

**รายงาน**

ระบบการจองตั๋วการรถไฟแห่งประเทศไทย

**เสนอ**

อาจารย์ เอิญ สุริยะฉาย

**จัดทำโดย**

1. นาย ปรเมษฐ์ 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 ) การสร้างตารางในPhpMyAmin |  | 15 |
| 5.2 ) การแก้ไขข้อมูลตารางใน PhpMyAmin |  | 16 |
| 5.3 ) วิธีการ Create Tableโดย SQL |  | 16 |
| **บทที่ 6 เพิ่มขบวนรถไฟ** |  | 17-27 |
| 6.1 ) โค้ดการ Insertข้อมูล |  | 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 ) การแก้ข้อมูลจากฐานข้อมูล |  | 30 |
| 7.4 ) relational algebra |  | 31-34 |

**บทที่ 1**

**บทนำ**

* 1. **คำอธิบายระบบ**

**ระบบการจองตั๋วการรถไฟแห่งประเทศไทย** **(Railway** **Reservation** **System)** เป็นระบบทำงานใน ลักษณะของ **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 ที่สแกนผ่านแอพมือถือเพื่อจ่ายเงินและอัพโหลดรูปภาพการชำระเงินเพื่อให้แอดมิน

ตรวจสอบ หลังจากนั้นจึงจะสามารถปริ้นตั๋วรถไฟได้

* **ส่วนแอดมิน** แอดมินสามารถแก้ไข เพิ่ม ลบ เส้นทางเดินรถ ขบวนรถไฟ ตู้รถไฟ และสามารถเข้าถึงรายการจองตั๋วและรายการชำระเงินทั้งหมดของทุกยูเซอร์ รวมไปถึงการตรวจสอบรายการขบวนรถไฟที่วิ่งอยู่ทั้งหมดได้

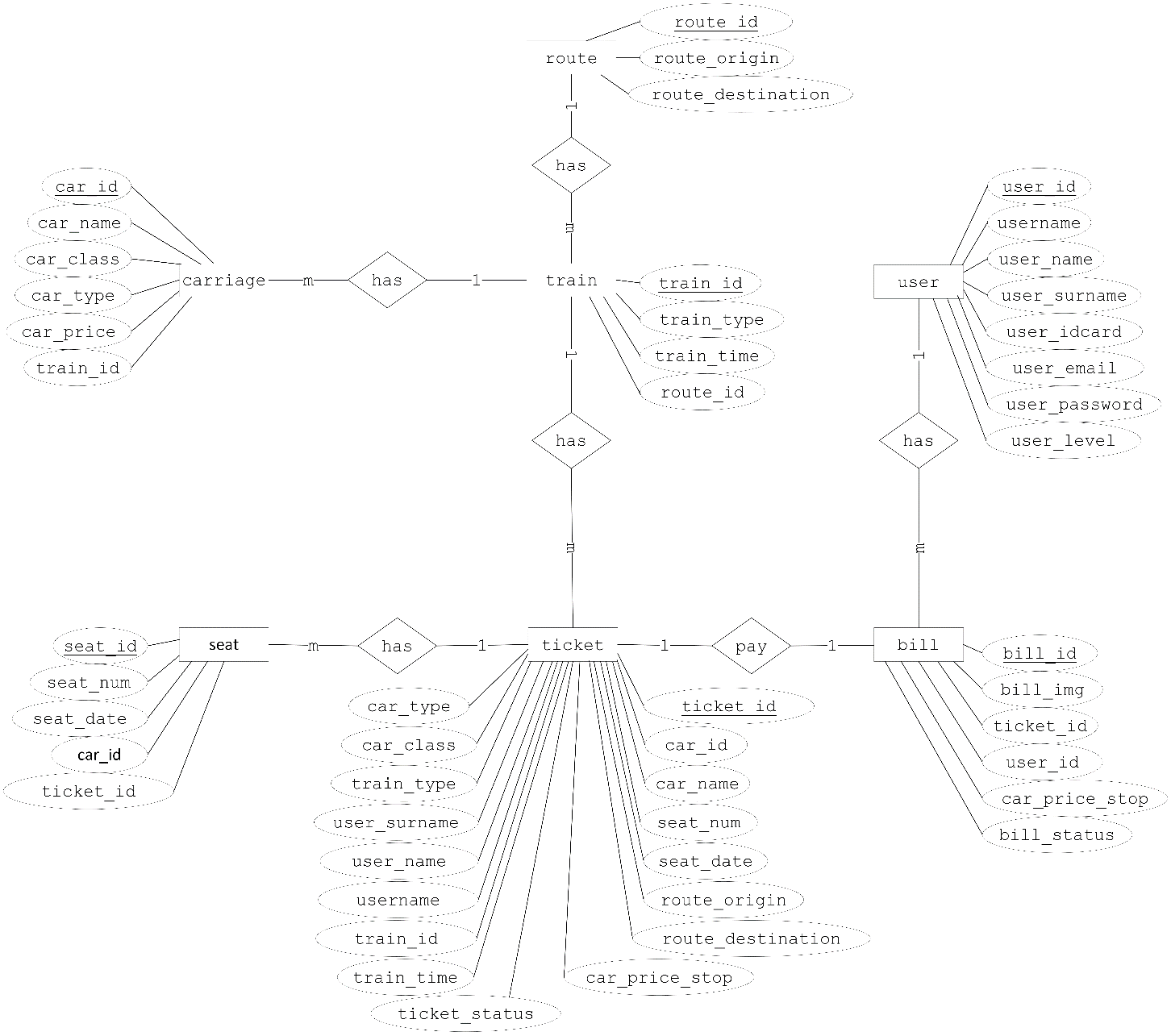
**1.2 ฟังก์ชันการทำงานภายในระบบ**

1. **การสมัครสมาชิก(Register) และเข้าสู้ระบบ(Login) :**
   1. ยูสเซอร์สามารถเปลี่ยนชื่อจริง นามสกุล อีเมลได้
   2. ในขั้นตอนการสมัครสมาชิกยูสเซอร์\*\*ต้องกรอกยูสเซอร์เนม (Username) ชื่อจริง นามสกุล เลขบัตรประจำตัวประชาชน อีเมล (Email) และพาสเวิร์ด (Password)
2. **การจองตั๋วรถไฟ (Railway Reservation) :**
   1. เลือกสถานีต้นทาง และปลายทาง รวมถึงวันที่ต้องการจะไป
   2. เลือกขบวนรถไฟและเลือกตู้รถไฟ
   3. เลือกที่นั่ง
3. **การชำระเงิน** 
   1. ยูสเซอร์สามารถชำระเงินทันทีหรือชำระเงินภายหลังได้
   2. ยูสเซอร์สามารถชำระเงินทันทีโดยการ Scan QR Code ได้เลย
4. **ข้อมูลส่วนตัว (Personal Information) :**
   1. ยูสเซอร์สามารถแก้ไขข้อมูลส่วนตัวด้วยตัวเองได้ตลอดเวลาที่แก้ไขข้อมูลผู้โดยสาร
5. **ยกเลิกการจองตั๋วรถไฟ (Cancel Reservation) :**
   1. ยูสเซอร์สามารถยกเลิกการจองตั๋วรถไฟได้

***หมายเหตุ หากชำระเงินแล้วสามารถยกเลิกการจองตั๋วได้แต่จะไม่ได้รับเงินคืน***

1. **แก้ไข ลบ เพิ่ม และตรวจสอบรายการขบวนรถไฟ:**
   1. Admin สามารถแก้ไข เพิ่ม ลบ เส้นทางเดินรถ ขบวนรถไฟ ตู้รถไฟ และสามารถเข้าถึงรายการจองตั๋วได้ รวมไปถึงการตรวจสอบรายการขบวนรถไฟที่วิ่งอยู่ทั้งหมดได้
2. **ประวัติการจอง (Reservation History) :** 
   1. **ส่วนยูสเซอร์** 
      1. แอดมินสามารถตรวจสอบ รหัสตั๋วโดยสาร ชื่อผู้ใช้ ชื่อจริง นามสกุล วันที่ของยูสเซอร์ได้
      2. แอดมินสามารถตรวจสอบหลักฐานการชำระเงินของยูสเซอร์ เพื่อยืนยันการออกตั๋วรถไฟให้ยูสเซอร์ได้ ซึ่งจะระบุหมายเลขบิล หมายเลชผู้ใช้ หมายเลขการจอง ยอดต้องชำระ และหลักฐานการชำระเงิน
   2. **ส่วนยูสเซอร์** 
      1. ยูสเซอร์สามารถเข้ามาปริ้นตั๋วรถไฟได้ที่ **ประวัติการจองตั๋วรถไฟ**
      2. กรณีที่ไม่ได้ชำระเงินในหน้าชำระเงินสามารถชำระเงินในหน้าต่างนี้ภายหลังได้ แล้วทางแอดมินจะตรวจสอบให้ จึงจะสามารถทำการปริ้นตั๋วรถไฟได้
3. **ออกจากระบบ (Logout)**
   1. ยูสเซอร์สามารถออกจากระบบเพื่อเปลี่ยนบัญชีผู้ใช้ได้

**บทที่ 2**

**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**

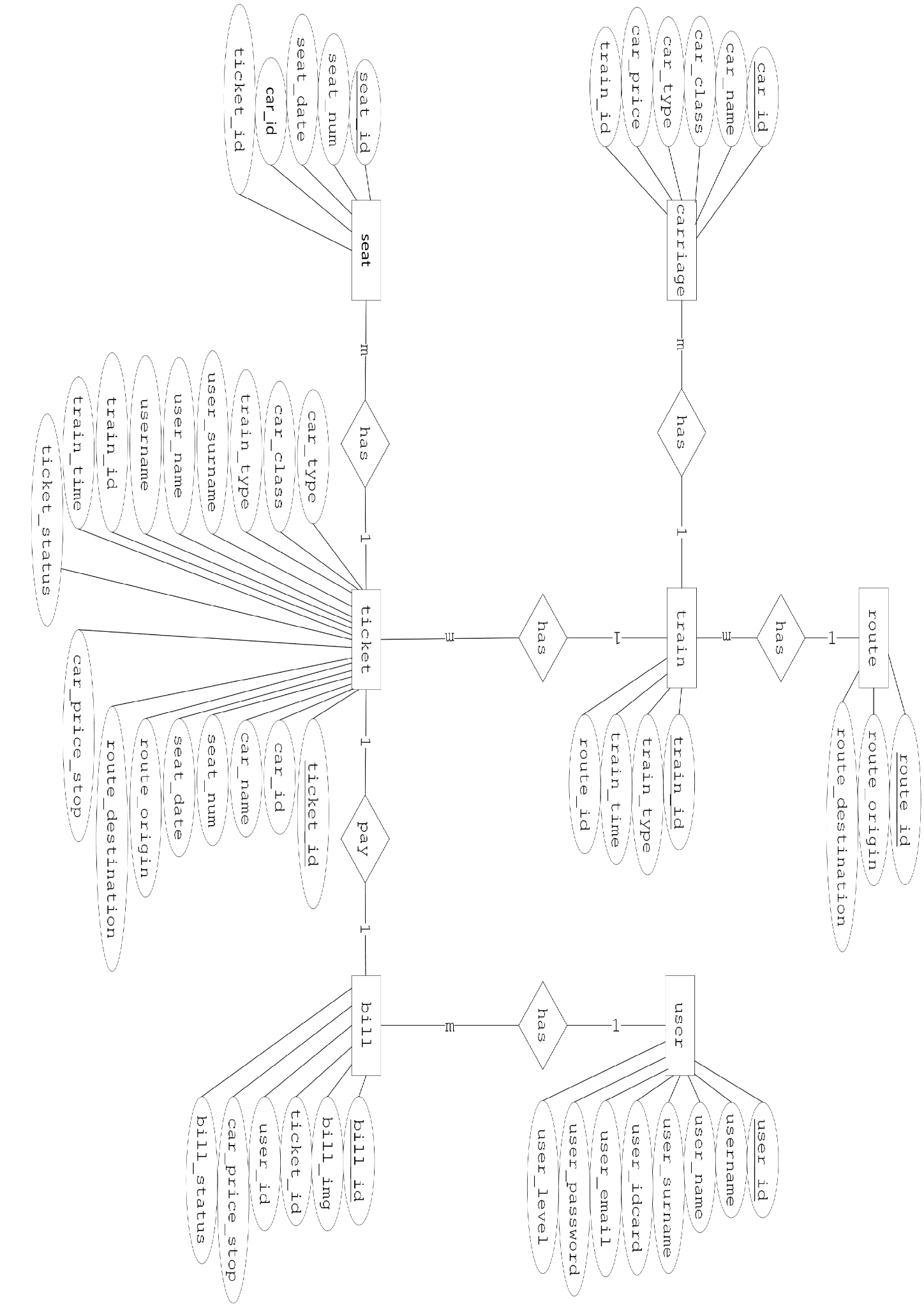
| **bill\_id** | **bill\_img** | **car\_price\_stop** | **bill\_status** | **ticket\_id** | **user\_id** |
| --- | --- | --- | --- | --- | --- |
| \*PK |  |  |  | \*FK | \*FK |

**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\_destinartion** | **car\_price\_stop** | **ticket\_status** | **trian\_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** | **Descrition** | **Type** | **Key Type** | **Reference Table** |
| --- | --- | --- | --- | --- |
| train\_id | รหัสรถไฟ | int | pk |  |
| train\_type | ชนิดรถไฟ | varchar(100) |  |  |
| train\_time | เวลารถไฟ | varchar(100) |  |  |
| route\_id | รหัสเส้นทาง | int | fk | Route |

**Table Name : Bill**

| **Attribute Name** | **Descrition** | **Type** | **Key Type** | **Reference Table** |
| --- | --- | --- | --- | --- |
| bill\_id | รหัสบิล | int | pk |  |
| bill\_img | ใบชำระเงิน | varbinary(100) |  |  |
| ticket\_id | รหัสตั๋ว | int | fk | Ticket |
| user\_id | รหัสผู้ใช้ | int |  |  |
| car\_price\_stop | ราคาสุทธิ | double |  |  |
| bill\_status | สถานะบิล | varchar(100) |  |  |

**Table Name : Carriage**

| **Attribute Name** | **Descrition** | **Type** | **Key Type** | **Reference** **Table** |
| --- | --- | --- | --- | --- |
| car\_id | รหัสตู้โดยสาร | int | pk |  |
| car\_name | ชื่อตู้โดยสาร | varchar(100) |  |  |
| car\_class | ระดับตู้โดยสาร | varchar(100) |  |  |
| car\_type | ประเภทตู้โดยสาร | varchar(100) |  |  |
| car\_price | ราคาตู้โดยสาร | double |  |  |
| train\_id | หมายเลขขบวน | int | fk | Train |

**Table Name : Route**

| **Attribute Name** | **Descrition** | **Type** | **Key Type** | **Reference Table** |
| --- | --- | --- | --- | --- |
| route\_id | รหัสเส้นทาง | int | pk |  |
| route\_origin | สถานีต้นทาง | varchar(100) |  |  |
| route\_destination | สถานีปลายทาง | varchar(100) |  |  |

**Table Name : Seat**

| **Attribute Name** | **Descrition** | **Type** | **Key Type** | **Reference Table** |
| --- | --- | --- | --- | --- |
| seat\_id | รหัสที่นั่ง | int | pk |  |
| seat\_num | เลขที่นั่ง | int |  |  |
| seat\_date | วันที่เดินทาง | varchar(100) |  |  |
| car\_id | รหัสตู้โดยสาร | int | fk | Carriage |
| ticket\_id | รหัสตั๋วโดยสาร | int | fk | Ticket |

**Table Name : Ticket**

| **Attribute Name** | **Descrition** | **Type** | **Key Type** | **Reference Table** |
| --- | --- | --- | --- | --- |
| ticket\_id | รหัสตั๋วโดยสาร | int | pk |  |
| car\_id | รหัสตู้โดยสาร | varchar  (1000) | fk | Carriage |
| car\_name | ชื่อตู้โดยสาร | varchar(100) |  |  |
| seat\_num | เลขที่นั่ง | varchar(100) |  |  |
| seat\_date | วันที่เดินทาง | varchar(100) |  |  |
| route\_origin | สถานีต้นทาง | varchar(100) |  |  |
| route\_destination | สถานีปลายทาง | varchar(100) |  |  |
| username | ชื่อผู้ใช้ | varchar(100) |  |  |
| user\_name | ชื่อ | varchar(100) |  |  |
| user\_surname | นามสกุล | varchar(100) |  |  |
| train\_type | ชนิดรถไฟ | varchar(100) |  |  |
| car\_class | ระดับตู้โดยสาร | varchar(100) |  |  |
| car\_type | ประเภทตู้โดยสาร | varchar(100) |  |  |
| car\_price\_status | สถานะการชำระเงิน | int |  |  |
| ticket\_status | สถานะการจองตั๋ว | varchar(100) |  |  |
| train\_time | เวลาเดินทาง | varchar(100) |  |  |
| train\_id | รหัสรถไฟ | int | fk | Train |

**Table Name : User**

| **Attribute Name** | **Descrition** | **Type** | **Key Type** | **Reference Table** |
| --- | --- | --- | --- | --- |
| user\_id | รหัสผู้ใช้ | int | pk |  |
| username | ชื่อผู้ใช้ | varchar(100) |  |  |
| user\_name | ชื่อ | varchar(100) |  |  |
| user\_surname | นามสกุล | varchar(100) |  |  |
| user\_idcard | เลขบัตรประจำตัวประชาชน | bigint |  |  |
| user\_email | อีเมลล์ผู้ใช้ | varchar(100) |  |  |
| user\_password | รหัสผ่าน | varchar(100) |  |  |
| user\_level | ระดับผู้ใช้ | varchar(100) |  |  |



**บทที่ 5**

**Create Table**





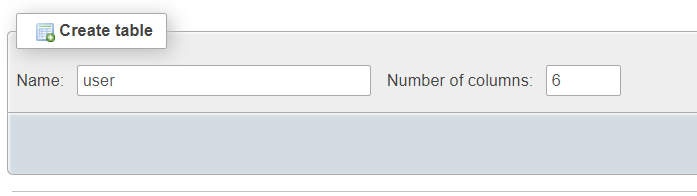






**5.1 ) การสร้างตารางใน PhpMyAmin**

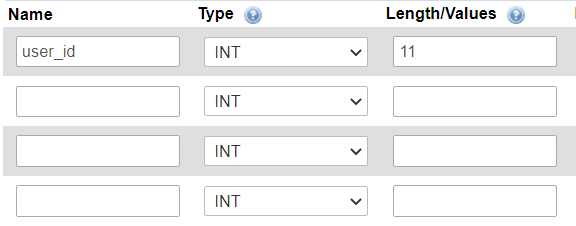
**ขั้นตอนที่ 1** ระบุชื่อ table และจำนวนคอลัมป์ สามารถเพิ่ม หรือ ลบ จำนวนคอลัมป์ได้ในภายหลัง





**ขั้นตอนที่ 2** ระบุชื่อคอลัมป์ ชนิดข้อมูล และขนาดของข้อมูล





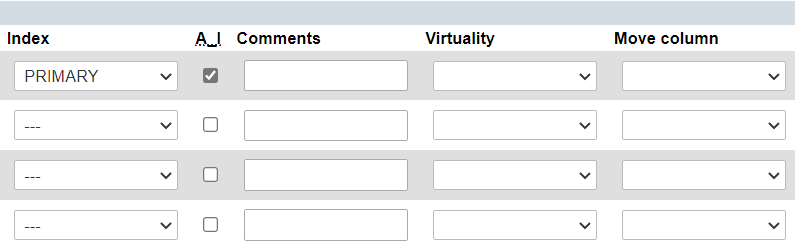


**ชั้นตอนที่ 3**

**Index -primary** เป็นการระบุคอลัมป์ เป็น **PRIMARY KEY**

**AI** เป็นการระบุให้คอลัมป์นั้นไม่ซ้ำกันและเพิ่มข้อมูลทันที โดยไม่ต้องเพิ่มข้อมูลในช่องนั้นๆ

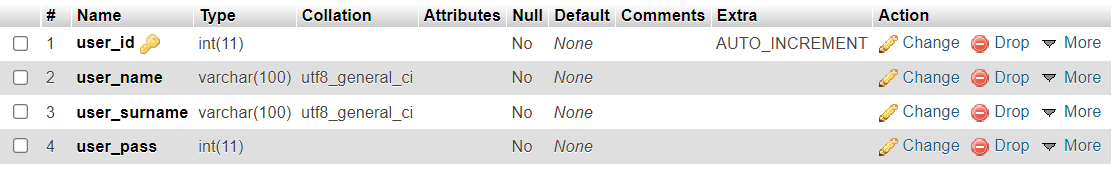
**Move column** ใช้ในการจัดเรียงคอลัมป์





**5.2 ) การแก้ไขข้อมูลตารางใน PhpMyAmin**

**Change** ใช้แก้ใขข้อมูลในคอลัมป์นั้นๆ

**Drop** ใช้ในการลบคอลัมป์นั้นๆ

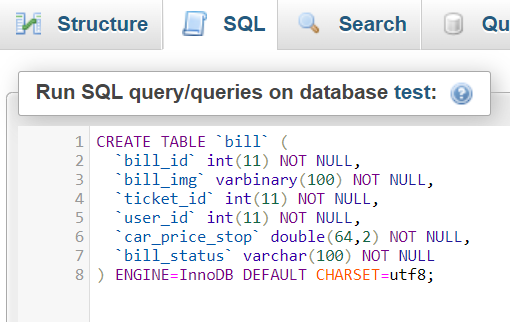


นอกจากนี้เราสามารถสร้าง table โดยวิธีการเขียนโค้ด sql ได้อีกด้วย

โดยใช้คำสั่ง **Create Table**

**5.3 ) วิธีการ Create Table**

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



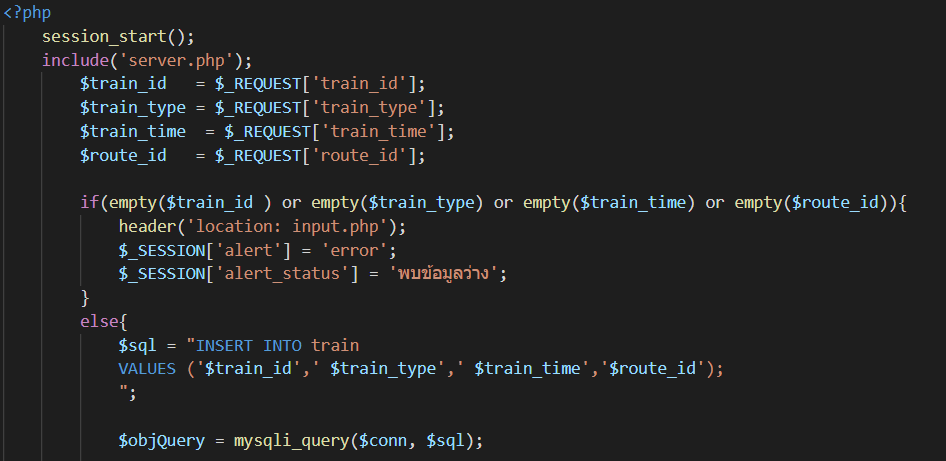


**บทที่ 6**

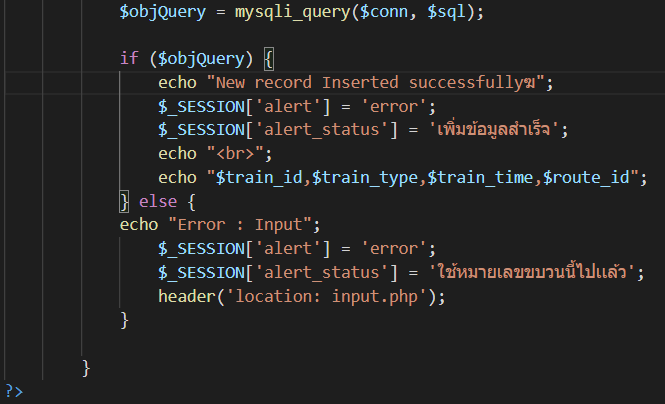
**Insert & Delete Data**

**6.1 ) โค้ดการ Insert ข้อมูล**

6.1.1) การ **Insert** ข้อมูลของตาราง **train**



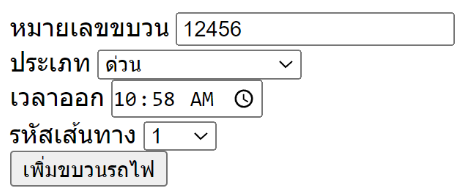


1. มีการประกาศเรียกใช้ function session start เพื่อเรียกใช้ตัวแปร $\_SESSION
2. มีการเชื่อมดาต้าเบสกับ server (server.php)
3. มีการรับค่ามาจากฟอร์มใน html ผ่านคำสั่ง $\_REQUEST
4. มีการดักค่าว่างถ้า user ไม่ได้กรอกค่าลงในฟอร์ม จะทำการรีเฟรชหน้าเว็บไซต์
5. มีการนำข้อมูลที่เก็บไว้ในตัวแปร $train\_id,$train\_type,$train\_time,$route\_id ไป Insert ข้อมูลในตาราง train ในดาต้าเบส



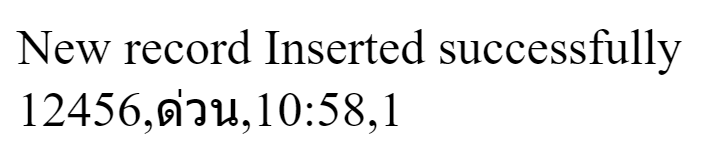
1. มีการเรียกใช้ฟังก์ชัน mysqli\_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ Insert ข้อมูลเข้าไป
2. มีการใช้ if เข้ามาแยกระหว่างข้อมูลที่สามารถ insert ผ่านและไม่ผ่าน

**เพิ่มขบวนรถไฟ**



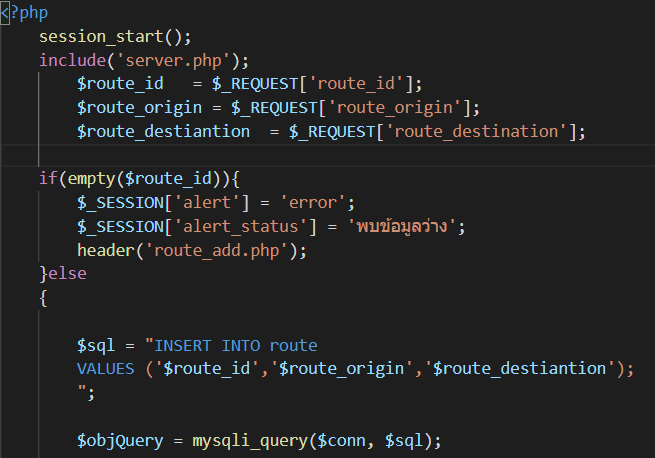
1. ทำการใส่ข้อมูลลงในฟอร์มทั้ง 4 ช่อง (หมายเลขขบวน,ประเภท,เวลาออก,รหัสเส้นทาง)
2. หลังจากนั้น **กดเพิ่มขบวนรถไฟ**
3. ตัว code จะพาไปหน้าต่างที่แสดงข้อมูลที่ถูกเพิ่มเข้าไป

**ข้อความหลังจากทำการ insert ข้อมูล**

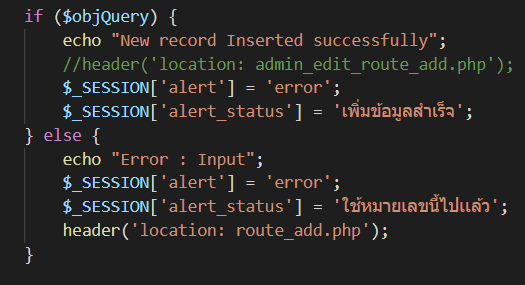


1. มีการแสดงข้อมูลว่าข้อมูลที่เพิ่มเข้าไป

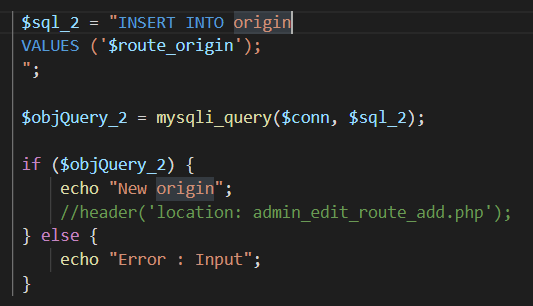
6.1.2) การ **Insert** ข้อมูลของตาราง **route**



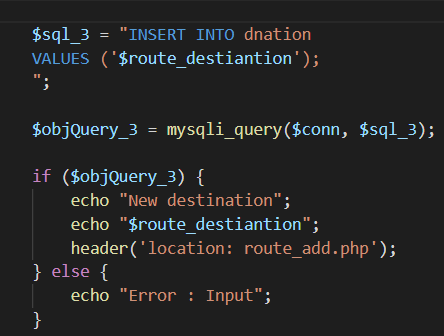
1. มีการประกาศเรียกใช้ function session start เพื่อเรียกใช้ตัวแปร $\_SESSION
2. มีการเชื่อมดาต้าเบสกับ server (server.php)
3. มีการรับค่ามาจากฟอร์มใน html ผ่านคำสั่ง $\_REQUEST
4. มีการดักค่าว่างถ้า user ไม่ได้กรอกค่าลงในฟอร์ม จะทำการรีเฟรชหน้าเว็บไซต์
5. มีการนำข้อมูลที่เก็บไว้ในตัวแปร $route\_id , $route\_origin , $route\_destination ไป Insert ข้อมูลในตาราง route ในดาต้าเบส
6. มีการเรียกใช้ฟังก์ชัน mysqli\_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ Insert ข้อมูลเข้าไป



1. ถ้าข้อมูลถูกต้องจะแจ้งเตือนว่าเพิ่มข้อมูขสำเร็จ ถ้ามี route\_id ซ้ำจะแจ้งเตือน error และให้กรอกค่าใหม่

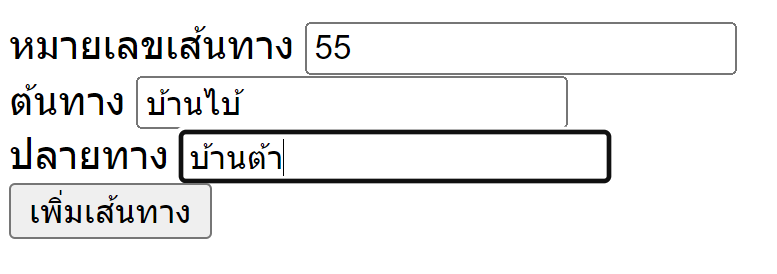
**

1. มีการนำค่าที่เก็บไว้ในตัวแปร $route\_origin ไป insertข้อมูลในตาราง origin ในดาต้าเบส

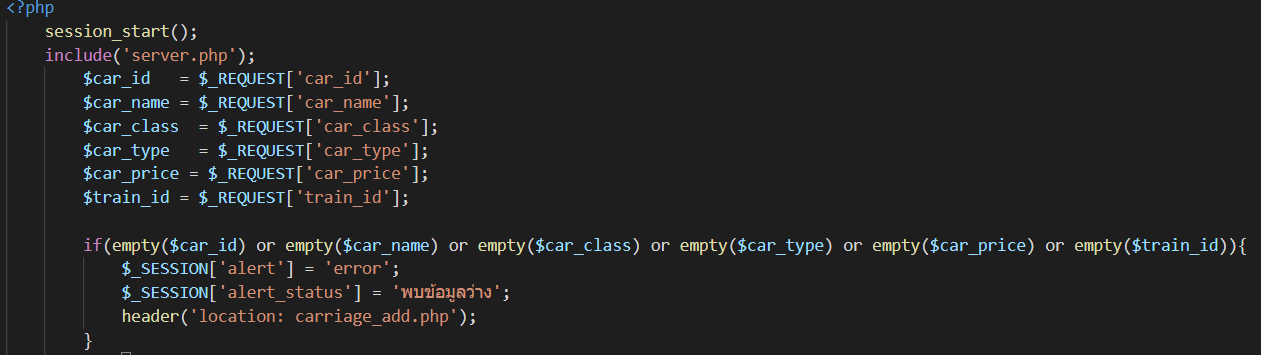


1. มีการนำค่าที่เก็บไว้ในตัวแปร $route\_destination ไป insert ข้อมูลในตาราง dnation ในดาต้าเบส

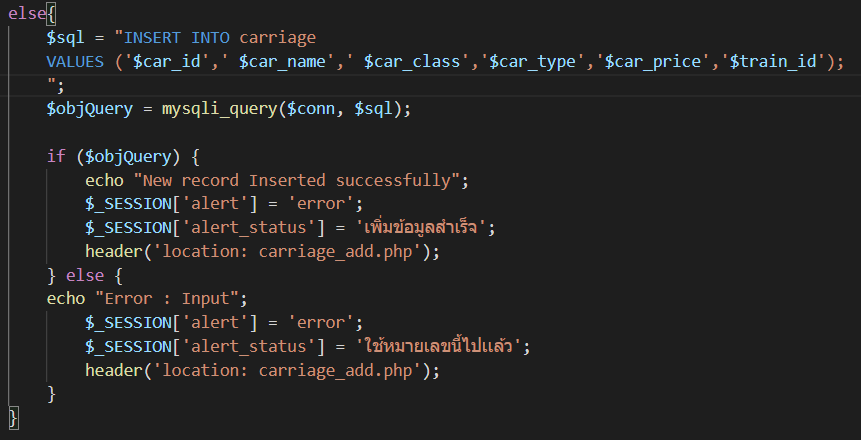
**เพิ่มเส้นทางเดินรถ**

****

6.1.3) การ **Insert** ข้อมูลในตาราง **carriage**

****

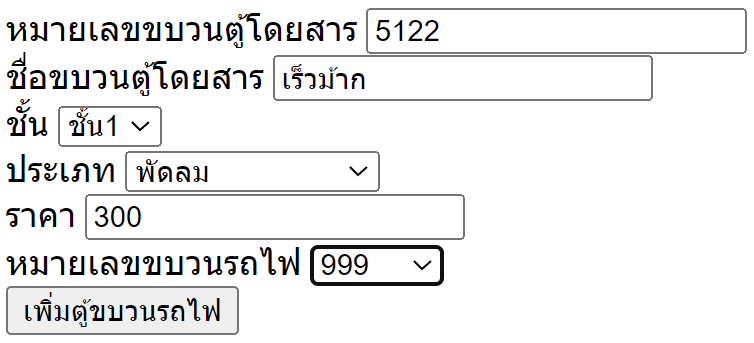
1. มีการประกาศเรียกใช้ function session start เพื่อเรียกใช้ตัวแปร $\_SESSION
2. มีการเชื่อมดาต้าเบสกับ server (server.php)
3. มีการรับค่ามาจากฟอร์มใน html ผ่านคำสั่ง $\_REQUEST
4. มีการดักค่าว่างถ้า user ไม่ได้กรอกค่าลงในฟอร์ม จะทำการรีเฟรชหน้าเว็บไซต์





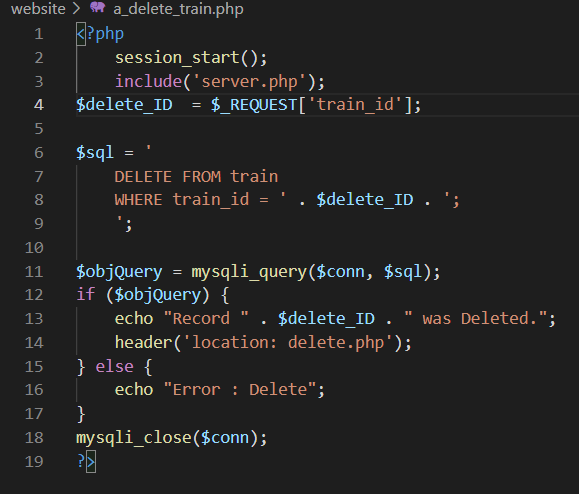
1. มีการนำข้อมูลที่เก็บไว้ในตัวแปร $car\_id , $car\_name , $car\_class , $car\_type , $car\_price , $train\_id ไป Insert ข้อมูลในตาราง carriage ในดาต้าเบส
2. มีการเรียกใช้ฟังก์ชัน mysqli\_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ Insert ข้อมูลเข้าไป
3. และถ้าหากมีการใส่ข้อมูลซ้ำจะ error และระบบจะให้ทำการกรอกใหม่อีกครั้ง

**เพิ่มข้อมูลตู้ขบวนรถไฟ**



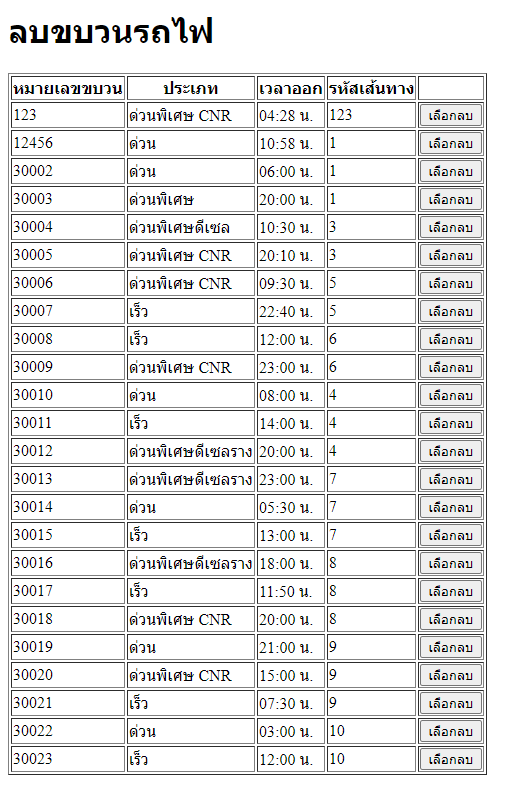


* 1. **)** **โค้ดการ delete** **ข้อมูล**
     1. การ **delete** ข้อมูลในตาราง **train**

****

* + 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 train\_id
    6. มีการเรียกใช้ฟังก์ชัน mysqli\_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ Delete ข้อมูล
    7. มีการใช้ฟังก์ชัน mysqli\_close เพื่อยุติการเชื่อมต่อกับฐานข้อมูล
    8. ถ้าการลบไม่สำเร็จจะขึ้นแจ้งเตือนแอดมินว่า “Error : Delete”

**หน้าต่างลบข้อมูลขบวนรถไฟ**





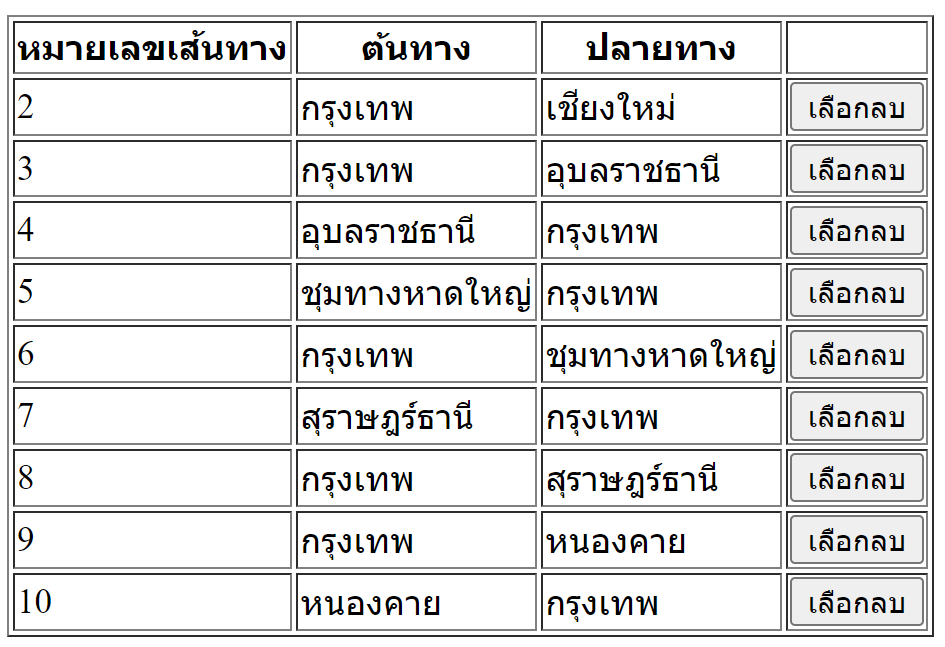
1. กดปุ่ม **เลือกลบ** เพื่อลบ row ที่ต้องการ
2. หลังจากลบ row ที่เลือกจะหายไปจากหน้าเว็บไซต์
   * 1. การ **delete** ข้อมูลในตาราง **route**

****



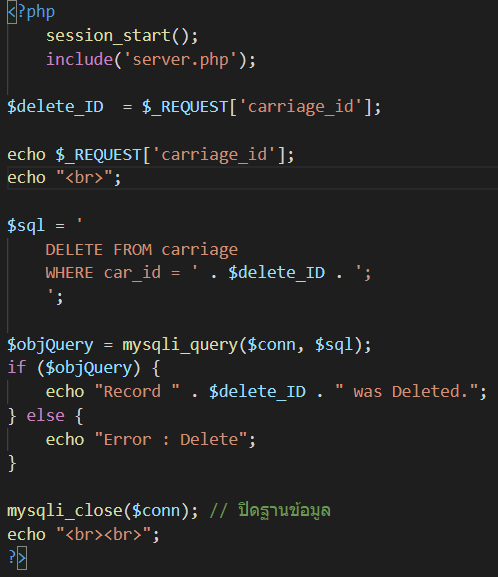
1. มีการประกาศเรียกใช้ function session start เพื่อเรียกใช้ตัวแปร $\_SESSION
2. มีการเชื่อมดาต้าเบสกับ server (server.php)
3. มีการรับค่ามาจากฟอร์มใน html ผ่านคำสั่ง $\_REQUEST ใส่ในตัวแปร $delete\_ID
4. มีการเรียกใช้ฟังก์ชัน mysqli\_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ delete ข้อมูล
5. มีการลบค่าโดยอ้างอิงข้อมูลมาจาก route\_id
6. มีการเรียกใช้ฟังก์ชัน mysqli\_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ Delete ข้อมูล
7. มีการใช้ฟังก์ชัน mysqli\_close เพื่อยุติการเชื่อมต่อกับฐานข้อมูล
8. ถ้าการลบไม่สำเร็จจะขึ้นแจ้งเตือนแอดมินว่า “Error : Delete”

**หน้าต่างลบเส้นทางเดินรถ**

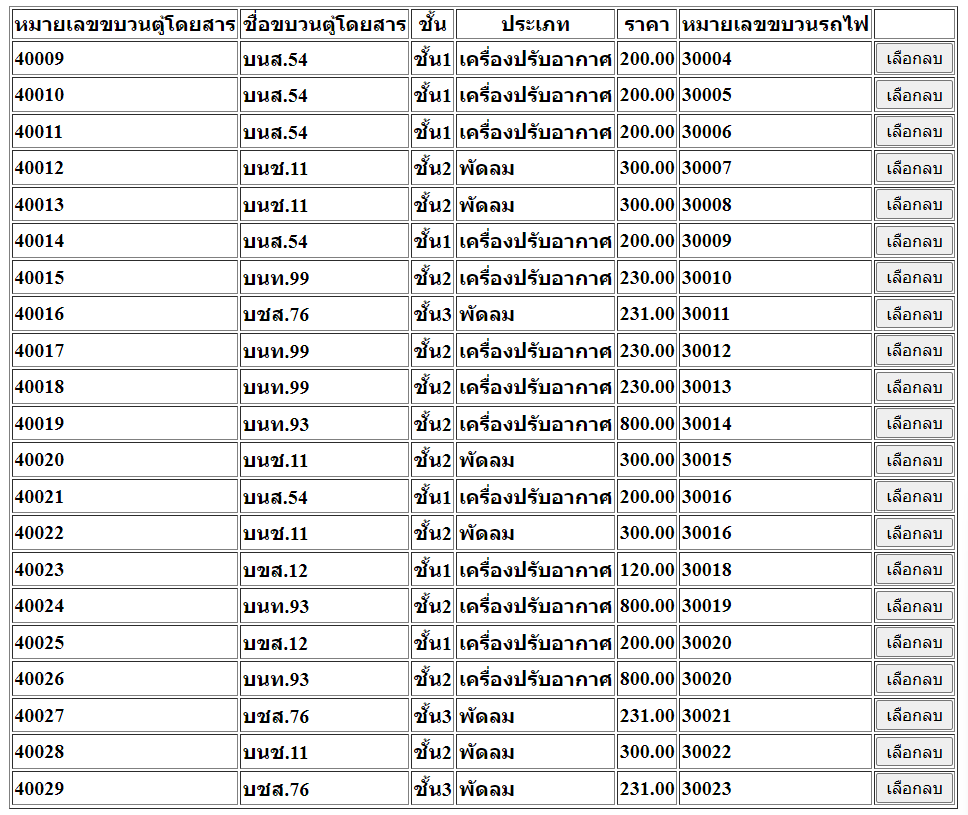




1. กดปุ่ม **เลือกลบ** เพื่อลบ row ที่ต้องการ
2. หลังจากลบ row ที่เลือกจะหายไปจากหน้าเว็บไซต์
   * 1. การ **delete** ข้อมูลในตาราง **carriage**



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 car\_id
6. มีการเรียกใช้ฟังก์ชัน mysqli\_query ให้มีการติดต่อไปที่ฐานข้อมูลเพื่อทำการ Delete ข้อมูล
7. มีการใช้ฟังก์ชัน mysqli\_close เพื่อยุติการเชื่อมต่อกับฐานข้อมูล
8. ถ้าการลบไม่สำเร็จจะขึ้นแจ้งเตือนแอดมินว่า “Error : Delete”

**หน้าต่างลบรายการตู้รถไฟ** 



1. กดปุ่ม **เลือกลบ** เพื่อลบ row ที่ต้องการ
2. หลังจากลบ row ที่เลือกจะหายไปจากหน้าเว็บไซต์

**บทที่ 7**

**การติดต่อฐานข้อมูล**

**7.1 ) การติดต่อฐานข้อมูล ในภาษาPHP**

คำสั่ง MYSQL\_CONNECT( ) ใช้ในการติดต่อฐานข้อมูล

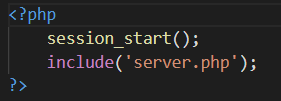






**7.2 ) การแสดงข้อมูลจากฐานข้อมูลบนหน้าเว็บไซต์**

สามารถแสดงข้อมูลจากฐานข้อมูลโดยใช้คำสั่ง **SELECT**

**ขั้นตอนที่1**

เรียกฟังก์ชันที่ใช้ติดต่อ SERVER

**SESSION\_START( );**

ใช้สำหรับประกาศตัวแปร **$\_SESSION**

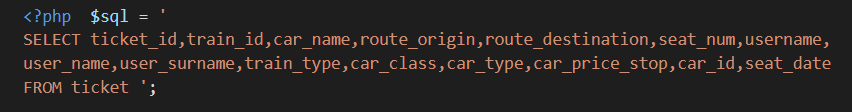
**Include(** )

ใช้ในการรวมไฟล์ โดยเราจะนำไฟล์ที่ใช้ติดต่อฐานข้อมูลมารวมด้วย

**ขั้นตอนที่2**

ใช้คำสั่ง **SELECT** ระบุชื่อคอลัมป์ และ ระบุชื่อตาราง ทีเราต้องการแสดงผลในหน้าเว็บ

จากนั้นสร้างตัวแปรมารับค่า

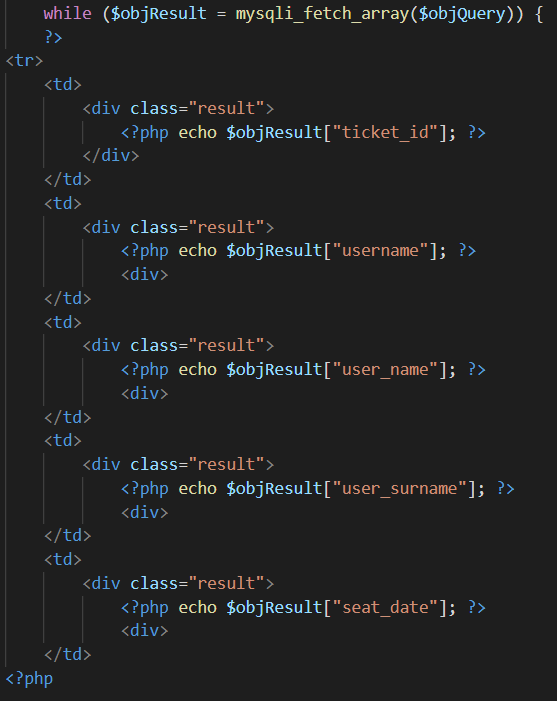
**MYSQLI\_QU ( )** ใช้ในการ QUERY ข้อมูลจาก DB 



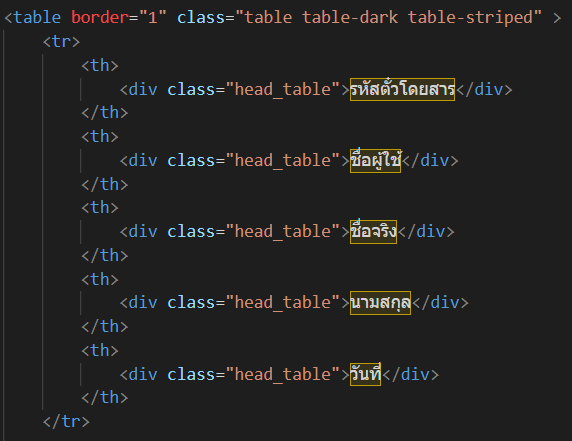
**ขั้นตอนที่** **3**

สร้าง TABLE ใช้สำหรับแสดงข้อมูล

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

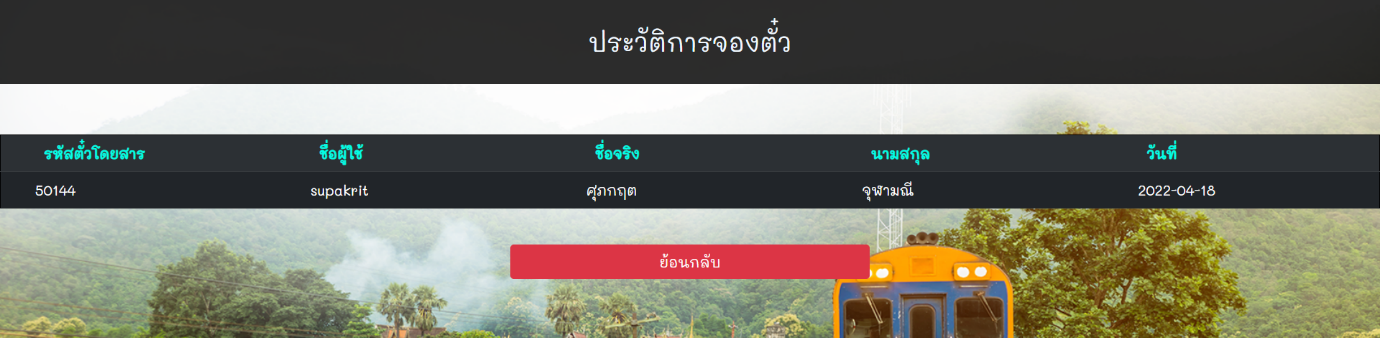
**MYSQLI\_FETCH\_ARRAY( )** 

เรียกแถวผลลัพท์เป็น ARRAY



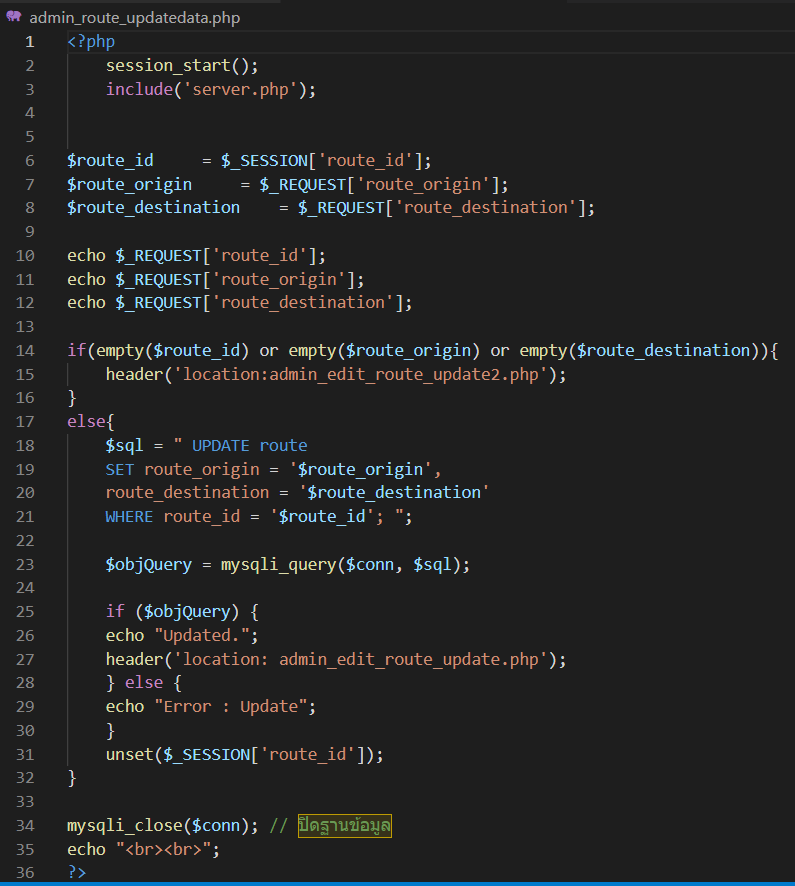


จะได้ผลลัพท์ดังนี้



**7.3 ) การแก้ข้อมูลจากฐานข้อมูล**

1. สามารถแก้ใขข้อมูลจากฐานข้อมูลโดยใช้คำสั่ง **UPDATE**













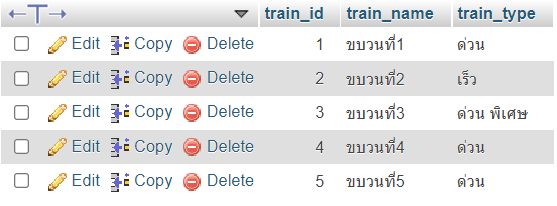
**7.4 relational algebra**

**SQL Operator**

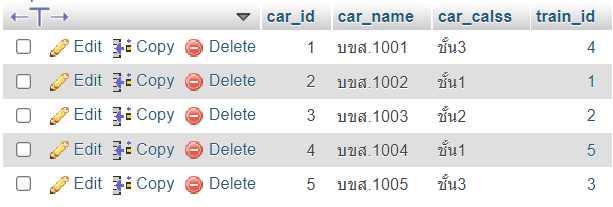
SQL Operator การทำงานที่เลือกข้อมูลในเงื่อนใข โดยระบบฐานข้อมูล จะมีการจอง key word บางคำ หรือ ตัวอักษรไว้สำหรับให้ SQL statement ใช้ระบุเงือนไข เพื่อเป็นการสั่งระบบให้ทำตามเงือนไขที่วางไว้ เช่น การเปรียบเทียบ การคำนวณทางคณิตศาสตร์ การทำกระบวนการเหล่านี้มักจะใช้ระบุเงือนไขภายใต้ SQL statement และ สามารถกำหนกได้หลายเงือนไขภายใน statement นั้นๆ

ตารางที่จะนำไปเป็นตัวอย่างการ operator

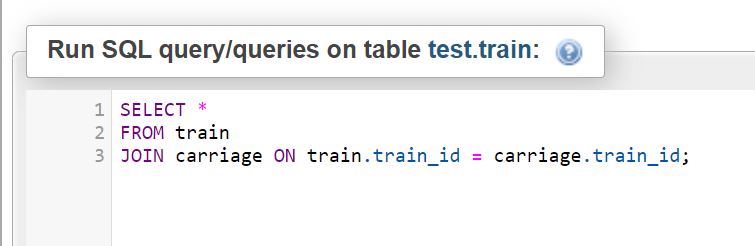
**Table : train**

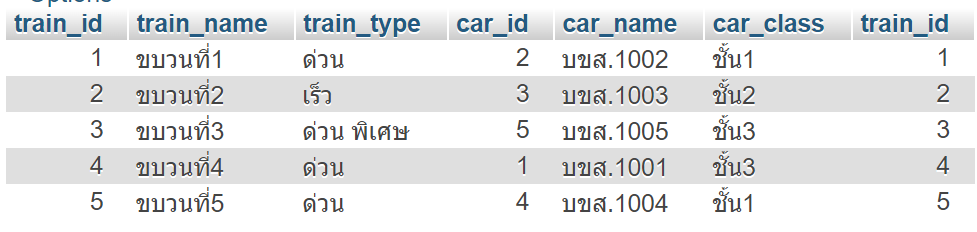
****



**Table = carriage**

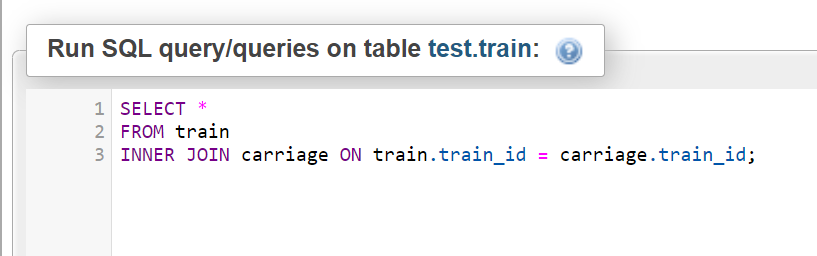


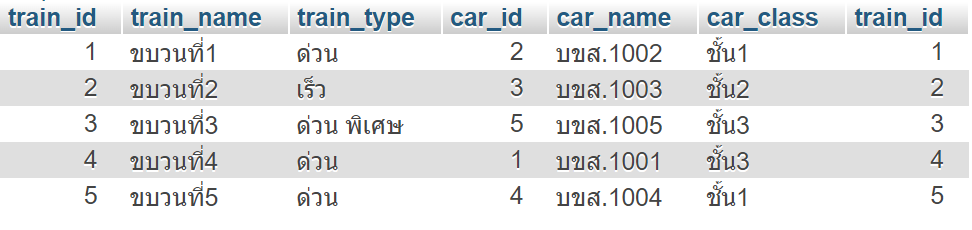
* + 1. การ **JOIN** แบบบปกติ



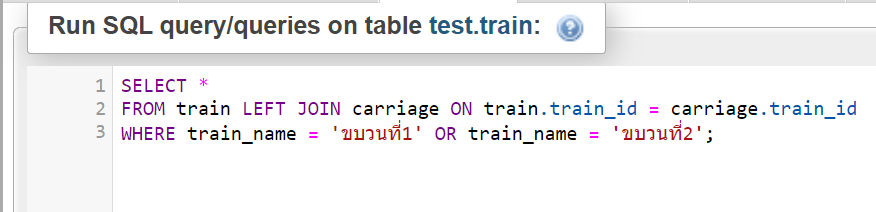
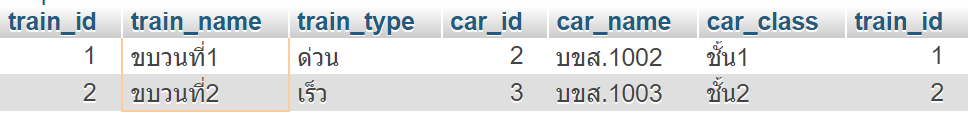


7.4.2) **INNER JOIN** แสดงผลข้อมูลที่มีค่าทั้ง 2 table เหมือนกันใน column ที่ระบุ

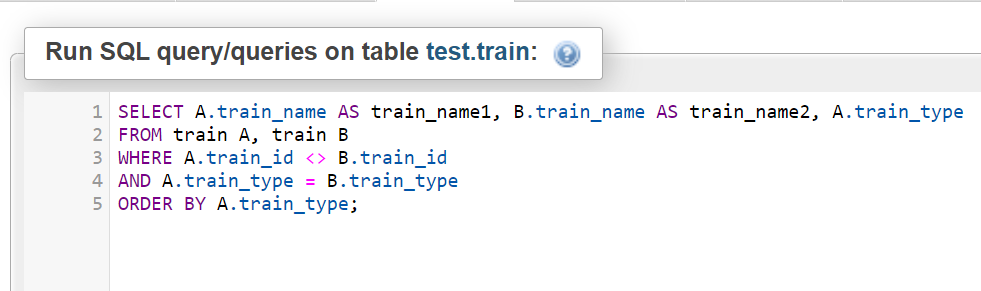


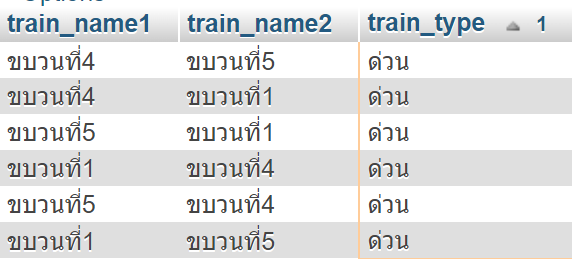


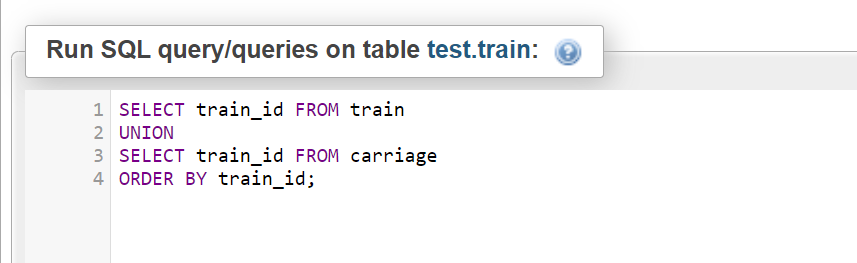


7.4.3) **LEFT JOIN** แสดงผลข้อมูลของ table ทางซ้ายทั้งหมด (table 1) และ ข้อมูลของ table ทางขวาที่ตรงเงื่อนไข (table 2)



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



7.4.5) **UNION** เราสามารถนำข้อมูลมารวมกันในเชิงบรรทัด ด้วยคำสั่ง SQL UNION statement

