



CHƯƠNG 4

NGÔN NGỮ TRUY VẤN SQL

Structured Query Language (SQL)

- SQL là ngôn ngữ chuẩn của hệ quản trị CSDL – RDBMS
- Ngôn ngữ chuẩn hiện nay là ANSI-92 (SQL-92)
- SQL được sử dụng rộng rãi trong Oracle, SQL Server, DB2, Microsoft Access...
- T-SQL dành riêng cho SQL server.

Transact-SQL (T-SQL)

Chia thành 3 nhóm lệnh cơ bản:

- **Data definition language (DDL)** - được dùng để tạo và chỉnh sửa cấu trúc CSDL
- **Data manipulation language (DML)** – được dùng để nhập liệu, chỉnh sửa hoặc rút trích dữ liệu trong CSDL
- **Data control language (DCL)** – được dùng để quản lý việc truy xuất dữ liệu của SQL server

Các phiên bản của SQL Server

- Personal edition
- Standard edition
- Enterprise edition
- Developer edition
- Windows CE edition
- Desktop edition

Quyền xác thực đăng nhập

Login authentication

- Để kết nối vào SQL server, user buộc phải có login account (tài khoản đăng nhập).
- SQL server hỗ trợ 2 cơ chế để xác thực tài khoản đăng nhập:
 - **Windows Authentication mode:** chỉ cho phép các tài khoản của Windows được kết nối. Các tài khoản đăng nhập của riêng SQL server không thể tạo ra và sử dụng được.
 - **Mixed mode:** cho phép cả 2 loại tài khoản của Windows và của SQL server đều được dùng để đăng nhập vào SQL server

Database objects

- Table
- Data type
- Default
- Rule
- Constraint
- Index
- View
- Stored procedure
- User-defined function (UDF)
- Trigger

Cách định danh cho đối tượng

Object identifiers

- Tất cả các đối tượng của CSDL đều phải có tên với quy định sau:
 - Ký tự đầu tiên: phải là ký tự chữ hoặc I trong các ký tự sau : _, @, #
 - Các ký tự còn lại:
 - Bất kỳ ký tự unicode chuẩn nào
 - Chữ số
 - @, \$, _, và #
 - Không được dùng khoảng trắng hoặc các từ khoá dành riêng (reversed word) để đặt tên cho đối tượng

Các kiểu dữ liệu hệ thống System data types

- binary[(n)]
- varbinary[(n)]
- bit
- decimal[(p[,s])]
- real
- float[(n)]
- bigint
- int
- smallint
- tinyint
- numeric [(p[,s])]
- money
- smallmoney
- char[(n)]
- nchar[(n)]
- varchar[(n)]
- nvarchar[(n)]
- datetime
- smalldatetime
- table
- text
- ntext

Table

- Bảng là một đối tượng của CSDL được dùng để lưu trữ dữ liệu.
- Dữ liệu trong bảng được tổ chức thành các hàng (rows) và cột (columns).
 - Mỗi hàng trong bảng biểu diễn một bản ghi (record) duy nhất. Mỗi cột biểu diễn một thuộc tính (attribute).
 - Tên cột trong 1 bảng không được trùng nhau nhưng cho phép tên cột có thể trùng nhau trong những bảng khác nhau của cùng 1 CSDL.
- SQL Server cho phép:
 - Tối đa 2 triệu bảng trong 1 CSDL.
 - Tối đa 1024 cột trong 1 bảng
 - Tối đa 8060 bytes trong 1 hàng

Các hướng dẫn chung khi thiết kế bảng

- Tên bảng phải là từ định danh hợp lệ.
- Không thể thay đổi dữ liệu cột hoặc huỷ cột mà có tham chiếu chỉ mục.
- Không thể xóa 1 bảng nếu có bất kỳ phụ thuộc giữa bảng đó với các đối tượng khác của SQL Server.

Constraints - Các ràng buộc

- SQL server 2000 hỗ trợ các loại constraint
 - Default
 - Check
 - Unique
 - Primary key
 - Foreign key
- Các ràng buộc dùng để bảo đảm các giá trị của dữ liệu không vi phạm quy luật bảo toàn dữ liệu

Tạo và hủy CSDL

- Tạo CSDL:

CREATE DATABASE <Tên CSDL>

Ví dụ: create database QLSV

- Sử dụng CSDL:

USE <Tên CSDL>

Ví dụ: use QLSV

- Hủy CSDL:

DROP DATABASE <Tên CSDL>

Ví dụ: drop database QLSV

Tạo bảng dữ liệu (Table)

CREATE TABLE <Tên bảng>

(<Column 1><Kiểu dữ liệu>[<RBTV>],

<Column 2><Kiểu dữ liệu>[<RBTV>]

,.....,

<Column n><Kiểu dữ liệu>[<RBTV>])

Ràng buộc toàn vẹn – RBTV:

NOT NULL, NULL, UNIQUE, DEFAULT, PRIMARY
KEY, FOREIGN KEY / REFERENCES, CHECK

Tạo bảng dữ liệu (Table) (2)

```
CREATE TABLE KHOA (  
    MA_KHOA CHAR(5) PRIMARY KEY,  
    TEN_KHOA VARCHAR(50) NOT NULL,  
)
```

Tạo bảng dữ liệu (Table) (3)

```
CREATE TABLE NHANVIEN (  
    HONV VARCHAR(10) NOT NULL,  
    TENLOT VARCHAR(20) NOT NULL,  
    TENNV VARCHAR(10) NOT NULL,  
    MANV CHAR(5) CONSTRAINT NV_MANV_PK PRIMARY KEY,  
    NGSINH      DATETIME,  
    DCHI VARCHAR(50),  
    DIENTHOAI CHAR(8) CHECK (DIENTHOAI LIKE '[0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9]'),  
    PHAI CHAR(3) CONSTRAINT NV_PHAI_CHK  
        CHECK (PHAI IN ('Nam', 'Nu')),  
    LUONG INT CONSTRAINT NV_LUONG_DF DEFAULT (1000),  
    MA_NQL CHAR(5),  
    PHG INT  
)
```

Hủy bảng dữ liệu

DROP TABLE <Tên bảng>

Ví dụ:

DROP TABLE KHOA

DROP TABLE NHANVIEN

Sửa bảng dữ liệu

Sửa cột

**ALTER TABLE <Tên bảng> ALTER COLUMN
<Column_Name> <Kiểu dữ liệu mới> [RBTV]**

Thêm cột

**ALTER TABLE <Tên bảng> ADD COLUMN <Column_Name>
<Kiểu dữ liệu> [RBTV]**

Xóa cột

ALTER TABLE <Tên bảng> DROP COLUMN <Column_Name>

Sửa bảng dữ liệu (2)

Ví dụ:

```
ALTER TABLE KHOA ADD COLUMN NAM_THANH_LAP  
DATETIME NULL
```

```
ALTER TABLE KHOA ALTER COLUMN TEN_KHOA  
NVARCHAR(100) NOT NULL
```

```
ALTER TABLE KHOA DROP COLUMN NAM_THANH_LAP
```

Sửa bảng dữ liệu (3)

Thêm ràng buộc toàn vẹn

**ALTER TABLE <Tên bảng> ADD CONSTRAINT <Tên RBTV>
<RBTV>**

Xóa ràng buộc toàn vẹn

ALTER TABLE <Tên bảng> DROP <Tên RBTV>

Sửa bảng dữ liệu (4)

**ALTER TABLE PHONGBAN ADD CONSTRAINT
PB_MAPHG_PK PRIMARY KEY (MAPHG)**

**ALTER TABLE PHONGBAN ADD
CONSTRAINT PB_TRPHG FOREIGN KEY (TRPHG)
REFERENCES NHANVIEN(MANV),
CONSTRAINT PB_NGNHANCHUC_DF DEFAULT
(GETDATE()) FOR (NG_NHANCHUC),**

Cập nhật dữ liệu

Thêm dữ liệu vào bảng:

INSERT INTO <Tên bảng> (Cột1, Cột2,..., Cột n)

VALUES (Giá trị 1, Giá trị 2,, Giá trị n)

Xóa dữ liệu khỏi bảng:

DELETE FROM <Tên bảng> [WHERE <Điều kiện>]

Sửa dữ liệu:

UPDATE <Tên bảng> SET <Tên thuộc tính>=<giá trị mới>,

<Tên thuộc tính>=<giá trị mới>.....

[WHERE <Điều kiện>]

Truy vấn dữ liệu

SELECT select_list
FROM table_name 1,.....,table_name n
[WHERE clause]
[GROUP BY clause]
[HAVING BY clause]
[ORDER BY clause]
[COMPUTE clause]

Mệnh đề SELECT

SELECT tên_cột_1, tên_cột_2, ..., tên_cột_n

- Thứ tự của các cột được nêu ra trong câu lệnh SELECT sẽ xác định thứ tự của các cột được hiển thị trong kết quả.
- Đổi tên các cột trong các kết quả
 tiêu_đề_cột = tên_cột

hoặc

 tên_cột tiêu_đề_cột

hoặc

 tên_cột AS tiêu_đề_cột

Mệnh đề SELECT

- Tính toán các giá trị trong câu lệnh SELECT

Ví dụ 2: Tiền thưởng của mỗi nhân viên bằng 10% lương của nhân viên đó.

```
SELECT HONV+' '+TENLOT+' '+TENNV  
      AS TEN, LUONG*10% AS THUONG  
FROM NHANVIEN
```


Mệnh đề SELECT

- Từ khóa DISTINCT được sử dụng trong câu lệnh SELECT nhằm loại bỏ những dòng dữ liệu có giá trị giống nhau

Mệnh đề FROM

- Mệnh đề FROM trong câu lệnh SELECT được sử dụng để chỉ định các bảng cần truy xuất dữ liệu
- Sau mệnh đề FROM là danh sách các bảng tham gia vào truy vấn (tên các bảng được ngăn cách với nhau bởi dấu phẩy)

SELECT *select_list*

FROM *table_name1*,...,*table_name n*

Mệnh đề FROM

- Có thể sử dụng bí danh cho các bảng (alias). Bí danh được gán sau tên bảng

SELECT *

FROM NHANVIEN

SELECT N1.HONV, N1.TENLOT,
N1.TENNV, N1.DCHI

FROM NHANVIEN N1

Mệnh đề WHERE

Xác định các điều kiện đối với việc truy xuất dữ liệu

- Các toán tử so sánh (=, >, <, >=, <=, <>, !=, <)
- Giới hạn (BETWEEN và NOT BETWEEN)
- Danh sách (IN và NOT IN)
- Khuôn dạng (LIKE và NOT LIKE)
- Các giá trị chưa biết (IS NULL và IS NOT NULL)
- Nối các biểu thức (AND, OR)

Mệnh đề WHERE

Giá trị NULL có thể được nhập vào một cột theo các cách:

- Người sử dụng trực tiếp đưa giá trị NULL
- Không có dữ liệu được đưa vào và không có mặc định cho cột hay kiểu dữ liệu trên cột đó
- Cột có kiểu dữ liệu là kiểu số sẽ chứa giá trị NULL nếu giá trị được nhập gây tràn số

Mệnh đề ORDER BY

Sử dụng để sắp xếp kết quả truy vấn theo một hay nhiều cột

- Tăng dần ASC (mặc định)
- Giảm dần DESC

Ví dụ 11: Sắp xếp các nhân viên theo thứ tự lương giảm dần, nếu lương bằng nhau thì sắp xếp theo thứ tự tăng dần của ngày sinh

SELECT *

FROM NHANVIEN

ORDER BY LUONG DESC, NGSINH

Phép toán tập hợp

- Phép hội

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
UNION [ALL]

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>

- Phép giao

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
INTERSECT [ALL]

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>

Phép toán tập hợp

- Phép trừ

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
EXCEPT [ALL]

SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>

Nếu sử dụng tùy chọn ALL thì các dòng giống nhau sẽ không bị loại bỏ.

Phép kết

- **Kết nội**

SELECT <danh sách các cột>

FROM R1 [**INNER**] **JOIN** R2 **ON** <biểu thức>

WHERE <điều kiện>

Phép kết

Kết nội

Ví dụ 14: Với mỗi phòng ban cho biết tên phòng ban và địa điểm phòng

```
SELECT PB.TENPHG, DD.DIADIEM  
FROM PHONGBAN PB, DIADIEM_PHG DD  
WHERE PB.MAPHG = DD.MAPHG
```

```
SELECT PB.TENPHG, DD.DIADIEM  
FROM  
    PHONGBAN PB INNER JOIN DIADIEM_PHG DD  
    ON PB.MAPHG = DD.MAPHG
```

Phép kết

- **Kết ngoại**

SELECT <danh sách các cột>

FROM R1 LEFT|RIGHT [OUTER] JOIN
R2 ON <biểu thức>

WHERE <điều kiện>

Phép kết

• Kết ngoại

Ví dụ 15: Với mỗi nhân viên cho biết họ tên nhân viên và họ tên người quản lý trực tiếp của nhân viên đó nếu có.

```
SELECT NV1.HONV+' '+NV1.TENLOT+' '+NV1.TENNV AS  
TENNV, NV2.HONV+' '+NV2.TENLOT+' '+NV2.TENNV AS  
TENNQL
```

```
FROM NHANVIEN NV1, NHANVIEN NV2  
WHERE NV1.MA_NQL =* NV2.MANV
```

```
SELECT NV1.HONV+' '+NV1.TENLOT+' '+NV1.TENNV AS  
TENNV, NV2.HONV+' '+NV2.TENLOT+' '+NV2.TENNV AS  
TENNQL
```

```
FROM NHANVIEN NV1 RIGHT OUTER JOIN NHANVIEN  
NV2 ON NV1.MA_NQL = NV2.MANV
```

Hàm kết hợp

- COUNT
 - COUNT(*) đếm số dòng
 - COUNT(<tên thuộc tính>) đếm số giá trị khác NULL của thuộc tính
 - COUNT(DISTINCT <tên thuộc tính>) đếm số giá trị khác nhau và khác NULL của thuộc tính
- MIN : Tính giá trị nhỏ nhất
- MAX : Tính giá trị lớn nhất
- SUM : Tính tổng các giá trị
- AVG : Tính trung bình các giá trị

Mệnh đề GROUP BY

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <điều kiện>

GROUP BY <danh sách các cột gom nhóm>

GROUP BY tổ chức dữ liệu vào các nhóm.

Sau khi gom nhóm, mỗi nhóm các bộ sẽ có cùng giá trị tại các thuộc tính gom nhóm.

Mệnh đề GROUP BY

- Danh sách các thuộc tính trong mệnh đề SELECT (trừ những thuộc tính trong các hàm kết hợp) **phải có** trong danh sách các thuộc tính trong mệnh đề GROUP BY, nếu không câu lệnh sẽ không hợp lệ.

Mệnh đề HAVING

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <điều kiện>

GROUP BY <danh sách các cột gom nhóm>

HAVING <điều kiện trên nhóm>

HAVING thiết lập các điều kiện lên các nhóm trong kết quả truy vấn

Mệnh đề HAVING

- Mệnh đề HAVING thiết lập các điều kiện cho mệnh đề GROUP BY tương tự như cách thức mệnh đề WHERE thiết lập các điều kiện cho mệnh đề SELECT.
- Điểm khác biệt là trong mệnh đề WHERE không được có các hàm kết hợp, còn mệnh đề HAVING được sử dụng các hàm kết hợp.
- Mệnh đề HAVING chỉ kiểm tra điều kiện trên nhóm, không là điều kiện lọc trên từng bộ
- Sau khi gom nhóm điều kiện trên nhóm mới được thực hiện.

Mệnh đề HAVING

Thứ tự thực hiện câu truy vấn có mệnh đề
GROUP BY và HAVING

1. Chọn ra những dòng thỏa điều kiện trong mệnh đề WHERE
2. Gom nhóm theo mệnh đề GROUP BY
3. Áp dụng các điều kiện trong mệnh đề HAVING lên các nhóm
4. Rút trích các giá trị của các cột và hàm kết hợp trong mệnh đề SELECT

Truy vấn lồng

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <so sánh tập hợp> (

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <điều kiện>)

Truy vấn lồng

Câu lệnh chứa truy vấn con thường có dạng:
WHERE <biểu thức><so sánh tập hợp><truy vấn con>

- WHERE <tên thuộc tính> [NOT] IN (truy vấn con)
- WHERE [NOT] EXIST (truy vấn con)
- WHERE <tên thuộc tính><toán tử so sánh> [ANY | ALL] (truy vấn con)

Lòng phân cấp

- Mệnh đề WHERE của truy vấn con không tham chiếu đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha
- Khi thực hiện, truy vấn con sẽ được thực hiện trước

Lồng tương quan

- Mệnh đề WHERE của truy vấn con tham chiếu ít nhất một thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha
- Khi thực hiện, câu truy vấn con sẽ được thực hiện nhiều lần, mỗi lần tương ứng với một bộ của truy vấn cha

Lồng tương quan

SELECT MANV, TENNV
FROM NHANVIEN, PHONGBAN
WHERE TENPHG='NGHIEN CUU' AND PHG=MAPHG

SELECT MANV, TENNV
FROM NHANVIEN
WHERE EXISTS (SELECT *
FROM PHONGBAN
WHERE TENPHG='NGHIEN CUU'
AND PHG=MAPHG)

Phép chia $R (A,B,C,D,E) \div S (D,E)$

```
SELECT R1.A, R1.B, R1.C
FROM   R  R1
WHERE  NOT EXISTS (
    SELECT *
    FROM   S
    WHERE  NOT EXISTS (
        SELECT *
        FROM   R  R2
        WHERE  R2.D = S.D AND R2.E = S.E
        AND R1.A=R2.A AND R1.B=R2.B AND R1.C=R2.C ))
```


Phép chia

Ví dụ 27: Tìm danh sách những nhân viên làm việc trong mọi đề án của công ty

R ÷ S

→ PHANCONG ÷ DEAN

Phép chia

```
SELECT R1.A, R1.B, R1.C
FROM R R1
WHERE NOT EXISTS (
    SELECT *
    FROM S
    WHERE NOT EXISTS (
        SELECT *
        FROM R R2
        WHERE R2.D=S.D AND R2.E=S.E
        AND R1.A=R2.A AND R1.B=R2.B AND R1.C=R2.C
    ))
```

Phép chia

```
SELECT PC1.MA_NVIENT
FROM PHANCONG PC1
WHERE NOT EXISTS (
    SELECT *
    FROM DEAN DA
    WHERE NOT EXISTS (
        SELECT *
        FROM PHANCONG PC2
        WHERE PC2.SODA=DA.MADA AND
        PC1.MA_NVIENT=PC2.MA_NVIENT
    )
)
```