BÁO CÁO KIẾN THỨC TÌM HIỂU VỀ MySQL

* SQL (Ngôn ngữ truy vấn có cấu trúc): là một ngôn ngữ lập trình phục vụ việc lưu trữ và xử lý thông tin trong cơ sở dữ liệu quan hệ. Cơ sở dữ liệu quan hệ lưu trữ thông tin dưới dạng bảng có các hàng và cột đại diện cho những thuộc tính dữ liệu và nhiều mối quan hệ khác nhau giữa các giá trị dữ liệu. Có thể sử dụng các câu lệnh SQL để lưu trữ, cập nhật, loại bỏ, tìm kiếm và truy xuất thông tin từ cơ sở dữ liệu, cũng có thể sử dụng SQL để duy trì và tối ưu hóa hiệu suất cơ sở dữ liệu.

Một số cú pháp SQL phổ biến:

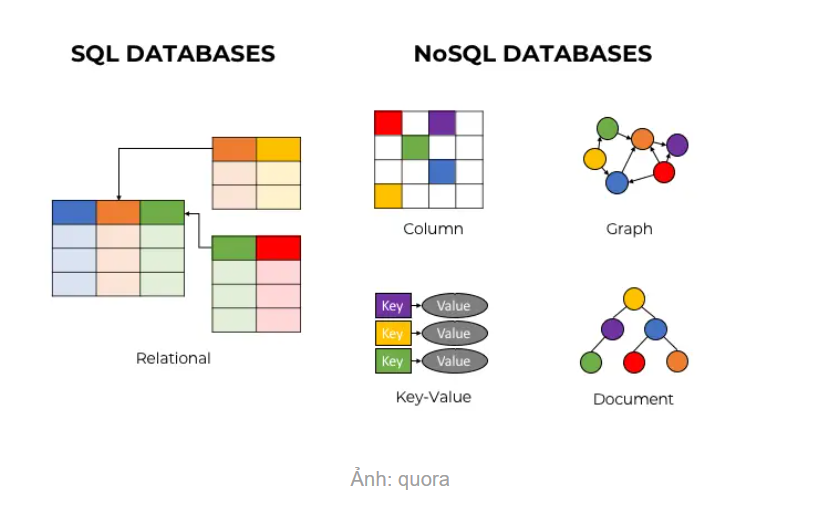
* SELECT – Trích xuất dữ liệu từ database
* UPDATE – Cập nhật dữ liệu vào database
* DELETE – Xóa dữ liệu từ database
* INSERT INTO – Chèn dữ liệu mới vào database
* CREATE DATABASE – Tạo database mới
* ALTER DATABASE – Sửa database ( sửa tên, trạng thái,…)
* CREATE TABLE – Tạo bảng mới
* ALTER TABLE – Sửa bảng
* DROP TABLE – Xóa bảng
* CREATE INDEX – Tạo chỉ mục trong bảng
* DROP INDEX – Xóa chỉ mục
* No SQL:

Một trong những lựa chọn cơ bản cần thực hiện khi phát triển một ứng dụng là sử dụng cơ sở dữ liệu (CSDL) SQL hay NoSQL để lưu trữ dữ liệu. SQL đã được phát triển từ rất lâu, được kiểm nghiệm qua thời gian và sử dụng rất nhiều. Chúng được thiết kế cho các giao dịch đáng tin cậy và các truy vấn đặc biệt trong các ứng dụng. Tuy nhiên, CSDL SQL không phải là không có hạn chế và và không phải lúc nào cũng phù hợp với tất cà các nhu cầu lưu trữ và truy xuất dữ liệu.

Và cơ sở dữ liệu NoSQL đã ra đời để đáp ứng những hạn chế đó. Hệ thống CSDL NoSQL lưu trữ và quản lý dữ liệu theo cách cho phép tốc độ hoạt động cao và tính linh hoạt cao.

* **Cơ sở dữ liệu NoSQL** là một Hệ thống quản lý dữ liệu không quan hệ (**non-relational Data Management System**) có lược đồ (schema) linh hoạt, dễ mở rộng.
* Mục đích chính của việc sử dụng cơ sở dữ liệu NoSQL là dành cho các kho dữ liệu phân tán với nhu cầu lưu trữ dữ liệu lớn. NoSQL được sử dụng cho Dữ liệu lớn và ứng dụng web thời gian thực.

Đối với hệ thống cơ sở dữ liệu NoSQL, nó bao gồm một loạt các công nghệ cơ sở dữ liệu có thể lưu trữ dữ liệu có cấu trúc, bán cấu trúc, phi cấu trúc và đa hình. Sơ đồ sau cho thấy CSDL NoSQL khác so với CSDL SQL:



* Join:

Trong SQL, mệnh đề JOIN được sử dụng để kết hợp các hàng từ hai hoặc nhiều bảng dựa trên một cột có liên quan giữa chúng.

* INNER JOIN: Trả về các record có giá trị khớp trong cả hai bảng
* LEFT JOIN: Trả về tất cả các record từ bảng bên trái (**table1**) và mọi record phù hợp từ bảng bên phải (**table2**). Nếu không khớp, giá trị **NULL** sẽ được hiển thị cho các cột của bảng bên phải (**table2**).
* RIGHT JOIN: Trả về tất cả các record từ bảng bên phải (**table2**) và mọi record phù hợp từ bảng bên trái (**table1**). Nếu không khớp, giá trị **NULL** sẽ được hiển thị cho các cột của bảng bên trái (**table1**).
* FULL JOIN: Trả về các record có giá trị khớp trong cả hai bảng, cũng như bất kỳ record nào chưa khớp từ một trong hai bảng **<==>** LEFT JOIN union RIGHT JOIN
* CROSS JOIN: Trả về tích Descartes của hai bảng (nghĩa là mọi tổ hợp các record từ cả hai bảng)
* SELF JOIN: Trả về các record có giá trị khớp với điều kiện khi một bảng được join với chính nó
* Function:

**Function (Hàm)** là một đối tượng trong cơ sở dữ liệu bao gồm một tập nhiều câu lệnh được nhóm lại với nhau và được tạo ra với mục đích sử dụng lại.

**Cú pháp:**

* **Tạo hàm**

/\* schema\_name: Tên schema (lược đồ) sở hữu function . function\_name: Tên gán cho function \*/

CREATE FUNCTION [schema\_name.]function\_name

/\* @parameter: Một hay nhiều tham số được truyền vào hàm.

type\_schema\_name: Kiểu dữ liệu của schema.

Datatype: Kiểu dữ liệu cho @parameter.

*Default:* Giá trị mặc định gán cho @parameter.

*READONLY:* @parameter không thể bị function ghi đè lên.

\*/

( [ @parameter [ AS ] [type\_schema\_name.] datatype

[ = default ] [ READONLY ]

, @parameter [ AS ] [type\_schema\_name.] datatype

[ = default ] [ READONLY ] ]

)

RETURNS return\_datatype /\*Kiểu dữ liệu của giá trị trả về.\*/

/\* *ENCRYPTION:* Mã nguồn (source) của function sẽ không được lưu trữ dưới dạng text trong hệ thống.

*SCHEMABINDING:* Đảm bảo các đối tượng không bị chỉnh sửa gây ảnh hưởng đến function.

*RETURNS NULL ON NULL INPUT*: Hàm sẽ trả về NULL nếu bất cứ parameter nào là NULL.

*CALL ON NULL INPUT*: Hàm sẽ thực thi cho dù bao gồm tham số là NULL.

*EXECUTE AS* clause: Xác định ngữ cảnh bảo mật để thực thi hàm.

\*/

[ WITH { ENCRYPTION

| SCHEMABINDING

| RETURNS NULL ON NULL INPUT

| CALLED ON NULL INPUT

| EXECUTE AS Clause ]

[ AS ]

BEGIN

[declaration\_section]

executable\_section

RETURN return\_value /\* Giá trị được trả về. \*/

END;

* **Xóa hàm**

DROP FUNCTION function\_name;

* Procedure: là một chương trình trong cơ sở dữ liệu gồm nhiều câu lệnh mà bạn lưu lại cho những lần sử dụng sau.

Cú pháp:

/\* *schema\_name:* Tên schema (lược đồ) sở hữu procedure.

*procedure\_name:* Tên gán cho procedure

\*/

CREATE { PROCEDURE | PROC } [schema\_name.]procedure\_name

/\* *@parameter:* Một hay nhiều tham số được truyền vào hàm.

*type\_schema\_name:* Kiểu dữ liệu của schema (nếu có).

*Datatype:* Kiểu dữ liệu cho @parameter.

*Default:* Giá trị mặc định gán cho @parameter.

*OUT/OUTPUT:* @parameter là một tham số đầu ra

*READONLY:* @parameter không thể bị procedure ghi đè lên.

\*/

[ @parameter [type\_schema\_name.] datatype

[ VARYING ] [ = default ] [ OUT | OUTPUT | READONLY ]

, @parameter [type\_schema\_name.] datatype

[ VARYING ] [ = default ] [ OUT | OUTPUT | READONLY ] ]

/\* *ENCRYPTION:* Mã nguồn (source) của procedure sẽ không được lưu trữ dưới dạng text trong hệ thống.

*RECOMPILE:* Truy vấn sẽ không được lưu ở bộ nhớ đệm (cache) cho thủ tục này.

*EXECUTE AS clause:*Xác định ngữ cảnh bảo mật để thực thi thủ tục.

*FOR REPLICATION:* Procedure đã lưu sẽ chỉ được thực thi trong quá trình replication (nhân bản).

\*/

[ WITH { ENCRYPTION | RECOMPILE | EXECUTE AS Clause } ]

[ FOR REPLICATION ]

AS

BEGIN

[declaration\_section]

executable\_section

END;

* Transaction: TRANSACTION trong SQL là tiến trình thực hiện một nhóm các câu lệnh SQL. Các câu lệnh này được thực thi một cách tuần tự và độc lập. Một Transaction được thực hiện thành công khi tất cả câu lệnh đều thành công, khi đó tất cả các thay đổi dữ liệu được thực hiện trong Transaction được lưu vào cơ sở dữ liệu. Tuy nhiên, nếu chỉ một trong số đó thất bại thì toàn bộ tiến trình sẽ thất bại, đồng nghĩa với việc dữ liệu phải rollback về trạng thái ban đầu (dữ liệu được khôi phục về trạng thái trước khi thực hiện Transaction).

Các câu lệnh cơ bản trong transaction:

* COMMIT: để lưu các thay đổi.
* ROLLBACK: để quay trở lại trạng thái trước khi có thay đổi.
* SAVEPOINT: tạo các điểm (point) bên trong các nhóm Transaction để ROLLBACK, tức là để quay trở lại điểm trạng thái đó.
* SET TRANSACTION: đặt một tên cho một Transaction.

Các lệnh điều khiển Transaction chỉ được sử dụng với các lệnh thao tác dữ liệu như INSERT, UPDATE và DELETE. Tuy nhiên chúng không thể được sử dụng trong lệnh CREATE TABLE hoặc DROP TABLE vì các hoạt động này được tự động xác định trong cơ sở dữ liệu.

* Index:

CREATE UNIQUE INDEX index\_name ON table\_name ( column1, column2,...);

Tạo Unique Index: có nghĩa là hai hàng không thể có cùng giá trị chỉ mục.

Để tạo chỉ lục các giá trị trong cột theo thứ tự giảm dần, thêm từ khóa DESC

CREATE UNIQUE INDEX index\_name ON table\_name ( column1, column2,... DESC);