
Ngôn ngữ T-SQL

1

SQL

- SQL – Structured Query Language
- Chức năng của SQL
 - Định nghĩa dữ liệu
 - Truy xuất và thao tác với dữ liệu
 - Điều khiển truy cập
 - Đảm bảo toàn vẹn dữ liệu
- Vai trò của SQL
 - Là ngôn ngữ hỏi có tính tương tác
 - Là ngôn ngữ lập trình CSDL
 - Là ngôn ngữ quản trị CSDL
 - Là ngôn ngữ cho hệ thống Client/Server
 - Là ngôn ngữ truy cập dữ liệu trên Web

2

Nội dung

- Tổng quan về SQL
- Giới thiệu về SQL Server
- T-SQL

3

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

4

Câu lệnh SQL

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

5

Kiểu dữ liệu trong SQL

Tên kiểu	Mô tả
CHAR (<i>n</i>)	Kiểu chuỗi với độ dài cố định
NCHAR (<i>n</i>)	Kiểu chuỗi với độ dài cố định hỗ trợ UNICODE
VARCHAR (<i>n</i>)	Kiểu chuỗi với độ dài chính xác
NVARCHAR (<i>n</i>)	Kiểu chuỗi với độ dài chính xác hỗ trợ UNICODE
INTEGER	Số nguyên có giá trị từ -2^{31} đến $2^{31} - 1$
INT	Như kiểu Integer
TINYINT	Số nguyên có giá trị từ 0 đến 255.
SMALLINT	Số nguyên có giá trị từ -2^{15} đến $2^{15} - 1$
BIGINT	Số nguyên có giá trị từ -2^{63} đến $2^{63} - 1$
NUMERIC (<i>p,s</i>)	Kiểu số với độ chính xác cố định.
DECIMAL (<i>p,s</i>)	Tương tự kiểu Numeric

6

Kiểu dữ liệu trong SQL

Tên kiểu	Mô tả
FLOAT	Số thực có giá trị từ -1.79E+308 đến 1.79E+308
REAL	Số thực có giá trị từ -3.40E + 38 đến 3.40E + 38
MONEY	Kiểu tiền tệ
BIT	Kiểu bit (có giá trị 0 hoặc 1)
DATETIME	Kiểu ngày giờ (chính xác đến phần trăm của giây)
SMALLDATETIME	Kiểu ngày giờ (chính xác đến phút)
TIMESTAMP	
BINARY	Dữ liệu nhị phân với độ dài cố định (tối đa 8000 bytes)
VARBINARY	Dữ liệu nhị phân với độ dài chính xác (tối đa 8000 bytes)
IMAGE	Dữ liệu nhị phân với độ dài chính xác (tối đa 2,147,483,647 bytes)
TEXT	Dữ liệu kiểu chuỗi với độ dài lớn (tối đa 2,147,483,647 ký tự)
NTEXT	Dữ liệu kiểu chuỗi với độ dài lớn và hỗ trợ UNICODE (tối đa 1,073,741,823 ký tự)

7

Nội dung

- Tổng quan về SQL
- Giới thiệu về SQL Server
- T-SQL

8

Giới thiệu SQL Server

- SQL Server 7.0, 2000, ..., 2014, 2016,...,2019
- SQL Server 2019:
 - Là một HQT CSDL quan hệ
 - Sử dụng Transact-SQL để trao đổi dữ liệu Client/
 - Các phiên bản:
 - Express
 - Standard
 - Business Intelligence
 - Enterprise
 - Link download: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>




Express

SQL Server 2019 Express is a free edition of SQL Server, ideal for development and production for desktop, web, and small server applications.

[Download now >](#)

[Download SQL Server Management Studio \(SSMS\)](#)

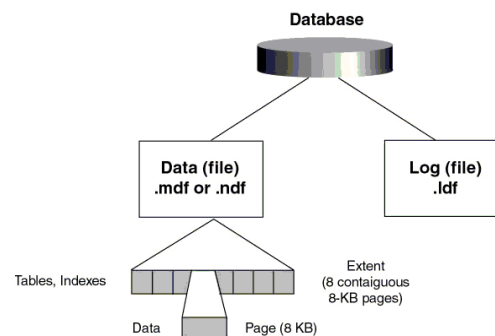
Download SSMS

 [Download SQL Server Management Studio \(SSMS\) 18.9.2](#)

9

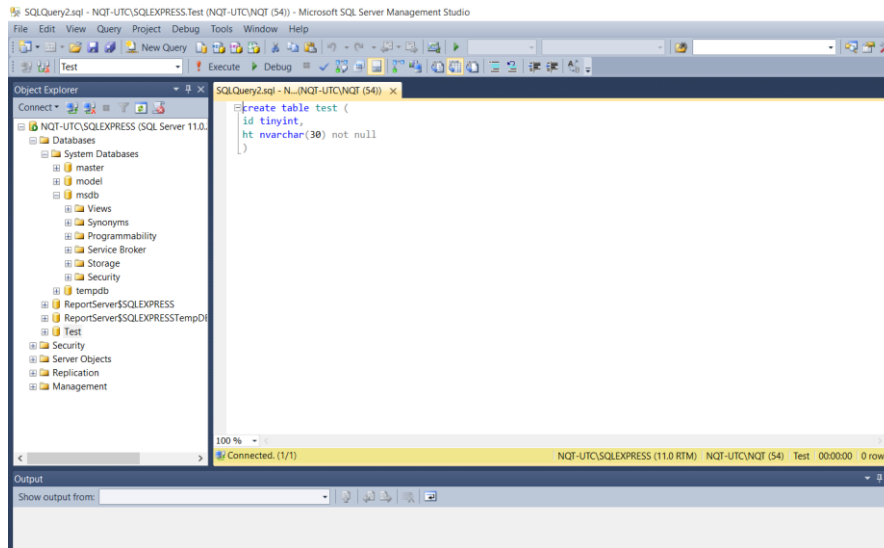
Giới thiệu SQL Server

- Cấu trúc của SQL Server
 - Multi-Instance: Default Instance
 - Mỗi Instance gồm 4 system databases và các users database
 - Master, Tempdb, Model, Msdb
- Cấu trúc vật lý của 1 SQL Server
 - Primary data file (.mdf)
 - Secondary data file (.ndf)
 - Transaction log file (.ldf)



10

Giới thiệu SQL Server



11

Nội dung

- Tổng quan về SQL
- Giới thiệu về SQL Server
- T-SQL

12

T-SQL

- Giới thiệu T-SQL
- Ngôn ngữ định nghĩa dữ liệu (DDL)
- Ngôn ngữ thao tác dữ liệu

13

T-SQL

- Transact-SQL
- Là ngôn ngữ mở rộng của SQL chuẩn của ISO và ANSI
- Khác với P-SQL(Procedural - SQL) dùng trong Oracle

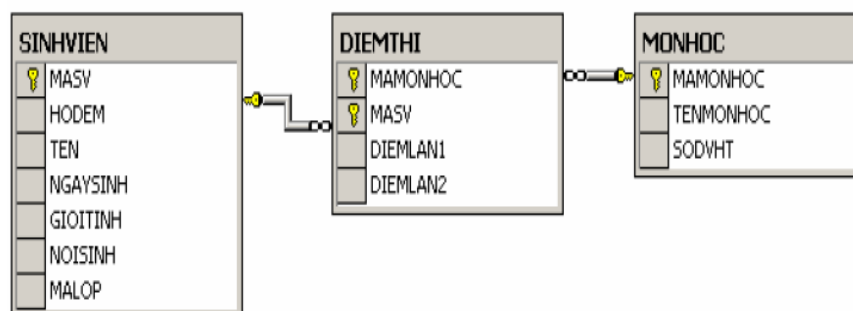
14

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

15



16

Tạo, xóa CSDL

- *Tạo Cơ sở dữ liệu*
 - *Cú pháp*
`CREATE DATABASE <Tên cơ sở dữ liệu> [;]`
 - *Ví dụ*
`CREATE DATABASE QuanlyDiem`
- *Sử dụng Cơ sở dữ liệu*
 - *Cú pháp*
`USE <Tên cơ sở dữ liệu>`
 - *Ví dụ*
`USE QuanlyDiem`
- *Xóa Cơ sở dữ liệu*
 - *Cú pháp*
`DROP DATABASE <Tên cơ sở dữ liệu>`
 - *Ví dụ*
`DROP DATABASE QuanlyDiem`

17

Tạo bảng - CREATE TABLE

- *Cú pháp* `CREATE TABLE tên_bảng`

```
(
    tên_cột    thuộc_tính_cột    các_ràng_buộc
    [, ...
    ,tên_cột_n thuộc_tính_cột_n  các_ràng_buộc_cột_n]
    [, các_ràng_buộc_trên_bảng]
)
```
- *Ví dụ*

```
create table SINHVIEN(
MaSV char(9) not null,
HoDem nvarchar(30) not null,
TenSV nvarchar(10) not null,
NgàySinh date,
NoiSinh nvarchar(100),
GioiTinh bit,
MaLop char(10)
)
```

18

CREATE TABLE

Ràng buộc PRIMARY KEY (Khóa chính)

- *Cú pháp*

```
[CONSTRAINT tên_ràng_buộc]
```

- *Ví dụ* PRIMARY KEY [(danh_sách_cột)]

```
create table SINHVIEN(
  MaSV varchar(15)
  constraint pk_SINHVIEN_MaSV
  primary key ,
  Hodem nvarchar(30),
  Ten nvarchar(10),
  NgaySinh date,
  GioiTinh bit,
  NoiSinh nvarchar(100),
  MaLop char(15)
)
```

```
create table DIEMTHI(
  MaSV varchar(15),
  MaMonHoc varchar(15),
  DiemLan1 numeric(2,2),
  DiemLan2 numeric(2,2)
  constraint pk_DIEMTHI primary key
  (MaSV, MaMonHoc)
)
```

19

CREATE TABLE

Ràng buộc FOREIGN KEY

- *Cú pháp*

```
[CONSTRAINT tên_ràng_buộc]
FOREIGN KEY [(danh_sách_cột)]
REFERENCES tên_bảng_tham_chiếu(danh_sách_cột_tham_chiếu)
[ON DELETE CASCADE | NO ACTION | SET NULL | SET DEFAULT]
[ON UPDATE CASCADE | NO ACTION | SET NULL | SET DEFAULT]
```

- *Ví dụ: Tạo bảng Điểm thi*

20

CREATE TABLE

Ràng buộc FOREIGN KEY

```
create table DIEM(
  MaMH varchar(5),
  MaSV varchar(15),
  DiemTP float,
  DiemKTHP float
  constraint pk_DIEM primary key (MaMH, MaSV),
  constraint fk_DIEM_MONHOC
  foreign key (MaMH)
  references MONHOC(MaMH),
  constraint fk_DIEM_SINHVIEN
  foreign key (MaSV)
  references SINHVIEN(MaSV)
)
```

21

CREATE TABLE

Ràng buộc UNIQUE

• Cú pháp

```
[CONSTRAINT tên_ràng_buộc]
UNIQUE [ (danh_sách_cột) ]
```

• Ví dụ

```
CREATE TABLE lop
(
  malop      NVARCHAR(10)      NOT NULL,
  tenlop     NVARCHAR(30)      NOT NULL,
  khoa       SMALLINT          NULL,
  hedaotao   NVARCHAR(25)      NULL,
  namnhaphoc INT               NULL,
  makhhoa    NVARCHAR(5),
  CONSTRAINT pk_lop PRIMARY KEY (malop),
  CONSTRAINT unique_lop_tenlop UNIQUE(tenlop)
)
```

22

CREATE TABLE

Ràng buộc DEFAULT

- *Cú pháp*

```
[CONSTRAINT tên_ràng buộc]
DEFAULT(giá trị) FOR Tên_trường
```

- *Ví dụ*

```
ALTER TABLE chitietdathang
ADD
    CONSTRAINT df_chitietdathang_soluong
    DEFAULT(1) FOR soluong,
```

23

CREATE TABLE

Ràng buộc CHECK

- *Cú pháp*

```
[CONSTRAINT tên_ràng buộc]
CHECK (điều kiện)
```

- *Ví dụ*

```
ALTER TABLE nhanvien
ADD
    CONSTRAINT chk_nhanvien_ngaylamviec
    CHECK (datediff(yy,ngaysinh,ngaylamviec)
        BETWEEN 18 AND 60)
```

24

Sửa nội dung bảng- ALTER TABLE

- *Bổ sung một cột vào bảng*
- *Xóa một cột khỏi bảng*
- *Thay đổi định nghĩa của một cột trong bảng*
- *Xóa hoặc bổ sung ràng buộc cho bảng*

25

ALTER TABLE

```
ALTER TABLE tên_bảng
  ADD định_nghĩa_cột |
  ALTER COLUMN tên_cột kiểu_dữ_liệu [NULL | NOT NULL] |
  DROP COLUMN tên_cột |
  ADD CONSTRAINT tên_ràng_buộc định_nghĩa_ràng_buộc |
  DROP CONSTRAINT tên_ràng_buộc
```

26

ALTER TABLE

- Ví dụ: Cho định nghĩa 2 bảng NHANVIEN và DONVI như sau

```
CREATE TABLE nhanvien
(
  manv          NVARCHAR(10)    NOT NULL,
  hoten         NVARCHAR(30)    NOT NULL,
  ngaysinh      DATETIME,
  diachi        CHAR(30)        NOT NULL
)

CREATE TABLE donvi
(
  madv          INT             NOT NULL PRIMARY KEY,
  tendv         NVARCHAR(30)    NOT NULL
)
```

27

ALTER TABLE

- Bài tập
 - Bổ sung thêm cột MADV cho bảng NHANVIEN
 - Định nghĩa lại kiểu dữ liệu của cột DIACHI trong bảng NHANVIEN cho phép nhận giá trị NULL
 - Xóa cột ngày sinh khỏi bảng NHANVIEN
 - Định nghĩa khóa chính cho bảng NHANVIEN là trường MANV
 - Định nghĩa khóa ngoài của bảng NHANVIEN trên cột MADV của bảng DONVI

28

ALTER TABLE

```

ALTER TABLE nhanvien
ADD
    dienthoai NVARCHAR(6)
    CONSTRAINT chk_nhanvien_dienthoai
    CHECK (dienthoai LIKE '[0-9][0-9][0-9][0-9][0-9][0-9]')

ALTER TABLE nhanvien
ADD
    madv INT NULL

ALTER TABLE nhanvien
ALTER COLUMN diachi NVARCHAR(100) NULL

```

29

ALTER TABLE

```

ALTER TABLE nhanvien
ADD
    CONSTRAINT pk_nhanvien PRIMARY KEY(manv)

ALTER TABLE nhanvien
ADD
    CONSTRAINT fk_nhanvien_madv
    FOREIGN KEY(madv) REFERENCES donvi(madv)
    ON DELETE CASCADE
    ON UPDATE CASCADE

```

30

Xóa bảng - DROP TABLE

• *Cú pháp*

```
DROP TABLE tên_bảng
```

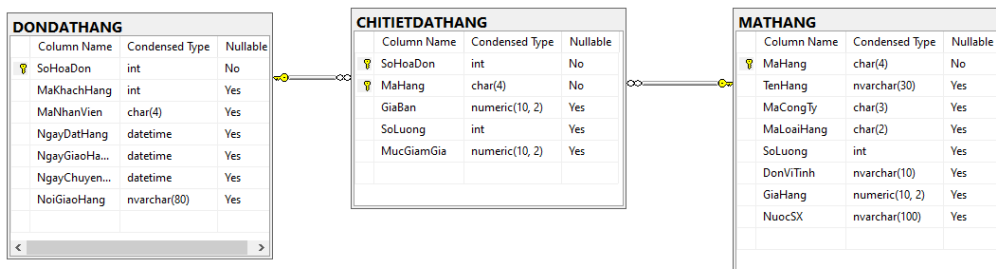
• *Ghi chú:*

- Câu lệnh này sẽ xóa các ràng buộc, chỉ mục, trigger liên quan đến bảng cần xóa
- Khi xóa bằng lệnh DROP không thể khôi phục lại được
- Không thể thực hiện được nếu vẫn còn ràng buộc về khóa ngoại
- Ví dụ: Xóa bảng DONVI

```
ALTER TABLE nhanvien
DROP CONSTRAINT fk_nhanvien_madv

DROP TABLE donvi
```

31



32

T-SQL

- Giới thiệu T-SQL
- Ngôn ngữ định nghĩa dữ liệu (DDL)
- Ngôn ngữ thao tác dữ liệu

33

Ngôn ngữ thao tác dữ liệu DML – Data Manipulation Language

- SELECT
- INSERT
- UPDATE
- DELETE

34

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]]
```

35

SELECT

- Ví dụ

```
SELECT manv, hoten, ngaysinh
FROM nhanvien
```

36

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

37

SELECT – Truy vấn thường

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

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

- Ví dụ:

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

38

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
```

39

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

40

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

41

Các toán tử so sánh

Toán tử	ý nghĩa
=	Bằng
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng
<>	Khác
!>	Không lớn hơn
!<	Không nhỏ hơn

42

BETWEEN

• Kiểm tra dữ liệu trong phạm vi nào đó

Cách sử dụng	Ý nghĩa
giá_trị BETWEEN a AND b	$a \leq \text{giá_trị} \leq b$
giá_trị NOT BETWEEN a AND b	$(\text{giá_trị} < a) \text{ AND } (\text{giá_trị} > b)$

• Ví dụ

```
SELECT hodem,ten,year(getdate())-year(ngaysinh) AS tuoi
FROM sinhvien
WHERE ten='Binh' AND
      YEAR(GETDATE())-YEAR(ngaysinh) BETWEEN 20 AND 22
```

43

Từ khóa LIKE

Ký tự đại diện	ý nghĩa
%	Chuỗi ký tự bất kỳ gồm không hoặc nhiều ký tự
_	Ký tự đơn bất kỳ
[]	Ký tự đơn bất kỳ trong giới hạn được chỉ định (ví dụ [a-f]) hay một tập (ví dụ [abcdef])
[^]	Ký tự đơn bất kỳ không nằm trong giới hạn được chỉ định (ví dụ [^a-f] hay một tập (ví dụ [^abcdef])).

```
SELECT hodem,ten FROM sinhvien
WHERE hodem LIKE 'Lê%'
```

44

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

45

Tạo bảng mới từ kết quả của SELECT

- Câu lệnh SELECT ... INTO
 - Tạo bảng mới có cấu trúc và dữ liệu được xác định từ kết quả của truy vấn
 - Bảng mới có số cột bằng số cột chỉ định và số dòng là số dòng kết quả truy vấn

- Ví dụ

```
SELECT hodem,ten, YEAR(GETDATE())-YEAR(ngaysinh) AS tuoi
      INTO tuoisv
      FROM sinhvien
```

46

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

47

Sắp xếp kết quả truy vấn

- Sử dụng ORDER BY
- Tối đa 16 cột

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

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

48

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

49

Phép nối

- Phép nối bằng (equi-joint) 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

50

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'
```

51

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`

52

Phép nối ngoài

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

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

53

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

54

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

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

55

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

56

Phép nối ngoài (FULL OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Đội ngoại
2	Hành chính
3	Kế toán
4	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
```

57

Phép nối ngoài (FULL OUTER JOIN)

Bảng DONVI

MADV	TENDV
1	Đội ngoại
2	Hành chính
3	Kế toán
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	Đội ngoại
Hoa	2	2	Hành chính
Nam	2	2	Hành chính
Vinh	1	1	Đội ngoại
Hung	5	NULL	NULL
Phuong	NULL	NULL	NULL
NULL	NULL	4	Kinh doanh
NULL	NULL	3	Kế toán

58

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'
```

59

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

60

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

61

Thống kê dữ liệu với GROUP BY

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

62

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 '
```

63

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.

64

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
```

65

Thống kê dữ liệu với GROUP BY

- Thống kê trên các nhóm

66

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

```
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
```

67

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

68

Thống kê dữ liệu với COMPUTE

- 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

- COMPUTE

- Sử dụng kết hợp hàm gộp và mệnh đề ORDER BY trong SELECT để cho biết kết quả của từng nhóm và chi tiết trong mỗi nhóm

69

Thống kê dữ liệu với COMPUTE

- Cú pháp

```
COMPUTE hàm_gộp(tên_cột) [,..., hàm_gộp (tên_cột)]
BY danh_sách_cột
```

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 khoa.makhoa,tenkhoa,malop,tenlop FROM khoa,lop
WHERE khoa.makhoa=lop.makhoa
ORDER BY khoa.makhoa
COMPUTE COUNT(malop) BY khoa.makhoa
```

70

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
-
-
- Truy vấn con

71

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])
```

72

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

73

Truy vấn con (Subquery)

- Ví dụ

```
SELECT *
FROM monhoc
WHERE sodvht>=(SELECT sodvht
                FROM monhoc
                WHERE mamonhoc='TI-001')

SELECT hodem,ten
FROM sinhvien JOIN lop ON sinhvien.malop=lop.malop
WHERE tenlop='Tin K25' AND
      ngaysinh<ALL(SELECT ngaysinh
                  FROM sinhvien JOIN lop
                  ON sinhvien.malop=lop.malop
                  WHERE lop.tenlop='Toán K25')
```

74

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)
```

75

Ngôn ngữ thao tác dữ liệu DML – Data Manipulation Language

- SELECT
- INSERT
- UPDATE
- DELETE

76

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

77

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')
```

78

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 kháchhang(makh, hokh, tenkh)

Select manv, honv, tennv from NhanVien where...

79

Ngôn ngữ thao tác dữ liệu DML – Data Manipulation Language

- SELECT

- UPDATE

- DELETE

80

Bổ sung 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      UPDATE nhatkypdong
SET sodvht = 3      SET tienphong=songay*CASE WHEN loaiphong='A' THEN 100
WHERE sodvht = 2      WHEN loaiphong='B' THEN 70
                        ELSE 50
                        END
```

81

Bổ sung dữ liệu: UPDATE

- Ví dụ

```
UPDATE nhatkymbanhang
SET thanh tien = soluong*gia
FROM mathang
WHERE nhatkymbanhang.mahang = mathang.mahang

UPDATE nhatkymbanhang
SET thanh tien = soluong*gia
FROM mathang
WHERE mathang.mahang =(SELECT mathang.mahang
                        FROM mathang
                        WHERE mathang.mahang=nhatkymbanhang.mahang)
```

82

Ngôn ngữ thao tác dữ liệu DML – Data Manipulation Language

- SELECT
- INSERT
- UPDATE
- DELETE

83

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)
```

84

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
```

85

Bài tập

Cho CSDL QuanlyBanhang.

Hãy viết các câu lệnh SQL thực hiện các truy vấn sau:

1. Cho biết danh sách các đối tác cung cấp hàng cho công ty.
2. Mã hàng, tên hàng và số lượng hiện có trong công ty
3. Địa chỉ, số điện thoại của nhà cung cấp có tên giao dịch VINAMILK là gì?
4. Cho biết mã và tên các mặt hàng có giá lớn hơn 100000 và số lượng hiện có ít hơn 50
5. Đơn đặt hàng số 1 do ai đặt, do nhân viên nào lập, thời gian và địa điểm giao hàng ở đâu?
6. Hiển thị những nhân viên có lương cơ bản cao nhất công ty
7. Nhân viên nào trong công ty bán được nhiều hàng nhất và số lượng bán được là bao nhiêu?
8. Hãy cho biết tổng số hàng của mỗi loại hàng
9. Tăng lương lên 50% cho những nhân viên bán được số lượng hàng ≥ 100 trong năm 2007.

86

Bài tập

10. Xóa những đơn hàng có ngày đặt hàng trước năm 2005 ra khỏi CSDL
11. Xóa khỏi bảng NHANVIEN những nhân viên đã làm việc cho công ty trên 40 năm