

MySQL DBA (3+ Years Experience) - Detailed Interview Questions & Answers

1. Explain InnoDB Architecture in Detail.

InnoDB architecture consists of memory components and disk components. Memory Components:

- Buffer Pool: Caches data and index pages to reduce disk I/O. Ideally 60–70% of RAM.
- Change Buffer: Buffers changes to secondary indexes.
- Adaptive Hash Index: Speeds up frequent lookups.
- Log Buffer: Stores redo log entries before flushing to disk.

Disk Components:

- System Tablespace: Stores metadata and undo logs.
- Redo Logs: Ensure durability and crash recovery.
- Undo Logs: Used for rollback and MVCC.
- Doublewrite Buffer: Prevents partial page corruption. In production, monitoring buffer pool hit ratio and redo log flushing is critical.

2. Explain MySQL Replication Internals.

Replication works in stages:

- Primary Server: - Writes changes to Binary Log (binlog).
- Replica Server: - I/O Thread fetches binlog and stores it in relay log.
- SQL Thread reads relay log and executes events.

Types:

- Asynchronous
- Semi-synchronous
- Group Replication

Replication lag may occur due to large transactions, missing indexes, or slow disk I/O.

3. How Does MySQL Crash Recovery Work?

When MySQL crashes, InnoDB uses redo logs to reapply committed transactions and undo logs to rollback incomplete ones. Important variables:

- innodb_flush_log_at_trx_commit
- sync_binlog

These control durability and performance tradeoffs.

4. How Would You Tune a High Write Production System?

Steps:

- Increase innodb_buffer_pool_size.
- Optimize innodb_log_file_size.
- Remove unused indexes.
- Enable row-based replication.
- Batch inserts at application level.
- Monitor disk latency and fsync frequency.

5. Explain Index Strategy for Large Tables (200GB+).

- Use short numeric primary keys.
- Avoid over-indexing.
- Create covering indexes where possible.
- Partition by date for archival tables.
- Monitor index selectivity and cardinality.

6. How Do You Handle Replication Failover?

- Confirm primary failure.
- Identify best replica (least lag).
- STOP SLAVE; RESET SLAVE ALL;
- Promote replica and redirect application.
- Reconfigure other replicas. GTID simplifies failover.

7. Explain Isolation Levels with Practical Examples.

- Read Uncommitted: Dirty reads possible.
- Read Committed: No dirty reads.
- Repeatable Read (default): Prevents non-repeatable reads using MVCC.
- Serializable: Strictest, higher locking overhead.

8. What Happens During a Long Running Transaction?

- Locks held longer.
- Blocks other transactions.
- Increases undo log size.
- Causes replication lag.
- Affects purge process. Monitor using information_schema.innodb_trx.

9. Explain Binary Logging Formats.

- Statement-Based: Logs SQL statements.
- Row-Based: Logs actual row changes (recommended).
- Mixed: Combination of both. Row-based is safer for replication.

10. Explain MySQL High Availability Architecture.

Common HA setup: Primary with multiple replicas. Tools:

- Orchestrator - MHA - MySQL Group Replication
- ProxySQL HA layers include database layer, proxy layer, and monitoring layer.

11. How Would You Investigate CPU Spikes?

- Check SHOW PROCESSLIST.
- Analyze slow query log.
- Identify full table scans.
- Use EXPLAIN ANALYZE.
- Check missing indexes.

12. Describe a Production Backup Strategy.

- Weekly full physical backups.
- Daily incremental backups.
- Enable binary logging.
- Test restore procedures regularly.
- Follow 3-2-1 backup rule.

13. Difference Between Sharding and Partitioning.

Partitioning: - Same server. - Transparent to application. Sharding: - Multiple servers. - Application-level routing. Used for massive scale systems.

14. What Metrics Do You Monitor as DBA?

- CPU usage
- Disk I/O
- Buffer pool hit ratio
- Threads running
- Replication lag
- Slow queries
- Queries per second (QPS)

15. How Do You Reduce Lock Contention?

- Proper indexing.
- Keep transactions small.
- Use row-level locking.
- Maintain consistent update order.
- Avoid long-running transactions.