

CHƯƠNG 7

NGÔN NGỮ TRUY VẤN CÓ CẤU TRÚC - SQL

NGÔN NGỮ SQL

- Các khái niệm cơ bản
- Các câu lệnh định nghĩa dữ liệu
- Các ràng buộc

MỘT SỐ KHÁI NIỆM CƠ BẢN

Bảng, trường, bản ghi

➤ Các bảng trong một cơ sở dữ liệu

MAKHOA	TENKHOA	DIENTHOAI
DHT01	Khoa Toán cơ - Tin học	054822407
DHT02	Khoa Công nghệ thông tin	054826767
DHT03	Khoa Vật lý	054823463
DHT04	Khoa Hoá học	
...	...	

MALOP	TENLOP	KHOA	HEDAOTAO	NAMNHAPHOC	SISO	MAKHOA
C24101	Toán K24	24	Chính quy 2000		5	DHT01
C24102	Tin K24	24	Chính quy 2000		8	DHT02
C24103	Lý K24	24	Chính quy 2000		7	DHT03
C24301	Sinh K24	24	Chính quy 2000		5	DHT05

MASV	HODEM	TEN	NGAYSINH	GIOTINH	NOISINH	MALOP
0241010001	Ngô Thị Nhật	Anh	Nov 27 1982	0	Quảng Ninh, Quảng Bình	C24101
0241010002	Nguyễn Thị Ngọc	Anh	Mar 21 1983	0	Tân Kỳ, Nghệ An	C24101
0241010003	Ngô Việt	Bắc	May 11 1982	1	Yên Khánh, Ninh Bình	C24101
0241010004	Nguyễn Đình	Bình	Oct 6 1982	1	Huế	C24101
0241010005	Hồ Đăng	Chiến	Jan 20 1982	1	Phong Điền, TTHuế	C24101
0241020001	Nguyễn Tuấn	Anh	Jul 15 1979	1	Đo Linh, Quảng Trị	C24102
0241020002	Trần Thị Kim	Anh	Nov 4 1982	0	Phong Điền, TTHuế	C24102
0241020003	Võ Đức	Ấn	May 24 1982	1	Huế	C24102
0241020004	Nguyễn Công	Bình	Jun 6 1979	1	Thăng Bình, Quảng Nam	C24102
0241020005	Nguyễn Thanh	Bình	Apr 24 1982	1	Huế	C24102
...

Bảng (Table)

- Bảng là đối tượng được tổ chức và lưu trữ dữ liệu.
- Một cơ sở dữ liệu bao gồm nhiều bảng và mỗi bảng được xác định duy nhất bởi tên bảng.
- Một bảng bao gồm tập các dòng và các cột: mỗi một dòng trong bảng biểu diễn cho một thực thể.
- Ví dụ: Bảng SINHVIEN
 - Mỗi một dòng trong bảng sinh viên tương ứng với một sinh viên.
 - Mỗi một cột biểu diễn cho một tính chất của thực thể (ví dụ cột Ngaysinh biểu diễn ngày sinh của sinh viên được lưu trữ trong bảng)

Bảng (Table)

- Một bảng bao gồm các yếu tố:
 - Tên của bảng: được sử dụng để xác định duy nhất một bảng trong cơ sở dữ liệu.
 - Cấu trúc của bảng: Tập các cột trong bảng, mỗi một cột trong bảng được xác định bởi tên cột và phải có một kiểu dữ liệu.
 - Dữ liệu của bảng: tập các hàng (các bản ghi) hiện có trong bảng.

Khóa chính và khóa ngoài

- Mỗi bảng phải có một cột hoặc một tập các cột mà giá trị dữ liệu của nó xác định duy nhất một dòng trong một tập các dòng của bảng. Tập một hoặc nhiều cột có tính chất này gọi là khóa của bảng.
- Việc chọn khoá của bảng có vai trò quan trọng trong việc thiết kế và cài đặt cơ sở dữ liệu quan hệ. Các dòng dữ liệu trong một bảng phải có giá trị khác nhau trên khoá.
 - Ví dụ: Bảng Môn học khóa chính là gì?

Khóa chính và khóa ngoài

MAKHOA	TENKHOA	DIENTHOAI
DHT01	Khoa Toán cơ - Tin học	054822407
DHT02	Khoa Công nghệ thông tin	054826767
DHT03	Khoa Vật lý	054823462
...

MALOP	TENLOP	KHOA	HEDAOTAO	NAMNHAPHOC	SISO	MAKHOA
C24101	Toán K24	24	Chính quy	2000	5	DHT01
C25101	Toán K25	25	Chính quy	2001	5	DHT01
C25102	Tin K25	25	Chính quy	2001	6	DHT02
C24102	Tin K24	24	Chính quy	2000	8	DHT02
...

Bảng LOP

Khóa chính và khóa ngoài

- Khái niệm khóa ngoài (foreign key) trong cơ sở dữ liệu quan hệ được sử dụng để biểu diễn mối quan hệ giữa các bảng dữ liệu. Một hay một tập các cột trong bảng mà giá trị của nó được xác định từ khóa chính từ một bảng khác được gọi là khóa ngoài.
- Câu hỏi: trong bảng lớp ở hình trên đâu là khóa ngoài?

Các ngôn ngữ thành phần của SQL

- SQL cung cấp cả hai phần:
 - Ngôn ngữ định nghĩa dữ liệu (DDL - Data Definition Language).
 - Ngôn ngữ thao tác dữ liệu (DML - Data Manipulation Language).

Các ngôn ngữ thành phần của SQL

- DDL: Các câu lệnh DDL trong SQL là biểu thức được xây dựng chung quanh những mệnh đề sau:
 - CREATE: Dùng để tạo mới các bảng, các trường và các chỉ mục.
 - ALTER: Dùng để sửa đổi các bảng bằng cách thêm trường, thay đổi định nghĩa của các trường.
 - DROP: Dùng để xóa các bảng và chỉ mục khỏi CSDL.

Các ngôn ngữ thành phần của SQL

- DML: Các câu lệnh DML là các biểu thức được xây dựng dựa trên các mệnh đề sau:
 - SELECT: Dùng để truy vấn CSDL để lấy được những bản ghi thoả mãn những tiêu chuẩn nào đó.
 - INSERT: Dùng để chèn một nhóm dữ liệu vào CSDL thông qua một thao tác.
 - UPDATE: Dùng để thay đổi giá trị của những trường, những bản ghi cụ thể.
 - DELETE: Dùng để loại bỏ những bản ghi ra khỏi CSDL.

Các kiểu dữ liệu

- CHAR (n) - Kiểu chuỗi với độ dài cố định
- NCHAR (n) - Kiểu chuỗi với độ dài cố định hỗ trợ UNICODE
- VARCHAR (n) - Kiểu chuỗi với độ dài chính xác
- NVARCHAR (n) - Kiểu chuỗi với độ dài chính xác hỗ trợ UNICODE
- INTEGER - Số nguyên có giá trị từ -2³¹ đến 2³¹ - 1
- INT - Như kiểu Integer
- TINYTINT - Số nguyên có giá trị từ 0 đến 255.
- SMALLINT - Số nguyên có giá trị từ -2¹⁵ đến 2¹⁵ - 1

Các kiểu dữ liệu

- BIGINT Số nguyên có giá trị từ -2^{63} đến $2^{63}-1$.
- NUMERIC (p,s) Kiểu số với độ chính xác cố định.
- DECIMAL (p,s) Tương tự kiểu Numeric.
- FLOAT Số thực có giá trị từ $-1.79E+308$ đến $1.79E+308$.
- REAL Số thực có giá trị từ $-3.40E + 38$ đến $3.40E + 38$.
- MONEY Kiểu tiền tệ BIT Kiểu bit (có giá trị 0 hoặc 1).
- DATETIME Kiểu ngày giờ (chính xác đến phần trăm của giây)

Các kiểu dữ liệu

- **SMALLDATETIME** - Kiểu ngày giờ (chính xác đến phút).
- **BINARY** - Dữ liệu nhị phân với độ dài cố định (tối đa 8000 bytes).
- **VARBINARY** - Dữ liệu nhị phân với độ dài chính xác (tối đa 8000 bytes).
- **IMAGE** - Dữ liệu nhị phân với độ dài chính xác (tối đa 2,147,483,647 bytes).
- **TEXT** - Dữ liệu kiểu chuỗi với độ dài lớn (tối đa 2,147,483,647 ký tự)
- **NTEXT** - Dữ liệu kiểu chuỗi với độ dài lớn và hỗ trợ.
- **UNICODE** (tối đa 1,073,741,823 ký tự)

Câu lệnh SQL

○ SQL chuẩn bao gồm khoảng 40 câu lệnh

Câu lệnh

Chức năng

Thao tác dữ liệu

SELECT

Truy xuất dữ liệu

INSERT

Bổ sung dữ liệu

UPDATE

Cập nhật dữ liệu

DELETE

Xoá dữ liệu

TRUNCATE

Xoá toàn bộ dữ liệu trong bảng

Câu lệnh SQL

CREATE TABLE	Tạo bảng
DROP TABLE	Xoa bảng
ALTER TABLE	Sửa đổi bảng
CREATE VIEW	Tạo khung nhìn
ALTER VIEW	Sửa đổi khung nhìn
DROP VIEW	Xoá khung nhìn
CREATE INDEX	Tạo chỉ mục
DROP INDEX	Xoá chỉ mục
CREATE SCHEMA	Tạo lược đồ cơ sở dữ liệu
DROP SCHEMA	Xoá lược đồ cơ sở dữ liệu
CREATE PROCEDURE	Tạo thủ tục lưu trữ
ALTER PROCEDURE	Sửa đổi thủ tục lưu trữ
DROP PROCEDURE	Xoá thủ tục lưu trữ

Câu lệnh SQL

CREATE FUNCTION	Tạo hàm (do người sử dụng định nghĩa)
ALTER FUNCTION	Sửa đổi hàm
DROP FUNCTION	Xoá hàm
CREATE TRIGGER	Tạo trigger
ALTER TRIGGER	Sửa đổi trigger
DROP TRIGGER	Xoá trigger

Câu lệnh SQL

Điều khiển truy cập

GRANT	Cấp phát quyền cho người sử dụng
REVOKE	Thu hồi quyền từ người sử dụng

Quản lý giao tác

COMMIT	Ủy thác (kết thúc thành công) giao tác
ROLLBACK	Quay lui giao tác
SAVE TRANSACTION	Đánh dấu một điểm trong giao tác

Lập trình

DECLARE	Khai báo biến hoặc định nghĩa con trỏ
OPEN	Mở một con trỏ để truy xuất kết quả truy vấn
FETCH	Đọc một dòng trong kết quả truy vấn (sử dụng con trỏ)
CLOSE	Đóng một con trỏ
EXECUTE	Thực thi một câu lệnh SQL

NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU DDL

Các ngôn ngữ giao tiếp

- **Ngôn ngữ định nghĩa dữ liệu** (*Data Definition Language - DDL*): cho phép khai báo cấu trúc bảng, các mối quan hệ và các ràng buộc.
- **Ngôn ngữ thao tác dữ liệu** (*Data Manipulation Language - DML*): cho phép thêm, xóa, sửa dữ liệu.
- **Ngôn ngữ truy vấn dữ liệu** (*Structured Query Language – SQL*): cho phép truy vấn dữ liệu.
- **Ngôn ngữ điều khiển dữ liệu** (*Data Control Language – DCL*): khai báo bảo mật thông tin, cấp quyền và thu hồi quyền khai thác trên cơ sở dữ liệu.

Ngôn ngữ định nghĩa dữ liệu

- **Lệnh tạo cơ sở dữ liệu CREATE**
- **Lệnh tạo bảng (CREATE)**
- **Lệnh sửa cấu trúc bảng (ALTER)**
- **Lệnh xóa bảng (DROP)**

Ngôn ngữ định nghĩa dữ liệu (DDL)

- **Tạo một cơ sở dữ liệu**
 - Cú pháp: Create Database <Tên CSDL>
 - Ví dụ: Tạo một cơ sở dữ liệu quản lý thư viện
Create Database QLTV

Ngôn ngữ định nghĩa dữ liệu (DDL)

- **Tạo bảng**

- Cú pháp:

```
CREATE TABLE <tên_bảng>
(
    <tên_cột1>    <kiểu_dữ_liệu> [not null],
    <tên_cột2>    <kiểu_dữ_liệu> [not null],
    ...
    <tên_cộtn>    <kiểu_dữ_liệu> [not null],
    [CONSTRAINT mệnh đề ]
)
```

Trong đó: CONSTRAINT cho phép ta khai báo các ràng buộc dữ liệu

Ngôn ngữ định nghĩa dữ liệu (DDL)

- **Tạo bảng**

- Tên của bảng được xác định ngay sau câu lệnh:

- CREATE TABLE.
 - Mỗi HQTCSDL có một quy tắc đặt tên riêng.
 - Tên bảng phải bắt đầu bằng một chữ cái, có dưới 30 kí tự (chữ cái, chữ số, và dấu '_').
 - Tên bảng phải khác tên gọi khác của bảng hay của khung nhìn và với tên gọi đã dành riêng của SQL.
 - Không phân biệt hoa, thường.

Ngôn ngữ định nghĩa dữ liệu (DDL)

- **Tạo bảng**

- Các thuộc tính:

- Tên thuộc tính
 - Loại dữ liệu và độ dài
 - Các ràng buộc có liên quan
 - Mỗi thuộc tính của một định nghĩa kết thúc bằng dấu ‘,’.

Tạo bảng

■ Một số kiểu dữ liệu

Kiểu dữ liệu	SQL Server
Chuỗi ký tự	varchar(n), char(n), nvarchar(n), nchar(n)
Số	tinyint, smallint, int, numeric (m,n), decimal(m,n), float, real, smallmoney, money
Ngày tháng	smalldatetime, datetime
Luận lý	bit

Ngôn ngữ định nghĩa dữ liệu (DDL)

- **Tạo bảng**

- Ví dụ: Tạo bảng **KHACHHANG** (MAKH, HOTEN, DCHI, SODT, NGSINH, DOANH SO, NGDK, CCCD).

Tạo bảng

Create table KHACHHANG

```
(  
    MAKH      char(4) primary key,  
    HOTEN     varchar(40),  
    DCHI      varchar(50),  
    SODT      varchar(20),  
    NGSINH    smalldatetime,  
    DOANHISO  money,  
    NGDK      smalldatetime,  
    CCCD      varchar(12)  
)
```

Tạo bảng

- Tạo cơ sở dữ liệu, tạo bảng cho *CSDL quản lý bán hàng* gồm có các quan hệ sau:

KHACHHANG (MAKH, HOTEN, DCHI, SODT, NGSINH, DOANHISO, NGDK, CMND)

NHANVIEN (MANV,HOTEN, NGVL, SODT)

SANPHAM (MASP,TENSP, DVT, NUOCSX, GIA)

HOADON (SOHD, NGHD, MAKH, MANV, TRIGIA)

CTHD (SOHD,MASP,SL)

Ngôn ngữ định nghĩa dữ liệu (DDL)

- **Tạo bảng**

- Các ràng buộc (constraint) gồm

- NOT NULL.
 - UNIQUE.
 - PRIMARY KEY.
 - FOREIGN KEY (Referential).
 - CHECK

Ngôn ngữ định nghĩa dữ liệu (DDL)

- **Tạo bảng**

- Các ràng buộc (constraint) gồm

- ***Cú pháp:***

- [CONSTRAINT constraint_name Loại ràng buộc (colname1, [colname2], ..., [colname n]))].

Các ràng buộc

- **NOT NULL - Không rỗng**

- Khi có mệnh đề: NOT NULL có trong định nghĩa của một cột thì ta bắt buộc thuộc tính này phải có giá trị. Nếu ta không chỉ thị gì trong định nghĩa của thuộc tính thì nó có thể có hoặc không có giá trị.

```
CREATE TABLE NHANVIEN (  
    MaNV NUMBER(10) NOT NULL,  
    TenNV CHAR(30) )
```

Các ràng buộc

- **UNIQUE-Duy nhất**

- Chỉ ra ràng buộc duy nhất, các giá trị của cột chỉ trong mệnh đề UNIQUE trong các row của table phải có giá trị khác biệt. Giá trị null là cho phép nêu UNIQUE dựa trên một cột.

```
CREATE TABLE NHANVIEN (  
    MaNV NUMBER(10) NOT NULL,  
    TenNV CHAR(30),  
    DiachiNV CHAR(50),  
    CCCD NUMBER(12),  
    CONSTRAINT UNI_CCCD UNIQUE(CCCD))
```

Các ràng buộc

- PRIMARY KEY
 - Chỉ ra ràng buộc duy nhất (giống UNIQUE), tuy nhiên khoá là dạng khoá UNIQUE cấp cao nhất.
 - Một table chỉ có thể có một PRIMARY KEY. Các giá trị trong PRIMARY KEY phải NOT NULL.

Các ràng buộc

- PRIMARY KEY.
- FOREIGN KEY

Các ràng buộc

- Ví dụ

```
CREATE TABLE NHANVIEN (  
    MaNV char(10) NOT NULL primary key,  
    TenNV char(30),  
    DiachiNV char(50) )
```

Các ràng buộc

- Ví dụ

Create table CTHD

(

SOHD int foreign key

references HOADON(SOHD),

MASP char(4) foreign key

references SANPHAM(MASP),

SL int,

constraint PK_CTHD primary key (SOHD,MASP)

)

Các ràng buộc

- **DEFAULT-Mặc định**

- Ràng buộc DEFAULT được sử dụng để quy định giá trị mặc định cho một cột. Giá trị này sẽ tự động gán cho cột nếu người sử dụng không nhập vào khi bổ sung bản ghi.

Các ràng buộc

- **DEFAULT-Mặc định**

- Ví dụ:

- CREATE TABLE NHANVIEN (
MaNV char(10) primary key,
TenNV char(30) not null,
Gioitinh char(3) DEFAULT 'Nam')

Các ràng buộc

- **DEFAULT-Mặc định**

- Ví dụ:

- CREATE TABLE NHANVIEN (
MaNV char(10) primary key,
TenNV char(30) not null,
Gioitinh char(3) DEFAULT 'Nam')

Ngôn ngữ định nghĩa dữ liệu (DDL)

▪ **Lệnh sửa cấu trúc bảng (ALTER)**

- Thêm thuộc tính.
- Sửa kiểu dữ liệu của thuộc tính.
- Xoá thuộc tính.
- Thêm ràng buộc toàn vẹn.
- Xoá ràng buộc toàn vẹn



Ngôn ngữ định nghĩa dữ liệu (DDL)

▪ **Lệnh sửa cấu trúc bảng (ALTER)**

- Thêm thuộc tính.
- Sửa kiểu dữ liệu của thuộc tính.
- Xoá thuộc tính.
- Thêm ràng buộc toàn vẹn.
- Xoá ràng buộc toàn vẹn



Ngôn ngữ định nghĩa dữ liệu (DDL)

▪ Lệnh sửa cấu trúc bảng (ALTER)

– Thêm thuộc tính.

- Cú pháp: **ALTER TABLE <Tên bảng> ADD <Tên thuộc tính> kiểu dữ liệu.**
- Ví dụ: thêm cột Ghi_chu vào bảng khách hàng
*ALTER TABLE KHACHHANG ADD GHI_CHU
varchar(20).*

Ngôn ngữ định nghĩa dữ liệu (DDL)

▪ Lệnh sửa cấu trúc bảng (ALTER)

– Thay đổi kiểu của thuộc tính

- Cú pháp: **ALTER TABLE <Tên_bảng> ALTER COLUMN Tên_cột Kiểu_mới.**

Lưu ý: Không phải sửa bất kỳ kiểu dữ liệu nào cũng được

Ngôn ngữ định nghĩa dữ liệu (DDL)

▪ Lệnh sửa cấu trúc bảng (ALTER)

– Thay đổi kiểu của thuộc tính

- Cú pháp: **ALTER TABLE <Tên_bảng> ALTER COLUMN Tên_cột Kiểu_mới.**

Lưu ý: Không phải sửa bất kỳ kiểu dữ liệu nào cũng được

Thay đổi kiểu thuộc tính

- **Ví dụ:** Sửa Cột Ghi_chu thành kiểu dữ liệu varchar(50).

*ALTER TABLE KHACHHANG ALTER COLUMN
GHI_CHU varchar(50)*

- Nếu sửa kiểu dữ liệu của cột Ghi_chu thành varchar(5), mà trước đó đã nhập giá trị cho cột Ghi_chu có độ dài hơn 5 ký tự thì không được phép.
- Hoặc sửa từ kiểu chuỗi ký tự sang kiểu số, ...

Ngôn ngữ định nghĩa dữ liệu (DDL)

▪ Lệnh sửa cấu trúc bảng (ALTER)

– Xóa thuộc tính.

- Cú pháp: *ALTER TABLE tên_bảng DROP COLUMN tên_cột.*

- Ví dụ: xóa cột Ghi_chu trong bảng KHACHHANG

ALTER TABLE NHANVIEN DROP COLUMN Ghi_chu.

Ngôn ngữ định nghĩa dữ liệu (DDL)

- **Lệnh sửa cấu trúc bảng (ALTER)**
 - Xóa đối tượng
 - Cú pháp: **DROP** <Object_name>

Ngôn ngữ định nghĩa dữ liệu (DDL)

▪ **Lệnh sửa cấu trúc bảng (ALTER)**

– Xóa đối tượng

- Cú pháp: **DROP <Object_name>**

– Xóa bảng

- Cú pháp: **DROP TABLE Tên_bảng**

- Ví dụ: xóa bảng KHACHHANG.

DROP TABLE KHACHHANG

- **Lưu ý:** khi muốn xóa một bảng phải xóa tất cả những khóa ngoại tham chiếu tới bảng đó trước.

Ngôn ngữ định nghĩa dữ liệu (DDL)

▪ Lệnh sửa cấu trúc bảng (ALTER)

– Thêm ràng buộc.

- Cú pháp: ALTER TABLE <Tên bảng> ADD
CONSTRAINT <Tên ràng buộc> <Loại ràng buộc>
(Tên cột).

Lệnh sửa cấu trúc bảng (ALTER)

- Cú pháp

**ALTER TABLE <tên_bảng>
ADD CONSTRAINT
<tên_ràng_buộc>**

UNIQUE tên_cột

PRIMARY KEY (tên_cột)

FOREIGN KEY (tên_cột)

REFERENCES tên_bảng
(cột_là_khóa_chính)

CHECK (tên_cột điều_kiện)⁵²

Lệnh sửa cấu trúc bảng (ALTER)

- **Ví dụ**

- ALTER TABLE NHANVIEN ADD CONSTRAINT PK_NV
PRIMARY KEY (MANV)
- ALTER TABLE CTHD ADD CONSTRAINT FK_CT_SP
FOREIGN KEY (MASP) REFERENCES SANPHAM(MASP)
- ALTER TABLE SANPHAM ADD CONSTRAINT
CK_GIA CHECK (GIA >=500)
- ALTER TABLE KHACHHANG ADD CONSTRAINT UQ_KH
UNIQUE (CCCD)

Ngôn ngữ định nghĩa dữ liệu (DDL)

- **Lệnh sửa cấu trúc bảng (ALTER)**
 - Xóa ràng buộc toàn vẹn.
 - Cú pháp: *ALTER TABLE tên_bảng DROP CONSTRAINT tên_ràng_buộc.*

Ngôn ngữ định nghĩa dữ liệu (DDL)

- **Lệnh sửa cấu trúc bảng (ALTER)**
 - Xóa ràng buộc toàn vẹn.
 - Ví dụ:
 - Alter table CTHD drop constraint FK_CT_SP
 - Alter table SANPHAM drop constraint ck_gia
- **Lưu ý:** đối với ràng buộc khóa chính, muốn xóa ràng buộc này phải xóa hết các ràng buộc khóa ngoại tham chiếu tới nó

Ngôn ngữ thao tác dữ liệu

- Gồm các lệnh:
 - Lệnh thêm dữ liệu (INSERT)
 - Lệnh sửa dữ liệu (UPDATE)
 - Lệnh xóa dữ liệu (DELETE)

Thêm dữ liệu

- **Cú pháp:**

INSERT INTO tên_bảng (cột1,...,cộtn) VALUES
(giá_trị_1,...., giá_trị_n)

INSERT INTO tên_bảng VALUES (giá_trị_1,
giá_trị_2,..., giá_trị_n)

- **Ví dụ:**

- insert into SANPHAM values('BC01','But chi', 'cay', 'Singapore', 3000)
- insert into SANPHAM(masp,tensp,dvt,nuocsx,gia) values ('BC01','But chi','cay','Singapore',3000)

Sửa dữ liệu

- **Cú pháp**

UPDATE tên_bảng

SET cột_1 = giá_trị_1, cột_2 = giá_trị_2

[WHERE điều_kiện]

- **Lưu ý:** cẩn thận với các lệnh xóa và sửa, nếu không có điều kiện ở WHERE nghĩa là xóa hoặc sửa tất cả.
- **Ví dụ:** Tăng giá 10% đối với những sản phẩm do “Trung Quoc” sản xuất

```
UPDATE SANPHAM
```

```
SET Gia = Gia*1.1
```

```
WHERE Nuocsx='Trung Quoc'
```

Xóa dữ liệu

- **Cú pháp**

DELETE FROM tên_bảng [WHERE điều_kiện]

- **Ví dụ:**

- Xóa toàn bộ nhân viên

DELETE FROM NHANVIEN

- Xóa những sản phẩm do Trung Quốc sản xuất có giá thấp hơn 10000

DELETE FROM SANPHAM

WHERE (Gia < 10000) and (Nuocsx = 'Trung Quoc')

Ngôn ngữ truy vấn dữ liệu có cấu trúc

- Câu truy vấn tổng quát
- Truy vấn đơn giản
- Phép kết
- Đặt bí danh, sử dụng *, distinct
- Các toán tử
- Câu truy vấn con (subquery)
- Phép chia
- Hàm tính toán, gom nhóm

Câu truy vấn tổng quát

- **Cú pháp**

SELECT [DISTINCT] * | tên_cột | hàm

FROM bảng

[WHERE điều_kiện]

[GROUP BY tên_cột]

[HAVING điều_kiện]

[ORDER BY tên_cột ASC | DESC]

Truy vấn đơn giản

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

Truy vấn đơn giản

- **Ví dụ 1:** Cho biết mã lớp, tên lớp và hệ đào tạo của các lớp hiện có:

```
SELECT malop, tenlop, hedaotao
```

```
FROM lop
```

- **Ví dụ 2:** `SELECT * FROM khoa`

Truy vấn đơn giản

- **Lưu ý:**

- Nếu truy vấn được thực hiện trên nhiều bảng/khung nhìn và trong các bảng có các trường trùng tên thì tên của những trường này nếu xuất hiện trong danh sách chọn phải được viết dưới dạng:

tên_bảng.tên_trường

```
SELECT malop, tenlop, lop.makhoa, tenkhoa FROM  
lop, khoa WHERE lop.makhoa = khoa.makhoa
```


Truy vấn đơn giản

- **Thay đổi tiêu đề các cột**

tiêu_đề_cột = tên_trường

hoặc

tên_trường AS tiêu_đề_cột hoặc tên_trường
tiêu_đề_cột

- **Ví dụ:** `SELECT 'Mã lớp'= malop, tenlop 'Tên lớp',
khoa AS 'Khoá' FROM lop`

Truy vấn đơn giản

- Loại bỏ các dòng dữ liệu trùng nhau trong kết quả truy vấn ta chỉ định thêm từ khóa **DISTINCT** ngay sau từ khoá **SELECT**.

- Ví dụ:



Truy vấn đơn giản

- **TOP:** Giới hạn số lượng dòng trong kết quả truy vấn
- **Ví dụ:** Hiển thị họ tên và ngày sinh của 5 sinh viên đầu tiên trong danh sách:

```
SELECT TOP 5 hodem, ten, ngaysinh FROM  
sinhvien
```

Truy vấn đơn giản

- Có thể chỉ định số lượng các dòng cần hiển thị theo tỷ lệ phần trăm bằng cách sử dụng thêm từ khoá PERCENT:
- **Ví dụ:** Hiển thị họ tên và ngày sinh của 10% số lượng sinh viên hiện có trong bảng SINHVIEN

```
SELECT TOP 10 PERCENT    hodem, ten,  
ngaysinh FROM sinhvien
```

Truy vấn đơn giản

- Chỉ định điều kiện truy vấn dữ liệu:
 - Mệnh đề WHERE trong câu lệnh SELECT được sử dụng nhằm xác định các điều kiện đối với việc truy xuất dữ liệu.
 - Sau mệnh đề WHERE là một biểu thức logic và chỉ những dòng dữ liệu nào thoả mãn điều kiện được chỉ định mới được hiển thị trong kết quả truy vấn.

Truy vấn đơn giản

- Ví dụ: Hiển thị danh sách các môn học có số đơn vị học trình lớn hơn 3:
○ `SELECT * FROM monhoc WHERE sodvht>3`

Truy vấn đơn giản

- Trong mệnh đề WHERE thường sử dụng:
 - Các toán tử kết hợp điều kiện (AND, OR).
 - Các toán tử so sánh
 - Kiểm tra giới hạn của dữ liệu (BETWEEN/ NOT BETWEEN) .
 - Danh sách
 - Kiểm tra khuôn dạng dữ liệu.
 - Các giá trị NULL

Truy vấn đơn giản

- Ví dụ 1:
 - `SELECT * FROM monhoc WHERE sodvht>3`
- Ví dụ 2: Cho biết mã, họ tên và ngày sinh của các sinh viên có tên là Anh và có tuổi nhỏ hơn hoặc bằng 20

```
SELECT masv,hodem,ten,ngaysinh FROM  
sinhvien WHERE (ten='Anh') AND  
(YEAR(GETDATE())-YEAR(ngaysinh)<=20)
```


Truy vấn đơn giản

- Các toán tử so sánh

Toán tử	ý nghĩa
=	Bằng
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
≠	Khác
!>	Không lớn hơn
!<	Không nhỏ hơn

Truy vấn đơn giản

- Ví dụ 1:
 - `SELECT * FROM monhoc WHERE sodvht>3`
- Ví dụ 2: Cho biết mã, họ tên và ngày sinh của các sinh viên có tên là Anh và có tuổi nhỏ hơn hoặc bằng 20

```
SELECT masv,hodem,ten,ngaysinh FROM  
sinhvien WHERE (ten='Anh') AND  
(YEAR(GETDATE())-YEAR(ngaysinh)<=20)
```

Truy vấn đơn giản

- Kiểm tra giới hạn của dữ liệu:

Để kiểm tra xem giá trị dữ liệu nằm trong (ngoài) một khoảng nào đó, ta sử dụng toán tử BETWEEN (NOT BETWEEN) như sau:

Cách sử dụng	Ý nghĩa
giá_trị BETWEEN a AND b	$a \leq \text{giá_trị} \leq b$
giá_trị NOT BETWEEN a AND b	$(\text{giá_trị} < a) \text{ AND } (\text{giá_trị} > b)$

Truy vấn đơn giản

- Ví dụ 1: Cho biết họ tên và tuổi của các sinh viên có tên là Bình và có tuổi nằm trong khoảng từ 20 đến 22?

```
SELECT hodem,ten,year(getdate())-  
year(ngaysinh) AS tuoi FROM  
sinhvien WHERE ten='Bình' AND  
YEAR(GETDATE())-YEAR(ngaysinh)  
BETWEEN 20 AND 22
```

Truy vấn đơn giản

- Tìm masp, tensp do “Trung Quoc” sản xuất có giá từ 20000 đến 30000

Select masp,tensp

From SANPHAM

Where nuocsx='Trung Quoc'

and gia between 20000 and 30000

Truy vấn đơn giản

- Từ khoá IN được sử dụng khi ta cần chỉ định điều kiện tìm kiếm dữ liệu cho câu lệnh SELECT là một danh sách các giá trị. Sau IN (hoặc NOT IN) có thể là một danh sách các giá trị hoặc là một câu lệnh SELECT khác.
- Ví dụ: Để biết danh sách các môn học có số đơn vị học trình là 2, 4 hoặc 5.

○ `SELECT * FROM monhoc WHERE sodvht=2
OR sodvht=4 OR sodvht=5.`

Tương đương:

`SELECT * FROM monhoc WHERE sodvht IN (2,4,5)`

Đặt bí danh, sử dụng *, distinct

- Đặt bí danh – Alias: cho thuộc tính và quan hệ: tên_cũ AS tên_mới.
 - Select manv, hoten as [ho va ten] From NHANVIEN
- Liệt kê tất cả các thuộc tính của quan hệ:
 - Select * from Nhanvien.
 - Select NHANVIEN.* from NHANVIEN
- Distinct: trùng chỉ lấy một lần.
 - Select distinct nuocsx from SANPHAM.
- Sắp xếp kết quả hiển thị: Order by
 - Select * from SANPHAM order by nuocsx, gia DESC

Toán tử truy vấn

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

Toán tử truy vấn

- **IS NULL, IS NOT NULL**
- **Toán tử so sánh, phép toán**
 - Select gia*1.1 as [gia ban] from SANPHAM where nuocsx<>'Viet Nam'
 - Select * from SANPHAM where (gia between 20000 and 30000) OR (nuocsx='Viet Nam')
- **Toán tử IN, NOT IN**
 - Select * from SANPHAM where masp NOT IN ('BB01','BB02','BB03')

Toán tử truy vấn

- Dữ liệu trong một cột cho phép NULL sẽ nhận giá trị NULL trong các trường hợp sau:
 - Nếu không có dữ liệu được nhập cho cột và không có mặc định cho cột hay kiểu dữ liệu trên cột đó.
 - Người sử dụng trực tiếp đưa giá trị NULL vào cho cột đó.

Toán tử so sánh

- Trong mệnh đề WHERE để kiểm tra giá trị của một cột có NULL hay không ta dùng câu lệnh:
 - WHERE tên_cột IS NULL
 - WHERE tên_cột IS NOT NULL
- Ví dụ:
 - Select sohd from HOADON where makh is Null
 - Select * from HOADON where makh is Not Null

Toán tử so sánh

- **Toán tử LIKE**

- So sánh chuỗi tương đối
- Cú pháp: s LIKE p, p có thể chứa % hoặc _
- % : thay thế một chuỗi ký tự bất kỳ
- _ : thay thế một ký tự bất kỳ

- **Ví dụ:** Select masp,tensp from SANPHAM
where masp like 'B%01'

Câu truy vấn con

- **In hoặc Exists**
- **Ví dụ:** Tìm các số hóa đơn mua cùng lúc 2 sản phẩm có mã số “BB01” và “BB02”.
 - select distinct sohd
from CTHD where masp='BB01' and sohd **IN**
(select distinct sohd from CTHD where masp='BB02')
 - select distinct A.sohd
from CTHD A where A.masp='BB01' and
EXISTS (select * from CTHD B
where B.masp='BB02' and A.sohd=B.sohd)

Phép chia

Sử dụng NOT EXISTS

- **Ví dụ:** Tìm số hóa đơn đã mua tất cả những sản phẩm do “Trung Quoc” sản xuất.
- Select sohd from HOADON where not exists
(select * from SANPHAM
where nuocsx=‘Trung Quoc’ and not exists
(select * from CTHD where
HOADON.sohd=CTHD.sohd and
CTHD.masp=SANPHAM.masp))

Câu truy vấn con

- **Các hàm tính toán cơ bản**
 - COUNT: Đếm số bộ dữ liệu của thuộc tính
 - MIN: Tính giá trị nhỏ nhất
 - MAX: Tính giá trị lớn nhất
 - AVG: Tính giá trị trung bình
 - SUM: Tính tổng giá trị các bộ dữ liệu

Câu truy vấn con

- **Các hàm tính toán thường đứng sau mệnh đề Select và Having của cấu trúc truy vấn.**
- **Ví dụ:** Tính tổng lương, lương cao nhất, thấp nhất và giá trị lương trung bình của các nhân viên trong công ty.
`SELECT SUM(Luong) as Tổng lương, MAX(Luong) as Lương cao nhất, MIN(Luong) as Lương thấp nhất, AVG(Luong) as Lương TB
FROM NHAN_VIEN`

Bài tập

NHANVIEN					
MANV	HOTEN	PHAI	MANQL	PHONG	LUONG
NV001	Nguyễn Ngọc Linh	Nữ	Null	NC	2.800.000
NV002	Đinh Bá Tiến	Nam	NV002	DH	2.000.000
NV003	Nguyễn Văn Mạnh	Nam	NV001	NC	2.300.000
NV004	Trần Thanh Long	Nam	NV002	DH	1.800.000
NV005	Nguyễn Thị Hồng Vân	Nữ	NV001	NC	2.500.000
NV006	Nguyễn Minh	Nam	NV002	DH	2.000.000
NV007	Hà Duy Lập	Nam	NV003	NC	1.800.000
NV008	Trần Kim Duyên	Nữ	NV003	NC	1.800.000
NV009	Nguyễn Kim Anh	Nữ	NV003	NC	2.000.000

Bài tập

1. Tính lương thấp nhất, cao nhất, trung bình và tổng lương của tất cả các nhân viên.
2. Có tất cả bao nhiêu nhân viên
3. Bao nhiêu nhân viên có người quản lý
4. Bao nhiêu phòng ban có nhân viên trực thuộc
5. Tính lương trung bình của các nhân viên
6. Tính lương trung bình của các nhân viên theo từng phòng ban

Các hàm tính toán và gom nhóm

- **Gom nhóm: mệnh đề GROUP BY**
 - Sử dụng hàm gom nhóm trên các bộ trong quan hệ.
 - Mỗi nhóm bộ bao gồm tập hợp các bộ có cùng giá trị trên các thuộc tính gom nhóm
 - Hàm gom nhóm áp dụng trên mỗi bộ độc lập nhau.
 - SQL có mệnh đề GROUP BY để chỉ ra các thuộc tính gom nhóm, các thuộc tính này phải xuất hiện trong mệnh đề SELECT

Các hàm tính toán và gom nhóm

- **Ví dụ:** Tính lương trung bình của các nhân viên theo từng phòng ban.

```
SELECT          phong, avg(LUONG) as  
LUONGTB  
FROM            NhanVien  
GROUP BY        phong
```

Các hàm tính toán và gom nhóm

- **Điều kiện sau gom nhóm: mệnh đề HAVING**
 - Lọc kết quả theo điều kiện, sau khi đã gom nhóm
 - Điều kiện ở HAVING được thực hiện sau khi gom nhóm, các điều kiện có liên quan đến thuộc tính Group By
- **Ví dụ:** tìm phòng có số lượng nhân viên “Nữ” trên 5 người

```
SELECT phong
FROM      NhanVien
WHERE phai = 'Nữ'
GROUP BY  phong
HAVING count(manv) > 5
```

Sắp xếp kết quả truy vấn

- Mặc định, các dòng dữ liệu trong kết quả của câu truy vấn tuân theo thứ tự của chúng trong bảng dữ liệu hoặc được sắp xếp theo chỉ mục
- Trong trường hợp muốn dữ liệu được sắp xếp theo chiều tăng hoặc giảm của giá trị của một hoặc nhiều trường, ta sử dụng thêm mệnh đề **ORDER BY** trong câu lệnh **SELECT**
- Dữ liệu được sắp xếp có thể theo chiều tăng (ASC) hoặc giảm (DESC), mặc định là sắp xếp theo chiều tăng.

Sắp xếp kết quả truy vấn

- **Ví dụ 1:** Câu lệnh dưới đây hiển thị danh sách các môn học và sắp xếp theo chiều giảm dần của số đơn vị học trình:
 - `SELECT * FROM monhoc ORDER BY sodvht DESC`
 - Nếu sau `ORDER BY` có nhiều cột thì việc sắp xếp dữ liệu sẽ được ưu tiên theo thứ tự từ trái qua phải

Sắp xếp kết quả truy vấn

- **Ví dụ 2:** SELECT hodem, ten, gioitinh, YEAR(GETDATE())-YEAR(ngaysinh) AS tuoi FROM sinhvien WHERE ten='Bình' ORDER BY gioitinh, tuoi