

Viện Đại học Mở Hà Nội
Khoa Công nghệ Điện tử - Thông tin

Lập trình ứng dụng

Ngôn ngữ SQL
Phần 1: Câu truy vấn đơn

Nội dung chính

- ❖ Tổng quan về SQL
- ❖ Các lệnh SQL cơ bản

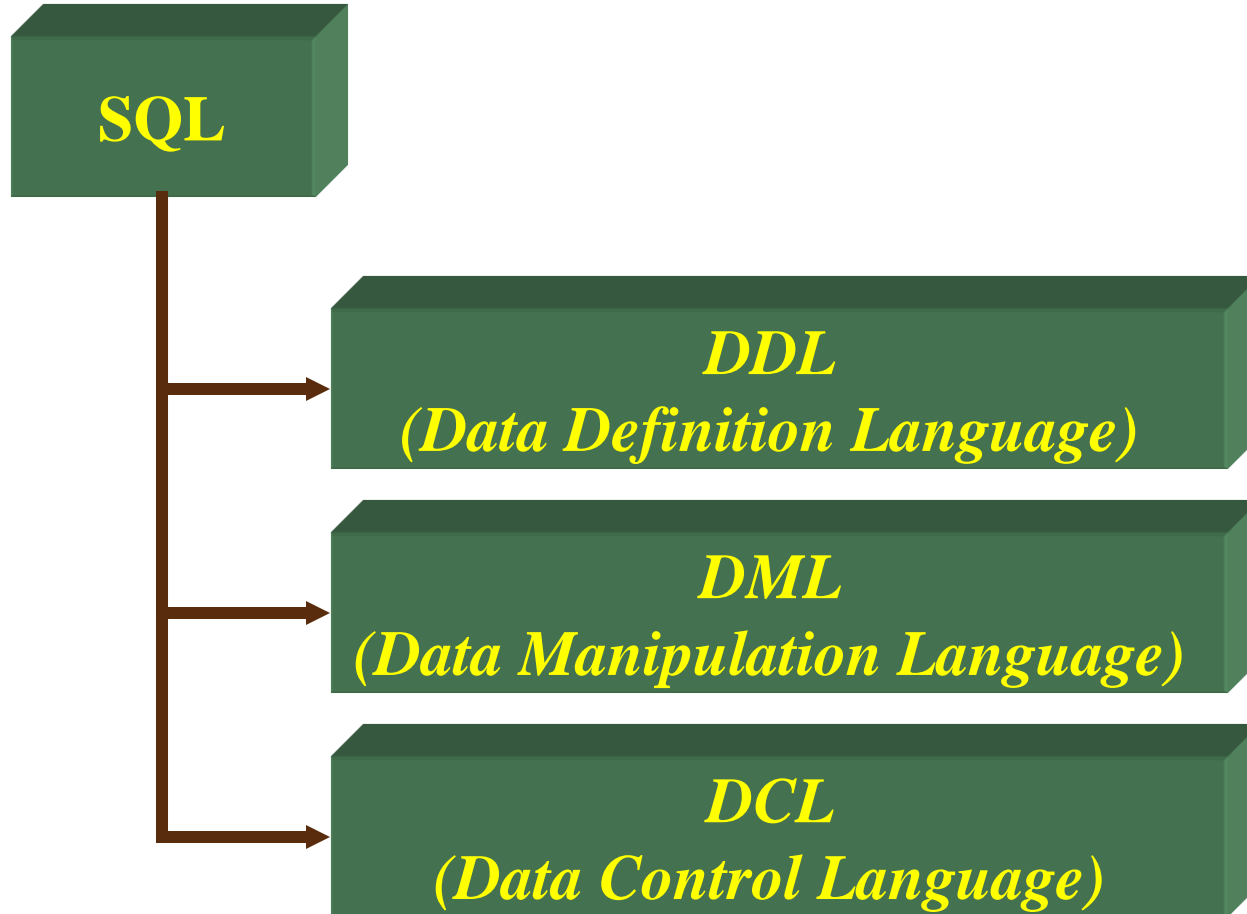
Tổng quan về SQL

- ❖ **SQL** (viết tắt của “Structured Query Language” – Ngôn ngữ truy vấn có cấu trúc) là tập các lệnh cho phép người dùng và cả các chương trình thực hiện các truy vấn dữ liệu trong cơ sở dữ liệu.
- ❖ Về mặt lịch sử, ban đầu nó có tên gọi là *SEQUEL*, (Structured English Query Language) do *Donald D. Chamberlin* và *Raymond F. Boyce* tại hãng IBM phát triển vào đầu những năm 70 của thế kỷ trước. Sau này nó mới được đổi tên thành SQL (và vẫn được phát âm là "sequel").
- ❖ Ngày nay, nó là ngôn ngữ chuẩn hóa của các hệ quản trị cơ sở dữ liệu quan hệ.

Các phiên bản SQL*

Year	Name	Alias	Comments
1986	SQL-86	SQL-87	First published by ANSI. Ratified by ISO in 1987.
1989	SQL-89	FIPS 127-1	Minor revision, adopted as FIPS 127-1.
1992	SQL-92	SQL2, FIPS 127-2	Major revision (ISO 9075), <i>Entry Level SQL-92</i> adopted as FIPS 127-2.
1999	SQL:1999	SQL3	Added regular expression matching, recursive queries, triggers, support for procedural and control-of-flow statements, non-scalar types, and some object-oriented features.
2003	SQL:2003		Introduced XML-related features, <i>window functions</i> , standardized sequences, and columns with auto-generated values (including identity-columns).
2006	SQL:2006		ISO/IEC 9075-14:2006 defines ways in which SQL can be used in conjunction with XML. It defines ways of importing and storing XML data in an SQL database, manipulating it within the database and publishing both XML and conventional SQL-data in XML form. In addition, it provides facilities that permit applications to integrate into their SQL code the use of XQuery, the XML Query Language published by the World Wide Web Consortium (W3C), to concurrently access ordinary SQL-data and XML documents.
2008	SQL:2008		Defines more flexible windowing functions, clarifies SQL 2003 items that were still unclear [1]

Các thành phần ngôn ngữ của SQL



Các thành phần ngôn ngữ của SQL

SELECT	Data retrieval
INSERT UPDATE DELETE	Data manipulation language (DML)
CREATE ALTER DROP TRUNCATE	Data definition language (DDL)
COMMIT ROLLBACK	Transaction control
GRANT REVOKE	Data control language (DCL)

Các lệnh định nghĩa dữ liệu (DDL)

Tên lệnh	Ý nghĩa
CREATE DATABASE <db-name>	Tạo một cơ sở dữ liệu với tên <i>db-name</i>
CREATE TABLE <tb-name> (...)	Tạo một bảng (quan hệ) với tên <i>tb-name</i> trong một CSDL cho trước
ALTER TABLE <tb-name>	Thay đổi cấu trúc bảng <i>tb-name</i> (thêm/bớt cột thuộc tính...)
DROP DATABASE <db-name>	Xóa cơ sở dữ liệu có tên <i>db-name</i>
DROP TABLE <tb-name>	Xóa bảng có tên <i>tb-name</i>

Ví dụ

```
CREATE DATABASE db_QLSV;
```

```
CREATE TABLE Lophoc
```

```
(  
  id_LH          int PRIMARY KEY;  
  tenlop        varchar(40) NOT NULL;  
  khoahoc       varchar(10);  
);
```

```
CREATE TABLE Sinhvien
```

```
(  
  id_SV          int PRIMARY KEY;  
  tenSV          varchar(40) NOT NULL;  
  ngaysinh       date;  
  diachi         varchar(150);  
  id_LD          int REFERENCES Lophoc(id_LH)  
);
```

Các lệnh thao tác dữ liệu (DML)

Tên lệnh	Ý nghĩa
INSERT [INTO] <tb-name> VALUES (v_1, v_2, \dots, v_k);	Bổ sung một bộ (v_1, v_2, \dots, v_k) vào bảng <i>tb-name</i> .
UPDATE <tb-name> SET $a_1 = v_1$ [, $a_2 = v_2, \dots$] [WHERE (đk)];	Cập nhật nội dung của một hoặc nhiều bộ trong một bảng (quan hệ) với tên <i>tb-name</i> trong một CSDL cho trước.
DELETE <tb-name> [WHERE (đk)];	Xóa một hoặc nhiều bộ trong bảng <i>tb-name</i> .
SELECT ...	Tìm kiếm các bộ trong một hoặc nhiều bảng theo một điều kiện nào đó.

Lệnh SELECT

Lệnh SELECT là một lệnh đa năng để truy vấn dữ liệu trong CSDL. Nó cho phép thực hiện tất cả các thao tác cơ bản trong đại số quan hệ như:

- ❖ **Chiều (Projection)**
- ❖ **Chọn (Selection)**
- ❖ **Nối (Joining)**
- ❖ **Hợp (Union)**
- ❖ **Trừ (Except)**

Projection

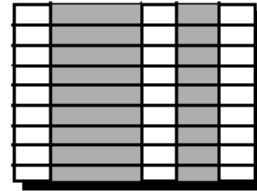


Table 1

Selection

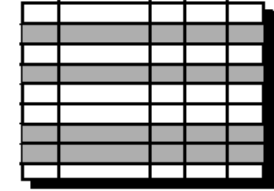
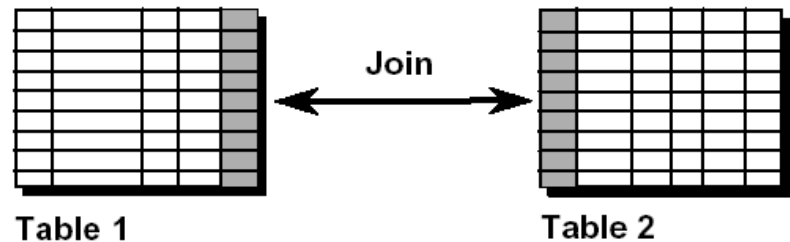


Table 1



Lệnh SQL đơn giản nhất: in ra toàn bộ nội dung của 1 bảng:

```
SELECT *  
FROM table-name;
```

Lệnh SELECT

❖ Phân loại:

- **Lệnh đơn:** là câu lệnh mà chỉ truy vấn thông tin từ 1 bảng
- **Lệnh phức:** là câu lệnh truy vấn thông tin từ nhiều bảng
- **Lệnh truy vấn con** (sub-query, hay còn gọi là lệnh SELECT lồng nhau): là câu lệnh SELECT mà bên trong nó cũng lại chứa 1 hay nhiều câu lệnh SELECT khác

Ghi chú

- ❖ Mỗi một lệnh (*statement*) trong SQL bao gồm một số mệnh đề (clause)

Ví dụ:

SELECT *

FROM employees

là một câu lệnh gồm có 2 mệnh đề

- ❖ Cú pháp trong SQL KHÔNG phân biệt chữ hoa với chữ thường
- ❖ Với các từ khóa nên viết hoa để dễ phân biệt với các từ khác

Lệnh đơn

1. Phép chiếu trong SQL
2. Phép chọn trong SQL
3. Đối sánh mẫu (Pattern matching) trong SQL
4. Giá trị NULL và 'Unknown'
5. Sắp xếp đầu ra

Cú pháp lệnh SELECT

SELECT [**ALL** | **DISTINCT**]

* | {column_name | expression [alias],...}

FROM table

- ❖ **SELECT** xác định các thuộc tính (cột) cần xuất ra
 - **ALL**: là lựa chọn mặc định, cho phép các hàng có giá trị trùng nhau cũng được xuất ra
 - **DISTINCT**: các hàng có giá trị trùng nhau chỉ được xuất ra 1 lần
- ❖ **FROM** xác định một hay nhiều bảng chứa các thông tin cần tìm

Ví dụ: SELECT tất cả các cột

```
SELECT *  
FROM PC
```

model	speed	hdd	screen	price
1003	2	250	14	1000
1004	2	500	18	900
1005	2.5	500	18	800
1006	2.5	500	18	800

Phép chiếu trong SQL

❖ Trong mệnh đề **SELECT**, thay vì sử dụng “*” để liệt kê toàn bộ các thuộc tính, ta có thể liệt kê từng thuộc tính mà muốn xuất ra.

❖ VD:

```
SELECT model, speed, price  
FROM PC
```

model	speed	price
1003	2	1000
1004	2	900
1005	2.5	800
1006	2.5	800

Có thể mở rộng phép chiếu sử dụng bí danh và biểu thức

```
SELECT model, price [price in USD],  
         price*20000 [price in VND]  
FROM    PC
```

model	price in USD	price in VND
1003	1000	20000000
1004	900	18000000
1005	800	16000000
1006	800	16000000

Bí danh (Alias)

- ❖ Là biện pháp cho phép đổi tên các thuộc tính (cột), hay tên các bảng trong câu lệnh SELECT
- ❖ Nó có thể đi kèm với từ khóa AS (không bắt buộc)
- ❖ Trong trường hợp bí danh có khoảng trắng thì cần đặt nó trong cặp “**bí danh**” hoặc [**bí danh**]

Loại bỏ các bộ trùng lặp với từ khóa **DISTINCT**

Bảng PC

model	speed	hdd	screen	price
1003	2	250	14	1000
1004	2	500	18	900
1005	2.5	500	18	800
1006	2.5	500	18	800

```
SELECT DISTINCT speed  
FROM PC
```

```
SELECT DISTINCT speed,hdd  
FROM PC
```

speed
2
2.5

speed	hdd
2	250
2	500
2.5	500

Phép chọn trong SQL

```
SELECT [ ALL | DISTINCT ]  
      * | {column_name | expression [alias],...}  
FROM   table  
WHERE  condition
```

Trong đó:

[condition]: biểu thức logic biểu diễn điều kiện chọn.

Các ví dụ cho lệnh chọn

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE department_id = 90;
```

```
SELECT last_name, job_id, department_id
FROM employees
WHERE last_name = 'Goyal';
```

```
SELECT last_name, salary
FROM employees
WHERE salary <= 3000;
```

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500;
```

Các ví dụ cho lệnh chọn

```
SELECT employee_id, last_name, salary, manager_id
FROM employees
WHERE manager_id IN (100, 101, 201);
```

```
SELECT first_name
FROM employees
WHERE first_name LIKE 'S%';
```

```
SELECT last_name, manager_id
FROM employees
WHERE manager_id IS NULL;
```

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >=10000
AND job_id LIKE '%MAN%';
```

Các phép toán trong SQL

- ❖ Các phép toán số học: +, -, *, /,
- ❖ Các phép toán so sánh: <, >, <=, >=, =, <> hoặc !=, BETWEEN .. AND
- ❖ Phép đối sánh mẫu: LIKE
- ❖ Các phép toán logic: AND, OR, NOT
- ❖ Các phép toán tập hợp: IN, UNION, INTERSECTION, EXCEPT (MINUS)

Đối sánh mẫu

- ❖ Khi so sánh các chuỗi, ngoài các phép toán quan hệ thông thường (<, >, =, v.v), SQL còn cung cấp khả năng so sánh theo mẫu (pattern), nó được gọi là “đối sánh mẫu” (pattern matching).
- ❖ Cú pháp như sau: `s LIKE p` trong đó:
 - s: là chuỗi ta muốn đem so sánh
 - p: là một mẫu cần so sánh. Nó có thể chứa các ký bất kỳ và 2 loại ký tự mẫu đại diện đặc biệt:
 - “%”: đại diện cho một chuỗi bất kỳ, kể cả chuỗi rỗng
 - “_”: đại diện cho đúng 1 ký tự bất kỳ

Một số ví dụ

EmployeeID	Last Name	First Name	Title	TitleOfCourtesy
1	Davolio	Nancy	Sales Representative	Ms.
2	Fuller	Andrew	Vice President, Sales	Dr.
3	Leverling	Janet	Sales Representative	Ms.
4	Peacock	Margaret	Sales Representative	Mrs.
5	Buchanan	Steven	Sales Manager	Mr.
6	Suyama	Michael	Sales Representative	Mr.
7	King	Robert	Sales Representative	Mr.
8	Callahan	Laura	Inside Sales Coordinator	Ms.
9	Dodsworth	Anne	Sales Representative	Ms.

Bảng Employees trong CSDL NorthWind

Một số ví dụ

Tìm các nhân viên có FirstName bắt đầu là ký tự 'A'?

```
SELECT EmployeeID, LastName, FirstName, Title  
FROM Employees  
WHERE FirstName LIKE 'A%'
```

EmployeeID	LastName	FirstName	Title
2	Fuller	Andrew	Vice President, Sales
9	Dodsworth	Anne	Sales Representative

Một số ví dụ

Tìm các nhân viên có LastName có độ dài ít nhất 2 ký tự và có ký tự cuối cùng là 'n' ?

```
SELECT EmployeeID, LastName, FirstName, Title
FROM Employees
WHERE LastName LIKE '%_n'
```

EmployeeID	LastName	FirstName	Title
5	Buchanan	Steven	Sales Manager
8	Callahan	Laura	Inside Sales Coordinator

Giá trị NULL và logic UNKNOWN

- ❖ **NULL** là giá trị đặc biệt được đưa vào để biểu diễn giá trị cho các thuộc tính mà *không có giá trị*.
- ❖ **NULL** không thuộc miền giá trị của bất kỳ kiểu dữ liệu nào, nên thực ra nó không được coi như một “giá trị” thực sự cho một thuộc tính, mà chỉ có ý nghĩa đánh dấu là thuộc tính này chưa có giá trị (chưa được khởi tạo, cũng như chưa được cập nhật giá trị).
- ❖ Do đó, việc so sánh một giá trị với **NULL** có thể không trả về giá trị logic *TRUE* hay *FALSE* như các giá trị thông thường. Chính vì vậy, các hệ QTCSDL đưa thêm vào một giá trị logic thứ ba để biểu diễn tình huống này và gọi nó là *Unknown*

Bảng giá trị logic với 'Unknown'

p	q	$p \text{ OR } q$	$p \text{ AND } q$	$p = q$
True	True	True	True	True
True	False	True	False	False
True	Unknown	True	Unknown	Unknown
False	True	True	False	False
False	False	False	False	True
False	Unknown	Unknown	False	Unknown
Unknown	True	True	Unknown	Unknown
Unknown	False	Unknown	False	Unknown
Unknown	Unknown	Unknown	Unknown	Unknown

p	NOT p
True	False
False	True
Unknown	Unknown

Sắp xếp kết quả tìm kiếm

```
SELECT [ ALL | DISTINCT ]  
      * | {column_name | expression [alias],...}  
FROM table  
[WHERE conditions]  
[ORDER BY {expression [ASC | DESC] ,...} ]
```

- ❖ Sử dụng mệnh đề ORDER BY, nó phải là mệnh đề cuối cùng trong lệnh SELECT.
- ❖ **Expression:** Xác định một hoặc nhiều thuộc tính trong số các thuộc tính mà ta muốn sắp xếp. Khi có nhiều thuộc tính, thì việc sắp xếp sẽ lần lượt theo thứ tự xuất hiện của các thuộc tính.

Ví dụ

In d/s nhân viên có sắp xếp theo ***LastName***:

```
SELECT EmployeeID, LastName, FirstName, Title, TitleOfCourtesy  
FROM Employees  
ORDER BY LastName
```

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy
10	Binh	Nguyen	NULL	NULL
5	Buchanan	Steven	Sales Manager	Mr.
8	Callahan	Laura	Inside Sales Coordinator	Ms.
1	Davolio	Nancy	Sales Representative	Ms.
9	Dodsworth	Anne	Sales Representative	Ms.
2	Fuller	Andrew	Vice President, Sales	Dr.
7	King	Robert	Sales Representative	Mr.
3	Leverling	Janet	Sales Representative	Ms.
4	Peacock	Margaret	Sales Representative	Mrs.
6	Suyama	Michael	Sales Representative	Mr.

Ví dụ

In d/s nhân viên có sắp xếp theo ***TitleOfCourtesy*** (theo thứ tự giảm dần) và ***FirstName***:

```
SELECT EmployeeID, LastName, FirstName, Title, TitleOfCourtesy
FROM Employees
ORDER BY TitleOfCourtesy DESC, 3
```

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy
9	Dodsworth	Anne	Sales Representative	Ms.
3	Leverling	Janet	Sales Representative	Ms.
8	Callahan	Laura	Inside Sales Coordinator	Ms.
1	Davolio	Nancy	Sales Representative	Ms.
4	Peacock	Margaret	Sales Representative	Mrs.
6	Suyama	Michael	Sales Representative	Mr.
7	King	Robert	Sales Representative	Mr.
5	Buchanan	Steven	Sales Manager	Mr.
2	Fuller	Andrew	Vice President, Sales	Dr.
10	Binh	Nguyen	NULL	NULL

Tóm tắt

- ❖ Mục đích câu lệnh SQL
- ❖ Các nhóm câu lệnh SQL
- ❖ Lệnh SELECT

Viện Đại học Mở Hà Nội
Khoa Công nghệ Điện tử - Thông tin

Thank You !