mindX Technology School

Computer Science Advanced

BÀI 5. GIỚI THIỆU VỀ CƠ SỞ DỮ LIỆU

1. Cơ Sở Dữ Liệu

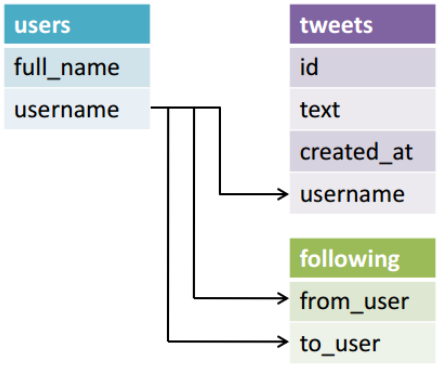
Ảnh: TechGig

Cơ sở dữ liệu *(Database)* là một *tập hợp dữ liệu được lưu trữ một cách có tổ chức.*

Một *cơ sở dữ liệu* được quản lý bằng một ***Hệ thống quản lý cơ sở dữ liệu*** *(Database Management System – DBMS)*. Hệ thống này là cầu nối từ cơ sở dữ liệu đến các ứng dụng và người dùng.

CÁC LOẠI CƠ SỞ DỮ LIỆU

Thông thường, ta chia cơ sở dữ liệu thành 2 nhóm dựa theo cách tổ chức dữ liệu.



Cơ sở dữ liệu quan hệ *(Relational Database):*

Dữ liệu được lưu trữ theo *cấu trúc bảng*, mỗi bảng đại diện cho một chủ thể. Các bảng có thể được thiết lập quan hệ với nhau.

Với cách tổ chức này, cơ sở dữ liệu quan hệ cho phép các chương trình *quản lý* và *truy vấn* thông qua ***Ngôn ngữ truy vấn có cấu trúc*** *(Structured Query Language – SQL)*.

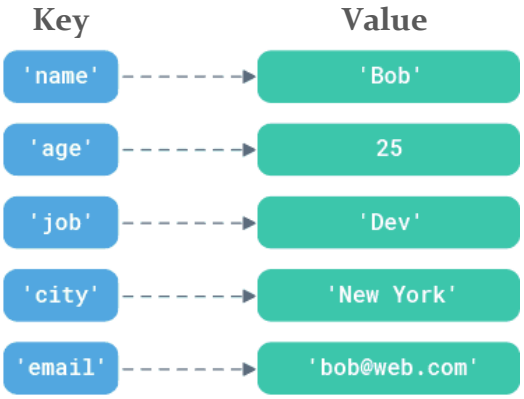
Cơ sở dữ liệu phi quan hệ *(Non-Relational Database):*

Đây là một khái niệm để gọi chung tất cả các cơ sở dữ liệu có *cách tổ chức khác với cơ sở dữ liệu quan hệ.*

Một số cách tổ chức phổ biến:

Document Database lưu trữ các dữ liệu có liên quan với nhau theo từng *"tài liệu"*. Mỗi tài liệu có thể lưu được dưới dạng JSON, XML, ...

Key-Value Database lưu trữ dữ liệu theo dạng *"khóa"* và *"giá trị"*. Mỗi giá trị có thể được truy vấn từ một khóa độc nhất.



Ngoài ra còn nhiều cách tổ chức dữ liệu khác như *Wide-Column* và *Graph Database*.

Vì các cơ sở dữ liệu này không dùng ngôn ngữ SQL để truy vấn nên còn được gọi là *noSQL Database*.

2. Cơ Sở Dữ Liệu Quan Hệ

I. Trong cơ sở dữ liệu quan hệ, dữ liệu được biểu diễn thành các bảng, mỗi bảng gồm các dòng và cột.

- Mỗi ***dòng*** biểu diễn một *đối tượng*.
- Mỗi ***cột*** biểu diễn một *thuộc tính* của đối tượng. Các thuộc tính này còn gọi là *trường*.

Ví dụ: Bảng **users** chứa dữ liệu về các tài khoản trên mạng xã hội Twitter:

- Mỗi dòng chứa thông tin về một tài khoản
- Mỗi cột chứa một thông tin của mỗi tài khoản, trong ví dụ này là *tên tài khoản* và *tên người dùng*.

users

username	full_name
_DreamLead	Boris Hadjur
GunnarSvalander	Gunnar Slavander
GEsoftware	GE Software

II. Một cơ sở dữ liệu có thể chứa nhiều bảng, mỗi bảng đại diện cho một chủ thể.

Ví dụ: Bảng **tweets** chứa dữ liệu về các bài đăng và bảng **following** lưu trạng thái theo dõi giữa các tài khoản.

tweets

id	text	created_at	username
1	What do you think about #emailing #campaigns…	Tue, 12 Feb 2013 08:43:09	_DreamLead
2	Bill Gates Talks Databases, Free Software on Reddit…	Tue, 12 Feb 2013 07:31:06	GunnarSvalander
3	RT @KirkDBorne: Readings in #Databases: excellent…	Tue, 12 Feb 2013 07:30:24	GEsoftware

following

from_user	to_user
_DreamLead	GunnarSvalander
GunnarSvalander	_DreamLead
_DreamLead	GEsoftware

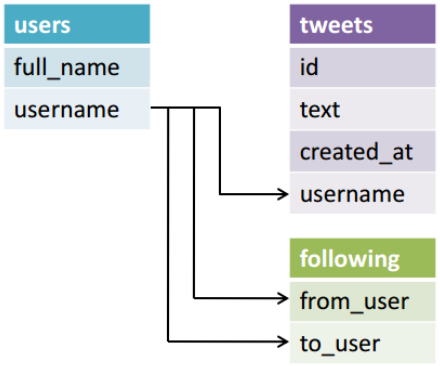
Chú ý: Mỗi bảng đều có một ***khóa chính*** *(primary key)* được tạo thành từ *một hoặc nhiều trường*. Trong cùng một bảng, khóa chính của mỗi dòng là *độc nhất*. Ta đặt khóa chính để đảm bảo *tính chặt chẽ của dữ liệu*.

- Bảng **users** có khóa chính là *username*, như vậy không tài khoản nào có cùng tên tài khoản.
- Bảng **tweets** có khóa chính là *id*.
- Bảng **following** có khóa chính được tạo thành bởi hai trường là *from_user* và *to_user*.

III. Các bảng trong cơ sở dữ liệu liên kết với nhau bằng quan hệ.

Trong ví dụ này, ta có:

- Mỗi *username* trong **tweets** đều phải tồn tại trong **users**.
  Ta gọi *username* trong **tweets** là ***khóa ngoại*** *(foreign key)*, phân biệt với *username* trong **users** là *khóa chính*.
- Mối quan hệ giữa *username* trong **users** và *username* trong **tweets** là ***quan hệ 1-nhiều***, với ý nghĩa:
  - *Một* **user** có thể viết *nhiều* **tweet**.
  - *Mỗi* **tweet** chỉ có thể được viết bởi *một* **user**.
- Tương tự, ta có *from_user* và *to_user* trong **following** là *khóa ngoại* của *username* trong **users**. Mối quan hệ giữa các trường này là *1-nhiều*.



Ảnh và ví dụ: envatotuts+

CÁC LOẠI QUAN HỆ CHÍNH

Ngoài quan hệ 1-nhiều, ta còn có *quan hệ 1-1* và *quan hệ nhiều-nhiều*.

Quan hệ 1-1: Giả sử ta tạo thêm bảng **user_details** lưu các thông tin cá nhân của người dùng. Khi đó quan hệ giữa *username* trong **users** và *username* trong **user_details** là *quan hệ 1-1*.

Quan hệ nhiều-nhiều: là quan hệ trong trạng thái theo dõi giữa các tài khoản:

- *Một* **user** có thể theo dõi *nhiều* **user**.
- *Mỗi* **user** được theo dõi cũng có thể theo dõi *nhiều* **user** khác.

Một quan hệ nhiều-nhiều khi đưa vào cơ sở dữ liệu sẽ được tách ra thành một bảng riêng, tạo thành *hai quan hệ 1-nhiều* như bảng **following**.

3. DBMS và Ngôn Ngữ SQL

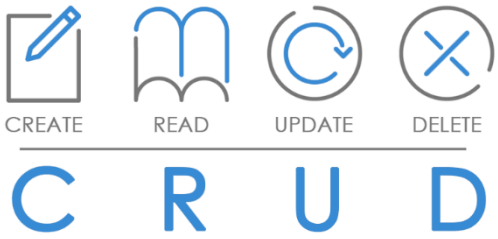
***Ngôn ngữ truy vấn có cấu trúc** (SQL)* được dùng để điều khiển các hoạt động quản lý dữ liệu trong một *cơ sở dữ liệu quan hệ* thông qua ***Hệ thống quản lý cơ sở dữ liệu** (DBMS)*.

MỘT SỐ DBMS PHỔ BIẾN

Microsoft SQL Server	MySQL	SQLite
Phần mềm thuộc Microsoft.	Phần mềm mã nguồn mở thuộc Oracle.	Phần mềm mã nguồn mở.
Chạy trên hệ điều hành Windows và Linux.	Chạy trên nhiều hệ điều hành, bao gồm Windows, Linux và OS X.	Không phụ thuộc hệ điều hành. Có thể chạy trên thiết bị di động.

CÁC HOẠT ĐỘNG QUẢN LÝ DỮ LIỆU VỚI SQL

- ***Create***: Khởi tạo cơ sở dữ liệu và các bảng.
- ***Read***: Đọc/truy vấn dữ liệu từ các bảng.
- ***Update***: Cập nhật các dòng và cột.
- ***Delete***: Xóa bảng, dòng và cột.



Hoạt động	Ví dụ

Create: Khởi tạo bảng
`CREATE TABLE tên_bảng(`
`  tên_cột kiểu_dữ_liệu,`
`  tên_cột kiểu_dữ_liệu,`
`  ...`
`);`
	`CREATE TABLE users(`
	`  username VARCHAR(63),`
	`  full_name VARCHAR(255)`
	`);`

Update: Thêm dòng vào bảng
`INSERT INTO tên_bảng(tên_cột, tên_cột, ...)`
`VALUES (giá_trị, giá_trị, ...);`
	`INSERT INTO users(username, full_name)`
	`VALUES ('_DreamLead', 'Boris Hadjur');`

Update: Thêm cột vào bảng
`ALTER TABLE tên_bảng`
`ADD tên_cột kiểu_dữ_liệu;`
	`ALTER TABLE users`
	`ADD email VARCHAR(255);`

Update: Thay đổi giá trị trong bảng
`UPDATE tên_bảng`
`SET tên_cột = giá_trị, tên_cột = giá_trị, ...`
`WHERE điều_kiện;`
	`UPDATE users`
	`SET full_name = 'John Doe'`
	`WHERE username = '_DreamLead';`

Hoạt động	Ví dụ

Delete: Xóa dòng trong bảng
`DELETE FROM tên_bảng`
`WHERE điều_kiện;`
	`DELETE FROM users`
	`WHERE username = '_DreamLead';`

Delete: Xóa cột trong bảng
`ALTER TABLE tên_bảng`
`DROP tên_cột;`
	`ALTER TABLE users`
	`DROP email;`

Delete: Xóa bảng `DROP TABLE tên_bảng;`	`DROP TABLE users;`

Chú ý:

- Các lệnh SQL *không thể hoàn lại sau khi được thực thi*. Như vậy, sau khi thực hiện các hoạt động *delete* hoặc *update*, ta không thể khôi phục lại dữ liệu cũ.
- Lệnh UPDATE... SET và DELETE FROM nếu không có điều kiện ở WHERE sẽ mặc định *thay đổi tất cả các dòng trong bảng*.

PHỤ LỤC: CÁC KIỂU DỮ LIỆU CƠ BẢN TRONG SQL

Tùy thuộc vào DBMS mà các kiểu dữ liệu có thể khác nhau. Tuy nhiên, mỗi DBMS đều hỗ trợ các kiểu dữ liệu cơ bản cho các giá trị *số nguyên, số thực, chuỗi,* và *thời gian*.

Kiểu dữ liệu	Nội dung
`INT`	Số nguyên
`DECIMAL`	Số thực
`VARCHAR(n)`	Chuỗi tối đa n ký tự
`BLOB`	Dữ liệu dạng nhị phân (như hình ảnh)
`DATE`	Ngày tháng (YYYY-MM-DD)
`TIMESTAMP`	Thời gian (YYYY-MM-DD HH:MM:SS)