THIẾT KẾ CƠ SỞ DỮ LIỆU QUAN HỆ (Relational Database Designing)

Phần II – NGÔN NGỮ TRUY VẤN SQL

(Structured Query Language = ngôn ngữ truy vấn có cấu trúc)

SQL = Structured Query Language

- Là ngôn ngữ dùng để truy vấn dữ liệu
- Ngôn ngữ = cú pháp (cấu trúc ngữ pháp) + các từ khóa (từ vựng) + hàm lập sẵn.
- Là 1 công cụ giao tiếp của Hệ Quản Trị CSDL
- Là cầu nối giữa :
  - Nhà phát triển (Lập trình viên) và Hệ quản trị CSDL
  - Người dùng cuối (End-user) và Hệ quản trị CSDL

SQL = Structured Query Language

- Ngôn ngữ SQL là một chuẩn chung tương đối giữa các Hệ quản trị CSDL khác nhau.
- 1 trong các cú pháp của SQL :

SELECT <tên các thuộc tính>

FROM <tên các quan hệ>

WHERE <điều kiện chọn>

…

Cú pháp SQL – Kiểu Dữ liệu

(data type)

- Chuỗi (String) : được đặt trong dấu nháy kép hoặc đơn.
  - Ví dụ :

```
SELECT *
FROM SINHVIEN
WHERE MASV = “SV01”
```

dữ liệu chuỗi

Cú pháp SQL – Kiểu Dữ liệu (t.t)

(data type)

- Số (number)
  - Ví dụ : 1024 ; 4.5 ; …
- Ngày tháng (date/time) : được đặt trong cặp dấu #, giữa ngày – tháng – năm là dấu phân cách “-” hoặc “/”, tên tháng có thể là số (1-12) hoặc viết tắt 3 chữ cái đầu.
  - Ví dụ : #12/2/2001# ; #1-Jan-94# ; …

Cú pháp SQL – Các toán tử số học

(Arithmetic Operations)

Toán tử	Ý nghĩa	Ví dụ	Kết quả
+	Cộng	5 + 2 #28/08/01# + 4	7 #01/09/01#
-	Trừ	#02/09/01# - 3	#30/08/01#
*	Nhân	5 * 2	10
/	Chia	5 / 2	2.5
\	Chia nguyên	5 \ 2	2
^	Lũy thừa	5 ^ 2	25
Mod	Chia dư	5 Mod 2	1

Cú pháp SQL – Các toán tử so sánh

(Comparative Operations)

Toán tử	Ý nghĩa	Ví dụ	Kết quả
<	Nhỏ hơn	3 < 5	True
<=	Nhỏ hơn hay bằng	2 <= 5	True
>	Lớn hơn	2 > 5	False
>=	Lớn hơn hay bằng	2 >= 5	False
=	Bằng nhau	2 = 5	False
<>	Khác nhau	2 <> 5	True

Cú pháp SQL – Các toán tử luận lý

(Logical Operations)

Toán tử	Ý nghĩa	Ví dụ	Kết quả
Not	Luật phủ định	Not (5 > 2) Not (2>5)	False True
And	Luật và	(5>2) And (2>5) (5>2) And (5>4)	False True
Or	Luật hay	(5>2) Or (2>5) (2>5) Or (4>5)	True False

Ví dụ

1. *SELECT* HO,TEN
   FROM SINHVIEN
   WHERE NOT(MASV = ‘SV01’)
2. *SELECT* MASV,HO,TEN
   FROM SINHVIEN
   WHERE (DIEMTB >= 5) *AND* (DIEMTB<=6.5)

Cú pháp SQL – Các toán tử *BETWEEN…AND*

Cú pháp :

value1 *Between* value2 *and* value3

Ý nghĩa : trả về True nếu value1 nằm giữa value2 và value3 ⇔value2<=value1<=value3

Ví dụ :

```
SELECT    *
FROM      SINHVIEN
WHERE     DIEMTB BETWEEN 5 AND 6.5
```

Cú pháp SQL – Các toán tử *LIKE*

Cú pháp :

value1 *LIKE* <khuôn mẫu giá trị>

Ý nghĩa :

_Trả về các value1 có dạng thức giống như <khuôn mẫu giá trị>

Các ký tự đại diện dùng trong khuôn mẫu :

* : đại diện cho tất cả ký tự bất kỳ

? : đại diện cho một ký tự bất kỳ

: đại diện cho 1 ký tự số

[A_1,A_2,…] : đại diện cho 1 ký tự thuộc tập {A_1, A_2, …}

[A_1 – A_2] : đại diện cho 1 ký tự thuộc khoảng ký tự từ A_1 đến A_2

Các toán tử *LIKE* – Ví dụ

Chọn tất cả các cột có trong quan hệ

SELECT *

FROM SINHVIEN

WHERE TEN *LIKE* ‘*Hoa’

Ý nghĩa : tìm tất cả sinh viên có từ Hoa trong phần cuối của tên, ví dụ : ‘Ngọc Thoa’, ‘Đào Hoa’, …

Các toán tử *LIKE* – Ví dụ (t.t)

SELECT *

FROM SINHVIEN

WHERE MASV *LIKE* ‘SV0[1-4]’

Ý nghĩa : tìm tất cả sinh viên có mã sinh viên là SV01, SV02, SV03 hoặc SV04

➔ Toán tử LIKE được sử dụng nhiều trong các truy vấn tìm kiếm dữ liệu

Cú pháp SQL – Các hàm lập sẵn

Cú pháp chung : <tênHàm>(Danh sách đối số)

Hàm **IIf**

Cú pháp : **IIf**(điều kiện,giá trị 1,giá trị 2)

Ý nghĩa : Trả về giá trị 1 nếu điều kiện đúng, ngược lại, trả về giá trị 2.

Ví dụ :

```
SELECT  *
FROM    SINHVIEN
WHERE   DIEMTB >= IIF(GIOITINH='Nam',6.5,6)
```

Hàm Date

Cú pháp : **Date**()

Ý nghĩa : Trả về giá trị ngày giờ hiện tại của hệ thống.

Ví dụ :

```
SELECT      *
FROM   HOADON
WHERE NGAYLAP >= (DATE()-5)
```

Hàm Sum

Cú pháp : **Sum**(<tên thuộc tính>)

Ý nghĩa : Trả về tổng của các giá trị tương ứng với <tên thuộc tính> của tất cả các bộ có trong quan hệ thỏa điều kiện WHERE.

Ví dụ :

SELECT	Sum(GIATRI)
FROM	HOADON
WHERE	NGAYLAP >= (*DATE()*-5)

Ý nghĩa : Trả về tổng giá trị của các hóa đơn có ngày lập trong vòng 6 ngày gần đây.

Hàm Max

Cú pháp : **Max**(<tên thuộc tính>)

Ý nghĩa : Trả về giá trị lớn nhất trong các giá trị tương ứng với <tên thuộc tính> của các bộ có trong quan hệ thỏa điều kiện WHERE.

Ví dụ :

SELECT Max(GIATRI)

FROM HOADON

WHERE NGAYLAP >= (*DATE()*-5)

Ý nghĩa : Trả về giá trị lớn nhất trong các hóa đơn có ngày lập trong vòng 6 ngày gần đây.

Một số hàm khác

- ***Day***(<biểu thức ngày>) : trả về chỉ số của ngày trong <biểu thức ngày>.
  - Ví dụ : ***Day***(#12/2/2005#) → 12
- ***Month***(<biểu thức ngày>) : trả về chỉ số của tháng trong <biểu thức ngày>.
- ***Year***(<biểu thức ngày>) : trả về chỉ số của năm trong <biểu thức ngày>.
- ***Len***(<giá trị chuỗi>) : trả về độ dài của chuỗi

Một số hàm khác (t.t)

- Ví dụ :

```
SELECT      *
FROM        SINHVIEN
WHERE LEN(TEN) > 4
```

- ***Chr***(<mã ASCII>) : trả về ký tự có mã ASCII tương ứng.
- ***InStr***(start,s1,s2) : trả về vị trí của chuỗi s2 trong chuỗi s1 kể từ vị trí start.
- ***LCase***(s) : trả về giá trị chuỗi in thường của chuỗi s
- ***UCase***(s) : trả về giá trị chuỗi in hoa của chuỗi s

Một số hàm khác (t.t)

- ***Left***(s,n) : trả về chuỗi gồm n ký tự bên trái của chuỗi s.
- ***Right***(s,n) : trả về chuỗi gồm n ký tự bên phải của chuỗi s.
- ***Mid***(s,i,n) : trả về chuỗi con của chuỗi s gồm n ký tự kể từ vị trí i.
- ***Nz***(v1,v2) : trả về giá trị v1 nếu v1 khác Null, ngược lại trả về giá trị v2.

Một số hàm khác (t.t)

- ***Min***(<tên thuộc tính>) : trả về giá trị nhỏ nhất trong các giá trị tương ứng với <tên thuộc tính> của các bộ thỏa điều kiện WHERE có trong quan hệ.
- ***Avg***(<tên thuộc tính>) : trả về giá trị trung bình cộng của các giá trị tương ứng với <tên thuộc tính> của các bộ thỏa điều kiện WHERE có trong quan hệ.
- ***Count***(<tên thuộc tính>) : trả về số lượng các giá trị tương ứng với <tên thuộc tính> của các bộ thỏa điều kiện WHERE và khác Null có trong quan hệ.

Các Loại Truy Vấn SQL

1. **Truy vấn chọn** **(Select query)**
   - Là các truy vấn bắt đầu bằng từ khóa *SELECT*
   - Trả về 1 giá trị hoặc 1 tập các bộ
2. **Truy vấn định nghĩa dữ liệu** **(Data Definition Query)**
   - Là các truy vấn bắt đầu bằng từ khóa *CREATE*, *DELETE*, *INSERT*, *ALTER*, …
   - Sử dụng để tạo,thêm,xóa,sửa các bảng (quan hệ), bộ, ràng buộc, … trong CSDL
3. **Truy vấn cập nhật dữ liệu** **(Data Modification Query)**

Truy vấn định nghĩa dữ liệu – Tạo lược đồ quan hệ

Ví dụ 1 :

CREATE TABLE SINHVIEN(

MASV *Text*(10) *CONSTRAINT* k1 *PRIMARY KEY*,

HOTEN *Text*(30), NGAYSINH *Date*, MALOP *Text*(10),

DIEMTB *Double*)

Ghi chú : _ Từ in nghiêng là từ khóa của SQL

_ *Text*, *Date*, *Double*, … : các kiểu dữ liệu (của thuộc tính)

_ *Text*(10) : kiểu dữ liệu Text, có độ dài 10 ký tự

Tạo lược đồ quan hệ (t.t)

_ MASV *Text*(10) *CONSTRAINT* k1 *PRIMARY KEY* :

Khai báo thuộc tính MASV là khóa chính với *quy tắc ràng buộc* tên là k1

Ví dụ 2 :

CREATE TABLE BANGDIEM(

MASV *Text*(10), MAMH *Text*(10), DIEM Double, *CONSTRAINT* k2 *PRIMARY KEY* (MASV, MAMH)

)

Thêm,xóa,sửa thuộc tính (cột)

Thêm thuộc tính và quan hệ

Ví dụ :

ALTER TABLE SINHVIEN *ADD COLUMN* GIOITINH *TEXT*(10)

Sửa kiểu dữ liệu của thuộc tính :

ALTER TABLE SINHVIEN *ALTER COLUMN* GIOITINH *BOOLEAN*

Xóa thuộc tính

Ví dụ :

ALTER TABLE SINHVIEN *DROP COLUMN* GIOITINH

Xóa,thêm các ràng buộc

Xóa ràng buộc khóa chính

Ví dụ :

ALTER TABLE SINHVIEN *DROP CONSTRAINT* k1

Thêm ràng buộc khóa chính

Ví dụ :

ALTER TABLE SINHVIEN *ADD CONSTRAINT* k1
PRIMARY KEY (MASV)

Thêm ràng buộc miền giá trị lên thuộc tính

Ví dụ :

ALTER TABLE SINHVIEN *ADD CONSTRAINT* k3
CHECK (DIEMTB>=0 AND DIEMTB<=10)

Truy vấn chọn

Ví dụ 1 : Chọn tất cả sinh viên có điểm trung bình >= 6.5

SELECT * *FROM* SINHVIEN *WHERE* DIEMTB >= 6.5;

Ví dụ 2 : Chọn 10 sinh viên có điểm trung bình cao nhất

SELECT TOP 10 *FROM* SINHVIEN;

Ví dụ 3 : Chọn 10% sinh viên có điểm trung bình cao nhất

SELECT TOP 10% *FROM* SINHVIEN;

Ví dụ 4 : Chọn có loại bỏ các bộ trùng : chọn các mức điểm khác nhau mà các sinh viên đã đạt được

SELECT DISTINCT DIEMTB *FROM* SINHVIEN;

Lưu ý : Dấu ; cho biết đã kết thúc câu lệnh SQL

Truy vấn chọn từ nhiều bảng

Ví dụ 1 : Tìm tất cả các tên học phần mà sinh viên mang mã số SV01 đã đăng ký.

```
SELECT    HOCPHAN.TENHP
FROM SINHVIEN,DANGKY_HOCPHAN,HOCPHAN
WHERE     SINHVIEN.MASV = 'SV01' AND
          SINHVIEN.MASV = DANGKY_HOCPHAN.MASV
          AND
           DANGKY_HOCPHAN.MAHP = HOCPHAN.MAHP;
```

Lưu ý : *FROM* $Q_1,Q_2,\ldots,Q_n \Leftrightarrow$

FROM $Q_1 x Q_2 x \ldots x Q_n$ (Tích Descartes)

Truy vấn chọn có kết

Ví dụ 1 : Tìm tất cả các tên học phần mà sinh viên mang mã số SV01 đã đăng ký.

```
SELECT      HOCPHAN.TENHP
FROM (SINHVIEN INNER JOIN DANGKY_HOCPHAN ON
                SINHVIEN.MASV = DANGKY_HOCPHAN.MASV)
                INNER JOIN HOCPHAN ON
                DANGKY_HOCPHAN.MAHP = HOCPHAN.MAHP
WHERE       MASV = 'SV01';
```

Lưu ý : Phép kết chính là phép chọn có điều kiện từ tích Descartes.

Truy vấn chọn có sắp thứ tự kết quả trả về

Ví dụ 1 : Tìm tất cả các tên sinh viên đã đăng ký học phần có mã là CSDL, sắp thứ tự kết quả trả về theo tên tăng dần, họ tăng dần và mã sinh viên giảm dần.

```
SELECT      MASV,HO,TEN
FROM   (SINHVIEN INNER JOIN DANGKY_HOCPHAN ON
                SINHVIEN.MASV = DANGKY_HOCPHAN.MASV
WHERE       MAHP = 'CSDL'
ORDER BY    TEN ASC, HO ASC, MASV DESC;
```

Lưu ý : Khi thuộc tính giữa các bảng được truy vấn sau từ khóa From không trùng tên thì ta có thể ghi tường minh tên thuộc tính, mà không cần phải ghi :

<Tên bảng>.<Tên thuộc tính>

Truy vấn chọn có sắp các kết quả trả về theo nhóm (group by)

Ví dụ 1 : Tìm tất cả các tên sinh viên đã đăng ký học phần ít nhất 3 học phần trở lên.

```
SELECT    SINHVIEN.MASV, SINHVIEN.HOTEN
FROM      DANGKY_HOCPHAN INNER JOIN SINHVIEN ON
          DANGKY_HOCPHAN.MASV=SINHVIEN.MASV
GROUP BY      SINHVIEN.MASV,SINHVIEN.HOTEN
HAVING        COUNT(DANGKY_HOCPHAN.MAHP)>=3
```

Truy vấn chọn lồng nhau (nested/sub query)

- Là câu lệnh truy vấn khi mà trong biểu thức điều kiện của WHERE hoặc HAVING là một câu truy vấn khác.

Ví dụ : Lấy về thông tin của sinh viên có điểm trung bình cao nhất.

SELECT MASV,HOTEN
FROM SINHVIEN
WHERE DIEMTB >=
ALL(*SELECT* DIEMTB *FROM* SINHVIEN)

Các từ khóa trong truy vấn lồng nhau

- ***ANY***, ***SOME*** : Kết quả các bộ trả về của query cha so sánh với 1 trong (bất kỳ) các bộ của query con.
- ***ALL*** : Kết quả các bộ trả về của query cha so sánh với tất cả các bộ của query con.
- ***IN*** : Kết quả các bộ trả về của query cha bằng với 1 trong (bất kỳ) các bộ của query con.
- ***NOT IN*** : Kết quả các bộ trả về của query cha không bằng với bất kỳ bộ nào của query con.
- ***EXISTS*** / ***NOT EXISTS*** : Kết quả các bộ trả về của query cha được thỏa khi query con có tồn tại ít nhất 1 bộ / không tồn tại bộ nào

Truy vấn lồng nhau – Ví dụ

Ví dụ : Lấy về thông tin của các sinh viên có đăng ký môn học CSDL.

SELECT MASV,HOTEN

FROM SINHVIEN

WHERE MASV ***IN***

(*SELECT* MASV *FROM* DANGKY_HOCPHAN *WHERE* MAHP=‘CSDL’)

Truy vấn lồng nhau – Ví dụ

Ví dụ : Lấy về thông tin của các sinh viên không có đăng ký môn học CSDL.

SELECT MASV,HOTEN

FROM SINHVIEN

WHERE MASV ***NOT IN***

(*SELECT* MASV *FROM* DANGKY_HOCPHAN *WHERE* MAHP=‘CSDL’)

Truy vấn lồng nhau – Ví dụ

Ví dụ : Trả về điểm trung bình cộng của các sinh viên nếu như có ít nhất 1 sinh viên có điểm trung bình >= 5.

SELECT *AVG*(DIEMTB)

FROM SINHVIEN

WHERE ***EXISTS***(*SELECT* DIEMTB *FROM* SINHVIEN *WHERE* DIEMTB>=5)

Truy vấn cập nhật dữ liệu – Cập nhật các bộ

Cú pháp :

UPDATE <TÊN BẢNG> *SET*

<TÊN THUỘC TÍNH 1> = <GIÁ TRỊ 1>,

<TÊN THUỘC TÍNH 2> = <GIÁ TRỊ 2>,

…

<TÊN THUỘC TÍNH n> = <GIÁ TRỊ n>

WHERE <ĐIỀU KIỆN>

Cập nhật các bộ (t.t)

Ví dụ : Cộng thêm 1 điểm cho các sinh viên có điểm trung bình >= 4 và <5

UPDATE SINHVIEN *SET* DIEMTB=DIEMTB+1

WHERE DIEMTB>=4 *AND* DIEMTB<5

Xóa các bộ

Cú pháp :

DELETE FROM <TÊN BẢNG>

WHERE <ĐIỀU KIỆN>

Ví dụ : Xóa các học sinh không có điểm trung bình

```
DELETE FROM     SINHVIEN
WHERE           DIEMTB = Null
```

Thêm các bộ vào quan hệ (bảng)

Cú pháp :

INSERT INTO <TÊN BẢNG>(
<TÊNTHUỘC TÍNH1>,<TÊNTHUỘCTÍNH2>,…)
VALUES(<GIÁ TRỊ 1>, <GIÁ TRỊ 2>, …)

Lưu ý : Các giá trị trong VALUES(…) phải tương ứng với các thuộc tính trong <TÊN BẢNG>(…)

Nếu có thuộc tính nào trong lược đồ quan hệ <TÊN BẢNG> không được khai báo trong <TÊN BẢNG>(…) và VALUES(…) thì giá trị của bộ mới được thêm vào ứng với thuộc tính đó sẽ được đặt bằng Null