

CƠ SỞ DỮ LIỆU

CHƯƠNG 4

NGÔN NGỮ SQL

NGÔN NGỮ CỎN DỮ LIỆU SQL

- SQL (Structure English Query Language)
- Tên gọi cũ là SEQUEL
- Được ANSI và ISO thừa nhận là ngôn ngữ chuẩn về dữ liệu; là phương tiện giao tiếp của nhiều hệ thống thông tin quản lý.
- Ra đời năm 1976 tại phòng thí nghiệm Chamberlin;
- SQL-86 hay SQL1(ANSI 1986) → SQL-92 hay SQL2 → SQL3.

NGÔN NGỮ CON DỮ LIỆU SQL

■ Đặc điểm:

- ◆ Là ngôn ngữ có cấu trúc, phi thủ tục
- ◆ Vừa đóng vai trò là ngôn ngữ con cá thể thao tác độc lập cho người sử dụng cuối, đồng thời có thể nhúng trong ngôn ngữ lập trình.

■ Chức năng của SQL

- ◆ Hỏi dữ liệu (Truy vấn)
- ◆ Định nghĩa dữ liệu
- ◆ Cập nhật dữ liệu
- ◆ Bảo mật và an toàn dữ liệu

NGÔN NGỮ CỎN DỮ LIỆU SQL

■ Thuật ngữ

- | | | |
|---|------------------------|--------------|
| ◆ | Quan hệ (Relation) | Bảng (Table) |
| ◆ | Thuộc tính (Attribute) | Cột (Column) |
| ◆ | Bộ (Tuple) | Hàng (Row) |

Không quan trọng thứ tự các cột và thứ tự các hàng !

SQL Data Definition Language

- DDL là một thành phần trong SQL, chứa các câu lệnh tạo/xoá bảng, định nghĩa indexes, thiết lập các ràng buộc quan hệ
- Các câu lệnh DDL quan trọng:
 - ◆ CREATE TABLE – tạo bảng mới
 - ◆ ALTER TABLE – thay đổi cấu trúc bảng
 - ◆ DROP TABLE – xoá bảng
 - ◆ CREATE INDEX – lập chỉ mục (index)
 - ◆ DROP INDEX – xoá chỉ mục

NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU TRONG SQL

■ Tạo bảng

CREATE TABLE < tên bảng>

(<tên cột 1> <kiểu dl> [NOT NULL] [DEFAULT <gt mặc định>,

<tên cột 2> <kiểu dl> [NOT NULL] [DEFAULT <gt mặc định>,

....

<tên cột n> <kiểu dl> [NOT NULL] [DEFAULT <gt mặc định>

[, PRIMARY KEY (<tên các cột tham gia vào khóa chính>)]

[, UNIQUE (<tên các cột tham gia vào khóa duy nhất>)]

*[, FOREIGN KEY (<tên cột là khóa ngoại lai>) REFERENCES <tên
bảng> (<tên cột liên kết>) [, FOREIGN KEY.....]]);*

NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU

- Đặt tên các ràng buộc: bổ sung thêm thành phần sau vào câu lệnh tạo bảng, đặt trước mỗi phần khai báo ràng buộc:

CONSTRAINT <tên ràng buộc>

- Ví dụ Xét CSDL gồm 3 bảng:
SV (MASV, HOTEN, GT, NS, QUE, LOP)
HP (MAHP, TENHP, SOTC)
KQHT (MASV, MAHP, DIEM)

NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU

```
CREATE TABLE SV
(MASV          CHAR(7) ,
HOTEN          VARCHAR(30) NOT NULL,
GT             CHAR(3),
NS             DATE,
QUE            VARCHAR(20),
LOP            CHAR(10),
PRIMARY KEY (MASV));
```


NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU

```
CREATE TABLE HP  
(MAHP CHAR(5),  
TENHP VARCHAR(20) NOT NULL,  
SOTC SMALLINT NOT NULL,  
PRIMARY KEY (MAHP),  
CONSTRAINT TEN_DUY_NHAT  
UNIQUE (TENHP));
```

NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU

```
CREATE TABLE KQHT
(MASV          CHAR(7),
MAHP           CHAR(5),
DIEM           DECIMAL(4,1) DEFAULT 0,
PRIMARY KEY    (MASV, MAHP),
FOREIGN KEY    (MASV) REFERENCES SV(MASV),
FOREIGN KEY    (MAHP) REFERENCES HP(MAHP));
```

NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU

■ Thay đổi cấu trúc bảng

- ◆ Thêm một cột mới cho bảng

ALTER TABLE <tên bảng>

ADD COLUMN <tên cột> <kiểu dữ liệu> [NOT NULL] [DEFFAULT <giá trị>];

- ◆ Xóa một cột của bảng

ALTER TABLE <tên bảng>

DROP COLUMN <tên cột> [CASCADE];

- ◆ Thay đổi kiểu dữ liệu của một cột trong bảng

ALTER TABLE <tên bảng>

CHANGE COLUMN <tên cột> <kiểu dữ liệu mới>;

NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU

■ Quy định DEFAULT cho một cột

◆ Thêm quy định DEFAULT

ALTER TABLE <tên bảng>

ALTER <tên cột> SET DEFAULT <giá trị>;

◆ Xóa quy định DEFAULT

ALTER TABLE <tên bảng>

ALTER <tên cột> DROP DEFAULT;

NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU

■ Thay đổi các ràng buộc trên bảng

- ◆ Xóa ràng buộc

ALTER TABLE <tên bảng>

DROP CONSTRAINT<tên ràng buộc>;

- ◆ Thêm ràng buộc

ALTER TABLE <tên bảng>

*ADD (CONSTRAINT <tên ràng buộc> PRIMARY KEY...| UNIQUE...|
FOREIGN KEY.....);*

■ Xóa bảng

DROP < tên bảng> [CASCADE];

NGÔN NGỮ THAO TÁC DỮ LIỆU



- **SQL Data Manipulation Language**
- **Là cú pháp để bổ sung, xoá, sửa chữa các hàng trong bảng và tìm kiếm dữ liệu.**
- **Các câu lệnh trong DML**
 - ◆ SELECT trích rút dữ liệu từ các bảng csdl
 - ◆ UPDATE sửa chữa dữ liệu trong bảng
 - ◆ DELETE xoá dữ liệu trong bảng
 - ◆ INSERT INTO chèn thêm dữ liệu mới vào bảng

CẬP NHẬT DỮ LIỆU CHO BẢNG

■ Bổ sung thêm một hàng cho bảng

*INSERT INTO <tên bảng> [(<danh sách các cột>)]
VALUES (<danh sách các giá trị tương ứng>
|<câu truy vấn >;*

■ Ví dụ

INSERT INTO HP VALUES ('M4', 'Toán rời rạc', 4);

Hoặc

INSERT INTO HP (TENHP, MAHP, SOTC)

VALUES ('Toán rời rạc', 'M4', 4);

INSERT INTO SV(MASV, HOTEN) VALUES ('BK1005', 'Hoa');

CẬP NHẬT DỮ LIỆU CHO BẢNG

■ Xóa các hàng trong bảng

DELETE FROM <tên bảng>

[WHERE <biểu thức điều kiện>];

■ Thay đổi giá trị cho các cột trong bảng

UPDATE <tên bảng>

SET <tên cột 1> = <biểu thức>

[, <tên cột2> = <biểu thức>, ...]

[WHERE <biểu thức điều kiện>];

TÌM KIẾM DỮ LIỆU

SELECT [DISTINCT] <danh sách chiếu>

FROM <danh sách tên bảng|tên view>

[WHERE [NOT] <biểu thức điều kiện>]

[GROUP BY <danh sách cột | biểu thức>

[HAVING [NOT] <biểu thức điều kiện>]]

***[ORDER BY < danh sách tên cột| số thứ tự các cột
trong danh sách chiếu|biểu thức> [ASC|DESC]];***

TÌM KIẾM DỮ LIỆU

■ Biểu thức

Một biểu thức được tạo nên từ các hằng, các tên cột và các hàm, được liên kết với nhau bởi các phép tính số học (+, -, *, /)

- Danh sách chiếu:

Danh sách các tên cột (hoặc biểu thức) cần có trong bảng kết quả. Khi danh sách chiếu là dấu *, có nghĩa danh sách chiếu là toàn bộ các cột của tất cả các bảng (hoặc view) được chỉ ra sau FROM

TÌM KIẾM DỮ LIỆU

■ Biểu thức điều kiện

- (a) **biểu thức logic** tạo nên từ các hằng, các tên cột và các hàm, được nối với nhau bởi các phép toán số học so sánh (>, >=, <, <=, !=) và logic (and, or, not)
- (b) **tân từ** ở các dạng sau:
 - (b1) <tên cột> **IS [NOT] NULL**
 - (b2) <tên cột> **[NOT] LIKE** <xâu kí tự>
 - (b3) <biểu thức> **[NOT] BETWEEN** <biểu thức 1>
AND <biểu thức2>
 - (b4) <tên cột> **[NOT] IN** (<danh sách các giá trị | truy vấn con>)
 - (b5) <tên cột> **θ ANY | ALL** (truy vấn con)
 - (b6) **EXISTS** (truy vấn con)
- (c) **biểu thức logic kết hợp với tân từ** bởi các phép toán logic

■ Ví dụ một trạng thái của CSDL

Quan hệ SV

MASV	HOTEN	NS	GT	QUE	LOP
BK1000	Nguyễn Minh Đức	1/1/1985	Nam	Hà Tây	Toán 1
BK1001	Trần Mai Ly	2/2/1986	Nữ	Hà Nội	Toán 1
BK1002	Lê Phương Nam	3/3/1985	Nam	Nam Hà	Toán 2

Quan hệ HP

MAHP	TENHP	SOTC
M1	Cơ sở dữ liệu	4
M2	Toán rời rạc	3
M3	Mạng MT	4

Quan hệ KQHT

MASV	MAHP	DIEM
BK1000	M1	4
BK1001	M1	6
BK1001	M2	7
BK1001	M3	5

TÌM KIẾM DỮ LIỆU

■ Tìm kiếm đơn giản

- Ví dụ 1: Cho biết danh sách tất cả các sinh viên

SELECT * FROM SV;

- Ví dụ 2: Cho biết mã số những sinh viên đã có ít nhất một điểm thi

SELECT DISTINCT MASV FROM KQHT;

∩ Biểu thức quan hệ tương đương

$\pi_{MASV}(KQHT)$

**Bảng
KQHT**

MASV	MAHP	DIEM
BK1000	M1	4
BK1001	M1	6
BK1001	M2	7
BK1001	M3	5

**SELECT MASV
FROM KQHT;**

MASV
BK1000
BK1001
BK1001
BK1001

**SELECT DISTINCT MASV
FROM KQHT;**

MASV
BK1000
BK1001

TÌM KIẾM DỮ LIỆU

■ Tìm kiếm với biểu thức logic dạng (a)

- Ví dụ 3: Cho biết mã số, tên và ngày sinh của các sinh viên nữ quê ở Hà Nội

SELECT MASV, HOTEN, NS

FROM SV

WHERE GT = 'nữ' and QUE = 'Hà Nội';

⌚ Biểu thức quan hệ tương đương

$\pi_{MASV, HOTEN, NS}(\sigma_{GT = 'Nữ' \wedge QUE = 'Hà Nội'}(SV))$

TÌM KIẾM DỮ LIỆU

■ Tìm kiếm với biểu thức điều kiện dạng (b1)

- Ví dụ 4: Cho biết danh sách mã số và tên những sinh viên chưa rõ quê quán

```
SELECT MASV, HOTEN  
FROM SV  
WHERE QUE IS NULL;
```

- Ví dụ 5: Cho biết thông tin về những sinh viên có họ ‘Nguyễn’

TÌM KIẾM DỮ LIỆU

■ Tìm kiếm với biểu thức điều kiện dạng (b2)

Khi dùng *LIKE* < *xâu kí tự*>, trong *xâu kí tự* có thể chứa các kí tự thay thế:

‘%’ thay thế cho một *xâu kí tự*

‘_’ thay thế cho một *kí tự*

➤ Ví dụ 5 *SELECT **
 FROM SV
 WHERE HOTEN LIKE ‘Nguyễn%’;

TÌM KIẾM DỮ LIỆU

■ Sử dụng dữ liệu kiểu ngày tháng

- Ví dụ 6: Cho biết thông tin về những sinh viên sinh vào tháng 12 năm 1979

```
SELECT * FROM SV  
WHERE '01/01/1980' - NS <= 31 AND  
      '01/01/1980' - NS > 0;
```

■ Tìm kiếm với biểu thức điều kiện dạng b3)

```
SELECT      * FROM SV  
WHERE      NS BETWEEN '01/12/1979' AND  
            '31/12/1979';
```

TÌM KIẾM DỮ LIỆU

■ Tìm kiếm với biểu thức điều kiện dạng (b3) và (c)

- Ví dụ 7: Cho biết mã số, họ tên và ngày sinh của những sinh viên lớp 'Toán 1' sinh vào tháng 12 năm 1979

```
SELECT  MASV, HOTEN, NS FROM SV  
WHERE   LOP = 'Toán 1' and NS BETWEEN  
        '01/12/1979' AND '31/12/1979';
```

- ◆ *NS BETWEEN '01/12/1979' AND '31/12/1979' tương đương:*
- ◆ *NS >= '01/12/1979' AND NS <= '31/12/1979'*

TÌM KIẾM DỮ LIỆU

■ Biểu diễn phép kết nối

- ◆ mô tả biểu thức kết nối trong mệnh đề WHERE
- ◆ Hoặc dùng phép JOIN trong mệnh đề FROM
- Lưu ý: Khi một cột A trong danh sách chiếu cùng xuất hiện trong nhiều bảng được chỉ ra sau mệnh đề FROM, để tránh sự nhập nhằng, phải ghi rõ T.A, có nghĩa cột A của bảng T

TÌM KIẾM DỮ LIỆU – Phép kết nối

- Sơ đồ chung: Kết nối tự nhiên R, S dựa trên cột A

- Đại số quan hệ: $\pi_{ABC}(\sigma_E(R \bowtie S))$

- SQL:
SELECT R.A, B,...
FROM R, S
WHERE E AND R.A = S.A;

Hoặc

- SELECT R.A, B,...*
FROM R JOIN S ON R.A = S.A
WHERE E;

TÌM KIẾM DỮ LIỆU – Phép kết nối

- Ví dụ 8: Cho biết mã số và tên những sinh viên thi học phần mã số 'm1' đạt điểm ≥ 4

■ Cách viết 1

SELECT SV.MASV, HOTEN

FROM SV, KQHT

WHERE SV.MASV=KQHT.MASV AND MAHP ='m1'

AND DIEM ≥ 4 ;

TÌM KIẾM DỮ LIỆU – Phép kết nối

■ Cách viết 2

```
SELECT SV.MASV, HOTEN  
FROM (SV JOIN KQHT ON SV.MASV = KQHT.MASV)  
WHERE MAHP = 'm1' AND DIEM >= 4;
```

■ Cách viết 3

```
SELECT SV.MASV, HOTEN  
FROM (SV NATURAL JOIN KQHT)  
WHERE MAHP = 'm1' AND DIEM >= 4;
```

■ Kết nối $\pi_{ABC}(\sigma_E(R \bowtie_{A \theta B} S))$

SELECT A, B, C,.... FROM R, S

WHERE E AND A θ B;

? Biểu thức đại số quan hệ nào tương đương với:

SELECT * FROM S, R WHERE E;

TÌM KIẾM DỮ LIỆU – Phép kết nối

■ INNER JOIN

- ◆ R INNER JOIN S: Kết quả chỉ bao gồm những hàng tạo bởi từ những hàng của R và những hàng của S mà kết nối được với nhau (JOIN mặc định là INNER JOIN).

■ OUTER JOIN

- ◆ LEFT OUTER JOIN,
- ◆ RIGHT OUTER JOIN,
- ◆ FULL OUTER JOIN

TÌM KIẾM DỮ LIỆU – Phép kết nối

■ ***R LEFT OUTER JOIN S*** cho kết quả gồm:

- ◆ những hàng tạo bởi từ những hàng của R và những hàng của S mà kết nối được với nhau,
- ◆ những hàng của R và các cột còn lại mang giá trị NULL nếu không có hàng nào trong S kết nối được với những hàng này.

■ ***Mặc định OUTER JOIN là LEFT OUTER JOIN***

- ◆ *R RIGHT OUTER JOIN S*
- ◆ *R FULL OUTER JOIN S*

TÌM KIẾM DỮ LIỆU – Phép kết nối

- Ví dụ 9: Cho biết mã số, tên sinh viên, mã học phần và điểm thi của tất cả các sinh viên. Với những sinh viên chưa có điểm thi học phần nào thì giá trị trên cột MAHP và DIEM là NULL.

```
SELECT SV.MASV, HOTEN, MAHP, DIEM  
FROM (SV OUTER JOIN KQHT ON SV.MASV =  
KQHT.MASV);
```

TÌM KIẾM DỮ LIỆU – Phép kết nối

■ SV [LEFT] OUTER JOIN KQHT

MASV	HOTEN	NS
BK1000	Nguyễn Minh Đức	1/1/1985
BK1001	Trần Mai Ly	2/2/1986
BK1002	Lê Phương Nam	3/3/1985

MASV	MAHP	DIEM
BK1000	M1	4
BK1000	M2	6
BK1001	M2	7

MASV	HOTEN	MAHP	DIEM
BK1000	Nguyễn Minh Đức	M1	4
BK1000	Nguyễn Minh Đức	M2	6
BK1001	Trần Mai Ly	M2	7
BK1002	Lê Phương Nam	NULL	NULL

Kết quả kết
nối ngoài

TÌM KIẾM DỮ LIỆU – Phép kết nối

- Ví dụ 10. Cho biết mã số, họ tên và điểm thi của những sinh viên thi học phần 'Cơ sở dữ liệu' đạt điểm ≥ 4 .

```
SELECT SV.MASV, HOTEN, DIEM  
FROM SV, KQHT, HP  
WHERE SV.MASV = KQHT.MASV AND KQHT.MAHP  
= HP.MAHP AND TENHP = 'Cơ sở dữ liệu' AND  
DIEM  $\geq$  4;
```

TÌM KIẾM DỮ LIỆU – Phép kết nối

```
SELECT SV.MASV, HOTEN, DIEM  
FROM ((SV JOIN KQHT ON SV.MASV = KQHT.MASV)  
JOIN HP ON KQHT.MAHP = HP.MAHP) )  
WHERE TENHP = 'Cơ sở dữ liệu' AND DIEM >= 4;
```

Hoặc

```
SELECT SV.MASV, HOTEN, DIEM  
FROM ((SV NATURAL JOIN KQHT) NATURAL JOIN HP)  
WHERE TENHP = 'Cơ sở dữ liệu' AND DIEM >= 4;
```

LUYỆN TẬP – Phép kết nối



- 1) Lập bảng điểm học phần CSDL của lớp “Toán 1” (gồm các cột MASV, HOTEN, DIEM)
- 2) Cho biết điểm học phần Cơ sở dữ liệu của sinh viên “BK1000”.
- 3) Lập bảng điểm của sinh viên “BK1000” (gồm MAHP, TENHP, SOTC, diem)

TÌM KIẾM DỮ LIỆU

SELECT [DISTINCT] <danh sách chiếu>

FROM <danh sách tên bảng|tên view>

[WHERE [NOT] <biểu thức điều kiện>]

[GROUP BY <danh sách cột | biểu thức>

[HAVING [NOT] <biểu thức điều kiện>]]

***[ORDER BY < danh sách tên cột| số thứ tự các cột
trong danh sách chiếu|biểu thức> [ASC|DESC]];***

TÌM KIẾM DỮ LIỆU

■ Biểu thức điều kiện

- (a) **biểu thức logic** tạo nên từ các hằng, các tên cột và các hàm, được nối với nhau bởi các phép toán số học so sánh (>, >=, <, <=, !=) và logic (and, or, not)
- (b) **tân từ** ở các dạng sau:
 - (b1) <tên cột> **IS [NOT] NULL**
 - (b2) <tên cột> **[NOT] LIKE** <xâu kí tự>
 - (b3) <biểu thức> **[NOT] BETWEEN** <biểu thức 1>
AND <biểu thức2>
 - (b4) <tên cột> **[NOT] IN** (<danh sách các giá trị | truy vấn con>)
 - (b5) <tên cột> **θ ANY | ALL** (truy vấn con)
 - (b6) **EXISTS** (truy vấn con)
- (c) **biểu thức logic kết hợp với tân từ** bởi các phép toán logic

TÌM KIẾM DỮ LIỆU – Truy vấn con

- Tìm kiếm với biểu thức điều kiện dạng (b4)
- Ví dụ 11: Cho biết mã số của những sinh viên có điểm <4 ít nhất là một trong 3 học phần m1, m2, m3

```
SELECT DISTINCT MASV  
FROM KQHT  
WHERE      DIEM < 4 AND  
           MAHP IN ('m1', 'm2', 'm3');
```

TÌM KIẾM DỮ LIỆU – Truy vấn con

- Tìm kiếm với biểu thức điều kiện dạng (b5)

Ví dụ 12. Cho biết mã số những sinh viên phải học lại (điểm < 4) ít nhất một học phần có số tín chỉ > 3

SELECT DISTINCT MASV

FROM KQHT

WHERE DIEM < 4 AND

MAHP IN (SELECT MAHP

FROM HP

WHERE SOTC > 3);

- *Hãy viết câu truy vấn trên bằng cách sử dụng phép kết nối*

TÌM KIẾM DỮ LIỆU – Truy vấn con

- Ví dụ 13: Cho biết mã số và tên những sinh viên có điểm < 4 ở ít nhất một học phần có số tín chỉ ≥ 3 .
- Ví dụ 14: Cho biết mã số và tên những sinh viên không có điểm học phần nào < 4.

TÌM KIẾM DỮ LIỆU

■ Tìm kiếm với biểu thức điều kiện dạng (b6)

$A \theta ANY S =$ *true, nếu $\exists X \in S: A \theta X$*
false, nếu ngược lại

$A \theta ALL S =$ *true, nếu $\forall X \in S: A \theta X$*
false, nếu ngược lại

θ là phép toán so sánh: $=, <> (!=), >, >=, <, <=$

TÌM KIẾM DỮ LIỆU

- Ví dụ 15: Cho biết mã số những sinh viên đạt điểm cao nhất học phần 'M1'.

```
SELECT MASV FROM KQHT  
WHERE MAHP = 'm1'  
AND DIEM >= ALL (SELECT DIEM  
FROM KQHT  
WHERE MAHP = 'M1');
```

TÌM KIẾM DỮ LIỆU

MASV	MAHP	DIEM
BK1000	M1	4
BK1000	M2	6
BK1001	M2	7
BK1002	M1	8
BK1003	M3	6
BK1003	M1	9
BK1004	M2	5

DIEM
4
8
9

```
SELECT DIEM FROM  
KQHT  
WHERE MAHP = 'M1');
```

MASV	MAHP	DIEM
BK1000	M1	4
BK1002	M1	8
BK1003	M1	9

TÌM KIẾM DỮ LIỆU

- Nếu kết quả của truy vấn con là một hằng thì không cần sử dụng từ khóa **ANY**, **ALL** trong điều kiện
- Ví dụ 16: Cho biết mã số và tên của những sinh viên có cùng ngày sinh với sinh viên có mã số 'BK1000'

```
SELECT  MASV, HOTEN FROM SV  
WHERE   NS = (SELECT    NS FROM SV  
                WHERE   MASV = 'BK1000');
```


TÌM KIẾM DỮ LIỆU

■ Tìm kiếm với biểu thức điều kiện dạng (b7)

EXISTS < truy vấn con > = true, nếu kết quả của
truy vấn con $\neq \emptyset$
false, nếu ngược lại

- Ví dụ 17: Cho biết mã số và tên của những sinh viên không phải học lại học phần nào.

SELECT MASV, HOTEN

FROM SV WHERE NOT EXISTS

(SELECT * FROM KQHT

WHERE MASV = SV. MASV AND DIEM <4);

MASV	MAHP	DIEM
BK1000	M1	4
BK1002	M1	8
BK1003	M1	9
BK1000	M2	2
BK1001	M2	7
BK1004	M2	3
BK1003	M3	6

***SELECT MASV, HOTEN
FROM SV
WHERE NOT EXISTS
(SELECT * FROM KQHT
WHERE MASV = SV. MASV AND
DIEM <4);***

TÌM KIẾM DỮ LIỆU

- Đặt bí danh cho cột, bảng

***SELECT <tên cột> [AS] <bí danh>, ...
FROM <tên bảng> AS <bí danh>,***

- Ví dụ 18:

***SELECT MASV FROM KQHT AS T
WHERE MAHP = 'm1' AND NOT EXISTS
(SELECT * FROM KQHT
WHERE MAHP = 'm1' AND DIEM > t.DIEM);***

TÌM KIẾM DỮ LIỆU

■ Sử dụng hàm thư viện trong danh sách chiếu

- Ví dụ 19: Cho biết tổng số sinh viên phải học lại ít nhất một học phần nào đó

```
SELECT COUNT (DISTINCT MASV) TSSV_thilai  
FROM KQHT  
WHERE DIEM < 4;
```

- Ví dụ 20:

```
SELECT MASV FROM KQHT  
WHERE      MAHP = 'm1' AND  
DIEM = (    SELECT MAX(DIEM) FROM KQHT  
          WHERE MAHP = 'm1');
```

TÌM KIẾM DỮ LIỆU - *Mệnh đề Order by*

■ Sử dụng mệnh đề ORDER BY

➤ Ví dụ 21:

***SELECT MASV, MAHP, DIEM FROM KQHT
ORDER BY MASV, DIEM DESC;***

■ Ghi chú:

- ◆ Chiều sắp xếp luôn mặc định là ASC
- ◆ Mệnh đề ORDER BY có thể viết:

ORDER BY 1, 3 DESC
- ◆ Khi sắp xếp dựa trên giá trị của nhiều cột thì thứ tự ưu tiên là từ trái sang phải

TÌM KIẾM DỮ LIỆU - Mệnh đề Group by

- Ví dụ 22: Đối với mỗi sinh viên, cho biết số học phần đã thi

```
SELECT MASV, COUNT(MAHP)  
FROM KQHT  
GROUP BY MASV;
```

MASV	MAHP	DIEM
BK1000	M1	4
BK1002	M1	8
BK1003	M1	9
BK1000	M2	2
BK1001	M2	7
BK1004	M2	3
BK1003	M3	6

MASV	MAHP	DIEM
BK1000	M1	4
BK1000	M2	2
BK1001	M2	7
BK1002	M1	8
BK1003	M3	6
BK1003	M1	9
BK1004	M2	3

MASV	count(MAHP)
BK1000	2
BK1001	1
BK1002	1
BK1003	2
BK1004	1

- Ví dụ 23: Đối với mỗi sinh viên, cho biết số học phần có điểm < 4

SELECT MASV, COUNT(MAHP) AS So_mon_TL

FROM KQHT

WHERE DIEM < 4

GROUP BY MASV;

MASV	MAHP	DIEM
BK1000	M1	4
BK1002	M1	8
BK1003	M1	9
BK1000	M2	2
BK1001	M2	7
BK1004	M2	3
BK1003	M3	6

MASV	MAHP	DIEM
BK1000	M1	4
BK1000	M2	2
BK1004	M2	3

MASV	So_mon_TL
BK1000	2
BK1004	1

TÌM KIẾM DỮ LIỆU

SELECT [DISTINCT] <danh sách chiếu>

FROM <danh sách tên bảng|tên view>

[WHERE [NOT] <biểu thức điều kiện>]

[GROUP BY <danh sách cột | biểu thức>

[HAVING [NOT] <biểu thức điều kiện>]]

***[ORDER BY < danh sách tên cột| số thứ tự các cột
trong danh sách chiếu|biểu thức> [ASC|DESC]];***

TÌM KIẾM DỮ LIỆU



■ Cách sử dụng mệnh đề Group by

- ◆ Mệnh đề GROUP BY dùng để phân hoạch dữ liệu trong bảng thành từng nhóm dựa trên giá trị của một hay nhiều cột (cột này gọi là cột nhóm). Mỗi nhóm tương ứng với một giá trị trên cột nhóm.
- ◆ Các hàm nhóm (COUNT, MAX, MIN, SUM, AVG) sẽ có tác dụng trong từng nhóm chứ không phải trên toàn bảng.
- ◆ Nếu khi có mệnh đề GROUP BY mà danh sách chiếu không sử dụng hàm nhóm thì GROUP BY chỉ có tác dụng sắp xếp các hàng thuộc cùng một nhóm xuất hiện liên tiếp nhau.

TÌM KIẾM DỮ LIỆU

- ◆ Khi cần nhóm dữ liệu dựa trên nhiều cột thì thứ tự ưu tiên tính từ trái sang phải và danh sách các cột nhóm cũng phải xuất hiện đầu tiên trong danh sách chiếu theo đúng thứ tự như trong mệnh đề GROUP BY. Chẳng hạn, khi 2 cột nhóm là A, B thì truy vấn phải có dạng sau:

SELECT A, B, <Danh sách hàm nhóm>

.....

GROUP BY A, B;

TÌM KIẾM DỮ LIỆU

- Ví dụ 24: Đối với mỗi lớp, cho biết mã số sinh viên trong lớp và điểm cao nhất mà sinh viên này đạt được.

```
SELECT LOP, SV.MASV, MAX(DIEM)  
FROM SV, KQHT  
WHERE SV.MASV = KQHT.MASV  
GROUP BY LOP, SV.MASV;
```

TÌM KIẾM DỮ LIỆU - *Mệnh đề Having*

■ Sử dụng mệnh đề HAVING

SELECT ... FROM ...

GROUP BY ... HAVING <biểu thức điều kiện>

- ◆ Mệnh đề HAVING chỉ được dùng khi trong truy vấn có mệnh đề GROUP BY.
- ◆ Mệnh đề HAVING dùng để lọc dữ liệu trong từng nhóm dữ liệu được phân hoạch bởi GROUP BY.
- ◆ Cách sử dụng mệnh đề HAVING tương tự như mệnh đề WHERE nhưng miền tác động của biểu thức điều kiện là trong mỗi nhóm được phân hoạch bởi GROUP BY.

TÌM KIẾM DỮ LIỆU

- Ví dụ 25: Cho biết tổng số học phần đã thi của mỗi sinh viên. Chỉ hiển thị trong bảng kết quả về những sinh viên đã thi ít nhất hai học phần.

```
SELECT MASV, COUNT(MAHP) AS SOMON  
FROM KQHT GROUP BY MASV  
HAVING COUNT(MAHP) >= 2;
```

TÌM KIẾM DỮ LIỆU

- Ví dụ 26. Cho biết tổng số học phần có điểm < 4 của mỗi sinh viên. Chỉ hiển thị trong bảng kết quả về những sinh viên có ít nhất 6 học phần có điểm < 4

```
SELECT MASV, COUNT(MAHP) AS SO_MON_THILAI  
FROM KQHT  
WHERE DIEM < 4  
GROUP BY MASV  
HAVING COUNT(MAHP) >= 6;
```

TÌM KIẾM DỮ LIỆU

- Cho biết số sinh viên học lại của mỗi học phần. Chỉ hiển thị những học phần có nhiều hơn 10 sinh viên phải học lại

```
SELECT MAHP, COUNT(MASV) AS SO_SV_THILAI  
FROM KQHT  
WHERE DIEM < 4  
GROUP BY MAHP  
HAVING COUNT(MASV) > 10;
```

TÌM KIẾM DỮ LIỆU

- Ví dụ 27. Cho biết số học phần phải học lại của những sinh viên phải thi lại nhiều hơn sinh viên 'BK1000'

```
SELECT MASV, COUNT(MAHP) FROM KQHT  
WHERE DIEM < 4 GROUP BY MASV  
HAVING COUNT(MAHP) > (SELECT COUNT(MAHP)  
FROM KQHT  
WHERE MASV = 'BK1000'  
AND DIEM < 4);
```


TÌM KIẾM DỮ LIỆU – *Phép tính tập hợp*

- Truy vấn có sử dụng các phép toán tập hợp
(SELECT...FROM...)
UNION | INTERSECT | MINUS | CONTAINS | =
(SELECT....FROM...);
- Phần lớn các cài đặt của SQL không hỗ trợ phép toán so sánh hai tập hợp =, CONTAINS.

TÌM KIẾM DỮ LIỆU

- Ví dụ 28: Cho biết mã số và tên của những sinh viên không phải học lại học phần nào

(SELECT MASV, HOTEN FROM SV)

MINUS

*(SELECT SV.MASV, HOTEN FROM SV, KQHT
WHERE SV.MASV = KQHT.MASV AND DIEM < 4);*

TÌM KIẾM DỮ LIỆU

- Ví dụ 29 (**): Câu hỏi ở ví dụ 28 có thể được hiểu là: cho biết mã số và tên của những sinh viên đạt điểm ≥ 4 ở tất cả các học phần đã thi.

SELECT MASV, HOTEN FROM SV

***WHERE (SELECT MAHP FROM KQHT
WHERE MASV = SV.MASV)***

=

***(SELECT MAHP FROM KQHT
WHERE MASV = SV.MASV AND DIEM \geq 4);***

BÀI TẬP

■ Dùng ngôn ngữ hỏi SQL biểu diễn các câu hỏi sau:

- 1) Cho biết mã số, tên học phần và điểm thi tất cả các học phần của sinh viên mã số là 'BK1000'.
- 2) Cho biết mã số, tên học phần và điểm thi ở những học phần mà sinh viên 'BK1000' phải thi lại ($DIEM < 4$)
- 3) Cho biết mã số và tên của những sinh viên đã thi ít nhất là một trong 3 học phần Cơ sở dữ liệu, Cấu trúc dữ liệu, Mạng máy tính.
- 4) Cho biết mã số và tên của những học phần mà sinh viên mã số là 'BK1000' chưa có điểm

BÀI TẬP



- 5) Cho biết mã số và tên của những sinh viên có điểm thi học phần m1 không thấp nhất khoa
- 6) Cho biết mã số và tên của những sinh viên có điểm thi học phần m1 lớn hơn điểm thi học phần m1 của sinh viên mã số là 'BK1000'
- 7) Cho biết số sinh viên phải học lại học phần Cơ sở dữ liệu
- 8) Đối với mỗi học phần, cho biết mã học phần, tên học phần và số sinh viên phải học lại học phần đó
- 9) Cho biết điểm cao nhất học phần M1 mà các sinh viên đạt được
- 10) Cho biết mã số, tên và lớp của sinh viên đạt điểm cao nhất học phần Cơ sở dữ liệu

BÀI TẬP



- 11) Cho biết điểm trung bình tích lũy của sinh viên mã số 'BK1000'
- 12) Với mỗi sinh viên, cho biết mã số, tên và điểm tích lũy của sinh viên đó
- 13) Đối với mỗi lớp, lập bảng điểm gồm mã số, tên sinh viên và điểm trung bình tích lũy. Sắp xếp danh sách theo chiều giảm của điểm trung bình tích lũy và chiều tăng của HOTEN
- 14) Cho biết mã số và số điểm < 4 của những sinh viên có hơn một nửa số điểm là < 4
- 15) Cho biết mã số và tên những sinh viên có hơn một nửa số điểm là < 4

BÀI TẬP

- 16) Đối với mỗi lớp, cho biết mã số và tên của những sinh viên phải học lại từ 3 học phần trở lên
- 17) Cho biết mã số và tên của những học phần học mà tất cả các sinh viên đều đạt điểm ≥ 4 (hay không có sinh viên phải học lại) (Viết theo 2 cách sử dụng phép so sánh hai tập hợp và không sử dụng phép so sánh 2 tập hợp).
- 18) Cho biết mã số và tên của sinh viên có điểm trung bình tích lũy ≥ 8.0 (giả thiết: thang điểm 10)
- 19) Cho biết mã số và tên của những sinh viên có điểm trung bình tích lũy.
- 20) Cho biết mã số và tên của những sinh viên phải học lại ở ít nhất là những học phần mà sinh viên mã số 'BK1000' phải học lại.

LẬP CHỈ MỤC (INDEX)

- Chỉ mục lưu trữ vị trí các bản ghi trong một bảng, dựa trên giá trị của một hay nhiều trường.
- Mục đích: Tăng tốc độ truy nhập thông tin trong CSDL
- Tạo index

CREATE [UNIQUE] INDEX <tên index> ON <tên bảng>(<tên cột> [ASC|DESC],...);

- Xóa index

DROP INDEX <tên index>;

KHUNG NHÌN (VIEW)

■ Tạo khung nhìn

CREATE VIEW <tên view>[(<danh sách các cột của view>)] AS <câu truy vấn> [WITH READ ONLY][WITH CHECK OPTION];

■ Xóa khung nhìn

DROP VIEW <tên view>;

■ Cập nhật dữ liệu cho view

View là một bảng ảo nên các câu lệnh cập nhật tương tự như đối với bảng.

NGÔN NGỮ KIỂM SOÁT DỮ LIỆU

■ Giao quyền

GRANT <các quyền> ON <đối tượng> TO <user> [WITH GRANT OPTION];

- ◆ Các quyền: ALTER, READ, SELECT, UPDATE, INSERT, INDEX, EXECUTE,...
- ◆ Đối tượng: bảng, view,...
- ◆ User: tên người sử dụng hoặc là PUBLIC để chỉ tất cả người sử dụng.

■ Thu hồi quyền

REVOKE <các quyền> On <đối tượng> FROM <user>;

SQL NHÚNG

- Khả năng được nhúng trong các ngôn ngữ lập trình như: C, C++, COBOL, PASCAL,....(ngôn ngữ chủ).
- Cú pháp của chỉ thị SQL phụ thuộc vào ngôn ngữ chủ.
Ví dụ: C(SQL/C), PASCAL (ALLBASE/SQL)
EXEC SQL <câu truy vấn hoặc thao tác dữ liệu trong SQL>;
Java: executeQuery(" SELECT...."), executeUpdate(),...
- Biến của ngôn ngữ chủ được dùng trong truy vấn SQL dạng nhúng phải có dấu ‘: ‘ đặt trước để phân biệt với các biến của SQL
- Kết quả của truy vấn không được đưa ra ngay mà sau khi được dịch bởi chương trình dịch của ngôn ngữ chủ

BÀI TẬP

- Cho CSDL gồm 3 quan hệ:

MH(MAMH, TENMH, GIA)

KH(MAKH, TENKH, DIACHI)

HD(SOHD, MAKH, NGÀY, TONGTIEN)

CTHD(SOHD, MAMH, GIABAN, SLMUA, THANHTIEN),
trong đó:

MAMH: Mã hàng	MAKH: Mã khách hàng	SOHD: Số hoá đơn
TENMH: Tên mặt hàng	TENKH: Tên khách hàng	NGAY: Ngày mua hàng
GIA: Đơn giá hiện tại của mặt hàng	DIACHI: Địa chỉ khách hàng	SLMUA: Số lượng hàng mua
GIABAN: Giá bán của mặt hàng trên hóa đơn	THANHTIEN: Thành tiền	TONGTIEN: Tổng tiền

BÀI TẬP

- Dùng đại số quan hệ để biểu diễn các câu hỏi sau:
 - 1) Cho biết danh sách tên và địa chỉ của tất cả các khách hàng.
 - 2) Cho biết tên và địa chỉ những khách hàng đã mua mặt hàng tên là 'video' ngày '10/05/2008'.
 - 3) Cho biết mã số và tên những mặt hàng mà trong tháng 9 năm 2008 không có khách hàng nào mua mặt hàng này.
 - 4) Cho biết mã số và tên khách hàng đã mua cả hai mặt hàng “H1” và ‘H2’ trên cùng một hoá đơn vào ngày 9/9/2008

BÀI TẬP

- Dùng ngôn ngữ SQL để biểu diễn các câu hỏi sau:
 - 1) Cho biết danh sách tất cả các mặt hàng (sắp xếp theo chiều tăng của tên hàng).
 - 2) Cho biết mã số và tên khách hàng mua hàng ngày '10/10/2008' với tổng tiền trên hoá đơn lớn hơn hoặc bằng 1 triệu.
 - 3) Cho biết mã số và tên mặt hàng mà trong tháng 10 năm 2008 không có khách hàng nào mua mặt hàng này.
 - 4) Cho biết mã số và tên khách hàng mua mặt hàng mã số 'H1' với số lượng nhiều nhất (trên 1 hoá đơn) trong ngày '10/10/2008'.
 - 5) Cho biết tổng số tiền bán hàng trong ngày 10/10/2008.
 - 6) Đối với mỗi mặt hàng được bán trong ngày 10/10/2008, cho biết mã số mặt hàng, tên hàng và tổng số tiền hàng.