

THAO TÁC DỮ LIỆU, TẠO KHUNG NHÌN

NỘI DUNG BÀI HỌC

- 1 THAO TÁC DỮ LIỆU: BỔ SUNG DỮ LIỆU
- 2 THAO TÁC DỮ LIỆU: CẬP NHẬT DỮ LIỆU
- 3 THAO TÁC DỮ LIỆU: XÓA DỮ LIỆU
- 4 KHUNG NHÌN - VIEW

XÉT CÁC BẢNG DỮ LIỆU SAU

Khoa	
	MaKhoa
	TenKhoa
	SoDT

Lop	
	MaLop
	TenLop
	MaKhoa

SinhVien	
	MaSV
	HoDem
	Ten
	NgaySinh
	GioiTinh
	QueQuan
	MaLop

BỔ SUNG DỮ LIỆU

❖ Bổ sung từng dòng dữ liệu với lệnh INSERT

INSERT INTO tên_bảng[(danh_sách_cột)] VALUES(danh_sách_trị)

- **Ví dụ 1:** bổ sung thêm một dòng dữ liệu vào bảng KHOA
INSERT INTO khoa VALUES('DHT10','Khoa Luật','054821135')
- **Ví dụ 2:** bổ sung thêm một dòng dữ liệu vào bảng KHOA, nhưng chỉ 2 cột
INSERT INTO khoa(MaKhoa,TenKhoa) VALUES('DHT11','Khoa CNTT')

BỔ SUNG NHIỀU DÒNG DỮ LIỆU TỪ BẢNG KHÁC

- ❖ Một cách sử dụng khác của câu lệnh INSERT được sử dụng để bổ sung nhiều dòng dữ liệu vào một bảng, các dòng dữ liệu này được lấy từ một bảng khác thông qua câu lệnh SELECT
- Cú pháp câu lệnh INSERT có dạng như sau:

INSERT INTO tên_bảng[(danh_sách_cột)] câu_lệnh_SELECT

Ví dụ: Giả sử ta có bảng **LUUSINHVIEN** bao gồm các trường HODEM, TEN, NGAYSINH. Câu lệnh dưới đây bổ sung vào bảng LUUSINHVIEN các dòng dữ liệu có được từ câu truy vấn SELECT:

```
INSERT INTO luusinhvien
```

```
SELECT hodem,ten,ngaysinh FROM sinhvien WHERE quequan like '%Hà Nội'
```

CẬP NHẬT DỮ LIỆU

- ❖ Câu lệnh UPDATE trong SQL được sử dụng để cập nhật dữ liệu trong các bảng.
- ❖ Câu lệnh này có cú pháp như sau:

```
UPDATE tên_bảng  
SET tên_cột = biểu_thức  
[, ..., tên_cột_k = biểu_thức_k]  
[FROM DS_bảng_khác]  
[WHERE điều_kiện]
```

Ví dụ 1: cập nhật lại số đơn vị học trình của các môn học có số đơn vị học trình ≤ 2

```
UPDATE monhoc  
SET sodvht = 3  
WHERE sodvht <= 2
```

SỬ DỤNG CẤU TRÚC CASE TRONG CÂU LỆNH UPDATE

❖ Ví dụ: Giả sử ta có bảng NHATKYPHONG sau đây

SOPHONG	LOAIPHONG	SONGAY	TIENPHONG
101	A	5	
202	B	5	
101	A	2	
102	C	3	

- Sau khi thực hiện câu lệnh:

UPDATE nhatkypdong

SET tienphong=songay*CASE WHEN loaiphong='A' THEN 100
WHEN loaiphong='B' THEN 70
ELSE 50

END

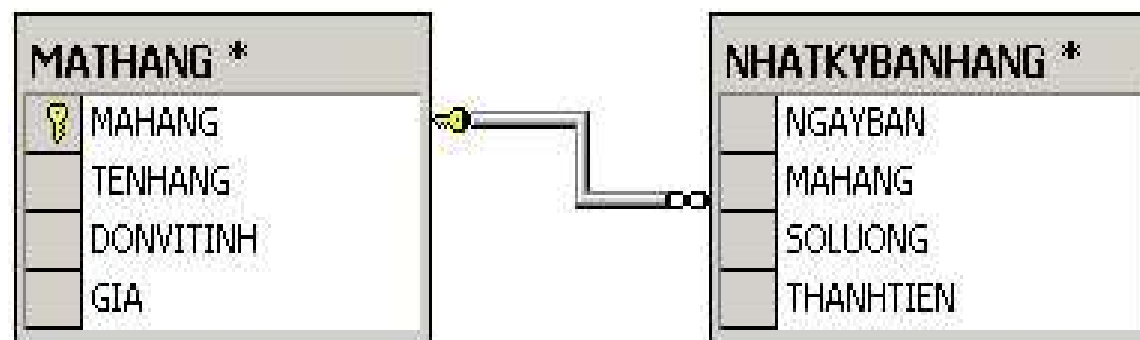
- Dữ liệu trong bảng sẽ là:

SOPHONG	LOAIPHONG	SONGAY	TIENPHONG
101	A	5	500
202	B	5	350
101	A	2	200
102	C	3	150



ĐIỀU KIỆN CẬP NHẬT DỮ LIỆU LIÊN QUAN ĐẾN NHIỀU BẢNG

- ❖ Mệnh đề FROM trong câu lệnh UPDATE được sử dụng khi cần chỉ định các điều kiện liên quan đến các bảng khác với bảng cần cập nhật dữ liệu. Trong trường hợp này, trong mệnh đề WHERE thường có điều kiện nối giữa các bảng.



Ví dụ: Cập nhật giá trị trường THANHTIEN = SOLUONG × GIA:

```
UPDATE nhatkynhanhang SET thanhtien = soluong*gia
FROM mathang
WHERE nhatkynhanhang.mahang = mathang.mahang
```



XOÁ DỮ LIỆU

❖ Cú pháp lệnh

```
DELETE FROM tên_bảng  
[FROM DS_bảng_khác]  
[WHERE điều_kiện]
```

Ví dụ 1: xoá khỏi bảng SINHVIEN những sinh viên quê tại Hà Nội

```
DELETE FROM sinhvien WHERE QueQuan LIKE N'%Hà Nội'
```

Ví dụ 2: xoá khỏi bảng SINHVIEN những sinh viên lớp 'CNTT2K14'

```
DELETE FROM sinhvien  
FROM lop  
WHERE lop.malop=sinhvien.malop AND tenlop='CNTT2K14'
```

SỬ DỤNG TRUY VẤN CON TRONG CÂU LỆNH DELETE

❖ Trong câu lệnh delete, người ta sử dụng truy vấn con để lấy điều kiện xóa trong 1 bảng khác để xóa.

Ví dụ: xóa khỏi bảng LOP những lớp không có sinh viên nào học
`DELETE FROM lop WHERE malop NOT IN (SELECT DISTINCT malop
FROM sinhvien)`

KHUNG NHÌN - VIEW

- ❖ Khung nhìn là một bảng tạm thời, có cấu trúc như một bảng, khung nhìn không lưu trữ dữ liệu mà nó được tạo ra khi sử dụng, khung nhìn là đối tượng thuộc CSDL.
- ❖ Khung nhìn được tạo ra từ câu lệnh truy vấn dữ liệu (lệnh SELECT), truy vấn từ một hoặc nhiều bảng dữ liệu.
- ❖ Có thể thực hiện truy vấn dữ liệu trên cấu trúc của khung nhìn.
- ❖ Sử dụng khung nhìn trong CSDL đem lại các lợi ích sau:
 - Đơn giản hoá các thao tác truy vấn dữ liệu
 - Tập trung trên dữ liệu được xác định
 - Bảo mật dữ liệu

KHUNG NHÌN - VIEW

❖ Tạo khung nhìn:

```
CREATE VIEW tên_khung_nhìn[(danh_sách_tên_cột)]
```

```
AS
```

```
câu_lệnh_SELECT
```

XÉT CÁC BẢNG DỮ LIỆU SAU

Lop	
	MaLop
	TenLop
	Khoa

SinhVien	
	MaSV
	HoDem
	Ten
	NgaySinh
	MaLop

KHUNG NHÌN - VIEW

- ❖ Ví dụ: Câu lệnh dưới đây tạo khung nhìn có tên DSSV từ câu lệnh SELECT truy vấn dữ liệu từ hai bảng SINHVIEN và LOP

```
CREATE VIEW dssv
```

```
AS
```

```
SELECT masv,hodem,ten, Year(GetDate())-Year(NgaySinh) as Tuoi,tenlop FROM sinhvien,lop  
WHERE sinhvien.malop=lop.malop
```

- ❖ thực hiện câu lệnh:

```
SELECT * FROM dssv => trả về kết quả câu truy vấn.
```

- ❖ Hoặc tạo view có tiêu đề thì phải trùng các fields trong select

```
CREATE VIEW dssv(mã,họ đệm,tên,tuổi,lớp)
```

```
AS
```

```
SELECT masv,hodem,ten, Year(GetDate())-Year(NgaySinh) as Tuoi,tenlop FROM sinhvien,lop  
WHERE sinhvien.malop=lop.malop
```

KHUNG NHÌN - VIEW

- ❖ Tất cả các cột trong view đều phải có tên cụ thể.
- ❖ Ví dụ: Câu lệnh dưới đây là câu lệnh sai do cột thứ 4 không xác định được tên cột

```
CREATE VIEW tuoisinhvien
```

```
AS
```

```
SELECT masv,hodem,ten, Year(GetDate())-Year(NgaySinh)
```

```
FROM sinhvien
```

CẬP NHẬT, BỔ SUNG VÀ XOÁ DL THÔNG QUA KHUNG NHÌN

- ❖ Đối với một số khung nhìn, ta có thể tiến hành thực hiện các thao tác cập nhật, bổ sung và xoá dữ liệu.
- ❖ Về mặt lý thuyết, để có thể thực hiện thao tác bổ sung, cập nhật và xoá, một khung nhìn trước tiên phải thoả mãn các điều kiện sau đây:
 - Trong câu lệnh **SELECT** định nghĩa khung nhìn không được sử dụng từ khoá **DISTINCT**, **TOP**, **GROUP BY** và **UNION**.
 - Các thành phần xuất hiện trong danh sách chọn của câu lệnh **SELECT** phải là các cột trong các bảng cơ sở. Trong danh sách chọn không được chứa các biểu thức tính toán, các hàm gộp.
- ❖ Ngoài những điều kiện trên, các thao tác thay đổi đến dữ liệu thông qua khung nhìn còn phải đảm bảo thoả mãn các ràng buộc trên các bảng cơ sở, tức là vẫn đảm bảo tính toàn vẹn dữ liệu.