



Giới thiệu ngôn ngữ truy vấn cấu trúc (SQL)

Bởi:

Khoa CNTT ĐHSP KT Hưng Yên

SQL là chuẩn ngôn ngữ ANSI để truy cập CSDL.

SQL là gì?

- SQL là viết tắt của **Structured Query Language** - Ngôn ngữ truy vấn cấu trúc.
- SQL cho phép bạn truy cập vào CSDL.
- SQL là một chuẩn ngôn ngữ của ANSI.
- SQL có thể thực thi các câu truy vấn trên CSDL.
- SQL có thể lấy dữ liệu từ CSDL.
- SQL có thể chèn dữ liệu mới vào CSDL.
- SQL có thể xóa dữ liệu trong CSDL.
- SQL có thể sửa đổi dữ liệu hiện có trong CSDL.

SQL là một chuẩn

SQL là một chuẩn của ANSI (American National Standards Institute - Viện tiêu chuẩn quốc gia Hoa kỳ) về truy xuất các hệ thống CSDL. Các câu lệnh SQL được sử dụng để truy xuất và cập nhật dữ liệu trong một CSDL.

SQL hoạt động với hầu hết các chương trình CSDL như **MS Access, DB2, Informix, MS SQL Server, Oracle, Sybase** v.v...

Lưu ý: Hầu hết các chương trình CSDL hỗ trợ SQL đều có phần mở rộng cho SQL chỉ hoạt động với chính chương trình đó.

Bảng CSDL

Một CSDL thường bao gồm một hoặc nhiều bảng (table). Mỗi bảng được xác định thông qua một tên (ví dụ **Customers** hoặc **Orders**). Bảng chứa các mẫu tin - dòng (record - row), là dữ liệu của bảng.

Giới thiệu ngôn ngữ truy vấn cấu trúc (SQL)

Dưới đây là một ví dụ về một bảng có tên là **Persons** (người):

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

Bảng ở trên bao gồm 3 mẫu tin (dòng), mỗi mẫu tin tương ứng với một người, và bốn cột (**LastName**, **FirstName**, **Address** và **City**).

Ngôn ngữ truy vấn cấu trúc (SQL)

Với SQL ta có thể truy vấn CSDL và nhận lấy kết quả trả về thông qua các câu truy vấn.

Một câu truy vấn như sau:

```
SELECT LastName FROM Persons
```

Sẽ trả về kết quả như sau:

LastName
Hansen
Svendson
Pettersen

Lưu ý: Một số hệ thống CSDL đòi hỏi câu lệnh SQL phải kết thúc bằng một dấu chấm phẩy (;). Chúng ta sẽ không dùng dấu chấm phẩy trong bài viết này.

SQL là ngôn ngữ thao tác dữ liệu (DML - Data Manipulation Language)

SQL là cú pháp để thực thi các câu truy vấn. SQL cũng bao gồm cú pháp để cập nhật - sửa đổi, chèn thêm và xoá các mẫu tin.

Sau đây là danh sách các lệnh và truy vấn dạng DML của SQL:

Giới thiệu ngôn ngữ truy vấn cấu trúc (SQL)

- SELECT - lấy dữ liệu từ một bảng CSDL.
- UPDATE - cập nhật/sửa đổi dữ liệu trong bảng.
- DELETE - xoá dữ liệu trong bảng.
- INSERT INTO - thêm dữ liệu mới vào bảng.

SQL là ngôn ngữ định nghĩa dữ liệu (DDL - Data Definition Language)

Phần DDL của SQL cho phép tạo ra hoặc xoá các bảng. Chúng ta cũng có thể định nghĩa các khoá (key), chỉ mục (index), chỉ định các liên kết giữa các bảng và thiết lập các quan hệ ràng buộc giữa các bảng trong CSDL.

Các lệnh DDL quan trọng nhất của SQL là:

- CREATE TABLE - tạo ra một bảng mới.
- ALTER TABLE - thay đổi cấu trúc của bảng.
- DROP TABLE - xoá một bảng.
- CREATE INDEX - tạo chỉ mục (khóa để tìm kiếm - search key).
- DROP INDEX - xoá chỉ mục đã được tạo.

Các kiểu dữ liệu

Kiểu dữ liệu sẽ qui định loại dữ liệu nào được phép lưu trữ trong cột. Sau đây là các kiểu dữ liệu thường dùng nhất trong SQL:

Kiểu dữ liệu	Mô tả
Kiểu dữ liệu binary - Kiểu dữ liệu binary bao gồm các số hecxa. Có ba dạng của kiểu dữ liệu binary.	
Binary: Cú pháp: binary [(n)]	
Varbinary: Cú pháp: varbinary [(n)]	
Image:Cú pháp: image	
Kiểu dữ liệu Text – Kiểu dữ liệu ký tự bao gồm một tổ hợp các chữ cái, các ký hiệu hoặc các chữ số.Kiểu dữ liệu Nchar và nvarchar lưu ký tự theo tiêu chuẩn Unicode.	
Char: Cú pháp: char [(n)]	
Varchar: Cú pháp: varchar [(n)]	
Nchar: Cú pháp: nchar [(n)]	
Nvarchar: Cú pháp: nvarchar [(n)]	

Text: Cú pháp: text	
Ntext: Cú pháp: ntext	
Date và time – Dữ liệu ngày tháng được đặt trong dấu nháy đơn. Thứ tự giữa chúng không quan trọng. Có thể đặt ngày tháng trước thời gian hoặc ngược lại.	
Datetime: Cú pháp : datetime	
SmalldatetimeCú pháp: Smalldatetime	
Dữ liệu Numeric – Dữ liệu này chỉ bao gồm các số. Nó gồm các số dương, số âm, và số thập phân.	
Dữ liệu Integer: Bao gồm các số nguyên dương và nguyên âm, như -15, 0, 5, 2509. Có bốn dạng khác nhau của kiểu dữ liệu Integerint, smallint, tinyint, bigint.	
Dữ liệu Decimal: float, real	
Kiểu dữ liệu Monetary - Kiểu dữ liệu Monetary (tiền tệ) lưu trữ các giá trị tiền tệ với độ chính xác là bốn chữ số sau dấu phẩy.	
Money: Cú pháp: Money	
Smallmoney:Cú pháp: smallmoney	
Kiểu dữ liệu Bit – Kiểu dữ liệu này có thể lưu giá trị 1 hoặc 0	Bit:Cú pháp: Bit

Câu lệnh định nghĩa dữ liệu (create: database, table. alter table, drop table)

Tạo một CSDL

CREATE DATABASE tên_CSDL

Tạo một bảng trong một CSDL

- Để định nghĩa một bảng
 - Tên bảng
 - Các thuộc tính
 - Tên thuộc tính
 - Kiểu dữ liệu
 - Các RBTV trên thuộc tính
- Cú pháp

```
CREATE TABLE <Tên bảng>(
<Tên cột><Kiểu dữ liệu>[<RBTV>],
<Tên cột><Kiểu dữ liệu>[<RBTV>],
...
[<RBTV>]
)
```

Ví dụ

Tạo một bảng tên **Person** có bốn cột: **LastName**, **FirstName**, **Address** và **Age**:

```
CREATE TABLE Person(LastName varchar,FirstName varchar,Address varchar,Age
int)
```

Tạo bảng và đặt kích thước tối đa của các cột:

```
CREATE TABLE Person(LastName varchar(30),FirstName varchar,Address
varchar,Age int)
```

Tạo bảng có tên là **nhanvien** bao gồm các cột manv – mã nhân viên, honv – họ nhân viên, tenlot – tên đệm, tennv – tên nhân viên, ngsinh – ngày sinh, dchi – địa chỉ, phai – giới tính, luong – lương của nhân viên, ma_nql – mã người quản lý, phg – mã phòng nhân viên, dienthoai – điện thoại

```
CREATE TABLE NHANVIEN (
MANV CHAR(9),
HONV VARCHAR(10),
TENLOT VARCHAR(20),
TENNV VARCHAR(10),
NGSINH DATETIME,
DCHI VARCHAR(50),
PHAI CHAR(3),
```

Giới thiệu ngôn ngữ truy vấn cấu trúc (SQL)

LUONG INT,

MA_NQL CHAR(9),

PHG INT,

DIENTHOAI CHAR(6) NULL

Các ràng buộc toàn vẹn (TBTV) và cách tạo ràng buộc trên bảng

- NOT NULL, NULL, UNIQUE, DEFAULT, CHECK
- PRIMARY KEY
- FOREIGN KEY / REFERENCES
- Ràng buộc CHECK

Ràng buộc CHECK được sử dụng để chỉ định các giá trị hay khuôn dạng dữ liệu có thể được chấp nhận đối với một cột. Trên một cột có thể có nhiều ràng buộc CHECK. Để khai báo ràng buộc CHECK đối với một cột nào đó, ta sử dụng cú pháp sau:

[CONSTRAINT constraint_name]

CHECK (expression)

Trong đó **expression** là một biểu thức logic quy định giá trị hay khuôn dạng của dữ liệu được cho phép

Ví dụ : khi tạo bảng NHANVIEN, để quy định số điện thoại của nhân viên có dạng ‘## ## ##’ (chẳng hạn 713153) chúng ta sẽ viết câu lệnh như sau:

CREATE TABLE NHANVIEN (

MANV CHAR(9),

HONV VARCHAR(10),

TENLOT VARCHAR(20),

TENNV VARCHAR(10),

NGSINH DATETIME,

DCHI VARCHAR(50),

PHAI CHAR(3),

LUONG INT,

MA_NQL CHAR(9),

PHG INT,

DIENTHOAI CHAR(6) NULL constraint check_dienthoai

CHECK (DIENTHOAI like '[0-9][0-9][0-9][0-9][0-9] [0-9]'))

- Ràng buộc DEFAULT

Ràng buộc DEFAULT được sử dụng để quy định giá trị mặc định cho một cột. Giá trị này sẽ tự động được gán cho cột này khi người sử dụng bỏ xung một bản ghi mà không chỉ định giá trị cho cột. Trên mỗi cột chỉ có thể có nhiều nhất một ràng buộc DEFAULT.

Để khai báo một giá trị cho một cột, ta chỉ định một ràng buộc DEFAULT cho cột bằng cách sử dụng cú pháp sau:

[CONSTRAINT constraint_name]

DEFAULT { const_expression

| nonarguments_function

| NULL)

Câu lệnh dưới đây chỉ định giá trị mặc định là ‘không biết’ cho cột Address trong bảng Person

CREATE TABLE Person

(

LastName varchar,

FirstName varchar,

Address varchar default 'không biết',

Age int

)

- Ràng buộc PRIMARY KEY

Ràng buộc PRIMARY KEY được sử dụng để định nghĩa khóa chính của bảng. Một ràng buộc PRIMARY KEY đảm bảo rằng không có các giá trị trùng lặp được đưa vào trên cột. Hay nói cách khác, giá trị của khóa chính sẽ giúp cho ta xác định được duy nhất một dòng (bản ghi) trong bảng dữ liệu. Mỗi một bảng chỉ có thể có duy nhất một khóa chính và bản thân khóa chính không chấp nhận giá trị NULL. Ràng buộc PRIMARY KEY là cơ sở cho việc đảm bảo tính toàn vẹn thực thể cũng như toàn vẹn tham chiếu .

Để khai báo một ràng buộc PRIMARY KEY, bạn sử dụng cú pháp sau:

```
[CONSTRAINT constraint_name ]
```

```
PRIMARY KEY [( colname [ , colname2 [..., colname16]])]
```

- Ràng buộc FOREIGN KEY

Các bảng bên trong một CSDL thường có mối quan hệ với nhau. Các mối quan hệ này được xác định dựa trên tính bằng nhau giữa một hay nhiều trường của bảng này với một hay nhiều trường của bảng khác . Nếu có một hay nhiều cột nào đó của một bảng có giá trị được xác định từ một hay nhiều trường khóa của bảng khác thì các cột đó được gọi là có ràng buộc khóa ngoại (foreign key). Các ràng buộc FOREIGN KEY được sử dụng kết hợp với các ràng buộc PRIMARY KEY và UNIQUE nhằm đảm bảo tính toàn vẹn tham chiếu giữa các bảng được chỉ định

Để khai báo khóa ngoại, bạn sử dụng câu lệnh như sau:

```
[ CONSTRAINT constraint_name ]
```

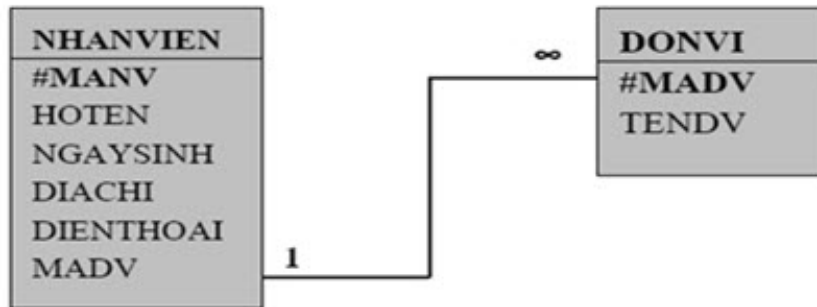
```
[ FOREIGN KEY (cloname [ ,cloname2 [..., colname16]])]
```

```
REFERENCES references_table [( ref_cloname
```

```
[, ref_colname2
```

```
[..., colname16]])]
```

Ví dụ: Tạo hai bảng Nhân Viên(MaNV, Hoten, NgaySinh, Diachi, Diênthoai, MaDV) và ĐƠN VỊ(MaDV,TenDV) theo mô hình dưới đây.



Chúng ta sẽ viết câu lệnh tạo hai bảng trên như sau

Create table DonVi(MaDV char(2)primary key, tenDV char(20)notnull)

Create table NhanVien(Manv char(10)notnull,Hoteen char(30)notnull,NgaySinh datetime null, Diachi char(50) null, Dien thoai char(6) null, MaDV char(2) foreign key(MaDV))

- Ràng buộc UNIQUE

Thay vì sử dụng khóa chính, bạn có thể sử dụng ràng buộc UNIQUE để đảm bảo tính toàn vẹn thực thể. Sử dụng ràng buộc UNIQUE trên một (hay nhiều) cột bắt buộc các giá trị dữ liệu trên một (hay nhiều) cột này không được trùng lặp nhau. Để khai báo một ràng buộc UNIQUE, bạn sử dụng cú pháp lệnh sau đây:

[CONSTRAINT constraint_name]

UNIQUE [cloname1 [, cloname2 [..., cloname16]]]

+ Ràng buộc NOT NULL quy định các giá trị khi thêm vào cột trong bảng CSDL phải luôn luôn khác rỗng

+ Ràng buộc NULL cho phép các giá trị rỗng được thêm vào cột trong bảng CSDL, khi tạo bảng nếu không khai báo ràng buộc thì các cột được mặc định ràng buộc NULL

Lệnh sửa đổi bảng (alter table)

Ví dụ - Thay đổi cấu trúc bảng

Thêm vào bảng NHANVIEN cột NGHENGHIEP

ALTER TABLE NHANVIEN ADD NGHENGHIEP CHAR(20)

Xóa cột NGHENGHIEP từ bảng NHANVIEN

Giới thiệu ngôn ngữ truy vấn cấu trúc (SQL)

ALTER TABLE NHANVIEN DROP COLUMN NGHENGHIEP

Thay đổi kiểu dữ liệu của cột DCHI thành VARCHAR(60)

ALTER TABLE NHANVIEN ALTER COLUMN DCHI VARCHAR(60)

Lệnh thay đổi ràng buộc của bảng

Thêm RBTV

ALTER TABLE <Tên bảng> ADD

CONSTRAINT<TEN_RBTV><RBTV>,

CONSTRAINT<TEN_RBTV><RBTV>,

.....

Xóa RBTV

ALTER TABLE <Tên_bảng>DROP<Tên_RBTV>

Ví dụ thay đổi ràng buộc toàn vẹn

Thêm ràng buộc khóa chính, ràng buộc khóa ngoại, ràng buộc mặc định và ràng buộc kiểm tra vào bảng NHANVIEN

ALTER TABLE PHONGBAN ADD

CONSTRAINT PB_MAPHG_PK PRIMARY KEY (MAPHG),

CONSTRAINT PB_TRPHG FOREIGN KEY (TRPHG) REFERENCES
NHANVIEN(MANV),

CONSTRAINT PB_NGNHANCHUC_DF DEFAULT GETDATE() FOR
NG_NHANCHUC,

Lệnh xóa bảng:

Được dùng để xóa cấu trúc bảng

Tất cả dữ liệu của bảng cũng bị xóa

Cú pháp

Giới thiệu ngôn ngữ truy vấn cấu trúc (SQL)

DROP TABLE <Tên_bảng>