1. Khái niệm

Cơ sở dữ liệu (Database) là một tập các dữ liệu có tổ chức, thường được lưu trong các hệ thống máy tính. Hay có thể hiểu đơn giản nó chính là thông tin được lưu trữ trong máy tính. Có hai loại cơ sở dữ liệu là cơ sở dữ liệu quan hệ và cơ sở dữ liệu phi quan hệ.

1. Phân loại cơ sở dữ liệu
2. SQL: cơ sở dữ liệu quan hệ

Đối với cơ sở dữ liệu quan hệ, dữ liệu được cấu trúc thành các bảng, mỗi bảng gồm có các hàng và các cột, mỗi dữ liệu được thêm vào phải đảm bảo cấu trúc đúng như đã quy định của bảng đó. Dữ liệu sẽ được ghi thông quan ngôn ngữ truy vấn SQL (Structured Query Language). Các cơ sở dữ liệu quan hệ có thể kể đến như MS SQL Server, MySQL, Oracle…

1. NoSQL: cơ sở dữ liệu phi quan hệ

Cơ sở dữ liệu phi quan hệ (thường được gọi là NoSQL database) là cơ sở dữ liệu không cần quy định trước cấu trúc, mỗi dữ liệu có thể có các trường thông tin khác nhau (lưu trữ cấu trúc khá giống với json). Cơ sở dữ liệu phi quan hệ thì không dùng ngôn ngữ truy vấn. Một số cơ sở dữ liệu phi quan hệ có thể kể đến như MongoDB, CouchDB…

Ví dụ: Trong cơ sở dữ liệu NoSQL, hồ sơ về một cuốn sách thường được lưu trữ dưới dạng văn bản JSON. Với từng quyển sách, mục, ISBN, Tên sách, Số phiên bản, Tên tác giả và ID tác giả được lưu trữ dưới dạng thuộc tính trong một văn bản duy nhất. Trong mô hình này, dữ liệu được tối ưu hóa cho việc phát triển trực quan và khả năng thay đổi quy mô theo chiều ngang

* Lợi ích của sử dụng noSQL:
* Cơ sở dữ liệu NoSQL là lựa chọn cực kỳ thích hợp cho nhiều ứng dụng hiện đại, ví dụ như di động, web và trò chơi đòi hỏi phải sử dụng cơ sở dữ liệu cực kỳ thiết thực, linh hoạt, có khả năng thay đổi quy mô và hiệu năng cao, dễ dàng mở rộng theo chiều rộng.
* Linh hoạt: Cơ sở dữ liệu NoSQL thường cung cấp các sơ đồ linh hoạt giúp công đoạn phát triển nhanh hơn và có khả năng lặp lại cao hơn. Mô hình dữ liệu linh hoạt biến cơ sở dữ liệu NoSQL thành lựa chọn lý tưởng cho dữ liệu không được tổ chức thành cấu trúc hoặc có cấu trúc chưa hoàn chỉnh.
* Khả năng thay đổi quy mô: Cơ sở dữ liệu NoSQL thường được thiết kế để tăng quy mô bằng cách sử dụng các cụm phần cứng được phân phối thay vì tăng quy mô bằng cách bổ sung máy chủ mạnh và tốn kém.
* Hiệu năng cao: Cơ sở dữ liệu NoSQL được tối ưu hóa theo các mô hình dữ liệu cụ thể và các mẫu truy cập giúp tăng hiệu năng cao hơn so với việc cố gắng đạt được mức độ chức năng tương tự bằng cơ sở dữ liệu quan hệ.
* Cực kỳ thiết thực: Cơ sở dữ liệu NoSQL cung cấp các API và kiểu dữ liệu cực kỳ thiết thực được xây dựng riêng cho từng mô hình dữ liệu tương ứng.

III. Mục đích ta sử dụng CSDL

* Là vì cho dễ quản lý và tìm kiếm, thống nhất việc lưu trữ thông tin. Có thể tìm kiếm một cách nhanh chóng khi cần thông tin nào đó và cũng có thể lôi các “dòng họ” liên quan với thông tin đó ra luôn.

IV. MySQL

* là một hệ quản trị cơ sở dữ liệu quan hệ
* open-source
* free
* Tốc độ nhanh, đáng tin cậy và dễ mở rộng theo chiều rộng, dễ dàng sử dụng
* cross-platform
* developed by Oracle

SQL được sử dụng để thêm, tìm kiếm, update và xóa các bản ghi dữ liệu trên cơ sở dữ liệu

SQL keywords không phân biệt chữ hoa và chữ thường

* dấu ; để ngăn cách các câu lệnh SQL và cho phép nhiều hơn một câu lệnh SQL được gửi đến server trong một lần gọi
* SELECT - extracts data from a database
* UPDATE - updates data in a database
* DELETE - deletes data from a database
* INSERT INTO - inserts new data into a database
* CREATE DATABASE - creates a new database
* ALTER DATABASE - modifies a database
* CREATE TABLE - creates a new table
* ALTER TABLE - modifies a table
* DROP TABLE - deletes a table
* CREATE INDEX - creates an index (search key)
* DROP INDEX - deletes an index

1. Select

SELECT column1, column2, ...

FROM table\_name;

nếu muốn lấy tất cả các trường

SELECT \* FROM *table\_name*;

SELECT DISTINCT được dùng để chỉ lấy những giá trị distinct: phân biệt

1. Create Table

CREATE TABLE

được dùng để tạo một bảng mới trong một database

Example: CREATE TABLE Persons (

PersonID int,

LastName varchar(255),

FirstName varchar(255),

Address varchar(255),

City varchar(255)

);

Một bảng mới cũng có thể được tạo ra từ một bảng cũ

CREATE TABLE TestTable AS

SELECT customername, contactname

FROM customers;

1. Where

được dùng để thêm điều kiện

| **Operator** | **Description** |
| --- | --- |
| = | Equal |
| > | Greater than |
| < | Less than |
| >= | Greater than or equal |
| <= | Less than or equal |
| <> | Not equal. Note: In some versions of SQL this operator may be written as != |
| BETWEEN | Between a certain range |
| LIKE | Search for a pattern |
| IN | To specify multiple possible values for a column |

1. Insert into

Dùng để thêm các bản ghi mới vào một bảng

* Có thể viết theo hai cách : có chú thích cột cụ thể và thêm cho toàn bảng

INSERT INTO *table\_name* (*column1*, *column2*, *column3*, ...)

VALUES (*value1*, *value2*, *value3*, ...);

OR

INSERT INTO *table\_name*

VALUES (*value1*, *value2*, *value3*, ...);

1. Giá trị NULL

Giá trị NULL là giá trị rỗng

* phải sử dụng các operator là IS NULL và IS NOT NULL

SELECT *column\_names*

FROM *table\_name*

WHERE *column\_name* IS NULL;

1. Update được dùng để thay đổi các bản ghi sẵn có trong bảng

* Phải có where nếu không sẽ update toàn bảng

UPDATE Customers

SET ContactName = 'Alfred Schmidt', City = 'Frankfurt'

WHERE CustomerID = 1;

1. Delete

The DELETE statement is used to delete existing records in a table.

DELETE FROM *table\_name* WHERE *condition*