

AS 12 PERGUNTAS DE BANCO DE DADOS QUE MAIS CAEM EM ENTREVISTA TÉCNICA.

Cada uma com a resposta que passa e o resumo em uma frase. Treine em voz alta antes da entrevista.

01 · ÍNDICES

O que é um índice e quando ele piora o desempenho?

Índice é uma estrutura (em geral uma B-tree) que acha linhas sem ler a tabela inteira. Acelera WHERE, JOIN e ORDER BY em colunas seletivas. Custa espaço e deixa INSERT, UPDATE e DELETE mais lentos, porque cada índice também precisa ser atualizado. Em coluna com poucos valores distintos ou tabela pequena, quase não ajuda.

Em uma frase: Índice acelera a leitura e cobra na escrita.

02 · ACID

O que significa ACID?

Atomicidade: a transação acontece inteira ou nada. Consistência: o banco sai de um estado válido para outro, respeitando as regras (constraints). Isolamento: transações concorrentes não enxergam o trabalho pela metade uma da outra. Durabilidade: depois do COMMIT, o dado sobrevive a queda (graças ao log de escrita, o WAL).

Em uma frase: Tudo ou nada, regras respeitadas, sem interferência, gravado de vez.

03 · NÍVEIS DE ISOLAMENTO

Quais são os níveis de isolamento e que problema cada um evita?

Read Uncommitted permite dirty read (ler dado não confirmado). Read Committed evita dirty read, mas permite non-repeatable read. Repeatable Read evita os dois, mas pode ter phantom read (linhas novas aparecendo). Serializable evita tudo, como se as transações rodassem em fila. Quanto mais isolamento, menos concorrência. PostgreSQL usa Read Committed por padrão; MySQL InnoDB usa Repeatable Read.

Em uma frase: Mais isolamento = menos anomalia e menos concorrência.

04 · PROBLEMA N+1

O que é o problema N+1 e como resolver?

Acontece quando o código busca uma lista (1 consulta) e depois faz mais uma consulta para cada item (N consultas). Comum em ORM com lazy loading. Resolva com JOIN, carregamento antecipado (eager loading: include, select_related, prefetch_related) ou um IN com todos os ids de uma vez.

Em uma frase: 1 consulta para a lista + 1 por item. Troque por JOIN ou eager loading.

05 · NORMALIZAÇÃO

O que é normalização e quando desnormalizar?

Normalizar é organizar as tabelas para evitar dado repetido e anomalias de atualização. 1FN: valores atômicos. 2FN: tudo depende da chave inteira. 3FN: nada depende de outra coluna que não seja a chave. Desnormalizar (duplicar de propósito) faz sentido em leitura muito pesada, relatórios e cache, aceitando o custo de manter as cópias em sincronia.

Em uma frase: Normalize para integridade. Desnormalize com motivo medido.

06 · TIPOS DE JOIN

Qual a diferença entre INNER, LEFT, RIGHT e FULL JOIN?

INNER JOIN traz só as linhas com correspondência nas duas tabelas. LEFT JOIN traz todas da esquerda e NULL onde a direita não tem par. RIGHT JOIN é o espelho. FULL OUTER JOIN traz tudo dos dois lados. CROSS JOIN faz o produto cartesiano. Pegadinha: filtro no WHERE sobre a tabela da direita transforma o LEFT JOIN em INNER.

Em uma frase: INNER = interseção. LEFT = tudo da esquerda + par quando existe.

07 · TRANSAÇÕES E DEADLOCK

O que é um deadlock e como evitar?

Deadlock é quando duas transações seguram um lock que a outra precisa e ficam esperando para sempre. O banco detecta e aborta uma delas. Para evitar: acesse as tabelas e linhas sempre na mesma ordem, mantenha as transações curtas, use índices para travar menos linhas e trate o erro com nova tentativa (retry).

Em uma frase: Mesma ordem de acesso, transação curta e retry.

Quando usar SQL e quando usar NoSQL?

SQL: dados relacionais, esquema definido, transações ACID e consultas com JOIN (pedidos, pagamentos). NoSQL: esquema flexível, escala horizontal fácil e um padrão de acesso conhecido (documentos, chave-valor, grafos, séries temporais). A resposta forte começa pelo padrão de acesso e pela consistência que o negócio exige, não pela moda.

Em uma frase: Escolha pelo padrão de acesso e pela consistência exigida.

Qual a diferença entre replicação e sharding?

Replicação copia os mesmos dados para várias máquinas: dá alta disponibilidade e escala a leitura (réplicas), com risco de atraso (replication lag). Sharding divide os dados entre máquinas por uma chave (ex.: user_id): escala a escrita e o volume, mas complica JOIN entre shards, transações e rebalanceamento. A escolha da chave de shard é a parte mais importante.

Em uma frase: Replicação copia. Sharding divide.

Como você descobre por que uma consulta está lenta?

Rodo EXPLAIN (ou EXPLAIN ANALYZE, que executa de verdade) para ver o plano: Seq Scan em tabela grande, estimativa de linhas muito diferente do real, ordenação em disco, JOIN em loop aninhado sobre muitas linhas. Depois ajusto: índice certo, reescrever a consulta, atualizar estatísticas (ANALYZE) e medir de novo.

Em uma frase: Não chute: leia o plano, corrija e meça de novo.

Qual a diferença entre chave primária e chave estrangeira?

Chave primária identifica cada linha de forma única: não repete e não aceita NULL, e uma tabela tem só uma. Chave estrangeira é uma coluna que aponta para a chave primária de outra tabela e garante integridade referencial (não existe pedido de cliente inexistente). Dá para definir o que acontece ao apagar o pai: CASCADE, RESTRICT ou SET NULL.

Em uma frase: PK identifica. FK garante a relação.

Em um índice composto, a ordem das colunas importa?

Importa. Um índice em (a, b) serve para filtros em a e em a + b, mas não para filtro só em b (regra do prefixo mais à esquerda). Coloque primeiro a coluna usada em igualdade e mais seletiva, depois a de intervalo ou ordenação. Um índice que cobre todas as colunas da consulta (covering index) evita até ler a tabela.

Em uma frase: Índice (a, b) não ajuda filtro só em b.

Na entrevista, não pare na definição. Diga o custo de cada escolha (o trade-off) e dê um exemplo real. É isso que separa júnior de sênior.

Versão online:

lucasalbuquerque.me/pt/banco