☑ Real-Time Scenario-Based SQL Server DBA Interview Q&A

---

1. Scenario: Database Performance Issue

Q: Users are complaining that the application is running very slowly. What steps would you take to identify and resolve the performance issue in SQL Server?

A:

- First, check system-level performance using tools like **Performance Monitor (PerfMon)** for CPU, memory, disk I/O.
- Use **Activity Monitor** or sp_who2 to identify blocking or high CPU sessions.
- Use **DMVs** like sys.dm_exec_requests, sys.dm_exec_query_stats, and sys.dm_exec_sql_text to find long-running queries.
- Check **wait statistics** using sys.dm_os_wait_stats.
- Look for **missing indexes**, **outdated statistics**, or **parameter sniffing** issues.
- Use **SQL Server Profiler** or **Extended Events** if needed to trace slow queries.
- Tune slow queries and add/rebuild indexes if necessary.

---

2. Scenario: Transaction Log Full

Q: The transaction log file has grown unexpectedly large and filled the disk. What could be the reason, and how do you resolve it?

A:

- Check the **recovery model**; if it's **FULL**, ensure regular transaction log backups are being taken.
- Run DBCC SQLPERF(LOGSPACE) to check log usage.
- Use sys.dm_tran_database_transactions to find open transactions.
- Backup the log if not done recently:
- BACKUP LOG [DBName] TO DISK = '...';
- Shrink the log file **only if necessary** using:
- DBCC SHRINKFILE('LogicalLogFileName', TargetSizeInMB);
- Investigate and fix the root cause (e.g., long-running transactions, uncommitted work).

---

3. Scenario: Database Backup Failed

Q: You receive an alert that a scheduled full backup failed last night. What steps would you take?

A:

- Check the **SQL Agent Job history** and **error logs** for failure reason.
- Verify the **disk space** on the backup drive.
- Confirm the **backup path** is accessible and the SQL Server service account has permissions.
- If the backup failed due to a transient issue, manually run it and monitor.
- Set up **alerts and notifications** for future failures.
- Review the **backup schedule** and consider adding retry logic.

---

4. Scenario: High CPU Usage

Q: CPU usage on the SQL Server is consistently over 90%. What could be causing this and how would you troubleshoot it?

A:

- Identify the queries using the most CPU:
- SELECT TOP 10 total_worker_time, execution_count, text
- FROM sys.dm_exec_query_stats
- CROSS APPLY sys.dm_exec_sql_text(sql_handle)
- ORDER BY total_worker_time DESC;
- Look for **missing indexes**, **bad query plans**, or **CPU-heavy operations**.
- Use **Resource Governor** to control CPU for specific workloads if needed.
- Check for **auto-update stats**, parallelism (MAXDOP), or excessive compilation/recompilation.
- Check for **virus scans** or 3rd-party processes on the server.

---

5. Scenario: Database Corruption

Q: How would you detect and fix corruption in a SQL Server database?

A:

- Run DBCC CHECKDB('DatabaseName') to detect corruption.
- If corruption is found:
    - Try to **restore from backup** if available.
    - If backup isn't available, attempt to repair using:
    - DBCC CHECKDB('DatabaseName', REPAIR_ALLOW_DATA_LOSS);

(Note: This may result in data loss – last resort only)

- Isolate and export salvageable data.
- Consider moving to a new database and importing data.
- Investigate root cause (disk issues, memory problems, etc.)

---

6. Scenario: Migration to New Server

Q: Your company is moving to a new SQL Server. What steps would you take to migrate a production database safely?

A:

- Document all current **server configurations** (logins, jobs, linked servers).
- Use **Backup/Restore**, **Detach/Attach**, or **Generate Scripts** to move databases.
- Migrate and test **logins, SQL Agent jobs, and SSIS packages**.
- Use **DBATools PowerShell** module for easier migration.
- Validate performance and functionality in a **staging environment**.
- Cut over during a **maintenance window** with rollback plan.
- Monitor closely post-migration.

---

7. Scenario: Blocking and Deadlocks

Q: You notice queries are getting blocked or deadlocks are occurring frequently. How do you identify and resolve them?

A:

- Use sp_who2, sys.dm_exec_requests, or sys.dm_tran_locks to identify blockers.
- Use **SQL Server Profiler** or **Extended Events** to capture **deadlock graphs**.
- Analyze the deadlock graph to identify the victim and culprit.
- Resolve by:
    - Optimizing queries to hold locks for shorter time.
    - Adding proper indexes.
    - Implementing **retry logic** or **deadlock priorities**.
    - Using **READ COMMITTED SNAPSHOT ISOLATION** if appropriate.

---

8. Scenario: Security Breach or Unauthorized Access

Q: You suspect someone has accessed the database without authorization. How do you investigate?

A:

- Check **SQL Server logs** and **Windows Event Viewer**.
- Enable and review **Login Auditing** (both failed and successful).
- Query sys.server_principals, sys.database_principals, and sys.server_permissions.
- Review **Default Trace** for login creation or permission changes.
- If enabled, check **SQL Audit** logs.
- Disable suspicious logins and force password reset.

---

9. Scenario: TempDB Full

Q: TempDB is filling up quickly and causing failures. What could be the cause and how do you handle it?

A:

- Check queries using tempdb heavily (large sorts, hash joins, temp tables).
- Monitor size using:
- DBCC SQLPERF('logspace');

or sys.dm_db_file_space_usage.

- Restart SQL Server (recreates TempDB) as a quick fix.
- Increase TempDB file size and number of files (one per logical processor up to 8 is typical).
- Optimize queries and avoid unnecessary temp table usage.

---

10. Scenario: AlwaysOn Availability Group Issue

Q: One of the secondary replicas in your AlwaysOn Availability Group is not synchronizing. How do you fix this?

A:

- Check **synchronization state** in SSMS or via sys.dm_hadr_database_replica_states.
- Verify network connectivity and endpoints.
- Restart **SQL Server Agent** and **AlwaysOn health sessions**.
- Check if the secondary database is **suspended** and resume it if needed.
- ALTER DATABASE [DBName] SET HADR RESUME;
- Review Windows Failover Cluster logs.
- Ensure both servers have same SQL Server version/patch level.