

# CHƯƠNG 4

## NGÔN NGỮ SQL: TRUY VẤN, RÀNG BUỘC

# NGÔN NGỮ SQL

- 4.1. Câu lệnh mô tả dữ liệu DDL (Data Definition Language)
- 4.2. Câu lệnh thao tác dữ liệu DML (Data Manipulation Language)
- 4.3. Câu lệnh truy vấn dữ liệu SQL (Structured Query Language)
- 4.4. Câu lệnh quản lý dữ liệu DCL (Data Control Language)

# KHÁI QUÁT VỀ NGÔN NGỮ DỮ LIỆU SQL

- Một DBMS phải có ngôn ngữ giao tiếp giữa người sử dụng với CSDL. Ngôn ngữ giao tiếp CSDL gồm các thành phần:
  - Ngôn ngữ mô tả dữ liệu (***Data Definition Language - DDL***): cho phép khai báo cấu trúc bảng, mối quan hệ, các quy tắc.
  - Ngôn ngữ thao tác dữ liệu (***Data manipulation Language - DML***): cho phép thêm, xoá, sửa.
  - Ngôn ngữ truy vấn dữ liệu hay ngôn ngữ hỏi đáp có cấu trúc (***Structured Query Language - SQL***): cho phép truy vấn các thông tin.
  - Ngôn ngữ quản lý dữ liệu (***Data Control Language - DCL***): cho phép thay đổi cấu trúc, khai báo bảo, cấp quyền.

# KHÁI QUÁT VỀ NGÔN NGỮ SQL...

- 1975-1976: SEQUEL (Structured English Query language) \_IBM.
  - Quan hệ SYSTEM-R
- 1976: SEQUEL2. 1978-1979 SEQUEL2
- 1978-1979: Ngôn ngữ truy vấn có cấu trúc
  - Hệ quản trị CSDL: SYSTEM-R.
- 1986: Chuẩn SQL- 86
- 1989: Chuẩn SQL- 89
- 1992: Chuẩn SQL- 92
- 1996: Chuẩn SQL- 96
  
- ☞ *Tất cả các hệ quản trị CSDL lớn trên thế giới cho phép truy cập bằng SQL và hầu hết theo chuẩn ANSI*

# KHÁI QUÁT VỀ NGÔN NGỮ SQL...

- Ngôn ngữ gần với ngôn ngữ tự nhiên (tiếng Anh)
- SQL là ngôn ngữ phi cấu trúc, tức là trong các lệnh của SQL người sử dụng CHỈ CẦN đưa ra yêu cầu hệ thống CÁI GÌ chứ không cần chỉ ra phải làm THẾ NÀO.
  - Ví dụ: Cho cấu trúc dữ liệu để quản lý học sinh như sau  
HOCSINH(MaHS, TenHS, ĐTB, Xeploai). Đưa ra TenHS, ĐTB của các học sinh có ĐTB $\geq$ 8.0.  
  
**Select** TenHS, ĐTB **From** HOCSINH **Where** ĐTB $\geq$ 8.0;
- SQL được chia 2 loại: SQL (ngôn ngữ hỏi) và PL/SQL (ngôn ngữ lập trình).

# KHÁI QUÁT VỀ NGÔN NGỮ SQL...

- Một số quy ước
  - Các biến cú pháp người sử dụng phải điền cụ thể vào khi viết lệnh (< >)
  - Các thành phần tùy chọn ([ ]).
  - Lựa chọn một trong các khả năng (|).
  - Thành phần bắt buộc phải chọn trong danh sách: ({ }).
  - Lệnh SQL có thể được viết trên nhiều dòng và kết thúc lệnh bởi dấu chấm phẩy ( ; ),
  - Từ khoá, tên, hàm, tên thuộc tính, tên bảng, tên đối tượng thì không được phép viết tách xuống hàng. SQL không phân biệt chữ hoa và chữ thường.

# 4.1.CÂU LỆNH MÔ TẢ DỮ LIỆU DDL (DATA DEFINITION LANGUAGE)



# CÂU LỆNH MÔ TẢ DỮ LIỆU DDL

- SQL chuẩn (86, 89, 92, 96) quy định cách đặt tên tên bảng, cột, View, ràng buộc toàn vẹn,... như sau:
  - Gồm tối đa 32 ký tự chữ cái, chữ số và dấu (\_), bắt đầu bằng chữ cái hoặc (\_).
  - Tên bảng phải là duy nhất trong CSDL và tên bảng trung gian, và không trùng với từ khoá.
  - Tên cột của một bảng là khác nhau, có thể giống nhau nếu chúng nằm trong các bảng khác nhau.
  - Một số HQTCSDDL cho phép tên có dấu cách, khi thao tác phải bao bởi cặp []
  - Không phân biệt hoa, thường
- Câu lệnh SQL kết thúc bằng dấu ;
- -- là chú thích



# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC

- SQL Server cung cấp 6 loại kiểu dữ liệu
- **1. Kiểu dữ liệu Exact Numeric (số chính xác, không sai số) trong SQL**

Kiểu dữ liệu	Từ	Tới
bigint	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
Int	-2,147,483,648	2,147,483,647
smallint	-32,768	32,767
tinyint	0	255
Bit	0	1
decimal	$-10^{38} + 1$	$10^{38} - 1$
numeric	$-10^{38} + 1$	$10^{38} - 1$
money	-922,337,203,685,477.5808	+922,337,203,685,477.5807
smallmoney	-214,748.3648	+214,748.3647

- **2. Kiểu dữ liệu Approximate Numeric trong SQL**

Kiểu dữ liệu	Từ	Tới
float	$-1.79E + 308$	$1.79E + 308$
Real	$-3.40E + 38$	$3.40E + 38$

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

## ▪ 3. Kiểu dữ liệu Date và Time trong SQL

Kiểu dữ liệu	Từ	Tới
<b>datetime</b>	Jan 1, 1753	Dec 31, 9999
<b>smalldatetime</b>	Jan 1, 1900	Jun 6, 2079
<b>date</b>	Lưu giữ date dạng June 30,2016	
<b>time</b>	Lưu giữ time dạng 12:30 P.M.	

## ▪ 4. Kiểu dữ liệu Character String (kiểu chuỗi) trong SQL

Kiểu dữ liệu	Miêu tả
<b>char</b>	Không chứa Unicode, độ dài tối đa là 8.000 ký tự (các ký tự không phải Unicode có độ dài cố định)
<b>varchar</b>	Không chứa Unicode, độ dài tối đa là 8.000 ký tự (dữ liệu không phải Unicode có độ dài có thể thay đổi)
<b>varchar(max)</b>	Không chứa Unicode, độ dài tối đa là 231 ký tự, dữ liệu không phải Unicode có độ dài có thể thay đổi (chỉ với SQL Server 2005)
<b>text</b>	Không chứa Unicode, độ dài tối đa là 2.147.483.647 ký tự, dữ liệu không phải Unicode có độ dài có thể thay đổi

## ▪ 5. Kiểu dữ liệu Unicode Character String trong SQL

Kiểu dữ liệu	Miêu tả
<b>nchar</b>	Độ dài tối đa là 4.000 ký tự (Unicode có độ dài cố định)
<b>nvarchar</b>	Độ dài tối đa là 4.000 ký tự (Unicode có độ dài có thể thay đổi)
<b>nvarchar(max)</b>	Độ dài tối đa là 231 ký tự, Unicode có độ dài có thể thay đổi (chỉ với SQL Server 2005)
<b>ntext</b>	Độ dài tối đa là 1.073.741.823 ký tự (Unicode có độ dài có thể thay đổi)

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

## ▪ 6. Kiểu dữ liệu Binary

Kiểu dữ liệu	Miêu tả
<b>binary</b>	Độ dài tối đa là 8.000 byte (dữ liệu binary có độ dài cố định )
<b>varbinary</b>	Độ dài tối đa là 8.000 byte (dữ liệu binary có độ dài có thể thay đổi)
<b>varbinary(max)</b>	Độ dài tối đa là 231 byte, dữ liệu binary có độ dài có thể thay đổi (chỉ với SQL Server 2005)
<b>image</b>	Độ dài tối đa là 2.147.483.647 byte (dữ liệu binary có độ dài có thể thay đổi)

Kiểu dữ liệu	Miêu tả
<b>sql_variant</b>	Lưu giữ các giá trị của các kiểu dữ liệu đa dạng được hỗ trợ bởi SQL Server, ngoại trừ text, ntext, và timestamp
<b>timestamp</b>	Lưu giữ một số duy nhất mà được cập nhật mỗi khi một hàng được cập nhật
<b>uniqueidentifier</b>	Lưu giữ một định danh chung (Globally Unique Identifier - GUID)
<b>xml</b>	Lưu giữ dữ liệu XML. Bạn có thể lưu giữ xml trong một column hoặc một biến (chỉ với SQL Server 2005)
<b>cursor</b>	Tham chiếu tới một đối tượng con trỏ (Cursor)
<b>table</b>	Lưu giữ một tập hợp kết quả để xử lý vào lần sau

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

- **Tạo CSDL**

- CREATE DATABASE <tên CSDL>

VD: Tạo CSDL QLHANGHOA

Create Database QLHANGHOA;

- **Tạo bảng CSDL**

- CREATE TABLE <tên bảng>

- ( <tên cột 1> <kiểu dữ liệu 1> (<kích thước 1>),

- ...,

- <tên cột n> <kiểu dữ liệu n> (<kích thước n>),

- [[CONSTRAINT <tên RB1>] <Ràng buộc 1>,

- ...,

- [CONSTRAINT <tên RBn>] <Ràng buộc n>] )

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

## ❑ Các dạng ràng buộc gồm:

- ❑ NOT NULL: Không rỗng
- ❑ UNIQUE: Duy nhất
- ❑ PRIMARY KEY: Khóa chính
- ❑ FOREIGN KEY ( Referential ) REFERENCES : Khóa ngoại
- ❑ CHECK: kiểm tra giá trị - DEFAULT: mặc định

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

- Ví dụ CSDL
  - Khách(Mak, tenk, diachi, dienthoai)
  - Hang(mah, tenh, slton, maloai )
  - HoaDon(SoHD, ngayHD, Mak)
  - ChitietHD(SoHD, mah, slb, dgia)

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

- Ví dụ: Tạo bảng HANG

```
Create table HANG (  
Mah char(5) not Null,  
Tenh varchar(30) ,  
Slton int) ;
```

Hang(mah, tenh, slton, ma\_loai )

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

## ■ Ví dụ: Tạo bảng HANG

```
Create table HANG (  
Mah char(5) not Null Primary Key,  
Tenh char(30) ,  
Slton int);
```

Hoặc

```
Create table HANG (  
Mah char(5) not Null,  
Tenh char(30) ,  
Slton int,  
Constraint H_PK Primary Key  
(Mah));
```

Hoặc

```
Create table HANG (  
Mah char(5) not Null,  
Tenh char(30) ,  
Slton int,  
Primary Key(mah));
```



# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

- Tạo bảng KHACH với tên và địa chỉ là duy nhất. Mặc định điện thoại là '0',

```
Create Table KHACH(  
Mak char(10) not null primary key,  
Tenk varchar(30),  
Diachi varchar(50),  
Dienthoai varchar(12) default '0',  
constraint UN_Ten_DC unique(Tenk,Diachi));
```

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

tạo bảng HoaDon

```
Create table HOADON  
(Sohd char(5) not null primary key,  
ngayhd date,  
mak char(10) ,  
foreign Key (mak) References khach(mak)) ;
```

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

- Ví dụ : Tạo bảng ChitietHD

```
Create table ChitietHD  
  (Sohd char(5) not null,  
  mah char(5) not null,  
  slban int,  
  Primary key (sohd,mah),  
  foreign Key (sohd) References hoadon(sohd),  
  check (slban>=0));
```

Hoặc

```
Create table ChitietHD  
  (Sohd char(5) not null,  
  mah char(5) not null,  
  slban int,  
  Constraint CT_PK Primary key (sohd,mah),  
  Constraint HD_FK foreign Key(sohd) References  
  hoadon(sohd),  
  Constraint CK_SLB check (slban>=0));
```

**EMPLOYEE**

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

**DEPARTMENT**

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

**DEPT\_LOCATIONS**

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

**PROJECT**

Pname	<u>Pnumber</u>	<u>Plocation</u>	Dnum
-------	----------------	------------------	------

**WORKS\_ON**

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

**DEPENDENT**

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

**Figure 9.2**  
Result of mapping the  
COMPANY ER schema  
into a relational database  
schema.

- NHANVIEN (MaNV, Ho, Hodem, Ten, NgaySinh, DiaChi, Gioitinh, Luong, MaNguoitheodoi, MaPhong)
- Phongban (MaPhong, TenPhong, MaNQL, NgayBDQL)
- Phong\_Diachi(MaPhong, Diachi)
- DuAn( MaDA, TenDA, DiachiDA, MaPhong)
- NV\_DA(MaNV, MaDA, SoGio)
- NguoiPT(MaNV, TenNPT, GioitinhNPT, Nsinh, Moiquanhe)

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

- **Xoá một bảng**
- `DROP TABLE <tên bảng>;`
- Ví dụ: xoá bảng khách hàng

`Drop table khách;`

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

- **Sửa đổi cấu trúc của bảng**
- ***Thêm một ràng buộc***
  - ALTER TABLE <tên bảng>  
ADD CONSTRAINT <Tên RB>;
- ***Thêm một cột***
  - ALTER TABLE <tên bảng>  
ADD <tên cột><kiểu dữ liệu>;
- ***Xoá một cột***
  - ALTER TABLE <tên bảng>  
DROP COLUMN <tên cột> [ràng buộc];
- ***Thay đổi kiểu dữ liệu của cột***
  - ALTER TABLE <tên bảng>  
ALTER COLUMN <tên cột> <kiểu dl mới>;

# CÁC LỆNH LIÊN QUAN ĐẾN CẤU TRÚC...

- Ví dụ:

- Thêm một ràng buộc CHECK

```
Alter table hang
```

```
    Add constraint check_SL check (Slton>0)
```

- Thêm cột Giới tính vào bảng Khách

```
Alter table khách
```

```
    Add GT char(3) ;
```

- Thay đổi độ rộng của cột địa chỉ trong bảng Khách

```
Alter table khách
```

```
    Alter column diachi char(40) ;
```

- Xóa bỏ cột GT trong bảng Khách

```
Alter table khách
```

```
    Drop column GT ;
```

## 4.2 Câu lệnh thao tác dữ liệu DML



# CÁC LỆNH CẬP NHẬT DỮ LIỆU

- **Bổ sung giá trị mới**

- Có thể thêm vào bảng mỗi lần một bản ghi hoặc nhiều bản ghi lấy kết quả từ một truy vấn nào đó.

- ***Bổ sung trực tiếp một bộ giá trị (them 1 bản ghi)***

INSERT INTO <tên bảng> [(tên cột 1>, <tên cột 2>,...)]

VALUES (<biểu thức 1>, <biểu thức 2>,...);

- Số lượng biểu thức và kiểu giá trị của các biểu thức phải tương ứng với số lượng và kiểu giá trị của các tên cột trong danh sách tên cột của bảng.

# CÁC LỆNH CẬP NHẬT DỮ LIỆU...

- Các giá trị phải phù hợp với các ràng buộc toàn vẹn định nghĩa trên quan hệ, trong đó có ràng buộc toàn vẹn về khoá chính (Primary key), khoá ngoại (Foreign key) và miền giá trị.
- Ví dụ : Thêm một khách hàng mới có nội dung
- Mak : K2000, Tenk : Đinh Gia Linh, Diachi : Hà Nội, Dienthoai : 048570581 vào bảng KHACH

**Insert Into KHACH**

**Values ('K2000' , N'Đinh Gia Linh', N'Hà Nội', '048570581') ;**

# CÁC LỆNH CẬP NHẬT DỮ LIỆU...


- ***Thêm một hay nhiều bộ giá trị từ truy vấn.***

INSERT INTO <tên bảng> [(<tên cột 1>, <tên cột 2>,...)]

SELECT <biểu thức 1>, <biểu thức 2>,...

FROM <danh sách các bảng nguồn>

[WHERE <điều kiện>]...

-  Nếu giá trị của các biểu thức sau từ khoá SELECT hoàn toàn phù hợp về số lượng, miền giá trị và thứ tự của các cột trong bảng thì danh sách tên các cột của bảng sau khi từ khoá INTO có thể được bỏ qua.

# CÁC LỆNH CẬP NHẬT DỮ LIỆU...

- Tạo mới một bảng với các bộ giá trị lấy từ CSDL

SELECT <biểu thức 1>, <biểu thức 2>,...

INTO <tên bảng>

FROM <danh sách các bảng nguồn>

[WHERE <điều kiện>]

GROUP BY <danh sách cột phân nhóm>

[HAVING <điều kiện>]

[ORDER BY <cột 1>[ASC | DESC], <cột 2> [ASC | DESC],...]

# CÁC LỆNH CẬP NHẬT DỮ LIỆU...

- Ví dụ : Tạo bảng mới có tên là KHHANOI gồm các khách hàng ở Hà Nội

Select mak, tenk, diachi, dienthoai

Into KHHANOI

From KHACH

Where DIACHI like N'Hà Nội';

# CÁC LỆNH CẬP NHẬT DỮ LIỆU...

- **Sửa nội dung của bộ**

UPDATE <tên bảng>

SET <tên cột 1> = <biểu thức 1>,

<tên cột 2> = <biểu thức 2>,...

<tên cột n> = <biểu thức n>

[WHERE <điều kiện>];

- Ví dụ: Sửa số lượng hàng tồn kho của tất cả các mặt hàng còn lại một nửa.

**Update HÀNG**

**Set SLTon=SLTon/2;**

# CÁC LỆNH CẬP NHẬT DỮ LIỆU...

- **Xoá bộ**

DELETE FROM <tên bảng>

[FROM <ds bang>]

[WHERE <điều kiện>];

- ☞ Các bản ghi thoả mãn điều kiện sau WHERE sẽ bị xoá khỏi bảng, nếu không có mệnh đề WHERE thì tất cả các bản ghi của bảng sẽ bị xoá khỏi bảng.

- Ví dụ: Xoá các khách hàng tại HaiPhong

**Delete from KHACH**

**Where dc like 'HaiPhong';**

## **4.3 Câu lệnh truy vấn dữ liệu SQL**



# CÂU LỆNH SELECT

- Cú pháp tổng quát của câu lệnh như sau:

SELECT [ DISTINCT][TOP n] <biểu thức 1>, <biểu thức 2>,...

[INTO tenbangmoi]

FROM <tên bảng 1>, <tên bảng 2>,...

[WHERE] <điều kiện chọn>

[ GROUP BY <tên cột 1>,<tên cột 2>,...]

[HAVING<điều kiện in kết quả>]

[ ORDER BY <tên cột 1> | <biểu thức số 1> [ASC| DESC],...];

# MỆNH ĐỀ SELECT

- **SELECT [DISTINCT] [TOP n]{\*| <biểu thức 1> [AS<tên mới 1>], <biểu thức 2> [AS <tên mới 2>],...}**

**FROM <tên bảng>;**

- VD: Cho biết tên các khách hàng của cửa hàng

```
Select      TENK  
  
From    KHACH;
```

- Nếu không muốn lấy tên các khách hàng trùng nhau thì dùng từ khoá DISTINCT.

```
Select Distinct  TENK  
  
From    KHACH;
```

- Muốn hiển thị hết tất cả các cột của bảng dùng ký tự đại diện “\*”

```
Select      *  
  
From    KHACH;
```

# MỆNH ĐỀ SELECT...

- Có thể dùng các phép toán số học +, -, \*, /, ^, %, các hàm tính toán đối với các cột kiểu số.

```
Select      MAH, SLTON*10 AS SL_TON_NEW  
  
From        HANG;
```

- Có thể thay đổi tên của các cột trong bảng kết quả ta dùng từ khoá AS

```
Select MAK AS ma_so_khach_hang  
  
From        KHACH;
```

# MỆNH ĐỀ SELECT...

- Nhận xét: Sau từ khoá SELECT ta còn có thể có từ khoá *TOP n*. Điều này cho phép chúng ta chỉ hiển thị *n* hàng trong bảng kết quả. Thông thường khi dùng TOP thì thường kết hợp với mệnh đề sắp xếp ORDER BY.
- Đưa ra 3 MAHANG đầu tiên trong danh sách.

Select TOP 3 MAH

From HANG;

Đưa ra 10 bạn có điểm CSDL cao nhất

# TỪ KHÓA WHERE

- **WHERE** <điều kiện chọn>

- Các bản ghi thoả mãn <điều kiện chọn > mới được thể hiện trong bảng kết quả.
- Điều kiện chọn có thể chứa các phép toán And, Or, Between, Not Between, like (so sánh xâu), In, is [not] Null, =, !=, <, <=, >, >=.
- Các ký tự thay thế: % thay thế cho một chuỗi ký tự  
\_ thay thế cho một ký tự bắt buộc
- Chú ý: Trong SQL hằng ký tự được bao bởi cặp ' '. Trong Access dấu \* thay thế cho một nhóm ký tự, dấu ? thay thế cho một ký tự, hằng ký tự là cặp dấu nháy kép "", hằng ngày tháng là cặp dấu # #.

# TỪ KHÓA WHERE...

- Ví dụ:
- Hiển thị các mặt hàng có số lượng tồn lớn hơn 100

```
Select  *  
From    HANG  
Where   SLTON>100;
```

- Cho hiển thị các khách hàng ở địa chỉ bắt đầu bằng chữ H

```
Select  *  
From    KHACH  
Where   diachi like 'H%' ; 'H_a%'
```

# TỪ KHÓA WHERE...

Toán tử so sánh	Diễn tả
=	So sánh bằng
<> hoặc !=	Khác
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
BETWEEN value1 AND value2	So sánh nằm trong khoảng value1 và value2
LIKE	So sánh chuỗi tương đối

# TỪ KHÓA WHERE...

- Cho phép so sánh một chuỗi với chuỗi khác (có chứa các ký tự đại diện) sử dụng toán tử LIKE
  - 1 ký tự (\_): 'H\_A'
  - Nhiều ký tự: '%OA%'

Ký tự	Mô tả
_	Thay cho một ký tự đơn
%	Thay cho một chuỗi
[]	Thay cho một ký tự đơn trong khoảng được bao bởi cặp dấu ngoặc vuông
[^]	Thay cho một ký tự đơn bất kỳ không nằm trong khoảng được bao bởi cặp dấu ngoặc vuông



# TỪ KHÓA FROM

- **FROM** <bảng1, bảng2,...>
- Dùng xác định các bảng cần có trong câu lệnh
- Cho biết các thông tin về khách hàng của các hoá đơn trong tháng 5/2010

```
Select KHACH.MaK, KHACH.TenK
```

```
From KHACH, HOADON
```

```
Where (month(NgayHD)= 5) and (year(NgayHD)=2010) and  
(HOADON.MaK = KHACH.MaK) ;
```

```
SELECT *
```

```
FROM KHACH
```

- Khách(Mak, tenk, diachi, dienthoai)
- HoaDon(SoHD, ngayHD, Mak)

# TỪ KHÓA FROM

- Dùng tên bảng và dấu chấm (.) để phân biệt 2 cột giống nhau.
- Có thể gán bí danh cho các bảng

```
Select  KH.*
```

```
From      KHACH KH,HOADON HD
```

```
Where  (month(NgayHD) = 5) and (year(NgayHD) = 2010)  
        and  (HD.MaK = KH.MaK) ;
```

# BÀI TẬP

Cho CSDL sau

NHANVIEN(MaNV, Hoten, Luong, GioiTinh, MaDV, MaNQL, CongViec)

DONVI (MaDV, TenDV, DiaChi)

1. Cho biết lương trung bình của các nhân viên nữ
2. Cho biết lương cao nhất, lương thấp nhất, lương trung bình trong công ty.
3. Cho biết trong công ty có bao nhiêu nhân viên nữ.
4. Cho biết công ty có bao nhiêu nhân viên nam là “Kỹ sư”
5. Cho biết Họ tên, Tên đơn vị, địa chỉ của các nhân viên nam
6. Cho biết các nhân viên có lương 200, 300, 500
7. Cho biết các nhân viên là người quản lý

# TỪ KHÓA ORDER BY

- **ORDER BY <tên cột> | <biểu thức> [ASC | DESC], <tên cột> | <biểu thức> [ASC | DESC],...**
- Biểu thức phải có giá trị số: nó thể hiện số thứ tự của cột trong bảng kết quả.
- Sắp xếp theo thứ tự tăng dần (ASC - mặc định là ASC), giảm dần (DESC) theo giá trị cột.
- Cho biết 3 mặt hàng đầu tiên có số lượng tồn kho cao nhất (sắp xếp giảm theo số lượng tồn kho)

```
Select TOP 3 MAH, TENH, SLTON
```

```
From  HANG
```

```
Order By  SLTON desc;
```

# TỪ KHÓA ORDER BY...

- Cho biết danh sách Top 10 sinh viên có điểm môn CSDL  $\geq 8$ ; nếu trùng điểm thì sx theo tên sv tăng dần; nếu trùng tên thì sắp xếp theo ho đệm tăng dần
- Cho biết các khách hàng được sắp xếp theo địa chỉ, nếu cùng địa chỉ thì giảm theo tên

```
Select *
```

```
From KHACH
```

```
Order By DIACHI Asc, TENK Desc;
```

# TỪ KHÓA GROUP BY –NHÓM DỮ LIỆU

- **GROUP BY** <tên cột 1>,<tên cột 2>,...

**[HAVING <điều kiện>]**

- Nhóm các bản ghi có giá trị giống nhau trên các cột chỉ định.
- HAVING theo sau GROUP BY dùng để kiểm tra điều kiện nhóm. Nhóm nào thoả mãn điều kiện sau HAVING thì mới được hiển thị.
- VD: Đưa ra số lượng khách của mỗi địa chỉ

```
Select diachi as Tinh, count(*) as SL_Khach
```

```
From KHACH
```

```
Group by diachi;
```

# TỪ KHÓA GROUP BY

- Cho biết các khách hàng có nhiều hơn 2 lần mua hàng

```
Select MAK, count(MAK) AS so_lan_mua
```

```
From HOADON
```

```
Group By MAK
```

```
Having count(MAK) > 2;
```

- Chú ý: nếu vừa có điều kiện **Where** và **Having** thì điều kiện sau **Where** được xử lý trước. Chỉ có những bộ nào thoả mãn điều kiện **Where** mới được nhóm và sau khi nhóm xong mới kiểm tra điều kiện sau **Having**.

# TỪ KHÓA GROUP BY..

- Cho biết các khách hàng có nhiều hơn 10 lần mua hàng trong tháng 2 năm 2010

```
Select Mak, count(Mak) as So_Lan_mua_hang
```

```
From HOADON
```

```
Where (month(NgayHd)=2) and (year(NgayHd)=2010)
```

```
Group By Mak
```

```
Having count(Mak)>10;
```

- Chú ý: Chỉ có các cột phân nhóm mới được thể hiện trên mệnh đề Select.



# TỪ KHÓA GROUP BY..

- Tính tổng số lượng hàng của mỗi hoá đơn bán

```
Select SOHD, sum(SLBAN) as So_luong_hang_ban  
From chitietHD  
Group By SOHD;
```

- *Khi phân nhóm dữ liệu ta có thể sử dụng các hàm phân nhóm để tính toán trên mỗi nhóm như count, sum, avg, max, min, ...*

# CÁC HÀM THAO TÁC DỮ LIỆU

## ■ Các hàm tính toán trên nhóm các bảng ghi

- COUNT (\*| <tên cột> - đếm số bản ghi có giá trị xác định tại cột được cho bởi <tên cột>
  - SUM (<biểu thức>) – tìm tổng giá trị các biểu thức
  - MIN (<biểu thức>) – tìm giá trị nhỏ nhất
  - MAX (<biểu thức>) – tìm giá trị lớn nhất
  - AVG (<biểu thức>) – tính giá trị trung bình của biểu thức dựa trên các bản ghi của các nhóm.
- *Các hàm này còn thường được đi kèm với mệnh đề GROUP BY*

# CÁC HÀM THAO TÁC DỮ LIỆU...

- Cho biết số lượng tồn lớn nhất và nhỏ nhất của các mặt hàng

```
Select Max (SLTON) , Min (SLTON)
```

```
From  HANG;
```

- Đưa ra số lượng lớn nhất của một mặt hàng trong mỗi đơn hàng.

```
Select SOHD, Max (SLBAN) as So_luong_Max
```

```
From ChitietHD
```

```
Group By SOHD;
```

- *Chú ý: các hàm SUM, MAX, MIN, AVG chỉ áp dụng với biểu thức kiểu số.*

# CÁC HÀM THAO TÁC DỮ LIỆU...

- **Các hàm tính toán trên bản ghi**

- ***Các hàm toán học.***

- **ABS (x)**      Trị tuyệt đối của x.
- **SQRT (x)**      Căn bậc hai của x (Access, SQL–Server :SQR (x) )
- **LOG (x)**      Logarit tự nhiên của x
- **EXP (x)**      Hàm mũ cơ số e của x
- **ROUND(x,n)**      Làm tròn tới n số lẻ (Access, SQL–Server, RND(x))
- **Các hàm lượng giác:** SIN, COS, TAN, ASIN, ACOS, ATAN...

# CÁC HÀM THAO TÁC DỮ LIỆU...

- $\text{ROUND}(n[,m])$ : Cho giá trị làm tròn của  $n$  (đến cấp  $m$ , mặc nhiên  $m=0$ ).
- $\text{TRUNC}(n[,m])$ : Cho giá trị  $n$  lấy  $m$  chữ số tính từ chấm thập phân.
- $\text{CEIL}(n)$ : Cho số nguyên nhỏ nhất lớn hơn hoặc bằng  $n$ .
- $\text{FLOOR}(n)$ : Cho số nguyên lớn nhất bằng hoặc nhỏ hơn  $n$ .
- $\text{POWER}(m,n)$ : Cho lũy thừa bậc  $n$  của  $m$ .

# CÁC HÀM THAO TÁC DỮ LIỆU...

- $\text{SQRT}(n)$ : Cho căn bậc 2 của  $n$ ,  $n \geq 0$
- $\text{SIGN}(n)$ : Cho dấu của  $n$ .  
 $n < 0$  có  $\text{SIGN}(n) = -1$   
 $n = 0$  có  $\text{SIGN}(n) = 0$   
 $n > 0$  có  $\text{SIGN}(n) = 1$
- $\text{ABS}(n)$ : Cho giá trị tuyệt đối
- $\text{MOD}(m, n)$ : Cho phần dư của phép chia  $m$  cho  $n$

# CÁC HÀM THAO TÁC DỮ LIỆU...

- ***Các hàm xử lý chuỗi ký tự.***
  - LEN (str) Cho chiều dài dãy ký tự
  - LEFT (str, n) Lấy n ký tự về phía trái của dãy str
  - RIGHT (str, n) Lấy n ký tự về phía phải của dãy str
  - MID (str, p, n) Lấy n ký tự của dãy str kể từ vị trí p trong dãy
  - CONCAT(char1, char2): Cho kết hợp của 2 chuỗi ký tự, tương tự như sử dụng toán tử.

# CÁC HÀM THAO TÁC DỮ LIỆU...

- ***Các hàm xử lý chuỗi ký tự.***

- **INITCAP(char):** Cho chuỗi với ký tự đầu các từ là ký tự hoa
- **LOWER(char):** Cho chuỗi ký tự viết thường (không viết hoa)
- **LPAD(char1, n [,char2]):** Cho chuỗi ký tự có chiều dài bằng n. Nếu chuỗi char1 ngắn hơn n thì thêm vào bên trái chuỗi char2 cho đủ n ký tự. Nếu chuỗi char1 dài hơn n thì giữ lại n ký tự tính từ trái sang



# CÁC HÀM THAO TÁC DỮ LIỆU...

- **LTRIM(char1, n [,char2]):** Bỏ các ký tự trống bên trái
- **NLS\_INITCAP(char):** Cho chuỗi với ký tự đầu các từ là chữ hoa, các chữ còn lại là chữ thường
- **REPLACE(char,search\_string[,replacement\_string]):** Thay tất cả các chuỗi search\_string có trong chuỗi char bằng chuỗi replacement\_string.
- **RPAD(char1, n [,char2]):**Giống LPAD(char1, n [,char2]) nhưng căn phải.
- **RTRIM(char1, n [,char2]):** Bỏ các ký tự trống bên phải
- **SOUNDEX(char):** Cho chuỗi đồng âm của char.
- **SUBSTR(char, m [,n]):** Cho chuỗi con của chuỗi char lấy từ vị trí m về phải n ký tự, nếu không chỉ n thì lấy cho đến cuối chuỗi

# CÁC HÀM THAO TÁC DỮ LIỆU...

- **TRANSLATE(char, from, to):** Cho chuỗi trong đó mỗi ký tự trong chuỗi from thay bằng ký tự tương ứng trong chuỗi to, những ký tự trong chuỗi from không có tương ứng trong chuỗi to sẽ bị loại bỏ.
- **UPPER(char):** Cho chuỗi chữ hoa của chuỗi char
- **ASCII(char):** Cho ký tự ASCII của byte đầu tiên của chuỗi char
- **INSTR(char1, char2 [,n[,m]]):** Tìm vị trí chuỗi char2 trong chuỗi char1 bắt đầu từ vị trí n, lần xuất hiện thứ m.

# CÁC HÀM THAO TÁC DỮ LIỆU...

- ***Các hàm xử lý ngày tháng và thời gian.***
  - **MONTH\_BETWEEN(d1, d2):** Cho biết số tháng giữa ngày d1 và d2.
  - **ADD\_MONTHS(d,n):** Cho ngày d thêm n tháng.
  - **NEXT\_DAY(d, char ):** Cho ngày tiếp theo ngày d có thứ chỉ bởi char.
  - **LAST\_DAY(d):** Cho ngày cuối cùng trong tháng chỉ bởi d.
  - **DATE ( ):** Cho ngày tháng năm hiện tại (oracle: SYSDATE)
  - **DAY (dd):** Cho số thứ tự ngày trong tháng của biểu thức ngày dd

# CÁC HÀM THAO TÁC DỮ LIỆU...

- ***Các hàm xử lý ngày tháng và thời gian.***
  - **MONTH (dd)** Cho số thứ tự tháng trong năm của biểu thức ngày dd
  - **YEAR (dd)** Cho năm của biểu thức ngày dd
  - **HOURL (tt)** Cho giờ trong ngày (0- 23)
  - **MINUTE (tt)** Cho số phút của thời gian tt
  - **SECONDS (tt)** Cho số giây của biểu thức giờ tt.

# CÁC HÀM THAO TÁC DỮ LIỆU...

- ***Các hàm chuyển đổi kiểu giá trị.***
  - **FORMAT (biểu thức, mẫu):** Đổi biểu thức có kiểu bất kỳ thành chuỗi theo mẫu đã cho trong tham số thứ 2. Có thể sử dụng hàm STR để thay thế.
  - Họ các hàm chuyển đổi biểu thức có kiểu bất kỳ thành một giá trị thuộc kiểu xác định: CSTR, CINT, CLNG, CSIN, CDBL,...

# CÁC PHÉP TOÁN TRÊN TẬP HỢP

- Các phép toán trên tập hợp gồm: Hợp (UNION) hoặc UNION ALL, Giao (INTERSECT), Trừ (MINUS)
- Điều kiện Các bảng có cùng số cột như nhau.
- **Phép UNION**
- VD: GS có KHACH1 lưu các khách ở miền bắc, KHACH2 là khách miền nam. Đưa ra Tên khách có Diachi ở 'Hanoi' hoặc 'BinhDuong'

```
SELECT tenk
FROM KHACH1
WHERE diachi='Hanoi'
UNION
SELECT tenk
FROM KHACH2
WHERE diachi='BinhDuong'
```

# TRUY VẤN THÔNG TIN TỪ NHIỀU BẢNG

- Các câu truy vấn trên nhiều bảng, về bản chất là giống như trên một bảng, tức là chỉ cần chỉ ra thông tin gì cần tìm và lấy từ các nguồn dữ liệu nào. Các bảng nguồn cần chỉ ra trong FROM.
- Nếu các bảng nguồn có các tên thuộc tính giống nhau thì tên thuộc tính này phải được viết tường minh trong biểu thức tìm kiếm với tên bảng đi kèm phía trước.
- Các bảng được liên kết với nhau qua phép kết nối của mệnh đề FROM hoặc thông qua điều kiện của mệnh đề WHERE. Nếu không thể hiện mối quan hệ này, kết quả sẽ là bảng tích Đề các của bảng 2.

# KẾT NỐI TỰ NHIÊN (EQUI-JOIN)

Select ...

From ...

Where <điều kiện kết nối>...

- Cho biết tên các khách hàng mua hàng trong năm 2011

```
Select KH.Tenk
```

```
From KHACH KH, HOADON HD
```

```
Where (KH.Mak = HD.Mak) and (year(NgayHD) = 2011) ;
```

- Ta có thể sử dụng phép kết nối nội **Inner join** để viết lại câu lệnh trên

```
Select KHACH.Tenk
```

```
From KHACH Inner Join HOADON on KHACH.Mak=HOADON.Mak
```

```
Where year(NgayHD) = 2011 ;
```



# KẾT NỐI NGOẠI (OUTER JOIN)

- Kết nối ngoại gồm 2 loại, kết nối trái (***Left Outer Join***), kết nối phải (***Right Outer Join***)
- Cho biết các thông tin về khách hàng và các đơn mua hàng của họ nếu có.

```
Select KHACH.*, HOADON.*
```

```
From KHACH Left Outer Join HOADON On
```

```
KHACH.Mak=HOADON.Mak
```

# KẾT NỐI NGOẠI (OUTER JOIN)

- Kết nối ngoại gồm 2 loại, kết nối trái (***Left Outer Join***), kết nối phải (***Right Outer Join***)
- **LEFT JOIN** trả về tất cả các dòng có ở bảng trái, mặc dù bảng phải không thỏa mãn. Nếu dữ liệu có ở bảng trái mà không có ở bảng phải thì dữ liệu vẫn hiển thị.
- **RIGHT JOIN** trả về tất cả các dòng có ở bảng phải, mặc dù bảng trái không thỏa mãn. Nếu dữ liệu có ở bảng phải mà không có ở bảng trái thì vẫn được hiển thị.

# KẾT NỐI NGOẠI (OUTER JOIN)

- Cho biết các thông tin về khách hàng và các đơn mua hàng của họ nếu có.

```
Select KHACH.*, HOADON.*  
From KHACH Left Outer Join HOADON On  
KHACH.Mak = HOADON.Mak
```

# TRUY VẤN LỒNG NHAU (QUERY WITH SUBQUERY)

- Một truy vấn lồng vào một truy vấn khác gọi là Subquery, Subquery cũng bao gồm các mệnh đề cơ bản như Query và có thể lồng nhau nhiều mức.
- Subquery được bao bởi hai dấu ngoặc và lồng vào truy vấn tại mệnh đề Where hoặc Having.
- *Có hai loại truy vấn lồng nhau:*
  - **Truy vấn lồng nhau phân cấp:** Mức cao hơn chỉ nhận kết quả của mức thấp. Khi thực hiện, các truy vấn cấp thấp hơn sẽ định trị trước một lần rồi cung cấp kết quả cho truy vấn cấp cao hơn.
  - **Truy vấn lồng nhau tương quan:** Mỗi một tính toán của truy vấn mức cao hơn có tham chiếu đến các truy vấn mức thấp hơn, mỗi lần tham chiếu như vậy các truy vấn mức thấp hơn phải định trị lại.

# TRUY VẤN LÒNG NHAU (QUERY WITH SUBQUERY)...

- Truy vấn lồng nhau phân cấp:
- Cho biết đầy đủ thông tin về những mặt hàng có tồn kho lớn nhất

```
Select *
```

```
From HÀNG
```

```
Where SLTON=(Select Max(SLTON) From HÀNG) ;
```

- ☞ Truy vấn con thực hiện trước và tìm ra số lượng hàng tồn lớn nhất, sau đó làm điều kiện cho truy vấn ngoài để liệt kê những mặt hàng có số lượng tồn bằng với số lượng tồn lớn nhất.


# TRUY VẤN LÒNG NHAU (QUERY WITH SUBQUERY)...

- Truy vấn lồng nhau tương quan:
- Cho biết n mặt hàng có tồn kho lớn nhất (VD n=5)

```
Select *
```

```
From HÀNG H
```

```
Where (Select count(*) From HÀNG Where  
SLTON>H.SLTON) <5;
```

-  Với mỗi mặt hàng của truy vấn ngoài, truy vấn con bên trong sẽ đếm các mặt hàng có số lượng tồn lớn hơn mặt hàng đó, nếu có ít hơn n mặt hàng có số lượng tồn lớn hơn chúng thì có nghĩa là nó nằm trong n mặt hàng lớn nhất.

# CÁC PHÉP TOÁN CÓ THỂ DÙNG ĐỐI VỚI TRUY VẤN LÒNG NHAU

- *Phép toán tập hợp In, Not in*
- Để xem một bản ghi có thuộc một bảng hay không ta dùng Subquery với toán tử **In** hoặc **Not In**.
- VD: Cho biết các khách hàng ở Hanoi mua hàng trong tháng 1/2011

```
Select *
```

```
From KHACH
```

```
Where DIACHI like 'Hanoi' and
```

```
Khach.MAK in (Select Hoadon.MAK From HOADON
```

```
Where (month(NGAYHD)=1) and
```

```
(year(NGAYHD)=2011)) ;
```

# CÁC PHÉP TOÁN CÓ THỂ DÙNG ĐỐI VỚI TRUY VẤN LÒNG NHAU

- Cho biết các mặt hàng chưa từng được bán

```
Select *
```

```
From HANG
```

```
Where Hang.MAH Not in (Select ChitietHD.MAH From  
ChitietHD);
```



# CÁC PHÉP TOÁN CÓ THỂ DÙNG ĐỐI VỚI TRUY VẤN LÒNG NHAU...

- *Phép so sánh tập hợp*
- <some, <=some, >some, >=some, =some, <>some Tương đương với:
- <any, <=any, >any, >=any, =any, <>any
- <all, <=all, >all, >=all, =all, <>all
- Chú ý: **=some** tương đương với **In** nhưng **<>some** không tương đương với **Not In**, **<>all** tương đương với **Not In**
- Liệt kê các mặt hàng không phải là mặt hàng có tồn kho lớn nhất

**Select \***

**From HANG**

**Where SLTON<some (Select SLTON From HANG) ;**

# CÁC PHÉP TOÁN CÓ THỂ DÙNG ĐỐI VỚI TRUY VẤN LÒNG NHAU...

- Cho biết số lượng trung bình một lần đặt hàng của một mặt hàng

**Select MAH, Avg(SLB)**

**From ChitietHD**

**Group By MAH;**

- Muốn biết mặt hàng có số lượng đặt hàng trung bình lớn nhất. Thường nghĩ đến dùng Max(Avg(SLB)), nhưng SQL không cho phép các hàm thống kê lồng nhau. Cách giải quyết là:

**Select MAH, Avg(SLB)**

**From ChitietHD**

**Group By MAH**

**Having Avg(SLB)>=All (Select Avg(SLB)**

**From ChitietHD**

**Group By MAH);**

# CÁC PHÉP TOÁN CÓ THỂ DÙNG ĐỐI VỚI TRUY VẤN LÒNG NHAU...

- *Phép toán kiểm tra bảng rỗng*
- $\text{Exists}(Q) = \text{True}$  nếu có ít nhất một bản ghi trong Q  
= false nếu ngược lại
- $\text{Not Exists}(Q) = \text{True}$  Q không có bộ nào  
= false nếu ngược lại
- Cho biết thông tin về các mặt hàng được bán trong tháng 7/2012  

```
Select H.*  
From HANG H  
Where Exists (Select *  
              From HOADON D, ChitietHD C  
              Where (year(NGAYHD)=2012) And  
                    (month(NGAYHD)=7) and      (D.SOHD=C.SOHD)  
                    and      (C.MAH=H.MAH) ) ;
```

# CÁC PHÉP TOÁN CÓ THỂ DÙNG ĐỐI VỚI TRUY VẤN LÒNG NHAU...

- ***Kiểm tra các bản ghi trùng nhau***
- $\text{Unique}(Q) = \text{True}$  nếu  $Q$  không có các bộ trùng nhau  
=  $\text{False}$  nếu ngược lại
- $\text{Not Unique}(Q) = \text{True}$  nếu  $Q$  có các bộ trùng nhau  
=  $\text{False}$  nếu ngược lại

# CÁC PHÉP TOÁN CÓ THỂ DÙNG ĐỐI VỚI TRUY VẤN LÒNG NHAU...

- Tìm các khách hàng chỉ mua hàng một lần

```
Select *
```

```
From KHACH K
```

```
Where Unique (Select MAK From HOADON H Where  
K.MAK=H.MAK) ;
```

- Tìm các khách hàng có ít nhất hai lần mua hàng

```
Select *
```

```
From KHACH K
```

```
Where Not Unique (Select MAK From HOADON H Where  
K.MAK=H.MAK) ;
```

## 4.4 CÁC LỆNH ĐIỀU KHIỂN

### DỮ LIỆU DCL

# CÁC LỆNH GIAO QUYỀN TRUY NHẬP CSDL

- GRANT dùng để cấp quyền cho người sử dụng trên đối tượng Cơ sở dữ liệu hoặc quyền thực thi các câu lệnh SQL SERVER.
- Cú pháp có 2 dạng như sau:
- **Dạng 1:** Cấp quyền đối với câu lệnh SQL

**GRANT** ALL | statement [,...,statementN ]  
TO account [, ...,accountN] [WITH GRANT OPTION]

- **Dạng 2:** Cấp quyền đối với các đối tượng trong CSDL

**GRANT** ALL | permission [,...,permissionN]  
ON table\_name |view\_name [(column1 [,..., columnN]))]  
|ON stored\_procedure  
TO account [, ...,accountN] [WITH GRANT OPTION]

# CÁC LỆNH GIAO QUYỀN TRUY NHẬP CSDL

- **Permission:**

- Quyền trên bảng/view: Select, Insert, Delete, Update
- Quyền trên cột của bảng/view: Select, Update
- Quyền trên các thủ tục: EXECUTE (thực thi)

- **Statement: quyền cho các câu lệnh**

- CREATE DATABASE
- CREATE TABLE
- CREATE VIEW
- CREATE PROCEDURE
- CREATE RULE
- CREATE DEFAULT
- BACKUP DATABASE
- BACKUP LOG

Được giao quyền cho người khác [WITH GRANT OPTION]



# CÁC LỆNH GIAO QUYỀN TRUY NHẬP CSDL...

- REVOKE được dùng để huỷ bỏ quyền đã được cấp phát cho người sử dụng. Câu lệnh này cũng có 2 dạng như GRANT
- Cú pháp có 2 dạng như sau:

- **Dạng 1:** Hủy quyền đối với câu lệnh SQL

**REVOKE** ALL | statement [,...,statementN]

FROM account [, ...,accountN]

- **Dạng 2:** Hủy quyền đối với các đối tượng trong CSDL

**REVOKE** ALL | permission [,... .,permissionN]}

ON table\_name | view\_name [(column [,...,columnN])]

| stored\_procedure

FROM account [, ...,accountN ]

# CÁC LỆNH GIAO QUYỀN TRUY NHẬP CSDL...

- Giao quyền SELECT, INSERT, DELETE cho GiaLinh với các bảng KHACH

```
GRANT SELECT, INSERT, DELETE  
ON KHACH  
TO GiaLinh WITH GRANT OPTION;
```

- Thu hồi lại quyền DELETE của GiaLinh đối với bảng HANG

```
REVOKE DELETE ON HANG FROM GiaLinh;
```

# BÀI TẬP CHƯƠNG 4

**4.1:** Cho cơ sở dữ liệu dùng để quản lý các chuyến đi của một công ty du lịch

## **1. DIADIEM(MADD, TENDD)**

Mỗi một địa điểm có một mã số( MADD) dùng để phân biệt với các địa điểm khác và có một tên (TENDD)

## **2. XE(BIENSO, KHTD)**

Mỗi một xe có một biển số duy nhất(BIENSO) để phân biệt với các xe khác và có số lượng khách tối đa mà xe đó có thể chở(KHTD)

## **3. HUONGDV(MAHDV, HTHDV, DCHDV)**

Mỗi một hướng dẫn viên của công ty có một mã số duy nhất để phân biệt(MAHDV), có họ tên(HTHDV) và địa chỉ của hướng dẫn viên(DCHDV)

## **4. CHUYENDI(MACD, TENCD, NGKH, NGKT, KHDK)**

Mỗi một chuyến đi có một mã số để phân biệt(MACD), thông tin về chuyến đi bao gồm: tên chuyến đi(TENCD), ngày khởi hành(NGKH), ngày kết thúc(NGKT) và số khách dự kiến(KHDK).



# BÀI TẬP CHƯƠNG 4...

## 5. CTIETCD(MACD, MADD, SNLUU)

Chi tiết của chuyến đi (MACD) là các địa điểm mà chuyến đi đó đi qua (MADD), (SNLUU) là số ngày lưu lại tại điểm du lịch đó.

## 6.HUONGDAN(MACD, MAHDV)

Ghi nhận các hướng dẫn viên(MAHDV) tham gia hướng dẫn cho chuyến đi (MACD)

## 7. KHACH(MACD, HTKH, TUOI, DCKH, DTKH)

Ghi nhận thông tin về khách hàng đăng ký vào chuyến đi(MACD), bao gồm: họ tên(HTKH), tuổi (TUOIKH), địa chỉ(DCKH) và điện thoại liên lạc của khách(DTKH)

## 8. XEPV(MACD, BIENSO)

Ghi nhận các xe (BIENSO) phục vụ cho chuyến đi (MACD)



# BÀI TẬP CHƯƠNG 4...

*Dùng câu lệnh SQL để thực hiện các yêu cầu sau:*

1. Tạo tất cả các bảng trên.
2. Cho biết danh sách các hướng dẫn viên của công ty.
3. Liệt kê đầy đủ thông tin về các điểm du lịch liên kết với công ty.
4. Cho biết đầy đủ thông tin về các địa điểm mà chuyến đi mã số CD2000 đi qua.
5. Liệt kê các lữ khách của chuyến đi CD2000.
6. Cho biết số lượng khách của chuyến đi CD1999.
7. Chuyến đi nào có số lượng khách lớn hơn số lượng dự kiến.
8. Cho biết tổng số lượng khách của tất cả các chuyến đi có ngày khởi hành trong tháng 12/2001.
9. Cho biết số ngày lưu lại trung bình, số ngày lưu lại lớn nhất, nhỏ nhất qua các điểm du lịch của chuyến đi CD2000.
10. Cho biết số lượng xe phục vụ cho chuyến đi CD2000.



# BÀI TẬP CHƯƠNG 4...

11. Điểm du lịch nào(Mã số, tên) có số ngày lưu lại lớn nhất của chuyến đi CD2000
12. Điểm du lịch nào(Mã số, tên) có số ngày lưu lại lớn hơn số ngày lưu lại trung bình qua các điểm của chuyến đi CD2000.
13. Điểm du lịch SaPa(mã số SP) có bao nhiêu chuyến đi ghé qua và khai thác được bao nhiêu ngày(tổng số ngày phục vụ).
14. Liệt kê 3 điểm du lịch đầu tiên của chuyến đi CD2000 có số ngày lưu lại lớn nhất.
15. Liệt kê 3 điểm du lịch đầu tiên của chuyến đi CD2000 có số ngày lưu lại ít nhất.
16. Liệt kê các điểm du lịch của chuyến đi CD2000 ngoại trừ điểm có số ngày lưu ít nhất
- 17.Cho biết số lượng các điểm du lịch, tổng số ngày lưu lại tại các địa điểm, số lượng các hướng dẫn viên, số lượng xe phục vụ cho từng chuyến đi có ngày khởi hành trong tháng 12/2000.



# BÀI TẬP CHƯƠNG 4...

**4.2:** Xét CSDL quản lý công chức viên chức CCVC, gồm các bảng DONVI, LOAIDV, NGACHCBVC, NGACHBACLUONG và CBVC.

1. DONVI(Madv, Tendv, loai) là quan hệ đơn vị gồm mã đơn vị, tên đơn vị, loại đơn vị.
2. LOAIDVI(Loai, Tenloaihinhh), là quan hệ về loại hình tổ chức của đơn vị gồm loại hình và tên loại hình.
3. NGACHCBVC(Ngach, Tenngach): quan hệ ngạch cán bộ viên chức gồm có ngạch và tên ngạch.
4. NGACHBACLUONG(Ngach, Bac, Hesoluong): quan hệ ngạch bậc và hệ số lương của cán bộ viên chức gồm có ngạch, bậc lương, hệ số lương.
5. CBVC(MaDV, MaCC, HT, GT, NS, Ngach, Bac, Ngayxep) là quan hệ về cán bộ viên chức gồm có Mã đơn vị, mã công chức, họ tên, giới tính, ngày tháng năm sinh, ngạch lương, bậc lương, ngày xếp lương



# BÀI TẬP CHƯƠNG 4...

*Hãy viết các câu lệnh truy vấn thông tin cho các câu hỏi sau đây:*

1. Cho danh sách CBVC theo thứ tự Alphabet của tên của các CBVC.
2. Cho danh sách CBVC có hệ số lương từ 3.0 trở lên.
3. Cho biết tổng hệ số lương của từng đơn vị.
4. Cho danh sách CBVC thuộc các đơn vị mà tên có chữ "phòng".
5. Cho danh sách CBVC thuộc các đơn vị có tên loại hình tổ chức là "hành chính"
6. Cho danh sách CBVC thuộc ngạch "cán sự" có bậc 7 trở lên, hoặc những người có hệ số lương lớn hơn 3.06
7. Cho danh sách CBVC (mà) có thời hạn xếp lương tính đến cuối năm 1998 là 3 năm trở lên đối với các ngạch chuyên viên và chuyên viên chính; hoặc 2 năm trở lên đối với các ngạch còn lại. *(Đây là danh sách CBVC đến hạn nâng lương trong năm 1998).*
8. Cho danh sách các CBVC có hệ số lương cao hơn hệ số lương của những người thuộc ngạch "cán sự".





# BÀI TẬP CHƯƠNG 4...

## 4.3: Cho lược đồ CSDL QLSV

### 1. KHOA(Makh, Vpkh)

Mỗi khoa có 1 mã số phân biệt (Makh), ta biết được vị trí của văn phòng khoa.

### 2. LOP(Malop, Makh)

Mỗi lớp có 1 mã số để phân biệt (Malop) thuộc duy nhất một khoa nào đó (Makh).

### 3. SINHVIEN(Masv, Hosv, Tensv, Nssv, Dcsv, Loptr, Malop)

Mỗi sinh viên có một mã số để phân biệt với các sinh viên khác (Masv), thông tin của từng sinh viên là họ và đệm (Hosv), tên (Tensv), năm sinh(Nssv), địa chỉ (Dcsv), có phải là lớp trưởng không (Loptr) và thuộc một lớp duy nhất nào đó (Malop)



# BÀI TẬP CHƯƠNG 4...

## 4. MONHOC(Mamh, Tenmh, LT, TH)

Mỗi môn học có một mã số duy nhất (Mamh), có một tên (Tenmh), số tiết lý thuyết (LT), số tiết thực hành (TH)

## 5. CTHOC(Malop, HK, Mamh)

Mỗi lớp học (Malop) trong từng học kỳ (HK) sẽ có một số môn học (Mamh) được giảng dạy cho lớp đó.

## 6. DIEMSV(Masv, Mamh, Lan, Diem)

Ghi nhận điểm của các môn học (Mamh) ở lần thi nào (Lan), của sinh viên(Masv).



# BÀI TẬP CHƯƠNG 4...

***Viết câu lệnh SQL để thực hiện yêu cầu sau:***

1. Cho biết danh sách lớp
2. Cho biết danh sách sinh viên lớp TH1.
3. Cho biết danh sách SV khoa CNTT
4. Cho biết chương trình học của lớp TH1
5. Điểm lần 1 môn CSDL của SV lớp TH1.
6. Điểm trung bình lần 1 môn CTDL của lớp TH1.
7. Số lượng SV của lớp TH2.
8. Lớp TH1 phải học bao nhiêu môn trong HK1 và HK2.
9. Cho biết 3 SV đầu tiên có điểm thi lần 1 cao nhất môn CSDL.
10. Cho biết sĩ số từng lớp.



# BÀI TẬP CHƯƠNG 4...

11. Khoa nào đông SV nhất.
12. Lớp nào đông nhất khoa CNTT.
13. Môn học nào mà ở lần thi 1 có số SV không đạt nhiều nhất.
14. Tìm điểm thi lớn nhất của mỗi SV cho mỗi môn học (vì SV được thi nhiều lần).
15. Điểm trung bình của từng lớp khoa CNTT ở lần thi thứ nhất môn CSDL.
16. Sinh viên nào của lớp TH1 đã thi đạt tất cả các môn học ở lần 1 của HK2.
17. Danh sách SV nhận học bổng học kỳ 2 của lớp TH2, nghĩa là đạt tất cả các môn học của học kỳ này ở lần thi thứ nhất.
18. Biết rằng lớp TH1 đã học đủ 6 học kỳ, cho biết SV nào đủ điều kiện thi tốt nghiệp, nghĩa là đã đạt đủ tất cả các môn.

