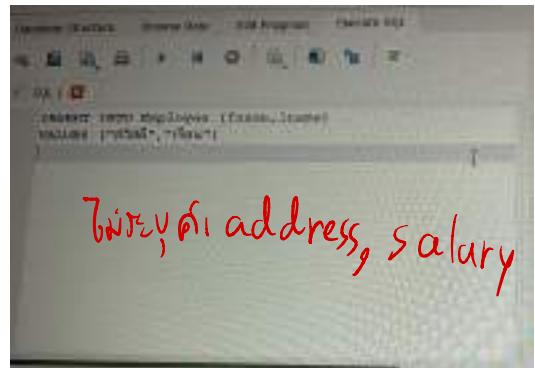
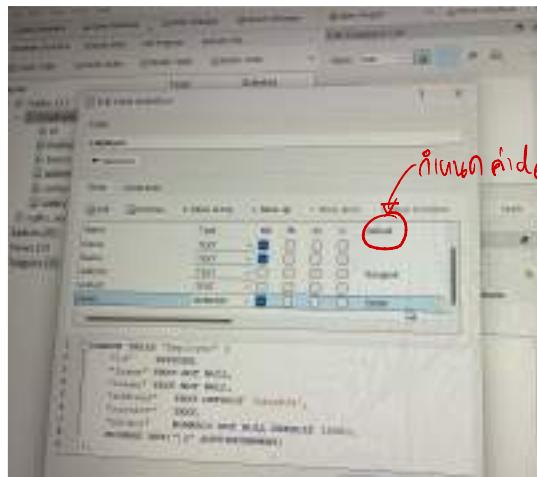
ID	Name	Address	Salary
1	John	123 Main St.	30000
2	Anna	111 Elm St.	25000
3	Peter	134 Oak St.	28000
4	Jane	155 Pine St.	26000
5	David	166 Cedar St.	40000
6	Sarah	177 Birch St.	40000
7	Emily	188 Maple St.	12000

ទីតាំង

ទីតាំងនៃរបៀបការងារ

EP6 : ចិនអីនិងរាយការ Default សម្រាប់

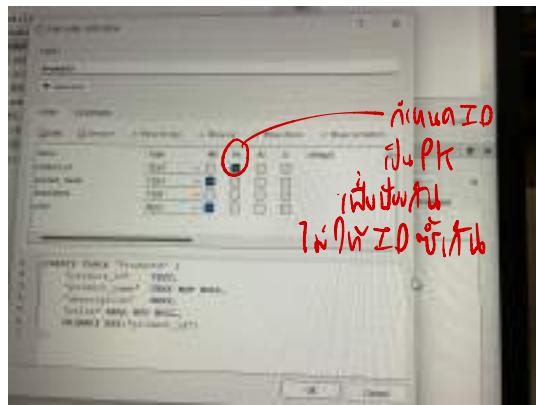
Let's see how to set up default values for columns in MySQL



ID	Name	Address	Salary	Notes
1	John Doe	123 Main St	50000	
2	Jane Doe	123 Main St	55000	
3	Mike Johnson	123 Main St	60000	
4	Sarah Johnson	123 Main St	65000	
5	David Johnson	123 Main St	70000	
6	Emily Johnson	123 Main St	75000	
7	Robert Johnson	123 Main St	80000	
8	Sarah Johnson	123 Main St	85000	
9	David Johnson	123 Main St	90000	
10	Emily Johnson	123 Main St	95000	
11	Robert Johnson	123 Main St	100000	

E P7 : ជាន់ទីក្រុងមុនវាន់ខ្លាត់ទៅ Unique ត្រូវបានពិនិត្យ

លើអាជីវកម្មនៃការងារ column តើរួចរាល់



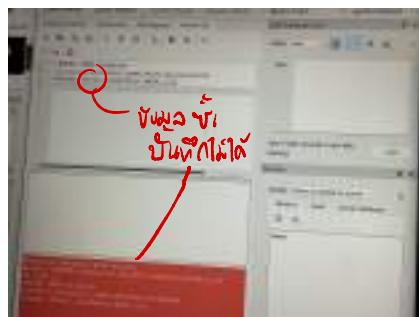
```
USE test;
SELECT * FROM product;
+-----+-----+-----+-----+
| id   | name | desc | price |
+-----+-----+-----+-----+
| 1    | apple| fruit| 150.0 |
+-----+-----+-----+-----+
```

SQL query results:

```
1 row(s) finished without errors.
Query executed successfully. Took 0.000 sec.
```

ការកំណត់ត្រាបច្ចុប្បន្ននៃការកំណត់ត្រាបច្ចុប្បន្ន។ ការកំណត់ត្រាបច្ចុប្បន្នត្រូវបានធ្វើឡើងនៅក្នុងការកំណត់ត្រាបច្ចុប្បន្ន។

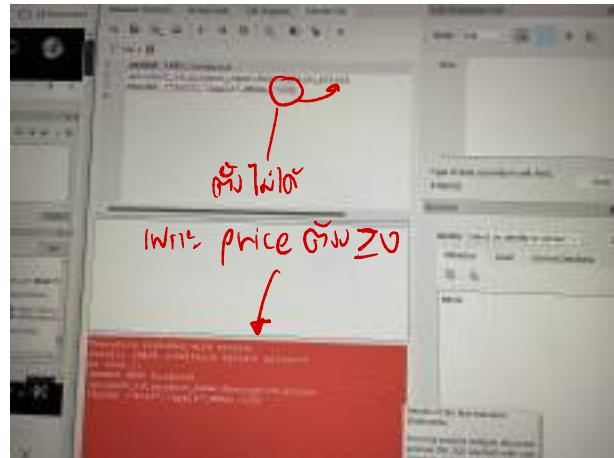
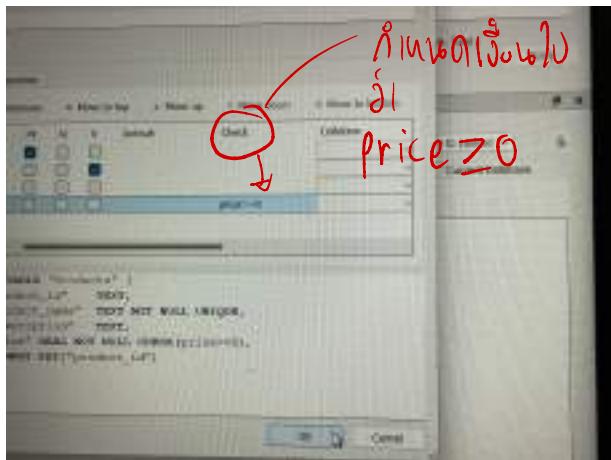
product_id	product_name	description	price
1	apple	fruit	150.0



ការកំណត់ត្រាបច្ចុប្បន្ននៃការកំណត់ត្រាបច្ចុប្បន្ន។ ការកំណត់ត្រាបច្ចុប្បន្នត្រូវបានធ្វើឡើងនៅក្នុងការកំណត់ត្រាបច្ចុប្បន្ន។

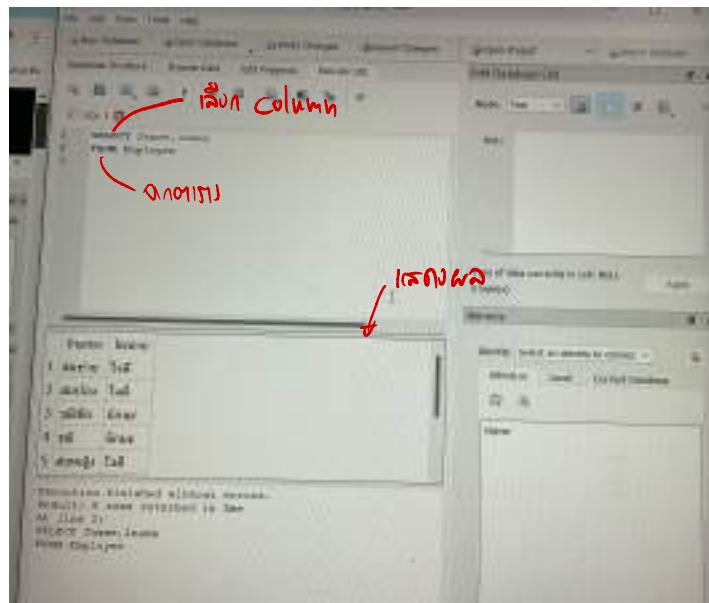
product_id	product_name	description	price
1	apple	fruit	150.0

EP8 : សិកសម្រាប់តាមវិធាននៃការ Check



EP9 : การสอบถามข้อมูล c SELECT

└ ภาษา Table ที่รับการตอบ



EP10 : លក្ខណៈបច្ចាម ចំណាំនូវ column

សម្រាប់ * តម្លៃអង្គភាព
ដែលពីរ និង Auto.

ToN

ID	Name	Address	CityID	Salary
10001	John Doe	123 Main St, Anytown, USA	10001	50000
10002	Jane Smith	456 Elm St, Anytown, USA	10002	45000
10003	Bob Johnson	789 Oak St, Anytown, USA	10003	60000
10004	Sarah Williams	210 Pine St, Anytown, USA	10004	55000
10005	David Lee	345 Cedar St, Anytown, USA	10005	48000

EP11 : ពិនិត្យ column ទៅ AS

តាមអាជីវកម្ម column នឹងរាយការណ៍បញ្ជី
ទាន និងនូវការក្រោរពន្លឺ Database

	Customer ID	Name	Salary
1	John	John Doe	30000
2	Mike	Mike Johnson	30000
3	Jane	Jane Doe	30000
4	Sam	Sam Johnson	30000
5	Angela	Angela Johnson	40000

MySQL Workbench Version 8.0.25
Copyright © 2023 Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

EP12 : ແລກປັນນີ້ ໂື່ສັບຕົວດໍາລັງ DISTINCT

Before

ນີ້ແມ່ນເວັບໄຈ^{ຈະ}
column fname
ມີເພິ່ນກຳໄວ

After

ມີຫຼຸດທີ່

ມີຫຼຸດທີ່
ມີຫຼຸດທີ່

ຈະ 12A16

EP13 : การสอบถามข้อมูล ที่มีเงื่อนไข (WHERE)

The screenshot shows the MySQL Workbench interface. In the top-left, there's a query editor window with the following SQL code:

```
SELECT * FROM Employees WHERE salary > 2000;
```

Below it is a results grid with the following data:

name	hireDate	salary
John Doe	2000-01-01	2000

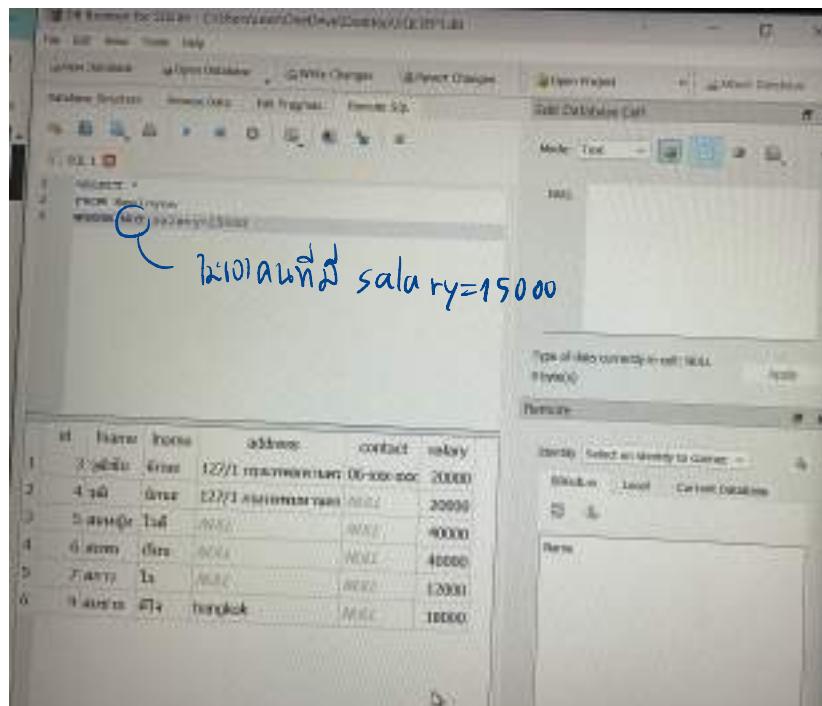
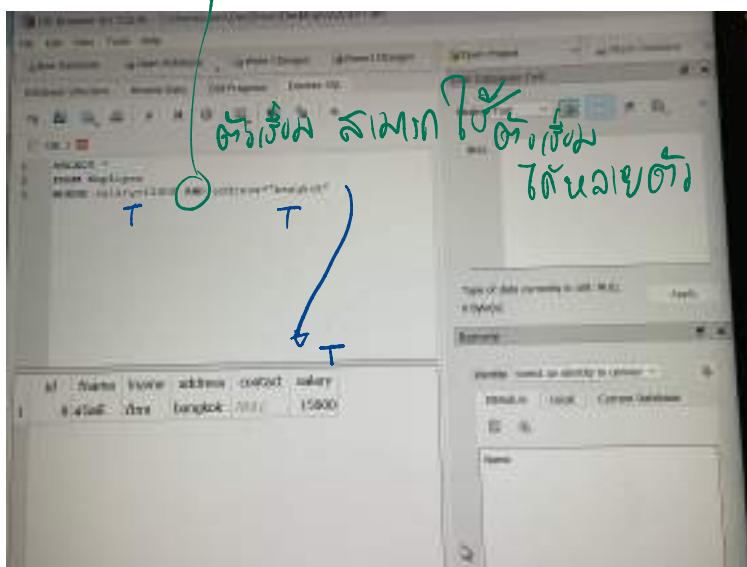
Handwritten annotations in red are present: "ตั้งเงื่อนไขเป็นค่าที่ต้องการ" (Set condition to the required value) and "ผลลัพธ์เดียว" (One result). A red arrow points from the word "เดียว" to the single row in the results grid.

ຈຳກັດຮຽນ | ເພີ່ມປະຕິບທີ່ຢູ່ປ
ແສ່ວຂອງຕາມຄວາມ

EP 14: ຕົວຢັນການກັບປຶກທີ່ໜີ

=
!= \neq ດັວກເກີດ
>
<
>=
<=

EP15 : ទីនោរណ៍នៃសម្រាប់ AND, OR, NOT



EP 16 : សេចក្តីរាយការត្រួវ IS NULL

សេចក្តីរាយការត្រួវ NOT NULL

ត្រូវបញ្ជី
រាយការត្រួវ
contact

ID	Name	Phone	Address	Contact	Salary
1	A. S.	0123456789	123/1 address road	093	20000
2	B. Seng	0123456789	123/2 address road	093	40000
3	C. Sam	0123456789	123/3 address road	093	40000
4	D. Anna	0123456789	123/4 address road	093	12000
5	E. Jack	0123456789	123/5 address road	093	15000
6	F. Sandra	0123456789	123/6 address road	093	18000

EP17 : ເຮັດວຽກພະລິບດ້ານ ORDER BY (ເປົ້າຜົນທີ່ໃຫຍ່ WHERE)

ID	Name	Address	Contact	Salary
1	John	1234 Main Street, New York	1234567890	30000
2	Jane	1234 Main Street, New York	1234567890	25000
3	Tom	1234 Main Street, New York	1234567890	22000
4	Karen	1234 Main Street, New York	1234567890	20000
5	David	1234 Main Street, New York	1234567890	18000
6	Emily	1234 Main Street, New York	1234567890	16000
7	Michael	1234 Main Street, New York	1234567890	15000
8	Sarah	1234 Main Street, New York	1234567890	14000
9	David	1234 Main Street, New York	1234567890	13000

ຫຼາຍ : ເຮັດວຽກພະລິບດ້ານ ORDER BY
ຫຼາຍ : ພະລິບ A-Z
ຫຼາຍ : ເຮັດວຽກພະລິບດ້ານ WHERE

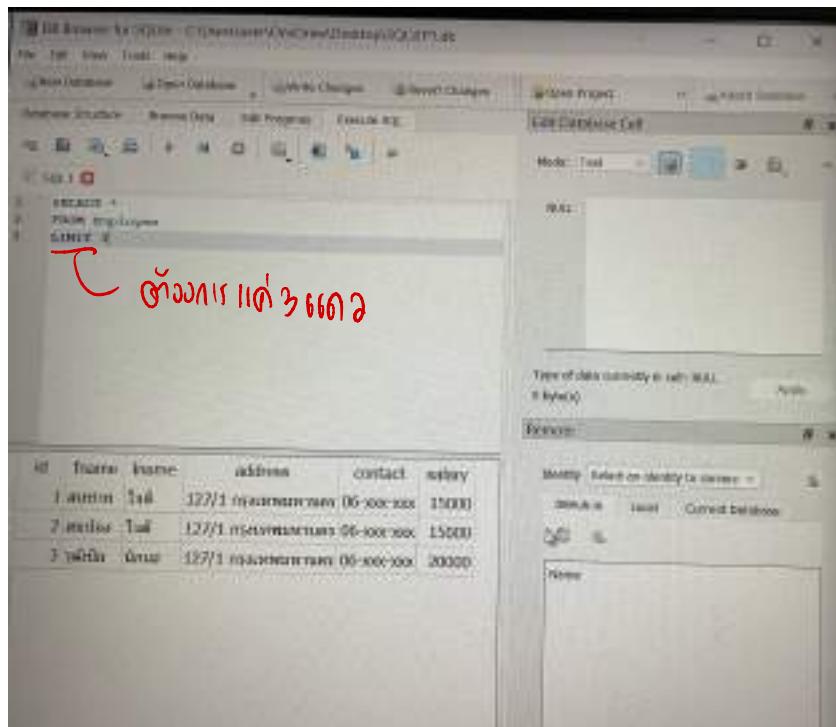
ASC → ເຮັດວຽກນັ້ນຈຳນວຍ

DESC → ເຮັດວຽກນັ້ນຈຳນວຍ

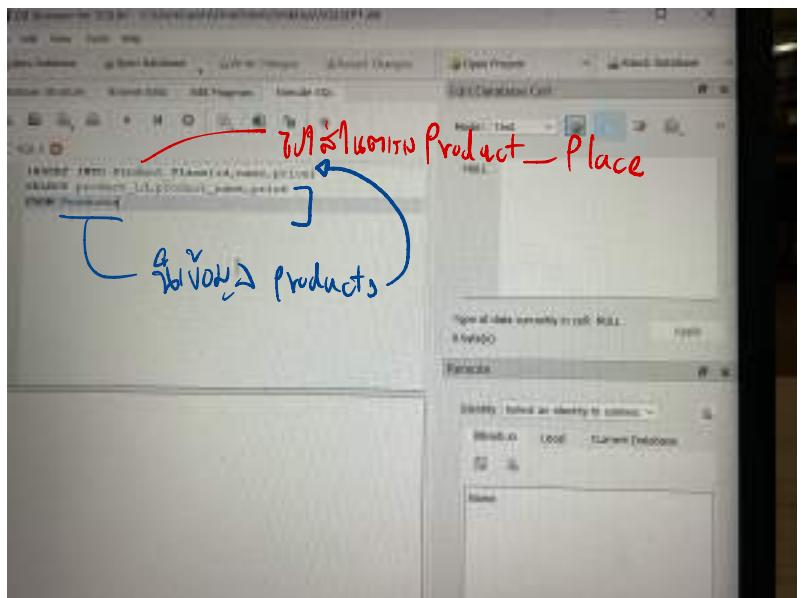
ID	Name	Address	Contact	Salary
1	John	1234 Main Street, New York	1234567890	30000
2	Jane	1234 Main Street, New York	1234567890	25000
3	Tom	1234 Main Street, New York	1234567890	22000
4	Karen	1234 Main Street, New York	1234567890	20000
5	David	1234 Main Street, New York	1234567890	18000
6	Emily	1234 Main Street, New York	1234567890	16000
7	Michael	1234 Main Street, New York	1234567890	15000
8	Sarah	1234 Main Street, New York	1234567890	14000
9	David	1234 Main Street, New York	1234567890	13000
10	John	1234 Main Street, New York	1234567890	12000

ຫຼາຍວຽກພະລິບດ້ານ WHERE

EP 18 : កំណត់វិនាទន ប្រចាំពេល LIMIT



EP19 : ដំឡើងតម្លៃផលិតផលទាហរវិន



	id	product	place	market
1	ID1	laptop	150.0	market
2	ID2	monitor	150.0	market
3	ID3	apple	150.0	market

EP20 : ໂັດຮັບເຄືອງກົມະລິການ

Before

	Name	Birth	Address	Phone	Salary
1.	Lander, Ted	1971-08-10		15000	
2.	Sandra, Ted	1973-06-20		15000	
3.	Sandra, Steve	1973-06-20		20000	
4.	Ted, Sean	1973-06-20		20000	
5.	Yamini, Ted	1971-05-01		40000	
6.	Sandra, Alice	1973-05-01		40000	
7.	Sandra, Tom	1973-05-01		12000	
8.	Sandra, Rita	1973-05-01		15000	

update ຈົດ
ກົມະລິການ Employee
ມີລົງທຶນ auto_inj_id = 6

	Name	Birth	Address	Phone	Salary
1.	Lander, Ted	1971-08-10		15000	
2.	Sandra, Ted	1973-06-20		15000	
3.	Sandra, Steve	1973-06-20		20000	
4.	Ted, Sean	1973-06-20		20000	
5.	Yamini, Ted	1971-05-01		40000	
6.	Sandra, Alice	1973-05-01		30000	
7.	Sandra, Tom	1973-05-01		12000	
8.	Sandra, Rita	1973-05-01		15000	

After

	Name	Birth	Address	Phone	Salary
1.	Lander, Ted	1971-08-10		15000	
2.	Sandra, Ted	1973-06-20		15000	
3.	Sandra, Steve	1973-06-20		20000	
4.	Ted, Sean	1973-06-20		20000	
5.	Yamini, Ted	1971-05-01		40000	
6.	Sandra, Alice	1973-05-01		30000	
7.	Sandra, Tom	1973-05-01		12000	
8.	Sandra, Rita	1973-05-01		15000	

EP21 : ໂປ່ງນູ້ໃນຕາງວາ

Before

ID	Product Name	Description	Price
1001	Banana	Fruit	150.0
1002	apple	Fruit	150.0
1003	mango	Fruit	150.0

ລາຍລະອຽດ Oitable

Transaction finished without errors.
Statement: query executed successfully. Rows affected:
At line 1:
DELETE FROM Products
Rows affected: 1

After

ບົດລົງກົດຈຳກັດ

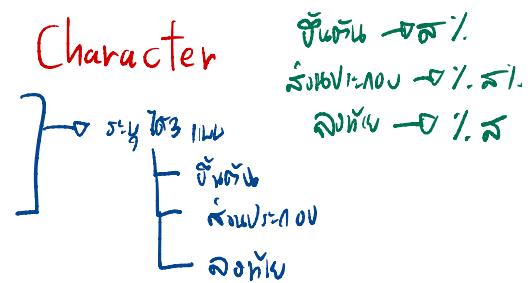
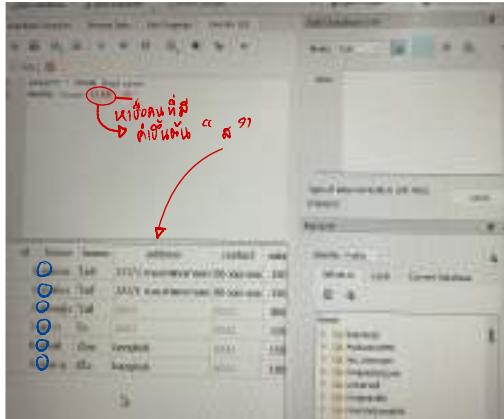
ລາຍລະອຽດ Oitable

ID	Product Name	Description	Price
1001	Banana	Fruit	150.0

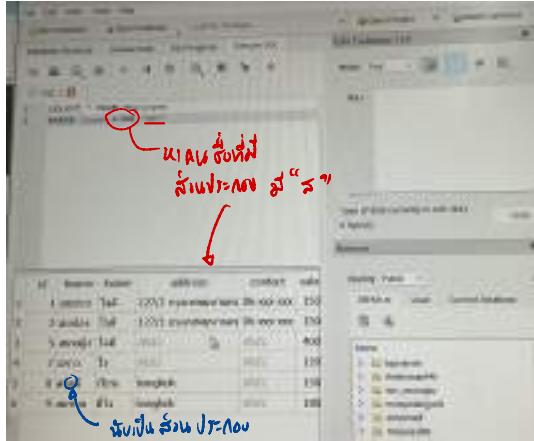
ID	Product Name	Description	Price
1001	Banana	Fruit	150.0

EP 22 : Like และ Wild Card Character

/.; ដើម្បីរាយការណ៍ ចុច្ចជាមួយសំខាន់ទិន្នន័យ
/.; នឹងបានពិនិត្យលទ្ធផល 1 ពិន្ទុ

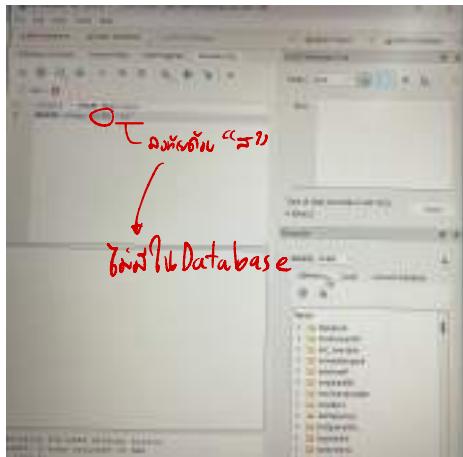



សំខាន់ទិន្នន័យ
 " %"

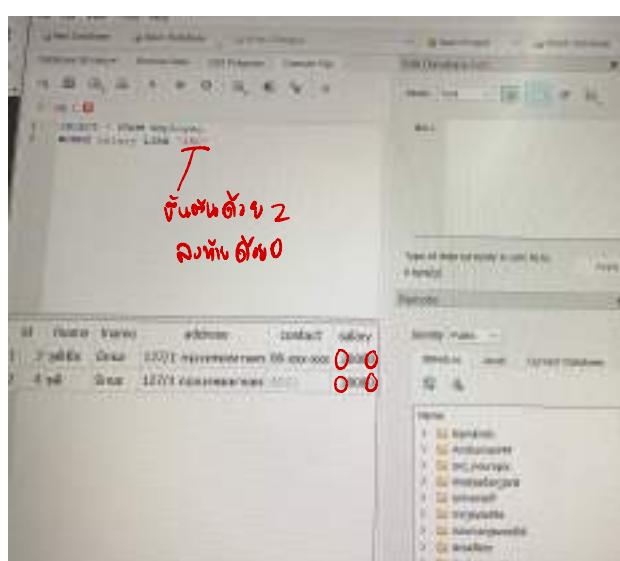


សំខាន់ទិន្នន័យ
 " %"

ដែលត្រូវការពិនិត្យលទ្ធផល



បញ្ជី Database



ត្រូវពិនិត្យទាំង 2
 លទ្ធផល នៅក្បាន់ទេ

ชื่อพนักงานต้องไม่บลanks

ต้องมีตั้งแต่ 1 ถึง 12,3
เป็นผลลัพธ์

	name	birthdate	address	contact	salary
1	John Doe	1990-01-01	New York, USA	123-456-7890	50000
2	Jane Smith	1985-05-20	Los Angeles, USA	123-456-7891	50000
3	Mike Johnson	1978-07-15	Chicago, USA	123-456-7892	40000
4	Sarah Davis	1982-03-10	Seattle, USA	123-456-7893	40000

ต้องมีตั้งแต่ 1 ถึง 12,3 ต้องไม่บลanks
ต้องมีตั้งแต่ 1 , จนถึง 10
เป็นผลลัพธ์

	name	birthdate	address	contact	salary
1	John Doe	1990-01-01	New York, USA	123-456-7890	50000
2	Jane Smith	1985-05-20	Los Angeles, USA	123-456-7891	50000
3	Mike Johnson	1978-07-15	Chicago, USA	123-456-7892	40000
4	Sarah Davis	1982-03-10	Seattle, USA	123-456-7893	40000

EP23 : ດີກິ່ນເອີນໄຊທາງ ດາວໂຫຼດກາລົກ

id	name	balance	available	vith
1	admin	10000	15000	14000
2	admin	10000	13000	14000
3	admin	20000	20000	19000
4	admin	20000	20500	19000
5	admin	100	40000	40500
6	admin	10000	40000	39000
7	admin	10	12000	11500
8	admin	10000	15500	14000
9	admin	10000	18500	17000

```
USE master
GO
CREATE DATABASE [test]
GO
USE test
GO
CREATE TABLE [members]
(
    id int,
    name nvarchar(50),
    balance int,
    available int,
    vith int
)
INSERT INTO members VALUES ('1','admin',10000,15000,14000)
INSERT INTO members VALUES ('2','admin',10000,13000,14000)
INSERT INTO members VALUES ('3','admin',20000,20000,19000)
INSERT INTO members VALUES ('4','admin',20000,20500,19000)
INSERT INTO members VALUES ('5','admin',100,40000,40500)
INSERT INTO members VALUES ('6','admin',10000,40000,39000)
INSERT INTO members VALUES ('7','admin',10,12000,11500)
INSERT INTO members VALUES ('8','admin',10000,15500,14000)
INSERT INTO members VALUES ('9','admin',10000,18500,17000)
```

+
-
*

/ current
% current

EP24: គំរែង IN និង NOT IN

IN : ការណែនាំ ត្រូវបានពិនិត្យក្នុង ការ
តម្លៃឡើយ OR

The screenshot shows a MySQL Workbench interface. In the center, there is a results grid for a query. The grid has columns labeled 'ID', 'name', 'last_name', 'address', 'contact', and 'salary'. There are two rows of data:

ID	name	last_name	address	contact	salary
1	John	Doe	123 Main St	1234567890	50000
2	Susan	Smith	456 Elm St	1234567890	45000

A green arrow points from the text above the grid to the 'name' column header. Another green arrow points from the text below the grid to the 'salary' column header.

Text annotations in red and green are overlaid on the screenshot:

- Red text: "ទីនេះក្រោមចុះឈ្មោះ និងអាណាពាណិជ្ជកម្ម"
- Green text: "នាមឈ្មោះនៅក្នុង និងអាណាពាណិជ្ជកម្ម"
- Red text: "ទីនេះក្រោមចុះឈ្មោះ និងអាណាពាណិជ្ជកម្ម"

In the bottom right corner of the results grid, there is a status bar message: "No rows affected (0.000 sec)".

EP25 : ការស្ថិត BETWEEN និង AND

សំណើន៍លទ្ធផលរបស់ខ្លួន

30,000 ≤ salary ≤ 50,000

ID	Name	fname	address	contact	salary
1	S. Somboon	Tall	NO.1	NULL	40000
2	B. Narong	Leborg	NO.2	NULL	40000

Execution finished without errors.
Results: 2 rows returned in 1ms

EP26 : ផ្លាស់បន្ទាន់លើកវិញ្ញាន

The screenshot shows the Microsoft SQL Server Management Studio interface. In the top-left pane, there is a tree view of database objects. In the main pane, a query window displays the following T-SQL code:

```
SELECT * FROM Employee
    ORDER BY EmployeeID
    OFFSET 1 ROWS
    FOR XML AUTO, ELEMENTS;
```

A red circle highlights the word "ROW" in the "OFFSET 1 ROWS" clause. A red arrow points from this highlighted text towards the top of the image, where the title "ផ្លាស់បន្ទាន់លើកវិញ្ញាន" is written in red.

In the bottom-left pane, a results grid shows one row of data:

EmployeeID	EmployeeName	Address	City	State	PostalCode	Country
1	John Smith	123 Main Street	Anytown	CA	90000	USA

Below the results grid, the status bar displays the message: "Execution completed without errors. 1 rows contained in result set (Type: Int32)." To the right of the results grid, there is a "Results" pane which is currently empty.

EP27 : CASE ... WHEN

ນັ້ນຍິນ if... else

Screenshot of MySQL Workbench showing a query editor and results pane.

The query editor contains the following SQL code:

```
SELECT name, salary, department,
CASE
    WHEN salary >= 40000 THEN 'A'
    WHEN salary < 40000 THEN 'B'
    ELSE 'C'
END AS grade
FROM employees;
```

Annotations on the code:

- A blue circle highlights the `AS grade` part of the `CASE` statement.
- An arrow points from the `AS grade` label to the `grade` column in the results table.
- Handwritten text above the `AS grade` label says "ເຊື່ອຕົວນີ້" (example).
- Handwritten text next to the `AS grade` label says "ກູ່ມີຄວາມສຳເນົາ" (those who have a high salary).
- Handwritten text next to the `AS grade` label says "ກູ່ມີຄວາມບໍ່ສຳເນົາ" (those who have a low salary).
- Handwritten text next to the `AS grade` label says "ກູ່ທີ່ບໍ່ໄດ້ຮັບຂໍາມະນຸດ" (those who don't receive a salary).
- Handwritten text below the `AS grade` label says "2 = salary".
- Handwritten text above the `WHEN` clause says "WHEN ... THEN ..." with arrows pointing to each `THEN` keyword.
- Handwritten text below the first `THEN` keyword says "ຈິງຈານ" (true).
- Handwritten text below the second `THEN` keyword says "ແລ້ວຈະບໍ່" (then not).

The results pane shows the following data:

name	salary	department	grade
John Doe	40000	Customer Service	A
Susan Williams	40000	Customer Service	A
Mark Jones	30000	Customer Service	B
David Williams	30000	Customer Service	B
James Smith	18000	IT	C
Anna Davis	15000	IT	C
Robert Williams	15000	IT	C
Sarah Johnson	12000	IT	C

MySQL command history at the bottom:

```
mysql> SELECT name, salary, department,
    CASE
        WHEN salary >= 40000 THEN 'A'
        WHEN salary < 40000 THEN 'B'
        ELSE 'C'
    END AS grade
    FROM employees;
Query OK, 9 rows affected, 1 warning, 0 errors
Time: 0.000 sec
mysql> SELECT name, salary, department
    FROM employees;
```

EP28 : LIKE กับ CASE .. WHEN

The screenshot shows a DB Browser for SQLite interface with a SQL editor and a results grid.

SQL Editor:

```
SELECT name
CASE WHEN language LIKE 'עברית' THEN 1
WHEN language = 'English' THEN 2
WHEN language LIKE 'תיכנית' THEN 3
ELSE 0 END AS 'Priority'
FROM skills
```

Annotations in Hebrew:

- 3גנום (3גנום)
- 2גנום (2גנום)
- 1גנום (1גנום)
- AND so on (开始了)
- NULL (NULL)

Results Grid:

name	Priority
english	1
עברית	1
hebreu	2
english	1
עברית	3
hebreu	1
hebreu	0
hebreu	3

EP 29 : ຈົດລຸ່ມ Group By

ນັ້ນວ່າເຕັມໄສ ທີ່ຈີ່
ອີກຕົວຜົນ, ກິດເປົ້າ
ໃຫຍ່ໄລຍະວາງ

ອີກກົມໃນ function
ໃຫຍ່ໄນ້ຕີ່ກິດ

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, the tabs 'Database Structure', 'Business Data', 'SQL Progress', and 'Execute SQL' are visible. Below the tabs, there's a connection status for '127.0.0.1' and a dropdown menu for 'SELECT From: DBMS_XPLAN'. The main area contains a query window with the following code:

```
SELECT * FROM DBMS_XPLAN
```

Below the code, the results grid displays the following data:

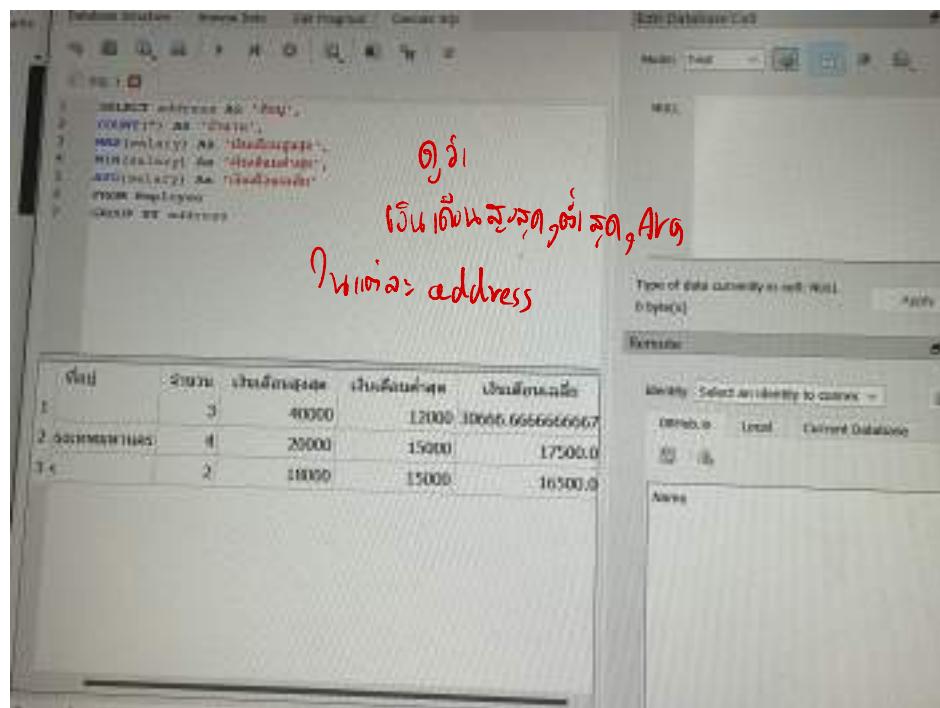
Name	Value
1. id	1
2. id	1
3. name	3
4. name	3
5. name	3
6. name	1
7. name	1
8. id	1

At the bottom of the results grid, there is a message: 'ORA-00933: SQL command not properly ended'. The status bar at the bottom of the screen also shows the message 'ORA-00933: SQL command not properly ended'.

EP30 : Group By และ function

① วิธี
เงินเดือนของแต่ละคน, Avg, Sum

หัวข้อ address



ที่อยู่	จำนวน	เดือนสูงสุด	เดือนต่ำสุด	เดือนเฉลี่ย
ถนนมหาชัย	4	20000	15000	17500.0
ถนนสุขุมวิท	2	18000	15000	16500.0

EP31 : (រៀបចំការណើនក្នុងមិន (Having) ពីចុះទូទៅ Group By

The screenshot shows a MySQL Workbench interface with a query editor and a results grid.

Query Editor:

```
SELECT address,
       COUNT(*) AS 'Count',
       SUM(salary) AS 'TotalSalary'
  FROM Employee
 GROUP BY address
 HAVING SUM(salary) > 20000;
```

Results Grid:

address	Count	TotalSalary
1. null	3	92000
2. 123 Main Street	4	20000
3. Bangkok	2	33000

Annotations:

- A red arrow points from the word "salary" in the query to the "TotalSalary" column header in the results grid.
- A red arrow points from the condition "salary > 20000" in the query to the "TotalSalary" column header in the results grid.
- A red circle highlights the value "20000" in the query editor.
- A red circle highlights the value "20000" in the "TotalSalary" column of the results grid.
- A red circle highlights the value "3" in the "Count" column of the results grid.

Message Bar:

Execution finished without errors.
Result: 3 rows returned in 0ms
At line 1
MySQL address:

EP 32 : Sub Query → ເນື້ນ Query ຢົກລວມ

ສອດໃຈລວມ

Main query
Sub query

Main query

Sub query

ເຖິງພົນໄດ້ຮັບຜົນກັບສູງ

id	name	address
1	John	123 Main Street
2	Alice	456 Elm Street

EP33 : សរបច្បាសនូវ Sub Query

The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a query editor window containing the following SQL code:

```
SELECT employee_name, address, salary FROM employees  
WHERE address =  
    (SELECT address FROM employees WHERE salary = 60000)
```

A red arrow points from the text "សរបច្បាស" (Sub Query) to the subquery part of the code. In the bottom-right pane, there is a results grid displaying the following data:

ID	employee_name	address	salary
1	R. Sathish	123 Main St.	65000
2	G. Andra	123 Main St.	68000

This screenshot shows a results grid from MySQL Workbench. The grid displays the same data as the previous screenshot, with two rows of results:

ID	employee_name	address	salary
1	R. Sathish	123 Main St.	65000
2	G. Andra	123 Main St.	68000

EP34 : การเข้ามาร่วมกัน (Join Table)

การเข้ามาร่วมกัน / การรวมตาราง (Join Table)



การนำเอาตาราง 2 ตารางมารวมกันโดยข้อมูลจะต้องมีส่วนที่เข้ามาร่วมกันได้ในทั้ง 2 ตาราง โดยคำนึงถึงการผ่าน Primary Key (PK) และ Foreign Key (FK)

ออกแบบตาราง

ตารางข้อมูลสินค้า (Products)

product_id	รหัสสินค้า (PK)
product_name	ชื่อสินค้า
price	ราคา
unit	หน่วย
type_id	ประเภทสินค้า (FK)
status	สถานะ

ตารางข้อมูลประเภทสินค้า (Categories)

id	รหัสประเภทสินค้า (PK)
name	ชื่อประเภทสินค้า

Key ๕๐ จับ Key ๔๐

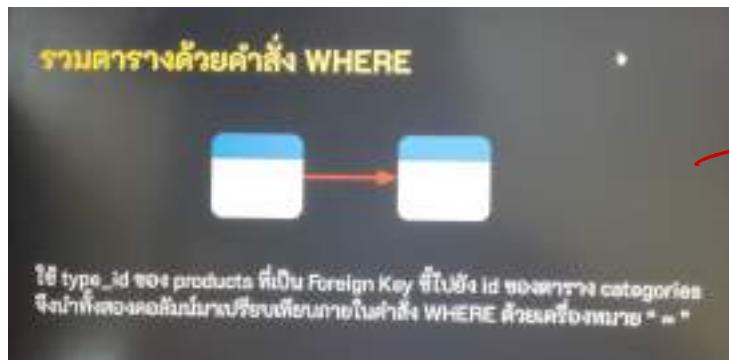
ตัวอย่างข้อมูลสินค้า (Products)

product_id	product_name	price	unit	type_id	status
P001	โน้ตบุ๊ค	1500.00	หน่วย	T001	1
P002	โทรศัพท์	3000.00	หน่วย	T001	2
P003	เมาส์	200.00	ชิ้น	T003	1
P004	เมาส์ไร้สาย	180.00	-	T003	2
P005	กล้อง	700.00	-	T004	1

ตัวอย่างข้อมูลประเภทสินค้า (Categories)

id	name
T001	คอมพิวเตอร์
T002	โทรศัพท์
T003	周邊
T004	อุปกรณ์

EP35 : รวมตารางด้วยคำสั่ง WHERE



[INNER JOIN]

▶ นำค่ามาร่วมกัน

ที่มีความสัมพันธ์กัน
มาบูรณาการ

SQL Editor:

```
SELECT products.product_name AS 'ผลิตภัณฑ์',
       products.type_id AS 'ประเภท',
       products.type AS 'ประเภท',
       categories.category_name AS 'ประเภท'
  FROM products
 INNER JOIN categories
    ON products.type_id = categories.id;
```

Results Grid:

ผลิตภัณฑ์	ประเภท	ประเภท
1 ลูกอม	150.0 THB	ผลิตภัณฑ์
2 น้ำผลไม้	200.0 THB	ผลิตภัณฑ์
3 เบเกอรี่	150.0 THB	ผลิตภัณฑ์
4 นมสด	250.0 THB	ผลิตภัณฑ์
5 ผักผลไม้	300.0 THB	ผลิตภัณฑ์
6 ไข่	250.0 THB	ผลิตภัณฑ์
7 ข้าว	200.0 THB	ผลิตภัณฑ์
8 กุ้งแม่น้ำ	300.0 THB	ผลิตภัณฑ์

Information: 21 rows selected, 0 rows inserted, 0 rows updated, 0 rows deleted.
Time: 0.000 sec (0.000 sec)

EP36: INNER JOIN

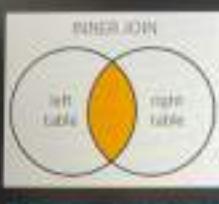
ความหมายของ JOIN



มีการรวมค่าที่เกี่ยวข้อง 2 ตารางเข้าด้วยกันโดยการเลือกหัวเม็ดที่มีความเหมือนกันในแต่ละ TABLE ที่เกี่ยวข้อง จึงเรียกว่า JOIN มีอยู่ 4 แบบ คือ LEFT JOIN , RIGHT JOIN , FULL JOIN และ INNER JOIN

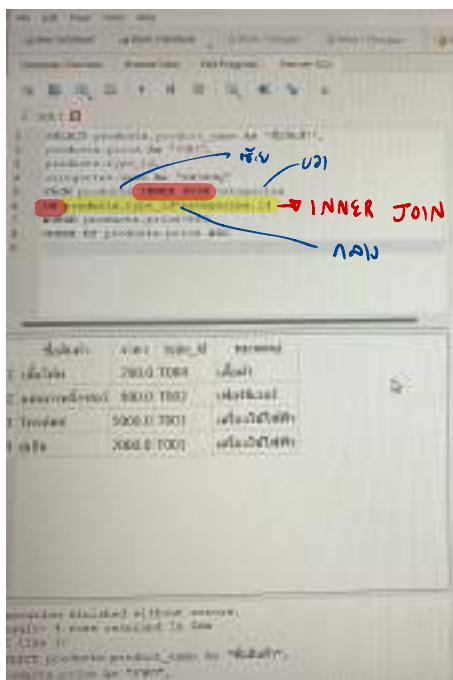
GROUP BY , HAVING , ORDER BY , LIMIT ก็จะมาjoin กัน JOIN ได้

INNER JOIN



```
SELECT column-name  
FROM table-name1 INNER JOIN table-name2  
ON column-name1 = column-name2  
WHERE condition
```

ใช้ INNER JOIN 2 ตารางที่มีหัวเม็ดเก็บไว้
คือเป็นรีตรูป (ถ้าเป็นเดียว)



```
SELECT product.product_name AS 'product_name'  
product_price FROM product  
product_type FROM product_type  
WHERE product.product_id = product_type.product_id;
```

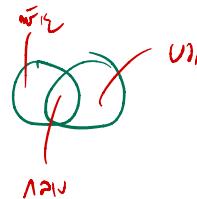
INNER JOIN

ผลลัพธ์:

product_name	price	type_id	product_id
กล้องดิจิตอล	2800.00	1	1
กล้องมือถือ	8800.00	2	2
โทรศัพท์	5900.00	3	3
โน๊ตบุ๊ค	20800.00	4	4

```
Rows: 4 | Total size: 0 | Elapsed: 0.000 sec | Threads: 1 / 4 | Status: Idle
```

```
mysql> SELECT product.product_name AS 'product_name'  
    product_price FROM product  
    product_type FROM product_type  
    WHERE product.product_id = product_type.product_id;  
    result set size: 4  
    rows affected: 4  
    status: OK  
    duration: 0.000 sec  
    lock time: 0.000 sec
```



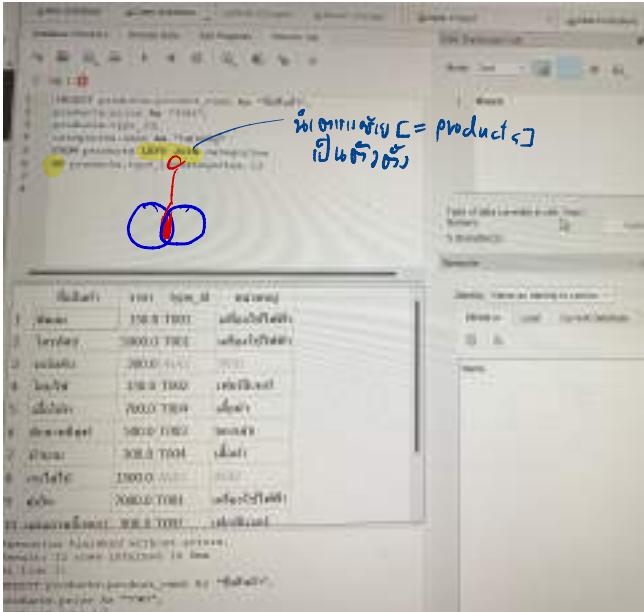
EP37 : LEFT JOIN

LEFT JOIN



```
SELECT * FROM table-name1 LEFT JOIN table-name2  
OR column-name1 = column-name2  
WHERE condition
```

จะเห็นว่ามีผลลัพธ์ 2 รายการใน left table ที่ไม่ได้มาจากการ JOIN คือ product_001 และ product_002 ที่ไม่มีค่าใน column ของ right table

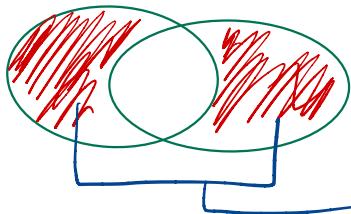


ผลลัพธ์ที่ได้คือ [= product_type]
เป็นตัวอย่าง

product_id	name	type_id	description
1	book	10001-10001	หนังสือไทย
2	pen	10002-10002	ปากกา
3	calculator	10003-10003	คалькуเลเตอร์
4	key	10004-10004	กุญแจ
5	clock	10005-10005	นาฬิกา
6	stationery	10006-10006	เครื่องเขียน
7	eraser	10007-10007	ลิปสติก
8	notebook	10008-10008	บันทึก
9	pen	10009-10009	ปากกาจี๊ด

```
SELECT * FROM products LEFT JOIN product_type ON products.product_id = product_type.product_id;
```

EP38 : OUTER JOIN



ມີ 2 ຊົນລວມທີ່ໄດ້ກຳນົດໄດ້ກຳນົດຈົກ

ຈົກມາຈົກ

Database Status Home Data Edit Progress Database SQL

10.1

```
1 SELECT products.product_name AS "ຄະດີນິກາ",
2 products.price AS "ລາຄາ",
3 products.type_id,
4 categories.name AS "ເມືອງ"
5 FROM products LEFT JOIN categories
6 ON products.type_id=categories.id
7 WHERE products.type_id='14' limit;
```

ມີສະບັບໃຈຕົວໃຫຍ່ NULL
(ຈົກມາຈົກລົບທີ່ມີ)

ຄະດີນິກາ	ລາຄາ	type_id	ນາມເມືອງ
1. morsky	300.0	NULL	NULL
2. emulajce	1500.0	NULL	NULL
3. vitalidns	400.0	NULL	NULL

Execution finished without errors.
result: 3 rows returned in 7ms.
line 1:
SELECT products.product_name AS "ຄະດີນິກາ",
products.price AS "ລາຄາ",
products.type_id,

EP39: RIGHT JOIN

RIGHT JOIN

```

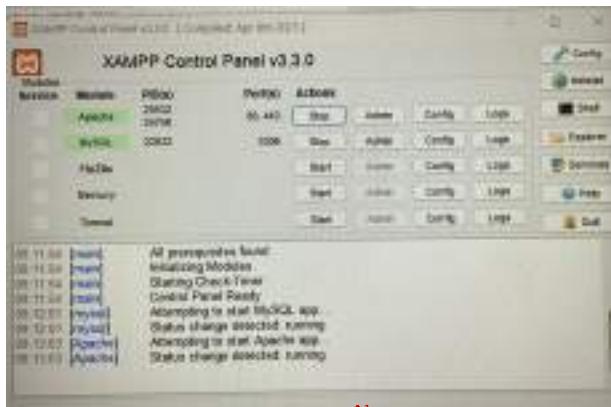
SELECT column-names
FROM table-name1 RIGHT JOIN table-name2
ON column-name1 = column-name2
WHERE condition
  
```

ແລະກົບຂອງມູນຄອງ 2 ດາວ່າງໄສ້ຈຳກັງທີ່ອື່ນ
ດ້ານຂາຍຂອງຄໍາສັ່ງ JOIN ເປັນດ້ວຍຕີ່ຈາກນັ້ນໄປ
ເລືອກຂອງມູນຄໍທີ່ອື່ນດ້ານຂາຍມີອື່ນຄໍາສັ່ງ JOIN ທີ່
ນີ້ມີຄື່ອງໃຫ້ກັນ

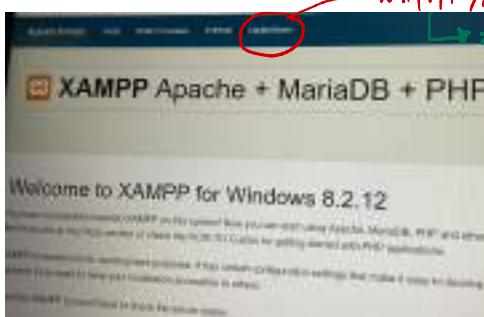
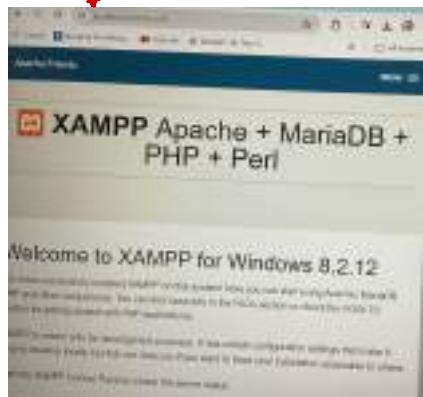
http://www.w3schools.com/sql/sql_right.asp

ກອກຮົບຕໍ່ເຊິ່ງ
RIGHT JOIN
TASUJ XAMPP
MySQL
CTURICRA SULL Lite
ນິກົນຂັບ)

(1) ໂັດ START ອີ່ MySQL , Apache ອີ່ XAMPP

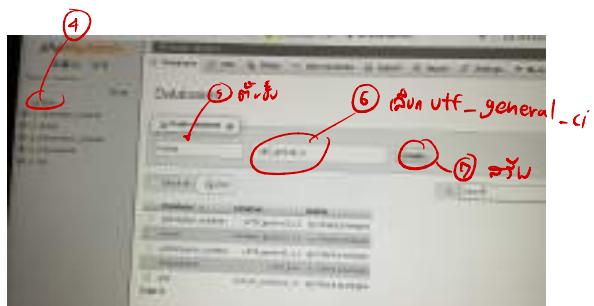


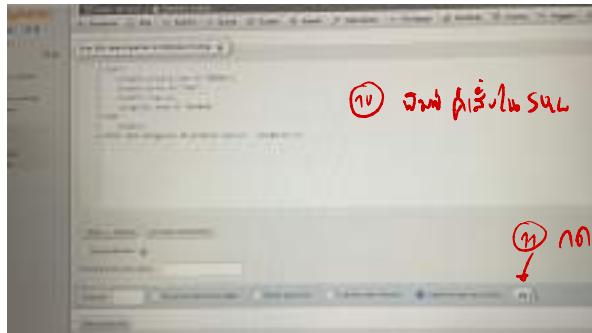
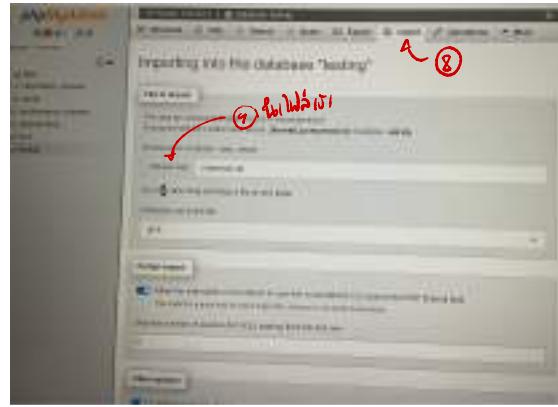
(2) ໂັດທີ່ Local host



c3) ອີ່ phpMyAdmin

ຄໍາຕົກລາງ ອີ່ MySQL





ໄດ້ ລົງທະບຽນ



EP 40 : FULL JOIN : ជីវិកសម្រាប់អនុ

[Right + Left = Full]

នៅលើកបែងចាយ XAMP

MySQL

(Universal Lite នៃពីរ)



Run SQL statement or Database object

1. SELECT products.product_name AS "Left",
product_descriptions.description AS "Middle",
categories.category_name AS "Right"
FROM products
INNER JOIN product_descriptions
ON products.product_id = product_descriptions.product_id;
2. SELECT products.product_name AS "Left",
product_descriptions.description AS "Middle",
categories.category_name AS "Right"
FROM products
INNER JOIN product_descriptions
ON products.product_id = product_descriptions.product_id
INNER JOIN categories
ON product_descriptions.product_id = categories.product_id;

Clear Run Get information Run

Run SQL query

SQL Editor SQL Editor History SQL Editor Recent SQL Editor Create Direct Key Cache

ទីវិទ្យាណាព័ត៌មាន

Left & Right

ទីវិទ្យាណាព័ត៌មាន

ឈ្មោះផ្លូវ	សម្រាប់	Type ID	ឈ្មោះ
ឈុន	150	T001	ឈុនសម្រាប់
ឈុន	1500	T001	ឈុនសម្រាប់
ឈុន	150	T002	ឈុនសម្រាប់
ឈុន	700	T004	ឈុន
ឈុនសម្រាប់	500	T022	ឈុនសម្រាប់
ឈុន	300	T004	ឈុន
ឈុន	2000	T001	ឈុនសម្រាប់
ឈុនសម្រាប់	900	T002	ឈុនសម្រាប់
ឈុន	NULL	NULL	ឈុនសម្រាប់
ឈុន	NULL	NULL	ឈុនសម្រាប់
ឈុន	300	NULL	
ឈុន	1500	NULL	
ឈុនសម្រាប់	400	NULL	