



CONCURSO PÚBLICO PARA PROFESSOR DE MAGISTÉRIO SUPERIOR

PERFIL DE ATUAÇÃO: Banco de Dados e Programação

ESPELHO DA PROVA

Ponto Sorteado: Ponto 10 – NoSQL: tipos de bancos de dados NoSQL (documento, chave-valor, grafo) e suas aplicações

1. Contextualização: Por que NoSQL?

- Motivação: Limitações do modelo relacional frente a grandes volumes, variedade e velocidade de dados (Big Data, Web 2.0).
- Conceito: “NoSQL” = Not Only SQL — categoria de sistemas que priorizam escalabilidade horizontal, flexibilidade de esquema e alta disponibilidade.
- CAP Theorem: (Consistency, Availability, Partition Tolerance) — trade-offs que influenciam o design dos bancos NoSQL.
- Evolução dos bancos de dados relacionais para os NoSQL. Exemplo: crescimento do volume de dados, Big Data, necessidade de escalabilidade e flexibilidade.

2. Características Gerais dos Bancos NoSQL

- Modelagem flexível, Escalabilidade horizontal, Desempenho, Consistência eventual;
- Consultas via APIs específicas (JSON, chave-valor, grafos).
- Comparação com bancos relacionais (SQL) — vantagens e limitações de cada um.

3. Tipos de Bancos de Dados NoSQL

- 3.1. Chave-Valor: armazena pares (chave, valor) — estrutura simples, similar a um dicionário, Exemplos: Redis, Amazon DynamoDB, Riak, Limitação: sem suporte nativo a consultas complexas ou relacionamentos. Aplicações: cache de dados, sessões de usuário, sistemas de recomendação em tempo real.
- 3.2. Documento: documentos estruturados (JSON, BSON, XML) armazenados em coleções, Exemplos: MongoDB, CouchDB, Firebase Firestore, Aplicações: sistemas web, catálogos de produtos, perfis de usuário, logs, Integração com programação: uso direto com linguagens orientadas a objetos e APIs REST.)
- 3.3. Grafos: nós (entidades) e arestas (relacionamentos) — dados modelados como grafo. Exemplos: Neo4j, OrientDB, Amazon Neptune, Aplicações: redes sociais, recomendação, detecção de fraudes, análise

de rotas, Diferencial: modelagem natural de relacionamentos e consultas em tempo real sobre conexões.

- 3.4. Bancos orientados a colunas - Armazenam dados em famílias de colunas (column families) em vez de linhas. Exemplos: Apache Cassandra, Google Bigtable, Apache HBase.

4. Critérios de Escolha e Integração

Critério técnico: tipo de dado, padrão de acesso, necessidade de relacionamentos.

5. Aplicações e Cenários Reais

- E-commerce: produtos (documento), carrinho de compras (chave-valor), recomendações (grafo).
- Redes Sociais: perfis e posts (documento), conexões (grafo), sessões (chave-valor).
- IoT / Log Analytics: armazenamento de dados não estruturados e consultas rápidas.

6. Tendências e Boas Práticas

- Modelagem orientada ao acesso — desenhar o esquema conforme as consultas.
- Uso combinado com SQL (NewSQL / multimodelo).
- Observabilidade e segurança: autenticação, auditoria, LGPD.
- Ferramentas atuais: Mongo Atlas, Neo4j Aura, Redis Stack, ArangoDB.

7. Desafios e limitações

- Dificuldade de padronização e consultas complexas.
- Curva de aprendizado para desenvolvedores.
- Integração com sistemas legados.

8. Conclusões

- NoSQL complementa, não substitui, o relacional.
- Escolha guiada pelo problema e pelo tipo de dado.
- Relevância crescente em cenários de Big Data, IA e aplicações.

NOME E ASSINATURA DA COORDENAÇÃO DAS BANCAS