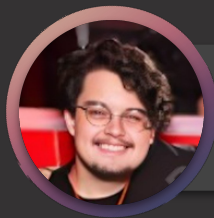


Database



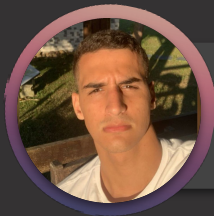
Grupo: Sherlock



Antonio Henrique Leitão Barros
Ciência da computação
Matrícula: 01647043



Daniel de Araújo Procópio da
Cunha
Sistemas de informação
Matrícula: 01643124



Nickolas Mac'Hamilton Renaux
Alves
Sistemas de informação
Matrícula: 01651053



Rafael José de Araújo Procópio da
Cunha
Sistemas de informação
Matrícula: 01589882



Gabriel de Luna Cavalcanti
Ciência da computação
Matrícula: 01331511

Cassandra



É um banco de dados NoSQL do tipo chave-valor, seu estilo é “BASE”, no entanto possui algumas poucas características semelhantes ao “ACID”.

Prioriza disponibilidade de dados, logo é considerado “Basically Available”.

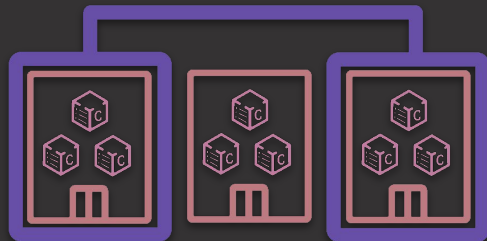
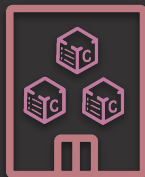
Não precisa ser consistente o tempo todo, sendo então “Soft-state”.

Consistente em momentos indeterminados, ou seja “Eventually consistent”

Suas funcionalidades foram arquitetadas com escalabilidade e alta avaliabilidade em mente, teoricamente, pode conter uma quantidade de nodos ilimitados em um cluster, e clusters são dispersos e os seus dados são transacionados entre diversos pontos.

Desnormalizado.

CAP & Estruturas



Nodes

Os nodos são instâncias individuais do Cassandra, cada nodo geralmente corresponde a uma máquina, seja ela uma unidade física ou virtualizada.

Racks

Agrupamento lógico de nodos em um mesmo datacenter, usados para definir o layout físico, comunicação e topologia.

Cluster

Os clusters são um conjunto de nodos organizados de maneira lógica e de formas diversas, podendo estar a grandes distâncias geograficamente ou fisicamente próximos.

Pontos

Positivos

- Basicamente disponível -> Muitos usuários podem acessar o banco de dados e garante disponibilidade até na presença de falhas.
- Arquitetura distribuída -> Dados são replicados em diversos nodos, que comunicam-se entre si através de ações como o “gossip protocol” e “hinted handoff”.
- Escalabilidade e “no-bottleneck” -> É possível aumentar o armazenamento e processamentos de forma horizontal onde cada nodo processa um certo workload.

Negativos

- Eventualmente consistente -> Seus dados são atualizados eventualmente.
- Complexidade da estrutura -> Seu “schema” deve ser cuidadosamente desenvolvido, já que desnormalização é uma prática comum no Cassandra, o que pode aumentar a complexidade da busca e processamento.
- Investimento em escalabilidade -> Escalar horizontalmente gera custos maiores, e também adiciona complexidade ao sistema.

DDL



CREATE KEYSPACE

ALTER KEYSPACE

DROP KEYSPACE

CREATE TABLE

ALTER TABLE

DROP TABLE

(ADD)

DML



INSERT

DELETE

UPDATE

(INTO)

(VALUES)

(SET)

(WHERE)

DQL



SELECT

(FROM)

(WHERE)

(COUNT)

(SUM)

(AVG)

(MIN)

(MAX)

(*)

Referências

O que é Cassandra

<https://www.scylladb.com/learn/apache-cassandra/introduction-to-apache-cassandra/#:~:text=Cassandra%20is%20not%20a%20fully,commits%2C%20rollbacks%20or%20locking%20mechanisms.>

Cassandra clusters

<https://www.scylladb.com/glossary/cassandra-cluster/#:~:text=A%20Cassandra%20cluster%20is%20a,physical%20outages%20of%20underlying%20infrastructure.>

“ACID” & “BASE”

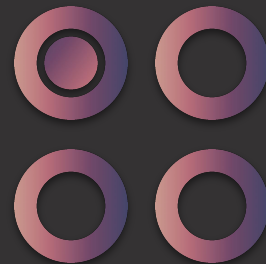
<https://aws.amazon.com/pt/compare/the-difference-between-acid-and-base-database/>

Comandos DDL, DML & DQL

https://docs.datastax.com/en/cql-oss/3.3/cql/cql_reference/cqlCommandsTOC.html

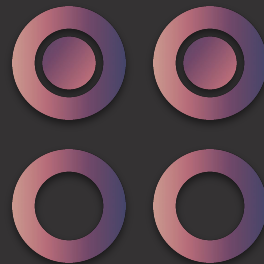
Cassandra no docker

```
MINGW64~/c/Users/Antonio H: x + v
Antonio Henrique@DESKTOP-KLL6V0M MINGW64 ~
$ docker pull cassandra
```



Cassandra no docker

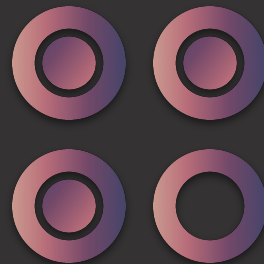
```
MINGW64/c/Users/Antonio H x + v
Antonio Henrique@DESKTOP-KLL6V0M MINGW64 ~
$ docker run -p 7000:7000 -p 7001:7001 -p 7199:7199 -p 9042:9042 -p 9160:9160 --name cs_db -d cassandra:latest
```



Cassandra no docker

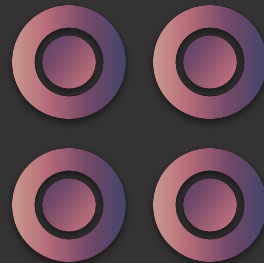
```
MINGW64/c/Users/Antonio H... x + v
Antonio Henrique@DESKTOP-KLL6V0M MINGW64 ~
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        NAMES
9a07f23de315   cassandra:latest  "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:7000-7001→7000-7001/tcp, 0.0.0.0:7199→7199/tcp, 0.0.0.0:9042→9042/tcp, 0.0.0.0:9160→9160/tcp  cs_db

Antonio Henrique@DESKTOP-KLL6V0M MINGW64 ~
$ |
```



Cassandra no docker

```
root@9a07f23de315: /  
Antonio Henrique@DESKTOP-KLL6V0M MINGW64 ~  
$ docker exec -it 9a07f23de315 bash  
root@9a07f23de315: /# cqlsh  
Connected to Test Cluster at 127.0.0.1:9042  
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]  
Use HELP for help.  
cqlsh>
```



CREATE & USE

```
CREATE KEYSPACE cs_db WITH replication = {'class':'SimpleStrategy', 'replication_factor' : 1};

USE cs_db;

CREATE TABLE IF NOT EXISTS player_info (
  player_id INT PRIMARY KEY,
  nickname TEXT,
  realname TEXT,
  age INT,
  country TEXT,
  current_team TEXT,
  teams SET<TEXT>,
  total_kills INT,
  total_deaths INT,
  headshot_percentage DOUBLE,
  maps_played INT,
  rounds_played INT,
  kd DOUBLE,
  kill_death_diff INT,
  total_opening_kills INT,
  total_opening_deaths INT,
  opening_kill_ratio DOUBLE,
  opening_kill_rating DOUBLE,
  first_kill_won_round DOUBLE,
  rating DOUBLE,
  rifle_kills INT,
  sniper_kills INT,
  smg_kills INT,
  pistol_kills INT,
  grenade_kills INT,
  other_kills INT,
  damage_per_round DOUBLE,
  grenade_dmg_per_round DOUBLE,
  kills_per_round DOUBLE,
  assists_per_round DOUBLE,
  deaths_per_round DOUBLE,
  saved_by_team DOUBLE,
  saved_team DOUBLE,
  rounds_with_kills DOUBLE,
  team_win_after_first_kill DOUBLE,
  zero_kill_rounds INT,
  one_kill_rounds INT,
  two_kill_rounds INT,
  three_kill_rounds INT,
  four_kill_rounds INT,
  five_kill_rounds INT
);
```

INSERT VALUES

```
INSERT INTO player_info (  
    player_id,  
    nickname,  
    realname,  
    age,  
    country,  
    current_team,  
    teams,  
    total_kills,  
    total_deaths,  
    headshot_percentage,  
    damage_per_round,  
    grenade_dmg_per_round,  
    maps_played,  
    rounds_played,  
    kd,  
    kills_per_round,  
    assists_per_round,  
    deaths_per_round,  
    saved_by_team,  
    saved_team,  
    rounds_with_kills,  
    kill_death_diff,  
    total_opening_kills,  
    total_opening_deaths,  
    opening_kill_ratio,  
    opening_kill_rating,  
    team_won_after_first_kill,  
    first_kill_won_round,  
    zero_kill_rounds,  
    one_kill_rounds,  
    two_kill_rounds,  
    three_kill_rounds,  
    four_kill_rounds,  
    five_kill_rounds,  
    rifle_kills,  
    sniper_kills,  
    smg_kills,  
    pistol_kills,  
    grenade_kills,  
    other_kills,  
    rating  
) VALUES (  
    18835,  
    'saffee',  
    'Rafael Costa',  
    27, 'Brazil',  
    'FURIA',  
    {'FURIA', 'paiN'},  
    8482,  
    6517,  
    30.9,  
    80.2,  
    3.0,  
    404,  
    10701,  
    1.3,  
    0.79,  
    0.1,  
    0.61,  
    0.08,  
    0.1,  
    5488,  
    1965,  
    1309,  
    892,  
    1.47,  
    1.14,  
    76.1,  
    17.3,  
    5213,  
    3288,  
    1548,  
    524,  
    114,  
    14,  
    2545,  
    4412,  
    264,  
    1209,  
    52,  
    28,  
    1.22  
);
```

SELECT

```
root@9a07f23de315: /
cqlsh:cs_db> select nickname from player_info;

nickname
-----
shiro
simple
deko
saffee
Zyw0o
(5 rows)
cqlsh:cs_db> SELECT nickname FROM player_info WHERE country = 'Brazil' ALLOW FILTERING;

nickname
-----
saffee
(1 rows)
cqlsh:cs_db> SELECT realname, nickname
... FROM player_info
... WHERE kill_death_diff > 5000
... ALLOW FILTERING;

realname | nickname
-----+-----
Dmitry Sokolov | shiro
Aleksandr Kostylev | simple
Mathieu Herbaut | Zyw0o
(3 rows)
cqlsh:cs_db> |
```

OBRIGADO!!!