**RavenDB vs MSSQL**

# Hiệu suất:

Chúng ta được nghe nói RavenDB xử lý nhanh và nhanh hơn cả MSSQL. Điều này thì thực sự không đúng. RavenDB thực sự chính nó không nhanh hơn MSSQL nhưng những ựng xây dựng với RavenDB thường nhanh hơn so với những ứng dụng xây dựng dựa trên MSSQL. Có một số lý do giải thích cho điêu này như sau:

* Giảm số lượng các yêu cầu truy cập tới database:
  + RavenDB cho phép bạn lưu trữ toàn bộ đối tượng và các tập hợp bên trong một document duy nhất. Đó là lý do tại sao chúng ta gửi yêu cầu tới RavenDB server ít hơn so với MSSQL.
* Không có join khi truy vấn:
  + Mặc dù RavenDB có một số khái niệm hỗ trợ join giữa các document, nhưng nó bao gồm luôn cả việc phi chuẩn hóa dữ liệu. Thay vì lấy tất cả thông tin cần thiết từ nhiều document khác nhau bằng cách gôm chung vào 1 câu truy vấn , chúng ta đã có tất cả thông tin cần thiết bên trong một document duy nhất. RavenDB rất nhanh trong việc load lên một document. Đó là lý do cho việc tăng hiệu suất ở đây.
* Indexes: tính toán trước các dữ liệu kiểu tập hợp
  + Map/reduce được sử dụng để tính toán bên dưới, chạy nền (không ảnh hưởng đến việc truy cập cơ sở dữ liệu của chương trình), và lưu trữ kết quả tính toán trên đĩa cứng. Vì thế những truy vấn đó rất ít tốn chi phí vì tất cả các tính toán đã được thực hiện. Tương tự như vậy, các indexes map thông thường cũng cho kết quả truy vấn nhanh. Tuy nhiên chúng ta cũng nên lưu ý là, indexes có BASE consistency (tính nhất quán cơ bản), điều đó có nghĩa là, kết quả chúng ta có được có thể là kết quả cũ và cần phải nắm bắt được thao tác ghi xuống cơ sở dữ liệu mới nhất. Điều đó cũng có nghĩa là trong những hệ thống lớn (busy systems), chúng ta không được đảm bảo rằng chúng ta sẽ luôn có kết quả mới nhất dựa vào index, nhưng điều đó lại là một điểm mạnh của RavenDB theo một hướng khác, bởi vì những thao tác viết xuống cơ sở dữ liệu không bị ảnh hưởng bởi số lượng các indexes. Nó hoàn toàn đơn giản và RavenDB cập nhật indexes dưới nền mà không làm chậm hiệu suất thao tác insert/update dữ liệu.

# Tính đơn giản

* Mặc dù RavenDB đã cố gắng làm cho việc sử dụng trở nên đơn giản nhất có thể, nhưng vẫn còn những trở ngại trong việc thay đổi suy nghĩ của nhà phát triển về việc mô hình dữ liệu. Chúng ta cần suy nghĩ về dữ liệu theo một cách khác so với trước đây đã từng làm với cơ sở dữ liệu quan hệ. Đây là những thứ mà chúng ta cần xác định rõ ràng trước khi chọn RavenDB làm công nghệ lưu trữ cho ứng dụng mới phát triển với thời hạn hoàn thành giới hạn.
* Tuy nhiên, học cách mô hình dữ liệu các document với RavenDB và làm việc với RavenDB sẽ là những kinh nghiệm bổ ích bởi vì chúng ta sẽ có được những thông tin phản hồi ngay lập tức. Và khi chúng ta hoàn thành việc học này rồi thì việc phát triển ứng dụng sẽ nhanh chóng và dễ dàng hơn rất nhiều.
* Mặc dù chúng ta có EF 4.1 code first, FNH với việc automappings, thậm chí là micro-orms như Massive hay là Petapoco làm rất tốt trong việc đơn giản quá trình phát triển ứng dụng khi làm việc với SQL server, nhưng nó còn dễ dàng hơn khi làm việc với RavenDB. Chúng ta không cần nghĩ về việc mapping và normalization, chỉ cần có đối tượng là xong.

# Tooling và ecosystem (công cụvà hệ sinh thái)

* Đây là những thứ chúng ta sẽ thiếu nếu đã quen làm việc với MSSQL hoặc sử dụng phần mềm của hãng thứ ba giúp chúng ta làm việc và quản lý SQL servers
* Một khó khăn nữa là khi làm việc với RavenDB (cũng như hầu hết các cơ sở dữ liệu NoSQL nói chung) là không thể nhận được hỗ trợ tốt ad-hoc reporting. RavenDB có một gói cho phép bạn sao chép dữ liệu sang SQL server để sử dụng cho mục đích reporting nhưng rõ ràng như thế sẽ rất bất tiện so với việc truy vấn trên cơ sở dữ liệu gốc.

# Nên chọn RavenDB hay MSSQL?

Câu hỏi nên chọn RavenDB hay MSSQL rõ ràng là phụ thuộc vào từng trường hợp cụ thể, kỹ năng của nhóm, môi trường phát triển … nhưng chúng ta có một số tiêu chí để chọn RavenDB hay MSSQL như sau:

* Chọn RavenDB khi:
  + Chúng ta có thể nghĩ dữ liệu của chúng ta theo kiểu tập hợp mà hoàn toàn độc lập với khối dữ liệu (ví dụ như: customer, order, product…)
  + Cần hiệu suất tốt trên việc tính toán và truy vấn kết hợp
  + Cần tìm kiếm phức tạp (full-text, facets,…)
  + Cần mô hình mở rộng tốt
  + Ứng dụng có tính sẵn sàng cao với chi phí thấp
* Chọn MSSQL khi:
  + Cần hỗ trợ người dùng tạo ra các báo cáo và phân tích dữ liệu động
  + Làm việc chủ yếu với dữ liệu quan hệ ( ví dụ: accounting, statistics…)
  + Muốn sử dụng Windows Azure
  + Khách hàng muốn phát triển ứng dụng trên MSSQL mà không biết có sự thay thế khác tốt hơn.

Lưu ý: chúng ta không nhất thiết phải chọn một trong 2 cơ sở dữ liệu trên hoặc là những cơ sở dữ liệu khác. Thực tế chúng ta có thể kết hợp chúng và tận dụng những điểm mạnh của từng loại cơ sở dữ liệu.