### What are Transactions

Trong MongoDB, một thao tác ghi trên một tài liệu là nguyên tử, ngay cả khi thao tác đó sửa đổi nhiều tài liệu nhúng trong một tài liệu. Khi một thao tác ghi duy nhất, như updateMany, sửa đổi nhiều tài liệu, việc sửa đổi từng tài liệu là nguyên tử, nhưng toàn bộ hoạt động không phải là nguyên tử. Một số trường hợp sử dụng cũng có thể yêu cầu bạn sửa đổi nhiều thao tác ghi như một phần của thao tác đơn. Trong các kịch bản như vậy, bạn cần các Transactions để thực thi tính nguyên tử trên nhiều thao tác ghi.

### Why use Transactions

Cho đến khi MongoDB v4.0, cách duy nhất bạn có thể mô phỏng các Transactions là thực hiện cam kết hai pha trong ứng dụng của bạn. Emmanuel Olaojo đã viết một bài báo, Fawn: Transactions trong MongoDB, trên các cam kết hai pha bằng mô-đun Fawn cho các ứng dụng Node.js. Xin lưu ý rằng các cam kết hai pha chỉ có thể cung cấp ngữ nghĩa giống như Transactions. Trong khi sử dụng cam kết hai pha, các ứng dụng có thể trả về dữ liệu trung gian trong quá trình cam kết hai pha hoặc khôi phục.

MongoDB v4.0 giới thiệu các Transactions đa tài liệu cho các bộ bản sao và có thể được sử dụng trên nhiều hoạt động, bộ sưu tập và tài liệu. Các Transactions đa tài liệu cung cấp một cái nhìn thống nhất về dữ liệu trên toàn cầu và thực thi tất cả hoặc không có gì để duy trì tính toàn vẹn dữ liệu. Các Transactions sẽ có các thuộc tính sau

Khi một Transactions được cam kết, tất cả các thay đổi dữ liệu được thực hiện trong Transactions sẽ được lưu lại.

Nếu bất kỳ hoạt động nào trong Transactions không thành công, Transactions sẽ hủy bỏ

Khi một Transactions bị hủy bỏ / hủy bỏ, tất cả các thay đổi dữ liệu được thực hiện trong Transactions sẽ bị loại bỏ.

Cho đến khi một Transactions được cam kết, không có thao tác ghi nào trong Transactions được hiển thị bên ngoài Transactions.

## \*Những tính năng của transaction

## Multi-Document Transactions

Khi một thao tác ghi đơn (ví dụ: db.collection.updateMany ()) sửa đổi nhiều tài liệu, việc sửa đổi từng tài liệu là atomic, nhưng toàn bộ hoạt động không phải là atomic.

Khi thực hiện các thao tác ghi nhiều tài liệu, cho dù thông qua một thao tác ghi hoặc nhiều thao tác ghi, các hoạt động khác có thể xen kẽ.

Bắt đầu từ phiên bản 4.0, đối với các tình huống yêu cầu tính nguyên tử để cập nhật nhiều tài liệu hoặc tính nhất quán giữa các lần đọc thành nhiều tài liệu, MongoDB cung cấp các Transactions đa tài liệu cho các bộ bản sao.

Trong hầu hết các trường hợp, multi-document transaction phải chịu chi phí hiệu năng cao hơn so với ghi tài liệu đơn lẻ và tính khả dụng của Transactions đa tài liệu không nên thay thế cho thiết kế lược đồ hiệu quả. Đối với nhiều kịch bản, mô hình dữ liệu không chuẩn hóa (tài liệu nhúng và mảng) sẽ tiếp tục tối ưu cho dữ liệu và trường hợp sử dụng của bạn. Đó là, đối với nhiều kịch bản, mô hình hóa dữ liệu của bạn một cách phù hợp sẽ giảm thiểu nhu cầu Transactions đa tài liệu.

## Concurrency Control(Kiểm soát đồng thời)

Concurrency control cho phép nhiều ứng dụng chạy đồng thời mà không gây ra sự mâu thuẫn hoặc xung đột dữ liệu.

Một cách tiếp cận là tạo một chỉ mục duy nhất trên một trường chỉ có thể có các giá trị duy nhất. Điều này ngăn việc chèn hoặc cập nhật tạo dữ liệu trùng lặp. Tạo một chỉ mục duy nhất trên nhiều trường để buộc tính duy nhất trên sự kết hợp các giá trị trường đó. Để biết ví dụ về các trường hợp sử dụng, hãy xem update () và Unique Index và findAndModify () và Unique Index.

Một cách tiếp cận khác là chỉ định giá trị hiện tại dự kiến của một trường trong vị từ truy vấn cho các thao tác ghi.

## ACID transactions in MongoDB 4.0

Các thuộc tính ACID nổi tiếng trong thế giới của các cơ sở dữ liệu quan hệ, nhưng hãy để lại tóm tắt ý nghĩa của từ viết tắt.  
  
Tính atomic: một nhóm các lệnh bên trong Transactions phải tuân theo mô hình toàn bộ hoặc không có gì. Nếu chỉ một trong các lệnh thất bại vì bất kỳ lý do nào, Transactions hoàn chỉnh cũng thất bại.  
**Consistency**: nếu một Transactions thực hiện thành công, nó sẽ lấy cơ sở dữ liệu từ một trạng thái phù hợp với trạng thái khác cũng phù hợp.  
**Isolation**: nhiều Transactions có thể chạy cùng một lúc trong hệ thống. Cách ly đảm bảo rằng mỗi Transactions không thể xem kết quả một phần của các Transactions khác. Thực hiện nhiều Transactions song song phải có kết quả giống như chạy chúng tuần tự  
**Durability**: nó đảm bảo rằng một Transactions đã cam kết sẽ tồn tại lâu dài, ngay cả trong trường hợp lỗi hệ thống

## Limitations of transactions:

Hỗ trợ cho các Transactions giới thiệu một số hạn chế:

một bộ sưu tập PHẢI tồn tại để sử dụng các Transactions

một bộ sưu tập không thể được tạo hoặc bỏ trong Transactions

một chỉ mục không thể được tạo hoặc bỏ trong Transactions

Các hoạt động không phải CRUD không được phép trong một Transactions (ví dụ: các lệnh quản trị như creatUser không được phép)

một Transactions không thể đọc hoặc ghi trong cấu hình, quản trị viên và cơ sở dữ liệu cục bộ

một Transactions không thể ghi vào hệ thống. \* bộ sưu tập

kích thước của một Transactions được giới hạn ở 16MB

một mục nhập oplog duy nhất được tạo trong quá trình cam kết: ghi trong Transactions don don có các mục oplog duy nhất như trong các truy vấn thông thường

giới hạn là kết quả của kích thước tối đa 16 MB của bất kỳ tài liệu BSON nào trong oplog

trong trường hợp Transactions lớn hơn, bạn nên xem xét chia chúng thành các Transactions nhỏ hơn

theo mặc định, một Transactions thực hiện lâu hơn 60 giây sẽ tự động hết hạn

bạn có thể thay đổi điều này bằng cách sử dụng tham số cấu hình TransactionsLifetimeLimitSeconds

các Transactions dựa trên khả năng chụp nhanh của WiredTiger và việc Transactions được thực hiện trong thời gian dài có thể dẫn đến áp lực cao đối với bộ đệm của WiredTiger, để duy trì các ảnh chụp nhanh và dẫn đến việc giữ lại rất nhiều thao tác không được lưu trong bộ nhớ.