

# SQL

แบบฝึกหัดพัฒนาทักษะภาษา SQL  
สำหรับโปรแกรมเมอร์

Practice with "Real" SQL Problems



โค้ดเชก

# SQL

แบบฝึกหัดพัฒนาทักษะภาษา SQL สำหรับโปรแกรมเมอร์

โค้ชเอก

แจกฟรีที่ <http://www.codingthailand.com/site/index.php?r=site/sqlbook>

1 มกราคม 2559

หนังสือเล่มนี้ผมตั้งใจจัดทำขึ้นเพื่ออยากแบ่งปัน และให้ความรู้เกี่ยวกับภาษา SQL ซึ่งเป็นภาษาสำคัญมากๆ สำหรับการจัดการฐานข้อมูล โดยเฉพาะมือใหม่ที่จะต้องทำความเข้าใจ และศึกษามาก่อนเป็นอันดับต้นๆ หวังว่าหนังสือเล่มนี้จะเป็นประโยชน์ ประหยัดเวลาการเรียนรู้ ขอให้มีความสุข และสนุกกับการเรียนรู้ทุกคน สวัสดิ์ปีใหม่ 2016 ครับ

“ฝันให้ใหญ่ แต่วินัยต้องเริ่มจากเล็กๆ”



©2015 - 2016 โค้ชเอก

# สารบัญ

แบบฝึกหัดที่ 1 ร้านขายอุปกรณ์คอมพิวเตอร์

แบบฝึกหัดที่ 2 ระบบจัดการข้อมูลพนักงาน

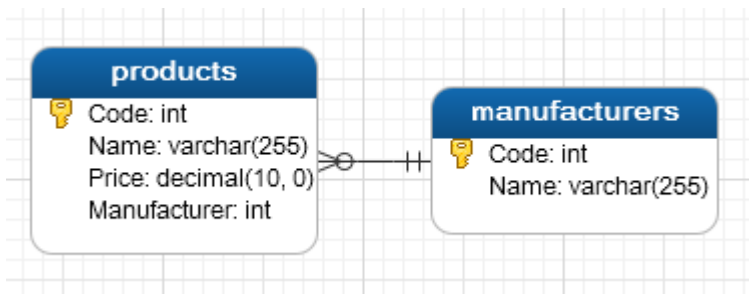
แบบฝึกหัดที่ 3 ระบบคลังสินค้า

แบบฝึกหัดที่ 4 ระบบโรงภาพยนตร์ (โรงหนัง)

แบบฝึกหัดที่ 5 ระบบจัดหาวัสดุ

**\*\*ตัวอย่างคำสั่ง SQL สำหรับสร้างตาราง เหมาะกับ MySQL/MariaDB แต่ก็สามารถประยุกต์  
ได้กับทุกฐานข้อมูล\*\***

## แบบฝึกหัดที่ 1 ร้านขายอุปกรณ์คอมพิวเตอร์



### คำอธิบายตาราง

ตารางสินค้า (products)

เก็บข้อมูลสินค้า

ตารางโรงงาน (manufacturers)

เก็บข้อมูลโรงงาน

### คำสั่ง SQL

```
CREATE TABLE Manufacturers (
```

```
Code INTEGER,
```

```
Name VARCHAR(255) NOT NULL,
```

```
PRIMARY KEY (Code)
```

```
);
```

```
CREATE TABLE Products (
```

```
Code INTEGER,
```

```
Name VARCHAR(255) NOT NULL ,
```

```
Price DECIMAL NOT NULL ,
```

```
Manufacturer INTEGER NOT NULL,
```

```
PRIMARY KEY (Code),
```

```
FOREIGN KEY (Manufacturer) REFERENCES Manufacturers(Code)
```

```
) ENGINE=INNODB;
```

```

INSERT INTO Manufacturers(Code,Name) VALUES(1,'Sony');
INSERT INTO Manufacturers(Code,Name) VALUES(2,'Creative Labs');
INSERT INTO Manufacturers(Code,Name) VALUES(3,'Hewlett-Packard');
INSERT INTO Manufacturers(Code,Name) VALUES(4,'Iomega');
INSERT INTO Manufacturers(Code,Name) VALUES(5,'Fujitsu');
INSERT INTO Manufacturers(Code,Name) VALUES(6,'Winchester');

INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(1,'Hard drive',240,5);
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(2,'Memory',120,6);
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(3,'ZIP drive',150,4);
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(4,'Floppy disk',5,6);
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(5,'Monitor',240,1);
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(6,'DVD drive',180,2);
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(7,'CD drive',90,2);
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(8,'Printer',270,3);
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(9,'Toner cartridge',66,3);
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(10,'DVD burner',180,2);

```

## แบบฝึกหัด

-- 1.1 แสดง ชื่อสินค้า ทั้งหมดจากตารางสินค้า

```
select name from products;
```

-- 1.2 แสดง ชื่อสินค้าและราคา ทั้งหมดจากตารางสินค้า

```
select name, price from products;
```

-- 1.3 แสดงชื่อสินค้า เฉพาะสินค้าที่มีราคาน้อยกว่าหรือเท่ากับ 200

```
select name from products where price <= 200;
```

-- 1.4 แสดงข้อมูลสินค้าทั้งหมด เฉพาะสินค้าที่มีราคาตั้งแต่ 60 ถึง 120

```
select * from products where price between $60 and $120;
```

```
select * from products where price >= 60 and price <= 120;
```

-- 1.5 แสดงชื่อสินค้า และราคาในหน่วย cents (เอาราคาคูณด้วย 100).

```
select name, price*100 from products;
```

```
select name, concat(price*100, ' cents') from products;
```

-- 1.6 แสดงค่าเฉลี่ยของราคาสินค้าทั้งหมด

```
select avg(price) from products;
```

```
select sum(price)/count(price) from products;
```

-- 1.7 แสดงค่าเฉลี่ยของราคาสินค้าทั้งหมด เฉพาะที่มีรหัสโรงงาน เท่ากับ 2

```
select avg(price) from products where Manufacturer = 2;
```

-- 1.8 นับจำนวนสินค้า ที่มีราคาน้อยกว่าหรือเท่ากับ \$180

```
select count(*) from products where price>=180;
```

-- 1.9 แสดงชื่อสินค้า และราคา ที่มีราคามากกว่าหรือเท่ากับ \$180 และให้

เรียงลำดับราคาจากมากไปหาน้อย และเรียงชื่อสินค้าจากน้อยไปหามาก

```
select name, price from products where price>=180 order by price desc,  
name asc;
```

-- 1.10 แสดงข้อมูลสินค้าทั้งหมด พร้อมด้วยข้อมูลโรงงานในแต่ละรายการ

```
select a.*, b.name from products a join Manufacturers b
```

```
on(a.manufacturer = b.code);
```

```
select a.*, b.name from products a, Manufacturers b where
```

```
a.manufacturer = b.code;
```

-- 1.11 แสดงข้อมูลชื่อสินค้า ราคา และชื่อโรงงาน ทั้งหมด

```
select a.name, a.price, b.name from products a join Manufacturers b
```

```
on(a.manufacturer = b.code);
```

```
SELECT Products.Name, Price, Manufacturers.Name
```

```
FROM Products INNER JOIN Manufacturers
```

```
ON Products.Manufacturer = Manufacturers.Code;
```

-- 1.12 แสดงค่าเฉลี่ยราคาสินค้า รหัสโรงงาน ในแต่ละรหัสโรงงาน

```
SELECT AVG(Price), Manufacturer
```

```
FROM Products
```

```
GROUP BY Manufacturer;
```

-- 1.13 แสดงค่าเฉลี่ยราคาสินค้า ชื่อโรงงาน ในแต่ละโรงงาน

```
select avg(a.price), b.name
```

```
from Products a join Manufacturers b
```

```
on a.manufacturer = b.code
```

```
group by b.name;
```

-- 1.14 แสดงค่าเฉลี่ยราคาสินค้า ชื่อโรงงาน ในแต่ละโรงงาน ที่มีค่าเฉลี่ยสินค้า  
ต้องมากกว่าหรือเท่ากับ \$150

```
select avg(a.price), b.name
```

```
from Manufacturers b join Products a
```

```
on b.code = a.Manufacturer
```

```
group by b.name
```

```
having avg(a.price)>=150;
```

```
SELECT AVG(Price), Manufacturers.Name
```

```
FROM Products, Manufacturers
```

```
WHERE Products.Manufacturer = Manufacturers.Code
```

```
GROUP BY Manufacturers.Name
```

```
HAVING AVG(Price) >= 150;
```

-- 1.15 แสดงชื่อสินค้า และราคาสินค้าที่ถูกที่สุด

```
select name, price from Products
```

```
where price = (
```

```
select min(price)
```

```
from products);
```



```
SELECT name,price  
  
FROM Products  
  
ORDER BY price ASC  
  
LIMIT 1;
```

```
SELECT Name, Price  
  
FROM Products  
  
WHERE Price = (SELECT MIN(Price) FROM Products);
```

```
-- 1.16 เพิ่มสินค้าใหม่ ชื่อสินค้า Loudspeakers ราคา $70 รหัสโรงงาน 2  
  
insert into Products values (11, 'Loudspeakers', 70, 2);
```

```
-- 1.17 แก้ไขชื่อสินค้า ของรหัส 8 เป็น "Laser Printer"  
  
update products  
  
set name = 'Laser Printer'  
  
where code=8;
```

```
-- 1.18 ลดราคาสินค้าทั้งหมดลง 10%  
  
update products  
  
set price=price*0.9;
```

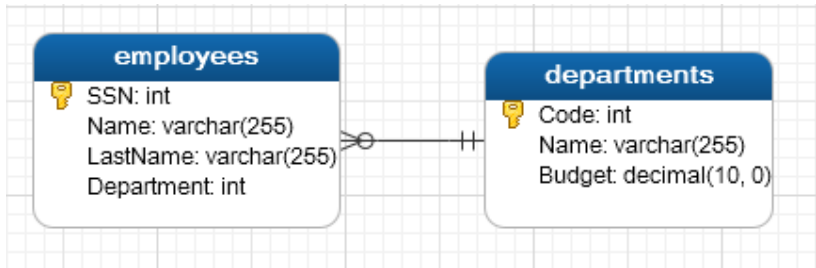
-- 1.19 ลดราคาสินค้าทั้งหมดลง 10% โดยให้ลดเฉพาะสินค้าที่มีราคามากกว่า  
หรือเท่ากับ \$120

update products

set price = price \* 0.9

where price >= 120;

## แบบฝึกหัดที่ 2 ระบบจัดการข้อมูลพนักงาน



### คำอธิบายตาราง

ตารางพนักงาน (employees)      เก็บข้อมูลพนักงาน

ตารางแผนก (departments)      เก็บข้อมูลแผนก

### คำสั่ง SQL

```
CREATE TABLE Departments (  
    Code INTEGER PRIMARY KEY,  
    Name varchar(255) NOT NULL ,  
    Budget decimal NOT NULL  
);
```

```
CREATE TABLE Employees (  
    SSN INTEGER PRIMARY KEY,  
    Name varchar(255) NOT NULL ,  
    LastName varchar(255) NOT NULL ,  
    Department INTEGER NOT NULL ,  
    foreign key (department) references Departments(Code)  
)  
ENGINE=INNODB;
```

```

INSERT INTO Departments(Code,Name,Budget) VALUES(14,'IT',65000);
INSERT INTO Departments(Code,Name,Budget) VALUES(37,'Accounting',15000);
INSERT INTO Departments(Code,Name,Budget) VALUES(59,'Human Resources',240000);
INSERT INTO Departments(Code,Name,Budget) VALUES(77,'Research',55000);

INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('123234877','Michael','Rogers',14);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('152934485','Anand','Manikutty',14);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('222364883','Carol','Smith',37);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('326587417','Joe','Stevens',37);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('332154719','Mary-Anne','Foster',14);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('332569843','George','ODonnell',77);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('546523478','John','Doe',59);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('631231482','David','Smith',77);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('654873219','Zacary','Efron',59);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('745685214','Eric','Goldsmith',59);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('845657245','Elizabeth','Doe',14);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('845657246','Kumar','Swamy',14);

```

## แบบฝึกหัด

-- 2.1 แสดงข้อมูลนามสกุลของพนักงานทั้งหมด

```
select LastName from Employees;
```

-- 2.2 แสดงนามสกุลพนักงาน เฉพาะนามสกุลที่ไม่ซ้ำกัน

```
select distinct LastName from employees;
```

-- 2.3 แสดงข้อมูลพนักงานทั้งหมด ที่มีนามสกุล 'Smith'

```
select * from employees where lastname = 'Smith';
```

-- 2.4 แสดงข้อมูลพนักงานทั้งหมด ที่มีนามสกุล "Smith" หรือ "Doe".

```
select * from Employees where lastname in ('Smith', 'Doe');
```

```
select * from Employees where lastname = 'Smith' or lastname = 'Doe';
```

-- 2.5 แสดงข้อมูลพนักงานทั้งหมด ที่ทำงานในแผนก 14

```
select * from Employees where department = 14;
```

-- 2.6 แสดงข้อมูลพนักงานทั้งหมด ที่ทำงานในแผนก 37 หรือแผนก 77

```
select * from employees where department = 37 or department = 77;
```

```
select * from employees where department in (37, 77);
```

-- 2.7 แสดงข้อมูลทั้งหมดของพนักงาน ที่มีนามสกุลขึ้นต้นด้วย S

```
select * from employees where LastName like 'S%';
```

-- 2.8 แสดงผลรวมงบประมาณ (Budget) ของทุกแผนก

```
select sum(budget) from Departments;
```

```
select Name, sum(Budget) from Departments group by Name;
```

-- 2.9 แสดงรหัสแผนก จำนวนพนักงาน ในแต่ละรหัสแผนก

```
select Department, count(*) from employees group by department;
```

```
SELECT Department, COUNT(*)  
  
FROM Employees  
  
GROUP BY Department;
```

-- 2.10 แสดงข้อมูลพนักงานทั้งหมด และข้อมูลแผนกทั้งหมด ของพนักงานแต่ละคน

```
select a.*, b.* from employees a join departments b on a.department =  
b.code;
```

```
SELECT SSN, E.Name AS Name_E, LastName, D.Name AS Name_D,  
Department, Code, Budget  
  
FROM Employees E INNER JOIN Departments D  
  
ON E.Department = D.Code;
```

-- 2.11 แสดงชื่อพนักงาน นามสกุล ของพนักงานที่ทำงานในแผนก ที่มี  
งบประมาณ (budget) มากกว่า \$60,000

```
select name, lastname  
  
from employees  
  
where department in (  
  
select code from departments where Budget>60000  
  
);
```

/\* Without subquery \*/

```
SELECT Employees.Name, LastName  
FROM Employees INNER JOIN Departments  
ON Employees.Department = Departments.Code  
AND Departments.Budget > 60000;
```

/\* With subquery \*/

```
SELECT Name, LastName FROM Employees  
WHERE Department IN  
(SELECT Code FROM Departments WHERE Budget > 60000);
```

-- 2.12 แสดงข้อมูลแผนกทั้งหมดที่มีงบประมาณมากกว่า งบประมาณเฉลี่ยทั้ง  
แผนก

```
select *  
from departments  
where budget > (  
select avg(budget) from departments  
);
```

```
SELECT *  
FROM Departments
```

WHERE Budget >

(

SELECT AVG(Budget)

FROM Departments

);

-- 2.13 แสดงชื่อ นามสกุล ของพนักงานที่ทำงานในแผนกที่มีงบประมาณต่ำที่สุด

2 แผนก

select name, lastname

from employees

where department =(

select temp.code

from (select \* from departments order by budget limit 2) temp

order by temp.budget desc limit 1

);

/\* With subquery \*/

SELECT e.Name, e.LastName

FROM Employees e

WHERE e.Department = (

SELECT sub.Code



FROM (SELECT \* FROM Departments d ORDER BY d.budget LIMIT

2) sub

ORDER BY budget DESC LIMIT 1);

-- 2.14

-- เพิ่มข้อมูลแผนกใหม่ ประกอบด้วย ชื่อแผนก "Quality Assurance" มี  
งบประมาณ \$40000 และรหัสแผนกเป็น 11

-- เพิ่มข้อมูลพนักงาน ประกอบด้วย ชื่อ-สกุล "Mary Moore" มีรหัส SSN เท่ากับ  
847219811 และให้อยู่ในแผนก 11

insert into departments values(11, 'Quality Assurnce', 40000);

insert into employees values(847219811, 'Mary', 'Moore', 11);

-- 2.15

-- ลดงบประมาณทุกแผนกลง 10%.

update departments

set budget = 0.9 \* budget;

-- 2.16

-- ลบข้อมูลพนักงาน ในแผนก IT ทั้งหมด (รหัส 14)

delete from employees

where department = 14;

-- 2.17

-- ลบข้อมูลพนักงานทั้งหมด ที่ทำงานในแผนก ที่มีงบประมาณมากกว่าหรือเท่ากับ \$60,000

delete from employees

where department in (

select code

from departments

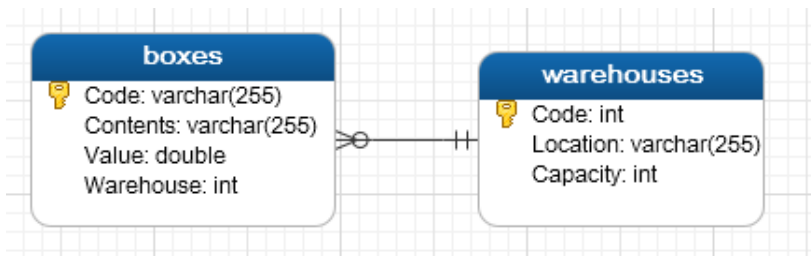
where budget>=60000

);

-- 2.18 ลบข้อมูลพนักงานทั้งหมด

delete from employees;

### แบบฝึกหัดที่ 3 ระบบคลังสินค้า



#### คำอธิบายตาราง

ตารางกล่องสินค้า (boxes)

เก็บข้อมูลกล่องสินค้า

ตารางคลังสินค้า (warehouses)

เก็บข้อมูลคลังสินค้า

#### คำสั่ง SQL

```
CREATE TABLE Warehouses (  
    Code INTEGER NOT NULL,  
    Location VARCHAR(255) NOT NULL ,  
    Capacity INTEGER NOT NULL,  
    PRIMARY KEY (Code)  
);  
  
CREATE TABLE Boxes (  
    Code VARCHAR(255) NOT NULL,  
    Contents VARCHAR(255) NOT NULL ,  
    Value REAL NOT NULL ,  
    Warehouse INTEGER NOT NULL,  
    PRIMARY KEY (Code),  
    FOREIGN KEY (Warehouse) REFERENCES Warehouses(Code)  
)  
ENGINE=INNODB;
```

```

INSERT INTO Warehouses(Code,Location,Capacity) VALUES(1,'Chicago',3);
INSERT INTO Warehouses(Code,Location,Capacity) VALUES(2,'Chicago',4);
INSERT INTO Warehouses(Code,Location,Capacity) VALUES(3,'New York',7);
INSERT INTO Warehouses(Code,Location,Capacity) VALUES(4,'Los Angeles',2);
INSERT INTO Warehouses(Code,Location,Capacity) VALUES(5,'San Francisco',8);

INSERT INTO Boxes(Code,Contents,Value,Warehouse) VALUES('0MN7','Rocks',180,3);
INSERT INTO Boxes(Code,Contents,Value,Warehouse) VALUES('4H8P','Rocks',250,1);
INSERT INTO Boxes(Code,Contents,Value,Warehouse) VALUES('4RT3','Scissors',190,4);
INSERT INTO Boxes(Code,Contents,Value,Warehouse) VALUES('7G3H','Rocks',200,1);
INSERT INTO Boxes(Code,Contents,Value,Warehouse) VALUES('8JN6','Papers',75,1);
INSERT INTO Boxes(Code,Contents,Value,Warehouse) VALUES('8Y6U','Papers',50,3);
INSERT INTO Boxes(Code,Contents,Value,Warehouse) VALUES('9J6F','Papers',175,2);
INSERT INTO Boxes(Code,Contents,Value,Warehouse) VALUES('LL08','Rocks',140,4);
INSERT INTO Boxes(Code,Contents,Value,Warehouse) VALUES('P0H6','Scissors',125,1);
INSERT INTO Boxes(Code,Contents,Value,Warehouse) VALUES('P2T6','Scissors',150,2);
INSERT INTO Boxes(Code,Contents,Value,Warehouse) VALUES('TU55','Papers',90,5);

```

## แบบฝึกหัด

--3.1 แสดงข้อมูลทั้งหมดของคลังสินค้า

```
select * from warehouses;
```

--3.2 แสดงข้อมูลกล่องสินค้าทั้งหมดที่มี Value มากกว่า \$150

```
select * from boxes where Value>150;
```

--3.3 แสดงข้อมูล contents ที่ซ้ำกันทั้งหมด

```
select distinct contents from boxes;
```

--3.4 แสดงค่าเฉลี่ย value ของกล่องสินค้าทั้งหมด

```
select avg(value) from boxes;
```

--3.5 แสดงข้อมูลรหัสคลังสินค้า และค่าเฉลี่ย value ของกล่องสินค้า ในแต่ละ  
รหัสคลังสินค้า

```
select warehouse, avg(value) from boxes group by warehouse;
```

```
SELECT Warehouse, AVG(Value)
```

```
FROM Boxes
```

```
GROUP BY Warehouse;
```

--3.6 แสดงข้อมูลรหัสคลังสินค้า และค่าเฉลี่ย value ของกล่องสินค้า ในแต่ละ  
รหัสคลังสินค้า ที่มีค่าเฉลี่ย value มากกว่า 150

```
select warehouse, avg(value)
```

```
from boxes
```

```
group by warehouse
```

```
having avg(value)> 150;
```

--3.7 แสดงรหัสกล่อง สถานที่เก็บ (location) ในแต่ละกล่องสินค้าทั้งหมด

```
select boxes.code, warehouses.location
```

```
from boxes join warehouses
```

```
on boxes.Warehouse = Warehouses.Code;
```

```
SELECT Boxes.Code, Location
```

```
FROM Warehouses
```

```
INNER JOIN Boxes ON Warehouses.Code = Boxes.Warehouse;
```

--3.8 แสดงรหัสคลังสินค้า และนับจำนวนกล่องสินค้าทั้งหมด ในแต่ละรหัส  
คลังสินค้า

```
select Warehouse, count(*)
```

```
from boxes
```

```
group by warehouse;
```

--3.9 แสดงรหัสกล่องสินค้า ที่อยู่ใน Chicago

```
select Boxes.code
```

```
from boxes join Warehouses
```

```
on boxes.warehouse = warehouses.code
```

```
where warehouses.location = 'Chicago';
```

```
/* Without subqueries */
```

```
SELECT Boxes.Code
```

```
FROM Warehouses LEFT JOIN Boxes
```

```
ON Warehouses.Code = Boxes.Warehouse
```

```
WHERE Location = 'Chicago';
```

```
/* With a subquery */
```

```
SELECT Code
```

```
FROM Boxes
```

```
WHERE Warehouse IN
```

```
(
```

```
SELECT Code
```

```
FROM Warehouses
```

```
WHERE Location = 'Chicago'
```

```
);
```

--3.10 เพิ่มกล่องสินค้าใหม่ รหัส "H5RT" ข้างในบรรจุ Papers มี value \$200  
และอยู่ในคลังสินค้าที่ 2

```
INSERT INTO Boxes VALUES ('H5RT', 'Papers', 200, 2);
```

--3.11 ลดมูลค่า (value) ของกล่องสินค้าทุกกล่องลง 15%

```
update boxes
```

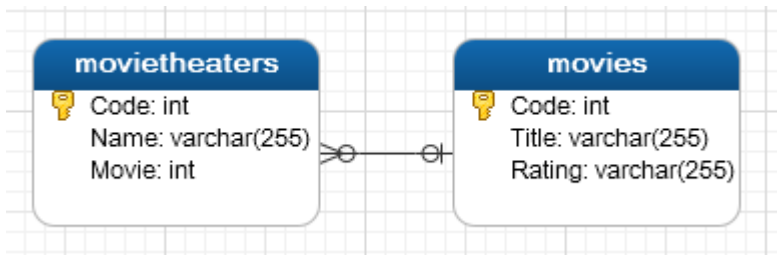
```
set value = value * 0.85;
```

--3.12 ลบข้อมูลกล่องสินค้าทั้งหมดที่มี value น้อยกว่า \$100

```
delete from boxes
```

```
where value < 100;
```

## แบบฝึกหัดที่ 4 ระบบโรงภาพยนตร์ (โรงหนัง)



### คำอธิบายตาราง

ตารางโรงฉาย movietheaters

เก็บข้อมูลโรงฉายภาพยนตร์

ตารางภาพยนตร์ (movie)

เก็บข้อมูลภาพยนตร์

### คำสั่ง SQL

```
CREATE TABLE Movies (  
    Code INTEGER PRIMARY KEY,  
    Title VARCHAR(255) NOT NULL,  
    Rating VARCHAR(255)  
);
```

```
CREATE TABLE MovieTheaters (  
    Code INTEGER PRIMARY KEY,  
    Name VARCHAR(255) NOT NULL,  
    Movie INTEGER,  
    FOREIGN KEY (Movie) REFERENCES Movies(Code)  
)  
ENGINE=INNODB;
```

```
INSERT INTO Movies(Code,Title,Rating) VALUES(1,'Citizen Kane','PG');
```

```
INSERT INTO Movies(Code,Title,Rating) VALUES(2,'Singin" in the Rain','G');
```



```

INSERT INTO Movies(Code,Title,Rating) VALUES(3,'The Wizard of Oz','G');
INSERT INTO Movies(Code,Title,Rating) VALUES(4,'The Quiet Man',NULL);
INSERT INTO Movies(Code,Title,Rating) VALUES(5,'North by Northwest',NULL);
INSERT INTO Movies(Code,Title,Rating) VALUES(6,'The Last Tango in Paris','NC-17');
INSERT INTO Movies(Code,Title,Rating) VALUES(7,'Some Like it Hot','PG-13');
INSERT INTO Movies(Code,Title,Rating) VALUES(8,'A Night at the Opera',NULL);

INSERT INTO MovieTheaters(Code,Name,Movie) VALUES(1,'Odeon',5);
INSERT INTO MovieTheaters(Code,Name,Movie) VALUES(2,'Imperial',1);
INSERT INTO MovieTheaters(Code,Name,Movie) VALUES(3,'Majestic',NULL);
INSERT INTO MovieTheaters(Code,Name,Movie) VALUES(4,'Royale',6);
INSERT INTO MovieTheaters(Code,Name,Movie) VALUES(5,'Paraiso',3);
INSERT INTO MovieTheaters(Code,Name,Movie) VALUES(6,'Nickelodeon',NULL);

```

## แบบฝึกหัด

-- 4.1 แสดงชื่อเรื่อง ของภาพยนตร์ทั้งหมด

```
select title from movies;
```

-- 4.2 แสดงอันดับ (rating) ของภาพยนตร์ที่มีอันดับซ้ำกันทั้งหมด

```
select distinct rating from movies;
```

-- 4.3 แสดงข้อมูลภาพยนตร์ทั้งหมด ที่ไม่ได้จัดอันดับ

```
select *
```

```
from movies
```

```
where rating is NULL;
```

-- 4.4 แสดงข้อมูลโรงหนังทั้งหมดที่ไม่ได้ฉายภาพยนตร์ ณ ตอนนี

```
select * from MovieTheaters
```

```
where Movie is NULL;
```

-- 4.5 แสดงข้อมูลโรงหนัง และภาพยนตร์ทั้งหมด โดยข้อมูลโรงหนัง จะมี  
ภาพยนตร์ฉายหรือไม่ฉายก็ให้แสดงด้วย

```
SELECT *
```

```
FROM MovieTheaters LEFT JOIN Movies
```

```
ON MovieTheaters.Movie = Movies.Code;
```

-- 4.6 แสดงชื่อภาพยนตร์ ที่ไม่ได้ฉายในโรงหนังใดๆเลย

```
select title
```

```
from movies
```

```
where code not in (
```

```
select movie from movietheaters
```

```
);
```

```
/* With JOIN */
```

```
SELECT Movies.Title
```

```
FROM MovieTheaters RIGHT JOIN Movies
```

```
ON MovieTheaters.Movie = Movies.Code
```

```
WHERE MovieTheaters.Movie IS NULL;
```

/\* With subquery \*/

SELECT Title FROM Movies

WHERE Code NOT IN

(

SELECT Movie FROM MovieTheaters

WHERE Movie IS NOT NULL

);

-- 4.7 จัดอันดับภาพยนตร์เป็น G เฉพาะภาพยนตร์ที่ยังไม่ได้รับการจัดอันดับ

update Movies

set Rating = 'G'

where Rating is NULL;

-- 4.8 ลบข้อมูลโรงหนัง ที่มีหนังจัดอันดับเป็น NC-17

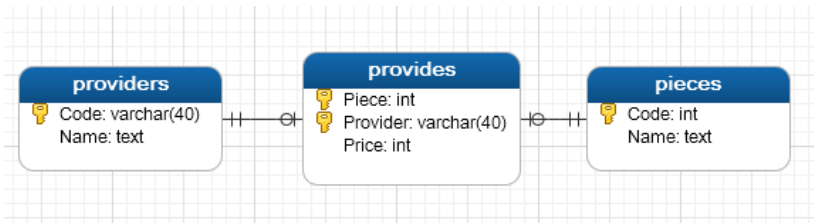
delete from MovieTheaters

where Movie in (

select Code from Movies where Rating = 'NC-17'

);

## แบบฝึกหัดที่ 5 ระบบจัดหาวัสดุ



### คำอธิบายตาราง

ตาราง providers

เก็บข้อมูลผู้จัดหาวัสดุ

ตาราง pieces

เก็บข้อมูลวัสดุ

ตาราง provides

เก็บรายละเอียดการจัดหา

### คำสั่ง SQL

```
CREATE TABLE Pieces (  
    Code INTEGER PRIMARY KEY NOT NULL,  
    Name TEXT NOT NULL  
);  
  
CREATE TABLE Providers (  
    Code VARCHAR(40)  
    PRIMARY KEY NOT NULL,  
    Name TEXT NOT NULL  
);  
  
CREATE TABLE Provides (  
    Piece INTEGER,  
    FOREIGN KEY (Piece) REFERENCES Pieces(Code),  
    Provider VARCHAR(40),  
    FOREIGN KEY (Provider) REFERENCES Providers(Code),
```

```
Price INTEGER NOT NULL,  
PRIMARY KEY(Piece, Provider)  
);
```

```
INSERT INTO Providers(Code, Name) VALUES('HAL','Clarke Enterprises');  
INSERT INTO Providers(Code, Name) VALUES('RBT','Susan Calvin Corp.');
```

```
INSERT INTO Providers(Code, Name) VALUES('TNBC','Skellington Supplies');
```

```
INSERT INTO Pieces(Code, Name) VALUES(1,'Sprocket');  
INSERT INTO Pieces(Code, Name) VALUES(2,'Screw');  
INSERT INTO Pieces(Code, Name) VALUES(3,'Nut');  
INSERT INTO Pieces(Code, Name) VALUES(4,'Bolt');
```

```
INSERT INTO Provides(Piece, Provider, Price) VALUES(1,'HAL',10);  
INSERT INTO Provides(Piece, Provider, Price) VALUES(1,'RBT',15);  
INSERT INTO Provides(Piece, Provider, Price) VALUES(2,'HAL',20);  
INSERT INTO Provides(Piece, Provider, Price) VALUES(2,'RBT',15);  
INSERT INTO Provides(Piece, Provider, Price) VALUES(2,'TNBC',14);  
INSERT INTO Provides(Piece, Provider, Price) VALUES(3,'RBT',50);  
INSERT INTO Provides(Piece, Provider, Price) VALUES(3,'TNBC',45);  
INSERT INTO Provides(Piece, Provider, Price) VALUES(4,'HAL',5);  
INSERT INTO Provides(Piece, Provider, Price) VALUES(4,'RBT',7);
```

## แบบฝึกหัด

-- 5.1 แสดงชื่อ วัสดุทั้งหมด

```
select Name from Pieces;
```

-- 5.2 แสดงข้อมูลทั้งหมด ของผู้จัดหา

select \* from providers;

-- 5.3 แสดงรหัสวัสดุ ค่าเฉลี่ยราคา ของการเก็บข้อมูลวัสดุ ในแต่ละรหัส

select piece, avg(price)

from Provides

group by piece;

-- 5.4 แสดงชื่อผู้จัดหาทั้งหมด ที่จัดหารหัสวัสดุรหัส 1

select Name

from Providers

where Code in (

select Provider from provides where Piece = 1

);

select Providers.Name

from Providers join Provides

on Providers.Code = Provides.Provider

where Provides.Piece = 1;

/\* Without subquery \*/

SELECT Providers.Name

```
FROM Providers INNER JOIN Provides
    ON Providers.Code = Provides.Provider
    AND Provides.Piece = 1;
```

```
/* With subquery */
```

```
SELECT Name
    FROM Providers
    WHERE Code IN
        (SELECT Provider FROM Provides WHERE Piece = 1);
```

```
-- 5.5 แสดงชื่อวัสดุ ที่มีผู้จัดหา รหัส HAL
```

```
select Name from Pieces
where Code in (
    select Piece from Provides where Provider = 'HAL'
);
```

```
select Pieces.Name
    from Pieces join Provides
    on (Pieces.code = Provides.Piece)
    where Provides.Provider = 'HAL';
```

```
/* With EXISTS subquery */
```

-- Interesting clause

SELECT Name

FROM Pieces

WHERE EXISTS

(

SELECT \* FROM Provides

WHERE Provider = 'HAL'

AND Piece = Pieces.Code

);

---

**\*\*ขอบคุณสำหรับการติดตาม และสนับสนุนผมตลอดมาครับ\*\***

โค้ชเอก

Codingthailand.com