

วศ 5012346

การออกแบบฐานข้อมูล และระบบสารสนเทศ

EN 5012346 Database and Information System Design

SS1: SQL

วิศวกรรมการสื่อสารและสารสนเทศ

คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏเทพสตรี

EN 5012346 Database and Information System Design

วิชาที่ต้องศึกษาก่อน : ไม่มี

แบบจำลองระบบฐานข้อมูล ทฤษฎีรูปแบบบรรทัดฐานและการออกแบบฐานข้อมูล ภาษาสอบถามพจนานุกรมข้อมูล การควบคุมภาวะพร้อมกัน การถูกสืบจากความล้มเหลว การวิเคราะห์ระบบสารสนเทศ ขั้นตอนต่างๆ ของการพัฒนาระบบ การสร้างแผนภูมิการไหลของข้อมูล การออกแบบส่วนติดต่อกับผู้ใช้ ส่วนเก็บข้อมูล การวิเคราะห์และการออกแบบระบบเชิงวัตถุ จดให้มีการปฏิบัติการที่สอดคล้องกับเนื้อหาวิชา

ระหว่างภาค 70 คะแนน

Contents

สอบเก็บคะแนน SQL ¹	30 คะแนน	SS1: SQL (1-5)
สอบเก็บคะแนน database ²	30 คะแนน	SS2: Database (6-10)
งานที่ได้รับมอบหมาย	10 คะแนน	SS3: Information System Design (11-15)

ปลายภาค 30 คะแนน

ต่ำกว่า 50 คะแนน -> ไม่ผ่าน (F)

¹ สอบสัปดาห์ที่ 6 (นัดสอบนอกตารางเรียน)² สอบสัปดาห์ที่ 11 (นัดสอบนอกตารางเรียน)



SQL (part I)

- Introduction
- Database
- MySQL
- Query-I
 - SELECT..FROM..WHERE
 - ORDER BY
 - LIMIT
 - DISTINCT
 - LIKE
- Query-II
 - Sub Query
 - Function
 - GROUP BY
 - Join
 - Left-Join
 - Export
 - Excel
 - SQL (Back up)

สาขาวิชาวิศวกรรมการสื่อสารและสารสนเทศ
- 1/79 คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏเทพสตรี

1. Introduction

1.1 Database

C : Create

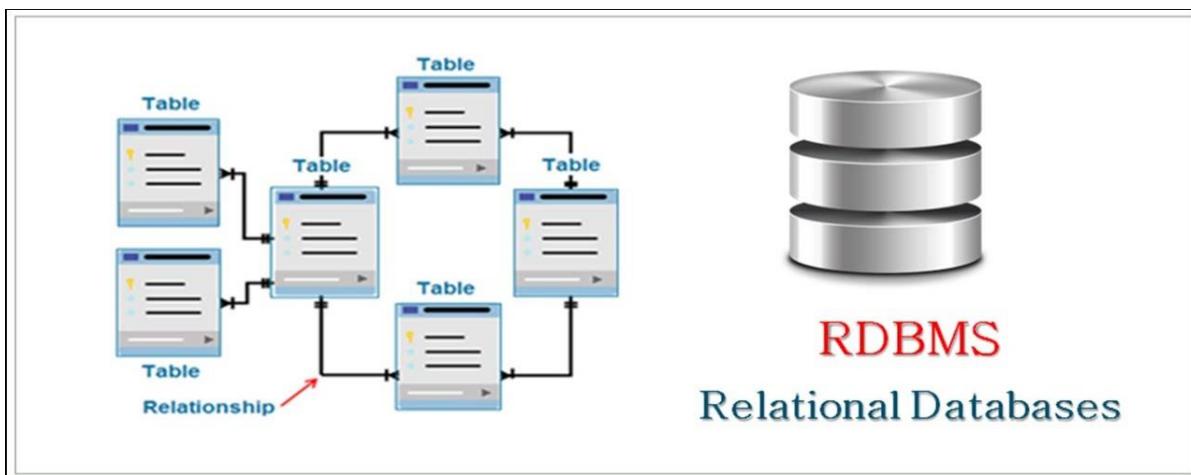
R : Read

U : Update

D : Delete

- 3/79 -

1.1 Database



Relational Database Management System

www.learncomputerscienceonline.com

- 4/79 -

1.2 MySQL



Website: <https://theepsatri.com/phpMyAdmin>

Username: data

Password: Data@2024

- 5/79 -

1.2 MySQL

Recent | Favorites

data_lab
Information_schema

Database

General settings

Appearance settings

Language English

Theme pmahomme

Font size 82%

More settings

Database server

- Server: Localhost via UNIX socket
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.4.10-MariaDB - MariaDB Server
- Protocol version: 10
- User: data@localhost
- Server charset: cp1252 West European (latin1)

Web server

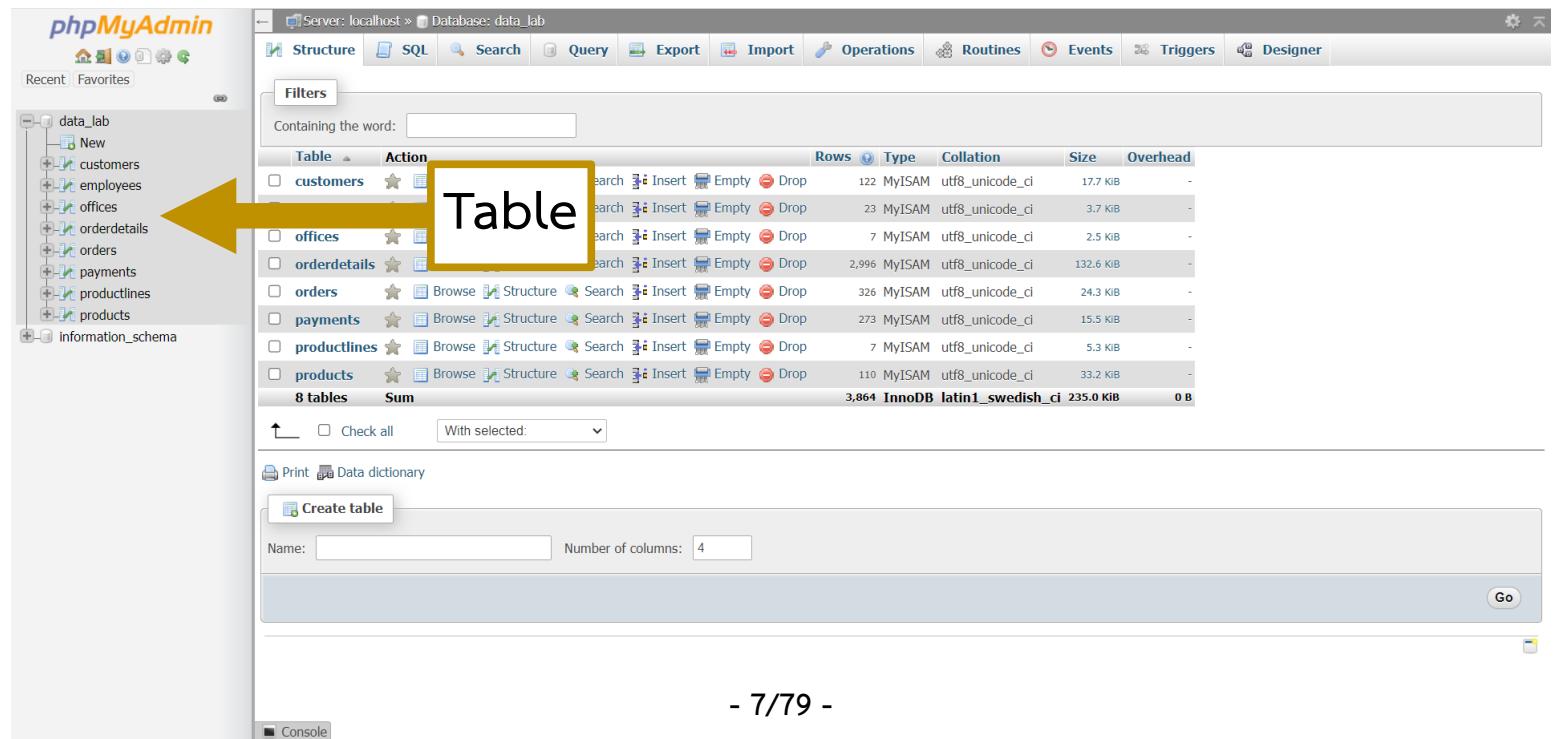
- Apache/2
- Database client version: libmysql - mysqld 5.0.12-dev - 20150407 - \$Id: 7cc7cc96e675fd72e5cf0f267f48e167c2abb2 \$
- PHP extension: mysqli curl mbstring
- PHP version: 7.3.12

phpMyAdmin

- Version information: 4.9.2
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes
- License

- 6/79 -

1.2 MySQL



phpMyAdmin

Server: localhost » Database: data_lab

Structure SQL Search Query Export Import Operations Routines Events Triggers Designer

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
customers	Insert Empty Drop	122	MyISAM	utf8_unicode_ci	17.7 kB	-
offices	Insert Empty Drop	23	MyISAM	utf8_unicode_ci	3.7 kB	-
orderdetails	Insert Empty Drop	2,996	MyISAM	utf8_unicode_ci	132.6 kB	-
orders	Insert Empty Drop	326	MyISAM	utf8_unicode_ci	24.3 kB	-
payments	Insert Empty Drop	273	MyISAM	utf8_unicode_ci	15.5 kB	-
productlines	Insert Empty Drop	7	MyISAM	utf8_unicode_ci	5.3 kB	-
products	Insert Empty Drop	110	MyISAM	utf8_unicode_ci	33.2 kB	-
8 tables	Sum	3,864	InnoDB	latin1_swedish_ci	235.0 kB	0 B

Print Data dictionary

Create table

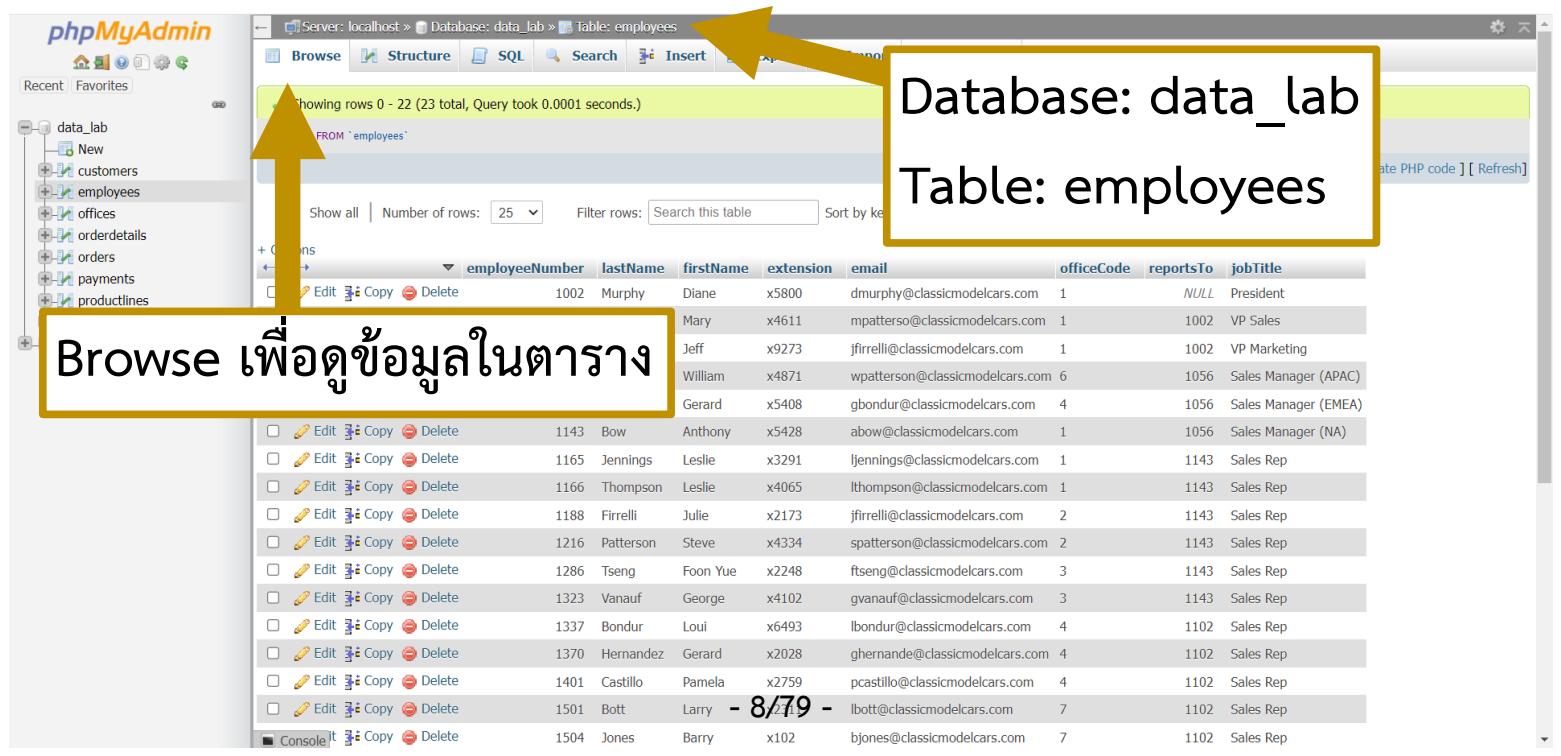
Name: Number of columns: 4

Go

Console

- 7/79 -

1.2 MySQL



phpMyAdmin

Recent Favorites

Server: localhost » Database: data_lab » Table: employees

Browse Structure SQL Search Insert

Showing rows 0 - 22 (23 total, Query took 0.0001 seconds.)

FROM `employees`

Show all Number of rows: 25 Filter rows: Search this table Sort by key

+ Columns

	employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle
1002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com	1	NULL	President	
Mary	x4611	mpatterso@classicmodelcars.com	1	1002	VP Sales			
Jeff	x9273	jfirrelli@classicmodelcars.com	1	1002	VP Marketing			
William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)			
Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sales Manager (EMEA)			
Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)			
Bow	1143	1143	1143	1143	Sales Rep			
Leslie	1165	Jennings	Le Jennings	1	1143	Sales Rep		
Thompson	1166	Thompson	Leslie	x4065	1	1143	Sales Rep	
Firrelli	1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep
Patterson	1216	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep
Tseng	1286	Tseng	Foon Yue	x2248	ftsing@classicmodelcars.com	3	1143	Sales Rep
Vanauf	1323	Vanauf	George	x4102	gvanauf@classicmodelcars.com	3	1143	Sales Rep
Bondur	1337	Bondur	Loui	x6493	lbondur@classicmodelcars.com	4	1102	Sales Rep
Hernandez	1370	Hernandez	Gerard	x2028	ghernande@classicmodelcars.com	4	1102	Sales Rep
Castillo	1401	Castillo	Pamela	x2759	pcastillo@classicmodelcars.com	4	1102	Sales Rep
Bott	1501	Bott	Larry	- 8/79 -	lbott@classicmodelcars.com	7	1102	Sales Rep
Jones	1504	Jones	Barry	x102	bjones@classicmodelcars.com	7	1102	Sales Rep

Console

Browse เพื่อดูข้อมูลในตาราง

Database: data_lab
Table: employees

- 8/79 -

1.2 MySQL

The screenshot shows the phpMyAdmin interface for a MySQL database named 'data'. The left sidebar lists databases like 'data_lab', 'customers', 'employees', etc. The main area shows the 'Structure' tab for the 'customers' table. A yellow arrow points from the text 'Structure เพื่อดูโครงสร้างของตาราง' to the table definition area. The table has 13 columns:

#	Name	Type	Collation	Action
1	customerNumber	int(11)	None	Change Drop More
2	customerName	varchar(50)	utf8_unicode_ci	Change Drop More
3	contactLastName	varchar(50)	utf8_unicode_ci	Change Drop More
4	contactFirstName	varchar(50)	utf8_unicode_ci	Change Drop More
5	phone	varchar(50)	utf8_unicode_ci	Change Drop More
6	addressLine1	varchar(50)	utf8_unicode_ci	Change Drop More
7	addressLine2	varchar(50)	utf8_unicode_ci	Change Drop More
8	city	varchar(50)	utf8_unicode_ci	Change Drop More
9	state	varchar(50)	utf8_unicode_ci	Change Drop More
10	postalCode	varchar(15)	utf8_unicode_ci	Change Drop More
11	country	varchar(50)	utf8_unicode_ci	Change Drop More
12	salesRepEmployeeNumber	int(11)	Yes NULL	Change Drop More
13	creditLimit	double	Yes NULL	Change Drop More

Below the table, there are buttons for 'Check all', 'With selected:', and various table operations like 'Print', 'Propose table structure', 'Move columns', 'Normalize', 'Add', and 'Indexes'.

1.2 MySQL

The screenshot shows the phpMyAdmin interface for the 'data_lab' database. The left sidebar lists databases like 'data_lab', 'customers', 'employees', etc. The main area shows the 'SQL' tab for the 'customers' table. A yellow arrow points from the text 'SQL เพื่อเขียนคำสั่ง query' to the SQL query editor. The query entered is:

```
1 SELECT * FROM `customers` WHERE 1
```

The right side of the screen shows a list of columns for the 'customers' table:

- customerNumber
- customerName
- contactLastName
- contactFirstName
- phone
- addressLine1
- addressLine2
- city
- state
- postalCode
- country
- salesRepEmployeeNumber
- creditLimit

Below the query editor, there are buttons for 'SELECT*', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Clear', 'Format', and 'Get auto-saved query'. There are also checkboxes for 'Bind parameters', 'Delimiter', 'Show this query here again', 'Retain query box', 'Rollback when finished', and 'Enable foreign key checks'. The 'Go' button is at the bottom right.

2 . Query- I

- 11/79 -

2 . 1 SELECT . . FROM . . WHERE

```
SELECT *
FROM   customers
```

- 12/79 -

2.1 SELECT . . FROM . . WHERE

The screenshot shows the phpMyAdmin interface for the 'customers' table in the 'data_lab' database. The 'SQL' tab is selected. A red box highlights the SQL query: 'SELECT * FROM customers'. Another red box highlights the text 'SQL เพื่อเขียนคำสั่ง query' (SQL to write a query) with an arrow pointing to the query text. The right panel shows table columns: customerNumber, customerName, addressLine1, addressLine2, city, state, postalCode, country, salesRepEmployeeNumber, and creditLimit.

```
1 SELECT *
2 FROM customers
3
```

SQL เพื่อเขียนคำสั่ง query

2.1 SELECT . . FROM . . WHERE

The screenshot shows the phpMyAdmin interface displaying the results of the query 'SELECT * FROM customers'. The results table has 122 rows. A red box highlights the status bar at the bottom of the results table showing '- 14/79 -'.

customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country
103	Atelier graphique	Schmitt	Carine	40.32.2555	54, rue Royale	NULL	Nantes	NULL	44000	France
112	Signal Gift Stores	King	Jean	7025551838	8489 Strong St.	NULL	Las Vegas	NV	83030	USA
114	Australian Collectors, Co.	Ferguson	Peter	03 9520 4555	636 St Kilda Road	Level 3	Melbourne	Victoria	3004	Australia
119	La Rochelle Gifts	Labrune	Janine	40.67.8555	67, rue des Cincinante Otages	NULL	Nantes	NULL	44000	France
121	Baane Mini Imports	Bergulfsen	Jonas	07-98 9555	Erling Skakkes gate 78	NULL	Stavarn	NULL	4110	Norway
124	Mini Gifts Distributors Ltd.	Nelson	Susan	4155551450	5677 Strong St.	NULL	San Rafael	CA	97562	USA
125	Havel & Zbyszek Co.	Piestrzewicz	Zbyszek	(26) 642-7555	ul. Filtrowa 68	NULL	Warszawa	NULL	01-012	Poland
128	Blauer See Auto, Co.	Keitel	Roland	+49 69 66 90 2555	Lyonerstr. 34	NULL	Frankfurt	NULL	60528	Germany
129	Mini Wheels Co.	Murphy	Julie	6505555787	5557 North Pendale Street	NULL	San Francisco	CA	94217	USA
131	Land of Toys Inc.	Lee	Kwai	2125557818	897 Long Airport Avenue	NULL	NYC	NY	10022	USA
141	Euro+ Shopping Channel	Freyre	Diego	(91) 555 94 44	C/ Moralzarzal, 86	NULL	Madrid	NULL	28034	Spain
144	Volvo Model Replicas, Co	Berglund	Christina	0921-12 3555	Berguvsvagen 8	NULL	Luleå	NULL	S-958 22	Sweden
145	Danish Wholesale	Peterson	Ivitta	31 12 3555	Vinbaeltet 34	NULL	Kopenhagen	NULL	1734	Denmark

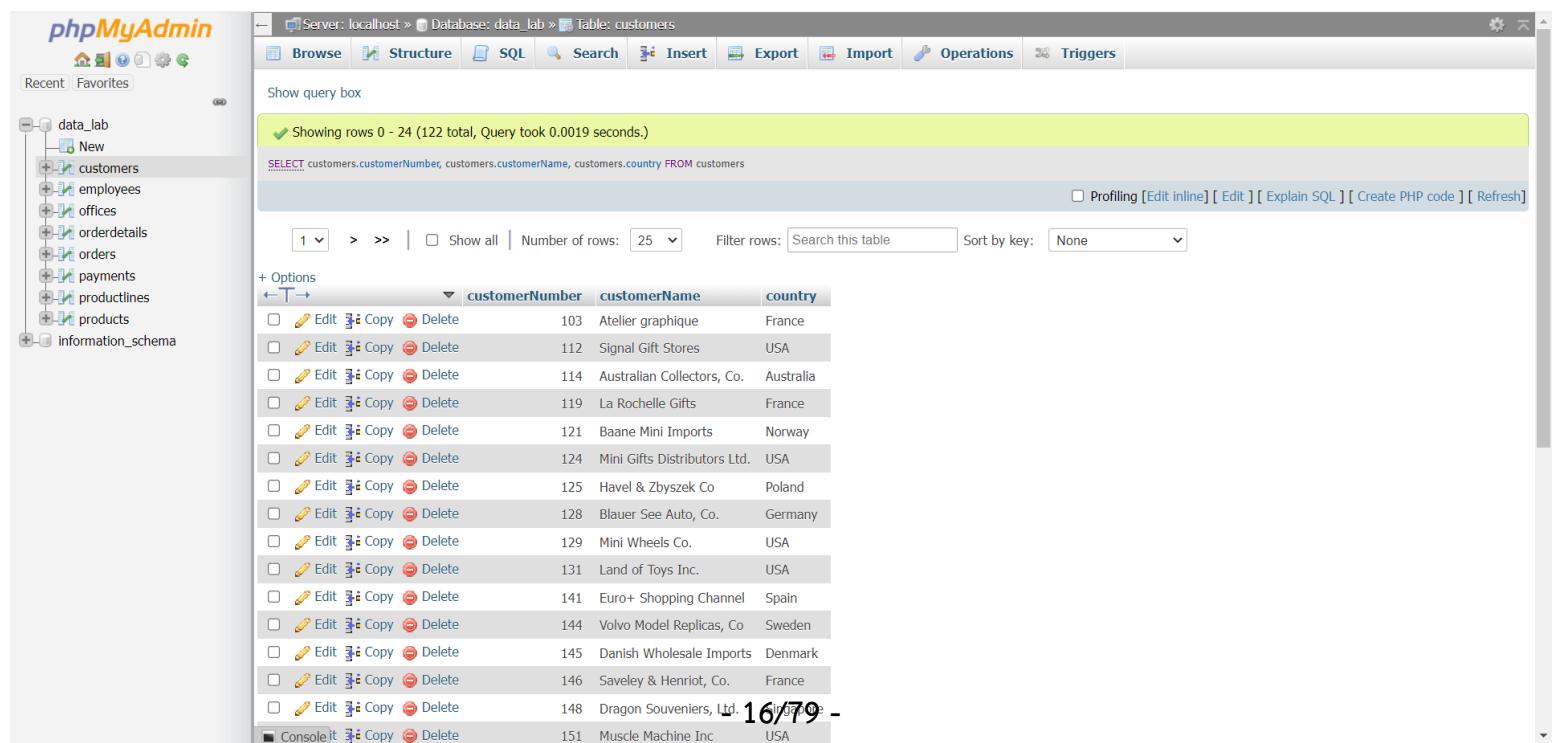
- 14/79 -

2.1 SELECT . . FROM . . WHERE

SELECT customers.customerNumber,
customers.customerName,
customers.country
FROM customers

- 15/79 -

2.1 SELECT . . FROM . . WHERE



The screenshot shows the phpMyAdmin interface with the following details:

- Server:** localhost
- Database:** data_lab
- Table:** customers
- Query Result:**
 - Showing rows 0 - 24 (122 total, Query took 0.0019 seconds.)
 - SQL: `SELECT customers.customerNumber, customers.customerName, customers.country FROM customers`
 - Buttons: Profiling, Edit inline, Edit, Explain SQL, Create PHP code, Refresh
 - Options: Show all, Number of rows: 25, Filter rows: Search this table, Sort by key: None
- Table Data:** A grid of customer information with columns: customerNumber, customerName, country.

	customerNumber	customerName	country
<input type="checkbox"/> Edit Copy Delete	103	Atelier graphique	France
<input type="checkbox"/> Edit Copy Delete	112	Signal Gift Stores	USA
<input type="checkbox"/> Edit Copy Delete	114	Australian Collectors, Co.	Australia
<input type="checkbox"/> Edit Copy Delete	119	La Rochelle Gifts	France
<input type="checkbox"/> Edit Copy Delete	121	Baane Mini Imports	Norway
<input type="checkbox"/> Edit Copy Delete	124	Mini Gifts Distributors Ltd.	USA
<input type="checkbox"/> Edit Copy Delete	125	Havel & Zbyszek Co	Poland
<input type="checkbox"/> Edit Copy Delete	128	Blauer See Auto, Co.	Germany
<input type="checkbox"/> Edit Copy Delete	129	Mini Wheels Co.	USA
<input type="checkbox"/> Edit Copy Delete	131	Land of Toys Inc.	USA
<input type="checkbox"/> Edit Copy Delete	141	Euro+ Shopping Channel	Spain
<input type="checkbox"/> Edit Copy Delete	144	Volvo Model Replicas, Co	Sweden
<input type="checkbox"/> Edit Copy Delete	145	Danish Wholesale Imports	Denmark
<input type="checkbox"/> Edit Copy Delete	146	Savely & Henriot, Co.	France
<input type="checkbox"/> Edit Copy Delete	148	Dragon Souvenirs, Ltd.	United Kingdom
<input type="checkbox"/> Edit Copy Delete	151	Muscle Machine Inc	USA

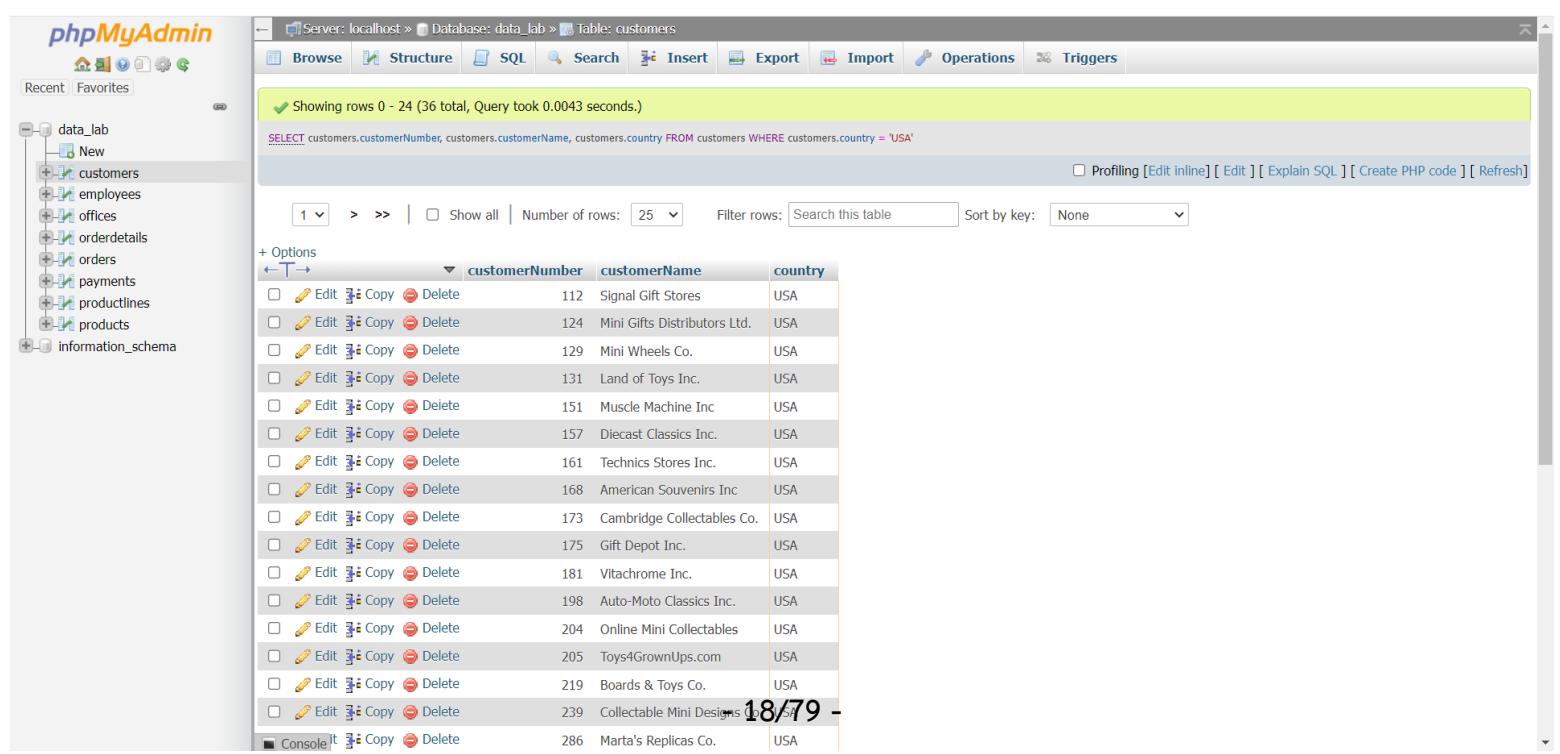
- 16/79 -

2.1 SELECT . . FROM . . WHERE

SELECT customers.customerNumber,
customers.customerName,
customers.country
FROM customers
WHERE customers.country = 'USA'

- 17/79 -

2.1 SELECT . . FROM . . WHERE



The screenshot shows the phpMyAdmin interface with the following details:

- Server:** localhost
- Database:** data_lab
- Table:** customers
- Query Result:** Showing rows 0 - 24 (36 total, Query took 0.0043 seconds.)
- SQL Query:** SELECT customers.customerNumber, customers.customerName, customers.country FROM customers WHERE customers.country = 'USA'
- Table Headers:** customerNumber, customerName, country
- Table Data:** A list of 24 rows where the country is USA. The first few rows are:

customerNumber	customerName	country
112	Signal Gift Stores	USA
124	Mini Gifts Distributors Ltd.	USA
129	Mini Wheels Co.	USA
131	Land of Toys Inc.	USA
151	Muscle Machine Inc	USA
157	Diecast Classics Inc.	USA
161	Technics Stores Inc.	USA
168	American Souvenirs Inc	USA
173	Cambridge Collectables Co.	USA
175	Gift Depot Inc.	USA
181	Vitachrome Inc.	USA
198	Auto-Moto Classics Inc.	USA
204	Online Mini Collectables	USA
205	Toys4GrownUps.com	USA
219	Bands & Toys Co.	USA
239	Collectable Mini Designs	USA
286	Marta's Replicas Co.	USA
- Page Number:** 18/79

2.1 SELECT . . FROM . . WHERE

ให้ query หาข้อมูล customers ประกอบไปด้วย customerNumber,
customerName และ country โดยแสดงเฉพาะลูกค้าในประเทศ Norway

- 19/79 -

2.1 SELECT . . FROM . . WHERE

ให้ query หาข้อมูล customers ประกอบไปด้วย customerNumber,
customerName และ country โดยแสดงเฉพาะลูกค้าในกลุ่มประเทศ Norway,
Denmark และ Finland

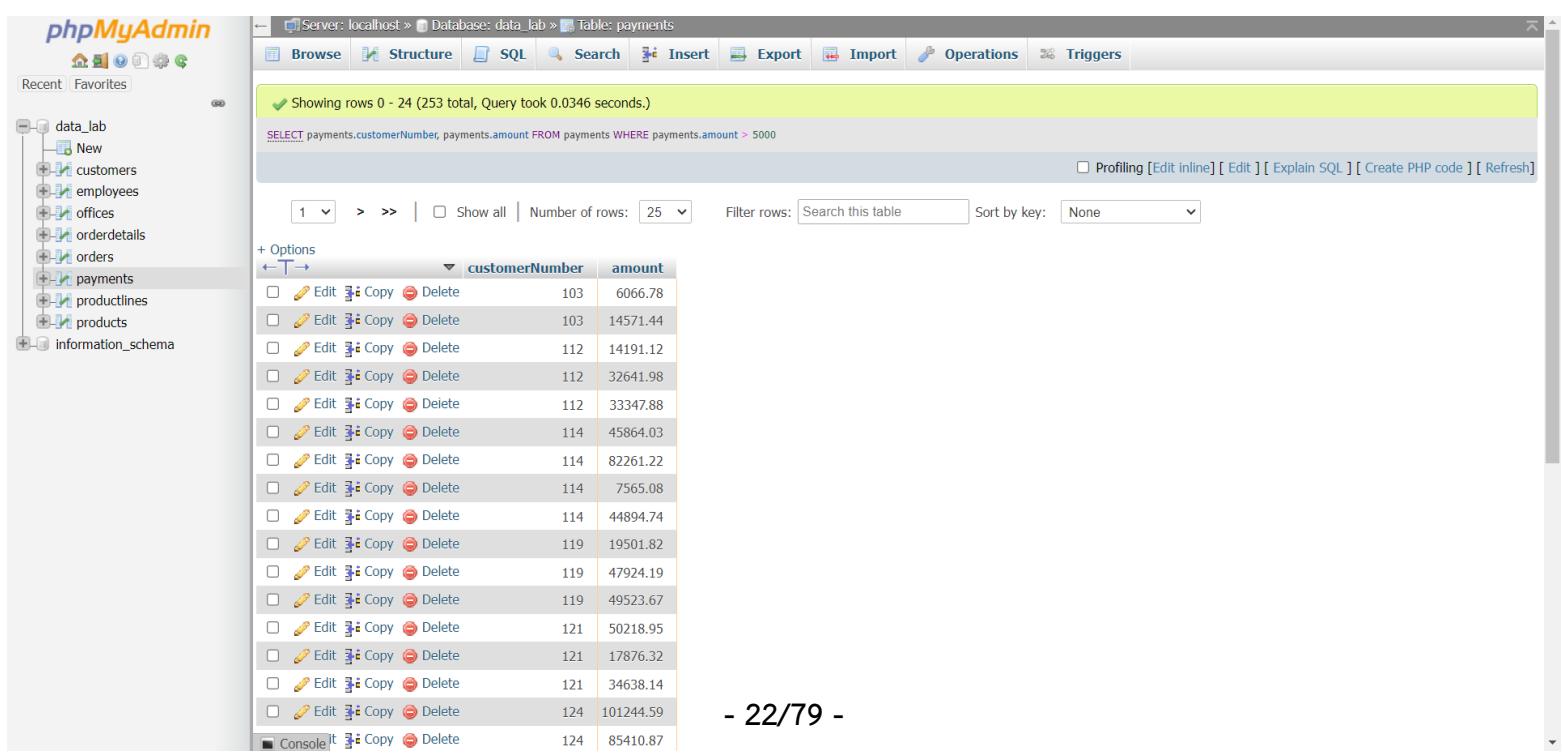
- 20/79 -

2.1 SELECT . . FROM . . WHERE

```
SELECT payments.customerNumber,  
       payments.amount  
FROM payments  
WHERE payments.amount > 5000
```

- 21/79 -

2.1 SELECT . . FROM . . WHERE



The screenshot shows the phpMyAdmin interface with the following details:

- Server:** localhost
- Database:** data_lab
- Table:** payments
- Query Result:** Showing rows 0 - 24 (253 total, Query took 0.0346 seconds.)
- SQL Query:** SELECT payments.customerNumber, payments.amount FROM payments WHERE payments.amount > 5000
- Table Headers:** customerNumber, amount
- Table Data:** A list of 24 rows where the amount is greater than 5000.

customerNumber	amount
103	6066.78
103	14571.44
112	14191.12
112	32641.98
112	33347.88
114	45864.03
114	82261.22
114	7565.08
114	44894.74
119	19501.82
119	47924.19
119	49523.67
121	50218.95
121	17876.32
121	34638.14
124	101244.59
124	85410.87

- 22/79 -

2.1 SELECT . . FROM . . WHERE

ให้ query หาข้อมูล orders ประกอบไปด้วย orderNumber, status และ customerNumber โดยแสดงเฉพาะข้อมูล customerNumber รหัส 124

- 23/79 -

2.1 SELECT . . FROM . . WHERE

ให้ query หาข้อมูล orders ประกอบไปด้วย orderNumber, status และ customerNumber โดยแสดงเฉพาะข้อมูล customerNumber รหัส 103, 112 และ 114

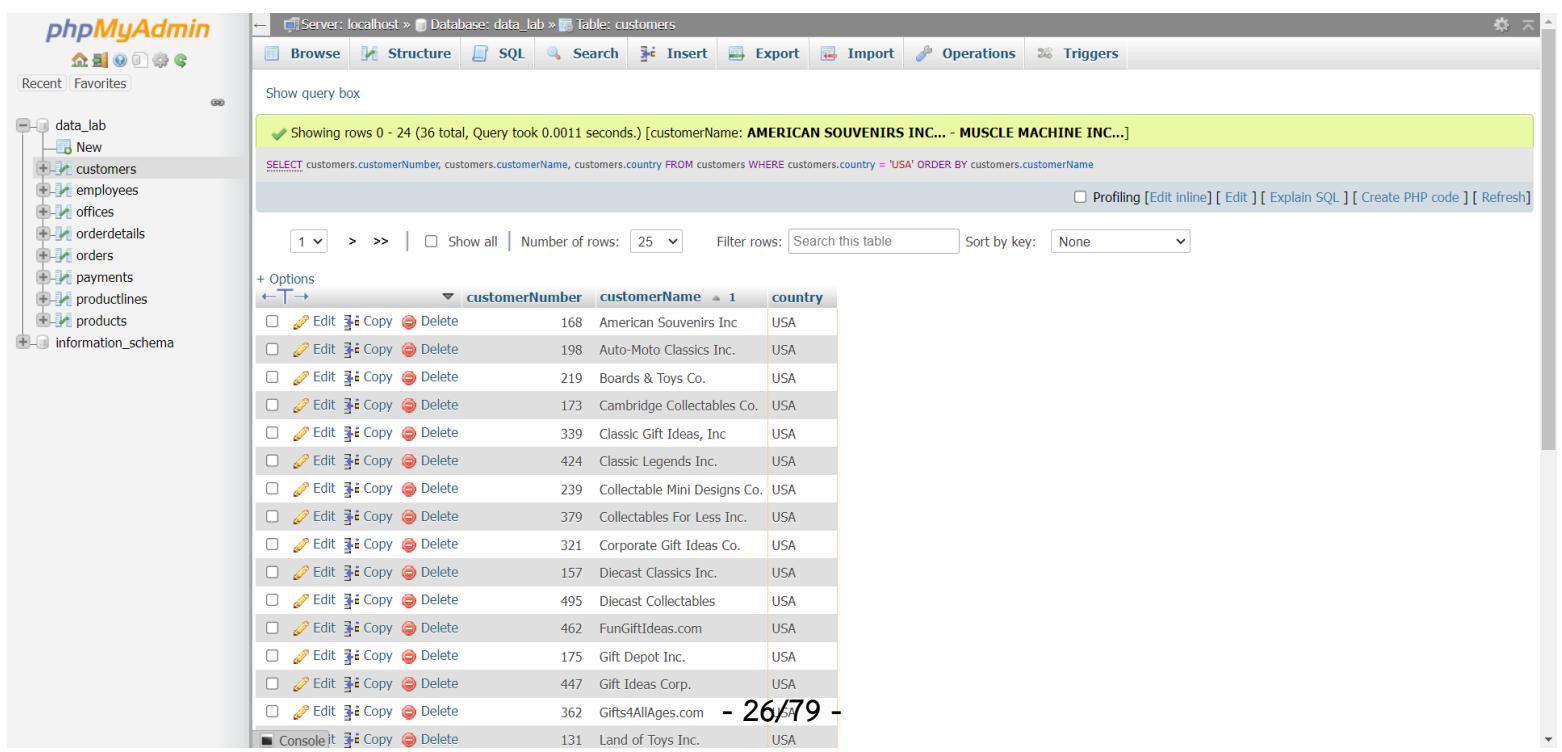
- 24/79 -

2.2 ORDER BY

```
SELECT customers.customerNumber,  
customers.customerName,  
customers.country  
FROM customers  
WHERE customers.country = 'USA'  
ORDER BY customers.customerName
```

- 25/79 -

2.2 ORDER BY



The screenshot shows the phpMyAdmin interface with the following details:

- Server:** localhost
- Database:** data_lab
- Table:** customers
- Query Result:** Rows 0 - 24 (36 total, Query took 0.0011 seconds.) [customerName: AMERICAN SOUVENIRS INC... - MUSCLE MACHINE INC...]
- SQL Query:** SELECT customers.customerNumber, customers.customerName, customers.country FROM customers WHERE customers.country = 'USA' ORDER BY customers.customerName
- Options:** Profiling, Edit inline, Edit, Explain SQL, Create PHP code, Refresh
- Table Headers:** customerNumber, customerName, country
- Table Data:** A list of 36 rows from the customers table where country is 'USA', ordered by customerName. The first few rows are:

customerNumber	customerName	country
168	American Souvenirs Inc.	USA
198	Auto-Moto Classics Inc.	USA
219	Bands & Toys Co.	USA
173	Cambridge Collectables Co.	USA
339	Classic Gift Ideas, Inc	USA
424	Classic Legends Inc.	USA
239	Collectable Mini Designs Co.	USA
379	Collectables For Less Inc.	USA
321	Corporate Gift Ideas Co.	USA
157	Diecast Classics Inc.	USA
495	Diecast Collectables	USA
462	FunGiftIdeas.com	USA
175	Gift Depot Inc.	USA
447	Gift Ideas Corp.	USA
362	Gifts4AllAges.com	USA
131	Land of Toys Inc.	USA

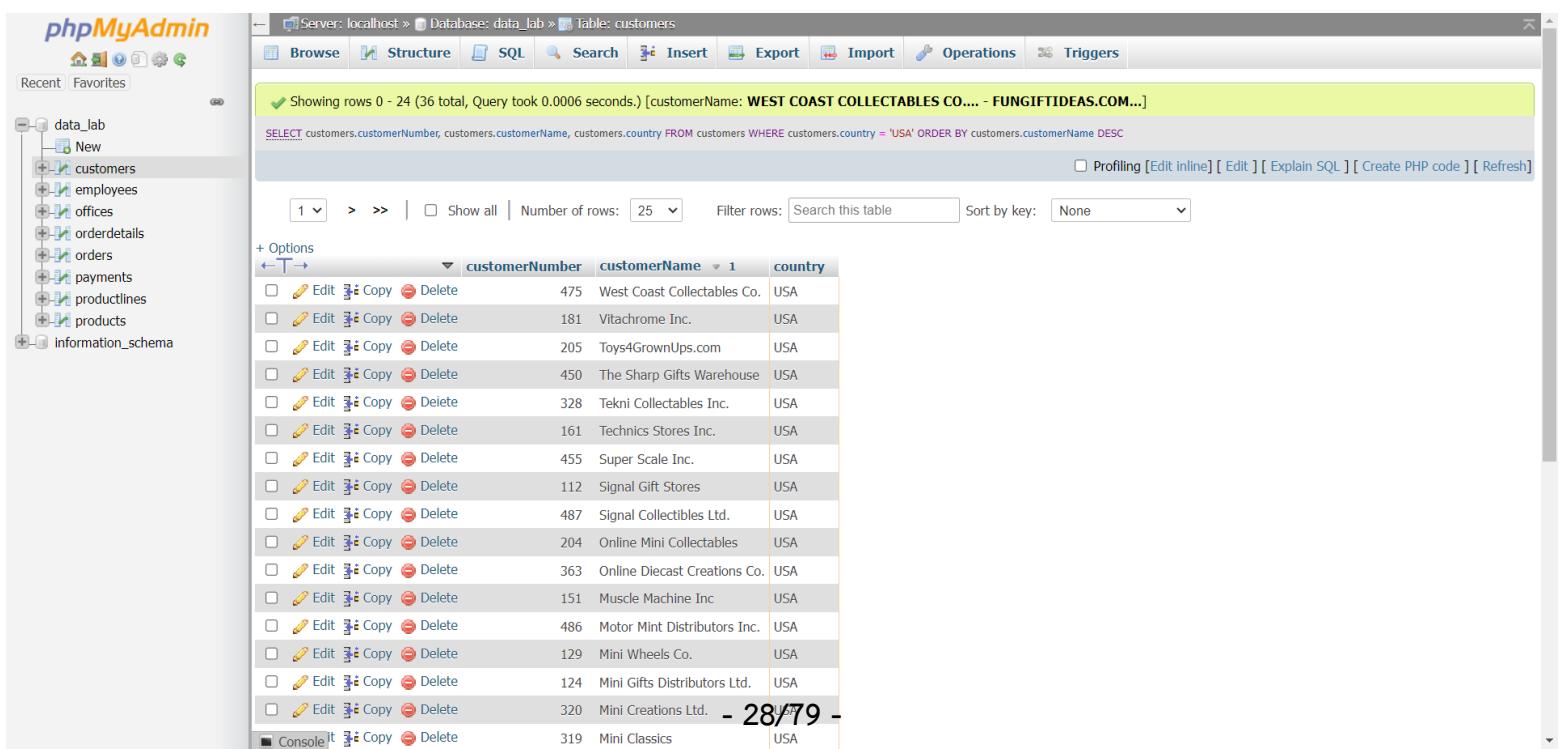
- 26/79 -

2.2 ORDER BY

```
SELECT customers.customerNumber,  
       customers.customerName,  
       customers.country  
FROM customers  
WHERE customers.country = 'USA'  
ORDER BY customers.customerName DESC
```

- 27/79 -

2.2 ORDER BY



The screenshot shows the phpMyAdmin interface with the following details:

- Server:** localhost
- Database:** data_lab
- Table:** customers
- Query Result:** Showing rows 0 - 24 (36 total, Query took 0.0006 seconds.) [customerName: WEST COAST COLLECTABLES CO..... - FUNGIFTIDEAS.COM...]
- SQL Query:** SELECT customers.customerNumber, customers.customerName, customers.country FROM customers WHERE customers.country = 'USA' ORDER BY customers.customerName DESC
- Options:** Profiling, Edit, Explain SQL, Create PHP code, Refresh
- Table Headers:** customerNumber, customerName, country
- Table Data:** A list of 24 rows from the customers table where country is 'USA', ordered by customerName in descending order.

customerNumber	customerName	country
475	West Coast Collectables Co.	USA
181	Vitachrome Inc.	USA
205	Toys4GrownUps.com	USA
450	The Sharp Gifts Warehouse	USA
328	Tekni Collectables Inc.	USA
161	Technics Stores Inc.	USA
455	Super Scale Inc.	USA
112	Signal Gift Stores	USA
487	Signal Collectibles Ltd.	USA
204	Online Mini Collectables	USA
363	Online Diecast Creations Co.	USA
151	Muscle Machine Inc	USA
486	Motor Mint Distributors Inc.	USA
129	Mini Wheels Co.	USA
124	Mini Gifts Distributors Ltd.	USA
320	Mini Creations Ltd.	USA
319	Mini Classics	USA

- 28/79 -

2.2 ORDER BY

ให้ query หาข้อมูล payments ประกอบไปด้วย customerNumber และ amount โดยเรียงลำดับตามยอด amount จากมากไปน้อย

- 29/79 -

2.2 ORDER BY

ให้ query หาข้อมูล employees ประกอบไปด้วย employeeNumber, firstName และ lastName โดยเรียงลำดับตาม employeeNumber จากมากไปน้อย

- 30/79 -

2 . 3 LIMIT

```
SELECT payments.customerNumber,  
       payments.amount  
FROM   payments  
ORDER BY payments.amount DESC  
LIMIT  0,3
```

- 31/79 -

2 . 3 LIMIT

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** localhost
- Database:** data_lab
- Table:** payments
- Query:**

```
1 SELECT payments.customerNumber, payments.amount
2 FROM payments
3 ORDER BY payments.amount DESC
4 LIMIT 0,3
```
- Results:** Showing rows 0 - 2 (3 total, Query took 0.0001 seconds.) [amount: 120166.58... - 111654.4...]
- Table Headers:** customerNumber, amount
- Table Data:**

customerNumber	amount
141	120166.58
141	116208.4
124	111654.4
- Operations:** Edit, Copy, Delete
- Buttons:** Go, Cancel, Profiling, Edit inline, Edit, Explain SQL, Create PHP code, Refresh
- Query results operations:** Print, Copy to clipboard, Export, Display chart, Create view
- Console:** (bottom left)

- 32/79 -

2 . 3 LIMIT

ให้ query หาข้อมูล products ประกอบไปด้วย productCode, productName และ buyPrice โดยแสดงเฉพาะข้อมูลตาม buyPrice มากที่สุด 10 ลำดับแรก

- 33/79 -

- 34/79 -

2.4 DISTINCT

**SELECT city
FROM customers
ORDER BY city**

The screenshot shows the MySQL Workbench interface with the following details:

- Server: localhost
- Database: data_lab
- Table: customers
- Query: `SELECT city FROM customers ORDER BY city`
- Result: Rows 0 - 24 (122 total). The results show multiple entries for the same city, such as Auckland appearing four times.
- UI Elements: Browse, Structure, SQL, Search, Insert, Export, Import, Operational buttons. A search bar at the top. Pagination controls (1, >, >>), a "Number of rows" dropdown set to 25, and a "Filter rows" search bar.
- A red box highlights the row for Auckland, and a red arrow points from it to a yellow box containing the Thai text "ซื้อประทุมฯ".

**SELECT DISTINCT city
FROM customers
ORDER BY city**

The screenshot shows the MySQL Workbench interface with the following details:

- Server: localhost
- Database: data_lab
- Table: customers
- Query: `SELECT DISTINCT city FROM customers ORDER BY city`
- Result: Rows 0 - 24 (95 total). The results show each city listed only once, such as Auckland appearing only once.
- UI Elements: Browse, Structure, SQL, Search, Insert, Export, Import, Operational buttons. A search bar at the top. Pagination controls (1, >, >>), a "Number of rows" dropdown set to 25, and a "Filter rows" search bar.

2 . 4 DISTINCT

ให้ query หาข้อมูลนามสกุลของพนักงานที่ไม่ซ้ำกัน

- 37/79 -

- 38/79 -

2.5 LIKE

```
SELECT customerName, city  
FROM customers  
WHERE customerName like 'Toy%'
```

+ Options

← →

	customerName	city
<input type="checkbox"/> Edit Copy Delete	Toys of Finland, Co.	Helsinki
<input type="checkbox"/> Edit Copy Delete	Toys4GrownUps.com	Pasadena

- 39/79 -

2.5 LIKE

```
SELECT firstName, lastName  
FROM employees  
WHERE firstName like 'M__'
```

+ Options

← →

	firstName	lastName
<input type="checkbox"/> Edit Copy Delete	Mary	Patterson
<input type="checkbox"/> Edit Copy Delete	Mami	Nishi

- 40/79 -

3 . Query-II

- 41/79 -

3.1 Sub Query

มีบริษัทลูกค้าอะไรบ้าง ที่อยู่เมืองเดียวกับบริษัท Vitachrome Inc.

- 42/79 -

3.1 Sub Query

ขั้นตอนแรก หาว่า Vitachrome Inc. อยู่ที่เมืองอะไร

```
SELECT      city
FROM        customers
WHERE       customerName='Vitachrome Inc.'
```

A screenshot of a MySQL query results table. The table has one row with the following data:

+ Options	← T →	city
<input type="checkbox"/>	Edit Copy Delete	NYC

- 43/79 -

3.1 Sub Query

ขั้นที่สอง นำเมืองที่ได้จากขั้นตอนแรกมาหาว่ามีบริษัทอะไรอยู่ที่เมืองนี้บ้าง

```
SELECT      customerName
FROM        customers
WHERE       city = 'NYC'
```

A screenshot of a MySQL query results table. The table has five rows with the following data:

+ Options	← T →	customerName
<input type="checkbox"/>	Edit Copy Delete	Land of Toys Inc.
<input type="checkbox"/>	Edit Copy Delete	Muscle Machine Inc.
<input type="checkbox"/>	Edit Copy Delete	Vitachrome Inc.
<input type="checkbox"/>	Edit Copy Delete	Classic Legends Inc.
<input type="checkbox"/>	Edit Copy Delete	Microscale Inc.

- 44/79 -

3.1 Sub Query

มีบริษัทลูกค้าอะไรบ้าง ที่อยู่เมืองเดียวกับบริษัท Vitachrome Inc.

สามารถเรียกคำสั่ง SQL ซ้อนกันได้ เพื่อทำงานเพียงครั้งเดียว

```
SELECT      columnName
FROM        customers
WHERE       city = (
              SELECT      city
              FROM        customers
              WHERE       customerName='Vitachrome Inc.'
            )
```

- 45/79 -

3.1 Sub Query

1. ให้แสดงรายชื่อพนักงานที่มีนามสกุลเดียวกับ William

2. ให้แสดงรายชื่อและนามสกุลพนักงานที่อยู่ออฟฟิศเดียวกับ William Patterson

- 46/79 -

3.2 Function

ยอดชำระเงินที่มากที่สุด น้อยที่สุด คือเท่าไร

```
SELECT MAX(amount), MIN(amount)  
FROM payments
```

- 47/79 -

3.2 Function

ยอดชำระเงินของลูกค้า 103

```
SELECT customerNumber, amount  
FROM payments  
WHERE customerNumber = 103
```

customerNumber	amount
103	6066.78
103	14571.44
103	1676.14

```
SELECT customerNumber, SUM(amount)  
FROM payments  
WHERE customerNumber = 103
```

customerNumber	SUM(amount)
103	22314.36

- 48/79 -

3.3 GROUP BY

Server: localhost » Database: data_lab » Table: customers

Browse Structure SQL Search Insert Export Import Operations Triggers

g rows 0 - 24 (122 total, Query took 0.0008 seconds.) [salesRepEmployeeNumber: 1702... - 1504...]

IM `customers` ORDER BY `customers`.`salesRepEmployeeNumber` DESC

Profile Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

	customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployeeNumber	creditLimit
	216	Enaco Distributors	Saavedra	Eduardo	(93) 203 4555	Rambla de Cataluña, 23	NULL	Barcelona	NULL	08022	Spain	1702	60300
	298	Vida Sport, Ltd	Holz	Mihael	0897-034555	Grenzacherweg 237	NULL	Genève	NULL	1203	Switzerland	1702	141300
	344	CAF Imports	Fernandez	Jesus	+34 913 728 555	Merchants House	27-30 Merchant's Quay	Madrid	NULL	28023	Spain	1702	59600
	376	Precious Collectables	Urs	Braun	0452-076555	Hauptstr. 29	NULL	Bern	NULL	3011	Switzerland	1702	0
	458	Corrida Auto Replicas, Ltd	So		+31 00 6342 5555	1-6-20 Dojima	NULL	Kita-ku	Osaka	556 0004	Japan	1702	104600
	484	Iberia Gift Imports, Corp.	Ro		+852 2251 1555	Bank of China Tower	1 Garden Road	Central Hong Kong	NULL	41101	Spain	1702	65700
	148	Dragon Souveniers, Ltd.	Na		+63 2 555 3587	15 McCallum Street	NatWest Center #13-03	Makati City	NULL	1227 MM	Philippines	1621	103800
	177	Osaka Souveniers Co.	Kentary	Mory	+81 3 3584 0000	2-2-8 Roppongi	NULL	Minato-ku	Tokyo	106-0032	Japan	1621	81200
	211	King Kong Collectables, Co.	Gao	Mike								1621	58600
	385	Cruz & Sons Co.	Cruz	Arnold								1621	81500
	398	Tokyo Collectables, Ltd	Shimamura	Akiko								1621	94400

3.3 GROUP BY

salesRepEmployeeNumber	COUNT(customers.customerNumber)
NULL	22
1165	6
1166	6
1188	6
1216	6
1286	7
1323	8
1337	6
1370	7
1401	10
1501	8
1504	9
1611	5
1612	5
1621	5
1702	6

3 . 3 GROUP BY

```
SELECT customers.salesRepEmployeeNumber,  
       COUNT(customers.customerNumber)  
FROM   customers  
GROUP BY customers.salesRepEmployeeNumber
```

- 51/79 -

3 . 3 GROUP BY

<code>salesRepEmployeeNumber</code>	<code>COUNT(customers.customerNumber)</code>
<code>NULL</code>	22
1165	6
1166	6
1188	6
1216	6
1286	7
1323	8
1337	6
1370	7
1401	10
1501	8
1504	9
1611	5
1612	5
1621	5
1702	6

- 52/79 -

3.3 GROUP BY

```
SELECT    orders.customerNumber,  
          COUNT(orders.orderNumber) as 'Number of Orders'  
FROM      orders  
GROUP BY  orders.customerNumber  
ORDER BY  'Number of Orders` DESC
```

- 53/79 -

3.3 GROUP BY

customerNumber	Number of Orders
141	26
124	17
148	5
114	5
145	5
353	5
323	5
321	4
381	4
128	4
496	4
282	4
121	4
382	4
161	4
131	4
450	4
276	4
398	4
151	4
166	4
157	4
144	4
119	4
424	3

- 54/79 -

3.3 GROUP BY

ให้ query หาข้อมูล orderdetails ประกอบไปด้วย productCode และจำนวนการสั่งสินค้าซึ่นน้ำ (Number of Orders)

- 55/79 -

3.3 GROUP BY

ยอดชำระเงินของลูกค้าแต่ละคน

```
SELECT      customerNumber, SUM(amount)
FROM        payments
GROUP BY    customerNumber
```

customerNumber	SUM(amount)
103	22314.36
112	80180.98
114	180585.06999999998
119	116949.68000000001
121	104224.79
124	584188.2400000001
128	75937.76000000001
129	66710.56
131	107639.94
141	715738.9800000001
144	43680.65

- 56/79 -

3.3 GROUP BY

ให้แสดงจำนวนพนักงานแยกตามออฟฟิศ

```
SELECT officeCode, COUNT(officeCode)
FROM employees
GROUP BY officeCode
```

officeCode	COUNT(officeCode)
1	6
2	2
3	2
4	5
5	2
6	4
7	2

- 57/79 -

- 58/79 -

4 . Join

- 59/79 -

4 . 1 Left-Join #1

employees

offices

employeeNumber	firstName	officeCode	officeCode	country
1002	Diane	1	1	USA
1056	Mary	1	1	USA
1076	Jeff	1	1	USA
1088	William	6	6	Australia
1102	Gerard	4	4	France
1143	Anthony	1	1	USA
1165	Leslie	1	1	USA
1166	Leslie	1	1	USA
1188	Julie	2	2	USA
1216	Steve	2	2	USA
1286	Foon Yue	3	3	USA
1323	George	3	3	USA
1337	Loui	4	4	France
1370	Gerard	4	4	France
1401	Pamela	4	4	France
1501	Larry	7	7	UK
1504	Barry	7	7	UK
1611	Andy	6	6	Australia
1612	Peter	6	6	Australia
1619	Tom	6	6	Australia
1621	Mami	5	5	Japan
1625	Yoshimi	5	5	Japan

- 60/79 -

4.1 Left-Join #1

employeeNumber	firstName	officeCode
1002	Diane	1
1056	Mary	1
1076	Jeff	1
1088	William	6
1102	Gerard	4
1143	Anthony	1
1165	Leslie	1
1166	Leslie	1
1188	Julie	2
1216	Steve	2
1286	Foon Yue	3
1323	George	3
1337	Loui	4
1370	Gerard	4

```
SELECT employeeNumber,  
       firstName,  
       officeCode  
  FROM employees
```

- 61/79 -

4.1 Left-Join #1

officeCode	country
1	USA
2	USA
3	USA
4	France
5	Japan
6	Australia
7	UK

```
SELECT officeCode,  
       country  
  FROM offices
```

- 62/79 -

4.1 Left-Join #1

employeeNumber	firstName	officeCode
1002	Diane	1
1056	Mary	1
1076	Jeff	1
1088	William	6
1102	Gerard	4
1143	Anthony	1
1165	Leslie	1
1166	Leslie	1
1188	Julie	2
1216	Steve	2
1286	Foon Yue	3
1323	George	3
1337	Loui	4
1370	Gerard	4

employees

officeCode	country
1	USA
2	USA
3	USA
4	France
5	Japan
6	Australia
7	UK

offices

- 63/79 -

4.1 Left-Join #1

SELECT

employees.employeeNumber,
 employees.firstName,
 employees.officeCode,
 offices.officeCode,
 offices.country

FROM

employees

LEFT JOIN

offices

ON

employees.officeCode = offices.officeCode

employeeNumber	firstName	officeCode	officeCode	country
1002	Diane	1	1	USA
1056	Mary	1	1	USA
1076	Jeff	1	1	USA
1088	William	6	6	Australia
1102	Gerard	4	4	France
1143	Anthony	1	1	USA
1165	Leslie	1	1	USA
1166	Leslie	1	1	USA
1188	Julie	2	2	USA
1216	Steve	2	2	USA
1286	Foon Yue	3	3	USA
1323	George	3	3	USA
1337	Loui	4	4	France
1370	Gerard	4	4	France
1401	Pamela	4	4	France
1501	Larry	7	7	UK
1504	Barry	7	7	UK
1611	Andy	6	6	Australia
1612	Peter	6	6	Australia
1619	Tom	6	6	Australia
1621	Mami	5	5	Japan
1625	Yoshimi	5	5	Japan

- 64/79 -

4.1 Left-Join #2

customers

customerName	salesRepEmployeeNumber	firstName	lastName
Atelier graphique	1370	Gerard	Hernandez
Signal Gift Stores	1166	Leslie	Thompson
Australian Collectors, Co.	1611	Andy	Fixter
La Rochelle Gifts	1370	Gerard	Hernandez
Baane Mini Imports	1504	Barry	Jones
Mini Gifts Distributors Ltd.	1165	Leslie	Jennings
Havel & Zbyszek Co	NULL	NULL	NULL
Blauer See Auto, Co.	1504	Barry	Jones
Mini Wheels Co.	1165	Leslie	Jennings
Land of Toys Inc.	1323	George	Vanauf
Euro+ Shopping Channel	1370	Gerard	Hernandez
Volvo Model Replicas, Co	1504	Barry	Jones
Danish Wholesale Imports	1401	Pamela	Castillo
Saveley & Henriot, Co.	1337	Loui	Bondur
Dragon Souveniers, Ltd.	1621	Mami	Nishi
Muscle Machine Inc	1286	Foon Yue	Tseng
Diecast Classics Inc.	1216	Steve	Patterson
Technics Stores Inc.	1165	Leslie	Jennings
Handji Gifts& Co	1612	Peter	Marsh
Herkku Gifts	1504	Barry	Jones
American Souvenirs Inc	1286	Foon Yue	Tsena

- 65/79 -

employees

4.1 Left-Join #2

customerName	salesRepEmployeeNumber	employeeNumber	firstName	lastName
Atelier graphique	1370	1002	Diane	Murphy
Signal Gift Stores	1166	1056	Mary	Patterson
Australian Collectors, Co.	1611	1076	Jeff	Firrelli
La Rochelle Gifts	1370	1088	William	Patterson
Baane Mini Imports	1504	1102	Gerard	Bondur
Mini Gifts Distributors Ltd.	1165	1143	Anthony	Bow
Havel & Zbyszek Co	NULL	1165	Slie	Jennings
Blauer See Auto, Co.	1504	1165	Slie	Thompson

SELECT **customerName,**
FROM **salesRepEmployeeNumber**
 customers

SELECT **employeeNumber,**
FROM **firstName,**
 lastName
 employees

4.1 Left-Join #2

```
SELECT      customers.customerName,  
            customers.salesRepEmployeeNumber,  
            employees.firstName,  
            employees.lastName  
FROM        customers  
LEFT JOIN   employees  
ON          customers.salesRepEmployeeNumber =  
                    employees.employeeNumber
```

- 67/79 -

4.1 Left-Join #2

customerName	salesRepEmployeeNumber	firstName	lastName
Atelier graphique	1370	Gerard	Hernandez
Signal Gift Stores	1166	Leslie	Thompson
Australian Collectors, Co.	1611	Andy	Fixter
La Rochelle Gifts	1370	Gerard	Hernandez
Baane Mini Imports	1504	Barry	Jones
Mini Gifts Distributors Ltd.	1165	Leslie	Jennings
Havel & Zbyszek Co	NULL	NULL	NULL
Blauer See Auto, Co.	1504	Barry	Jones
Mini Wheels Co.	1165	Leslie	Jennings
Land of Toys Inc.	1323	George	Vanauf
Euro+ Shopping Channel	1370	Gerard	Hernandez
Volvo Model Replicas, Co	1504	Barry	Jones
Danish Wholesale Imports	1401	Pamela	Castillo
Saveley & Henriot, Co.	1337	Loui	Bondur
Dragon Souveniers, Ltd.	1621	Mami	Nishi
Muscle Machine Inc	1286	Foon Yue	Tseng
Diecast Classics Inc.	1216	Steve	Patterson
Technics Stores Inc.	1165	Leslie	Jennings
Handji Gifts& Co	1612	Peter	Marsh
Herkku Gifts	- 68/79 -	Barry	Jones
American Souvenirs Inc	1286	Foon Yue	Tsena

4.1 Left-Join #3

productCode	Number of Orders
S18_3232	53
S24_2840	28
S700_1691	28
S32_3522	28
S18_3029	28
S18_2949	28
S24_3816	28
S32_1374	28
S24_1937	28
S700_4002	28
S18_2319	28
S24_3949	28
S18_4668	28
S10_4962	28
S12_2823	28
S700_2047	28
S700_2824	28
S18_3856	28
S18_2957	28
S24_2022	28

```
SELECT orderdetails.productCode,  
       COUNT(orderdetails.orderNumber) AS  
          'Number of Orders'  
FROM orderdetails  
GROUP BY orderdetails.productCode
```

- 69/79 -

4.1 Left-Join #3

productCode	productName
S10_1678	1969 Harley Davidson Ultimate Chopper
S10_1949	1952 Alpine Renault 1300
S10_2016	1996 Moto Guzzi 1100i
S10_4698	2003 Harley-Davidson Eagle Drag Bike
S10_4757	1972 Alfa Romeo GTA
S10_4962	1962 LanciaA Delta 16V
S12_1099	1968 Ford Mustang
S12_1108	2001 Ferrari Enzo
S12_1666	1958 Setra Bus
S12_2823	2002 Suzuki XREO
S12_3148	1969 Corvair Monza
S12_3380	1968 Dodge Charger
S12_3891	1969 Ford Falcon
S12_3990	1970 Plymouth Hemi Cuda
S12_4473	1957 Chevy Pickup
S12_4675	1969 Dodge Charger

```
SELECT products.productCode,  
       products.productName  
FROM products
```

- 70/79 -

4.1 Left-Join #3

```
SELECT      orderdetails.productCode,  
products.productName,  
COUNT(orderdetails.orderNumber) AS 'Number of Orders'  
FROM        orderdetails
```

LEFT JOIN products

ON orderdetails.productCode = products.productCode

GROUP BY orderdetails.productCode

ORDER BY `Number of Orders` DESC

- 71/79 -

4.1 Left-Join #3

orderdetails

products

productCode	productName	Number of Orders
S18_3237	1992 Ferrari 360 Spider red	53
S18_2949	1913 Ford Model T Speedster	28
S24_3816	1940 Ford Deluxe Sedan	28
S32_4289	1928 Ford Phaeton Deluxe	28
S32_1374	1997 BMW F650 ST	28
S18_2057	1934 Ford V8 Coupe	28
S24_1444	1970 Dodge Coronet	28
S50_1341	1930 Buick Marquette Phaeton	28
S18_3136	18th Century Vintage Horse Carriage	28
S24_2840	1958 Chevy Corvette Limited Edition	28
S700_1691	American Airlines: B767-300	28
S18_3320	1917 Maxwell Touring Car	28
S700_2047	HMS Bounty	28
S18_4600	1940s Ford truck	28
S32_2509	1954 Greyhound Scenicruiser	28
S700_2466	America West Airlines B757-200	28
S18_2325	1932 Model A Ford J-Coupe	28
S18_4668	1939 Cadillac Limousine	28
S700_2834	ATA: B757-300	28
S18_2795	1928 Mercedes-Benz SSK	28
S24_2300	1962 Volkswagen Microbus	28
S50_1392	Diamond T620 Semi-Skirted Tanker	28

- 72/79 -

5 . Export

- 73/79 -

5 . 1 Excel

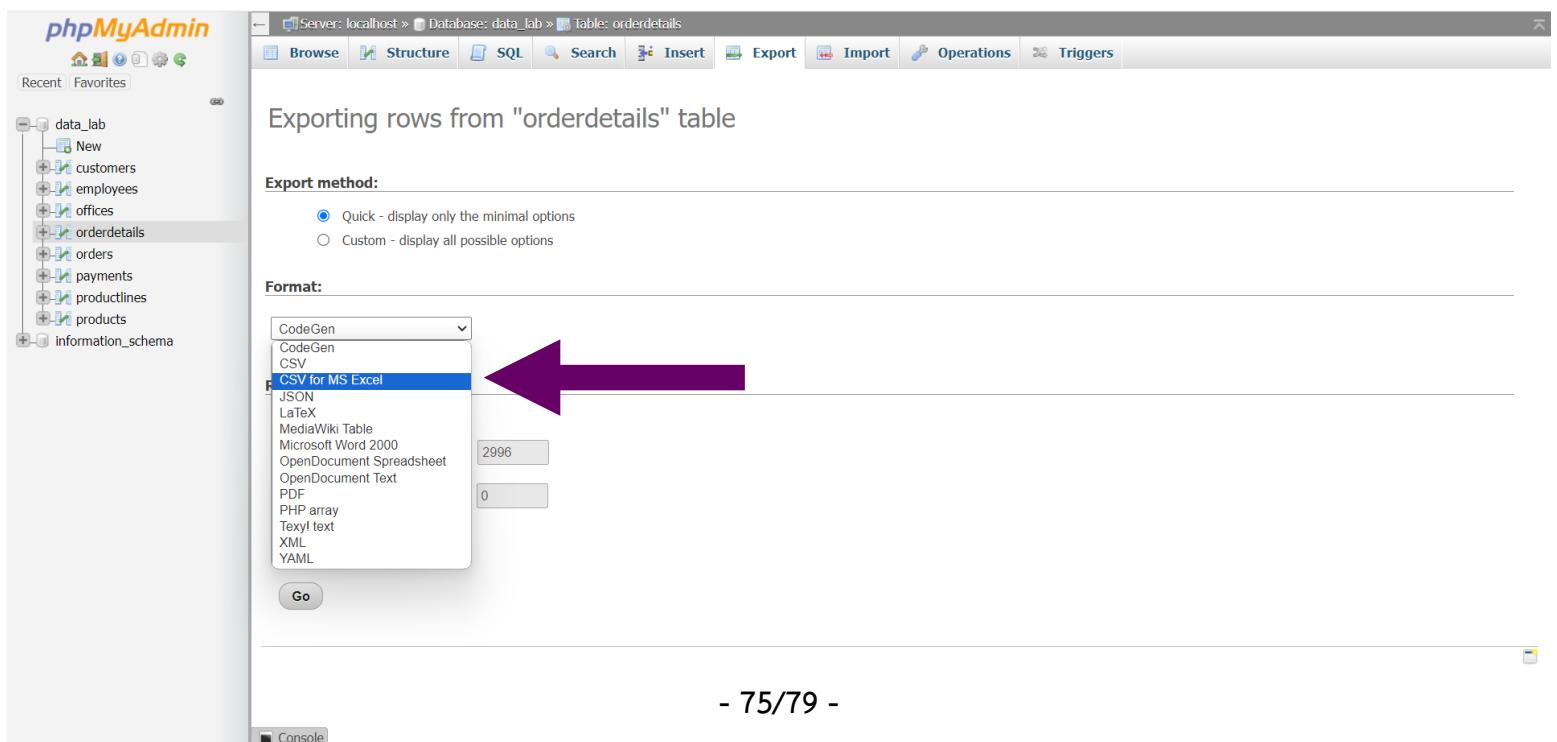
The screenshot shows the phpMyAdmin interface for a database named 'data_lab'. The left sidebar lists databases, tables, and schema. The main area displays a table titled 'products' with columns: productCode, productName, and Number of Orders. A purple arrow points from the bottom-left towards the 'Export' button in the toolbar below the table.

productCode	productName	Number of Orders
S18_3232	1992 Ferrari 360 Spider red	53
S18_2325	1932 Model A Ford J-Coupe	28
S18_4668	1939 Cadillac Limousine	28
S700_2834	ATA: B757-300	28
S18_2795	1928 Mercedes-Benz SSK	28
S24_2300	1962 Volkswagen Microbus	28
S50_1392	Diamond T620 Semi-Skirted Tanker	28
S700_3167	F/A 18 Hornet 1/72	28
S24_1937	1939 Chevrolet Deluxe Coupe	28
S24_4258	1936 Chrysler Airflow	28
S18_1662	1980s Black Hawk Helicopter	28
S700_4002	American Airlines: MD-11S	28
S24_2022	1938 Cadillac V-16 Presidential Limousine	28
S32_1268	1980's GM Manhattan Express	28
S10_4757	1972 Alfa Romeo GTA	28
S18_2581	P-51-D Mustang	28
S72_1253	Boeing X-32A JSF	28
S18_1342	1937 Lincoln Berlin	28
S32_3522	1996 Peterbilt 379 Stake Bed with Outrigger	28
S18_3029	1999 Yamaha Speed Boat	28
S10_1678	1969 Harley Davidson Ultimate Copper	28
S18_1367	1936 Mercedes-Benz 500K Special Roadster	28
S700_2824	1982 Camaro Z28	28
S18_3856	1941 Chevrolet Special De Luxe Cabriolet	28
S10_2016	1996 Moto Guzzi 1100i	28

Query results operations: Print, Copy to clipboard, Export, Display chart, Create view

- 74/79 -

5.1 Excel



Exporting rows from "orderdetails" table

Export method:

Quick - display only the minimal options
 Custom - display all possible options

Format:

CodeGen
CodeGen
CSV
CSV for MS Excel
JSON
LaTeX
MediaWiki Table
Microsoft Word 2000
OpenDocument Spreadsheet
OpenDocument Text
PDF
PHP array
Texty text
XML
YAML

2996
0

Go

- 75/79 -

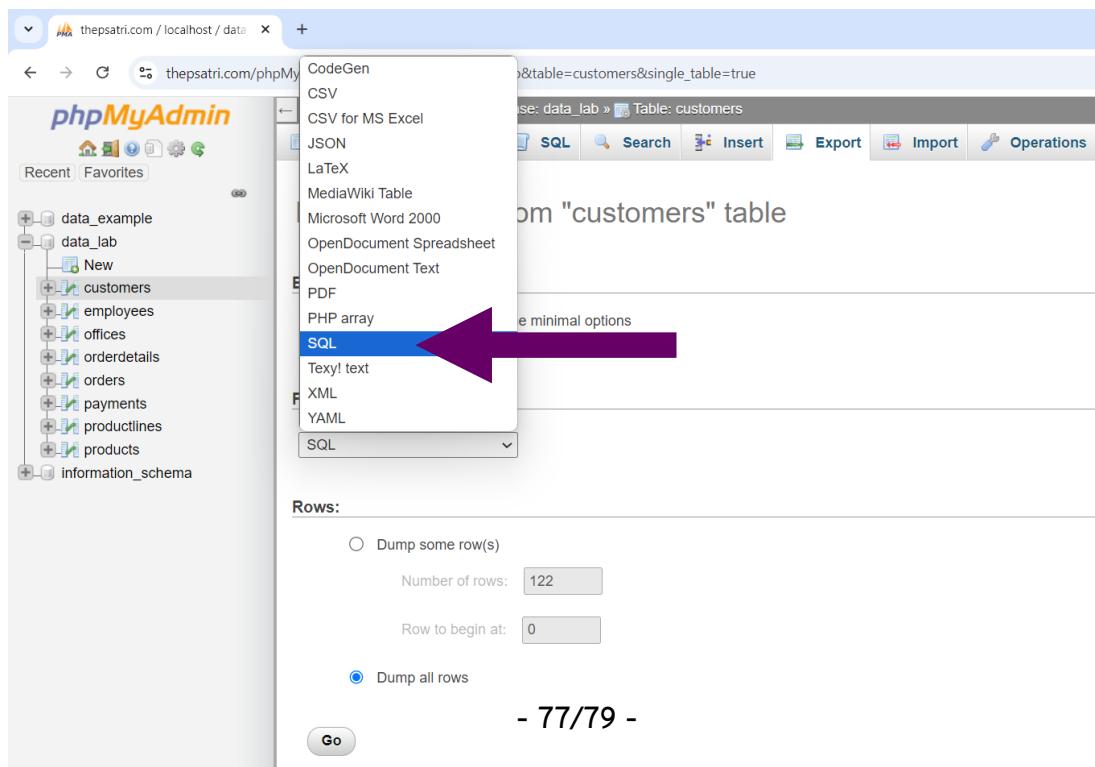
Console

5.1 Excel

A	B	C	D	E	F	G	H	I	J	K
productCode	productName	Number of Orders								
S18_3232	1992 Ferrari 360 Spider red	53								
S18_2949	1913 Ford Model T Speedster	28								
S24_3816	1940 Ford Delivery Sedan	28								
S32_4289	1928 Ford Phaeton Deluxe	28								
S32_1374	1997 BMW F650 ST	28								
S18_2957	1934 Ford V8 Coupe	28								
S24_1444	1970 Dodge Coronet	28								
S50_1341	1930 Buick Marquette Phaeton	28								
S18_3136	18th Century Vintage Horse Carriage	28								
S24_2840	1958 Chevy Corvette Limited Edition	28								
S700_1691	American Airlines: B767-300	28								
S18_3320	1917 Maxwell Touring Car	28								
S700_2047	HMS Bounty	28								
S18_4600	1940s Ford truck	28								
S32_2509	1954 Greyhound Scenicruiser	28								
S700_2466	America West Airlines B757-200	28								
S18_2325	1932 Model A Ford J-Coupe	28								
S18_4668	1939 Cadillac Limousine	28								
S700_2834	ATA: B757-300	28								
S18_2795	1928 Mercedes-Benz SSK	28								
S24_2300	1962 Volkswagen Microbus	28								
S50_1392	Diamond T620 Semi-Skirted Tanker	28								
S700_3167	F/A 18 Hornet 1/72	28								
S24_1937	1939 Chevrolet Deluxe Coupe	28								
S24_4258	1936 Chrysler Airflow	28								
S18_1662	1980s Black Hawk Helicopter	28								
S700_4002	Antique Automobile MD 146	28								

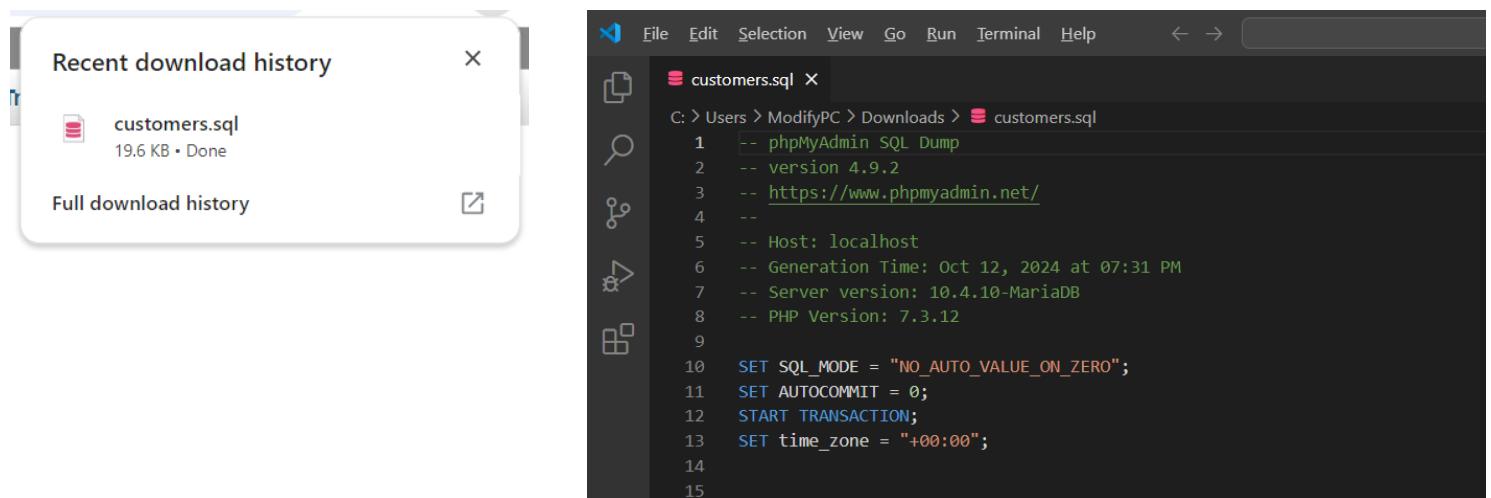
- 76/79 -

5.2 SQL (Back up)



The screenshot shows the phpMyAdmin interface for a database named 'data_lab'. The 'customers' table is selected. A context menu is open over the table, with the 'SQL' option highlighted by a purple arrow. The menu also includes options like CodeGen, CSV, CSV for MS Excel, JSON, LaTeX, MediaWiki Table, Microsoft Word 2000, OpenDocument Spreadsheet, OpenDocument Text, PDF, PHP array, SQL, Texy! text, XML, YAML, and another SQL option at the bottom.

5.2 SQL (Back up)



The screenshot shows a file manager window with a 'Recent download history' section. It lists a file named 'customers.sql' which is 19.6 KB and has been completed ('Done'). Below this is a 'Full download history' section. To the right, a code editor window titled 'customers.sql' shows the generated SQL dump. The dump includes standard header information such as the MySQL version, host, generation time, server version, and PHP version, followed by the actual data definition language (DDL) statements for creating tables and their structures.

```
1 -- phpMyAdmin SQL Dump
2 -- version 4.9.2
3 -- https://www.phpmyadmin.net/
4 --
5 -- Host: localhost
6 -- Generation Time: Oct 12, 2024 at 07:31 PM
7 -- Server version: 10.4.10-MariaDB
8 -- PHP Version: 7.3.12
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET AUTOCOMMIT = 0;
12 START TRANSACTION;
13 SET time_zone = "+00:00";
14
15
```



- 79/79 -

แนวข้อสอบปฏิบัติการ SQL (part I)

1. ให้แสดงข้อมูลสำนักงาน (officeCode, city, state, country) ที่อยู่ในประเทศ USA
2. ให้แสดงข้อมูลลูกค้า (Customers) ดังนี้ customerName, contactLastName และ city โดยให้เรียงลำดับจากมากไปน้อยตาม customerName
3. ให้แสดงข้อมูลพนักงาน (Employees) ดังนี้ firstName, lastName และ email โดยเลือกเฉพาะ jobTitle เป็น Sales Manager
4. ให้แสดงข้อมูลพนักงาน (Employees) ดังนี้ firstName, lastName และ email โดยเลือกเฉพาะพนักงานที่มี office อยู่ในประเทศ USA, Japan
5. ให้แสดงข้อมูลลูกค้า (Customers) โดยใหม่ข้อมูลดังนี้ customerName, city, country และเลือกแสดงเฉพาะลูกค้าที่ไม่อยู่ในประเทศ Japan, USA
6. ให้แสดงข้อมูลของ products ที่มี productCode, productName และ quantityInStock โดยเลือกแสดงเฉพาะ product ที่มี productLine เป็น Vintage Cars
7. ให้แสดงราคาเฉลี่ยของการชำระเงิน (Payments)
8. ให้แสดงชื่อลูกค้า (customerName) ที่มีจำนวนครั้งในการสั่งซื้อร่วมสูงสุด
9. ให้แสดงชื่อและนามสกุลพนักงาน (firstName, lastName) ที่มีจำนวนลูกค้าที่ดูแลรวมสูงสุด
10. ให้แสดงข้อมูลลูกค้าซึ่งใหม่ข้อมูลดังนี้ customerName, contactFirstName, contactLastName และแสดงรายละเอียดของพนักงานที่เป็นคนดูแลลูกค้ารายนั้นๆ (firstName และ lastName)
11. ให้แสดงข้อมูลลูกค้าที่ประกอบไปด้วย customerName, contactFirstName และ phone ที่มีพนักงานที่ดูแลคนเดียวกับลูกค้าที่ชื่อ Vida Sport, Ltd
12. ให้แสดงข้อมูล orderNumber ที่สั่งสินค้า (productDescription) มี Chrome Trim เป็นส่วนประกอบ และมีสถานะเป็น Shipped
13. ให้แสดงข้อมูลที่ประกอบไปด้วย customerName, country และ ค่าเฉลี่ยของ amount (ใน payments) ของลูกค้ารายนั้นๆ
14. ให้แสดงชื่อลูกค้า (customerName) ที่สั่งสินค้าใน productLine เป็น Motorcycles และ Classic Cars พร้อมทั้งแสดงชื่อ-สกุลพนักงานที่ดูแลด้วย
15. ให้แสดงชื่อลูกค้า (customerName) ที่สั่งสินค้า (productDescription) มี Chrome Trim เป็นส่วนประกอบ และมียอดสั่งซื้อ (quantityOrdered) เกิน 50 ชิ้น

SQL Basics Cheat Sheet

SQL

SQL, or Structured Query Language, is a language to talk to databases. It allows you to select specific data and to build complex reports. Today, SQL is a universal language of data. It is used in practically all technologies that process data.

SAMPLE DATA

COUNTRY			
id	name	population	area
1	France	66600000	640680
2	Germany	80700000	357000
...

CITY				
id	name	country_id	population	rating
1	Paris	1	2243000	5
2	Berlin	2	3460000	3
...

QUERYING SINGLE TABLE

Fetch all columns from the country table:

```
SELECT *  
FROM country;
```

Fetch id and name columns from the city table:

```
SELECT id, name  
FROM city;
```

Fetch city names sorted by the rating column in the default ASCending order:

```
SELECT name  
FROM city  
ORDER BY rating [ASC];
```

Fetch city names sorted by the rating column in the DESCending order:

```
SELECT name  
FROM city  
ORDER BY rating DESC;
```

ALIASES

COLUMNS

```
SELECT name AS city_name  
FROM city;
```

TABLES

```
SELECT co.name, ci.name  
FROM city AS ci  
JOIN country AS co  
ON ci.country_id = co.id;
```

FILTERING THE OUTPUT

COMPARISON OPERATORS

Fetch names of cities that have a rating above 3:

```
SELECT name  
FROM city  
WHERE rating > 3;
```

Fetch names of cities that are neither Berlin nor Madrid:

```
SELECT name  
FROM city  
WHERE name != 'Berlin'  
AND name != 'Madrid';
```

TEXT OPERATORS

Fetch names of cities that start with a 'P' or end with an 's':

```
SELECT name  
FROM city  
WHERE name LIKE 'P%'  
OR name LIKE '%s';
```

Fetch names of cities that start with any letter followed by 'ublin' (like Dublin in Ireland or Lublin in Poland):

```
SELECT name  
FROM city  
WHERE name LIKE '_ublin';
```

OTHER OPERATORS

Fetch names of cities that have a population between 500K and 5M:

```
SELECT name  
FROM city  
WHERE population BETWEEN 500000 AND 5000000;
```

Fetch names of cities that don't miss a rating value:

```
SELECT name  
FROM city  
WHERE rating IS NOT NULL;
```

Fetch names of cities that are in countries with IDs 1, 4, 7, or 8:

```
SELECT name  
FROM city  
WHERE country_id IN (1, 4, 7, 8);
```

QUERYING MULTIPLE TABLES

INNER JOIN

JOIN (or explicitly INNER JOIN) returns rows that have matching values in both tables.

```
SELECT city.name, country.name  
FROM city  
[INNER] JOIN country  
ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
3	Warsaw	4	3	Iceland

LEFT JOIN

LEFT JOIN returns all rows from the left table with corresponding rows from the right table. If there's no matching row, NULLs are returned as values from the second table.

```
SELECT city.name, country.name  
FROM city  
LEFT JOIN country  
ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
3	Warsaw	4	NULL	NULL

RIGHT JOIN

RIGHT JOIN returns all rows from the right table with corresponding rows from the left table. If there's no matching row, NULLs are returned as values from the left table.

```
SELECT city.name, country.name  
FROM city  
RIGHT JOIN country  
ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
NULL	NULL	NULL	3	Iceland

FULL JOIN

FULL JOIN (or explicitly FULL OUTER JOIN) returns all rows from both tables - if there's no matching row in the second table, NULLs are returned.

```
SELECT city.name, country.name  
FROM city  
FULL [OUTER] JOIN country  
ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
3	Warsaw	4	NULL	NULL
NULL	NULL	NULL	3	Iceland

CROSS JOIN

CROSS JOIN returns all possible combinations of rows from both tables. There are two syntaxes available.

```
SELECT city.name, country.name  
FROM city  
CROSS JOIN country;
```

```
SELECT city.name, country.name  
FROM city, country;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
1	Paris	1	2	Germany
2	Berlin	2	1	France
2	Berlin	2	2	Germany

NATURAL JOIN

NATURAL JOIN will join tables by all columns with the same name.

```
SELECT city.name, country.name  
FROM city  
NATURAL JOIN country;
```

CITY			COUNTRY	
country_id	id	name	name	id
6	6	San Marino	San Marino	6
7	7	Vatican City	Vatican City	7
5	9	Greece	Greece	9
10	11	Monaco	Monaco	10

NATURAL JOIN used these columns to match rows: city.id, city.name, country.id, country.name

NATURAL JOIN is very rarely used in practice.

SQL Basics Cheat Sheet

AGGREGATION AND GROUPING

GROUP BY **groups** together rows that have the same values in specified columns. It computes summaries (aggregates) for each unique combination of values.

CITY		
id	name	country_id
1	Paris	1
101	Marseille	1
102	Lyon	1
2	Berlin	2
103	Hamburg	2
104	Munich	2
3	Warsaw	4
105	Cracow	4

→

CITY		
country_id	count	
1	3	
2	3	
4	2	

AGGREGATE FUNCTIONS

- **avg(expr)** – average value for rows within the group
- **count(expr)** – count of values for rows within the group
- **max(expr)** – maximum value within the group
- **min(expr)** – minimum value within the group
- **sum(expr)** – sum of values within the group

EXAMPLE QUERIES

Find out the number of cities:

```
SELECT COUNT(*)  
FROM city;
```

Find out the number of cities with non-null ratings:

```
SELECT COUNT(rating)  
FROM city;
```

Find out the number of distinctive country values:

```
SELECT COUNT(DISTINCT country_id)  
FROM city;
```

Find out the smallest and the greatest country populations:

```
SELECT MIN(population), MAX(population)  
FROM country;
```

Find out the total population of cities in respective countries:

```
SELECT country_id, SUM(population)  
FROM city  
GROUP BY country_id;
```

Find out the average rating for cities in respective countries if the average is above 3.0:

```
SELECT country_id, AVG(rating)  
FROM city  
GROUP BY country_id  
HAVING AVG(rating) > 3.0;
```

SUBQUERIES

A subquery is a query that is nested inside another query, or inside another subquery. There are different types of subqueries.

SINGLE VALUE

The simplest subquery returns exactly one column and exactly one row. It can be used with comparison operators =, <, <=, >, or >=.

This query finds cities with the same rating as Paris:

```
SELECT name FROM city  
WHERE rating = (  
    SELECT rating  
    FROM city  
    WHERE name = 'Paris'  
);
```

MULTIPLE VALUES

A subquery can also return multiple columns or multiple rows. Such subqueries can be used with operators IN, EXISTS, ALL, or ANY.

This query finds cities in countries that have a population above 20M:

```
SELECT name  
FROM city  
WHERE country_id IN (  
    SELECT country_id  
    FROM country  
    WHERE population > 20000000  
);
```

CORRELATED

A correlated subquery refers to the tables introduced in the outer query. A correlated subquery depends on the outer query. It cannot be run independently from the outer query.

This query finds cities with a population greater than the average population in the country:

```
SELECT *  
FROM city main_city  
WHERE population > (  
    SELECT AVG(population)  
    FROM city average_city  
    WHERE average_city.country_id = main_city.country_id  
);
```

This query finds countries that have at least one city:

```
SELECT name  
FROM country  
WHERE EXISTS (  
    SELECT *  
    FROM city  
    WHERE country_id = country.id  
);
```

SET OPERATIONS

Set operations are used to combine the results of two or more queries into a single result. The combined queries must return the same number of columns and compatible data types. The names of the corresponding columns can be different.

CYCLING		
id	name	country
1	YK	DE
2	ZG	DE
3	WT	PL
...

SKATING		
id	name	country
1	YK	DE
2	DF	DE
3	AK	PL
...

UNION

UNION combines the results of two result sets and removes duplicates. UNION ALL doesn't remove duplicate rows.

This query displays German cyclists together with German skaters:

```
SELECT name  
FROM cycling  
WHERE country = 'DE'  
UNION / UNION ALL  
SELECT name  
FROM skating  
WHERE country = 'DE';
```

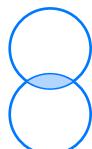


INTERSECT

INTERSECT returns only rows that appear in both result sets.

This query displays German cyclists who are also German skaters at the same time:

```
SELECT name  
FROM cycling  
WHERE country = 'DE'  
INTERSECT  
SELECT name  
FROM skating  
WHERE country = 'DE';
```



EXCEPT

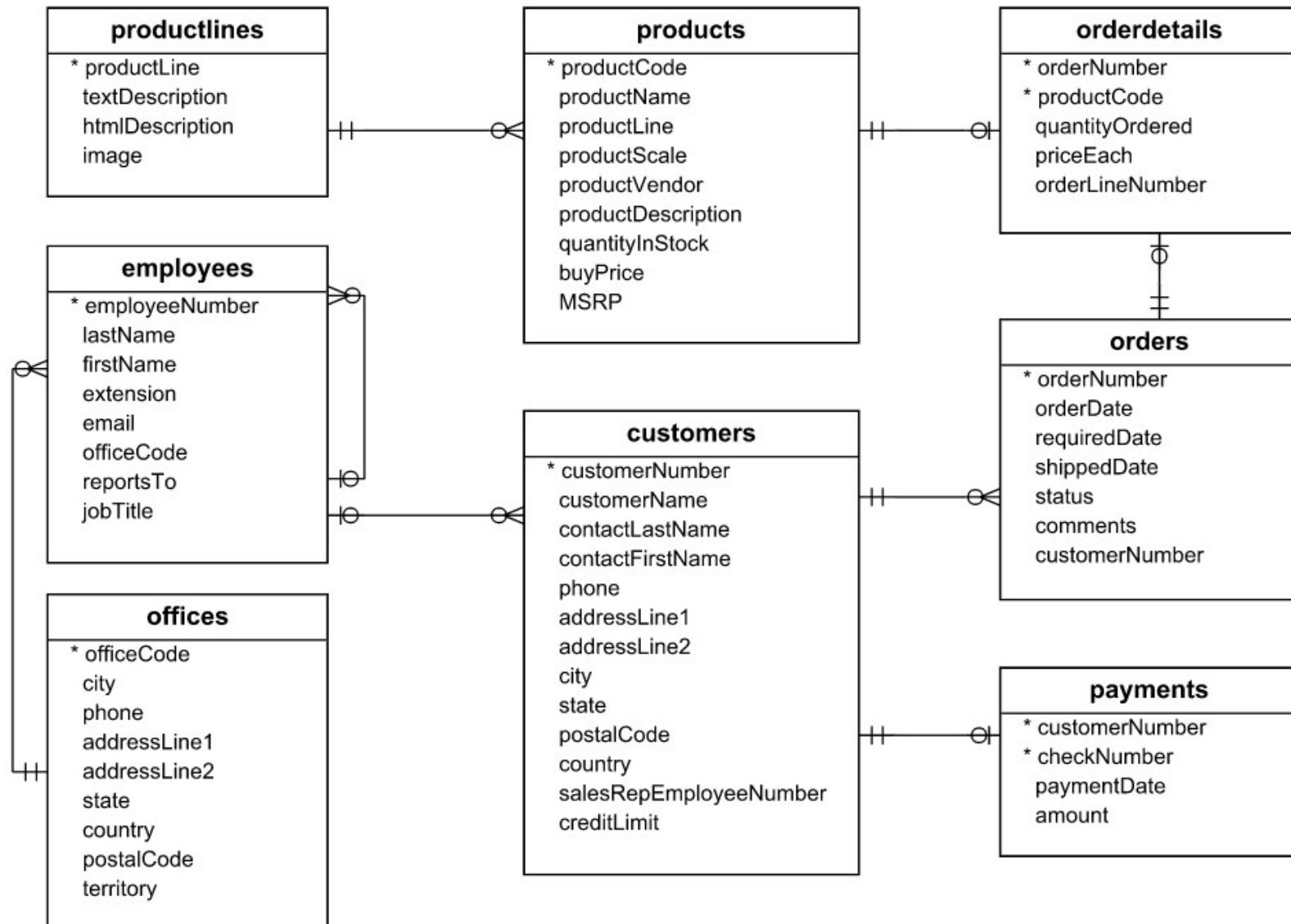
EXCEPT returns only the rows that appear in the first result set but do not appear in the second result set.

This query displays German cyclists unless they are also German skaters at the same time:

```
SELECT name  
FROM cycling  
WHERE country = 'DE'  
EXCEPT / MINUS  
SELECT name  
FROM skating  
WHERE country = 'DE';
```



MySQL Sample Database Diagram





SQL (part 2)

- Recap.
- Database
- MySQL
- Import
- Import DB
- Insert
- Update
- Delete

สาขาวิชาวิศวกรรมการสื่อสารและสารสนเทศ

- 1/38 คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏเทพสตรี

1 . Recap.

1.1 Database

C : Create

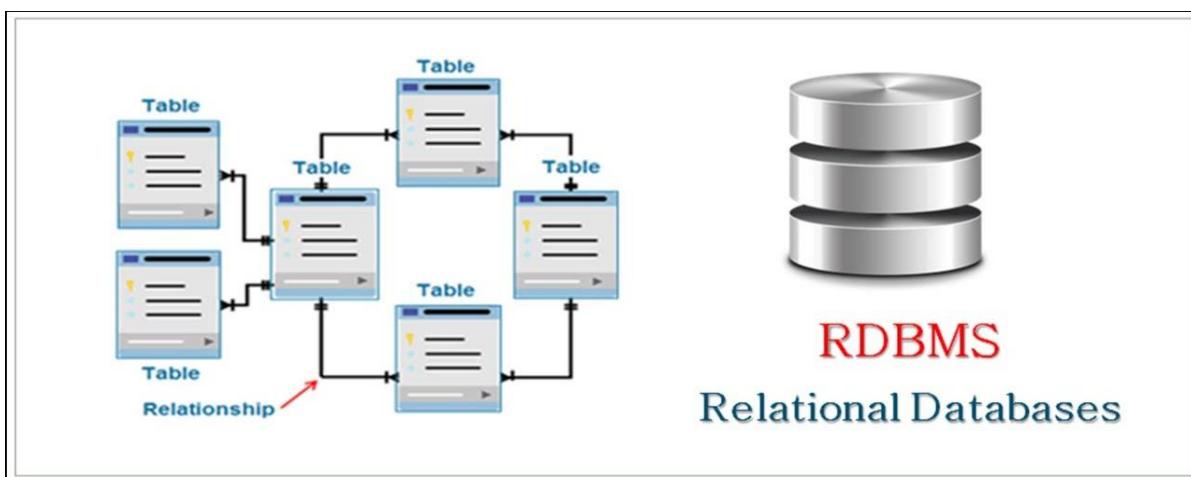
R : Read

U : Update

D : Delete

- 3/38 -

1.1 Database



Relational Database Management System

www.learncomputerscienceonline.com

- 4/38 -

1.2 MySQL



Website: <https://thepsatri.com/phpMyAdmin>

Username: data

Password: Data@2024

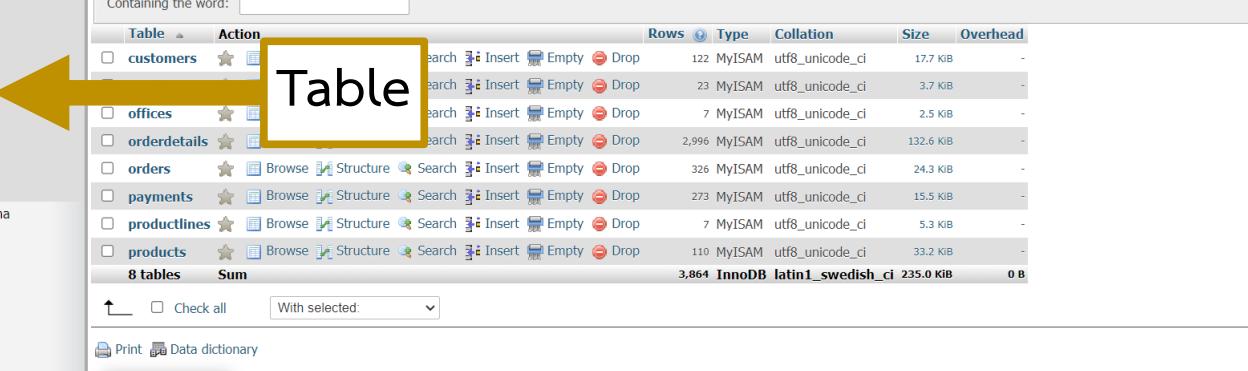
- 5/38 -

1.2 MySQL

The screenshot shows the phpMyAdmin interface. On the left, there's a sidebar with 'Recent' and 'Favorites' sections, showing databases like 'data_lab' and 'information_schema'. The main area has tabs for 'Databases', 'SQL', 'Status', 'Export', 'Import', 'Settings', 'Variables', 'Charsets', 'Engines', and 'Plugins'. A large yellow box highlights the 'Database' tab. To the right, there are several panels: 'Database server' (Server: localhost via UNIX socket, Server type: MariaDB, Server connection: SSL is not being used, Server version: 10.4.10-MariaDB - MariaDB Server, Protocol version: 10, User: data@localhost, Server charset: cp1252 West European (latin1)); 'Web server' (Apache/2, Database client version: libmysql - mysqld 5.0.12-dev - 20150407 - \$Id: 7cc7cc96e675fd72e5cf0f267f48e167c2abb2 \$, PHP extension: mysqli curl mbstring); and 'phpMyAdmin' (Version information: 4.9.2, Documentation, Official Homepage, Contribute, Get support, List of changes, License). At the bottom, there's a 'Console' link.

- 6/38 -

1.2 MySQL



The screenshot shows the phpMyAdmin interface for the 'data_lab' database. The left sidebar lists tables: customers, employees, offices, orderdetails, orders, payments, productlines, and products. The main area displays a table of database statistics. A large yellow arrow points from the left sidebar to the 'Table' heading in the main table list. A yellow box highlights the 'Table' heading.

Table	Action	Rows	Type	Collation	Size	Overhead
customers	Search Insert Empty Drop	122	MyISAM	utf8_unicode_ci	17.7 Kib	-
offices	Search Insert Empty Drop	23	MyISAM	utf8_unicode_ci	3.7 Kib	-
orderdetails	Search Insert Empty Drop	7	MyISAM	utf8_unicode_ci	2.5 Kib	-
orders	Browse Structure Search Insert Empty Drop	326	MyISAM	utf8_unicode_ci	24.3 Kib	-
payments	Browse Structure Search Insert Empty Drop	273	MyISAM	utf8_unicode_ci	15.5 Kib	-
productlines	Browse Structure Search Insert Empty Drop	7	MyISAM	utf8_unicode_ci	5.3 Kib	-
products	Browse Structure Search Insert Empty Drop	110	MyISAM	utf8_unicode_ci	33.2 Kib	-
8 tables	Sum	3,864	InnoDB	latin1_swedish_ci	235.0 Kib	0 B

Print Data dictionary

Create table

Name: Number of columns: 4

Go

- 7/38 -

1.2 MySQL

The screenshot shows the phpMyAdmin interface. On the left, the database structure for 'data_lab' is visible, including tables like customers, employees, offices, and orders. The 'employees' table is selected. The main area displays the contents of the 'employees' table with columns: employeeNumber, lastName, firstName, extension, email, officeCode, reportsTo, and jobTitle. The data includes rows for Diane Murphy, Mary Patterson, Jeff Firrelli, William Bondur, Gerard Hernandez, Anthony Bow, and others. A large yellow box highlights the 'Browse' button and the table data. Another yellow box highlights the database and table names at the top right. A large orange arrow points from the text 'Browse เพื่อดูข้อมูลในตาราง' to the 'Browse' button.

phpMyAdmin

Recent | Favorites

Server: localhost » Database: data_lab » Table: employees

Browse Structure SQL Search Insert Export

Showing rows 0 - 22 (23 total, Query took 0.0001 seconds.)

FROM `employees`

Show all Number of rows: 25 Filter rows: Search this table Sort by key

+ Options

employeeNumber lastName firstName extension email officeCode reportsTo jobTitle

1002 Murphy Diane x5800 dmurphy@classicmodelcars.com 1 NULL President

Mary x4611 mpatterso@classicmodelcars.com 1 1002 VP Sales

Jeff x9273 jfirrelli@classicmodelcars.com 1 1002 VP Marketing

William x4871 wpatterson@classicmodelcars.com 6 1056 Sales Manager (APAC)

Gerard x5408 gbondur@classicmodelcars.com 4 1056 Sales Manager (EMEA)

Anthony x5428 abow@classicmodelcars.com 1 1056 Sales Manager (NA)

1143 Bow

1165 Jennings Leslie x3291 ljennings@classicmodelcars.com 1 1143 Sales Rep

1166 Thompson Leslie x4065 lthompson@classicmodelcars.com 1 1143 Sales Rep

1188 Firrelli Julie x2173 jfirrelli@classicmodelcars.com 2 1143 Sales Rep

1216 Patterson Steve x4334 spatterson@classicmodelcars.com 2 1143 Sales Rep

1286 Tseng Foon Yue x2248 ftseng@classicmodelcars.com 3 1143 Sales Rep

1323 Vanauf George x4102 gvanauf@classicmodelcars.com 3 1143 Sales Rep

1337 Bondur Loui x6493 lbondur@classicmodelcars.com 4 1102 Sales Rep

1370 Hernandez Gerard x2028 ghernande@classicmodelcars.com 4 1102 Sales Rep

1401 Castillo Pamela x2759 pcastillo@classicmodelcars.com 4 1102 Sales Rep

1501 Bott Larry x2211 lbott@classicmodelcars.com 7 1102 Sales Rep

1504 Jones Barry x102 bjones@classicmodelcars.com 7 1102 Sales Rep

- 8/38 -

Browse เพื่อดูข้อมูลในตาราง

Database: data_lab
Table: employees

1.2 MySQL

Structure เพื่อตู้โครงสร้างของตาราง

#	Name	Type	Collation	Action	
1	customerNumber	int(11)	No None	Change Drop More	
2	customerName	varchar(50)	utf8_unicode_ci	No None	Change Drop More
3	contactLastName	varchar(50)	utf8_unicode_ci	No None	Change Drop More
4	contactFirstName	varchar(50)	utf8_unicode_ci	No None	Change Drop More
5	phone	varchar(50)	utf8_unicode_ci	No None	Change Drop More
6	addressLine1	varchar(50)	utf8_unicode_ci	No None	Change Drop More
7	addressLine2	varchar(50)	utf8_unicode_ci	Yes NULL	Change Drop More
8	city	varchar(50)	utf8_unicode_ci	No None	Change Drop More
9	state	varchar(50)	utf8_unicode_ci	Yes NULL	Change Drop More
10	postalCode	varchar(15)	utf8_unicode_ci	Yes NULL	Change Drop More
11	country	varchar(50)	utf8_unicode_ci	No None	Change Drop More
12	salesRepEmployeeNumber	int(11)		Yes NULL	Change Drop More
13	creditLimit	double		Yes NULL	Change Drop More

Print [Propose table structure](#) [Move columns](#) [Normalize](#)
Add 1 column(s) after creditLimit [Go](#)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	customerNumber	122	A	No	

Create an index on 1 columns [Go](#)

1.2 MySQL

SQL เพื่อเขียนคำสั่ง query

```
Run SQL query/queries on table data_lab.customers:
```

1 SELECT * FROM `customers` WHERE 1

SELECT * INSERT UPDATE DELETE Clear Format Get auto-saved query Bind parameters

[Delimiter :] Show this query here again Retain query box Rollback when finished Enable foreign key checks [Go](#)

customerNumber
customerName
contactLastName
contactFirstName
phone
addressLine1
addressLine2
city
state
postalCode
country
salesRepEmployeeNumber
creditLimit

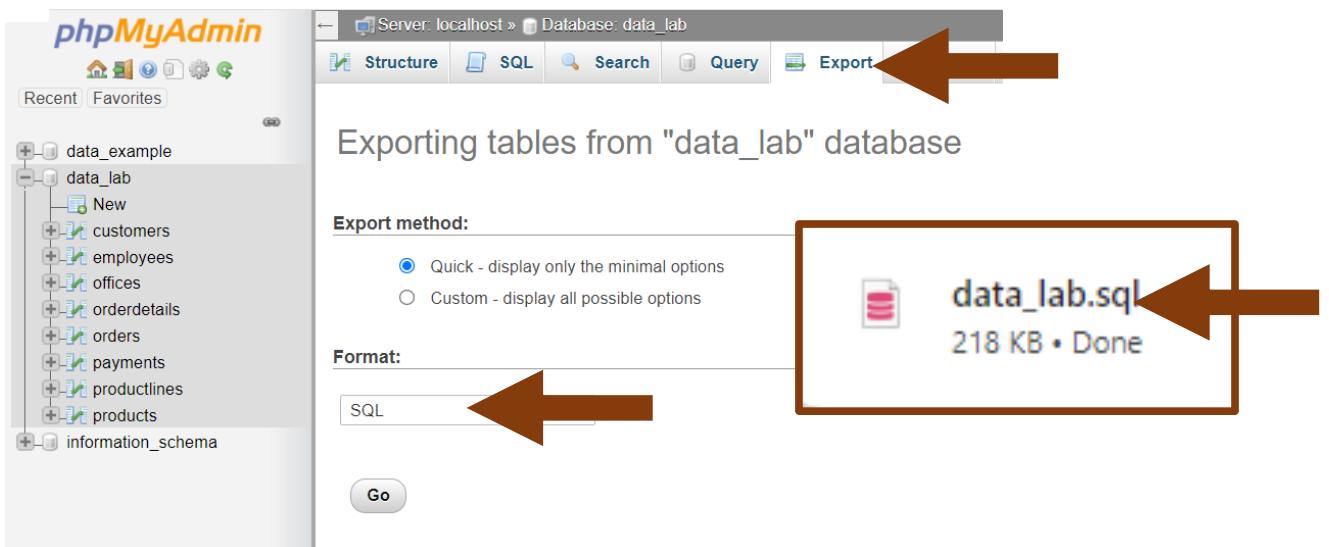
- 10/38 -

2 . Import

- 11/38 -

2 . Import

Export



- 12/38 -

2 . Import

Back-End (DirectAdmin)

DirectAdmin Login Page

Please enter your Username and Password

Username:	s44010305
Password:	*****

Website: <http://thepsatri.com:2222>

Username: s + รหัสนักศึกษา

Password: Ss + รหัสนักศึกษา

- 13/38 -

2 . Import

DirectAdmin

Your Account

Domain Setup Site Summary Statistics / Logs
Change Password FTP Management
Login History Subdomain Management
DNS Management MySQL Management
Support Center Password Protected Directories
Installed Perl Modules File Manager
Create/Restore Backups

E-Mail Management

E-Mail Accounts SPAM Filters

Message System

Your Account

Bandwidth

Disk Space

Used	Max
Disk Space (MB) 0.0	9000
Bandwidth (GB) 0.0000	8.7890
E-Mails 1	10
Ftp Accounts 1	1
Databases 0	5
Inodes 0	unlimited

Current Domain
44010305.thepsatri.com

Logout



- 14/38 -

2 . Import

MySQL Management

The screenshot shows the MySQL Management interface. At the top, there are five icons: Home, Webmail, Password, Help, and Files. Below them is a message: "Create new Database" and "You won't be able to login to phpMyAdmin until you create a Database." A large brown arrow points from the text message down to the "Create new Database" form. The form has tabs for "Database", "Number of Users", and "Download Backup". The "Database" tab is selected. It contains fields for "Database Name" (s44010305_lab), "Database Username" (s44010305_lab), "Username Password" (redacted), and "Confirm Password" (redacted). There are "Repair", "Check", and "Optimize" buttons at the bottom. To the right, a modal window titled "Create new Database" shows the same input fields with the "Create" button highlighted.

- 15/38 -

2 . Import

phpMyAdmin

The screenshot shows the phpMyAdmin login page. At the top is the phpMyAdmin logo featuring a sailboat icon and the text "phpMyAdmin". Below it is a "Welcome to phpMyAdmin" message. A "Language" dropdown menu is set to "English". A "Log in" button is followed by input fields for "Username" (s44010305) and "Password" (redacted). A "Go" button is at the bottom right of the form.

- 16/38 -

2 . Import

Import Database

The screenshot shows the phpMyAdmin interface for a database named 's44010305_lab'. A large red arrow points from the text 'เลือก database' (Select database) to the database tree on the left, where 's44010305_lab' is selected. Another red arrow points from the word 'Import' to the 'Import' tab in the top navigation bar.

- 17/38 -

2 . Import

Import Database

The screenshot shows the phpMyAdmin interface for the same database 's44010305_lab'. A large red arrow points to the 'Import' tab in the top navigation bar. The main area displays the message 'Importing into the database "s44010305_lab"'.

- 18/38 -

2 . Import

Import Database

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** localhost
- Database:** s44010305_lab
- Structure, SQL, Search, Query, Export, Import, Operations, More** tabs are visible.
- Import Status:** "Import has been successfully finished, 38 queries executed. (data_lab.sql)"
- Log Output:** MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)
-- phpMyAdmin SQL Dump -- version 4.9.2 -- https://www.phpmyadmin.net/ -- -- Host: localhost -- Generation Time: Oct 14, 2024 at 11:18 AM -- Server version: 10.4.10-MariaDB -- PHP Version: 7.3.12 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO"
[Edit inline] [Edit] [Create PHP code]
- Log Output:** MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)
SET AUTOCOMMIT = 0
[Edit inline] [Edit] [Create PHP code]
- Log Output:** MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)
START TRANSACTION
[Edit inline] [Edit] [Create PHP code]
- Progress:** - 19/38 -

3. Insert

- 21/38 -

3. Insert

ก่อน Insert ตาราง employees จะมี 23 rows

The screenshot shows the phpMyAdmin interface for the 'employees' table in the 's44010305_lab' database. The table has 23 rows. The columns are: employeeNumber, lastName, firstName, extension, and email. The data includes rows for Murphy, Patterson, Firrelli, and Patterson.

	employeeNumber	lastName	firstName	extension	email
<input type="checkbox"/> Edit Copy Delete	1002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com
<input type="checkbox"/> Edit Copy Delete	1056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com
<input type="checkbox"/> Edit Copy Delete	1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com
<input type="checkbox"/> Edit Copy Delete	1088	Patterson	William	x4871	wpatterson@classicmodelcars.com

- 22/38 -

3. Insert

```
INSERT INTO employees  
VALUES (employeeNumber, lastName, firstName)  
(44010305, 'Dangkham', 'Piyapong')
```

<input type="checkbox"/>		Edit		Copy		Delete	1625	Kato	Yoshimi	x102	ykato@classicmodelcars.com	
<input type="checkbox"/>		Edit		Copy		Delete	1702	Gerard	Martin	x2312	mgerard@classicmodelcars.com	
<input type="checkbox"/>		Edit		Copy		Delete	44010305	Dangkham	Piyapong			
<hr/>												
<hr/>												
	<input type="checkbox"/>	Check all	With selected:			Edit		Copy		Delete		Export

- 23/38 -

3. Insert

หลัง Insert ตาราง employees จะมี 24 rows

The screenshot shows the phpMyAdmin interface for the 'employees' table. The table has columns: employeeNumber, lastName, firstName, extension, and email. The data includes rows for Murphy, Patterson, Firrelli, and Patterson again. The status bar at the bottom indicates 24/38 rows.

		Edit		Copy		Delete	employeeNumber	lastName	firstName	extension	email
<input type="checkbox"/>		Edit		Copy		Delete	1002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com
<input type="checkbox"/>		Edit		Copy		Delete	1056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com
<input type="checkbox"/>		Edit		Copy		Delete	1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com
<input type="checkbox"/>		Edit		Copy		Delete	1088	Patterson	William	x4871	wpatterson@classicmodelcars.com
<input type="checkbox"/>		Edit		Copy		Delete	1102	Bondur	Gerard	x5408	gbondur@classicmodelcars.com

- 24/38 -

3. Insert

ให้เพิ่ม employees อีก 1 คน มีรายละเอียดดังนี้

employeeNumber	555
firstName	Kylie
lastName	Minogue
officeCode	5
email	kylie@kylie.com

- 25/38 -

3. Insert

ให้เพิ่มข้อมูล payments มีรายละเอียดดังนี้

customerNumber	495
checkNumber	GG155234
paymentDate	2024-05-14
amount	57890.75

- 26/38 -

4 . Update

- 27/38 -

4 . Update

<input type="checkbox"/>	Edit Copy Delete	1625	Kato	Yoshimi	x102	ykato@classicmodelcars.com
<input type="checkbox"/>	Edit Copy Delete	1702	Gerard	Martin	x2312	mgerard@classicmodelcars.com
<input type="checkbox"/>	Edit Copy Delete	44010305	Dangkham	Piyapong		

employeeNumber 44010305
lastName Dangkham
firstName Piyapong --> Piya

- 28/38 -

4 . Update

```
UPDATE employees  
SET firstName = 'Piya'  
WHERE employeeNumber = 44010305
```

<input type="checkbox"/>	Edit	Copy	Delete	1625	Kato	Yoshimi	x102	ykato@classicmodelcars.com
<input type="checkbox"/>	Edit	Copy	Delete	1702	Gerard	Martin	x2312	mgerard@classicmodelcars.com
<input type="checkbox"/>	Edit	Copy	Delete	44010305	Dangkham	Piya		

- 29/38 -

4 . Update

ให้แก้ไขข้อมูล employees รหัส 555 ให้มีรายละเอียดดังนี้

firstName	Beyonce
lastName	Knows

- 30/38 -

4 . Update

ให้แก้ไขข้อมูล employees ที่มี jobTitle เป็น Sales Rep ให้มี officeCode เป็น 7

- 31/38 -

- 32/38 -

asfsafasf

5 . Delete

- 33/38 -

5 . Delete

ก่อน Insert ตาราง employees จะมี 24 rows

The screenshot shows the phpMyAdmin interface for the 'employees' table in the 's44010305_lab' database. The left sidebar lists various tables: information_schema, s44010305_lab, New, customers, employees, offices, orderdetails, orders, payments, productlines, and products. The 'employees' table is selected. The main area displays the results of a SELECT query: 'Showing rows 0 - 23 (24 total, Query took 0.0001 seconds.)'. Below the results, there is an SQL editor with the query 'SELECT * FROM `employees`' and several control buttons: Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. At the bottom, there are options to Show all (checkbox), Number of rows (dropdown set to 25), Filter rows (text input 'Search this table'), Sort by key (dropdown set to None), and a 'Options' section with a 'T' icon and a 'Rows' dropdown set to 25.

employeeNumber	lastName	firstName	extension	email
1002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com
1056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com
1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com
1088	Patterson	William	x4871	wpatterson@classicmodelcars.com

- 34/38 -

5 . Delete

**DELETE FROM employees
WHERE employeeNumber = 44010305**

Showing rows 0 - 22 (23 total, Query took 0.0003 seconds.)

SELECT * FROM `employees`

	employeeNumber	lastName	firstName	extension	email
<input type="checkbox"/> Edit Copy Delete	1002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com
<input type="checkbox"/> Edit Copy Delete	1056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com
<input type="checkbox"/> Edit Copy Delete	- 35/38 -	1076	Firrelli	Jeff	x9273

5 . Delete

ให้ลบข้อมูล employees ที่อยู่ office หมายเลข 3

5. Delete

ให้ลบข้อมูล employees ที่มีนามสกุลเดียวกับพนักงานรหัส 1056

- 37/38 -

37



- 38/38 -