



รายงาน

ระบบการจูงตัวการรถไฟแห่งประเทศไทย

เสนอ

อาจารย์ เอิญ สุริยะฉาย

จัดทำโดย

- | | |
|------------------------------|---------------------|
| 1. นาย ประเมษฐ์ พกุลานนท์ | 6404062620010 เซค 1 |
| 2. นาย อลงกรณ์ มานะสกุลกิจ | 6404062620028 เซค 1 |
| 3. นาย รุ่งสิทธิ์ ประทีปะเสน | 6404062620150 เซค 1 |
| 4. นาย กฤติภัทร ไชประณิธาน | 6404062620176 เซค 1 |
| 5. นาย ศุภกฤต จุฬามณี | 6404062620184 เซค 1 |
| 6. นาย กันต์ธี จิตรแก้ว | 6404062610456 เซค 1 |

นักศึกษาชั้นปีที่ 1

รายงานนี้เป็นส่วนหนึ่งของวิชา ระบบฐานข้อมูล (040613301)

สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยพระจอมเกล้าพระนครเหนือ

ภาคเรียนที่ 2 ปีการศึกษา 2564

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	3-4
1.1 คำอธิบายระบบ	3
1.2 ฟังก์ชันการทำงานภายในระบบ	3-4
บทที่ 2 ER-DIAGRAM	5
บทที่ 3 แปลง ER-DIAGRAM เป็นตาราง	6-7
บทที่ 4 DATA DICTIONARY	8-11
บทที่ 5 Create Table	12-16
5.1) การสร้างตารางใน PhpMyAmin	15
5.2) การแก้ไขข้อมูลตารางใน PhpMyAmin	16
5.3) วิธีการ Create Table โดย SQL	16
บทที่ 6 เพิ่มขบวนการไฟ	17-27
6.1) โค้ดการ Insert ข้อมูล	17-21
6.2) โค้ดการ delete ข้อมูล	22-27
บทที่ 7 การติดต่อฐานข้อมูล	28-34
7.1) การติดต่อฐานข้อมูล ในภาษาPHP	28
7.2) การแสดงข้อมูลจากฐานข้อมูลบนหน้าเว็บไซต์	28-30
7.3) การแก้ไขข้อมูลจากฐานข้อมูล	30
7.4) relational algebra	31-34

บทที่ 1

บทนำ

1.1 คำอธิบายระบบ

ระบบการจองตั๋วการรถไฟแห่งประเทศไทย (Railway Reservation System) เป็นระบบทำงานในลักษณะของ Web Application โดยจะแบ่งการทำงานเป็นสองระบบ ได้แก่ **ส่วนยูสเซอร์** หรือ ส่วนที่ให้ผู้ใช้งานสามารถจองตั๋วรถไฟได้ และ **ส่วนแอดมิน** หรือ ส่วนที่ให้ผู้ดูแลระบบสามารถแก้ไข เพิ่ม ลบเส้นทางเดินรถขบวนรถไฟตู้รถไฟและสามารถเข้าถึงรายการจองตั๋วและรายการชำระเงินทั้งหมดของทุกยูสเซอร์ รวมไปถึงการตรวจสอบรายการขบวนรถไฟที่วิ่งอยู่ทั้งหมดได้

- **ส่วนยูสเซอร์** ยูสเซอร์สามารถจองตั๋วรถไฟโดยเลือกสถานีต้นทาง สถานีปลายทาง และวันที่เดินทาง ระบบจะให้ยูสเซอร์สามารถเลือกรถไฟและตู้ได้ ซึ่งยูสเซอร์สามารถเลือกประเภทได้ตามที่ต้องการ เช่น ประเภทรถไฟ เวลาออก ประเภทตู้รถไฟ ชั้น ราคาต่อที่นั่ง และที่นั่ง เป็นต้น และในส่วนของการชำระเงินจะเป็นการชำระเงินแบบ QR code ที่สแกนผ่านแอปมือถือเพื่อจ่ายเงินและอัปโหลดรูปภาพการชำระเงินเพื่อให้แอดมินตรวจสอบ หลังจากนั้นจึงจะสามารถปริ้นตั๋วรถไฟได้
- **ส่วนแอดมิน** แอดมินสามารถแก้ไข เพิ่ม ลบ เส้นทางเดินรถ ขบวนรถไฟ ตู้รถไฟ และสามารถเข้าถึงรายการจองตั๋วและรายการชำระเงินทั้งหมดของทุกยูสเซอร์ รวมไปถึงการตรวจสอบรายการขบวนรถไฟที่วิ่งอยู่ทั้งหมดได้

1.2 ฟังก์ชันการทำงานภายในระบบ

- i) **การสมัครสมาชิก(Register) และเข้าสู่ระบบ(Login) :**
 - 1) ยูสเซอร์สามารถเปลี่ยนชื่อจริง นามสกุล อีเมลได้
 - 2) ในขั้นตอนการสมัครสมาชิกยูสเซอร์**ต้องกรอกยูสเซอร์เนม (Username) ชื่อจริง นามสกุล เลขบัตรประจำตัวประชาชน อีเมล (Email) และพาสเวิร์ด (Password)
- ii) **การจองตั๋วรถไฟ (Railway Reservation) :**
 - 1) เลือกสถานีต้นทาง และปลายทาง รวมถึงวันที่ต้องการจะไป
 - 2) เลือกขบวนรถไฟและเลือกตู้รถไฟ
 - 3) เลือกที่นั่ง

iii) การชำระเงิน

- 1) ยูสเซอร์สามารถชำระเงินทันทีหรือชำระเงินภายหลังได้
- 2) ยูสเซอร์สามารถชำระเงินทันทีโดยการ Scan QR Code ได้เลย

iv) ข้อมูลส่วนตัว (Personal Information) :

- 1) ยูสเซอร์สามารถแก้ไขข้อมูลส่วนตัวด้วยตัวเองได้ตลอดเวลาที่แก้ไขข้อมูลผู้โดยสาร

v) ยกเลิกการจองตั๋วรถไฟ (Cancel Reservation) :

- 1) ยูสเซอร์สามารถยกเลิกการจองตั๋วรถไฟได้

หมายเหตุ หากชำระเงินแล้วสามารถยกเลิกการจองตั๋วได้แต่จะไม่ได้รับเงินคืน

vi) แก้ไข ลบ เพิ่ม และตรวจสอบรายการขบวนรถไฟ:

- 1) Admin สามารถแก้ไข เพิ่ม ลบ เส้นทางเดินรถ ขบวนรถไฟ ตู้รถไฟ และสามารถเข้าถึงรายการจองตั๋วได้ รวมไปถึงการตรวจสอบรายการขบวนรถไฟที่วิ่งอยู่ทั้งหมดได้

vii) ประวัติการจอง (Reservation History) :

1) ส่วนยูสเซอร์

- i. แอดมินสามารถตรวจสอบ รหัสตัวโดยสาร ชื่อผู้ใช้ ชื่อจริง นามสกุล วันที่ของยูสเซอร์ได้
- ii. แอดมินสามารถตรวจสอบหลักฐานการชำระเงินของยูสเซอร์ เพื่อยืนยันการออกตั๋วรถไฟให้ยูสเซอร์ได้ ซึ่งจะระบุหมายเลขบิล หมายเลขผู้ใช้ หมายเลขการจอง ยอดต้องชำระ และหลักฐานการชำระเงิน

2) ส่วนยูสเซอร์

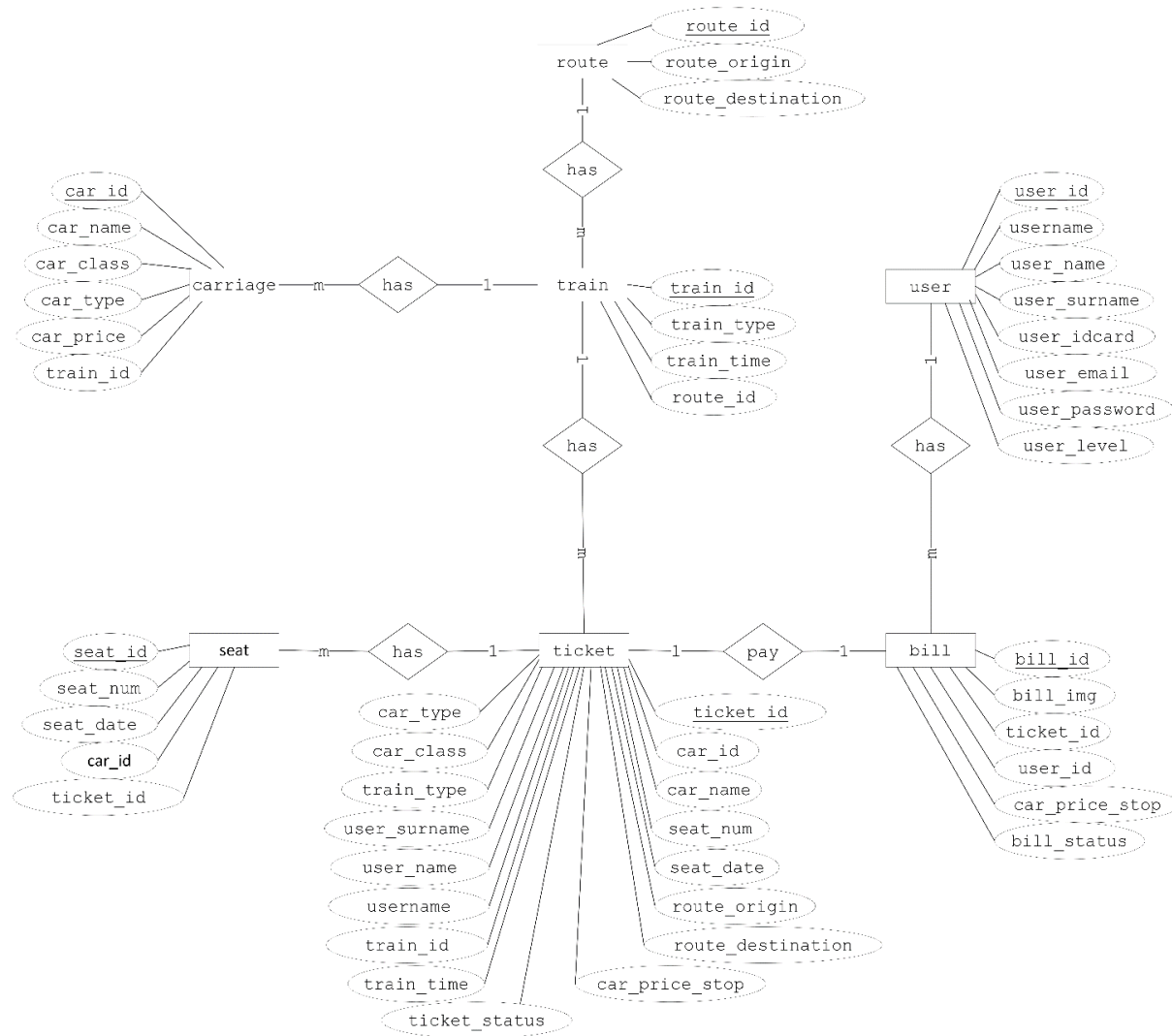
- i. ยูสเซอร์สามารถเข้ามาปรี้นตั๋วรถไฟได้ที่ ประวัติการจองตั๋วรถไฟ
- ii. กรณีที่ไม่ได้ชำระเงินในหน้าชำระเงินสามารถชำระเงินในหน้าต่างนี้ภายหลังได้ แล้วทางแอดมินจะตรวจสอบให้ จึงจะสามารถทำการปรี้นตั๋วรถไฟได้

viii) ออกจากระบบ (Logout)

- 1) ยูสเซอร์สามารถออกจากระบบเพื่อเปลี่ยนบัญชีผู้ใช้ได้

บทที่ 2

ER-DIAGRAM



ภาพที่ 2.1 ER Diagram

บทที่ 3

แปลง ER เป็นตาราง

ตาราง user

user_id	username	user_name	user_surname	user_idcard	user_email	user_password	user_level
*PK							

ตาราง bill

bill_id	bill_img	car_price_stop	bill_status	ticket_id	user_id
*PK				*FK	*FK

ตาราง ticket

ticket_id	car_name	car_id	seat_num	seat_date	route_origin	train_type	car_type	car_class
*PK		*FK						

ตาราง ticket(ต่อ)

route_destination	car_price_stop	ticket_status	train_time	username	user_name	user_surname	train_id
							*FK

ตาราง seat

seat_id	seat_num	seat_date	car_id	ticket_id
*PK				*FK

ตาราง train

train_id	trian_type	train_time	route_id
*PK			*FK

ตาราง carriage

car_id	car_name	car_class	car_type	car_price	train_id
*PK					*FK

ตาราง route

route_id	route_origin	route_destination
*PK		*FK

บทที่ 4

DATA DICTIONARY

Table Name : Train

Attribute Name	Description	Type	Key Type	Reference Table
train_id	รหัสรถไฟ	int	pk	
train_type	ชนิดรถไฟ	varchar		
train_time	เวลารถไฟ	varchar		
route_id	รหัสเส้นทาง	int	fk	Route

Table Name : Bill

Attribute Name	Description	Type	Key Type	Reference Table
bill_id	รหัสบิล	int	pk	
bill_img	ใบชำระเงิน	varbinary		
ticket_id	รหัสตั๋ว	int	fk	Ticket
user_id	รหัสผู้ใช้	int		
car_price_stop	ราคาสุทธิ	double		
bill_status	สถานะบิล	varchar		

Table Name : Carriage

Attribute Name	Description	Type	Key Type	Reference Table
car_id	รหัสตู้โดยสาร	int	pk	
car_name	ชื่อตู้โดยสาร	varchar		
car_class	ระดับตู้โดยสาร	varchar		
car_type	ประเภทตู้โดยสาร	varchar		
car_price	ราคาตู้โดยสาร	double		
train_id	หมายเลขขบวน	int	fk	Train

Table Name : Route

Attribute Name	Description	Type	Key Type	Reference Table
route_id	รหัสเส้นทาง	int	pk	
route_origin	สถานีต้นทาง	varchar		
route_destination	สถานีปลายทาง	varchar		

Table Name : Seat

Attribute Name	Description	Type	Key Type	Reference Table
seat_id	รหัสที่นั่ง	int	pk	
seat_num	เลขที่นั่ง	int		
seat_date	วันที่เดินทาง	varchar		
car_id	รหัสตู้โดยสาร	int	fk	Carriage
ticket_id	รหัสตั๋วโดยสาร	int	fk	Ticket

Table Name : Ticket

Attribute Name	Description	Type	Key Type	Reference Table
ticket_id	รหัสตั๋วโดยสาร	int	pk	
car_id	รหัสตู้โดยสาร	varchar	fk	Carriage
car_name	ชื่อตู้โดยสาร	varchar		
seat_num	เลขที่นั่ง	varchar		
seat_date	วันที่เดินทาง	varchar		
route_origin	สถานีต้นทาง	varchar		
route_destination	สถานีปลายทาง	varchar		
username	ชื่อผู้ใช้	varchar		
user_name	ชื่อ	varchar		

user_surname	นามสกุล	varchar		
train_type	ชนิดรถไฟ	varchar		
car_class	ระดับตู้โดยสาร	varchar		
car_type	ประเภทตู้โดยสาร	varchar		
car_price_status	สถานะการชำระเงิน	int		
ticket_status	สถานะการจองตั๋ว	varchar		
train_time	เวลาเดินทาง	varchar		
train_id	รหัสรถไฟ	int	fk	Train

Table Name : User

Attribute Name	Description	Type	Key Type	Reference Table
user_id	รหัสผู้ใช้	int	pk	
username	ชื่อผู้ใช้	varchar		
user_name	ชื่อ	varchar		
user_surname	นามสกุล	varchar		
user_idcard	เลขบัตรประจำตัวประชาชน	bigint		
user_email	อีเมลล์ผู้ใช้	varchar		
user_password	รหัสผ่าน	varchar		
user_level	ระดับผู้ใช้	varchar		

บทที่ 5

Create Table

Create Table

เพื่อสร้างตารางในฐานข้อมูล

ตาราง user

```
CREATE TABLE `user` (
  `user_id` int(11) NOT NULL,
  `username` varchar(100) NOT NULL,
  `user_name` varchar(100) NOT NULL,
  `user_surname` varchar(100) NOT NULL,
  `user_idcard` bigint(13) NOT NULL,
  `user_email` varchar(100) NOT NULL,
  `user_password` varchar(100) NOT NULL,
  `user_level` varchar(100) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ภาพที่ 5.0.1 Create Table

ตาราง ticket

```
CREATE TABLE `ticket` (
  `ticket_id` int(11) NOT NULL,
  `car_id` varchar(11) NOT NULL,
  `car_name` varchar(1000) NOT NULL,
  `seat_num` varchar(100) NOT NULL,
  `seat_date` varchar(100) NOT NULL,
  `route_origin` varchar(100) CHARACTER SET
  `route_destination` varchar(100) NOT NULL,
  `username` varchar(100) NOT NULL,
  `user_name` varchar(100) NOT NULL,
  `user_surname` varchar(100) NOT NULL,
  `train_type` varchar(100) NOT NULL,
  `car_class` varchar(100) NOT NULL,
  `car_type` varchar(100) NOT NULL,
  `car_price_stop` int(11) NOT NULL,
  `ticket_status` varchar(100) NOT NULL,
  `train_time` varchar(100) NOT NULL,
  `train_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ภาพที่ 5.0.2 Create Table

Create Table

เพื่อสร้างตารางในฐานข้อมูล

ตาราง bill

```
CREATE TABLE `bill` (  
  `bill_id` int(11) NOT NULL,  
  `bill_img` varbinary(100) NOT NULL,  
  `ticket_id` int(11) NOT NULL,  
  `user_id` int(11) NOT NULL,  
  `car_price_stop` double(64,2) NOT NULL,  
  `bill_status` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ภาพที่ 5.0.3 Create Table

ตาราง route

```
CREATE TABLE `route` (  
  `route_id` int(11) NOT NULL,  
  `route_origin` varchar(100) NOT NULL,  
  `route_destination` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ภาพที่ 5.0.4 Create Table

Create Table

เพื่อสร้างตารางในฐานข้อมูล

ตาราง train

```
CREATE TABLE `train` (
  `train_id` int(11) NOT NULL,
  `train_type` varchar(100) NOT NULL,
  `train_time` varchar(100) NOT NULL,
  `route_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ภาพที่ 5.0.5 Create Table

ตาราง seat

```
CREATE TABLE `seat` (
  `seat_id` int(11) NOT NULL,
  `seat_num` int(11) NOT NULL,
  `seat_date` varchar(100) NOT NULL,
  `car_id` int(11) NOT NULL,
  `ticket_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ภาพที่ 5.0.6 Create Table

ตาราง carriage

```
CREATE TABLE `carriage` (
  `car_id` int(11) NOT NULL,
  `car_name` varchar(100) NOT NULL,
  `car_class` varchar(100) NOT NULL,
  `car_type` varchar(100) NOT NULL,
  `car_price` double(64,2) NOT NULL,
  `train_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ภาพที่ 5.0.7 Create Table

5.1) การสร้างตารางใน PhpMyAdmin

ขั้นตอนที่ 1 ระบุชื่อ table และจำนวนคอลัมป์ สามารถเพิ่ม หรือ ลบ จำนวนคอลัมป์ได้ในภายหลัง

ภาพที่ 5.1.1 Create Table

ขั้นตอนที่ 2 ระบุชื่อคอลัมป์ ชนิดข้อมูล และขนาดของข้อมูล

Name	Type	Length/Values
user_id	INT	11
	INT	
	INT	
	INT	

ภาพที่ 5.1.2 Create Table

ขั้นตอนที่ 3

Index -primary เป็นการระบุคอลัมป์ เป็น PRIMARY KEY

AI เป็นการระบุให้คอลัมป์นั้นไม่ซ้ำกันและเพิ่มข้อมูลทันที โดยไม่ต้องเพิ่มข้อมูลในช่องนั้นๆ

Move column ใช้ในการจัดเรียงคอลัมป์

Index	A.I.	Comments	Virtuality	Move column
PRIMARY	<input checked="" type="checkbox"/>			
---	<input type="checkbox"/>			
---	<input type="checkbox"/>			
---	<input type="checkbox"/>			

ภาพที่ 5.1.3 Create Table

5.2) การแก้ไขข้อมูลตารางใน PhpMyAdmin

Change ใช้แก้ไขข้อมูลในคอลัมป์นั้นๆ

Drop ใช้ในการลบคอลัมป์นั้นๆ

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	user_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	user_name	varchar(100)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/> 3	user_surname	varchar(100)	utf8_general_ci		No	None			Change Drop More
<input type="checkbox"/> 4	user_pass	int(11)			No	None			Change Drop More

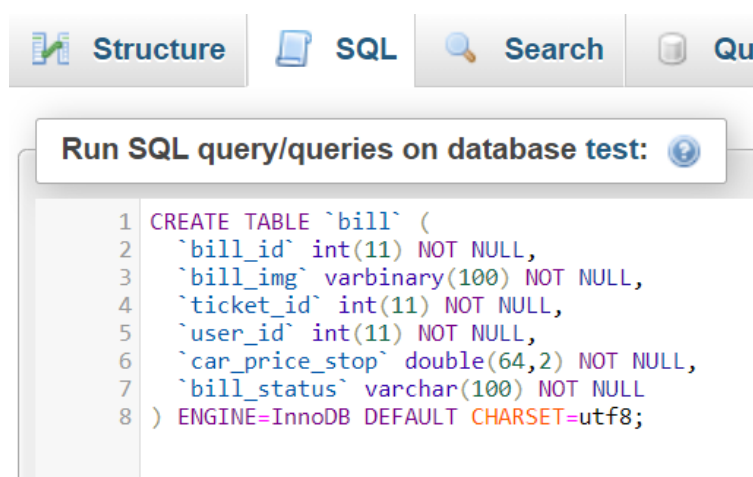
ภาพที่ 5.2.1 Create Table

นอกจากนี้เราสามารถสร้าง table โดยวิธีการเขียนโค้ด sql ได้อีกด้วย

โดยใช้คำสั่ง Create Table

5.3) วิธีการ Create Table

CREATE TABLE `ชื่อตาราง` (`ชื่อคอลัมป์` ชนิดข้อมูล(ระบุขนาดข้อมูล) NOT NULL,)



ภาพที่ 5.3.1 Create Table

บทที่ 6

Insert & Delete Data

6.1) โค้ดการ Insert ข้อมูล

6.1.1) การ Insert ข้อมูลของตาราง train

```
<?php
session_start();
include('server.php');
$train_id = $_REQUEST['train_id'];
$train_type = $_REQUEST['train_type'];
$train_time = $_REQUEST['train_time'];
$route_id = $_REQUEST['route_id'];

if(empty($train_id) or empty($train_type) or empty($train_time) or empty($route_id)){
    header('location: input.php');
    $_SESSION['alert'] = 'error';
    $_SESSION['alert_status'] = 'พบข้อมูลว่าง';
}
else{
    $sql = "INSERT INTO train
VALUES ('$train_id',' $train_type',' $train_time',' $route_id');
";
}
```

ภาพที่ 6.1.1.1 Insert

- 1) มีการประกาศเรียกใช้ function session start เพื่อเรียกใช้ตัวแปร \$_SESSION
- 2) มีการเชื่อมดาต้าเบสกับ server (server.php)
- 3) มีการรับค่ามาจากฟอร์มใน html ผ่านคำสั่ง \$_REQUEST
- 4) มีการตรวจสอบว่าถ้า user ไม่ได้กรอกค่าลงในฟอร์ม จะทำการรีเฟรชหน้าเว็บไซต์
- 5) มีการนำข้อมูลที่เก็บไว้ในตัวแปร \$train_id,\$train_type,\$train_time,\$route_id ไป

Insert ข้อมูลในตาราง train ในดาต้าเบส

```
$objQuery = mysqli_query($conn, $sql);

if ($objQuery) {
    echo "New record Inserted successfully";
    $_SESSION['alert'] = 'error';
    $_SESSION['alert_status'] = 'เพิ่มข้อมูลสำเร็จ';
    echo "<br>";
    echo "$train_id,$train_type,$train_time,$route_id";
} else {
    echo "Error : Input";
    $_SESSION['alert'] = 'error';
    $_SESSION['alert_status'] = 'ใช้หมายเลขบนนี้ไปแล้ว';
    header('location: input.php');
}
}
```

ภาพที่ 6.1.1.5 Insert

- 6) มีการเรียกใช้ฟังก์ชัน mysqli_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ Insert ข้อมูลเข้าไป
- 7) มีการใช้ if เข้ามาแยกระหว่างข้อมูลที่สามารถ insert ผ่านและไม่ผ่าน

เพิ่มขบวนรถไฟ

หมายเลขขบวน

ประเภท

เวลาออก

รหัสเส้นทาง

ภาพ หน้าต่างเพิ่มขบวนรถไฟ

- 1) ทำการใส่ข้อมูลลงในฟอร์มทั้ง 4 ช่อง (หมายเลขขบวน,ประเภท,เวลาออก,รหัสเส้นทาง)
- 2) หลังจากนั้น กดเพิ่มขบวนรถไฟ
- 3) ตัว code จะพาไปหน้าต่างที่แสดงข้อมูลที่ถูกเพิ่มเข้าไป

ข้อความหลังจากทำการ insert ข้อมูล

New record Inserted successfully
12456,ด่วน,10:58,1

ภาพข้อความ insert ข้อมูล

- 4) มีการแสดงข้อมูลว่าข้อมูลที่เพิ่มเข้าไป

6.1.2) การ Insert ข้อมูลของตาราง route

```
<?php
session_start();
include('server.php');
$route_id = $_REQUEST['route_id'];
$route_origin = $_REQUEST['route_origin'];
$route_destiantion = $_REQUEST['route_destination'];

if(empty($route_id)){
    $_SESSION['alert'] = 'error';
    $_SESSION['alert_status'] = 'พบข้อมูลว่าง';
    header('route_add.php');
}else
{
    $sql = "INSERT INTO route
VALUES ('$route_id','$route_origin','$route_destiantion')";

    $objQuery = mysqli_query($conn, $sql);
```

ภาพที่ 6.1.2.1 Insert

- 1) มีการประกาศเรียกใช้ function session start เพื่อเรียกใช้ตัวแปร \$_SESSION
- 2) มีการเชื่อมดาต้าเบสกับ server (server.php)

- 3) มีการรับค่ามาจากฟอร์มใน html ผ่านคำสั่ง \$_REQUEST
- 4) มีการตั้งค่าว่างถ้า user ไม่ได้กรอกค่าลงในฟอร์ม จะทำการรีเฟรชหน้าเว็บไซต์
- 5) มีการนำข้อมูลที่เก็บไว้ในตัวแปร \$route_id , \$route_origin , \$route_destination ไป Insert ข้อมูลในตาราง route ในดาต้าเบส
- 6) มีการเรียกใช้ฟังก์ชัน mysqli_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ Insert ข้อมูลเข้าไป

```
if ($objQuery) {
    echo "New record Inserted successfully";
    //header('location: admin_edit_route_add.php');
    $_SESSION['alert'] = 'error';
    $_SESSION['alert_status'] = 'เพิ่มข้อมูลสำเร็จ';
} else {
    echo "Error : Input";
    $_SESSION['alert'] = 'error';
    $_SESSION['alert_status'] = 'ใช้หมายเลขนี้ไปแล้ว';
    header('location: route_add.php');
}
```

ภาพที่ 6.1.2.6

Insert

- 7) ถ้าข้อมูลถูกต้องจะแจ้งเตือนว่าเพิ่มข้อมูลสำเร็จ ถ้ามี route_id ซ้ำจะแจ้งเตือน error และให้กรอกค่าใหม่

```
$sql_2 = "INSERT INTO origin
VALUES ('$route_origin');
";

$objQuery_2 = mysqli_query($conn, $sql_2);

if ($objQuery_2) {
    echo "New origin";
    //header('location: admin_edit_route_add.php');
} else {
    echo "Error : Input";
}
```

ภาพที่ 6.1.2.7 Insert

- 8) มีการนำค่าที่เก็บไว้ในตัวแปร \$route_origin ไป insert ข้อมูลในตาราง origin ในดาต้าเบส

```

$sql_3 = "INSERT INTO dnation
VALUES ('$route_destiantion');
";

$objQuery_3 = mysqli_query($conn, $sql_3);

if ($objQuery_3) {
    echo "New destination";
    echo "$route_destiantion";
    header('location: route_add.php');
} else {
    echo "Error : Input";
}

```

ภาพที่ 6.1.2.8 Insert

- 9) มีการนำค่าที่เก็บไว้ในตัวแปร \$route_destination ไป insert ข้อมูลในตาราง dnation ในดาต้าเบส

เพิ่มเส้นทางเดินรถ

หมายเลขเส้นทาง

ต้นทาง

ปลายทาง

ภาพ การเพิ่มเส้นทางเดินรถ

6.1.3) การ Insert ข้อมูลในตาราง carriage

```

<?php
session_start();
include('server.php');
$car_id = $_REQUEST['car_id'];
$car_name = $_REQUEST['car_name'];
$car_class = $_REQUEST['car_class'];
$car_type = $_REQUEST['car_type'];
$car_price = $_REQUEST['car_price'];
$train_id = $_REQUEST['train_id'];

if(empty($car_id) or empty($car_name) or empty($car_class) or empty($car_type) or empty($car_price) or empty($train_id)){
    $_SESSION['alert'] = 'error';
    $_SESSION['alert_status'] = 'พบข้อผิดพลาด';
    header('location: carriage_add.php');
}

```

- 1) มีการประกาศเรียกใช้ function session start เพื่อเรียกใช้ตัวแปร \$_SESSION
- 2) มีการเชื่อมดาต้าเบสกับ server (server.php)
- 3) มีการรับค่ามาจากฟอร์มใน html ผ่านคำสั่ง \$_REQUEST
- 4) มีการตั้งค่าว่างถ้า user ไม่ได้กรอกค่าลงในฟอร์ม จะทำการรีเฟรชหน้าเว็บไซต์

ภาพที่ 6.1.3.1 Insert

```

else{
    $sql = "INSERT INTO carriage
VALUES ('$car_id',' $car_name',' $car_class','$car_type','$car_price','$train_id');
";
    $objQuery = mysqli_query($conn, $sql);

    if ($objQuery) {
        echo "New record Inserted successfully";
        $_SESSION['alert'] = 'error';
        $_SESSION['alert_status'] = 'เพิ่มข้อมูลสำเร็จ';
        header('location: carriage_add.php');
    } else {
        echo "Error : Input";
        $_SESSION['alert'] = 'error';
        $_SESSION['alert_status'] = 'ให้หมายเลขนี้ไปแล้ว';
        header('location: carriage_add.php');
    }
}

```

ภาพที่ 6.1.3.4 Insert

- 5) มีการนำข้อมูลที่เก็บไว้ในตัวแปร \$car_id , \$car_name , \$car_class , \$car_type , \$car_price , \$train_id ไป Insert ข้อมูลในตาราง carriage ในดาต้าเบส
- 6) มีการเรียกใช้ฟังก์ชัน mysqli_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ Insert ข้อมูลเข้าไป
- 7) และถ้าหากมีการใส่ข้อมูลซ้ำจะ error และระบบจะให้ทำการกรอกใหม่อีกครั้ง

เพิ่มข้อมูลตู้ขบวนรถไฟ

หมายเลขขบวนตู้โดยสาร

ชื่อขบวนตู้โดยสาร

ชั้น

ประเภท

ราคา

หมายเลขขบวนรถไฟ

ภาพ การเพิ่มข้อมูลขบวน

6.2) โค้ดการ delete ข้อมูล

6.2.1) การ delete ข้อมูลในตาราง train

```

website > a_delete_train.php
1  <?php
2      session_start();
3      include('server.php');
4      $delete_ID = $_REQUEST['train_id'];
5
6      $sql = '
7          DELETE FROM train
8          WHERE train_id = ' . $delete_ID . ' ;
9      ' ;
10
11     $objQuery = mysqli_query($conn, $sql);
12     if ($objQuery) {
13         echo "Record " . $delete_ID . " was Deleted.";
14         header('location: delete.php');
15     } else {
16         echo "Error : Delete";
17     }
18     mysqli_close($conn);
19     ?>

```

ภาพที่ 6.2.1 delete

- 1) มีการประกาศเรียกใช้ function session start เพื่อเรียกใช้ตัวแปร \$_SESSION
- 2) มีการเชื่อมดาต้าเบสกับ server (server.php)
- 3) มีการรับค่ามาจากฟอร์มใน html ผ่านคำสั่ง \$_REQUEST ใส่ในตัวแปร \$delete_ID
- 4) มีการเรียกใช้ฟังก์ชัน mysqli_query ให้มีการติดต่อไปพื้นฐานข้อมูลเพื่อทำการ delete ข้อมูล
- 5) มีการลบค่าโดยอ้างอิงข้อมูลมาจาก train_id
- 6) มีการเรียกใช้ฟังก์ชัน mysqli_query ให้มีการติดต่อไปพื้นฐานข้อมูลเพื่อทำการ Delete ข้อมูล
- 7) มีการใช้ฟังก์ชัน mysqli_close เพื่อยุติการเชื่อมต่อกับฐานข้อมูล
- 8) ถ้าการลบไม่สำเร็จจะขึ้นแจ้งเตือนแอตมินว่า “Error : Delete”

หน้าต่างลบข้อมูลขบวนรถไฟ

หมายเลขขบวน	ประเภท	เวลาออก	รหัสเส้นทาง	
123	ด่วนพิเศษ CNR	04:28 น.	123	<input type="button" value="เลือกลบ"/>
12456	ด่วน	10:58 น.	1	<input type="button" value="เลือกลบ"/>
30002	ด่วน	06:00 น.	1	<input type="button" value="เลือกลบ"/>
30003	ด่วนพิเศษ	20:00 น.	1	<input type="button" value="เลือกลบ"/>
30004	ด่วนพิเศษดีเซล	10:30 น.	3	<input type="button" value="เลือกลบ"/>
30005	ด่วนพิเศษ CNR	20:10 น.	3	<input type="button" value="เลือกลบ"/>
30006	ด่วนพิเศษ CNR	09:30 น.	5	<input type="button" value="เลือกลบ"/>
30007	เร็ว	22:40 น.	5	<input type="button" value="เลือกลบ"/>
30008	เร็ว	12:00 น.	6	<input type="button" value="เลือกลบ"/>
30009	ด่วนพิเศษ CNR	23:00 น.	6	<input type="button" value="เลือกลบ"/>
30010	ด่วน	08:00 น.	4	<input type="button" value="เลือกลบ"/>
30011	เร็ว	14:00 น.	4	<input type="button" value="เลือกลบ"/>
30012	ด่วนพิเศษดีเซลราง	20:00 น.	4	<input type="button" value="เลือกลบ"/>
30013	ด่วนพิเศษดีเซลราง	23:00 น.	7	<input type="button" value="เลือกลบ"/>
30014	ด่วน	05:30 น.	7	<input type="button" value="เลือกลบ"/>
30015	เร็ว	13:00 น.	7	<input type="button" value="เลือกลบ"/>
30016	ด่วนพิเศษดีเซลราง	18:00 น.	8	<input type="button" value="เลือกลบ"/>
30017	เร็ว	11:50 น.	8	<input type="button" value="เลือกลบ"/>
30018	ด่วนพิเศษ CNR	20:00 น.	8	<input type="button" value="เลือกลบ"/>
30019	ด่วน	21:00 น.	9	<input type="button" value="เลือกลบ"/>
30020	ด่วนพิเศษ CNR	15:00 น.	9	<input type="button" value="เลือกลบ"/>
30021	เร็ว	07:30 น.	9	<input type="button" value="เลือกลบ"/>
30022	ด่วน	03:00 น.	10	<input type="button" value="เลือกลบ"/>
30023	เร็ว	12:00 น.	10	<input type="button" value="เลือกลบ"/>

ภาพ หน้าต่างลบข้อมูลขบวนรถไฟ

- 1) กดปุ่ม **เลือกลบ** เพื่อลบ row ที่ต้องการ
- 2) หลังจากลบ row ที่เลือกจะหายไปจากหน้าเว็บไซต์

6.2.2) การ delete ข้อมูลในตาราง route

```
<?php
    session_start();
    include('server.php');
    $delete_ID = $_REQUEST['route_id'];

    echo $_REQUEST['route_id'];

    $sql = '
        DELETE FROM route
        WHERE route_id = ' . $delete_ID . '
    ';

    $objQuery = mysqli_query($conn, $sql);
    if ($objQuery) {
        echo "Record " . $delete_ID . " was Deleted.";
        header('location: route_delete.php');
    } else {
        echo "Error : Delete";
    }
    mysqli_close($conn);
?>
```

ภาพที่ 6.2.2 delete

- 1) มีการประกาศเรียกใช้ function session start เพื่อเรียกใช้ตัวแปร \$_SESSION
- 2) มีการเชื่อมดาต้าเบสกับ server (server.php)
- 3) มีการรับค่ามาจากฟอร์มใน html ผ่านคำสั่ง \$_REQUEST ใส่ในตัวแปร \$delete_ID
- 4) มีการเรียกใช้ฟังก์ชัน mysqli_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ delete ข้อมูล
- 5) มีการลบค่าโดยอ้างอิงข้อมูลมาจาก route_id
- 6) มีการเรียกใช้ฟังก์ชัน mysqli_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ Delete ข้อมูล
- 7) มีการใช้ฟังก์ชัน mysqli_close เพื่อยุติการเชื่อมต่อกับฐานข้อมูล
- 8) ถ้าการลบไม่สำเร็จจะขึ้นแจ้งเตือนแอดมินว่า “Error : Delete”

หน้าต่างลบเส้นทางเดินรถ

หมายเลขเส้นทาง	ต้นทาง	ปลายทาง	
2	กรุงเทพ	เชียงใหม่	เลือกลบ
3	กรุงเทพ	อุบลราชธานี	เลือกลบ
4	อุบลราชธานี	กรุงเทพ	เลือกลบ
5	ชุมทางหาดใหญ่	กรุงเทพ	เลือกลบ
6	กรุงเทพ	ชุมทางหาดใหญ่	เลือกลบ
7	สุราษฎร์ธานี	กรุงเทพ	เลือกลบ
8	กรุงเทพ	สุราษฎร์ธานี	เลือกลบ
9	กรุงเทพ	หนองคาย	เลือกลบ
10	หนองคาย	กรุงเทพ	เลือกลบ

ภาพ หน้าต่างลบเส้นทางเดิน

- 1) กดปุ่ม **เลือกลบ** เพื่อลบ row ที่ต้องการ
- 2) หลังจากลบ row ที่เลือกจะหายไปจากหน้าเว็บไซต์

6.2.3) การ delete ข้อมูลในตาราง carriage

```
<?php
    session_start();
    include('server.php');

    $delete_ID = $_REQUEST['carriage_id'];

    echo $_REQUEST['carriage_id'];
    echo "<br>";

    $sql = '
        DELETE FROM carriage
        WHERE car_id = ' . $delete_ID . '
    ';

    $objQuery = mysqli_query($conn, $sql);
    if ($objQuery) {
        echo "Record " . $delete_ID . " was Deleted.";
    } else {
        echo "Error : Delete";
    }

    mysqli_close($conn); // ปิดฐานข้อมูล
    echo "<br><br>";
?>
```

ภาพที่ 6.2.3 delete

- 1) มีการประกาศเรียกใช้ function session start เพื่อเรียกใช้ตัวแปร \$_SESSION
- 2) มีการเชื่อมต่อด้านเบสกับ server (server.php)
- 3) มีการรับค่ามาจากฟอร์มใน html ผ่านคำสั่ง \$_REQUEST ใส่ในตัวแปร \$delete_ID
- 4) มีการเรียกใช้ฟังก์ชัน mysqli_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ delete ข้อมูล
- 5) มีการลบค่าโดยอ้างอิงข้อมูลมาจาก car_id
- 6) มีการเรียกใช้ฟังก์ชัน mysqli_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ Delete ข้อมูล
- 7) มีการใช้ฟังก์ชัน mysqli_close เพื่อยุติการเชื่อมต่อกับฐานข้อมูล
- 8) ถ้าการลบไม่สำเร็จจะขึ้นแจ้งเตือนแอดมินว่า “Error : Delete”

หน้าต่างลบรายการตู้รถไฟ

หมายเลขขบวนตู้โดยสาร	ชื่อขบวนตู้โดยสาร	ชั้น	ประเภท	ราคา	หมายเลขขบวนรถไฟ	
40009	บนส.54	ชั้น1	เครื่องปรับอากาศ	200.00	30004	เลือกลบ
40010	บนส.54	ชั้น1	เครื่องปรับอากาศ	200.00	30005	เลือกลบ
40011	บนส.54	ชั้น1	เครื่องปรับอากาศ	200.00	30006	เลือกลบ
40012	บนข.11	ชั้น2	พัดลม	300.00	30007	เลือกลบ
40013	บนข.11	ชั้น2	พัดลม	300.00	30008	เลือกลบ
40014	บนส.54	ชั้น1	เครื่องปรับอากาศ	200.00	30009	เลือกลบ
40015	บนท.99	ชั้น2	เครื่องปรับอากาศ	230.00	30010	เลือกลบ
40016	บนส.76	ชั้น3	พัดลม	231.00	30011	เลือกลบ
40017	บนท.99	ชั้น2	เครื่องปรับอากาศ	230.00	30012	เลือกลบ
40018	บนท.99	ชั้น2	เครื่องปรับอากาศ	230.00	30013	เลือกลบ
40019	บนท.93	ชั้น2	เครื่องปรับอากาศ	800.00	30014	เลือกลบ
40020	บนข.11	ชั้น2	พัดลม	300.00	30015	เลือกลบ
40021	บนส.54	ชั้น1	เครื่องปรับอากาศ	200.00	30016	เลือกลบ
40022	บนข.11	ชั้น2	พัดลม	300.00	30016	เลือกลบ
40023	บนส.12	ชั้น1	เครื่องปรับอากาศ	120.00	30018	เลือกลบ
40024	บนท.93	ชั้น2	เครื่องปรับอากาศ	800.00	30019	เลือกลบ
40025	บนส.12	ชั้น1	เครื่องปรับอากาศ	200.00	30020	เลือกลบ
40026	บนท.93	ชั้น2	เครื่องปรับอากาศ	800.00	30020	เลือกลบ
40027	บนส.76	ชั้น3	พัดลม	231.00	30021	เลือกลบ
40028	บนข.11	ชั้น2	พัดลม	300.00	30022	เลือกลบ
40029	บนส.76	ชั้น3	พัดลม	231.00	30023	เลือกลบ

ภาพ หน้าต่างลบรายการตู้รถไฟ

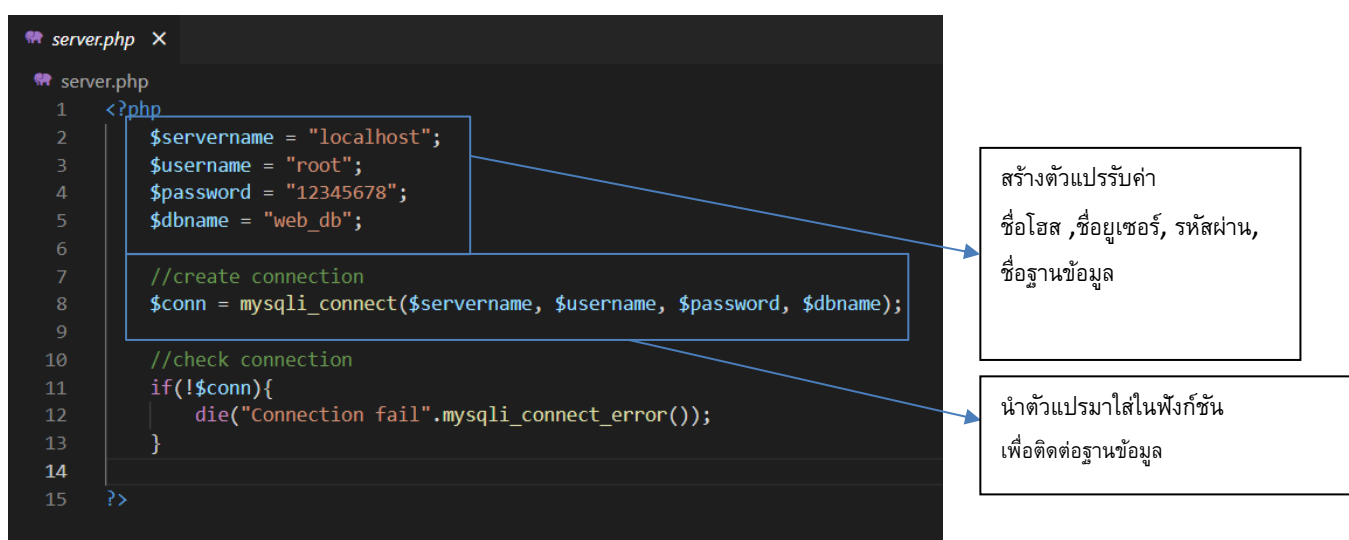
- 1) กดปุ่ม **เลือกลบ** เพื่อลบ row ที่ต้องการ
- 2) หลังจากลบ row ที่เลือกจะหายไปจากหน้าเว็บไซต์

บทที่ 7

การติดต่อฐานข้อมูล

7.1) การติดต่อฐานข้อมูล ในภาษาPHP

คำสั่ง `MYSQL_CONNECT()` ใช้ในการติดต่อฐานข้อมูล



ภาพที่ 7.1 การติดต่อฐานข้อมูล

7.2) การแสดงข้อมูลจากฐานข้อมูลบนหน้าเว็บไซต์

สามารถแสดงข้อมูลจากฐานข้อมูลโดยใช้คำสั่ง `SELECT`

ขั้นตอนที่1

เรียกฟังก์ชันที่ใช้ติดต่อ SERVER

`SESSION_START();`

ใช้สำหรับประกาศตัวแปร `$_SESSION`

`Include()`

ใช้ในการรวมไฟล์ โดยเราจะนำไฟล์ที่ใช้ติดต่อฐานข้อมูลมารวมด้วย

```

<?php
    session_start();
    include('server.php');
?>
  
```

ภาพที่ 7.2.1 การแสดงข้อมูลจากฐานข้อมูลบนหน้าเว็บไซต์

ขั้นตอนที่ 2

ใช้คำสั่ง **SELECT** ระบุชื่อคอลัมป์ และ ระบุชื่อตาราง ที่เราต้องการแสดงผลในหน้าเว็บ
จากนั้นสร้างตัวแปรมารับค่า

MYSQLI_QU () ใช้ในการ QUERY ข้อมูลจาก DB

```
<?php $sql = '
SELECT ticket_id,train_id,car_name,route_origin,route_destination,seat_num,username,
user_name,user_surname,train_type,car_class,car_type,car_price_stop,car_id,seat_date
FROM ticket ';
```

```
$objQuery = mysqli_query($conn, $sql) or die("Error Query [" . $sql . "]);
```

ภาพที่ 7.2.2 การแสดงข้อมูลจากฐานข้อมูลบนหน้าเว็บไซต์

ขั้นตอนที่ 3

สร้าง TABLE ใช้สำหรับแสดงข้อมูล

< TR > ส่วนของ ROW , < TH > แสดงหัวตาราง , < TD > ข้อมูลในตาราง

MYSQLI_FETCH_ARRAY()

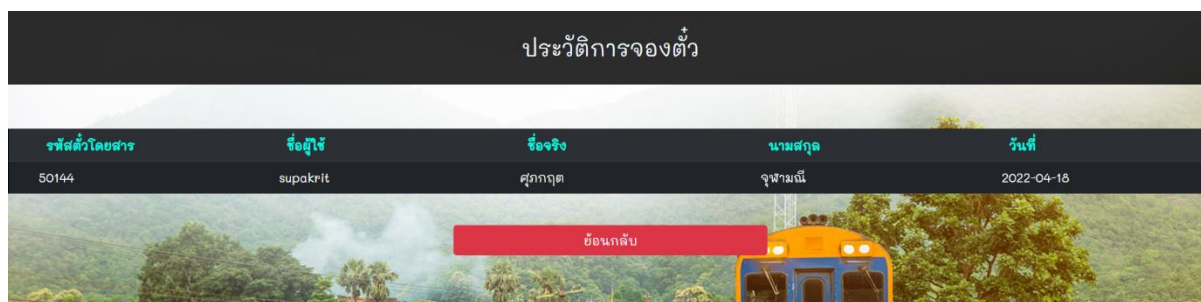
เรียกแถวผลลัพธ์เป็น ARRAY

```
<table border="1" class="table table-dark table-striped" >
<tr>
<th>
<div class="head_table">รหัสตัวโดยสาร</div>
</th>
<th>
<div class="head_table">ชื่อผู้โดยสาร</div>
</th>
<th>
<div class="head_table">ชื่อจริง</div>
</th>
<th>
<div class="head_table">นามสกุล</div>
</th>
<th>
<div class="head_table">วันที่</div>
</th>
</tr>
```

```
while ($objResult = mysqli_fetch_array($objQuery)) {
?>
<tr>
<td>
<div class="result">
<?php echo $objResult["ticket_id"]; ?>
</div>
</td>
<td>
<div class="result">
<?php echo $objResult["username"]; ?>
</div>
</td>
<td>
<div class="result">
<?php echo $objResult["user_name"]; ?>
</div>
</td>
<td>
<div class="result">
<?php echo $objResult["user_surname"]; ?>
</div>
</td>
<td>
<div class="result">
<?php echo $objResult["seat_date"]; ?>
</div>
</td>
</tr>
<?php
```

ภาพที่ 7.2.3 การแสดงข้อมูลจากฐานข้อมูลบนหน้าเว็บไซต์

จะได้ผลลัพธ์ดังนี้



ภาพที่ 7.2.4 การติดต่อฐานข้อมูล

7.3) การแก้ไขข้อมูลจากฐานข้อมูล

- a) สามารถแก้ไขข้อมูลจากฐานข้อมูลโดยใช้คำสั่ง UPDATE

```

admin_route_updatedata.php
1  <?php
2      session_start();
3      include('server.php');
4
5
6  $route_id    = $_SESSION['route_id'];
7  $route_origin = $_REQUEST['route_origin'];
8  $route_destination = $_REQUEST['route_destination'];
9
10 echo $_REQUEST['route_id'];
11 echo $_REQUEST['route_origin'];
12 echo $_REQUEST['route_destination'];
13
14 if(empty($route_id) or empty($route_origin) or empty($route_destination)){
15     header('location:admin_edit_route_update2.php');
16 }
17 else{
18     $sql = " UPDATE route
19             SET route_origin = '$route_origin',
20                 route_destination = '$route_destination'
21             WHERE route_id = '$route_id' ";
22
23     $objQuery = mysqli_query($conn, $sql);
24
25     if ($objQuery) {
26         echo "Updated.";
27         header('location: admin_edit_route_update.php');
28     } else {
29         echo "Error : Update";
30     }
31     unset($_SESSION['route_id']);
32 }
33
34 mysqli_close($conn); // ปิดฐานข้อมูล
35 echo "<br><br>";
36 ?>
    
```

ข้อมูลที่ต้องการนำไปแก้ไข

ตรวจสอบว่าข้อมูลมีค่าว่าง

ภาพที่ 7.3.1 การแก้ไขข้อมูลจากฐานข้อมูล

คำสั่ง UPDATE

UPDATE __ตารางแก้ไข__, SET คอลัมป์ = คอลัมป์ใหม่, WHERE = ตำแหน่งที่แก้ไข

7.4 relational algebra

SQL Operator

SQL Operator การทำงานที่เลือกข้อมูลในเงื่อนไข โดยระบบฐานข้อมูล จะมีการจอง key word บางคำ หรือ ตัวอักษรไว้สำหรับให้ SQL statement ใช้ระบุเงื่อนไข เพื่อเป็นการสั่งระบบให้ทำตามเงื่อนไขที่วางไว้ เช่น การเปรียบเทียบ การคำนวณทางคณิตศาสตร์ การทำกระบวนการเหล่านี้ มักจะใช้ระบุเงื่อนไขภายใต้ SQL statement และ สามารถกำหนดได้หลายเงื่อนไขภายใน statement นั้นๆ

ตารางที่จะนำไปเป็นตัวอย่างการ operator

Table : train

	train_id	train_name	train_type
<input type="checkbox"/> Edit Copy Delete	1	ขบวนที่1	ด่วน
<input type="checkbox"/> Edit Copy Delete	2	ขบวนที่2	เร็ว
<input type="checkbox"/> Edit Copy Delete	3	ขบวนที่3	ด่วน พิเศษ
<input type="checkbox"/> Edit Copy Delete	4	ขบวนที่4	ด่วน
<input type="checkbox"/> Edit Copy Delete	5	ขบวนที่5	ด่วน

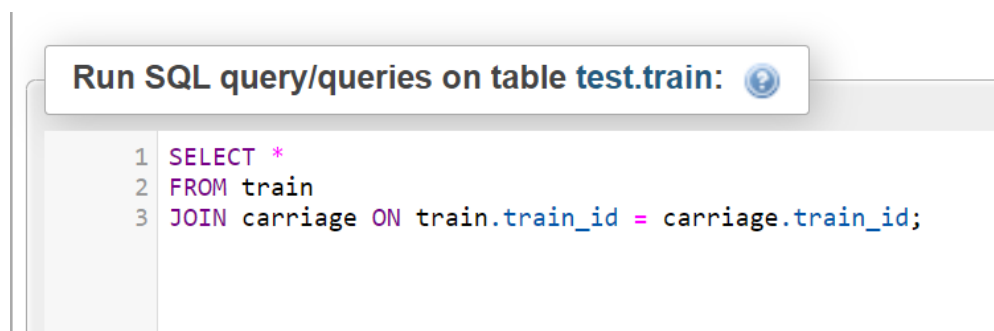
ภาพที่ 7.4.1 relational algebra

Table = carriage

	car_id	car_name	car_calss	train_id
<input type="checkbox"/> Edit Copy Delete	1	บขส.1001	ชั้น3	4
<input type="checkbox"/> Edit Copy Delete	2	บขส.1002	ชั้น1	1
<input type="checkbox"/> Edit Copy Delete	3	บขส.1003	ชั้น2	2
<input type="checkbox"/> Edit Copy Delete	4	บขส.1004	ชั้น1	5
<input type="checkbox"/> Edit Copy Delete	5	บขส.1005	ชั้น3	3

ภาพที่ 7.4.2 relational algebra

7.4.1) การ JOIN แบบบปกติ

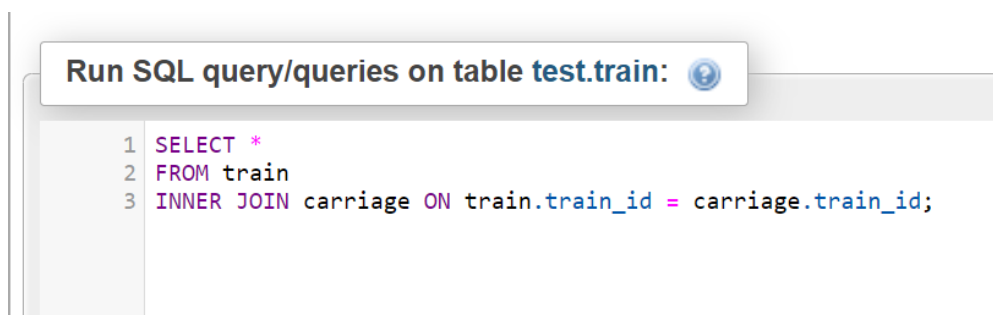


ภาพที่ 7.4.1.1 JOIN

train_id	train_name	train_type	car_id	car_name	car_class	train_id
1	ขบวนที่1	ด่วน	2	บขส.1002	ชั้น1	1
2	ขบวนที่2	เร็ว	3	บขส.1003	ชั้น2	2
3	ขบวนที่3	ด่วน พิเศษ	5	บขส.1005	ชั้น3	3
4	ขบวนที่4	ด่วน	1	บขส.1001	ชั้น3	4
5	ขบวนที่5	ด่วน	4	บขส.1004	ชั้น1	5

ภาพที่ 7.4.1.2 JOIN

7.4.2) INNER JOIN แสดงผลข้อมูลที่มีค่าทั้ง 2 table เหมือนกันใน column ที่ระบุ

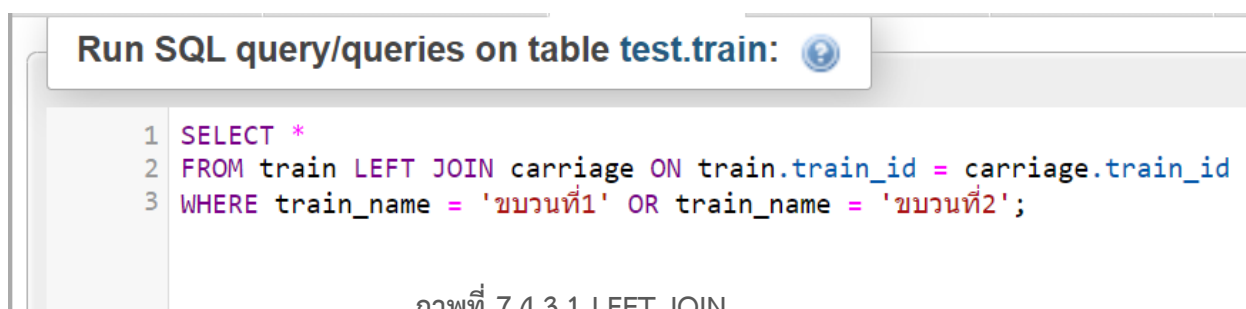


ภาพที่ 7.4.2.1 INNER JOIN

train_id	train_name	train_type	car_id	car_name	car_class	train_id
1	ขบวนที่1	ด่วน	2	บขส.1002	ชั้น1	1
2	ขบวนที่2	เร็ว	3	บขส.1003	ชั้น2	2
3	ขบวนที่3	ด่วน พิเศษ	5	บขส.1005	ชั้น3	3
4	ขบวนที่4	ด่วน	1	บขส.1001	ชั้น3	4
5	ขบวนที่5	ด่วน	4	บขส.1004	ชั้น1	5

ภาพที่ 7.4.2.2 INNER JOIN

7.4.3) LEFT JOIN แสดงผลข้อมูลของ table ทางซ้ายทั้งหมด (table 1) และ ข้อมูลของ table ทางขวาที่ตรงเงื่อนไข (table 2)

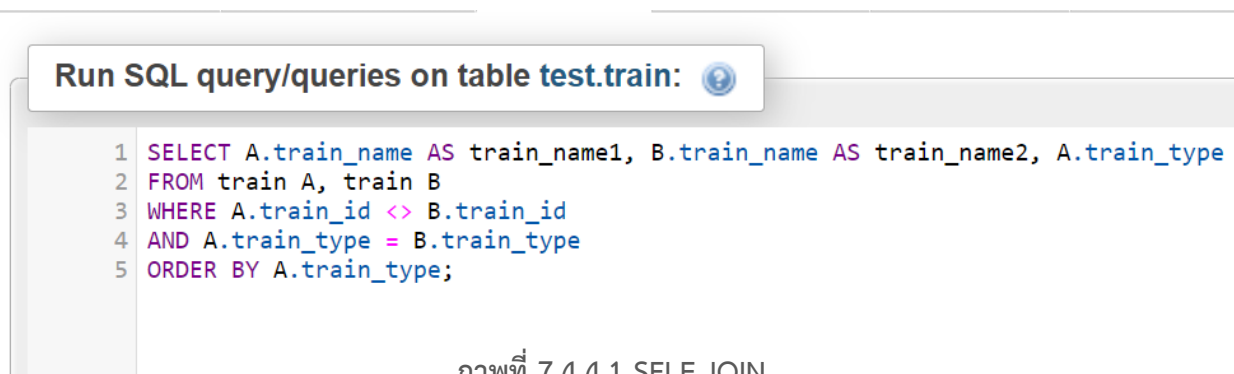


ภาพที่ 7.4.3.1 LEFT JOIN

train_id	train_name	train_type	car_id	car_name	car_class	train_id
1	ขบวนที่1	ด่วน	2	บขส.1002	ชั้น1	1
2	ขบวนที่2	เร็ว	3	บขส.1003	ชั้น2	2

ภาพที่ 7.4.3.2 LEFT JOIN

7.4.4) SELF JOIN ทำการเชื่อมความสัมพันธ์ของ table ตัวเอง โดยเปรียบเสมือนสร้าง table ตัวเองเป็นอีก table หนึ่งมา JOIN กันเอง

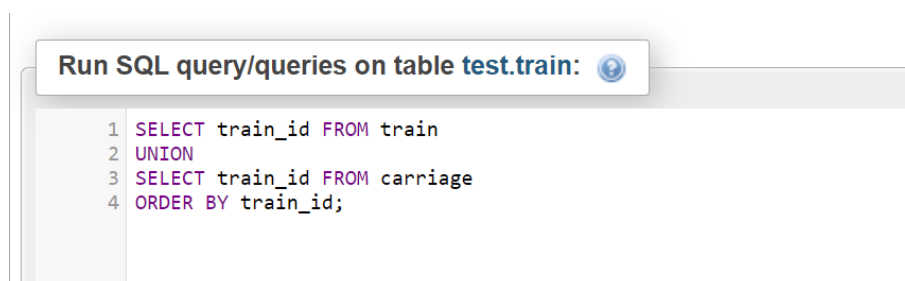


ภาพที่ 7.4.4.1 SELF JOIN

train_name1	train_name2	train_type	1
ขบวนที่4	ขบวนที่5	ด่วน	
ขบวนที่4	ขบวนที่1	ด่วน	
ขบวนที่5	ขบวนที่1	ด่วน	
ขบวนที่1	ขบวนที่4	ด่วน	
ขบวนที่5	ขบวนที่4	ด่วน	
ขบวนที่1	ขบวนที่5	ด่วน	

ภาพที่ 7.4.4.2 SELF JOIN

7.4.5) UNION เราสามารถนำข้อมูลมารวมกันในเชิงบรรทัด ด้วยคำสั่ง SQL UNION statement



ภาพที่ 7.4.5.1 UNION

train_id	1
	1
	2
	3
	4
	5

ภาพที่ 7.4.5.2 UNION