Ex03\_Mysql

Phạm Hải Triều

Báo cáo

[SQL là gì? 1](#_Toc99618744)

[NoSQL là gì? 2](#_Toc99618745)

[﻿﻿﻿﻿Lệnh Join (Tham gia) là gì? 5](#_Toc99618747)

[Các loại lệnh Join 5](#_Toc99618748)

[Lệnh Join tham Gia Chéo ( Cross Join ) 5](#_Toc99618749)

[Lệnh Join: INNER JOIN 6](#_Toc99618750)

[Lệnh Join: Outer Join 7](#_Toc99618751)

[Lệnh Join: Left Join 7](#_Toc99618752)

[Lệnh Join: Right Join 7](#_Toc99618753)

# SQL là gì?

SQL chính là viết tắt của Structured Query language, được phát âm là "S-Q-L" hoặc đôi khi là "See-Quel" là ngôn ngữ chuẩn để xử lý Cơ sở dữ liệu quan hệ. Một cơ sở dữ liệu quan hệ xác định các mối quan hệ dưới dạng các bảng.

Lập trình SQL có thể được sử dụng hiệu quả để chèn, tìm kiếm, cập nhật, xóa các bản ghi cơ sở dữ liệu.

Điều đó không có nghĩa là SQL không thể làm được gì ngoài việc đó. Nó có thể làm rất nhiều thứ bao gồm, nhưng không giới hạn, tối ưu hóa và duy trì cơ sở dữ liệu.

Các cơ sở dữ liệu quan hệ như Cơ sở dữ liệu MySQL, Oracle, MS SQL Server, Sybase, vv sử dụng SQL.

# NoSQL là gì?

NoSQL là một DMS không quan hệ, không yêu cầu một lược đồ cố định, tránh các phép nối, và dễ dàng mở rộng. Cơ sở dữ liệu NoSQL được sử dụng cho các kho dữ liệu phân tán với nhu cầu lưu trữ dữ liệu khổng lồ. NoSQL được sử dụng cho dữ liệu lớn và các ứng dụng web thời gian thực. Ví dụ như các công ty như Twitter, Facebook, Google thu thập hàng terabyte dữ liệu người dùng mỗi ngày.

RDBMS truyền thống sử dụng cú pháp SQL để lưu trữ và truy xuất dữ liệu để có thêm thông tin chi tiết. Thay vào đó, một hệ thống cơ sở dữ liệu NoSQL bao gồm một loạt các công nghệ cơ sở dữ liệu có thể lưu trữ dữ liệu có cấu trúc, bán cấu trúc, không có cấu trúc và đa hình.

# ﻿﻿﻿﻿Lệnh Join (Tham gia) là gì?

**Lệnh join**giúp lấy dữ liệu từ hai hoặc nhiều bảng cơ sở dữ liệu. Các bảng có liên kết với nhau bằng cách sử dụng khóa chính và khóa ngoài.

# Các loại lệnh Join

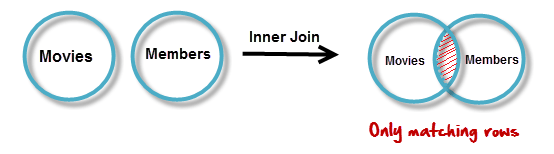
## Lệnh Join tham Gia Chéo ( Cross Join )

Tham Gia Chéo là một hình thức Tham Gia đơn giản nhất nó khớp với từng hàng từ một bảng cơ sở dữ liệu với tất cả các hàng khác.

Cú pháp : SELECT \* FROM `movies` CROSS JOIN `members`

## Lệnh Join: INNER JOIN

Inner Join là một **lệnh Join** được sử dụng để kết hợp các hàng từ cả hai bảng thỏa mãn điều kiện đã cho.



SELECT members.`first\_name` , members.`last\_name` , movies.`title`

FROM members ,movies

WHERE movies.`id` = members.`movie\_id`

Lưu ý kết quả trên cũng có thể được viết như sau để đạt được kết quả tương tự.

SELECT A.`first\_name` , A.`last\_name` , B.`title`

FROM `members`AS A

INNER JOIN `movies` AS B

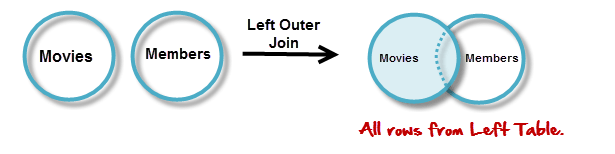
ON B.`id` = A.`movie\_id` `

## Lệnh Join: Outer Join

Outer Join là **lệnh Join** sử dụng cho tất cả kết quả các bản ghi khớp từ cả hai bảng.

Nó có thể phát hiện các bản ghi không có kết quả khớp trong bảng đã tham gia. Nó cho giá trị NULL các bản ghi của bảng đã tham gia nếu không tìm thấy kết quả khớp.

## Lệnh Join: Left Join



Left Join cho kết quả tất cả các hàng từ bảng bên trái ngay cả khi không tìm thấy hàng phù hợp trong bảng bên phải. Trường hợp không tìm thấy kết quả phù hợp trong bảng bên phải, kết quả sẽ là [Null](https://gitiho.com/blog/huong-dan-tim-hieu-ve-null-va-not-null-trong-mysql.html).

SELECT A.`title` , B.`first\_name` , B.`last\_name`

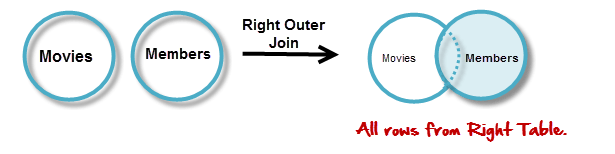
FROM `movies` AS A

LEFT JOIN `members` AS B

ON B.`movie\_id` = A.`id`

## Lệnh Join: Right Join

Right Join trái ngược với Left Join. Right Join là **lệnh Join** cho tất cả kết quả các cột từ bảng bên phải ngay cả khi không tìm thấy hàng phù hợp trong bảng bên trái. Trường hợp không tìm thấy kết quả phù hợp trong[bảng](https://gitiho.com/blog/cach-xu-ly-loi-va-dong-trong-tren-bang-bao-cao-pivottable-cua-excel-macbook.html) bên trái, kết quả là Null



SELECT A.`first\_name` , A.`last\_name`, B.`title`

FROM `members` AS A

RIGHT JOIN `movies` AS B

ON B.`id` = A.`movie\_id`

# Function và Store Procedure (hàm/thủ tục)

##### Function và Store Procedure là gì

* Là đoạn chương trình kịch bản (programming scripts) với các câu lệnh SQL nhúng (embedded SQL) được lưu dưới dạng đã được biên dịch và thi hành thực tiếp bởi MySQL server.
* SP cho phép lưu trữ các logic ứng dụng trên CSDL.

##### Cú pháp

* Tạo hàm/thủ tục

CREATE FUNCTION name ([parameterlist]) RETURNS datatype [options] sqlcode

CREATE PROCEDURE name ([parameterlist]) [options] sqlcode

Ví dụ:

DELIMITER $$

CREATE PROCEDURE count\_people\_with\_age(IN age\_in INT, OUT quantity INT)

BEGIN

SELECT \*

FROM people

WHERE age >= age\_in;

SELECT FOUND\_ROWS() INTO quantity;

END

* DELIMITER $$: Giải phóng để cho phép sử dụng ‘;’ trong thủ tục
* Gán giá trị cho biến
  + Sử dụng SET hoặc SELECT INTO.
* Gọi thủ tục:

Call count\_people\_with\_age (18,@people\_count);

Select @ people \_count;

* Xóa Hàm/Thủ tục:

DROP FUNCTION [IF EXISTS] name

DROP PROCEDURE [IF EXISTS] name

* Hiện thị thông tin

SHOW FUNCTION STATUS;

SHOW PROCEDURE STATUS;

SHOW FUNCTION STATUS LIKE 'repeat%';

SHOW PROCEDURE STATUS LIKE 'film%';

SHOW CREATE PROCEDURE 'name';

##### Một số cấu trúc điều khiển dùng trong Hàm, thủ tục:

* Mệnh đề IF THEN

IF condition THEN

commands;

[ELSE IF condition THEN

commands;]

[ELSE

commands;]

END IF;

* Mệnh đề REPEAT UNTIL

[loopname:]

REPEAT commands;

UNTIL condition

END REPEAT

[loopname];

* Mệnh đề CASE

CASE expression

WHEN value1 THEN commands;

[WHEN value2 THEN commands;]

[ELSE commands;]

END CASE;

* Mệnh đề WHILE

[loopname:]

WHILE condition

DO commands;

END WHILE

[loopname];

##### Con trỏ (cursor) trong hàm/thủ tục

* MySQL hỗ trợ con trỏ trong hàm/thủ tục, trigger
* Con trỏ được sử dụng để duyệt các dòng kết quả trả lại bởi truy vấn, và xử lý từng dòng riêng lẻ
* Cú pháp Con trỏ:

DECLARE cursor\_name CURSOR FOR SELECT\_statement;

OPEN cursor\_name;

* Lấy ra từng dòng và chuyển tới dòng kết quả tiếp theo sử dụng lệnh FETCH

FETCH cursor\_name INTO variable list;

CLOSE cursor\_name;

##### Xử lý lỗi thông qua Handlers

* Luôn có khả năng hàm/thủ tục gặp lỗi trong khi thi hành các lệnh SQL. MySQL cung cấp kỹ thuật xử lý lỗi thông qua handler
* Một handler cần định nghĩa sau khai báo các biến, con trỏ và điều kiện, nhưng trước các lệnh SQL
* Cú pháp Handler:

DECLARE type HANDLER FORcondition1, condition2, condition3, ... statement;

* type: CONTINUE hoặc EXIT
* condition(s): Các điều kiện mà handler sẽ được gọi : Not found, SqlException,..
* statement: Câu lệnh sẽ thi hành khi có điều kiện xảy ra

##### Ưu nhược điểm

* Ưu điểm
  + Giảm dư thừa mã chương trình: Các đoạn mã tương tự trong các ứng dụng như thêm, cập nhật có thể lưu ở phía CSDL
  + Bảo trì: Nếu có sự thay đổi trong CSDL, mã lệnh cần thay đổi có thể xác định trong các SP
  + An ninh CSDL tốt hơn: Trong các ứng dụng an ninh cao, với hàm/thủ tục có thể kiểm soát truy cập dữ liệu và đưa ra các qui định an ninh tập trung
* Nhược điểm
  + Thiếu tính khả chuyển (Lack of Portability)  
    SP khó chuyển từ một DBMS sang một DBMS khác. Yêu cầu lập trình và kiểm thử lại đáng kể
  + Tải DB Server  
    Sử dụng hàm/thủ tục nhiều có thể gây quá tải MySQL server
  + Hạn chế ngôn ngữ lập trình  
    Lập trình SP không phong phú như các nền tảng phát triển khác như Java hay PHP  
    Lưu ý: Các phiên bản tương lai của MySQL có thể cung cấp các giao diện cho phép tạo các hàm/thủ tục sử dụng các ngôn ngữ bên ngoài, ví dụ Java

# Transaction là gì?

Trong MySQL, transaction bắt đầu với câu lệnh BEGIN WORK và kết thúc với một trong hai câu lệnh COMMIT hoặc ROLLBACK. Các câu lệnh SQL đặt giữa [BEGIN WORK … COMMIT/ROLLBACK] là phần chính của một transaction.

## Lệnh COMMIT:

* Khi một transaction hoàn chỉnh được hoàn thành thì lệnh COMMIT phải được đưa ra để mọi sự thay đổi đều được tác động đến tất cả các table liêu quan.

## Lệnh ROLLBACK :

* Nếu bị lỗi, thì lệnh ROLLBACK nên được đưa ra để đưa tất cả các table liên quan với transaction về lại trạng thái trước đó. Ta có thể điều khiển hành vi của một transaction bằng cách thiết lập một biến phiên, được gọi là Autocommit. Nếu Autocommit được thiết lập bằng 1, SET AUTOCOMMIT = 1 (default), thì mỗi lệnh SQL (trong hoặc ngoài transaction) được xem như một transaction hoàn chỉnh, điều này được duy trì cho đến khi nó kết thúc. Khi Autocommit được thiết lập bằng 0, SET AUTOCOMMIT = 0, thì một tập, có thứ tự, của các câu lệnh sẽ đóng vai trò như là một transaction và không có một hoạt động nào là được xuất phát cho đến có một lệnh Commit khác được đưa ra. Ta có thể sử dụng các câu lệnh này trong PHP bằng cách sử dụng hàm mysql\_query().

# . MySQL Index là gì?

Index là dữ liệu có cấu trúc như B-Tree giúp cải thiện tốc độ tìm kiếm trên một bảng, làm giảm chi phí thực hiện truy vấn. Việc tối ưu hóa chỉ mục sẽ giúp xác định được vị trí của dữ liệu cần tìm thay vì phải dò theo thứ tự hàng triệu record trong bảng.