Note:

PK - Khóa chính

NN - Không Null

BIN - nhị phân (lưu trữ dữ liệu dưới dạng chuỗi nhị phân. Không có bộ ký tự nào để phân loại và so sánh dựa trên giá trị số của byte trong giá trị.)

UN - Unsigned (các số không âm. Vì vậy nếu phạm vi từ -500 đến 500, thay vì 0 - 1000, phạm vi là giống nhau nhưng nó bắt đầu ở 0)

UQ - Tạo / xóa khóa duy nhất

ZF - Không được điền (nếu chiều dài là 5 như INT (5) thì mỗi trường được điền bằng 0 vào giá trị thứ 5. 12 = 00012, 400 = 00400, v.v.)

AI - Tăng trưởng tự động

G - Cột được tạo. tức là giá trị được tạo bởi công thức dựa trên các cột khác

1. **SELECT \*FROM** **classicmodels.customers;**

* Hiển thị toàn bộ thông tin của table “classicmodels.customers”.

1. **SELECT** **customerName, phone FROM classicmodels.customers;**

* Hiển thị thông tin các bản ghi của trường tùy chọn là “customerName, phone” của table “classicmodels.customers”.

1. **SELECT** **customernumber, contactlastname**

**FROM customers**

**WHERE contactfirstname LIKE '%A%'**

* Đối chiếu những giá trị nào trong trường “contactfirstname” của bảng có cụm từ “A” thì hiển thị hết các trường “customernumber, contactlastname” của bảng “customers” tương ứng.

1. **SELECT customernumber, contactlastname**

**FROM customers**

**WHERE** **customernumber BETWEEN 103 AND 125;**

* Đối chiếu những giá trị nào trong trường “customernumber” của bảng nằm trong khoảng 103->125 thì hiển thị hết các trường “customernumber, contactlastname” của bảng “customers” tương ứng.

1. **SELECT customernumber, contactlastname**

**FROM customers**

**WHERE** **city in (****'USA', 'France', 'Spain');**

* Đối chiếu những giá trị nào trong trường “**city**” của bảng có giá trị **'USA', 'France', 'Spain'** thì hiển thị hết các trường “customernumber, contactlastname” của bảng “customers” tương ứng.

1. **SELECT customernumber, contactlastname**

**FROM customers**

**WHERE** **country = 'USA' OR city = 'berlin';**

* Đối chiếu những giá trị nào trong trường “**country**” của bảng có giá trị **'USA'** hoặc trường **city = 'berlin'** thì hiển thị hết các trường “customernumber, contactlastname” của bảng “customers” tương ứng.

1. **SELECT** **customernumber, contactlastname**

**FROM** **customers**

**ORDER BY contactfirstname ASC ;**

* Sắp xếp các giá trị của trường ‘contactfirstname’ trong bảng ‘customers’ theo thứ tự a-z và hiển thị thông tin các trường ‘customernumber, contactlastname’ tương ứng.

1. **SELECT customernumber, contactlastname**

**FROM customers**

**ORDER BY contactfirstname DESC**

**limit 10; (top 10)**

* Sắp xếp các giá trị của trường ‘contactfirstname’ trong bảng ‘customers’ theo thứ tự z-a và hiển thị thông tin các trường ‘customernumber, contactlastname’ tương ứng và hiển thị 10 phần tử đầu tiên.

1. **SELECT count(country) AS 'Số khách hàng số ở France'**

**FROM customers**

**WHERE city = 'France' ;**

* Đếm số khách hàng có ‘city’ ở France của bảng ‘customers’.

1. **UPDATE customers SET addressLine2='Level 0' WHERE customerNumber='103';**

* Tìm trong table “customers” bản ghi thuộc trường “customerNumber” có giá trị là “103”, sau đó update thông tin của trường “addressLine2” của bản ghi đó lên “Level 0”.

1. **DELETE FROM `classicmodels`.`customers` WHERE `customerNumber`='103';**

* Xóa 1 record

1. **ALTER TABLE table\_name**

**DROP COLUMN column\_name;**

* Xóa 1 cột trong bảng

1. **INSERT INTO** classicmodels.customers(customerNumber,customerName,contactLastName,contactFirstName,phone,addressLine1,city,country) value('105', 'LyNa222','Be','Na',0905315397,'TamKy','Quang Nam','Viet Nam');

* Thêm 1 record

1. **SELECT \*FROM customers WHERE customerName='LeMinhHai';**

* Tìm bản ghi chứa từ cần tìm của 1 cột nào đó trong bảng

1. **SELECT \*FROM customers WHERE customerName like '%MinhHai%';**

- Tìm các bản ghi chứa cụm từ “MinhHai” của cột “customerName” trong table “customers”.

1. **SELECT \*FROM customers WHERE salesRepEmployeeNumber in (1166);**

Or **SELECT \*FROM customers WHERE customerName in (‘MinhHai’, ‘LyNa’);**

* Hiển thị các bản ghi chứa thông tin là “Minh Hải” của trường “customerName” trong table “customer”.

1. **foreign key (acc\_category\_id) references furama.accomodation\_category(acc\_category\_id)**

* **Tạo khóa ngoại liên kết giữa các bảng.**

1. **CREATE DATABASE name;**

* Tạo cơ sở dữ liệu

1. **CREATE TABLE name;**

* Tạo 1 bảng

**Vd1: CREATE TABLE contacts**

**(contact\_id int(11) not null auto\_increment,**

**last\_name varchar(30) not null,**

**first\_name varchar(25),**

**birthday date,**

**constraint contacts\_pk primary key(contact\_id)**

**);**

**Vd2: CREATE TABLE suppliers**

**(supplier\_id int(11) not null auto\_increment,**

**supplier\_name varchar(50) not null,**

**account\_rep varchar(30) not null default 'TBD',**

**constraint suppliers\_pk primary key (supplier\_id)**

**);**

* Tạo bảng “contacts” với thông số các trường

1. **DROP TABLE name1, name2;**

* Xóa 1 hoặc nhiều bảng.

1. **ALTER TABLE contacts**

**ADD phone int(11) after first\_name,**

**ADD address varchar(50) after phone;**

* Thêm trường “phone”, “address” cho bảng.

1. **ALTER TABLE contacts**

**MODIFY last\_name varchar(50) not null,**

**MODIFY first\_name varchar(30);**

* Sửa thông số các trường “last\_name”, “first\_name” trong bảng.

1. **ALTER TABLE contacts**

**DROP COLUMN first\_name;**

* Xóa trường first\_name trong bảng.

1. **ALTER TABLE contacts CHANGE COLUMN phone phoneNumber int(11);**

* Thay đổi tên cột “phone” thành “phoneNumber” trong bảng contacts

1. **ALTER TABLE contacts RENAME TO people;**

* Đổi tên bảng “contacts” thành “people”

1. **ALTER TABLE table\_name  
    ADD PRIMARY KEY(primary\_key\_column);**

* Tạo thêm khóa chính sau khi bảng đã được tạo.

1. **SELECT** **productCode, productName, quantityInStock, buyPrice FROM** **carsalemanagement.products**

**WHERE** **buyPrice > 29000 AND quantityInStock > 100;**

* Hiển thị thông tin các trường “productCode, productName, quantityInStock, buyPrice” từ bảng “carsalemanagement.products” với điều kiện các thực thể “buyPrice > 29000 and quantityInStock > 100”

1. **SELECT productCode,productName,buyPrice,textDescription**

**FROM carsalemanagement.products**

**INNER JOIN carsalemanagement.productlines**

**ON products.****productLine = productlines.productLine**

**WHERE buyPrice>24000 AND buyPrice<55000;**

* Tạo 1 bảng mới và hiển thị các trường “productCode,productName,buyPrice,textDescription” trong 2 bảng “carsalemanagement.products, carsalemanagement.productlines”, tham chiếu so sánh giá trị khóa ngoại “productLine” trong 2 bảng và trả về giá trị.

1. **SELECT** **productCode, productName, buyPrice, quantityInStock, productVendor, productLine**

**from** **carsalemanagement.products**

**where** **productVendor='mazda' or productLine='Motorcycles';**

* Hiển thị thông tin các trường “productCode, productName, buyPrice, quantityInStock, productVendor, productLine” từ bảng “carsalemanagement.products” với điều kiện “productVendor='mazda' or productLine='Motorcycles’.

1. **SELECT classicmodels.customers.customerNumber, customers.customerName, customers.phone, customers.city, payments.paymentDate, payments.amount**

**FROM customers**

**LEFT JOIN** **payments**

**ON customers.customerNumber = payments.customerNumber**

**WHERE amount is null;**

* Hiển thị thông tin các trường “classicmodels.customers.customerNumber, customers.customerName, customers.phone, customers.city, payments.paymentDate, payments.amount” từ 2 bảng “customers, payments” bằng cách gộp theo “LEFT JOIN”.
* Phần “ON” tham chiếu 2 trường “customerNumber” của 2 bảng.
* Hiển thị tất cả các trường thông tin của bảng “customers”, còn các trường thông tin của bảng “payments” phụ thuộc vào tham chiếu của trường “customerNumber”
* Điều kiện where đó để hiển thị các giá trị “amount is null ” trong bảng.