

AULA 16 – DATA WAREHOUSE

PROFA. DRA. LEILA BERGAMASCO

CC5232 – Banco de Dados

NA AULA DE HOJE

- NoSQL

BANCO DE DADOS NÃO-RELACIONAIS, ORIENTADOS A
DOCUMENTOS, NOSQL, PÓS RELACIONAIS, HELP!

BANCO DE DADOS NÃO RELACIONAL

- Classe de banco de dados que fornecem um mecanismo para armazenamento e recuperação de dados que são modelados de formas diferentes das relações tabulares usadas nos bancos de dados relacionais.
- Características:
 - Contém todas as informações importantes em um único documento,
 - Livre de esquemas
 - Possui identificadores únicos universais (UUID)
 - Possibilitar a consulta de documentos através de métodos avançados de agrupamento e filtragem (MapReduce) e...
 - Permite redundância e inconsistência.



BANCO DE DADOS NÃO RELACIONAL

- NoSQL – (Not Only SQL)
 - Bem aceito na comunidade, porém as características de uma banco não relacional vão além de não ser baseado em SQL
- Orientado a Documento ou Chave-valor
 - Parcialmente correto, os bancos não relacionais podem ser do tipo Orientado a documento ou chave-valor
- Pós-relacional
 - Bem aceito mas INCORRETO. O conceito existe desde 1960! Não é evolução do relacional, é uma alternativa para DETERMINADAS SITUAÇÕES
- NoRELAT – (Não relacional)
 - Taxonomia correta porém pouco aceita na comunidade.

VARIAÇÕES

- Orientado a documento
 - Chave valor
 - Orientado a coluna
 - Orientado a grafo
-
- Mongo DB é o líder
 - Orientado a documento

VARIAÇÕES

- Orientado a documento
 - “banco de dados não relacional projetado para armazenar e consultar dados como documentos do tipo JSON. Os bancos de dados de documentos facilitam para que os desenvolvedores armazenem e consultem dados usando o mesmo formato de modelo de documento que usam no código do aplicativo. A natureza flexível, semiestruturada e hierárquica dos documentos e dos bancos de dados de documentos permite que eles evoluam conforme as necessidades dos aplicativos” (AWS doc)
- JSON - JavaScript Object Notation

```
[
  {
    "year" : 2013,
    "title" : "Turn It Down, Or Else!",
    "info" : {
      "directors" : [ "Alice Smith", "Bob Jones"],
      "release_date" : "2013-01-18T00:00:00Z",
      "rating" : 6.2,
      "genres" : ["Comedy", "Drama"],
      "image_url" : "http://ia.media-imdb.com/images/N/O9ERWAU7FS797AJ7LU8HN09AMUP908RLlo5JF90EWR7LJKQ7@@._V1_SX400_.jpg",
      "plot" : "A rock band plays their music at high volumes, annoying the neighbors.",
      "actors" : ["David Matthewman", "Jonathan G. Neff"]
    }
  },
  {
    "year": 2015,
    "title": "The Big New Movie",
    "info": {
      "plot": "Nothing happens at all.",
      "rating": 0
    }
  }
]
```


VARIAÇÕES

- Chave valor
 - “banco de dados não relacional que usa um método de chave-valor simples para armazenar dados. Um banco de dados de chave-valor armazena dados como um conjunto de pares de chave-valor em que uma chave funciona como um identificador exclusivo. A chave e os valores podem ser qualquer coisa, desde objetos simples até objetos compostos complexos” (AWS doc)
 - Possibilita a visualização do BD como uma grande tabela hash;
 - Maior nível de eficiência;
 - Possui as estruturas mais simples das soluções NoSQL.

```
json
{
  "biblioteca": {
    "A10:3": [
      {
        "titulo": "Harry Potter e a Ordem da Fênix",
        "autor": "J.K. Rowling"
      },
      {
        "titulo": "Harry Potter e a Pedra Filosofal",
        "autor": "J.K. Rowling"
      }
    ],
    "A10:12": [
      {
        "titulo": "O Hobbit",
        "autor": "J.R.R. Tolkien"
      },
      {
        "titulo": "O Senhor dos Anéis: A Sociedade do Anel",
        "autor": "J.R.R. Tolkien"
      }
    ]
  }
}
```

Chave	Valor
16	nome = NoSQL Essencial, ano - 2014
Connor	idade = 22, interesse = programação
2	nome = True Blood, gênero = fantasia, classificação = 16 anos
Laís	ocupação = estudante

VARIAÇÕES

- Baseado em colunas
 - “O armazenamento orientado a colunas para tabelas do banco de dados é um fator importante para a performance de consulta analítica, pois ele reduz expressivamente os requisitos gerais de E/S de disco e diminui a quantidade de dados que você precisa carregar do disco.” (AWS doc)

VARIAÇÕES

Id	Nome	Sobrenome	Idade
1	João	Pereira	32
2	Carlos	Gonçalves	41
3	Ana	Silva	13

■ BDs tradicionais

- Possuem índices, chaves primárias
- Fazem armazenamento linear

1	João	Pereira	32	2	Carlos	Gonçalves	41	3	Kondado	Inteligência	13
---	------	---------	----	---	--------	-----------	----	---	---------	--------------	----

- Saber os dados do Id=1 → OK! 1 acesso
- Saber a soma das idades → Percorrer vários blocos de memória (E/S)
- Evitar busca sequencial → Índices!

■ BD colunares

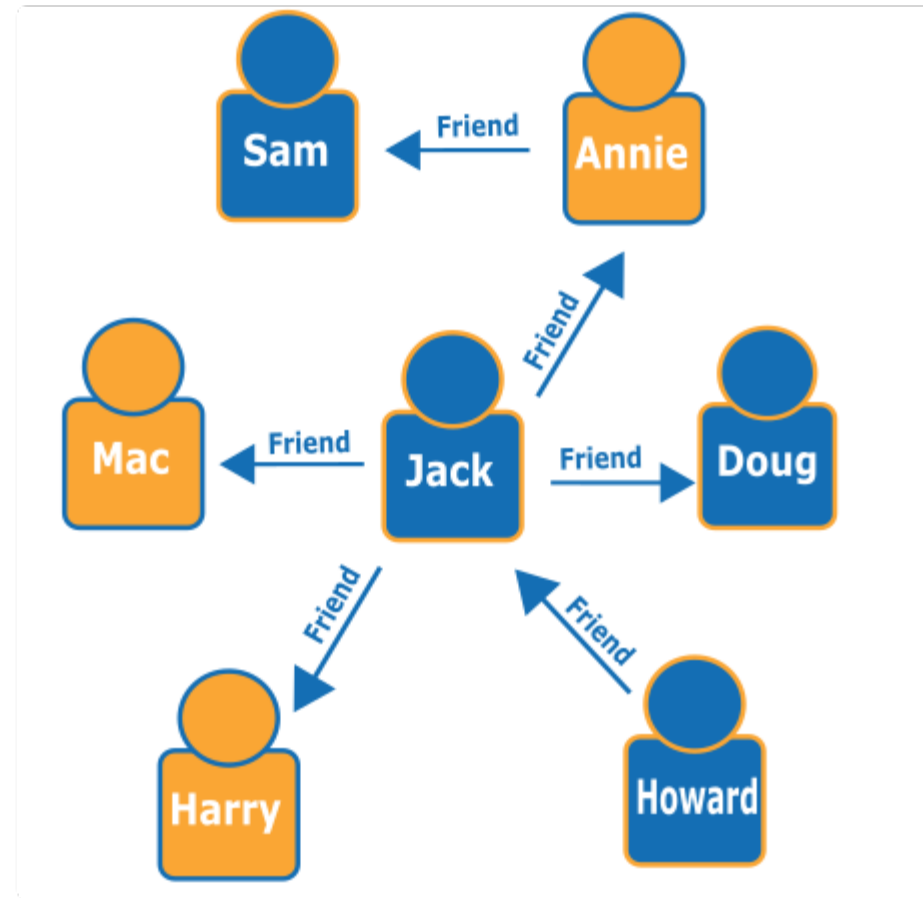
- Maior compressão – já que dados de tipos iguais são armazenados juntos, há uma otimização de espaço utilizado
- Eliminação da necessidade de índices – não é necessário
- Alta-performance para operações de agregação

1	João	Pereira	32	2	Carlos	Gonçalves	41	3	Kondado	Inteligência	13
1	2	3	João	Carlos	Kondado	Pereira	Gonçalves	Inteligência	32	41	13

Avaliar as consultas é muito importante!!

VARIAÇÕES

- Baseado em grafos
 - “criados especificamente para possibilitar o armazenamento de relacionamentos e a navegação por eles. Os relacionamentos são elementos distintos que agregam a maior parte do valor para os bancos de dados grafo. Os bancos de dados grafo usam nós para armazenar entidades de dados e bordas para armazenar os relacionamentos entre as entidades.” (AWS doc)



MONGO DB

- Open Source
- Vantagens
 - Não precisa projetar o esquema do banco de dados ao trabalhar com o mongoDB;
 - Fornece grande flexibilidade para os campos nos documentos;
 - Trabalha com dados heterogêneos;
 - Não requer nenhuma adição ou injeção de SQL;
 - Facilmente integrável com o Big Data Hadoop (com diversas versões open source).
- Desvantagens
 - Utiliza muita memória para armazenar e estocar dados;
 - Limite de 16 MB de dados para armazenar nos documentos;
 - Limite para aninhar dados nos arquivos BSON em até 100 níveis.

SINTAXE MONGO DB

- **Databases:** Container físicos dos dados. Cada *database* contém seus próprios arquivos no sistema de arquivos. Cada servidor pode possuir uma ou mais *databases*.
- **Collections:** Uma *collection* é uma coleção de documentos, é o equivalente a uma tabela de um banco de dados SQL. Cada *collection* existe dentro de uma *database* e elas não possuem um esquema definido. Os documentos dentro de uma *collection* podem ter campos diferentes, mas cada um destes elementos geralmente são relativos a um mesmo propósito ou algo similar.
- **Documentos:** Um documento é um par de chave-valor, os quais não possuem um esquema, conforme explicado anteriormente, o que significa que documentos diferentes dentro de uma mesma *collection* podem ter campos diferentes e mesmo os campos iguais podem ser de tipos diferentes de dados.

SINTAXE

```
{
  _id: ObjectId(57d0849edab45f72a3994895)
  titulo: 'Exemplo de MongoDB',
  descricao: 'MongoDB é uma base NoSQL',
  por: 'Professor Dirceu Semighini Filho',
  site: 'http://www.dirceu.professor.blogspot.com.br',
  tags: ['mongodb', 'banco de dados', 'NoSQL'],
  likes: 100,
  comentarios: [
    {
      usuario: 'aluno1',
      mensagem: 'Comentário de exemplo',
      dataCriacao: new Date(2016,8,20,2,15),
      likes: 0
    },
    {
      usuario: 'aluno2',
      mensagem: 'Outro comentário de exemplo',
      dataCriacao: new Date(2016,8,25,7,45),
      likes: 5
    }
  ]
}
```

SINTAXE

```
DBCollection collection =  
mongoDB.getCollection(professor),  
collection.insert(inserir);
```

```
{  
  _id: ObjectId(57d0849edab45f72a3994895)  
  nome: 'Dirceu'  
  titulo: 'Exemplo de MongoDB',  
  descricao: 'MongoDB é uma base NoSQL',  
  por: 'Professor Dirceu Semighini Filho',  
  site:  
    'http://www.dirceu.professor.blogspot.com.br',  
  tags: ['mongodb', 'banco de dados', 'NoSQL'],  
  likes: 100,  
  comentarios: [  
    {  
      usuario: 'aluno1',  
      mensagem: 'Comentário de exemplo',  
      dataCriacao: new Date(2016,8,20,2,15),  
      likes: 0  
    },  
    {  
      usuario: 'aluno2',  
      mensagem: 'Outro comentário de exemplo',  
      dataCriacao: new Date(2016,8,25,7,45),  
      likes: 5  
    }  
  ]  
}
```

SINTAXE

- Para listar as collections da base de dados que estamos usando faça:

```
db.getCollectionNames()
```

```
[ "professor", "system.indexes" ]
```

- Para apagar a sua base de dados, digite:

```
db.dropDatabase()
```

- Para listar os dados de uma collection use o comando **find**, da seguinte maneira:

```
db.professor.find()
```

```
{ "_id" : ObjectId("57d0849edab45f72a3994895"), "nome" : "Dirceu" }
```

- Para apagar uma collection execute o comando drop:

```
db.professor.drop()
```

SINTAXE

```
SELECT * FROM usuarios WHERE idade > 25 AND idade <= 50
db.usuarios.find(
    { idade: { $gt: 25, $lte: 50 } } )
```

```
SELECT * FROM usuarios WHERE nome like "Higor%"
db.usuarios.find(
    { nome: /^Higor/ }
)
```

```
SELECT * FROM usuarios WHERE cidade = "Porto Alegre" ORDER BY nome DESC
db.usuarios.find({cidade : "Porto Alegre"}).sort({nome:-1})
```

```
SELECT * FROM usuarios u INNER JOIN eventos e ON e.id = u.id
    WHERE e.publicado is not null
    GROUP BY u.email
db.usuarios.find(('eventos.publicado': ($ne: null)))
```

SINTAXE

- Find()
 - \$gt – greater than
 - \$gte – greater than or equal
 - \$lt – less than
 - \$lte – less than or equal
- aggregate()
 - \$sort – order by
 - \$Project - limit

VANTAGENS E DESVANTAGENS

Vantagens	Desvantagens
Base de dados orientada para objetos <u>flexível</u> (não estando presa a um esquema de inter-relação de tabelas).	Não irão resolver problemas de escalabilidade de um Website.
Facilidade em <u>adicionar novos dados</u> . <u>Dados sempre disponíveis</u> .	Mais recentes do que as bases de dados Relacionais, não contando com tantas alternativas como o modelo Relacional.
Excelente maneira de lidar com o problema do <u>Big Data</u> (gere bem o armazenamento de grandes quantidades de informação).	Suporte Técnico: Devido ao facto de grande parte dos sistemas NoSQL serem Open Source, o suporte é geralmente oferecido por empresas muito mais pequenas, sem o peso de grandes nomes existentes no mercado das bases de dados relacionais, como é o caso da Oracle, Microsoft ou IBM.
Necessitam de muito <u>menos manutenção</u> , sendo a reparação e distribuição de dados feita, na sua maioria, automaticamente. Numa base de dados Relacional (SQL), é sempre necessário um administrador, responsável pela mesma.	Dificuldade de instalação e esforço para manutenção.
O <u>custo</u> desta base de dados é <u>mais reduzido</u> .	Problemático trabalhar com muitos joins

FERRAMENTAS

←EROSPIKE



Clustrix



Couchbase



amazon
DynamoDB

APACHE
HBASE



MarkLogic



memsql
Speed. Scale. Simplicity.



mongoDB



NUODB

ORACLE®

NOSQL
DATABASE



riak

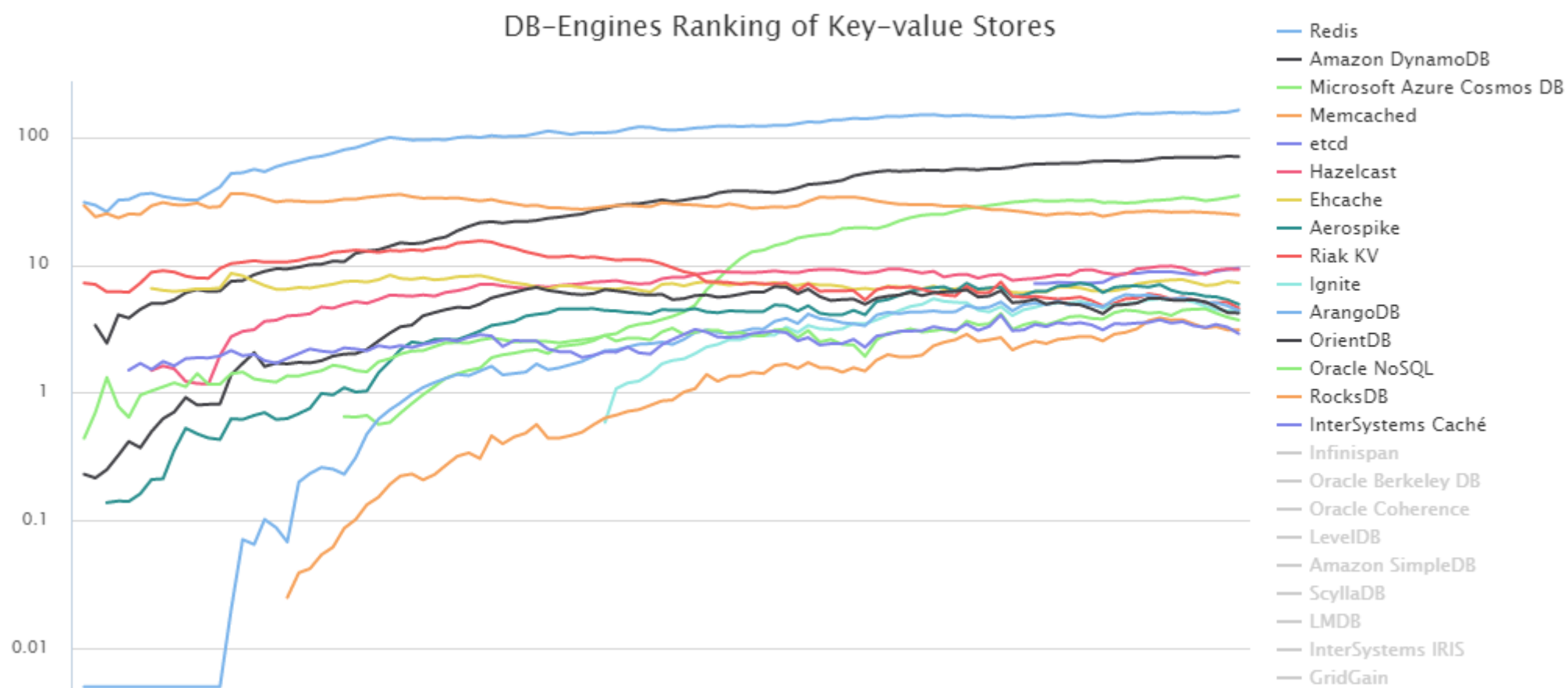
splice
MACHINE

TRANS/ATTICE



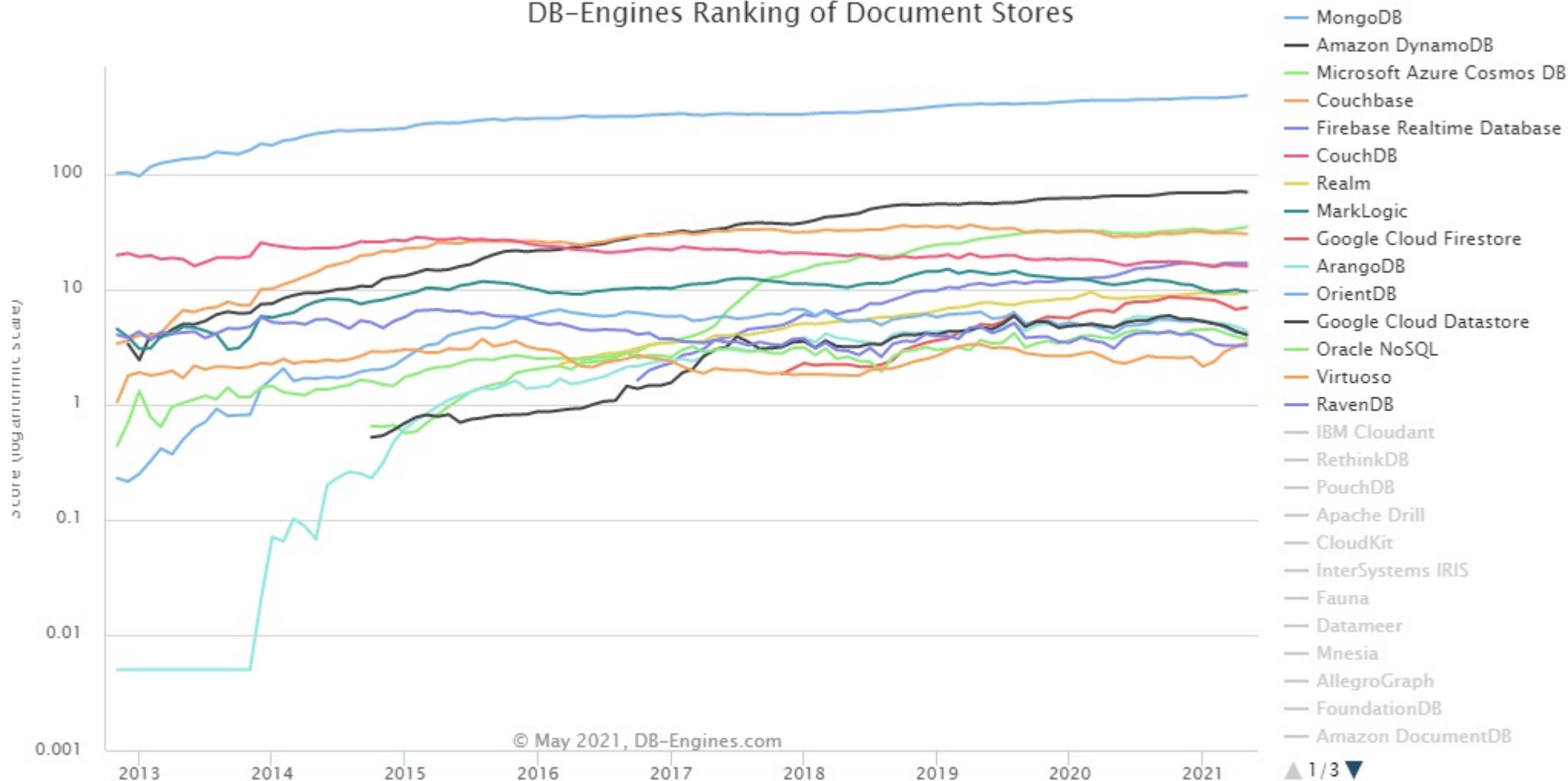
VOLTDDB

FERRAMENTAS



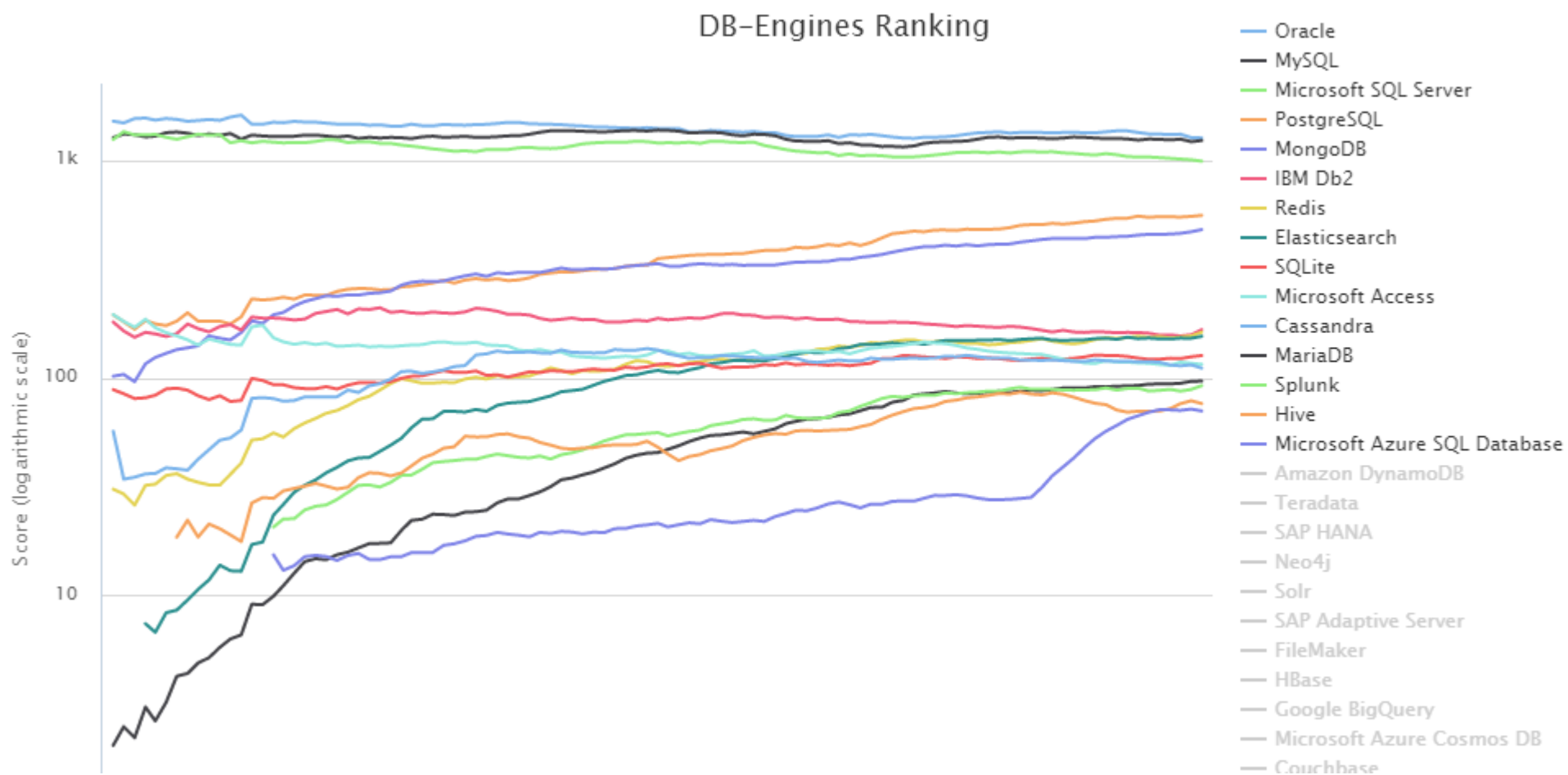
FERRAMENTAS

DB-Engines Ranking of Document Stores



FERRAMENTAS

May 2021



PREVISÕES

- *Até 2021, banco de dados relacionais continuarão sendo utilizados por pelo menos 70% dos novos aplicativos/projetos de DBMS.”*
- ***“Até 2022, os produtos de DBMS open sources serão responsáveis por mais de 25% da receita de DBMS.***
- *“Até 2023, 75% de todos os bancos de dados estarão em uma plataforma Cloud, o que mudará drasticamente o cenário dos fornecedores de DBMS*

RECURSOS

- Para NoSQL não temos muitos recursos para treinar
 - Playground Mongo
 - W3resource – exercícios mas sem editor.
 - Muito bom para aprofundar comandos noSQL
- Playground
 - bson multiple collections

OBRIGADO E ATÉ A PRÓXIMA AULA!