

Integrantes: Edgar Vaccani de Araujo

Irving Barreira Servulo da Silva

Miguel Angelo Vieira Monteiro de Oliveira Dias T Thiago da Silva Marques

**Contents**

[1. Introdução 3](#_Toc455141645)

[2. Linguagem 4](#_Toc455141646)

[2.1. Tables: 4](#_Toc455141647)

[2.2. Joins: 4](#_Toc455141648)

[2.3. Projeções: 5](#_Toc455141649)

[2.4. Distinct: 5](#_Toc455141650)

[3. Versões e Pré-requisitos 6](#_Toc455141651)

[4. OrientDB Studio 7](#_Toc455141652)

[4.1. Criando um novo banco de dados no DB 7](#_Toc455141653)

[4.2. Execução de consultas: 8](#_Toc455141654)

[4.3. Trabalhando com Json: 9](#_Toc455141655)

[4.4. Schema Mananger: 9](#_Toc455141656)

[4.5. Editor grafo: 11](#_Toc455141657)

[4.6. Consultas 12](#_Toc455141658)

[5. Clientes 13](#_Toc455141659)

[6. Características Operacionais 15](#_Toc455141660)

[6.1. Sharding 15](#_Toc455141661)

[6.2. Replicação 16](#_Toc455141662)

[6.3. Índices 18](#_Toc455141663)

[6.4. Cache 19](#_Toc455141664)

[6.5. Fail-over 19](#_Toc455141665)

[6.6. Auto-recovery. 20](#_Toc455141666)

[6.7. Tipo de transações: 20](#_Toc455141667)

[7. Referencias 21](#_Toc455141668)

# Introdução

O banco OrientDB é um Multi-model DBMS (Document, Graph e key/value). Fabricado pela Orient Technologies LTD, desenvolvido em Java, o projeto foi lançado em 2011. Atualmente possui downloads superior a 70.000 por mês, mais de 100 colaboradores da comunidade e 1000, de usuários de produção. OrientDB combina a conexão de grafos, a agilidade de documentos com o familiar dialeto SQL de modo que você pode usar OrientDB como um substituto para um produto em qualquer uma dessas categorias. No entanto, a principal razão pela qual os usuários escolhem OrientDB é a sua capacidade de agir como um verdadeiro Multi-Modelo DBMS, combinando todos os recursos dos 3 modelos em um.

O Orientdb é incrivelmente rápido, amazenando 220 mil registros por segundo em hardware comum. Suporta schema-less, schema-full e schema-mixed. Alguns de seus clientes incluem: Ericsson, as Nações Unidas, Cisco, CenturyLink e Sonatype.

OrientDB vem em duas edições. A Community Edition gratuito para qualquer uso e a Enterprise Edition que é um software comercial desenvolvido pela mesma equipe que desenvolveu o motor OrientDB e tem o serviço profissional de suporte.

No universo Bigdata, o banco OrientDB se enquadra no conceito de documentos e Grafos, por possuir ao mesmo tempo, um schema flexível encontrados nos bancos de documentos, alinhado a uma boa performance e facilidade nos relacionamentos entre diferentes entidades encontrados nos bancos de Grafos. Atualmente ele é bem aceito no mercado de bancos de dados, ocupando a 41º posição no ranking geral e apenas o 3ª entres os bancos multi-model. Fonte: **db-engines.**

# Linguagem

Orient DB usa SQL como linguagem de consulta e adiciona algumas extensões para ativar a funcionalidade gráfico. Existem algumas diferenças entre a sintaxe SQL padrão e o a usada por OrientDB, mas a maior parte das sintaxes são parecidas. Segue abaixo algumas diferenças encontradas no OrientBD:

## Tables:

OrientDB é todo orientado a objeto, por esse motivo, as tabelas são tratadas como Classes ao invés de Tabelas como nos bancos relacionais. Ex:

CREATE CLASS POST

CREATE CLASS COMMENT

## Joins:

Pelo fato do OrientDB ser um banco de Grafos, os relacionamentos são representados por links ao em vez de JOIN dos bancos relacionais.

Por esta razão, a sintaxe clássica JOIN não é suportada. OrientDB usa a notação de "ponto (.) " para navegar nos LINKS. Exemplo 1:

SELECT \*

FROM empregado A, City B

ONDE A.city = B.id

E B.name = 'Roma'

Em OrientDB uma operação equivalente seria:

SELECT \* FROM EMPREGADOS WHERE city.name = 'Roma'

Isso é muito mais simples e poderoso! Se você usar várias associações, o equivalente OrientDB SQL será um benefício ainda maior. Exemplo 2:

SELECT \*

FROM empregado A, B Cidade, País C,

ONDE A.city = B.id

E B.country = C.id

E C.name = 'Itália'

Em OrientDB uma operação equivalente seria:

SELECT \* FROM empregado quando city.country.name = 'Itália'

## Projeções:

As projeções no SQL são de preenchimento obrigatório e pode usar o caractere asterisco \* para incluir todos os campos. Com OrientDB este tipo de projecção é opcional. Exemplo: No SQL, para selecionar todas as colunas de cliente é feito da seguinte forma:

SELECT \* FROM cliente

Em OrientDB o \* é opcional:

SELECT FROM cliente

## Distinct:

Em SQL, DISTINCT é uma palavra-chave, mas em OrientDB é uma função ex:

SELECT DISTINCT nome FROM Cidade

Em OrientDB é: SELECT DISTINCT (nome) FROM Cidade

# Versões e Pré-requisitos

OrientDB está disponível em duas edições:

* Community Edition: Esta edição é lançado como um projeto de código aberto que permite o uso livre irrestrito tanto para open source e projetos comerciais.
* Enterprise Edition: É um software comercial construído em cima do Community Edition. Enterprise foi desenvolvido pela mesma equipe que desenvolveu o motor OrientDB. Ele serve como uma extensão do Community Edition, fornecendo recursos da empresa, tais como:

Query Profiler

Distributed Clustering configuration

Metrics Recording

Live Monitoring with configurable Alerts

* Pré-requisitos: Ambas as edições de OrientDB podem ser executados em qualquer sistema operacional que implementa a máquina virtual Java (JVM). Exemplos destes incluem:

Linux, todas as distribuições, incluindo ARM (Raspberry Pi, etc.)

Mac OS X

Microsoft Windows, a partir de 95 / NT e mais tarde

Solaris

HP-UX

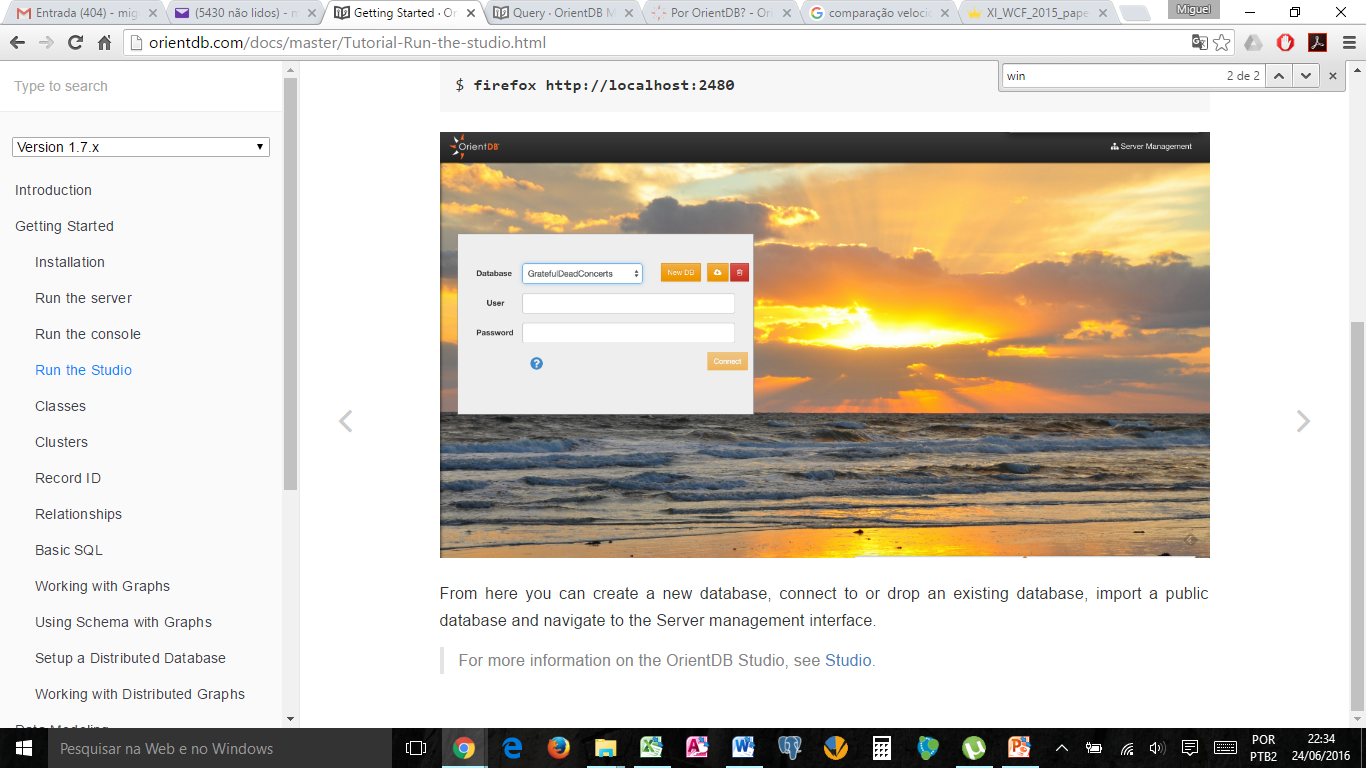
IBM AIX

O site da fabricante informa apenas que OrientDB requer Java, versão 1.7 ou superior e nenhum requisito mínimo de hardware é pré-estabelecido. No entanto, vários desenvolvedores recomendam uma configuração mínima de 512 de memória RAM e processador duo core.

# OrientDB Studio

Studio é uma interface web para a administração de OrientDB que vem em pacote com a distribuição OrientDB.Se você executar OrientDB na sua máquina a interface web pode ser acessado através da URL:

http: // localhost: 2480

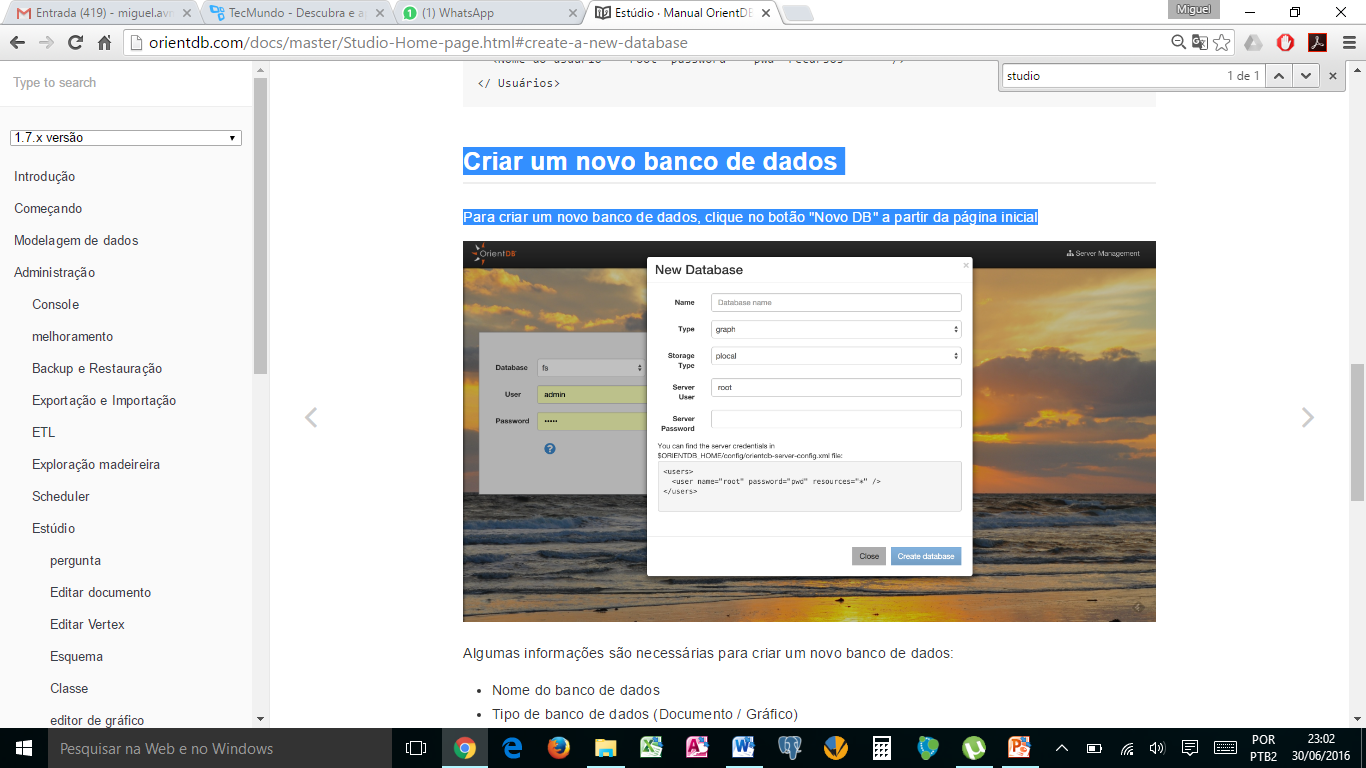


A partir daqui, você pode:

* Conectar a um banco de dados existente; Soltar um banco de dados existente; Criar um novo banco de dados; Importar um banco de dados público; Vá para a [UI Management Server](http://orientdb.com/docs/master/Studio-Server-Management.html).

## Criando um novo banco de dados no DB

Para criar um novo banco de dados, clique no botão "Novo DB" a partir da página inicial.

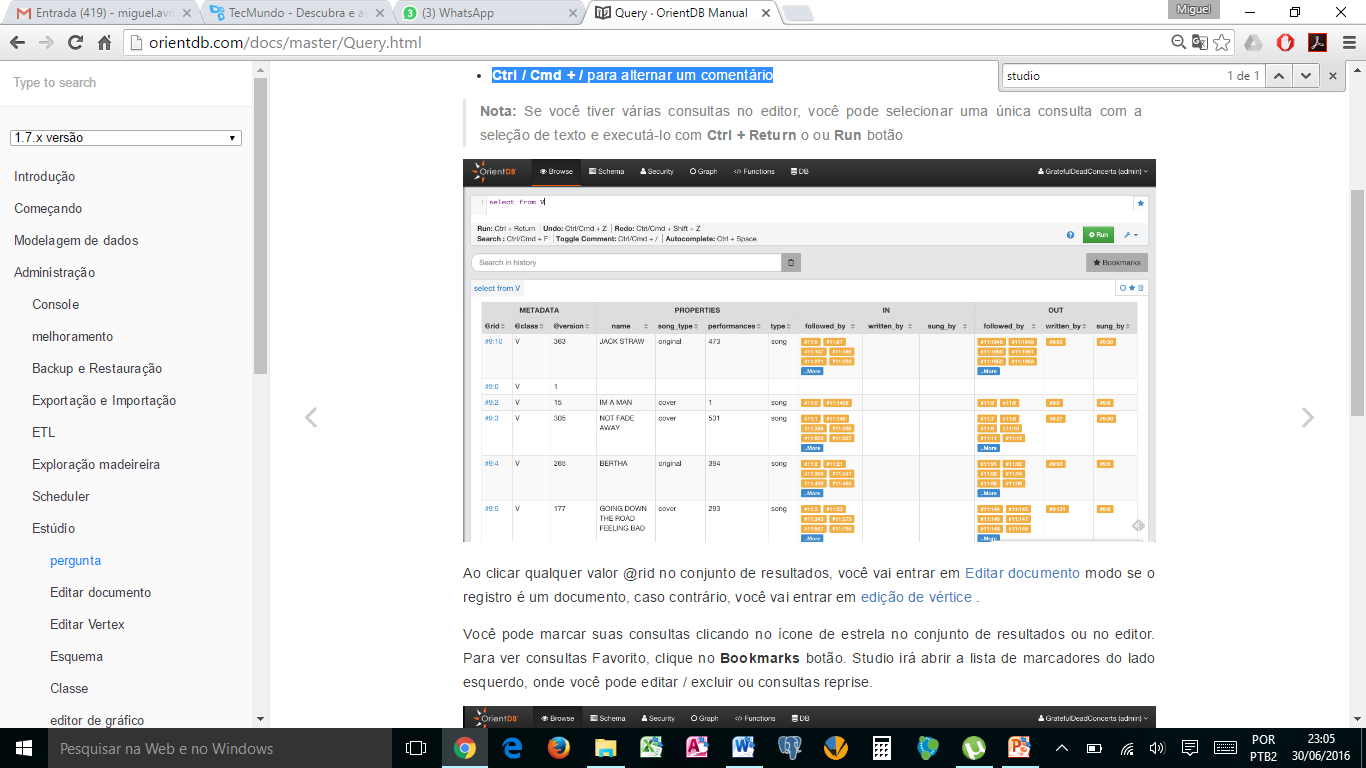


## Execução de consultas:

Studio suporta o reconhecimento automático do idioma que você está usando entre as apoiadas: [SQL](http://orientdb.com/docs/master/SQL.html) e[Gremlin](http://orientdb.com/docs/master/Gremlin.html) . Ao escrever, usar o recurso de auto-completar pressionando Ctrl + Space.

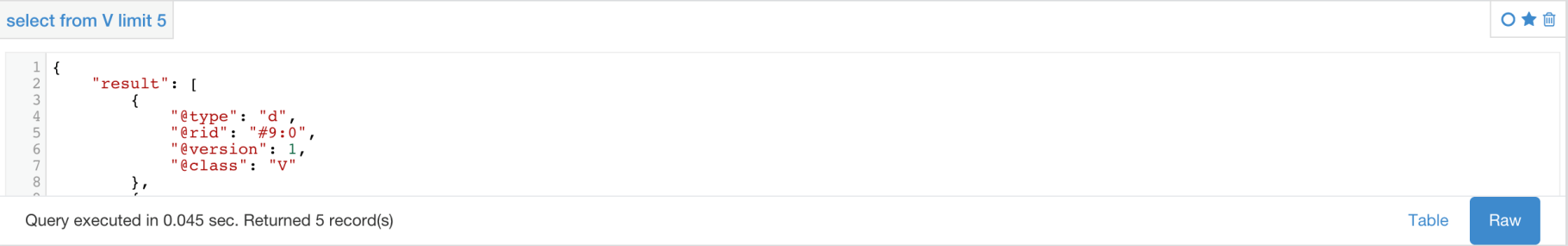
* Outros atalhos estão disponíveis no editor de consulta:

**Ctrl + Return** para executar a consulta ou apenas clique no **Run** botão; **Ctrl / Cmd + Z** para desfazer alterações; **Ctrl / Cmd + Shift** + Z para refazer alterações; **Ctrl / Cmd + F** para procurar no editor; **Ctrl / Cmd + /** para alternar um comentário.

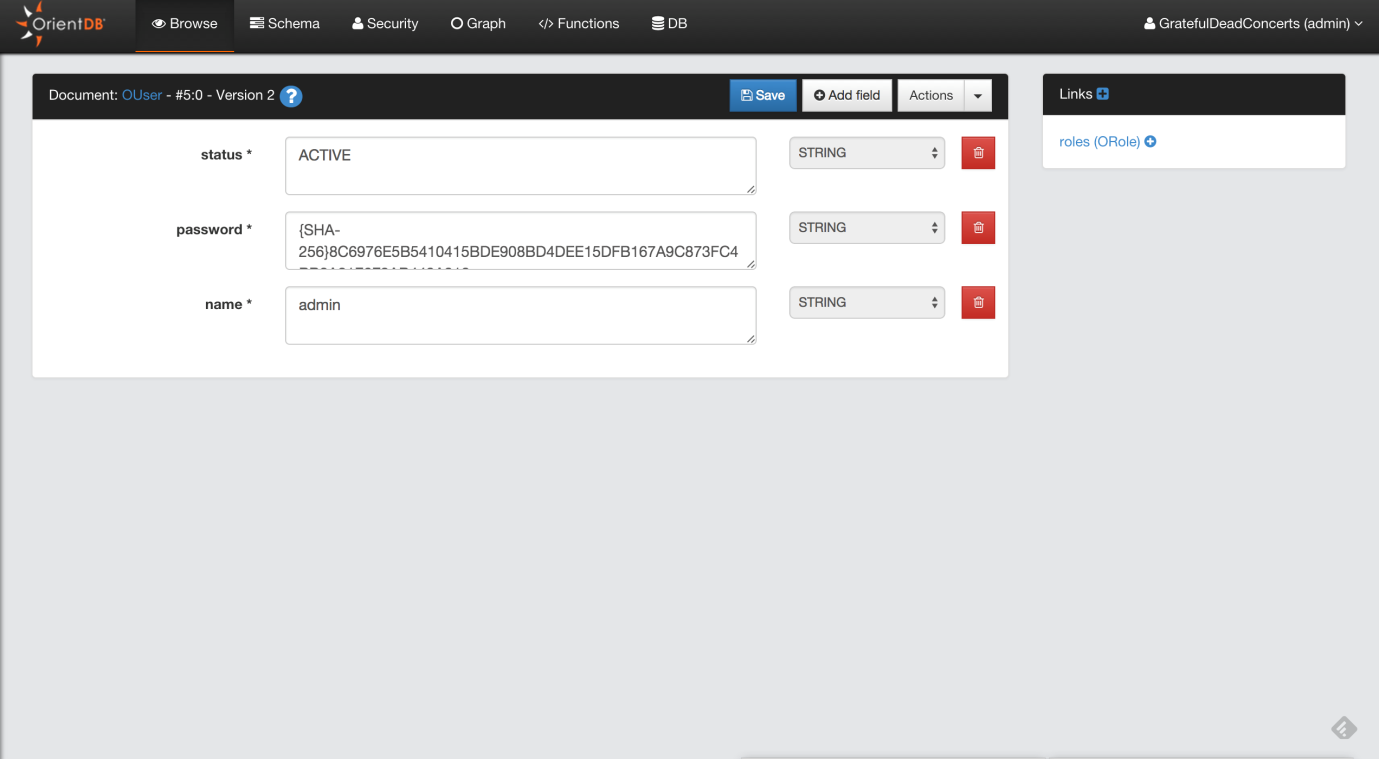


## Trabalhando com Json:

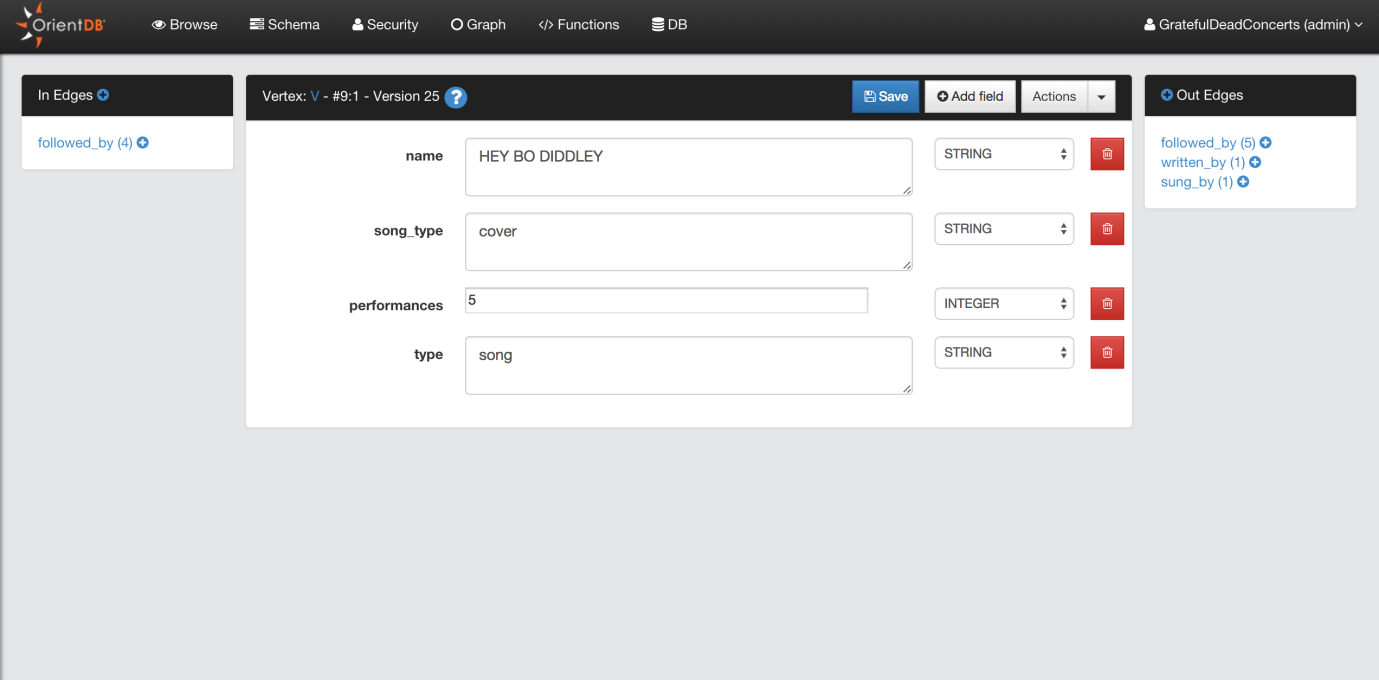
Estúdio comunica com o servidor OrientDB usando [o protocolo HTTP / REST + JSON](http://orientdb.com/docs/master/OrientDB-REST.html) . Para ver a saída em formato JSON, pressione o **RAW** guia.



Edição de Documento:

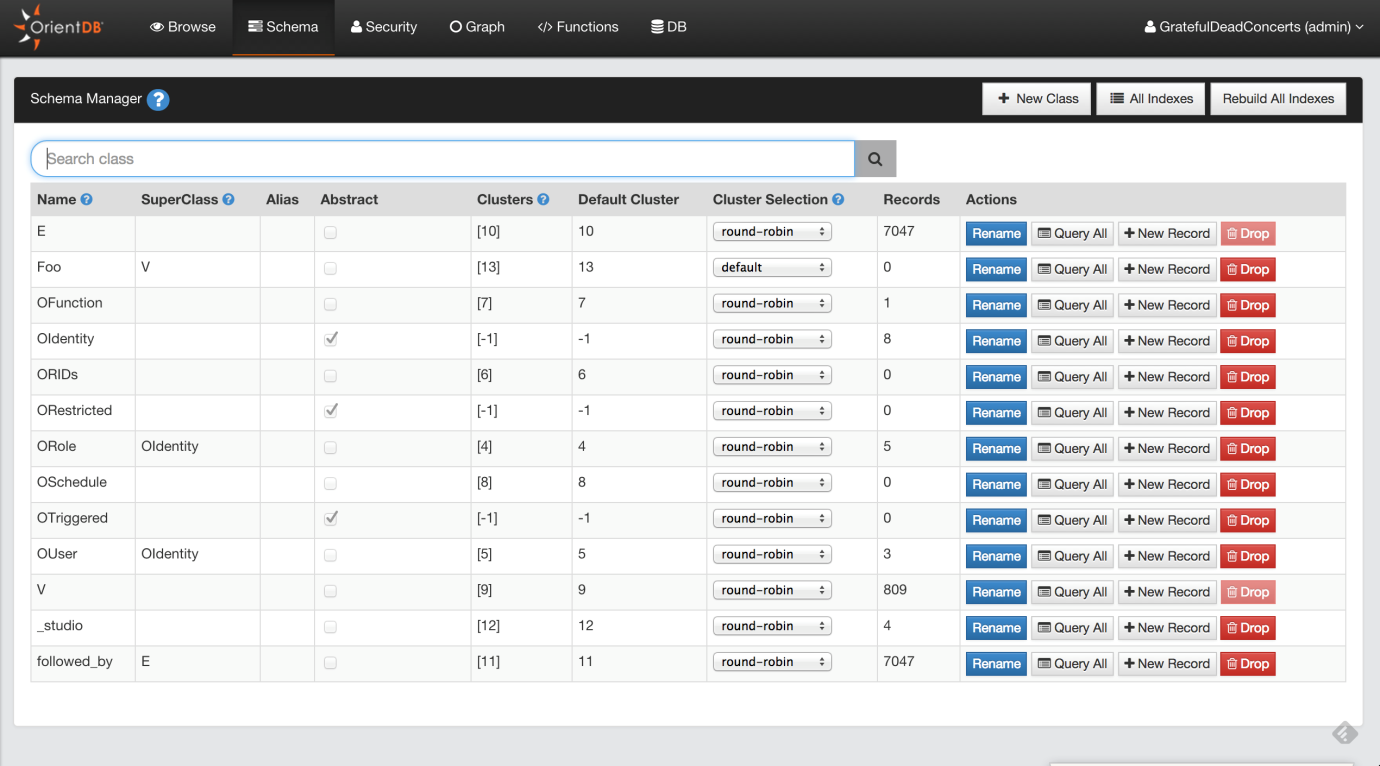


Edição de Vértice:



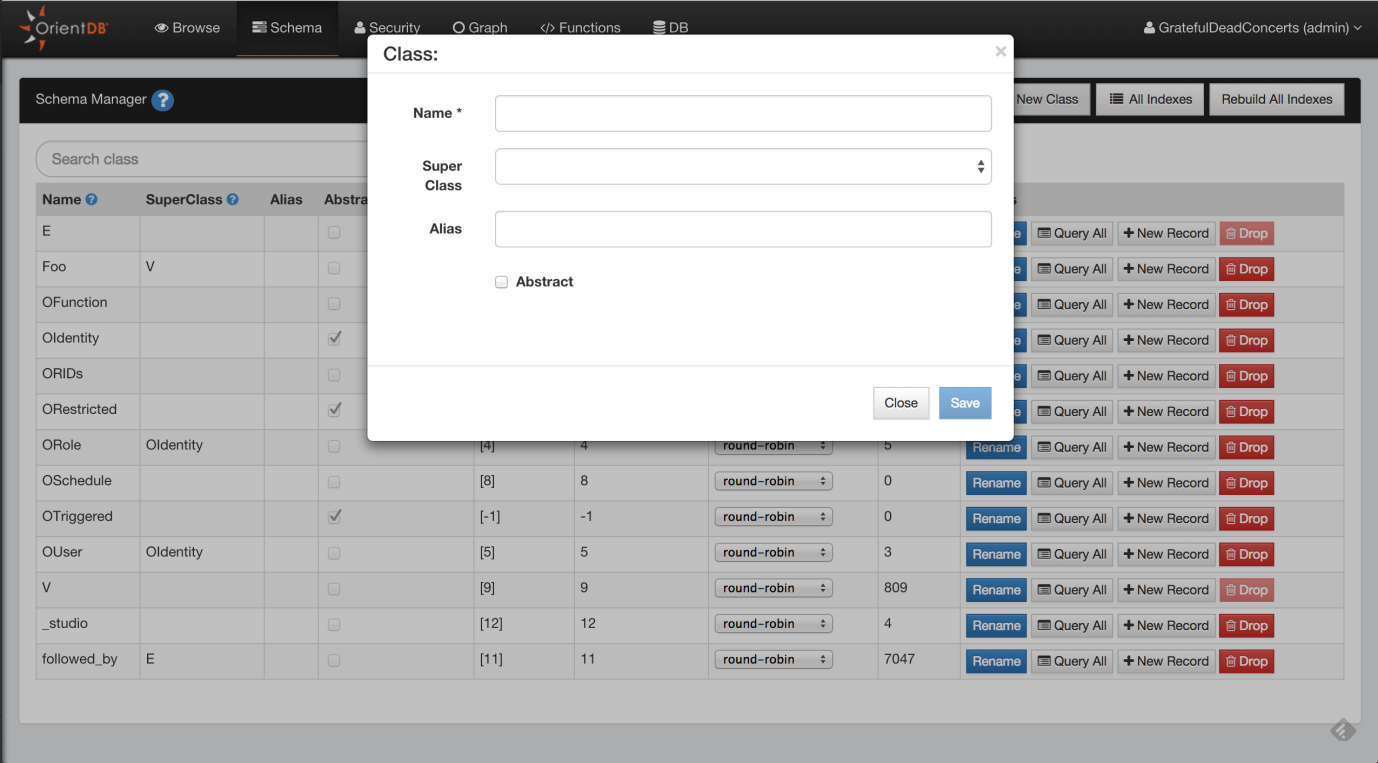
## Schema Mananger:

OrientDB pode funcionar em modo sem esquema, o modo de esquema ou uma mistura de ambos. Aqui vamos discutir o modo de esquema. Para saber mais sobre esquema no OrientDB.

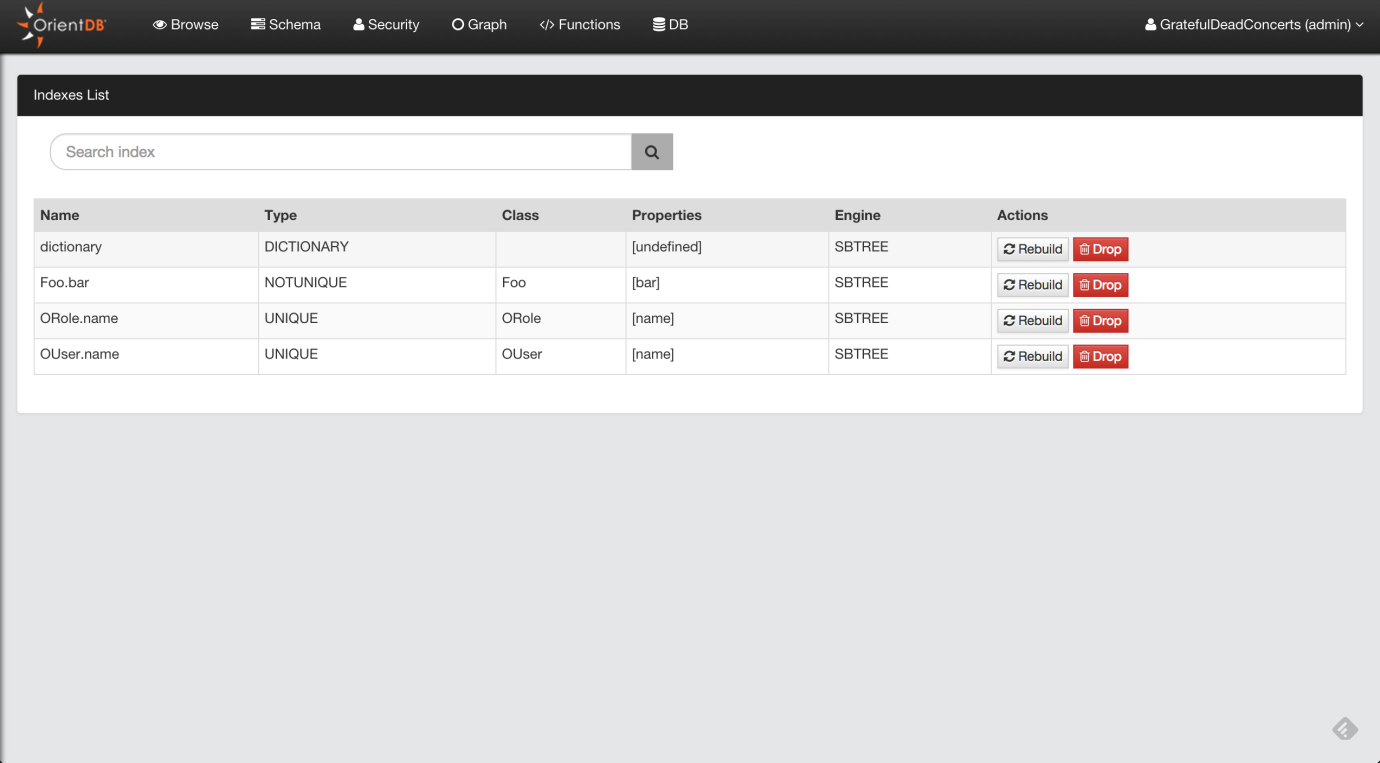


* Criar uma nova Classe:

Para criar uma nova classe, basta clicar na nova classe botão. Algumas informações é necessário para criar a nova classe.



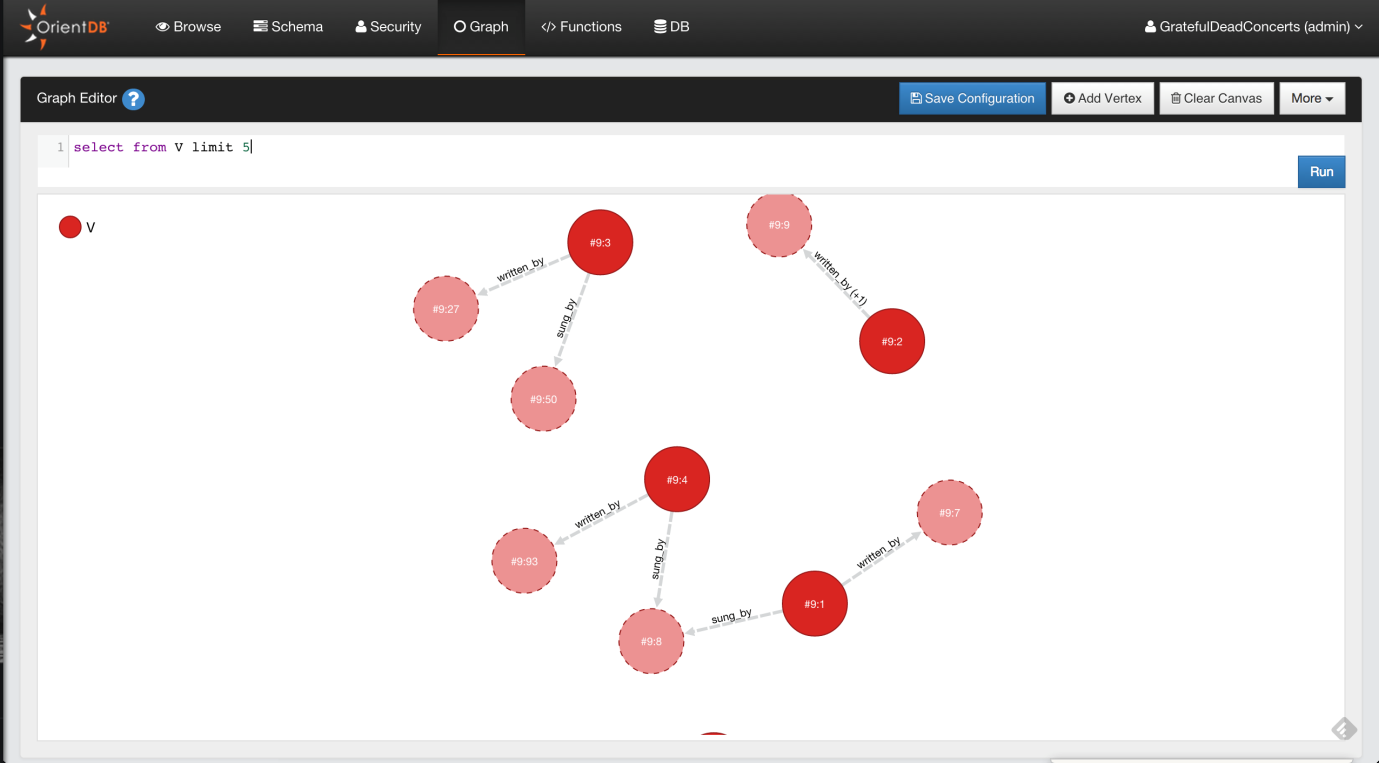
* Ver Índices:



## Editor grafo:

Desde Studio 2.0, temos um novo editor gráfico marca. Não apenas você pode visualizar seus dados de um modo gráfico, mas você também pode interagir com o gráfico e modificá-lo.

Para preencher a área do gráfico basta digitar uma consulta no editor de consulta ou utilizar a funcionalidade **Enviar para representar graficamente** a partir da [interface do usuário Navegar](http://orientdb.com/docs/master/Query.html).



* Operações possíveis com o editor grafo:

Adicionar Vértices; Salve a configuração Rendering Graph; Limpar o gráfico Rendering Canvas; excluir Vértices; Remover Vértices de lona; Editar Vértices; Inspecione Vértices; Alterar a configuração Rendering de vértices; navegando Relações; Criar arestas entre os vértices; Excluir arestas entre vértices; Inspecione Edges; Edit Edges.

O stuidio permite não só essas funções que mostramos, mas diversas outras nas áreas de Segurança, database management, painel de instrumentos, Server Managment, gerenciamento de cluster, Teleporter e outros.

## Consultas

Orient DB utiliza o SQL como linguagem de consulta, portanto os comandos e sintaxes são bem parecidos com o que estamos familiarizados nos bancos relacionais. Abaixo alguns exemplos:

* Select simples na class ouser:

select from OUser

* Similar ao SQL server, Orient DB suporta where nas condições das buscas:

select from Employee where city = 'Rome' order by surname asc, name asc

* Insert:

insert into Employee (name, surname, gender) values ('Jay', 'Miner', 'M')

* Update:

update Employee set local = true where city = 'London'

* Delete:

delete from Employee where city <> 'London'

# Clientes

CISCO, as nações unidas e Ericsson, são alguns dos principais clientes que utilizam oriente DB para armazenamento dos seus dados. Abaixo um caso de sucesso descrito pelo Dave Sims – Fundador e CEO da Floify. Um Portal Mutuário, que automatiza o processo de recolha de documentos de empréstimo de seus mutuários e envia atualizações de status de empréstimo para agentes imobiliários e mutuários:

"Quando eu comecei Floify, eu olhei para um banco de dados que preencheram todos os critérios abaixo:

* Bancos de dados que não fossem baseada em servidor (Postgres, etc).
* Bancos de dados que não fossem baseadas em Java (MongoDB, etc).
* Bancos de dados que suportam transações ACID.

Eu estreitei minhas escolhas para OrientDB e Neo4j, porem abaixo algumas razões pelas quais eu finalmente escolheu OrientDB sobre Neo4j:

* O desempenho Benchmarks de OrientDB foram muito impressionantes.
* A equipe de desenvolvimento OrientDB era muito ativo.
* Custo. OrientDB é gratuito. Neo4j custava mais do que o que eu estava disposto a pagar ou o que eu podia pagar.

Aqui estão alguns dos meus recursos favoritos em OrientDB:

* Eu posso criar o banco de dados em código.
* Eu não tenho de usar SQL para consultar, mas na maioria das vezes, eu faço. Eu já sei SQL, e é apenas fácil para mim.
* Eu uso o banco de dados de documentos, e é muito agradável inserir novos documentos em Java.
* Eu posso armazenar vários megabytes de objetos binários diretamente no banco de dados.
* Meu banco de dados é armazenado em um diretório no disco. Quando escalabilidade exige isso, eu posso atualizar para uma base de dados distribuída de dois servidores.
* Velocidade. Para mim, OrientDB é muito rápido, e nos poucos anos que eu tenho usado ele, tornou-se mais rápido.

No futuro, como as necessidades de desempenho e escalabilidade do Floify exigi-lo, eu vou investigar uma configuração de banco de dados multi-servidor em OrientDB. Nesse meio tempo, eu estou preparando para atualizar para OrientDB 2.0, que foi lançado recentemente e promete ainda mais velocidade."

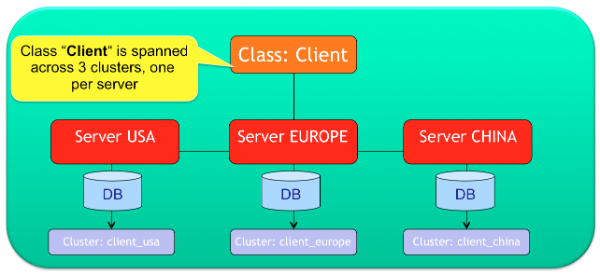
# Características Operacionais

## Sharding

OrientDB suporta sharding dos dados a nível de classe, usando vários clusters por classe, onde cada cluster tem sua própria lista de servidor onde os dados são replicados. No caso, todos os registros armazenados em grupos que fazem parte da mesma classe, são registros dessa classe.

Segue abaixo um exemplo de classe "Cliente" dividido em 3 clusters:

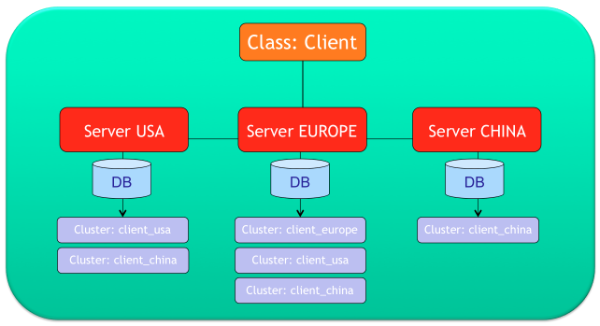
[Class](http://orientdb.com/docs/last/Concepts.html#class) **Client** -> [Clusters](http://orientdb.com/docs/last/Concepts.html#cluster) [ client\_usa, client\_europe, client\_china ]



É possível atribuir um cluster para um ou mais servidores(Nodos). Se mais servidores são adicionados, os registros serão copiados em todos os servidores. O primeiro servidor na lista será o servidor master para esse cluster.

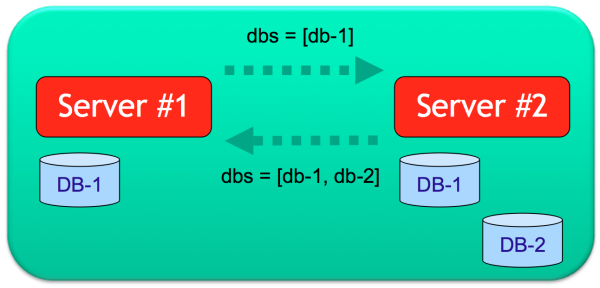
Abaixo um exemplo de configuração onde a classe Cliente foi dividido em 3 clusters: cliente\_usa, cliente\_europa e cliente\_china, cada um com configuração diferente:

* Cliente\_usa, será gerido por "EUA" e "Europa" nodes
* Cliente europa, será gerida apenas pelo node "europa"
* Client\_china, será gerido por todos os nodes

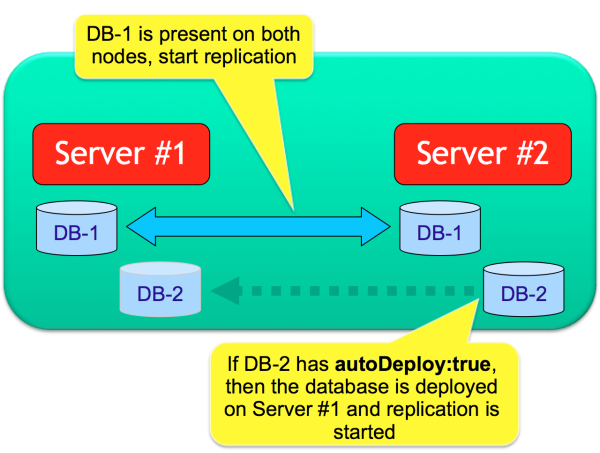


## Replicação

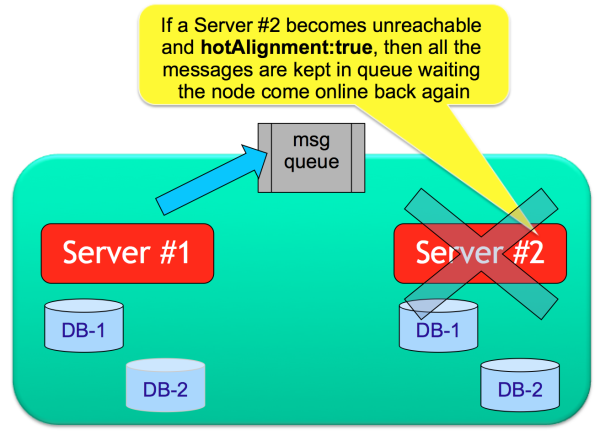
OrientDB suporta a replicação multi Master ou Master-Master. Isto significa que todos os nós do cluster são mestres e são capazes de ler e escrever no banco de dados. Quando um servidor OrientDB está iniciando, ele envia a lista de bancos de dados atuais para todos os nós do cluster. Se outros nós têm bancos de dados com o mesmo nome, a replicação é automaticamente definida.



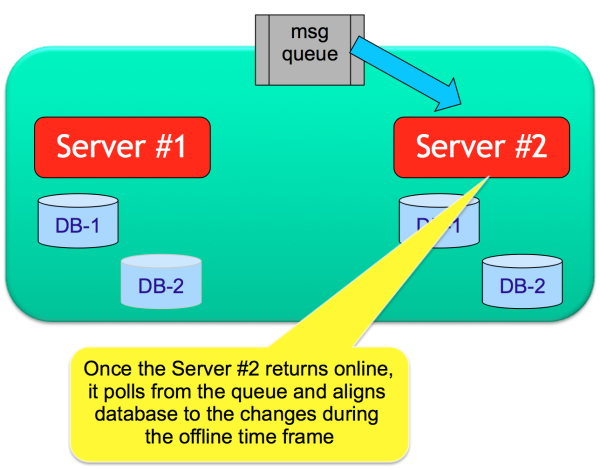
Se a configuração do banco de dados tem a configuração "autodeploy": true, então as bases de dados são automaticamente replicadas para os outros nós assim que eles forem implantados no cluster.



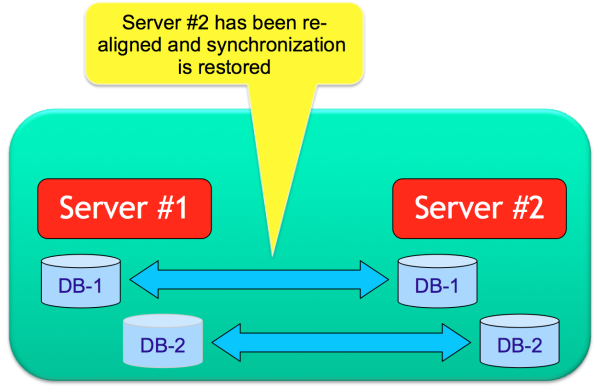
No caso de um servidor ficar inacessível, o nó é removido por configuração de banco de dados, a menos que a configuração seja "hotAlignment": true. Neste caso, todas as novas mensagens de sincronização são mantidas em uma fila distribuído.



Assim que o servidor ficar online novamente, é iniciado a fase de sincronização (status = sincronização). Todas os dados antes armazenados no pool, são transferidas para o nó reativado.



Uma vez que o alinhamento estiver terminado, o nó torna-se on-line (status = Online) e a replicação continua como antes.



## Índices

Em OrientDB podemos usar Índices ou configuração de caches para melhorar o desempenho de consultas. OrientDB suporta quatro algoritmos de índice:

* **SB-Tree Index** - Fornece uma mistura de recursos disponíveis nos outros tipos de índice, bom para uso geral. É durável, transacional e suporta varias consultas. É o tipo de índice padrão.
* **Hash Index** - Fornece pesquisa de lookup rápida e utiliza pouco espaço em disco. É durável e transacional, mas não suporta varias consultas. Ele funciona como um HashMap, o que torna mais rápido em pesquisas pontuais e consome menos recursos do que outros tipos de índice.
* **Lucene Full Text Index** - Fornece bons índices de texto completo, mas não pode ser usado para outros tipos de índice. É durável, transacional e suporta varias consultas.
* **Lucene Spatial Index** - Fornece bons índices espaciais, mas não pode ser usado para outros tipos de índice. É durável, transacional e suportes variam consultas.

OrientDB permite executar consultas em qualquer campo, indexados ou não-indexados. O motor de SQL reconhece automaticamente se nenhum índice pode ser usado para acelerar a execução. Também é possível consultar os índices diretamente usando índice: <index-name> como um alvo.

Exemplo: SELECT FROM INDEX:myIndex WHERE key = 'Jay'

## Cache

OrientDB possui níveis de cache. Quando OrientDB é executado em Distributed-Architecture, cada servidor tem seu próprio cache. Todos os caches em cada servidor são independentes.

É possível também ter um cache compartilhado entre os servidores, habilitando o cache de nível 2 Hazelcast. Para ativá-la, é necessario definir a propriedade **cache.level2.impl** no arquivo **orientdb-dserver-config.xml** com valor **com.orientechnologies.orient.server.hazelcast.Hazelcast:**

Nota: Essa configuração vai deixar lento as inserções de dados, mas vai melhorar as operações de consulta e pesquisa.

## Fail-over

Uma instância OrientDB pode falhar por várias razões:

* Problemas de HW, tais como perda de erro de alimentação ou de disco
* Problemas SW, tais como uma falha do sistema operacional
* Problema de aplicativo, como um bug que trave o seu aplicativo que está ligado ao Orient engine.

É possivel usar o OrientDB engine diretamente no mesmo processo da aplicação. Isso dá um desempenho superior devido à falta de comunicação entre processos. Neste caso, se aplicação falhar (por qualquer motivo), o Mecanismo de OrientDB também falha.

Em casos onde há servidor OrientDB conectado remotamente, caso aplicação falhe o engine orient db continua trabalhando, mas qualquer transação pendente de propriedade do cliente será revertida.

## Auto-recovery.

No start-up do OrientDB Engine é verificado se ele está reiniciando a partir de um acidente. Casos seja reiniciado a partir de um acidente a fase de auto-recuperação é iniciado, que reverte todas as transações pendentes. OrientDB tem diferentes níveis de durabilidade com base no tipo de armazenamento e configurações.

## Tipo de transações:

OrientDB suporta Transações otimistas, este modo utiliza o bem conhecido conceito multi Version Control System (MVCC), permitindo que várias leituras e escrita sobre os mesmos registros. A verificação de integridade é feita sobre commit. Se o registro foi salvo por outra transação no intermediaria, então a exceção ConcurrentModificationException será lançada. A aplicação pode optar por repetir a operação ou abortá-lo. O banco não aceita transação pessimista, ou seja, bloqueio do registro na fonte de dados enquanto o dado sofre modificações.

NOTA: OrientDB mantém a transação em memória RAM do cliente, de modo que o tamanho da transação é afetada pelo espaço disponível de RAM na JVM. Para operações que envolvem muitos registros, considere dividi-lo em várias transações.

# Referencias

http://orientdb.com/orientdb/