Ngôn ngữ T-SQL

Câu lệnh SQL

- SQL chuẩn bao gồm khoảng 40 câu lệnh
- Cú pháp chi tiết một số câu lệnh có thể thay đổi tùy vào HQTCSDL
- Ví dụ:
  - Định nghĩa dữ liệu
    - CREATE TABLE
    - DROP TABLE
    - ALTER TABLE
  - Thao tác dữ liệu:
    - SELECT
    - INSERT
    - UPDATE
    - DELETE

Câu lệnh SQL

- Điều khiển truy cập
  - GRANT
  - REVOKE
  - DENY
- Lập trình
  - DECLARE
  - OPEN
  - FETCH
  - EXECUTE
  - CLOSE
  - DELETE

Ngôn ngữ định nghĩa dữ liệu

- DDL – Data Definition Language
- Gồm các lệnh định nghĩa các đối tượng trong CSDL
  - CREATE object_Name
  - ALTER object_Name
  - DROP object_Name

  object_Name: table, view, stored procedure, indexes

Ngôn ngữ thao tác dữ liệu
DML – Data Manipulation Language

- SELECT
- INSERT
- UPDATE
- DELETE

SELECT

- Chức năng
  - Truy xuất dữ liệu từ các dòng và các cột của một hoặc nhiều bảng
  - Phép chiếu
  - Phép chọn
  - Phép kết
- Cú pháp

```
SELECT [ALL | DISTINCT][TOP n] danh_sách_chọn
[INTO tên_bảng_mới]
FROM  danh_sách_bảng/khung_nhìn
[WHERE     điều_kiện]
[GROUP BY  danh_sách_cột]
[HAVING điều_kiện]
[ORDER BY  cột_sắp_xếp]
[COMPUTE  danh_sách_hàm_gộp [BY danh_sách_cột]]
```

SELECT

- Truy vấn thường
- Truy vấn có điều kiện
- Tạo mới bảng dữ liệu từ kết quả của SELECT
- Sắp xếp kết quả truy vấn
- Phép nối
- Thống kê dữ liệu với GROUP BY
- Thống kê dữ liệu với COMPUTE
- Truy vấn con

SELECT – Truy vấn thường

- Chọn tất cả các trường: *
- Chọn một số cột
  Tenbang.Tentruong
- Thay đổi tiêu đề các cột

- Ví dụ:

```
tiêu_đề_cột = tên_trường
tên_trường AS tiêu_đề_cột
tên_trường  tiêu_đề_cột
```

```
SELECT 'Mã lớp'= malop,tenlop 'Tên lớp',khoa AS 'Khoá'
FROM lop
```

SELECT – Truy vấn thường

- Cấu trúc CASE trong danh sách chọn

```
CASE biểu_thức
   WHEN biểu_thức_kiểm_tra THEN kết_quả
   [ ... ]
   [ELSE kết_quả_của_else]
END
```

```
CASE
   WHEN điều_kiện THEN kết_quả
   [ ... ]
   [ELSE kết_quả_của_else]
END
```

```
SELECT masv,hodem,ten,
       CASE gioitinh
          WHEN 1 THEN 'Nam'
          ELSE 'Nữ'
       END AS gioitinh
FROM sinhvien
```

```
SELECT masv,hodem,ten,
       CASE
          WHEN gioitinh=1 THEN 'Nam'
          ELSE 'Nữ'
       END AS gioitinh
FROM sinhvien
```

SELECT

- Truy vấn thông thường
- Truy vấn có điều kiện
- Tạo mới bảng dữ liệu từ kết quả của SELECT
- Sắp xếp kết quả truy vấn
- Phép nối
- Thống kê dữ liệu với GROUP BY
- Thống kê dữ liệu với COMPUTE
- Truy vấn con

SELECT – Truy vấn có điều kiện

- Sử dụng mệnh đề WHERE
- Sau mệnh đề WHERE là một biểu thức logic để lọc các kết quả thỏa mãn

- Các toán tử so sánh
- Toán tử BETWEEN
- Danh sách (IN và NOT IN)
- Toán tử LIKE

Sắp xếp kết quả truy vấn

- Sử dụng ORDER BY
- Tối đa 16 cột

```
SELECT * FROM monhoc
ORDER BY sodvht DESC
```

```
SELECT hodem,ten,gioitinh,
       YEAR(GETDATE())-YEAR(ngaysinh) AS tuoi
FROM sinhvien
WHERE ten='Bình'
ORDER BY gioitinh,tuoi
```

Phép nối

- Phép nối bằng (equi-join) và phép nối tự nhiên (natural-join)
- Phép nối ngoài (Outer-join)
  - Phép nối ngoài trái (*=)
  - Phép nối ngoài phải (=*)
  - Phép nối ngoài đầy đủ
- Phép nối nhiều bảng

Phép nối

- Phép nối trong

```
tên_bảng_1 [INNER] JOIN tên_bảng_2 ON điều_kiện_nối
```

```
SELECT hodem,ten,ngaysinh
FROM sinhvien,lop
WHERE tenlop='Tin K24' AND
      sinhvien.malop=lop.malop
```

```
SELECT hodem,ten,ngaysinh
FROM sinhvien INNER JOIN lop
     ON sinhvien.malop=lop.malop
WHERE tenlop='Tin K24'
```

Phép nối ngoài

- Phép nối ngoài trái (LEFT OUTER JOIN)
- Phép nối ngoài phải (RIGHT OUTER JOIN)
- Phép nối ngoài đầy đủ (FULL OUTER JOIN)

```
tên_bảng_1 LEFT|RIGHT|FULL [OUTER] JOIN tên_bảng_2
        ON điều_kiện_nối
```

Phép nối ngoài

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
9	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

```
SELECT *
FROM nhanvien LEFT OUTER JOIN donvi
     ON nhanvien.madv=donvi.madv
```

Phép nối ngoài (LEFT OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
4	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

HOTEN	MADV	MADV	TENDV
Thanh	1	1	Doi ngoai
Hoa	2	2	Hanh chinh
Nam	2	2	Hanh chinh
Vinh	1	1	Doi ngoai
Hung	5	NULL	NULL
Phuong	NULL	NULL	NULL

Phép nối ngoài (RIGHT OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
9	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

```
SELECT *
FROM nhanvien RIGHT OUTER JOIN donvi
     ON nhanvien.madv=donvi.madv
```

Phép nối ngoài (RIGHT OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
4	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

HOTEN	MADV	MADV	TENDV
Thanh	1	1	Doi ngoai
Vinh	1	1	Doi ngoai
Hoa	2	2	Hanh chinh
Nam	2	2	Hanh chinh
NULL	NULL	3	Ke toan
NULL	NULL	4	Kinh doanh

Phép nối ngoài (FULL OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
9	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

```
SELECT *
FROM nhanvien FULL OUTER JOIN donvi
     ON nhanvien.madv=donvi.madv
```

Phép nối ngoài (FULL OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Doi ngoai
2	Hanh chinh
3	Ke toan
4	Kinh doanh

Bảng NHANVIEN

HOTEN	MADV
Thanh	1
Hoa	2
Nam	2
Vinh	1
Hung	5
Phuong	NULL

HOTEN	MADV	MADV	TENDV
Thanh	1	1	Doi ngoai
Hoa	2	2	Hanh chinh
Nam	2	2	Hanh chinh
Vinh	1	1	Doi ngoai
Hung	5	NULL	NULL
Phuong	NULL	NULL	NULL
NULL	NULL	4	Kinh doanh
NULL	NULL	3	Ke toan

Phép nối nhiều bảng

- Cho phép thực hiện phép nối nhiều bảng một cách rõ ràng
- Phép nối thực hiện theo thứ tự định nghĩa
- Ví dụ:

```
SELECT hodem,ten,ngaysinh
FROM (sinhvien INNER JOIN lop
                ON sinhvien.malop=lop.malop)
      INNER JOIN khoa ON lop.makhoa=khoa.makhoa
WHERE tenkhoa= 'Khoa công nghệ thông tin'
```

Thống kê dữ liệu với GROUP BY

- Mệnh đề GROUP BY cho phép phân hoạch các dòng dữ liệu thành các nhóm dữ liệu và thực hiện các phép toán trên các nhóm dữ liệu đó
- Các hàm gộp được sử dụng để tính toán trên toàn bảng, hoặc trên mỗi nhóm dữ liệu
- Các hàm gộp được sử dụng như là các cột trong danh sách các cột trong SELECT hoặc HAVING, không được xuất hiện sau WHERE

Thống kê dữ liệu với GROUP BY

Hàm gộp	**Chức năng**
SUM([ALL | DISTINCT] *biểu_thức*)	Tính tổng các giá trị.
AVG([ALL | DISTINCT] *biểu_thức*)	Tính trung bình của các giá trị
COUNT([ALL | DISTINCT] *biểu_thức*)	Đếm số các giá trị trong biểu thức.
COUNT(*)	Đếm số các dòng được chọn.
MAX(*biểu_thức*)	Tính giá trị lớn nhất
MIN(*biểu_thức*)	Tính giá trị nhỏ nhất

Thống kê dữ liệu với GROUP BY

- Thống kê trên toàn bộ dữ liệu

```
SELECT AVG(diemlan1)
FROM diemthi
```

```
SELECT MAX(YEAR(GETDATE())-YEAR(ngaysinh)),
       MIN(YEAR(GETDATE())-YEAR(ngaysinh)),
       AVG(YEAR(GETDATE())-YEAR(ngaysinh))
FROM sinhvien
WHERE noisinh='Hà nội '
```

Thống kê dữ liệu với GROUP BY

- Thống kê trên các nhóm
  - Ví dụ
    - Viết câu lệnh SQL để hiển thị số sinh viên của mỗi lớp
    - Cho biết điểm trung bình thi lần 1 các môn học của các sinh viên.

Thống kê dữ liệu với GROUP BY

- Thống kê trên các nhóm

```
SELECT lop.malop,tenlop,COUNT(masv) AS siso
FROM lop,sinhvien
WHERE lop.malop=sinhvien.malop
GROUP BY lop.malop,tenlop
```

```
SELECT sinhvien.masv,hodem,ten,
       sum(diemlan1*sodvht)/sum(sodvht)
FROM sinhvien,diemthi,monhoc
WHERE sinhvien.masv=diemthi.masv AND
      diemthi.mamonhoc=monhoc.mamonhoc
GROUP BY sinhvien.masv,hodem,ten
```

Thống kê dữ liệu với GROUP BY

- Thống kê trên các nhóm

Thống kê dữ liệu với GROUP BY

- Chỉ định điều kiện đối với hàm gộp (HAVING)
  - HAVING sử dụng sau GROUP BY
  - Khác với WHERE:
    - HAVING cho phép sử dụng hàm gộp còn WHERE thì không
- Ví dụ 1:

```
SELECT sinhvien.masv,hodem,ten,
       SUM(diemlan1*sodvht)/sum(sodvht)
FROM sinhvien,diemthi,monhoc
WHERE sinhvien.masv=diemthi.masv AND
      diemthi.mamonhoc=monhoc.mamonhoc
GROUP BY sinhvien.masv,hodem,ten
HAVING sum(diemlan1*sodvht)/sum(sodvht)>=5
```

- VD2: Tìm những hóa đơn có trị giá lớn hơn 1000000 (QLBH)

Thống kê dữ liệu với ROLLUP

- GROUP BY
  - Cho phép thống kê dữ liệu trên từng nhóm
  - Không biết được dữ liệu chi tiết trên từng nhóm
  - Ví dụ

```
SELECT khoa.makhoa,tenkhoa,COUNT(malop) AS solop
FROM khoa,lop
WHERE khoa.makhoa=lop.makhoa
GROUP BY khoa.makhoa,tenkhoa
```

    cho biết số lượng lớp của từng khoa chứ không cho biết được trong khoa đó gồm những lớp nào
- WITH ROLLUP
  - Sử dụng kết hợp hàm gộp và mệnh đề GROUP BY trong SELECT để cho biết kết quả của tổng các nhóm

Thống kê dữ liệu với ROLLUP

- Cú pháp

  SELECT danh sách cột

  hàm gộp (tên trường tính)

  FROM danh sách bảng

  GROUP BY danh sách cột

  WITH ROLLUP

Trong đó

- Các hàm gộp có thể dùng: SUM, AVG, MAX, MIN, COUNT
- danh_sách_cột: là danh sách các cột để nhóm dữ liệu

- Ví dụ:

```
select NgayBan, tHoaDonBan.SoHDB, sum(SLBan) as SoSanPham
from tHoaDonBan join tChiTietHDB  on tHoaDonBan.SoHDB=tChiTietHDB.SoHDB
group by NgayBan, tHoaDonBan.SoHDB
with rollup
```

Truy vấn con (Subquery)

- Là truy vấn được lồng trong một câu lệnh SELECT, INSERT, UPDATE, DELETE hoặc trong một truy vấn con khác
- Sử dụng để biểu diễn những truy vấn trong đó điều kiện truy vấn là kết quả của một truy vấn khác
- Cú pháp

```
(SELECT [ALL | DISTINCT] danh_sách_chọn
FROM  danh_sách_bảng
[WHERE     điều_kiện]
[GROUP BY  danh_sách_cột]
[HAVING điều_kiện])
```

Truy vấn con (Subquery)

- Một số chú ý khi sử dụng truy vấn con:
  - Một truy vấn con phải đặt trong ()
  - Kết quả của truy vấn con thường là một cột
  - Tên cột trong truy vấn con có thể là các cột trong truy vấn ngoài
  - Mệnh đề COMPUTE, ORDER BY không có trong truy vấn con

Truy vấn con (Subquery)

- Ví dụ

```
select MaSach, TenSach
from tSach
where MaSach not in
     (select distinct MaSach from tChiTietHDB)
```

Truy vấn con (Subquery)

- Sử dụng EXISTS với truy vấn con

```
WHERE [NOT] EXISTS (truy_vấn_con)
```

để kiểm tra xem truy vấn con có trả về kết quả nào không

- Ví dụ:

```
SELECT hodem,ten
FROM sinhvien
WHERE NOT EXISTS(SELECT masv FROM diemthi
                 WHERE diemthi.masv=sinhvien.masv)
```

Ngôn ngữ thao tác dữ liệu
DML – Data Manipulation Language

- SELECT
- INSERT
- UPDATE
- DELETE

Bổ sung dữ liệu: INSERT

- Bổ sung từng dòng dữ liệu với mỗi câu lệnh INSERT
- Bổ sung nhiều dòng dữ liệu bằng cách truy xuất dữ liệu từ các bảng khác

Bổ sung dữ liệu: INSERT

- Bổ sung từng dòng dữ liệu với mỗi câu lệnh INSERT

- Ví dụ

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

```
INSERT INTO khoa
VALUES('DHT10','Khoa Luật','054821135')

INSERT INTO sinhvien(masv,hodem,ten,gioitinh,malop)
VALUES('0241020008','Nguyễn Công','Chính',1,'C24102')

INSERT INTO sinhvien
VALUES('0241020008','Nguyễn Công','Chính',
       NULL,1,NULL,'C24102')
```

Bổ sung dữ liệu: INSERT

- Bổ sung nhiều dòng dữ liệu từ các bảng khác

```
INSERT  INTO tên_bảng[(danh_sách_cột)] câu_lệnh_SELECT
```

- Ví dụ

```
INSERT INTO luusinhvien
SELECT  hodem,ten,ngaysinh
FROM sinhvien
WHERE noisinh like '%Hà nội %'
```

- Chú ý
  - Kết quả của câu lệnh SELECT phải có số cột bằng số cột được chỉ định trong bảng đích và phải tương thích về kiểu dữ liệu

Insert into khachhang(makh, hokh, tenkh)

Select manv, honv, tennv from NhanVien where…

Ngôn ngữ thao tác dữ liệu
DML – Data Manipulation Language

- SELECT

- UPDATE
- DELETE

Cập nhật dữ liệu: UPDATE

- Cú pháp

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

- Ví dụ

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

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

Bổ sung dữ liệu: UPDATE

- Ví dụ

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

UPDATE nhatkybanhang
SET  thanhtien = soluong*gia
FROM mathang
WHERE mathang.mahang =(SELECT mathang.mahang
      FROM mathang
      WHERE mathang.mahang=nhatkybanhang.mahang)
```

Ngôn ngữ thao tác dữ liệu
DML – Data Manipulation Language

- SELECT
- INSERT
- UPDATE
- DELETE

Xóa dữ liệu: DELETE

- Cú pháp

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

- Ví dụ

```
DELETE FROM sinhvien
WHERE noisinh LIKE '%Hà nội %'

DELETE FROM sinhvien
FROM lop
WHERE lop.malop=sinhvien.malop AND tenlop='Tin K24'

DELETE FROM lop
WHERE malop NOT IN (SELECT DISTINCT malop
                    FROM sinhvien)
```

Xóa dữ liệu: DELETE

- Xóa toàn bộ dữ liệu trong bảng
  - Sử dụng câu lệnh DELETE... FROM không có WHERE
  - TRUNCATE

```
TRUNCATE TABLE tên_bảng
```