**SQL Server DBA Training**

\*--------------------------------------------------------------------------------------------------------\*

--Anjaiah Karnati

1. SQL Server Versions & Editions:
   1. Versions:
      1. SQL Server 2000 🡪 8.0
      2. SQL Server 2005 🡪 9.0
      3. SQL Server 2008 🡪 10.0
      4. SQL Server 2008 R2 🡪 10.50
      5. SQL Server 2012 🡪 11.0
      6. SQL Server 2014 🡪 12.0
      7. SQL Server 2016 🡪 13.0
      8. SQL Server 2017 🡪 14.0
      9. SQL Server 2019 🡪 15.0
   2. Operating Systems for SQL Server
      1. Windows Server 2000
      2. Windows Server 2003
      3. Windows Server 2008 R2
      4. Windows Server 2012
      5. Windows Server 2019
   3. Editions of SQL Server (https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-version-15?view=sql-server-ver15)
      1. Enterprise: it supports higher hardware, and which will be used for larger organization. It supports unlimited RAM and 524PB of database size and supports up to maximum processors.
      2. Standard: It is a medium level edition. It supports 128GB of RAM and 524PB of database size and supports up to Limited to lesser of 4 sockets or 24 cores.
      3. Developer Edition: It is mainly used for developing application and testing the application.
      4. Express: it is free edition. It supports up to 10GB of database size. SQL Server Agent disabled completely.
      5. Compact: It is used in mobiles and laptops.
      6. Web: It is a low-level production edition which will be used in smaller organization and the options supported smaller hardware.

**SQL Server Structure:**

SQL Server

↓

Database

↓

Tables

↓

Data

**Architecture of Database**

**Database Structure:**

DATA

MDF (Master Data File)

Containing data

LOG

LDF (Log Data File)

Contains Transactions

Database is a collection of data. Every database contains 2 files data and log file. Data file extension is .MDF and log file extension is .LDF. DATA file contains data and log file contains transactions.

We can add multiple data files are called secondary data file and extension is .NDF (New Data File).

**Requirements for installing SQL Server:**

1. .NET Framework X.X (for 2005 – 2.0, 2008 R2- 3.5 SP1, 2012 – 4.0 or 3.5 SP1)
2. Windows installer X.X (for 2005 – 3.5, 2008 R2 – 4.5, 2012 – 4.5)

**Services:**

1. **DB Services:** - These are only used for managing DB & Data
2. **Integration Services:** - used to integrate different data sources into a single data source
3. **Reporting Services:** - used for reporting purpose
4. **Analysis Services:** - used for analysing purpose

User databases

System databases

User defined databases

Default databases

(Master, Model, MSDB, Tempdb and Resource)

**Master:**  Master DB is the master of all other databases, which contains information of all other DB’s. it also contains logins information and without master DB SQL Services will not start.

**Model:** Model DB is role model or template to create the new DB. The new DB will have the same properties of model DB.

**MSDB:** MSDB (Microsoft DB) contains information about jobs, alerts, operators, and backup & restore information.

**TempDB:** TempDB is a temporary DB, which contains all temporary transactions. Whenever we restart the SQL Services new tempDB will be created.

**Resource DB:** Resource DB is introduced in 2005 SQL Server onwards. Resource DB contains all system objects. Resource DB is a hidden DB. It will not be visible from SQL Server Management Studio. Resource DB mainly used for upgradation of SQL Server.

**SQL Server Naming Convention:**

Server name or Machine name = xyz

Default instance = xyz (or) . (or) local

Named instance = xyz\SQL1

SQL Server

SQL Server Agent

Jobs, Alerts, Operators

**String Datatypes:**

* Char [N] - 1 byte
* Varchar [N] - 1 byte
* Text -16 byte
* Varchar [MAX]-16GB

**Moving the System Databases:**

* In two scenarios, we will prefer to move system DB’s to other location.
  + If any problem occurs in OS
  + If the disk size is not sufficient in current drive.

**Moving TempDB:**

1. Find the current location of data and log file using below query.

Use TempDB

Go

EXEC SP\_HELPFILE

Go

1. Run the below query in MASTER database to modify file location/path

Use MASTER

Go

ALTER DATABASE TempDB MODIFY FILE (NAME=TEMPDEV, FILENAME=’C:\DATA\TEMPDB.MDF’)

Go

ALTER DATABASE TempDB MODIFY FILE (NAME=TEMPLOG, FILENAME=’C:\DATA\TEMPLOG.LDF’)

Go

1. Restart the SQL Services
2. Verify the current location of the tempDB data and log file using step#1

**Moving Model and MSDB:**

**(**For MSDB, stop the agent before moving the DB files)

**For MSDB: SS are started. But the agent won’t start until the data and log files were placed in new path**

1. Find the current location (path) using

Use MSDB/Model

EXEC SP\_HELPFILE

Go

1. Run the below query in MASTER database to modify file location/path

Use MASTER

Go

ALTER DATABASE MSDB MODIFY FILE (NAME=msdb, FILENAME= ’C:\DATA\MSDB.MDF’)

Go

ALTER DATABASE MSDB MODIFY FILE (NAME=msdblog, FILENAME=’C:\DATA\msdblog.LDF’)

Go

1. Copy the files from old location to new location.
2. Restart the SQL Services
3. Verify the current location of the MSDB data and log file using step#1

**Moving Master DB:**

1. Find the current location (path) using

Use MSDB/Model

EXEC SP\_HELPFILE

Go

1. Run the below query in MASTER database to modify file location/path

Use MASTER

Go

ALTER DATABASE MSDB MODIFY FILE (NAME=msdb, FILENAME= ’C:\DATA\MSDB.MDF’)

Go

ALTER DATABASE MSDB MODIFY FILE (NAME=msdblog, FILENAME=’C:\DATA\msdblog.LDF’)

Go

1. Copy the files from current or old location to new location.
2. Go to configuration manager -> SQL Server -> Properties -> Advanced -> Startup Parameters, modify the startup parameters path with new paths.
3. Restart the SQL Services
4. Verify the current location of the MASTER data and log file using step#1

**Types of pages in SQL Server:**

**Data Page:** It holds the data entered by the user but not the datatype.

**Index:** It stores the index entries.

**Text/Image:** It stores LOB (Large Object Data) like text, ntext, varchar(max), nvarchar(max), varbinary(max), Image and XML data.

**GAM & SGAM:** (Global allocation Map & Shared Global Allocation Map): These are used for saving information related to the allocation of extents.

**PFS** (Page Free Space): Information related to the page allocation and unused space available on pages.

**IAM** (Index Allocation Map): Information pertaining to extents that are used by a table or index per allocation unit.

**BCM** (Bulk Changed Map): Keeps information about the extents changed in a BULK operation.

**DCM** (Differential Change Map): This is the information of extents that have modified since the last BACKUP DATABASE statement.

**Database Backup and Different types of Backups:**

***Full Backup:*** Full Backup will backup the entire database.

**Command**: BACKUP DATABASE [DATABASE\_NAME] TO DISK=’C:\BACKUPS\DBNAME.BAK’ WITH INIT, STATS=10, COMPRESSION, COPY\_ONLY

INIT 🡪overwrites the existing backup file.

Compression 🡪performs the backup compression

Stats 🡪 will provide status for every 10% completed.

Format 🡪 will format the backup file.

***Differential Backup:*** Differential Backup will backup the modified data from the last full backup to the current time.

**Command**: BACKUP DATABASE [DATABASE\_NAME] TO DISK=’C:\BACKUPS\DBNAME.DIFF’ WITH DIFFERENTIAL, STATS=10, COMPRESSION

***Log Backup:*** Log Backup will backup all the transactions which are available in log file.

**Command**: BACKUP LOG [DATABASE\_NAME] TO DISK=’C:\BACKUPS\DBNAME.TRN’ WITH, STATS=10, COMPRESSION

***File & File Group Backup:*** By using file & File group backup we can backup the entire database or we can backup single file group or we can backup multiple file groups.

**Command**: BACKUP DATABASE [DATABASE\_NAME] FILEGROUP=’PRIMARY’ TO DISK=’C:\BACKUPS\DBNAME.BAK’ WITH STATS=10

***Copy-only Backup:*** By using Copy-only backup we can backup full backup, log backup without breaking the LSN (log sequence number) value.

**Command:** BACKUP DATABASE [DATABASE\_NAME] TO DISK=’C:\BACKUPS\DBNAME.BAK’ WITH COPY\_ONLY, STATS=10, COMPRESSION

***Split Backup:*** By using split backup we can divide the backup file into multiple files with equal sizes.

BACKUP DATABASE [DATABASE\_NAME] TO DISK=’C:\BACKUPS\SPILT1\_DBNAME.BAK’,

DISK=’C:\BACKUPS\SPLIT2\_DBNAME.BAK’ WITH STATS=10, COMPRESSION

***Mirror Backup:***  By using mirror backup we can take the backup of database in multiple locations simultaneously.

**Command:** BACKUP DATABASE [DATABASE\_NAME] TO DISK=’C:\BACKUPS\DBNAME.BAK’, MIRROR TO DISK=’C:\BACKUPS\DBNAME.BAK’ WITH STATS=10, COMPRESSION

**Tail log Backup:** A tail log backup captures the log records that have not been backed up since the recent loast log backup to current **point in time** before restoring.

**Command**: BACKUP LOG [DATABASE\_NAME] TO DISK=’C:\BACKUPS\DBNAME.TRN’ WITH, NO\_TRUNCATE | CONTINUE\_AFTER\_ERROR | NORECOVERY, STATS=10, COMPRESSION

***NO\_TRUNCATE:*** will use it if DB is damaged, and you want to produce a backup of the log if possible.

***CONTINUE\_AFTER\_ERROR:*** will use it when we got an error while trying to take tail log backup

***NORECOVERY:*** will use it if DB is healthy and you want to set the database in restoring state with the backup command.

**Database Restore and Different Restoration Options:**

**RESTORE WITH RECOVERY:** if we selectthis option, the DB will recover completely and come online. It is not possible to restore the additional backups.

**Command:** RESTORE DATABASE [DBNAME] FROM DISK=’C:\BACKUPS\DBNAME.BAK’

WITH MOVE ‘db data file logical name’TO ’D:\SQLDATA\DBNAME.MDF’,

MOVE ‘db LOG file logical name’TO ’E:\SQLLOG\DBNAME\_LOG.LDF’, RECOVERY, STATS=10

**RESTORE WITH NORECOVERY:** if we selectthis option, the DB is in restoring state and It is possible to restore the additional backups.

**Command:** RESTORE DATABASE [DBNAME] FROM DISK=’C:\BACKUPS\DBNAME.BAK’

WITH MOVE ‘db data file logical name’TO ’D:\SQLDATA\DBNAME.MDF’,

MOVE ‘db LOG file logical name’TO ’E:\SQLLOG\DBNAME\_LOG.LDF’, NORECOVERY, STATS=10

**RESTORE WITH STANDBY:** if we selectthis option, the DB is in standby state, in which the database is available for limited read-only access. This option is equivalent to specifying WITH STANDBY in a RESTORE statement.

**Command:** RESTORE DATABASE [DBNAME] FROM DISK=’C:\BACKUPS\DBNAME.BAK’

WITH MOVE ‘db data file logical name’TO ’D:\SQLDATA\DBNAME.MDF’,

MOVE ‘db LOG file logical name’TO ’E:\SQLLOG\DBNAME\_LOG.LDF’, STANDBY, STATS=10

**Restore a full backup**

The RESTORE DATABASE option allows you to restore either a full, differential, file or file group backup.

RESTORE DATABASE AdventureWorks FROM DISK = 'C:\AdventureWorks.BAK'

GO

Restore a full backup allowing additional restores such as a differential or transaction log backup (NORECOVERY)

The NORECOVERY option leaves the database in a restoring state after the restore has completed. This allows you to restore additional files to get the database more current. By default this option is turned off.

RESTORE DATABASE AdventureWorks FROM DISK = 'C:\AdventureWorks.BAK' WITH NORECOVERY

GO

**Restore a differential backup**

To restore a differential backup, the options are exactly the same. The first thing that has to happen is to do a full restore using the NORECOVERY option. Then the differential can be restored.

RESTORE DATABASE AdventureWorks FROM DISK = 'C:\AdventureWorks.BAK' WITH NORECOVERY

GO

RESTORE DATABASE AdventureWorks FROM DISK = 'C:\AdventureWorks.BAK'

GO

**Restore a transaction log backup**

To restore a transaction log backup the database need to be in a restoring state.  This means that you would have to restore a full backup and possibly a differential backup as well.

RESTORE LOG AdventureWorks FROM DISK = 'C:\AdventureWorks.TRN'

GO

**Restore multiple transaction log files (NORECOVERY)**

The NORECOVERY option leaves the database in a restoring state after the restore has completed. This allows you to restore additional files to get the database more current.

RESTORE LOG AdventureWorks FROM DISK = 'C:\AdventureWorks\_1.TRN' WITH NORECOVERY

GO

RESTORE LOG AdventureWorks FROM DISK = 'C:\AdventureWorks\_2.TRN'

GO

**RESTORE HEADERONLY:**

The RESTORE HEADERONLY option allows you to see the backup header information for all backups for a particular backup file.

RESTORE HEADERONLY FROM DISK = 'C:\AdventureWorks.BAK'

GO

**RESTORE LABELONLY:**

The RESTORE LABELONLY option allows you to see the backup media information for the backup file.

RESTORE LABELONLY FROM DISK = 'C:\AdventureWorks.BAK'

GO

**RESTORE FILELISTONLY:**

The RESTORE FILELISTONLY option allows you to see a list of the files that were backed up.

RESTORE FILELISTONLY FROM DISK = 'C:\AdventureWorks.BAK'

**RESTORE VERIFYONLY:**

The RESTORE VERIFYONLY command checks the backup to ensure it is complete and the entire backup is readable (useful for the restore)

RESTORE VERIFYONLY FROM DISK = C:\AdventureWorks.BAK

GO

**Simple Scenario:**

**->** Full backup is happening on every Sunday at 9PM

-> Differential backup daily at 9PM

-> Log backup for every 30mins

***And Database corrupted on Friday at 10.45AM.***

***Restoring Process to recover the database:***

1. *Take the tail log backup.*
2. *Restore Sunday Full backup with NORECOVERY (+WNR).*
3. *Restore Thursday Differential backup with NORECOVERY (+WNR).*
4. *Restore all log backups from Thursday (after differential backup) 10PM to Friday 10.30AM.*
5. *Finally restore the TAIL log backup.*

**System Databases Restoration Process:**

***Restore for MODEL and MSDB database:***

1. Before restoring the MSDB OR MODEL, Stop the SQL Server Agent.
2. Take the backup of MSDB

BACKUP DATABASE MSDB TO DISK=’C:\BACKUPS\MSDB.BAK’

1. Verify exclusive access
2. Restore MSDB

RESTORE DATABASE MSDB FROM DISK=’C:\BACKUPS\MSDB.BAK’

1. Start SQL Server Agent

***Restore for MODEL and MSDB database:***

1. Backup the MASTER database.
2. Stop all the services except SQL Server
3. Keep the SQL Server in SINGLE USER MODE (add **-m** parameter in startup parameter option).
4. Open command prompt 🡪 SQLCMD or open query analyser and run below command
   1. In command prompt:

Connecting the server from command prompt:

Default Instance: run -> SQLCMD

Names Instance: SQLCMD /E /S ServerName\InstanceName

C:\users\username:SQLCMD

* RESTORE DATABASE MASTER FROM DISK=’C:\BACKUPS\MASTER.BAK’ WITH REPLACE; GO

**RECOVERY MODELS**

**Full recovery model**

Full recovery allows point-in-time (or point-of-failure) recovery, provided you have the transaction log backups at that point in time. Point-in-time recovery signifies the ability to recover a database to a specific time, restoring all committed transactions and rolling back all incomplete transactions.

Recovery to a point of failure or point in time involves

* Backing up the currently active transaction log
* Restoring the most recent full database backup without recovery
* Restoring the most recent differential database backup without recovery
* Restoring in sequence any transaction log backups without recovery
* Restoring the last transaction log backup with recovery

***Backups Possible:*** *Full, Differential, Log (F+D+L)*

**Bulk-logged recovery model**

Databases using the bulk-logged recovery model minimally log bulk operations to the Microsoft SQL Server transaction log. These operations include CREATE INDEX, SELECT … INTO, writetext, updatetext, and BULK INSERT. The transaction log does not record sufficient information to recover these changes if media failure occurs after a bulk operation. You can recover a database to the point of failure, but your data may not be consistent if it was changed by a bulk operation. The process of restoration is the same as that of full database recovery.

***Backups Possible: Full, Differential, Log (F+D+L)***

**Simple recovery model**

The simple recovery model does not use the transaction log for recovery. If you use the simple recovery model, you can only restore full database backups. The advantage to using the simple recovery model is that there is less transaction log management. This topic is covered more fully in [Simple backup and recovery](http://webhelp.esri.com/arcgisserver/9.3/java/geodatabases/simple_2049735039.htm).

***Backups Possible: Full, Differential, Log (F+D)***

If you are running SQL Server 2005 or later, you can use this script instead:

|  |
| --- |
| SELECT name, recovery\_model\_desc FROM master.sys.databases ORDER BY name |

This is how you can change a database's recovery model to the simple recovery model:

|  |
| --- |
| ALTER DATABASE AdventureWorks SET RECOVERY SIMPLE |

to the full recovery model:

|  |
| --- |
| ALTER DATABASE AdventureWorks SET RECOVERY FULL |

and to the bulk-logged recovery model:

|  |
| --- |
| ALTER DATABASE AdventureWorks SET RECOVERY BULK\_LOGGED |

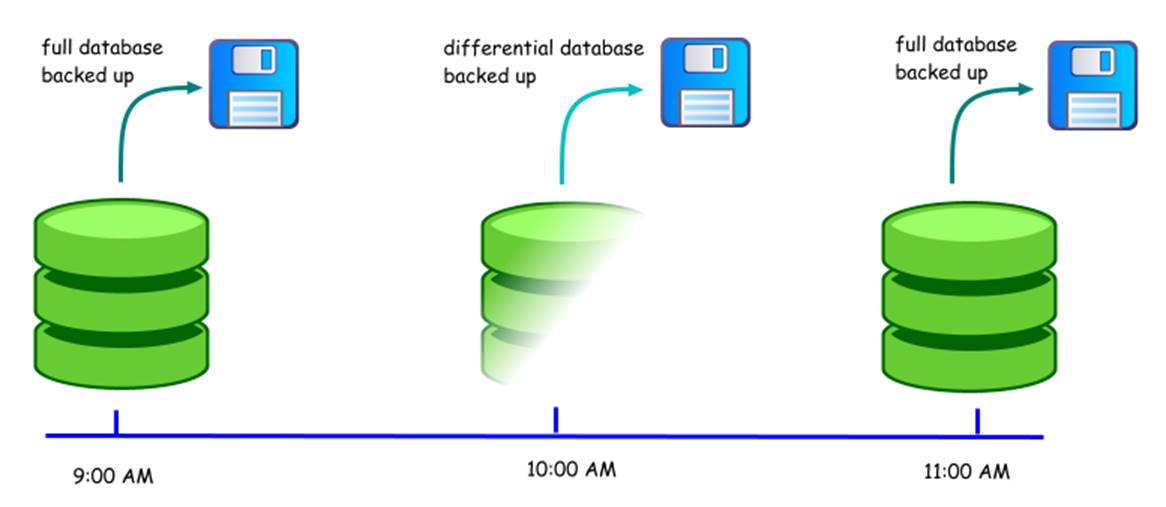
Which recovery model do you need to use?  It depends on how much data you can afford to lose.  Let's use the following diagram to illustrate the difference between the recovery models, where a full database backup is performed at 9:00 a.m, and 11 a.m.



***Recovery model scenarios:***

**The simple recovery model:**

Assume that there was a hardware failure at 10:45  a.m.  If the database was using the simple recovery model, you would have lost 105 minutes of work.  The latest point at which you can recover the database to is 9:00 a.m, since that was the time the last full backup that was made.  You could schedule differential backups to run periodically e.g.

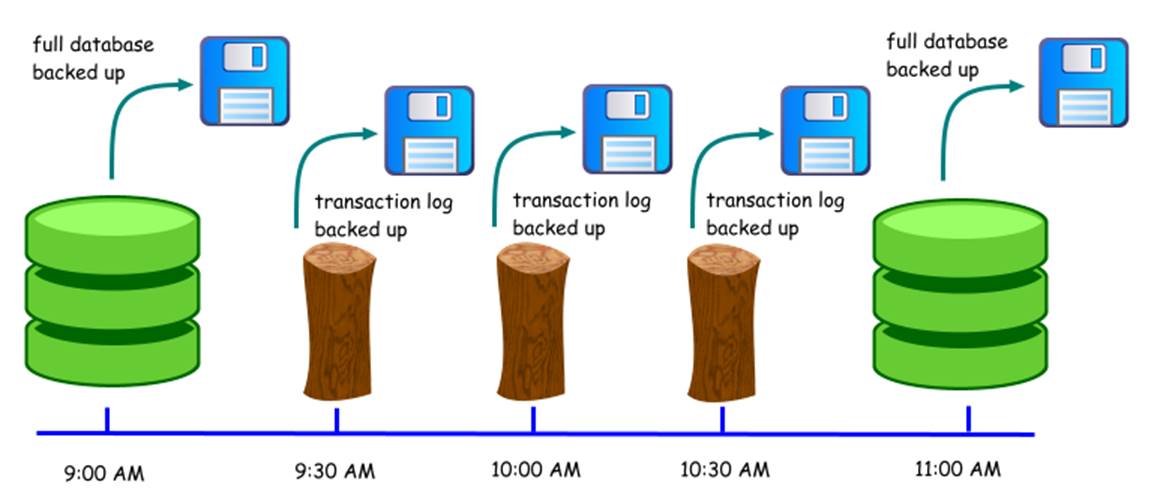


In this case, you would lose 45 minutes of work.  Now, assuming that a user deleted a critical table at 9:50 a.m.  Can you recover to the point in time just before the deletion?  No.  The differential backup contains only the changed data pages.  It cannot be used to restore to a specific point in time.  You'll have to restore the database to its 9 a.m state, and redo 49 minutes of work.  Then, you'll also need to redo the work that was performed after the deletion up to the time the error was discovered.

**The full recovery model:**

If no transaction log backups are made between 9 a.m and 11 a.m, you would face the same situation as you would if the database had been using the simple recovery model.  In addition, your transaction log file would be much larger, as SQL Server will not truncate committed and checkpointed transactions until they have been backed up.

Assume that the transaction log is backed up every 30 minutes:

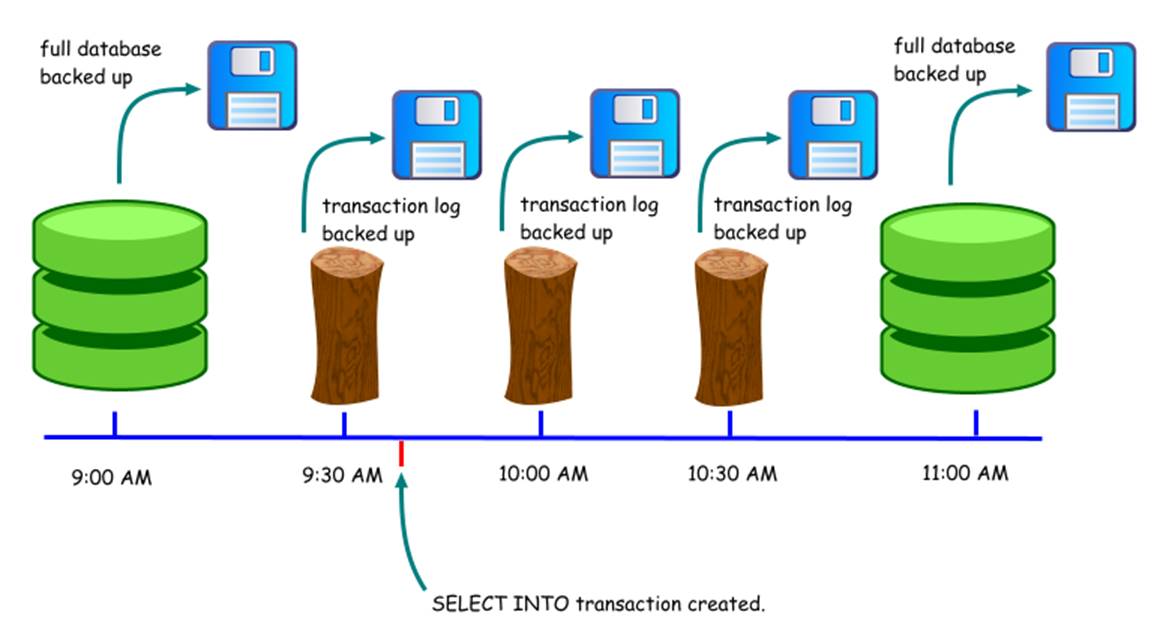


If a hardware failure occurs at 10:45 a.m, you would lose 15 minutes of work.  You can use the full database backup at 9 a.m, and apply the transaction logs to bring the database to its state at 10:30 a.m.  What if some critical data was deleted at 9:50 a.m?  Using the transaction log backup that was made at 10:00 a.m, you can restore the database to its state at 9:49 a.m.  You would still need to redo the work that was performed between the time of the deletion up to the time the error was discovered, as you cannot restore up to 9:49 a.m, skip the 9:50 a.m transactions, and restore the later transactions.  Admittedly, that would be great.

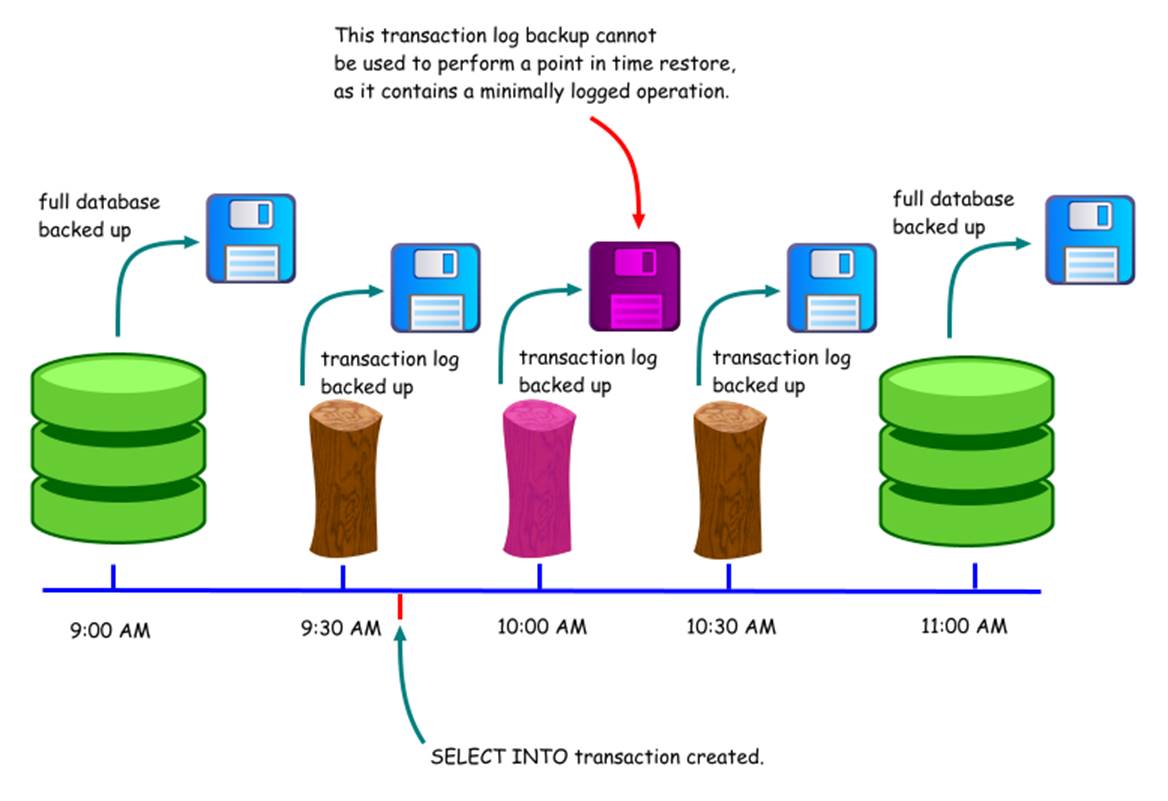
|  |
| --- |
| There are tools in the market that can simply reverse the changes created by user errors, while keeping the other transactions intact.  You still need transaction log backups to do that if the transaction is no longer in the active transaction log i.e. it has been backed up. |

**The bulk-logged recovery model:**

The situation with the bulk-logged recovery model is identical to the full recovery model if no minimally logged transactions are created in the database.  An example of a minimally logged transaction is a SELECT INTO operation.  Say that such a transaction occurred at 9:40 a.m.



This transaction would be minimally logged, which means that SQL Server only records the changed database pages arising from the transaction.  It does not log every insert into the new table.  Again, say that a critical table was deleted at 9:50 a.m.  What happens now?  You can no longer restore the database to its state at 9:49 a.m, as the transaction log backup created at 10:00 a.m cannot be used to restore to a specific point in time.  The best you can do is to restore the database to its state at 9:30 a.m.  Whenever a transaction log backup contains one or more minimally logged transactions, you cannot use that backup to perform a point in time restore.



Why then would anyone use the bulk-logged recovery model?  The main reason is performance.  Let's use the example of a SELECT INTO operation, say creating a large table from the results of a query.  If using the full recovery model, the details of every insertion into the new table is logged, and the transaction log will consume more space.  If using the bulk-logged recovery model, only details of the modified data pages are logged, allowing for better performance.  As we have seen, this comes at the expense of being able to perform a point in time restore using the transaction log.

The operations that are minimally logged are as follows (taken from SQL Server 2005 Books Online):

·Bulk import operations (bcp, INSERT ... SELECT \* FROM OPENROWSET(BULK...), and BULK INSERT).

·text, ntext, and image operations using the WRITETEXT and UPDATETEXT statements when inserting or appending new data. Note that minimal logging is not used when existing values are updated.

·SELECT INTO operations.

·Partial updates to large value data types using the UPDATE statement's .WRITE clauses when inserting or appending new data. Note that minimal logging is not used when existing values are updated.

·If the database recovery model is set to either bulk-logged or simple, some INDEX DDL operations are minimally logged whether the operation is executed offline or online.

·DROP INDEX new heap rebuild (if applicable).

|  |
| --- |
| When a database's data files are unavailable, perhaps due to a hardware failure, you can still back up the transaction log if the media it is on is still available.  However, you need to issue the **BACKUP LOG**command with the NO\_TRUNCATE option.  This will allow you to take a backup of the transaction log up to the point just before the hardware failure, and is usually referred to as backing up the tail of the transaction log.  However, if your database is using the bulk-logged recovery model, and the transaction log contains minimally logged transactions, the data files which contain the modified pages must also be available.  If those data files are unavailable, you will not be able to back up the tail of the transaction log.  This is another point to consider when using the bulk-logged recovery model. |

In summary, the simple recovery model provides the least number of options for recovery but is the simplest to manage.  The full recovery model allows for the most flexibility when it comes to restoring databases. The bulk-logged recovery model sacrifices some of those flexibility, in return for better performance for bulk operations.

|  |
| --- |
| If you have a new database that's running a full or bulk-logged recovery model, but have never made a full database backup of that database, the transaction log will be truncated every time a checkpoint occurs.  This behavior is similar to when the database is running the simple recovery model.  So if you have a database that's running the full recovery model and you find that the transaction log never grows even though you have not backed it up, you need to check if you have even made a full backup of that database yet. |

**T-SQL COMMANDS: TO AWARE**

DDL(Data Definition Language): DROP, CREATE, ALTER, TRUNCATE (D CAT)

DML(Data Manipulation Language): INSERT, DELETE, UPDATE, SELECT (I DUS)

DCL(Data Control Language): GRANT, REVOKE

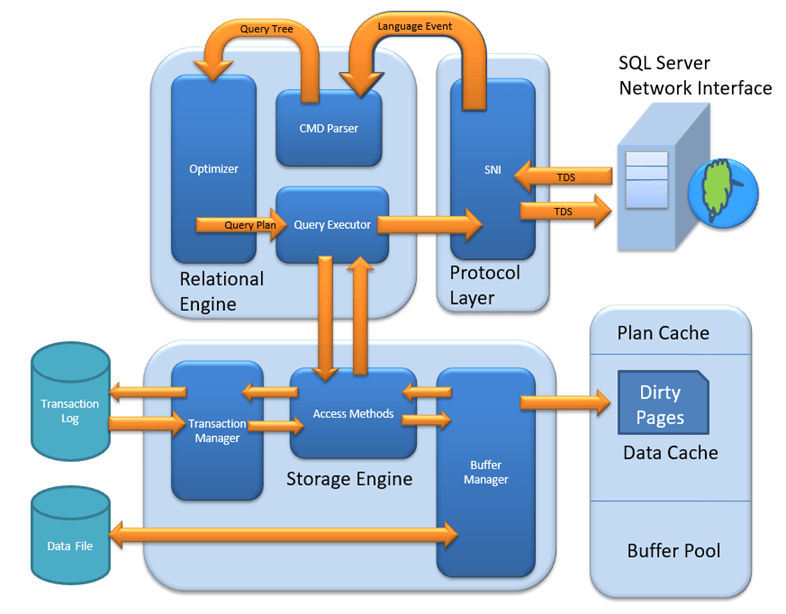
TCL(Transaction Control Language): ROLLBACK, COMMIT, SAVE TRAN

**SQL SERVER ARCHITECHTURE: IMPORTANT**

MS SQL Server is a client-server architecture. MS SQL Server process starts with the client application sending a request. The SQL Server accepts, processes and replies to the request with processed data. Let’s discuss in detail the entire architecture shown below:

As the below Diagram depicts there are three major components in SQL Server Architecture:

1. Protocol Layer
2. Relational Engine
3. Storage Engine

SQL Server Architecture Diagram

Let’s discuss in detail about all the three above major modules. In this tutorial, you will learn.

* [**Protocol Layer – SNI**](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#1)
* [Shared Memory](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#2)
* [TCP/IP](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#3)
* [Named Pipes](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#4)
* [What is TDS?](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#5)
* [**Relational Engine**](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#6)
* [CMD Parser](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#7)
* [Optimizer](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#8)
* [Query Executor](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#9)
* [**Storage Engine**](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#10)
* [File types](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#11)
* [Access Method](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#12)
* [Buffer Manager](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#13)
* [Plan Cache](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#14)
* [Data Parsing: Buffer cache & Data Storage](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#15)
* [Transaction Manager](https://www.guru99.com/sql-server-architecture.html?nowprocket=1#16)

**Protocol Layer – SNI**

MS SQL SERVER PROTOCOL LAYER supports 3 Type of Client Server Architecture. We will start with **“Three Type of Client Server Architecture”** which MS SQL Server supports.

**Shared Memory**

Let’s reconsider an early morning Conversation scenario.



MOM and TOM – Here Tom and his Mom, were at the same logical place, i.e. at their home. Tom was able to ask for Coffee and Mom was able it serve it hot.

**MS SQL SERVER –** Here **MS SQL** server provides **SHARED MEMORY PROTOCOL**. Here **CLIENT** and **MS SQL** server run on the same machine. Both can communicate via Shared Memory protocol.

**Analogy:** Lets map entities in the above two scenarios. We can easily map Tom to Client, Mom to SQL server, Home to Machine, and Verbal Communication to Shared Memory Protocol.

**From the desk of configuration and installation:**

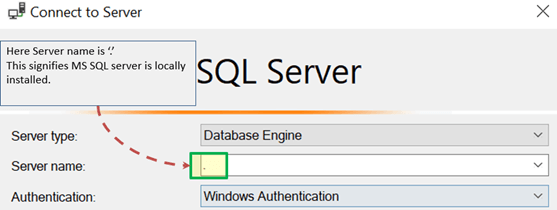
For Connection to Local DB – In SQL Management Studio, “Server Name” Option could be

“.”

“localhost”

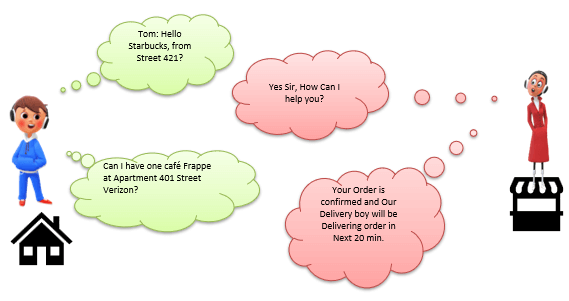
“127.0.0.1”

“Machine\Instance”



**TCP/IP**

Now consider in the evening, Tom is in the party mood. He wants a Coffee ordered from a well-known Coffee Shop. The Coffee shop is located 10 km away from his home.

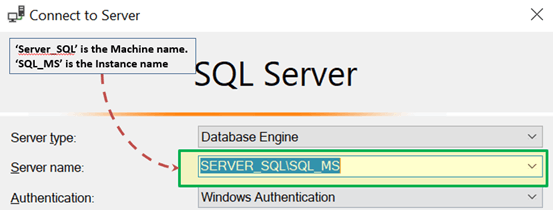


Here Tom and Starbuck are in different physical location. Tom at home and Starbucks at the busy marketplace. They’re communicating via Cellular network. Similarly, MS SQL SERVER provides the capability to interact via TCP/IP protocol, where CLIENT and MS SQL Server are remote to each other and installed on a separate machine.

**Analogy:** Lets map entities in the above two scenarios. We can easily map Tom to Client, Starbuck to SQL server, the Home/Market place to Remote location and finally Cellular network to TCP/IP protocol.

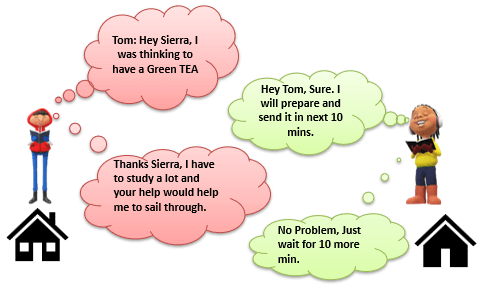
**Notes from the desk of Configuration/installation:**

* In SQL Management Studio – For Connection via TCP\IP, “Server Name” Option has to be “Machine\Instance of the server.”
* SQL server uses port 1433 in TCP/IP.



**Named Pipes**

Now finally at night, Tom wanted to have a light green tea which her neighbor, Sierra prepare very well.



Here **Tom** and his **Neighbor**, Sierra, are in same **physical** **location, being each other’s neighbor.** They’re communicating via **Intra network.** Similarly, **MS SQL SERVER** provides the capability to interact via the **Named Pipe** protocol. Here the CLIENT and **MS SQL SERVER** are in connection via **LAN**.

**Analogy:** Lets map entities in the above two scenarios. We can easily map Tom to Client, Sierra to SQL server, Neighbor to LAN and finally Intra network to Named Pipe Protocol.

**Notes from the desk of Configuration/installation:**

* **For Connection via Named Pipe.** This option is disabled by default and needs to be enabled by the SQL Configuration Manager.

**What is TDS?**

Now that we know that there are three types of Client-Server Architecture, lets us have a glance at TDS:

* TDS stands for Tabular Data Stream.
* All 3 protocols use TDS packets. TDS is encapsulated in Network packets. This enables data transfer from the client machine to the server machine.
* TDS was first developed by Sybase and is now Owned by Microsoft

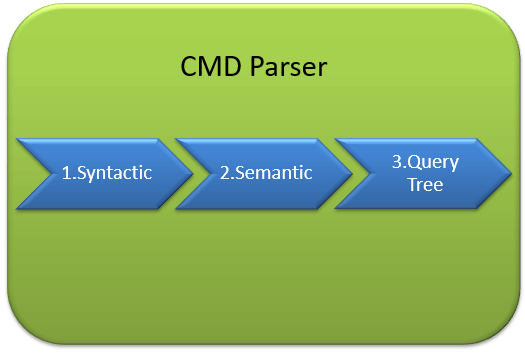
**Relational Engine**

The Relational Engine is also known as the Query Processor. It has the SQL Server components that determine what exactly a query needs to do and how it can be done best. It is responsible for the execution of user queries by requesting data from the storage engine and processing the results that are returned.

As depicted in the Architectural Diagram there are **3 major components** of the Relational Engine. Let’s study the components in detail:

**CMD Parser**

Data once received from Protocol Layer is then passed to Relational Engine. **“CMD Parser”** is the first component of Relational Engine to receive the Query data. The principal job of CMD Parser is to check the query for **Syntactic and Semantic error.** Finally, it **generates a Query Tree**. Let’s discuss in detail.



**Syntactic check:**

* Like every other Programming language, MS SQL also has the predefined set of Keywords. Also, SQL Server has its own grammar which SQL server understands.
* SELECT, INSERT, UPDATE, and many others belong to MS SQL predefined Keyword lists.
* CMD Parser does syntactic check. If users’ input does not follow these language syntax or grammar rules, it **returns an error.**

**Example:** Let’s say a Russian went to a Japanese restaurant. He orders fast food in the Russian language. Unfortunately, the waiter only understands Japanese. What would be the most obvious result?

The Answer is – the waiter is unable to process the order further.

There should not be any deviation in Grammar or language which SQL server accepts. If there are, SQL server cannot process it and hence will return an error message.

We will learn about MS SQL query more in upcoming tutorials. Yet, consider below most basic Query Syntax as

SELECT \* from <TABLE\_NAME>;

Now, to get the perception of what syntactic does, say if the user runs the basic query as below:

SELECR \* from <TABLE\_NAME>

Note that instead of ‘SELECT’ user typed “SELECR.”

**Result:** THE CMD Parser will parse this statement and will throw the error message. As “SELECR” does not follow the predefined keyword name and grammar. Here CMD Parser was expecting “SELECT.”

**Semantic check:**

* This is performed by **Normalizer**.
* In its simplest form, it checks whether Column name, Table name being queried exist in Schema. And if it exists, bind it to Query. This is also known as **Binding**.
* Complexity increases when user queries contain VIEW. Normalizer performs the replacement with the internally stored view definition and much more.

Let’s understand this with help of below example –

SELECT \* from USER\_ID

**Result:** THE CMD Parser will parse this statement for Semantic check. The parser will throw an error message as Normalizer will not find the requested table (USER\_ID) as it does not exist.

**Create Query Tree:**

* This step generates different execution tree in which query can be run.
* Note that, all the different trees have the same desired output.

**Optimizer**

The work of the optimizer is to create an execution plan for the user’s query. This is the plan that will determine how the user query will be executed.

Note that not all queries are optimized. Optimization is done for DML (Data Modification Language) commands like SELECT, INSERT, DELETE, and UPDATE. Such queries are first marked then send to the optimizer. DDL commands like CREATE and ALTER are not optimized, but they are instead compiled into an internal form. The query cost is calculated based on factors like CPU usage, Memory usage, and Input/ Output needs.

Optimizer’s role is to find the **cheapest, not the best, cost-effective execution plan.**

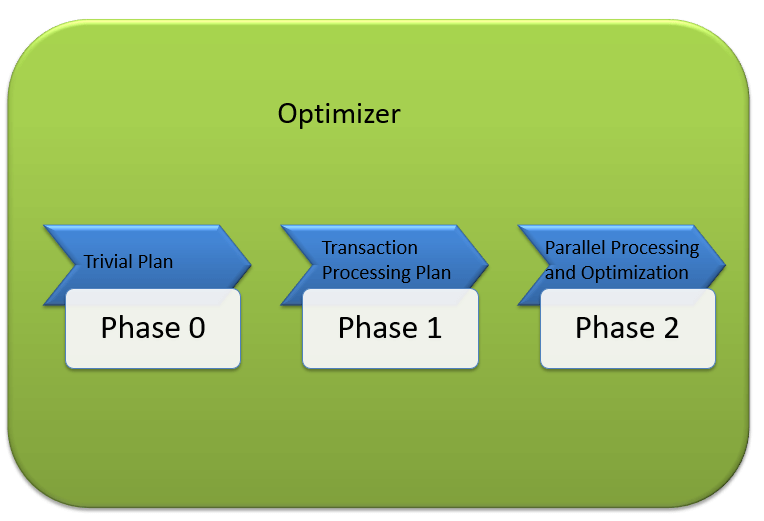
Before we Jump into more technical detail of Optimizer consider below real-life example:

**Example:**

Let’s say, you want to open an online Bank account. You already know about one Bank which takes a maximum of 2 Days to open an account. But, you also have a list of 20 other banks, which may or may not take less than 2 days. You can start engaging with these banks to determine which banks take less than 2 days. Now, you may not find a bank which takes less than 2 Days, and there is additional time lost due to the search activity itself. It would have been better to open an account with the first bank itself.

**Conclusion:** It’s is more important to select wisely. To be precise, choose which **option is best, not the cheapest.**

Similarly, MS **SQL Optimizer works on inbuilt exhaustive/heuristic algorithms. The goal is to minimize query run time.** All the Optimizer algorithms are **propriety of Microsoft and a secret.** Although**,** below are the high-level steps performed by MS SQL Optimizer. Searches of Optimization follows three phases as shown in the below diagram:



**Phase 0: Search for Trivial Plan:**

* This is also known as **Pre-optimization stage**.
* For some cases, there could be only one practical, workable plan, known as a trivial plan. There is no need for creating an optimized plan. The reason is, searching more would result in finding the same run time execution plan. That too with the extra cost of Searching for optimized Plan which was not required at all.
* If no Trivial plan found, then 1st Phase starts.

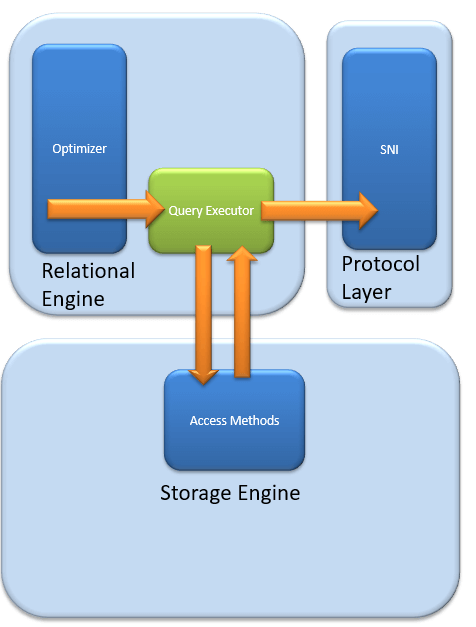
**Phase 1: Search for Transaction processing plans**

* This includes the search for **Simple and Complex Plan**.
* Simple Plan Search: Past Data of column and Index involved in Query, will be used for Statistical Analysis. This usually consists but not restricted to one Index Per table.
* Still, if the simple plan is not found, then more complex Plan is searched. It involves Multiple Index per table.

**Phase 2: Parallel Processing and Optimization.**

* If none of the above strategies work, Optimizer searches for Parallel Processing possibilities. This depends on the Machine’s processing capabilities and configuration.
* If that is still not possible, then the final optimization phase starts. Now, the final optimization aim is finding all other possible options for executing the query in the best way. Final optimization phase Algorithms are Microsoft Propriety.

**Query Executor**

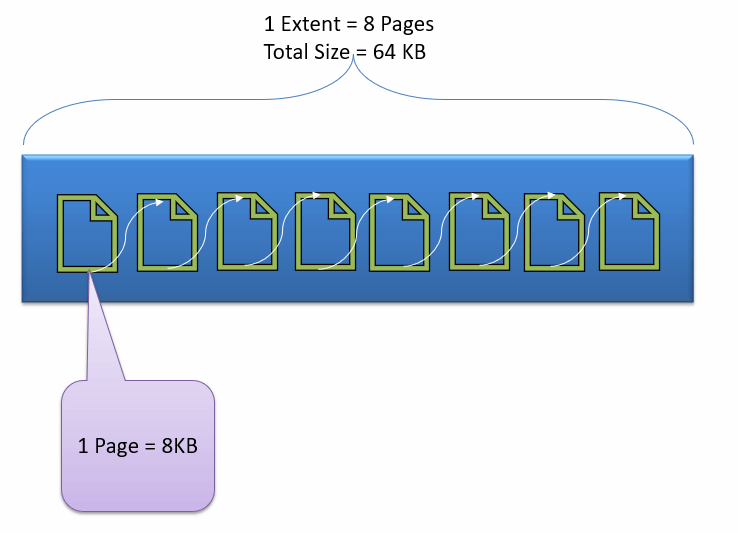


Query executer calls **Access Method.** It provides an execution plan for data fetching logic required for execution.Once data is received from Storage Engine, the result gets published to the Protocol layer. Finally, data is sent to the end user.

**Storage Engine**

The work of the Storage Engine is to store data in a storage system like Disk or SAN and retrieve the data when needed. Before we deep dive into Storage engine, let’s have a look at how data is stored in **Database** and **type of files available.**

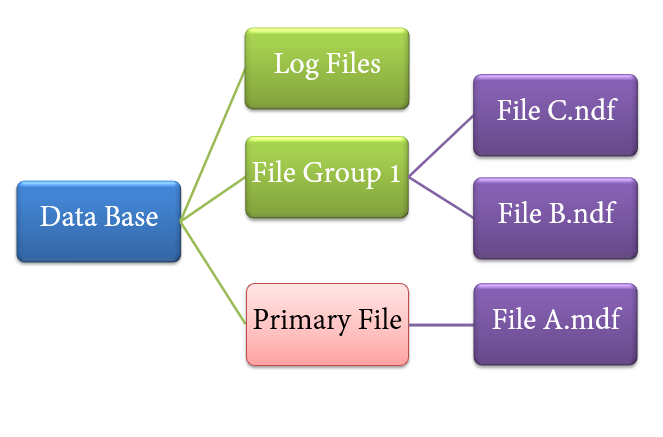
**Data File and Extent:**



Data File, physically stores data in the form of data pages, with each data page having a size of 8KB, forming the smallest storage unit in SQL Server. These data pages are logically grouped to form extents. No object is assigned a page in SQL Server.

The maintenance of the object is done via extents. The page has a section called the Page Header with a size of 96 bytes, carrying the metadata information about the page like the Page Type, Page Number, Size of Used Space, Size of Free Space, and Pointer to the next page and previous page, etc.

**File types**



1. **Primary file**

* Every database contains one Primary file.
* This store all important data related to tables, views, Triggers, etc.
* Extension is .**mdf** usually but can be of any extension.

1. **Secondary file**

* Database may or may not contains multiple Secondary files.
* This is optional and contain user-specific data.
* Extension is .**ndf** usually but can be of any extension.

1. **Log file**

* Also known as Write ahead logs.
* Extension is .**ldf**
* Used for Transaction Management.
* This is used to recover from any unwanted instances. Perform important task of Rollback to uncommitted transactions.

Storage Engine has 3 components; let’s look into them in detail.

**Access Method**

It acts as an interface between query executor and Buffer Manager/Transaction Logs.

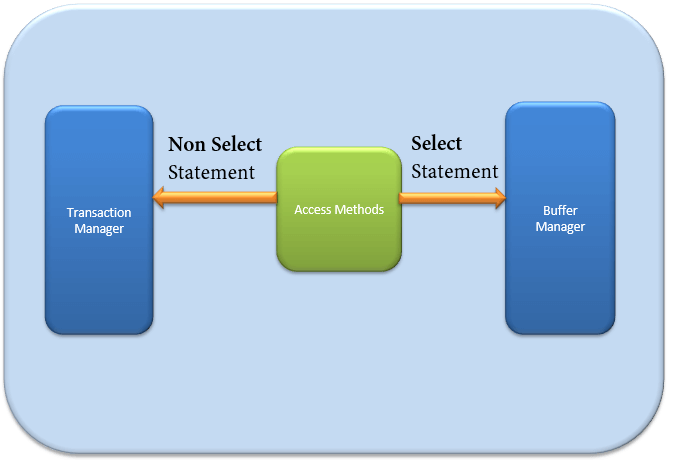
Access Method itself does not do any execution.

The first action is to determine whether the query is:

1. Select Statement (DDL)
2. Non- Select Statement (DDL & DML)

Depending upon the result, the Access Method takes the following steps:

1. If the query is **DDL**, SELECT statement, the query is pass to the **Buffer Manager** for further processing.
2. And if query if **DDL, NON-SELECT statement**, the query is pass to Transaction Manager. This mostly includes the UPDATE statement.

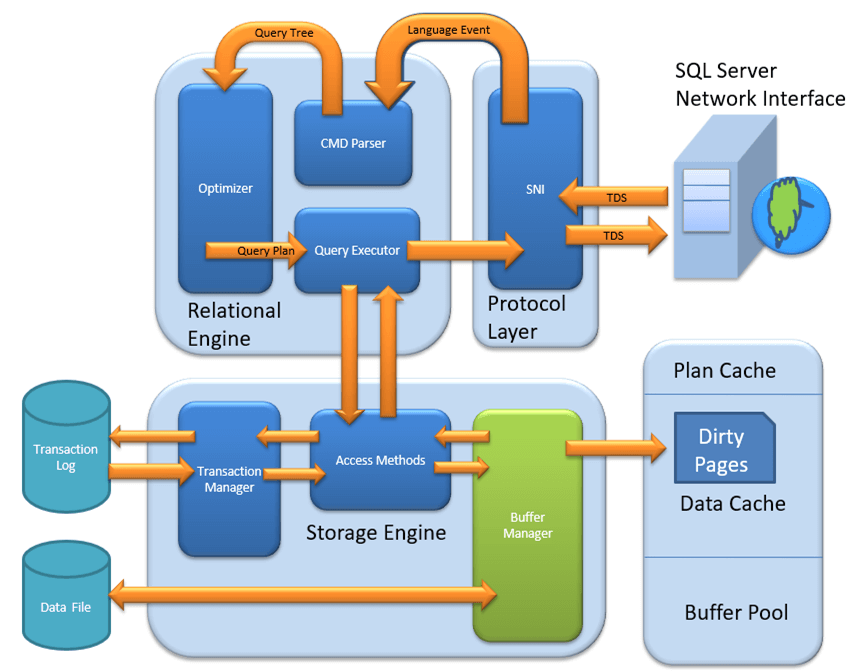


**Buffer Manager**

Buffer manager manages core functions for modules below:

* Plan Cache
* Data Parsing: Buffer cache & Data storage
* Dirty Page

We will learn Plan, Buffer and Data cache in this section. We will cover Dirty pages in the Transaction section.



**Plan Cache**

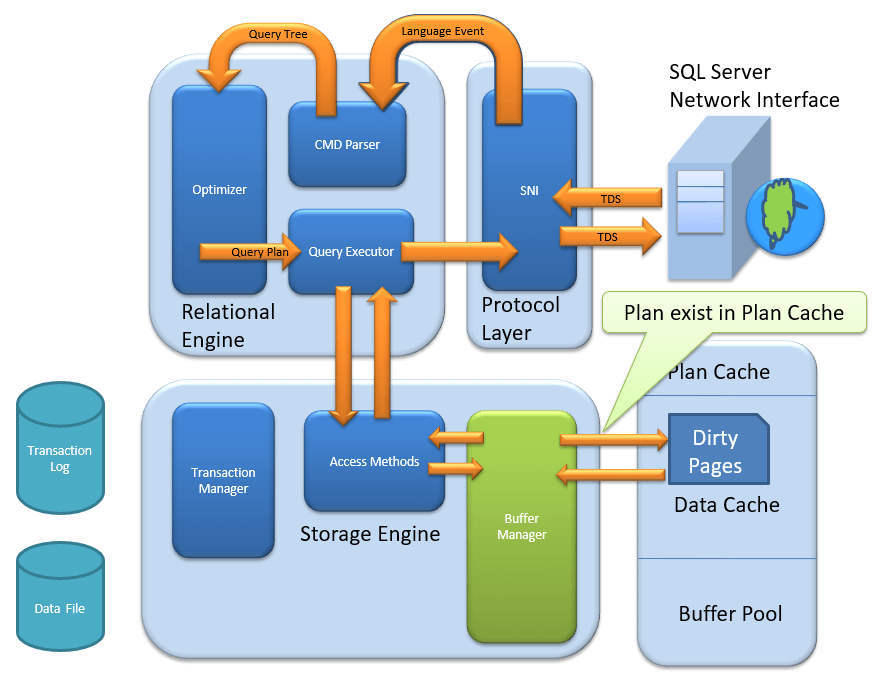
* **Existing Query plan:** The buffer manager checks if the execution plan is there in the stored Plan Cache. If Yes, then query plan cache and its associated data cache is used.
* **First time Cache plan:** Where does existing Plan cache come from?

If the first-time query execution plan is being run and is complex, it makes sense to store it in in the Plane cache. This will ensure faster availability when the next time SQL server gets the same query. So, it’s nothing else but the query itself which Plan execution is being stored if it is being run for the first time.

**Data Parsing: Buffer cache & Data Storage**

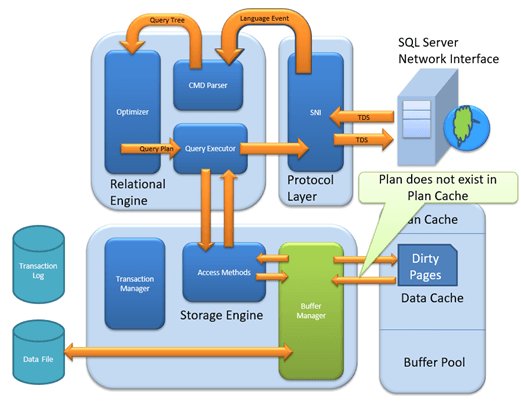
Buffer manager provides access to the data required. Below two approaches are possible depending upon whether data exist in the data cache or not:

**Buffer Cache – Soft Parsing:**



Buffer Manager looks for Data in Buffer in Data cache. If present, then this Data is used by Query Executor. This improves the performance as the number of I/O operation is reduced when fetching data from the cache as compared to fetching data from Data storage.

**Data Storage – Hard Parsing:**

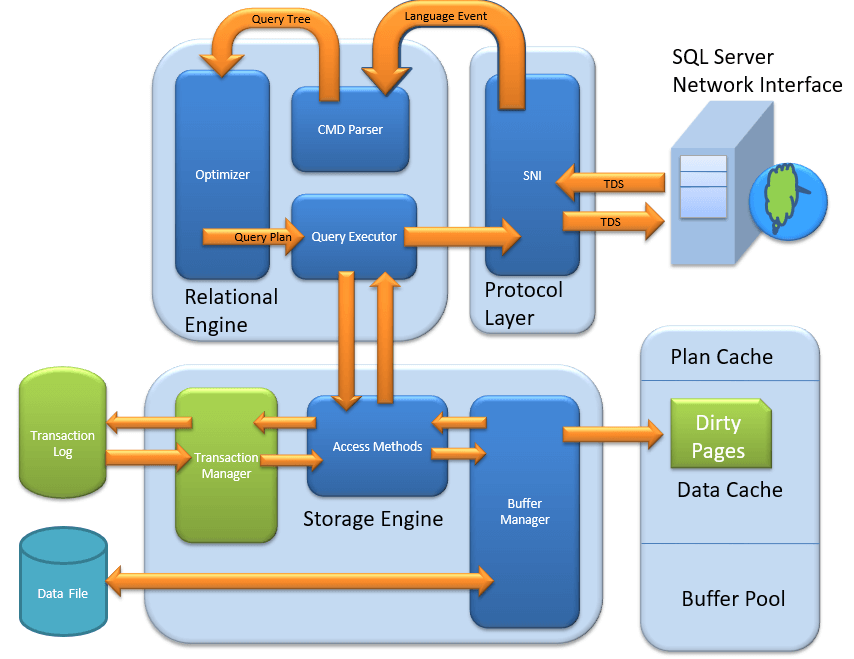


If data is not present in Buffer Manager than required Data is searched in Data Storage. It also stores data in the data cache for future use.

**Dirty Page**

It is stored as a processing logic of Transaction Manager. We will learn in detail in Transaction Manager section.

**Transaction Manager**



Transaction Manager is invoked when access method determines that Query is a Non-Select statement.

**Log Manager**

* Log Manager keeps a track of all updates done in the system via logs in Transaction Logs.
* Logs have **Logs Sequence Number with the Transaction ID and Data Modification Record**.
* This is used for keeping track of **Transaction Committed and Transaction Rollback**.

**Lock Manager**

1. During Transaction, the associated data in Data Storage is in the Lock state. This process is handled by Lock Manager.
2. This process ensures **data consistency and isolation**. Also known as ACID properties.

**Execution Process**

* Log Manager start logging and Lock Manager locks the associated data.
* Data’s copy is maintained in the Buffer cache.
* Copy of data supposed to be updated is maintained in Log buffer and all the events updates data in Data buffer.
* Pages which store the data is also known as **Dirty Pages**.
* **Checkpoint and Write-Ahead Logging:** This process run and mark all the page from Dirty Pages to Disk, but the page remains in the cache. Frequency is approximately 1 run per minute. But the page is first pushed to Data page of the log file from Buffer log. This is known as **Write Ahead Logging.**
* **Lazy Writer:** The Dirty page can remain in memory. When SQL server observes a huge load and Buffer memory is needed for a new transaction, it frees up Dirty Pages from the cache. It operates on **LRU** – Least recently used Algorithm for cleaning page from buffer pool to disk.

**Summary:**

* Three Type of Client Server Architecture exist: 1) Shared Memory 2) TCP/IP 3) Named Pipes
* TDS, developed by Sybase and now owned by Microsoft, is a packet which is encapsulated in Network packets for data transfer from the client machine to the server machine.
* Relational Engine contains three major components:

**CMD Parser:** This is responsible for Syntactic and Semantic error & finally generate a Query Tree.

**Optimizer:** Optimizer role is to find the cheapest, not the best, cost-effective execution plan.

**Query Executor:** Query executer calls Access Method and provides execution plan for data fetching logic required for execution.

* Three type of files exists Primary file, Secondary file, and Log files.
* Storage Engine: Has following important components

**Access Method:** This Component Determine whether the query is Select or Non-Select Statement. Invokes Buffer and Transfer Manager accordingly.

**Buffer Manager:** Buffer manager manages core functions for Plan Cache, Data Parsing & Dirty Page.

**Transaction Manager:** It manager Non-Select Transaction with help of Log and Lock Managers. Also, facilitates important implementation of Write Ahead logging and Lazy writers.

**PAGE:** Page is the default storage allocation of SQL Server. Page size is 8kb. Page contains Header & Footer. Header size is 96bytes. Header contains information about page type, free space in the page and information about the table page it’s belongs to.

**Footer** contains row offset pointing to every row.

**Extent** is collection of 8 pages of size 64kb. All the data allocations inside the SQL Server are in the form of extents only.

EXTENTS

MIXED EXTENT

UNIFORM EXTENT

**Uniform Extent:**  If all 8 pages are occupied by a single table then it is called as Uniform Extent.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | T1 | T1 | T1 | T1 | T1 | T1 | T1 |

**Mixed Extent:** If all 8 pages are occupied by different tables then it is called as Uniform Extent.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | T1 | T2 | T2 | T3 | T3 | T4 | T4 |

**HIGH AVAILABILITY TECHNOLOGIES:**

We have below HA’s:

* 1. Log Shipping
  2. Database Mirroring
  3. Replication
  4. Clustering
  5. AlwaysOn

**Log Shipping:**

RESTORING MODE

STANDBY MODE

LS BACKUP JOB

LS RESTORE JOB

LOG BACKUP

SHARED FOLDER1

SHARED FOLDER2

LS COPY JOB

🡪 Log shipping involves copying a database backup and subsequent transaction log backups from primary server and restoring database and transaction log backups on one more secondary (standby\restoring) server.

🡪 The target database is in a STANDBY or NORECOVERY mode on the secondary server(s) which allows subsequent transaction logs to be backed up on primary and shipped or copied to the secondary sever and then applied (restored) there.

**Permissions**: To setup a log shipping you must have SYSADMIN right on the server.

**Minimum Requirements:**

1. SQL Server 2005 or later. Standard, work group or enterprise editions must be installed on all server instances involved in log shipping.
2. The server involved in in log shipping should have the same CASE SENSITIVE settings.
3. The database must use the full recovery or bulk-logged recovery model.
4. A shared folder for copying T-Log backup files.
5. SQL Server agent must be configured properly.

**What is TUF file:** TUF stands for T-Transaction U-Undo F-File. So, it is transaction undo file.

**TUF:** when restoring transaction log with STANDBY option, SQL Server will go over the transaction log and save all uncommitted transactions to a file with TUF extension.

**Example:** Let us assume a rebuild index on a big table that has run for 45 mins and let’s say 3 log backups were taken in this duration. So, first log backup uncommitted transactions will be written to the TUF file while restoration and when the second log backup is being restored, the SQL Server will REDO all the uncommitted transactions from the TUF file and will continue the restoring the log backup but the second log backup also has uncommitted transactions. So, SQL Server will again UNDO the index operation and will save the uncommitted transactions to TUF file and this process continues until transaction is committed in further log backups causing the delay in log restore.

**Database Mirroring:**

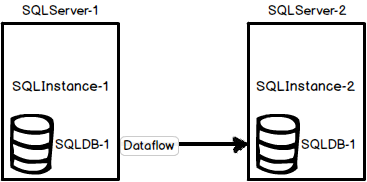
Database mirroring is a feature in SQL Server 2005, 2008 and later versions that provides a High Availability solution for Databases and Database Disaster Recovery. This feature can be enabled and used only on a database with FULL recovery models. The database can be mirrored from one SQL Server instance to another SQL Server instance. The source instance is called **Principal** server and the target instance is called **Mirrored** server. We could have one more server called **Witness** server.

**How does database mirroring work?**

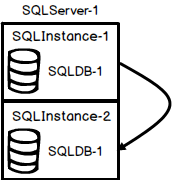
The Principal server sends the active transaction log record to the mirrored server. The mirrored server applies the transaction log record one by one in sequence.

**Implementation examples:**

One of the common mirroring configuration is the environment with two SQL Servers (SQLServer-1 and SQLServer-2), two instances (SQLInstance-1 and SQLInstance-2), and one mirrored database named SQLDB-1



The second common configuration is the environment with one SQL Server machine, two SQL Server instances, and one mirrored database named SQLDB-1. This solution has a major flaw because if SQLServer-1 goes down, both instances will be unavailable



**Operating modes:**

SQL Server database mirroring can be configured in two different modes, High-Safety mode also known as Synchronous mode and High-Performance mode also know as Asynchronous mode. The term Synchronous and Asynchronous says it all.

In **Synchronous** mode, the principal server sends the transaction to the mirrored server and waits until the transaction is committed on mirrored server then the transaction is committed on the principal server.

In Synchronous mode if you perform any transaction on the principal side. The transaction will not apply directly on principal. First it will apply on mirror server and send an acknowledge to the principal server then only the transaction applies on principal server.

In Synchronous mode always two databases are in SYNC.

In **Asynchronous** mode, the principal server sends the transaction to the mirrored server and does not wait for the transaction on mirrored to commit.

In Asynchronous mode first the transaction will be applied on the principal, and it will not wait for whether the transaction is applied on the mirror server. Principal server keep on accepting the transactions. If something happens to the principal server we will have data loss.

* **High Availability mode** – In this mode we have 3 servers 1) Principal Server 2) Mirror Server 3) Witness Server. Witness server is mandatory only in High Availability mode. The main advantage of High Availability mode is automatic failover.
* **High Protection mode:** – In this mode there won’t be any witness server and there won’t be any automatic failover.
* **High performance** – In this mode also will not have automatic failover. This mode available in Asynchronous mode.

***Prerequisites:***

1. For a mirroring session to be established, the partner and the witness if any must be running on the same version of SQL Server.
2. The two partners, that is the principal and mirror server must be running on the same edition of SQL Server. The witness if any can run on any edition of SQL Server that supports database mirroring.
3. The database must use the FULL recovery model. The SIMPLE and BULK-LOGGED recovery models do not support database mirroring.
4. Verify that the mirror server has sufficient disk space for the mirror database.
5. When you are creating the mirror database on the mirror server make sure that you restore the backup of the principal database specifying the same database name WITH NORECOVERY. Also, all log backups that were created after that backup was taken must also be applied again WITH NORECOVERY.

**Restrictions:**

1. Only user databases can be mirrored. You cannot mirror the system databases (Master, Model, MSDB, TempDB).
2. A mirrored database cannot be removed during a database mirroring session.
3. Database mirroring does not support FILESTREAM.
4. On a 32-bit system database mirroring can support a maximum of about 10 databases per server instance because of the number of worker threads that are consumed by each database mirroring session.

*Useful commands in database mirroring:*

*To drop database mirroring -> ALTER DATABASE <DB NAME> SET PARTNER OFF*

*To re-establish database mirroring -> ALTER DATABASE <DB NAME> SET PARTNER ON*

*To temporarily suspend database mirroring -> ALTER DATABASE <DB NAME> SET PARTNER SUSPEND*

*To resume suspended mirroring -> ALTER DATABASE <DB NAME> SET PARTNER RESUME*

*To perform failover -> ALTER DATABASE <DB NAME> SET PARTNER FAILOVER*

**Advantages and disadvantages of using SQL Server database mirroring:**

Using SQL Server database mirroring has multiple benefits: a built-in SQL Server feature, relatively easy to set up, can provide automatic failover in high safety mode, etc. Database mirroring can be combined with other disaster recovery options such as clustering, log shipping, and replication

[Database mirroring will be removed from SQL Server](http://technet.microsoft.com/en-us/library/ms189852.aspx) in future versions in favour of AlwaysOn Availability Groups. Also, database mirroring is per database only solution, which means that logins and jobs from the principal SQL Server must be manually recreated on the mirror. There is also possibility for delay, which can only be reduced with better hardware.

**Replication:**

Replication is a process of transferring the data from one server to another server. The main purpose of replication is for reporting purpose. We have different types of replications.

* + 1. Snapshot Replication
    2. Merge Replication
    3. Transactional Replication
    4. Peer to Peer Replication

**Snapshot Replication:** In snapshot replication, it will take complete data as a snapshot and applied to the subscriber side. Snapshot replication mainly used when we have less changes/updates on the publication side.

The main disadvantage of snapshot replication is we are not getting the continuous data from publisher to subscriber.

The agents available in snapshot replication are a) Snapshot Agent and b) Distributor Agent.

**Merge Replication:** It is bidirectional replication. Initially the publisher will transfer all the data to all subscribers if there are any modifications at the subscriber side the publisher will take these modifications and will transfer these changes to all subscribers.

In merge replication some subscribers may or may not connect to the publication so whenever they connected the data will be transferred.

In merge replication a) Snapshot Agent and b) Merge Agents are involved.

**Transactional Replication:** If you want to transfer the live data from publisher to subscriber you have to configure transactional replication.

In transactional replication, Snapshot Agent will take the initial data from publisher to subscriber then whatever the transactions we applied on the publisher database will be taking care by Log Reader Agent and will be applied to Distributor then the Distributor Agent will take care of all the transactions on Distribution and will be applied to the Subscriber. This is how the data transfer from Publisher to Subscriber continuously.

The agents involved in transactional replication are a) Snapshot Agent b) Log Reader Agent c) Distributor Agent.

Snapshot Agent

Log Reader Agent Distributor Agent

**Transactional Replication with Updatable Subscriber:** It is a bidirectional replication if we applied any transactions on Subscriber side. Queue Reader Agent will take care and apply to the distributor side.

Mainly four agents are involved a) Snapshot Agent b) Log Reader Agent c) Distributor Agent d) Queue Reader Agent.

**Peer to Peer Replication:** It is the extension of transactional replication. Here publisher will have multiple subscribers and all subscribers are now in connected state.

* We will configure the distributor once (it is a onetime process)

**Configure Replication using Backup option:**

1. Configure the publication normally and don’t select the initialization with snapshot option.
2. Go to publication properties -> subscription options -> allow subscription option using backup -> TRUE
3. Take the publication database backup and restore in subscriber.
4. After the database backup has been restored on the subscriber, on the publisher we used to add a subscription to this publication

Note the last three parameters in the command below. This is telling SQL Server how to initiate the subscriber:

@SYNC\_TYPE – using a backup

@BACKUPDEVICETYPE – where the backup was taken or disk

@BACKUPDEVICENAME – the name of the backup file name that was created.

**On publisher run the below command**

Use publication\_database

go

EXEC sp\_addsubscription

@publication = TSPMWebFiles\_Publisher,

@subscriber = 'AZPMSQL',

@destination\_db = TSPMWebFiles,

@sync\_type = 'initialize with /',

@backupdevicetype ='disk',

@backupdevicename = 'H:\SQLBackups\TSPMWebFiles\taillog1.trn'

**Performance:**

1. What are the things we are verifying in execution plan?

In execution plan we will verify whether query is using **table scan, index scan, index seek**, and **operator cost** and **CPU** and **IO**.

If query is using **table scan**, then we have to create the clustered index.

If query is using **index scan**, then we have to create non-clustered index.

If query is using **key-look-up**, then we have to create covering index by adding multiple columns to the existing non-clustered index.

1. What is the difference between Scan and Seek?

Scan mean identifying each row to find the required data. Seek means directly pointing the data.

1. How will improve query performance?

We will be verifying the execution plan and based on the plan we will fix the issue. If required, we will run the profiler to capture all the queries and we provide the profiler output to the input of a database tuning advisor (DTA). Where the DTA will analyse the query and will provide recommendations to improve query performance. It will provide the index and statistics.

1. What are Statistics?

Statistics will improve the selectivity of an index, if the statistics are not updated then the query will choose the wrong index which will hamper the performance. So, we have to update the statistics on regular basis. We can identify the statistics by using the command **SP\_UPDATESTATS**.

**FRAGMENTATION:**

Storing data non-uniformly inside the disk is called fragmentation.

Fragmentation

Internal Fragmentation

External Fragmentation

**Internal Fragmentation:** The data is not in uniform inside the page is called internal fragmentation.

**External Fragmentation:** The pages are inside the extent are not in uniform is called external fragmentation.

We can identify the fragmentation by using the following command: **DBCC SHOWCONTIG**

In the DBCC output we can verify the scan density and logical fragmentation. Always scan density closed to 100% and logical fragmentation closed to 0%.

In the DMV output we can verify the column average fragmentation percentage

Fragmentation

Fragmentation > 30% Fragmentation< 30%

Rebuild

Reorg

**What is the difference between Rebuild and Reorg?**

Rebuild operation will be chosen if the fragmentation is **more than 30%.** Rebuild operation will drop the current index and recreate the index again.

Reorg operation will be chosen if the fragmentation is **less than 30%.** Reorg operation will not drop any index. It will just rearrange the index internally.

**Fill Factor:** Fill factor is the amount of free space which we left in a page for the feature references.

Default fill factor is **Zero (0).**

*--> What is recommended fill factor?* **90**

**Heap:** if the table doesn’t have clustered index, then it is called as heap table.

**Explain Log Architecture:**

Internally the log file divided into multiple VLF’s (Virtual log files). The VLF will have different states.

Active: if the transaction is running the VLF is in active state.

Recoverable: transaction is completed but the log backup is not completed then the VLF is in recoverable state.

Reusable: if the transaction is completed and log backup is completed then the VLF state is in reusable state.

Unused: it is not allocated to any transaction still any VLF can use it.

**Security: Fixed Server Level Roles:**

**Sysadmin:**  Members of the sysadmin fixed server role can perform any activity in the server.

**Serveradmin:** Members of the serveradmin fixed server role can change server wide configuration options and shut down the server.

**Securityadmin:** Members of the securityadmin fixed server role manage logins and their properties. They can GRANT, DENY, and REVOKE server level permissions. They can also GRANT, DENY, and REVOKE database level permissions if they have access to a database. Additionally, they can reset passwords for SQL Server logins.

**Processadmin:** Members of the processadmin fixed server role can end processes that are running in an instance of SQL Server.

**Setupadmin:** Members of the setupadmin fixed server role can add and remove linked servers by using T-SQL statements. (Sysadmin membership is needed when using management studio).

**Bulkadmin:** Members of the bulkadmin fixed server role can run the BULK INSERT statements.

**Diskadmin:** The diskadmin fixed server role is used for managing disk files.

**DBCreator:** Members of the dbcreator fixed server role can create, alter, drop and restore any database.

**Public:** Every SQL Server login belongs to the public server role. When a server principal has not been granted or denied specific permissions on a securable object, the user inherits the permissions on any object when you want the object to be available to all users. You cannot change membership in public.

Server

Server

Logins

Database

**Agent roles in SQL Server:**

1. **SQLServerAgentRole:** Ability to manage jobs that they own
2. **SQLAgentReaderRole:** a). All of the SQLAgentUserRole rights. b). This ability to review multi server jobs, their configurations and history.
3. **SQLAgentOperatorRole:** a). All of the SQLAgentReaderRole rights. b). The ability to review operators, proxies, and alerts. c). Execute, stop, or start all local jobs. d). Delete the job history of any local job. e). Enable or disable all local jobs and schedules.

**Finding and fixing the Orphan users:**

**->**Logins available in SQL Server -> Select \* from master.dbo.syslogins

->Users available in Database -> Select \* from sysusers.

i. To list the orphaned user: **EXEC SP\_CHANGE\_USERS\_LOGIN ‘Report’**

ii. if you already have a login id and password for this user, fix it by doing:

EXEC SP\_CHANGE\_USERS\_LOGIN ‘AUTO\_FIX’,’USER’;

iii. if you want to create a new login id and password for this user, fix it by doing:

EXEC SP\_CHANGE\_USERS\_LOGIN ‘AUTO\_FIX’,’LOGINNAME’, NULL,’PASSWORD’;

**Orphan Users:**

**1. Server1**

**Database**

Test

Users

A, B, C

**A, B, C**

**2. Server2**

**Database**

Test01

Users

A, B, C(B and C are Orphan users as there is no login exist)

**A, X, Y**

**Locks:**

**Shared Lock:** When we are reading the data shared lock will be placed. Shared lock will be compatible with the other locks.

**Update Lock:** When we are performing the update operation, initially while reading the data shared lock will be placed, after identifying the data shared lock will converted to exclusive lock.

**Exclusive Lock:** When we are modifying the data exclusive lock will be placed.

**Intent Lock:** Intent locks are not the table level locks. Intent locks are range level locks means lock will not be placed on the entire table. It will place the lock on specific range and the remaining range will be accessible to the users.

**Lock:** When we are trying to access the data SQL Server will place a lock on table.

**Blocking: User1** is running a query on **table1** at the same time **User2** is running a query on different transaction on same table then query1 will block the query2.

**Deadlock:** U1 is having lock on T1 and U2 is having lock on T2. In order to complete the U1 query he requires a lock on T2 and in order to complete U2 query he requires a lock on T1. This is called deadlock.

When deadlock occurs SQL Server will continue one transaction and it will rollback another transaction. The transaction which is rolled back is called as deadlock victim. It will rollback the transaction which will take least resources to rollback.

**How will you capture the deadlocks?**

We can capture the deadlock by using trace flag 1204 and 1222.

1204 -> will provide the information of the queries in normal format, whereas 1222-> will provide deadlock information in XML format. Apart from that if you want to capture the deadlock deeply we have to run profiler using the command lock\_deadlockchain, lock\_deadlockgraph.

**AlwaysOn:**

1. An availability group supports a replicated environment for a discrete set of user databases, known as availability databases. You can create an availability group for high availability (HA) or for read scale. An HA availability group is a group of databases that fail over together. A read-scale availability group is a group of databases that are copied to other instances of SQL Server for read-only workload. An availability group supports one set of primary databases and one to eight sets of corresponding secondary databases. Secondary databases are not backups. Continue to back up your databases and their transaction logs on a regular basis.
2. You can create any type of backup of a primary database. Alternatively, you can create log backups and copy-only full backups of secondary databases. For more information, see [Active Secondaries: Backup on Secondary Replicas (Always On Availability Groups)](https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/active-secondaries-backup-on-secondary-replicas-always-on-availability-groups?view=sql-server-2017).
3. An availability replica hosts each set of availability database. Two types of availability replicas exist: a single primary replica. Which hosts the primary databases, and one to eight secondary replicas, each of which hosts a set of secondary databases and serves as a potential failover targets for the availability group. An availability group fails over at the level of an availability replica. An availability replica provides redundancy only at the database level-for the set of databases in one availability group. Failovers are not caused by database issues such as a database becoming suspect due to a loss of a data file or corruption of a transaction log.
4. The primary replica makes the primary databases available for read-write connections from clients. The primary replica sends transaction log records of each primary database to every secondary database. This process - known as data synchronization - occurs at the database level. Every secondary replica caches the transaction log records (hardens the log) and then applies them to its corresponding secondary database. Data synchronization occurs between the primary database and each connected secondary database, independently of the other databases. Therefore, a secondary database can be suspended or fail without affecting other secondary databases, and a primary database can be suspended or fail without affecting other primary databases.

**Click 🡪**

**Clustering:**

Clustering is a technology that automatically allows one physical server to take over the

tasks and responsibilities of another physical server that has failed. The goal is to ensure

those users running mission-critical applications will have very less downtime when a

failure occurred. A cluster is a group of independent computer systems, referred to as

nodes, working together as a unified computing resource. A cluster provides a single name

for clients to use and a single administrative interface, and it guarantees that data is

consistent across nodes.

**What is the difference between Active – Active and Active – Passive?**

In Active-Active (A-A) in two nodes two different SQL Servers are running.

In Active-Passive, only active node contains SQL Server and passive node is empty and passive node is waiting for the failover of active node.

Cluster

A-A A-P

SS1

SS1

SS2

Node1 Node2 Node1 Node2

Node1 Node2

Shared Disk

Looks Alive

Heartbeat Communication

Is Alive

Node1 Node2

Look Alive – 5 Secs

Is Alive – 60 Sec

Select @@ServerName

Node1 Node2

Private Network

Users & Application

Public NW Users and Applications Public NW

->In 2005 we can configure up to 8 nodes whereas in 2008 we can configure up to 16 nodes.

**Binary files of SQL Server:**

**N1 VSQL N2**

**VSS1**

Where it is running

Connection String: VSQL

Gmail

(7)

Windows Cluster

Public (2)

Public (1)

Private (4)

Private (3)

Private (6)

Public (5)

How many IP address are required to configure a two-node clustering?

A-A -> 7

A-P -> 6

A-A-A -> 10

A-P-P -> 8

What is the difference between public and private network?

**Private Network** is used for internal communication of cluster which is know as Heartbeat. Whereas **public network** is used for connecting the SQL Server to the external world.

**Heartbeat** is the signal which will be the signal sending to the other node to identify the health of the other node. This heartbeat signal will run for every 5secs.

**Looks Alive** is the normal check which runs for every 5secs.

**Is Alive** is thorough check when the server is not responding for Looks Alive. Is Alive will decide the failover process. It will send a command SELECT @@SERVERNAME to identify other server is running or not. If the server is not responding for Looks Alive and Is Alive then it will start initiating for the failover process.

**QUORUM:** Quorum drive contains cluster configuration information and a quorum.txt file will be created in quorum drive. Quorum will decide the failover of the SQL Server from one node to another node. Different types of quorum types **(1)**. Node majority (odd no of nodes). **(2)**. Node and disk majority (Even no of nodes). **(3)**. Node and File Share majority.

**What is the difference between 2005 cluster and 2008 cluster?**

2005 cluster supports up to 8 nodes

2008 cluster supports up to 16 nodes

In 2005 cluster, if you install the SQL Server in one node automatically the **binaries will** install in another node. We required to perform one installation.

In 2008 or later cluster, if you install the SQL Server in one node the **binaries will not** automatically install in other nodes. We have to add the binaries manually by using add node option in SQL Server installation center.

**Is MSDDC mandatory to install in SQL Server 2005?**

No, it is not mandatory, it is just optional.

**What are the things you will require to install the SQL Server cluster?**

->We should require the network name

->And we should require knowing the disk selection for keeping the data, log, backup and tempDB files.

->We should know the IP address of the SQL Server

->Apart from that we should know what services we have to install and default instance and named instance and Service Account (SA) and password.

**What is the start-up type of the services in Cluster?**

Manual

**How will you open failover cluster manager?**

In windows server 2003 🡪 go to run -> type Clustadmin

In windows server 2008 or later 🡪 go to run -> type Clustadmin.msc

**How will you apply Service Pack on A-P cluster?**

a). Login into node2 and apply the patching on node2 and then reboot the server.

b). Now come to Node1 and perform the SQL Server failover from Node1 to Node2 and apply the patch in Node1 after that reboot the server.

**How will you apply the patch in A-A cluster?**

a). login into to node2 and perform the failover of SQL Server2 from node2 to node1.

b). then apply the patch in node2 and reboot it.

c). Now login into node1 and perform the failover of SQL Server1 and SQL Server2 from node1 to node2 then apply the patch in node1 and reboot it.

d). After that balance the SQL Server (bring the SQL Server1 from node2 to node1)

**How will you perform manual failover of SQL instance in cluster?**

Go to Cluster manager -> Services and applications -> Select the SQL Server -> right click -> move this server or application to another node.

**How will you add a new disk to the cluster?**

Open failover cluster manager -> services & applications -> select SQL Server -> Right click -> add storage -> add the drive.

**How will you install the cluster?**

First, windows team will install the failover cluster manager and then thet will add the nodes then network team will configure the public network and private network then the storage team will configure the storage for the cluster.

First, we have to verify the validation of the cluster by performing all the tests if there are any issues, fix at that time itself then perform the main installation in node1 and go to node2 and perform add node installation.

**How will you verify the cluster log?**

Cluster log will be available in C:\windows\system32\.

**How will you troubleshoot cluster issues?**

First, we have to verify the cluster log and then we can verify SQL Server error log and then we can verify the event viewer and then we can verify critical cluster event logs to identify the error.

**What are the 2012 SQL Server new features?**

1. AlwaysOn
2. Contained databases
3. Default backup location
4. Column store index
5. Restart free SQL Server patching

**AlwaysOn:** AlwaysOn is a combination of windows cluster and DB mirroring.

In normal mirroring does **not** support for **One-Many**. But in AlwaysOn we can configure One-Many (1 principal -> 4 Replicas) in 2012.

In normal mirroring, it is not possible to read the secondary, but in AlwaysOn we can read the secondary database and we can use it for reporting purpose.

**What are the basic prerequisites to configure AlwaysOn?**

1). We have to configure the windows cluster (SQL Cluster is not required).

2). We don’t require any shared storage. Normal disks are sufficient to configure the AlwaysOn.

3). Require windows server 2008 R2 or later

4). Enable AlwaysOn feature by going to the SQL Server Configuration manager.

5). Both SQL & OS Editions, Versions should be at same level on all participating nodes.

6). Make sure you have a separate NIC’s for public and private communication

7). Make sure you have two free IP’s each for windows cluster IP and Always on listener IP

8). The Windows Cluster Account (Windows Cluster’s name) installed on these 2 servers needs to be granted the Create Computer Object privilege in the OU (Organization Unit) where these 2 servers reside in Active Directory.

9). Nodes in cluster (Ex: here 2 servers) must have drives of the same size and with the same name. There must be the same paths inside the drives. The reason is that; To get a database to SQL Server Availability Group, it is more convenient to have the same drives and paths on the secondary server

10). Recommend to have same collation on all replicas

11). Create shared network share on all participating nodes

12). Make sure your databases are in Full Recovery Mode, not Simple or Bulk Logged

13). Databases included in your AlwaysOn group must be user databases. System databases cannot participate in AlwaysOn Availability Groups.

14). Make sure full backups of each of your databases are made prior to installing AlwaysOn

**What are the basic prerequisites for Database to add in availability group?**

The database should be in full recovery model and at least one full backup should be taken to the database then only we can add the database to the availability group.

**What is contained database?**

By using contained database, we can avoid the orphan users. In contained databases we can create the user without any login.

So, when we move the contained database from one server to another server, we don’t require to fix the orphan users.

->we can configure 4 replicas in 2012 and 8 replicas in 2014.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Anjaiah Karnati\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Question and Answers:

1. What is the difference between MDF and NDF?

MDF is Master Data File, NDF is New Data File. MDF file will be created when database created.

When MDF is full then we go for additional file called NDF. NDF file will not create when we create database.

1. What is the difference between Default instance and Named instance?

**Default Instance:** If you are installing SQL Server as a default instance then the SQL server name will be the windows server’s name. We can install only one default instance per server.

**Named Instance:** If you need more than one instance then we must choose named instance option. For named instance we must provide name for SQL Server. We can install 49 named instances per server.

1. What are the things you will consider before installing the SQL Server?
   1. When a customer request to install SQL Server, we must know
      1. The basic pre-requisites (dotNet framework, windows installer and required hardware) of SQL Server.
      2. We have to choose default or named instance. If it a named instance, then we have to know the name of the SQL Server instance.
      3. What services to be installed on the server (DB Services, Integration Services, Reporting Services, Analysis Services).
      4. As a best practice, we have to know the location of the DATA file and LOG file.
      5. We require Service Account name and password.
2. What is the instance ID of default instance?

MSSQLSERVER

1. What is the instance ID or Name of named instance?

Whatever the name we have provided that will be the ID or name of the named instance.

1. What is collation?

Collation is the language of the SQL Server, which is displayed to the END User. The default collation is ‘**SQL\_Latin1\_General\_CP1\_CI\_AS’.**

1. What is Service Account?

Service Account is mainly used to run the services of SQL Server. The main advantage of service account is it will not locked-out and its password never expire.

1. What is the difference between windows and mixed mode?

In windows mode only windows users can access the SQL Server. Whereas in mixed mode windows users and SQL users can access the SQL Server.

1. How will you change the Authentication mode?

SQL Server -> Right Click -> Properties -> Security -> Change the Authentication mode -> Windows <->Mixed. But we need SQL Server Restart to apply changes.

1. How will you troubleshoot if installation fails in middle?
   1. First, we have to verify summary.txt log file. This summary log file contains information about all the services we have installed and each service. We will get the info of fail services.
   2. Apart from that we can also verify system configuration check report to identify the errors.
   3. Apart from that we can also verify the event viewer to identify the error.
2. What is the default location of the Summary.txt file?

By default, the summary.txt file will be going to ‘C:\Program Files\Microsoft SQL Server\140\Setup Bootstrap\Log’.

1. What is the difference between 2005, 2008R2 installation?
   1. In 2008R2 and later, we have the installation center option whereas in 2005 we do not have installation center option.
   2. In 2005, if we want to install Reporting Services IIS (Internet Information Services) is required. Where as in 2008R2 IIS is not required for SSRS installation.
   3. In 2008 R2, we can place Data and Log files, Backup and TempDB in different locations whereas in 2005 we have option for Data files only.
   4. In 2019, addition to the above, it will calculate number of Tempdb files to created, memory recommendation, MAXDOP recommendation, etc.
2. What is difference between 32bit and 64bit machine?

In 32bit server the data transfer rate is 32bits where as in 64bit server the data transfer rate is 64bits and the hardware supported by 32bit is very low and if supports only 4GB memory whereas in 64bit there is no limit.

1. What are the precautions we have to take before applying Service Packs or CU?

Before applying Service Pack, please take the backup of system databases and user databases. After applying the service pack, we have to perform the rebooting the windows.

1. How will you troubleshoot if service pack installation gets failed?

In 2005 SQL Server in-order to roll back the Service Pack we have to restore all system and user databases.

In SQL Server 2008R2 we can uninstall the Service Pack by going to the location control panel -> view installed updates -> select the service pack which you want to uninstall -> click uninstall.

1. What is the database ID for Resource DB?

32767

1. What are the start-up parameters available in SQL Server?

-d -> Master database data file

-l -> Master database log file

-e -> Error log file

1. What is the use of configuration manager in SQL Server?

SQL Server Configuration Manager is used to start the services and stop the services. Apart from that we can also manage the SQL Server. And we can change SQL Server port number and enable/disable protocols, etc.

1. How will you change Service Account of SQL Services?

SQL Server Configuration Manager -> SQL Server -> Properties -> Logon -> change the Service Account of SQL Server. We have to restart the SQL Services in-order to apply the new service account.

1. Shortcut to open configuration manager?

For 2005 – SQLServerManager.msc

For 2008 and 2008 R2 – SQLServerManager10.msc

For later – SQLServerManagerXX.msc – Replace XX with corresponding version number.

1. If SQL Server is not starting, then how will you troubleshoot?

(i). First, we have to verify the error log file by going to the start-up parameters (configuration manager).

(ii). Verify the error log and identify the error

(iii). Verify master data & log file location, file names and their extensions.

1. How will you identify, how many SQL Servers are running?

SQL Server configuration manager -> SQL Services -> there you can see list of SQL Servers running.

1. What is the difference between Char and Varchar datatype?

Char is a fixed length datatype, whereas Varchar is a variable length datatype. In char free space will be wasted and in varchar free space will not be wasted.

1. What is the difference between TRUNCATE and DELETE?

DELETE: deletes the data row by row whereas TRUNCATE deletes the data page by page manner.

DELETE is a logged operation and TRUNCATE is a non-logged operation.

For DELETE, after deleting the data rollback is possible.

For TRUNCATE, after deleting data rollback is not possible.

1. What is the default port number of SQL Server?

1433

1. What is the default port number for DAC?

1434

1. Default port number for DB main?

25

1. What is PAGE?

It is the default storage allocation of the SQL Server.

1. Default size of PAGE?

8KB

1. PAGE header & footer size?

96 bytes

1. What is the difference between GAM and SGAM?

GAM covers 64000 extents which is of size 4GB. GAM maintains uniform extents. If the bit is One (1) the extent is free, if the bit is Zero (0) means the extent is allocated.

SGAM also covers 64000 extents which is of size 4GB. SGAM maintains mixed extents if the bit is Zero (0) means it is not used if the bit is One (1) means it is mixed extent with free pages.

1. Is it possible to take full or differential backup of tempDB?

No.

1. How will you troubleshoot a failed job?

Right click on job -> view history -> based on the error troubleshoot the error.

1. How will you identify server name, database name, user details from a backup file?

By using RESTORE HEADERONLY option we will identify the backup details

RESTORE HEADERONLY FROM DISK=’C:\BACKUP.BAK’

1. How will you identify whether the backup file is valid or not?

By using RESTORE VERIFYONLY command we can identify the backup is valid or not.

RESTORE VERIFYONLY FROM DISK=’C:\BACKUP.BAK’

1. What is tail log backup?

Tail log backup will backup all transactions from the last log backup to the corrupted time.

1. What is the command to take the tail log backup?

BACKUP LOG <DB NAME> TO DISK=’C:\TAILLOGBACKUP.TRN’ WITH NO\_TRUNCATE

1. How will you perform the point in time restore to the specific time?

RESTORE LOG <DB NAME> FROM DISK=’C:\BACKUP.TRN’ WITH STOPAT =’2015-12-18 T19:46:59’

1. What is the checkpoint?

Checkpoint flush the data from buffer pool to hard disk. Checkpoint interval depends on recovery interval based on the recovery interval checkpoint will invoke and also checkpoint runs when SQL Server stops, when you change the recovery model and when log reaches 90% of the total space. In this scenarios checkpoint will run.

1. What is the difference between log backup and tail log backup?

Normal log backup will backup all the committed transactions available in the log file.

Tail log backup will backup all the committed and non-committed transactions.

While taking log backup database is online. While taking tail log backup database is corrupted or offline.

**Log backup command ->** BACKUP LOG <DB NAME> TO DISK=’C:\BACKUP.TRN’

**Tail log backup command->** BACKUP LOG <DB NAME> TO DISK=’C:\BACKUP.TRN’ WITH NO\_TRUNCATE, NORECOVERY, CONTINUE\_AFTER\_ERROR

1. What are the pages available in SQL Server?

DATA, INDEX, GAM, SGAM, PFS, BCM, DCM, IAM

1. How SQL Server identify differential or how SQL Server identifies which data is modified after the full backup?

By using DCM page SQL Server identifies data modification after full backup. In DCM page if the extent is modified then it will mark as ‘1’. If the extent is not modified, then it will mark as 0. While taking the differential backup it will backup all the extents which are in DCM page

DCM – Differential Change Map

1 – modified 0 – not modified

0

0

0

1

1

0

1

1

1. How SQL Server identifies BULK transactions?

By using BCM (Bulk Changed Map)

1. Explain about the log shipping?

->Log shipping is the process of shipping log backup from one server to another server.

->After configuring the log shipping there are 3 jobs mainly created. 1). Log backup job.2). Copy job 3). Restore job.

->Log backup job available in primary server and copy & restore job are available in secondary server.

->Log backup job will backup the transactions and will kept in shared folder1.

->Copy job will copy the backups from shared folder1 to shared folder2.

->Restore job will restore the backups into secondary server. This is how the data transfer from primary to secondary server. In case of primary server down we have to restore the secondary server and we have to use the secondary server.

1. What is TUF file?

TUF stands for Transaction Undo File, which contains all uncommitted transaction. TUF file will be created when we keep the database in STANDBY mode in log shipping.

1. What happens if my TUF file deleted?

Log shipping will break.

1. If log shipping breaks because of LSN mismatch. How will you troubleshoot?

->If there is no full backup happened after the LSN mismatch then we can take the differential backup of primary database and we can restore it on the secondary server, this will make the log shipping continue to work.

-> If there is a full backup happened after the LSN mismatch then we cannot use the differential backup. We have to use the same full backup on the secondary and re-configure the log shipping.

1. How will you identify the error details in the log shipping?

Go to MSDB -> verify the table -> log\_shipping\_monitor\_error\_details

1. What are the different severities we have?

Severity 1, Severity 2, Severity 3, Severity 4

SLA -> Service Level Agreement

Severity 1 -> should be resolved in 4 hours

Severity 2 -> 8 hours

Severity 3 -> 24 hours

Severity 4 -> 48 hours

1. What are the ticketing tools you have used?

Remedy -> 7.0, Goss, Service Desk, BugZero, FootPrints, HPSM(HP Service Manager).

1. What is the default port number for Mirroring?

5022, 5023

1. What is the default ENDPOINT name in Mirroring?

Mirroring.

1. What is endpoint?

Endpoint is a communication channel between application and database.

1. How many endpoints required to configure mirroring?

1

1. Is it possible to change default port number in mirroring?

Yes, we can change.

1. How will you bring mirror database online?

a). First, we have to make the partner off – **ALTER DATABASE <DB NAME> SET PARTNER OFF**

b). post breaking the mirroring, we can recover the mirror database by below command – **RESTORE DATABASE <DB NAME> WITH RECOVERY**

1. If you configure the mirroring in High-Protection mode. How will you bring the mirror online?

We can bring the mirror online by using the command – **ALTER DATABASE <DB NAME> SET PARTNER FORCE\_SERVICE\_ALLOW\_DATA\_LOSS**

1. What are the different database states in mirroring?

SYNCHRONIZED, SYNCHRONIZING, DISCONNECTED, SUSPENDED

1. Is it possible to access the mirror database in mirroring?

No, it is not possible to access mirror database directly. But we can access by configuring the database snapshot.

1. Difference between Log shipping and Mirroring?

**Log shipping:**

**a).** we can configure lower version to higher version.

b). Manual failover in log shipping

c). we can configure for **One to Many**

d). Manual failover will take more time.

**Mirroring:**

**a).** Both primary and secondary servers should be in same version.

b). Automatic failover

c). Mirroring only for **One to One**

d). Automatic failover will not take much time.

1. What is the difference between Clustered Index and Non-Clustered Index?

**Clustered Index:**

a). We can create only one clustered index per table.

b). Clustered Index column doesn’t contain duplicate and NULL values.

c). Clustered Index contains the data pages at leaf level.

d). Clustered index will be created when we create a **Primary key** on the column.

**Non-Clustered Index:**

a). We can create 249 non-clustered indexes in 2005 and 999 in 2008 R2 or later.

b). As non-clustered index column will have **single NULL** value and multiple duplicate values.

c). Non-clustered Index doesn’t contain the data pages at leaf level. It has a pointer pointing towards clustered index.

if there is no clustered index in the table then the non-clustered index points towards the disk.

d). Non-Clustered Index will be created when we create a **unique key** on the column

1. What is the difference between Login and User?

->Login scope is server level.

->User scope is database level.

->Login will exist without user. But user cannot exist without Login.

1. Can we create a user without Login?

No.

1. How will you provide permissions to a single table?

By using the command GRANT (All/Specific permission) on table to user (WITH GRANT option).

Or

Create login -> don’t provide any server level role -> select user mapping -> select database -> select specific role

Go to database -> security -> user properties -> search select all objects -> select table -> provide permission.

1. How will you provide a permission to manage the jobs or what permissions you will provide to manage the jobs?

We must provide ‘**SQLAgentOperatorRole’** in MSDB database.

1. What permission you provide to manage SSIS packages?

We must provide **db\_ssisadmin** in MSDB database.

1. What permissions you provide to run the profiler?

Use Master

Go

**GRANT ALTER TRACE TO <LOGIN OR USER>**

1. How will you perform restore a database from production to test or dev environment or lower environments?

->Take the backup of production database.

->Copy the backup file to test or dev server.

->Restore the database in test or dev server

->Fix the orphan users.

1. How will you troubleshoot blockings?

We can identify the blocking by using the commands:

**SP\_WHO** (or) **SP\_WHO2** (or) **SELECT \* FROM SYS.SYSPROCESSES WHERE BLOCKED<>0**

After identifying the blocking, we have to identify the queries by using the command: **DBCC INPUTBUFFER(SPID)**

If SELECT query is blocking with any other query, then we can kill the SELECT command by using KILL option: **KILL <SPID>**

if any query other than SELECT then we have to check with user and take the approval from the user and we have to kill that command.

1. How will you migrate the database from one server to another server?

a). Backup/Restore

b). Attach/Detach (downtime is required)

c). Import/Export

d). Copy database wizard

1. Is it possible to upgrade the SQL Server 2000 to SQL Server 2012?

No, directly it is not possible. First, we have to upgrade the SQL Server 2000 to SQL Server 2005 -> 2008 -> then we have to upgrade to 2012.

1. What is the difference between detaching Database and taking offline database?

When you detach the database, the database is not visible in SSMS. When you take database offline, the database is visible in SSMS. But it will not accept any connections.

1. Compatibility level of SQL Server?

2000 -> 80 -> 8.0

2005 -> 90 -> 9.0

2008 -> 100 -> 10.0

2012 -> 110 -> 11.0

2014 -> 120 -> 12.0

2016 -> 130 -> 13.0

2017 -> 140 -> 14.0

2019 -> 150 -> 15.0

1. How will you upgrade the Database from one server to another server?

a). Take the backup of the source database.

b). Copy the backup to new server

c). Restore it on new server

d). Then change the compatibility level for the database.

e). Then fix the orphan users.

1. What is the output of DBCC CHECKDB?

CHECKDB found 0 allocation errors and 0 consistency errors in database 'DB Name’.

1. Best recommendation for configuring TEMPDB?

It is recommended to configure multiple TempDB data files equals to maximum processes. But maximum suggested by Microsoft is 8.

1. How will you troubleshoot the disk space issue?

1). First, we have to verify the disk and check what are the files available in the disk. There must be 3 files available in the disk a. DATA, b. LOG, c. BACKUP files.

If the disk contains backup file, then we have to identify old backups and delete the old backup files and release the space.

If the disk contains DATA files, then we have to create a new file from another drive then we have to stop the **Auto Growth** for the previous file.

If the disk contains LOG files, then to release the space, first we have to take the log backup and then shrink the log file.

1. How will you troubleshoot if the Log file is full?

1). Verify the open transaction by using the command: **DBCC OPENTRAN()**

If there is open transaction running then we have to wait until the transaction is completed, then we have to take log backup and shrink the log file.

If there is no open transaction running, then we have to log backup and shrink the log file.

Sometimes it will not shrink log file when you take initial log backup then we have to take log backup again and try shrinking the log file. Even still if you are unable to shrink the file, we run the command: **SELECT \* FROM SYS.DATABASES** and we can verify the column **LOG\_REUSE\_WAIT\_DESC**

1. How will you troubleshoot if the database goes into SUSPECT mode?

i). We have to reset the database status by using the command **EXEC SP\_RESETSTATUS (DB NAME)**

ii). Then we have to keep the database in EMERGENCY mode by using the command

**ALTER DATABASE <DB NAME> SET EMERGENCY**

iii). We keep the database in single user mode by using the below command:

**ALTER DATABASE <DB NAME> SET SINGLE\_USER WITH ROLLBACK IMMEDIATE**

iv). Run the **DBCC CHECKDB** to check the database health then it there are any pages are corrupted then run the below command

**DBCC CHECKDB(‘DB NAME’) REPAIR\_ALLOW\_DATA\_LOSS**

v). Finally, after repair is done, we have to keep the database in multi user mode by using the command **ALTER DATABASE <DB NAME> SET MULTI\_USER**

1. Difference between Incident and Problem and Change?

ITIL process -> information technology infrastructure library.

-> **What is an Incident?**

An incident is something the needs to be resolved immediately. This can be either be thorough a permanent fix, a workaround, or a temporary fix.

**Example:** Disk space issue, Backup failure and job failure and SQL Server restarted.

**-**>**What is problem?**

if an incident is repeatedly occurring then we can consider it as a problem. Problems however are not incidents. An incident can raise a problem, especially if there is high possibility that the incident might happen again.

**Example:** if a disk space issue occurs every day, then we can consider this as a problem, and we can work for the permanent fix.

**-**>**What is a Change?**

If you want to perform any modification on the configuration settings in SQL Server, we require a change. The primary objective of the change management is to enable beneficial changes to IT services.

**Example:** 1-> Installing a new SQL Server

2->Applying a patch for SQL Server (require business approvals). Like all the implementations in SQL Server will comes under Change Management.

1. DBCC commands:

i). **DBCC INPUTBUFFER(SPID)** – used to identify the query from a SPID.

ii). **DBCC OPENTRAN** – used to identify the open transactions in a database (currently running transactions).

iii). **DBCC SQLPERF(LOGSPACE)** – used to identify the details about log file size and space usage of log file for all the databases.

iv). **DBCC TRACEON (1117)** – to enable the trace flag

v). DBCC TRACEOFF (1117) – to disable trace flag

vi). DBCC CHECKDB – To check the health of the database.

Used to identify the integrity of the database when we run checkdb in database internally it will run 3 more commands.

**DBCC CHECKALLOC**

**DBCC CHECKTABLE**

**DBCC CHECKCATLOG**

vii). DBCC SHRINKDATABASE – used to shrink the database.

viii). DBCC SHRINKFILE – used to shrink the data of log file.

1. DMV commands:

i). dm\_exec\_requests – to check current running process

ii). dm\_db\_index\_physical\_stats – to check fragmentation details

iii). dm\_exec\_query\_plan – to check query plan

iv). dm\_exec\_connections – to check current connections

v). dm\_exec\_cached\_palns – to check cached plans

vi). dm\_exec\_cursors – to check cursors

vii). dm\_exec\_query\_stats – to check query stats

viii). dm\_os\_performace\_counter – to check performance counter values

ix). dm\_os\_windows\_info – to check windows version from SQL Server.

x). dm\_db\_log\_space\_usage – to check log file usage in bytes.

1. What is Lazy write?

**lazy writer**: is a system process that keeps free buffers available by removing infrequently used pages from the buffer cache. Dirty pages are first written to disk. The eager write process writes dirty data pages associated with minimally logged operations such as bulk insert and select into. This process allows creating and writing new pages to take place in parallel. That is, the calling operation does not have to wait until the entire operation finishes before writing the pages to disk.

1. What is Listener?

**Listener**:-

TCP protocols connection only

**Modes** :-

Application intents: Read/Write or Read-only routing

Multi Subnet failover:- True /False --> It works/checks in parallel for all subnets .

**Monitor**:-

Exec Sp\_server\_diagnostics 5

Health check option:- Failure Condition Level:-

5 options have

3 is default

Along with listener we have another option to connect clients sql server in 2014 direct go to SQL server which will not disconnect connection if listener goes down or AG failovers too.

In Asynchronous failover --> it disconnect all the database, suspended on all relates servers, it can loss data and table too.. But synchronous no issues, all will be fine.

Missing data can be get back using Snapshot of database. We will have to take snapshot of database after the manual failover..

If we want to move back AG group from Asynchronous to synchronous without losing data then we have to commit synchronized on aynscronus node. As below:-

Syncronized\_Commit

After failover we can set asynchronous back to asynchronous.

Alter availability group <AG Name> Failover.

**Regular Usage Scripts:**

Regular usage scripts SQL DBA

=====================

1. How to find on which drive the database files (mdf & ldf) are located?

use master

select \*from sysaltfiles where filename like 'd%'

(OR)

select \* from master..sysaltfiles where filename like 'd%'

**for particular database:**

sp\_helpfile

2.How to shrinklog file if it is full?

backup log dbname with no\_log

dbcc shrinkfile(2)

3. how to see open transactions in a database?

dbcc opentran

4. HOW TO see if particular process is blocked?

Use master

select \* from sysprocessess where blocked>0 and spid=57

5. why buffer cache hit ratio <90?

if the bottlenecks are there

then the number of pages read into the cache will vary

and the ratio of finding the data by the server in the cache will decrease

6 how to find attribute of particular file?

Execute spFileDetails 'c:\autoexec.bat'

7. how to see the status of database?

dbcc showfilestats

8.How to see the CPU usage? (If buffer cache hit ratio<90)

select \*from sys.sysprocesses order by cpu asc

9.How to find out the info of a specific table on a database?

select \* from sysobjects where name like '%table\_name%'

10. What is Transaction Undo File? Why is it required?

Undo file is needed in standby state because while restoring the log backup, uncommitted transactions will be recorded to the undo file and only committed transactions will be written to disk there by making users to read the database. When you restore next t-log backup SQL server will fetch the uncommitted transactions from undo file and check with the new t-log backup whether the same is committed or not. If it’s committed the transactions will be written to disk else, it will be stored in undo file until it gets committed or rolled back.

11.How to get the data the from linked server?

Select \* from LinkedServer.DBName.SchemaName.TableName

12.How to find the service pack version?

SELECT SERVERPROPERTY('PRODUCTLEVEL')

13.How to see the collation setting for server level?

SP\_HELPSORT

14.How to see log usage info in a database?

select \* from sys.databases where name like '%log\_reuse\_wait\_desc%'

15.How to see the connection info by using DMV'S?

select \* from sys.dm\_exec\_sessions

16.How to see the transaction isolation level?

DBCC USEROPTIONS

17.SQL Server 2000 database compatible level to SQL Server 2005

EXEC sp\_dbcmptlevel AdventureWorks, 90;

18.How to see the memory ocupation of eache query running under Query Analyzer

select \* from sys.dm\_exec\_query\_memory\_grants

19.How to clear the data from buffer cache

DBCC DROPCLEANBUFFERS

20.How to see how much space is allocated in TEMPDB for a particular task

Select \* from sys.dm\_db\_session\_space\_usage

OR

sys.dm\_db\_task\_space\_usage

21.The following DMV query can be used to get useful information about the index usage for all objects in all databases

Select \* from from sys.dm\_db\_index\_usage\_stats

order by object\_id, index\_id

22.The following query to list all the schedulers and look at the number of runnable tasks.

Select \* from from sys.dm\_os\_schedulers where scheduler\_id < 255

23.Which Query is taking more cpu time

select \* from sys.dm\_exec\_query\_stats

24.How to find whether any active requests are running in parallel for a given session by using the following query.

select \* from sys.dm\_exec\_requests

25. if SQL server installation failed in sql2000, where can I see that failed info?

sqlstp.log, summary.txt

26. How to see the lock info?

select \* from sys.dm\_tran\_locks

27. How to open cluadmin from windows 2008 server?

cluadmin.msc