ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

**KHOA HỆ THỐNG THÔNG TIN**



**BÁO CÁO TIỂU LUẬN MÔN HỌC: CƠ SỞ DỮ LIỆU PHÂN TÁN**

ĐỀ TÀI : Hệ Quản Trị Cơ Sở Dữ Liệu OrientDB

Giảng viên hướng dẫn: Hồ Duy Tri

**Sinh viên thực hiện:**

Tô Thái Duy - 20521247

Nguyễn Văn Khang – 20521438

Trần Duy Khánh – 20521457

Phạm Huy Hùng - 20521372

# **LỜI CẢM ƠN**

Đầu tiên, nhóm tác giả xin gởi lời cảm ơn chân thành đến tập thể quý Thầy Cô Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM và quý Thầy Cô khoa Hệ thống thông tin đã giúp cho nhóm tác giả có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này.

Đặc biệt, nhóm tác giả xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới Thầy Hồ Duy Tri (Giảng viên môn Cơ sở Dữ liệu Phân tán). Thầy đã trực tiếp hướng dẫn tận tình, sửa chữa và đóng góp nhiều ý kiến quý báo giúp nhóm tác giả hoàn thành tốt báo cáo môn học của mình.

Trong thời gian một học kỳ thực hiện đề tài, nhóm tác giả đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, nhóm tác giả vận dụng tối đa những gì đã thu thập được để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, trong quá trình thực hiện, nhóm tác giả không tránh khỏi những thiếu sót. Chính vì vậy, nhóm tác giả rất mong nhận được những sự góp ý từ phía Thầy nhằm hoàn thiện những kiến thức mà nhóm tác giả đã học tập và là hành trang để nhóm tác giả thực hiện tiếp các đề tài khác trong tương lai.

Xin chân thành cảm ơn Thầy!

**MỤC LỤC**

[**LỜI CẢM ƠN** 2](#_Toc122291175)

[**I. Giới thiệu** 4](#_Toc122291176)

[1. OrientDB 4](#_Toc122291177)

[2. Lịch sử hình thành. 5](#_Toc122291178)

[3. Tác giả, tổ chức quản lý 5](#_Toc122291179)

[4. Mô hình lưu trữ (key – value, document, graph, hybrid…) 7](#_Toc122291180)

[5. Ngôn ngữ thao tác với dữ liệu 10](#_Toc122291182)

[6. Cơ chế phân tán (cách phân tán, mô hình phân tán dữ liệu…) 17](#_Toc122291183)

[**II. Hướng dẫn cài đặt:** 18](#_Toc122291184)

[1. Cài đặt OrientDB tại trang chủ 18](#_Toc122291185)

[2. Sau khi tải, khởi chạy server.sh để khởi động server 18](#_Toc122291186)

[3. Khởi tạo server thành công 19](#_Toc122291187)

[4. Truy cập theo đường dẫn 20](#_Toc122291188)

[**III. Thực nghiệm mô phỏng phân tán.** 22](#_Toc122291189)

[1. Mô tả bài toán đặt ra với dữ liệu. 22](#_Toc122291190)

[2. Mô tả cấu trúc dữ liệu sử dụng. 22](#_Toc122291191)

[3. Các bước thực nghiệm chi tiết từ lúc đứa dữ liệu vào đến lúc truy vấn phân tán, kèm theo hình minh hoạ. 22](#_Toc122291192)

[**IV. Tổng kết.** 23](#_Toc122291193)

[1. Kết quả. 23](#_Toc122291194)

[2. Hạn chế. 23](#_Toc122291195)

# **I. Giới thiệu**

## 1. OrientDB

OrientDB là một hệ quản trị cơ sở dữ liệu NoSQL đa mô hình với sự hỗ trợ cho dữ liệu đồ thị và tài liệu, điểm đáng chú ý của OrientDB đó chính là nó được phát triển bằng ngôn ngữ lập trình Java nên có thể được cài đặt trên bất kỳ hệ điều hành nào các bạn muốn, đây là một lợi thế lớn so với các cơ sở dữ liệu NoSQL khác.

Cơ sở dữ liệu OrientDB cũng mang đầy đủ các đặc tính của một cơ sở dữ liệu NoSQL. Nó có một hệ thống lập hồ sơ bảo mật mạnh mẽ dựa trên người dùng và vai trò, đồng thời hỗ trợ truy vấn bằng [Gremlin](https://en.wikipedia.org/wiki/Gremlin_(programming_language)) cùng với [SQL](https://en.wikipedia.org/wiki/SQL) được mở rộng để duyệt đồ thị.

OrientDB sử dụng một số cơ chế lập chỉ mục dựa trên [B-tree](https://en.wikipedia.org/wiki/B-tree) và [Extendible hashing](https://en.wikipedia.org/wiki/Extendible_hashing) , cái cuối cùng được gọi là "hash index", có kế hoạch triển khai các chỉ mục dựa trên chỉ mục cây [LSM-tree](https://en.wikipedia.org/wiki/LSM-tree) và [Fractal .](https://en.wikipedia.org/wiki/Fractal_tree_index) Mỗi bản ghi có [khóa Thay thế](https://en.wikipedia.org/wiki/Surrogate_key) cho biết vị trí của bản ghi bên trong [Danh sách mảng](https://en.wikipedia.org/wiki/Array_list) , các liên kết giữa các bản ghi được lưu trữ dưới dạng một giá trị của vị trí bản ghi được lưu trữ bên trong liên kết giới thiệu hoặc dưới dạng [B-tree](https://en.wikipedia.org/wiki/B-tree) của các vị trí bản ghi (được gọi là record IDs hoặc RIDs) cho phép truyền tải nhanh (với [độ phức tạp O](https://en.wikipedia.org/wiki/Big_O_notation#Orders_of_common_functions) ) của các mối quan hệ một-nhiều và thêm/xóa nhanh các liên kết mới. OrientDB là cơ sở dữ liệu đồ thị phổ biến thứ năm theo bảng xếp hạng cơ sở dữ liệu đồ thị [DB-Engines](https://en.wikipedia.org/wiki/DB-Engines_ranking) .

Điểm mạnh của OrientDB đó chính là tốc độ thực thi, trả về kết quả các truy vấn rất nhanh, giúp tăng tốc độ ứng dụng của các bạn. Nếu các bạn đang thiết kế một ứng dụng với nhiều thao tác lấy dữ liệu hơn so với ghi dữ liệu thì OrientDB là một sự lựa chọn tốt để cải thiện hiệu năng ứng dụng.

Hiện tại, cơ sở dữ liệu OrientDB đã có phiên bản 3 với tên gọi đầy đủ là OrientDB Enterprise Edition 3.0 cho phép các bạn tải về sử dụng miễn phí. Các bạn có thể tìm hiểu thêm cơ sở dữ liệu NoSQL OrientDB tại địa chỉ [https://orientdb.com](https://orientdb.com/).

## 2. Lịch sử hình thành.

OrientDB ban đầu được tác giả bởi [Luca Garulli](https://en.wikipedia.org/w/index.php?title=Luca_Garulli&action=edit&redlink=1) vào năm 2010. Luca đã viết nó dưới dạng Java viết lại lớp liên tục nhanh của cơ sở dữ liệu Orient ODBMS (ban đầu được phát triển bởi [Luca Garulli](https://en.wikipedia.org/w/index.php?title=Luca_Garulli&action=edit&redlink=1) vào năm 1999 bằng C++). Trong những năm 2012–2014, công cụ lưu trữ đã được phát triển lại bởi Andrey Lomakin. Nó có tên mới là "plocal" viết tắt của "cục bộ được phân trang". Tên này ngụ ý rằng công cụ lưu trữ mới dựa trên khái niệm chia nhỏ các tệp dữ liệu theo trang và trang được coi là một đơn vị thay đổi nguyên tử duy nhất. Kể từ năm 2012, dự án được tài trợ bởi [OrientDB LTD](https://en.wikipedia.org/w/index.php?title=OrientDB_LTD&action=edit&redlink=1) (trước đây là Orient Technologies LTD), một công ty vì lợi nhuận với Luca là Giám đốc điều hành và người sáng lập. Vào năm 2013, Andrey Lomakin đã gia nhập công ty với tư cách là kỹ sư trưởng R&D và đồng sở hữu công ty.

Lần đầu tiên từ "đa mô hình" được liên kết với cơ sở dữ liệu là vào ngày 30 tháng 5 năm 2012 tại Cologne, Đức, trong bài phát biểu quan trọng của Luca Garulli " Áp dụng NoSQL – Bước tiếp theo là gì? ". Luca Garulli đã hình dung ra sự phát triển của các sản phẩm NoSQL thế hệ thứ nhất thành các sản phẩm mới với nhiều tính năng hơn có thể được sử dụng bởi nhiều trường hợp sử dụng. OrientDB là sản phẩm đầu tiên bao gồm các tài liệu, biểu đồ, khóa-giá trị, không gian địa lý và mô hình phản ứng trong cùng một sản phẩm, ở cấp độ cốt lõi. Điều này có nghĩa là nhiều mô hình đã được tích hợp vào lõi mà không cần sử dụng các lớp. Vì lý do này, OrientDB là [cơ sở dữ liệu Đa mô hình](https://en.wikipedia.org/wiki/Multi-model_database) "Gốc" .

OrientDB đã được các phương tiện truyền thông đưa tin và là người chiến thắng giải thưởng InfoWorld Bossie 2015.

Vào ngày 15 tháng 9 năm 2017, công ty OrientDB LTD đã được [CallidusCloud](https://en.wikipedia.org/wiki/CallidusCloud) mua lại , một công ty đại chúng được giao dịch trên [NASDAQ](https://en.wikipedia.org/wiki/NASDAQ) .

Vào ngày 30 tháng 1 năm 2018, có thông báo rằng [SAP (công ty)](https://en.wikipedia.org/wiki/SAP_(company)) đã mua lại CallidusCloud với giá 2,4 tỷ USD. Và do đó, OrientDB hiện được hỗ trợ bởi [SAP (công ty)](https://en.wikipedia.org/wiki/SAP_(company)) .

Vào ngày 1 tháng 9 năm 2021, người sáng lập ban đầu [Luca Garulli](https://en.wikipedia.org/w/index.php?title=Luca_Garulli&action=edit&redlink=1) đã rời [SAP (công ty)](https://en.wikipedia.org/wiki/SAP_(company)) và chuyển dự án thành [ArcadeDB](https://en.wikipedia.org/w/index.php?title=ArcadeDB&action=edit&redlink=1) sau khi [SAP](https://en.wikipedia.org/wiki/SAP) quyết định ngừng cung cấp hỗ trợ thương mại cho OrientDB.

## 3. Tác giả, tổ chức quản lý

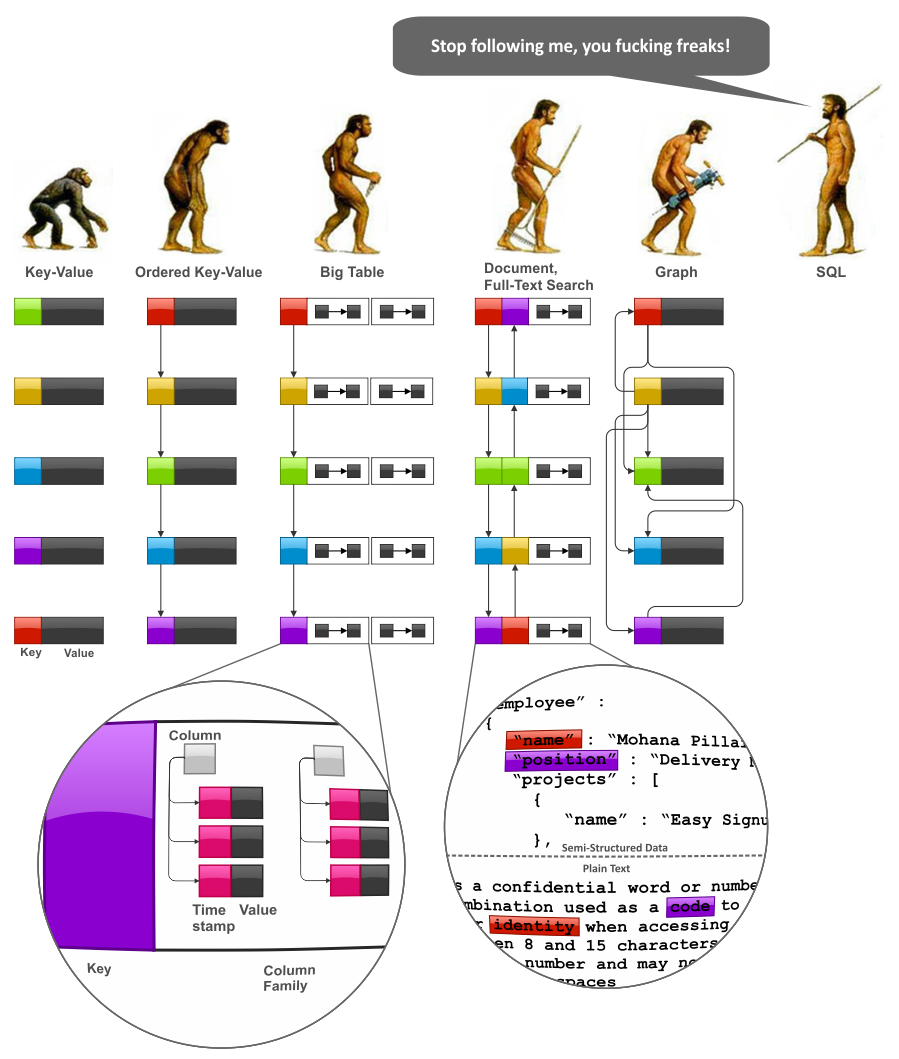
* OrientDB ban đầu được tác giả bởi [Luca Garulli](https://en.wikipedia.org/w/index.php?title=Luca_Garulli&action=edit&redlink=1) vào năm 2010. Luca đã viết nó dưới dạng Java viết lại lớp liên tục nhanh của cơ sở dữ liệu Orient ODBMS (ban đầu được phát triển bởi [Luca Garulli](https://en.wikipedia.org/w/index.php?title=Luca_Garulli&action=edit&redlink=1) vào năm 1999 bằng C++). Trong những năm 2012–2014, công cụ lưu trữ đã được phát triển lại bởi Andrey Lomakin.
* Luca Garulli đã hình dung ra sự phát triển của các sản phẩm NoSQL thế hệ thứ nhất thành các sản phẩm mới với nhiều tính năng hơn có thể được sử dụng bởi nhiều trường hợp sử dụng. OrientDB là sản phẩm đầu tiên bao gồm các tài liệu, biểu đồ, khóa-giá trị, không gian địa lý và mô hình phản ứng trong cùng một sản phẩm, ở cấp độ cốt lõi.

Các công cụ tổ chức quản lý:

* Cài đặt nhanh chóng. OrientDB có thể được cài đặt và chạy trong vòng chưa đầy 60 giây
* Giao dịch đầy đủ: hỗ trợ các giao dịch [ACID](https://en.wikipedia.org/wiki/ACID) đảm bảo rằng tất cả các giao dịch cơ sở dữ liệu được xử lý một cách đáng tin cậy và trong trường hợp xảy ra sự cố, tất cả các tài liệu đang chờ xử lý đều được khôi phục và cam kết.
* [Mô hình dữ liệu có cấu trúc đồ thị](https://en.wikipedia.org/wiki/Graph_database) : quản lý riêng các đồ thị. Hoàn toàn tương thích với [Apache](https://en.wikipedia.org/wiki/Apache_Software_Foundation) TinkerPop Gremlin (trước đây gọi là Bản thiết kế) khung điện toán đồ thị mã nguồn mở.
* [SQL](https://en.wikipedia.org/wiki/SQL) : hỗ trợ các truy vấn SQL có phần mở rộng để xử lý các mối quan hệ mà không cần [tham gia SQL](https://en.wikipedia.org/wiki/SQL_join) , quản lý cây và đồ thị của các tài liệu được kết nối.
* Công nghệ web: vốn hỗ trợ [HTTP](https://en.wikipedia.org/wiki/HTTP) , giao thức [RESTful](https://en.wikipedia.org/wiki/RESTful) và các thư viện hoặc thành phần bổ sung [JSON .](https://en.wikipedia.org/wiki/JSON)
* Phân tán: hỗ trợ đầy đủ cho [sao chép đa chủ](https://en.wikipedia.org/wiki/Multi-master_replication) bao gồm các cụm phân tán theo địa lý.
* Chạy ở mọi nơi: được triển khai bằng [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) thuần túy cho phép nó chạy trên [Linux](https://en.wikipedia.org/wiki/Linux) , [OS X](https://en.wikipedia.org/wiki/OS_X) , [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) hoặc bất kỳ hệ thống nào có [JVM](https://en.wikipedia.org/wiki/Java_virtual_machine) tuân thủ .
* Có thể nhúng: chế độ cục bộ để sử dụng cơ sở dữ liệu bỏ qua Máy chủ. Hoàn hảo cho các kịch bản mà cơ sở dữ liệu được nhúng.
* [Giấy phép Apache 2](https://en.wikipedia.org/wiki/Apache_2_License) : luôn miễn phí cho mọi mục đích sử dụng. Không có phí hoặc tiền bản quyền cần thiết để sử dụng nó.
* Toàn bộ máy chủ có dung lượng khoảng 512 [MB](https://en.wikipedia.org/wiki/Megabyte) .
* Hỗ trợ thương mại có sẵn từ OrientDB.
* So khớp mẫu: Được giới thiệu trong phiên bản 2.2, câu lệnh Match truy vấn cơ sở dữ liệu theo cách khai báo, sử dụng khớp mẫu.
* Các tính năng bảo mật được giới thiệu trong OrientDB 2.2 cung cấp một khung mở rộng để thêm trình xác thực bên ngoài, xác thực mật khẩu, nhập LDAP của vai trò cơ sở dữ liệu và người dùng, khả năng kiểm tra nâng cao và hỗ trợ nhật ký hệ thống. OrientDB Enterprise Edition cung cấp hỗ trợ trình duyệt [SPNEGO xác thực](https://en.wikipedia.org/wiki/SPNEGO) [Kerberos (giao thức)](https://en.wikipedia.org/wiki/Kerberos_(protocol)) đầy đủ . Khi nói đến mã hóa cơ sở dữ liệu, bắt đầu từ phiên bản 2.2, OrientDB có thể mã hóa các bản ghi trên đĩa. Điều này ngăn người dùng trái phép truy cập nội dung cơ sở dữ liệu hoặc thậm chí bỏ qua bảo mật của OrientDB.
* Teleporter: Cho phép nhập nhanh cơ sở dữ liệu quan hệ vào OrientDB chỉ bằng vài bước đơn giản.
* Sẵn sàng cho đám mây: OrientDB có thể được triển khai trên đám mây và hỗ trợ các nhà cung cấp sau: Amazon Web Services, Microsoft Azure, CenturyLink Cloud, Jelastic, DigitalOcean.

## 4. Mô hình lưu trữ (key – value, document, graph, hybrid …)

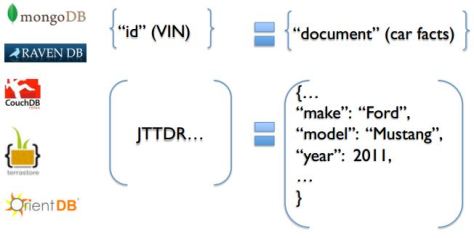
Có nhiều cách phân loại các cơ sở dữ liệu NoSQL khác nhau, mỗi loại với các loại và loại con khác nhau, một số trong số đó có thể chồng chéo lên nhau. Một phân loại cơ bản dựa trên mô hình dữ liệu, với các ví dụ:

* Document Database
* Graph Database
* **Document Database**

### **Giới thiệu**: Mỗi object sẽ được lưu trữ trong database dưới dạng một document. Dữ liệu sẽ được lưu trữ dưới dạng BSON/JSON/XML dưới database. Dữ liệu không schema cứng như SQL, do đó ta **có thể thêm/sửa field, thay đổi table**, … rất nhanh và đơn giản. Database dạng này có tốc độ truy vấn nhanh, có thể thực hiện các câu truy vấn phức tạp, dễ mở rộng (scalability). Mỗi database có **một kiểu truy vấn riêng**, khá là loạn xà ngầu (RavenDB dùng Lucene, MongoDB lại dùng query document).

**Database tiêu biểu**: MongoDB, RavenDB, CouchDB, TerraStone, OrientDB

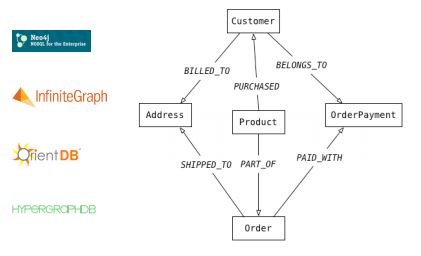
**Ứng dụng**: Do nhanh và linh động, document database thường đóng vài trò làm **database cho các ứng dụng prototype**, **big data**, e-commerce, CMS. Ngoài ra, ta còn dùng nó để lưu log hoặc history.



* **Graph Database**

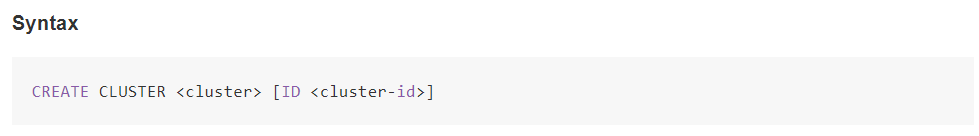
**Giới thiệu**: Dữ liệu trong graph database được lưu dưới dạng các node. Mỗi node sẽ có 1 label, 1 số properties như một row trong SQL. Các node này được kết nối với nhau bằng các relationship. Graph database tập trung nhiều vào relationship giữa các node, áp dụng nhiều thuật toán duyệt node để tăng tốc độ.

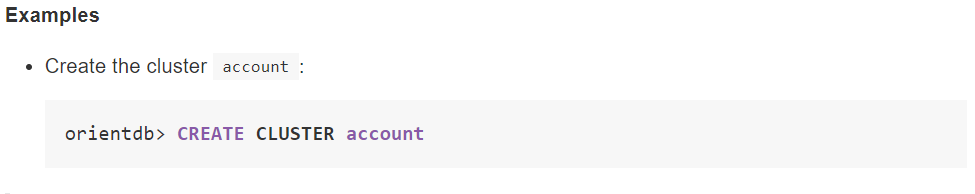
**Database tiêu biểu**: Neo4j, InfiniteGraph, OrientDB, HYPERGRAPHDB

**Ứng dụng**: Khi cần truy vấn các mối quan hệ, graph database **truy vấn nhanh và dễ hơn nhiều** so với database. Nó được dùng trong các hệ thống: mạng nơ ron, chuyển tiền bạc, mạng xã hội (tìm bạn bè), **giới thiệu sản phẩm** (dựa theo sở thích/lịch sử mua sắm của người dùng)…

## 5. Ngôn ngữ thao tác với dữ liệu

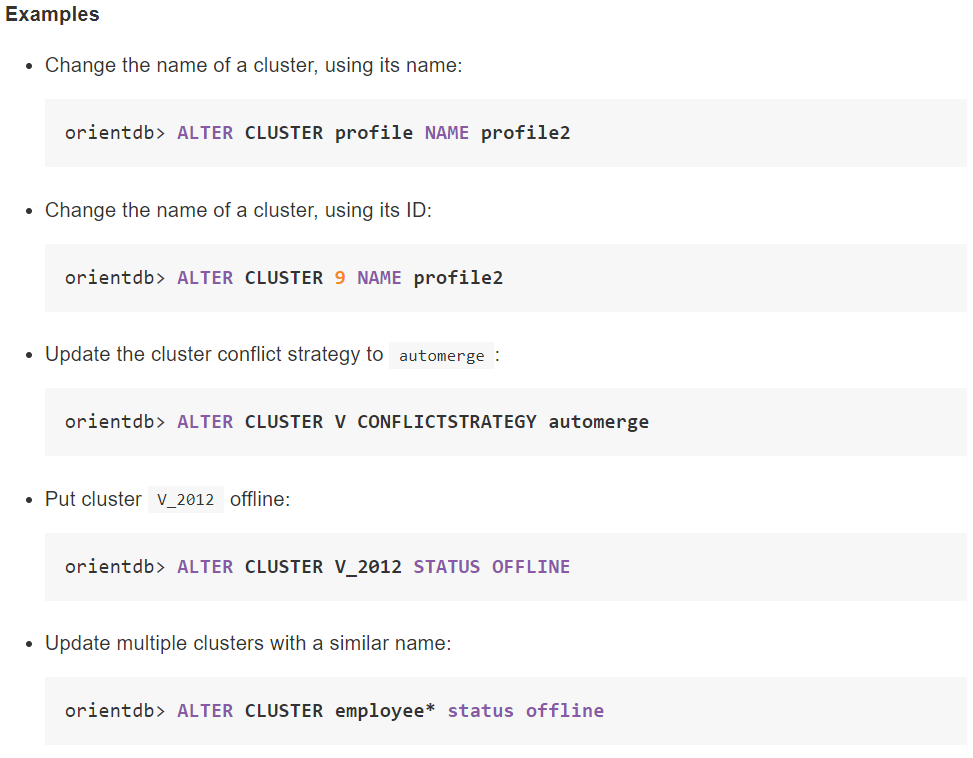
- CLUSTER

+ CREATE:

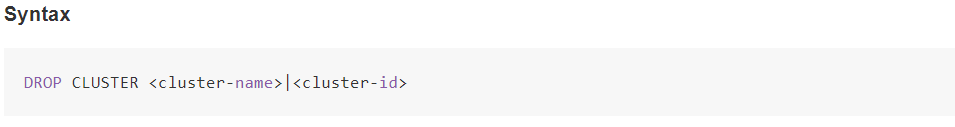
****

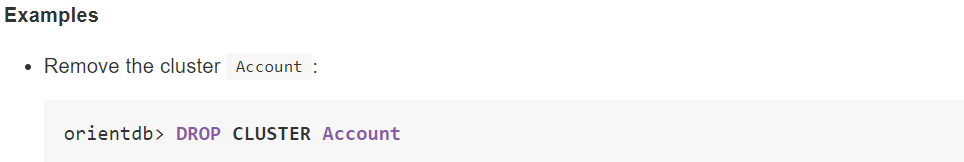
+ ALTER:

****

****

+ DROP:

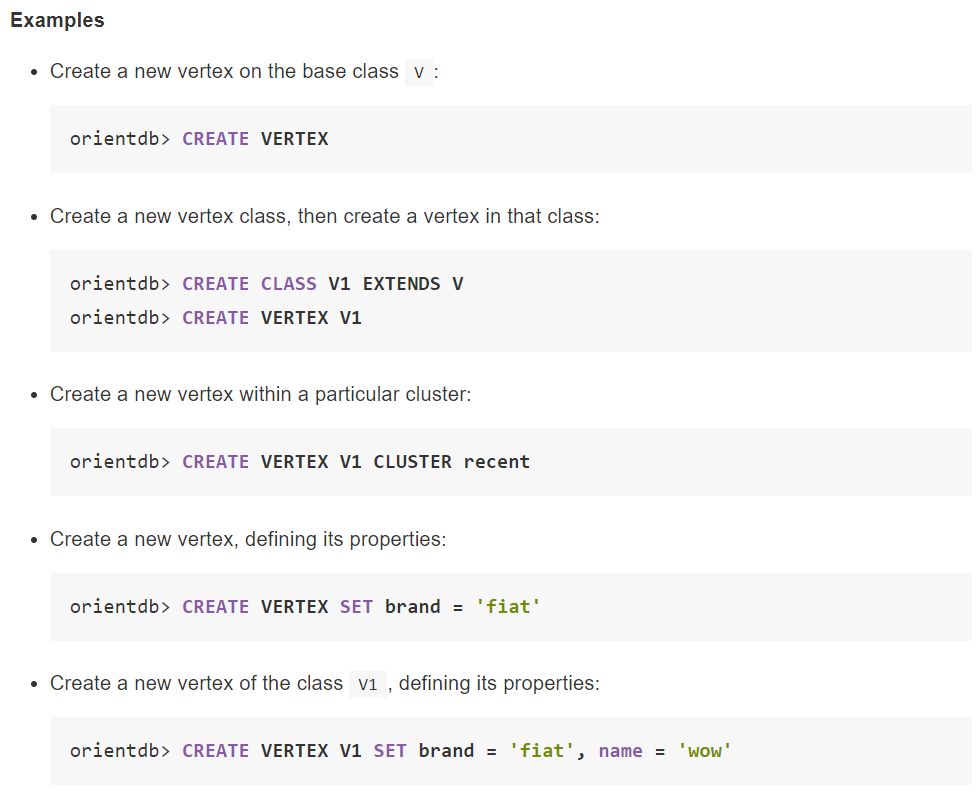
****



* GRAPH

+ CREATE VERTEX:







+ CREATE EDGE:

**Syntax**

CREATE EDGE <class> [CLUSTER <cluster>]

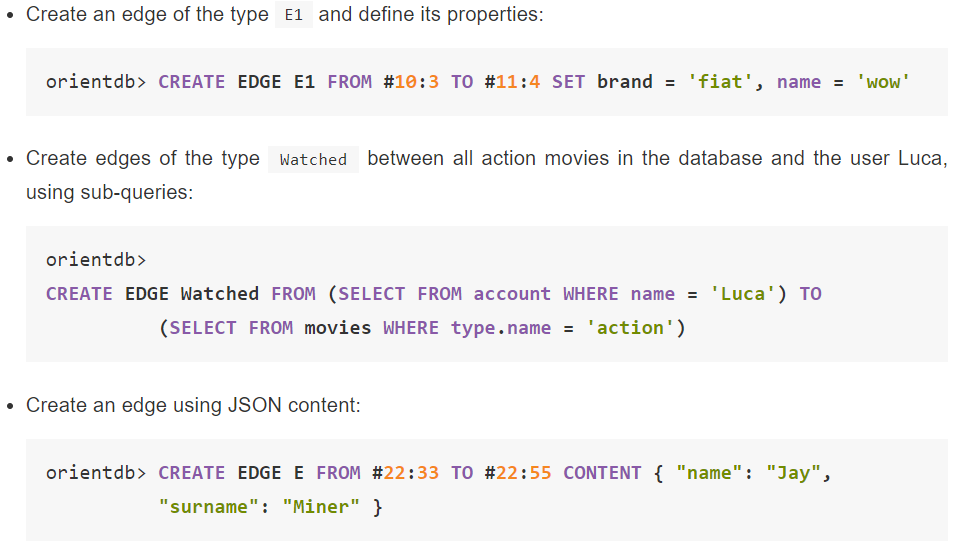
FROM <rid> | ( <query> ) | [ <rid> (, <rid>)\* ]

TO <rid> | ( <query> ) | [ <rid> (, <rid>)\* ]

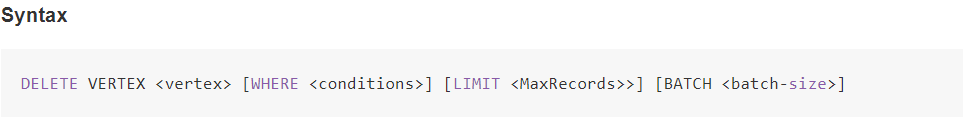
[ SET <field> = <expression>[,]\* ] | CONTENT {<JSON>}

[RETRY <retry> [WAIT <pauseBetweenRetriesInMs]] [BATCH <batch-size>]





+ DELETE VERTEX:





+ DELETE EDGE:

**Syntax**

DELETE EDGE

( <rid>

|

[<rid> (, <rid>)\*]

|

( [ FROM (<rid> | <select\_statement> ) ] [ TO ( <rid> | <select\_statement> ) ] )

|

[<class>]

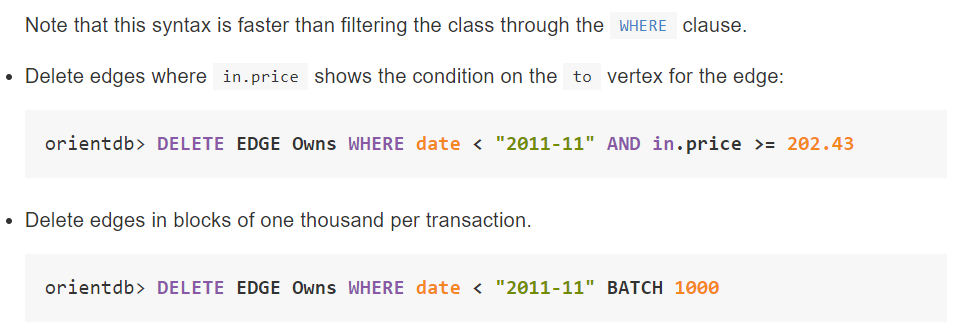
(

[WHERE <conditions>]

[LIMIT <MaxRecords>]

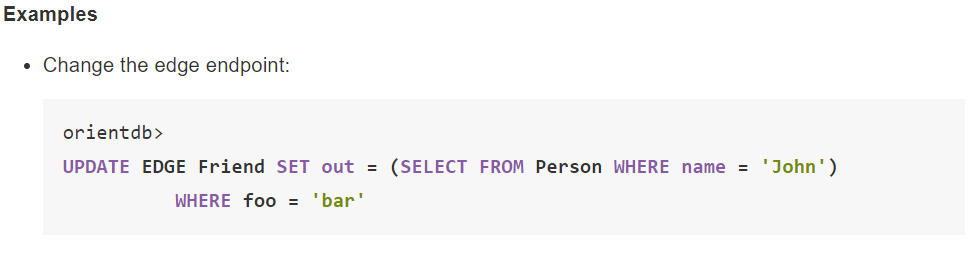
[BATCH <batch-size>]





+ UPDATE EDGE:





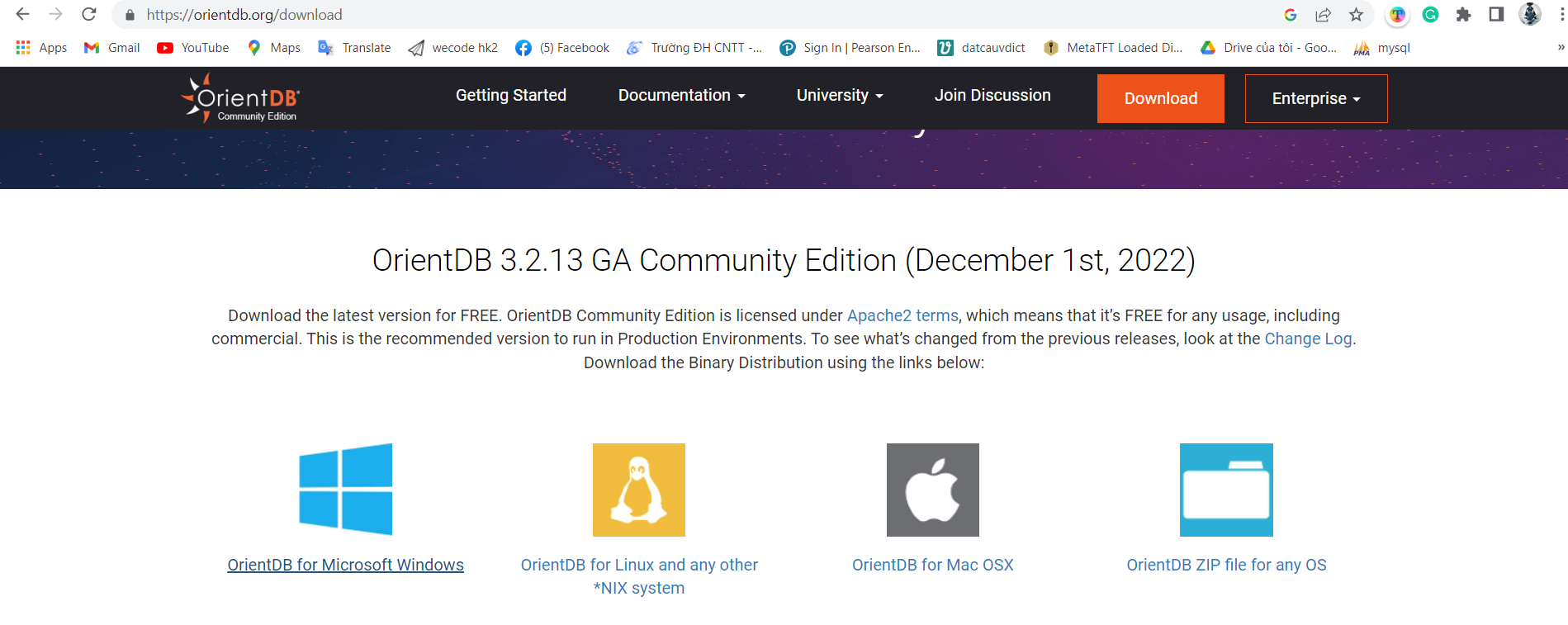
## 6. Cơ chế phân tán (cách phân tán, mô hình phân tán dữ liệu…)

* OrientDB có thể được phân phối trên các máy chủ khác nhau và được sử dụng theo nhiều cách khác nhau để đạt được hiệu suất, khả năng mở rộng và độ mạnh mẽ tối đa.
* OrientDB sử dụng [dự án Hazelcast Open Source](http://www.hazelcast.com/) để tự động khám phá các nút, lưu trữ cấu hình cụm thời gian chạy và đồng bộ hóa các hoạt động nhất định giữa các nút. Một số tài liệu tham khảo trong trang này được liên kết với tài liệu chính thức của Hazelcast để có thêm thông tin về chủ đề đó.
* *LƯU Ý: Khi bạn chạy ở chế độ phân tán, OrientDB cần thêm RAM. Tối thiểu là 2GB heap, nhưng chúng tôi khuyên bạn nên sử dụng ít nhất 4GB bộ nhớ heap. Để thay đổi đống, hãy sửa đổi cài đặt bộ nhớ Java trong thùng / máy chủ file.sh (hoặc server.bat trên Windows).*
* *Vai trò phân tán:*
* *OrientDB có kiến trúc phân tán đa chủ (còn được gọi là "master-less") nơi mỗi máy chủ có thể đọc và ghi. Bắt đầu từ v2.1, OrientDB hỗ trợ vai trò của "REPLICA", trong đó máy chủ ở chế độ chỉ đọc, chỉ chấp nhận các lệnh Đọc và Truy vấn. Hơn nữa, khi máy chủ tham gia cụm phân tán dưới dạng "REPLICA",* các cụm bản ghi riêng không được tạo giống như các nút "MASTER".

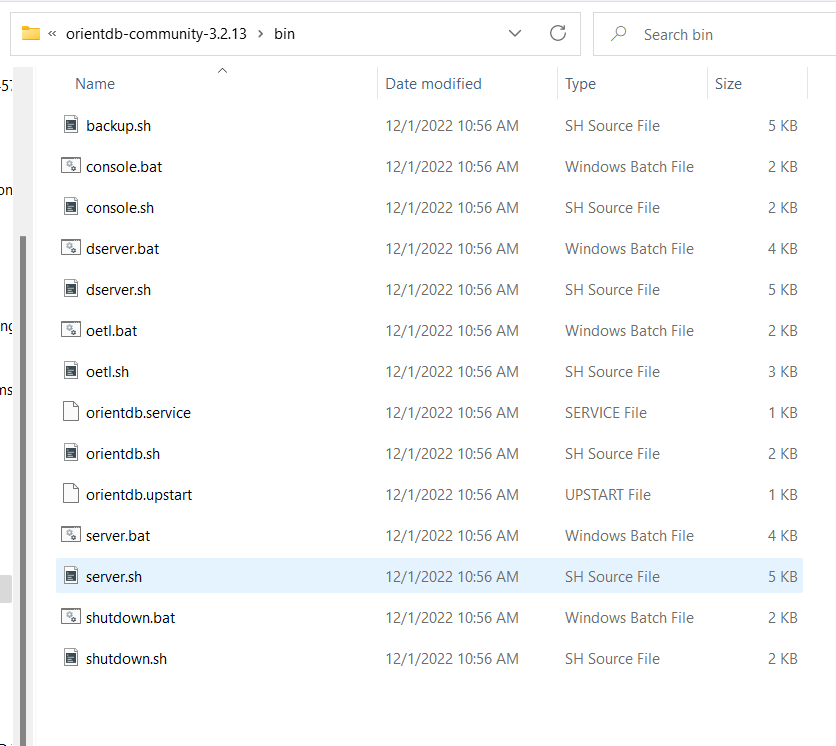
# **II. Hướng dẫn cài đặt:**

* Cài đặt trên một máy:

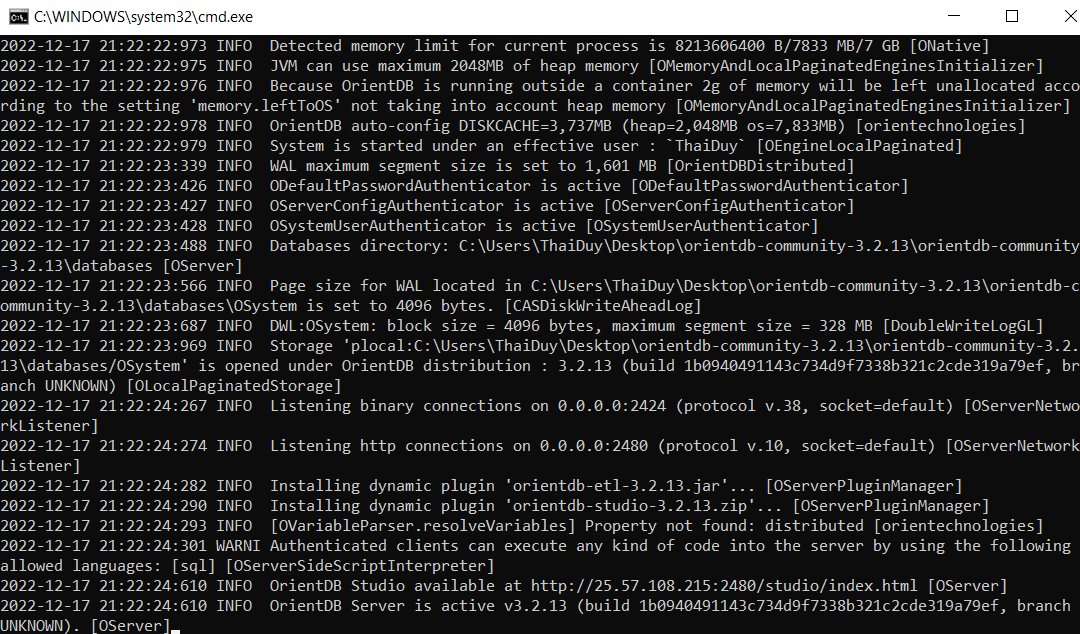
## Cài đặt OrientDB tại trang chủ: https://orientdb.org/download



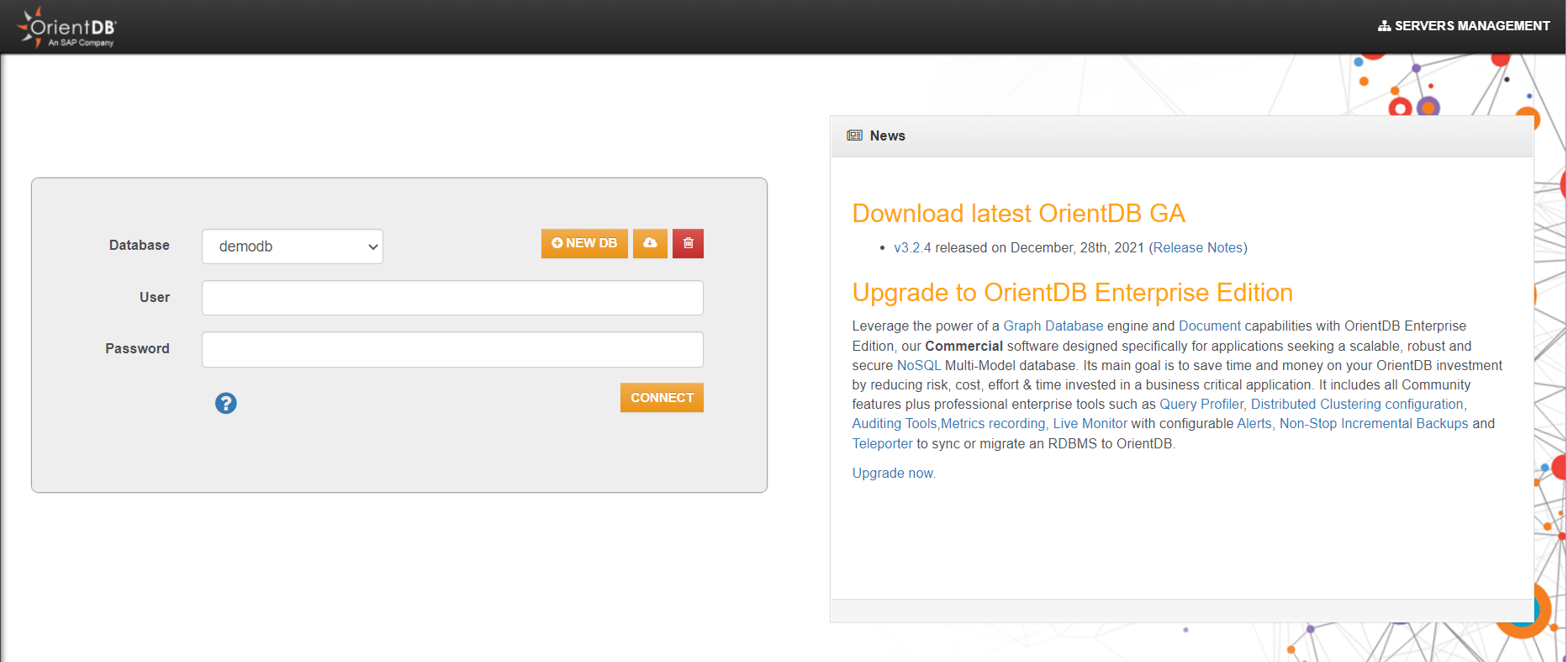
1. Sau khi tải, khởi chạy server.sh để khởi động server (C:\Users\orientdb-community-3.2.13\orientdb-community-3.2.13\bin)



## Khởi tạo server thành công



## Truy cập theo đường dẫn : @ip-address:port để vào trình duyệt của OrientDB



* Cài đặt trên cụm máy phân tán:

Có hai cách để chia sẻ dữ liệu trên nhiều máy chủ:

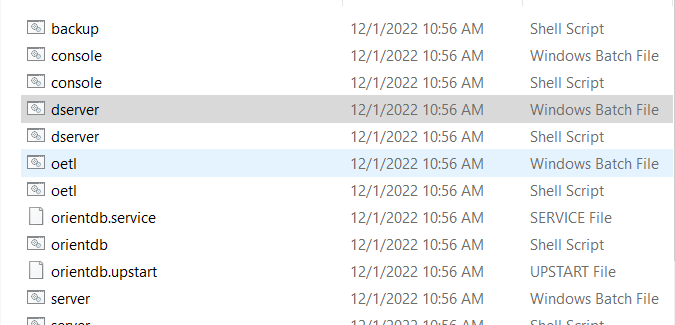
Cách 1: Trước khi khởi động, sao chép thư mục chứa cơ sở dữ liệu vào tất cả các máy chủ.( $ORIENTDB\_HOME/database)

Cách 2: Giữ cơ sở dữ liệu trên máy chủ đầu tiên, sau đó khởi động các máy chủ khác. Trong các cấu hình mặc định, OrientDB sẽ tự động chia sẻ cơ sở dữ liệu với các máy chủ tham gia.

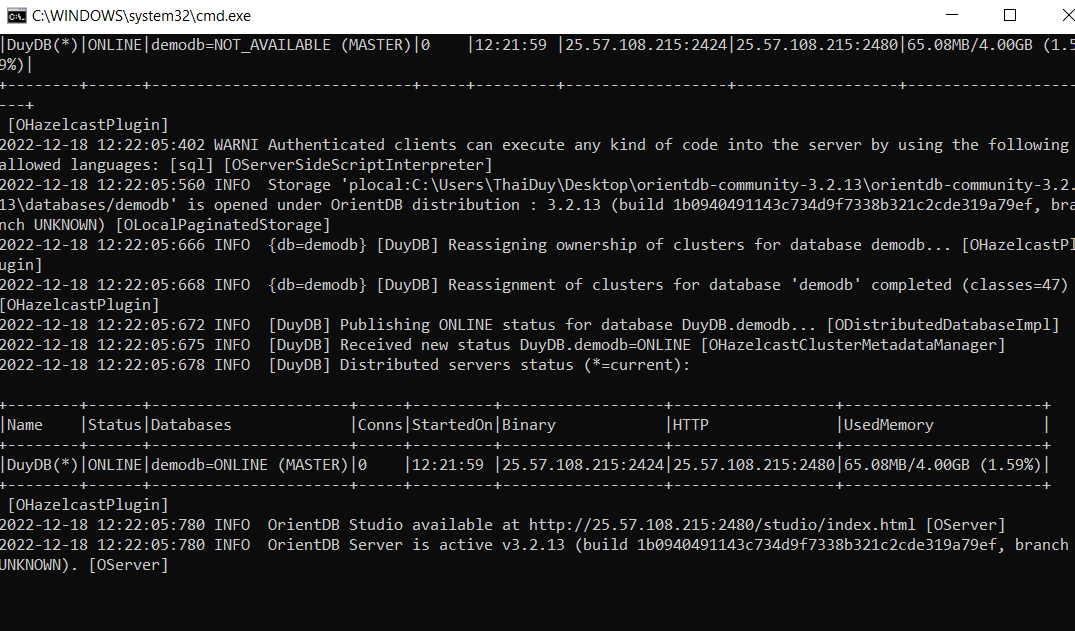
LƯU Ý: Khi bạn chạy ở chế độ phân tán, OrientDB cần thêm RAM. Tối thiểu là 2GB heap, nhưng chúng tôi khuyên bạn nên sử dụng ít nhất 4GB bộ nhớ heap. Để thay đổi heap sửa đổi cài đặt bộ nhớ Java trong tệp bin/dserver.sh (hoặc dserver.bat trên Windows).

Bước 1 : Khởi tạo máy chủ đầu tiên

Truy cập vào thư mục bin tìm để dserver.bat để khởi động máy chủ phân tán đầu tiên:



Bước 2: Khởi chạy và đặt tên cho máy chủ



Bước 3: Tiến hành thêm các máy chủ khác

* Chỉnh sửa trong file hazelcast.xml để tiến hành thêm các máy chủ khác.



* Hoặc sao chép thư mục cài đặt OrientDB từ nút đầu tiên sang từng nút tiếp theo $ **scp user@ip\_address $ORIENTDB\_HOME**
* Điều này sao chép cả cơ sở dữ liệu và tệp cấu hình của chúng vào nút máy chủ phân tán mới.

Bước 4: Kết nối các máy chủ với nhau

Sau khi tiến hành thêm máy chủ chạy câu lệnh kết nối trong console:

***CONNECT remote:localhost/database username password***

# **III. Thực nghiệm mô phỏng phân tán.**

1. Mô tả bài toán đặt ra với dữ liệu.
2. Mô tả cấu trúc dữ liệu sử dụng.
3. Các bước thực nghiệm chi tiết từ lúc đứa dữ liệu vào đến lúc truy vấn phân tán, kèm theo hình minh hoạ.

# **IV. Tổng kết.**

## Kết quả.

* Kết nối cơ sở dữ liệu ở thiết bị thành công.
* Kết nối ổn định, duy trì tốt trong suốt quá trình truy vấn.
* Toàn vẹn dữ liệu khi truy vấn ở các máy chủ khác.
* Quá trình phân tán sang máy chủ phân tán mới diễn ra ổn định.
* OrientBD hoạt động tốt, tốc độ thực thi cao, trả về kết quả truy vấn nhanh và không vượt quá bộ nhớ cho phép trong quá trình phân tán.
* Nhóm đã thu được kiến thức quan trong trong việc phân tán cơ sở dữ liệu bằng hệ quản trị cơ sở dữ liệu OrientDB NoSQL.

1. Hạn chế.

* Quá trình kết nối còn gặp nhiều khó khăn
* Kết nối ở 2 máy khác nhau nên khác mạng cần dùng công cụ để hỗ trợ
* Các thao tác với cơ sở dữ liệu ban đầu còn khá mới nên khá bất tiện
* Laptop không đáp ứng đủ nhu cầu của ứng dụng, hay tắt máy và bị sập nguồn.