

Q1. What do you about SQL Server file groups? How many secondary file groups can you create?

SQL Server has four file groups: primary, secondary/user-defined, memory-optimized, and filestream:

- Primary filegroup: The default filegroup that contains the primary data file and any secondary files not in other filegroups. It's automatically created when a new SQL database is created.
- Secondary/user-defined filegroup: Created by a user to manage database data. For example, you can create a secondary filegroup and table to keep frequently accessed tables on faster disks.
- Memory-optimized filegroup: Stores In-memory OLTP tables and table variables.
- Filestream: Another type of filegroup.

You can create any number of secondary files. For example, a simple database named Sales can have one primary file and a log file, while a more complex database named Orders can have one primary file and five secondary files.

Q2.What is auto growth and auto shrink properties in SQL Server? Under what use cases would you consider using auto shrink?

auto-growth is the process by which the database file expands when it runs out of space. Auto-shrink is a database option that allows the database to be automatically shrunk by a background task when more than 25% of its space is unused. The auto_shrink database property is set to FALSE by default.

Auto-shrink is not recommended because it can cause performance degradation in several ways, including: File-system and index fragmentation, Resource intensive, Not necessary if you manage your backups correctly, and Squandering resources in massive quantity.

However, auto-shrink can be a blessing in some situations. For example, you can disable it but still do the shrinks manually.

Q3.What do you know about SQL Server Recovery and what are the different kinds of recovery model in SQL Server 2014?

SQL Server recovery models determine how transactions are logged and how backups can be used to restore a database in case of failure. The three main recovery models in SQL Server 2014 are Simple, Full, and Bulk-Logged:

- Simple
Truncation can take place immediately upon the occurrence of a CHECKPOINT operation.
- Full
Bulk operations logging changes from minimal logging to full logging when switching to the full recovery model during a bulk operation.
- Bulk-Logged
Point-in-time restore is possible in certain cases. This model increases the performance of bulk operations due to minimal logging.

Q4. What do you know about buffer pool extension in SQL Server? How do you configure this?

The Buffer Pool Extension (BPE) feature in SQL Server 2014 allows you to extend the buffer pool size by using a fast external storage device, such as an SSD or a cloud storage service. This feature is only supported in 64 bit servers.

The BPE acts as a second-level cache, where less frequently used pages are moved from the buffer pool to the extension file. This allows the buffer pool to retain more “buffers”, avoiding an extra paging activity. The BPE significantly improves I/O throughput because of the lower latency and better random I/O performance of SSDs.

You can enable and disable the BPE without interfering in the instance's availability. No restart is required, and no application change is needed. The BPE deals with clean pages only, so there is no possibility of data loss.

To configure the BPE size, you can use `ALTER SERVER CONFIGURATION`. For example, if you have 64 GB of physical RAM, your Max Server Memory setting would be 54272 MB (53 GB), which means your BPE should be a maximum of 848 GB (53 * the ratio of 16) or less. 424 GB or 212 GB is recommended for optimal size.

Q5. How would you add constraints and relationships in SQL Server tables?

You can add constraints and relationships in SQL Server tables using the `CREATE TABLE` and `ALTER TABLE` commands:

- `CREATE TABLE`

Use this command to create constraints while creating a new table. For example, to create a primary key constraint on multiple attributes, you can list all the attributes in the parenthesis.

- **ALTER TABLE**

Use this command to create constraints while altering an existing table. For example, to add a foreign key constraint, you can use the ALTER TABLE T-SQL statement after the table creation.

To add relationships between tables in SQL Server, you can use SQL Server Management Studio:

1. Right-click the table that will be on the foreign-key side of the relationship and select Design.
2. From the Table Designer menu, select Relationships.
3. In the Foreign-key Relationships dialog box, select Add.
4. Select the relationship in the Selected Relationship list.

Q6. Can you describe in brief the SQL Server backup and restore process and different strategies that you would employ as discussed in the class?

The SQL Server backup and restore process is a way to protect your data from loss or corruption. It involves creating a copy of your database and storing it in a safe place. If your database is ever damaged or lost, you can restore it from the backup.

There are two main types of SQL Server backups: full backups and differential backups. A full backup copies the entire database, while a differential backup only copies the data that has changed since the last full backup.

You can also use transaction log backups to protect your data. A transaction log backup copies the log of all the changes that have been made to the database since the last backup. This can be useful for recovering from a specific point in time.

To restore a SQL Server database, you can use the SQL Server Management Studio (SSMS). SSMS is a graphical tool that allows you to manage your SQL Server databases.

To restore a database using SSMS, follow these steps:

1. Open SSMS and connect to the SQL Server instance that contains the database you want to restore.
2. In Object Explorer, right-click the Databases node and select Restore Database.

3. In the Restore Database dialog box, select the database you want to restore and the backup file you want to use.
4. Click OK to restore the database.

There are a few different strategies that you can use to back up and restore your SQL Server databases. One common strategy is to perform a full backup once a week and differential backups every day. This will ensure that you have a complete backup of your data and that you can recover from any point in time.

Another strategy is to use transaction log backups. This can be useful for recovering from a specific point in time, such as if you accidentally delete some data.

The best strategy for you will depend on your specific needs. You should consider the size of your database, the frequency of changes, and the level of protection you need.

Here are some additional tips for backing up and restoring your SQL Server databases:

- Use a variety of backup methods. This will help to protect your data from different types of failures.
- Store your backups in a safe place. This could be an offsite location or a cloud storage service.
- Test your backups regularly. This will ensure that you can restore your data successfully if you need to.
- Keep a copy of your backup and restore procedures in a safe place. This will help you to restore your data quickly and easily if you need to.



Q7. How would you perform a database restore and transaction logs restore?

To restore a database in SQL Server using SSMS, you can do the following:

1. Go to Object Explorer
2. Right-click the database
3. Select Restore Database
4. Select the database to restore

5. Choose the FULL backup
6. Select the differential sets to restore
7. Press OK

To restore a transaction log backup in SQL Server, you can do the following:

1. Right-click the database
2. Point to Tasks
3. Point to Restore
4. Click Transaction Log
5. Use the RESTORE LOG command

You can restore the entire transaction log, or to a specific point in time or transaction mark. The database must be in a restoring state, which means you must first restore a full backup and possibly a differential backup. You can do this using the NOCOVERY option.

A best practice is to restore all the log backups, then recover the database in a separate operation. After the recovery process completes, the database goes online, and no more transaction log backups can be applied to the database.

Q8. What is the difference between SQL Server authentication v/s. windows authentication?

SQL Server authentication and Windows authentication are both authentication modes for SQL Server. Windows authentication is the default mode and is more secure than SQL Server authentication.

Here are some differences between SQL Server authentication and Windows authentication:

- Authentication method

SQL Server authentication requires users to provide valid credentials to authenticate themselves to SQL Server. Windows authentication authenticates users through the Windows principal token in the OS.

- Password

SQL Server authentication requires users to create a new username and password that gets stored in SQL Server. Windows authentication uses a user's existing domain account ID and password.

- **Security**

Windows authentication uses Kerberos security protocol, provides password policy enforcement, supports account lockout, and supports password expiration.

- **Convenience**

When accessing SQL Server from the same computer it is installed on, you shouldn't be prompted to type in an username and password if you're using Windows Authentication.

Q9. What do you know about SQL Server audits and error logging using audit feature?

SQL Server Audit tracks and logs events that occur on the Database Engine. It can be used to audit and collect actions to monitor at the SQL Server level, at the database level, or at both levels.

SQL Server auditing can help detect potential threats and vulnerabilities by tracking and reviewing all activities happening on SQL servers. This includes server setting changes, who modified a value in a specific table in the database, and more.

Here are some features of SQL Server auditing:

- C2 auditing
- Common Criteria compliance
- Login auditing
- SQL Trace
- Extended Events
- Change data capture
- DML, DDL, and login triggers

You can define an audit using SQL Server Management Studio or Transact-SQL. The audit can have multiple targets, including: A file, The Windows Security event log, and The Windows Application event log.

Audit logs are written to Append Blobs in an Azure Blob Storage on your Azure subscription. Audit logs are in .xel format and can be opened with SQL Server Management Studio (SSMS).

By default, the error log files are stored in the following location: Program Files\Microsoft SQL Server\MSSQL<n>.\<instance name>\MSSQL\LOG\ERRORLOG, where <n> is an archived version number, and <instance name> is the name of the instance.

Q10. What do you know about SQL Server user roles?

Microsoft SQL Server provides roles to help database administrators manage permissions to structured data. Server-level roles, as their name implies, grant access server-wide, similar to groups in the Windows world. Each SQL database can also have its own unique permissions and roles.

Q11. What do you know about highly availability and how do you configure this in SQL Server?

1. Replication. Let's start with a feature which often comes up in the discussion: the replication feature within Microsoft SQL Server. ...
2. Log Shipping. Okay, let's get the obvious out of the way... ...
3. Mirroring. ...
4. Always On Failover Clustering. ...
5. Always On Availability Groups.