

Integrantes:

Anderson Silva

Gabriel Guimarães

Graziele Loureiro

Matheus Freitas

Índice

[1. Histórico 4](#_Toc455222170)

[2. Novidades 5](#_Toc455222171)

[3. Características 5](#_Toc455222172)

[4. Exemplos 6](#_Toc455222173)

[4.1. Consultas 6](#_Toc455222174)

[5. Requisitos Mínimos para Instalação 7](#_Toc455222175)

[6. Casos de Sucesso 8](#_Toc455222177)

[7. Sharding 8](#_Toc455222183)

[8. Segurança dos Dados Persistidos 8](#_Toc455222191)

[9. Referências 9](#_Toc455222194)

# Histórico

Desenvolvido por uma startup de mesmo nome, o RethinkDB é o primeiro banco de dados open-source construído para aplicações web real-time. Fundado em 2009, seu código se tornou público em 2012, com a versão 1.2. Em abril de 2015 foi lançada a 2.0, sua primeira versão production-ready.

O RethinkDB surgiu para facilitar respostas em tempo real, funcionam em formato Push, onde o servidor contata o cliente, em intervalos determinados, a empresa recomenda seu uso para aplicativos web colaborativos, aplicações analíticas com atualizações frequentes, jogos multiplayer, lojas online em tempo real e aplicativos conectados.

No início de Abril de 2016, foi lançada a versão 2.3 do Rethink introduzindo melhorias quanto à segurança e desempenho da linguagem de consulta.

Segundo o <http://db-engines.com/en/ranking> o RethinkDB está na 45º posição entre todos os databases no ranking de popularidade, em Março de 2016. Um ano antes ocupava a posição 86. Entre os bancos de dados de documentos, ele ocupava em Março de 2015, a 12º posição e atualmente está como o 7º mais popular.

# Novidades

A nova versão de casamento de comandos é construída com o [re2](https://code.google.com/p/re2/) do Google, e permite que expressões regulares possam ser utilizadas em filtros ou em índices secundários para acesso mais eficiente.

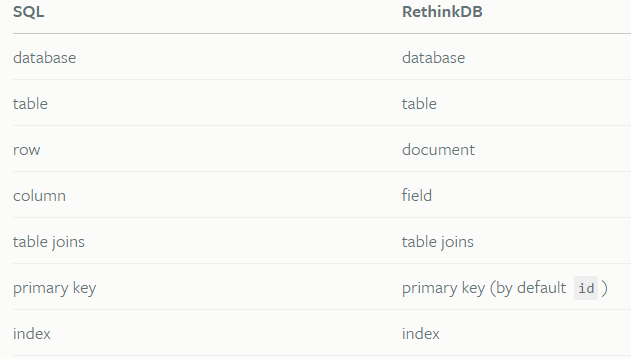
As novas operações de array abrange operações como adição, alteração e remoção de elementos (prepend, append, insertAt, deleteAt, changeAt), mesclagem e separação de arrays (spliceAt, add, mul, difference), além de testes de conteúdo de array (count, indexesOf, isEmpty).

Também existem as operações para grupos como inserção, união, diferença e interseção.  
  
 O RethinkDB está disponível para [download](http://www.rethinkdb.com/docs/install/), como executável e [código fonte](http://www.rethinkdb.com/docs/build/), para Mac OS X e Ubuntu. O servidor está sob a licença GNU AGPL v3.0, enquanto os drivers do cliente estão sob a licença Apache 2.0.

# Características

O Rethink é um banco de dados que armazena documentos que utilizando JSON.

Os documentos são armazenados em tabelas, mas ao contrário do que é conhecido como tabela nos bancos de dados relacionais, no Rethink, tabelas se referem apenas ao ‘espaço lógico’ onde ficam os documentos (Rethink é schema-less).



No RethinkDB existem databases e tables a linha se compara a um “document”, a coluna um field, existem joins.

# Exemplos

ReQL é a linguagem de consultas do Rethink, ela é diferente das outras linguagens NoSQL e possui 3 princípios-chave:

1. Incorpora sua linguagem de programação. Se sua aplicação é construída em Python, por exemplo, você pode usar a syntax convencional de Python para escrever suas queries.
2. Todas as queries são encadeáveis. Você inicia com uma tabela e incrementa encadeando transformadores no fim query utilizando o operador ‘.’.
3. Todas as queries são executadas no servidor. A função ‘run’, encadeada no final da query, faz com que ela seja executada no servidor e retorne apenas a saída.

ReQL ainda possibilita a utilização de ‘join’.

## Consultas

**Create and drop the database:**

r.dbCreate('my\_database')

r.dbDrop('my\_database')

r.tableCreate('users', {primaryKey: "id"})

r.table("users").delete()

**Insert into the table:**

r.table("users").insert({  
   userId: "f62255a8259f",  
  age: 30,  
  name: "Peter"  
 })

**Select**

r.table("users")

r.table("users")  
 .pluck("userId", "name")

r.table("users").filter({  
 name: "Peter"  
 })

**Update**

r.table("users").filter(  
   r.row("age").lt(18)  
 ).update({age: 18})

**Delete**

r.table("users").delete()

r.table("users")  
    .filter(r.row("age").lt(18))  
   .delete()

**Join (Inner join, outer join and left join):**

r.table("posts").innerJoin(  
  r.table("users"),  
  function (post, user) {  
    return post("userId").eq(user("id"));  
 }).zip()

**Listen for changes:**

r.db("advanced\_db").table("landmarks").changes();

# Requisitos Mínimos para Instalação

# Não existe uma restrição forte de requisitos, ele roda em linux em 32 e 64 bits, enquanto em windows apenas em 64 bits e recomenda-se pelo menos 2 GB de Ram para rodar, devido ao seu mecanismo de cache.

# Casos de Sucesso

# No lançamento da versão 2.0 Slava Akhmechet (@coffeemug) diz que mais de 100.000 desenvolvedores estão utilizando e cita exemplos como:

# Jive Software e Mediafly usam para aplicativos e páginas web reativas.

# Pristine.io e Narrative Clip usam para conectar aparelhos conectados a infraestrutura de núvem deles.

# Platzi e Workshape.io usam para analises em tempo real.

# CMUNE e NodeCraft usam para jogos de multiplayer massivos e escaláveis.

# Sharding

# No Rethinkdb o sharding se dá por base de tabela, primeiro o serviço tem que ser iniciado com um parâmetro para dizer que ele é parte de um ou mais clusters, para em seguida configurar a replicação da tabela.

# O Rethinkdb possui replicação que funciona de modo master-slave e tem mecanismos de failover automatico caso um número maior que a metade dos nós esteja disponível para votar o novo master.

# Sharding, o Rethinkdb paraleliza as consultas para melhorar a performance, então ao replicar os dados se ganha performance sem ter que modificar as consultas, a exceção são consultas de order by sem índice, a ordem com que os comandos são encadeados pode afetar a performance, dado que só alguns comandos são executados em paralelo. Também existem estruturas chamadas índices secundarios, que sempre necessitam que a query seja modificada para que haja ganho de performance, esses índices podem ser:

# Índices simples: compostos de apenas um campo;

# Índices compostos: podem conter várias colunas;

# Multi-indice: baseado em arrays de valores (Ex.: Tags e um Blog Post);

# Índices baseados em expressões arbitrárias.

# Segurança dos Dados Persistidos

# A recém-lançada versão 2.3 do RethinkDB possui melhorias quanto à administração do banco de dados. Possui criptografia de rede, TLS para criptografar as conexões com o BD.

# Usuários e permissões: você pode conseguir o controle granular sobre o acesso de dados através da criação de contas de usuário e atribuir permissões.

# Referências

<https://www.rethinkdb.com/>