TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN GIỮA KỲ

MÔN HỌC BẢO MẬT CƠ SỞ DỮ LIỆU

ĐỀ TÀI:

TÌM HIỂU CƠ SỞ DỮ LIỆU NOSQL VÀ DEMO CƠ CHẾ BẢO MẬT TRÊN ORIENTDB

Mã lớp học phần: DBSE431284_24_1_02CLC

GVHD: Ths. Lê Thị Minh Châu

Sinh viên thực hiện (nhóm 06):

Nguyễn Phú Thành	21110299
Cao Thị Ngọc Phụng	21110276
Trần Văn Tiến	21110319
Nguyễn Văn Hào	21110175

TP Hồ Chí Minh, tháng 11 năm 2024

DANH SÁCH SINH VIÊN THAM GIA

Mã học phần: DBSE431284_24_1_02CLC

Nhóm: 06

Tên đề tài: Tìm hiểu cơ sở dữ liệu NoSQL và demo các cơ chế bảo mật của OrientDB

STT	HỌ VÀ TÊN THÀNH VIÊN	MÃ SỐ SINH VIÊN	TỶ LỆ THAM GIA
1	Nguyễn Phú Thành	21110299	100%
2	Cao Thị Ngọc Phụng	21110276	100%
3	Trần Văn Tiến	21110319	100%
4	Nguyễn Văn Hào	21110175	100%

Ghi chú:

Tỷ lệ %: Mức độ phần trăm hoàn thành của từng sinh viên tham gia.

Trưởng nhóm: Nguyễn Phú Thành

Điểm số: …….

Nhận xét của giáo viên

...……

……....

……

……

……....

……....

……....

……....

PHỤ LỤC

1. BẢNG KẾ HOẠCH LÀM VIỆC VÀ PHÂN CÔNG NHIỆM VỤ

STT	Nội dung công việc	Thời gian	Thành viên thực hiện	
			Họ tên	MSSV
1	Tìm hiểu lý thuyết Server Security Cài đặt OrientDB Demo Server Securiy trên OrientDB, Disable Default User		Nguyễn Phú Thành	21110299
2	Tìm hiểu Database Security Cài đặt OrientDB Demo Tạo user, role, cấp quyền		Cao Thị Ngọc Phụng	21110276
3	Tìm hiểu lý thuyết SSL Connection Cài đặt OrientDB Demo tạo user, role, phân quyền và bảo mật cấp bảng ghi trên OrientDB		Trần Văn Tiến	21110319
4	Tìm hiểu lý thuyết Database Encryption Cài đặt OrientDB Demo user, role, cấp quyền, encryption		Nguyễn Văn Hào	21110175

LỜI CẢM ƠN

Lời đầu tiên, nhóm em xin gửi lời cảm ơn chân thành nhất đến ***Cô Lê Thị Minh Châu*** – giảng viên bộ môn ***Bảo mật Cơ sở dữ liệu*** của chúng em. Trong quá trình học tập và tìm hiểu bộ môn, chúng em đã nhận được sự quan tâm giúp đỡ, hướng dẫn rất tận tình và tâm huyết từ Cô. Cô đã giúp chúng em tích lũy thêm nhiều kiến thức để có cái nhìn sâu sắc và hoàn thiện hơn trong lĩnh vực Công nghệ thông tin. Để từ đó, ứng dụng những kiến thức mà Cô truyền tải, nhóm em xin trình bày lại những gì mà mình đã học hỏi được thông qua việc thực hiện đề tài "***Tìm hiểu cơ sở dữ liệu NoSQL và demo các cơ chế bảo mật của OrientDB"***

Kiến thức là vô hạn và sự tiếp nhận kiến thức của bản thân mỗi người luôn tồn tại những hạn chế nhất định. Do đó, trong phạm vi khả năng của bản thân, nhóm em đã rất cố gắng để hoàn thành đề tài một cách tốt nhất. Tuy nhiên, chắc chắn không tránh khỏi những thiếu sót, nhóm chúng em rất mong nhận được sự cảm thông và những ý kiến đóng góp đến từ Cô để đề tài của nhóm em được hoàn thiện hơn.

Một lần nữa, ***nhóm em xin chân thành cảm ơn Cô*** đã tận tình hướng dẫn, chỉ bảo các thành viên nhóm em trong suốt quá trình học tập và thực hiện đồ án này.

Kính chúc Cô sức khỏe, hạnh phúc thành công trên con đường sự nghiệp giảng dạy.

Trân trọng

Đại diện nhóm

Nguyễn Phú Thành

MỤC LỤC

PHỤ LỤC

PHẦN MỞ ĐẦU *1*

1. Lời mở đầu *1*

2. Mục đích, nhiệm vụ của đề tài *1*

3. Phạm vi thực hiện đề tài *1*

CHƯƠNG 1: TÌM HIỂU VỀ CƠ SỞ DỮ LIỆU NOSQL *2*

1.1 Khái niệm cơ sở dữ liệu NoSQL *2*

1.2 Loại cơ sở dữ liệu NoSQL *4*

1.3 Đặc điểm của cơ sở dữ liệu NoSQL *5*

1.4 Nhược điểm của cơ sở dữ liệu NoSQL *6*

1.5 Sự lựa chọn và ứng dụng của cơ sở dữ liệu NoSQL *6*

CHƯƠNG 2: TÌM HIỂU CÁC CƠ CHẾ BẢO MẬT TRONG ORIENTDB *8*

2.1 Bảo mật Server *8*

2.1.1 Quản lý User Server 9

2.1.2 Tài nguyên Server 10

2.2 Bảo mật cơ sở dữ liệu *12*

2.2.1 User 12

2.2.2 Roles 14

2.2.3 Modes 17

2.2.4 Security Policies 18

2.2.5 Resources 20

2.3 Mã hóa Cơ sở dữ liệu *21*

2.4 Kết nối SSL ***24***

2.4.1 Secure SSL connections 24

2.4.2 Cấu hình Client để Sử dụng SSL 29

CHƯƠNG 3: DEMO CƠ CHẾ BẢO MẬT TRÊN ORIENTDB ***33***

3.1 Demo Server Security ***33***

3.1.1 User server 33

3.1.2. User database 41

3.1.3. Disable Default user 42

3.2 Demo Front Office Manager ***43***

3.2.1 Kết nối database 43

3.2.2 Tạo Class 44

3.2.3. Tạo role 48

3.2.4. Tạo user 51

3.2.5. Kiểm tra lại các user 53

3.3 Demo Quản lý nhân viên ***60***

3.3.1. Tạo class và thêm bản ghi vào class 60

3.3.2. Tạo role 63

3.3.3 Tạo các user 64

3.3.4. Cấp các quyền thao tác trên class cho role 64

3.3.5. Cấp các quyền cho Role để đăng nhập 65

3.3.6. Kiểm tra kết quả 66

3.4 Demo Customer Staff Service ***70***

3.4.1 Kết nối vào cơ sở dữ liệu HotelDB 70

3.4.2 Bảo mật cao ở bảng Employee cho nhân viên chỉ xem chính mình 70

3.4.3 Tạo Các Lớp và Thuộc Tính 72

3.4.4 Tạo Các Role và Cấp Quyền 74

3.4.5 Tạo User và Gán Quyền Truy Cập 77

3.4.6 Thêm Thông Tin Khách Hàng, Booking, và Feedback 78

3.4.7 Cấp Quyền Truy Cập Bổ Sung cho Nhân Viên emp_an 80

3.4.8 Kiểm Tra Các Quyền Của customer_user 84

PHẦN KẾT LUẬN ***88***

1. Ưu điểm ***88***

2. Nhược điểm ***88***

3. Hướng phát triển ***88***

TÀI LIỆU THAM KHẢO

PHẦN MỞ ĐẦU

1. Lời mở đầu

Với sự bùng nổ của dữ liệu lớn và sự đa dạng của các loại dữ liệu, các cơ sở dữ liệu NoSQL như OrientDB đã trở thành một lựa chọn phổ biến. Khác với các CSDL quan hệ truyền thống, NoSQL cung cấp khả năng linh hoạt cao trong việc lưu trữ và quản lý dữ liệu có cấu trúc không cố định. OrientDB, với mô hình dữ liệu đồ thị mạnh mẽ, đặc biệt phù hợp với các ứng dụng đòi hỏi quản lý mối quan hệ giữa các thực thể. Đề tài này sẽ tập trung vào việc tìm hiểu cấu trúc dữ liệu, cơ chế truy vấn và đặc biệt là các tính năng bảo mật của OrientDB. Qua đó, giúp người đọc hiểu rõ hơn về vai trò của OrientDB trong việc xử lý dữ liệu lớn và đưa ra những lựa chọn phù hợp trong các dự án thực tế.

2. Mục đích, nhiệm vụ của đề tài

Hiểu rõ về cơ sở dữ liệu NoSQL. Cài đặt và cấu hình môi trường cho OrientFB. Thực hành các cơ chế bảo mật của OrientDB

3. Phạm vi thực hiện đề tài

Đề tài này thực hiện chủ yếu các lý thuyết cơ bản về OrientDB, cài đặt OrientDB và thực hành các cơ chế bảo mật trong OrientDB

CHƯƠNG 1: TÌM HIỂU VỀ CƠ SỞ DỮ LIỆU NOSQL

1.1 Khái niệm cơ sở dữ liệu NoSQL

Cơ sở dữ liệu NoSQL (tên gốc là "Non SQL" (phi SQL) hoặc "non relational" (phi quan hệ)) cung cấp một cơ chế để lưu trữ và truy xuất dữ liệu được mô hình hóa khác với các quan hệ bảng được sử dụng trong các cơ sở dữ liệu kiểu quan hệ.

Vào năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSQL trong một hội thảo về cơ sở dữ liệu nguồn mở phân tán. Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ database mới: distributed (phân tán) + non-relational (không ràng buộc). Đây là 2 đặc tính quan trọng nhất.

Sự khác biệt giữa NoSQL và SQL

	SQL Database	NoSQL Database
Lịch sử	Phát triển vào những năm 70 với làn sóng ứng dụng lưu trữ dữ liệu đầu tiên	Phát triển vào những năm 2000 để giải quyết những hạn chế của CSDL SQL, đặc biệt liên quan đến quy mô, nhân rộng và lưu trữ dữ liệu phi cấu trúc
Hệ quản trị CSDL đại diện	MySQL, Postgress, Oracle Database	MogoDB, Cassandra, HBase, Neo4j
Mô hình dữ liệu	Các bản ghi riêng lẻ được lưu trữ dưới dạng các hàng trong bảng, với mỗi cột lưu trữ một phần dữ liệu cụ thể về bản ghi đó. Các kiểu dữ liệu riêng biệt được lưu trữ trong các bảng riêng biệt và sau đó được nối với nhau khi các truy vấn	Khác nhau dựa trên loại cơ sở dữ liệu NoSQL. Ví dụ: key – value store hoạt động tương tự như CSDL SQL, nhưng chỉ có 2 cột (“Khóa - key” và “giá trị - value”). Document database loại bỏ hoàn toàn mô hình table – and – row, lưu trữ tất cả dữ liệu có

	phức tạp hơn được thực thi.	liên quan với nhau tạo một "document" duy nhất trong JSON, XML hoặc định dạng khác, có thể lồng các giá trị theo thứ bậc.
Khả năng mở rộng	Theo chiều dọc, có nghĩa là máy chủ duy nhất phải ngày càng mạnh mẽ để đáp ứng nhu cầu mở rộng của dữ liệu. Có thể mở rộng CSDL SQL trên nhiều máy chủ, nhưng cần bổ sung thêm kỹ thuật quan trọng.	Theo chiều ngang, có nghĩa là để thêm dung lượng, người quản trị CSDL chỉ cần thêm nhiều máy chủ hoặc đám mây. CSDL noSQL tự động phân tán dữ liệu trên các máy chủ khi cần thiết.
Mô hình phát triển	Kết hợp mã nguồn mở (ví dụ: Postgress, MySQL) và mã nguồn đóng (ví dụ: CSDL Oracle)	Mã nguồn mở
Thao tác dữ liệu	Ngôn ngữ cụ thể bằng cách sử dụng các câu lệnh Select, Insert, Update.	Thông qua các API hướng đối tượng
Tính nhất quán	Có tính nhất quán mạnh mẽ	Phụ thuộc vào hệ thống. Có hệ thống ưu tiên cung cấp tính nhất quán (ví dụ: MongoDB) trong khi một số khác cung cấp tính nhất quán cuối cùng (ví dụ: canssandra)

Lược đồ	Cấu trúc và kiểu dữ liệu được cố định trước. Để lưu trữ thông tin về một mục dữ liệu mới, toàn bộ CSDL phải được thay đổi, trong thời gian đó CSDL phải được ngoại tuyến	Các bản ghi có thể thêm thông tin mới một cách nhanh chóng, không giống như các hàng trong bảng SQL, dữ liệu khác nhau có thể được lưu trữ cùng nhau khi cần thiết. Đối với một số CSDL việc thêm các trường mới một cách linh hoạt sẽ khó khăn hơn

1.2 Loại cơ sở dữ liệu NoSQL

CSDL NoSQL được chia thành nhiều loại dựa trên cách lưu trữ và tổ chức dữ liệu. Dưới đây là một số loại CSDL NoSQL phổ biến:

Key – Value Stores: lưu trữ dữ liệu dưới dạng cặp khóa – giá trị. Mỗi đối tượng được lưu trữ dưới dạng một khóa duy nhất và giá trị tương ứng. Các ví dụ về loại CSDL này như là Redis, DynamoDB, ….

Document Stores: lưu trữ dữ liệu dưới dạng tài liệu có cấu trúc linh hoạt như JSON hoặc XML. Mỗi tài liệu được lưu trữ là một đối tượng độc lập, và dữ liệu trong tài liệu có thể được truy xuất và cập nhật một cách linh hoạt. Ví dụ: MongoDB và CouchBase, …

Column – Family Stores: lưu trữ dữ liệu dưới dạng cột thay vì dòng như trong CSDL quan hệ. Các cột có thể được nhóm lại thành các family tùy theo nhu cầu: Ví dụ: HBase, Cassandra, ….

Graph database: lưu trữ liệu dưới dạng các đỉnh (vertex) và các cạnh (edge) để biểu dữ quan hệ giữa các đối tượng. CSDL đồ thị phù hợp cho việc phân tích quan hệ phức tạp và xử lý các loại truy vấn liên quan đến mạng lưới. Ví dụ: Neo4j , ….

Ngoài ra, còn có một số loại dữ liệu NoSQL khác như Time – series database, spatial databases (CSDL địa lý), Memory – based databases (CSDL trên bộ nhớ). Mỗi loại

CSDL NoSQL có ứng dụng và đặc điểm riêng biệt, phù hợp với các trường hợp sử dụng và yêu cầu khác nhau.

1.3 Đặc điểm của cơ sở dữ liệu NoSQL

Mô hình dữ liệu linh hoạt: Cơ sở dữ liệu NoSQL cho phép lưu trữ dữ liệu không có cấu trúc hoặc có cấu trúc linh hoạt. Điều này cho phép lưu trữ và xử lý dữ liệu đa dạng, thay đổi và không theo mô hình quan hệ truyền thống. Các loại cơ sở dữ liệu NoSQL như Document Stores và Key-Value Stores cho phép lưu trữ dữ liệu dưới dạng tài liệu và cặp khóa-giá trị một cách linh hoạt.

Khả năng mở rộng ngang: Cơ sở dữ liệu NoSQL được thiết kế để mở rộng ngang (horizontal scalability). Điều này có nghĩa là có thể thêm các máy chủ và nút vào hệ thống để tăng cường khả năng lưu trữ và xử lý dữ liệu. Khả năng mở rộng ngang giúp cơ sở dữ liệu NoSQL xử lý lưu lượng truy cập lớn và khối lượng dữ liệu tăng dần một cách hiệu quả.

Hiệu suất và tốc độ truy vấn cao: Cơ sở dữ liệu NoSQL thường được tối ưu hóa để đạt hiệu suất và tốc độ truy vấn cao. Các cơ sở dữ liệu NoSQL thường có cấu trúc lưu trữ và cách truy xuất dữ liệu khác nhau so với cơ sở dữ liệu quan hệ, cho phép xử lý dữ liệu nhanh chóng và hiệu quả hơn.

Hỗ trợ cho các mô hình dữ liệu đặc biệt: Các loại cơ sở dữ liệu NoSQL như Graph Databases (đồ thị) và Time-Series Databases (chuỗi thời gian) được thiết kế để hỗ trợ các mô hình dữ liệu đặc biệt. Điều này cho phép xử lý và truy vấn dữ liệu phức tạp, như phân tích quan hệ đồ thị, theo dõi dữ liệu thời gian thực và nhiều ứng dụng khác.

Tính mở và cộng đồng phát triển: Các cơ sở dữ liệu NoSQL thường có tính mở và hỗ trợ từ cộng đồng phát triển mạnh mẽ. Điều này đồng nghĩa với việc có sự phát triển và cải tiến liên tục, sự linh hoạt và khả năng tuỳ chỉnh cao cho phù hợp với nhu cầu và yêu cầu cụ thể của ứng dụng.

Tuy nhiên, cũng cần lưu ý rằng cơ sở dữ liệu NoSQL không phải là lựa chọn tốt

nhất cho mọi trường hợp sử dụng. Việc chọn cơ sở dữ liệu nên dựa trên yêu cầu của dự án và tính chất đặc thù của dữ liệu.

1.4 Nhược điểm của cơ sở dữ liệu NoSQL

Khả năng truy vấn phức tạp hạn chế: hạn chế trong việc hỗ trợ các truy vấn phức tạp và các tính năng như khóa ngoại và transaction.

Thiếu chuẩn hóa và công cụ hỗ trợ: tạo khó khăn trong việc quản lý và vận hành cơ sở dữ liệu.

Độ phức tạp trong việc tích hợp với các hệ thống hiện có: trong trường hợp tích hợp với các hệ thống hiện có hoặc các ứng dụng sử dụng cơ sở dữ liệu quan hệ, việc chuyển đổi sang cơ sở dữ liệu NoSQL có thể gặp khó khăn và tốn thời gian.

Hạn chế trong việc thực hiện các giao transction: Một số loại cơ sở dữ liệu NoSQL không hỗ trợ giao transction ACID (Atomicity, Consistency, Isolation, Durability) như cơ sở dữ liệu quan hệ. Điều này có thể là vấn đề đối với các ứng dụng yêu cầu tính toàn vẹn dữ liệu cao.

1.5 Sự lựa chọn và ứng dụng của cơ sở dữ liệu NoSQL

Ứng dụng web và di động: Cơ sở dữ liệu NoSQL thường được sử dụng trong các ứng dụng web và di động, nơi cần lưu trữ và truy vấn dữ liệu theo mô hình phi quan hệ hoặc có tính mở rộng ngang cao. Ví dụ, MongoDB thường được sử dụng để lưu trữ thông tin người dùng, dữ liệu sản phẩm, bài viết, và các thông tin có tính chất tài liệu.

Phân tích dữ liệu lớn (Big Data): Cơ sở dữ liệu NoSQL thích hợp cho việc lưu trữ và xử lý dữ liệu lớn và phân tán. Với khả năng mở rộng ngang và hiệu suất cao, các hệ quản lý cơ sở dữ liệu NoSQL như Cassandra và HBase được sử dụng trong các hệ thống xử lý dữ liệu lớn, phân tích dữ liệu thời gian thực, và lưu trữ log.

Mạng xã hội và đồ thị: Cơ sở dữ liệu NoSQL đồ thị như Neo4j được sử dụng để lưu trữ và truy vấn dữ liệu mạng xã hội, mạng quan hệ, và các dữ liệu có cấu trúc phức tạp. Các loại cơ sở dữ liệu NoSQL đồ thị giúp tối ưu hóa việc truy vấn quan hệ

giữa các đối tượng và tính toán các thông tin liên kết.

Caching và xử lý sự kiện (Event-driven): Cơ sở dữ liệu NoSQL key-value như Redis thường được sử dụng để làm lớp cache trung gian để cải thiện hiệu suất truy vấn. Ngoài ra, Redis cũng được sử dụng trong các ứng dụng xử lý sự kiện thời gian thực, như xử lý hàng đợi (queue), đếm lượt thích (likes), và hệ thống chat.

IoT và dữ liệu thời gian thực: Với tính năng lưu trữ và truy vấn dữ liệu thời gian thực, cơ sở dữ liệu NoSQL thường được sử dụng trong các ứng dụng IoT (Internet of Things) và dữ liệu sensor. Các hệ quản lý cơ sở dữ liệu NoSQL như Cassandra và InfluxDB hỗ trợ lưu trữ và xử lý dữ liệu thời gian thực từ các thiết bị IoT.

CHƯƠNG 2: TÌM HIỂU CÁC CƠ CHẾ BẢO MẬT TRONG ORIENTDB

2.1 Bảo mật Server

Các máy chủ OrientDB riêng lẻ có thể quản lý nhiều cơ sở dữ liệu cùng một lúc và mỗi cơ sở dữ liệu có thể có nhóm người dùng riêng. Khi sử dụng OrientDB thông qua giao thức HTTP, máy chủ OrientDB sử dụng một vùng cho mỗi cơ sở dữ liệu.

Mặc dù người dùng và mật khẩu mặc định vẫn ổn khi bạn đang thiết lập hệ thống của mình, nhưng sẽ không thể tránh khỏi việc để chúng ở chế độ sản xuất. Để giúp hạn chế người dùng không đáng tin cậy truy cập vào máy chủ OrientDB, hãy thêm người dùng mới và thay đổi mật khẩu trong ***config/orientdb-server-config.xml*** tệp cấu hình máy chủ.

Để hạn chế người dùng trái phép tự cấp đặc quyền trên máy chủ OrientDB, hãy tắt quyền ghi vào tệp cấu hình. Để giúp ngăn họ xem mật khẩu, hãy tắt quyền truy cập đọc. Lưu ý rằng ngay cả khi mật khẩu đã được băm, có nhiều kỹ thuật có sẵn để bẻ khóa băm hoặc đốn mật khẩu thực.

Bắt đầu từ phiên bản 2.2, bảng điều khiển OrientDB cung cấp một loạt lệnh để quản lý người dùng:

Người dùng cấp máy chủ (Server-level user):

- Được tạo bằng lệnh SET SERVER USER.
- Có quyền truy cập vào toàn bộ các cơ sở dữ liệu trên máy chủ OrientDB.
- Quyền của người dùng này bao gồm việc quản lý, truy cập, và thao tác trên bất kỳ cơ sở dữ liệu nào có trên máy chủ, tùy thuộc vào vai trò (role) mà người dùng được gán.

Người dùng cấp cơ sở dữ liệu (Database-level user):

- Được tạo bằng lệnh INSERT INTO OUser trong một cơ sở dữ liệu cụ thể.

- Quyền truy cập chỉ giới hạn trong cơ sở dữ liệu mà người dùng được tạo.
- Người dùng này không thể truy cập vào các cơ sở dữ liệu khác trên cùng máy chủ OrientDB. Mọi thao tác và quyền hạn chỉ có hiệu lực trong phạm vi của cơ sở dữ liệu hiện tại.

2.1.1 Quản lý User Server

Đầu tiên, ta kết nối với máy chủ OrientDB

CONNECT remote:localhost root root

```
orientdb> CONNECT remote:localhost root root
OK
orientdb {server=remote:localhost}>
```

LIST SERVER USERS:

- Lệnh này hiển thị danh sách tất cả các người dùng đã được tạo trên máy chủ OrientDB.
- Đây là cách để kiểm tra nhanh danh sách người dùng hiện có, cùng với các thông tin cơ bản của họ (nếu có).

```
orientdb {server=remote:localhost}> LIST SERVER USERS

SERVER USERS

No users found
orientdb {server=remote:localhost}>
```

Hiện tại chưa có user nào

SET SERVER USER:

- Lệnh này dùng để **tạo mới hoặc chỉnh sửa** một người dùng.
- Khi tạo người dùng mới, bạn sẽ cần chỉ định tên người dùng, mật khẩu, và các quyền hạn (quyền truy cập) cho người dùng đó.
- Nếu người dùng đã tồn tại, lệnh này có thể được sử dụng để thay đổi mật khẩu hoặc điều chỉnh quyền truy cập của người dùng.

Cú pháp:

SET SERVER USER <user-name> <user-password> <user-permissions>

trong đó

<user-permissions> là resources bên dưới

```
OrientDB console v.3.2.35 (build ${buildNumber}, branch UNKNOWN) https://www.orientdb.com
Type 'help' to display all the supported commands.
orientdb> set server user phuthanhadmin phuthanhadmin *

Server user 'phuthanhadmin' set correctly
orientdb> set server user phuthanh phuthanh server.connect,database.listDatabases

Server user 'phuthanh' set correctly
orientdb> set server user ngocphung ngocphung server.connect,database.listDatabases,database.create

Server user 'ngocphung' set correctly
orientdb> set server user vantien vantien server.connect,database.listDatabases,database.create,database.drop

Server user 'vantien' set correctly
orientdb> list server users
```

DROP SERVER USER:

- Lệnh này dùng để **xóa một người dùng** khỏi máy chủ OrientDB.
- Sau khi lệnh được thực thi, người dùng đó sẽ không còn quyền truy cập vào cơ sở dữ liệu nữa.

Cú pháp: DROP USER SERVER <tên_user>

```
orientdb {server=remote:localhost}> DROP SERVER USER phuthanhadmin

Server user 'phuthanhadmin' dropped correctly
orientdb {server=remote:localhost}> LIST SERVER USERS

SERVER USERS

- 'phuthanhuser', permissions: reader
orientdb {server=remote:localhost}>
```

2.1.2 Tài nguyên Server

Tài nguyên (resources) và quyền truy cập (permissions) mà người dùng trong OrientDB có thể định nghĩa để quản lý quyền truy cập đến các tài nguyên cụ thể trên máy chủ. Dưới đây là giải thích chi tiết từng phần:

1. Tài nguyên và quyền truy cập

- Tài nguyên (resources): Đây là các thành phần mà người dùng có thể yêu cầu quyền truy cập, chẳng hạn như cơ sở dữ liệu, cấu hình máy chủ, và thông tin máy chủ.
- Quyền truy cập (permissions): Đây là các hành động mà người dùng có thể thực hiện trên tài nguyên. Ví dụ: tạo, xóa, kiểm tra sự tồn tại, hoặc truy cập

vào máy chủ.

2. Wildcard (*)

- Ký tự đại diện `*`: Cho phép một người dùng có quyền truy cập vào tất cả các tài nguyên. Người dùng `root` mặc định có tất cả quyền hạn, có nghĩa là họ có thể thực hiện bất kỳ hành động nào trên tất cả các cơ sở dữ liệu và tài nguyên quản lý.

3. Danh sách các tài nguyên và quyền hạn

Đoạn văn cung cấp danh sách các tài nguyên có sẵn trong OrientDB cùng với quyền hạn tương ứng:

- `database.create`: Quyền tạo một cơ sở dữ liệu mới trên máy chủ.
- `database.drop`: Quyền xóa một cơ sở dữ liệu.
- `database.exists`: Quyền kiểm tra sự tồn tại của một cơ sở dữ liệu.
- `database.freeze`: Quyền đóng băng quyền truy cập vào một cơ sở dữ liệu (không cho phép thay đổi).
- `database.release`: Quyền mở khóa một cơ sở dữ liệu đã bị đóng băng.
- `server.config.get`: Quyền lấy thông tin cấu hình của máy chủ.
- `server.config.set`: Quyền cập nhật cấu hình của máy chủ.
- `server.connect`: Quyền kết nối đến máy chủ.
- `server.info`: Quyền lấy thông tin và thống kê của máy chủ.
- `server.listDatabases`: Quyền liệt kê các cơ sở dữ liệu có sẵn trên máy chủ.
- `server.listDatabases.system`: Quyền bao gồm cơ sở dữ liệu OSystem trong danh sách các cơ sở dữ liệu.
- `server.replication`: Quyền thực hiện lệnh sao chép từ một máy chủ khác.

- `server.replication.start`: Quyền bắt đầu sao chép một cơ sở dữ liệu.
- `server.replication.stop`: Quyền dừng sao chép một cơ sở dữ liệu.
- `server.replication.config`: Quyền cập nhật cấu hình sao chép.
- `server.shutdown`: Quyền tắt máy chủ.

Lúc nãy ta có tạo ra user *phuthanhuser* thì khi mở file cấu hình lên sẽ có hiện

```
    <users>
        <user resources="reader" password="{PBKDF2WithHmacSHA256}C68C044DA164EDE13CC694F12B99B6FBC058E7DDC124EF4C:CBDF1F7CA8060F59027146BD5EEE18CD0EA378222E95BA45:65536"
name="phuthanhuser"/>
    </users>
    <properties>
        <entry value="false" name="profiler.enabled"/>
    </properties>
    <isAfterFirstTime>false</isAfterFirstTime>
</orient-server>
```

2.2 Bảo mật cơ sở dữ liệu

2.2.1 User

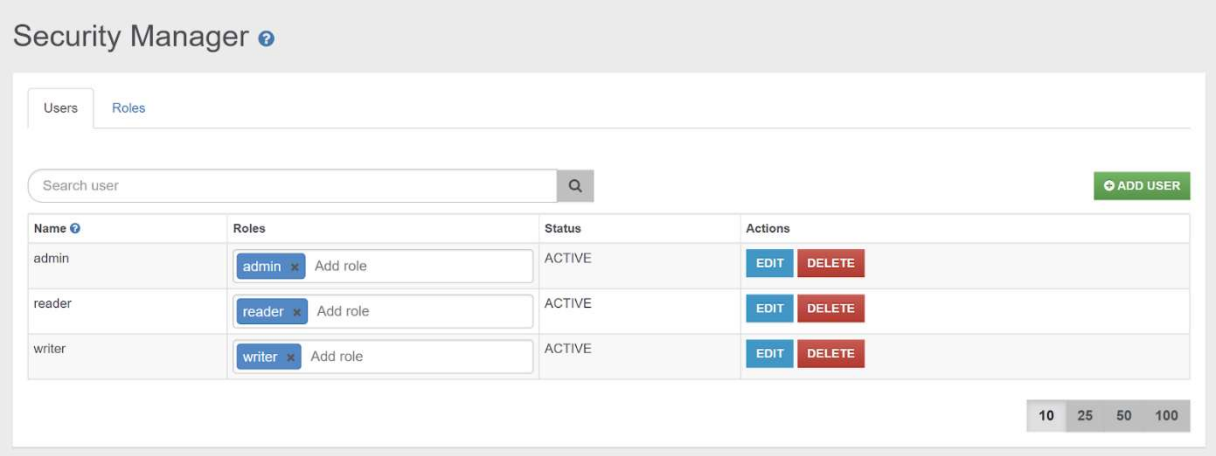
Theo mặc định, mỗi lần tạo cơ sở dữ liệu mới, OrientDB sẽ tạo ba người dùng mặc định với mật khẩu trùng tên người dùng:

1. admin: có quyền truy cập đầy đủ vào mọi chức năng của cơ sở dữ liệu.

2. reader: chỉ có quyền đọc, không thể chỉnh sửa hoặc xóa dữ liệu.

3. writer: giống reader, nhưng có thể tạo, cập nhật và xóa dữ liệu.

Người dùng được lưu dưới dạng bản ghi trong cụm ouser. OrientDB lưu mật khẩu dưới dạng mã băm và sử dung thuật toán PBKDF2 từ phiên bản 2.2 (trước đó là SHA-256). Trạng thái người dùng có thể là SUSPENDED hoặc ACTIVE, và chỉ người dùng ACTIVE mới có thể đăng nhập.



2.2.1.1 Kiểm tra user

Truy vấn người dùng hiện tại bằng câu lệnh SELECT trên lớp OUser:

```
SELECT * FROM OUser
```

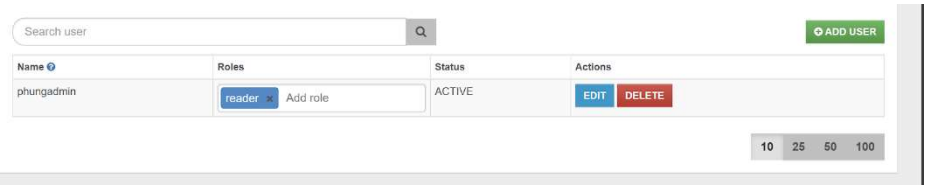
2.2.1.2 Tạo user

Sử dụng lệnh INSERT, đặt trạng thái là ACTIVE và chỉ định vai trò hợp lệ.

```
INSERT INTO OUser SET name = 'phungadmin', password = 'phungadmin', roles
= (SELECT FROM ORole WHERE name = 'reader')
```

```
3 item(s) found. Query executed in 0.027 sec(s).
orientdb {db=TestSecDB}> INSERT INTO OUser SET name = 'phungadmin', password = 'phungadmin', status = 'ACTIVE', roles =
(SELECT FROM ORole WHERE name = 'reader')

Inserted record '[OUser#6:0{name:phungadmin,password:{PBKDF2WithHmacSHA256}B31989A5269189FA3B9CF942B622A8C8C7B0526C9CB58
DAF:D0CF1237AC2778D436A10E21D1F08EF806E2B35BC3DC499B:65536,status:ACTIVE,roles:[1]} v1]' in 0.458000 sec(s).
```

Trên UI ta cũng có thể thấy là reader



2.2.1.3 Cập nhật user

ở trên kia khi tạo ngocphungadmin lỡ set role là reader muốn sửa lại thành admin thì cập nhật thành lệnh sau

```
UPDATE OUser SET roles = (SELECT @RID FROM ORole WHERE name = 'admin') WHERE name = 'phungadmin';
```

Tìm kiếm người dùng với tên là phungadmin trong class OUser. Nếu tìm thấy, trường roles của người dùng đó sẽ được cập nhật thành RID của vai trò admin. Nếu không tìm thấy người dùng, sẽ không có bản ghi nào bị thay đổi.

```
2 item(s) found. Query executed in 0.0 sec(s).
orientdb {db=TestSecDB}> UPDATE OUser SET roles = (SELECT @RID FROM ORole WHERE name = 'admin') WHERE name = 'phungadmin
';

Updated record(s) '[{count:1}]' in 0.018000 sec(s).

orientdb {db=TestSecDB}> SELECT * FROM OUser
```

Name	Roles	Status	Actions
phungadmin	admin × Add role	ACTIVE	EDIT DELETE

Thử đổi tên người dùng

```
UPDATE OUser SET name = 'phungreader' WHERE name = 'phungread'
```

Tìm tất cả người dùng có tên 'phungread' và thay đổi tên của họ thành 'phungreader'

```
orientdb {db=TestSecDB}> UPDATE OUser SET name = 'phungreader' WHERE name = 'phungread'

Updated record(s) '[{count:1}]' in 0.034000 sec(s).
```

phungreader	reader × Add role	ACTIVE	EDIT DELETE

2.2.1.4 Vô hiệu hóa user

Sử dụng UPDATE để chuyển trạng thái từ ACTIVE sang SUSPENDED, ví dụ để vô hiệu hóa tất cả trừ admin

```
UPDATE OUser SET status = 'SUSPENDED' WHERE name = 'phungreader'
```

```
orientdb {db=TestSecDB}> UPDATE OUser SET status = 'SUSPENDED' WHERE name = 'phungreader'

Updated record(s) '[{count:1}]' in 0.010000 sec(s).
```

Name	Roles	Status	Actions
phungadmin	admin × Add role	ACTIVE	EDIT DELETE
phungreader	reader × Add role	SUSPENDED	EDIT DELETE

2.2.2 Roles

2.2.2.1 Tổng quan về roles

Sử dụng hệ thống bit như thế này cho phép bạn dễ dàng quản lý quyền truy cập bằng

cách kết hợp các quyền khác nhau trong một giá trị duy nhất, giúp tiết kiệm không gian lưu trữ và tăng hiệu suất xử lý.

NONE: #0000 - 0
CREATE: #0001 - 1
READ: #0010 - 2
UPDATE: #0100 - 4
DELETE: #1000 - 8
ALL: #1111 - 15

Có thể kết hợp các quyền bằng cách cộng các giá trị tương ứng

Quyền CREATE + READ sẽ là 1 + 2 = 3 (0001 + 0010).

Đây là bảng giá trị

Giá trị	Quyền hạn	Mô tả
1	CREATE	Quyền tạo mới dữ liệu.
2	READ	Quyền đọc dữ liệu.
4	UPDATE	Quyền chỉnh sửa dữ liệu.
8	DELETE	Quyền xóa dữ liệu.
15	CREATE + READ + UPDATE + DELETE	Quyền tạo mới, đọc, chỉnh sửa, xóa dữ liệu.
16	EXECUTE	Quyền thực thi các lệnh hoặc thủ tục.
3	CREATE + READ	Quyền tạo và đọc dữ liệu.
6	READ + UPDATE	Quyền đọc và chỉnh sửa dữ liệu.
14	READ + UPDATE + DELETE	Quyền đọc, chỉnh sửa và xóa dữ liệu.
31	Tất cả các quyền	Quyền đầy đủ bao gồm tạo, đọc, chỉnh sửa, xóa và thực thi.

2.2.2.2 Truy vấn

Truy vấn các vai trò hiện tại bằng lệnh SELECT trên lớp Orole để xem các roles trong cơ sở dữ liệu

```
SELECT * FROM ORole
```

VÍ DỤ

RID	mode	name	rules
0		admin	{database=31, database.function=31, database.schema=31, database.command.gremlin=31, *=...|{*:#4:0}
1		reader	{database.cluster.internal=2, database.cluster.orole=0, database=2, database.function=2...|{database.class.*.*:#4:0, database:#4:1, database.schema:#4:1, database.cluster.internal:#4:1, database.cluster.or...
2		writer	{database.cluster.internal=2, database.class.oschedule=2, database.class.osequence=2, d...|{database.class.*.*:#4:0, database:#4:1, database.schema:#4:3, database.cluster.internal:#4:1, database.cluster.or...

Khi tạo mới db luôn luôn có sẵn 3 role

```
0 item(s) found. Query executed in 0.006 sec(s).
orientdb {db=TestSecDB}> SELECT RID, mode, name, policies, rules FROM ORole

+----+----+----+------+-------------------------------------------------------------+-----------------------------------
-----------------------------------+
|#   |RID |mode|name  |rules                                                        |policies
                                    |
+----+----+----+------+-------------------------------------------------------------+-----------------------------------
-----------------------------------+
|0   |    |    |admin |{database=31, database.function=31, database.schema=31, d...|{*=#4:0}
                                    |
|1   |    |    |reader|{database.cluster.internal=2, database.cluster.orole=0, d...|{database.cluster.internal=#4:1, dat
abase.class.*=#4:1, database.func...|
|2   |    |    |writer|{database.cluster.internal=2, database.class.oschedule=2,...|{database.cluster.internal=#4:1, dat
abase.class.*=#4:0, database.func...|
+----+----+----+------+-------------------------------------------------------------+-----------------------------------
```

- Ta có thể thấy database.cluster.internal=2 tức là reader có quyền đọc trên các cluster nội bộ

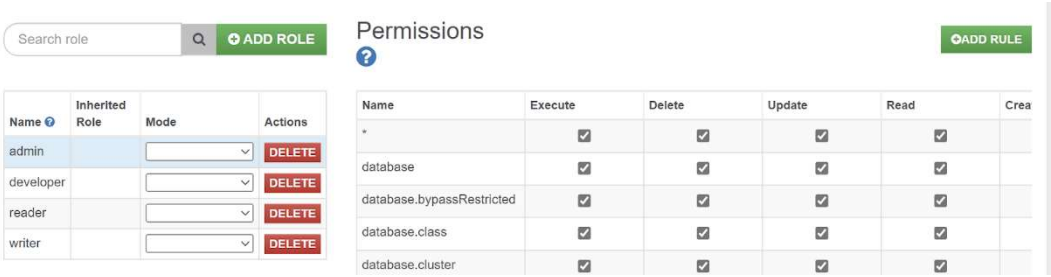
- Ta có thể thấy database.cluster.orole=0 tức là reader không thể đọc được các role

2.2.2.3 Tạo quyền

Tạo quyền tên là developer

```
INSERT INTO ORole SET name = 'developer'
```

```
orientdb {db=TestSecDB}> INSERT INTO ORole SET name = 'developer'

Inserted record '[ORole#5:3{name:developer} v1]' in 0.052000 sec(s).
```



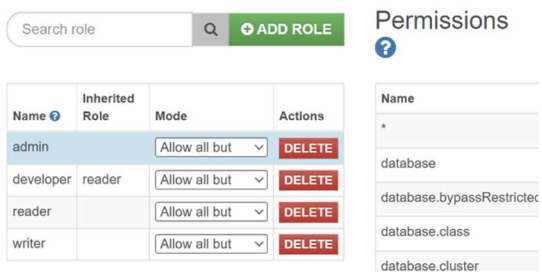
2.2.2.4 Kế thừa quyền

Vai trò có thể kế thừa quyền từ các vai trò khác theo kiểu hướng đối tượng. Để cho phép một vai trò mở rộng vai trò khác, hãy thêm vai trò chính vào thuộc tính.

```
UPDATE ORole SET inheritedRole = (SELECT FROM ORole WHERE name = 'reader') WHERE name = 'developer';
```

Lệnh này cho phép role developer kế thừa những quyền của reader

```
orientdb {db=TestSecDB}> UPDATE ORole SET inheritedRole = (SELECT FROM ORole WHERE name = 'reader') WHERE name = 'develo
per';

Updated record(s) '[{count:1}]' in 0.049000 sec(s).
```



2.2.3 Modes

Trong các phiên bản mới của OrientDB (từ v3.1 trở đi), hệ thống không còn dùng

các con số như 0, 1, 2 để biểu thị quyền nữa.

- **Allow All But (1)**: Cho phép tất cả trừ những quyền bị cấm qua các quy tắc. Nếu không có quy tắc nào cho tài nguyên, quyền truy cập sẽ được cho phép. Thường dùng cho admin.
- **Deny All But (0)**: Từ chối mọi quyền trừ khi có quy tắc cho phép. Mặc định áp dụng cho người dùng thông thường như reader và writer.

2.2.4 Security Policies

Chính sách bảo mật là phiên bản nâng cao của quyền thao tác, giúp kiểm soát truy cập chi tiết hơn. Một chính sách bảo mật là một bản đồ gồm các cặp **hoạt động - điều kiện**.

Hoạt động (operation) là một trong các loại sau

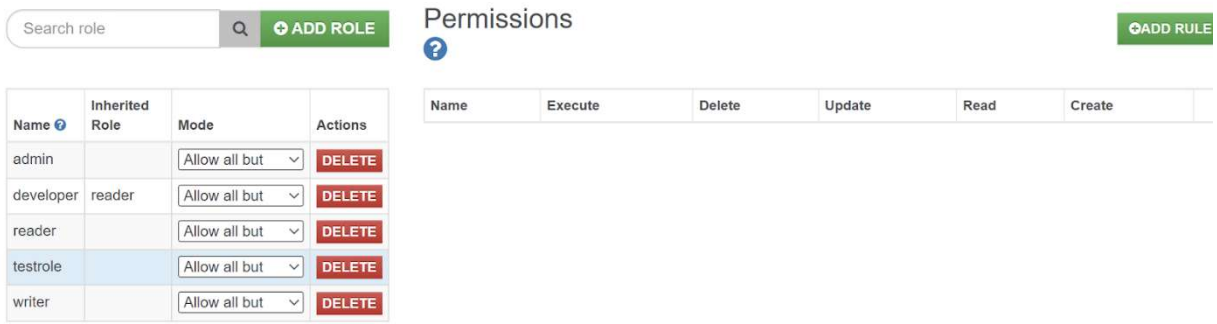
NONE: #0000 - 0
CREATE: #0001 - 1
READ: #0010 - 2
UPDATE: #0100 - 4
DELETE: #1000 - 8
ALL: #1111 - 15

. Điều kiện (predicate) là một điều kiện SQL, có thể đơn giản (ví dụ: TRUE hoặc FALSE) hoặc phức tạp (ví dụ: "owner = $currentUser"), có thể bao gồm cả các truy vấn con hoặc các thao tác trên đồ thị.

Chính sách bảo mật được áp dụng cho từng bản ghi ở mỗi giai đoạn của vòng đời (tạo, đọc, cập nhật, xóa) để quyết định xem hoạt động đó có được phép hay không.

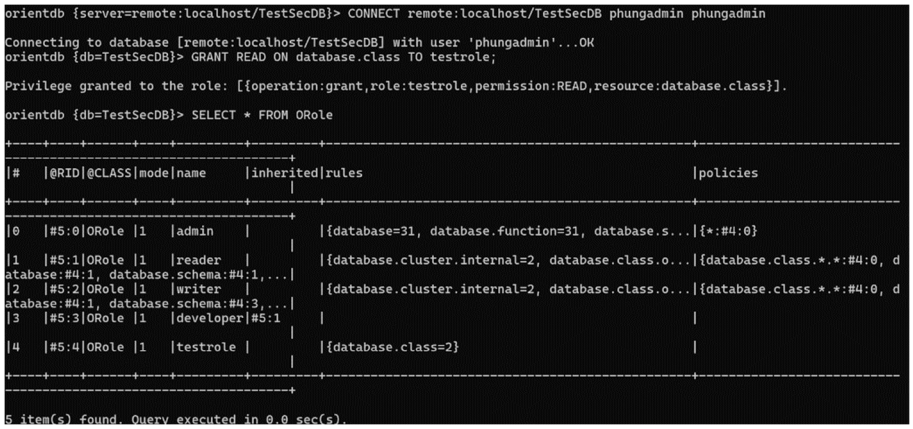
VÍ DỤ:

Giả sử testrole không có bất kì quyền nào cả

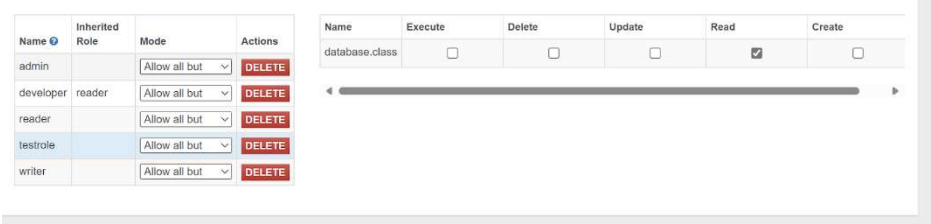


GRANT CREATE ON database.class TO testrole;

lệnh này cho phép role testrole có quyền tạo các class



```
orientdb {server=remote:localhost/TestSecDB}> CONNECT remote:localhost/TestSecDB phungadmin phungadmin

Connecting to database [remote:localhost/TestSecDB] with user 'phungadmin'...OK
orientdb {db=TestSecDB}> GRANT READ ON database.class TO testrole;

Privilege granted to the role: [{operation:grant,role:testrole,permission:READ,resource:database.class}].

orientdb {db=TestSecDB}> SELECT * FROM ORole

+----+----+------+----+---------+---------+-----------------------------------------------+-------------------------
-------------------------------------+
|#   |@RID|@CLASS|mode|name     |inherited|rules                                          |policies
                                      |
+----+----+------+----+---------+---------+-----------------------------------------------+-------------------------
-------------------------------------+
|0   |#5:0|ORole |1   |admin    |         |{database=31, database.function=31, database.s...|{*:#4:0}
                                      |
|1   |#5:1|ORole |1   |reader   |         |{database.cluster.internal=2, database.class.o...|{database.class.*.*:#4:0, d
atabase:#4:1, database.schema:#4:1,...|
|2   |#5:2|ORole |1   |writer   |         |{database.cluster.internal=2, database.class.o...|{database.class.*.*:#4:0, d
atabase:#4:1, database.schema:#4:3,...|
|3   |#5:3|ORole |1   |developer|#5:1     |                                               |
                                      |
|4   |#5:4|ORole |1   |testrole |         |{database.class=2}                             |
                                      |
+----+----+------+----+---------+---------+-----------------------------------------------+-------------------------
-------------------------------------+

5 item(s) found. Query executed in 0.0 sec(s).
```



2.2.5 ORestricted

ORestricted trong OrientDB là một cơ chế bảo mật cấp bản ghi, cho phép quản lý quyền truy cập dữ liệu chi tiết tới từng bản ghi trong một lớp (class). Khi một lớp kế thừa từ ORestricted, mỗi bản ghi trong lớp sẽ có các trường ẩn giúp xác định quyền truy cập:

- _allowRead: Xác định danh sách user hoặc role được phép đọc bản ghi.
- _allowUpdate: Xác định danh sách user hoặc role được phép sửa bản ghi.
- _allowDelete: Xác định danh sách user hoặc role được phép xóa bản ghi.

Đặc điểm: Nếu user không được liệt kê trong các trường trên thì không thể truy cập bản ghi, ngay cả khi họ có quyền truy cập lớp (class). - Chỉ các user được liệt kê hoặc quản trị viên (admin) mới có thể truy cập và thực hiện các thao tác

2.2.6 Resources

Tài nguyên là các chuỗi ký tự liên kết với các khái niệm trong OrientDB và phân biệt chữ hoa, chữ thường.

Các loại tài nguyên phổ biến và cách kiểm tra quyền truy cập của chúng:

database: Kiểm tra khi truy cập vào cơ sở dữ liệu.

database.class.<class-name>: Kiểm tra khi truy cập vào một lớp cụ thể.

database.class.<class-name>.<property-name>: Kiểm tra khi truy cập vào một thuộc tính cụ thể của lớp.

database.cluster.<cluster-name>: Kiểm tra khi truy cập vào một cụm cụ thể.

database.query: Kiểm tra khi thực thi truy vấn.

database.command: Kiểm tra khi thực thi lệnh.

database.schema: Kiểm tra quyền truy cập vào schema.

database.function: Kiểm tra khi thực thi hàm.

database.config: Kiểm tra khi truy cập cấu hình cơ sở dữ liệu.

database.hook.record: Kiểm tra khi sử dụng các hook bản ghi.

server.admin: Kiểm tra khi truy cập quản trị máy chủ từ xa.

database.bypassRestricted: Cho phép người dùng bỏ qua các hạn chế về quyền truy cập trong cơ sở dữ liệu.

database.command.gremlin: Cho phép thực hiện các lệnh Gremlin trong cơ sở dữ liệu.

database.systemclusters: Kiểm tra quyền truy cập vào các cluster hệ thống.

VÍ DỤ:

Cho phép quyền truy cập tất cả các lớp

```
UPDATE ORole SET rules['database'] = 31 WHERE name = "testrole";
```

31 là giá trị thể hiện tất cả các quyền vì vậy lệnh này update cho phép testrole có tất cả các quyền

```
orientdb {db=TestSecDB}> UPDATE ORole SET rules['database'] = 31 WHERE name = "testrole";

Updated record(s) '[{count:1}]' in 0.002000 sec(s).

orientdb {db=TestSecDB}> SELECT * FROM ORole

+----+----+------+----+---------+---------+-----------------------------------------------+-------------------------
-------------------------------------+
|#   |@RID|@CLASS|mode|name     |inherited|rules                                          |policies
                                       |
+----+----+------+----+---------+---------+-----------------------------------------------+-------------------------
-------------------------------------+
|0   |#5:0|ORole |1   |admin    |         |{database=31, database.function=31, database.s...|{*:#4:0}
                                       |
|1   |#5:1|ORole |1   |reader   |         |{database.cluster.internal=2, database.cluster...|{database.class.*.*:#4:0, d
atabase:#4:1, database.schema:#4:1,...|
|2   |#5:2|ORole |1   |writer   |         |{database.cluster.internal=2, database.class.o...|{database.class.*.*:#4:0, d
atabase:#4:1, database.schema:#4:3,...|
|3   |#5:3|ORole |1   |developer|#5:1     |                                               |
                                       |
|4   |#5:4|ORole |1   |testrole |         |{database.class=3, database=31}                |
                                       |
+----+----+------+----+---------+---------+-----------------------------------------------+-------------------------
-------------------------------------+
```

2.3 Mã hóa Cơ sở dữ liệu

Từ phiên bản 2.2, OrientDB hỗ trợ mã hóa dữ liệu trên đĩa, giúp ngăn người dùng trái phép truy cập nội dung cơ sở dữ liệu hoặc vượt qua bảo mật. Khóa mã hóa không được lưu trong cơ sở dữ liệu mà cần cung cấp khi chạy, nếu mất khóa thì không thể truy cập vào dữ liệu đã mã hóa.

Mã hóa hoạt động ở cấp cụm (collection) và dùng thuật toán AES. Bạn có thể cấu hình mã hóa chung cho tất cả các cơ sở dữ liệu qua Java API hoặc thông qua tham số JVM:

```
OGlobalConfiguration.STORAGE_ENCRYPTION_KEY.setValue("T1JJRU5UREJfSVNfQ09PTA==");
```

Chạy với lệnh:

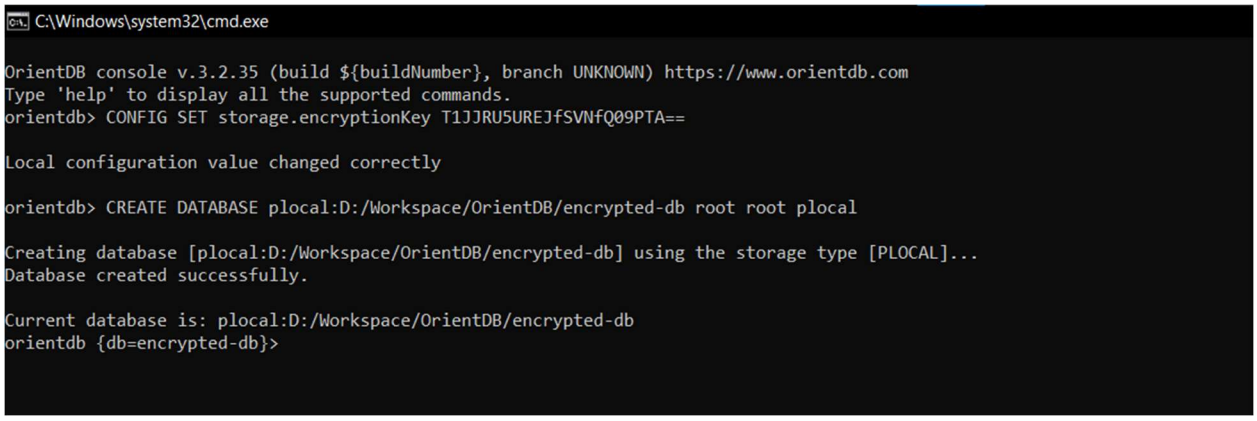
```
java ... -Dstorage.encryptionKey="T1JJRU5UREJfSVNfQ09PTA=="
```

Lưu ý: mã hóa không hỗ trợ giao thức từ xa, chỉ dùng cho môi trường nhúng hoặc trên máy chủ.

Để tạo cơ sở dữ liệu được mã hóa trong OrientDB, bạn cần thiết lập khóa mã hóa **storage.encryptionKey** trước khi dùng lệnh **CREATE DATABASE** trên console hoặc Java API.

Ví dụ trên console:

```
orientdb> CONFIG SET storage.encryptionKey
T1JJRU5UREJfSVNfQ09PTA==
orientdb> CREATE DATABASE plocal:/tmp/db/encrypted-db user password
plocal

Hoặc (đối với người dùng Windows OS)
orientdb> CONFIG SET storage.encryptionKey
T1JJRU5UREJfSVNfQ09PTA==
orientdb> CREATE DATABASE plocal:D:/Workspace/OrientDB/encrypted-db root root
plocal
```



Ví dụ Java API:

```
try (final OrientDB orientDB = new OrientDB("embedded:" +
dbDirectoryFile.getAbsolutePath(), OrientDBConfig.defaultConfig())) {
```

```
  final OrientDBConfig orientDBConfig = OrientDBConfig.builder()
            .addConfig(OGlobalConfiguration.STORAGE_ENCRYPTION_KEY,
"T1JJRU5UREJfSVNfQ09PTA==").build();
  orientDB.create("encryption", ODatabaseType.PLOCAL, orientDBConfig);
}
```

Để mở cơ sở dữ liệu được mã hóa trong OrientDB, bạn cần cung cấp khóa mã hóa **storage.encryptionKey** trước khi dùng lệnh **CONNECT** trên console hoặc Java API

Ví dụ trên console:

```
orientdb> CONFIG SET storage.encryptionKey
T1JJRU5UREJfSVNfQ09PTA==
orientdb> CONNECT plocal:D:/Workspace/OrientDB/encrypted-db root root

Hoặc (đối với người dùng Windows OS)
orientdb> CONFIG SET storage.encryptionKey
T1JJRU5UREJfSVNfQ09PTA==
orientdb> CONNECT plocal:D:/Workspace/OrientDB/encrypted-db root root
```

```
orientdb> CONNECT plocal:D:/Workspace/OrientDB/encrypted-db root root

Connecting to database [plocal:D:/Workspace/OrientDB/encrypted-db] with user 'root'...OK
orientdb {db=encrypted-db}>
```

Ví dụ Java API:

```
try (final OrientDB orientDB = new OrientDB("embedded:" +
dbDirectoryFile.getAbsolutePath(), OrientDBConfig.defaultConfig())) {
  final OrientDBConfig orientDBConfig = OrientDBConfig.builder()
```

```
    .addConfig(OGlobalConfiguration.STORAGE_ENCRYPTION_KEY,
"T1JJRU5UREJfSVNfQ09PTA==").build();
  orientDB.open("encryption", "user", "password", orientDBConfig);
}
```

Trong trường hợp bạn truyền vào khóa rỗng (null) hoặc không hợp lệ khi mở cơ sở dữ liệu, OrientDB sẽ đưa ra một ngoại lệ **ODatabaseException**.

```
C:\Windows\system32\cmd.exe
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at com.orientechnologies.common.console.OConsoleApplication.execute(OConsoleApplication.java:456)
        at com.orientechnologies.common.console.OConsoleApplication.executeCommands(OConsoleApplication.java:283)
        at com.orientechnologies.common.console.OConsoleApplication.run(OConsoleApplication.java:144)
        at com.orientechnologies.orient.console.OConsoleDatabaseApp.main(OConsoleDatabaseApp.java:162)
 [OWOWCache]
2024-11-02 15:55:36:023 SEVER In storage encrypted-db file with name 'config.bd' has broken page under the index 1 [OLocalPaginatedStorage]
2024-11-02 15:55:36:083 SEVER Exception `3F56875E` in storage `plocal:D:\Workspace\OrientDB/encrypted-db`: 3.2.35 (build ${buildNumber}, branch UNKNOWN) [OLocalPaginatedStorage]
com.orientechnologies.orient.core.exception.OStorageException: Exception during execution of atomic operation inside of storage encrypted-db
        at com.orientechnologies.orient.core.storage.impl.local.paginated.atomicoperations.OAtomicOperationsManager.executeInsideAtomicOperation(OAtomicOperationsManager.java:146)
        at com.orientechnologies.orient.core.storage.impl.local.OAbstractPaginatedStorage.open(OAbstractPaginatedStorage.java:545)
        at com.orientechnologies.orient.core.db.OrientDBEmbedded.getAndOpenStorage(OrientDBEmbedded.java:528)
        at com.orientechnologies.orient.core.db.OrientDBEmbedded.open(OrientDBEmbedded.java:480)
        at com.orientechnologies.orient.core.db.OrientDB.open(OrientDB.java:226)
        at com.orientechnologies.orient.core.db.OrientDB.open(OrientDB.java:212)
        at com.orientechnologies.orient.console.OConsoleDatabaseApp.connect(OConsoleDatabaseApp.java:261)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at com.orientechnologies.common.console.OConsoleApplication.execute(OConsoleApplication.java:456)
        at com.orientechnologies.common.console.OConsoleApplication.executeCommands(OConsoleApplication.java:283)
        at com.orientechnologies.common.console.OConsoleApplication.run(OConsoleApplication.java:144)
        at com.orientechnologies.orient.console.OConsoleDatabaseApp.main(OConsoleDatabaseApp.java:162)
Caused by: java.lang.NullPointerException
        at com.orientechnologies.orient.core.storage.impl.local.paginated.base.ODurablePage.<init>(ODurablePage.java:75)
        at com.orientechnologies.orient.core.storage.index.sbtree.singlevalue.v1.CellBTreeBucketSingleValueV1.<init>(CellBTreeBucketSingleValueV1.java:58)
        at com.orientechnologies.orient.core.storage.index.sbtree.singlevalue.v1.CellBTreeSingleValueV1$CellBTreeSpliteratorForward.tryAdvance(CellBTreeSingleValueV1.java:1695)
        at java.util.Spliterator.forEachRemaining(Spliterator.java:326)
        at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:481)
        at java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:471)
        at java.util.stream.ReduceOps$ReduceOp.evaluateSequential(ReduceOps.java:708)
        at java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)
        at java.util.stream.ReferencePipeline.collect(ReferencePipeline.java:499)
        at com.orientechnologies.orient.core.storage.config.OClusterBasedStorageConfiguration.preloadConfigurationProperties(OClusterBasedStorageConfiguration.java:1122)
        at com.orientechnologies.orient.core.storage.config.OClusterBasedStorageConfiguration.load(OClusterBasedStorageConfiguration.java:258)
        at com.orientechnologies.orient.core.storage.impl.local.OAbstractPaginatedStorage.lambda$open$1(OAbstractPaginatedStorage.java:551)
        at com.orientechnologies.orient.core.storage.impl.local.paginated.atomicoperations.OAtomicOperationsManager.executeInsideAtomicOperation(OAtomicOperationsManager.java:140)
        ... 14 more


!ERROR: Cannot open database 'encrypted-db'
-> Exception during execution of atomic operation inside of storage encrypted-db
-> null
orientdb {server=plocal:D:/Workspace/OrientDB/encrypted-db}>
```

Lưu ý: từ phiên bản 3.x mã hóa cụm không còn được hỗ trợ.

2.4 Kết nối SSL

2.4.1 Secure SSL connections

SSL (Secure Sockets Layer) và TLS (Transport Layer Security) là các cơ chế bảo mật cho phép mã hóa các kết nối giữa server và client, giúp ngăn chặn truy cập trái phép và bảo vệ dữ liệu truyền qua mạng. Với SSL/TLS thì các kết nối HTTP và BINARY trong OrientDB đều được mã hóa giúp bảo vệ dữ liệu khỏi các cuộc tấn

công trung gian (man-in-the-middle).

- Mục đích của việc dùng SSL/TLS

+ Bảo vệ dữ liệu: SSL/TLS mã hóa dữ liệu truyền tải, đảm bảo rằng chỉ người gửi và người nhận mới có thể hiểu được dữ liệu.

+ Xác thực server: SSL giúp client biết rằng họ đang kết nối đúng server mà họ tin tưởng.

+ Xác thực client (tuỳ chọn): Nếu cần, server cũng có thể yêu cầu xác thực client, đảm bảo rằng chỉ người dùng đã được cấp quyền mới có thể truy cập.

- Các bước cấu hình SSL/TLS trong OrientDB:

+ Key Store và Trust Store: Key Store là nơi lưu chứng chỉ của server (giúp xác thực server). Trust Store là nơi lưu chứng chỉ của các bên mà server hoặc client tin tưởng.

+ Bước 1: Tạo Key Store cho Server

Tạo Key Store để lưu chứng chỉ của server. Key Store này sẽ giúp mã hóa dữ liệu và xác minh server.

keytool -genkey -alias server -keystore orientdb.ks -keyalg RSA -keysize 2048 -validity 3650

- Alias (server): Tên của chứng chỉ.
- Keystore (orientdb.ks): Tên file Key Store sẽ chứa chứng chỉ của server.
- keyalg RSA: Loại mã hóa là RSA.
- keysize 2048: Độ dài khóa là 2048 bit đảm bảo tính bảo mật cao.
- validity 3650: Thời hạn của chứng chỉ là 10 năm (3650 ngày).

```
C:\Users\LG>keytool -genkey -alias server -keystore orientdb.ks -keyalg RSA -keysize 2048 -validity 3650
Enter keystore password:

Re-enter new password:

They don't match. Try again
Enter keystore password:

Re-enter new password:

What is your first and last name?
  [Unknown]:  tien
What is the name of your organizational unit?
  [Unknown]:  spkt
What is the name of your organization?
  [Unknown]:  spkt
What is the name of your City or Locality?
  [Unknown]:  HoChiMinh
What is the name of your State or Province?
  [Unknown]:  HCM
What is the two-letter country code for this unit?
  [Unknown]:  VN
Is CN=tien, OU=spkt, O=spkt, L=HoChiMinh, ST=HCM, C=VN correct?
  [no]:  yes

Enter key password for <server>
        (RETURN if same as keystore password):

Re-enter new password:
```

Password: Tien2024@

+ Bước 2: Xuất Chứng chỉ của Server để Client có thể Xác Thực

Client cần chứng chỉ của server để xác minh server. Để chia sẻ chứng chỉ này với client, bạn cần xuất nó từ Key Store:

keytool -export -alias server -keystore orientdb.ks -file orientdb.cert

```
C:\Users\LG>keytool -export -alias server -keystore orientdb.ks -file orientdb.cert
Enter keystore password:

Certificate stored in file <orientdb.cert>
```

Bước 3: Tạo Key Store và Trust Store cho Client

Client cũng cần có Key Store để mã hóa kết nối khi truy cập vào server.

keytool -genkey -alias console -keystore orientdb-console.ks -keyalg RSA -keysize 2048 -validity 3650

```
C:\Users\LG>keytool -genkey -alias console -keystore orientdb-console.ks -keyalg RSA -keysize 2048 -validity 3650
Enter keystore password:

Re-enter new password:

What is your first and last name?
  [Unknown]:  tientran
What is the name of your organizational unit?
  [Unknown]:  ute
What is the name of your organization?
  [Unknown]:  hcmute
What is the name of your City or Locality?
  [Unknown]:  HoChiMinh
What is the name of your State or Province?
  [Unknown]:  HCM
What is the two-letter country code for this unit?
  [Unknown]:  VN
Is CN=tientran, OU=ute, O=hcmute, L=HoChiMinh, ST=HCM, C=VN correct?
  [no]:  yes

Enter key password for <console>
        (RETURN if same as keystore password):

Re-enter new password:
```

PassWord: Test123@

Bước 4: Tạo Trust Store cho Client và Nhập Chứng chỉ của Server

Client cần nhập chứng chỉ của server vào Trust Store để biết rằng đây là server tin cậy:

keytool -import -alias server -keystore orientdb-console.ts -file orientdb.cert

```
C:\Users\LG>keytool -import -alias server -keystore orientdb-console.ts -file orientdb.cert
Enter keystore password:

Re-enter new password:

Owner: CN=tien, OU=spkt, O=spkt, L=HoChiMinh, ST=HCM, C=VN
Issuer: CN=tien, OU=spkt, O=spkt, L=HoChiMinh, ST=HCM, C=VN
Serial number: 36e19ea9
Valid from: Sat Nov 02 10:59:56 ICT 2024 until: Tue Oct 31 10:59:56 ICT 2034
Certificate fingerprints:
         MD5:  3B:AC:70:CD:79:ED:80:4E:48:9B:9F:F4:DD:FD:3E:2A
         SHA1: 0E:28:84:62:DC:60:0D:98:CF:FF:6C:74:73:FC:0D:73:F7:E6:90:15
         SHA256: 94:06:AD:52:F5:19:3A:BF:85:6A:B6:41:F3:35:A3:74:4A:E1:6A:58:42:E8:A4:4D:30:E2:35:37:B2:84:BF:87
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 64 84 5A 1E 1C C5 F4 B5   09 49 1F 07 93 40 A3 3E  d.Z......I...@.>
0010: 3D 5C 73 4E                                        =\sN
]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore

C:\Users\LG>
```

+ Cấu hình SSL/TLS trong OrientDB Server

Sau khi có Key Store và Trust Store cần cấu hình để OrientDB có thể sử dụng SSL. Trong tệp cấu hình $ORIENTDB_HOME/config/orientdb-server-config.xml cần

bật SSL:

```
<sockets>
  <socket
implementation="com.orientechnologies.orient.server.network.OServerSSLSock
etFactory" name="ssl">
 <socket
implementation="com.orientechnologies.orient.server.network.OServerTLSSock
etFactory" name="ssl">
            <parameters>
               <parameter value="false" name="network.ssl.clientAuth"/>
               <parameter value="E:\Database_Security\Project_Giuaky\orientdb-
community-3.2.35\config\cert\orientdb.p12" name="network.ssl.keyStore"/>
               <parameter value="Tien2024@"
name="network.ssl.keyStorePassword"/>
               <parameter value="E:\Database_Security\Project_Giuaky\orientdb-
community-3.2.35\config\cert\orientdb-console.p12"
name="network.ssl.trustStore"/>
               <parameter value="Test123@"
name="network.ssl.trustStorePassword"/>
            </parameters>
         </socket>

  <listener protocol="binary" ip-address="0.0.0.0" port-range="2434-2440"
socket="ssl"/>
```

```
<network>
    <sockets>
        <socket implementation="com.orientechnologies.orient.server.network.OServerTLSSocketFactory" name="ssl">
            <parameters>
                <parameter value="false" name="network.ssl.clientAuth"/>
                <parameter value="E:\Database_Security\Project_Giuaky\orientdb-community-3.2.35\config\cert\orientdb.p12" name="network
                <parameter value="Tien2024@" name="network.ssl.keyStorePassword"/>
                <parameter value="E:\Database_Security\Project_Giuaky\orientdb-community-3.2.35\config\cert\orientdb-console.p12" name=
                <parameter value="Test123@" name="network.ssl.trustStorePassword"/>
            </parameters>
        </socket>
    </sockets>
    <protocols>
        <!-- Default registered protocol. It reads commands using the HTTP protocol
            and write data locally -->
        <protocol name="binary"
                  implementation="com.orientechnologies.orient.server.network.protocol.binary.ONetworkProtocolBinary"/>
        <protocol name="http"
```

```
<listeners>
    <listener protocol="binary" ip-address="0.0.0.0" port-range="2424-2430" socket="default"/>
    <listener protocol="binary" ip-address="0.0.0.0" port-range="2434-2440" socket="ssl"/>
    <listener protocol="http" ip-address="0.0.0.0" port-range="2480-2490" socket="default">
```

Giải thích:

- network.ssl.clientAuth: false có nghĩa là server không yêu cầu client phải xác thực (dùng xác thực một chiều).
- network.ssl.keyStore và network.ssl.trustStore: Đường dẫn đến Key Store và Trust Store.
- network.ssl.keyStorePassword và network.ssl.trustStorePassword: Mật khẩu của Key Store và Trust Store.

- Cấu hình SSL trên Console

1. Đảm bảo rằng KEYSTORE, TRUSTSTORE, và mật khẩu được cấu hình đúng trong tệp console.sh.
2. Đặt thuộc tính client.ssl.enabled thành true để bật SSL cho console.

2.4.2 Cấu hình Client để Sử dụng SSL

Để client có thể kết nối với server bằng SSL, bạn cần cấu hình các thuộc tính hệ thống trong Java như sau:

```
ORIENTDB_SETTINGS="-Djna.nosys=true -Djava.util.logging.config.file=\"$ORIENTDB_HOME/config/orientdb-client-log.properties\" -Djava.awt.hea
#JAVA_OPTS=-Xmx1024m
KEYSTORE="$ORIENTDB_HOME/config/cert/orientdb-console.ks"
KEYSTORE_PASS=Tien2024@
TRUSTSTORE="$ORIENTDB_HOME/config/cert/orientdb-console.ts"
TRUSTSTORE_PASS=Test123@
SSL_OPTS="-Dclient.ssl.enabled=true"

exec "$JAVA" -client $JAVA_OPTS $ORIENTDB_OPTS_MEMORY $ORIENTDB_SETTINGS $SSL_OPTS \
    -Dfile.encoding=utf-8 -Dorientdb.build.number="UNKNOWN@r${buildNumber}; 2024-10-29 10:52:12+0000" \
    -cp "$ORIENTDB_HOME/lib/orientdb-tools-3.2.35.jar:$ORIENTDB_HOME/lib/*:$ORIENTDB_HOME/plugins/*" \
    "-Djavax.net.ssl.keyStore=$KEYSTORE" \
    "-Djavax.net.ssl.keyStorePassword=$KEYSTORE_PASS" \
    "-Djavax.net.ssl.trustStore=$TRUSTSTORE" \
    "-Djavax.net.ssl.trustStorePassword=$TRUSTSTORE_PASS" \
    com.orientechnologies.orient.console.OConsoleDatabaseApp $*
```

- **javax.net.ssl.keyStore** và **javax.net.ssl.trustStore**: Đường dẫn tới Key Store và Trust Store của client.
- **javax.net.ssl.keyStorePassword** và **javax.net.ssl.trustStorePassword**: Mật khẩu tương ứng của Key Store và Trust Store.

+ Xác thực Client (Nếu Cần)

Nếu server yêu cầu client cũng cung cấp chứng chỉ:

Bước 1: Xuất chứng chỉ của client để server xác minh.

```
keytool -export -alias console -keystore orientdb-console.ks -file orientdb-console.cert
```

Bước 2: Tạo Trust Store cho server và nhập chứng chỉ của client

```
keytool -import -alias console -keystore orientdb.ts -file orientdb-console.cert
```

Bước 3: Trong $ORIENTDB_HOME/config/orientdb-server-config.xml, bật xác thực client:

```
<sockets>
  <socket
implementation="com.orientechnologies.orient.server.network.OServerSSLSocketFactory" name="ssl">
    <parameters>
      <parameter value="true" name="network.ssl.clientAuth"/>
```

```
    <parameter value="config/cert/orientdb.ks" name="network.ssl.keyStore"/>
    <parameter value="password" name="network.ssl.keyStorePassword"/>
    <parameter value="config/cert/orientdb.ts" name="network.ssl.trustStore"/>
    <parameter value="password" name="network.ssl.trustStorePassword"/>
   </parameters>
  </socket>
</sockets>
```

+ Kiểm tra kết nối SSL

Kết nối server và client để đảm bảo rằng chúng có thể giao tiếp với nhau qua SSL mà không gặp lỗi xác thực.

```
rding to the setting 'memory.leftToOS' not taking into account heap memory [OMemoryAndLocalPaginatedEnginesInitializer]
2024-11-02 21:41:18:983 INFO  OrientDB auto-config DISKCACHE=4,014MB (heap=1,963MB os=8,025MB) [orientechnologies]
2024-11-02 21:41:18:983 INFO  System is started under an effective user : `LG` [OEngineLocalPaginated]
2024-11-02 21:41:19:201 INFO  WAL maximum segment size is set to 6,144 MB [OrientDBDistributed]
2024-11-02 21:41:20:858 INFO  Databases directory: E:\Database_Security\Project_Giuaky\orientdb-community-3.2.35\databas
es [OServer]
2024-11-02 21:41:21:030 INFO  Page size for WAL located in E:\Database_Security\Project_Giuaky\orientdb-community-3.2.35
\databases\OSystem is set to 4096 bytes. [CASDiskWriteAheadLog]
2024-11-02 21:41:21:264 INFO  DWL:OSystem: block size = 4096 bytes, maximum segment size = 2314 MB [DoubleWriteLogGL]
2024-11-02 21:41:21:546 INFO  Storage 'plocal:E:\Database_Security\Project_Giuaky\orientdb-community-3.2.35\databases/OS
ystem' is opened under OrientDB distribution : 3.2.35 (build ${buildNumber}, branch UNKNOWN) [OLocalPaginatedStorage]
2024-11-02 21:41:21:936 INFO  Listening binary connections on 0.0.0.0:2424 (protocol v.38, socket=default) [OServerNetwo
rkListener]
2024-11-02 21:41:22:249 INFO  Listening binary connections on 0.0.0.0:2434 (protocol v.38, socket=ssl) [OServerNetworkLi
stener]
2024-11-02 21:41:22:249 INFO  Listening http connections on 0.0.0.0:2480 (protocol v.10, socket=default) [OServerNetwork
Listener]
2024-11-02 21:41:22:327 INFO  Using ORIENTDB_ROOT_PASSWORD variable, for root's password [OServer]
2024-11-02 21:41:22:342 INFO  Installing dynamic plugin 'orientdb-etl-3.2.35.jar'... [OServerPluginManager]
2024-11-02 21:41:22:342 INFO  Installing dynamic plugin 'orientdb-studio-3.2.35.zip'... [OServerPluginManager]
2024-11-02 21:41:22:342 INFO  [OVariableParser.resolveVariables] Property not found: distributed [orientechnologies]
2024-11-02 21:41:22:358 WARNI Authenticated clients can execute any kind of code into the server by using the following
allowed languages: [sql] [OServerSideScriptInterpreter]
2024-11-02 21:41:23:749 INFO  OrientDB Studio available at http://192.168.1.3:2480/studio/index.html [OServer]
2024-11-02 21:41:23:749 INFO  OrientDB Server is active v3.2.35 (build ${buildNumber}, branch UNKNOWN). [OServer]
2024-11-02 21:41:47:256 INFO  Page size for WAL located in E:\Database_Security\Project_Giuaky\orientdb-community-3.2.35
\databases\demodb is set to 4096 bytes. [CASDiskWriteAheadLog]
2024-11-02 21:41:58:310 INFO  DWL:demodb: block size = 4096 bytes, maximum segment size = 2314 MB [DoubleWriteLogGL]
2024-11-02 21:41:58:732 INFO  Storage 'plocal:E:\Database_Security\Project_Giuaky\orientdb-community-3.2.35\databases/de
modb' is opened under OrientDB distribution : 3.2.35 (build ${buildNumber}, branch UNKNOWN) [OLocalPaginatedStorage]
```

Sử dụng openssl để kiểm tra kết nối SSL với server trên cổng 2434

```
C:\Users\LG>openssl s_client -connect 127.0.0.1:2434
Loading 'screen' into random state - done
CONNECTED(00000214)
depth=0 /C=VN/ST=HCM/L=HoChiMinh/O=spkt/OU=spkt/CN=tien
verify error:num=18:self signed certificate
verify return:1
depth=0 /C=VN/ST=HCM/L=HoChiMinh/O=spkt/OU=spkt/CN=tien
verify return:1
---
Certificate chain
 0 s:/C=VN/ST=HCM/L=HoChiMinh/O=spkt/OU=spkt/CN=tien
   i:/C=VN/ST=HCM/L=HoChiMinh/O=spkt/OU=spkt/CN=tien
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDVzCCAj+gAwIBAgIENuGeqTANBgkqhkiG9w0BAQsFADBcMQswCQYDVQQGEwJW
TjEMMAoGA1UECBMDSENNMRIwEAYDVQQHEwlIb0NoaU1pbmgxDTALBgNVBAoTBHNw
a3QxDTALBgNVBAsTBHNwa3QxDTALBgNVBAMTBHRpZW4wHhcNMjQxMTAyMDM1OTU2
WhcNMzQxMDMxMDM1OTU2WjBcMQswCQYDVQQGEwJWTjEMMAoGA1UECBMDSENNMRIw
EAYDVQQHEwlIb0NoaU1pbmgxDTALBgNVBAoTBHNwa3QxDTALBgNVBAsTBHNwa3Qx
DTALBgNVBAMTBHRpZW4wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDU
VOV38DFvRjb8l06pRxsA737ArU1gcLk7cIbXIaVMtYjvY+iQEZ6Ut8eCKd6fACl t
EzEcaMEC+tJpQXm0y0UgLt3JRm3+zqD49txOr/YtXSeXG5Y6Ey2AdJCVLaSyXUer
Zmz0kAsZG+TxfYiJ+B8W1jlwfSwfH+u9YVIpHcYFMHn0ZR+JNoHcBzO3GqksAjtX
9cSjalKMLKppv2fnvX3JudSinTBtKXMMjZOkmCA9nfnbRwMaJjTe05wXiBSXsYi9
qGsvdGJReIjngBYehZubdJw68qIM62vH94EPEpoVpK4fON/H8EHnpRbQ2Da44hcP
zoPjKkRL95iB+Wgwv2MtAgMBAAGjITAfMB0GA1UdDgQWBBRkhFoeHMX0tQlJHweT
QKM+PVxzTjANBgkqhkiG9w0BAQsFAAOCAQEAw26YPLXEm+WPmW/B35ayuLAjQfa8
nUCjY3sFuuAkk1EvfwKfUkuNOzT/Kfd8LvAtyN3gO3DNciPX3NKJoR7hZmopl8Fk
6aSOHmd9Ie5Pr3w6E2A0XNHp+rBkM7vZpwt4+zJ5ejLhxI6VKFH+WpKtEUijZRO0
/c1NiRzHE/qf95I/Tp9f2pnGii7nsSZfVQR7spbfm4Dl9V1O450C5tpHcfaIDrOr
```

```
qGsvdGJReIjngBYehZubdJw68qIM62vH94EPEpoVpK4fON/H8EHnpRbQ2Da44hcP
zoPjKkRL95iB+Wgwv2MtAgMBAAGjITAfMB0GA1UdDgQWBBRkhFoeHMX0tQlJHweT
QKM+PVxzTjANBgkqhkiG9w0BAQsFAAOCAQEAw26YPLXEm+WPmW/B35ayuLAjQfa8
nUCjY3sFuuAkk1EvfwKfUkuNOzT/Kfd8LvAtyN3gO3DNciPX3NKJoR7hZmopl8Fk
6aSOHmd9Ie5Pr3w6E2A0XNHp+rBkM7vZpwt4+zJ5ejLhxI6VKFH+WpKtEUijZRO0
/c1NiRzHE/qf95I/Tp9f2pnGii7nsSZfVQR7spbfm4Dl9V1O450C5tpHcfaIDrOr
JHtjDffojOb1rTHbpc8xCc1rmL/yOpMSCN05/tgse5H0YiJILi+khvnoeshfFZxj
EVjFJoQZgZ9lMSffVMnWqNuJ6bXZPtEfLCHuEqEKGWHQ/6tUI+1210R96Q==
-----END CERTIFICATE-----
subject=/C=VN/ST=HCM/L=HoChiMinh/O=spkt/OU=spkt/CN=tien
issuer=/C=VN/ST=HCM/L=HoChiMinh/O=spkt/OU=spkt/CN=tien
---
No client certificate CA names sent
---
SSL handshake has read 1536 bytes and written 322 bytes
---
New, TLSv1/SSLv3, Cipher is DHE-RSA-AES256-SHA
Server public key is 2048 bit
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : TLSv1
    Cipher    : DHE-RSA-AES256-SHA
    Session-ID: 67263EAE4DA722F96E0770897D88BDE748C49467384A7DCA68B6CB79E367B8CE
    Session-ID-ctx:
    Master-Key: 5CC0982DF07A900C34DFB80B91BBB9ADA8550268AA14A6590FC826940F8BC6C616C31F1A948FC49DAE5886D396141B6C
    Key-Arg   : None
    Start Time: 1730559662
    Timeout   : 300 (sec)
    Verify return code: 18 (self signed certificate)
---
&
```

CHƯƠNG 3: DEMO CƠ CHẾ BẢO MẬT TRÊN ORIENTDB

3.1 Demo Server Security

3.1.1 User server

Ta xem danh sách các user server bằng lệnh ***List server users***

```
orientdb> list server users

SERVER USERS

- 'phuthanhadmin', permissions: *
orientdb>
```

Tiến hành tạo các ***user server*** bằng lệnh SET SERVER USER <user-name> <user-password> <user-permissions>

```
SET SERVER USER thanh thanh server.connect,server.listDatabases

SET SERVER USER phung phung
server.connect,server.listDatabases,database.create

SET SERVER USER tien tien
server.connect,server.listDatabases,database.create,database.drop

SET SERVER USER hao hao
server.connect,server.listDatabases,database.create,database.drop,*.read

SET SERVER USER huong huong
server.connect,server.listDatabases,*.read,database.freeze

SET SERVER USER hung hung
server.connect,server.listDatabases,*.read,database.release
```

thanh: có quyền kết nối tới server, xem danh sách các database

phung: có quyền kết nối tới server, xem danh sách các database, tạo database

tien: có quyền kết nối tới server, xem danh sách các database, tạo database, xóa database

hao: có quyền kết nối tới server, xem danh sách các database, tạo database, xóa database, truy cập vào database

huong: có quyền kết nối tới server, xem danh sách các database, truy cập vào database, đóng băng database

hung: có quyền kết nối tới server, xem danh sách các database, truy cập vào database, mở database bị đóng băng

(*Note: Nếu user muốn vào database phải có quyền *.read)*

```
orientdb> SET SERVER USER thanh thanh server.connect,server.listDatabases
SET SERVER USER phung phung server.connect,server.listDatabases,database.create
SET SERVER USER tien tien server.connect,server.listDatabases,database.create,database.drop
SET SERVER USER hao hao server.connect,server.listDatabases,database.create,database.drop,*.read
SET SERVER USER huong huong server.connect,server.listDatabases,*.read,database.freeze
SET SERVER USER hung hung server.connect,server.listDatabases,*.read,database.release

Server user 'thanh' set correctly
orientdb>
Server user 'phung' set correctly
orientdb>
Server user 'tien' set correctly
orientdb>
Server user 'hao' set correctly
orientdb>
Server user 'huong' set correctly
orientdb>
Server user 'hung' set correctly
orientdb>
```

Xem lại danh sách các user server bằng lệnh LIST SERVER USERS

```
orientdb> list server users

SERVER USERS

- 'hung', permissions: server.connect,server.listDatabases,*.read,database.release
- 'hao', permissions: server.connect,server.listDatabases,database.create,database.drop,*.read
- 'tien', permissions: server.connect,server.listDatabases,database.create,database.drop
- 'thanh', permissions: server.connect,server.listDatabases
- 'huong', permissions: server.connect,server.listDatabases,*.read,database.freeze
- 'phuthanhadmin', permissions: *
- 'phung', permissions: server.connect,server.listDatabases,database.create
orientdb>
```

Kiểm tra quyền của các user server

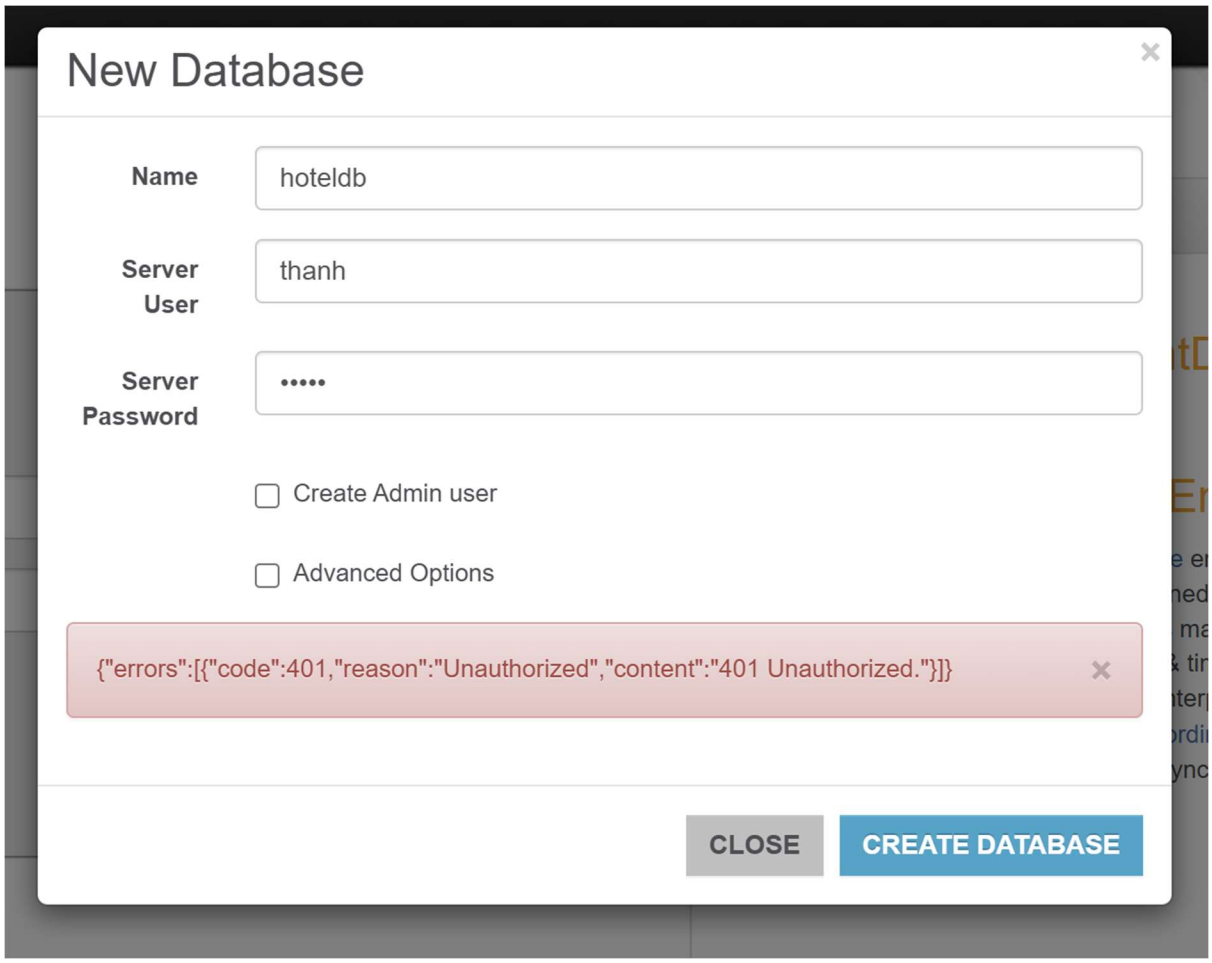
User thanh

User server thanh chỉ có quyền connect tới server, xem các databases của server. Ta thử connect tới server và list databases của server bằng user thanh thì sẽ thành công

Connect remote:localhost thanh thanh *List databases*

```
orientdb> connect remote:localhost thanh thanh
OK
orientdb {server=remote:localhost}> list databases

Found 1 databases:

* demodb

orientdb {server=remote:localhost}>
```

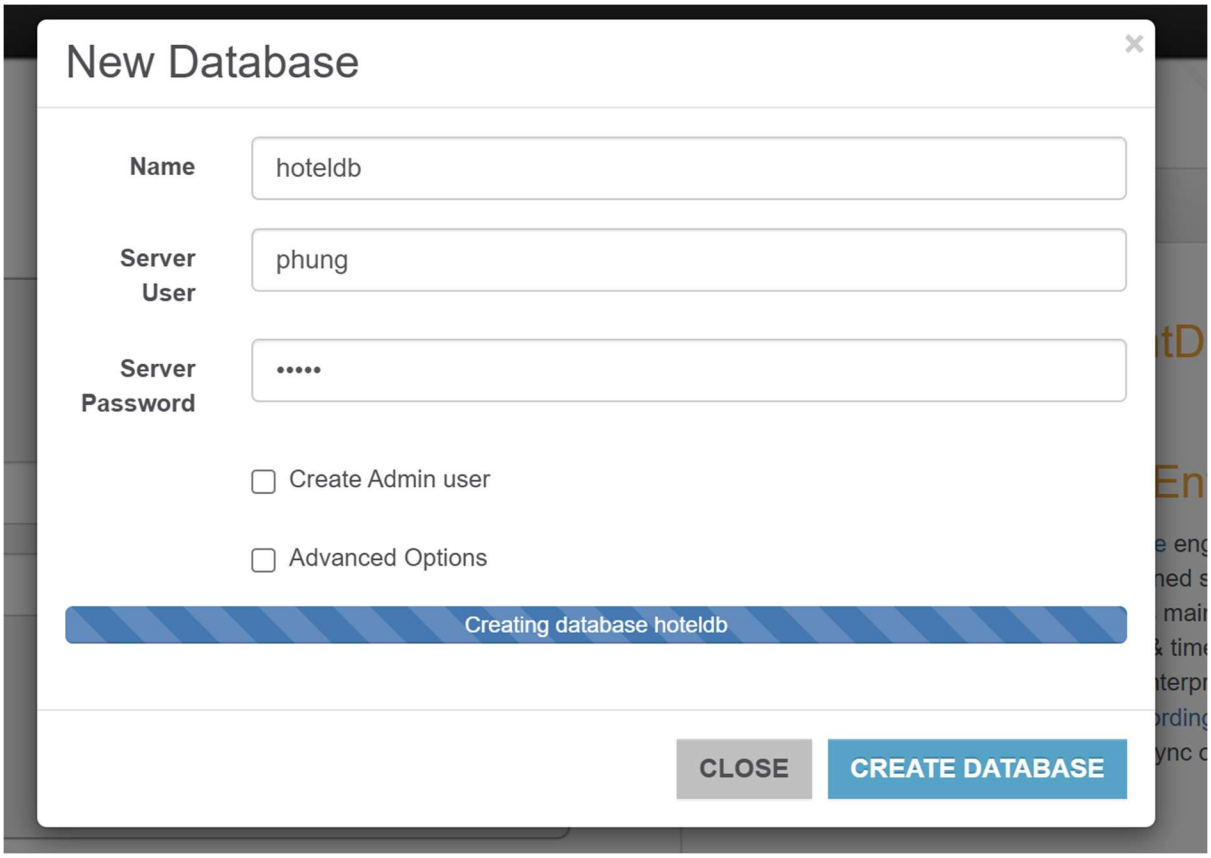
Bây giờ ta sẽ dùng user server thanh để tạo database ***hoteldb*** để xem user thanh có tạo được không

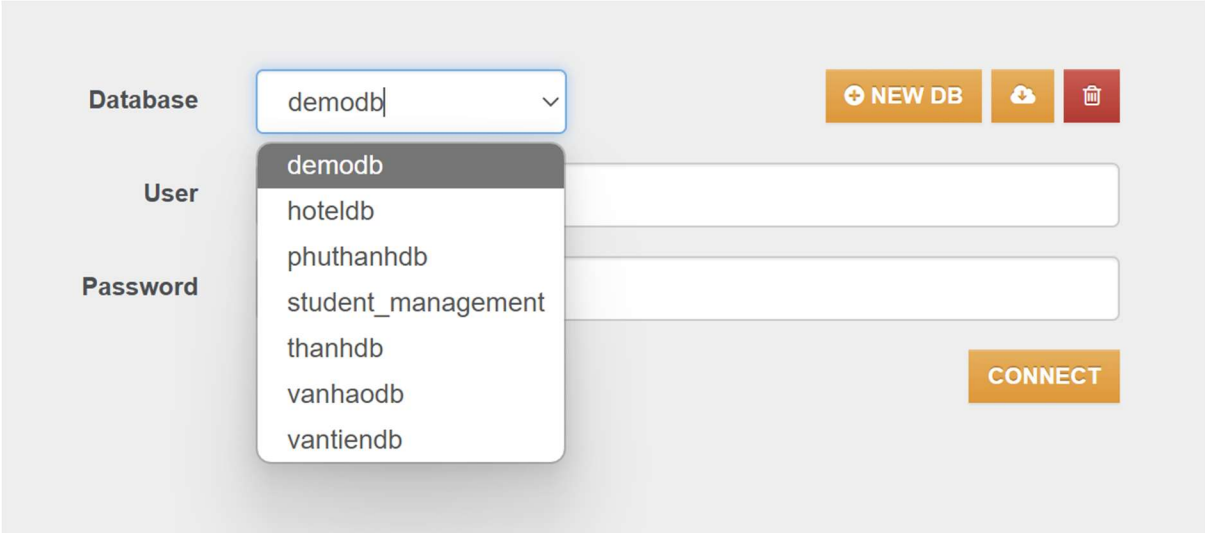


user phuthanh không thể tạo database được

User phung

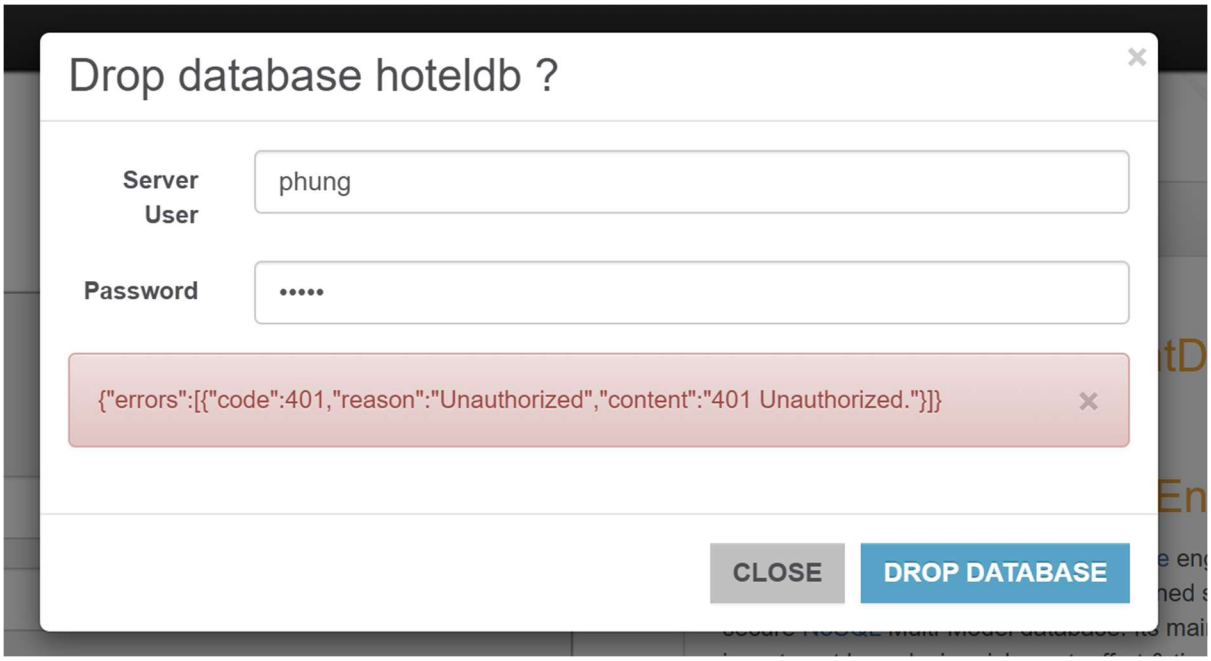
User server phung có quyền connect tới server và create database. Bây giờ ta sẽ dùng user server phung để tạo database để xem user phung có tạo được không





Ta thấy user phung tạo thành công database hoteldb nhưng không truy cập vào được vì không có quyền

Ta thử dùng user phung xóa database vừa tạo



Nó sẽ báo lỗi vì phung không được cấp quyền xóa nên không authorize được

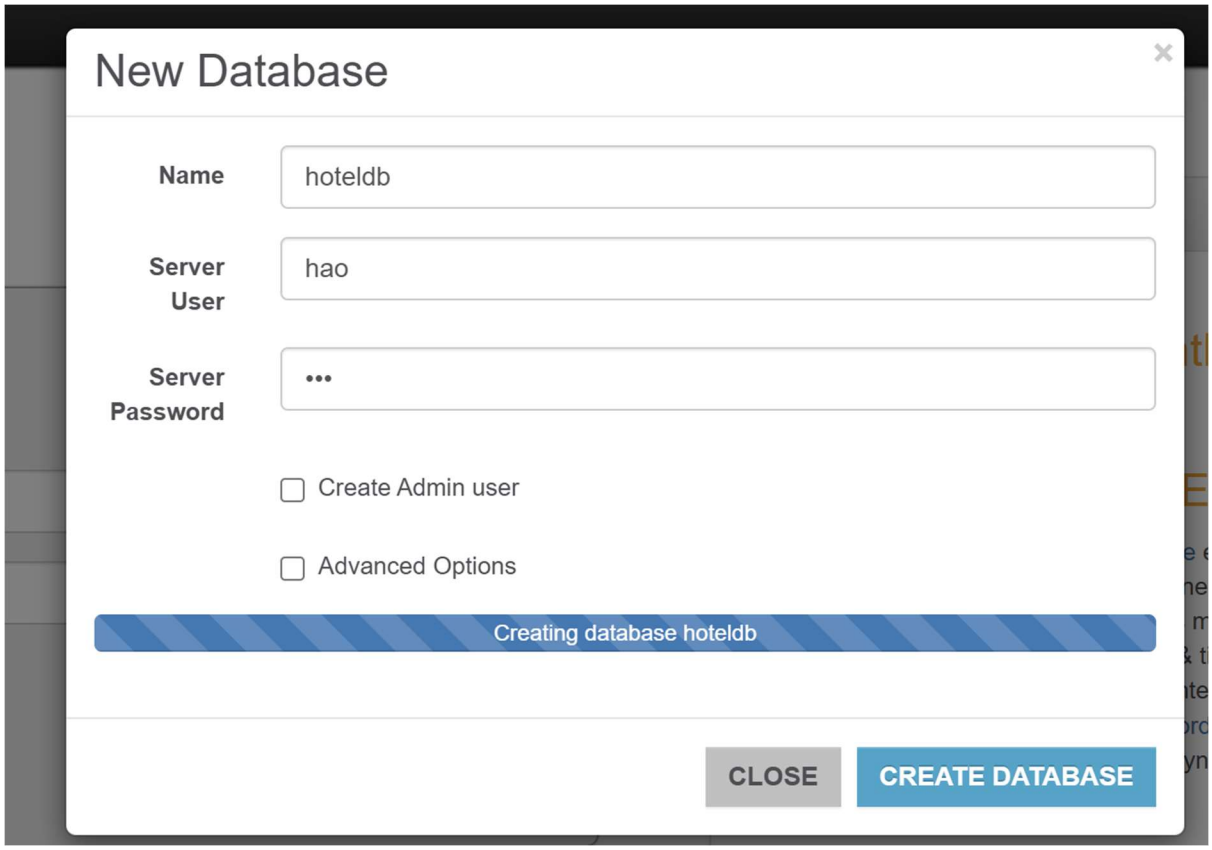
User tien

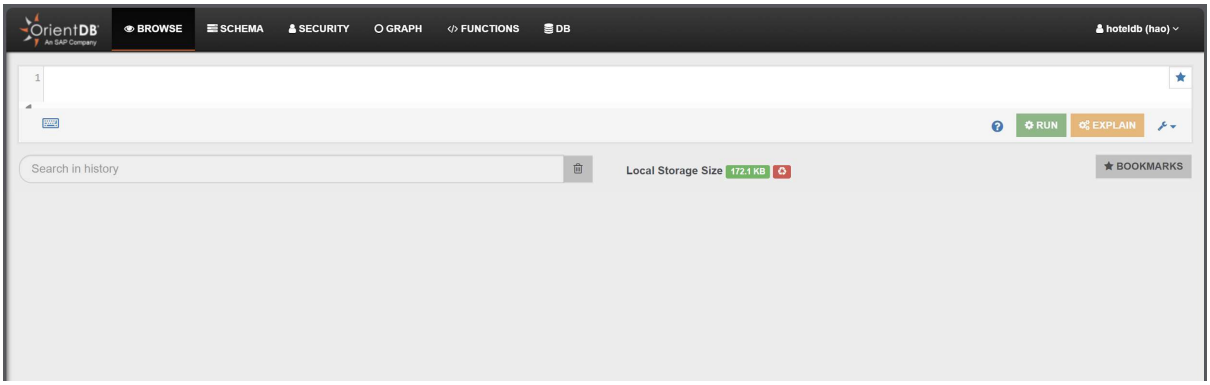
User server tien có quyền connect tới server và create database, drop database. Bây giờ ta sẽ dùng user server tien để xóa database ***hoteldb*** để xem user tien có xóa được không

Như vậy user tien có thể xóa được database

User hao

User server hao có quyền connect tới server và create database, drop database, và đọc được database (tức là truy cập vào đọc được). Bây giờ ta sẽ dùng user server hao để tạo database hoteldb để xem





Ta thấy user hao có thể tạo và truy cập được vào database

Ta dùng user hao để thử đóng bang database hoteldb thử bằng lệnh ***freeze hoteldb***

nhưng sẽ không thành công do không được cấp quyền

```
orientdb> connect remote:localhost/hoteldb hao hao

Connecting to database [remote:localhost/hoteldb] with user 'hao'...OK
orientdb {db=hoteldb}> freeze database

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'hao' cannot access to the resource [d
atabase.freeze]. Use another server user or change permission in the file config/orientdb-server-config.xml

orientdb {db=hoteldb}>
```

User huong

User huong có quyền kết nối tới server, xem danh sách các database, truy cập vào database, đóng băng database. Ta dùng user huong truy cập database hoteldb và đóng băng thì sẽ thành công

```
orientdb> connect remote:localhost/hoteldb hao hao

Connecting to database [remote:localhost/hoteldb] with user 'hao'...OK
orientdb {db=hoteldb}> freeze database

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'hao' cannot access to the resource [d
atabase.freeze]. Use another server user or change permission in the file config/orientdb-server-config.xml

orientdb {db=hoteldb}> connect remote:localhost/hoteldb huong huong

Disconnecting from the database [hoteldb]...OK

Connecting to database [remote:localhost/hoteldb] with user 'huong'...OK
orientdb {db=hoteldb}> freeze database


Database 'hoteldb' was frozen successfully
orientdb {db=hoteldb}>
```

Tuy nhiên mở đóng bang database hoteldb sẽ không thành công vì không có quyền

```
orientdb {db=hoteldb}> connect remote:localhost/hoteldb huong huong

Disconnecting from the database [hoteldb]...OK

Connecting to database [remote:localhost/hoteldb] with user 'huong'...OK
orientdb {db=hoteldb}> freeze database


Database 'hoteldb' was frozen successfully
orientdb {db=hoteldb}> release database

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'huong' cannot access to the resource
[database.release]. Use another server user or change permission in the file config/orientdb-server-config.xml
        DB name="hoteldb"
        DB name="hoteldb"

orientdb {db=hoteldb}>
```

User hung: có quyền kết nối tới server, xem danh sách các database, truy cập vào database, mở database bị đóng băng

Ta thử dùng user hung kết nối tới database hoteldb mà thử mở đóng bang thì sẽ thành

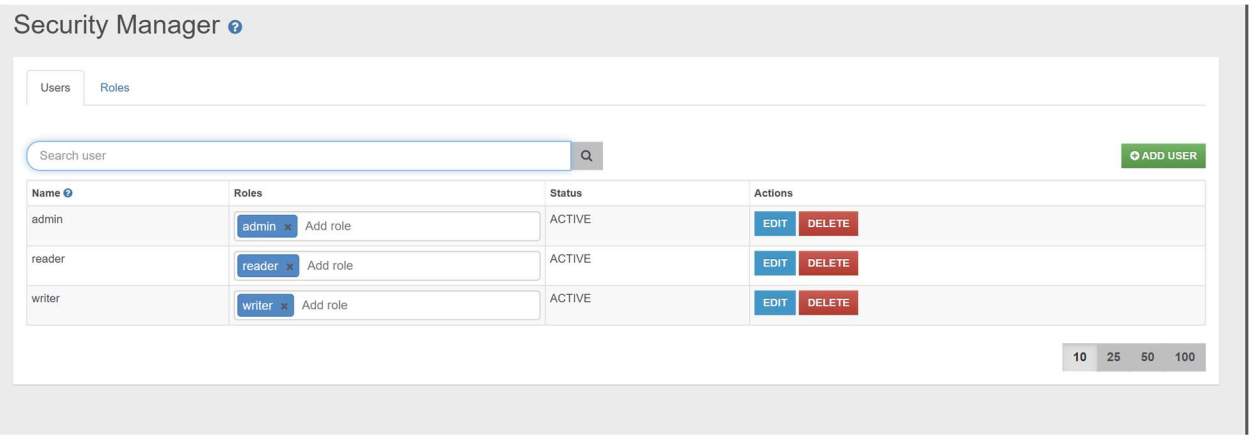
công

```
orientdb {db=hoteldb}> connect remote:localhost/hoteldb hung hung

Disconnecting from the database [hoteldb]...OK

Connecting to database [remote:localhost/hoteldb] with user 'hung'...OK
orientdb {db=hoteldb}> release database


Database 'hoteldb' was released successfully
orientdb {db=hoteldb}>
```

3.1.2. User database

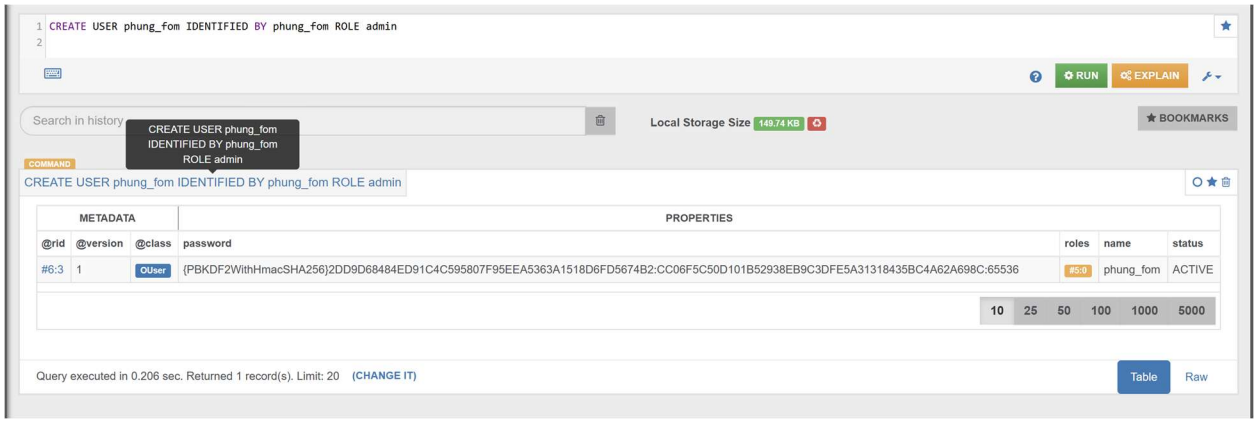
Khi ta tạo một database bất kì, hệ thống sẽ tự tạo 3 user mặc định

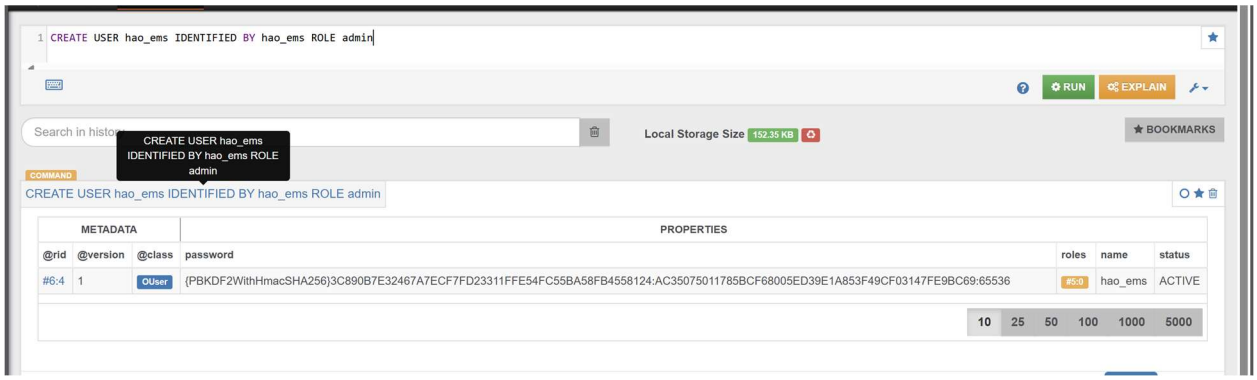
Admin với role admin, reader với role reader, writer với role writer

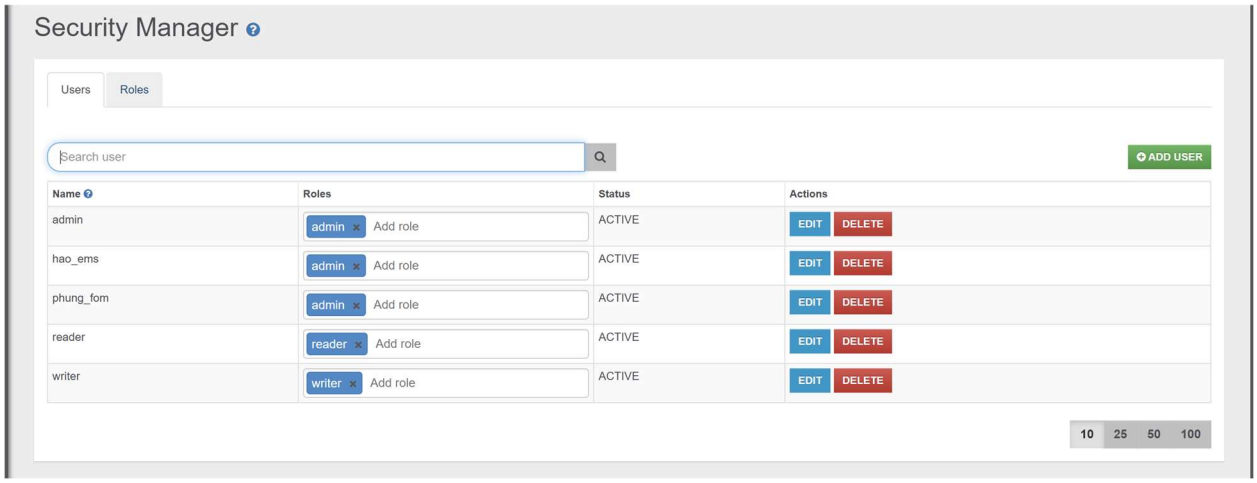


Bây giờ ta tạo thử 2 user phung_fom và hao_ems với role là admin thử

CREATE USER phung_fom IDENTIFIED BY phung_fom ROLE admin
CREATE USER hao_ems IDENTIFIED BY hao_ems ROLE admin







3.1.3. Disable Default user

Nếu muốn tắt tính năng 3 user mặc định là admin reader writer thì ta sẽ vào file ***security.json*** trong thư mục config

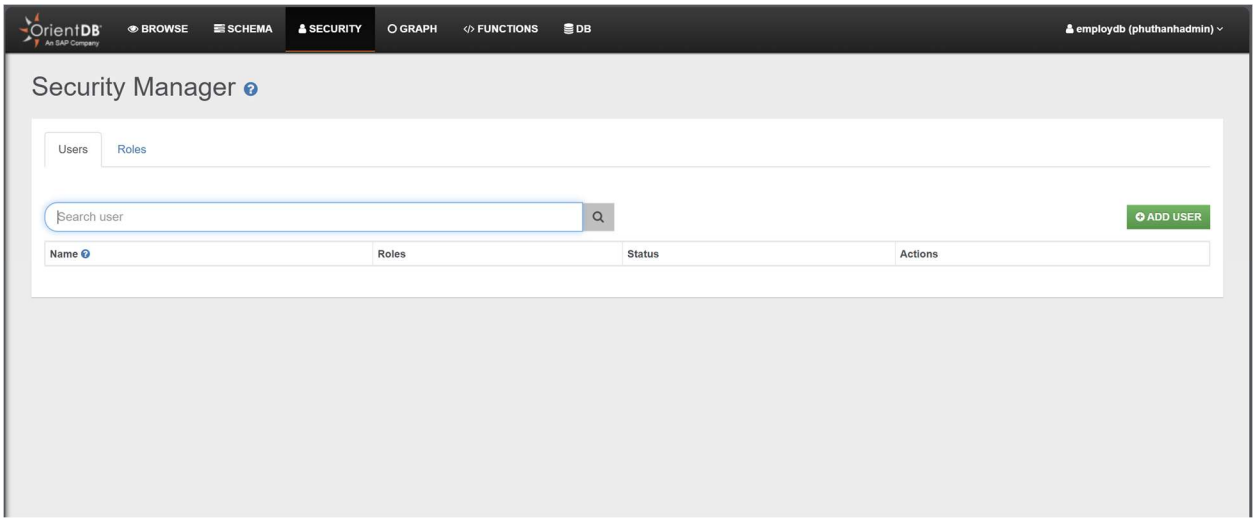
```
"server": {
  "createDefaultUsers": false
```

```
},
```

File Edit View

```
{
  "enabled": true,
  "debug": false,
  "server": {
    "createDefaultUsers": false
  },
```

Khi đó khi ta tạo 1 dababases mới employdb thì không có người dùng mặc định nào



3.2 Demo Front Office Manager

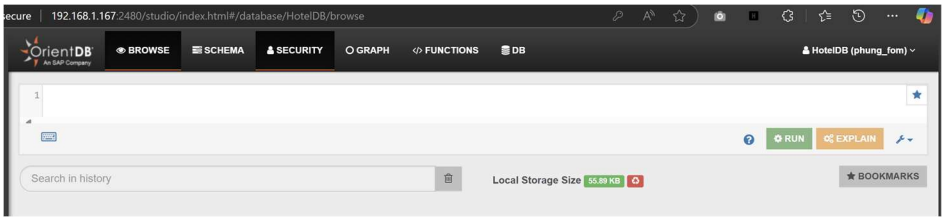
3.2.1 Kết nối database

Kết nối database bằng quản lý

```
CONNECT remote:localhost/HotelDB phung_fom phungfom;
```

```
orientdb> CONNECT remote:localhost/HotelDB phung_fom phungfom;

Connecting to database [remote:localhost/HotelDB] with user 'phung_fom'...OK
```

Thử kết nối bằng UI



3.2.2 Tạo Class

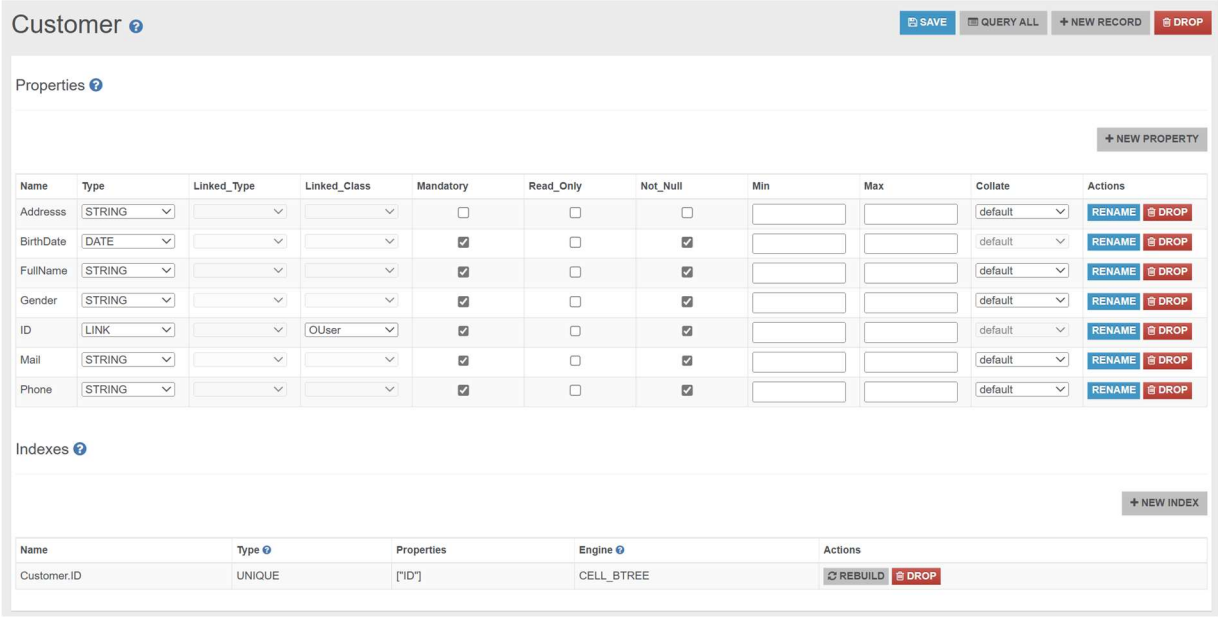
3.2.2.1 Class Customer

Tạo class chứa thông tin của khách hàng

```
CREATE CLASS Customer EXTENDS V;
CREATE PROPERTY Customer.ID LINK OUser (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Customer.FullName STRING (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Customer.Gender STRING (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Customer.BirthDate DATE (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Customer.Mail STRING (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Customer.Phone STRING (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Customer.Addresss STRING;
CREATE INDEX Customer.ID UNIQUE;
```

```
orientdb {db=HotelDB}> CREATE CLASS Customer EXTENDS V;

Class created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Customer.ID LINK OUser (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Customer.FullName STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Customer.Gender STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Customer.BirthDate DATE (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Customer.Mail STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Customer.Phone STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Customer.Addresss STRING;

Property created successfully.

orientdb {db=HotelDB}> CREATE INDEX Customer.ID UNIQUE;


Creating index...
Created index successfully in 0.035000 sec(s).


Index created successfully
```

Kiểm tra class đã được tạo trên UI

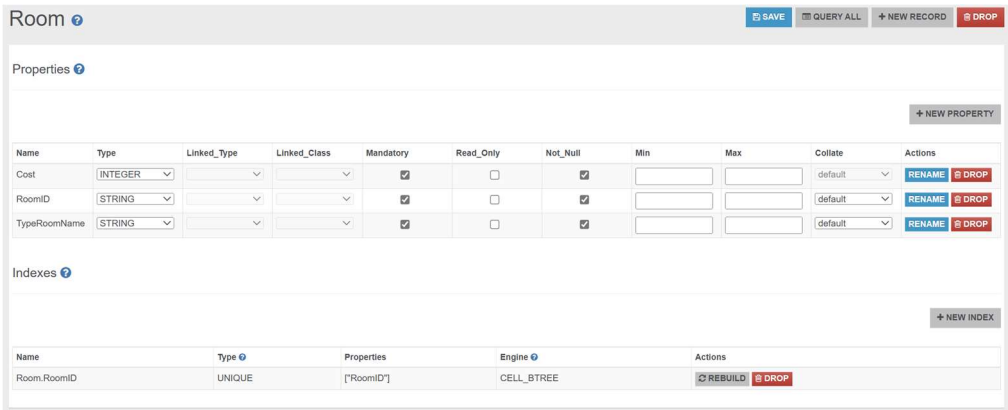


3.2.2.2 Class Room

```
CREATE CLASS Room EXTENDS V;
CREATE PROPERTY Room.RoomID STRING (MANDATORY TRUE,
NOTNULL TRUE);
CREATE PROPERTY Room.TypeRoomName STRING (MANDATORY
```

```
TRUE, NOTNULL TRUE);
CREATE PROPERTY Room.Cost INTEGER (MANDATORY TRUE,
NOTNULL TRUE);
CREATE INDEX Room.RoomID UNIQUE;
```

```
orientdb {db=HotelDB}> CREATE CLASS Room EXTENDS V;

Class created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Room.RoomID STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Room.TypeRoomName STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Room.Cost INTEGER (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE INDEX Room.RoomID UNIQUE;


Creating index...
Created index successfully in 0.034000 sec(s).


Index created successfully
```



3.2.2.3 Class Booking

```
CREATE CLASS Booking EXTENDS E;
CREATE PROPERTY Booking.BookID STRING (MANDATORY TRUE,
NOTNULL TRUE);
CREATE PROPERTY Booking.CustomerID LINK Customer
(MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.RoomID LINK Room (MANDATORY
TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.BookDate DATETIME (MANDATORY
```

TRUE, NOTNULL TRUE);

CREATE PROPERTY Booking.CheckInDate DATETIME (MANDATORY TRUE, NOTNULL TRUE);

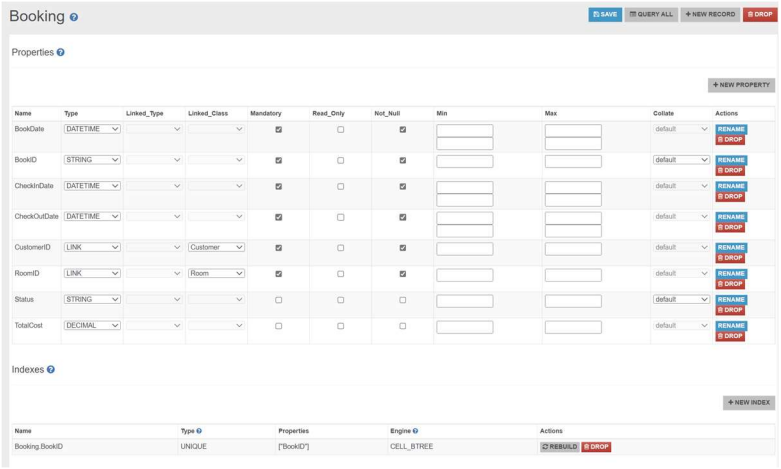
CREATE PROPERTY Booking.CheckOutDate DATETIME (MANDATORY TRUE, NOTNULL TRUE);

CREATE PROPERTY Booking.TotalCost DECIMAL;

CREATE PROPERTY Booking.Status STRING;

CREATE INDEX Booking.BookID UNIQUE;

```
orientdb {db=HotelDB}> CREATE CLASS Booking EXTENDS E;
CREATE PROPERTY Booking.BookID STRING (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.CustomerID LINK Customer (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.RoomID LINK Room (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.BookDate DATETIME (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.CheckInDate DATETIME (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.CheckOutDate DATETIME (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.TotalCost DECIMAL;
CREATE PROPERTY Booking.Status STRING;
CREATE INDEX Booking.BookID UNIQUE;

Class created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>

Creating index...
Created index successfully in 0.028000 sec(s).


Index created successfully
```



3.2.3. Tạo role

3.2.3.1 Receptionist

Tạo role lễ tân

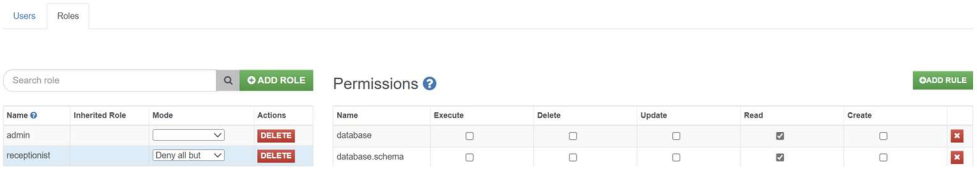
```
INSERT INTO ORole SET name = 'receptionist', mode = 0;
```

```
orientdb {db=HotelDB}> INSERT INTO ORole SET name = 'receptionist', mode = 0;
Inserted record '[ORole#5:4{name:receptionist,mode:0} v1]' in 0.006000 sec(s).
```

Cấp quyền được đăng nhập

```
GRANT READ ON database TO receptionist;
GRANT READ ON database.schema TO receptionist;
```

```
orientdb {db=HotelDB}> GRANT READ ON database TO receptionist;
Privilege granted to the role: [{operation:grant,role:receptionist,permission:READ,resource:database}].
orientdb {db=HotelDB}> GRANT READ ON database.schema TO receptionist;
Privilege granted to the role: [{operation:grant,role:receptionist,permission:READ,resource:database.schema}].
```



-Cho phép lễ tân tạo account khách hàng và tạo mới, chỉnh sửa thông tin khách hàng.

```
GRANT READ ON database.class.orole TO receptionist;
GRANT READ ON database.class.ouser TO receptionist;
```

```
GRANT CREATE ON database.class.ouser TO receptionist;
GRANT READ ON database.class.customer TO receptionist;
GRANT CREATE ON database.class.customer TO receptionist;
GRANT UPDATE ON database.class.customer TO receptionist;
GRANT READ ON database.cluster TO receptionist;
GRANT CREATE ON database.cluster TO receptionist;
GRANT UPDATE ON database.cluster TO receptionist;
```

```
orientdb {db=HotelDB}> GRANT READ ON database.class.orole TO receptionist;
GRANT READ ON database.class.ouser TO receptionist;
GRANT CREATE ON database.class.ouser TO receptionist;
GRANT READ ON database.class.customer TO receptionist;
GRANT CREATE ON database.class.customer TO receptionist;
GRANT UPDATE ON database.class.customer TO receptionist;
GRANT READ ON database.cluster TO receptionist;
GRANT CREATE ON database.cluster TO receptionist;
GRANT UPDATE ON database.cluster TO receptionist;

Privilege granted to the role: [{operation:grant,role:receptionist,permission:READ,resource:database.class.orole}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:receptionist,permission:READ,resource:database.class.ouser}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:receptionist,permission:CREATE,resource:database.class.ouser}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:receptionist,permission:READ,resource:database.class.customer}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:receptionist,permission:CREATE,resource:database.class.customer}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:receptionist,permission:UPDATE,resource:database.class.customer}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:receptionist,permission:READ,resource:database.cluster}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:receptionist,permission:CREATE,resource:database.cluster}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:receptionist,permission:UPDATE,resource:database.cluster}].
```

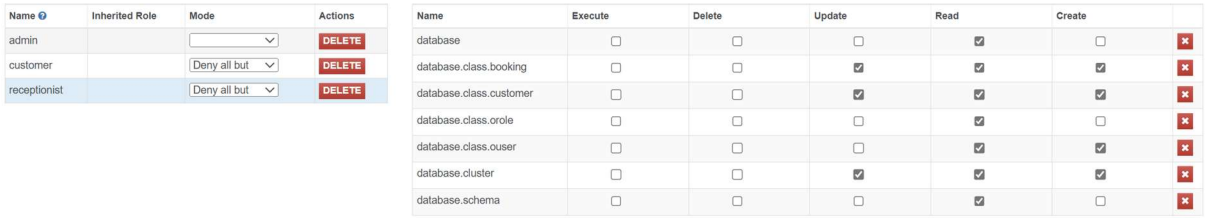
Name	Inherited Role	Mode	Actions
admin			DELETE
customer		Deny all but	DELETE
receptionist		Deny all but	DELETE

Name	Execute	Delete	Update	Read	Create	
database	☐	☐	☐	☑	☐	✖
database.class.customer	☐	☐	☑	☑	☑	✖
database.class.orole	☐	☐	☐	☑	☐	✖
database.class.ouser	☐	☐	☐	☑	☑	✖
database.cluster	☐	☐	☑	☑	☑	✖
database.schema	☐	☐	☐	☑	☐	✖

Cho phép lễ tân tạo, cập nhật, chỉnh sửa đặt phòng . Tuy nhiên không cho lễ tân thay đổi phòng vì phòng trường hợp lễ tân sẽ sửa giá tiền của phòng đó

```
GRANT READ ON database.class.booking TO receptionist;
GRANT CREATE ON database.class.booking TO receptionist;
GRANT UPDATE ON database.class.booking TO receptionist;
```

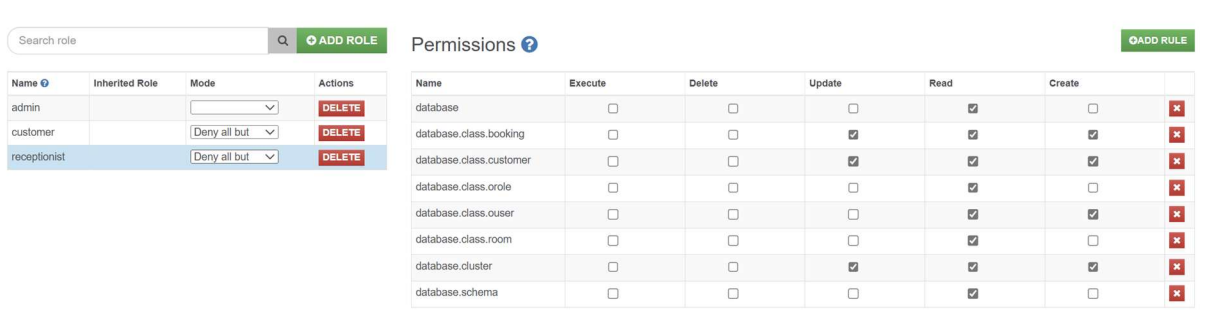
```
orientdb {db=HotelDB}> GRANT READ ON database.class.booking TO receptionist;
GRANT CREATE ON database.class.booking TO receptionist;
GRANT UPDATE ON database.class.booking TO receptionist;

Privilege granted to the role: [{operation:grant,role:receptionist,permission:READ,resource:database.class.booking}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:receptionist,permission:CREATE,resource:database.class.booking}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:receptionist,permission:UPDATE,resource:database.class.booking}].
```



Tuy nhiên chỉ cho lễ tên xem phòng chứ không cho lễ tân thay đổi phòng vì phòng trường hợp lễ tân sẽ sửa giá tiền của phòng đó

```
GRANT UPDATE ON database.class.room TO receptionist;
```

```
orientdb {db=HotelDB}> GRANT READ ON database.class.room TO receptionist;

Privilege granted to the role: [{operation:grant,role:receptionist,permission:READ,resource:database.class.room}].
```



3.2.3.2 Customer

Tạo role khách hàng

```
INSERT INTO ORole SET name = 'customer', mode = 0;
```

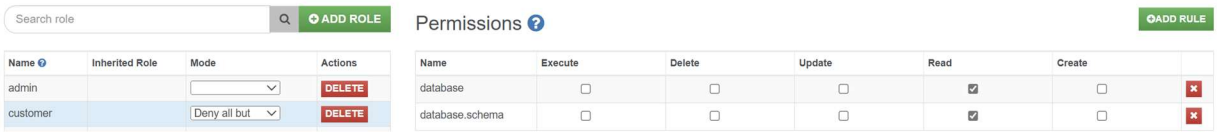
```
orientdb {db=HotelDB}> INSERT INTO ORole SET name = 'customer', mode = 0;

Inserted record '[ORole#5:5{name:customer,mode:0} v1]' in 0.010000 sec(s).
```

Cấp quyền được đăng nhập cho khách hàng khi cấp quyền này user có role như thế sẽ được thế các class

GRANT READ ON database TO customer;

GRANT READ ON database.schema TO customer;

```
orientdb {db=HotelDB}> GRANT READ ON database TO customer;

Privilege granted to the role: [{operation:grant,role:customer,permission:READ,resource:database}].

orientdb {db=HotelDB}> GRANT READ ON database.schema TO customer;

Privilege granted to the role: [{operation:grant,role:customer,permission:READ,resource:database.schema}].
```



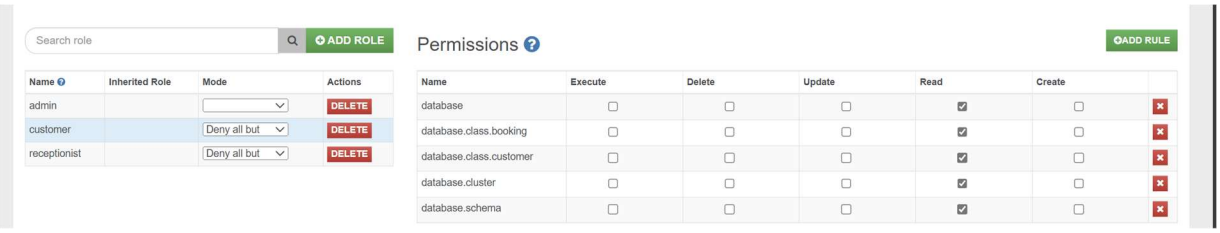
cho phép khách hàng đọc thông tin bảng Customer và bảng Booking

GRANT READ ON database.class.customer TO customer;

GRANT READ ON database.class.booking TO customer;

GRANT READ ON database.cluster TO customer;

```
orientdb {db=HotelDB}> GRANT READ ON database.class.customer TO customer;
GRANT READ ON database.class.booking TO customer;

Privilege granted to the role: [{operation:grant,role:customer,permission:READ,resource:database.class.customer}]

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:customer,permission:READ,resource:database.class.booking}].
```

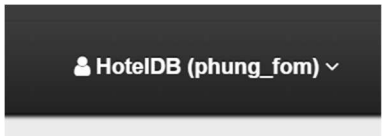


3.2.4. Tạo user

Kết nối bằng user quản lí, chỉ có quản lý mới có thể tạo user cho khách hàng lẫn lễ tân

CONNECT remote:localhost/HotelDB phung_fom phungfom;

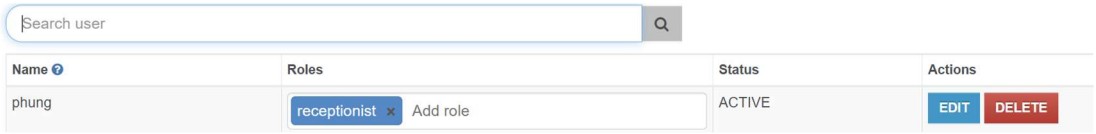
```
orientdb {db=HotelDB}> CONNECT remote:localhost/HotelDB phung_fom phungfom;

Disconnecting from the database [HotelDB]...OK

Connecting to database [remote:localhost/HotelDB] with user 'phung_fom'...OK
```



3.2.4.1 Phung (Receptionist)

```
INSERT INTO OUser SET name = 'phung', password = 'phung', roles =
(SELECT FROM ORole WHERE name = 'receptionist'), status = 'ACTIVE';
```

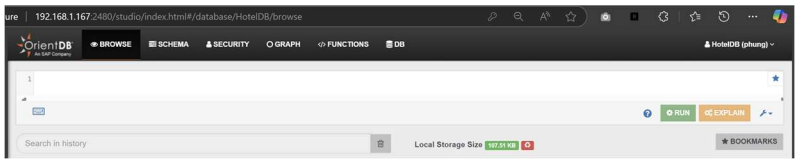
```
orientdb {db=HotelDB}> INSERT INTO OUser SET name = 'phung', password = 'phung', roles = (SELECT FROM ORole WHERE name = 'receptionist'), status = 'ACTIVE';

Inserted record '[OUser#6:10{name:phung,password:{PBKDF2WithHmacSHA256}A7894F19555F1898BDBB842B434E290E3083589E709E8D0C:13119AFDDDA671C6FEC1D2D1B28CA86FB2534C7178
770F66:65536,roles:[1],status:ACTIVE} v1]' in 0.273000 sec(s).
```




```
CONNECT remote:localhost/HotelDB phung phung;
```

```
orientdb {db=HotelDB}> CONNECT remote:localhost/HotelDB phung_fom phungfom;

Disconnecting from the database [HotelDB]...OK

Connecting to database [remote:localhost/HotelDB] with user 'phung_fom'...OK
```

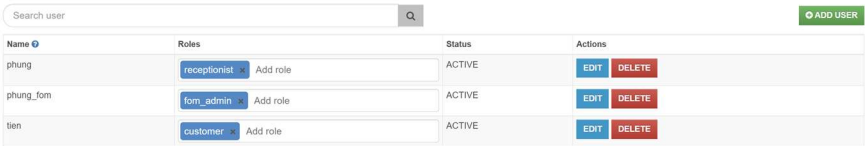
- kiểm tra trên UI xem có thể đăng nhập được không



3.2.4.2 Tien (Customer)

```
INSERT INTO OUser SET name = 'tien', password = 'tien', roles = (SELECT
FROM ORole WHERE name = 'customer'), status = 'ACTIVE';
```

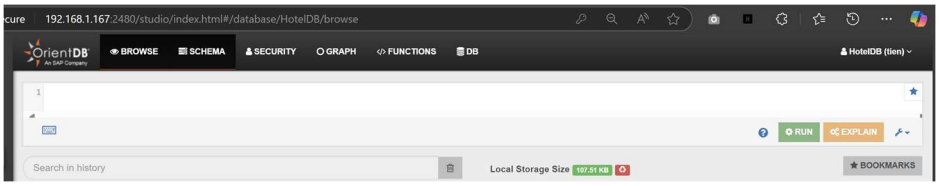
```
orientdb {db=HotelDB}> INSERT INTO OUser SET name = 'tien', password = 'tien', roles = (SELECT FROM ORole WHERE name = '
customer'), status = 'ACTIVE';

Inserted record '[OUser#6:9{name:tien,password:{PBKDF2WithHmacSHA256}CC8D63CE67B692C3516D8AF75EA9A43396BF0AF8722CAA04:06
61CA138994F74603EC7CFEDF4B65371533D6E227168806:65536,roles:[1],status:ACTIVE} v1]' in 0.122000 sec(s).
```




```
CONNECT remote:localhost/HotelDB tien tien;
```

```
orientdb {db=HotelDB}> CONNECT remote:localhost/HotelDB tien tien;
Disconnecting from the database [HotelDB]...OK
Connecting to database [remote:localhost/HotelDB] with user 'tien'...OK
```

kiểm tra trên UI xem có thể đăng nhập được không



3.2.5. Kiểm tra lại các user

3.2.5.1 phung_fom (admin)

-Kết nối bằng user phung_fom

```
CONNECT remote:localhost/HotelDB phung_fom phungfom;
```

-thêm thông tin vào bảng phòng

```
INSERT INTO Room SET Cost = 100, RoomID = 'P101', TypeRoomName = 'Deluxe'
INSERT INTO Room SET Cost = 150, RoomID = 'R102', TypeRoomName = 'Standard'
INSERT INTO Room SET Cost = 200, RoomID = 'R103', TypeRoomName = 'Luxury'
```

```
orientdb {db=HotelDB}> INSERT INTO Room SET Cost = 100, RoomID = 'R101', TypeRoomName = 'Deluxe'
Inserted record '[Room#74:0{Cost:100,RoomID:R101,TypeRoomName:Deluxe} v1]' in 0.029000 sec(s).
orientdb {db=HotelDB}> INSERT INTO Room SET Cost = 150, RoomID = 'R102', TypeRoomName = 'Standard'
Inserted record '[Room#75:0{Cost:150,RoomID:R102,TypeRoomName:Standard} v1]' in 0.014000 sec(s).
orientdb {db=HotelDB}> INSERT INTO Room SET Cost = 200, RoomID = 'R103', TypeRoomName = 'Luxury'
Inserted record '[Room#76:0{Cost:200,RoomID:R103,TypeRoomName:Luxury} v1]' in 0.009000 sec(s).
```

SELECT * FROM Room

METADATA			PROPERTIES		
@rid	@version	@class	TypeRoomName	RoomID	Cost
#74:0	1	Room	Deluxe	R101	100
#75:0	1	Room	Standard	R102	150
#76:0	1	Room	Luxury	R103	200

10 25 50 100 1000 5000

chỉnh sửa thông tin bảng phòng

UPDATE Room SET Cost = 120, TypeRoomName = 'Suite' WHERE RoomID = 'R101'

```
orientdb {db=HotelDB}> UPDATE Room SET Cost = 120, TypeRoomName = 'Suite' WHERE RoomID = 'R101'

Updated record(s) '[{count:1}]' in 0.016000 sec(s).
```

COMMAND

SELECT * FROM Room

METADATA			PROPERTIES		
@rid	@version	@class	TypeRoomName	RoomID	Cost
#74:0	2	Room	Suite	R101	120
#75:0	1	Room	Standard	R102	150
#76:0	1	Room	Luxury	R103	200

10 25 50 100 1000 5000

qua đó ta có thể thấy phòng R101 có version 2 tức là đã qua chỉnh sửa 1 lần

thêm thông tin khách hàng cần tạo user trước

INSERT INTO OUser SET name = 'hao', password = 'hao', roles = (SELECT FROM ORole WHERE name = 'customer'), status = 'ACTIVE';

INSERT INTO Customer SET Addresss = '123 Main St', BirthDate = '2003-05-15', FullName = 'Nguyen Van Hao', Gender = 'Male', ID = (SELECT FROM OUser WHERE name = 'hao'), Mail = 'haone@example.com', Phone = '123-456-7890'

```
orientdb {db=HotelDB}> INSERT INTO OUser SET name = 'hao', password = 'hao', roles = (SELECT FROM ORole WHERE name = 'customer'), status = 'AC
TIVE';
INSERT INTO Customer SET Addresss = '123 Main St', BirthDate = '2003-05-15', FullName = 'Nguyen Van Hao', Gender = 'Male', ID = (SELECT FROM O
User WHERE name = 'hao'), Mail = 'haone@example.com', Phone = '123-456-7890'

Inserted record '[OUser#6:7{name:hao,password:{PBKDF2WithHmacSHA256}2D6C20F4C9D2A76340553D29F946F612C4598C5DAEB9D64C:F1DA3132661DD0E41872E6461
5FAD16E975EC360A4EF17FB:65536,roles:[1],status:ACTIVE} v1]' in 0.140000 sec(s).

orientdb {db=HotelDB}>
Inserted record '[Customer#58:0{Addresss:123 Main St,BirthDate:Thu May 15 00:00:00 ICT 2003,FullName:Nguyen Van Hao,Gender:Male,ID:#6:7,Mail:h
aone@example.com,Phone:123-456-7890} v1]' in 0.021000 sec(s).
```

select * from `Customer`

METADATA			PROPERTIES						
@rid	@version	@class	Mail	Phone	FullName	Addresss	Gender	ID	BirthDate
#58:0	1	Customer	haone@example.com	123-456-7890	Nguyen Van Hao	123 Main St	Male	#6:7	2003-05-15 00:00:00

10 25 50 100 1000 5000

chỉnh sửa thông tin khách hàng

UPDATE Customer SET Mail = 'haonguyen@gmail.com' WHERE FullName = 'Nguyen Van Hao'

```
orientdb {db=HotelDB}> UPDATE Customer SET Mail = 'haonguyen@gmail.com' WHERE FullName = 'Nguyen Van Hao'

Updated record(s) '[{count:1}]' in 0.018000 sec(s).
```

COMMAND

select * from `Customer`

METADATA			PROPERTIES						
@rid	@version	@class	Mail	Phone	FullName	Addresss	Gender	ID	BirthDate
#58:0	2	Customer	haonguyen@gmail.com	123-456-7890	Nguyen Van Hao	123 Main St	Male	#6:7	2003-05-15 00:00:00

10 25 50 100 1000 5000

thêm đơn đặt phòng

```
INSERT INTO Booking SET BookDate = sysdate(), BookID = 'B001', CheckInDate = '2024-11-10 14:00:00', CheckOutDate = '2024-11-15 12:00:00', CustomerID = (SELECT FROM Customer WHERE FullName = 'Nguyen Van Hao'), RoomID = (SELECT FROM Room WHERE RoomID = 'R102'), Status = 'Confirmed', TotalCost = 120.00
```

```
orientdb {db=HotelDB}> INSERT INTO Booking SET BookDate = sysdate(), BookID = 'B001', CheckInDate = '2024-11-10 14:00:00', CheckOutDate = '202
4-11-15 12:00:00', CustomerID = (SELECT FROM Customer WHERE FullName = 'Nguyen Van Hao'), RoomID = (SELECT FROM Room WHERE RoomID = 'R102'), S
tatus = 'Confirmed', TotalCost = 120.00

Inserted record '[Booking#90:0{BookDate:Thu Nov 07 00:35:26 ICT 2024,BookID:B001,CheckInDate:Sun Nov 10 14:00:00 ICT 2024,CheckOutDate:Fri Nov
 15 12:00:00 ICT 2024,CustomerID:#58:0,RoomID:#75:0,Status:Confirmed,TotalCost:120.0} v1]' in 0.032000 sec(s).
```

COMMAND

select * from `Booking`

METADATA			PROPERTIES							
@rid	@version	@class	Status	TotalCost	BookID	CheckInDate	BookDate	CustomerID	RoomID	CheckOutDate
#90:0	1	Booking	Confirmed	120	B001	2024-11-10 14:00:00	2024-11-07 00:35:26	#58:0	#75:0	2024-11-15 12:00:00

10 25 50 100 1000 5000

3.2.5.2 Phung (Receptionist)

Kết nối bằng user phung_fom

```
CONNECT remote:localhost/HotelDB phung phung;
```

```
orientdb {db=HotelDB}> CONNECT remote:localhost/HotelDB phung phung;

Disconnecting from the database [HotelDB]...OK

Connecting to database [remote:localhost/HotelDB] with user 'phung'...OK
```

thêm thông tin vào bảng phòng

```
INSERT INTO Room SET Cost = 100, RoomID = 'P101', TypeRoomName = 'Deluxe'
```

```
orientdb {db=HotelDB}> INSERT INTO Room SET Cost = 100, RoomID = 'P101', TypeRoomName = 'Deluxe'

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'phung' does not have permission to execute the operation 'C
reate' against the resource: ResourceGeneric [name=CLASS, legacyName=database.class].Room
        DB name="HotelDB"
        DB name="HotelDB"
```

Không có quyền thêm loại phòng

- để tránh trường hợp lễ tân tự ý xóa phòng hoặc sửa giá tiền

Thêm thông tin khách hàng trường hợp có khách hàng mới

INSERT INTO OUser SET name = 'ngoc', password = 'ngoc', roles = (SELECT FROM ORole WHERE name = 'customer'), status = 'ACTIVE';

INSERT INTO Customer SET Addresss = '123 Tan Lap St', BirthDate = '2002-05-15', FullName = 'Bui Thanh Ngoc', Gender = 'Male', ID = (SELECT FROM OUser WHERE name = 'ngoc'), Mail = 'ngoc@example.com', Phone = '123-456-6890'

```
orientdb {db=HotelDB}> INSERT INTO Information SET Addresss = '123 Tan Lap St', BirthDate = '2002-05-15', FullName = 'Bu
i Thanh Ngoc', Gender = 'Male', ID = (SELECT FROM OUser WHERE name = 'ngoc'), Mail = 'ngoc@example.com', Phone = '123-45
6-6890'

Inserted record '[Information#61:0{Addresss:123 Tan Lap St,BirthDate:Wed May 15 00:00:00 ICT 2002,FullName:Bui Thanh Ngo
c,Gender:Male,ID:#6:12,Mail:ngoc@example.com,Phone:123-456-6890} v1]' in 0.011000 sec(s).
```

COMMAND

select * from `Customer`

METADATA			PROPERTIES						
@rid	@version	@class	Mail	Phone	FullName	Addresss	Gender	ID	BirthDate
#58:0	2	Customer	haonguyen@gmail.com	123-456-7890	Nguyen Van Hao	123 Main St	Male	#6:7	2003-05-15 00:00:00
#59:0	1	Customer	ngoc@example.com	123-456-6890	Bui Thanh Ngoc	123 Tan Lap St	Male	#6:5	2002-05-15 00:00:00

10 25 50 100 1000 5000

- xóa khách hàng ‘Hào’

DELETE FROM Customer WHERE FullName = 'Nguyen Van Hao'

```
orientdb {db=HotelDB}> DELETE FROM Customer WHERE FullName = 'Nguyen Van Hao'

Error: com.orientechnologies.orient.core.exception.OCommandExecutionException: Cannot safely delete a vertex, please use DELETE VERTEX or UNSA
FE
        DB name="HotelDB"
        DB name="HotelDB"
```

Không có quyền xóa khách hàng

- sửa thông tin khách hàng

UPDATE Customer SET Mail = 'vanhao@gmail.com' WHERE FullName = 'Nguyen Van Hao'

```
orientdb {db=HotelDB}> UPDATE Customer SET Mail = 'vanhao@gmail.com' WHERE FullName = 'Nguyen Van Hao'

Updated record(s) '[{count:1}]' in 0.006000 sec(s).

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'phung' does not have permission to execute the operation 'R
ead' against the resource: ResourceGeneric [name=SYSTEM_CLUSTER, legacyName=database.systemclusters].null
        DB name="HotelDB"
        DB name="HotelDB"
```

 Không có quyền sửa khách hàng

- Thêm hóa đơn đặt hàng

INSERT INTO Booking SET BookDate = sysdate(), BookID = 'B002',

CheckInDate = '2025-1-10 14:00:00', CheckOutDate = '2025-1-15 12:00:00', CustomerID = (SELECT FROM Customer WHERE FullName = 'Bui Thanh Ngoc'), RoomID = (SELECT FROM Room WHERE RoomID = 'R102'), Status = 'Confirmed', TotalCost = 120.00

```
orientdb {db=HotelDB}> INSERT INTO Booking SET BookDate = sysdate(), BookID = 'B002', CheckInDate = '2025-1-10 14:00:00', CheckOutDate = '2025
-1-15 12:00:00', CustomerID = (SELECT FROM Customer WHERE FullName = 'Bui Thanh Ngoc'), RoomID = (SELECT FROM Room WHERE RoomID = 'R102'), Sta
tus = 'Confirmed', TotalCost = 120.00

Inserted record '[Booking#91:0{BookDate:Thu Nov 07 00:43:22 ICT 2024,BookID:B002,CheckInDate:Fri Jan 10 14:00:00 ICT 2025,CheckOutDate:Wed Jan
 15 12:00:00 ICT 2025,CustomerID:#59:0,RoomID:#75:0,Status:Confirmed,TotalCost:120.0} v1]' in 0.019000 sec(s).
```

select * from `Booking`

select * from `Booking`

METADATA			PROPERTIES							
@rid	@version	@class	Status	TotalCost	BookID	CheckInDate	BookDate	CustomerID	RoomID	CheckOutDate
#90:0	1	Booking	Confirmed	120	B001	2024-11-10 14:00:00	2024-11-07 00:35:26	#58:0	#75:0	2024-11-15 12:00:00
#91:0	1	Booking	Confirmed	120	B002	2025-01-10 14:00:00	2024-11-07 00:43:22	#59:0	#75:0	2025-01-15 12:00:00

10 25 50 100 1000 5000

- Sửa totalcost

UPDATE Booking SET TotalCost = '20000' WHERE BookID = 'B002'

```
orientdb {db=HotelDB}> UPDATE Booking SET TotalCost = '20000' WHERE BookID = 'B002'

Updated record(s) '[{count:1}]' in 0.010000 sec(s).

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'phung' does not have permission to execute the operation 'R
ead' against the resource: ResourceGeneric [name=SYSTEM_CLUSTER, legacyName=database.systemclusters].null
        DB name="HotelDB"
        DB name="HotelDB"
```

Không thể chỉnh sửa booking trường hợp tự ý đổi giá tiền

3.2.5.3 Tien (Customer)

CONNECT remote:localhost/HotelDB tien tien;

```
orientdb> CONNECT remote:localhost/HotelDB tien tien;

Connecting to database [remote:localhost/HotelDB] with user 'tien'...OK
```

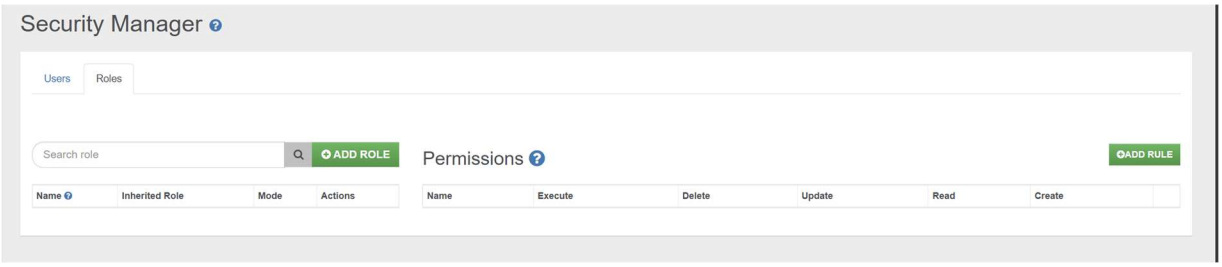
Security Manager

Users Roles

Search user

ADD USER

Name	Roles	Status	Actions



khách hàng không xem được gì ở phần security

UPDATE Room SET Cost = 120, TypeRoomName = 'Suite' WHERE RoomID = 'P101'

```
orientdb {db=HotelDB}> UPDATE Room SET Cost = 120, TypeRoomName = 'Suite' WHERE RoomID = 'P101'

Updated record(s) '[{count:0}]' in 0.004000 sec(s).

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'tien' does not have permission to execute the operation 'Re
ad' against the resource: ResourceGeneric [name=SYSTEM_CLUSTER, legacyName=database.systemclusters].null
        DB name="HotelDB"
        DB name="HotelDB"
```

không sửa thông tin bảng room

DELETE FROM Room WHERE RoomID = 'P101'

```
orientdb {db=HotelDB}> DELETE FROM Room WHERE RoomID = 'P101'

Delete record(s) '[{count:0}]' in 0.010000 sec(s).

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'tien' does not have permission to execute the operation 'Re
ad' against the resource: ResourceGeneric [name=SYSTEM_CLUSTER, legacyName=database.systemclusters].null
        DB name="HotelDB"
        DB name="HotelDB"
```

không xóa được

UPDATE Customer SET Mail = 'thanhthanh@gmail.com' WHERE FullName = 'Tran Van Tien'

```
orientdb {db=HotelDB}> UPDATE Customer SET Mail = 'thanhthanh@gmail.com' WHERE FullName = 'Tran Van Tien'

Updated record(s) '[{count:0}]' in 0.010000 sec(s).

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'tien' does not have permission to execute the operation 'Re
ad' against the resource: ResourceGeneric [name=SYSTEM_CLUSTER, legacyName=database.systemclusters].null
        DB name="HotelDB"
        DB name="HotelDB"
```

không có quyền update

DELETE FROM Customer WHERE FullName = 'Tran Van Tien'

```
orientdb {db=HotelDB}> DELETE FROM Customer WHERE FullName = 'Tran Van Tien'

Delete record(s) '[{count:0}]' in 0.008000 sec(s).

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'tien' does not have permission to execute the operation 'Re
ad' against the resource: ResourceGeneric [name=SYSTEM_CLUSTER, legacyName=database.systemclusters].null
        DB name="HotelDB"
        DB name="HotelDB"
```

không xóa được

UPDATE Booking SET CheckOutDate = '2025-01-16 12:00:00' WHERE BookID = 'B002'

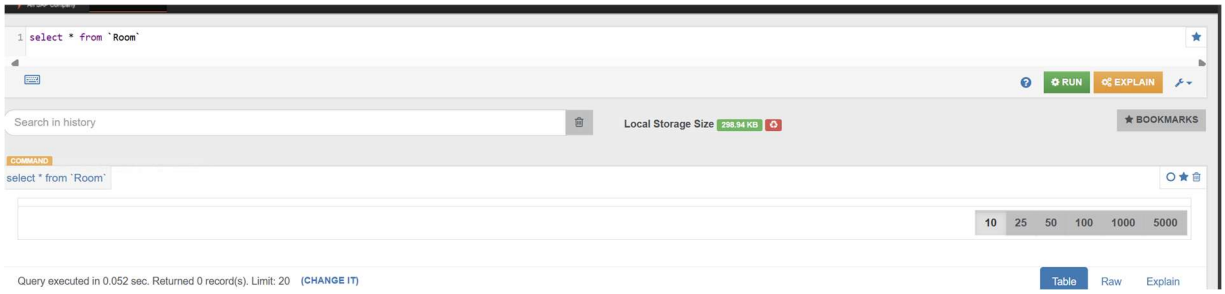
```
orientdb {db=HotelDB}> UPDATE Booking SET CheckOutDate = '2025-01-16 12:00:00' WHERE BookID = 'B002'

Updated record(s) '[{count:0}]' in 0.000000 sec(s).

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'tien' does not have permission to execute the operation 'Re
ad' against the resource: ResourceGeneric [name=SYSTEM_CLUSTER, legacyName=database.systemclusters].null
        DB name="HotelDB"
        DB name="HotelDB"
```

không có update được

DELETE FROM Booking WHERE BookID = 'B002'

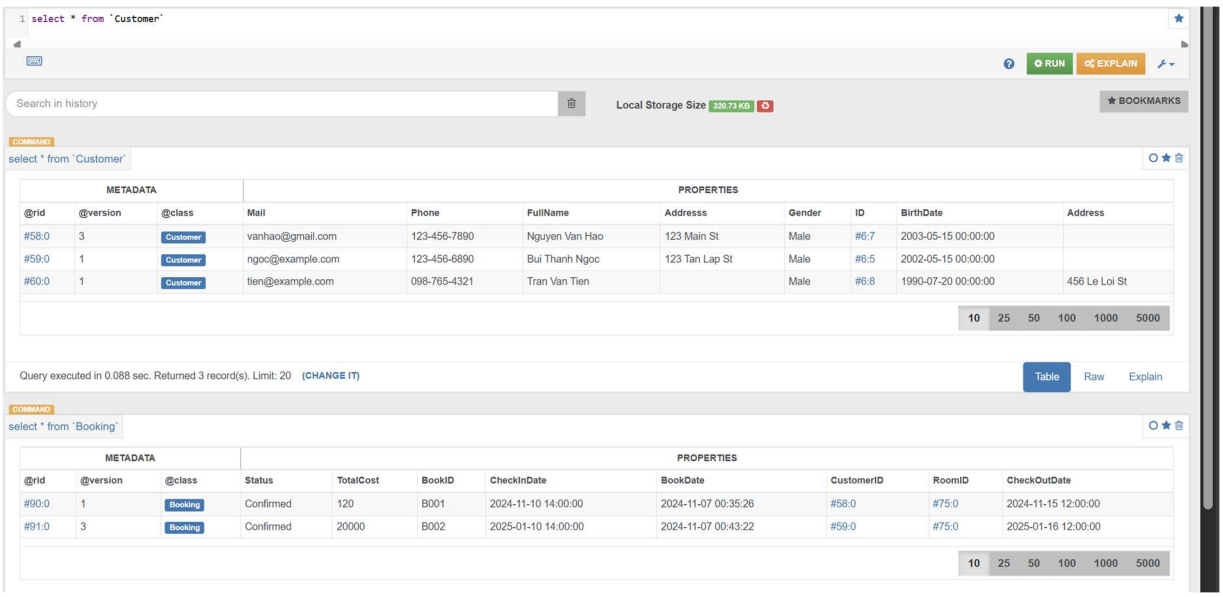
```
orientdb {db=HotelDB}> DELETE FROM Booking WHERE BookID = 'B002'

Delete record(s) '[{count:0}]' in 0.011000 sec(s).

Error: com.orientechnologies.orient.core.exception.OSecurityAccessException: User 'tien' does not have permission to execute the operation 'Re
ad' against the resource: ResourceGeneric [name=SYSTEM_CLUSTER, legacyName=database.systemclusters].null
        DB name="HotelDB"
        DB name="HotelDB"
```

không xóa được



không xem được thông tin các phòng

Khách hàng chỉ có thể xem bảng Customer và bảng Booking



3.3 Demo Employee Manager

Mô tả:

Quản lý nhân viên (user hao_ems) có quyền xem và điều chỉnh thông tin của tất cả nhân viên, còn trong nhân viên có thể có nhân viên chỉ xem và có nhân viên xem và sửa thông tin. Ở phần này, chỉ dừng lại ở việc nhân viên sửa thông tin bảng dựa trên quyền của Role, phần sau sẽ thực hiện chi tiết về việc chỉnh sửa thông tin của bảng ở mức dòng (Nhân viên chỉ được chỉnh sửa của bản thân họ). OrientDB không hỗ trợ kiểm soát quyền truy cập chi tiết đến mức cột (thuộc tính) trực tiếp thông qua lệnh GRANT (Nhân viên được chỉnh sửa tất cả thông tin của mình, không ngoại trừ một thuộc tính nào). Hao_ems là user do phuthanhadmin tạo

```
orientdb {db=HotelDB}> create user hao_ems identified by 'hao_ems' ROLE admin;

User created successfully.

orientdb {db=HotelDB}> select * from OUser

+----+----+------+------+------+-------+------------------
```

```
CREATE PROPERTY Information.FullName STRING (MANDATORY TRUE, NOTNULL
TRUE);
CREATE PROPERTY Information.Gender STRING (MANDATORY TRUE, NOTNULL
TRUE);
CREATE PROPERTY Information.BirthDate DATE (MANDATORY TRUE, NOTNULL
TRUE);
CREATE PROPERTY Information.Mail STRING (MANDATORY TRUE, NOTNULL
TRUE);
CREATE PROPERTY Information.Phone STRING (MANDATORY TRUE, NOTNULL
TRUE);
CREATE PROPERTY Information.Addresss STRING;

CREATE INDEX Information.ID UNIQUE;
```

```
orientdb> connect remote:localhost/HotelDB hao_ems hao_ems

Connecting to database [remote:localhost/HotelDB] with user 'hao_ems'...OK
orientdb {db=HotelDB}>

orientdb {db=HotelDB}> CREATE CLASS Information EXTENDS V;

Class created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Information.ID INTEGER (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Information.FullName STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Information.Gender STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Information.BirthDate DATE (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Information.Mail STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Information.Phone STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Information.Addresss STRING;

Property created successfully.

orientdb {db=HotelDB}> CREATE INDEX Information.ID UNIQUE;


Creating index...
Created index successfully in 0.033000 sec(s).


Index created successfully
orientdb {db=HotelDB}>
```

- **Chèn 2 bản ghi vào class Information**

```
INSERT INTO Information SET ID = 1, FullName = 'Nguyen Van An',
Gender = 'Nam', BirthDate = '1990-01-01', Mail =
```

```
'nguyenvanan@gmail.com', Phone = '0123456789', Addresss = '123 Duong
ABC';

INSERT INTO Information SET ID = 2, FullName = 'Tran Thi Chi', Gender
= 'Nu', BirthDate = '1992-02-02', Mail = 'tranthichi@gmail.com',
Phone = '0987654321', Addresss = '456 Duong XYZ';
```

```
orientdb {db=HotelDB}> INSERT INTO Information SET ID = 1, FullName = 'Nguyen Van An', Gender = 'Nam', BirthDate = '1990-01-01', Mail = 'nguyenvanan@gmail.com', Phone = '0123456789', Addresss = '123 Duong ABC';

Inserted record '[Information#35:0{ID:1,FullName:Nguyen Van An,Gender:Nam,BirthDate:Mon Jan 01 00:00:00 ICT 1990,Mail:nguyenvanan@gmail.com,Phone:0123456789,Addresss:123 Duong ABC} v1]' in 0.003000 sec(s).

orientdb {db=HotelDB}> INSERT INTO Information SET ID = 2, FullName = 'Tran Thi Chi', Gender = 'Nu', BirthDate = '1992-02-02', Mail = 'tranthichi@gmail.com', Phone = '0987654321', Addresss = '456 Duong XYZ';

Inserted record '[Information#36:0{ID:2,FullName:Tran Thi Chi,Gender:Nu,BirthDate:Sun Feb 02 00:00:00 ICT 1992,Mail:tranthichi@gmail.com,Phone:0987654321,Addresss:456 Duong XYZ} v1]' in 0.004000 sec(s).

orientdb {db=HotelDB}> select * from Information

+----+-----+-----------+----+-------------+------+-------------------+---------------------+----------+-------------+
|#   |@RID |@CLASS     |ID  |FullName     |Gender|BirthDate          |Mail                 |Phone     |Addresss     |
+----+-----+-----------+----+-------------+------+-------------------+---------------------+----------+-------------+
|0   |#35:0|Information|1   |Nguyen Van An|Nam   |1990-01-01 00:00:00|nguyenvanan@gmail.com|0123456789|123 Duong ABC|
|1   |#36:0|Information|2   |Tran Thi Chi |Nu    |1992-02-02 00:00:00|tranthichi@gmail.com |0987654321|456 Duong XYZ|
+----+-----+-----------+----+-------------+------+-------------------+---------------------+----------+-------------+

2 item(s) found. Query executed in 0.003 sec(s).
orientdb {db=HotelDB}>
```

- **Class Employee:**

```
CREATE CLASS Employee EXTENDS V;
CREATE PROPERTY Employee.ID INTEGER (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Employee.Position STRING (MANDATORY TRUE, NOTNULL
TRUE);
CREATE PROPERTY Employee.Salary STRING (MANDATORY TRUE, NOTNULL
TRUE);

CREATE INDEX Employee.ID UNIQUE;

CREATE PROPERTY Employee.Information LINK Information;
```

```
orientdb {db=HotelDB}> CREATE CLASS Employee EXTENDS V;

Class created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Employee.ID INTEGER (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Employee.Position STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE PROPERTY Employee.Salary STRING (MANDATORY TRUE, NOTNULL TRUE);

Property created successfully.

orientdb {db=HotelDB}> CREATE INDEX Employee.ID UNIQUE;


Creating index...
Created index successfully in 0.031000 sec(s).


Index created successfully
orientdb {db=HotelDB}> CREATE PROPERTY Employee.Information LINK Information;

Property created successfully.

orientdb {db=HotelDB}>
```

- **Chèn 2 bản ghi vào class Employee:**

```
INSERT INTO Employee SET ID = 1, Position = 'Receptionist', Salary =
'1000', Information = (SELECT FROM Information WHERE ID = 1);

INSERT INTO Employee SET ID = 2, Position = 'Internship', Salary =
'600', Information = (SELECT FROM Information WHERE ID = 2);
```

```
orientdb {db=HotelDB}> INSERT INTO Employee SET ID = 1, Position = 'Receptionist', Salary = '1000', Information = (SELECT FROM Information WHERE ID = 1);

Inserted record '[Employee#46:0{ID:1,Position:Receptionist,Salary:1000,Information:#35:0} v1]' in 0.004000 sec(s).

orientdb {db=HotelDB}> INSERT INTO Employee SET ID = 2, Position = 'Internship', Salary = '600', Information = (SELECT FROM Information WHERE ID = 2);

Inserted record '[Employee#47:0{ID:2,Position:Internship,Salary:600,Information:#36:0} v1]' in 0.003000 sec(s).

orientdb {db=HotelDB}> select * from Employee

+----+-----+--------+----+------------+------+-----------+
|#   |@RID |@CLASS  |ID  |Position    |Salary|Information|
+----+-----+--------+----+------------+------+-----------+
|0   |#46:0|Employee|1   |Receptionist|1000  |#35:0      |
|1   |#47:0|Employee|2   |Internship  |600   |#36:0      |
+----+-----+--------+----+------------+------+-----------+

2 item(s) found. Query executed in 0.003 sec(s).
orientdb {db=HotelDB}>
```

3.3.2. Tạo role

Tạo 2 role **EmpRoleRead** và **EmpRoleModify**:

```
INSERT INTO ORole SET name = 'EmpRoleModify', mode = 0;

INSERT INTO ORole SET name = 'EmpRoleRead', mode = 0;
```

```
orientdb {db=HotelDB}> INSERT INTO ORole SET name = 'EmpRoleModify', mode = 0;

Inserted record '[ORole#5:21{name:EmpRoleModify,mode:0} v1]' in 0.002000 sec(s).

orientdb {db=HotelDB}> INSERT INTO ORole SET name = 'EmpRoleRead', mode = 0;

Inserted record '[ORole#5:22{name:EmpRoleRead,mode:0} v1]' in 0.003000 sec(s).

orientdb {db=HotelDB}> select * from ORole

+----+-----+-----+----+--------+---------+-------------------------------------------------+--------------------------------------------------------------+
|#   |@RID |@CLASS|mode|name    |inherited|rules                                            |policies                                                      |
+----+-----+-----+----+--------+---------+-------------------------------------------------+--------------------------------------------------------------+
|0   |#5:0 |ORole |    |admin   |         |{database=31, database.function=31, database.s...|{*:#4:0}                                                      |
|1   |#5:1 |ORole |    |reader  |         |{database.cluster.internal=2, database.cluster...|{database.class.*.*:#4:0, database:#4:1, database.schema...|
|2   |#5:2 |ORole |    |writer  |         |{database.cluster.internal=2, database.class.o...|{database.class.*.*:#4:0, database:#4:1, database.schema...|
|3   |#5:21|ORole |0   |EmpRol...|        |                                                 |                                                              |
|4   |#5:22|ORole |0   |EmpRol...|        |                                                 |                                                              |
+----+-----+-----+----+--------+---------+-------------------------------------------------+--------------------------------------------------------------+

5 item(s) found. Query executed in 0.003 sec(s).
orientdb {db=HotelDB}>
```

3.3.3 Tạo các user

Tạo ra **2** user tương ứng với 2 employee và gán role **EmpRoleRead**, **EmpRoleModify** cho từng user

```
CREATE USER emp_chi IDENTIFIED BY 'emp_chi' ROLE EmpRoleRead;

CREATE USER emp_an IDENTIFIED BY 'emp_an' ROLE EmpRoleModify;
```

```
orientdb {db=HotelDB}> CREATE USER emp_chi IDENTIFIED BY 'emp_chi' ROLE EmpRoleRead;

User created successfully.

orientdb {db=HotelDB}> CREATE USER emp_an IDENTIFIED BY 'emp_an' ROLE EmpRoleModify;

User created successfully.

orientdb {db=HotelDB}> select * from OUser

+----+-----+------+------+-------+------+--------------------------------------------------------------------------------------------------------+
|#   |@RID |@CLASS|status|name   |roles |password                                                                                                |
+----+-----+------+------+-------+------+--------------------------------------------------------------------------------------------------------+
|0   |#6:0 |OUser |ACTIVE|admin  |[#5:0] |{PBKDF2WithHmacSHA256}60DB0A27530406F6A76BD93B23E348FFB80544C308FF8A49:F2A2E32898DF3B378237717B0D5E30C20E07D2D...|
|1   |#6:1 |OUser |ACTIVE|reader |[#5:1] |{PBKDF2WithHmacSHA256}C7408FD2ED56B62A7D20E6E502082D11853BFE2012F9D5CD:4B859870DAEDBAE409F025C93FAD3F4A4B5DA37...|
|2   |#6:2 |OUser |ACTIVE|writer |[#5:2] |{PBKDF2WithHmacSHA256}196531DB321A25EF37B6D8C38E77114FDC3278FD762977B6:9A53138DC575EA8C10E126865CAD44DFE7A78BB...|
|3   |#6:3 |OUser |ACTIVE|hao_ems|[#5:0] |{PBKDF2WithHmacSHA256}5CCC49A87ED1A2B356BEACBFC1565CF4D5E0DE004EE6198E:3993182DA6915A8E86B3C20FC99E403BC274225...|
|4   |#6:19|OUser |ACTIVE|emp_chi|[#5:22]|{PBKDF2WithHmacSHA256}968100A2002D4CD9F902498E9EAFBFFA83C04340A3CC1702:941DC54D255C03030203761E05FB2A6B03C3281...|
|5   |#6:20|OUser |ACTIVE|emp_an |[#5:21]|{PBKDF2WithHmacSHA256}36D6DC6FAFEFBB54015B9406B259EE12D232C9866F2D85EA:3C4B394B707D0071690016D6DE968FA6B522B9F...|
+----+-----+------+------+-------+------+--------------------------------------------------------------------------------------------------------+

6 item(s) found. Query executed in 0.002 sec(s).
orientdb {db=HotelDB}>
```

3.3.4. Cấp các quyền thao tác trên class cho role

- Employee thuộc role EmpRoleRead có quyền đọc nội dung 2 class:

```
GRANT READ ON database.cluster TO EmpRoleRead;
GRANT READ ON database.class.Information TO EmpRoleRead;
GRANT READ ON database.class.Employee TO EmpRoleRead;
```

```
orientdb {db=HotelDB}> GRANT READ ON database.cluster TO EmpRoleRead;

Privilege granted to the role: [{operation:grant,role:EmpRoleRead,permission:READ,resource:database.cluster}].

orientdb {db=HotelDB}> GRANT READ ON database.class.Information TO EmpRoleRead;

Privilege granted to the role: [{operation:grant,role:EmpRoleRead,permission:READ,resource:database.class.Information}].

orientdb {db=HotelDB}> GRANT READ ON database.class.Employee TO EmpRoleRead;

Privilege granted to the role: [{operation:grant,role:EmpRoleRead,permission:READ,resource:database.class.Employee}].

orientdb {db=HotelDB}>
```

- Employee thuộc role EmpRoleModify có quyền xem và sửa thông tin, nhưng không sửa được ID và Salary và Position → không cấp quyền UPDATE trên bảng Employee cho EmpRoleModify

```
GRANT READ ON database.cluster TO EmpRoleModify;
GRANT UPDATE ON database.cluster TO EmpRoleModify;
GRANT READ ON database.class.Information TO EmpRoleModify;
GRANT READ ON database.class.Employee TO EmpRoleModify;
GRANT UPDATE ON database.class.Information TO EmpRoleModify;
```

```
orientdb {db=HotelDB}> GRANT READ ON database.cluster TO EmpRoleModify;

Privilege granted to the role: [{operation:grant,role:EmpRoleModify,permission:READ,resource:database.cluster}].

orientdb {db=HotelDB}> GRANT UPDATE ON database.cluster TO EmpRoleModify;

Privilege granted to the role: [{operation:grant,role:EmpRoleModify,permission:UPDATE,resource:database.cluster}].

orientdb {db=HotelDB}> GRANT READ ON database.class.Information TO EmpRoleModify;

Privilege granted to the role: [{operation:grant,role:EmpRoleModify,permission:READ,resource:database.class.Information}].

orientdb {db=HotelDB}> GRANT READ ON database.class.Employee TO EmpRoleModify;

Privilege granted to the role: [{operation:grant,role:EmpRoleModify,permission:READ,resource:database.class.Employee}].

orientdb {db=HotelDB}> GRANT UPDATE ON database.class.Information TO EmpRoleModify;

Privilege granted to the role: [{operation:grant,role:EmpRoleModify,permission:UPDATE,resource:database.class.Information}].

orientdb {db=HotelDB}>
```

3.3.5. Cấp các quyền cho Role để đăng nhập

- Cấp quyền cho role EmpRoleRead để user có thể đăng nhập vào trong database:

```
GRANT READ ON database TO EmpRoleRead;
GRANT READ ON database.schema to EmpRoleRead;
```

```
orientdb {db=HotelDB}> GRANT READ ON database TO EmpRoleRead;

Privilege granted to the role: [{operation:grant,role:EmpRoleRead,permission:READ,resource:database}].

orientdb {db=HotelDB}> GRANT READ ON database.schema to EmpRoleRead;

Privilege granted to the role: [{operation:grant,role:EmpRoleRead,permission:READ,resource:database.schema}].

orientdb {db=HotelDB}>
```

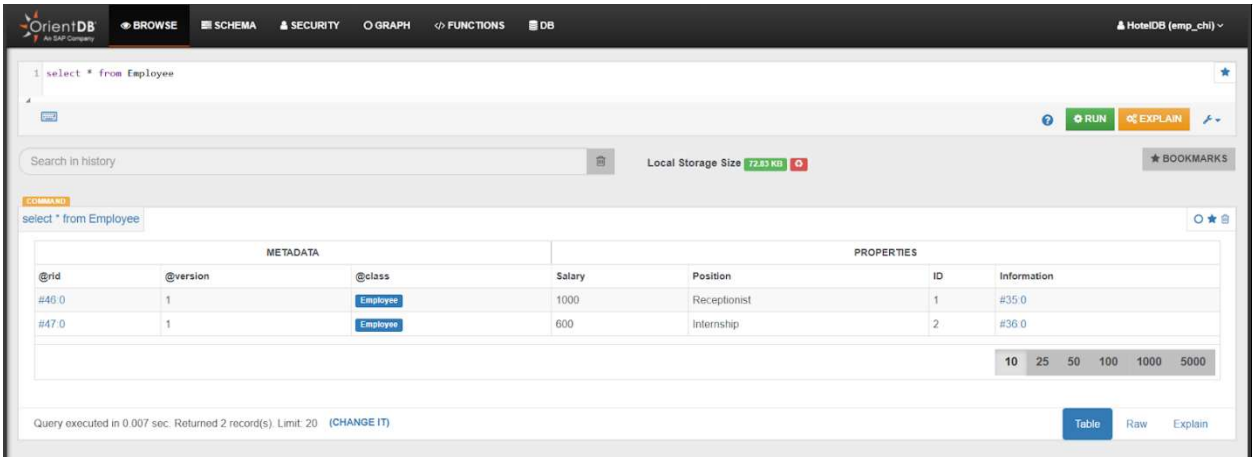
- Cấp quyền cho role EmpRoleModify để user có thể đăng nhập vào trong database:

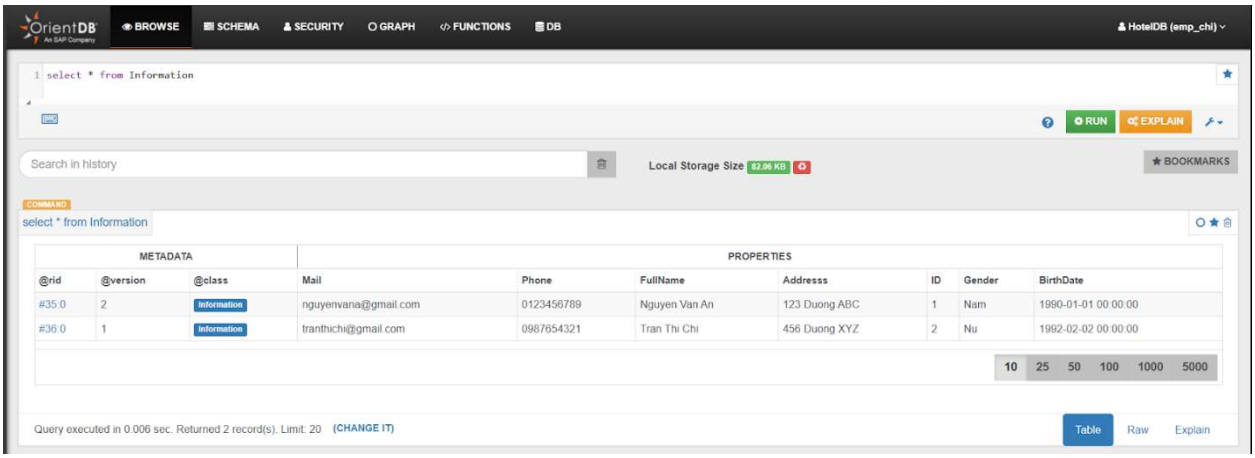
```
GRANT READ ON database TO EmpRoleModify;
GRANT READ ON database.schema to EmpRoleModify;
```

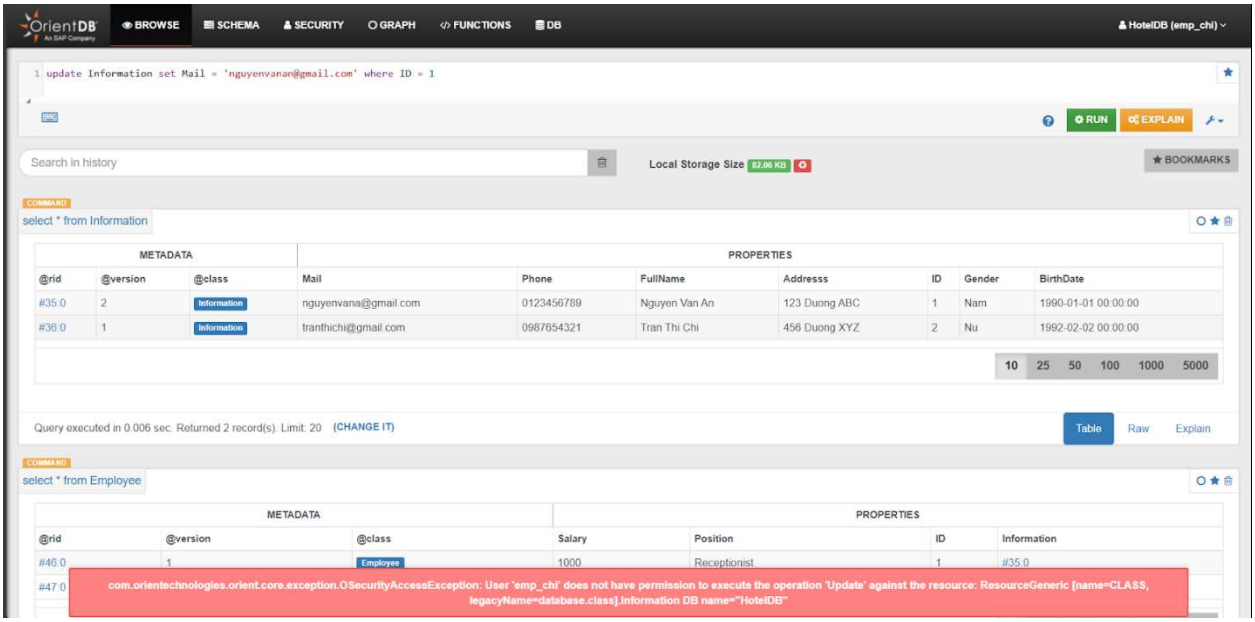
```
orientdb {db=HotelDB}> GRANT READ ON database TO EmpRoleModify;

Privilege granted to the role: [{operation:grant,role:EmpRoleModify,permission:READ,resource:database}].

orientdb {db=HotelDB}> GRANT READ ON database.schema to EmpRoleModify;

Privilege granted to the role: [{operation:grant,role:EmpRoleModify,permission:READ,resource:database.schema}].

orientdb {db=HotelDB}>
```

3.3.6. Kiểm tra kết quả

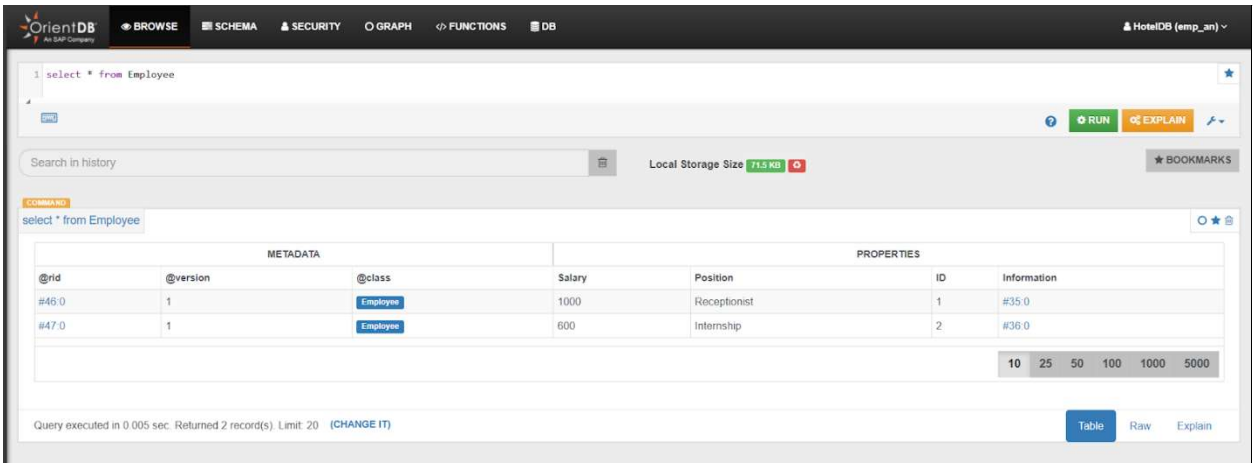
- **user emp_chi (thuộc role EmpRoleRead)**

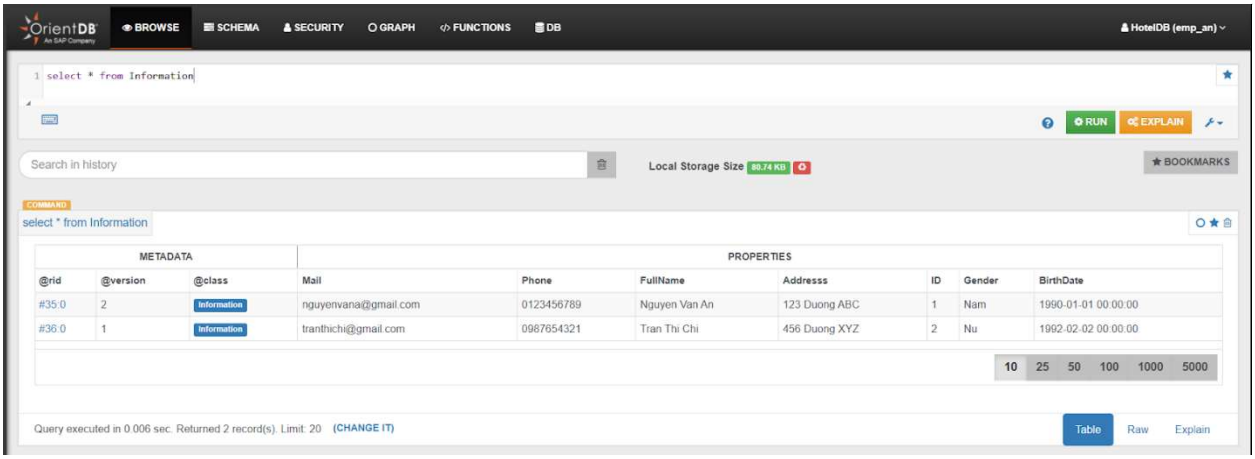


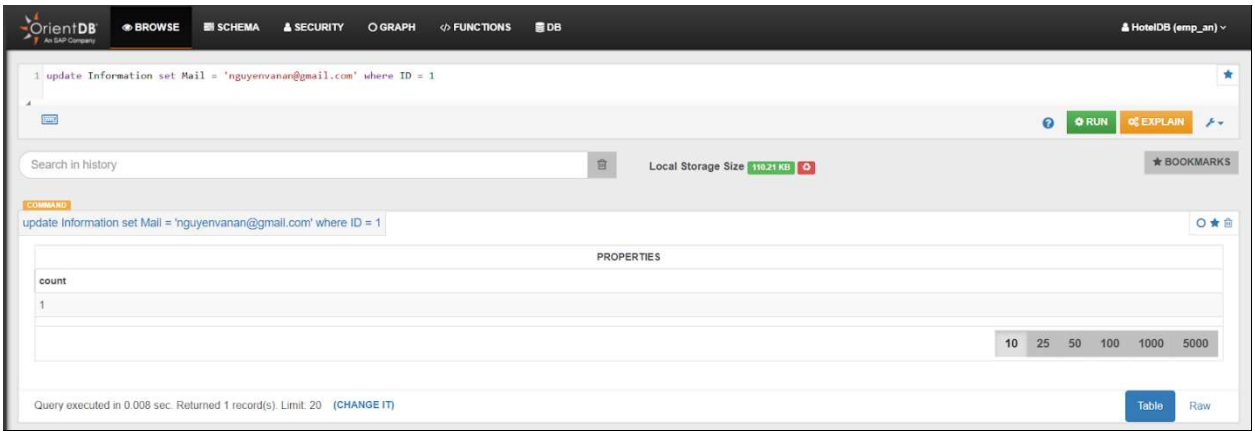




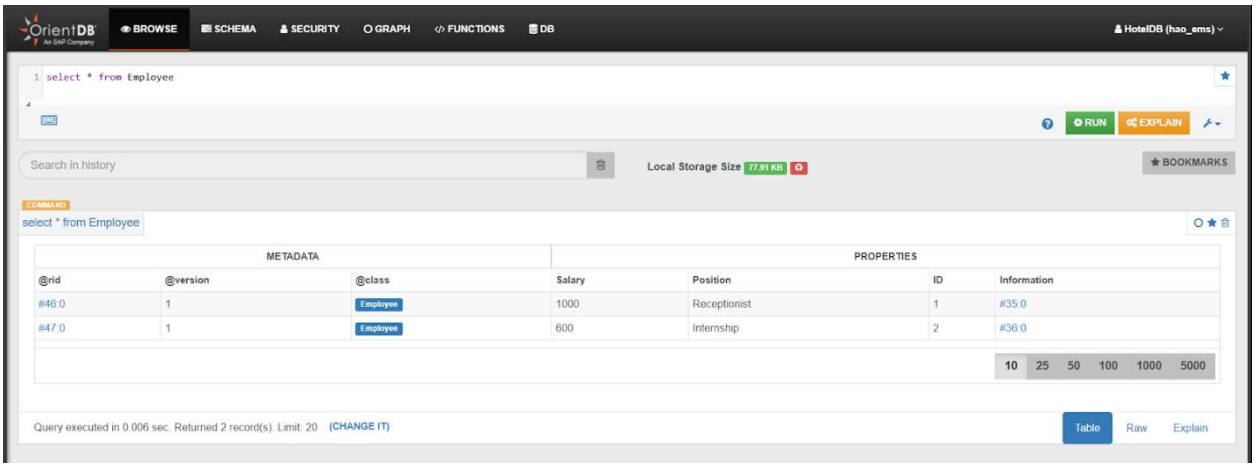
- **user emp_an (thuộc role EmpRoleModify)**

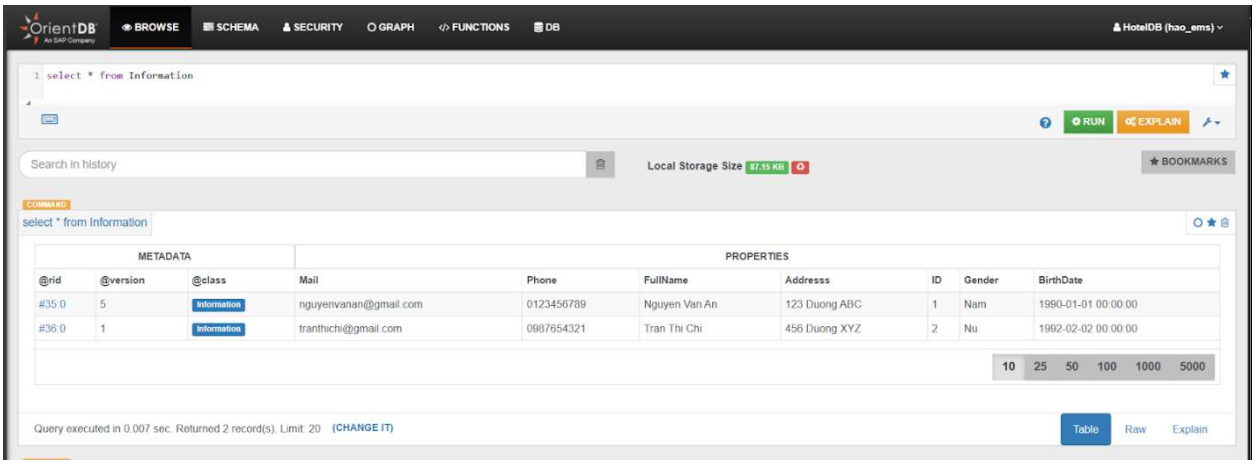


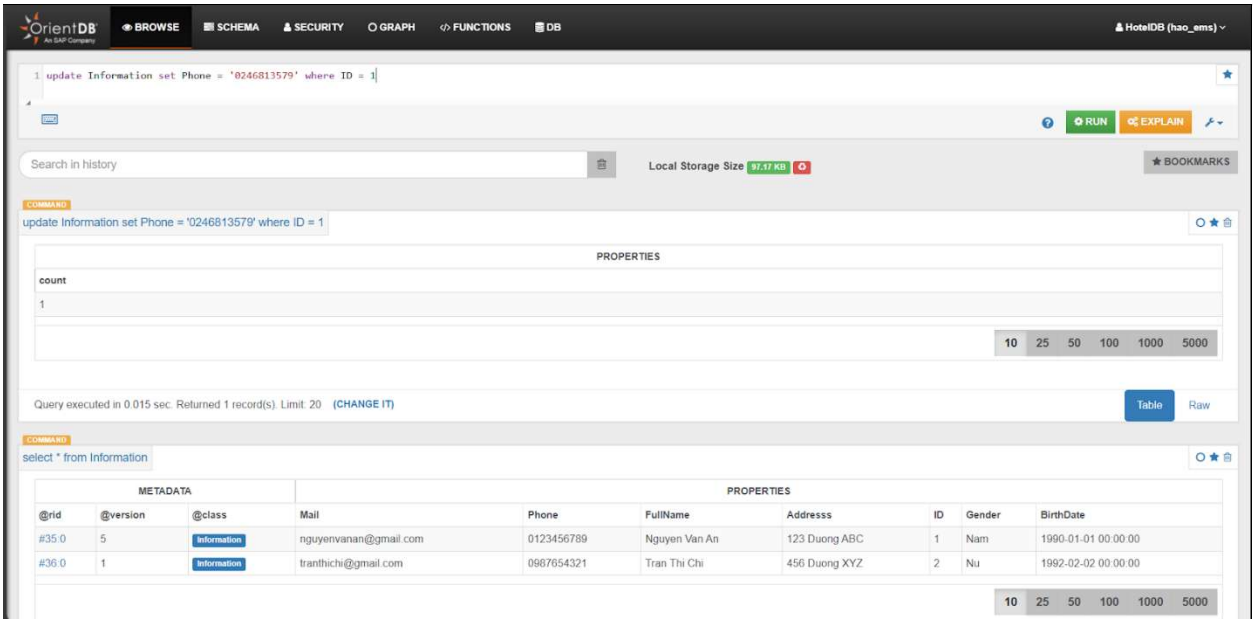




- **user hao_ems (admin)**


OrientDB
BROWSE
SCHEMA
SECURITY
GRAPH
FUNCTIONS
DB
HotelDB (hao_ems)
1 select * from Employee
RUN
EXPLAIN
Search in history
Local Storage Size 77.91 KB
BOOKMARKS
COMMAND
select * from Employee
METADATA
PROPERTIES
@rid @version @class Salary Position ID Information
#46:0 1 Employee 1000 Receptionist 1 #35:0
#47:0 1 Employee 600 Internship 2 #36:0
10 25 50 100 1000 5000
Query executed in 0.006 sec. Returned 2 record(s). Limit: 20 (CHANGE IT)
Table Raw Explain


OrientDB
BROWSE
SCHEMA
SECURITY
GRAPH
FUNCTIONS
DB
HotelDB (hao_ems)
1 select * from Information
RUN
EXPLAIN
Search in history
Local Storage Size 87.15 KB
BOOKMARKS
COMMAND
select * from Information
METADATA
PROPERTIES
@rid @version @class Mail Phone FullName Addresss ID Gender BirthDate
#35:0 5 Information nguyenvanan@gmail.com 0123456789 Nguyen Van An 123 Duong ABC 1 Nam 1990-01-01 00:00:00
#36:0 1 Information tranthichi@gmail.com 0987654321 Tran Thi Chi 456 Duong XYZ 2 Nu 1992-02-02 00:00:00
10 25 50 100 1000 5000
Query executed in 0.007 sec. Returned 2 record(s). Limit: 20 (CHANGE IT)
Table Raw Explain


OrientDB
BROWSE
SCHEMA
SECURITY
GRAPH
FUNCTIONS
DB
HotelDB (hao_ems)
1 update Information set Phone = '0246813579' where ID = 1
RUN
EXPLAIN
Search in history
Local Storage Size 97.17 KB
BOOKMARKS
COMMAND
update Information set Phone = '0246813579' where ID = 1
PROPERTIES
count
1
10 25 50 100 1000 5000
Query executed in 0.015 sec. Returned 1 record(s). Limit: 20 (CHANGE IT)
Table Raw
COMMAND
select * from Information
METADATA
PROPERTIES
@rid @version @class Mail Phone FullName Addresss ID Gender BirthDate
#35:0 5 Information nguyenvanan@gmail.com 0123456789 Nguyen Van An 123 Duong ABC 1 Nam 1990-01-01 00:00:00
#36:0 1 Information tranthichi@gmail.com 0987654321 Tran Thi Chi 456 Duong XYZ 2 Nu 1992-02-02 00:00:00
10 25 50 100 1000 5000

3.4 Demo Customer Staff Service

Mô tả:

Trong hệ thống quản lý khách sạn nhóm đã áp dụng bảo mật cấp bản ghi bằng cách mở rộng các bảng từ lớp ORestricted trong OrientDB. Điều này cho phép thiết lập quyền truy cập chi tiết cho từng người dùng trên từng bản ghi thông qua các thuộc tính _allowRead, _allowUpdate, và _allowDelete, đảm bảo rằng mỗi người dùng chỉ có thể truy cập thông tin mà họ được phép:

+ User hao_ems - quản trị viên với quyền truy cập toàn bộ.

+ Nhân viên emp_chi - có quyền chỉ đọc thông tin và xem lương của chính mình.

+ Nhân viên emp_an - có quyền đọc và sửa thông tin, xem lương của chính mình, đồng thời là nhân viên phục vụ, có thể xem đơn đặt hàng, khách hàng và phản hồi.

+ Customer - có quyền xem booking của chính mình, có quyền xem và chỉnh thông tin cá nhân kèm theo quyền thêm, xóa, sửa feedback .

3.4.1 Kết nối vào cơ sở dữ liệu HotelDB

```
CONNECT remote:localhost/HotelDB hao_ems hao_ems;
```

```
orientdb {db=HotelDB}>  CONNECT remote:localhost/HotelDB hao_ems hao_ems

Disconnecting from the database [HotelDB]...OK

Connecting to database [remote:localhost/HotelDB] with user 'hao_ems'...OK
orientdb {db=HotelDB}>
```

3.4.2 Bảo mật cao ở bảng Employee cho nhân viên chỉ xem chính mình

Tạo Lớp Employee:

```
CREATE CLASS Employee EXTENDS V, ORestricted;;
CREATE PROPERTY Employee.ID INTEGER (MANDATORY TRUE,
```

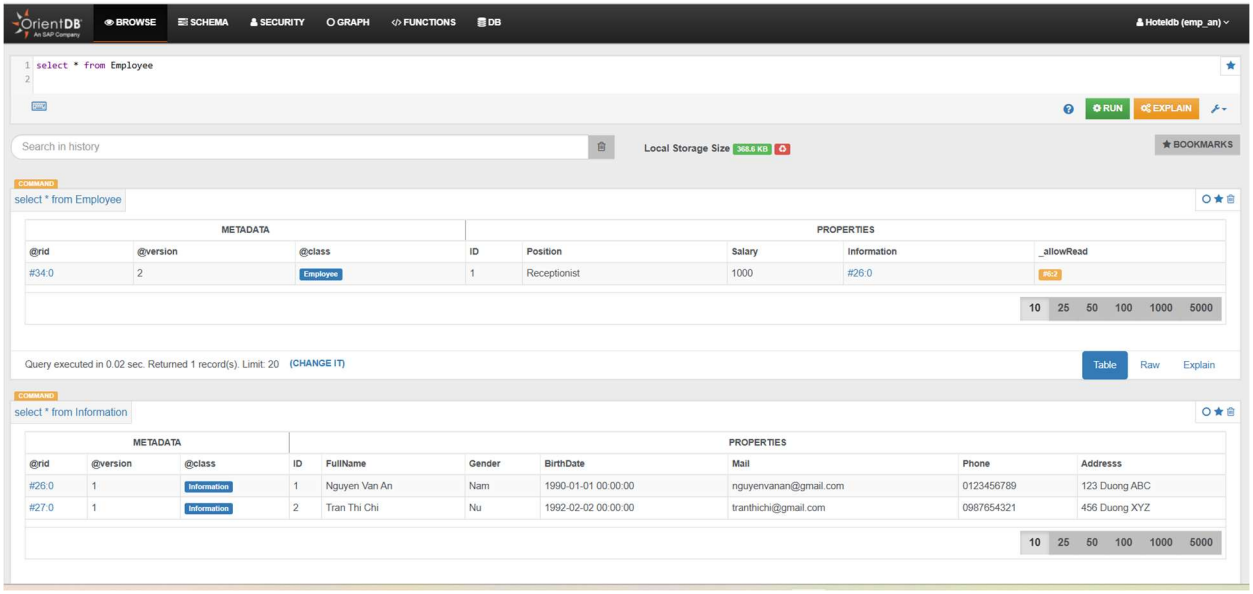
```
NOTNULL TRUE);
CREATE PROPERTY Employee.Position STRING (MANDATORY TRUE,
NOTNULL TRUE);
CREATE PROPERTY Employee.Salary STRING (MANDATORY TRUE,
NOTNULL TRUE);
CREATE INDEX Employee.ID UNIQUE;
CREATE PROPERTY Employee.Information LINK Information;
```

Chèn 2 bản ghi vào class Employee:

```
INSERT INTO Employee SET ID = 1, Position = 'Receptionist', Salary = '1000',
Information = (SELECT FROM Information WHERE ID = 1), _allowRead =
(SELECT @RID FROM OUser WHERE name = 'emp_an');
INSERT INTO Employee SET ID = 2, Position = 'Internship', Salary = '600',
Information = (SELECT FROM Information WHERE ID = 2), _allowRead =
(SELECT @RID FROM OUser WHERE name = 'emp_chi');
```

```
orientdb {db=Hoteldb}> INSERT INTO Employee SET ID = 1, Position = 'Receptionist', Salary = '1000', Information = (SELEC
T FROM Information WHERE ID = 1), _allowRead = (SELECT @RID FROM OUser WHERE name = 'emp_an');

Inserted record '[Employee#34:0{ID:1,Position:Receptionist,Salary:1000,Information:#26:0,_allowRead:[0]} v1]' in 0.01800
0 sec(s).

orientdb {db=Hoteldb}> INSERT INTO Employee SET ID = 2, Position = 'Internship', Salary = '600', Information = (SELECT F
ROM Information WHERE ID = 2), _allowRead = (SELECT @RID FROM OUser WHERE name = 'emp_chi');

Inserted record '[Employee#35:0{ID:2,Position:Internship,Salary:600,Information:#27:0,_allowRead:[0]} v1]' in 0.007000 s
ec(s).

orientdb {db=Hoteldb}>
```

An có thể xem thông tin ở tất cả ở bảng Information nhưng chỉ xem được Lương của chính mình



3.4.3 Tạo Các Lớp và Thuộc Tính

Tạo Lớp Customers:

```
-- Lớp lưu thông tin khách hàng
CREATE CLASS Customers EXTENDS V, ORestricted;
CREATE PROPERTY Customers.FullName STRING (MANDATORY TRUE,
NOTNULL TRUE);
CREATE PROPERTY Customers.Contact STRING;
```

```
orientdb {db=HotelDB}> CREATE CLASS Customers EXTENDS V, ORestricted;
CREATE PROPERTY Customers.FullName STRING (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Customers.Contact STRING;

Class created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
```

Tạo Lớp Booking:

```
-- Lớp lưu thông tin booking
CREATE CLASS Booking EXTENDS E, ORestricted;
```

```
CREATE PROPERTY Booking.BookID STRING (MANDATORY TRUE,
NOTNULL TRUE);
CREATE PROPERTY Booking.RoomID LINK Room (MANDATORY TRUE,
NOTNULL TRUE);
CREATE PROPERTY Booking.BookDate DATETIME (MANDATORY TRUE,
NOTNULL TRUE);
CREATE PROPERTY Booking.CheckInDate DATETIME (MANDATORY
TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.CheckOutDate DATETIME (MANDATORY
TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.TotalCost DECIMAL;
CREATE PROPERTY Booking.Status STRING;
CREATE INDEX Booking.BookID UNIQUE;
```

```
orientdb {db=Hoteldb}> CREATE CLASS Booking EXTENDS E, ORestricted;
CREATE PROPERTY Booking.BookID STRING (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.InformationID LINK Information (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.RoomID LINK Room (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.BookDate DATETIME (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.CheckInDate DATETIME (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.CheckOutDate DATETIME (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Booking.TotalCost DECIMAL;
CREATE PROPERTY Booking.Status STRING;

Class created successfully.

orientdb {db=Hoteldb}>
Property created successfully.

orientdb {db=Hoteldb}>
Property created successfully.

orientdb {db=Hoteldb}>
```

```
orientdb {db=Hoteldb}>
Property created successfully.

orientdb {db=Hoteldb}>
Property created successfully.

orientdb {db=Hoteldb}>
Property created successfully.

orientdb {db=Hoteldb}>
Property created successfully.

orientdb {db=Hoteldb}>
Property created successfully.

orientdb {db=Hoteldb}> CREATE INDEX Booking.BookID UNIQUE;


Creating index...
Created index successfully in 0.050000 sec(s).
```

Tạo Lớp Feedback:

```
-- Lớp lưu thông tin phản hồi
CREATE CLASS Feedback EXTENDS V, ORestricted;
CREATE PROPERTY Feedback.customer STRING (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Feedback.comment STRING;
CREATE PROPERTY Feedback.rating INTEGER;
```

```
orientdb {db=HotelDB}> CREATE CLASS Feedback EXTENDS V, ORestricted;
CREATE PROPERTY Feedback.customer STRING (MANDATORY TRUE, NOTNULL TRUE);
CREATE PROPERTY Feedback.comment STRING;
CREATE PROPERTY Feedback.rating INTEGER;

Class created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
Property created successfully.

orientdb {db=HotelDB}>
```

3.4.4 Tạo Các Role và Cấp Quyền

Cấp quyền xem (READ) cho role EmpRoleModify trên bảng Orders, Feedback,

Customers để nhân viên emp_an có thể xem được thông tin order, feedback và khách hàng.

```
GRANT READ ON database.class.Booking TO EmpRoleModify;
GRANT READ ON database.class.Feedback TO EmpRoleModify;
GRANT READ ON database.class.Customers TO EmpRoleModify;
```

```
orientdb {db=HotelDB}> GRANT READ ON database.class.Orders TO EmpRoleModify;
GRANT READ ON database.class.Feedback TO EmpRoleModify;
GRANT READ ON database.class.Customers TO EmpRoleModify;

Privilege granted to the role: [{operation:grant,role:EmpRoleModify,permission:READ,resource:database.class.Orders}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:EmpRoleModify,permission:READ,resource:database.class.Feedback}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:EmpRoleModify,permission:READ,resource:database.class.Customers}].

orientdb {db=HotelDB}>
```

Tạo role cho khách hàng

```
INSERT INTO ORole SET name = 'CustomerRole', mode = 0;
```

```
orientdb {db=HotelDB}> INSERT INTO ORole SET name = 'CustomerRole', mode = 1;

Inserted record '[ORole#5:5{name:CustomerRole,mode:1} v1]' in 0.000000 sec(s).

orientdb {db=HotelDB}> select * from Orole

+----+----+------+----+---------+---------+-----------------------------------------------+--------------------------
-------------------------------------+
|#   |@RID|@CLASS|mode|name     |inherited|rules                                          |policies
                                      |
+----+----+------+----+---------+---------+-----------------------------------------------+--------------------------
-------------------------------------+
|0   |#5:0|ORole |    |admin    |         |{database=31, database.function=31, database.s...|{*:#4:0}
                                      |
|1   |#5:1|ORole |    |reader   |         |{database.cluster.internal=2, database.cluster...|{database.class.*.*:#4:0, d
atabase:#4:1, database.schema:#4:1,...|
|2   |#5:2|ORole |    |writer   |         |{database.cluster.internal=2, database.class.o...|{database.class.*.*:#4:0, d
atabase:#4:1, database.schema:#4:3,...|
|3   |#5:3|ORole |1   |EmpRol...|         |{database=2, database.class.information=6, dat...|
                                      |
|4   |#5:4|ORole |1   |EmpRol...|         |{database=2, database.class.information=2, dat...|
                                      |
|5   |#5:5|ORole |1   |Custom...|         |                                               |
                                      |
+----+----+------+----+---------+---------+-----------------------------------------------+--------------------------
-------------------------------------+

6 item(s) found. Query executed in 0.0 sec(s).
orientdb {db=HotelDB}>
```

```
orientdb {db=HotelDB}> select * from Orole

+----+----+------+----+---------+---------+-----------------------------------------------+---------------------------
-------------------------------------+
|#   |@RID|@CLASS|mode|name     |inherited|rules                                          |policies
                                      |
+----+----+------+----+---------+---------+-----------------------------------------------+---------------------------
-------------------------------------+
|0   |#5:0|ORole |    |admin    |         |{database=31, database.function=31, database.s...|{*:#4:0}
                                      |
|1   |#5:1|ORole |    |reader   |         |{database.cluster.internal=2, database.cluster...|{database.class.*.*:#4:0, d
atabase:#4:1, database.schema:#4:1,...|
|2   |#5:2|ORole |    |writer   |         |{database.cluster.internal=2, database.class.o...|{database.class.*.*:#4:0, d
atabase:#4:1, database.schema:#4:3,...|
|3   |#5:3|ORole |1   |EmpRol...|         |{database=2, database.class.information=6, dat...|
                                      |
|4   |#5:4|ORole |1   |EmpRol...|         |{database=2, database.class.information=2, dat...|
                                      |
|5   |#5:5|ORole |1   |Custom...|         |{database=2, database.class.feedback=6, databa...|
                                      |
|6   |#5:6|ORole |1   |ViewDa...|         |{database.class.feedback=2, database.class.ord...|
                                      |
+----+----+------+----+---------+---------+-----------------------------------------------+---------------------------
-------------------------------------+

7 item(s) found. Query executed in 0.0 sec(s).
orientdb {db=HotelDB}>
```

Cấp quyền cho CustomerRole để xem và sửa thông tin cá nhân và thêm phản hồi

```
GRANT READ ON database.class.Customers TO CustomerRole;
GRANT UPDATE ON database.class.Customers TO CustomerRole;
GRANT READ ON database.class.Customers TO CustomerRole;
GRANT READ ON database.class.Feedback TO CustomerRole;
GRANT CREATE ON database.class.Feedback TO CustomerRole;
GRANT DELETE ON database.class.Feedback TO CustomerRole;
GRANT READ ON database.class.Booking TO CustomerRole;
GRANT READ ON database.cluster TO CustomerRole;
GRANT UPDATE ON database.cluster TO CustomerRole;
GRANT DELETE ON database.cluster TO CustomerRole;
GRANT CREATE ON database.cluster TO CustomerRole;
```

```
orientdb {db=HotelDB}> GRANT READ ON database.class.Customers TO CustomerRole;

Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:READ,resource:database.class.Customers}].

orientdb {db=HotelDB}> GRANT UPDATE ON database.class.Customers TO CustomerRole;

Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:UPDATE,resource:database.class.Customers}]
.
```

```
orientdb {db=HotelDB}> GRANT READ ON database.class.Feedback TO CustomerRole;

Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:READ,resource:database.class.Feedback}].

orientdb {db=HotelDB}>
```

```
orientdb {db=HotelDB}> GRANT CREATE ON database.class.Feedback TO CustomerRole;
GRANT DELETE ON database.class.Feedback TO CustomerRole;

Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:CREATE,resource:database.class.Feedback}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:DELETE,resource:database.class.Feedback}].

orientdb {db=HotelDB}>
```

```
orientdb {db=HotelDB}> GRANT UPDATE ON database.class.Feedback TO CustomerRole;

Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:UPDATE,resource:database.class.Feedback}].

orientdb {db=HotelDB}>
```

```
orientdb {db=HotelDB}> GRANT READ ON database.cluster TO CustomerRole;

Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:READ,resource:database.cluster}].

orientdb {db=HotelDB}> GRANT UPDATE ON database.cluster TO CustomerRole;

Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:UPDATE,resource:database.cluster}].

orientdb {db=HotelDB}>
```

```
orientdb {db=HotelDB}> GRANT DELETE ON database.cluster TO CustomerRole;

Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:DELETE,resource:database.cluster}].

orientdb {db=HotelDB}> GRANT CREATE ON database.cluster TO CustomerRole;

Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:CREATE,resource:database.cluster}].

orientdb {db=HotelDB}>
```

3.4.5 Tạo User và Gán Quyền Truy Cập

Cấp các quyền cho Role để user có thể đăng nhập vào trong database:

```
GRANT READ ON database TO CustomerRole;
GRANT READ ON database.schema to CustomerRole;
```

```
orientdb {db=HotelDB}> GRANT READ ON database TO CustomerRole;
GRANT READ ON database.schema to CustomerRole;
GRANT READ ON database.class to CustomerRole;

Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:READ,resource:database}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:READ,resource:database.schema}].

orientdb {db=HotelDB}>
Privilege granted to the role: [{operation:grant,role:CustomerRole,permission:READ,resource:database.class}].

orientdb {db=HotelDB}>
```

Tạo user customer_user, customer_user2 **và gán role CustomerRole.**

CREATE USER customer_user IDENTIFIED BY ‘customer_user’ ROLE CustomerRole;

CREATE USER customer_user2 IDENTIFIED BY 'customer_user2' ROLE CustomerRole;

```
orientdb {db=HotelDB}> CREATE USER customer_user IDENTIFIED BY 'customer_user' ROLE CustomerRole;

User created successfully.

orientdb {db=HotelDB}>
```

```
orientdb {db=HotelDB}> CREATE USER customer_user2 IDENTIFIED BY 'customer_user2' ROLE CustomerRole;

User created successfully.

orientdb {db=HotelDB}>
```

3.4.6 Thêm Thông Tin Khách Hàng, Booking, và Feedback

Thêm Thông Tin Khách Hàng cho customer_user

INSERT INTO Customers SET FullName = 'Tran Van Tien', Contact = '0123456789', _allowRead = (SELECT @RID FROM OUser WHERE name = 'customer_user'), _allowUpdate = (SELECT @RID FROM OUser WHERE name = 'customer_user')

```
owRead = ['customer_user'],  _allowUpdate = ['customer_user']   DB name="HotelDB"
-> java.lang.String cannot be cast to com.orientechnologies.orient.core.db.record.OIdentifiable
orientdb {db=HotelDB}> INSERT INTO Customers SET FullName = 'Tran Van Tien', Contact = '0123456789',  _allowRead = (SELE
CT @RID FROM OUser WHERE name = 'customer_user'), _allowUpdate = (SELECT @RID FROM OUser WHERE name = 'customer_user')

Inserted record '[Customers#42:0{FullName:Tran Van Tien,Contact:0123456789,_allowRead:[1],_allowUpdate:[1],_allow:[1]} v
1]' in 0.078000 sec(s).

orientdb {db=HotelDB}>
```

Thêm Thông Tin Khách Hàng cho customer_user2

INSERT INTO Customers SET FullName = 'Nguyen Thi Lan', Contact = '0987654321', _allowRead = (SELECT @RID FROM OUser WHERE name = 'customer_user2'), _allowUpdate = (SELECT @RID FROM OUser WHERE name = 'customer_user2');

```
orientdb {db=HotelDB}> INSERT INTO Customers SET FullName = 'Nguyen Thi Lan', Contact = '0987654321', _allowRead = (SELECT @RID FROM
OUser WHERE name = 'customer_user2'), _allowUpdate = (SELECT @RID FROM OUser WHERE name = 'customer_user2');

Inserted record '[Customers#42:1{FullName:Nguyen Thi Lan,Contact:0987654321,_allowRead:[1],_allowUpdate:[1],_allow:[1]} v1]' in 0.008
000 sec(s).

orientdb {db=HotelDB}>
```

Thêm Booking cho customer_user

INSERT INTO Booking SET BookDate = sysdate(), BookID = 'B001', CheckInDate = '2024-11-10 14:00:00', CheckOutDate = '2024-11-15 12:00:00', CustomerID = (SELECT FROM Customers WHERE FullName = 'Tran Van Tien'), RoomID = (SELECT FROM Room WHERE RoomID = 'R102'), Status = 'Confirmed', TotalCost = 120.00, _allowRead = (SELECT @RID FROM OUser WHERE name = 'customer_user')

```
orientdb {db=Hoteldb}> INSERT INTO Booking SET BookDate = sysdate(), BookID = 'B001', CheckInDate = '2024-11-10 14:00:00
', CheckOutDate = '2024-11-15 12:00:00', CustomerID = (SELECT FROM Customers WHERE FullName = 'Tran Van Tien'), RoomID =
 (SELECT FROM Room WHERE RoomID = 'R102'), Status = 'Confirmed', TotalCost = 120.00, _allowRead = (SELECT @RID FROM OUse
r WHERE name = 'customer_user')

Inserted record '[Booking#58:0{BookDate:Thu Nov 07 01:24:29 ICT 2024,BookID:B001,CheckInDate:Sun Nov 10 14:00:00 ICT 202
4,CheckOutDate:Fri Nov 15 12:00:00 ICT 2024,CustomerID:[1],RoomID:#67:0,Status:Confirmed,TotalCost:120.0,_allowRead:[1],
_allow:[1]} v1]' in 0.022000 sec(s).

orientdb {db=Hoteldb}>
```

Thêm Booking cho customer_user2

INSERT INTO Booking SET BookDate = sysdate(), BookID = 'B002', CheckInDate = '2024-11-15 14:00:00', CheckOutDate = '2024-11-20 12:00:00', CustomerID = (SELECT FROM Customers WHERE FullName = 'Nguyen Thi Lan'), RoomID = (SELECT FROM Room WHERE RoomID = 'R103'), Status = 'Confirmed', TotalCost = 200.00, _allowRead = (SELECT @RID FROM OUser WHERE name = 'customer_user2')

```
orientdb {db=Hoteldb}> INSERT INTO Booking SET BookDate = sysdate(), BookID = 'B002', CheckInDate = '2024-11-15 14:00:00
', CheckOutDate = '2024-11-20 12:00:00', CustomerID = (SELECT FROM Customers WHERE FullName = 'Nguyen Thi Lan'), RoomID
= (SELECT FROM Room WHERE RoomID = 'R103'), Status = 'Confirmed', TotalCost = 200.00, _allowRead = (SELECT @RID FROM OUs
er WHERE name = 'customer_user2')

Inserted record '[Booking#59:0{BookDate:Thu Nov 07 01:28:16 ICT 2024,BookID:B002,CheckInDate:Fri Nov 15 14:00:00 ICT 202
4,CheckOutDate:Wed Nov 20 12:00:00 ICT 2024,CustomerID:[1],RoomID:#68:0,Status:Confirmed,TotalCost:200.0,_allowRead:[1],
_allow:[1]} v1]' in 0.005000 sec(s).

orientdb {db=Hoteldb}>
```

Thêm Phản Hồi từ customer_user

INSERT INTO Feedback SET customer = ' Tran Van Tien', comment = 'Dich vu

tuyet voi!', rating = 5, _allowRead = (SELECT @RID FROM OUser WHERE name = 'customer_user'), _allowUpdate = (SELECT @RID FROM OUser WHERE name = 'customer_user'), _allowDelete = (SELECT @RID FROM OUser WHERE name = 'customer_user')

```
orientdb {db=HotelDB}> INSERT INTO Feedback SET customer = ' Tran Van Tien', comment = 'Dich vu tuyet voi!', rating = 5,
 _allowRead = (SELECT @RID FROM OUser WHERE name = 'customer_user'), _allowUpdate = (SELECT @RID FROM OUser WHERE name =
 'customer_user'), _allowDelete = (SELECT @RID FROM OUser WHERE name = 'customer_user')

Inserted record '[Feedback#58:0{customer: Tran Van Tien,comment:Dich vu tuyet voi!,rating:5,_allowRead:[1],_allowUpdate:
[1],_allowDelete:[1],_allow:[1]} v1]' in 0.016000 sec(s).

orientdb {db=HotelDB}>
```

Thêm Phản Hồi từ customer_user2

INSERT INTO Feedback SET customer = 'Nguyen Thi Lan', comment = 'Phong dep va sach se!', rating = 4, _allowRead = (SELECT @RID FROM OUser WHERE name = 'customer_user2'), _allowUpdate = (SELECT @RID FROM OUser WHERE name = 'customer_user2'), _allowDelete = (SELECT @RID FROM OUser WHERE name = 'customer_user2');

```
orientdb {db=HotelDB}> INSERT INTO Feedback SET customer = 'Nguyen Thi Lan', comment = 'Phong dep va sach se!', rating = 4, _allowRea
d = (SELECT @RID FROM OUser WHERE name = 'customer_user2'), _allowUpdate = (SELECT @RID FROM OUser WHERE name = 'customer_user2'), _a
llowDelete = (SELECT @RID FROM OUser WHERE name = 'customer_user2');

Inserted record '[Feedback#58:1{customer:Nguyen Thi Lan,comment:Phong dep va sach se!,rating:4,_allowRead:[1],_allowUpdate:[1],_allow
Delete:[1],_allow:[1]} v1]' in 0.014000 sec(s).

orientdb {db=HotelDB}>
```

3.4.7 Cấp Quyền Truy Cập Bổ Sung cho Nhân Viên emp_an

Lấy @RID của nhân viên emp_an

```
orientdb {db=HotelDB}> select @RID from OUser WHERE name = 'emp_an';

+----+----+------+
|#   |@RID|@CLASS|
+----+----+------+
|0   |#6:2|OUser |
+----+----+------+

1 item(s) found. Query executed in 0.003 sec(s).
orientdb {db=HotelDB}>
```

Cấp quyền xem thông tin của khách hàng cho emp_an

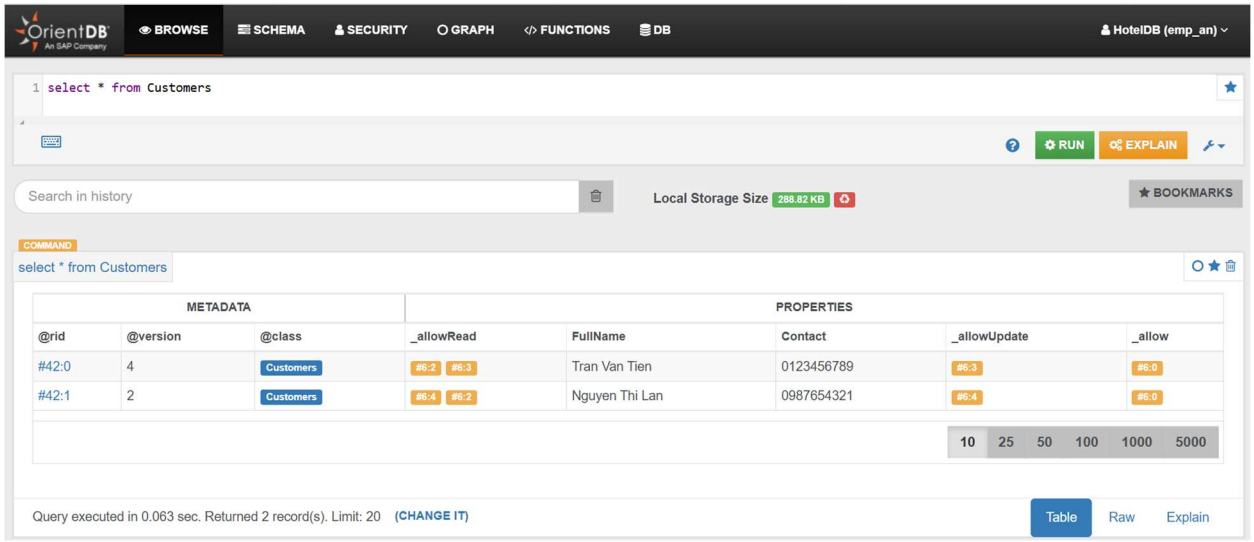
UPDATE Customers SET _allowRead = _allowRead || [#6:2];

UPDATE Customers SET _allowRead = _allowRead || [#6:2] WHERE NOT

(_allowRead CONTAINS #6:2);

```
orientdb {db=HotelDB}> UPDATE Customers SET _allowRead = _allowRead || [#6:2];

Updated record(s) '[{count:1}]' in 0.015000 sec(s).
```

```
orientdb {db=HotelDB}> UPDATE Customers SET _allowRead = _allowRead || [#6:2] WHERE NOT (_allowRead CONTAINS #6:2);

Updated record(s) '[{count:1}]' in 0.009000 sec(s).

orientdb {db=HotelDB}> select * from Customers

+----+-----+---------+-----------+-------------+----------+------------+------+
|#   |@RID |@CLASS   |_allowRead |FullName     |Contact   |_allowUpdate|_allow|
+----+-----+---------+-----------+-------------+----------+------------+------+
|0   |#42:0|Customers|[#6:2,#6:3]|Tran Van Tien |0123456789|[#6:3]      |[#6:0]|
|1   |#42:1|Customers|[#6:4,#6:2]|Nguyen Thi Lan|0987654321|[#6:4]      |[#6:0]|
+----+-----+---------+-----------+-------------+----------+------------+------+

2 item(s) found. Query executed in 0.0 sec(s).
orientdb {db=HotelDB}>
```

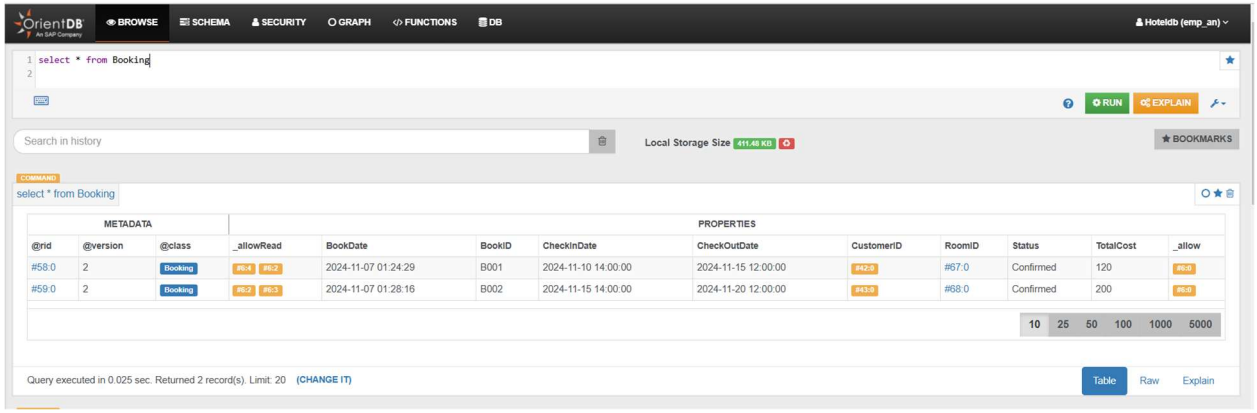


Cấp quyền xem thông tin booking của khách hàng cho emp_an

UPDATE Booking SET _allowRead = _allowRead || [#6:2];

UPDATE Booking SET _allowRead = _allowRead || [#6:2] WHERE NOT (_allowRead CONTAINS #6:2);

```
orientdb {db=HotelDB}> UPDATE Orders SET _allowRead = _allowRead || [#6:2] WHERE NOT (_allowRead CONTAINS #6:2);

Updated record(s) '[{count:1}]' in 0.011000 sec(s).

orientdb {db=HotelDB}> select * from Orders

+----+-----+------+-----------+-------------+--------+--------+------------+------+
|#   |@RID |@CLASS|_allowRead |customer     |item    |quantity|status      |_allow|
+----+-----+------+-----------+-------------+--------+--------+------------+------+
|0   |#50:0|Orders|[#6:2,#6:3]|Tran Van Tien |Bua sang|2       |Da phuc vu  |[#6:0]|
|1   |#50:1|Orders|[#6:4,#6:2]|Nguyen Thi Lan|Ca phe  |1       |Cho xac nhan|[#6:0]|
+----+-----+------+-----------+-------------+--------+--------+------------+------+

2 item(s) found. Query executed in 0.0 sec(s).
orientdb {db=HotelDB}>
```



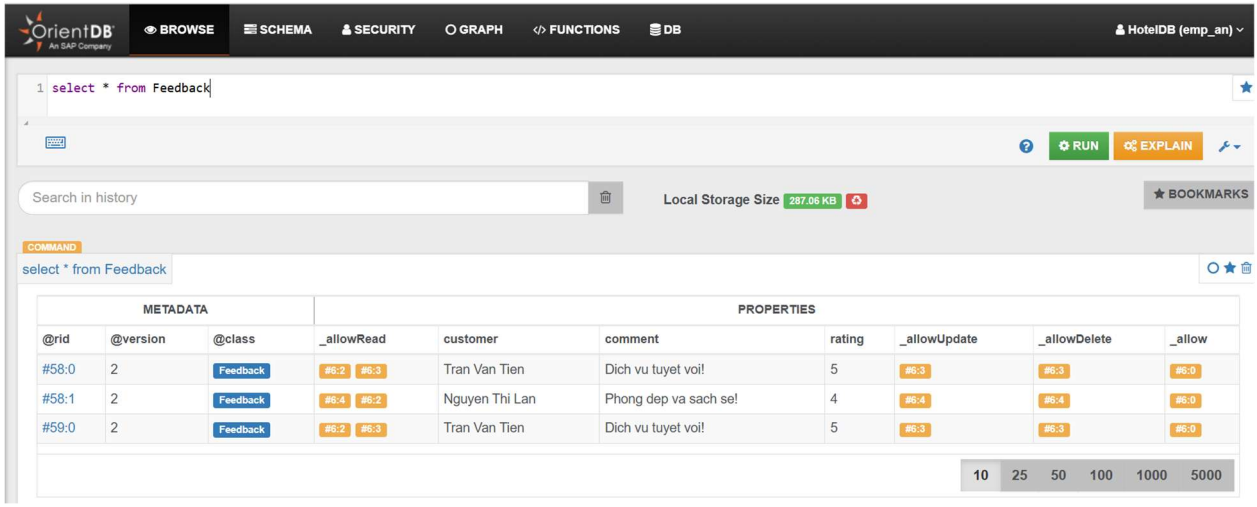
Cấp quyền xem thông tin feedback của khách hàng cho emp_an

UPDATE Feedback SET _allowRead = _allowRead || [#6:2];

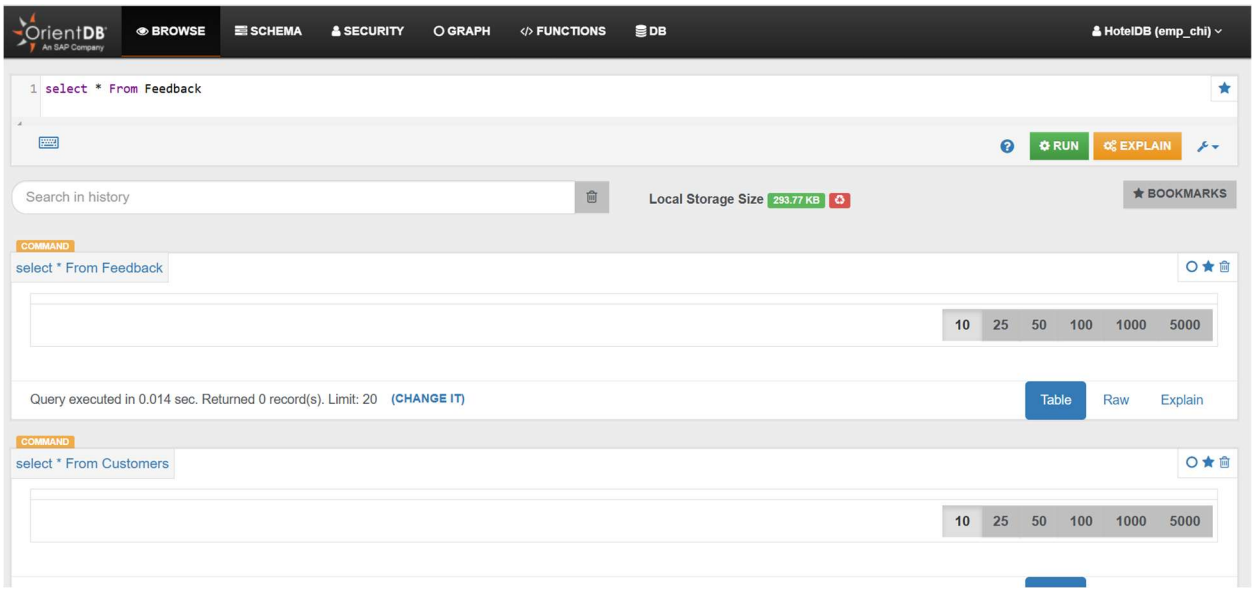
UPDATE Feedback SET _allowRead = _allowRead || [#6:2] WHERE NOT (_allowRead CONTAINS #6:2);

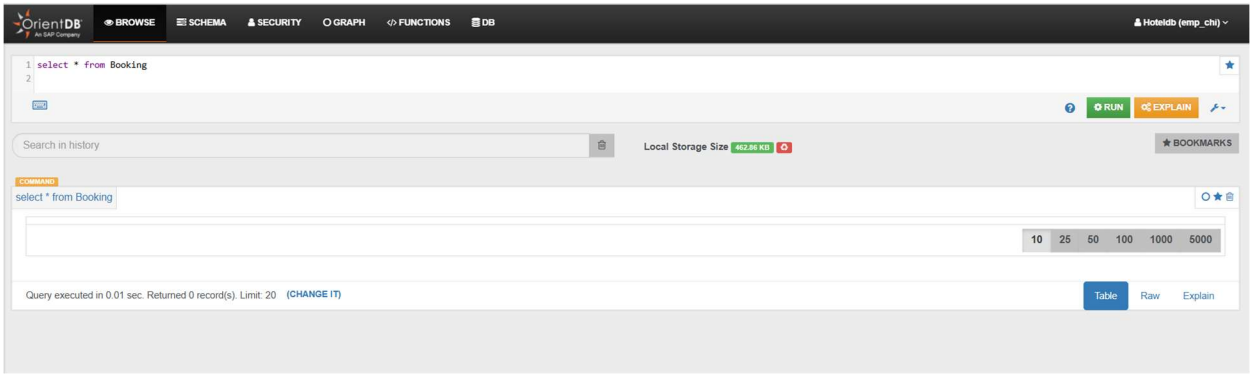
```
orientdb {db=HotelDB}> UPDATE Feedback SET _allowRead = _allowRead || [#6:2];

Updated record(s) '[{count:2}]' in 0.016000 sec(s).

orientdb {db=HotelDB}>
```

```
orientdb {db=HotelDB}> UPDATE Feedback SET _allowRead = _allowRead || [#6:2] WHERE NOT (_allowRead CONTAINS #6:2);

Updated record(s) '[{count:1}]' in 0.008000 sec(s).

orientdb {db=HotelDB}> select * from Feedback

+----+-----+--------+-----------+--------------+---------------------+------+------------+------------+------+
|#   |@RID |@CLASS  |_allowRead |customer      |comment              |rating|_allowUpdate|_allowDelete|_allow|
+----+-----+--------+-----------+--------------+---------------------+------+------------+------------+------+
|0   |#58:0|Feedback|[#6:2,#6:3]| Tran Van Tien|Dich vu tuyet voi!   |5     |[#6:3]      |[#6:3]      |[#6:0]|
|1   |#58:1|Feedback|[#6:4,#6:2]|Nguyen Thi Lan|Phong dep va sach se!|4     |[#6:4]      |[#6:4]      |[#6:0]|
|2   |#59:0|Feedback|[#6:2,#6:3]| Tran Van Tien|Dich vu tuyet voi!   |5     |[#6:3]      |[#6:3]      |[#6:0]|
+----+-----+--------+-----------+--------------+---------------------+------+------------+------------+------+

3 item(s) found. Query executed in 0.0 sec(s).
orientdb {db=HotelDB}> |
```



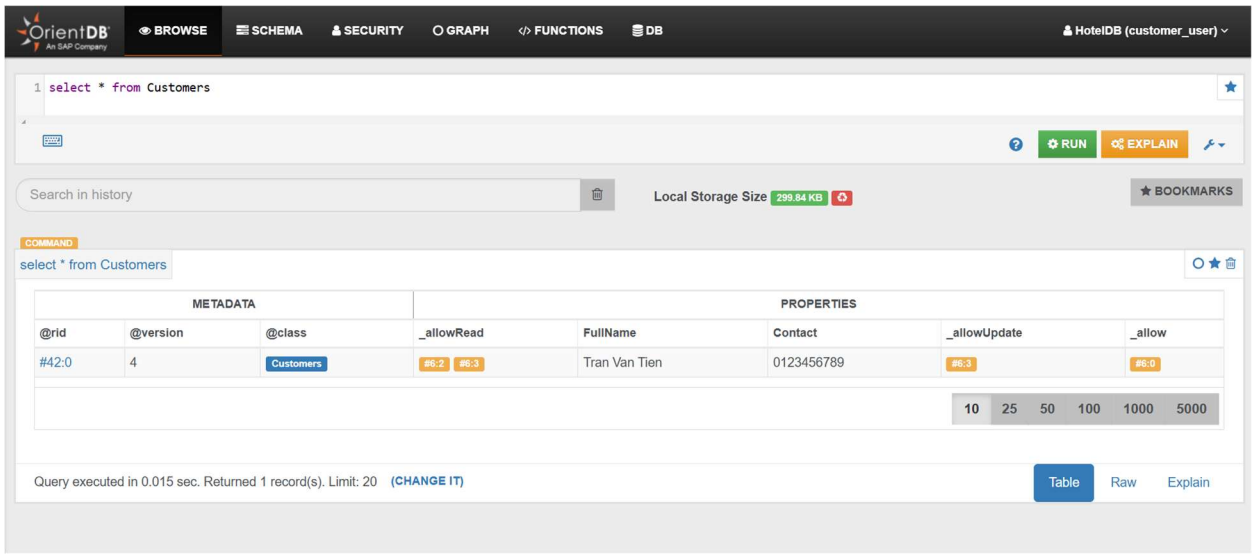
Nhân viên emp_chi không thể xem thông tin khách hàng, booking và feedback

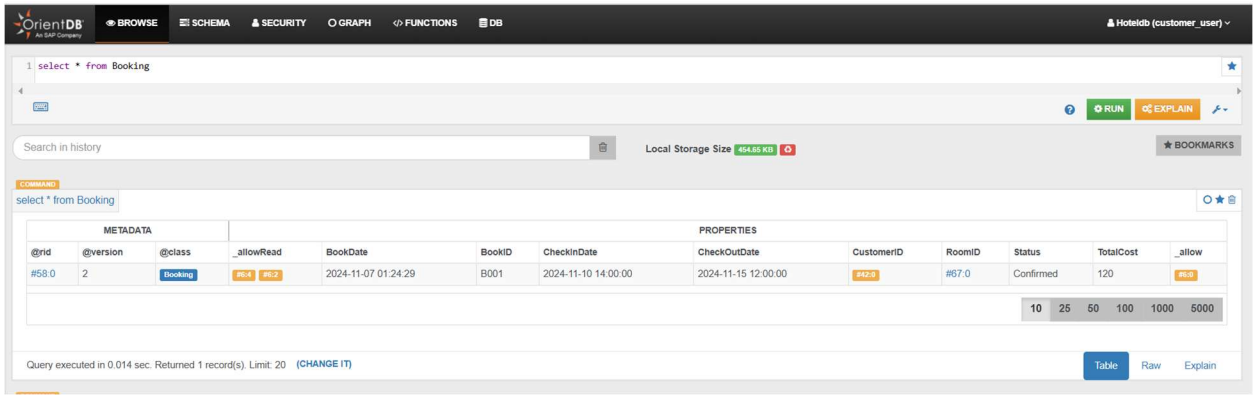


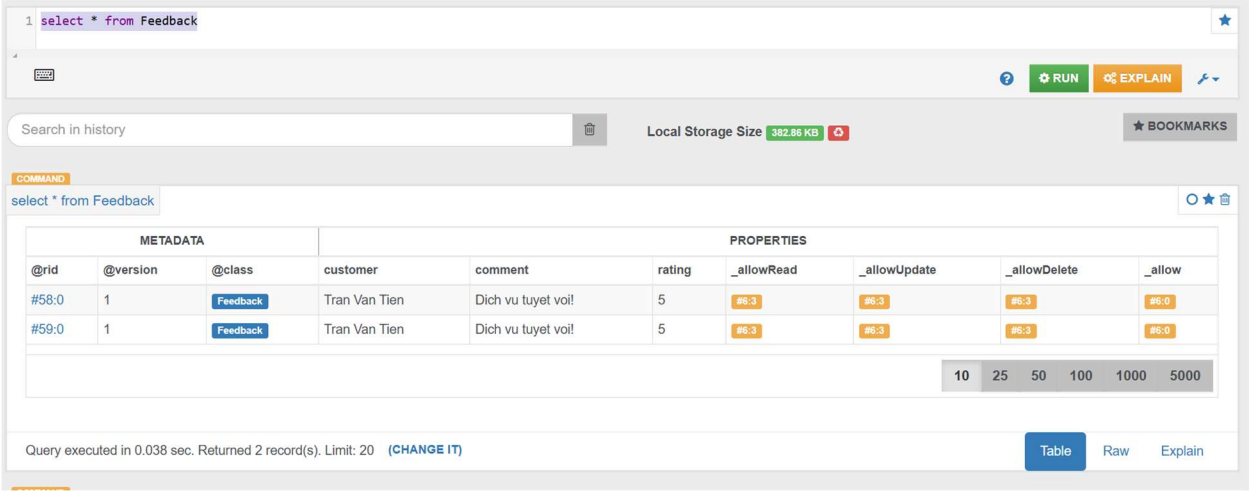


3.4.8 Kiểm Tra Các Quyền Của customer_user

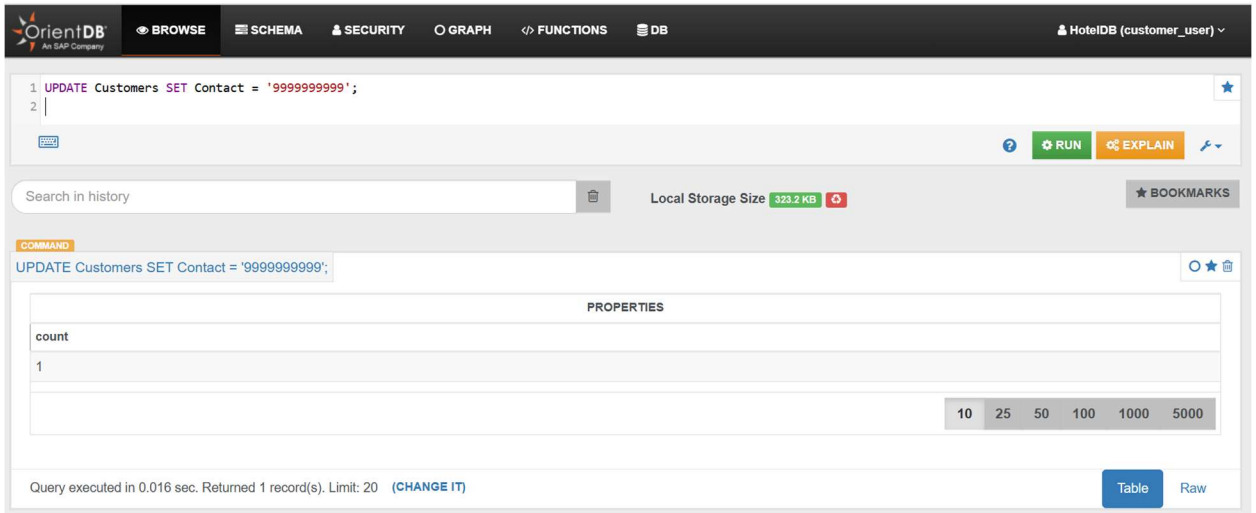
Customer_user Tien chỉ xem được thông tin của chính mình

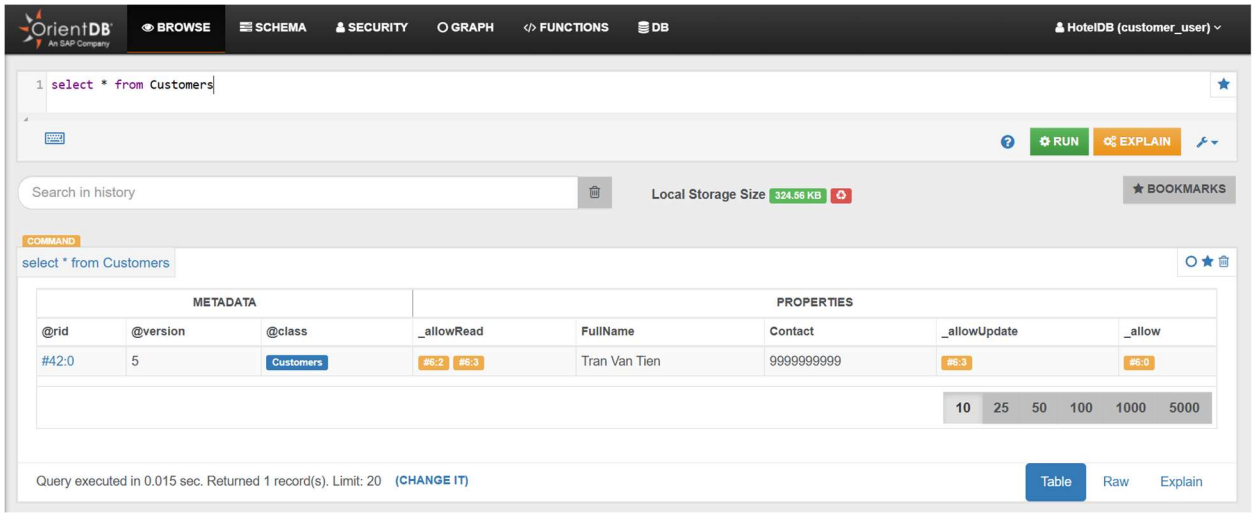






Chỉnh sửa thông tin của chính mình





Tạo, xóa, chỉnh sửa feedback của chính mình

INSERT INTO Feedback SET customer = 'Tran Van Tien', comment = 'Rất hài lòng với dịch vụ!', rating = 4, _allowRead = (SELECT @RID FROM OUser WHERE name = 'customer_user') || [#6:2], _allowUpdate = (SELECT @RID FROM OUser WHERE name = 'customer_user'), _allowDelete = (SELECT @RID FROM OUser WHERE name = 'customer_user');

OrientDB An SAP Company | BROWSE | SCHEMA | SECURITY | GRAPH | FUNCTIONS | DB | HotelDB (customer_user)

```
INSERT INTO Feedback SET customer = 'Tran Van Tien', comment = 'Rất hài lòng với dịch vụ!', rating = 4, _allowRead = (SELECT @RID FROM OUser WHERE name =
'customer_user') || [#6:2], _allowUpdate = (SELECT @RID FROM OUser WHERE name = 'customer_user'), _allowDelete = (SELECT @RID FROM OUser WHERE name = 'customer_user');
```

RUN | EXPLAIN

Search in history | Local Storage Size 345.37 KB | BOOKMARKS

COMMAND

INSERT INTO Feedback SET customer = 'Tran Van Tien', comment = 'Rất hài lòng với dịch vụ!', rating = 4, _allowRead = (SELECT @RID FROM OUser WHERE name = 'cust...

METADATA			PROPERTIES						
@rid	@version	@class	customer	comment	rating	_allowRead	_allowUpdate	_allowDelete	_allow
#62:0	1	Feedback	Tran Van Tien	Rất hài lòng với dịch vụ!	4	#6:2 #6:3	#6:3	#6:3	#6:3

10 25 50 100 1000 5000

Query executed in 0.028 sec. Returned 1 record(s). Limit: 20 (CHANGE IT) | Table | Raw

DELETE VERTEX Feedback WHERE rating = 4;

OrientDB An SAP Company | BROWSE | SCHEMA | SECURITY | GRAPH | FUNCTIONS | DB | HotelDB (customer_user)

```
DELETE VERTEX Feedback WHERE rating = 4;
```

RUN | EXPLAIN

Search in history | Local Storage Size 345.29 KB | BOOKMARKS

COMMAND

DELETE VERTEX Feedback WHERE rating = 4;

PROPERTIES
count
1

10 25 50 100 1000 5000

Query executed in 0.141 sec. Returned 1 record(s). Limit: 20 (CHANGE IT) | Table | Raw

OrientDB An SAP Company | BROWSE | SCHEMA | SECURITY | GRAPH | FUNCTIONS | DB | HotelDB (customer_user)

```
UPDATE Feedback SET comment = 'Dich vu tot hon mong doi!' WHERE rating = 5;
```

RUN | EXPLAIN

Search in history | Local Storage Size 334.91 KB | BOOKMARKS

COMMAND

UPDATE Feedback SET comment = 'Dich vu tot hon mong doi!' WHERE rating = 5;

PROPERTIES
count
2

10 25 50 100 1000 5000

Query executed in 0.02 sec. Returned 1 record(s). Limit: 20 (CHANGE IT) | Table | Raw

OrientDB An SAP Company | BROWSE | SCHEMA | SECURITY | GRAPH | FUNCTIONS | DB | HotelDB (customer_user)

```
1 select * from Feedback
```

RUN | EXPLAIN

Search in history | Local Storage Size 336.36 KB | BOOKMARKS

COMMAND

select * from Feedback

METADATA			PROPERTIES						
@rid	@version	@class	rating	_allowRead	customer	comment	_allowUpdate	_allowDelete	_allow
#58:0	3	Feedback	5	#6:2 #6:3	Tran Van Tien	Dich vu tot hon mong doi!	#6:3	#6:3	#6:0
#59:0	3	Feedback	5	#6:2 #6:3	Tran Van Tien	Dich vu tot hon mong doi!	#6:3	#6:3	#6:0

10 | 25 | 50 | 100 | 1000 | 5000

Query executed in 0.017 sec. Returned 2 record(s). Limit: 20 (CHANGE IT)

Table | Raw | Explain

PHẦN KẾT LUẬN

1. Ưu điểm

Nắm được các kiến thức cũng như những vấn đề liên quan về OrientDB. OrientDB cho phép linh hoạt trong việc thiết kế và thay đổi cấu trúc dữ liệu, phù hợp với các ứng dụng có yêu cầu thay đổi liên tục. Mô hình đồ thị của OrientDB giúp biểu diễn các mối quan hệ phức tạp giữa các thực thể một cách trực quan và hiệu quả. Với khả năng lập chỉ mục mạnh mẽ và tối ưu hóa truy vấn, OrientDB cung cấp hiệu suất truy vấn nhanh chóng

2. Nhược điểm

Do thời gian tìm hiểu giới hạn, nhóm chưa tìm hiểu sâu vào các khía cạnh khác của OrientDB. Các cơ chế nâng cao của OrientDB

3. Hướng phát triển

Từ những nghiên cứu ban đầu về OrientDB. Nhóm sẽ khám phá các khả năng tích hợp OrientDB với các công nghệ như Spark, Kafka, Elasticsearch để xây dựng các hệ thống xử lý dữ liệu lớn, thực thời và phân tích dữ liệu phức tạp

TÀI LIỆU THAM KHẢO

1. OrientDB, Security, Server security - OrientDB
2. Cao Lê Viết Tiến, *Cơ sở dữ liệu NoSQL là gì? So sánh cơ sở dữ liệu SQL với NoSQL,* NoSQL là gì? Các thông tin về cơ sở dữ liệu NoSQL
3. ChatGPT