



Các lệnh và các mệnh đề cơ bản trong SQL

Bởi:

unknown

Lệnh CREATE TABLE

- Cú pháp: CREATE TABLE Tên_bảng

Tên_cột Loại_dữ_liệu [Not Null],

Primary Key(Tên khoá chính),

Foreign Key(Tên khoá ngoài),...);

Trong đó:

+ *Tên_bảng*: là xâu kí tự không chứa các ký tự trống và không trùng với các từ khoá

+ *Tên_cột*: là xâu kí tự bất kì không chứa kí tự trống, tên cột trong một bảng là duy nhất, thứ tự các cột không quan trọng

+ *Loại_dữ_liệu*: gồm một số loại dữ liệu sau:

- *integer*: số nguyên từ -2147483648 đến 2147483647
- *smallinteger*: số nguyên từ -32768 đến 32767
- *decimal(n, p)*: số thập phân với độ dài tối đa là n kể cả p chữ số phần thập phân (không tính dấu chấm thập phân). Từ khoá Number trong SQL được dùng trong dạng dữ liệu này.
- *Float*: số dấu phẩy động
- *Char(n)*: xâu kí tự có độ dài cố định n, (n<=255)
- *Varchar(n)*: xâu kí tự có độ dài biến đổi, độ dài xâu có thể từ 0 đến n và được xác định tại thời điểm đưa dữ liệu vào lưu trữ.
- *Longvarchar*: xâu kí tự có độ dài không cố định. Độ dài này có thể từ 4Kbs đến 32 Kbs
- *Date*: dữ liệu dạng ngày tháng

Các lệnh và các mệnh đề cơ bản trong SQL

- Ví dụ:

+ Tạo bảng sinh viên như sau:

```
CREATE TABLE sinhvien  
(Hodem Varchar(20) Not Null,  
Ten Varchar(15) Not Null,  
Nsinh Date,  
MaSV Varchar(5) Not Null,  
Que Varchar(25),  
Hocluc Decimal(4,2),  
PRIMARY KEY (MaSV));
```

+ Tạo bảng đề tài như sau:

```
CREATE TABLE detai  
(MaDT Varchar(10) Not Null,  
TenDT Varchar(30) Not Null,  
ChuNhiem Varchar(25),  
Kinhphi Decimal(10,2),  
KetQua Decimal(4,2),  
PRIMARY KEY (MaDT));
```

+ Tạo bảng sinh viên đề tài như sau:

```
CREATE TABLE sv_detai  
(MaSV Varchar(5) Not Null,  
MaDT Varchar(10) Not Null,
```

Các lệnh và các mệnh đề cơ bản trong SQL

NoiTT Varchar(30) Not Null,

KM Decimal(10,2),

PRIMARY KEY (MaDT, MaSV),

FOREIGN KEY (MaDT) REFERENCES detai(MaDT),

FOREIGN KEY (MaSV) REFERENCES sinhvien(MaSV));

Lệnh INSERT INTO

- Cú pháp:

INSERT INTO Tên_bảng

VALUES(gia trị1, giá trị 2, ..., giá trị n);

trong đó: giá trị1, giá trị 2, ..., giá trị n là những giá trị để chèn vào các cột tương ứng từ cột 1 đến cột n của một bản ghi trong bảng

- Chức năng:

Chèn một bản ghi vào bảng

- Ví dụ:

Chèn vào bảng sinh viên một bản ghi như sau:

INSERT INTO sinhvien

VALUES("Nguyễn Hồng", "Son", "2/12/1976", "A420", "Hà Nội", 9.5);

Lệnh DELETE

- Cú pháp:

DELETE FROM Tên_bảng

WHERE <Điều kiện xoá >;

- Chức năng:

Xoá các bản ghi trong bảng thoả mãn điều kiện xoá

Các lệnh và các mệnh đề cơ bản trong SQL

- Ví dụ:

Xóa những sinh viên có học lực <5 trong bảng sinh viên

```
DELETE FROM sinhvien
```

```
WHERE (HocLuc<5);
```

Lệnh UPDATE

- *Cú pháp:*

```
UPDATE Tên_bảng
```

```
SET Giá_trị_mới
```

```
WHERE <Điều kiện sửa đổi>;
```

- *Chức năng:*

Sửa đổi giá trị các trường của các bản ghi trong bảng

- Ví dụ:

Sửa học lực của sinh viên có MaSV = “B401” lên 8

```
UPDATE sinhvien
```

```
SET HocLuc = 8
```

```
WHERE MaSV = “B401”;
```

Lệnh SELECT

SELECT là một lệnh hỏi dữ liệu cơ bản trong SQL. Có rất nhiều mệnh đề con tùy chọn trong câu lệnh SELECT, vì vậy chúng ta sẽ làm quen lần lượt từng bước một.

Cấu trúc SELECT – FROM – WHERE

Cấu trúc SELECT – FROM – WHERE là cấu trúc đơn giản nhất của SQL.

- *Cú pháp:*

```
SELECT <Danh sách các cột>
```

Các lệnh và các mệnh đề cơ bản trong SQL

FROM <Danh sách bảng>

WHERE <Điều kiện>;

Trong đó:

+ <Danh sách các cột>: Là danh sách các cột hoặc biểu thức của các cột được đưa vào kết quả truy vấn

+ <Danh sách bảng>: Là danh sách các bảng mà từ đó các cột được lấy ra

+ <Điều kiện>: Là một biểu thức logic xác định các bản ghi thoả mãn điều kiện của câu lệnh.

- Ví dụ 1: Hiển thị họ đệm, tên của bảng sinh viên:

```
SELECT Hodem,Ten
```

```
FROM sinhvien;
```

- Ví dụ 2: Hiển thị họ đệm, tên, học lực của những sinh viên có học lực ≥ 8 trong bảng sinh viên:

```
SELECT Hodem, Ten, HocLuc
```

```
FROM sinhvien
```

```
WHERE HocLuc $\geq$ 8;
```

Truy vấn sử dụng các hàm MAX, MIN, AVG, SUM, COUNT

Hàm MAX

- Chức năng:

Cho giá trị lớn nhất trong cột

- Ví dụ:

Hiển thị học lực cao nhất trong danh sách sinhvien

```
SELECT Max(HocLuc) AS DiemCaoNhat
```

```
FROM sinhvien;
```

Các lệnh và các mệnh đề cơ bản trong SQL

(AS để đặt tên cho cột Max(HocLuc))

Hàm MIN

- Chức năng:

Cho giá trị nhỏ nhất trong cột

- Ví dụ:

Hiển thị học lực nhỏ nhất trong danh sách sinh viên

```
SELECT Min(HocLuc) AS DiemThapNhat
```

```
FROM sinhvien;
```

Hàm AVG

- Chức năng: Cho giá trị trung bình cộng trong cột

- Ví dụ: Hiển thị học lực trung bình của cột học lực trong danh sách sinh viên

```
SELECT AVG(HocLuc) AS DiemTB
```

```
FROM sinhvien;
```

Hàm SUM

- Chức năng: Cho tổng giá trị trong cột

- Ví dụ: Hiển thị tổng học lực của cột học lực trong danh sách sinh viên

```
SELECT SUM(HocLuc) TongHL
```

```
FROM sinhvien;
```

Hàm COUNT

- Chức năng: Cho biết số phần tử (hàng) trong cột

- Ví dụ: Đếm số bản ghi (hàng) của cột học lực trong danh sách sinh viên

```
SELECT COUNT(HocLuc) AS SoSinhVien
```

Các lệnh và các mệnh đề cơ bản trong SQL

FROM sinhvien;

Truy vấn sử dụng các phép AND, OR, IN, BETWEEN, NOT, ALL

+ AND: Phép và logic

+ OR: Phép hoặc logic

+ IN: Là phần tử của...

+ BETWEEN: Là phần tử giữa các phần tử ...

+ NOT: Phép phủ định

+ ALL: Là tất cả những phần tử ...

- Ví dụ 1: Đưa ra danh sách những sinh viên có điểm ≥ 9 và có quê = “Hà Nội”

```
SELECT *
```

```
FROM sinhvien
```

```
WHERE (diem $\geq$ 9) AND ( que = “Hà Nội”);
```

- Ví dụ 2: Đưa ra danh sách những sinh viên có quê = “Thái Bình” hoặc “Thái Nguyên”

```
SELECT *
```

```
FROM sinhvien
```

```
WHERE (que = “Thái Bình”) OR (que = “Thái Nguyên”);
```

- Ví dụ 3: Đưa ra danh sách những nhân viên có kết quả là 8,9,10

```
SELECT *
```

```
FROM sv_dtai
```

```
WHERE KetQua IN (8, 9, 10);
```

- Ví dụ 4: Đưa ra danh sách những sinh viên có kết quả nằm trong khoảng [8,10]

```
SELECT *
```

Các lệnh và các mệnh đề cơ bản trong SQL

```
FROM sv_dtai
```

```
WHERE KetQua BETWEEN 8 and 10;
```

- Ví dụ 5: Đưa ra danh sách những sinh viên có quê không phải là “Hà Nội”

```
SELECT *
```

```
FROM sinhvien
```

```
WHERE que NOT(SELECT que
```

```
FROM sinhvien
```

```
WHERE (que = “Hà Nội”));
```

- Ví dụ 6: Hiển thị tất cả kết quả của sinh viên

```
SELECT ALL KetQua
```

```
FROM sv_detai;
```

Truy vấn thay đổi tên cột, tên bảng và hiển thị các cột từ nhiều bảng khác nhau,

- Muốn hiển thị các cột từ nhiều bảng khác nhau thì trong câu lệnh SELECT chúng ta phải làm như sau:

```
SELECT < Danh sách Tên_bang.Tên_cột>
```

```
FROM <Danh sách Tên_bảng>
```

```
WHERE <Điều kiện nối bảng>;
```

- Có thể đặt tên các cột trong kết quả các truy vấn bằng cách đặt tên mới vào sau cột được chọn ngăn cách bởi từ khoá AS, tương tự ta có thể đặt tên mới cho các bảng

Ví dụ: Hiển thị danh sách sinh viên bao gồm họ đệm, tên, kết quả từ bảng sinhvien và bảng sv_dt:

```
SELECT sinhvien.HoDem AS Ho, sinhvien.Ten AS Ten, sv_dt.KetQua AS KQ
```

```
FROM sinhvien AS sv, sv_dt AS sd
```

```
WHERE (sv.MaSV = sd.MaSV);
```


Các lệnh và các mệnh đề cơ bản trong SQL

Truy vấn sử dụng lượng từ DISTINCT/ ALL

* Để tránh tình trạng đưa ra các bộ (hàng/bản ghi) trùng lặp trong các kết quả truy vấn thì SQL có lượng từ DISTINCT.

Ví dụ: Hiển thị các mã đề tài được sinh viên đăng ký trong bảng đề tài

```
SELECT DISTINCT MaDT
```

```
FROM sinhvien;
```

* Để hiển thị tất cả các hàng (lấy cả các hàng có giá trị trùng nhau) ta dùng lượng từ ALL

Ví dụ: Hiển thị tất cả các MaDT mà bảng sv_dtai có

```
SELECT ALL MaDT
```

```
FROM sv_dtai;
```

Chú ý: Ngầm định (nếu không viết Distinct/All) thì máy hiểu là All

Truy vấn sử dụng mệnh đề GROUP BY

- Để hiển thị các bản ghi theo nhóm ta dùng mệnh đề GROUP BY

- Ví dụ: Hiển thị bảng đề tài theo nhóm mã đề tài

```
SELECT MaDT
```

```
FROM detai
```

```
GROUP BY MaDT;
```

Truy vấn có sử dụng mệnh đề HAVING

- Mệnh đề HAVING thường được sử dụng cùng mệnh đề GROUP BY. Sau HAVING là biểu thức điều kiện. Biểu thức điều kiện này không tác động vào toàn bảng được chỉ ra ở mệnh đề FROM mà chỉ tác động lần lượt từng nhóm các bản ghi đã chỉ ra tại mệnh đề GROUP BY.

- Ví dụ: Đếm xem có bao nhiêu đề tài đã được sinh viên đăng ký tham gia

```
SELECT MaDT
```

Các lệnh và các mệnh đề cơ bản trong SQL

FROM detail

GROUP BY MaDT

HAVING COUNT(*)

Truy vấn có sử dụng mệnh đề ORDER BY

- Mệnh đề ORDER BY được dùng để sắp xếp dữ liệu trong bảng theo chiều tăng hoặc giảm (ASC hoặc DESC) của một cột nào đó.

- Mệnh đề ORDER BY nếu đứng sau GROUP BY thì miền tác động của sắp xếp là trong từng nhóm của cột được chỉ ra trong GROUP BY.

- Ví dụ: Sắp xếp bảng sinhvien theo chiều giảm dần của cột học lực

SELECT *

FROM sinhvien

ORDER BY hl DESC;

Truy vấn lồng nhau

- Trong lệnh SELECT có thể được lồng nhiều mức

- Ví dụ1: Hiển thị sinh viên có học lực cao nhất

SELECT *

FROM sinhvien

WHERE hl = (SELECT MAX(hl)

FROM sinhvien);

- Ví dụ 2: Hiển thị họđệm,tên của những sinh viên có kết quả ≥ 9 và có tên đề tài là Access

SELECT hodem,ten

FROM sinhvien

WHERE MaSV IN (SELECT MaSV

Các lệnh và các mệnh đề cơ bản trong SQL

```
FROM sv_dtai
```

```
WHERE (kq >= 9) AND
```

```
( MaDT IN ( SELECT MaDT
```

```
FROM detai
```

```
WHERE tendt = "Access")));
```