

# Mục Lục

Bài 1	CÁC KHÁI NIỆM CƠ BẢN.....	1
I.	Các khái niệm .....	1
1.	Cơ sở dữ liệu.....	1
2.	Hệ quản trị cơ sở dữ liệu.....	1
II.	Thiết kế cơ sở dữ liệu .....	1
1.	Cơ sở dữ liệu quan hệ .....	1
2.	Quy trình thiết kế cơ sở dữ liệu .....	2
3.	Các bước chuẩn hoá.....	4
Bài 2	TỔNG QUAN VỀ CSDL MYSQL.....	6
I.	Tổng quan về MySQL .....	6
1.	Mô hình Client/Server .....	6
2.	Quy trình cài đặt (trên Windows) .....	6
3.	Khởi động & tắt dịch vụ MySQL .....	13
4.	Thao tác tại client.....	13
5.	Thay đổi password cho tài khoản root .....	14
II.	Các thao tác cơ bản trên CSDL .....	14
1.	Xem các cơ sở dữ liệu.....	14
2.	Tạo cơ sở dữ liệu mới .....	15
3.	Xoá cơ sở dữ liệu .....	15
4.	Thêm tài khoản người dùng (user).....	15
Bài 3	BẢNG - TABLE.....	16
I.	Định nghĩa .....	16
II.	Các kiểu dữ liệu.....	16
1.	Kiểu dữ số (numeric) .....	16
2.	Kiểu chuỗi kí tự (string).....	16
3.	Kiểu hỗn hợp (Miscellaneous).....	17
4.	Kiểu ngày giờ.....	17
5.	Các từ khoá định nghĩa cột (modifiers) .....	18
III.	Các thao tác trên bảng.....	18
1.	Tạo bảng .....	18
2.	Xem thông tin của CSDL, Bảng .....	19
3.	Xoá bảng .....	19
4.	Thay đổi cấu trúc bảng.....	19
5.	Chèn dữ liệu vào bảng .....	19

Bài 4	THAO TÁC TRÊN CSDL .....	21
I.	Import dữ liệu .....	21
1.	Import từ file text .....	21
2.	Import từ file sql .....	22
3.	Import từ file Access.....	23
II.	Export dữ liệu .....	25
1.	Chương trình mysqldump .....	25
2.	Lệnh Select ... Into Outfile.....	25
3.	Chương trình mysql client .....	25
Bài 5	TRUY VẤN – QUERY .....	26
I.	Định nghĩa .....	26
1.	Định nghĩa.....	26
2.	Các từ khoá .....	26
II.	Các thao tác truy vấn dữ liệu .....	26
1.	Truy vấn chọn dữ liệu .....	26
2.	Truy vấn thống kê dữ liệu .....	27
3.	Truy vấn lồng .....	28
4.	Mệnh đề Join.....	28
5.	Truy vấn chèn dữ liệu .....	28
6.	Truy vấn xoá dữ liệu .....	28
7.	Truy vấn cập nhật dữ liệu .....	28
III.	Toán tử & hàm .....	28
1.	Toán tử .....	28
2.	Hàm toán học .....	29
3.	Hàm điều kiện .....	29
4.	Hàm logic .....	29
5.	Hàm chuỗi.....	30
6.	Hàm thời gian .....	30
Bài 6	FUNCTION – PROCEDURE – TRIGGER.....	33
I.	Function .....	33
1.	Cấu trúc tạo Function.....	33
2.	Sử dụng Function .....	33
II.	Procedure .....	34
1.	Cấu trúc tạo Procedure.....	34
2.	Sử dụng Procedure .....	34
III.	Trigger .....	34

Bài 7	BẢO MẬT VÀ QUẢN TRỊ .....	36
I.	Bảo mật CSDL .....	36
1.	Phương thức bảo mật trong MySQL.....	36
2.	Tạo tài khoản người dùng .....	36
3.	Xoá tài khoản người dùng.....	36
4.	Cấp quyền & xoá quyền cho tài khoản người dùng.....	36
II.	Quản trị.....	37
1.	Backup (sao lưu) dữ liệu.....	37
2.	Restore (phục hồi) dữ liệu.....	37
3.	Các hàm trên hệ thống của MySQL.....	37
4.	Sử dụng mysqladmin .....	37
Bài 8	SỬ DỤNG CÔNG CỤ GIAO DIỆN ĐỒ HOẠ - MySQL GUI TOOLS .....	39
I.	Giới thiệu:.....	39
II.	MySQL Administrator.....	39
III.	MySQL Query Browser .....	40
<b>Bài Tập</b>	.....	42

# Bài 1 CÁC KHÁI NIỆM CƠ BẢN

## I. Các khái niệm

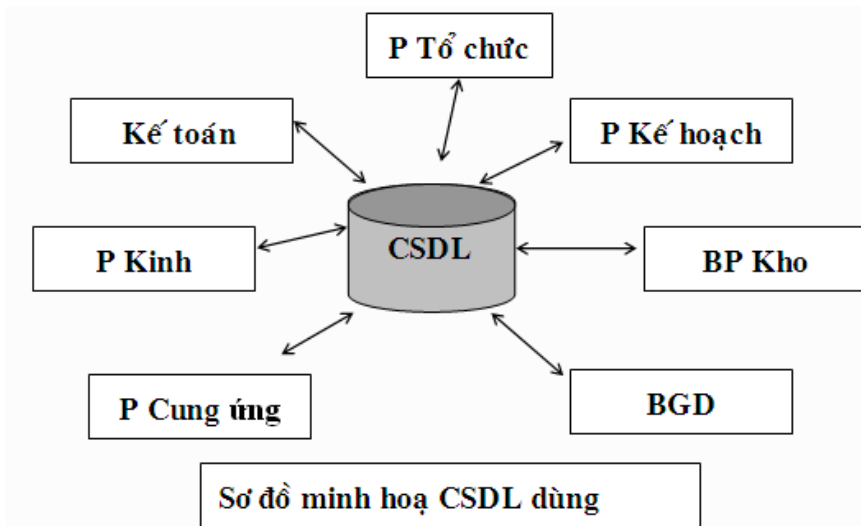
### 1. Cơ sở dữ liệu

*Cơ sở dữ liệu* (CSDL) hay còn gọi là *database* là tập hợp dữ liệu được tổ chức một cách có cấu trúc để phục vụ cho nhiều mục đích khác của người dùng.

**Vd:** Công ty có tập tin lưu trữ **danh sách nhân viên** trên máy tính, cùng lúc ban giám đốc cần xem xét các nhân viên để khen thưởng – phòng tài vụ lại cần lập bảng lương tháng cho các nhân viên.

Như vậy **danh sách nhân viên** được cả ban giám đốc và phòng tài vụ khai thác cùng một lúc, dĩ nhiên thông tin về nhân viên phải **nhất quán** nghĩa là dù ở đâu - ở ban giám đốc hay ở phòng tài vụ - thông tin ấy là phải như nhau.

Như vậy, để việc quản trị một công ty có sự nhất quán ta cần phải tổ chức dữ liệu, thông tin của công ty đó một cách đồng nhất - nghĩa là phải tổ chức CSDL cho công ty.



### 2. Hệ quản trị cơ sở dữ liệu

*Hệ quản trị CSDL* là các phần mềm cung cấp các công cụ để xây dựng CSDL, và các thao tác trên các CSDL đó. Ví dụ như *Foxpro, Access, Microsoft SQL Server, MySQL.....*

## II. Thiết kế cơ sở dữ liệu

### 1. Cơ sở dữ liệu quan hệ

#### a. Định nghĩa

Một cơ sở dữ liệu quan hệ là một cơ sở dữ liệu được tạo nên bởi các bảng. Mỗi bảng bao gồm nhiều cột, và các bảng có mối quan hệ với nhau dựa vào những giá trị khoá.

Vd : một cơ sở dữ liệu về bán hàng sẽ có một bảng là **Đơn Đặt Hàng** bao gồm các thông tin như *Mã số đơn đặt hàng, ngày đặt hàng, ngày giao hàng,...* Và một bảng khác là **Khách Hàng** bao gồm các thông tin như *họ tên, địa chỉ,...* Hai bảng này sẽ có liên quan với nhau, vì không thể một đơn đặt hàng mà không có khách hàng.

### b. Thực thể & Thuộc tính

*Thực thể* là một đối tượng cụ thể hay trừu tượng trong thế giới thực và được chuyển đổi vào trong CSDL thành một bảng. Trong một CSDL ta phải quản lý nhiều đối tượng dữ liệu, mỗi đối tượng đó được chuyển đổi từ các thực thể trong thế giới thực vào CSDL.

Vd : Trong một trường học có các thực thể: *giáo viên, học sinh, môn học...* Trong một thư viện có các thực thể: *sách, loại sách, nhà xuất bản...* Trong một công ty buôn bán có các thực thể: *mặt hàng* (cụ thể), *công nợ* (trừu tượng). Mỗi một thực thể như vậy sẽ là một bảng trong CSDL.

Mỗi thực thể có các tính chất riêng gọi là các **thuộc tính**.

Vd : thực thể *học sinh* có các thuộc tính như : *họ tên, ngày sinh, quê quán...*

Mỗi một thuộc tính sẽ được thể hiện thành một cột tương ứng trong bảng thể hiện thực thể đó.

### c. Khoá

Để phân biệt các đối tượng (thành phần) trong cùng một thực thể, ta sẽ chọn ra những thuộc tính dùng để phân biệt chúng, những thuộc tính đó gọi là **khóa**.

Ta có thể chọn ra trong các thuộc tính của thực thể một hoặc nhiều thuộc tính làm khoá hoặc cũng có thể đặt ra một thuộc tính mới (thường được gọi là Mã) để làm khoá cho thực thể đó.

Vd : Để phân biệt giữa các sinh viên trong cùng một thực thể sinh viên, ta đặt ra thuộc tính MaSV, và mỗi sinh viên có một mã số duy nhất để phân biệt với các sinh viên khác. Do đó, MaSV là khóa của thực thể SinhVien.

### d. Mối quan hệ

Mối quan hệ (hay còn gọi là ràng buộc) dùng để thể hiện sự liên quan dữ liệu giữa các bảng với nhau.

Có 3 loại quan hệ (sẽ được trình bày chi tiết ở phần sau):

- Quan hệ 1 – 1
- Quan hệ 1 – n
- Quan hệ n – n

## 2. Quy trình thiết kế cơ sở dữ liệu

### Bước 1: Xác định mục đích của CSDL

Xác định được mục đích của CSDL, sẽ giúp chúng ta quyết định đưa những thông tin nào vào CSDL

Ví dụ: Một công ty cần quản lý những thông tin về buôn bán của công ty. Như vậy, các thông tin cần lưu trữ là đơn đặt hàng, mặt hàng, khách hàng, nhân viên bán hàng.

### Bước 2: xác định các bảng (table) cần thiết.

Khi đã xác định được mục đích của CSDL, chúng ta có thể chia thông tin theo các chủ đề phân biệt. Mỗi chủ đề là một bảng (table) trong CSDL.

Ví dụ: với những thông tin của công ty trên chúng ta có thể tạo các bảng như sau: HÓA ĐƠN, CHI TIẾT HÓA ĐƠN, HÀNG HÓA, KHÁCH HÀNG, NHÂN VIÊN BÁN HÀNG...

Bước 3: Xác định các field (hay còn gọi là trường hoặc field) cho mỗi bảng.

Khi xác định các field cho bảng ta cần chú ý một số điều: Không nên tạo field chứa thông tin có thể rút ra từ những field khác, hoặc chứa kết quả có thể tính toán được. Nên tách thông tin ra phần nhỏ nhất; Các field phải phủ được các thông tin cần thiết.

Xác định khóa chính trong bảng (Primary Key): Một field là khóa chính khi mà giá trị của nó trong bảng là duy nhất, không trùng lặp. Trong một bảng có thể có hai field (hoặc hơn) kết hợp lại tạo thành một khóa chính, giá trị hợp của các field này trong bảng là duy nhất.

Field khóa chính phải có kích thước nhỏ, để tốc độ truy xuất CSDL được nhanh.

Field khóa chính không chấp nhận giá trị rỗng.

Ví dụ: trong bảng HÓA ĐƠN field Mã hóa đơn là field khóa chính, vì mỗi hóa đơn có một mã riêng.

Bước 4: Xác định mối quan hệ giữa các bảng

Chúng ta lưu trữ các dữ liệu có quan hệ với nhau trên nhiều bảng riêng lẻ khác nhau. Vì vậy chúng ta cần xác định mối quan hệ (Relationship) giữa các bảng. Để đặt quan hệ giữa hai bảng: bảng A và bảng B, chúng ta thêm khóa chính của một bảng đến bảng còn lại, vì vậy khóa này xuất hiện trong cả hai bảng. Nhưng chúng ta phải xác định sử dụng khóa chính của bảng nào? Để xác định mối quan hệ cho đúng, chúng ta phải hiểu rõ các loại quan hệ của chúng. Có 3 loại quan hệ giữa hai bảng.

- Quan hệ 1-nhiều (One – to – Many): Đây là mối quan hệ phổ biến trong CSDL quan hệ. Trong quan hệ này một mẫu tin trong bảng A có thể có quan hệ với nhiều mẫu tin trong bảng B.

Ví dụ: Trong một cơ quan, mỗi một tỉnh thành (nơi sinh của nhân viên) có nhiều nhân viên. Nhưng một nhân viên chỉ thuộc một tỉnh thành (nơi sinh)

- Quan hệ nhiều - nhiều (Many-to-Many): Mỗi mẫu tin trong bảng A có quan hệ với nhiều mẫu tin trong bảng B, và ngược lại một mẫu tin trong bảng B, cũng có quan hệ với nhiều mẫu tin trong bảng A. Quan hệ này khó thể hiện trong thiết kế. Trong trường hợp này ta phải tạo ra thêm một bảng trung gian C. Và chúng ta sẽ tạo mối quan hệ 1-nhiều giữa A và C, và mối quan hệ 1-nhiều giữa B và C.

- Quan hệ 1-1 (One-to-One): Một mẫu tin trong bảng A chỉ có quan hệ với một mẫu tin trong bảng B. Và trong một mẫu tin trong bảng B chỉ có một mẫu tin trong bảng A. Đây là trường hợp bất thường trong quan hệ, bởi vì hầu như thông tin quan hệ theo kiểu này có thể sẽ được lưu trữ trên cùng một bảng.

Trong trường hợp tạo quan hệ 1-1, trước hết bạn xem có thể đưa các dữ liệu đó vào cùng một bảng được hay không. Nếu vì một lý do nào đó mà không thể làm được điều đó, sau đây là một vài cách đặt quan hệ:

Nếu hai bảng có cùng chủ đề, ta lập mỗi quan hệ bằng cách sử dụng khóa chính cho cả hai bảng.

Nếu hai bảng có chủ đề khác nhau với khóa chính khác nhau, ta chọn một trong hai bảng và đặt field dùng làm khóa chính của nó trong bảng còn lại như là khóa ngoại.

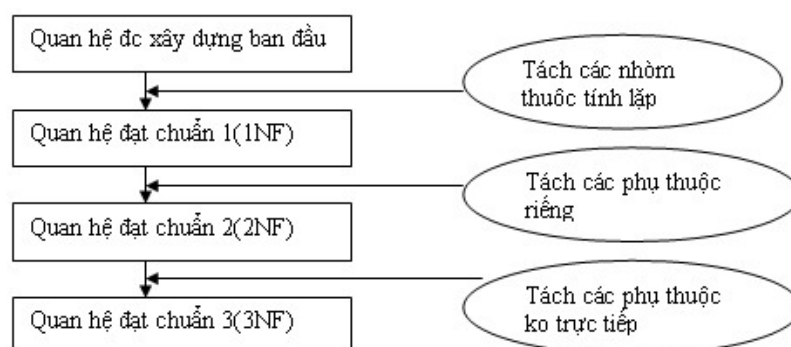
### Bước 5: Tinh chế lại thiết kế

Sau khi ta tạo các bảng, các field trong từng bảng, thiết lập quan hệ giữa các bảng, chúng ta cần xem xét lại một số vấn đề sau:

- Nhập một số mẫu tin để kiểm tra sai sót như có thiếu sót hoặc dư thừa hoặc trùng lặp thông tin trong CSDL không?
- Xem có thiếu field nào không
- Đã chọn khóa chính cho mỗi bảng chưa
- Kiểm tra nếu có trùng lặp thông tin trong một bảng không. Nếu có, và có thể chia bảng đó ra thành hai bảng với quan hệ 1- nhiều.

Có bảng nào nhiều field, mà ít mẫu tin hay không và có field nào bỏ trống trong một số mẫu tin hay không. Nếu có hãy thiết kế lại...

### 3. Các bước chuẩn hoá



Để thực hiện tốt việc tinh chế lại thiết kế, ta sẽ dựa vào 3 nguyên tắc chuẩn hoá sau.

- Phương thức chuẩn hoá 1 (1NF – First Normal Form)

Quan hệ là 1NF nếu ko chứa các thuộc tính lặp, các thuộc tính phải là đơn, nghĩa là giá trị của các ô là giao của hàng và cột phải có giá trị đơn, như vậy, mọi quan hệ đều là 1NF. Nếu bảng dữ liệu chứa các thuộc tính lặp thì ko phải quan hệ, để chuyển bảng dữ liệu có lặp thành quan hệ, có thể tách các thuộc tính lặp thành một hoặc nhiều bảng khác và nếu cần thiết thì tăng cường khóa cho các bảng mới này. Tiếp tục xem xét các bảng mới để đảm bảo sao cho các bảng này cũng là quan hệ, tức là đạt chuẩn 1.

- Phương thức chuẩn hoá 2 (2NF – Second Normal Form)

Một quan hệ R là dạng chuẩn 2(2NF) nếu nó là 1NF và các phụ thuộc hàm giữa các thuộc tính ngoài khóa và khóa đều là các phụ thuộc hàm sơ đẳng, nói cách khác, mọi thuộc tính ngoài khóa đều ko có phụ thuộc hàm vào bộ phận của khóa. Nếu quan hệ R chứa những thuộc tính có phụ thuộc hàm vào một bộ phận của khóa thì cần tách các nhóm thuộc tính phụ thuộc vào bộ phận của khóa và bổ sung thêm cho các nhóm này một phần khóa mà chúng có phụ thuộc hàm, để thành quan hệ. Nhóm còn lại tạo thành một quan hệ với khóa như cũ. Các quan hệ được tạo lập đều là 2NF.

- Phương thức chuẩn hoá 3 (3NF – Third Normal Form)

Một quan hệ R là dạng chuẩn 3 (3NF) nếu nó là 2NF và các phụ thuộc hàm giữa các thuộc tính khóa ngoài và khóa đều là các phụ thuộc hàm trực tiếp-nghĩa là ko tồn tại những phụ thuộc hàm ngoài khóa.. Nếu R không phải là 3NF, nghĩa là trong R tồn tại thuộc tính không phụ thuộc hàm trực tiếp vào khóa, thì tách các nhóm thuộc tính có phụ hàm vào thuộc tính khóa thành một quan hệ. khóa của quan hệ mới này chính là thuộc tính mà chúng có phụ thuộc hàm.



## Bài 2 TỔNG QUAN VỀ CSDL MYSQL

### I. Tổng quan về MySQL

MySQL là một hệ quản trị cơ sở dữ liệu đa luồng mã nguồn mở theo mô hình client/server, và ở mức độ chuyên dụng cho doanh nghiệp. MySQL được phát triển bởi một công ty tư vấn và phát triển ứng dụng của Thụy Điển có tên là TcX.

MySQL là một hệ quản trị cơ sở dữ liệu có tốc độ truy xuất rất nhanh và uyển chuyển. MySQL được phát triển phổ biến cho hệ điều hành Linux, tuy nhiên, với các phiên bản mới hiện nay, nó đã có thể sử dụng tốt trên của hệ điều hành Windows.

Chúng ta cần phân biệt giữa MySQL và SQL, SQL là ngôn ngữ dùng để truy xuất cơ sở dữ liệu được hãng phần mềm IBM phát triển và được sử dụng ở đa số các hệ quản trị CSDL hiện nay như MySQL, Microsoft SQL Server, DB2, Sysbase Adapter Server, SQL Lite, Oracle...

#### 1. Mô hình Client/Server

Mô hình client server (Client – server model) là một mô hình đã được sử dụng khá lâu trong lĩnh vực lập trình và phát triển ứng dụng, mô hình mạng trong một công ty, cơ quan... Trong mô hình này, sẽ có một máy chủ (server) có cấu hình khá mạnh về khả năng lưu trữ, bộ nhớ, tốc độ xử lý... được đặt làm trung tâm để lưu trữ tất cả các dữ liệu, quản lý các máy con... trong cơ quan, công ty hay quy trình phát triển ứng dụng nào đó.

Vì tính chất đặc thù của nhiều nhiệm vụ khác nhau nên máy chủ được chia làm nhiều loại : Database server (máy chủ yếu lưu trữ, quản lý cơ sở dữ liệu), Web server (máy chủ lưu trữ, quản lý ứng dụng web), Network server (máy chủ quản lý hệ thống mạng)... Ở đây, chúng ta sẽ chỉ bàn đến Database server.

Các Database server sẽ được cài đặt một hoặc nhiều hệ quản trị CSDL khác nhau, dùng để lưu trữ và xử lý các tiến trình truy cập, truy xuất, thống kê... liên quan đến dữ liệu. Khi này, client (máy con, máy khách) sẽ đóng vai trò gửi các yêu cầu, câu hỏi đến server, khi đó server sẽ tiếp nhận và xử lý các yêu cầu đó và gửi kết quả về cho client. Mô hình này còn có thể gọi là mô hình 2 tầng (2-tiers).

#### 2. Quy trình cài đặt (trên Windows)

##### a. Download MySQL

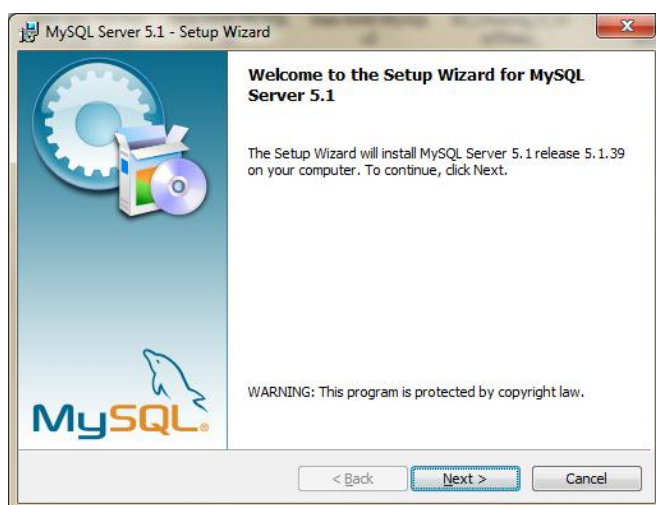
Để tải ứng dụng MySQL ta truy cập vào địa chỉ trang web <http://dev.mysql.com/downloads/>. Hiện nay, tại thời điểm viết tài liệu này, phiên bản mới nhất của MySQL là **5.4 beta**, tuy nhiên các bạn nên sử dụng phiên bản **5.1** (phiên bản được khuyến cáo sử dụng – recommend), trong giáo trình này sẽ sử dụng phiên bản **5.1**.

Bạn phải chọn phiên bản phù hợp với hệ điều hành (32bit hay 64bit). Nếu bạn sử dụng Windows 32bit (như Win2000, WinXP, Win7 32bit, WinVista 32bit) thì hãy chọn hệ điều hành Windows, ngược lại chọn Windows x64.

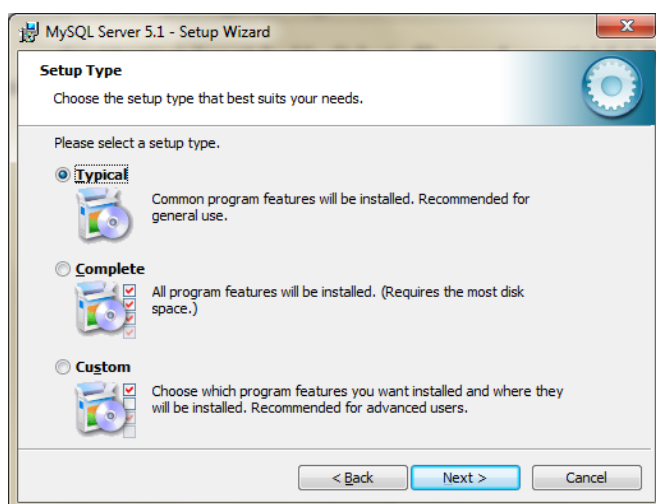
Sau khi chọn hệ điều hành, bạn hãy tải file Windows MSI Installer (file \*.msi) cho tiện việc cài đặt.

##### b. Quy trình cài đặt

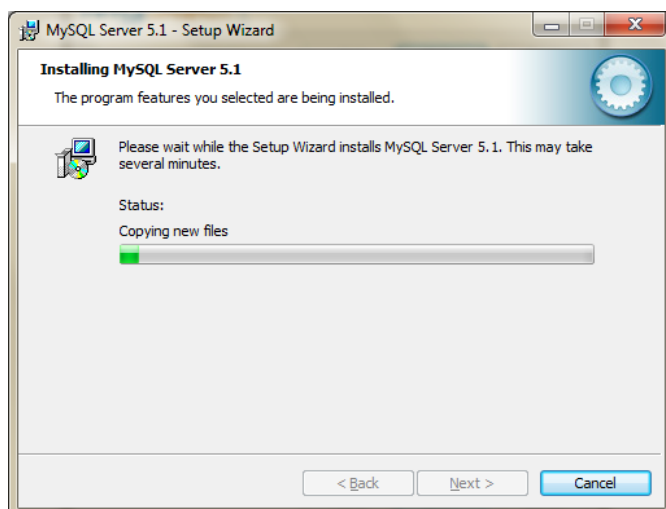
Sau khi tải về, bạn sẽ double click vào file **mysql-essential-5.1.39-win32.msi** để tiến hành cài đặt.



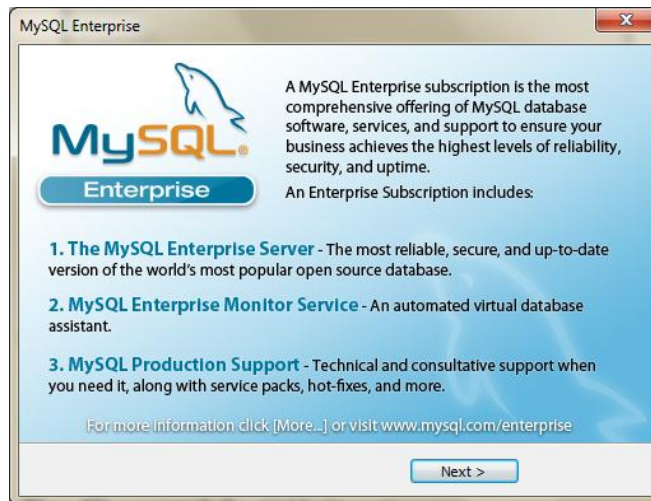
Màn hình chào của quá trình cài đặt, click Next.



Màn hình chọn phương thức cài đặt, ta chọn Typical (cài đặt mặc định), chương trình sẽ được cài đặt vào **C:\Program Files\MySQL\MySQL Server 5.1**. Click Next.



Màn hình tiến trình cài đặt.



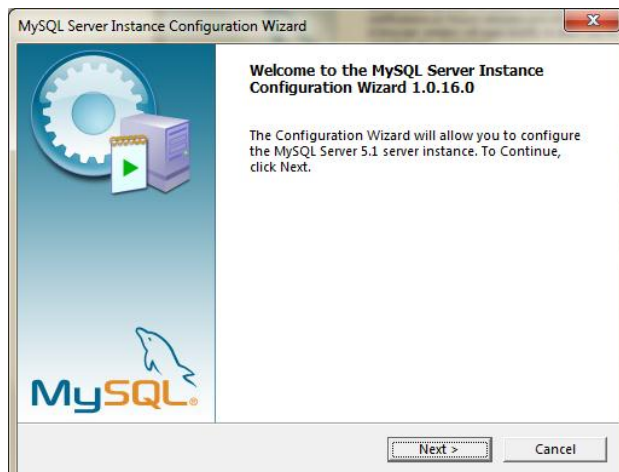
Màn hình giới thiệu các tính năng của MySQL, click Next.



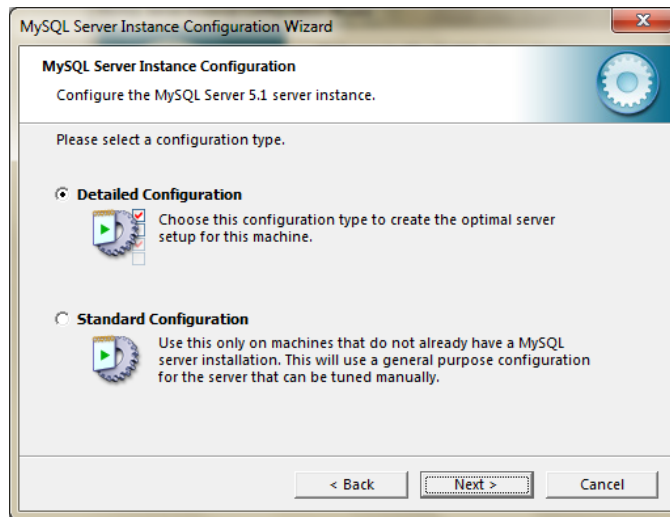
Màn hình tùy chọn sau cài đặt.

- Configure MySQL Server now : tiếp tục tiến hành cấu hình server sau khi cài đặt xong.
- Register the MySQL Server now : tiếp tục tiến hành đăng ký với MySQL.

Bạn chọn tùy chọn 1, bỏ tùy chọn 2, click Next.



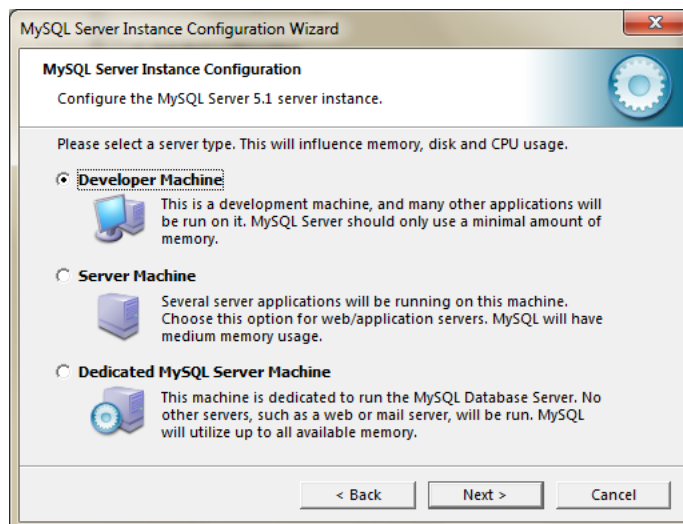
Màn hình tiến hành cấu hình Server, click Next



Màn hình tùy chọn cách cấu hình :

- Detailed Configuration : cấu hình chi tiết server.
- Standard Configuration : cấu hình theo chuẩn mặc định.

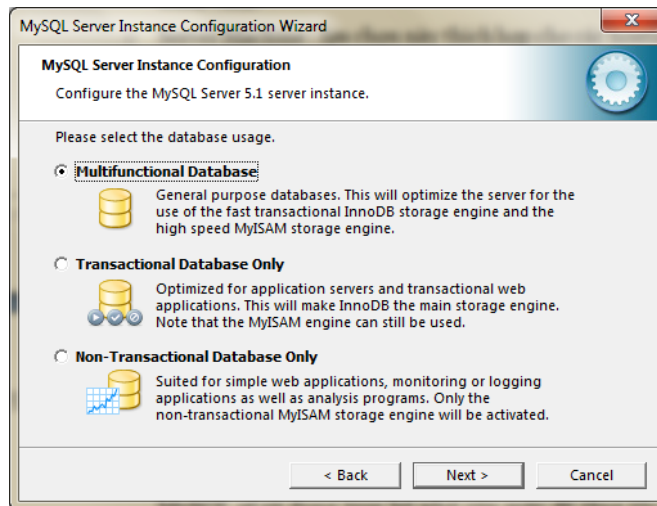
Chọn tùy chọn 1, click Next.



Màn hình lựa chọn loại server thích hợp :

- Developer Machine : lựa chọn này MySQL sẽ sử dụng ít bộ nhớ nhất, thích hợp cho các máy để phát triển các ứng dụng, website của các lập trình viên.
- Server Machine : lựa chọn này thích hợp cho các hosting server, chuyên cung cấp dịch vụ lưu trữ CSDL online.
- Dedicated MySQL Server Machine : tùy chọn này chỉ thích hợp cho các server chuyên chạy MySQL, không chạy thêm dịch vụ nào khác. MySQL sẽ sử dụng trọn bộ nhớ của máy để tăng tốc truy xuất CSDL.

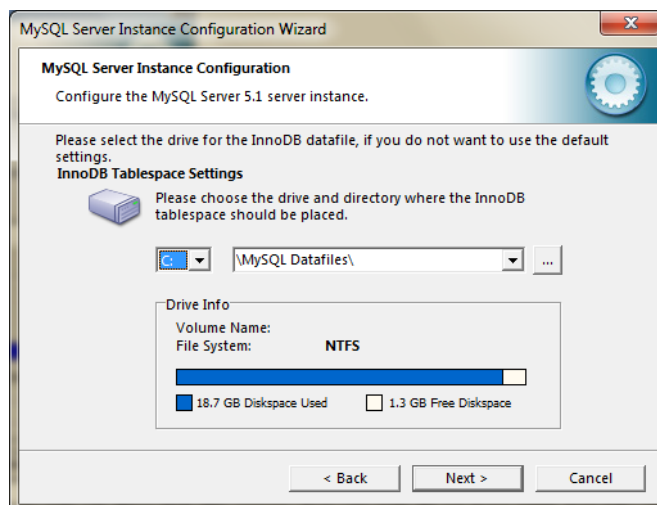
Bạn chọn tùy chọn 1, click Next.



Màn hình tùy chọn phương thức tối ưu hoá CSDL:

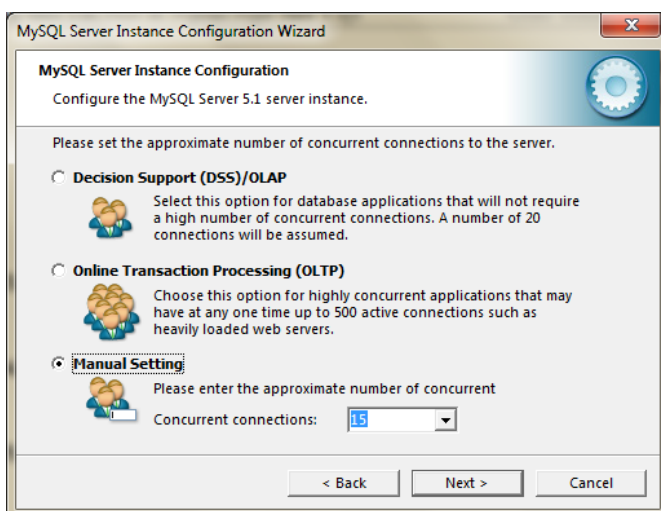
- Multifunctional Database : phương thức tối ưu hoá chung, thích hợp cho CSDL sử dụng với nhiều mục đích khác nhau.
- Transactional Database Only : Tối ưu hoá để sử dụng kiểu bảng InnoDB, thích hợp cho CSDL sử dụng nhiều đến Transaction.
- Non-Transactional Database Only : tối ưu hoá để sử dụng kiểu bảng MyISAM, thích hợp cho các CSDL lưu và phân tích log.

Bạn chọn tùy chọn 1, click Next.

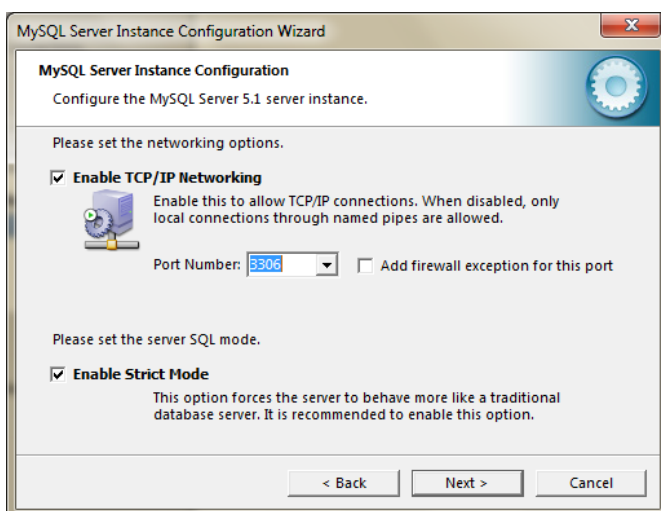


Màn hình lựa chọn nơi lưu trữ CSDL, thường sẽ được lưu trữ trong thư mục MySQL Datafiles trong thư mục cài MySQL (C:\Program Files\MySQL\MySQL Server 5.1\MySQL Datafiles).

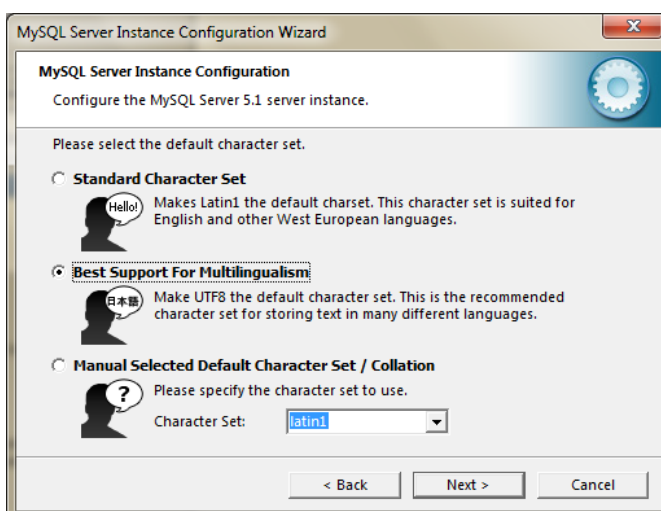
Bạn có thể đổi vị trí lưu bằng cách click dấu (...)



Màn hình lựa chọn số lượng kết nối đến server. Bạn hãy tùy chọn theo số lượng mà bạn ước lượng, tuy nhiên số lượng hợp lý là khoảng 15 – 20 cho các CSDL giành cho các ứng dụng (như Website).

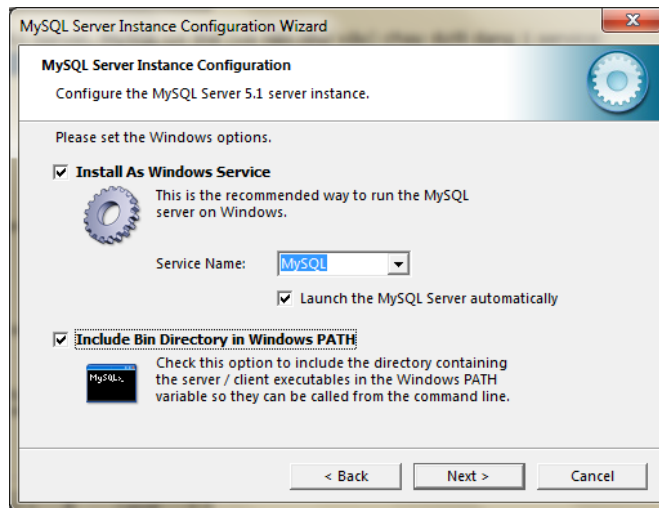


Màn hình tùy chọn phương thức kết nối đến server, để mặc định, click Next.



Màn hình tùy chọn bộ kí tự (charset) cho CSDL. Chọn tùy chọn **Best Support for Multilingualism** thích hợp cho lưu tiếng Việt (hỗ trợ UTF-8).





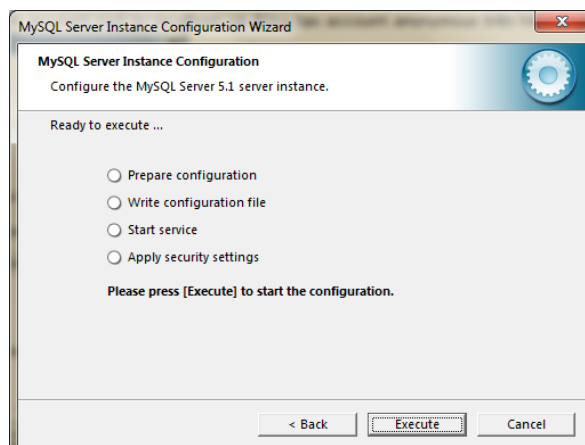
Màn hình lựa chọn phương thức chạy MySQL :

- Install As Windows Service : chạy MySQL như là một dịch vụ chạy ẩn.
- Include Bin Directory in Windows PATH : chèn đường dẫn gốc của MySQL vào các biến môi trường của Windows, không cần khai báo đường dẫn khi chạy.

Chọn cả 2 tùy chọn, click Next.



Chọn password cho tài khoản cao nhất của MySQL (tài khoản root). Hãy nhớ thật kỹ password này. Không chọn tùy chọn 2 vì lý do bảo mật.

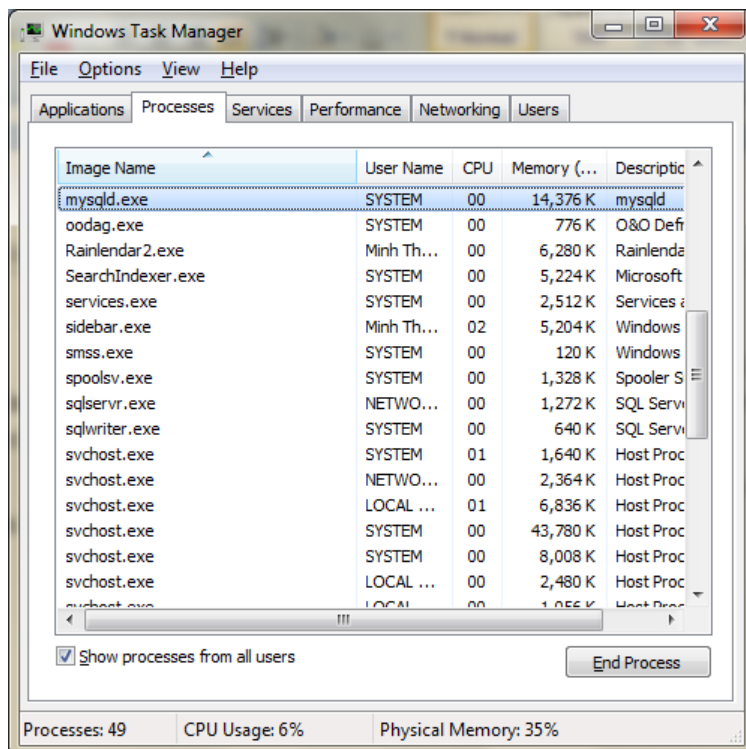


Click Execute để chạy và khởi động dịch vụ.

### 3. Khởi động & tắt dịch vụ MySQL

Sau khi cài đặt, MySQL Server sẽ chạy như là một dịch vụ (chương trình chạy ẩn) trên Windows. Bây giờ, máy của bạn vừa đóng vai trò là một server vừa đóng vai trò là một client, vì vậy tên server của máy sẽ là localhost.

Để tắt dịch vụ MySQL Server ta sẽ vào mở Task Manager, chọn thẻ Process, tìm đến tiến trình **mysqld**, click chọn và nhấn End Task để tắt dịch vụ.



Để khởi động lại dịch vụ, ta vào thư mục đã cài đặt MySQL (C:\Program Files\MySQL\MySQL Server 5.1\bin), double click vào file **MySQLInstanceConfig.exe**, thực hiện lại tiến trình cấu hình Server và chạy dịch vụ.



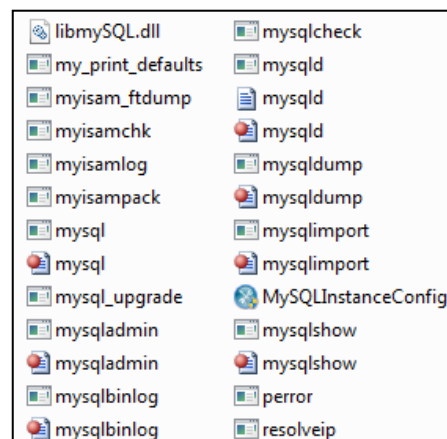
### 4. Thao tác tại client

Trong thư mục cài đặt, thư mục Bin chứa tất cả các file thực thi, các file lệnh của MySQL, các file này thường được chạy ở giao diện dòng lệnh (command line).

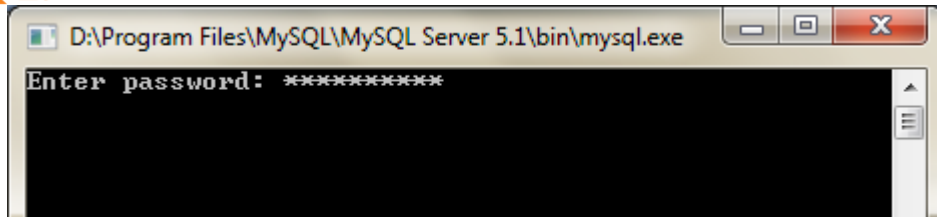
Để chạy các thao tác ở phía client ta thực thi file mysql.exe ở command line với câu lệnh :

```
mysql -u root -p
```

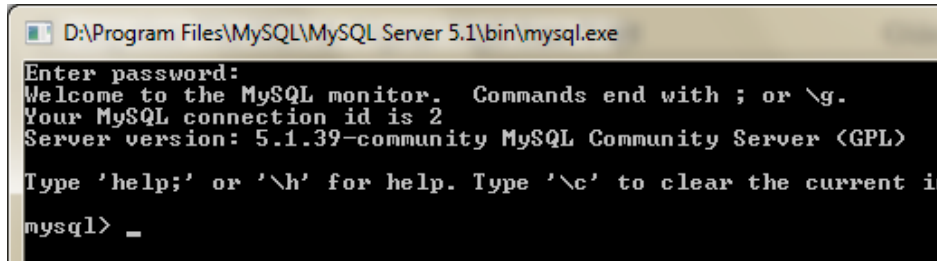
(hoặc vào Start → Programs → MySQL → MySQL Command Line Client).







Nhập password đăng nhập MySQL server, do mới chỉ có tài khoản root được tạo, nên bạn hãy nhập password của root khi cài đặt để đăng nhập server với tài khoản root.



Màn hình sau khi đăng nhập thành công sẽ là câu chào “welcome...”

Để thực hiện truy vấn CSDL ta gõ lệnh và kết thúc bằng dấu ";". Các lệnh của MySQL sẽ được trình bày chi tiết ở phần sau.

Vd : thực hiện lệnh thoát chương trình ta gõ : **quit;**

Hoặc lệnh : **exit;**

Chú ý : trong MySQL câu không phân biệt chữ hoa, chữ thường.

## 5. Thay đổi password cho tài khoản root

Mật khẩu của root là rất quan trọng, vì root là tài khoản cao nhất của MySQL Server, do đó nếu mật khẩu bị người khác biết sẽ có thể truy cập dữ liệu và làm tất cả các thao tác trên đó. Vì vậy việc thay đổi mật khẩu root khi cần thiết là việc cần làm. Để thực hiện thay đổi mật khẩu root, ta thực hiện câu lệnh sau :

```
set password for root@localhost = password("new_password");
```

Để thay đổi password cho các tài khoản khác, ta thực hiện lệnh :

```
set password for account_name@server_name = password("new_password");
```

Trong đó:

- *account\_name* : tên tài khoản.
- *server\_name* : tên server muốn truy cập, nếu server cũng là máy cục bộ, ta sử dụng localhost.

Để xem tình trạng của server, ta thực hiện lệnh

```
status;
```

## II. Các thao tác cơ bản trên CSDL

### 1. Xem các cơ sở dữ liệu

Để xem các cơ sở dữ liệu hiện có trên server, ta dùng lệnh :

```
show database;
```

## 2. Tao cơ sở dữ liệu mới

Sau khi đã phân tích và thiết kế cơ sở dữ liệu, ta có thể xây dựng cơ sở dữ liệu đó trên một hệ quản trị. Để xây dựng, thao tác đầu tiên là tạo cơ sở dữ liệu, để CSDL trên MySQL ta dùng lệnh :

```
create database database name;
```

Trong đó : database name là tên CSDL muốn tạo.

**Vd : create database *QLBanHang*;**

Sau khi tạo CSDL, để thực hiện truy xuất lên CSDL nào đó, ta dùng lệnh :

```
use database_name;
```

```
vd : use OLBanHang;
```

### 3. Xoá cơ sở dữ liệu

Để xoá một CSDL, ta dùng lệnh :

***drop database database\_name;***

vd : drop OLBanHang;

#### 4. Thêm tài khoản người dùng (user)

Sau khi đã tạo được cơ sở dữ liệu, ta có thể tạo thêm các tài khoản người dùng khác để sử dụng CSDL đó (không nên chỉ để tài khoản root sử dụng). Để thêm được tài khoản người dùng lên một CSDL ta phải hiểu cách MySQL quản lý user trên các CSDL như thế nào.

MySQL có một CSDL mặc định cũng mang tên **mysql**. Trong CSDL này bao gồm 3 bảng : **user**, **db**, **host**.

- Bảng **user** : dùng để quản lý các tài khoản người dùng, mật khẩu, thuộc những host nào, và tài khoản đó có những quyền hạn gì.
- Bảng **db** : dùng để quản lý các tài khoản người dùng được phép truy cập những CSDL nào.
- Bảng **host** : dùng để quản lý các database thuộc những hostname nào.

Trước tiên, ta phải thêm vào bảng host dòng dữ liệu quản lý CSDL vừa tạo được truy cập trên những hostname nào.

Vd : CSDL với tên là **QLDonDatHang**, được tạo trên server **localhost**.

```
use mysql;
```

```
insert into host values('localhost','QLDonDatHang','Y','Y','Y','Y','Y','Y','Y',
```

Tiếp theo, ta phải thêm vào bảng user tài khoản muốn tạo.

```
insert into user values('localhost','inzaghithanh',password('9967722'),'Y','Y','Y',  
'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y',  
'Y','Y','Y',' ',' ',' ',' ',20,20,20,20);
```

Sau cùng, ta sẽ thêm vào bảng db cho phép tài khoản vừa tạo sử dụng CSDL.

```
insert into db values('localhost','QLDonDatHang','inzaghiithanh','Y','Y','Y','Y',
    'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');
```

## Bài 3 BẢNG - TABLE

### I. Định nghĩa

Bảng là nơi lưu trữ dữ liệu. Trước khi thực hiện truy vấn và sử dụng dữ liệu, thì nhiệm vụ đầu tiên là phải lưu trữ nó.

Bảng có thể chứa dữ liệu về các đối tượng thông tin như HÀNG HÓA, NHÂN VIÊN.... Mỗi dòng (row) của bảng chứa một thể hiện riêng của đối tượng nào đó. Ví dụ thông tin về một hàng hóa, thông tin về một nhân viên.... Dòng được thành lập từ các cột (column), mỗi cột là một thông tin về đối tượng được lưu trữ trong bản. Ví dụ trong bảng NHANVIEN chúng ta có các field như sau: Field MaNV(mã nhân viên), field HoNV (họ và chữ lót), field TenNV (tên), field Nu (nam/nữ), field NgaySinh (ngày sinh).....

### II. Các kiểu dữ liệu

MySQL cung cấp các kiểu dữ liệu sau :

#### 1. Kiểu dữ số (numeric)

Tên kiểu (type name)	Bộ nhớ (memory space) bytes	Vùng giá trị (value range)	Vùng giá trị dương (Unsigned)
Tinyint	1	-128 ÷ 172	0 ÷ 255
Smallint	2	-32768 ÷ 32767	0 ÷ 65535
Mediumint	3	-8388608 ÷ 8388607	0 ÷ 16777215
Int	4	-2147483648 ÷ 2147483647	0 ÷ 4294967295
Bigint	8	-9223372036854775808 ÷ 9223372036854775807	0 ÷ 18446744073709550 615
Float (M,D)	4		
Double (M,D)	8		
Decimal (M,D)	M + 2		

#### 2. Kiểu chuỗi kí tự (string)

Tên kiểu	Kích thước tối đa (bytes)	Khoảng lưu trữ (bytes)
Char(x)	255	X
Varchar(x)	255	X + 1
Tinytext	255	X + 1
Tinyblob	255	X + 2
Text	65535	X + 2
Blob	65535	X + 2

Mediumtext	1.6 MB	X + 3
Mediumblob	1.6 MB	X + 3
Longtext	4.2 GB	X + 4
Longblob	4.2 GB	X + 4

### 3. Kiểu hỗn hợp (Miscellaneous)

**Enum** : kiểu dữ liệu liệt kê, cho phép định nghĩa trước các giá trị cho một cột, cột sẽ chỉ lưu trữ một trong các giá trị định sẵn đó.

Vd :

```
CREATE TABLE Test(
    Return ENUM('Y','N') DEFAULT 'N',
    Size ENUM('S','M','L','XL','XXL'),
    Color ENUM('Black','Red','White')
)
```

**Set** : kiểu dữ liệu liệt kê, tương tự **enum** nhưng cho phép cột lưu trữ nhiều giá trị trong các giá trị định sẵn, mỗi giá trị cách nhau bởi dấu ', '.

Vd :

```
CREATE Table Test(
    Advertiser SET('Web Page','Television','Newspaper')
)
```

### 4. Kiểu ngày giờ

Tên kiểu	Định dạng chuẩn	Giá trị 0
DATETIME	YYYY-MM-DD HH:MM:SS	0000-00-00 00:00:00
DATE	YYYY-MM-DD	0000-00-00
TIME	HH:MM:SS	00:00:0
YEAR	YYYY	0000
TIMESTAMP	(Xem phần sau)	0000000000000000 (dài nhất)

Định dạng kiểu TIMESTAMP

Tên kiểu	Định dạng
TIMESTAMP(14)	YYYYMMDDHHMMSS
TIMESTAMP(12)	YYMMDDHHMMSS
TIMESTAMP(10)	YYMMDDHHMM
TIMESTAMP(8)	YYYYMMDD
TIMESTAMP(6)	YYMMDD
TIMESTAMP(4)	YYMM
TIMESTAMP(2)	YY

## 5. Các từ khoá định nghĩa cột (modifiers)

Từ khoá	Kiểu dữ liệu thích hợp	Ý nghĩa
Auto_Increment	Int	Tự động tăng dữ liệu của cột
Binary	Char, Varchar	Thiết lập lưu trữ chuỗi nhị phân (phân biệt chữ hoa – thường)
Default	Tất cả trừ text, blob	Thiết lập giá trị mặc định cho cột
Not null	Tất cả	Không cho phép để giá trị null
Null	Tất cả	Cho phép để giá trị null
Primary key	Tất cả	Thiết lập khoá chính cho cột
Unique	Tất cả	Giá trị duy nhất
Unsigned	Kiểu numeric	Chỉ lưu giá trị số nguyên
Zerofill	Kiểu numeric	Điền giá trị 0 cho đủ chiều dài số

## III. Các thao tác trên bảng

### 1. Tạo bảng

- Tạo bảng mới

**Create table *table\_name* ( *column\_names* *datatypes* *modifiers* )**

Vd : Tạo bảng Customers (khách hàng)

```
CREATE TABLE Customers (Customer_ID INT NOT NULL
PRIMARY KEY AUTO_INCREMENT, First_Name VARCHAR(20)
NOT NULL, Last_Name VARCHAR(30) NOT NULL,
Address VARCHAR(50), City VARCHAR(20),
State VARCHAR(2), Zip VARCHAR(20),
E-Mail VARCHAR(20), Age INT, Race VARCHAR(20),
Gender ENUM('M', 'F') DEFAULT 'F',
Eye_Color VARCHAR(10), Hair_Color VARCHAR(10),
Favorite_Activity ENUM('Programming', 'Eating', 'Biking', 'Running', 'None')
DEFAULT 'None',
Favorite_Movie VARCHAR(50), Occupation VARCHAR(30), Smoker CHAR(0));
```

- Tạo bảng tạm (bảng phụ) mới

**Create temporary table *table\_name* ( *column\_names* *datatypes* *modifiers* )**

- Tạo bảng tạm (bảng phụ) từ một câu truy vấn

**Create temporary table select *column\_name* from *table\_name***

vd : create temporary table select \* from Customers

- Tạo bảng sao chép từ một bảng khác

**Create table *table\_name* select *column\_name* from *table\_name\_1***

vd : create table Customers\_copy select \* from Customers

- Kiểm tra sự tồn tại của bảng trước khi tạo

**Create table If not Exists *table\_name* (*column\_names* *datatypes* *modifiers*)**

## 2. Xem thông tin của CSDL, Bảng

- Xem các bảng của CSDL

**Show tables [from *database\_name*];**

- Xem các cột của bảng

**Show columns [from *table\_name*];**

- Xem cấu trúc của bảng

**Discribe *table\_name* [from *database\_name*];**

## 3. Xoá bảng

**Drop table *table\_name* [from *database\_name*];**

## 4. Thay đổi cấu trúc bảng

- Thay đổi tên cột

**Alter table *table\_name* change *old\_column\_name*  
*new\_column\_name* *old\_datatype*;**

Vd :        ALTER TABLE Customers  
              CHANGE First\_Name FirstName VARCHAR(20);

- Thay đổi kiểu dữ liệu

**Alter table *table\_name* change *column\_name* *column\_name* *new\_datatype*;**

Vd :        ALTER TABLE Customers  
              CHANGE Last\_Name Last\_Name VARCHAR(50);

- Đổi tên bảng

**Alter table *table\_name* Rename *new\_table\_name*;**

Vd :        ALTER TABLE Customers RENAME Customer\_Table;

- Thêm cột vào bảng

**Alter table *table\_name* add *column\_name* *datatype*;**

Vd :        ALTER TABLE Customer ADD Last\_Name VARCHAR(30);

- Xoá một cột

**Alter table *table\_name* Drop *column\_name*;**

Vd :        ALTER TABLE Customers DROP Last\_Name;

- Thêm khoá chính

**Alter table *table\_name* Add Primary Key (*column\_names*);**

Vd :        ALTER TABLE Customers ADD PRIMARY KEY (Customer\_ID);

- Xoá khoá chính

**Alter table *table\_name* Drop Primary Key;**

## 5. Chèn dữ liệu vào bảng

- Chèn một dòng dữ liệu

**Insert into table\_name (column\_names) values (column\_values);**

Vd : Tạo bảng một bảng có tên Test\_Table và chèn dữ liệu

```
CREATE TABLE Test_Table
(Test_ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
Test_Name VARCHAR(30),
Test_Date DATETIME,
Test_Giver VARCHAR(30));
INSERT INTO Test_Table (Test_ID, Test_Name, Test_Date, Test_Giver)
VALUES (NULL, 'Test','2000-01-01','Glen');
```

- Chèn nhiều dòng dữ liệu

**Insert into table\_name (column\_names) values (column\_values),  
(column\_values), (...);**

Vd :

```
INSERT INTO Test_Table (Test_ID, Test_Name, Test_Date, Test_Giver)
VALUES  (NULL, 'John','2000-01-01','Glen'),
        (NULL, 'Thomas','2000-01-01','Jose');
```

## Bài 4 THAO TÁC TRÊN CSDL

### I. Import dữ liệu

Sau khi thiết kế và tạo CSDL, ta có thể nhập dữ liệu vào các bảng bằng các dòng lệnh, nhưng phương pháp đó sẽ không khả thi nếu chúng ta đã có một nguồn dữ liệu khá lớn từ trước. Ngoài ra, nếu ta muốn lấy dữ liệu từ một hệ quản trị khác, thì ta cũng không thể nhập từng dòng lệnh. Vì vậy, các hệ quản trị luôn có chức năng Import cho phép người dùng chèn một lượng dữ liệu lớn có trước vào trong CSDL một cách nhanh nhất và ít tốn thời gian.

#### 1. Import từ file text

Để thực hiện Import dữ liệu từ một file text, ta sử dụng chương trình `mysqlimport.exe` trong thư mục Bin cài đặt MySQL. Câu lệnh import ở command line như sau :

**`mysqlimport.exe database_name table_name.txt`**

Vd : `mysqlimport QLBanHang Customer.txt`

Dữ liệu trong file text sẽ được tải vào bảng có tên cùng với tên file text. Nếu bảng chưa tồn tại, chương trình sẽ tự tạo bảng mới. Định dạng của file text phải được trình bày theo quy định sau :

- Mỗi dòng dữ liệu được trình bày trên 1 dòng.
- Giá trị text phải được đóng bằng dấu nháy đơn (') hoặc nháy kép (").
- Các giá trị cách bởi dấu phẩy (,).
- Các giá trị phải được sắp theo thứ tự tương ứng

Vd : khi import vào bảng có các cột sau *Customer\_ID int, Last\_Name varchar(25), First\_Name varchar(15)*, dữ liệu file text phải theo mẫu sau :

*1, "Nguyen Minh", "Thanh"*

Các tùy chọn của chương trình `mysqlimport`

Tùy chọn	Ý nghĩa
-d hoặc --delete	Xoá tất cả dữ liệu đang có trong bảng trước khi chèn dữ liệu mới
-f hoặc --force	Tiếp tục chèn dữ liệu khi gặp lỗi
-i hoặc --ignore	Bỏ qua những dòng dữ liệu chèn giống với những dòng có dữ liệu duy nhất
-L hoặc --local	Chỉ định sử dụng file import trên máy cục bộ
-l hoặc --lock-tables	Khoá các bảng trước khi chèn
-r hoặc --replace	Thay thế các dòng dữ liệu duy nhất bằng dòng dữ liệu chèn
--fields-enclosed-by=char	Chỉ định kí tự đóng dữ liệu ( vd : 'ABC')
--fields-escaped-by=char	Chỉ định kí tự loại trừ cho các kí tự đặc biệt
--fields-optionally-terminated-by=char	Chỉ định kí tự phân chia các giá trị
--fields-terminated-by=char	Chỉ định kí tự phân chia các giá trị
--lines-enclosed-by=str	Chỉ định kí tự kết thúc một dòng dữ liệu



Ví dụ : một file text có tên Orders.txt (Hoá đơn) với dữ liệu theo định dạng sau :

*"1", "ORD89876", "1 Dozen Roses", "19991226"*

Câu lệnh import có cấu trúc sau :

*Mysqlexport.exe -f -fields-enclosed-by="*  
*--fields-terminated-by=, QLBanHang Orders.txt*

## 2. Import từ file sql

Một cách khác để import dữ liệu đó là thực thi hàng loạt các câu lệnh sql từ một file \*.sql (hay còn gọi là batching).

Một mẫu ví dụ về file data.sql như sau :

```
USE QLBanHang;
INSERT INTO Customers (Customer_ID, Last_Name, First_Name)
VALUES(NULL, "Nguyen Minh", "Thanh");
INSERT INTO Customers (Customer_ID, Last_Name, First_Name)
VALUES(NULL, "Nguyen Thien", "Nam");
INSERT INTO Customers (Customer_ID, Last_Name, First_Name)
VALUES(NULL, "Nguyen Khoa", "Danh");
```

Để thực thi file sql ta sẽ sử dụng lệnh sau :

**Load Data Infile *filename.sql* Into Table *table\_name*;**

Vd : `LOAD DATA INFILE "C:\MyDocs\data.sql" INTO TABLE Orders;`

Nếu muốn chỉ định file sql nằm trên máy cục bộ, cá nhân :

**Load Data Local Infile *filename.sql* Into Table *table\_name*;**

Để thay thế các dòng giá trị trùng nhau :

**Load Data Local Infile *filename.sql* Replace Into Table *table\_name*;**

Tuy nhiên, ta cũng có thể sử dụng phương thức Load Data này cho các file text

**Load Data Infile *filename.txt* Into Table *table\_name*;**

Các tùy chọn được dùng thêm khi tải dữ liệu từ file text (được dùng sau từ khoá Fields) :

- **Terminated by *char***
- **Enclosed by *char***
- **Escaped by *char***

Vd :

```
LOAD DATA INFILE "Orders.txt" REPLACE INTO TABLE Orders
FIELDS TERMINATED BY ',' ENCLOSED BY '"';
```

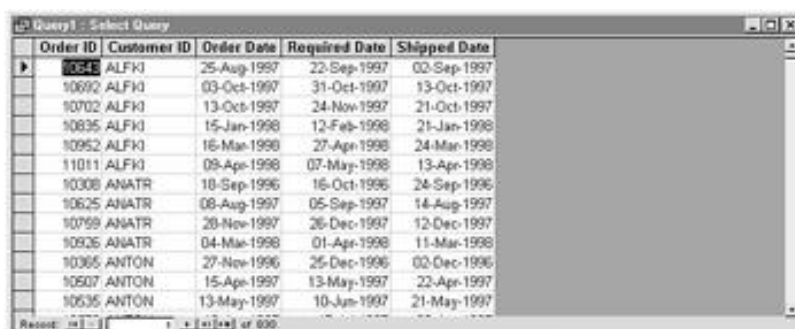
Chỉ định các cột được tải dữ liệu:

```
vd : LOAD DATA INFILE "/home/Order.txt"
      INTO TABLE Orders
      (Order_Number, Order_Date, Customer_ID);
```

### 3. Import từ file Access

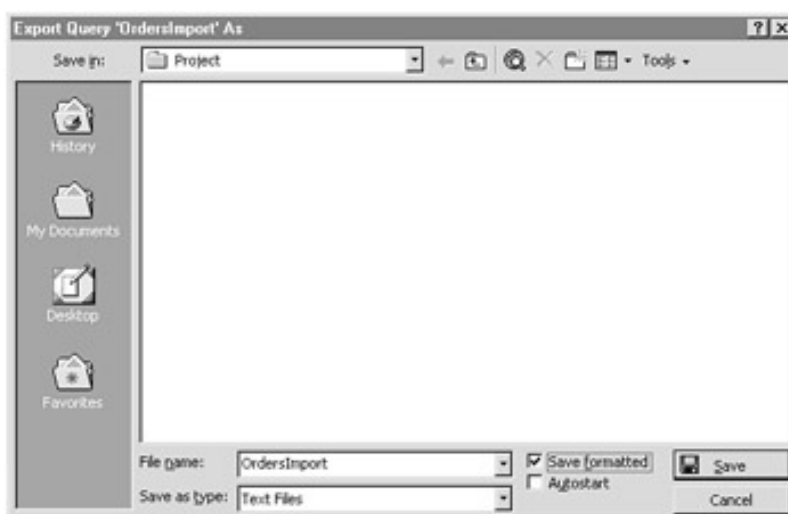
Để import dữ liệu từ một file Access, ta sẽ dùng phần mềm Access để xuất dữ liệu ra một file text trước, rồi dùng file text đó tải vào trong MySQL.

Ví dụ ta có một bảng dữ liệu **Orders** sau trong Access :



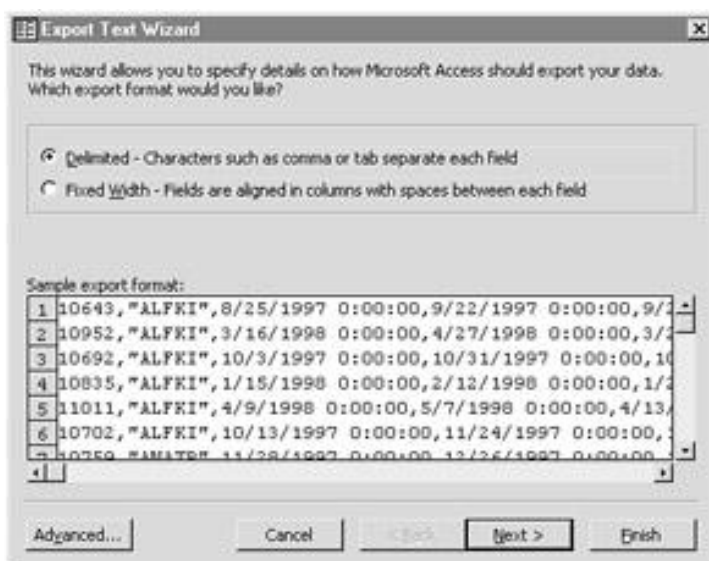
Order ID	Customer ID	Order Date	Required Date	Shipped Date
10643	ALFKI	25-Aug-1997	22-Sep-1997	02-Sep-1997
10692	ALFKI	03-Oct-1997	31-Oct-1997	13-Oct-1997
10702	ALFKI	13-Oct-1997	24-Nov-1997	21-Oct-1997
10835	ALFKI	15-Jan-1998	12-Feb-1998	21-Jan-1998
10952	ALFKI	16-Mar-1998	27-Apr-1998	24-Mar-1998
11011	ALFKI	09-Apr-1998	07-May-1998	13-Apr-1998
10308	ANATR	10-Sep-1996	16-Oct-1996	24-Sep-1996
10625	ANATR	08-Aug-1997	05-Sep-1997	14-Aug-1997
10759	ANATR	20-Nov-1997	26-Dec-1997	12-Dec-1997
10926	ANATR	04-Mar-1998	01-Apr-1998	11-Mar-1998
10365	ANTON	27-Nov-1996	25-Dec-1996	02-Dec-1996
10607	ANTON	15-Apr-1997	13-May-1997	22-Apr-1997
10636	ANTON	13-May-1997	10-Jun-1997	21-May-1997

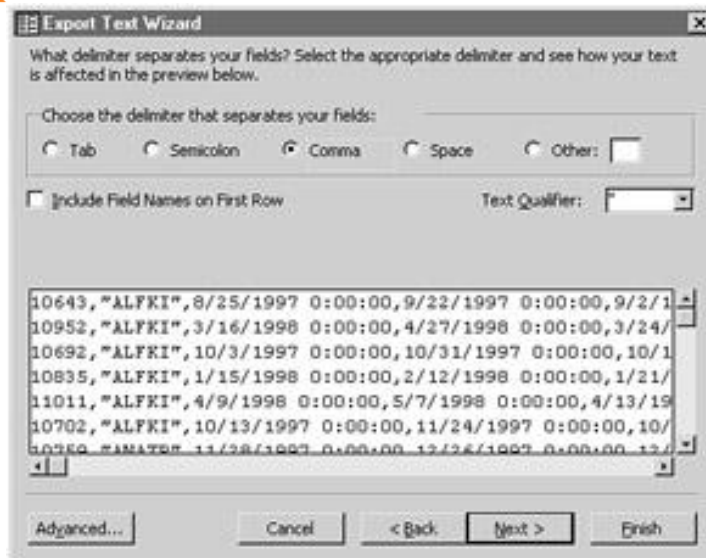
Ta sử dụng chức năng Export (trong menu File), cửa sổ Export sẽ xuất hiện :



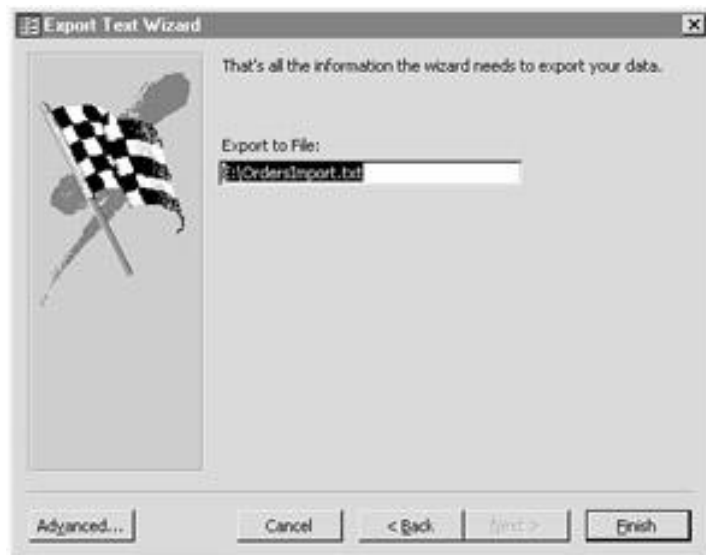
Nhập tên file trong ô filename và click vào nút chọn Save Formatted để lưu giữ các định dạng của dữ liệu.

Tiếp theo, ta sẽ chọn dấu phân cách giữa các giá trị (Delimited : phân cách bằng dấu phẩy, Fixed Width : phân cách bằng khoảng trắng và được canh lề).

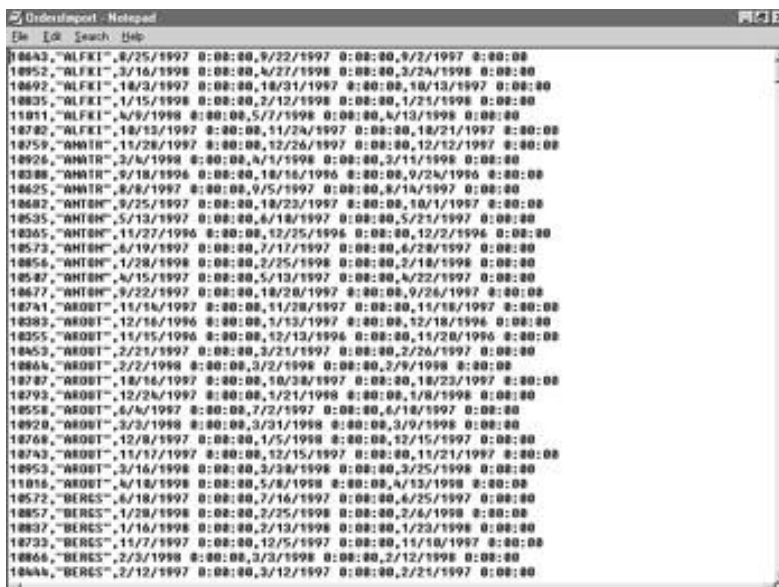




Chọn đường dẫn cho file xuất



Kết quả xuất như sau :



Sử dụng các thao tác import từ file text để import file dữ liệu vừa xuất.

## II. Export dữ liệu

### 1. Chương trình mysqldump

Để export cấu trúc định nghĩa và cả dữ liệu trong bảng ra file .txt ta dùng chương trình **mysqldump** trong thư mục Bin của MySQL. File kết xuất sẽ nằm trên server.

***Mysqldump.exe -u username -p database\_name table\_name > filename.txt***

vd : ***mysqldump -u root -p qlBanHang Customers > C:\Customer.txt***

Sau khi gõ lệnh, ta sẽ nhập password của tài khoản sử dụng. Nếu bạn không muốn tạo từng file .txt cho từng bảng, mà muốn làm cho cả CSDL, ta sử dụng lệnh

***Mysqldump.exe -u username -p database\_name > filename.txt***

### 2. Lệnh Select ... Into Outfile

Để thực hiện kết xuất dữ liệu từ một câu truy vấn (sẽ trình bày ở chương sau) ra file .txt trên server, ta thực hiện câu lệnh sau :

***Select column\_names from table\_name Into Outfile filename.txt***

***[ Fields***

***Terminated by char***

***Enclosed by char***

***Line Terminated by char ]***

vd :

***SELECT \* FROM Customer INTO OUTFILE 'c:\customers.txt'***

***FIELDS TERMINATED BY ',' ENCLOSED BY ' '***

***LINES TERMINATED BY '\r\n';***

Tuy nhiên, câu lệnh này sẽ kết xuất file trên server.

### 3. Chương trình mysql client

Hai thao tác trên đều kết xuất file trên server, nếu bạn phát triển ứng dụng trên máy cục bộ (local) thì máy sẽ vừa đóng vai trò là một server vừa đóng vai trò là một client. Tuy nhiên, nếu bạn làm việc thực sự trong môi trường server/client, thì việc kết xuất này sẽ làm mất thời gian là bạn phải chép file về máy.

Để thực hiện thao tác kết xuất trực tiếp lên máy client ta sẽ dùng chương trình mysql nhưng không đăng nhập theo cách chúng ta hay làm. Câu lệnh thực thi như sau :

***Mysql.exe -e "Select column\_names from table\_name"***

***--skip-column-names \ database\_name > filename.txt***

Vd :

***Mysql.exe -e "SELECT \* FROM Customers"***

***--skip-column-names \ qlBanHang > customers.txt***

## Bài 5 TRUY VẤN – QUERY

### I. Định nghĩa

#### 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...

Các bài trước chúng ta cũng đã sử dụng một số câu lệnh về tạo và chỉnh sửa cấu trúc bảng, lệnh tạo CSDL và chèn dữ liệu... Ở bài này, chúng ta sẽ tập chung vào vấn đề truy vấn dữ liệu, tức là chọn, thống kê, xóa và cập nhật dữ liệu.

#### 2. Các từ khoá

Các từ khoá của ngôn ngữ truy vấn SQL

Create	On	Limit	Into	Or
Alter	Order by	Drop	Where	Like
Insert	Join	Delete	Group by	As
Select	Cross Join	Update	Left Join	
From	Right Join	Set	And	

Cấu trúc của một câu lệnh truy vấn dữ liệu như sau :

**SELECT** \* | *column\_names*

**FROM** *table\_names*

**WHERE** *criteria*

**GROUP BY** *column\_names*

**ORDER BY** *column\_names*

### II. Các thao tác truy vấn dữ liệu

#### 1. Truy vấn chọn dữ liệu

- Chọn tất cả cột trong bảng

**SELECT** \* **FROM** *table\_name*

Vd : select \* from Customers;

- Chọn một vài cột trong bảng

**SELECT** *column\_names* **FROM** *table\_name*

Vd : select Lastname, Firstname from Customers;

- Chọn dữ liệu với điều kiện trích lọc dữ liệu

**SELECT** \* | *column\_names* **FROM** *table\_name*

**WHERE** *criteria\_1 and/or criteria\_2*

Vd : Select \* from Customers

Where Fistname="Thành" or Fistname="Nam";

Truy vấn gần giống với chuỗi kí tự, ta sử dụng kí tự thay thế (%)

Vd :       Select \* from Customers  
             Where Lastname like ="Nguyễn%";

- Giới hạn số lượng dòng kết quả

**SELECT \* / *column\_names* FROM *table\_name*  
WHERE *criteria* LIMIT *number*?**

Vd :       Select \* from Customers Limit 5;

- Chọn dữ liệu từ nhiều bảng

**SELECT \* / *column\_names* FROM *table\_name\_1*, *table\_name\_2*  
WHERE *table\_relationship***

Vd :       Select \* from Customers C, Orders O  
             Where C.Customer\_ID = O. Customer\_ID;

- Sắp xếp dữ liệu

**SELECT \*/*column\_names* FROM *table\_names*  
WHERE *table\_relationships* and/or *criteria*  
ORDER BY *column\_names* ASC|DESC**

Vd :       Select \* from Customers C, Orders O  
             Where C.Customer\_ID = O. Customer\_ID;  
             Order by OrderID;

- Đặt tên mới cho cột - bảng :

**Table\_name AS new\_table\_name  
Column\_name AS new\_column\_name**

Vd :       Select Lastname as HoLot, Firstname as Ten, Order\_ID as MaHoaDon  
             From Customers As C, Orders As O  
             Where C.Customer\_ID = O. Customer\_ID;  
             Order by Order\_ID;

## 2. Truy vấn thống kê dữ liệu

**SELECT *column\_names*, Aggregate\_function(*#column\_name*)  
FROM *table\_names*  
where *table\_relationships* and/or *criteria*  
GROUP BY *column\_names***

Vd :       Select Customer\_ID, Count(Order\_ID)  
             From Customers C, Orders O  
             Where C.Customer\_ID = O.Customer\_ID  
             Group by Customer\_ID;

Các Aggregate\_function (hàm thống kê) : Max(), Min(), Sum(), Count(Distinct), Avg(), Std().



Mệnh đề Group by trong truy vấn thống kê có tác dụng gom nhóm các dữ liệu giống nhau của các cột và các hàm thống kê sẽ thống kê trên các nhóm dữ liệu đó.

### 3. Truy vấn lồng

Khi một truy vấn được sử dụng trong một truy vấn khác, thì đó là câu truy vấn lồng. Câu truy vấn bên trong gọi là truy vấn con (subquery).

```
Vd :      Select Customer_ID, Max(TongTri) as TongTriLonNhat
          From (Select Customer_ID, Sum(ThanhTien) As TongTri From Orders
                Group by Customer_ID) as A
          Group by Customer_ID;
```

### 4. Mệnh đề Join

Dùng thiết lập quan hệ giữa các bảng khi thực hiện truy vấn.

**Inner Join** : các dòng dữ liệu có ở 2 bảng sẽ được giữ lại làm kết quả. Đây là mệnh đề thường được dùng nhất.

```
Vd : Select Lastname, Firstname, Order_ID
      From Customer as C (INNER) Join Order as O On C.Customer_ID =
      O.Customer_ID;
```

**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.

```
Vd : Select Lastname, Firstname, Order_ID
      From Customer as C LEFT Join Order as O On C.Customer_ID =
      O.Customer_ID;
```

### 5. Truy vấn chèn dữ liệu

**INSERT INTO table\_name(column\_names) VALUES(column\_values)**

```
Vd : Insert Into Customers(Customer_ID, Firstname, Lastname)
      Values(null, "Thành", "Nguyễn Minh");
```

### 6. Truy vấn xóa dữ liệu

**DELETE FROM table\_names WHERE criterias**

```
Vd : Delete From Customers Where Customer_ID=2;
```

### 7. Truy vấn cập nhật dữ liệu

**UPDATE table\_name SET column\_name=value,... WHERE criterias**

```
Vd : Update From Customers Set Firstname="Danh" Where Customer_ID=3;
```

## III. Toán tử & hàm

### 1. Toán tử

- Phép cộng / trừ / nhân / chia : dùng để tính toán 2 cột dữ liệu số

```
Vd_1 : Select ThanhTien + Thue as TongSoTien From Orders as HoaDon;
```

```
Vd_2 : Select ThanhTien - GiamGia as TongSoTien From Orders as HoaDon;
```

Vd\_3 : *Select SoLuong \* DonGia as ThanhTien From Orders as HoaDon;*

- Phép ghép chuỗi :

Vd : *Select Lastname & Firstname as Fullname From Customers;*

- Phép IN : xác định một giá trị có nằm trong một tập hợp

Vd : *Select Customer\_ID, Lastname, Firstname From Customer*

*Where Customer\_ID In (Select Customer\_ID, Count(Order\_ID) From Customer C, Order O Where C.Customer\_ID = O.Customer\_ID*

*Group by Customer\_ID Order By Count(Order\_ID))*

## 2. Hàm toán học

- Mod (số bị chia, số chia) : lấy phần dư của phép chia.

Vd : *Select Mod(ThanhTien,2) From Orders as HoaDon;*

- Round(số, vị trí làm tròn) : hàm làm tròn số.

Vd : *Select Round(ThanhTien,1) From Orders as HoaDon;*

Vị trí làm tròn :

2 : 2 số thập phân

1 : 1 số thập phân

0 : 0 số thập phân

-1 : hàng đơn vị

-2 : hàng chục

## 3. Hàm điều kiện

- IF(logic\_expression,true\_result,false\_result) : hàm kiểm tra điều kiện đúng/sai.

Vd : *Select If(SoLuong>20,5%,2%) as GiamGia From Orders;*

- IFNULL(result\_1,result\_2) : hàm trả về kết quả result\_1 nếu nó không null ngược lại sẽ trả về result\_2.

Vd : *Select IfNull(10/0,1) as Exam*

- CASE value WHEN expression THEN result\_1 ELSE result\_2 : hàm trả về result\_1 khi expression đúng, ngược lại trả về result\_2.

Vd : *Select CASE 1 WHEN Column1="Y" THEN 1 WHEN Column2="Y" THEN 2 WHEN Column3="Y" THEN 3 ELSE "NONE";*

## 4. Hàm logic

Các hàm logic được dùng trên các biểu thức điều kiện (logic\_expression).

- AND : phép và.
- OR : phép hoặc.
- NOT : phép phủ định.

Vd : *Select \* From Orders O, OrderDetails D, Products P*

*Where O.Order\_ID = D.Order\_ID AND D.Product\_ID = P. Product\_ID;*



## 5. Hàm chuỗi

- LTRIM, RTRIM, TRIM : cắt bỏ những khoảng trắng thừa.  
Vd : `Select LTRIM(" ABC") → "ABC"`  
Vd : `Select RTRIM("ABC ") → "ABC"`  
Vd : `Select TRIM(" ABC ") → "ABC"`  
Ta cũng có thể sử dụng để cắt bỏ chuỗi kí tự được chỉ định với các từ khoá (
- SUBSTRING(chuỗi, vị trí bắt đầu) : lấy một chuỗi con từ vị trí bắt đầu  
Vd : `Select Substring("Hello World",7) → "World"`
- LOCATE(chuỗi 1, chuỗi 2, vị trí bắt đầu) : xác định vị trí chuỗi 1 trong chuỗi 2 từ vị trí bắt đầu.  
Vd : `Select Locate("lo","Hello World",1) → 4`
- REPLACE(chuỗi 1, chuỗi 2, chuỗi 3) : thay thế chuỗi 2 bằng chuỗi 3 trong chuỗi 1.  
Vd : `select Replace("Filename.xxx","xxx","123") → "Filename.123"`
- UCASE, LCASE : chuyển chữ thường thành in hoa và ngược lại.
- REVERSE : đảo ngược chuỗi.

## 6. Hàm thời gian

- MONTHNAME(date) : trả về tên tháng của date.  
Vd : `Select monthname(20000105) → January`
- DAYOFYEAR(date) : trả về số ngày tính từ đầu năm đến date.  
Vd : `Select dayofyear(20000201) → 32`  
Vd : `Select dayofyear("2000/02/01") → 32`
- DAYOFMONTH(date) : trả về số ngày tính từ đầu tháng đến date.  
Vd : `Select dayofyear(20000201) → 1`  
Vd : `Select dayofyear("2000/02/01") → 1`  
Vd : `Select dayofyear("00-02-01") → 1`
- DAYOFWEEK(date) : trả về số ngày tính từ đầu tuần đến date.  
Vd : `Select dayofyear(20000201) → 3`  
Vd : `Select dayofyear("0-2-1") → 3`
- YEAR(date) : trả về năm của date.
- QUARTER(date) : trả về quý của date.
- MONTH(date) : trả về tháng của date.
- DAY(date) : trả về ngày của date.
- WEEK(date) : trả về số tuần của date tính từ ngày đầu tiên của năm.
- YEARWEEK(date) : trả về số tuần của date tính từ ngày đầu tiên của năm.
- NOW(), SYSDATE(), CURRENT\_TIMESTAMP : trả về ngày giờ hệ thống.  
Vd : `Select NOW() → 2009-10-28 18:05:11`

Vd : Select NOW()+0 → 20091028180511

- CURDATE() , CURRENT\_DATE : trả về ngày của hệ thống.
- CURTIME() , CURRENT\_TIME : trả về ngày của hệ thống.
- HOUR(time) : trả về giờ của time.
- MINUTE(time) : trả về phút của time.
- SECOND(time) : trả về giây của time.
- DATE\_FORMAT(date,format) : định dạng date theo format.
- TIME\_FORMAT(time,format) : định dạng date theo format.

Bảng tham số định dạng	
%r	12-giờ (hh:mm:ss AM PM)
%T	24-giờ (hh:mm:ss AM PM)
%Y	Định dạng năm, 4 chữ số
%y	Định dạng năm, 2 chữ số
%m	Gắn số 0 vào đầu tháng (01, 02...)
%c	Bỏ số 0 đầu tháng (1, 2...)
%M	Tên tháng (đầy đủ)
%b	Tên tháng (viết tắt)
%D	Ngày (1 <sup>st</sup> , 2 <sup>nd</sup> ...)
%d	Ngày có số 0 đầu (01, 02, 03...)
%e	Ngày không có số 0 đầu (1, 2, 3...)
%W	Ngày trong tuần (Sunday, Monday...)
%a	Ngày trong tuần (viết tắt)
%H	Giờ (00, 01...23)
%k	Giờ (0, 1, 2,...23)
%h	Giờ (01, 02, ... 12)
%L	Giờ (01, 02, ... 12)
%I	Giờ (1, 2, ... 12)
%i	Phút (00, 01,...59)
%S	Giây (00, 01,...59)
%s	Giây (00, 01,...59)
%P	AM   PM
%U	Số tuần trong năm (với Sunday là ngày đầu tiên)
\$u	Số tuần trong năm (với Monday là ngày đầu tiên)
%X&%V	Năm và số tuần (với Sunday là ngày đầu tiên)
%x&%v	Năm và số tuần (với Month là ngày đầu tiên)
%j	Số ngày trong năm (001, 002,...366)
%w	Số ngày trong tuần (0 = Sunday, 1 = Monday)
%%	Định dạng %

Vd : Select Date\_Format("1999-12-31 23:00:00","%r on % W")

→ kết quả : 11:00:00 PM on Friday

- TO\_DAYS(date) : trả về số ngày từ ngày đầu tiên sau **Công Nguyên** đến date.

Vd : TO\_DAYS("2000-01-01") → 730485

- FROM\_DAYS(*number*) : trả về ngày sau **Công Nguyên** *number* ngày.  
Vd : FROM\_DAYS(1000000) → 2737-11-28
- TIME\_TO\_SEC(*time*) : trả về thời gian tính bằng giây từ 0:00:00 đến *time*.
- SEC\_TO\_TIME(*seconds*) : trả về thời điểm sau 0:00:00 một khoảng thời gian là *seconds* giây.
- DATE\_ADD(*date*, INTERVAL *expression type*) : cộng thêm một khoảng thời gian vào *date*.
- ADDDATE(*date*, INTERVAL *expression type*) : cộng thêm một khoảng thời gian vào *date*.
- DATE\_SUB(*date*, INTERVAL *expression type*) : bớt đi một khoảng thời gian trong *date*.
- SUBDATE(*date*, INTERVAL *expression type*) : bớt đi một khoảng thời gian trong *date*.

Vd: ADDDATE("1980-07-04",INTERVAL 15 YEAR) → 1995-07-04

Vd: SUBDATE(20001201, INTERVAL "10 4" YEAR\_MONTH)→ 1990-08-01

Bảng các giá trị Type trong công thức

Type	Định dạng dữ liệu
SECOND	Giây
MINUTE	Phút
HOURL	Giờ
DAY	Ngày
MONTH	Tháng
YEAR	Năm
MINUTE_SECOND	"Phút : Giây"
HOURL_MINUTE	"Giờ : Phút"
DAY_HOURL	"Ngày Giờ"
YEAR_MONTH	"Năm Tháng"
HOURL_SECOND	"Giờ : Phút : Giây"
DAY_MINUTE	"Ngày Giờ : Phút"
DAY_SECOND	"Ngày Giờ : Phút : Giây"

## Bài 6 FUNCTION – PROCEDURE – TRIGGER

### I. Function

Function là phương thức dùng để định nghĩa ra các hàm mới do chính người quản trị CSDL tạo ra. Function dùng để tạo ra các hàm tương tự như những hàm đã được trình bày ở trên (NOW(), TRIM()...), Function sẽ được định nghĩa bởi một chương trình (các dòng lệnh của MySQL ghép lại, mỗi dòng lệnh sẽ được kết thúc bằng dấu ‘;’) để thực hiện một yêu cầu tính toán nào đó và trả về một giá trị (kết quả) mà ta cần. Để tạo được Function bạn cần có quyền CREATE ROUTINE.

Vd : tạo hàm tính giá trị tuổi trung bình của khách hàng trong Customers.

#### 1. Cấu trúc tạo Function

```
CREATE FUNCTION function_name(arguments)
RETURNS datatype characteristics
BEGIN
    ...
    RETURN value;
END
```

Trong đó :

- Function\_name : tên function muốn tạo
- Arguments : danh sách các tham số cần thiết cho function
- Datatype : kiểu dữ liệu mà function trả về
- Characteristics : các đặc trưng của function  
DETERMINISTIC, NO SQL, READS SQL DATA
- RETURN : kết quả trả về của function.

Vd : tạo hàm tính giá trị tuổi trung bình của khách hàng trong Customers.

```
CREATE FUNCTION TuổiTB()
RETURNS Float Reads SQL Data
BEGIN
    RETURN select avg(Year(Birthday)) From Customer;
END
```

#### 2. Sử dụng Function

Ta có thể dùng Function mới tạo như các function có sẵn của MySQL (ở bài 3). Function sẽ được lưu lại trong chính CSDL đang sử dụng. Tuy nhiên, chúng ta không thể liệt kê danh sách các Function, vì vậy chúng ta phải nhớ các Function đã tạo.

Vd : *Select TuổiTB();*

## II. Procedure

Procedure là phương thức dùng để tạo ra các hàm xử lý tuy nhiên Procedure không trả về giá trị (kết quả) như Function. Procedure thường được dùng để thao tác lên CSDL nhiều hơn. Đối với Function, ta không thể chỉnh sửa dữ liệu, như đối với Procedure ta có thể tạo ra các hàm dùng để chỉnh sửa, xoá... dữ liệu. Kết quả của Procedure không phải là một giá trị như Function, mà là một bộ giá trị (kết quả của một truy vấn...)

### 1. Cấu trúc tạo Procedure

```
CREATE PROCEDURE procedure_name(IN | OUT | INOUT arguments)
BEGIN
    ....
END
```

Trong đó :

- Procedure\_name : tên Procedure muốn tạo
- Arguments : các tham số cần thiết cho procedure, các tham số phải sử dụng các từ khoá In, Out, InOut để chỉ định loại argument (In – tham số chỉ cho giá trị vào, Out – tham số chỉ cho giá trị ra, InOut – tham số cho giá trị vào và ra).

Vd : liệt kê danh sách các khách hàng theo phái.

```
CREATE PROCEDURE GetCustomersByGender(IN gen varchar(1))
BEGIN
    Select * From Customer Where Gender = gen;
END
```

### 2. Sử dụng Procedure

Để thực thi một Procedure đã được tạo ta sử dụng từ khoá Call như sau :

```
CALL procedure_name(argument_values);
```

Vd : Call GetCustomersByGender('F');

## III. Trigger

Trigger là phương thức để tạo các ràng buộc cho CSDL khi CSDL có sự thay đổi như thêm, xoá, sửa dữ liệu. Các trigger sẽ tự động được CSDL thực thi khi những hành động thay đổi đó xảy ra.

Cấu trúc tạo Trigger :

```
CREATE TRIGGER trigger_name
[BEFORE / AFTER] [INSERT / UPDATE / DELETE]
FOR EACH ROW
BEGIN
    ...
END
```

Trong đó :

- Trigger\_name : tên trigger.
- Before | After : thao tác trước hay sau.
- Insert | Update | Delete : thao tác khi chèn mới | cập nhật | xóa
  - NEW : chỉ định dòng mới được chèn, cập nhật.
  - OLD : chỉ định dòng mới được xóa.

Vd: Trong bảng Customers có một cột AGE (tuổi), cột này không nhập nhưng sẽ được tự tính từ cột Birthday khi chèn thông tin một khách hàng mới

```
CREATE TRIGGER Update_Age_Customer AFTER INSERT
FOR EACH ROW
BEGIN
    Update Customers Set Age = Year(NEW.Birthday)
    WHERE Customer_ID = New.Customer_ID;
END
```

## Bài 7 BẢO MẬT VÀ QUẢN TRỊ

### I. Bảo mật CSDL

#### 1. Phương thức bảo mật trong MySQL

Như đã trình bày trong bài 2 (phần II.4), MySQL có một CSDL đặc biệt có tên `mysql`. Trong CSDL này có các bảng chứa thông tin về server, đặc biệt chúng ta đã biết đến 3 bảng là `db`, `host`, `user`. Ngoài ra còn các bảng như `func`, `column_priv`, `tables_priv` miêu tả về quyền hạn và chức năng của các tài khoản người dùng. Người duy nhất được truy cập vào các bảng này là người quản trị CSDL hay người nắm tài khoản root. Tất cả các dữ liệu trên server được lưu trữ trong thư mục `data` (được chứa trong thư mục cài đặt `mysql`).

#### 2. Tạo tài khoản người dùng

Tài khoản với tất cả các quyền truy cập trên máy cục bộ.

```
GRANT ALL PRIVILEGES ON *.* | database.table  
TO username@localhost IDENTIFIED BY "password"
```

Vd : `GRANT ALL PRIVILEGES ON *.*`

```
TO inzaghithanh@localhost IDENTIFIED BY "9967722";
```

Tài khoản với tất cả các quyền truy cập từ xa.

```
GRANT ALL PRIVILEGES ON *.* | database.table  
TO username@host IDENTIFIED BY "password"
```

Vd : `GRANT ALL PRIVILEGES ON *.*`

```
TO inzaghithanh@"saoviet.edu.vn" IDENTIFIED BY "9967722";
```

Vd : `GRANT ALL PRIVILEGES ON *.*`

```
TO inzaghithanh@"%" IDENTIFIED BY "9967722";
```

Vd : `GRANT ALL PRIVILEGES ON *.*`

```
TO inzaghithanh@"203.162.4.190" IDENTIFIED BY "9967722";
```

#### 3. Xóa tài khoản người dùng

Xóa tài khoản trong bảng `user`

```
DELETE FROM user WHERE user="value" and host="value"
```

Vd : `DELETE FROM user WHERE user="inzaghithanh" and host="localhost";`

#### 4. Cấp quyền & xóa quyền cho tài khoản người dùng

- Cấp quyền :

```
GRANT privilege_list ON *.* | database.table  
TO username@host IDENTIFIED BY "password"
```

Vd : `GRANT select, insert On *.*`

```
TO khoadanh@localhost IDENTIFIED BY "9967722";
```

- Xóa quyền :

```
REVOKE ALL | PRIVILEGE_TYPE
```

**ON \*.\* | Database.Table From user@host**

Vd : REVOKE ALL ON \*.\* From khoadanh@localhost

## II. Quản trị

### 1. Backup (sao lưu) dữ liệu

Để sao lưu lại cơ sở dữ liệu, ta sử dụng thao tác Export dữ liệu (bài 4 phần II) ra một file text hay file sql và lưu trữ lại.

### 2. Restore (phục hồi) dữ liệu

Để phục hồi lại cơ sở dữ liệu, ta sử dụng thao tác Import dữ liệu (bài 4 phần I) từ một file text hay file sql để đưa dữ liệu vào.

### 3. Các hàm trên hệ thống của MySQL

- DATABASE() : trả về CSDL đang sử dụng.
- USER(), SYSTEM\_USER(), SESSION\_USER() : trả về tài khoản đang sử dụng.
- VERSION() : xem thông tin phiên bản MySQL
- PASSWORD(string) : trả về giá trị string được mã hoá.

### 4. Sử dụng mysqladmin

Chương trình mysqladmin chứa các lệnh để quản lý CSDL MySQL, chương trình được thực thi ở chế độ dòng lệnh

Các lệnh của mysqladmin

- Create database\_name : tạo CSDL với tên chỉ định.
- Drop database\_name : xoá CSDL với tên chỉ định.
- Status : xem tình trạng server.
- Version : xem phiên bản MySQL.
- Extended-status : xem thông tin mở rộng của server.
- Variables : xem các biến thông số có giá trị.
- Processlist : xem các kết nối đến server.
- Password new\_password : thay đổi password.
- Flush-hosts : lưu tất cả các host trong bộ đệm.
- Flush-logs : lưu tất cả các nhật ký.
- Flush-tables : lưu tất cả các bảng.
- Flush-privileges : cập nhật lại các bảng cấp quyền.
- Kill id1, id2 : tắt các luồng xử lý của MySQL.
- Ping : gửi tín hiệu đến MySQL để kiểm tra các hàm.
- Reload : tải lại các quyền từ các bảng lưu trữ quyền.
- Refresh : lưu lại tất cả các bảng quyền, đóng lại và mở các file nhật ký.
- Shutdown : tắt server.



### Các tham số của mysqladmin

- -#, --debug : xuất ra file nhật ký để kiểm tra.
- -f, --force : xoá
- -u, --user=*username* : chỉ định tài khoản sử dụng mysqladmin.
- -p, --password[=*password*] : chỉ định password sử dụng mysqladmin.
- -h, --host=*hostname* : kết nối đến một host chỉ định.
- -C, --compress : sử dụng giao thức nén giữa client/server.
- -P, --port[=*portname*] : chỉ định port cho kết nối.
- -?, --help : hướng dẫn, giúp đỡ.
- -V, --version : xem phiên bản và thoát.

Vd : mysqladmin -u root -p create qlBanhang

Vd : mysqladmin status

## Bài 8 SỬ DỤNG CÔNG CỤ GIAO DIỆN ĐỒ HOẠ - MySQL GUI TOOLS

### I. Giới thiệu:

MySQL GUI Tool là bộ công cụ với giao diện đồ hoạ thân thiện, giúp người sử dụng thao tác trên MySQL hơn so với thao tác bằng giao diện câu lệnh.

Ta có thể tải MySQL GUI Tools trên trang web chính thức của MySQL (<http://dev.mysql.com/downloads/gui-tools/5.0.html>). Bộ công cụ bao gồm 3 công cụ chính :

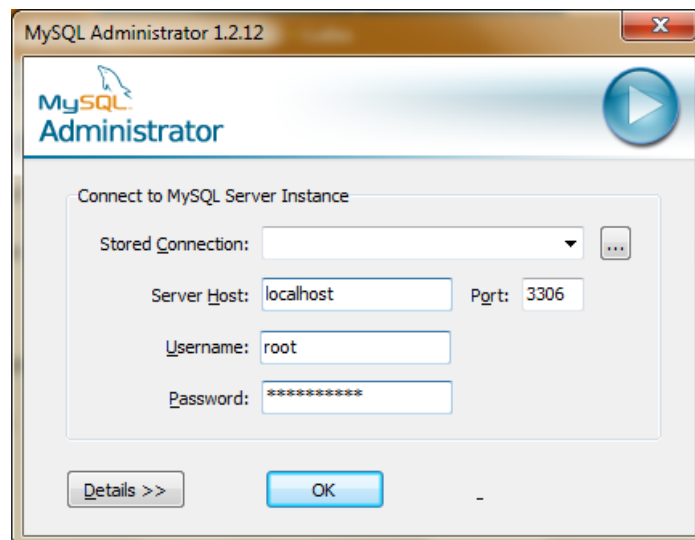
- MySQL Administrator : công cụ quản trị MySQL Server.
- MySQL Query Browser : công cụ thực hiện các câu truy vấn trực quan.
- MySQL Migration Toolkit : công cụ cho phép chuyển đổi giữa các hệ CSDL khác nhau.

### II. MySQL Administrator

Sau khi cài đặt MySQL GUI Tools, ta vào :

Start → Program → MySQL → MySQL Administrators

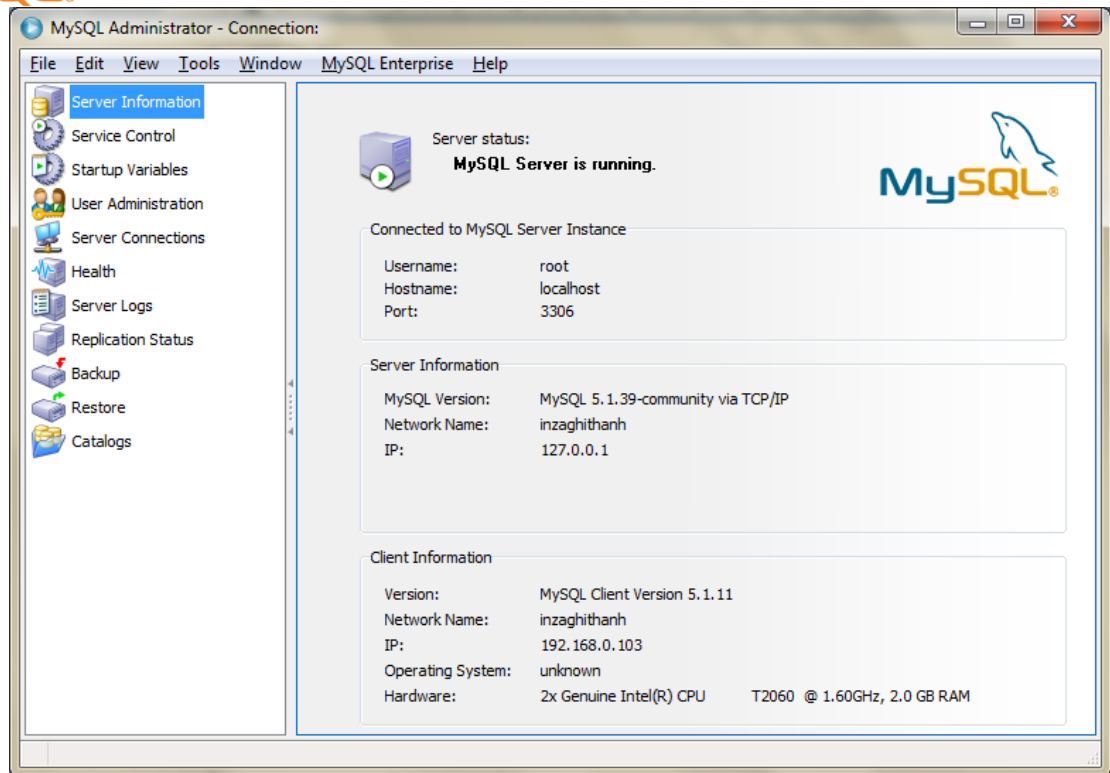
Màn hình đăng nhập hệ thống server :



- Server Host : nhập địa chỉ MySQL server muốn đăng nhập
- Username : tài khoản đăng nhập.
- Password : mật khẩu đăng nhập.

Màn hình chính của chương trình :

- Server Information : thông tin về server.
- Service Control : quản lý dịch vụ MySQL chạy trên Server.
- Startup Variables : quản lý các biến thông số tự chạy.
- User Administrator : quản lý tài khoản người dùng.



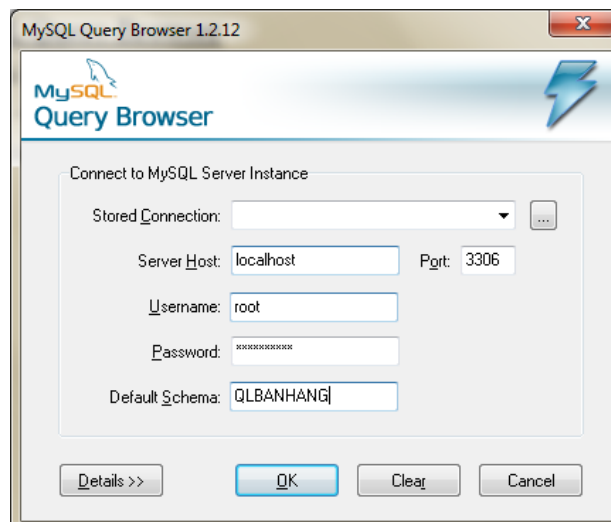
- Server Connections : quản lý các kết nối trên đến server.
- Health : quản lý tài nguyên trên server.
- Server Logs : nhật kí ghi nhận trên server.
- Replication Status : quản lý tình trạng trả lời từ server cho các client.
- Backup : thực hiện sao lưu dữ liệu.
- Restore : thực hiện phục hồi dữ liệu.
- Catalogs : danh mục các CSDL trên server.

### III. MySQL Query Browser

Sau khi cài đặt MySQL GUI Tools, ta vào :

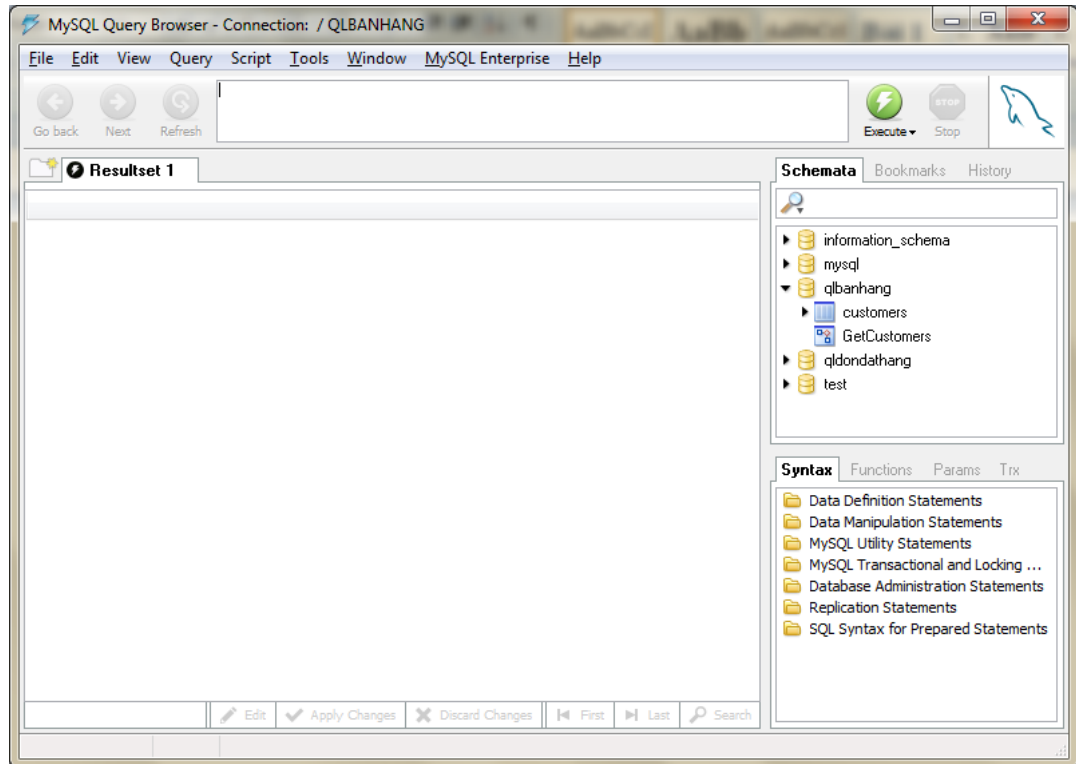
Start → Program → MySQL → MySQL Administrators

Màn hình đăng nhập hệ thống server :



- Server Host : nhập địa chỉ MySQL server muốn đăng nhập
- Username : tài khoản đăng nhập.
- Password : mật khẩu đăng nhập.
- Default Schema : Cơ sở dữ liệu sử dụng mặc định.

Màn hình chính của chương trình :



Để sử dụng, ta nhập câu truy vấn vào ô textbox phía trên và nhấn phím Execute để thực thi. Ngoài ra, ta cũng có thể chọn Procedure ở cửa sổ Schemata để thực thi câu Procedure đó.

# Bài Tập

## Bài Tập Thực Hành Số 1

- Cài đặt MySQL 5.1 trên Windows với các yêu cầu sau :
  - Chọn thư mục chứa dữ liệu MySQL Datafiles trong thư mục gốc cài đặt MySQL.
  - Thiết lập password cho root@localhost là : *saovietadmin*.
- Sử dụng MySQL Command Line Client, đăng nhập Server với tài khoản root.
 

Xem các CSDL hiện có trên Server.

Xem tình trạng của Server.

Đổi password của root@localhost là : *saoviet*.

Thoát khỏi chương trình.
- Sử dụng cmd (command line) của Windows đăng nhập MySQL Server với tài khoản root.
 

Tạo CSDL mới có tên là : *Test*.

Sử dụng CSDL vừa tạo.

Xoá CSDL vừa tạo.

## Bài Tập Thực Hành Số 2 : Tạo Bảng

- Tạo CSDL có tên : *QuanLyBanHang*.
- Sử dụng CSDL *QuanLyBanHang*, tạo các bảng sau :

**Bảng TỉnhTP**

Tên cột	Kiểu dữ liệu	Thuộc tính
MaTP	CHAR(2)	Primary key, Not Null
TenTP	VARCHAR(30)	Not Null, Default ' '

**Bảng DonVi**

Tên cột	Kiểu dữ liệu	Thuộc tính
MaDV	CHAR(2)	Primary key, Not Null
TenDV	VARCHAR(30)	Not Null, Unique

**Bảng NhaCungCap**

Tên cột	Kiểu dữ liệu	Thuộc tính
MaNCC	INT	Primary key, Not Null, Auto_Increment
TenNCC	VARCHAR(30)	Not Null, Unique
DiaChi	VARCHAR(60)	Not Null, Default ' '
DienThoai	VARCHAR(11)	Not Null, Default ' '

**Bảng MatHang**

Tên cột	Kiểu dữ liệu	Thuộc tính
MaMH	INT	Primary key, Not Null, Auto_Increment

TenMH	VARCHAR(30)	Not Null, Unique
DonViTinh	VARCHAR(20)	Not Null, Default ''
DonGia	INT	Not Null, Unsigned, Default 0
HinhSP	TEXT	Null
MaNCC	INT	Not Null

**Bảng KháchHang**

Tên cột	Kiểu dữ liệu	Thuộc tính
MaKH	VARCHAR(10)	Primary key, Not Null
TenKH	VARCHAR(40)	Not Null, Unique, Default ''
DiaChi	VARCHAR(60)	Not Null, Default ''
DienThoai	VARCHAR(11)	Not Null, Default ''
MaTP	CHAR(2)	Not Null

**Bảng NhanVien**

Tên cột	Kiểu dữ liệu	Thuộc tính
MaNV	INT	Primary key, Not Null, Auto_Increment
HoLot	VARCHAR(30)	Not Null, Default ''
TenNV	VARCHAR(20)	Not Null, Default ''
Phai	ENUM('F','M')	Not Null, Default 'F'
LuongCB	INT	Not Null, Unsigned, Default 0
NgayNViec	DATE	Not Null
DiaChiNV	VARCHAR(60)	Not Null, Default ''
DienThoaiNV	VARCHAR(11)	Null, Default ''
HinhNV	TEXT	Null

**Bảng DonDatHang**

Tên cột	Kiểu dữ liệu	Thuộc tính
MaDDH	CHAR(5)	Primary key, Not Null
MaKH	VARCHAR(10)	Not Null
MaNV	INT	Not Null
NgayDH	DATE	Not Null
NgayGiao	DATE	

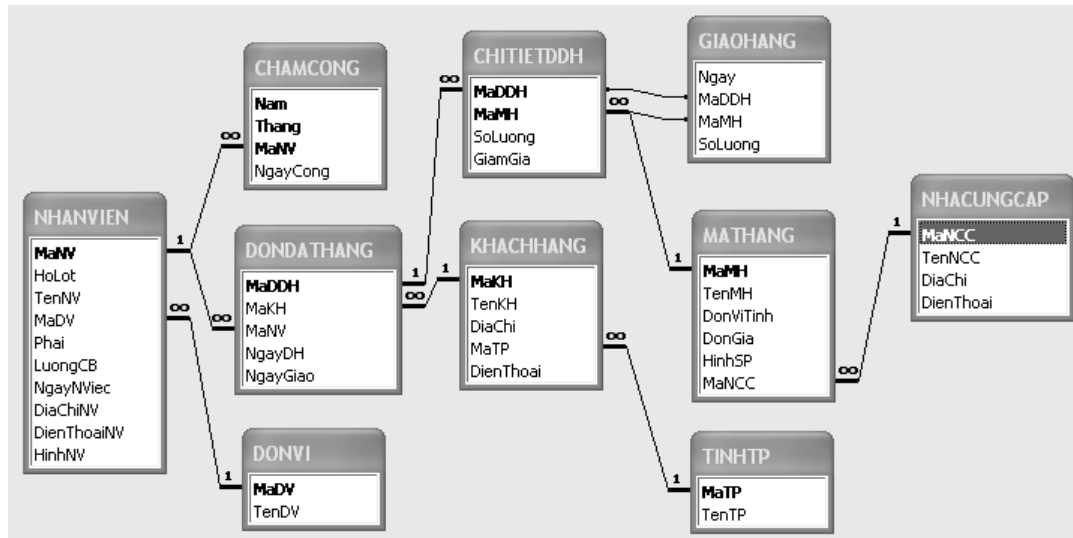
**Bảng ChiTietDDH**

Tên cột	Kiểu dữ liệu	Thuộc tính
MaDDH	CHAR(5)	Primary key, Not Null
MaMH	INT	Primary key, Not Null
SoLuong	INT	Not Null
GiamGia	DATE	Default 0

**Bảng ChamCong**

Tên cột	Kiểu dữ liệu	Thuộc tính
Thang	TINYINT	Primary key, Not Null, Unsigned
Nam	SMALLINT	Primary key, Not Null, Unsigned
MaNV	INT	Primary key, Not Null
NgàyCong	TINYINT	Default 0

### 3. Mô hình quan hệ giữa các bảng :



### 4. Thực hiện một số câu lệnh trên bảng :

Xem danh sách các bảng của CSDL *QuanlyBanHang*.

Xem danh sách của các cột trong từng bảng.

Xem cấu trúc của từng bảng.

### 5. Nhập dữ liệu vào bảng

Bảng **TinhTP**

MaTP	TenTP
CT	Cần Thơ
HN	Hà Nội
HP	Hải Phòng
HU	Huế
ND	Nam Định
NT	Nha Trang
TP	TP. Hồ Chí Minh
VI	Vinh

Bảng **DonVi**

MaDV	TenDV
GD	Ban Giám Đốc
KT	Kế Toán
NV	Nghiệp Vụ
TC	Tổ Chức



**Bảng NhaCungCap**

MaNCC	TenNCC	DiaChi	DienThoai
1	CTY Đồ Hộp Tân Bìn	TP.HCM	123456
2	CTY Rượu Bình Tân	TP.HCM	342342
3	CTY Đức Phát	TP.HCM	453456
4	CTY Rau Quả Việt Nam	Hà Nội	867567
5	CTY Thủy Sản Tiền Giang	Tiền Giang	876876
6	CTY VINAMILK	TP.HCM	646478
7	CTY Nông Sản Lâm Đồng	Đà Lạt	565467

**Bảng MatHang**

MaMH	TenMH	DonViTinh	DonGia	HinhSP	MaNCC
1	Rượu	Chai	\$230.00		7
2	Tiêu	Hộp	\$40.00		7
3	Bánh Kem	Cái	\$10.00		3
4	Bơ	Kg	\$38.00		6
5	Bánh Mì	Cái	\$8.00		3
6	Nem	Kg	\$23.79		1
7	Táo	Kg	\$5.00		4
8	Cá Hộp	Thùng	\$62.50		1

**Bảng KhachHang**

MaKH	TenKH	DiaChi	MaTP	DienThoai
CODACO	Cơ Khí Dân Dụng	388 Hòa Hảo P5	TP	1757828
COTEC	Công Nghệ Mới	319 Nguyễn Trãi P7	CT	7657657
DHP	Đại Hưng Phát	343 Nhật Tảo P16	NT	6546565
GIAY32	Giày 32	105A Ngô Quyền P11	CT	6565468
RUBIMEX	Công Nghiệp Cao Su	220 Lê Văn Sỹ	NT	2987912
SAMACO	SXKD Dịch Vụ Tổng Hợp	56 Nguyễn Biểu P2	CT	1987987
SAMECO	Cơ khí Sài Gòn	156 Lê Đại Hành P7	HN	2397997
TRACODI	Đầu tư phát triển GTVT	Hàm Nghi Q1	HP	0976766
VITICO	Hóa Nhựa Vĩnh Tiến	35 Tô Hiến Thành P14	TP	3987980
VTP	Van HƯng Phát	119 Bạch Vân P17	HN	3765768

Bảng **NhanVien** (Import từ file NhanVien.txt)

Bảng **ChamCong** (Import từ file ChamCong.txt)

Bảng **DonDatHang** (Import từ file DonDatHang.txt)

Bảng **ChiTietDDH** (Import từ file ChiTietDDH.txt)

### Bài Tập Thực Hành Số 3 : Truy vấn

Thực hiện các câu truy vấn chọn sau :

- Liệt kê danh sách các mặt hàng, bao gồm các thông tin sau : MaMH, TenMH, DonViTinh, DonGia.
- Liệt kê các mặt hàng có đơn giá  $\geq 62$  thông tin gồm các cột sau: MaMH, TenMH, DonViTinh, DonGia.
- Liệt kê các mặt hàng mà nhà cung cấp có MaNCC là 4 hoặc 2 thông tin gồm các cột sau: MaNCC, TenNCC, MaMH, TenMH, DonViTinh, DonGia.

4. Liệt kê các mặt hàng mà nhà cung cấp có địa chỉ tại TP.HCM thông tin gồm các cột sau: MaNCC, TenNCC, DiaChi, MaMH, TenMH, DonViTinh, DonGia.
5. Liệt kê các đơn hàng trong khoảng thời gian từ ngày 01/01/1997 đến ngày 31/01/1997 thông tin gồm các trường sau: MaDDH, NgayDH, MaKH, TenKH, MaMH, TenMH, SoLuong.
6. Liệt kê các đơn hàng trong tháng 2 năm 1997 thông tin gồm các trường sau : MaDDH, NgayDH, NgayGiao, MaNV, HoTenNV.
7. Liệt kê các đơn đặt hàng giao trước ngày 01/03/1997 và nhân viên phục vụ có MaNV là 4 hay 5. Thông tin gồm các trường sau : MaDDH, NgayDH, MaKH, TenKH, MaNV, HoTenNV.
8. Liệt kê các nhân viên thuộc phái Nam. Thông tin gồm các trường sau: MaNV, HoLot, TenNV, DiaChi, NgayNViec, Phai
9. Liệt kê các nhân viên nhận việc từ ngày 01/07/1991 đến ngày 30/11/1992 và lương cơ bản <300 thông tin gồm các Field sau: MaDV, MaNV, HoLot, TenNV, Phai, LuongCB, NgayNV.
10. Liệt kê các nhân viên Nữ cư trú tại Q1 Hay Nam. Thông tin gồm các field sau: MaNV, HoLot, Ten,NV, Phai, DiaChi.
11. Liệt kê các nhân viên có ngày công làm việc trong tháng 1 năm 1999 nhỏ hơn 25 ngày thông tin gồm các Field sau: MaNV, HoLot, TenNV, MaDV, TenDV, Thang, Nam, NgayCong.

*Thực hiện các câu truy vấn chọn (có biểu thức) sau :*

12. Liệt kê các đơn hàng mà nhân viên phục vụ thuộc phái nữ thông tin gồm các trường sau: MaDDH, NgayDH, NgayGiao, MaNV, HoTenNV, Phai.
13. Liệt kê các đơn hàng được hẹn giao trong vòng 15 ngày sau khi đặt hàng. Thông tin gồm các trường sau: MaDDH, NgayDH, NgayGiao, MaKH, TenKH, Thời gian (Từ ngày đặt hàng đến ngày giao hàng)
14. Liệt kê các đơn đặt hàng. Thông tin gồm các trường sau : MaDDH, NgayDH, MaMH, TenMH, SoLuong, DonGia, ThanhTien (Biết  $ThanhTien = SoLuong * DonGia * (1 - GiamGia)$ )
15. Liệt kê các đơn hàng được đặt trong quý 1 năm 1997. Thông tin gồm các trường sau : Nam, Quy, NgayDH, MaDDH, MaKH, TenKH, MaNV, TenNV
16. Liệt kê các đơn đặt hàng có chứa các mặt hàng có mã như sau: 1,3,5,7 thông tin gồm : MaDDH, NgayDH, MaMH, TenMH

*Thực hiện các câu truy vấn thống kê sau :*

17. Đếm tổng số khách hàng theo từng TỉnhTp. Thông tin gồm: MaTP, TongsoKH.
18. Đếm tổng số nhân viên theo từng Đơn Vị. Thông tin gồm: MaDV, TenDV, TongSoNV.
19. Đếm số đơn hàng. Thông tin gồm : Nam, MaKH, TenKH, TongSoDDH.
20. Tính tổng trị (Thành tiền) của từng đơn hàng. Thông tin gồm : Nam, MaDDH, MaKH, NgayDH, NgayGiao, TongTri.

21. Tính tổng thành tiền của từng khách hàng trong từng năm. Thông tin gồm : Nam, MaKH, TenKH, TongTien
22. Liệt kê danh sách khách hàng có tổng trị đặt hàng nhiều nhất trong năm 1998. thông tin gồm : MaKH, TenKH, Nam, TongTienMax.
23. Liệt kê danh sách các mặt hàng có tổng giá trị đặt hàng nhiều nhất trong 1997 thông tin gồm : MaMH, TenMH, Nam, TongTienMax
24. Cuối năm công ty muốn tặng lịch cho khách hàng với quy định như sau: 1 đơn đặt hàng 2 cuốn lịch. Tạo truy vấn gồm các thông tin sau: MaKH, TenKH, DiaChi, SoDDH, SoLichTang.
25. Thống kê tổng lương nhân viên của từng Đơn vị trong năm 1999. thông tin gồm : MaDV, TenDV, Thang, TongLuong.
26. Liệt kê Đơn vị có tổng lương trong năm 1999. lớn nhất thông tin gồm : MaDV, TenDV, Thang, TongLuongMax.

*Thực hiện các câu truy vấn hành động sau :*

27. Tạo Table LuuDDH gồm các đơn đặt hàng trong năm 1997,1998 và 1999 gồm các trường sau : MaDDH, MaKH, NgayDH, NgayGiao, MaMH, SoLuong, GiamGia.
28. Xóa Trong Table LuuDDH các đơn đặt hàng 1997
29. Tăng 2% giảm giá trong Table LuuDDH cho các đơn đặt hàng trong năm 1999
30. Tạo table Luong9901 (bảng lương tháng 1 năm 1999) gồm các Field sau: Thang, Nam, MaNV, HoTen, LuongCB, NgayCong, Luong. (biết lương = Lương căn bản \* Ngày công)
31. Tăng lương các nhân viên trong bảng lương Luong9901 tăng 10% đối với các nhân viên Nam và 20% đối với các nhân viên nữ và điều kiện là có ngày công > 20 ngày.
32. Liệt kê các mặt hàng không có đặt hàng trong khoảng thời gian [từ 1/5/1997] [đến 1/5/1998]

### **Bài Tập Thực Hành Số 3 : Function – Procedure - Trigger**

1. Tạo Function tính tổng doanh thu của năm 1997.
2. Tạo Function tính tổng doanh thu của một năm (được truyền tham số).
3. Tạo Function tính tổng thành tiền của một đơn đặt hàng (được truyền tham số).
4. Tạo Function tính tổng doanh thu của một nhân viên (được truyền tham số).
5. Thực hiện lại các câu truy vấn trên với Procedure.
6. Tạo Procedure liệt kê danh sách nhân viên theo phái (được truyền tham số).
7. Tạo Procedure liệt kê danh sách mặt hàng theo đơn đặt hàng (được truyền tham số).
8. Tạo Procedure liệt kê danh sách mặt hàng theo đơn đặt hàng (được truyền tham số).
9. Tạo Procedure liệt kê danh sách mặt hàng theo nhà cung cấp (được truyền tham số).
10. Tạo Procedure liệt kê danh sách đơn đặt hàng được đặt từ ngày A đến ngày B (A, B được truyền tham số).

**Bài Tập Thực Hành Số 4 : Bảo Mật**

1. Tạo tài khoản mới (chỉ truy cập trên local) với đầy đủ quyền có  
**tên : saoviet**  
**mật khẩu : saoviet.**
2. Thực hiện đăng nhập với tài khoản saoviet.
3. Thực hiện cho phép tài khoản saoviet có thể truy cập ở bất kì đâu.
4. Tạo tài khoản mới (chỉ truy cập trên local) chỉ có quyền chọn (select) :  
**tên : abc**  
**mật khẩu : 123456.**
5. Đăng nhập với tài khoản abc. Và thực hiện chọn, chèn dữ liệu vào bảng **MatHang**.
6. Đăng nhập với tài khoản root, xóa tài khoản abc.
7. Sử dụng các hàm hệ thống để xem thông tin Server.
8. Thực hiện Backup toàn bộ dữ liệu của QuanLyBanHang ra file QuanLyBanHang.sql