Question 15 of 15

What is the most effective way to handle 'read-your-writes' consistency issues with PostgreSQL read replicas?

1 Use higher isolation levels

2 Increase the number of read replicas

3 Always use synchronous replication

[✓ 4 Route critical reads to the primary database]

Correct!

Read-your-writes consistency issues occur when users don't see their own changes due to replication lag. The most effective solution is to route reads that need immediate consistency (like showing a user their just-posted content) to the primary database, while using replicas for other reads.

Question 1 of 15

PostgreSQL's GIN indexes can efficiently handle both full-text search queries and JSONB containment queries.

[✓ 1 True]

2 False

Correct!

GIN (Generalized Inverted Index) indexes in PostgreSQL work like the index at the back of a book, storing mappings of values to their locations. This makes them perfect for both full-text search (mapping words to documents) and JSONB queries (mapping JSON keys/values to rows).

Question 2 of 15

What is the primary bottleneck for write performance in a single PostgreSQL instance?

1 CPU processing power

2 Network bandwidth

3 Available RAM

[✓ 4 Disk I/O for WAL writes]

Correct!

PostgreSQL's Write-Ahead Log (WAL) must be written to disk before transactions can commit to ensure durability. Since WAL writes are sequential and synchronous, disk I/O becomes the primary bottleneck for write throughput, typically limiting a well-tuned instance to around 5,000 writes per second per core.

Question 3 of 15

PostgreSQL's default Read Committed isolation level prevents phantom reads.

1 True ✗

2 False ✓

Incorrect.

PostgreSQL's Read Committed isolation level only prevents dirty reads. It allows both non-repeatable reads and phantom reads. While PostgreSQL's Repeatable Read level prevents phantom reads (stronger than the SQL standard), Read Committed does not.

Question 4 of 15

Which PostgreSQL feature allows you to avoid needing a separate Elasticsearch cluster for many applications?

1 JSONB columns

2 PostGIS extension

3 Table partitioning

[✓ 4 Built-in full-text search with GIN indexes]

Correct!

PostgreSQL's built-in full-text search with GIN indexes supports word stemming, relevance ranking, multiple languages, and complex queries with AND/OR/NOT operations. This eliminates the need for Elasticsearch in many use cases, keeping the architecture simpler.

Question 5 of 15

What is the main advantage of PostgreSQL's covering indexes?

1 They can satisfy queries entirely from the index without table lookups ✓

2 They use less disk space

3 They work faster with write operations

4 They automatically update more frequently ✗

Incorrect.

Covering indexes include not just the columns being filtered on, but also the columns being selected using the INCLUDE clause. This allows PostgreSQL to return results entirely from the index without additional table lookups, significantly improving performance.

Question 6 of 15

PostgreSQL's JSONB data type allows you to get NoSQL-like flexibility while maintaining ACID guarantees.

[✓ 1 True]

2 False

Correct!

JSONB columns in PostgreSQL provide schema flexibility like NoSQL databases - you can store different attributes for different records. Unlike regular JSON columns, JSONB is stored in a binary format and can be efficiently indexed with GIN indexes, all while maintaining PostgreSQL's ACID properties.

Question 7 of 15

When should you consider using PostgreSQL's FOR UPDATE clause?

1 To enable full-text search

2 To improve write performance

3 To speed up read queries

[✓ 4 To ensure atomic operations across multiple related rows]

Correct!

FOR UPDATE provides row-level locking to ensure atomic operations. For example, in an auction system, you'd use FOR UPDATE when checking the current bid and placing a new one to prevent two users from simultaneously bidding based on the same outdated information.

Question 8 of 15

PostgreSQL's asynchronous replication provides better write performance than synchronous replication but may result in data loss during failover.

1 [✓ True]

2 False

Correct!

With asynchronous replication, the primary confirms writes immediately without waiting for replica acknowledgment, providing better performance. However, if the primary fails before changes are replicated, those changes can be lost. Synchronous replication waits for replica confirmation, ensuring no data loss but with higher latency.

Question 9 of 15

What type of spatial index does PostgreSQL's PostGIS extension use for efficient geospatial queries?

1 GIST indexes with R-tree implementation ✓

2 GIN indexes

3 Hash indexes

4 B-tree indexes

Correct!

PostGIS uses GIST (Generalized Search Tree) indexes, which implement R-tree indexing under the hood. R-trees are specifically designed for geometric data and can efficiently handle spatial queries like 'find all points within X kilometers' without checking every row.

Question 10 of 15

PostgreSQL's partial indexes can significantly improve performance and reduce storage when most queries only need a subset of data.

[✓ 1 True]

2 False

Correct!

Partial indexes only index rows that meet specific conditions (e.g., WHERE status = 'active'). This makes the index smaller, faster to maintain, and more efficient for queries that typically filter on the same conditions, especially when most of your data is inactive or deleted records.

Question 11 of 15

Which PostgreSQL scaling strategy involves distributing data across multiple PostgreSQL instances based on a chosen column?

1 Table partitioning ✗

2 Read replicas

3 Connection pooling

4 Horizontal sharding ✓

Incorrect.

Horizontal sharding distributes data across multiple PostgreSQL instances based on a sharding key (like user_id). Unlike PostgreSQL's built-in table partitioning (which splits data within a single instance), sharding requires external logic to route queries to the correct instance.

Question 12 of 15

PostgreSQL can efficiently combine full-text search, JSONB queries, and geospatial queries in a single SQL statement.

1 [✓ True]

2 False

Correct!

PostgreSQL's rich indexing capabilities allow combining multiple specialized queries efficiently. For example, you can find all video posts within 5km of San Francisco that mention 'food' and are tagged with 'restaurant' using GIN indexes for text search, JSONB containment, and GIST indexes for geospatial queries.

Question 13 of 15

What is one reason you would choose a different database over PostgreSQL for a system design?

1 When you need full-text search capabilities

2 When you need complex relationships between data

[✓ 3 When you need millions of writes per second globally]

4 When you need ACID compliance

✓ **Correct!**

PostgreSQL excels at complex relationships, ACID compliance, and full-text search. However, for extreme write throughput (millions of writes/second) or global multi-region active-active deployments, PostgreSQL's single-primary architecture and WAL bottlenecks make alternatives like Cassandra or CockroachDB more suitable.

Question 14 of 15

PostgreSQL's Repeatable Read isolation level provides stronger guarantees than the SQL standard requires, preventing phantom reads.

[✓ 1 True]

2 False

Correct!

PostgreSQL's Repeatable Read implementation is stronger than the SQL standard. While the SQL standard allows phantom reads at this level, PostgreSQL's implementation prevents them by creating a consistent snapshot that doesn't allow new rows to appear that match your query conditions.