

Banco de Dados II



Bancos de Dados NoSQL

PROF. DR. THIAGO ELIAS

Relacional *versus* NoSQL (*Not Only SQL*)



- NoSQL (Not Only SQL) é uma espécie de “guarda-chuva” que engloba bancos de dados com vários paradigmas.
- Relacional e NoSQL não competem entre si. Foram criados para propósitos distintos.

Relacional *versus* NoSQL (*Not Only SQL*)



- Relacional:
 - Esquema rígido com características fixas.
 - Baseiam-se em tabelas (linhas e colunas).
 - Precisam de um esquema rígido do BD.
 - As tabelas se relacionam entre si (por isso no nome relacional)
 - Consistência dos dados. Sempre vai fornecer dados íntegros e consistentes.
 - Provê um excelente controle transacional. Isso garante a consistência dos dados.
 - Possui um custo computacional alto e problema de desempenho, principalmente, por conta das operações joins.
 - Problema de escalabilidade.

Relacional *versus* NoSQL (*Not Only SQL*)



- NoSQL:
 - Não são relacionais (podem possuir relacionamentos).
 - Não precisam acessar várias tabelas para acessar uma dada informação. Eles armazenam os dados de forma isolada (desnormalizada).
 - Possuem diferentes modelos/formas de armazenamento de dados (grafos, documentos, colunas, etc).
 - Possuem esquemas de dados flexíveis.
 - ✦ Ex; Num BD orientado a documentos, podem existir documentos com diferentes propriedades na mesma coleção.

Relacional *versus* NoSQL (*Not Only SQL*)



- NoSQL:
 - É possível armazenar um objeto num único registro.
 - ✦ Exemplo: o objeto pessoa (senha, dependentes, endereço...) pode ser armazenado num único registro. No relacional, seriam necessárias várias tabelas . As consultas seriam “caras” por conta das junções.
 - Indicado para BD's muito grandes, com muitas ações de escrita.
 - Escala horizontal: ao contrário dos BDs relacionais, o NoSQL é bem adaptado ao escalonamento horizontal, ou seja, aumentando a quantidade de computadores, a performance aumenta.
 - ✦ Exemplo: Em bancos orientados a documentos e/ou colunas, os dados armazenados são “independentes”.

Relacional *versus* NoSQL (*Not Only SQL*)



Rank			DBMS	Database Model	Score		
Sep 2024	Aug 2024	Sep 2023			Sep 2024	Aug 2024	Sep 2023
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1286.59	+28.11	+45.72
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1029.49	+2.63	-82.00
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	807.76	-7.41	-94.45
4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ	644.36	+6.97	+23.61
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	410.24	-10.74	-29.18
6.	6.	6.	Redis +	Key-value, Multi-model ⓘ	149.43	-3.28	-14.26
7.	7.	↑ 11.	Snowflake +	Relational	133.72	-2.25	+12.83
8.	8.	↓ 7.	Elasticsearch	Search engine, Multi-model ⓘ	128.79	-1.04	-10.20
9.	9.	↓ 8.	IBM Db2	Relational, Multi-model ⓘ	123.05	+0.04	-13.67
10.	10.	↓ 9.	SQLite +	Relational	103.35	-1.44	-25.85
11.	11.	↑ 12.	Apache Cassandra +	Wide column, Multi-model ⓘ	98.94	+1.94	-11.11
12.	12.	↓ 10.	Microsoft Access	Relational	93.76	-2.61	-34.81
13.	13.	↑ 14.	Splunk	Search engine	93.02	-3.08	+1.63
14.	↑ 15.	↑ 17.	Databricks +	Multi-model ⓘ	84.24	-0.22	+9.06
15.	↓ 14.	↓ 13.	MariaDB +	Relational, Multi-model ⓘ	83.44	-3.09	-17.01
16.	16.	↓ 15.	Microsoft Azure SQL Database	Relational, Multi-model ⓘ	72.95	-2.08	-9.78
17.	17.	↓ 16.	Amazon DynamoDB +	Multi-model ⓘ	70.06	+1.15	-10.85
18.	↑ 19.	18.	Apache Hive	Relational	53.07	-2.17	-18.76
19.	↓ 18.	↑ 20.	Google BigQuery +	Relational	52.67	-2.86	-3.80
20.	20.	↑ 21.	FileMaker	Relational	45.20	-1.47	-8.40
21.	21.	↑ 23.	Neo4j +	Graph	42.68	-1.22	-7.71
22.	↑ 23.	↓ 19.	Teradata	Relational, Multi-model ⓘ	41.47	-0.78	-18.86

Fonte: DB-Engines Ranking

Principais tipos de BDs NoSQL



- Orientados a colunas:
 - Talvez, os que mais se assemelham aos relacionais.
 - Porém, ao invés de armazenar os dados em linhas, estes são armazenados em colunas.
 - Colunas podem possuir sub-colunas e assim por diante.
 - ✦ Ex: Cassandra, Hypertable, MonetDB

Principais tipos de BDs NoSQL



- Bancos do tipo Chave-Valor:
 - Se assemelham à estrutura de dicionários ou tabelas Rash.
 - Todo dado armazenado possui uma chave que o identifica além do valor que ele contém.
 - ✦ Ex: Table Storage, DynamoDB e Redis.

Principais tipos de BDs NoSQL



- Bancos orientados a grafos:
 - Grafos são estrutura de dados compostos por vértices e arestas.
 - As arestas interligam os vértices.
 - Os vértices representam dados e as arestas representam os relacionamentos existentes entre os vértices.
 - ✦ Ex: Neo4j, Infinite Graph e ArangoDB

Principais tipos de BDs NoSQL



- Bancos orientados a documentos:
 - Armazenam os registros em forma de documentos (normalmente, json ou xml).
 - ✦ Ex: MongoDB, CouchDB e DocumentDB
- **OBS:** Alguns desses SGBDs anteriormente apresentados podem implementar mais de um modelo/paradigma.

Quando escolher NoSQL?



- Dificuldade para modelar BD's complexos de forma relacional.
 - ✦ Exemplo: redes sociais, onde existirão postagens, vídeos, curtidas, comentários de amigos, etc. Cada comentário pode conter likes, comentários ou outros vídeos, e assim por diante.
 - ✦ Essa modelagem seria possível num BD relacional. Porém, a montagem de uma “linha do tempo” com as postagens de um usuário teria um custo computacional muito alto devido aos joins de várias tabelas, sem falar da complexidade desse código. **Isso comprometeria o desempenho da aplicação.**
- Já usando um BD orientado a documentos, por exemplo, uma postagem poderia ser um documento onde as curtidas e comentários e etc seriam os atributos desse documento. Os documentos seriam auto-suficientes.

Quando escolher NoSQL?



- Necessidade de armazenamento de dados com esquemas flexíveis.
 - Estrutura dos dados mais intuitiva.
 - Baixa complexidade para mudanças.
 - Já numa tabela, todos os registros devem possuir a mesma estrutura. Caso precise alterar a propriedade de um registro, teríamos que alterar toda a tabela e todas as demais linhas passariam a ter aquela propriedade.
- Em NoSQL, podemos alterar a propriedade de um registro sem alterar as propriedades dos demais.