

Transaction trong MySQL

Transaction trong SQL là một đơn vị công việc được thực hiện bởi một Database. Transaction là đơn vị hoặc dãy công việc được thực hiện theo một thứ tự logic và hợp lý, có thể được thao tác bởi người dùng hoặc bởi một Database program.

Một transaction là một sự lan truyền của một hoặc nhiều thay đổi tới Database. Ví dụ, nếu bạn đang tạo một bản ghi hoặc cập nhật một bản ghi hoặc xóa một bản ghi từ một bảng, thì bạn đang thực hiện transaction trên bảng đó. Nó là quan trọng để điều khiển các transaction để bảo đảm toàn vẹn dữ liệu và để xử lý các Database Error.

Nói cách khác, một Transaction sẽ không bao giờ hoàn thành trừ khi mỗi hoạt động riêng bên trong nhóm được thực hiện thành công. Nếu bất kỳ hoạt động nào bên trong Transaction thất bại, thì toàn bộ Transaction đó sẽ thất bại.

Thực tế, bạn sẽ gộp nhiều truy vấn SQL vào thành một nhóm và bạn sẽ thực thi tất cả chúng cùng với nhau như là một Transaction.

Property của Transaction trong MySQL

Transaction có 4 property chuẩn và thường được biết đến với tên lược danh là ACID:

- **Atomicity:** bảo đảm rằng tất cả hoạt động bên trong đơn vị công việc được hoàn thành một cách thành công; nếu không, transaction bị ngừng ở điểm thất bại, và các hoạt động trước được trao trả về trạng thái trước đó.
- **Consistency:** bảo đảm rằng Database thay đổi một cách chính xác trạng thái theo một transaction đã được ký thác thành công.
- **Isolation:** cho các transaction khả năng hoạt động một cách độc lập và không liên quan đến nhau.
- **Durability:** bảo đảm rằng kết quả hoặc tác động của một transaction, mà đã được ký thác, vẫn còn tồn tại trong trường hợp hệ thống thất bại.

Trong MySQL, các Transaction bắt đầu với lệnh BEGIN WORK và kết thúc với hoặc một lệnh COMMIT hoặc lệnh ROLLBACK. Các lệnh SQL giữa các lệnh bắt đầu và kết thúc cấu tạo nên Transaction.

Lệnh COMMIT và ROLLBACK trong MySQL

Hai từ khóa **Commit** và **Rollback** được sử dụng chủ yếu cho Transaction trong MySQL.

- Khi một Transaction được hoàn thành, lệnh COMMIT nên được thông báo để mà tất cả thay đổi tới các bảng liên quan sẽ có hiệu quả.
- Nếu một lỗi xuất hiện, một lệnh ROLLBACK nên được thông báo để trả bảng đã tham chiếu trong Transaction về trạng thái trước đó.

Bạn có thể điều khiển cách vận hành của một Transaction bằng việc thiết lập biến session gọi là **AUTOCOMMIT**. Nếu AUTOCOMMIT được thiết lập là 1 (mặc định), thì mỗi lệnh SQL (bên trong một Transaction hoặc không) được xem như là một Transaction đã kết thúc và được ký thác theo mặc định khi nó kết thúc. Khi AUTOCOMMIT được thiết lập là 0, thì bằng việc thông báo lệnh SET AUTOCOMMIT=0, các chuỗi lệnh theo sau hoạt động như một Transaction và không có hoạt động nào được ký thác tới khi một lệnh COMMIT tường minh được thông báo.

Ví dụ chung về Transaction trong MySQL

Dãy sự kiện là độc lập với ngôn ngữ chương trình được sử dụng; logic path có thể được tạo trong bất kỳ ngôn ngữ nào bạn sử dụng để tạo ứng dụng của mình.

Bạn có thể thực thi các lệnh SQL trong PHP bởi sử dụng hàm **mysql_query()**.

- Bắt đầu Transaction bằng việc thông báo lệnh **BEGIN WORK**.
- Thông báo một hay nhiều lệnh SQL như SELECT, INSERT, UPDATE hoặc DELETE.
- Kiểm tra xem có lỗi nào hay không và mọi thứ có theo như yêu cầu của bạn không.
- Nếu có bất kỳ lỗi nào, thì bạn thông báo lệnh ROLLBACK, nếu không có, bạn thông báo một lệnh COMMIT.

Các loại bảng bảo vệ Transaction trong MySQL

Bạn không thể sử dụng các Transaction một cách trực tiếp, bởi vì chúng không an toàn và không được bảo vệ. Nếu bạn muốn sử dụng các Transaction trong lập trình MySQL, thì bạn cần tạo các bảng theo một cách đặc biệt. Có nhiều loại bảng hỗ trợ cho các Transaction nhưng loại phổ biến nhất là **InnoDB**.

Sự hỗ trợ cho các bảng InnoDB yêu cầu một tham số biên dịch cụ thể khi biên dịch MySQL từ nguồn. Nếu phiên bản MySQL của bạn không có hỗ trợ cho InnoDB, bạn yêu cầu nhà cung cấp dịch vụ Internet để xây dựng một phiên bản MySQL với sự hỗ trợ cho loại bảng InnoDB hoặc tải và cài đặt MySQL-Max Binary Distribution cho Windows hoặc Linux/UNIX và làm việc với loại bảng này trong môi trường phát triển.

Nếu cài đặt MySQL của bạn hỗ trợ các bảng InnoDB, bạn chỉ cần đơn giản thêm định nghĩa **TYPE=InnoDB** ở phần lệnh tạo bảng. Ví dụ sau tạo một bảng InnoDB gọi là `sinhvienk60`:

```
//Lua chon co so du lieu USE sinhvien; //Tao bang sinhvienk60 CREATE TABLE
sinhvienk60 ( ten VARCHAR(40) NOT NULL, diemthi INT ) TYPE=InnoDB;
```

Bạn có thể sử dụng các loại bảng khác như **GEMINI** hoặc **BDB**, tùy thuộc vào cài đặt của bạn có hỗ trợ các loại bảng này không.