



Các ngôn ngữ dữ liệu đối với mô hình quan hệ

Ngôn ngữ SQL

Nguyễn Thị Oanh

oanhnt@soict.hust.edu.vn

Bộ môn Hệ thống thông tin (<http://is.hust.edu.vn/>)

Viện CNTT&TT

Trường Đại học Bách Khoa Hà Nội

Hệ

Ứng dụng

- Định nghĩa DL
- Thao tác DL

Hệ QTCSDL

CSDL

CSDL

Nội dung

- Đặt vấn đề
 - Giới thiệu một số ngôn ngữ và phân loại
 - So sánh và đánh giá
- Ngôn ngữ dữ liệu mức thấp:
 - **Đại số quan hệ**
 - Tính toán vị từ
- Một số ngôn ngữ dữ liệu mức cao
 - QBE (Query By Example)
 - **SQL** (Structured Query Language)
- Kết luận



Đặt vấn đề

- Mục đích của ngôn ngữ dữ liệu
- Tại sao có nhiều ngôn ngữ dữ liệu?
- Ngôn ngữ cấp thấp vs. Ngôn ngữ cấp cao?

Ví dụ

- Tìm tên của các sinh viên nào sống ở Bundoora
 - Tìm các bộ của bảng Student có Suburb = Bundoora
 - Đưa ra các giá trị của thuộc tính Name của các bộ này

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Ví dụ (2)

- Tìm các sinh viên đăng ký khoá học có mã số 113
 - Tìm các giá trị SID trong bảng Enrol có trường course tương ứng là 113
 - Đưa các bộ của bảng Student có Id trong các giá trị tìm thấy ở trên

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Enrol

SID	Course
3936	101
1108	113
8507	101

Course

No	Name	Dept
113	BCS	CSCE
101	MCS	CSCE

Phân loại ngôn ngữ truy vấn

- Ngôn ngữ cấp thấp
 - **Đại số quan hệ**
 - 1 câu hỏi = 1 tập các phép toán trên các quan hệ
 - Được biểu diễn bởi một biểu thức đại số (quan hệ)
 - **Tính toán vị từ**
 - 1 câu hỏi = 1 mô tả của các bộ mong muốn
 - Được đặc tả bởi một vị từ mà các bộ phải thoả mãn
 - Phân biệt 2 lớp:
 - ngôn ngữ tính toán vị từ biến bộ
 - ngôn ngữ tính toán vị từ biến miền
- Ngôn ngữ cấp cao
 - **QBE**
 - **SQL**



Ngôn ngữ SQL

SQL (*Structured Query Language*)

- 1975: SEQUEL
 - System-R
- 1976: SEQUEL2
- 1978/79: SQL
 - System-R
- 1986: chuẩn SQL-86
- 1989: chuẩn SQL-89
- 1992: chuẩn SQL-92 (SQL2)
- 1996: chuẩn SQL-96
- 1999: chuẩn SQL-99 (SQL3)
- Newers: 2003, 2006, 2008, 2011, 2016.

Các thành phần của SQL

- Ngôn ngữ mô tả dữ liệu (Data Definition Language)
 - Cấu trúc các bảng CSDL
 - Các mối liên hệ của dữ liệu
 - Quy tắc, ràng buộc áp đặt lên dữ liệu
- Ngôn ngữ thao tác dữ liệu (Data Manipulation Language)
 - Thêm, xoá, sửa và truy vấn dữ liệu trong CSDL
- Ngôn ngữ quản lý dữ liệu (Data Control Language)
 - Thay đổi cấu trúc của các bảng dữ liệu
 - Khai báo bảo mật thông tin
 - Quyền hạn của người dùng trong khai thác CSDL

Định nghĩa dữ liệu với SQL

- Các thông tin được định nghĩa bao gồm
 - Sơ đồ quan hệ
 - Kiểu dữ liệu hay miền giá trị của mỗi thuộc tính
 - Các ràng buộc toàn vẹn
 - Các chỉ số đối với mỗi bảng
 - Thông tin an toàn và uỷ quyền đối với mỗi bảng
 - Cấu trúc lưu trữ vật lý của mỗi bảng trên đĩa
- Được biểu diễn bởi các lệnh định nghĩa dữ liệu

Cú pháp

- Tạo bảng

```
CREATE TABLE tab(  
    col1 type1(size1)[NOT NULL], ...,  
    col2 type2(size2)[NOT NULL], ...,  
    ....  
    [CONSTRAINT <constraint name> <constraint type> clause]  
    ...  
)
```

- Xóa bảng

```
DROP TABLE tab
```

Quy ước đặt tên và kiểu dữ liệu

- Quy ước đặt tên
 - 32 ký tự: chữ cái, số, dấu _
- Kiểu dữ liệu (SQL-92)
 - CHAR(n)
 - VARCHAR(n)
 - Int
 - Smallint
 - Numeric(p,d)
 - Real, double
 - float(n)
 - Date
 - time

Kiểu ràng buộc

- Ràng buộc toàn vẹn (RBTV) về giá trị miền
CONSTRAINT <name>
CHECK <condition>
- RBTV về khoá chính
CONSTRAINT <name> **PRIMARY KEY** (fk1,fk2,...)
- RBTV về khoá ngoại hay phụ thuộc tồn tại
CONSTRAINT <name> **FOREIGN KEY** (fk1,fk2,...)
REFERENCES tab(k1,k2, ...)

Thêm/xoá/sửa cột của các bảng

- Thêm

ALTER TABLE <tên bảng>

ADD COLUMN <tên cột> <kiểu dữ liệu> [NOT NULL]

- Xoá

ALTER TABLE <tên bảng>

DROP COLUMN <tên cột>

- Sửa

ALTER TABLE <tên bảng>

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

Thêm/xóa các ràng buộc

- Thêm

ALTER TABLE <tên bảng>

ADD CONSTRAINT <tên ràng buộc> <kiểu ràng buộc>

- Xóa

ALTER TABLE <tên bảng>

DROP CONSTRAINT <tên ràng buộc>

Ví dụ

```
CREATE TABLE Student(  
  Id char(10)NOT NULL,  
  Name varchar(30)NOT NULL,  
  Suburb varchar(30),  
  CONSTRAINT key_Stud  
    PRIMARY KEY Id  
)
```

```
CREATE TABLE Takes(  
  SID char(10)NOT NULL,  
  SNO varchar(5)NOT NULL,  
  CONSTRAINT key_takes  
    PRIMARY KEY (SID,SNO),  
  CONSTRAINT key_2Stud  
    FOREIGN KEY (SID)  
    REFERENCES Student(Id)  
)
```

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

SID	SNO
1108	21
1108	23
8507	23
8507	29

Thao tác dữ liệu với SQL

- Các thao tác chính với dữ liệu
 - Thêm : **INSERT ...**
 - Xóa: **DELETE ...**
 - Sửa (Cập nhật): **UPDATE ...**
 - Tìm kiếm (truy vấn) dữ liệu: **SELECT ...**
- Được biểu diễn bởi các lệnh thao tác dữ liệu

Các câu lệnh cập nhật dữ liệu

○ Thêm

- **INSERT INTO** table[(col1,col2,...)]
 VALUES (exp1,exp2,...)
- **INSERT INTO** table[(col1,col2,...)]
 SELECT col1,col2, ...
 FROM tab1, tab2, ...
 WHERE <dieu_kien>

○ Ví dụ

- **INSERT INTO** Student(Id, Name, Suburb)
 VALUES ("1179","David","Evr")

Các câu lệnh cập nhật dữ liệu (2)

- Xoá

```
DELETE FROM      table
WHERE            cond_exp;
```

- Sửa

```
UPDATE   table
SET      col1 = exp1,
           col2=exp2,
           ...
           colN=expN
WHERE    cond_exp;
```

- Ví dụ

- **DELETE FROM** Student
WHERE Suburb = "Bundoora";
- **UPDATE** Student
SET Suburb = "Evry"
WHERE Suburb = "Evr";

Cú pháp câu lệnh truy vấn SQL

SELECT [**DISTINCT**] <bt1>, <bt2>, ...
FROM <bang1>,<bang2>, ...
[**WHERE** <dieu kien chon>]
[**GROUP BY** <tt1>, <tt2>, ...
[**HAVING** <dieu kien in ket qua>]]
[**ORDER BY** <tt1> | <bieu thuc so 1> [**ASC** | **DESC**]]

Truy vấn đơn giản trên 1 bảng

○Tìm thông tin từ các cột của bảng

- **SELECT** ColumnName, ColumnName, ...
FROM TableName
- **SELECT** *
FROM TableName

○Ví dụ

SELECT Name
FROM Student

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

$\Pi_{name}(Student)$



Name
Robert
Glen
Norman
Mary

Truy vấn với điều kiện lựa chọn

- Chọn các bản ghi (dòng)

SELECT ColumnName, ColumnName, ...
FROM TableName
WHERE condition_expression;

- Ví dụ

SELECT *
FROM Student
WHERE Suburb = 'Bundoora' ;

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

$\sigma_{\text{suburb} = \text{"Bundoora"}}(\text{Student})$



Id	Name	Suburb
3936	Glen	Bundoora
8507	Norman	Bundoora

Biểu diễn điều kiện lựa chọn

- Các phép toán quan hệ: =, !=, <, >, <=, >=

- attr = 100

- Các phép toán logic: NOT, AND, OR

- attr < 10 **OR** attr > 100

- Phép toán phạm vi: BETWEEN, IN, LIKE

- Kiểu dữ liệu số

- attr **BETWEEN** val1 **AND** val2 (\Leftrightarrow (attr >= val1) and (attr <= val2))

- attr **IN** (val1, val2, ...) (\Leftrightarrow (attr = val1) or (attr = val2) or ...)

- Kiểu dữ liệu chuỗi

- **LIKE**: sử dụng đối sánh mẫu chuỗi với các ký tự _ (hoặc %) (thay thế cho 1 ký tự bất kỳ), % (hoặc *) (thay thế cho 1 chuỗi ký tự bất kỳ):

- attr **LIKE** "_CS"

- _ và % : mysql, oracle, postgresql, mariadb

- attr **LIKE** "%CS%"

- % và *: MS Access

Truy vấn phức tạp trên nhiều bảng

- Điều kiện kết nối

```
SELECT    T1.C1,T1.C2,T2.C1,T2.C4, ...  
FROM      T1, T2  
WHERE      condition_expression
```

- Ví dụ: đưa ra danh sách mã sinh viên (Id), tên sinh viên (Name), thành phố (Suburb), mã khoá học (Course) mà các sinh viên đã đăng ký

```
SELECT    Id, Name, Suburb,Course  
FROM      Student,Enrol  
WHERE      Id=SID
```

Bài tập

- Đưa ra *danh sách các khoá học* (Course) mà tên của khoá học chứa cụm “CS”
- Viết câu lệnh SQL đưa ra *danh sách các sinh viên* đăng ký các khoá học có mã 113 hoặc 101
- Viết câu lệnh SQL đưa ra *danh sách tên sinh viên* học môn “Database” hoặc môn “VB”

Ví dụ

Takes

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

SID	SNO
1108	21
1108	23
1108	29
8507	23
8507	29

Enrol

SID	Course
3936	101
1108	113
8507	101

Course

No	Name	Dept
113	BCS	CSCE
101	MCS	CSCE

Subject

No	Name	Dept
21	Systems	CSCE
23	Database	CSCE
29	VB	CSCE
18	Algebra	Maths

Phép toán đổi tên

- Từ khoá **AS** : *<bt> [as] <ten_thay_the>*

```
SELECT      c1 [as] name1, c2 [as] name2
FROM        TableName
```

- Ví dụ: đưa ra danh sách mã sinh viên, tên sinh viên và tên môn học mà sinh viên đó tham gia

```
SELECT      SID , Student.Name as SName,
              Subject.Name as Subject
FROM        Student,Takes, Subject
WHERE       (Id=SID) and (SNO = No)
```

Lưu ý: ten_thay_the cho cột được dùng trong mệnh đề **ORDER BY** chứ KHÔNG dùng được trong **WHERE** hay **HAVING**

Sử dụng biến bộ trong SQL

- Sử dụng từ khoá **AS** trong mệnh đề FROM
<ten_bang> [as] <ten_thay_the>

- Ví dụ

```
SELECT SID, St.Name as SName, Sub.Name as Subj  
FROM Student as St, Takes, Subject as Sub  
WHERE (Id=SID) and (SNO = No)
```

- Từ khoá **AS** : optional

```
SELECT SID, St.Name SName, Sub.Name Subj  
FROM Student St, Takes, Subject Sub  
WHERE (Id=SID) and (SNO = No)
```

Loại trừ các bản ghi trùng nhau

- Từ khoá **DISTINCT**

```
SELECT DISTINCT    <bt1>, <bt2>, ...  
FROM              <bang1>,<bang2>, ...
```

- Ví dụ: đưa ra danh sách tên các khoa (dept) tương ứng với các khoá học (Course). Mỗi giá trị chỉ hiện thị một lần

```
SELECT DISTINCT    Dept  
FROM              Course
```

Tìm kiếm có sắp xếp

- Sắp xếp các bản ghi kết quả theo một thứ tự cho trước

```
SELECT      <bt1>, <bt2>, ...  
FROM        <bang1>,<bang2>, ...  
[WHERE      <dieu kien chon>]  
ORDER BY <tt1> | <bieu thuc so 1> [ASC | DESC]
```

- Ví dụ: đưa ra danh sách tên các sinh viên theo thứ tự tăng dần

```
SELECT      Name  
FROM        Student  
ORDER BY Name ASC
```

Phân nhóm các bản ghi kết quả

- Phân nhóm các bản ghi kết quả theo giá trị của 1 hoặc nhiều thuộc tính

```
SELECT          <bt1>, <bt2>, ...  
FROM           <bang1>,<bang2>, ...  
[WHERE          <dieu kien chon>]  
GROUP BY <tt1>, <tt2>, ...
```

- Ví dụ đưa ra tên thành phố và số lượng sinh viên tương ứng đến từ các thành phố đó

```
SELECT      Suburb, Count(Id)  
FROM       Student  
GROUP BY   Suburb
```


Điều kiện hiển thị các bản ghi kết quả

- Lựa chọn các bản ghi kết quả để hiển thị

```
SELECT      <bt1>, <bt2>, ...  
FROM        <bang1>,<bang2>, ...  
[WHERE      <dieu kien chon>]  
GROUP BY    <bt1>, <bt2>  
HAVING      <dieu kien in ket qua>
```

- Ví dụ: đưa ra tên các thành phố có nhiều hơn 3 sinh viên

```
SELECT      Suburb, COUNT(ID)  
FROM        Student  
GROUP BY    Suburb  
HAVING      COUNT(ID) > 3
```

Các phép toán tập hợp

- **UNION, MINUS, INTERSECT**
- Ví dụ: đưa ra danh sách tên các môn học không có sinh viên nào tham dự

```
SELECT DISTINCT Subject.Name  
FROM Subject
```

MINUS

```
SELECT DISTINCT Subject.Name  
FROM Takes, Subject  
WHERE Takes.SNO = Subject.No
```

Các câu truy vấn lồng nhau

- Là trường hợp các câu truy vấn (con) được viết lồng nhau
- Thường được sử dụng với để
 - Kiểm tra thành viên tập hợp (**IN**)
 - So sánh tập hợp (**>ALL, >=ALL, <ALL, <=ALL, =ALL, NOT IN, SOME,)**
 - Kiểm tra các bảng rỗng (**EXISTS** hoặc **NOT EXISTS**)
- Các truy vấn con lồng nhau thông qua mệnh đề **WHERE**

Các câu truy vấn lồng nhau (...)

- CSDL: S(sid, sname, size, city)
 P(pid, pname, color, weight, city)
 SP(sid, pid, quantity)
- Đưa ra mã số của các hãng cung ứng đồng thời 2 mặt hàng P1 và P2:
SELECT DISTINCT sid
FROM SP
WHERE pid = "P1"
 AND sid IN (SELECT sid FROM SP WHERE pid = "P2");
- Đưa ra tên của các hãng có số nhân viên đông nhất:
SELECT sname FROM S
WHERE size ≥ ALL(SELECT size FROM S)



Các hàm thư viện

- Hàm tính toán trên nhóm các bản ghi
 - MAX/MIN
 - SUM
 - AVG
 - COUNT

- Đưa ra tên của hãng S1 và tổng số mặt hàng mà hãng đó cung ứng
`SELECT sname, SUM(quantity)`
`FROM S, SP`

`WHERE S.sid = SP.sid AND S.sid = 'S1'`
- Đưa ra mã số các hãng cung ứng và số lượng trung bình các mặt hàng được cung ứng bởi từng hãng
`SELECT sid, AVG(quantity)`
`FROM SP`
`GROUP BY sid;`
- Đưa ra mã số các hãng cung ứng mà số lượng mặt hàng trung bình được cung cấp bởi hãng đó là trong khoảng từ 75 đến 100
`SELECT sid, AVG(quantity)`
`FROM SP`
`GROUP BY sid`
`HAVING AVG(quantity) BETWEEN 75 AND 100`

Các hàm thư viện (2)

- Hàm tính toán trên bản ghi
 - Hàm toán học: ABS, SQRT, LOG, EXP, SIGN, ROUND
 - Hàm xử lý chuỗi ký tự: LEN, LEFT, RIGHT, MID
 - Hàm xử lý thời gian: DATE, DAY, MONTH, YEAR, HOUR, MINUTE, SECOND
 - Hàm chuyển đổi kiểu giá trị: FORMAT

Cần tra cứu cú pháp cụ thể của các hàm theo tài liệu của từng hệ quản trị CSDL.

-
- Đưa ra thông tin môn học có tên "Database" (không phân biệt chữ hoa/thường)

```
SELECT *  
FROM subject  
WHERE lower(name) = "database";
```

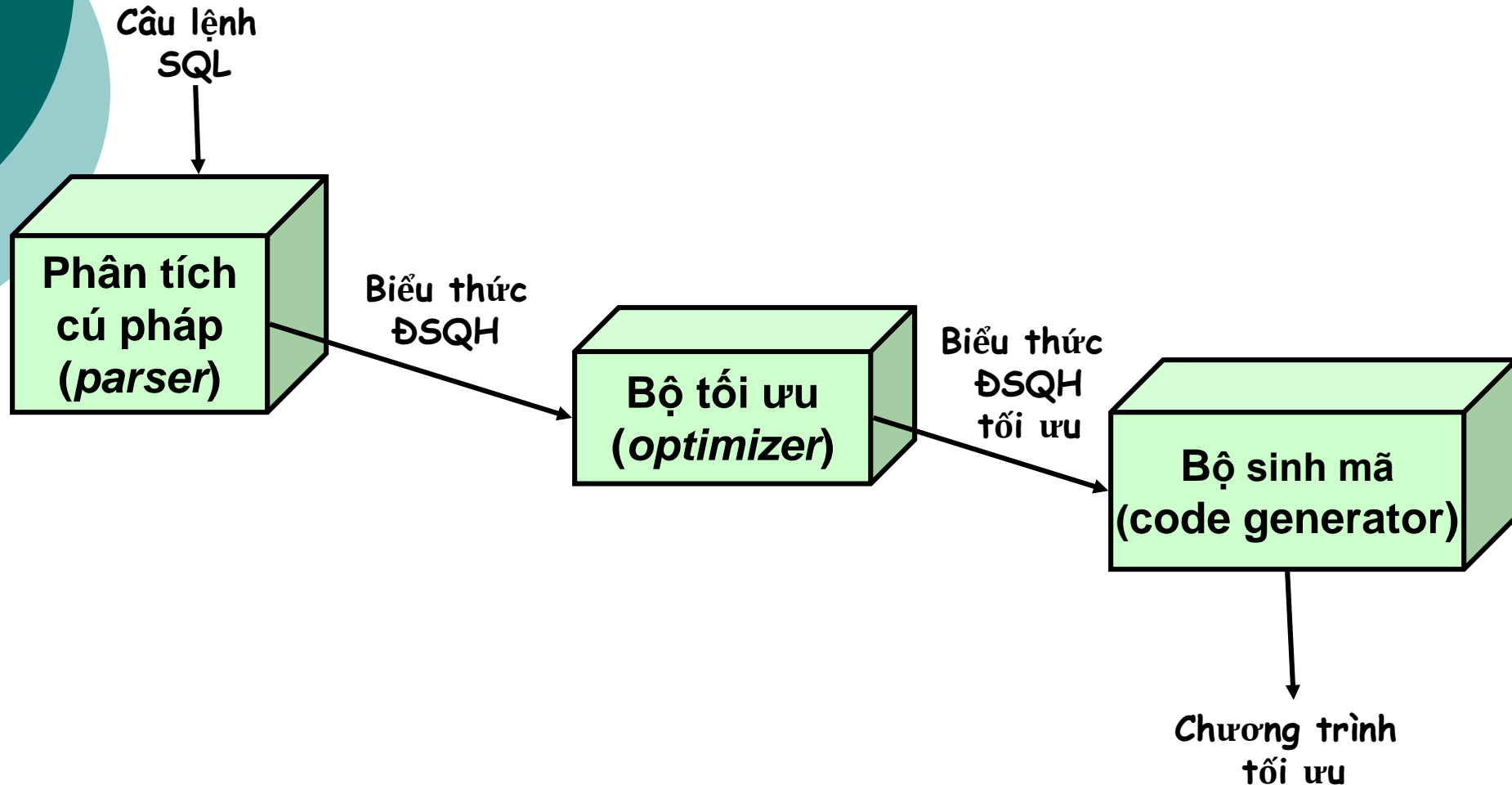
- Lấy ngày hiện tại

```
SELECT current_date;
```

- Lấy ngày, tháng, năm hiện tại

```
SELECT extract(day from current_date) as "Ngày",  
       extract(month from current_date) as "Tháng",  
       extract(year from current_date) as "Năm";
```


Xử lý câu hỏi truy vấn



Các điểm cần lưu ý

- Các ngôn ngữ dữ liệu
 - ĐSQH
 - SQL
- Sự tương đương của các ngôn ngữ
 - Biến đổi giữa câu truy vấn SQL và biểu thức đại số quan hệ

