

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỆ THỐNG THÔNG TIN



BÀI TẬP LỚN 2 – NOSQL

MÔN HỌC: CƠ SỞ DỮ LIỆU PHÂN TÁN

ĐỀ TÀI: Tìm hiểu và cài đặt MongoDB

Giảng viên giảng dạy: Nguyễn Minh Nhựt

Lớp: IS211.P21

Nhóm sinh viên thực hiện

- | | |
|--------------------------|----------|
| 1. Nguyễn Thị Hải Hà | 22520372 |
| 2. Nguyễn Thị Thanh Hiếu | 22520443 |
| 3. Nguyễn Hồng Nhật Long | 22520816 |
| 4. Phạm Đức Mạnh | 22520852 |

TPHCM, ngày 03 tháng 06 năm 2025

LỜI CẢM ƠN

Trước tiên, nhóm chúng em xin được bày tỏ lòng biết ơn sâu sắc đến toàn thể quý thầy cô Trường Đại học Công nghệ Thông tin – Đại học Quốc gia TP.HCM, đặc biệt là quý thầy cô Khoa Hệ thống Thông tin, những người đã luôn tận tâm trong giảng dạy và tạo điều kiện thuận lợi cho sinh viên được tiếp cận với kiến thức tiên tiến, môi trường học tập hiện đại và các công nghệ mới trong lĩnh vực khoa học dữ liệu và công nghệ thông tin. Trong suốt quá trình học tập và nghiên cứu, chúng em không chỉ được trang bị kiến thức nền tảng vững chắc mà còn được truyền cảm hứng học thuật và tinh thần đổi mới sáng tạo.

Đặc biệt, nhóm xin gửi lời cảm ơn chân thành và lòng kính trọng sâu sắc đến **Thầy Nguyễn Minh Nhựt**, người đã trực tiếp hướng dẫn và đồng hành cùng chúng em trong suốt quá trình thực hiện báo cáo. Thầy không chỉ hỗ trợ về mặt chuyên môn với sự tận tụy, nghiêm túc và trách nhiệm, mà còn luôn khuyến khích chúng em suy nghĩ độc lập, khai thác nhiều nguồn tư liệu. Những đóng góp quý báu từ thầy đã giúp chúng em mở rộng tư duy, rèn luyện khả năng nghiên cứu và tiếp cận vấn đề một cách khoa học, logic và sáng tạo hơn.

Chúng em cũng xin chân thành cảm ơn toàn thể các thành viên trong lớp đã cùng nhau học tập, chia sẻ tài liệu, kinh nghiệm và động viên nhau trong suốt quá trình qua. Sự gắn kết và tinh thần tập thể chính là yếu tố then chốt giúp môi trường học tập trở nên tích cực, hỗ trợ lẫn nhau cùng tiến bộ.

Mặc dù đã nỗ lực hết mình, nhưng với kinh nghiệm còn hạn chế, nên báo cáo không tránh khỏi những thiếu sót. Vì vậy, nhóm rất mong nhận được những nhận xét, phản hồi và góp ý chân thành từ quý thầy để không ngừng nâng cao năng lực bản thân, rút kinh nghiệm cho những dự án học thuật và nghề nghiệp trong tương lai.

Chúng em xin chân thành cảm ơn!

TPHCM, ngày 03 tháng 06 năm 2025

Nguyễn Thị Hải Hà – 22520372

Nguyễn Thị Thanh Hiếu – 22520443

Nguyễn Hồng Nhật Long - 22520816

Phạm Đức Mạnh - 22520852

MỤC LỤC

1. NOSQL Database Management Systems	Error! Bookmark not defined.
1.1. Giới thiệu chung	Error! Bookmark not defined.
1.1.1. Định nghĩa	Error! Bookmark not defined.
1.1.2. Lịch sử và nguồn gốc của NoSQL	Error! Bookmark not defined.
1.2. Các loại NoSQL.....	Error! Bookmark not defined.
1.2.1. Document-Based (Dựa trên tài liệu).....	Error! Bookmark not defined.
1.2.2. Key-Value Store (Khoá – Giá trị).....	Error! Bookmark not defined.
1.2.3. Column-Family (Dựa trên cột).....	Error! Bookmark not defined.
1.2.4. Graph-Based (Dựa trên đồ thị)	Error! Bookmark not defined.
1.3. Công cụ sử dụng	Error! Bookmark not defined.
2. MONGODB.....	Error! Bookmark not defined.
2.1. Giới thiệu	Error! Bookmark not defined.
2.2. Tính năng của MongoDB	Error! Bookmark not defined.
2.3. Kiến trúc hệ thống	Error! Bookmark not defined.
2.4. Lưu trữ dữ liệu	Error! Bookmark not defined.
2.5. Data Flow	Error! Bookmark not defined.
2.6. Nhận xét MongoDB.....	Error! Bookmark not defined.
2.6.1. Ưu điểm	Error! Bookmark not defined.
2.6.2. Hạn chế	Error! Bookmark not defined.
3. CÀI ĐẶT	Error! Bookmark not defined.
3.1.1. Cài đặt MongoDB trên hai máy.....	Error! Bookmark not defined.
3.1.2. Chạy MongoDB Server (mongod)	Error! Bookmark not defined.
4. THỬ NGHIỆM.....	Error! Bookmark not defined.
5. PHÂN CÔNG CÔNG VIỆC	Error! Bookmark not defined.
6. TÀI LIỆU THAM KHẢO	Error! Bookmark not defined.

1. NOSQL Database Management Systems	5
1.1. Giới thiệu chung	5
1.1.1. Định nghĩa	5
1.1.2. Lịch sử và nguồn gốc của NoSQL	5
1.2. Các loại NoSQL.....	5
1.2.1. Document-Based (Dựa trên tài liệu).....	5
1.2.2. Key-Value Store (Khoá – Giá trị).....	6
1.2.3. Column-Family (Dựa trên cột).....	6
1.2.4. Graph-Based (Dựa trên đồ thị)	7
1.3. Ưu điểm	7
2. MONGODB.....	8
2.1. Giới thiệu	8
2.1.1. Định nghĩa	8
2.1.2. Nguồn gốc ra đời	8
2.1.3. Tại sao nên sử dụng MongoDB?	9
2.1.4. Khi nào nên sử dụng MongoDB?.....	10
2.1.5. RDBMS vs MongoDB	10
2.1.6. Ứng dụng thực tế	11
2.2. Tính năng của MongoDB	11
2.3. Kiến trúc hệ thống	12
2.4. Lưu trữ dữ liệu	14
2.5. Data Flow	14
2.5.1. Luồng dữ liệu cơ bản trong MongoDB	14
2.5.2. Luồng dữ liệu khi tích hợp với hệ thống khác	15
2.5.3. Luồng dữ liệu trong replica set (sao chép dữ liệu).....	15
2.6. Nhận xét MongoDB.....	15
2.6.1. Ưu điểm	15

2.6.2. Hạn chế	16
3. CÀI ĐẶT	17
3.1.1. Cài đặt MongoDB trên hai máy.....	17
3.1.2. Chạy MongoDB Server (mongod)	21
3.1.3. Cài đặt MongoShell.....	22
3.1.4. Kết nối giữa 2 máy	26
3.1.5. Tạo database	32
4. THAO TÁC DỮ LIỆU QUA LẠI GIỮA HAI MÁY	33
4.1. Cấu hình.....	33
4.2. Thêm dữ liệu.....	42
4.3. Xoá, sửa trong MongoDB	44
4.4. Truy vấn trong MongoDB	44
4.4.1. Câu 1: Tìm khách hàng có nhiều hóa đơn nhất.....	44
4.4.2. Câu 2: Tìm sản phẩm bán chạy nhất	45
4.4.3. Câu 3: Tìm sản phẩm tồn kho ở chi nhánh 1 nhưng không có ở chi nhánh 2.	45
4.4.4. Câu 4: Danh sách khách hàng đã mua hàng ở cả 2 chi nhánh	46
4.4.5. Câu 5: Tìm những khách hàng từng mua sản phẩm Vợt Lining Turbo Charging	47
4.4.6. Câu 6: Thống kê số lượng hóa đơn của từng nhân viên.....	47
4.4.7. Câu 7: Thống kê tồn kho từng sản phẩm ở mỗi chi nhánh.....	48
4.4.8. Câu 8: Top 3 sản phẩm giá cao nhất của mỗi chi nhánh	49
4.4.9. Câu 9: Tìm hóa đơn chưa thanh toán của chi nhánh 2	50
4.4.10. Câu 10: Tìm khách hàng đã mua toàn bộ sản phẩm.....	51
5. PHÂN CÔNG CÔNG VIỆC	53
6. TÀI LIỆU THAM KHẢO	54
Bảng 1. Phân công công việc	53

1. NOSQL Database Management Systems

1.1. Giới thiệu chung

1.1.1. Định nghĩa

1.1.2. Lịch sử và nguồn gốc của NoSQL

NoSQL là viết tắt của "**Not Only SQL**", thể hiện rằng hệ cơ sở dữ liệu này không chỉ sử dụng SQL như các hệ truyền thống (RDBMS), mà còn hỗ trợ nhiều mô hình dữ liệu linh hoạt hơn.

- **1998:** Carlo Strozzi đưa ra hệ thống “NoSQL” đầu tiên là Strozzi NoSQL (không dùng SQL) nhưng vẫn là hệ thống quan hệ.
- **2009:** Thuật ngữ “NoSQL” được tái định nghĩa bởi **Eric Evans** tại một hội thảo do **Johan Oskarsson** tổ chức (Last.fm), nhấn mạnh đến:
 - + Phi quan hệ (non-relational),
 - + Phân tán (distributed),
 - + Quy mô lớn (scalable),
 - + Hiệu suất cao (high performance).

NoSQL xuất hiện như một giải pháp thay thế cho RDBMS trong bối cảnh “Big Data” và nhu cầu xử lý dữ liệu theo thời gian thực ở các công ty lớn như **Google, Amazon, Facebook...**

1.2. Các loại NoSQL

NoSQL không có một kiến trúc duy nhất. Dựa trên cách tổ chức dữ liệu, NoSQL được chia làm **4 loại chính:**

1.2.1. Document-Based (Dựa trên tài liệu)

- **Đặc điểm**
 - + Lưu trữ tài liệu dưới dạng document (JSON/BSON/XML) với nested data. Mỗi tài liệu có cấu trúc độc lập và linh hoạt.
 - + Hỗ trợ index trên nhiều trường, truy vấn phức tạp.
 - + Schema linh hoạt, cho phép thay đổi cấu trúc document.
- **Ví dụ:** MongoDB, CouchDB.

- **Ưu điểm:**
 - + Schema linh hoạt, dễ thay đổi.
 - + Hỗ trợ lồng nhau, truy vấn mạnh mẽ.
 - + Rất phù hợp với mô hình dữ liệu hướng đối tượng.
- **Ứng dụng:** Quản lý hồ sơ người dùng với attributes động, giỏ hàng, CMS, app thời gian thực.

1.2.2. Key-Value Store (Khoá – Giá trị)

- **Đặc điểm:**
 - + Lưu trữ dạng key-value đơn giản, key là định danh duy nhất.
 - + Không có schema định trước, giá trị có thể là string, JSON, binary...
 - + Tối ưu cho truy xuất nhanh với độ phức tạp O(1).
- **Ví dụ:** Redis, Riak, DynamoDB.
- **Ưu điểm:**
 - + Tốc độ truy xuất cực nhanh.
 - + Cực kỳ đơn giản, lý tưởng cho cache, session store.
- **Ứng dụng:** Cache hệ thống, quản lý session người dùng, hệ thống real-time analytics.

1.2.3. Column-Family (Dựa trên cột)

- **Đặc điểm:**
 - + Tổ chức dữ liệu theo column families thay vì rows.
 - + Tối ưu cho truy vấn theo cột, xử lý lượng lớn dữ liệu.
 - + Phân tán dữ liệu qua sharding tự động.
- **Ví dụ:** Cassandra, HBase.
- **Ưu điểm:**
 - o Hiệu quả khi truy vấn dữ liệu lớn theo chiều dọc.
 - o Tối ưu cho xử lý phân tích, tổng hợp dữ liệu (analytics).
- **Ứng dụng:** phân tích Big Data, Time-series data, hệ thống recommendation.

1.2.4. Graph-Based (Dựa trên đồ thị)

- **Mô hình**
 - + Biểu diễn dữ liệu dưới dạng nodes và edges.
 - + Tối ưu cho truy vấn quan hệ phức tạp (đồ thị).
 - + Sử dụng thuật toán traversal để phân tích mối quan hệ.
- **Ví dụ:** Neo4j, ArangoDB.

– Ưu điểm

- + Mạnh trong biểu diễn mối quan hệ phức tạp.
- + Truy vấn theo quan hệ nhanh và hiệu quả.
- **Ứng dụng:** Mạng xã hội, hệ thống recommendation, phát hiện gian lận.

1.3. Ưu điểm

- **Hiệu quả về chi phí (Cost-effectiveness)**
 - + NoSQL hỗ trợ mở rộng theo chiều ngang (horizontal scaling) dễ dàng – giúp phân phối tài nguyên linh hoạt, giảm chi phí và tăng hiệu quả.
- **Linh hoạt về mô hình dữ liệu (Flexibility)**
 - + Không yêu cầu lược đồ cố định, dễ dàng thay đổi cấu trúc dữ liệu khi ứng dụng phát triển.
 - + Hỗ trợ dữ liệu thay đổi nhanh hoặc có cấu trúc đa dạng – lý tưởng cho các mô hình phát triển linh hoạt (agile), triển khai nhanh (CI/CD), và cải tiến liên tục.
- **Nhân bản và sẵn sàng cao (Replication)**
 - + Hầu hết NoSQL hỗ trợ tự động nhân bản dữ liệu trên nhiều máy chủ (replication). Điều này đảm bảo:
 - Dữ liệu không bị mất khi một máy chủ gặp sự cố.
 - Truy cập liên tục, ngay cả khi có gián đoạn hoặc bảo trì hệ thống.
- **Tốc độ cao (Speed)**
 - + NoSQL cho phép lưu trữ và xử lý dữ liệu nhanh chóng hơn, mang lại lợi ích cho:

- Lập trình viên: truy vấn dễ, phát triển nhanh
 - Doanh nghiệp: phản hồi khách hàng tốt hơn
 - Khách hàng: trải nghiệm mượt mà hơn
- + Đặc biệt phù hợp với ứng dụng web hiện đại, thương mại điện tử, ứng dụng di động có yêu cầu hiệu năng cao.

2. MONGODB

2.1. Giới thiệu

2.1.1. Định nghĩa



MongoDB lưu trữ dữ liệu dưới dạng các tài liệu linh hoạt giống như JSON. Điều này có nghĩa là mỗi tài liệu có thể có các trường (fields) khác nhau, và có thể thay đổi cấu trúc dữ liệu bất cứ lúc nào.

Mô hình tài liệu của MongoDB giống với cách làm việc với đối tượng trong ngôn ngữ lập trình, nên việc xử lý dữ liệu trở nên tự nhiên và dễ dàng hơn.

MongoDB hỗ trợ các truy vấn linh hoạt, tạo chỉ mục và tổng hợp dữ liệu theo thời gian thực, giúp tìm kiếm và phân tích dữ liệu nhanh chóng và hiệu quả.

Vì là một cơ sở dữ liệu phân tán, MongoDB có sẵn các tính năng như: độ sẵn sàng cao, mở rộng quy mô dễ dàng (thêm máy chủ), và có thể triển khai ở nhiều vị trí địa lý khác nhau một cách thuận tiện.

2.1.2. Nguồn gốc ra đời

MongoDB được tạo ra bởi **Dwight Merriman** và **Eliot Horowitz**, hai kỹ sư từng làm việc tại **DoubleClick** – một công ty quảng cáo trực tuyến (sau này được Google mua lại). Trong quá trình phát triển các ứng dụng web tại DoubleClick, họ gặp phải nhiều vấn đề về hiệu năng và khả năng mở rộng với cơ sở dữ liệu quan hệ truyền thống (RDBMS).

Để giải quyết các vấn đề này, họ quyết định xây dựng một hệ cơ sở dữ liệu mới – và đó chính là MongoDB.

Tên gọi “MongoDB” bắt nguồn từ từ "humongous" (nghĩa là “khổng lồ”), thể hiện mục tiêu ban đầu là hỗ trợ lưu trữ lượng dữ liệu rất lớn.

- **Thành lập công ty và thương mại hóa**

- + Năm **2007**, Merriman và Horowitz cùng các cộng sự thành lập công ty 10Gen Inc. để phát triển và thương mại hóa MongoDB.
- + Đến năm **2013**, công ty đổi tên thành MongoDB Inc.
- + MongoDB chính thức niêm yết trên sàn chứng khoán NASDAQ vào tháng 10 năm **2017**, với mã cổ phiếu là MDB.

- **Phát hành mã nguồn mở**

- + MongoDB được phát hành dưới dạng mã nguồn mở từ năm 2009.
- + Từ đó đến nay, MongoDB luôn được cập nhật, nâng cấp và hỗ trợ cộng đồng phát triển mạnh mẽ.

2.1.3. Tại sao nên sử dụng MongoDB?

MongoDB được xây dựng dựa trên kiến trúc **mở rộng theo chiều ngang (scale-out)** – một xu hướng ngày càng phổ biến trong giới lập trình viên nhằm phát triển các ứng dụng có khả năng mở rộng cao và thích ứng linh hoạt với các schema dữ liệu thay đổi liên tục.

Là một cơ sở dữ liệu hướng tài liệu (**document database**), MongoDB giúp lập trình viên **dễ dàng lưu trữ cá dữ liệu có cấu trúc lẫn không có cấu trúc**. Nó sử dụng định dạng tương tự JSON để lưu trữ các tài liệu (documents). Định dạng này ánh xạ trực tiếp với các đối tượng trong hầu hết các ngôn ngữ lập trình hiện đại, nhờ đó lập trình viên không cần quá lo lắng về việc chuẩn hóa dữ liệu như trong các hệ thống SQL truyền thống.

MongoDB cũng có khả năng xử lý **khối lượng dữ liệu lớn và mở rộng cả theo chiều dọc (vertical) lẫn chiều ngang (horizontal)** để đáp ứng nhu cầu tăng trưởng dữ liệu mạnh mẽ.

MongoDB được thiết kế dành cho những người xây dựng các ứng dụng internet và ứng dụng doanh nghiệp cần **phát triển nhanh và mở rộng hệ thống một cách linh hoạt**. Hiện nay, các công ty và đội ngũ phát triển ở mọi quy mô đều đang sử dụng MongoDB.

2.1.4. Khi nào nên sử dụng MongoDB?

- **Khi cần tích hợp khối lượng lớn dữ liệu đa dạng**
 - + Nếu hệ thống của bạn cần kết hợp dữ liệu từ nhiều nguồn khác nhau (từ hàng chục đến hàng trăm hệ thống), MongoDB với mô hình dữ liệu linh hoạt sẽ giúp tạo ra một cái nhìn tổng thể, điều mà các hệ cơ sở dữ liệu truyền thống thường khó đáp ứng.
- **Khi cần mô tả dữ liệu phức tạp và dễ thay đổi**
 - + Trong các hệ thống có dữ liệu phân cấp hoặc mối quan hệ phức tạp, MongoDB cho phép lồng các tài liệu, hỗ trợ thay đổi cấu trúc dữ liệu dễ dàng mà không cần chỉnh sửa lại toàn bộ schema. Ngoài ra, MongoDB hỗ trợ tốt các loại dữ liệu chuyên biệt như dữ liệu địa lý (geospatial).
- **Khi ứng dụng yêu cầu hiệu năng cao và quy mô lớn**
 - + Với khả năng mở rộng theo chiều ngang, MongoDB có thể xử lý lượng lớn giao dịch và dữ liệu ở quy mô cực lớn. Đây là lựa chọn phù hợp cho các hệ thống cần hiệu suất cao mà vẫn muốn triển khai đơn giản.
- **Khi cần triển khai linh hoạt (local, datacenter, multi-cloud)**
 - + MongoDB có thể chạy trên máy cá nhân, cụm máy chủ tại trung tâm dữ liệu hoặc trên các nền tảng đám mây như MongoDB Atlas. Điều này giúp hệ thống dễ dàng thích ứng với các yêu cầu triển khai hiện tại và tương lai.
- **Khi cần hỗ trợ phát triển linh hoạt và cộng tác nhóm hiệu quả**
 - + Với mô hình tài liệu, MongoDB cho phép lập trình viên thao tác dữ liệu giống như làm việc với mã nguồn. Bên cạnh đó, MongoDB hỗ trợ cộng tác hiệu quả khi nhiều nhóm có thể phụ trách các phần khác nhau của cùng một tài liệu, phục vụ mô hình phát triển linh hoạt và phân tán.

2.1.5. RDBMS vs MongoDB

Hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) có một thiết kế lược đồ (schema) điển hình, thể hiện số lượng bảng và mối quan hệ giữa các bảng này. Trong khi đó, MongoDB là cơ sở dữ liệu hướng tài liệu (document-oriented). Không có khái niệm lược đồ hay quan hệ giữa các bảng như trong RDBMS.

MongoDB không hỗ trợ các giao dịch phức tạp vì không có các phép join phức tạp như trong RDBMS.

MongoDB cho phép một cấu trúc tài liệu rất linh hoạt và có khả năng mở rộng cao. Ví dụ, một tài liệu trong cùng một collection có thể có hai trường, trong khi tài liệu khác có thể có bốn trường, mà vẫn hoàn toàn hợp lệ.

MongoDB nhanh hơn so với RDBMS nhờ vào các kỹ thuật lưu trữ và đánh chỉ mục hiệu quả.

2.1.6. Ứng dụng thực tế

MongoDB đã được sử dụng bởi nhiều tổ chức lớn và đa dạng ngành nghề:

Tổ chức/Website	Mục đích sử dụng
MetLife (bảo hiểm)	Ứng dụng dịch vụ khách hàng (customer service apps)
Craigslist	Lưu trữ và lưu trữ dữ liệu cũ (archiving data)
CERN (trung tâm vật lý hạt nhân châu Âu)	Tổng hợp và khai phá dữ liệu (data aggregation & discovery)
The New York Times	Xây dựng ứng dụng biểu mẫu gửi ảnh (form builder cho ảnh từ độc giả)

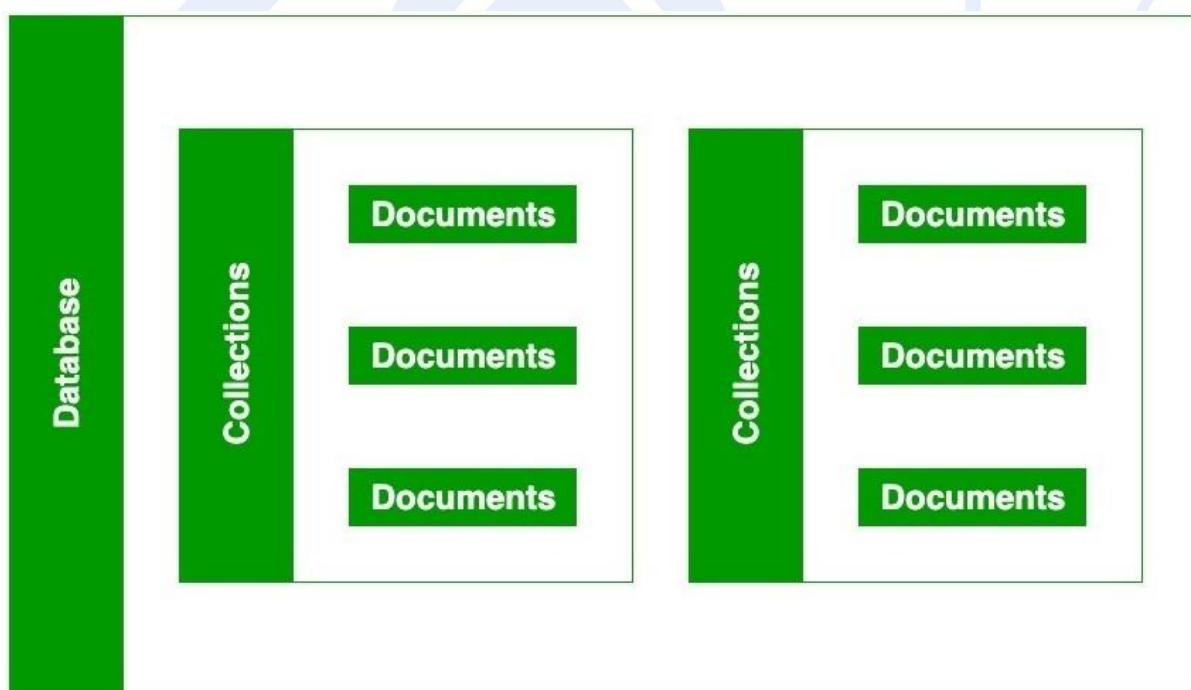
2.2. Tính năng của MongoDB

- **Ad hoc query:** Hỗ trợ truy vấn theo trường, phạm vi, biểu thức chính quy (regex), trả về toàn bộ hoặc một phần document
- **Indexing:** Tạo nhiều loại chỉ mục như: đơn trường, nhiều trường (compound), mảng (multikey), văn bản, không gian địa lý, hash
- **Replication:** Hỗ trợ replica set – có nhiều bản sao dữ liệu; nếu server chính bị lỗi, tự động chuyển đổi sang server phụ
- **Document-oriented:** Lưu trữ dưới dạng key-value trong BSON giúp linh hoạt hơn bảng RDBMS
- **Scalability:** Hỗ trợ sharding – phân tán dữ liệu qua nhiều server giúp xử lý dữ liệu lớn hiệu quả

- **Aggregation:** Tương tự GROUP BY trong SQL giúp xử lý và tính toán dữ liệu tập hợp
- **Load balancing:** Cân bằng tải đọc/ghi nhờ replica set và sharded cluster
- **Batch processing:** Hỗ trợ xử lý hàng loạt qua các phương pháp như ghi bulk – giúp giảm chi phí mạng và tăng tốc độ

2.3. Kiến trúc hệ thống

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
Column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Khóa mặc định _id do chính MongoDB cung cấp)



Hình 2-1. Cấu trúc dữ liệu của MongoDB

- **_id**
 - + **_id** là trường đặc biệt và bắt buộc phải có trong mỗi document của MongoDB, dùng để định danh duy nhất cho từng document trong một collection.

- + Vai trò của `_id` tương tự như khóa chính (primary key) trong các hệ quản trị cơ sở dữ liệu quan hệ.
 - + Giá trị `_id` mặc định là một chuỗi số thập lục phân gồm 12 byte, đảm bảo không trùng lặp giữa các document. Cấu trúc của 12 byte này gồm:
 - 4 byte đầu: thời gian tạo (dựa theo Unix Epoch).
 - 3 byte tiếp theo: mã định danh của máy tính tạo ra document.
 - 2 byte tiếp theo: ID của tiến trình MongoDB trên máy chủ.
 - 3 byte cuối: bộ đếm tăng dần để đảm bảo tính duy nhất.
 - + Khi tạo document mới, nếu không chỉ định `_id`, MongoDB sẽ tự động sinh ra giá trị này.
- **Document**
- + **Document** là đơn vị lưu trữ dữ liệu nhỏ nhất trong MongoDB, tương ứng với một bản ghi (row) trong hệ cơ sở dữ liệu quan hệ.
 - + Dữ liệu trong document được tổ chức dưới dạng các cặp key-value (field-value), linh hoạt về cấu trúc và kiểu dữ liệu.
 - + Các document trong cùng một collection không bắt buộc phải có cùng cấu trúc hoặc cùng số lượng trường. Cùng một trường ở các document khác nhau có thể chứa các kiểu dữ liệu khác nhau.
- **Collection**
- + **Collection** là tập hợp các document, tương tự như một bảng (table) trong cơ sở dữ liệu quan hệ. Collection không có cấu trúc schema cố định, cho phép các document bên trong có cấu trúc khác nhau.
 - + Thông thường, một collection sẽ chứa các document có cùng mục đích hoặc liên quan đến nhau về mặt dữ liệu.
- **Database**
- + **Database** trong MongoDB là một vùng lưu trữ vật lý, chứa nhiều collection. Một database có thể không chứa collection nào hoặc có rất nhiều collection.
 - + Một máy chủ MongoDB có thể quản lý nhiều database cùng lúc, và số lượng database chỉ bị giới hạn bởi tài nguyên hệ thống (bộ nhớ, dung lượng lưu trữ).

2.4. Lưu trữ dữ liệu

- **Về định dạng lưu trữ:** MongoDB sử dụng định dạng BSON (Binary JSON) – là phiên bản mở rộng của JSON, hỗ trợ kiểu dữ liệu nâng cao như: dữ liệu nhị phân, ngày tháng và mảng lồng nhau (nested array) giúp cải thiện hiệu suất truy vấn và lưu trữ.
- **Về cách lưu trữ dữ liệu và truy vấn:**
 - + Dữ liệu được lưu trong các **document** bên trong **collection**, giúp dữ liệu có tính linh hoạt cao.
 - + MongoDB hỗ trợ các cơ chế chỉ mục mạnh mẽ và truy vấn thời gian thực, đảm bảo tốc độ đọc/ghi nhanh hơn.
 - + Sử dụng **MongoDB Query Language (MQL)** tương tự cú pháp JSON để truy vấn dữ liệu.

2.5. Data Flow

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL dạng document, nơi dữ liệu được lưu trữ dưới dạng các document BSON (Binary JSON) trong các collection.

2.5.1. Luồng dữ liệu cơ bản trong MongoDB

- **Nhập dữ liệu (Input):** Dữ liệu được gửi từ các ứng dụng hoặc hệ thống bên ngoài thông qua driver (thư viện kết nối) của MongoDB. Dữ liệu này thường ở định dạng JSON hoặc BSON.
- **Xử lý dữ liệu:** MongoDB thực hiện các thao tác như insert, update, delete hoặc query trên các collection. Các thao tác này được tối ưu hóa nhờ hệ thống index, giúp truy vấn nhanh chóng mà không cần quét toàn bộ dữ liệu.
- **Lưu trữ dữ liệu:** Dữ liệu sau khi được xử lý sẽ được lưu trữ trong các collection thuộc database của MongoDB. Mỗi document có một trường `_id` duy nhất để định danh.
- **Truy xuất dữ liệu (Output):** Khi có yêu cầu truy vấn, MongoDB trả về dữ liệu cho ứng dụng hoặc người dùng cuối, cũng ở dạng JSON/BSON.

2.5.2. Luồng dữ liệu khi tích hợp với hệ thống khác

MongoDB thường được sử dụng làm nguồn hoặc đích dữ liệu trong các pipeline ETL (Extract, Transform, Load) hoặc các hệ thống xử lý dữ liệu lớn như Hadoop, BigQuery, Dataflow,...

Ví dụ về luồng dữ liệu từ MongoDB sang Hadoop bằng Apache NiFi:

- **GetMongo Processor:** Kết nối tới MongoDB, thực hiện truy vấn (theo lịch) để lấy dữ liệu từ collection dựa trên điều kiện (ví dụ theo ngày).
- **PutHDFS Processor:** Dữ liệu lấy từ MongoDB được đẩy lên hệ thống lưu trữ phân tán HDFS để phục vụ các tác vụ phân tích dữ liệu lớn.

Tương tự, với Dataflow của Google Cloud, có thể sử dụng template để trích xuất dữ liệu từ MongoDB và ghi sang BigQuery, với các tùy chọn như lưu toàn bộ document ở dạng JSON hoặc tách các trường thành cột riêng biệt.

2.5.3. Luồng dữ liệu trong replica set (sao chép dữ liệu)

Trong môi trường production, MongoDB thường triển khai theo mô hình replica set để đảm bảo tính sẵn sàng và an toàn dữ liệu:

- **Primary Node:** Nhận các thao tác ghi (write).
- **Secondary Node:** Sao chép dữ liệu từ primary qua cơ chế replication.
- **Flow Control:** Để tránh độ trễ lớn khi đồng bộ, MongoDB sử dụng cơ chế flow control, giới hạn tốc độ ghi dựa vào độ trễ giữa các node, đảm bảo dữ liệu được đồng bộ một cách nhất quán và ổn định.

2.6. Nhận xét MongoDB

2.6.1. Ưu điểm

- **Dễ sử dụng cho lập trình viên**
 - + Lập trình viên có thể tải xuống nhanh chóng hoặc sử dụng MongoDB trực tuyến (qua MongoDB Atlas).
 - + Việc làm việc với document (JSON) dễ hiểu và đơn giản hơn so với việc thiết kế bảng, khóa chính – khóa ngoại như trong RDBMS.
- **Hiệu quả**

- + Dữ liệu được lưu dưới dạng JSON nên nhẹ, dễ đọc và thuận tiện trong xử lý.
- + MongoDB dùng định dạng nhị phân BSON, giúp lưu trữ nhỏ gọn hơn và phân tích nhanh hơn so với văn bản thuần (text).
- **Schema linh hoạt**
 - + Mỗi document có thể có cấu trúc khác nhau, không cần schema cố định → dễ mở rộng và thích ứng với thay đổi của ứng dụng.
- **Ngôn ngữ truy vấn đơn giản**
 - + **MongoDB Query Language (MQL)** có cú pháp đơn giản, dễ tiếp cận, hỗ trợ truy vấn phức tạp và sử dụng chỉ mục để tăng tốc.
- **Hỗ trợ cloud tốt (Cloud-native)**
 - + **MongoDB Atlas** được phát triển cho môi trường cloud, cập nhật thường xuyên, dễ tích hợp công nghệ mới.

2.6.2. Hạn chế

- **Hạn chế về giao dịch (Transactions)**
 - + Hỗ trợ giao dịch nhiều tài liệu (multi-document transactions) chưa mạnh bằng cơ sở dữ liệu quan hệ.
 - + Giao dịch phức tạp hoặc kéo dài qua nhiều thao tác có thể khó cài đặt và không tối ưu.
- **Tính nhất quán dữ liệu chưa mạnh**
 - + MongoDB dùng mô hình “eventual consistency” cho replica set → có thể dẫn đến việc người dùng khác nhau thấy dữ liệu không đồng nhất tại cùng thời điểm.
 - + Với ứng dụng yêu cầu tính nhất quán cao (strong consistency), đây là vấn đề lớn.
- **Không hỗ trợ JOIN như SQL**
 - + MongoDB không có JOIN tự nhiên như SQL. Tuy có \$lookup để mô phỏng, nhưng hiệu năng thường kém hơn và câu truy vấn trở nên phức tạp, đặc biệt với mối quan hệ nhiều tầng.

- **Sử dụng nhiều bộ nhớ (RAM)**
 - + MongoDB lưu dữ liệu và chỉ mục thường dùng lên RAM → hiệu năng phụ thuộc vào RAM.
 - + Dễ dẫn đến việc tốn tài nguyên máy hơn các hệ quản trị khác.
- **Tốn dung lượng lưu trữ**
 - + Vì mỗi document tự chứa dữ liệu và metadata, nên dễ gây trùng lặp dữ liệu, phân mảnh, từ đó tăng dung lượng lưu trữ và chi phí.
 - + Không tối ưu bằng bảng quan hệ đã chuẩn hóa trong SQL.
- **Hạn chế về chỉ mục**
 - + MongoDB hỗ trợ nhiều loại chỉ mục, nhưng nếu tạo quá nhiều index, hiệu năng ghi (write) sẽ giảm mạnh.
 - + Mỗi lần ghi dữ liệu phải cập nhật nhiều index, khiến truy vấn nhanh nhưng ghi chậm.
- **Chi phí triển khai**
 - + Nếu dùng MongoDB cho các hệ thống high availability + horizontal scaling, chi phí phần cứng/cloud có thể rất cao.
 - + Cần nhiều RAM, ổ cứng và tài nguyên cho các bản sao (replica), làm tăng chi phí tổng thể.

3. CÀI ĐẶT

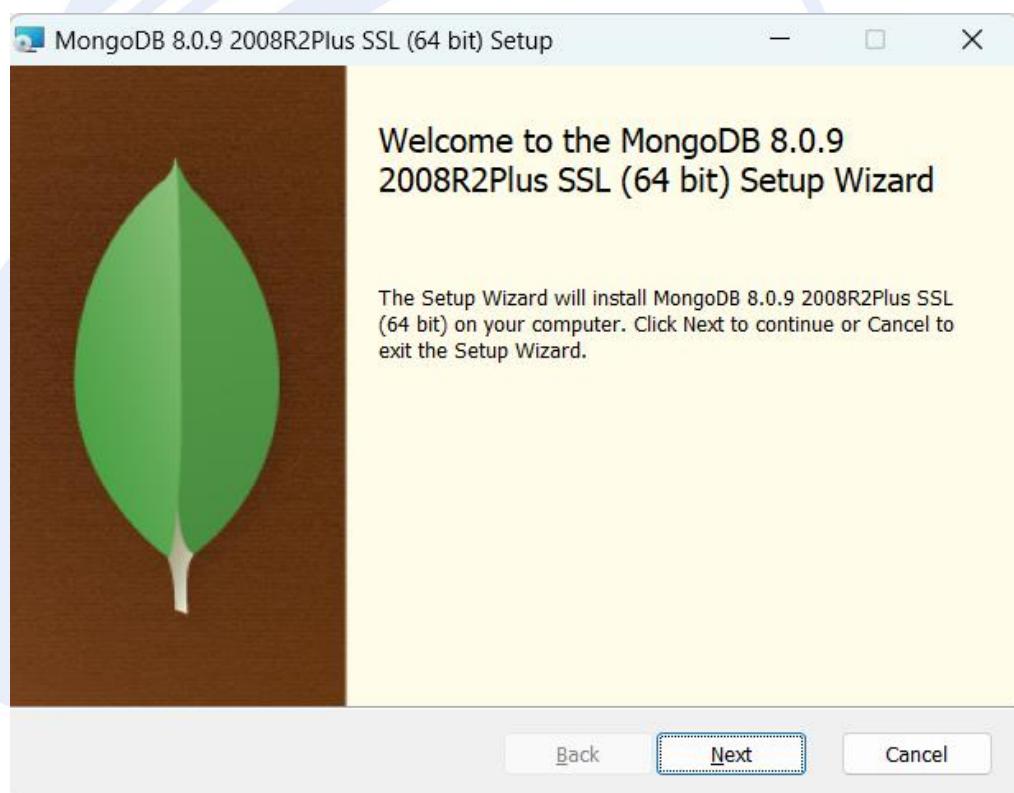
3.1.1. Cài đặt MongoDB trên hai máy

Bước 1: Download MongoDB Community Server về máy theo đường link:
<https://www.mongodb.com/try/download/community>.

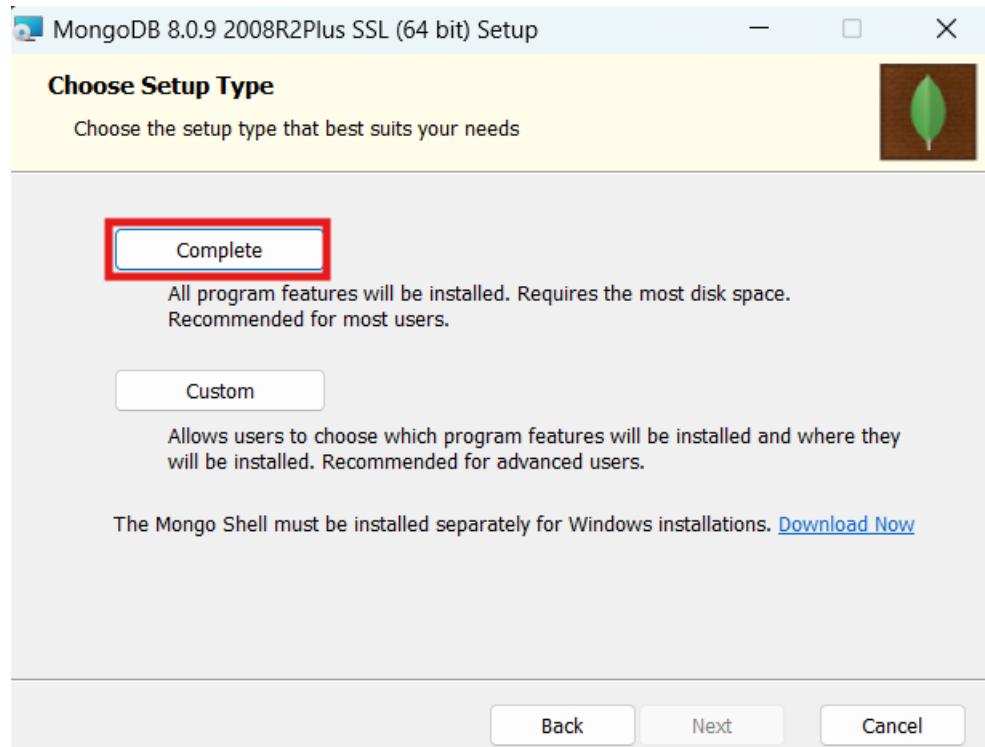
The screenshot shows the MongoDB website's product selection interface. On the left, there's a sidebar with options like MongoDB Atlas, MongoDB Enterprise Advanced, MongoDB Community Edition, MongoDB Community Server (which is selected and highlighted in green), Tools, and Atlas SQL Interface. The main area displays a command-line snippet for Mac users: '\$ brew install mongodb-atlas' and '\$ atlas setup'. Below this, configuration fields show 'Version: 8.0.9 (current)', 'Platform: Windows x64', and 'Package: msi'. At the bottom right of this section are 'Download' (highlighted with a red box), 'Copy link', and 'More Options' buttons.

Bước 2: Cài đặt MongoDB

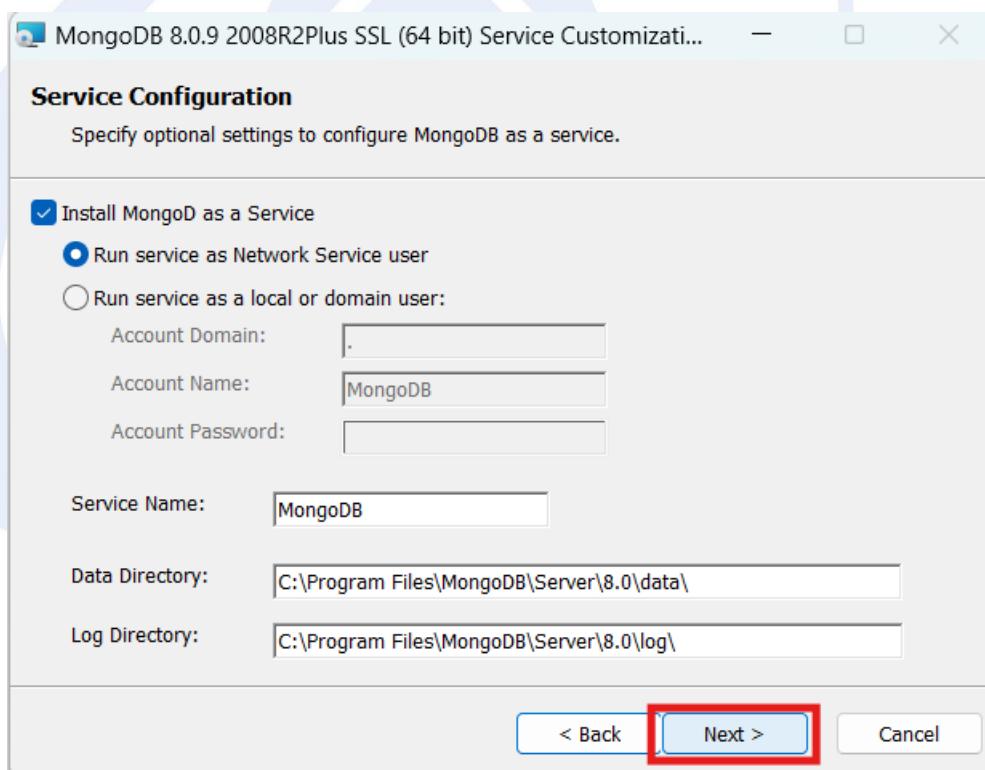
- Mở tệp .msi vừa tải về, sẽ hiện lên như ảnh dưới, nhấn **Next**.



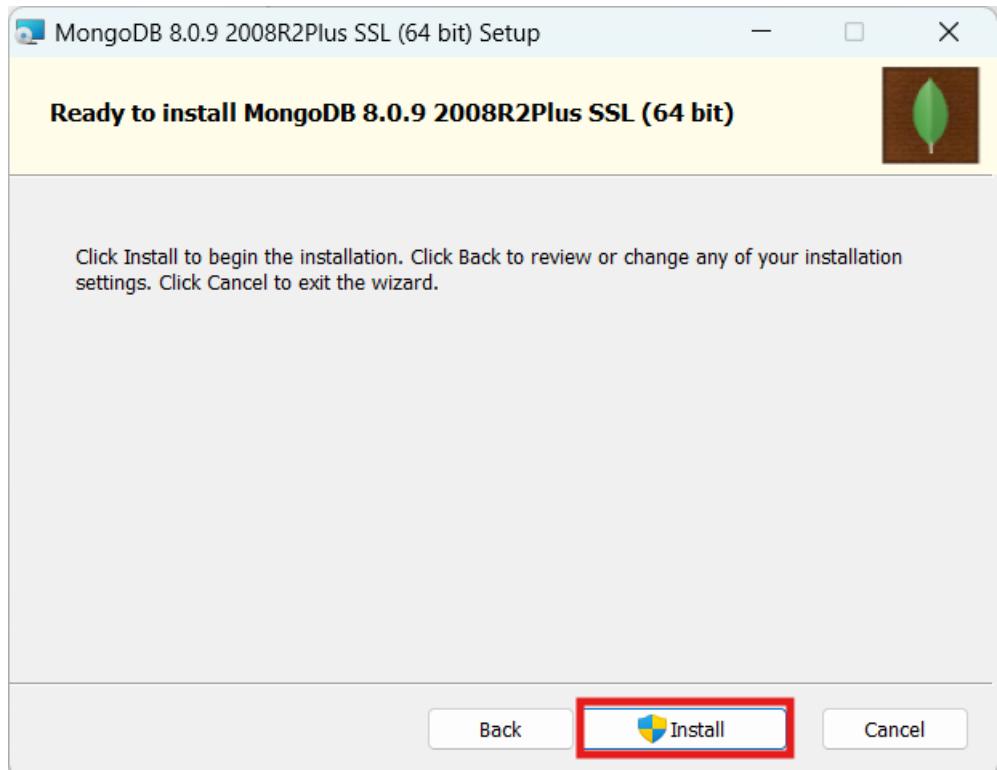
- Sau khi chấp nhận End-User License Agreement → **Next** → chọn **Complete**.



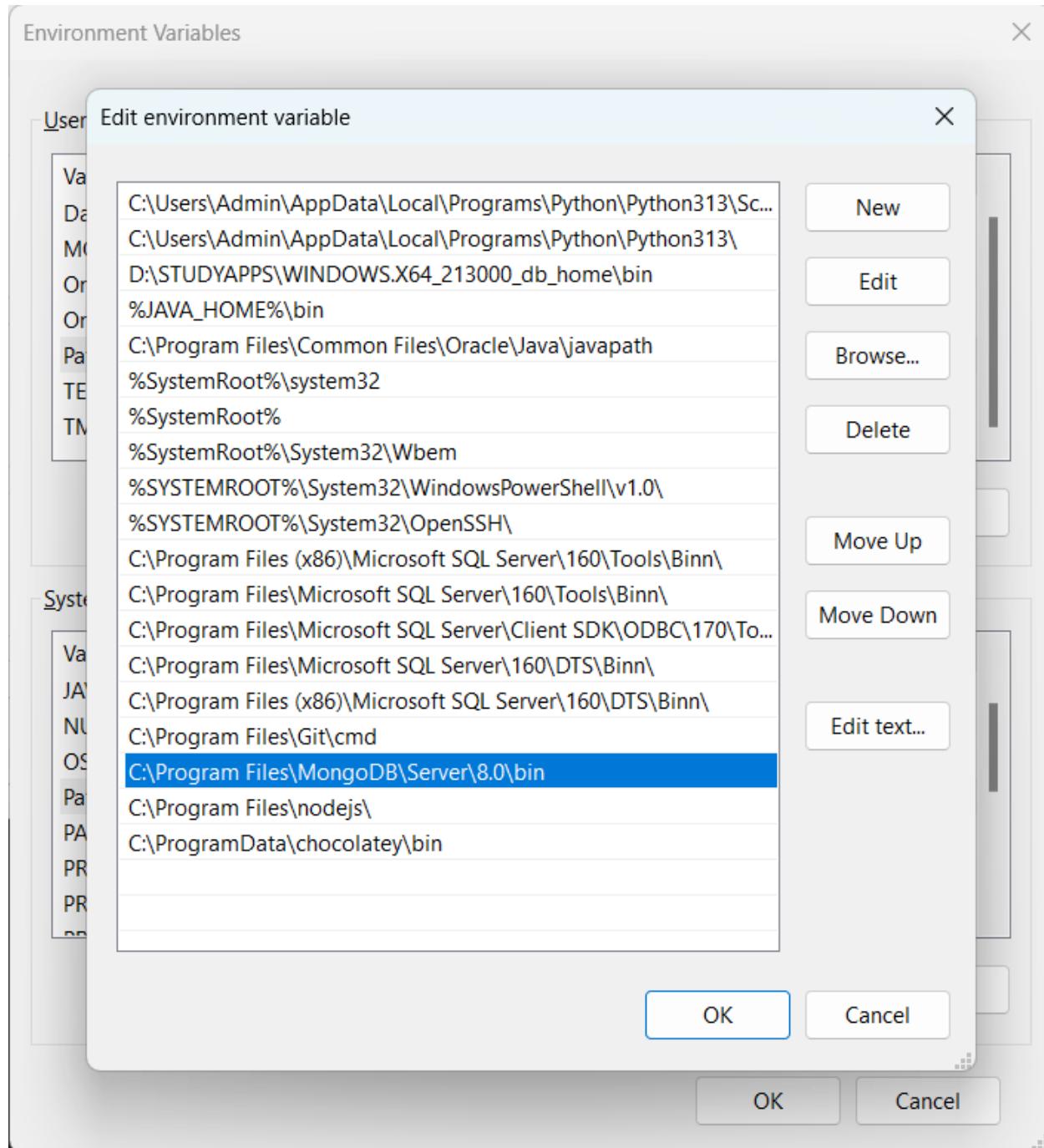
- + Tiếp theo màn hình sẽ hiện giao diện cấu hình MongoDB service như bên dưới, nhấn **Next**.



- + Nhấn **Install** để thực hiện cài đặt MongoDB.



- + Thiết lập biến môi trường: **Environment Variables** → ở System Variables chọn **Edit Path** → Thêm đường dẫn đến thư mục bin MongoDB.



3.1.2. Chạy MongoDB Server (mongod)

- Mở Command Prompt và chạy câu lệnh: **mongod --version** để kiểm tra xem đã cài được MongoDB hay chưa.

```
C:\Users\Admin>mongod --version
db version v8.0.6
Build Info: {
    "version": "8.0.6",
    "gitVersion": "80f21521ad4a3dfd5613f5d649d7058c6d46277f",
    "modules": [],
    "allocator": "tcmalloc-gperf",
    "environment": {
        "distmod": "windows",
        "distarch": "x86_64",
        "target_arch": "x86_64"
    }
}

C:\Users\Admin>
```

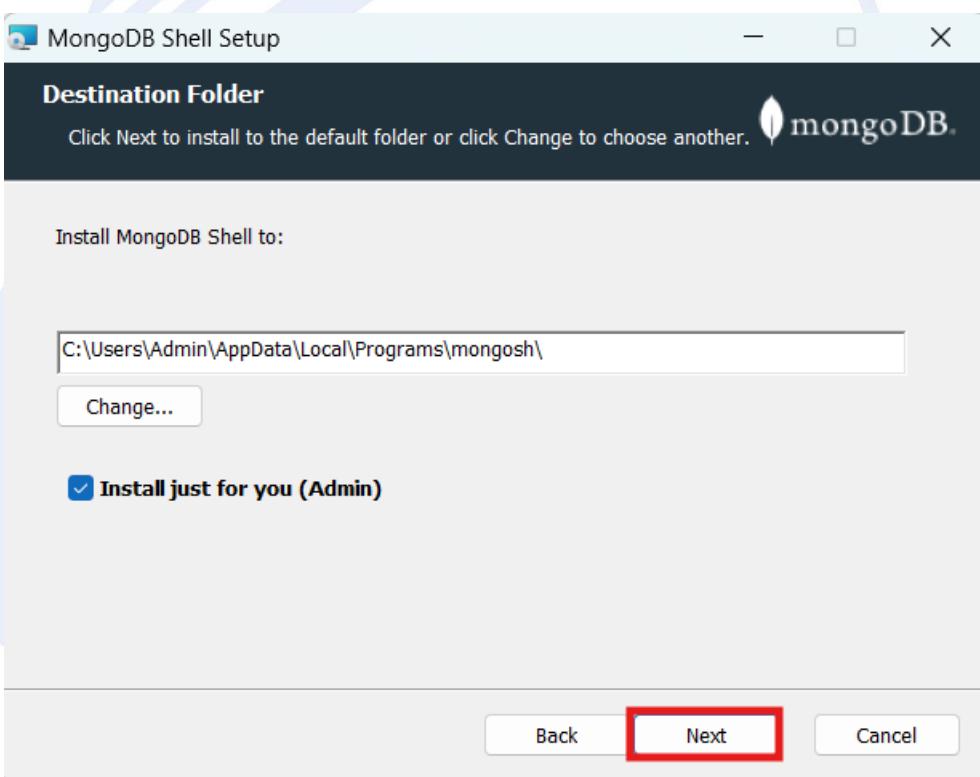
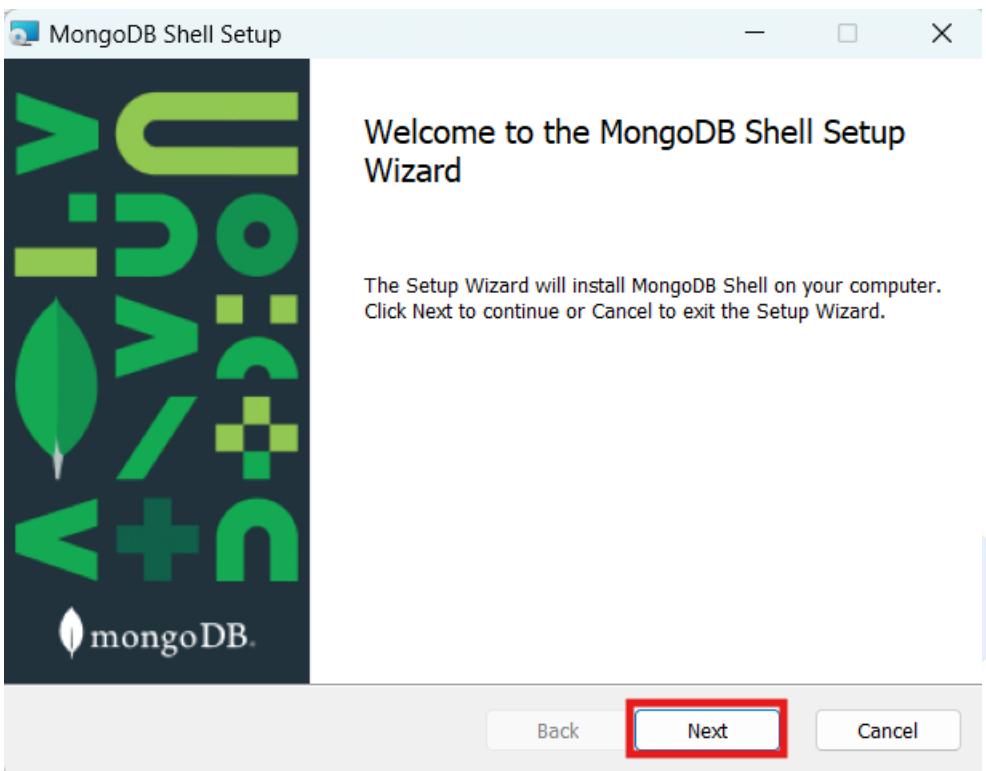
3.1.3. Cài đặt MongoShell

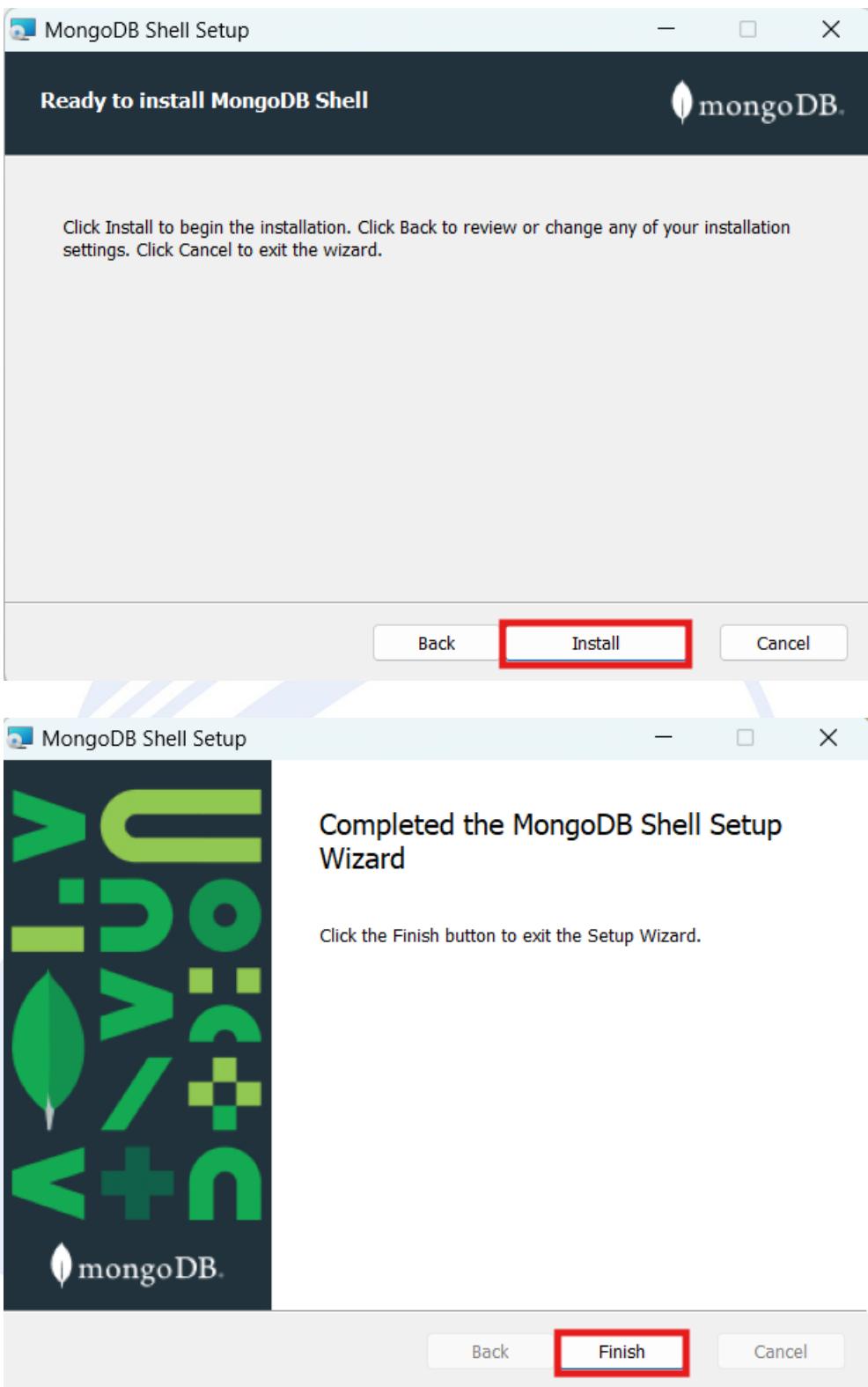
Bước 1: Download MongoDB Shell về máy theo đường link:

<https://www.mongodb.com/try/download/shell>.

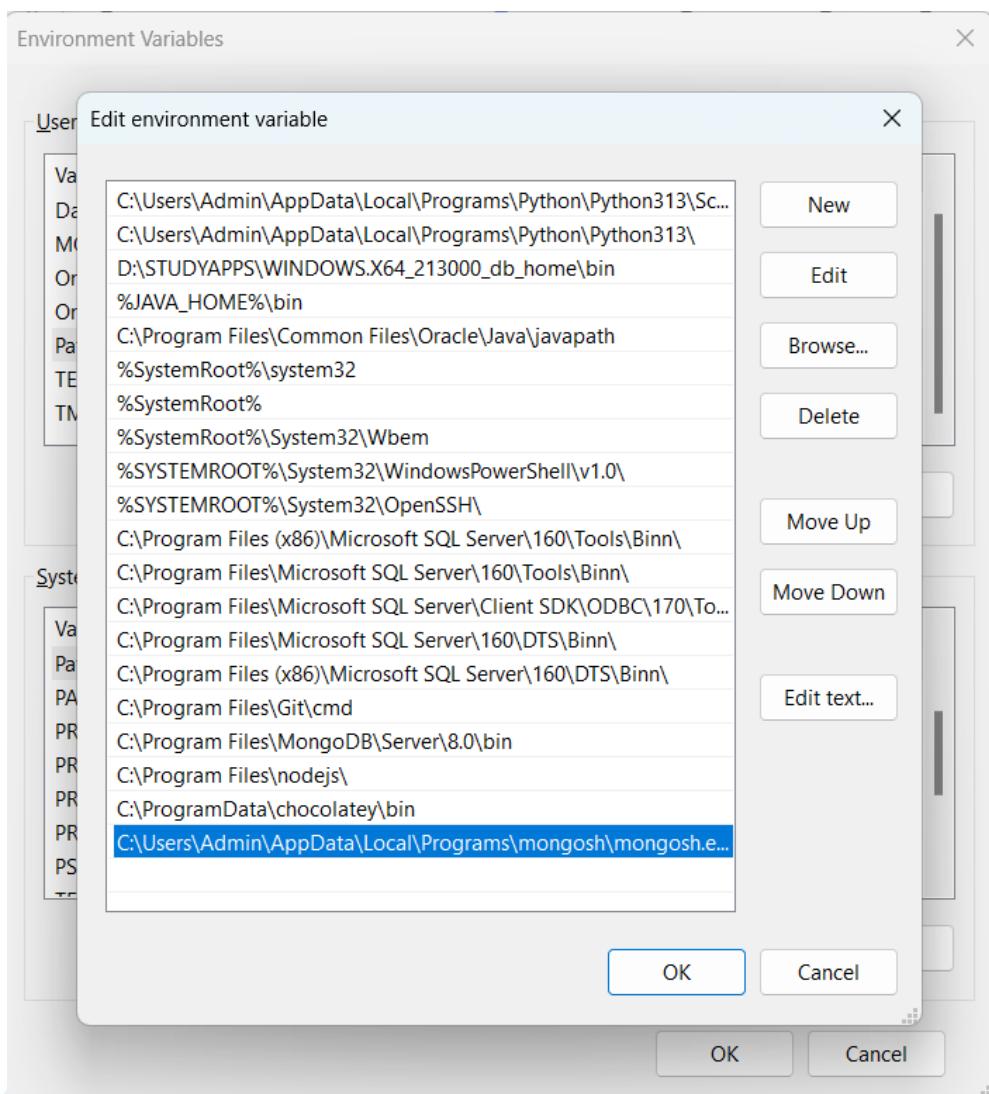
The screenshot shows the MongoDB download page. On the left, there's a sidebar with links like MongoDB Atlas, MongoDB Enterprise Advanced, MongoDB Community Edition, Tools (MongoDB Shell, MongoDB Compass, Atlas CLI, etc.), and others. The main area shows the MongoDB Shell download details. It includes a note that MongoDB Shell is an open source product developed separately from the MongoDB Server. It allows selecting Version (2.5.2), Platform (Windows x64 (10+)), and Package (zip). At the bottom, there are 'Download' (highlighted with a red box), 'Copy link', and 'More Options' buttons.

Bước 2: Mở file vừa tải và tiến hành cài đặt.





Bước 3: Thiết lập biến môi trường: ***Environment Variables*** → Ở System Variables chọn ***Edit Path*** → Thêm đường dẫn đến mongosh.



Bước 4: Mở Command Prompt và chạy câu lệnh **mongosh** để kiểm tra mongosh đã cài đặt thành công hay chưa.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2
Microsoft Windows [Version 10.0.22631.5335]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>mongosh
Current Mongosh Log ID: 683f097b15c997601550eb66
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2
Using MongoDB:      8.0.6
Using Mongosh:       2.5.2

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2025-05-22T19:47:26.278+07:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
test> |
```

Như vậy là hoàn tất quá trình cài đặt MongoDB.

3.1.4. Kết nối giữa 2 máy

Bước 1: Truy cập vào <https://cloud.mongodb.com/>, đăng nhập vào thì ta được giao diện như hình dưới, nhấn **New Project** để tạo project mới.

The screenshot shows the MongoDB Cloud interface. On the left, there's a sidebar with options like Alerts, Activity Feed, Settings, Integrations, Access Manager, Resource Policies, Billing, Support, and Live Migration. The main area is titled 'Projects' and shows a table with one row: 'Project 0' (0 Clusters, + Add Tags), 'Users' (1 User), 'Teams' (0 Teams), and 'Alerts' (0 Alerts). At the top right of the main area, there's a red-bordered 'New Project' button. The bottom of the screen has a footer with system status, last login, and various links.

Bước 2: Đặt tên cho project và nhấn **Next**.

The screenshot shows the 'Create a Project' form. On the left, there's a sidebar with the same set of options as the previous screenshot. The main area is titled 'Create a Project' and has two tabs: 'Name Your Project' (which is active) and 'Add Members'. Under 'Name Your Project', it says 'Project names have to be unique within the organization (and other restrictions)' and shows a text input field containing 'Mucvanhungnguoiban' with a red border around it. Below that is a section for 'Add Tags (Optional)' with a table for adding tags. The table has columns for 'Key' and 'Value', both with dropdown menus and a trash icon. There's also a '+ Add tag' button and a '0 TAGS' count. At the bottom right, there are 'Cancel' and 'Next' buttons, with 'Next' being red and highlighted with a red box.

Bước 3: Thêm thành viên của dự án và set quyền cho họ.

Bước 4: Sau khi tạo xong project ta sẽ có giao diện như hình dưới, tiếp tục sẽ nhấn vào **Create** để tiến hành tạo Cluster.

Bước 5: Điền username và password để tạo user cho database, sau đó nhấn **Create Database User**.

The screenshot shows the 'Connect to Cluster0' wizard in progress. Step 2, 'Choose a connection method', is displayed. A large callout bubble highlights the 'Create Database User' button, which is enclosed in a red box. The 'Close' and 'Choose a connection method' buttons are also visible.

Connect to Cluster0

1 Set up connection security 2 Choose a connection method 3 Connect

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more ↗](#)

1. Add a connection IP address

Your current IP address (1.54.152.31) has been added to enable local connectivity. Only an IP address you add to your Access List will be able to connect to your project's clusters. Add more later in [Network Access ↗](#).

2. Create a database user

This first user will have [atlasAdmin ↗](#) permissions for this project.

We autogenerated a username and password. You can use this or create your own.

ⓘ You'll need your database user's credentials in the next step. Copy the database user password.

Username	Password
22520372	1 HIDE

Create Database User

Close **Choose a connection method**

Bước 6: Sau khi tạo thành công Database User thì chọn phương thức kết nối bằng cách nhấp vào **Choose a connection method**.

Connect to Cluster0

1 Set up connection security 2 Choose a connection method 3 Connect

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more ↗](#)

1. Add a connection IP address

✓ Your current IP address (1.54.152.31) has been added to enable local connectivity. Only an IP address you add to your Access List will be able to connect to your project's clusters. Add more later in [Network Access ↗](#).

2. Create a database user

✓ A database user has been added to this project. Create another user later in [Database Access ↗](#). You'll need your database user's credentials in the next step.

Close **Choose a connection method**

Bước 7: Chọn phương thức kết nối, ở đây ta sẽ chọn **MongoDB for VSCode**.

Connect to Cluster0



Set up connection security



Choose a connection method



Connect

Connect to your application



Drivers

Access your Atlas data using MongoDB's native drivers (e.g. Node.js, Go, etc.)



Access your data through tools



Compass

Explore, modify, and visualize your data with MongoDB's GUI



Shell

Quickly add & update data using MongoDB's Javascript command-line interface



MongoDB for VS Code

Work with your data in MongoDB directly from your VS Code environment



Atlas SQL

Easily connect SQL tools to Atlas for data analysis and visualization



[Go Back](#)

[Close](#)

- Giao diện được hiển thị như hình dưới đây, ta sẽ copy dòng trong khung đỏ.

Connect to Cluster0

Connecting with MongoDB for VS Code

- 1. Install MongoDB for VS Code.**
In [VS Code](#), open "Extensions" in the left navigation and search for "MongoDB for VS Code." Select the extension and click install.
- 2. In VS Code, open the Command Palette.**
Click on "View" and open "Command Palette." Search "MongoDB: Connect" on the Command Palette and click on "Connect with Connection String."
- 3. Connect to your MongoDB deployment.**
Paste your connection string into the Command Palette.
 Show Password [?](#)

mongodb+srv://22520372:1@cluster0.amz3qgq.mongodb.net/

The password for **22520372** is included in the connection string for your first time setup. **This password will not be available again after exiting this connect flow.**

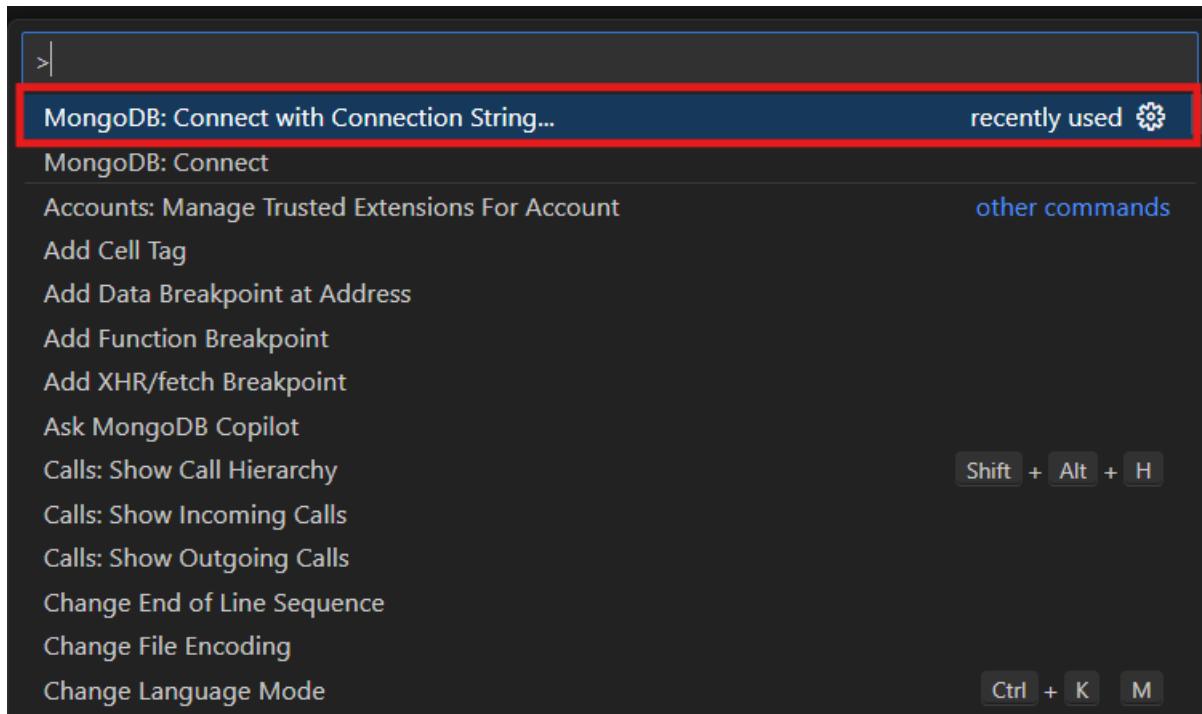
4. Click “Create New Playground” in MongoDB for VS Code to get started.
[Learn more about Playgrounds ↗](#)

RESOURCES

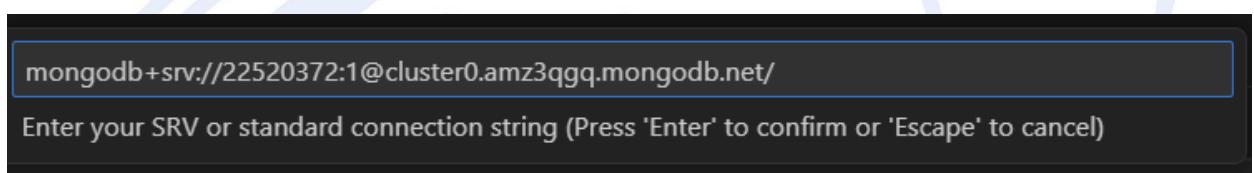
Connect to MongoDB through VSCode ↗	Explore your data with playgrounds ↗
Access your Database Users ↗	Troubleshoot Connections ↗

[Go Back](#)
[Done](#)

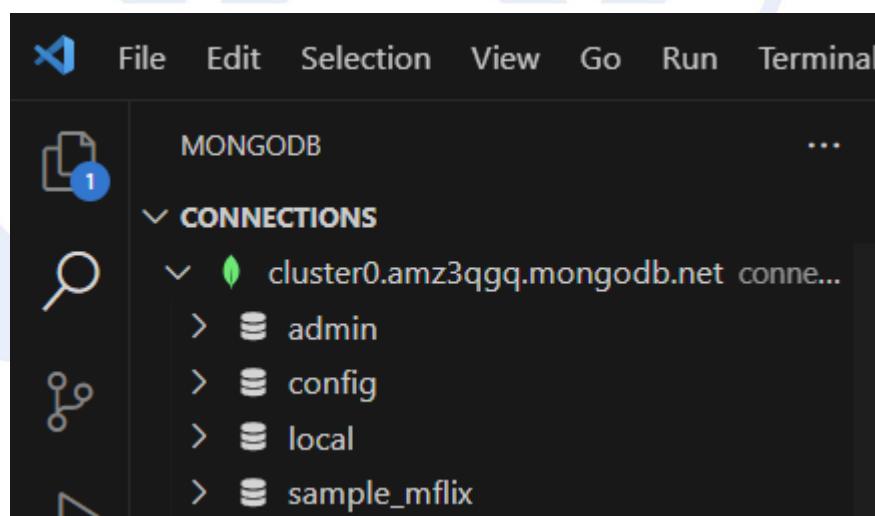
- Mở VSCode (đã được cài Extension *MongoDB for VS Code*) → ***View*** → ***Command Palette...*** → ***MongoDB: Connect with Connection String...***



- Sau đó paste connection string mình đã copy vào và nhấn **Enter**.



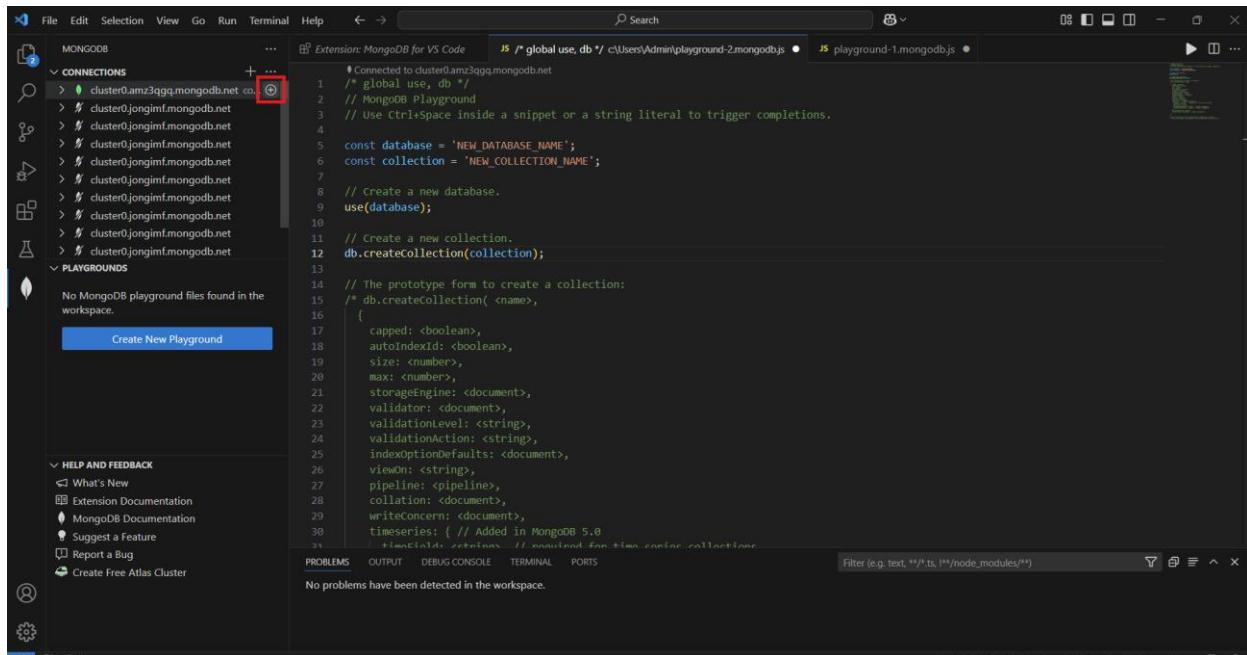
Bước 8: Đã kết nối thành công vào project.



Các user khác thực hiện tương tự.

3.1.5. Tạo database

Sau khi kết nối thành công vào project, ta nhấn dấu "+" để tạo database, VS Code sẽ hiển thị giao diện như hình bên dưới.



4. THAO TÁC DỮ LIỆU QUA LẠI GIỮA HAI MÁY

4.1. Cấu hình

Replica set

Bước 1: Trên từng máy, mở File Explorer và tạo:

- **C:\data\db** (có thể đặt máy 1 là db1, máy 2 là db2)
 - **C:\data\log**

 db1	6/1/2025 5:07 PM	File folder
 log	6/3/2025 8:47 PM	File folder

Bước 2: Chạy mongod (server) trên cả hai máy

Trên cả hai máy, chạy:

```
"C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe" --replSet "storeRS" --dbpath
```

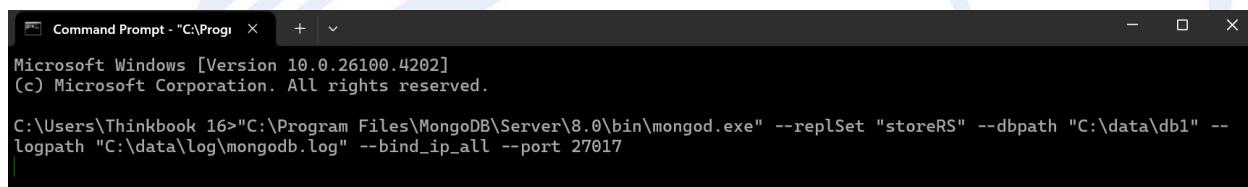
```
"C:\data\db1" --logpath "C:\data\log\mongodb.log" --bind_ip_all --port 27017
```

Trong đó :

- "C:\ProgramFiles\MongoDB\Server\8.0\bin\mongod.exe": Đường dẫn tới file thực thi **mongod** (MongoDB Server daemon).

- **-replicaSet "storeRS"**: Khai báo đây là **thành viên của một replica set** (tên là storeRS).
- **-dbpath "C:\data\db1"**: Chỉ định **thư mục lưu trữ dữ liệu database**.
- **-logpath "C:\data\log\mongodb.log"**: Chỉ định **đường dẫn file log** để ghi lại thông tin hoạt động, lỗi cảnh báo...
- **--bind_ip_all**: Cho phép **kết nối đến server MongoDB** này từ **tất cả các địa chỉ IP**
- **--port 27017**: Khởi động mongod **lắng nghe (listen)** ở cổng **27017** (mặc định của MongoDB).

Lưu ý: Để cửa sổ này mở nguyên (nếu đóng lại, MongoDB sẽ tắt)



```
Command Prompt - "C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe" --replSet "storeRS" --dbpath "C:\data\db1" --logpath "C:\data\log\mongodb.log" --bind_ip_all --port 27017
```

Bước 3: Khởi tạo replica set (chỉ trên một máy)

Mở CMD mới trên máy, chuyển vào thư mục có mongosh (nếu đã cài mongosh):

mongosh

Khởi tạo Replica Set:

```
rs.initiate({
  _id: "storeRS",
  members: [
    { _id: 0, host: "26.73.81.60:27017" },
    { _id: 1, host: "26.121.107.27:27017" }
  ]
})
```

Trong đó:

- **rs.initiate({...})**: Lệnh **khởi tạo (initiate)** một replica set.
- **_id: "storeRS"**: Đặt **tên cho replica set** là "storeRS".(giống với khi khởi động mongod)
- **members:[...]**: Danh sách các **thành viên** (nodes) trong replica set này.

- `{_id:0,host:"26.73.81.60:27017"}`: Thành viên thứ nhất của replica set (thường là PRIMARY đầu tiên).
- `{_id:1,host:"26.121.107.27:27017"}`: Thành viên thứ hai của replica set (có thể là SECONDARY).

```
test> rs.initiate({
...   _id: "storeRS",
...   members: [
...     { _id: 0, host: "26.73.81.60:27017" },
...     { _id: 1, host: "26.121.107.27:27017" }
...   ]
... })
...
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1748772037, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1748772037, i: 1 })
}
storeRS [direct: secondary] test>
```

Kiểm tra trạng thái bằng lệnh: `rs.status()`

```
stateStr: 'SECONDARY',
uptime: 11,
optime: { ts: Timestamp({ t: 1748772037, i: 1 }), t: Long('-1') },
optimeDurable: { ts: Timestamp({ t: 1748772037, i: 1 }), t: Long('-1') },
optimeWritten: { ts: Timestamp({ t: 1748772037, i: 1 }), t: Long('-1') },
optimeDate: ISODate('2025-06-01T10:00:37.000Z'),
optimeDurableDate: ISODate('2025-06-01T10:00:37.000Z'),
optimeWrittenDate: ISODate('2025-06-01T10:00:37.000Z'),
lastAppliedWallTime: ISODate('2025-06-01T10:00:37.037Z'),
lastDurableWallTime: ISODate('2025-06-01T10:00:37.037Z'),
lastWrittenWallTime: ISODate('2025-06-01T10:00:37.037Z'),
lastHeartbeat: ISODate('2025-06-01T10:00:48.081Z'),
lastHeartbeatRecv: ISODate('2025-06-01T10:00:48.082Z'),
pingMs: Long('10'),
lastHeartbeatMessage: '',
syncSourceHost: '26.73.81.60:27017',
syncSourceId: 0,
infoMessage: '',
configVersion: 1,
configTerm: 1
},
ok: 1,
'$clusterTime': {
  clusterTime: Timestamp({ t: 1748772048, i: 10 }),
  signature: {
    hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
    keyId: Long('0')
  }
},
operationTime: Timestamp({ t: 1748772048, i: 10 })
}
storeRS [direct: primary] test>
```

Sharding

Bước 1: Trên từng máy, mở File Explorer và tạo:

- **C:\data\shard**
- **C:\data\log**
- **C:\data\config**

Name	Date modified	Type	Size
config	6/3/2025 10:02 PM	File folder	
db1	6/3/2025 11:21 PM	File folder	
log	6/3/2025 8:47 PM	File folder	
shard	6/3/2025 10:03 PM	File folder	

Bước 2: Khởi động config server trên mỗi máy

```
"C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe" --configsvr --replSet configReplSet --port 27019 --dbpath C:\data\config --logpath C:\data\log\config.log --bind_ip_all
```



```
Command Prompt - "C:\Progr X + \v
Microsoft Windows [Version 10.0.26100.4202]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Thinkbook 16>"C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe" --configsvr --replSet configReplSet --port 27019 --dbpath C:\data\config --logpath C:\data\log\config.log --bind_ip_all
{"t": {"$date": "2025-06-03T16:22:15.025Z"}, "s": "I", "c": "CONTROL", "id": 20697, "ctx": "thread1", "msg": "Renamed existing log file", "attr": {"oldLogPath": "C:\\data\\log\\config.log", "newLogPath": "C:\\data\\log\\config.log.2025-06-03T16-22-15"}}
```

Bước 3: Khởi tạo replica set cho config server

Trên máy 1, mở mongosh: **mongosh --port 27019**

Khởi tạo replica set

```
rs.initiate({
  _id: "configReplSet",
  configsvr: true,
  members: [
    { _id: 0, host: "26.73.81.60:27019" },
    { _id: 1, host: "26.121.107.27:27019" }
  ]
})
```

```
test> rs.initiate({
...   _id: "configReplSet",
...   configsvr: true,
...   members: [
...     { _id: 0, host: "26.73.81.60:27019" },
...     { _id: 1, host: "26.121.107.27:27019" }
...   ]
... })
...
{
```

```
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1748784761, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1748784761, i: 1 })
}
configReplSet [direct: secondary] test>
```

Bước 4: Khởi động mongod shard trên mỗi máy

Máy 1 : "C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe" --shardsvr --replSet shardReplSet1 --port 27018 --dbpath C:\data\shard --logpath C:\data\log\shard.log --bind_ip_all

Máy 2: "C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe" --shardsvr --replSet shardReplSet2 --port 27018 --dbpath C:\data\shard --logpath C:\data\log\shard.log --bind_ip_all

The screenshot shows two Command Prompt windows running on Microsoft Windows 10. Both windows have the title 'Command Prompt - "C:\Program' and show the command 'C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe" --shardsvr --replSet shardReplSet1 --port 27018 --dbpath C:\data\shard --logpath C:\data\log\shard.log --bind_ip_all'. The first window's log output is visible, showing a log entry with timestamp, log file path, and new log path.

```
C:\Users\Thinkbook 16>"C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe" --shardsvr --replSet shardReplSet1 --port 27018 --dbpath C:\data\shard --logpath C:\data\log\shard.log --bind_ip_all
{"t": {"$date": "2025-06-03T16:22:44.225Z"}, "s": "I", "c": "CONTROL", "id": 20697, "ctx": "thread1", "msg": "Renamed existing log file", "attr": {"oldLogPath": "C:\\data\\log\\shard.log", "newLogPath": "C:\\data\\log\\shard.log.2025-06-03T16-22-44"}}
```

Bước 5: Khởi tạo replica set cho từng shard

- Máy 1, vào mongosh: **mongosh --port 27018**

Khởi tạo replica set

```
rs.initiate({
  _id: "shardReplSet1",
  members: [
    { _id: 0, host: "26.73.81.60:27018" }
  ]
})
```

```
test> rs.initiate({
...   _id: "shardReplSet1",
...   members: [
...     { _id: 0, host: "26.73.81.60:27018" }
...   ]
... })
```

```
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1748784852, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1748784852, i: 1 })
}
shardReplSet1 [direct: secondary] test> |
```

– Máy 2, vào mongosh: **mongosh --port 27018**

Khởi tạo replica set

```
rs.initiate({
  _id: "shardReplSet2",
  members: [
    { _id: 0, host: "26.121.107.27:27018" }
  ]
})
```

```
test> rs.initiate({
...   _id: "shardReplSet2",
...   members: [
...     { _id: 0, host: "26.121.107.27:27018" }
...   ]
... })
```

```
{
  "ok": 1,
  "$clusterTime": {
    "clusterTime": Timestamp({ t: 1748784904, i: 1 }),
    "signature": {
      "hash": Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      "keyId": Long('0')
    }
  },
  "operationTime": Timestamp({ t: 1748784904, i: 1 })
}
shardReplSet2 [direct: secondary] test> |
```

Bước 6: Khởi động mongos trên máy 1 (có thẻ cả máy 2)

```
"C:\Program Files\MongoDB\Server\8.0\bin\mongos.exe" --configdb
configReplSet/26.73.81.60:27019,26.121.107.27:27019 --port 27017 --bind_ip_all --
logpath C:\data\log\mongos.log
```



```
Microsoft Windows [Version 10.0.26100.4202]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Thinkbook 16>"C:\Program Files\MongoDB\Server\8.0\bin\mongos.exe" --configdb configReplSet/26.73.81.60:27019,26
.121.107.27:27019 --port 27017 --bind_ip_all --logpath C:\data\log\mongos.log
{"t": {"$date": "2025-06-03T16:23:55.523Z"}, "s": "I", "c": "SHARDING", "id": 24132, "ctx": "thread1", "msg": "Running a shard
ed cluster with fewer than 3 config servers should only be done for testing purposes and is not recommended for producti
on."}
{"t": {"$date": "2025-06-03T16:23:55.526Z"}, "s": "I", "c": "CONTROL", "id": 20697, "ctx": "thread1", "msg": "Renamed existin
g log file", "attr": {"oldLogPath": "C:\\data\\log\\mongos.log", "newLogPath": "C:\\data\\log\\mongos.log.2025-06-03T16-23-55
"}}
```

Bước 7: Thêm shard vào cluster qua mongos

- Trên **máy 1**, vào mongosh port 27017:

mongosh --host 26.73.81.60 --port 27017

```
sh.addShard("shardReplSet1/26.73.81.60:27018")
```

```
sh.addShard("shardReplSet2/26.121.107.27:27018")
```

```
[direct: mongos] test> sh.addShard("shardReplSet1/26.73.81.60:27018"
{
  shardAdded: 'shardReplSet1',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1748784927, i: 22 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1748784927, i: 22 })
}
[direct: mongos] test>
```

```
[direct: mongos] test> sh.addShard("shardReplSet2/26.121.107.27:27018"
{
  shardAdded: 'shardReplSet2',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1748784935, i: 23 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1748784935, i: 17 })
}
[direct: mongos] test> use badminton_store
```

Bước 8. Bật sharding cho database và collection

```
use badminton_store
```

```
sh.enableSharding("badminton_store")
```

```
sh.shardCollection("badminton_store.bills", {branch_id: 1})
```

```
sh.shardCollection("badminton_store.inventory", {branch_id: 1})
```

```
sh.shardCollection("badminton_store.employees", {branch_id: 1})
```

```
// (products/customers thường dùng chung, không cần shard theo branch_id)
```

```
[direct: mongos] test> use badminton_store
switched to db badminton_store
[direct: mongos] badminton_store> sh.enableSharding("badminton_stor
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1748784955, i: 9 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1748784955, i: 6 })
}
[direct: mongos] badminton_store> sh.shardCollection("badminton_sto
```

```
[direct: mongos] badminton_store> sh.shardCollection("badminton_sto
  1})
{
  collectionsharded: 'badminton_store.employees',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1748784987, i: 37 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1748784987, i: 37 })
}
```

Tạo bảng

Replica Set

```
from pymongo import MongoClient

# Kết nối MongoDB (thay connection string của bạn)
client = MongoClient("mongodb://26.73.81.60:27017,26.121.107.27:27017/?replicaSet=storeRS")

# Chọn database
db = client["badminton_store"]

# Tạo các collection (tạo trước, giống SQL "CREATE TABLE")
db.create_collection("products")
db.create_collection("customers")
db.create_collection("employees")
db.create_collection("inventory")
db.create_collection("bills")

print("✓ Đã tạo toàn bộ collections!")
```

Sharding

```

from pymongo import MongoClient

client = MongoClient("mongodb://26.73.81.60:27017")
db = client["badminton_store"]

# Tạo các collection (tạo trước, giống SQL "CREATE TABLE")
db.create_collection("products")
db.create_collection("customers")
db.create_collection("employees")
db.create_collection("inventory")
db.create_collection("bills")

print("✅ Đã tạo toàn bộ collections!")

```

4.2. Thêm dữ liệu

Replica Set

```

from pymongo import MongoClient
from datetime import datetime

client = MongoClient("mongodb://26.73.81.60:27017,26.121.107.27:27017/?replicaSet=storeRS")
db = client["badminton_store"]

# Insert nhiều trường cho các bảng
db.products.insert_many([
    {"_id": 1, "name": "Vợt Yonex Astrox 88D", "brand": "Yonex", "type": "attack", "price": 2000000, "warranty_months": 12},
    {"_id": 2, "name": "Vợt Victor Brave Sword 12", "brand": "Victor", "type": "defense", "price": 1850000, "warranty_months": 12}
])

db.customers.insert_many([
    {"_id": 1, "name": "Trần Văn C", "phone": "0901111222", "dob": "1998-05-11", "address": "Hà Nội"},
    {"_id": 2, "name": "Lê Thị D", "phone": "0903333444", "dob": "2001-09-22", "address": "Đà Nẵng"}
])

db.employees.insert_many([
    {"_id": 101, "name": "Nguyễn Văn A", "position": "Thủ ngân", "branch_id": 1, "phone": "0911222333"},
    {"_id": 102, "name": "Ngô Văn B", "position": "Bán hàng", "branch_id": 2, "phone": "0913444555"}
])

db.inventory.insert_many([
    {"_id": 1001, "product_id": 1, "quantity": 20, "branch_id": 1, "last_update": datetime.now()},
    {"_id": 1002, "product_id": 2, "quantity": 30, "branch_id": 2, "last_update": datetime.now()}
])

db.bills.insert_many([
    {
        "_id": 5001,
        "customer_id": 1,
        "employee_id": 101,
        "products": [{"product_id": 1, "qty": 2, "price": 2000000}],
        "total": 4000000,
        "date": datetime(2024, 5, 30, 15, 23),
        "status": "Đã thanh toán",
        "branch_id": 1
    },
    {
        "_id": 5002,
        "customer_id": 2,
        "employee_id": 102,
        "products": [{"product_id": 2, "qty": 1, "price": 1850000}],
        "total": 1850000,
        "date": datetime(2024, 5, 30, 17, 5),
        "status": "Chưa thanh toán",
        "branch_id": 2
    }
])

print("Đã insert dữ liệu mẫu!")

```

Sharding

```

1  from pymongo import MongoClient
2  from datetime import datetime, timedelta
3  import random
4
5  client = MongoClient("mongodb://26.73.81.60:27017") # Đổi IP nếu dùng mongos trên máy khác
6  db = client["badminton_store"]
7
8  # Xóa dữ liệu cũ nếu có
9  # Sản phẩm
10 product_names = [
11     ("Vợt Yonex Astrox 88D", "Yonex", "attack"),
12     ("Vợt Victor Brave Sword 12", "Victor", "defense"),
13     ("Vợt Lining Turbo Charging", "Lining", "attack"),
14     ("Quả cầu RSL", "RSL", "shuttlecock"),
15     ("Giày Mizuno Wave", "Mizuno", "shoes"),
16     ("Quả cầu Vina Star", "Vina", "shuttlecock"),
17     ("Vợt ProAce Stroke 318", "ProAce", "allround"),
18     ("Vợt Apacs Virtuoso", "Apacs", "allround"),
19     ("Giày Yonex Eclipsion Z", "Yonex", "shoes"),
20     ("Vợt Fleet Triotec", "Fleet", "attack")
21 ]
22 products = []
23 for i, (name, brand, type_) in enumerate(product_names, 1):
24     products.append({
25         "id": i,
26         "name": name,
27         "brand": brand,
28         "type": type_,
29         "price": random.randint(600000, 2500000),
30         "warranty_months": random.choice([0, 6, 12])
31     })
32 db.products.insert_many(products)
33
34 # Thêm 300 khách hàng
35 cities = ["Hà Nội", "Đà Nẵng", "Hải Phòng", "Hồ Chí Minh", "Cần Thơ", "Nghệ An", "Bắc Giang", "Huế", "Quảng Nam"]
36 customers = []
37 for i in range(1, 301):
38     customers.append({
39         "id": i,
40         "name": f"Khách hàng {i}",
41         "phone": "09" + ''.join(str(random.randint(0,9)) for _ in range(8)),
42         "dob": f"19{random.randint(70, 99)}-{random.randint(1,12)}-{random.randint(1,28)}:02d",
43         "address": random.choice(cities)
44     })
45 db.customers.insert_many(customers)
46
47 # Thêm 10 nhân viên (2 chi nhánh)
48 employees = []
49 for i in range(1, 11):
50     employees.append({
51         "id": 100 + i,
52         "name": f"Nhân viên {i}",
53         "position": random.choice(["Quản lý", "Bán hàng", "Thu ngân"]),
54         "branch_id": random.choice([1, 2]),
55         "phone": "09" + ''.join(str(random.randint(0,9)) for _ in range(8))
56     })
57 db.employees.insert_many(employees)
58

```

4.3. Xoá, sửa trong MongoDB

Replica Set

```
from pymongo import MongoClient

# Kết nối MongoDB (thay IP nếu cần)
client = MongoClient("mongodb://26.73.81.60:27017,26.121.107.27:27017/?replicaSet=storeRS")

# Chọn database
db = client["badminton_store"]

# Xóa từng collection (nếu tồn tại)
db.products.drop()
db.customers.drop()
db.employees.drop()
db.inventory.drop()
db.bills.drop()

print("✅ Đã xóa toàn bộ collections!")
```

Sharding

```
from pymongo import MongoClient

# Kết nối MongoDB (thay IP nếu cần)
client = MongoClient("mongodb://26.73.81.60:27017")

# Chọn database
db = client["badminton_store"]

# Xóa từng collection (nếu tồn tại)
db.products.drop()
db.customers.drop()
db.employees.drop()
db.inventory.drop()
db.bills.drop()

print("✅ Đã xóa toàn bộ collections!")
```

4.4. Truy vấn trong MongoDB

4.4.1. Câu 1: Tìm khách hàng có nhiều hóa đơn nhất.

```
from pymongo import MongoClient
client = MongoClient("mongodb://26.73.81.60:27017")
db = client["badminton_store"]
print("Khách hàng có nhiều hóa đơn nhất:")
pipeline = [
    {"$group": {"_id": "$customer_id", "count": {"$sum": 1}}},
    {"$sort": {"count": -1}},
```

```

    {"$limit": 1},
    {"$lookup": {
        "from": "customers",
        "localField": "_id",
        "foreignField": "_id",
        "as": "customer"
    }},
    {"$unwind": "$customer"}
]
for row in db.bills.aggregate(pipeline):
    print(row)

```

Kết quả:

```

Khách hàng có nhiều hóa đơn nhất:
{'_id': 247, 'count': 6, 'customer': {'_id': 247, 'name': 'Khách hàng 247', 'phone': '0955108283', 'dob': '1991-02-20',
'address': 'Huế'}}

```

4.4.2. Câu 2: Tìm sản phẩm bán chạy nhất

```

from pymongo import MongoClient
client = MongoClient("mongodb://26.73.81.60:27017")
db = client["badminton_store"]
print("Sản phẩm bán chạy nhất:")
pipeline = [
    {"$unwind": "$products"},
    {"$group": {"_id": "$products.product_id", "total_sold": {"$sum": "$products.qty"}}},
    {"$sort": {"total_sold": -1}},
    {"$limit": 1},
    {"$lookup": {
        "from": "products",
        "localField": "_id",
        "foreignField": "_id",
        "as": "product"
    }},
    {"$unwind": "$product"}
]
for row in db.bills.aggregate(pipeline):
    print(row)

```

Kết quả:

```

Sản phẩm bán chạy nhất:
{'_id': 5, 'total_sold': 521, 'product': {'_id': 5, 'name': 'Giày Mizuno Wave', 'brand': 'Mizuno', 'type': 'shoes',
'price': 2189050, 'warranty_months': 12}}

```

4.4.3. Câu 3: Tìm sản phẩm tồn kho ở chi nhánh 1 nhưng không có ở chi nhánh 2

```

from pymongo import MongoClient
client = MongoClient("mongodb://26.73.81.60:27017")
db = client["badminton_store"]
print("Tồn kho chỉ có ở CN1 mà không có ở CN2:")

```

```
# Tìm product_id có ở branch_id=1 nhưng không có ở branch_id=2
products_cn1 = set([x['product_id'] for x in db.inventory.find({"branch_id": 1})])
products_cn2 = set([x['product_id'] for x in db.inventory.find({"branch_id": 2})])
only_cn1 = products_cn1 - products_cn2
for inv in db.inventory.find({"branch_id": 1, "product_id": {"$in": list(only_cn1)}}):
    print(inv)
```

Kết quả:

```
Tồn kho chỉ có ở CN1 mà không có ở CN2:
{'_id': 1011, 'product_id': 11, 'quantity': 10, 'branch_id': 1, 'last_update': datetime.datetime(2025, 6, 3, 21, 8, 1, 970000)}
{'_id': 1012, 'product_id': 12, 'quantity': 5, 'branch_id': 1, 'last_update': datetime.datetime(2025, 6, 3, 21, 8, 1, 970000)}
```

4.4.4. Câu 4: Danh sách khách hàng đã mua hàng ở cả 2 chi nhánh

```
from pymongo import MongoClient
client = MongoClient("mongodb://26.73.81.60:27017")
db = client["badminton_store"]
print("Danh sách khách hàng có hóa đơn ở cả 2 chi nhánh:")
pipeline = [
    {
        "$group": {
            "_id": "$customer_id",
            "branches": {"$addToSet": "$branch_id"}
        }
    },
    {
        "$match": {
            "branches": {"$all": [1, 2]}
        }
    },
    {
        "$lookup": {
            "from": "customers",
            "localField": "_id",
            "foreignField": "_id",
            "as": "customer"
        }
    },
    {
        "$unwind": "$customer"
    }
]
for row in db.bills.aggregate(pipeline):
    print(row["customer"])
```

Kết quả:

Danh sách khách hàng có hóa đơn ở cả 2 chi nhánh:

```
{
  "_id": 47, "name": "Khách hàng 47", "phone": "0900194812", "dob": "1997-02-21", "address": "Hà Nội"},
  {"_id": 29, "name": "Khách hàng 29", "phone": "0986963698", "dob": "1994-06-01", "address": "Nghệ An"},
  {"_id": 124, "name": "Khách hàng 124", "phone": "0946403256", "dob": "1994-04-08", "address": "Huế"},
  {"_id": 155, "name": "Khách hàng 155", "phone": "0987986655", "dob": "1975-01-13", "address": "Quảng Nam"},
  {"_id": 263, "name": "Khách hàng 263", "phone": "0915549345", "dob": "1995-02-08", "address": "Hồ Chí Minh"},
  {"_id": 214, "name": "Khách hàng 214", "phone": "0930549778", "dob": "1976-10-05", "address": "Cần Thơ"},
  {"_id": 38, "name": "Khách hàng 38", "phone": "0990775163", "dob": "1996-06-10", "address": "Huế"},
  {"_id": 142, "name": "Khách hàng 142", "phone": "0968469743", "dob": "1999-04-12", "address": "Hải Phòng"},
  {"_id": 174, "name": "Khách hàng 174", "phone": "0968365765", "dob": "1994-07-13", "address": "Cần Thơ"},
  {"_id": 213, "name": "Khách hàng 213", "phone": "0921166216", "dob": "1975-12-08", "address": "Nghệ An"},
  {"_id": 21, "name": "Khách hàng 21", "phone": "0964435578", "dob": "1978-12-21", "address": "Đà Nẵng"},
  {"_id": 197, "name": "Khách hàng 197", "phone": "0970905301", "dob": "1991-08-10", "address": "Bắc Giang"},
  {"_id": 90, "name": "Khách hàng 90", "phone": "0902564597", "dob": "1996-06-28", "address": "Nghệ An"},
  {"_id": 95, "name": "Khách hàng 95", "phone": "0971013566", "dob": "1987-06-04", "address": "Đà Nẵng"},
  {"_id": 37, "name": "Khách hàng 37", "phone": "0930808011", "dob": "1995-01-23", "address": "Bắc Giang"},
  {"_id": 241, "name": "Khách hàng 241", "phone": "0944758154", "dob": "1984-08-18", "address": "Nghệ An"},
  {"_id": 242, "name": "Khách hàng 242", "phone": "0912668212", "dob": "1979-03-15", "address": "Quảng Nam"},
  {"_id": 133, "name": "Khách hàng 133", "phone": "0999493612", "dob": "1971-11-15", "address": "Cần Thơ"},
  {"_id": 69, "name": "Khách hàng 69", "phone": "0915843154", "dob": "1989-05-15", "address": "Đà Nẵng"},
  {"_id": 45, "name": "Khách hàng 45", "phone": "0917929375", "dob": "1973-12-21", "address": "Đà Nẵng"},
  {"_id": 222, "name": "Khách hàng 222", "phone": "0950809512", "dob": "1999-08-06", "address": "Cần Thơ"}
}
```

4.4.5. Câu 5: Tìm những khách hàng từng mua sản phẩm Vợt Lining Turbo Charging

```
from pymongo import MongoClient
client = MongoClient("mongodb://26.73.81.60:27017")
db = client["badminton_store"]
print("Khách hàng từng mua Vợt Lining Turbo Charging:")
# Tìm product_id của 'Vợt Lining Turbo Charging'
product = db.products.find_one({"name": "Vợt Lining Turbo Charging"})
if product:
    product_id = product["_id"]
    bill_cust_ids = db.bills.distinct("customer_id", {"products.product_id": product_id})
    for c in db.customers.find({"_id": {"$in": bill_cust_ids}}):
        print(c)
```

Kết quả:

```
Khách hàng từng mua Vợt Lining Turbo Charging:
[{"_id": 2, "name": "Khách hàng 2", "phone": "0914817805", "dob": "1995-05-21", "address": "Hồ Chí Minh"}, {"_id": 4, "name": "Khách hàng 4", "phone": "0964570399", "dob": "1992-04-24", "address": "Nghệ An"}, {"_id": 8, "name": "Khách hàng 8", "phone": "0947536995", "dob": "1995-02-10", "address": "Bắc Giang"}, {"_id": 9, "name": "Khách hàng 9", "phone": "0984033252", "dob": "1979-06-17", "address": "Nghệ An"}, {"_id": 15, "name": "Khách hàng 15", "phone": "0946004118", "dob": "1985-11-13", "address": "Quảng Nam"}, {"_id": 16, "name": "Khách hàng 16", "phone": "0978583375", "dob": "1974-02-27", "address": "Cần Thơ"}, {"_id": 18, "name": "Khách hàng 18", "phone": "0968373320", "dob": "1999-08-11", "address": "Hà Nội"}, {"_id": 21, "name": "Khách hàng 21", "phone": "0964435578", "dob": "1978-12-21", "address": "Đà Nẵng"}, {"_id": 29, "name": "Khách hàng 29", "phone": "0986963698", "dob": "1994-06-01", "address": "Nghệ An"}, {"_id": 31, "name": "Khách hàng 31", "phone": "0962510713", "dob": "1978-03-18", "address": "Cần Thơ"}, {"_id": 33, "name": "Khách hàng 33", "phone": "0950444114", "dob": "1987-03-07", "address": "Đà Nẵng"}, {"_id": 34, "name": "Khách hàng 34", "phone": "0953661243", "dob": "1991-11-25", "address": "Quảng Nam"}, {"_id": 36, "name": "Khách hàng 36", "phone": "0998162645", "dob": "1984-12-17", "address": "Hải Phòng"}, {"_id": 38, "name": "Khách hàng 38", "phone": "0990775163", "dob": "1996-06-10", "address": "Huế"}, {"_id": 41, "name": "Khách hàng 41", "phone": "0967783815", "dob": "1984-11-18", "address": "Hải Phòng"}, {"_id": 42, "name": "Khách hàng 42", "phone": "0942825575", "dob": "1973-08-17", "address": "Nghệ An"}, {"_id": 43, "name": "Khách hàng 43", "phone": "0934405689", "dob": "1974-07-11", "address": "Bắc Giang"}, {"_id": 47, "name": "Khách hàng 47", "phone": "0900194812", "dob": "1997-02-21", "address": "Hà Nội"}, {"_id": 49, "name": "Khách hàng 49", "phone": "0976909409", "dob": "1996-11-24", "address": "Hồ Chí Minh"}, {"_id": 50, "name": "Khách hàng 50", "phone": "0903600365", "dob": "1975-03-24", "address": "Bắc Giang"}, {"_id": 52, "name": "Khách hàng 52", "phone": "0924570843", "dob": "1983-11-03", "address": "Bắc Giang"}, {"_id": 53, "name": "Khách hàng 53", "phone": "0925876821", "dob": "1991-07-17", "address": "Bắc Giang"}, {"_id": 59, "name": "Khách hàng 59", "phone": "0992985742", "dob": "1977-12-02", "address": "Cần Thơ"}, {"_id": 64, "name": "Khách hàng 64", "phone": "0936798697", "dob": "1993-10-25", "address": "Huế"}]
```

4.4.6. Câu 6: Thống kê số lượng hóa đơn của từng nhân viên

```
from pymongo import MongoClient
client = MongoClient("mongodb://26.73.81.60:27017")
db = client["badminton_store"]
print("\nSố lượng hóa đơn của từng nhân viên:")
```

```

pipeline = [
    {"$group": {"_id": "$employee_id", "num_bills": {"$sum": 1}}},
    {"$sort": {"num_bills": -1}},
    {"$lookup": {
        "from": "employees",
        "localField": "_id",
        "foreignField": "_id",
        "as": "employee"
    }},
    {"$unwind": "$employee"}
]
for row in db.bills.aggregate(pipeline):
    print(row)

```

Kết quả:

```

Số lượng hóa đơn của từng nhân viên:
{"_id": 105, "num_bills": 57, "employee": {"_id": 105, "name": "Nhân viên 5", "position": "Bán hàng", "branch_id": 1, "phone": "0916893189"}}
{"_id": 108, "num_bills": 56, "employee": {"_id": 108, "name": "Nhân viên 8", "position": "Thu ngân", "branch_id": 2, "phone": "0942702943"}}
{"_id": 106, "num_bills": 55, "employee": {"_id": 106, "name": "Nhân viên 6", "position": "Thu ngân", "branch_id": 2, "phone": "0911096357"}}
{"_id": 102, "num_bills": 55, "employee": {"_id": 102, "name": "Nhân viên 2", "position": "Thu ngân", "branch_id": 1, "phone": "0992798990"}}
{"_id": 109, "num_bills": 49, "employee": {"_id": 109, "name": "Nhân viên 9", "position": "Thu ngân", "branch_id": 2, "phone": "0925257868"}}
{"_id": 104, "num_bills": 47, "employee": {"_id": 104, "name": "Nhân viên 4", "position": "Quản lý", "branch_id": 1, "phone": "0952883844"}}
{"_id": 110, "num_bills": 47, "employee": {"_id": 110, "name": "Nhân viên 10", "position": "Bán hàng", "branch_id": 2, "phone": "0982470026"}}
{"_id": 101, "num_bills": 46, "employee": {"_id": 101, "name": "Nhân viên 1", "position": "Bán hàng", "branch_id": 1, "phone": "0934350933"}}
{"_id": 103, "num_bills": 45, "employee": {"_id": 103, "name": "Nhân viên 3", "position": "Quản lý", "branch_id": 2, "phone": "0974397451"}}
{"_id": 107, "num_bills": 43, "employee": {"_id": 107, "name": "Nhân viên 7", "position": "Bán hàng", "branch_id": 2, "phone": "0920963943"}}

```

4.4.7. Câu 7: Thống kê tồn kho từng sản phẩm ở mỗi chi nhánh

```

from pymongo import MongoClient
client = MongoClient("mongodb://26.73.81.60:27017")
db = client["badminton_store"]
print("\nTồn kho của từng sản phẩm ở từng chi nhánh:")
pipeline = [
    {"$group": {
        "_id": {"product_id": "$product_id", "branch_id": "$branch_id"},
        "quantity": {"$sum": "$quantity"}
    }},
    {"$lookup": {
        "from": "products",
        "localField": "_id.product_id",
        "foreignField": "_id",
        "as": "product"
    }},
    {"$unwind": "$product"},

    {"$project": {
        "branch_id": "_id.branch_id",
        "product_id": "_id.product_id",
        "name": "$product.name",
    }}
]
for row in db.products.aggregate(pipeline):
    print(row)

```

```

        "quantity": 1
    },
    {"$sort": {"branch_id": 1, "quantity": -1}}
]
for row in db.inventory.aggregate(pipeline):
    print(row)

```

Kết quả:

```
Tổng kho của từng sản phẩm ở từng chi nhánh:
[{"_id": {"product_id": 8, "branch_id": 1}, "quantity": 60, "branch_id": 1, "product_id": 8, "name": "Vợt Apacs Virtuoso"}, {"_id": {"product_id": 10, "branch_id": 1}, "quantity": 56, "branch_id": 1, "product_id": 10, "name": "Vợt Fleet Triotec"}, {"_id": {"product_id": 9, "branch_id": 1}, "quantity": 48, "branch_id": 1, "product_id": 9, "name": "Giày Yonex Eclipsion Z"}, {"_id": {"product_id": 6, "branch_id": 1}, "quantity": 38, "branch_id": 1, "product_id": 6, "name": "Quả cầu Vina Star"}, {"_id": {"product_id": 4, "branch_id": 1}, "quantity": 34, "branch_id": 1, "product_id": 4, "name": "Quả cầu RSL"}, {"_id": {"product_id": 3, "branch_id": 1}, "quantity": 25, "branch_id": 1, "product_id": 3, "name": "Vợt Lining Turbo Charging"}, {"_id": {"product_id": 5, "branch_id": 1}, "quantity": 22, "branch_id": 1, "product_id": 5, "name": "Giày Mizuno Wave"}, {"_id": {"product_id": 7, "branch_id": 1}, "quantity": 19, "branch_id": 1, "product_id": 7, "name": "Vợt ProAce Stroke 318"}, {"_id": {"product_id": 1, "branch_id": 1}, "quantity": 5, "branch_id": 1, "product_id": 1, "name": "Vợt Yonex Astrox 88D"}, {"_id": {"product_id": 2, "branch_id": 1}, "quantity": 5, "branch_id": 1, "product_id": 2, "name": "Vợt Victor Brave Sword 12"}, {"_id": {"product_id": 8, "branch_id": 2}, "quantity": 56, "branch_id": 2, "product_id": 8, "name": "Vợt Apacs Virtuoso"}, {"_id": {"product_id": 10, "branch_id": 2}, "quantity": 43, "branch_id": 2, "product_id": 10, "name": "Vợt Fleet Triotec"}, {"_id": {"product_id": 5, "branch_id": 2}, "quantity": 31, "branch_id": 2, "product_id": 5, "name": "Giày Mizuno Wave"}, {"_id": {"product_id": 1, "branch_id": 2}, "quantity": 27, "branch_id": 2, "product_id": 1, "name": "Vợt Yonex Astrox 88D"}, {"_id": {"product_id": 6, "branch_id": 2}, "quantity": 24, "branch_id": 2, "product_id": 6, "name": "Quả cầu Vina Star"}, {"_id": {"product_id": 4, "branch_id": 2}, "quantity": 23, "branch_id": 2, "product_id": 4, "name": "Quả cầu RSL"}, {"_id": {"product_id": 2, "branch_id": 2}, "quantity": 15, "branch_id": 2, "product_id": 2, "name": "Vợt Victor Brave Sword 12"}, {"_id": {"product_id": 3, "branch_id": 2}, "quantity": 14, "branch_id": 2, "product_id": 3, "name": "Vợt Lining Turbo Charging"}, {"_id": {"product_id": 7, "branch_id": 2}, "quantity": 10, "branch_id": 2, "product_id": 7, "name": "Vợt ProAce Stroke 318"}, {"_id": {"product_id": 9, "branch_id": 2}, "quantity": 6, "branch_id": 2, "product_id": 9, "name": "Giày Yonex Eclipsion Z"}]
```

4.4.8. Câu 8: Top 3 sản phẩm giá cao nhất của mỗi chi nhánh

```

from pymongo import MongoClient
client = MongoClient("mongodb://26.73.81.60:27017")
db = client["badminton_store"]
print("\nTop 3 sản phẩm giá cao nhất của mỗi chi nhánh:")
pipeline = [
    {
        "$lookup": {
            "from": "products",
            "localField": "product_id",
            "foreignField": "_id",
            "as": "product"
        }
    },
    {"$unwind": "$product"},

    {
        "$group": {
            "_id": {"branch_id": "$branch_id", "product_id": "$product_id"},
            "name": {"$first": "$product.name"},
            "price": {"$first": "$product.price"},
            "quantity": {"$sum": "$quantity"}
        }
    },
    {"$sort": {"_id.branch_id": 1, "price": -1}},
    {
        "$group": {
            "_id": "$_id.branch_id",
            "products": {
                "$push": {

```

```

        "product_id": "$_id.product_id",
        "name": "$name",
        "price": "$price",
        "quantity": "$quantity"
    }
}
},
{
    "$project": {
        "products": {"$slice": ["$products", 3]}
    }
}
]
for row in db.inventory.aggregate(pipeline):
    print(f"Chi nhánh {row['_id']}:")
    for prod in row["products"]:
        print(" ", prod)

```

Kết quả:

```

Top 3 sản phẩm giá cao nhất của mỗi chi nhánh:
Chi nhánh 1:
{'product_id': 5, 'name': 'Giày Mizuno Wave', 'price': 2189050, 'quantity': 22}
{'product_id': 8, 'name': 'Vợt Apacs Virtuoso', 'price': 1949001, 'quantity': 60}
{'product_id': 4, 'name': 'Quả cầu RSL', 'price': 1923844, 'quantity': 34}
Chi nhánh 2:
{'product_id': 5, 'name': 'Giày Mizuno Wave', 'price': 2189050, 'quantity': 31}
{'product_id': 8, 'name': 'Vợt Apacs Virtuoso', 'price': 1949001, 'quantity': 56}
{'product_id': 4, 'name': 'Quả cầu RSL', 'price': 1923844, 'quantity': 23}

```

4.4.9. Câu 9: Tìm hóa đơn chưa thanh toán của chi nhánh 2

```

from pymongo import MongoClient
client = MongoClient("mongodb://26.73.81.60:27017")
db = client["badminton_store"]
print("\nHóa đơn chưa thanh toán của chi nhánh 2:")
for bill in db.bills.find({"status": "Chưa thanh toán", "branch_id": 2}):
    print(bill)

```

Kết quả:

```
Hóa đơn chưa thanh toán của chi nhánh 2:
[{"_id": 10009, "customer_id": 57, "employee_id": 106, "products": [{"product_id": 1, "qty": 1, "price": 754280}, {"product_id": 7, "qty": 4, "price": 1188021}], "total": 5506364, "date": datetime.datetime(2023, 6, 23, 19, 47), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10010, "customer_id": 104, "employee_id": 104, "products": [{"product_id": 6, "qty": 4, "price": 774067}, {"product_id": 7, "qty": 5, "price": 1188021}, {"product_id": 1, "qty": 6, "price": 754280}], "total": 13562053, "date": datetime.datetime(2023, 4, 15, 15, 13), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10012, "customer_id": 54, "employee_id": 108, "products": [{"product_id": 5, "qty": 2, "price": 2189058}, {"product_id": 10, "qty": 4, "price": 886546}], "total": 7924284, "date": datetime.datetime(2023, 7, 22, 15, 28), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10018, "customer_id": 238, "employee_id": 104, "products": [{"product_id": 5, "qty": 3, "price": 2189050}, {"product_id": 9, "qty": 7, "price": 1421530}], "total": 1651786, "date": datetime.datetime(2023, 3, 9, 17, 29), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10019, "customer_id": 120, "employee_id": 102, "products": [{"product_id": 3, "qty": 2, "price": 652128}, {"product_id": 4, "qty": 3, "price": 1923844}, {"product_id": 10, "qty": 7, "price": 886546}], "total": 13281594, "date": datetime.datetime(2023, 11, 14, 11, 52), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10021, "customer_id": 152, "employee_id": 110, "products": [{"product_id": 1, "qty": 5, "price": 754280}, {"product_id": 6, "qty": 4, "price": 774067}], "total": 6867668, "date": datetime.datetime(2023, 8, 12, 20, 4), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10033, "customer_id": 211, "employee_id": 103, "products": [{"product_id": 8, "qty": 5, "price": 1949001}, {"product_id": 10, "qty": 3, "price": 886546}], "total": 1240464, "date": datetime.datetime(2023, 11, 25, 12, 7), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10035, "customer_id": 157, "employee_id": 109, "products": [{"product_id": 4, "qty": 3, "price": 1923844}, {"product_id": 6, "qty": 3, "price": 774067}, {"product_id": 5, "qty": 5, "price": 2189050}, {"product_id": 8, "qty": 4, "price": 1949001}], "total": 26834987, "date": datetime.datetime(2023, 4, 20, 12, 48), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10038, "customer_id": 255, "employee_id": 103, "products": [{"product_id": 5, "qty": 6, "price": 2189050}, {"product_id": 6, "qty": 5, "price": 774067}, {"product_id": 10, "qty": 2, "price": 886546}, {"product_id": 1, "qty": 5, "price": 754280}], "total": 22549127, "date": datetime.datetime(2023, 5, 19, 13, 6), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10049, "customer_id": 8, "employee_id": 104, "products": [{"product_id": 4, "qty": 7, "price": 1923844}, {"product_id": 5, "qty": 6, "price": 2189050}], "total": 26601208, "date": datetime.datetime(2023, 1, 23, 20, 31), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10050, "customer_id": 269, "employee_id": 108, "products": [{"product_id": 5, "qty": 3, "price": 2189050}, {"product_id": 9, "qty": 2, "price": 1421530}], "total": 9410210, "date": datetime.datetime(2023, 8, 22, 8, 19), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10051, "customer_id": 132, "employee_id": 106, "products": [{"product_id": 10, "qty": 4, "price": 886546}, {"product_id": 5, "qty": 4, "price": 2189050}, {"product_id": 1, "qty": 1, "price": 754280}], "total": 13856664, "date": datetime.datetime(2023, 11, 4, 16, 26), "status": "Chưa thanh toán", "branch_id": 2}
[{"_id": 10057, "customer_id": 146, "employee_id": 106, "products": [{"product_id": 1, "qty": 7, "price": 754280}, {"product_id": 5, "qty": 7, "price": 2189050}], "total": 20603310, "date": datetime.datetime(2023, 5, 12, 8, 2), "status": "Chưa thanh toán", "branch_id": 2}
```

4.4.10. Câu 10: Tìm khách hàng đã mua toàn bộ sản phẩm

```
from pymongo import MongoClient
client = MongoClient("mongodb://26.73.81.60:27017")
db = client["badminton_store"]
product_ids = set(p["_id"] for p in db.products.find())
pipeline = [
    {"$unwind": "$products"}, 
    {"$group": {
        "_id": "$customer_id",
        "purchased_products": {"$addToSet": "$products.product_id"}
    }},
    {"$project": {
        "all_products": {"$setEquals": ["$purchased_products", 
list(product_ids)]}
    }},
    {"$match": {"all_products": true}},
    {"$lookup": {
        "from": "customers",
        "localField": "_id",
        "foreignField": "_id",
        "as": "customer"
    }},
    {"$unwind": "$customer"}
]
print("Danh sách khách hàng đã mua toàn bộ sản phẩm:")
for row in db.bills.aggregate(pipeline):
    print(row["customer"])
```

Kết quả:

```
Danh sách khách hàng đã mua toàn bộ sản phẩm:
[{"_id": 288, "name": "Khách hàng 288", "phone": "0959666300", "dob": "1983-03-08", "address": "Hải Phòng"}, {"_id": 118, "name": "Khách hàng 118", "phone": "0986143059", "dob": "1991-10-03", "address": "Hà Nội"}, {"_id": 247, "name": "Khách hàng 247", "phone": "0955108283", "dob": "1991-02-20", "address": "Huế"}]
```



5. PHÂN CÔNG CÔNG VIỆC

Bảng 1. Phân công công việc

Tên thành viên	Nội dung công việc	Mức độ hoàn thành	Mức độ đóng góp
Nguyễn Thị Hải Hà	Viết báo cáo chính Chạy demo trực tiếp	100%	25%
Nguyễn Thị Thanh Hiếu	Viết báo cáo Làm powerpoint Chạy demo trực tiếp	100%	25%
Nguyễn Hồng Nhật Long	Chạy demo thử nghiệm Chỉnh sửa video dự phòng	100%	25%
Phạm Đức Mạnh	Chạy demo thử nghiệm	100%	25%

6. TÀI LIỆU THAM KHẢO

- [1] TopDev, “MongoDB là gì? Định nghĩa đầy đủ và chi tiết nhất về MongoDB”, 2023. [Online]. Available: <https://topdev.vn/blog/mongodb-la-gi/>
- [2] thanhtv.96, “MongoDB là gì? Cơ sở dữ liệu phi quan hệ”, 2018. [Online]. Available: <https://viblo.asia/p/mongodb-la-gi-co-so-du-lieu-phi-quan-he-bJzKmgoPl9N>
- [3] geeksforgeeks, “MongoDB: An introduction”, [Online]. Available: <http://geeksforgeeks.org/mongodb-an-introduction/>
- [4] Oracle, “What Is MongoDB? An Expert Guide”, 2024. [Online]. Available: <https://www.oracle.com/vn/database/mongodb/>
- [5] mongodb, “Why Use MongoDB and When to Use It?”, [Online]. Available: <https://www.mongodb.com/resources/products/fundamentals/why-use-mongodb>
- [6] ibm, “What is a NoSQL database??”, 2022. [Online]. Available: <https://www.ibm.com/think/topics/nosql-databases>