

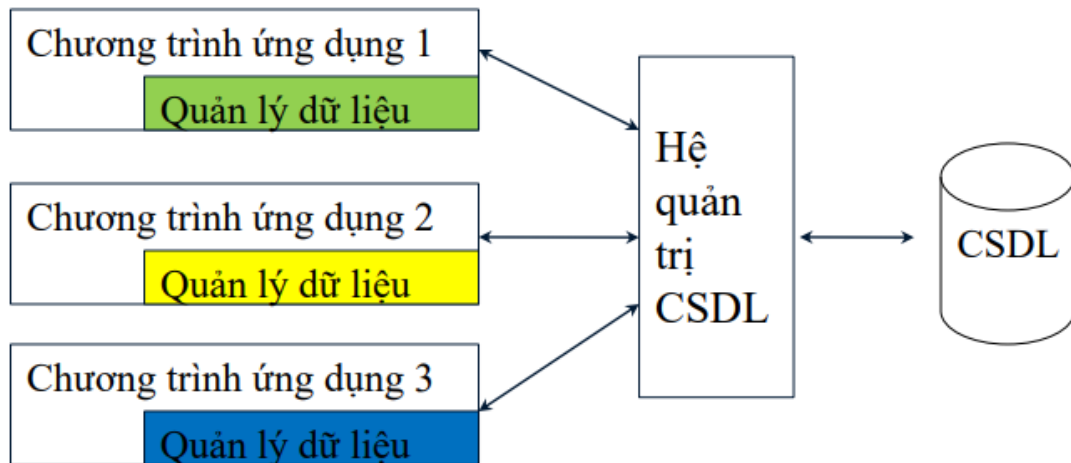
MỤC LỤC

PHẦN 1: TỔNG QUAN CƠ SỞ DỮ LIỆU	2
BÀI 1: CƠ SỞ DỮ LIỆU	2
BÀI 2: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU	3
PHẦN 2: CÁC KHÁI NIỆM CƠ BẢN MYSQL	4
BÀI 1: GIỚI THIỆU HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL	4
BÀI 2: THAO TÁC CƠ BẢN TRÊN CƠ SỞ DỮ LIỆU	5
BÀI 3: CÁC KIỂU DỮ LIỆU	6
PHẦN 3: THAO TÁC TRÊN BẢNG	9
BÀI 1: TẠO VÀ XÓA BẢNG.....	9
BÀI 2: THAY ĐỔI CẤU TRÚC BẢNG	10
BÀI 4: NGÔN NGỮ THAO TÁC DỮ LIỆU.....	12
PHẦN 4: TRUY VẤN (QUERY)	14
BÀI 1: ĐỊNH NGHĨA	14
BÀI 2: CÁC TOÁN TỬ TRUY VẤN	15
BÀI 3: TRUY VẤN ĐƠN GIẢN	16
BÀI 4: GOM NHÓM (GROUP BY)	21

PHẦN 1: TỔNG QUAN CƠ SỞ DỮ LIỆU

BÀI 1: CƠ SỞ DỮ LIỆU

- Cơ sở dữ liệu (Database) là một tập hợp các dữ liệu liên quan đến một chủ đề cụ thể và được tổ chức theo một cấu trúc nhất định để lưu trữ, truy xuất và quản lý dữ liệu một cách hiệu quả.
- Cơ sở dữ liệu thường bao gồm các bảng, các cột trong bảng và các quan hệ giữa các bảng, tạo thành một mô hình dữ liệu đồng nhất. Mỗi bảng đại diện cho một loại thực thể (entity) trong thế giới thực hoặc ảo, trong đó mỗi dòng (row) đại diện cho một trường hợp của thực thể đó và các cột (column) đại diện cho các thuộc tính (attribute) của thực thể.
- Cơ sở dữ liệu là một công cụ hữu ích để lưu trữ, truy xuất và quản lý các dữ liệu, cho phép các tổ chức và cá nhân lưu trữ, truy xuất và quản lý thông tin một cách hiệu quả. Các ứng dụng của cơ sở dữ liệu rất đa dạng, bao gồm quản lý thông tin khách hàng, quản lý hàng tồn kho, quản lý bán hàng, quản lý dữ liệu nhân sự,...



BÀI 2: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

- Hệ quản trị cơ sở dữ liệu (Database Management System - DBMS) là một phần mềm được sử dụng để quản lý và tổ chức cơ sở dữ liệu. Nó cho phép người dùng tạo, sửa đổi, truy xuất và xóa các dữ liệu trong cơ sở dữ liệu một cách hiệu quả và bảo mật.
- Hệ quản trị cơ sở dữ liệu cung cấp các tính năng như:
 - Tạo cơ sở dữ liệu mới
 - Tạo các bảng và các cột trong bảng
 - Thêm, sửa và xóa các bản ghi trong bảng
 - Tìm kiếm, truy vấn và lọc dữ liệu từ bảng
 - Định nghĩa các quan hệ giữa các bảng
 - Thực hiện sao lưu và phục hồi cơ sở dữ liệu
 - Cung cấp bảo mật dữ liệu, bao gồm phân quyền truy cập dữ liệu cho các người dùng khác nhau.
- Một số hệ quản trị cơ sở dữ liệu phổ biến là MySQL, Oracle, Microsoft SQL Server, PostgreSQL, MongoDB, ...

PHẦN 2: CÁC KHÁI NIỆM CƠ BẢN MYSQL

BÀI 1: GIỚI THIỆU HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

- MySQL là hệ thống quản trị cơ sở dữ liệu mã nguồn mở được sử dụng rộng rãi trên toàn thế giới. Nó cho phép lưu trữ và truy xuất dữ liệu hiệu quả, đồng thời cung cấp các tính năng an toàn và bảo mật.
- MySQL là một hệ quản trị cơ sở dữ liệu phổ biến và miễn phí mã nguồn mở.
- MySQL được sử dụng rộng rãi cho các ứng dụng web, trong đó nó cung cấp cơ sở dữ liệu để lưu trữ và truy xuất các thông tin được sử dụng bởi các ứng dụng web như các trang web thương mại điện tử, các diễn đàn và các ứng dụng web khác.
- MySQL hỗ trợ nhiều loại cơ sở dữ liệu như cơ sở dữ liệu quan hệ, dữ liệu không quan hệ, cơ sở dữ liệu bộ nhớ và cơ sở dữ liệu dạng đồ thị.

BÀI 2: THAO TÁC CƠ BẢN TRÊN CƠ SỞ DỮ LIỆU

- Tạo cơ sở dữ liệu: Sử dụng lệnh `CREATE DATABASE` để tạo cơ sở dữ liệu mới.
- Ví dụ: `CREATE DATABASE mydatabase;`
- Xóa cơ sở dữ liệu: Sử dụng lệnh `DROP DATABASE` để xóa cơ sở dữ liệu.
- Ví dụ: `DROP DATABASE mydatabase;`
- Xem các cơ sở dữ liệu có trong database ta dùng: `SHOW DATABASE`
- Sử dụng cơ sở dữ liệu ta dùng: `USE DATABASE_NAME`
- Ví dụ: `USE QuanLiHoaDon;`

BÀI 3: CÁC KIỂU DỮ LIỆU

- Mysql cung cấp các kiểu dữ liệu:
 - Kiểu số nguyên:

Tên kiểu dữ liệu	Bộ nhớ (bytes)	Vùng giá trị có dấu	Vùng giá trị không dấu
Tinyint	1	-128 đến 127	0 đến 255
Smallint	2	-32768 đến 32767	0 đến 65535
Mediumint	3	-8388608 đến 8388607	0 đến 16777215
Int	4	-2147483648 đến 2147483647	0 đến 4294967295
Bigint	8	-9223372036854775808 đến 9223372036854775807	0 đến 18446744073709550 615

- Kiểu số thực:

Tên kiểu dữ liệu	Bộ nhớ (bytes)	Vùng giá trị có dấu	Vùng giá trị không dấu
Float (M,D)	4	-128 đến 127	0 đến 255
Double(M,D)	8	-1.7976931348623157E+308 đến -2.2250738585072014E-308	0 đến 65535
Decimal(M,D)	M + D	-8388608 đến 8388607	số 0 và khoảng giá trị từ 2.2250738585072014E-308 đến 1.7976931348623157E+308

- Kiểu chuỗi: CHAR, VARCHAR, TEXT, BLOB.

Tên kiểu dữ liệu	Số ký tự chứa tối đa
Char	255
Varchar	255
Tinytext	255
Tinyblob	255
Text	65535
Blob	65535
Mediumtext	16777215
Mediumblob	16777215
Longtext	4294967295
Longblob	4294967295

- Kiểu ngày giờ:

Tên kiểu dữ liệu	Cách sử dụng	Định dạng kiểu dữ liệu	Vùng chứa
Date	Lưu trữ thông tin ngày trong cột bảng	YYYY-MM-DD (năm, tháng, ngày)	từ '1000-01-01' đến '9999-12-31'
Time	Lưu trữ giờ	HH:MM:SS (giờ, phút, giây)	từ '-838:59:59' đến '838:59:59'
Datetime	Lưu trữ cả ngày và giờ (*)	YYYY-MM-DD HH:MM:SS (năm, tháng, ngày và giờ, phút, giây)	từ '1000-01-01 00:00:00' đến '9999-12-31 23:59:59'
Timestamp	Lưu trữ cả giá trị ngày và giờ. (*)	YYYY-MM-DD HH:MM:SS (năm, tháng, ngày và giờ, phút, giây)	từ '1970-01-01 00:00:01' UTC đến '2038-01-19 03:14:07' UTC

Year	Lưu trữ các giá trị năm trong cột	YYYY (năm)	từ '1901' đến '2155'
------	-----------------------------------	------------	----------------------

- (*) Sự khác biệt chính của DATETIME và TIMESTAMP là giá trị của TIMESTAMP được chuyển đổi từ múi giờ hiện tại sang UTC trong khi lưu trữ, và chuyển ngược trở lại từ UTC sang múi giờ hiện tại trong lúc lấy ra. Còn kiểu dữ liệu DATETIME thì không có gì thay đổi.
- Kiểu Boolean: MySQL không có kiểu dữ liệu boolean (hoặc bool). Thay vào đó, nó chuyển đổi các giá trị boolean (true và false) trong cột thành kiểu dữ liệu TINYINT (1 và 0). Khi tạo một cột có kiểu dữ liệu boolean, MySQL xuất dữ liệu là 0 nếu sai (false) và 1 nếu đúng (true) .
- Kiểu ENUM: Kiểu dữ liệu liệt kê, cho phép định nghĩa một danh sách các chuỗi (có thể lên đến 65,535 chuỗi). Khi chèn vào bảng, dữ liệu phải là một trong số các giá trị nằm trong danh sách, nếu không bảng sẽ lưu trữ một chuỗi rỗng. Ví dụ: ENUM('red', 'green', 'blue') thì cột đó chỉ được lưu chuỗi có giá trị bằng 'red', 'green' hoặc 'blue'.
- Kiểu SET: kiểu dữ liệu liệt kê, tương tự như enum nhưng cho phép cột không chứa giá trị nào (chuỗi rỗng) hoặc lưu trữ nhiều giá trị trong các giá trị trong danh sách được xác định trước, mỗi giá trị cách nhau bởi dấu ', '. Ví dụ: SET('apple', 'banana', 'orange').

PHẦN 3: THAO TÁC TRÊN BẢNG

- Lược đồ CSDL quản lý bán hàng gồm có các quan hệ sau:
- **KHACHHANG** (MAKH, HOTEN, DCHI, SODT, NGSINH, DOANH SO, NGDK)
- **NHANVIEN** (MANV, HOTEN, NGVL, SODT)
- **SANPHAM** (MASP, TENS P, DVT, NUOCSX, GIA)
- **HOADON** (SOHD, NGHD, MAKH, MANV, TRIGIA)
- **CTHD** (SOHD, MASP, SL)

BÀI 1: TẠO VÀ XÓA BẢNG

Tạo bảng

```
create table ten_bang (  
    tencot1 kieu_dulieu1 rang_buoc1,  
    tencot2 kieu_dulieu2 rang_buoc2,  
    ...  
)
```

Ví dụ: Tạo bảng kháchhang

```
create table kháchhang(  
    makh char (4) primary key,  
    hoten varchar (50),  
    dchi varchar(50),  
    sodt varchar(20),  
    ngsinh date,  
    doanhso decimal (15,2),  
    ngdk date  
);
```

Xóa bảng

```
drop table ten_bang [from ten_database];
```

Ví dụ: Xóa bảng kháchhang

```
drop table kháchhang
```

BÀI 2: THAY ĐỔI CẤU TRÚC BẢNG

- Thay đổi tên bảng

```
alter table ten_bang rename ten_bang_moi;
```

Ví dụ: Thay đổi tên bảng sanpham thành sp

```
alter table sanpham rename sp
```

- Thay đổi tên cột

```
alter table ten_bang change ten_cot_cu ten_cot_moi kieu_du_lieu_cu;
```

Ví dụ: Thay đổi tên cột dvt (thuộc bảng sanpham) thành donvitinh

```
alter table sanpham change dvt donvitinh varchar
```

- Thay đổi kiểu dữ liệu

```
alter table ten_bang change ten_cot ten_cot kieu_du_lieu_moi;
```

Ví dụ: Thay đổi kiểu dữ liệu của cột từ varchar thành char

```
alter table sanpham change dvt dvt char(5)
```

- Thêm khóa chính

```
alter table ten_bang add primary key ten_cot;
```

Ví dụ: Thêm khóa chính cho bảng nhanvien

```
alter table nhanvien add primary key manv
```

- Xóa khóa chính

```
alter table ten_bang drop primary key;
```

Ví dụ: Xóa khóa chính của bảng nhanvien

```
alter table nhanvien drop primary key
```

- Thêm cột

```
alter table ten_bang add ten_cot kieu_du_lieu;
```

Ví dụ: Thêm cột nghd cho bảng hoadon

```
alter table hoadon add nghd date
```

- Xóa cột

```
alter table ten_bang drop ten_cot;
```

Ví dụ: Xóa cột nghd của bảng hoadon

```
alter table hoadon drop nghd
```

BÀI 4: NGÔN NGỮ THAO TÁC DỮ LIỆU

Thêm dữ liệu (INSERT)

- Cú pháp 1: Phải điền đủ các giá trị của các cột

```
insert into ten_bang values (gia_tri_1, gia_tri_2,..., gia_tri_n)
```

Ví dụ: bảng sanpham có 5 cột thì khi viết câu lệnh insert phải điền đủ 5 giá trị tương ứng, nếu không khi chạy sẽ bị lỗi.

```
insert into sanpham values ('TV01','Tap 100 giay  
mong','quyen','Trung Quoc',2500)
```

- Cú pháp 2: Số lượng giá trị được điền trong câu lệnh phải bằng với số lượng cột được điền trong câu lệnh

```
insert into ten_bang (cot_1,...,cot_n) values (gia_tri_1,...,  
gia_tri_n)
```

Ví dụ: trong câu lệnh insert, ta muốn thêm 5 giá trị cho 5 cột trong bảng sanpham thì phải viết đủ tên 5 cột và 5 giá trị tương ứng đó, nếu không khi chạy sẽ bị lỗi.

```
insert into sanpham (masp,tensp,dvt,nuocsx,gia) values  
('TV01','Tap 100 giay mong','quyen','Trung Quoc',2500)
```

Xóa dữ liệu (DELETE)

```
delete from ten_bang [where dieu_kien]
```

Ví dụ: Viết câu lệnh xóa những sản phẩm do Việt Nam sản xuất có giá thấp hơn 25000 (có điều kiện where)

```
delete from sanpham where (gia<25000) and (nuocsx='Viet
```

- Lưu ý: câu lệnh delete nếu không có điều kiện WHERE nghĩa là xóa tất cả các hàng dữ liệu trong bảng.

Ví dụ: Viết câu lệnh xóa tất cả các hàng trong bảng sanpham (không có điều kiện where)

```
delete from sanpham
```

Sửa dữ liệu (UPDATE)

```
update ten_bang set cot_1 = gia_tri_1, cot_2 = gia_tri_2 .... [where dieu_kien]
```

Ví dụ: Tăng giá 5% đối với những sản phẩm do "Singapore" sản xuất

```
update sanpham  
set gia = gia*1.05  
where nuocsx = 'Singapore'
```

Ví dụ: Sửa lại địa chỉ của khách hàng có tên 'TRAN NGOC LINH' thành '105 VO VAN TAN, Q3, TPHCM'

```
update khachhang  
set dchi = '34 TRUONG  
DINH,Q3,TPHCM'  
where hoten = 'TRAN NGOC LINH'
```

- Lưu ý: câu lệnh update nếu không có điều kiện WHERE nghĩa là sửa tất cả.

Ví dụ: Sửa lại địa chỉ của tất cả khách hàng thành 105 VO VAN TAN, Q3, TPHCM'

```
update khachhang  
set dchi = '34 TRUONG  
DINH,Q3,TPHCM'
```

- `<!DOCTYPE html>` - Khai báo loại tài liệu HTML. Để sử dụng phiên bản HTML5 thì ta thêm thẻ này vào vị trí đầu
- Các thẻ khác như `<h1>`, `<p>`, ``, ``, `<a>` được sử dụng để định dạng và hiển thị nội dung trang web.

PHẦN 4: TRUY VẤN (QUERY)

BÀI 1: ĐỊNH NGHĨA

- Truy vấn là sử dụng các câu lệnh của ngôn ngữ SQL (Structured Query Language – ngôn ngữ truy vấn có cấu trúc) để thực hiện việc tạo (create), thêm (insert), chọn (select), xóa (delete) và sửa (Update, Alter) dữ liệu, bảng, cột, dòng...
- Cấu trúc tổng quát của câu truy vấn

```
select [distinct] * | ten_cot | ham  
from ten_bang  
[where dieu_kien]  
[group by ten_cot]  
[having dieu_kien]  
[order by ten_cot ASC | DESC]
```

- Select: Tương đương phép chiếu của ĐSQH. Liệt kê các thuộc tính muốn hiển thị trong kết quả
- From: Liệt kê các quan hệ cần thiết, các phép kết
- Where: Tương ứng với điều kiện chọn trong ĐSQH. Điều kiện liên quan tới thuộc tính, sử dụng các phép nối luận lý AND, OR, NOT, các phép toán so sánh, BETWEEN

BÀI 2: CÁC TOÁN TỬ TRUY VẤN

- Toán tử so sánh: =, >, <, >=, <=, <>
- Toán tử logic: AND, OR, NOT
- Phép toán: +, -, *, /
- BETWEEN AND
- IS NULL, IS NOT NULL
- LIKE (_ %)
- IN, NOT IN
- EXISTS , NOT EXISTS
- SOME, ALL

BÀI 3: TRUY VẤN ĐƠN GIẢN

- Lược đồ CSDL quản lý bán hàng gồm có các quan hệ sau:
- **KHACHHANG** (MAKH, HOTEN, DCHI, SODT, NGSINH, DOANHSONGDK)
- **NHANVIEN** (MANV, HOTEN, NGVL, SODT)
- **SANPHAM** (MASP, TENSPP, DVT, NUOCSX, GIA)
- **HOADON** (SOHD, NGHDS, MAKH, MANV, TRIGIA)
- **CTHD** (SOHD, MASP, SL)

Ví dụ: Chọn tất cả các cột trong bảng hoadon

```
select * from hoadon;
```

Ví dụ: Chọn 3 cột: sohd, nghd, trigia trong bảng hoadon

```
select sohd, nghd, trigia from hoadon;
```

- Chọn dữ liệu với điều kiện trích lọc dữ liệu

Ví dụ: Tìm sohd, nghd của những hóa đơn được lập trước ngày '1/1/2007'

```
select sohd, nghd  
from hoadon  
where nghd < '2007-01-01';
```

Ví dụ: Tìm sohd, nghd của những hóa đơn được lập trước ngày '1/1/2007' có trị giá từ 300000 đến 500000

```
select sohd, nghd  
from hoadon  
where nghd < '2007-01-01' and trigia >= 300000 and trigia <= 500000;
```


Ví dụ: Tìm sohd, nghd của những hóa đơn được lập trước ngày '1/1/2007' có trị giá từ 300000 đến 500000 (sử dụng between)

```
select sohd, nghd
from hoadon
where nghd < '2007-01-01' and trigia between 300000 and 500000;
```

- Sử dụng toán tử LIKE để truy vấn so sánh chuỗi tương đối.
- Ký tự % để thay thế một chuỗi ký tự bất kỳ.
- Ký tự _ để thay thế một ký tự bất kỳ.

Ví dụ: Tìm sohd, makh, trigia của hóa đơn có makh gần giống với 'KH%'

```
select sohd, mahd, trigia
from hoadon
where makh like 'KH%';
```

Ví dụ: Tìm sohd, makh, trigia của hóa đơn có makh gần giống với 'KH_'

```
select sohd, mahd, trigia
from hoadon
where makh like 'KH1_';
```

- hiển thị kết quả dưới một tên cột khác không giống tên cột có sẵn

```
select tensp as 'Ten san pham', nuocsx as 'Nuoc san xuất', dvt as 'Đơn
vi tính'
from sanpham;
```

- hiển thị kết quả dưới một tên cột khác không giống tên cột có sẵn

```
select tensp as 'Ten san pham', nuocsx as 'Nuoc san xuất', dvt as 'Đơn
vi tính'
from sanpham;
```

Distinct

- Trong một bảng, một cột có khi sẽ có nhiều hàng có giá trị giống nhau. Để chọn ra các hàng có giá trị khác nhau, ta dùng từ khóa distinct.

Ví dụ: Chọn ra các tên sản phẩm khác nhau trong bảng sản phẩm

```
select distinct tensp from sanpham
```

- Từ khóa ORDER BY thường được dùng kết hợp với lệnh SELECT để sắp xếp các kết quả trả về theo thứ tự tăng dần hoặc giảm dần (dựa trên giá trị của một cột nào đó).
 - Sử dụng ASC để sắp xếp kết quả theo thứ tự tăng dần.
 - Sử dụng DESC để sắp xếp kết quả theo thứ tự giảm dần.

Ví dụ: Sắp xếp các khách hàng theo doanh số tăng dần

```
select makh, hoten, doanhso  
from khachhang  
order by doanhso asc;
```

Ví dụ: Sắp xếp các khách hàng theo doanh số giảm dần

```
select makh, hoten, doanhso  
from khachhang  
order by doanhso desc;
```

- is null, is not null

```
select *  
from hoadon  
where makh is NULL
```

```
select *  
from hoadon  
where makh is NOT NULL
```

- in, not in

```
select *  
from sanpham  
where masp in ('TV01','TV02');
```

```
select *  
from sanpham  
where masp not in ('TV01','TV02');
```

Inner join

- INNER JOIN được sử dụng để kết hợp các hàng từ hai bảng lại thành một bảng tổng hợp.
- Các hàng bên trong cái bảng thứ nhất sẽ được bắt cặp lần lượt với các hàng bên trong bảng thứ hai
- Những hàng mà dữ liệu trên thỏa điều kiện (condition) thì sẽ được giữ lại, còn những hàng không thỏa điều kiện thì sẽ bị loại bỏ.

Ví dụ: Chọn ra tên sản phẩm tương ứng với mã sản phẩm trong bảng cthd

```
select cthd.sohd, cthd.masp, sanpham.tensp  
from cthd inner join sanpham  
on cthd.masp = sanpham.masp;
```

Ví dụ: Chọn ra tên nhân viên tương ứng với sohđ do nhân viên đó lập

```
select hoadon.sohđ, hoadon.ngghđ, hoadon.manv,  
nhanvien.hoten  
from hoadon inner join nhanvien
```

Left Join

- Trả về các dòng dữ liệu của bảng bên trái và các dòng được kết nối bên bảng bên phải. Nếu các dòng bên trái không có kết nối với cột nào bên phải, thì các cột của bảng bên phải sẽ là null.

Ví dụ: Chọn ra sohd tương ứng với tên khách hàng đã mua hóa đơn đó

```
select hoadon.sohd, hoadon.nghd, hoadon.makh,  
khachhang.hoten  
from hoadon left join khachhang
```

Right Join

- Trả về các dòng dữ liệu của bảng bên phải và các dòng được kết nối bên bảng bên trái. Nếu các dòng bên phải không có kết nối với cột nào bên trái, thì các cột của bảng bên trái sẽ là null.

Ví dụ: Chọn ra tên khách hàng tương ứng với sohd đã mua

```
select khachhang.makh, khachhang.hoten, hoadon.sohd,  
hoadon.nghd  
from khachhang right join hoadon
```

BÀI 4: GOM NHÓM (GROUP BY)

- GROUP BY được dùng để nhóm những hàng có cùng giá trị dựa trên một cột nào đó lại với nhau.
- Mệnh đề GROUP BY chỉ ra các thuộc tính gom nhóm, các thuộc tính này phải xuất hiện trong mệnh đề select.
- Các hàm tính toán sử dụng với mệnh đề group by
 - COUNT: Đếm số bộ dữ liệu của thuộc tính
 - MIN: Tính giá trị nhỏ nhất
 - MAX: Tính giá trị lớn nhất
 - AVG: Tính giá trị trung bình
 - SUM: Tính tổng giá trị các bộ dữ liệu

Ví dụ: Hiển thị hóa đơn có trị giá thấp nhất

```
select min(trigia) as hd_thap_nhat
from hoadon;
```

Ví dụ: Hiển thị hóa đơn có trị giá cao nhất trong năm 2007

```
select max(trigia) as hd_cao_nhat_2007
from hoadon;
where nqhd like '2007%';
```

Ví dụ: Hiển thị tổng trị giá và trung bình trị giá của các hóa đơn trong năm 2006

```
select sum(trigia) as tong, avg(trigia) as
      trung_binh
from hoadon;
```

Ví dụ: Hiển thị tổng số hóa đơn của từng khách hàng đã mua hàng (hiển thị makh, hoten của khách hàng)

```
select hd.makh, kh.hoten, count(sohd) as
      tong_so_hoa_don
from hoadon hd, khachhang kh
where hd.makh = kh.makh
```