MS-SQL DBA

25 November 2024 10:08

- Primary key
- Foreign key of a table must n should be primary key of some other table
- Candidate key - one or more columns that could serve as primary key (one of these is set to primary key and another is called alternate key)
- Alrenate key -another candidate key which is not primary key (when user doesn't know primary key then they can search using alternate key -ex:train name and number)
- Unique key
- Composite key
- Seconady key
- Super key
- Database objects
  - Tables,views,indexes,procedures(using create command all are objects)
  - View - create view viewname
    - Why? - if a person wants to see only particular part of data when there are lot of entries
    - Abstraction acquired by view(end user can't see the actual base table)
    - Simple and complex view
  - Inbuilt ,user defined types
  - Index(key+ addresss of the key)
    - Culstured -sorted form
    - Non-clustered- non sorted form
  - Sequence - automatic generation (random number), auto increment, usually to generate primary key
- Collection of fields - records ; collection of records - file
- Properties
  - Atomicity -transaction is treated as single (if any single manipulation fails whole transaction is rolled back and no updation in database)

  - Locks - table lock,row lock, page lock
- OLTP - day to day /daily current operational data
- OLAP - historical data -cannot be changed
- Heap - not stored in sorted manner(data is appended not sorted)
- IOT table(index organized table)- sorts the data
- Normalization (to avoid redundancy,update,insert anamolies(kinda errors),ambiguous) -process of organising and structure the data
  - 1NF - each table cell should contain signle value and each record should be unique value
  - 2NF - be in 1NF and single column primary key that does not functionally dependent on any subset of candidate key relation
    dividing the table into two tables and relating each other with primary key and foreign key(when we have composite key and non-key attributes depends on only part of composite key)

Student_ID	Course_ID	Instructor_ID	Instructor_Name	Grade
1	101	1001	Dr. Smith	A
2	102	1002	Dr. Brown	B
1	102	1002	Dr. Brown	A
3	101	1001	Dr. Smith	C

To check if the relation is in **2NF**, we need to identify **partial dependencies**:

- The **primary key** is **Student_ID, Course_ID**.
- The **Grade** is fully dependent on the **entire primary key** (**Student_ID, Course_ID**). So, there is no partial dependency for the **Grade** attribute.
- However, **Instructor_ID** depends only on **Course_ID**, which is **part** of the composite primary key (**Student_ID, Course_ID**), and not the **entire** key. This is a **partial dependency**.
- **Instructor_Name** depends on **Instructor_ID**, which in turn depends on **Course_ID**. Since **Instructor_ID** is dependent on **Course_ID** (a part of the primary key), this is also a partial dependency.

Thus, the relation is **not in 2NF** because it has partial dependencies.

  - 3NF- be in 2NF and when there is transitive dependency (a->b,b->c and a->c) - non-key attribute dependent on another non-key attribute and indirectly on primary key
    - In order to achieve that non-key attribute directly depends on primary key we make a relationship

EmployeeID	Name	Department	DepartmentHead	DepartmentLocation
101	Alice	HR	John Smith	Building A
102	Bob	IT	Mary Johnson	Building B
103	Carol	IT	Mary Johnson	Building B

- **Primary Key**: EmployeeID
- **Non-key attributes**: Name, Department, DepartmentHead, DepartmentLocation

Issue:

- The **DepartmentHead** and **DepartmentLocation** depend on **Department**, not directly on **EmployeeID**. This is a **transitive dependency** because:
  - **Department → DepartmentHead** (i.e., a department has a specific head),
  - **Department → DepartmentLocation** (i.e., a department is located in a specific place),
  - **EmployeeID → Department** (i.e., each employee is assigned to a department).

So, we have **transitive dependencies**: **EmployeeID → Department → DepartmentHead** and **EmployeeID → Department → DepartmentLocation**.

- In the new structure:
  - The first table now contains **Department** as the primary key.
  - The second table uses **EmployeeID** as the primary key, and **Department** becomes a foreign key that references the **Department** table.

Resulting Tables After Applying 3NF:

1. **Employee Table**:

EmployeeID	Name	Department
101	Alice	HR
102	Bob	IT
103	Carol	IT

2. **Department Table**:

Department	DepartmentHead	DepartmentLocation
HR	John Smith	Building A
IT	Mary Johnson	Building B

  - BCNF - be in 3NF and
    - **BCNF Rule**: A relation is in **BCNF** if, for every functional dependency A→B, the determinant A is a **candidate key**. This means that no non-prime attribute (an attribute not part of any candidate key) can act as a determinant for any other attribute in the relation.

Student_ID	Course_ID	Instructor_ID	Instructor_Name
1	101	1001	Dr. Smith
2	101	1001	Dr. Smith
3	102	1002	Dr. Brown
4	103	1003	Dr. Clark

5	101	1001	Dr. Smith

- **Student_ID → Course_ID**: **Student_ID** is a candidate key, so this is fine in BCNF.
- **Course_ID → Instructor_ID**: **Course_ID** is **not** a candidate key because it does not uniquely identify the entire row (you would need both **Student_ID** and **Course_ID** to uniquely identify each row in the relation). Since **Course_ID** is not a candidate key and it determines **Instructor_ID**, this violates BCNF.
- **Instructor_ID → Instructor_Name**: **Instructor_ID** is a candidate key for the **Instructor** table, so this functional dependency is fine in BCNF.

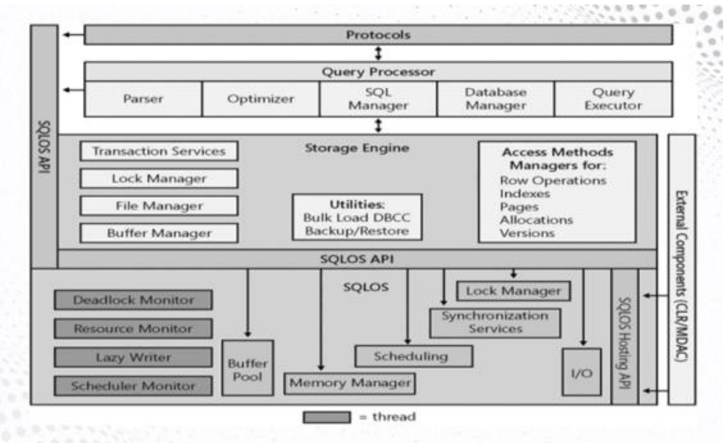
Conclusion:

- In **BCNF**, every **determinant** must be a **candidate key**. If a determinant is not a candidate key, then the relation violates BCNF and needs to be decomposed into smaller relations.

- **RAID -**reduntant array of independent disks

- T-SQL-(enhanced sql which includes loops,varibales,conditionals,stored procedures-execute batch of queries)
  - EXAMPLE

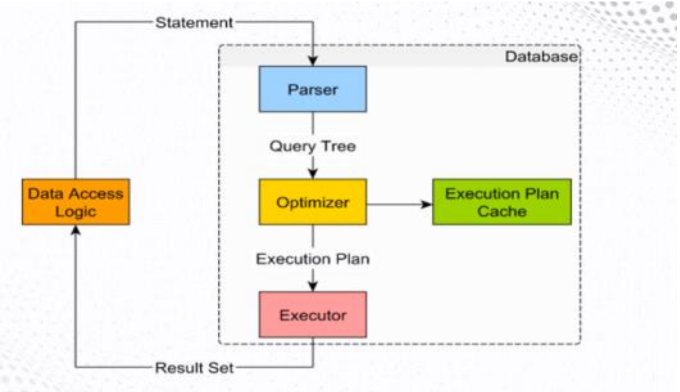
```
-- Create a table for demonstration
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName NVARCHAR(50),
    LastName NVARCHAR(50),
    Department NVARCHAR(50),
    Salary DECIMAL(10, 2)
);

-- Insert some sample data
INSERT INTO Employees (EmployeeID, FirstName, LastName, Department, Salary)
VALUES
(1, 'John', 'Doe', 'HR', 50000),
(2, 'Jane', 'Smith', 'IT', 60000),
(3, 'Bill', 'Johnson', 'Sales', 45000);

-- Declare a variable to hold the department name
DECLARE @Department NVARCHAR(50);
SET @Department = 'IT';

-- Declare a variable to hold the average salary
DECLARE @AverageSalary DECIMAL(10, 2);

-- Try to calculate the average salary for a department
BEGIN TRY
    -- Calculate the average salary for the given department
    SELECT @AverageSalary = AVG(Salary)
    FROM Employees
    WHERE Department = @Department;

    -- If no rows are found, throw an error
    IF @AverageSalary IS NULL
    BEGIN
        THROW 50000, 'No employees found in this department.', 1;
    END

    -- Output the average salary
    PRINT 'Average salary in ' + @Department + ' department: ' + CAST(@AverageSalary AS NVARCHAR(50));

END TRY
BEGIN CATCH
    -- Error handling if something goes wrong
    PRINT 'Error: ' + ERROR_MESSAGE();
END CATCH;
```

  - If there is no value entered '0' is filled unlike sql where null is filled
- MSSQL SERVER ARCHITECTURE
  - Client server-based architecture - server response to client query(2 tier architecture-client and database on different systems)
  - 1 tier architecture - client and database on same system
  - Components of sql server
    - Database engine - creates and drives relational databases
    - Analysis service(ssas)- data analysis componenet
    - Reporting services(ssrs)- generating and deploying report
    - Integration service(ssis) - etl (extract,transform and load)
  - Sql server architecture
    - 

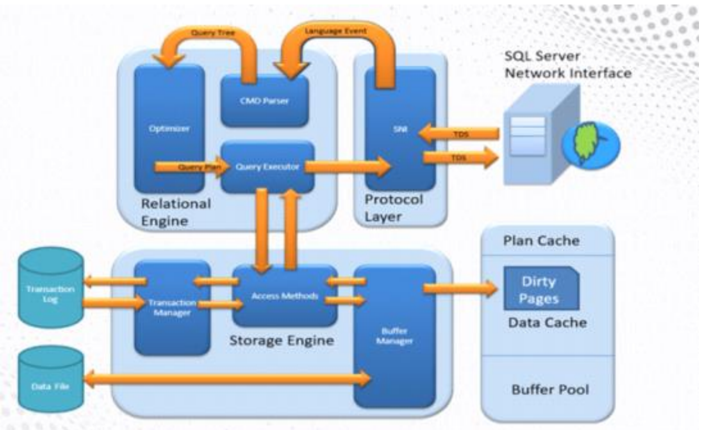
SQL Server Architecture

    - Protocol layer-server network interface-(shared memory,tcp/ip,named pipes and tds-tabular data stream:this enables data transfer from client to server machine)
      - Named pipes used by inter-process communication
    - Relational engine(cmd parser,optimizer and query executor)
      - Optimizer
        - Phase0- trivial plan
        - Phase1 -transaction processing plan
        - Phase2- parallel processing and optimizations
    - Storage engine(file types,access method,buffer manager,plan cache and transaction manager) - responsible for storage and retrieval from storage systems disk/san

- Data file and log file are at disk level
- Buffer manager
  - Plan chache
  - Data parsing
  - Dirty page
- Access methid
  - Ddl,select stament - buffer manager
  - Ddl, non-select - transaction manager

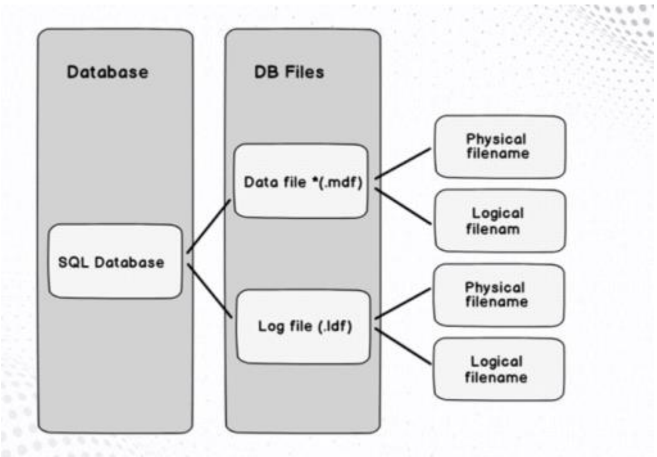
- Query execution process in SQL server
  - 

Query execution process in Server

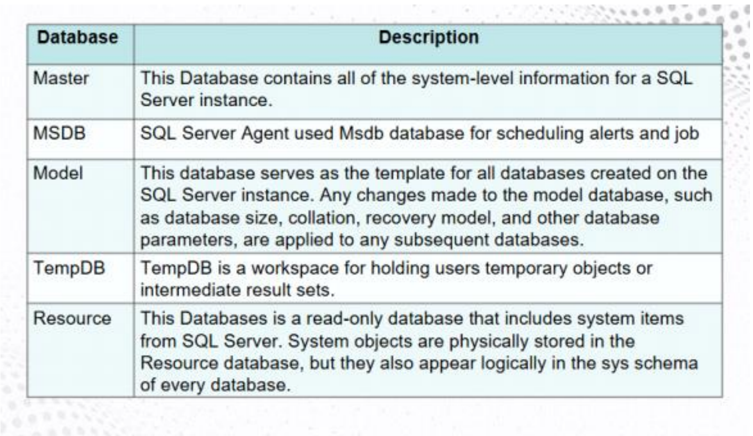
  - 

Process of SQL Server working on queries

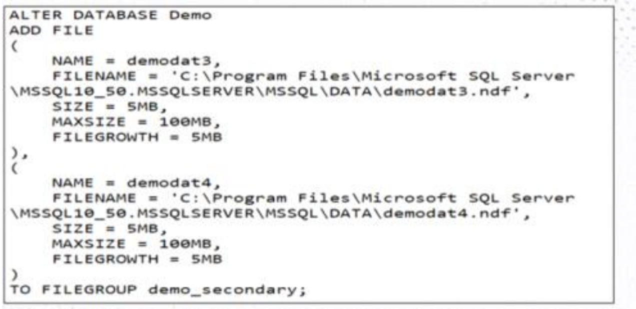
- Logical storage
  - The data is organized and accessed (databses ,tables,indexes and schemas)
  - It is how users and applications interact with the data via SQL queries, regardless of the physical storage structure.
  - Data is stored in data file as data pages(8kb)
  - Types of data pages
    - Data page
    - Text/image
    - Index
    - Pfs(page free space)
    - Bcm(bulk change map)
    - Iam(index allocation map)
    - Dcm(differential change map)
  - 1 extent = collention of 8 pages -The extent is the smallest unit of storage that SQL Server allocates to store data on disk.
    - Uniform extent- single object(table) is stored in 8 pages
    - Mixed extent -upto 8 objects can share mixed extent - each of eight pages is owned by different object
      - Global allocation map
      - Shared global allocation map
  - Each page has 96-byte header at the top(free space,page number,page type)
  - Row offset at end of page- stores the base location /postion/address of specific row - stored at end of page in reverse order
- Physical storage
  - It handles the mapping of logical objects (like tables and indexes) to physical storage locations on disk, ensuring data persistence and performance optimization.
  - Data file
    - Physical storage(.mdf)
    - Data files can be created together in file groups fro allocation,distribution and administration purposs
    - Primary data file(.mdf)
      - Data related to objects like tables,indexes,triggres .. Are stored here
      - Every database must have atleast one primary file
    - Secondary data file(.ndf)
      - Optional and contains user-specific data
      - Used to extend the storage of database
  - Log file
    - Stores a series of log records(.ldf)
    - Used for transaction management
    - Transaction log is imp for disaster recovery

- 

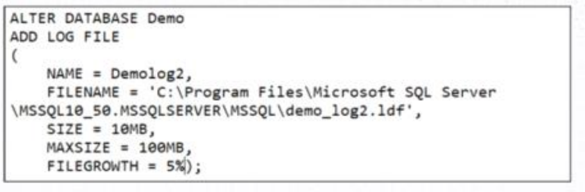
  - SQL server managers
    - Buffer manager - provides access to the necessary data(soft parse/hard parse)
    - Transaction manager - used when access method concludes request is an operator with no choice
    - Log manager-it monitors all system updates via transaction logs
    - Dirty pages-stored as transaction manager processing logic

- MSSQL DATABSAE CREATION
  - Database types
    - System database
      - Master-
      - Msdb
      - Model
      - Tempdb
      - Resource
      - 

Database	Description
Master	This Database contains all of the system-level information for a SQL Server instance.
MSDB	SQL Server Agent used Msdb database for scheduling alerts and job
Model	This database serves as the template for all databases created on the SQL Server instance. Any changes made to the model database, such as database size, collation, recovery model, and other database parameters, are applied to any subsequent databases.
TempDB	TempDB is a workspace for holding users temporary objects or intermediate result sets.
Resource	This Databases is a read-only database that includes system items from SQL Server. System objects are physically stored in the Resource database, but they also appear logically in the sys schema of every database.

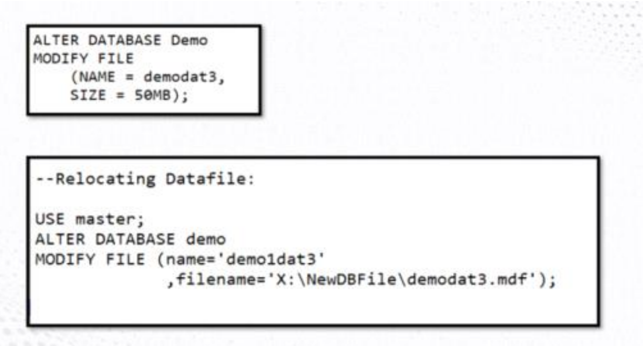
    - User databse
    - Add filegroup in sql server
      - Alter databse dbname add filegroup fgname
    - Add file to filegroup name
      - 

```
ALTER DATABASE Demo
ADD FILE
(
     NAME = demodat3,
     FILENAME = 'C:\Program Files\Microsoft SQL Server
\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\demodat3.ndf',
     SIZE = 5MB,
     MAXSIZE = 100MB,
     FILEGROWTH = 5MB
),
(
     NAME = demodat4,
     FILENAME = 'C:\Program Files\Microsoft SQL Server
\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\demodat4.ndf',
     SIZE = 5MB,
     MAXSIZE = 100MB,
     FILEGROWTH = 5MB
)
TO FILEGROUP demo_secondary;
```

  - Add log file to sql server
    - 

```
ALTER DATABASE Demo
ADD LOG FILE
(
    NAME = Demolog2,
    FILENAME = 'C:\Program Files\Microsoft SQL Server
\MSSQL10_50.MSSQLSERVER\MSSQL\demo_log2.ldf',
    SIZE = 10MB,
    MAXSIZE = 100MB,
    FILEGROWTH = 5%);
```

  - Modfiy data file in sql server

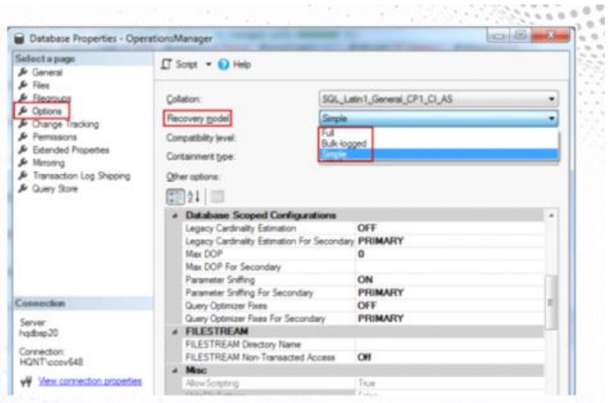


```
ALTER DATABASE Demo
MODIFY FILE
    (NAME = demodat3,
    SIZE = 50MB);
```

-
  ```
  --Relocating Datafile:

  USE master;
  ALTER DATABASE demo
  MODIFY FILE (name='demo1dat3'
                ,filename='X:\NewDBFile\demodat3.mdf');
  ```

- Data operations in sql server
  - Create new db
  - Add new filegroup
  - Add new datafile in primary/seconadry file
  - Change datafile size
  - Remove data files
  - Drop database

- MSSQL RECOVERY MODEL
  - Types of recovery model
    - Simple(by default)
      - Full /differential/file backup ,no transactional backup
      - Point in time not supported
      - It is used in Developing/testing
    - Full
      - All transactions Ddl+dml documented in transaction log file
      - Supports point in time
      - Data loss can be avoided
      - Bank transactions,finacial
      - More filegroups - to recover piecemeal of secondary file/read only file
    - Bulk logged
      - To prevent log file growth
  - Ways to change recovery model
    - Ssms



Database Properties - OperationsManager

Collation: SQL_Latin1_General_CP1_CI_AS

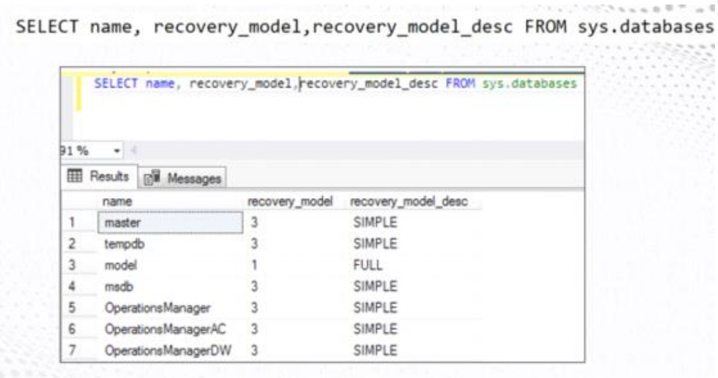
Recovery model: Simple

Legacy Cardinality Estimation: OFF
Legacy Cardinality Estimation For Secondary: PRIMARY
Max DOP: 0
Max DOP For Secondary:
Parameter Sniffing: ON
Parameter Sniffing For Secondary: PRIMARY
Query Optimizer Fixes: OFF
Query Optimizer Fixes For Secondary: PRIMARY
FILESTREAM Directory Name:
FILESTREAM Non-Transacted Access: Off
Allow Scripting: True

Server: hqdbsp20
Connection: HQNT\ccov648

Steps to take Simple backup in GUI mode

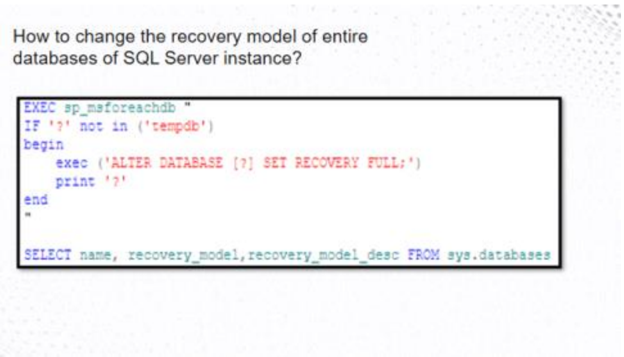
    - t-sql



```
SELECT name, recovery_model,recovery_model_desc FROM sys.databases
```

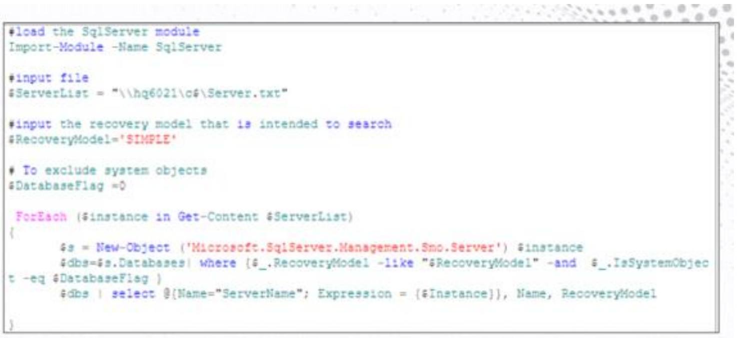
	name	recovery_model	recovery_model_desc
1	master	3	SIMPLE
2	tempdb	3	SIMPLE
3	model	1	FULL
4	msdb	3	SIMPLE
5	OperationsManager	3	SIMPLE
6	OperationsManagerAC	3	SIMPLE
7	OperationsManagerDW	3	SIMPLE

Query to select the recovery model

- ○ 

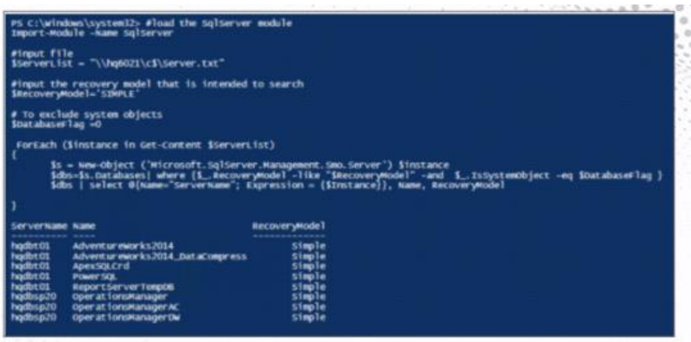
  How to change the recovery model of entire databases of SQL Server instance?

```
EXEC sp_msforeachdb "
IF '?' not in ('tempdb')
begin
    exec ('ALTER DATABASE [?] SET RECOVERY FULL;')
    print '?'
end
"

SELECT name, recovery_model,recovery_model_desc FROM sys.databases
```

- Powershell
  - ○ 

```
#load the SqlServer module
Import-Module -Name SqlServer

#input file
$ServerList = "\\hq6021\c$\Server.txt"

#input the recovery model that is intended to search
$RecoveryModel='SIMPLE'

# To exclude system objects
$DatabaseFlag =0

 ForEach ($instance in Get-Content $ServerList)
{
        $s = New-Object ('Microsoft.SqlServer.Management.Smo.Server') $instance
        $dbs=$s.Databases| where {$_.RecoveryModel -like "$RecoveryModel" -and  $_.IsSystemObjec
t -eq $DatabaseFlag }
        $dbs | select @{Name="ServerName"; Expression = {$Instance}}, Name, RecoveryModel

}
```

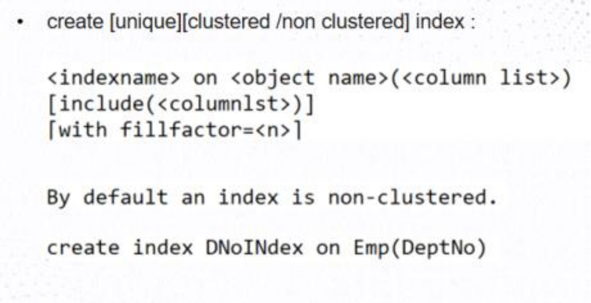
  SQL Server Command line interface for Change recovery model

  - ○ 

```
PS C:\windows\system32> #load the SqlServer module
Import-Module -Name SqlServer

#input file
$ServerList = "\\hq6021\c$\Server.txt"

#input the recovery model that is intended to search
$RecoveryModel='SIMPLE'

# To exclude system objects
$DatabaseFlag =0

 ForEach ($instance in Get-Content $ServerList)
{
       $s = New-Object ('Microsoft.SqlServer.Management.Smo.Server') $instance
       $dbs=$s.Databases| where {$_.RecoveryModel -like "$RecoveryModel" -and  $_.IsSystemObject -eq $DatabaseFlag }
       $dbs | select @{Name="ServerName"; Expression = {$Instance}}, Name, RecoveryModel

}

ServerName Name                              RecoveryModel
---------- ----                              -------------
hqdbt01    Adventureworks2014                       Simple
hqdbt01    Adventureworks2014_DataCompress          Simple
hqdbt01    ApexSQLCrd                               Simple
hqdbt01    PowerSQL                                 Simple
hqdbt01    ReportServerTempDB                       Simple
hqdbsp20   OperationsManager                        Simple
hqdbsp20   OperationsManagerAC                      Simple
hqdbsp20   OperationsManagerDW                      Simple
```

  The changed recovery model for available database

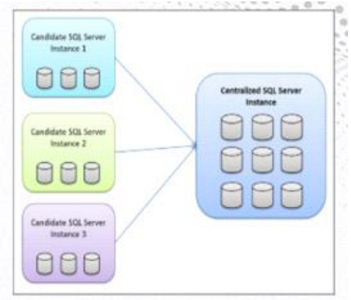
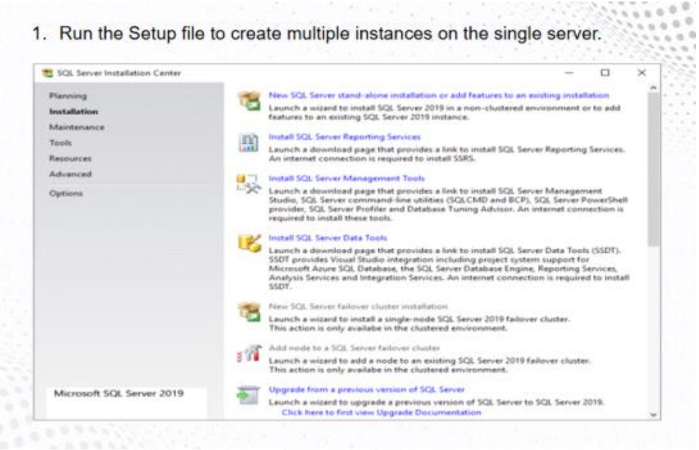
- MSSQL Index
  - Index will have keys+address
  - Clustered index
    - Used to sort and store the data
    - Clustered index keys serves as row finders
  - Non-clustered index
    - Row locator is pointer to the row
  - Table scan
    - There is system table called sysindexes that contains the information of indexes on tables
    - If no index, then there will be one row which says that there is no index on table
    - When select query is given sql server refere 'indi/index'

  - Index creation
    - 

      • create [unique][clustered /non clustered] index :

```
<indexname> on <object name>(<column list>)
[include(<columnlst>)]
[with fillfactor=<n>]

By default an index is non-clustered.

create index DNoINdex on Emp(DeptNo)
```

  - Single and composite indexes
    - If index is created on single column it is simple index
    - If index is created on conjuction of two tables it is composite table
    - Create index notedidx on emp(deptno asc,job desc)
    - Sp_helpindex emp - to find already created examples
    - Drop index indexname on objectname
  - Index used by optimizer
    - Indexes that are well designed an reduce I/O operation and use fewer system resourses
    - Query optimizer considers all available data retrieval methods and chooses the best one
    - The approach can be table can or scan of one or more indexes if exists
    - Query optimizer scans every row in table during a table scan and extracts the row that satisfies the query requirements(it uses lot of resources and numerous disk IO operations)
    - When query optimizer uses indexes , it search the index key columns for the storage location of row data

        - Index search is better than table search , unlike table index frequemtly have very few columns per row and rows are sorted
    - Monitor index maintenance and update statistics
        - Remove index index fragmentation
            - A heavily fragmented index might degrade the performance because it requires additional IO. We can prform reorg or rebuild on index based on its fragmentation values
        - Remove unused indexes
            - An unused index also consumes storage and increases maintenance overhead
        - Below 30% fragmentation we opt for reorganize
        - Above 30% we opt for rebuild
        - Whenever we go for reorganize /rebuild -update statistics is mandatory
        - If update statistics is not performed query optimizer might produce incorrect execution plan
    - Index selection and query optimizer
        - Query optimizer uses indexes to produce better execution plan - index selection is the most significant approach in query optimization
        - Unused indexes should be removed
        - Sys.dm_db_index_usage_stats DMV tool: gives the existing unused indexes
- MSSQL ManagingInstance
    - An instance of the database engine is copy of sqlserver.exe executable that runs as an operating system service((copy of serv er files,databases and security credentials)
    - Each instances manages several system databases and one or more user databases.
    - Each computer can run multiple instances of database engines
    - 
Candidate SQL Server Instance 1
Candidate SQL Server Instance 2
Candidate SQL Server Instance 3
Centralized SQL Server Instance
    - Multiple instances in sql server
        - Well protected from one another,don't interfere and each instance has own databases ,logins and security
        - Multiple instances gives the ability to spread load over tempdb
    - Facts to choose multiple instance
        - Memory usage
        - Tempdb disk usage
        - Cpu usage
        - Io
        - Ease of administration
        - Cost of liecensing
        - Monitoring
    - Steps to create multiple instances in SQL server
        - 

1. Run the Setup file to create multiple instances on the single server.

SQL Server Installation Center

New SQL Server stand-alone installation or add features to an existing installation
Launch a wizard to install SQL Server 2019 in a non-clustered environment or to add features to an existing SQL Server 2019 instance.

Install SQL Server Reporting Services
Launch a download page that provides a link to install SQL Server Reporting Services. An internet connection is required to install SSRS.

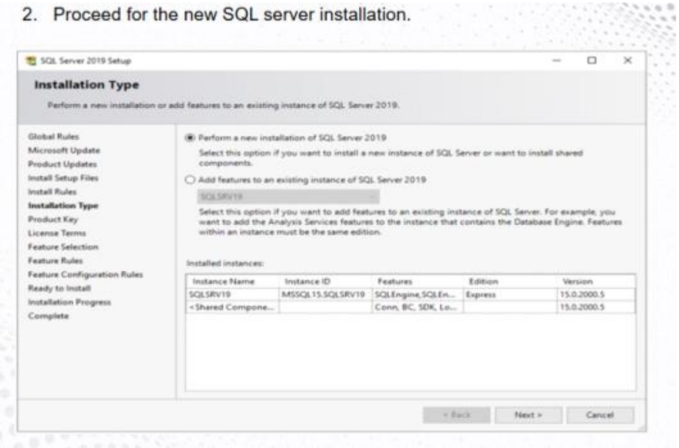
Install SQL Server Management Tools
Launch a download page that provides a link to install SQL Server Management Studio, SQL Server command-line utilities (SQLCMD and BCP), SQL Server PowerShell provider, SQL Server Profiler and Database Tuning Advisor. An internet connection is required to install these tools.

Install SQL Server Data Tools
Launch a download page that provides a link to install SQL Server Data Tools (SSDT). SSDT provides Visual Studio integration including project system support for Microsoft Azure SQL Database, the SQL Server Database Engine, Reporting Services, Analysis Services and Integration Services. An internet connection is required to install SSDT.

New SQL Server failover cluster installation
Launch a wizard to install a single-node SQL Server 2019 failover cluster. This action is only availabe in the clustered environment.

Add node to a SQL Server failover cluster
Launch a wizard to add a node to an existing SQL Server 2019 failover cluster. This action is only availabe in the clustered environment.

Upgrade from a previous version of SQL Server
Launch a wizard to upgrade a previous version of SQL Server to SQL Server 2019.
Click here to first view Upgrade Documentation

        - 

2. Proceed for the new SQL server installation.

SQL Server 2019 Setup

Installation Type

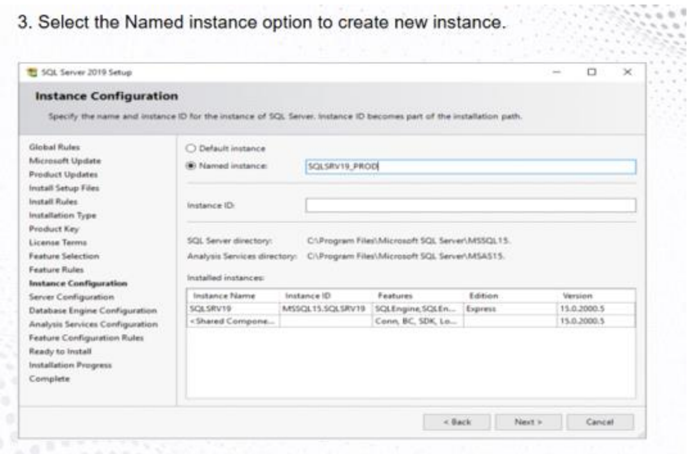
Perform a new installation or add features to an existing instance of SQL Server 2019.

● Perform a new installation of SQL Server 2019
Select this option if you want to install a new instance of SQL Server or want to install shared components.

○ Add features to an existing instance of SQL Server 2019
SQLSRV19
Select this option if you want to add features to an existing instance of SQL Server. For example, you want to add the Analysis Services features to the instance that contains the Database Engine. Features within an instance must be the same edition.

Installed instances:

Instance Name	Instance ID	Features	Edition	Version
SQLSRV19	MSSQL15.SQLSRV19	SQLEngine,SQLEn...	Express	15.0.2000.5
<Shared Compone...		Conn, BC, SDK, Lo...		15.0.2000.5

- 
  3. Select the Named instance option to create new instance.

  SQL Server 2019 Setup

  Instance Configuration

  Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.

  ○ Default instance

  ● Named instance: SQLSRV19_PROD

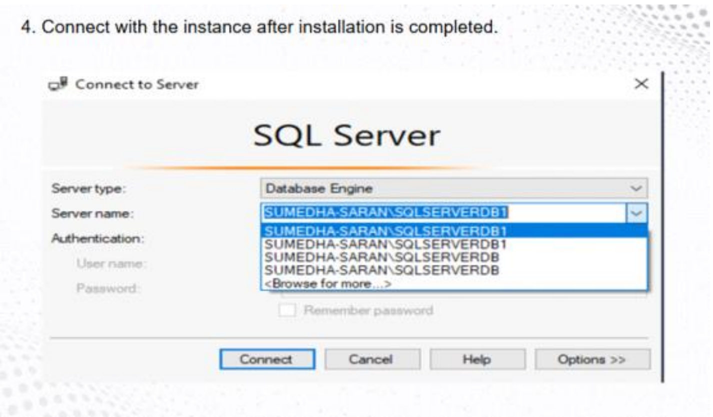
  Instance ID:

  SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL15.

  Analysis Services directory: C:\Program Files\Microsoft SQL Server\MSAS15.

  Installed instances:

Instance Name	Instance ID	Features	Edition	Version
SQLSRV19	MSSQL15.SQLSRV19	SQLEngine,SQLEn...	Express	15.0.2000.5
<Shared Compone...		Conn, BC, SDK, Lo...		15.0.2000.5

- 
  4. Connect with the instance after installation is completed.

  Connect to Server

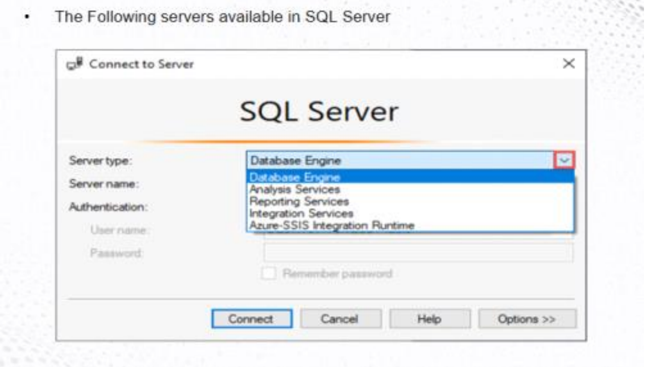
  SQL Server

  Server type: Database Engine

  Server name: SUMEDHA-SARAN\SQLSERVERDB1

  SUMEDHA-SARAN\SQLSERVERDB1
  SUMEDHA-SARAN\SQLSERVERDB1
  SUMEDHA-SARAN\SQLSERVERDB
  SUMEDHA-SARAN\SQLSERVERDB
  <Browse for more...>

  - Types of server
    - Database engine : core component of sql server instance ,used for storing,processing and securing
    - Analysis services - SSAS :is a multidimensional analysis tool that features OLAP , powerful data mining abilities and reporting used in business intelligence to ease your managing data tasks
    - Reporting servives - SSRS is a reporting and visualization tool for server , and by using it , one can create ,manage and publish reports and dashboards
    - Integration services - SSIS is fast and flexible data warehousing tool that can be used for data extraction and executing a wide range of data migration task
    - Azure SSIS integration runtime is the compute infrastructure used by azure data factory to provide the data integration capabilities across different network environments

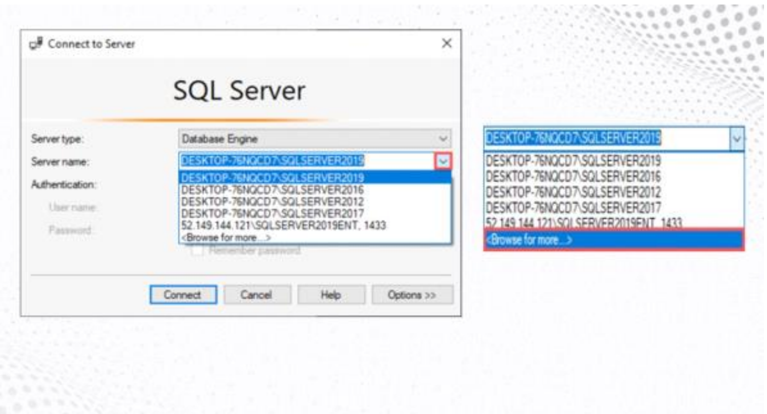
- 
  - The Following servers available in SQL Server

  Connect to Server

  SQL Server

  Server type: Database Engine

  Database Engine
  Analysis Services
  Reporting Services
  Integration Services
  Azure-SSIS Integration Runtime

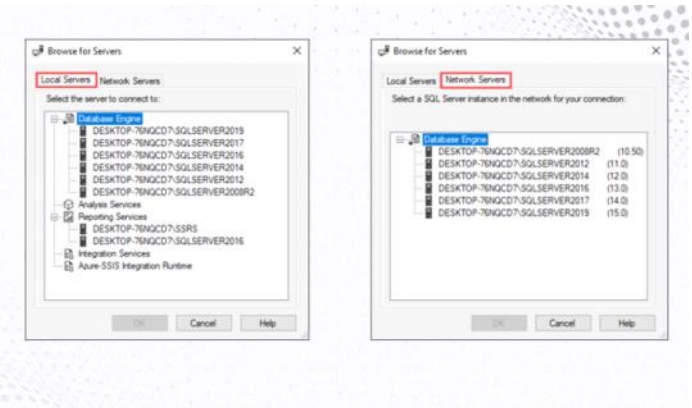
- 
  Connect to Server

  SQL Server

  Server type: Database Engine

  Server name: DESKTOP-76NQCD7\SQLSERVER2019

  DESKTOP-76NQCD7\SQLSERVER2019
  DESKTOP-76NQCD7\SQLSERVER2016
  DESKTOP-76NQCD7\SQLSERVER2012
  DESKTOP-76NQCD7\SQLSERVER2017
  52.149.144.121\SQLSERVER2019ENT, 1433
  <Browse for more...>

- 

  Browse for Servers

  Local Servers | Network Servers

  Select the server to connect to:

  Database Engine
  DESKTOP-76NQCD7\SQLSERVER2019
  DESKTOP-76NQCD7\SQLSERVER2017
  DESKTOP-76NQCD7\SQLSERVER2016
  DESKTOP-76NQCD7\SQLSERVER2014
  DESKTOP-76NQCD7\SQLSERVER2012
  DESKTOP-76NQCD7\SQLSERVER2008R2
  Analysis Services
  Reporting Services
  DESKTOP-76NQCD7\SSRS
  DESKTOP-76NQCD7\SQLSERVER2016
  Integration Services
  Azure-SSIS Integration Runtime

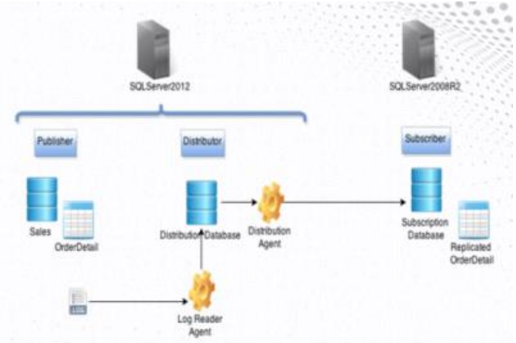
  Browse for Servers

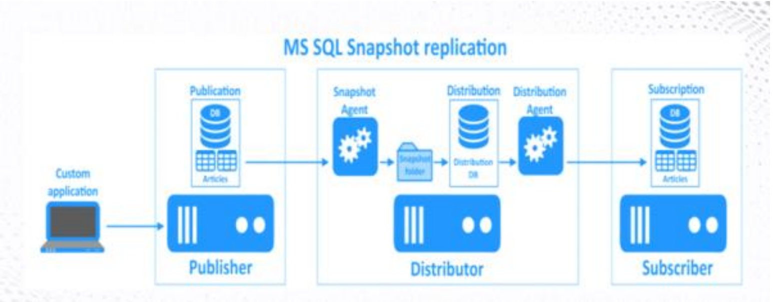
  Local Servers | Network Servers

  Select a SQL Server instance in the network for your connection:

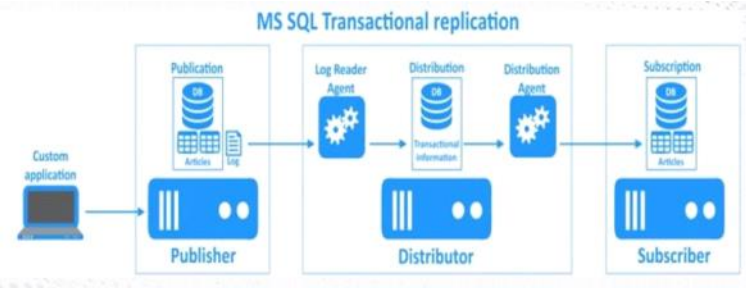
  Database Engine
  DESKTOP-76NQCD7\SQLSERVER2008R2 (10.50)
  DESKTOP-76NQCD7\SQLSERVER2012 (11.0)
  DESKTOP-76NQCD7\SQLSERVER2014 (12.0)
  DESKTOP-76NQCD7\SQLSERVER2016 (13.0)
  DESKTOP-76NQCD7\SQLSERVER2017 (14.0)
  DESKTOP-76NQCD7\SQLSERVER2019 (15.0)

  - SQl server components
    - Windows authentication :this mode allows to connect through windows user account
    - SQL server aunthentication : anytime this mode is used proper login and password is required
    - Azure active directorty
      - Universal with MFA :this mode delivers strong authentication with a different verification options user can choose from - phone call,text message,smart cards with a pin or mobile app notifications
      - Password : use this mode connecting to sql server instance database and when logged into windows using azure AD credentials from a domain not combined with azure
      - Integrated : this mode is used when connecting to sql database and logged into windows using azure active directory credentials from federated domain

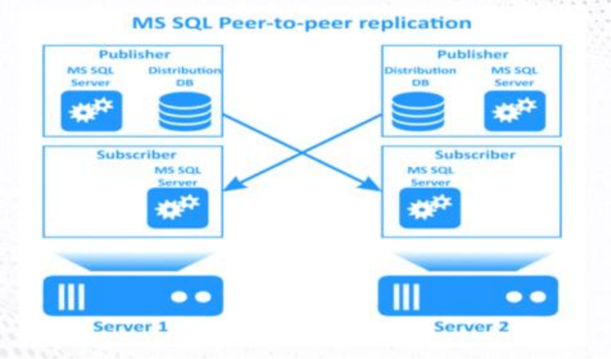
- MSSQL Replication Overview
  - Replication of group of techniques for transferring data and database objects between databases while synchronizing them to preserve consistency
  - Replication is used to distribute data to several locations as well as remote or mobile users via local and wide area networks, dial-up connections, wireless connections, and the internet
  - Sql server replication is used for copying and synchronizing data continuously or it can be scheduled to run at predetermined intervals
  - There are several different replication techniques that support a variety of data synchronization approches :
    - One-way
    - One-to-many
    - Many-to-one
    - Bi-directional
    - Note :Also keep several datasets in sync with each other
  - Transactional replication is commonly applied in server -to-server settings requiring high throughout ,such as:
    - Improving scalability and availability
    - Data warehousing and reporting
    - Integrating data from multiple sites
    - Integrating heterogeneous data
    - Offloading batch processing
  - Main roles in replication : distributor ,publisher,subscriber
    - 

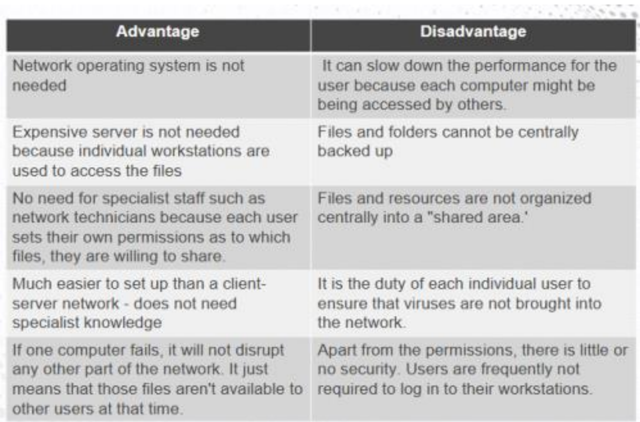
      SQL Server Replication main roles and Components
  - Benefits of replication
    - Improves performance and reduces locking conflicts
    - High availability
    - To work with its own copy of data,each location can have own rules and procedures
  - Snapshot replication : replicates the data exactly as it is appeared at the time of database snapshot taken
    - 

    - Advantages :
      - Simple to set up
      - Low maintenance
      - Database tables need not have primary keys
    - Disadvt:
      - Modifications made to the subscriber are lost after the new snapshot
      - Snapshot agent generates high impact when it runs
  - Transactional replication:
    - Periodic automated replication of data from a master database to a database replica in real time

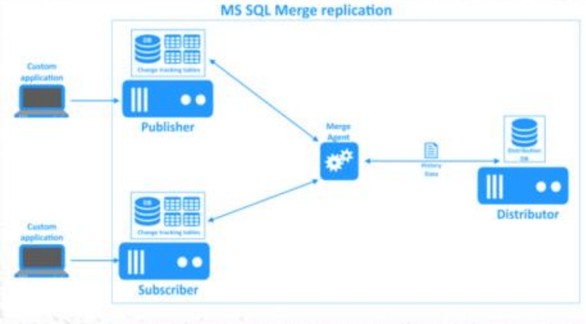
- 

- Advantages:
    - Suitable for high transactional data
    - Requires low latency
- Disadvantages:
    - Setting up transactional replication can be challenging risk,especially for beginners
    - It is prone to faults and can be affected events such as primary key violation

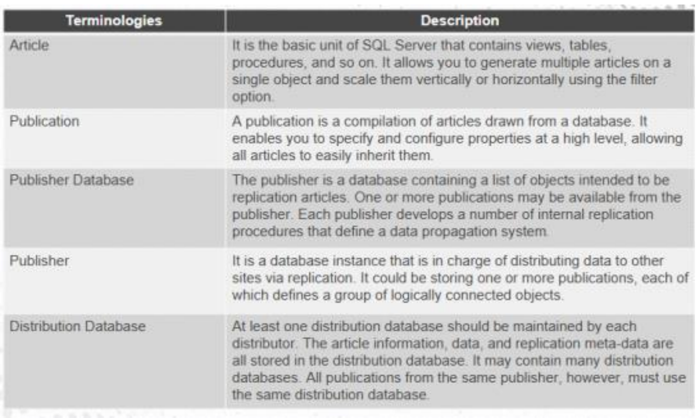
- Peer- to-peer replication :simultaneously replicating the databse data to numerous subscribers
    - 

    - 

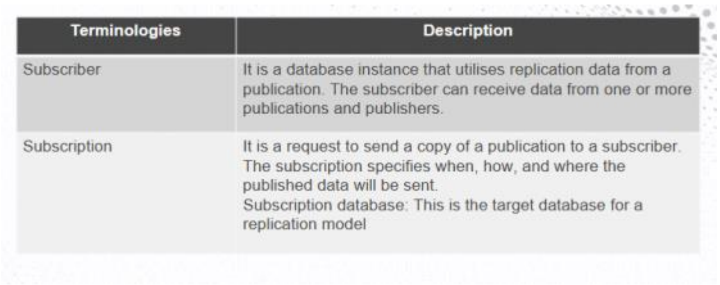
Advantage	Disadvantage
Network operating system is not needed	It can slow down the performance for the user because each computer might be being accessed by others.
Expensive server is not needed because individual workstations are used to access the files	Files and folders cannot be centrally backed up
No need for specialist staff such as network technicians because each user sets their own permissions as to which files, they are willing to share.	Files and resources are not organized centrally into a "shared area.'
Much easier to set up than a client-server network - does not need specialist knowledge	It is the duty of each individual user to ensure that viruses are not brought into the network.
If one computer fails, it will not disrupt any other part of the network. It just means that those files aren't available to other users at that time.	Apart from the permissions, there is little or no security. Users are frequently not required to log in to their workstations.

- Merge applicatin : bi-directional replication that is commonly used in server-to-client contexts to synchronize data between database servers that are non continuously linked
    - 

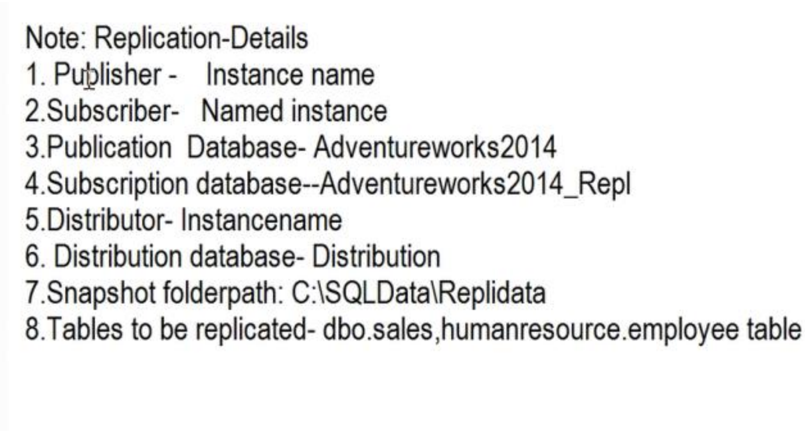
    - Advantages:
        - Let subscribers update records
        - It allows offline subscribers to make change to data
    - Disadvantages:
        - Setting up is risk
        - It is prone to create conflicts between publisher and subscribers
- Requirements for sql server replication
    - The following ports should be opened for inbound traffic
        - Tcp 1433,1434,2383,2382,,135,80,443
        - Udp 1434
        - Before installing msmsql server,configure windows firewall and open the relevant poerts for inbound traffic on each host
        - Before setting mssql replication,following softwares should be installed
            - .net framework - a set of libraries
            - Msmsql server- the database server software
            - Mssql server management studio (ssms)
- Terminologies in sql

- 

Terminologies	Description
Article	It is the basic unit of SQL Server that contains views, tables, procedures, and so on. It allows you to generate multiple articles on a single object and scale them vertically or horizontally using the filter option.
Publication	A publication is a compilation of articles drawn from a database. It enables you to specify and configure properties at a high level, allowing all articles to easily inherit them.
Publisher Database	The publisher is a database containing a list of objects intended to be replication articles. One or more publications may be available from the publisher. Each publisher develops a number of internal replication procedures that define a data propagation system.
Publisher	It is a database instance that is in charge of distributing data to other sites via replication. It could be storing one or more publications, each of which defines a group of logically connected objects.
Distribution Database	At least one distribution database should be maintained by each distributor. The article information, data, and replication meta-data are all stored in the distribution database. It may contain many distribution databases. All publications from the same publisher, however, must use the same distribution database.

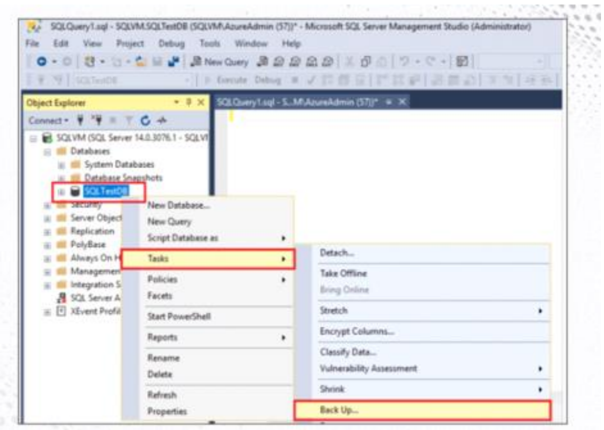
- 

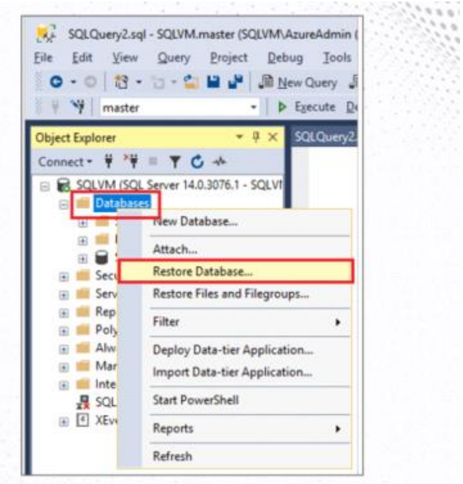
Terminologies	Description
Subscriber	It is a database instance that utilises replication data from a publication. The subscriber can receive data from one or more publications and publishers.
Subscription	It is a request to send a copy of a publication to a subscriber. The subscription specifies when, how, and where the published data will be sent. Subscription database: This is the target database for a replication model

  - 

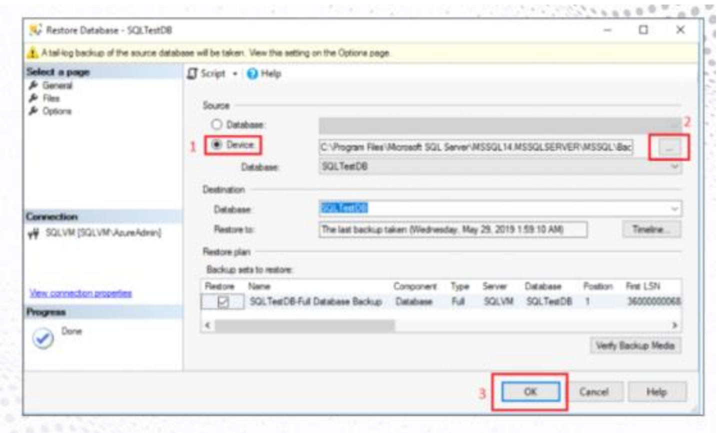
    Note: Replication-Details
    1. Publisher - Instance name
    2.Subscriber- Named instance
    3.Publication Database- Adventureworks2014
    4.Subscription database--Adventureworks2014_Repl
    5.Distributor- Instancename
    6. Distribution database- Distribution
    7.Snapshot folderpath: C:\SQLData\Replidata
    8.Tables to be replicated- dbo.sales,humanresource.employee table

- MSSQL BACKUP RECOVERY
  - Backing up is the process of making duplicates of crucial data on computer .
  - Process of restoring deleted or damaged data from backups is called recovery
  - Copies of Backups should be stored in secured,off-site place
  - Back and recovery strategy should be made and efficiently and tested properly,then only we can implement or use it
  - Best practice recommendations
    - Use separate storage
    - Choose appropriate recovery model
    - Design backup strategy
    - Estimate the size of full databse backup
    - Schedule bacjups
    - Test your backups
    - Verify media stability and consistency
    - Document backup/recovery strategy
    - Monitor progress with xevent
  - Types of backups
    - Full backups
    - Differential backups
    - File backups
    - Filegroup backups
    - Partila backups
    - Copy-only backups
    - Mirror backups
    - Transaction log backups

- 

- 
SQLQuery2.sql - SQLVM.master (SQLVM\AzureAdmin (
File Edit View Query Project Debug Tools
New Query
master
Execute
Object Explorer
Connect
SQLVM (SQL Server 14.0.3076.1 - SQLVM
Databases
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Filter
Deploy Data-tier Application...
Import Data-tier Application...
Start PowerShell
Reports
Refresh

  Restore the Database from the available backup

- 

  A tail-log backup of the source database will be taken. View this setting on the Options page.

  Source
  ○ Database:
  ● Device: C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Bac
  Database: SQLTestDB

  Destination
  Database: SQLTestDB
  Restore to: The last backup taken (Wednesday, May 29, 2019 1:59:10 AM)

  Restore plan
  Backup sets to restore:

Restore	Name	Component	Type	Server	Database	Position	First LSN
☑	SQLTestDB-Full Database Backup	Database	Full	SQLVM	SQLTestDB	1	36000000068

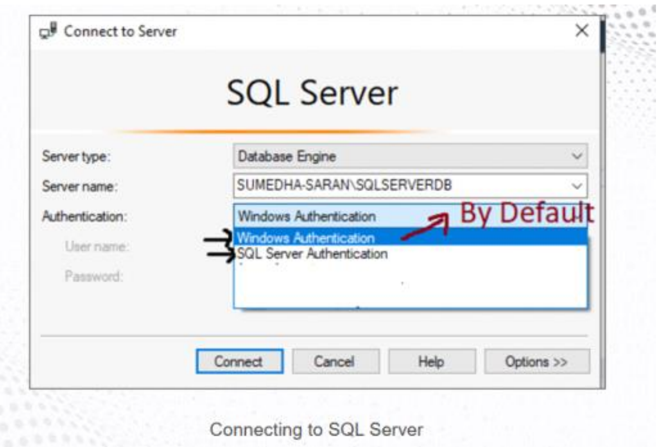
  Connection: SQLVM [SQLVM\AzureAdmin]

  Select the backup for the Database by selecting the device

- Clean resources :
  - Run the following t-sql command to remove the databse you created , along with its backup history in the msdb databse
    - Exec msdb.dbo.sp_delete_database_backuphistory @database_name=N'SQL TESTDB'

➢ SQL ARCHITECTURE MANAGEMENT

- Sql server architecture management
  - Sql database is created with one or more data files (.mdf and .ndf) and one transaction log file(.ldf). The data files includes the schema and data ,whereas the log file records the most recent modifications
  - Data is organized into pages (8kb)
  - Sql server - read,writes and modifies
- Data retrieval
  - Sql server accesses data by reading a complete page from disc and storing it in memory(RAM)
  - Pages are kept in memory until they are no longer needed
  - As sql works on same data set,same data page is regularly chnaged or read
- Data modification
  - Sql server uses table based on rows and columns to connect data(data consistency and security)
  - After each db modification - transaction occurance,data will be saved , and a log record will be written into the transaction log . All the chnages will be written sequentially
- Sql server authentication and authorization
  - Supports two authentication modes
    - Windows authentication
    - Sql server authentication

- 

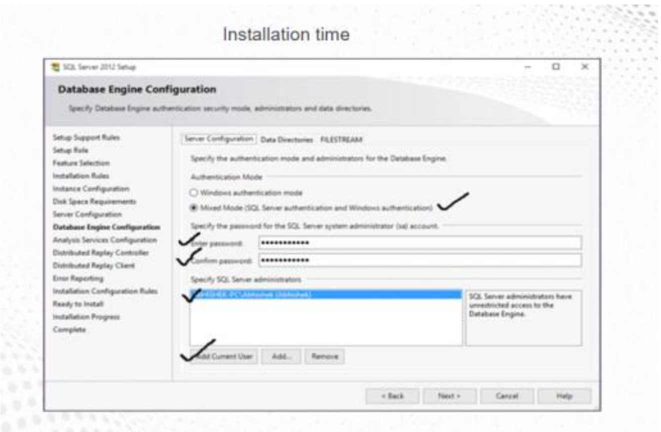
  Connect to Server

  SQL Server

  Server type: Database Engine

  Server name: SUMEDHA-SARAN\SQLSERVERDB

  Authentication: Windows Authentication

  Connecting to SQL Server
- Windows authentication is integrated with windows user and group accounts,making it possible to use domain or local windows account to log into sql server
- When a windows user connects to sql server ,the db engine validates the login credentials against windows principalauthentication,eleminating the need of separate sql credentials
- Setting up authentication:
  - Installation time
  - 

    Installation time

    SQL Server 2012 Setup

    Database Engine Configuration

    Specify Database Engine authentication security mode, administrators and data directories.

    Specify the authentication mode and administrators for the Database Engine.

    Authentication Mode

    ○ Windows authentication mode

    ● Mixed Mode (SQL Server authentication and Windows authentication) ✓

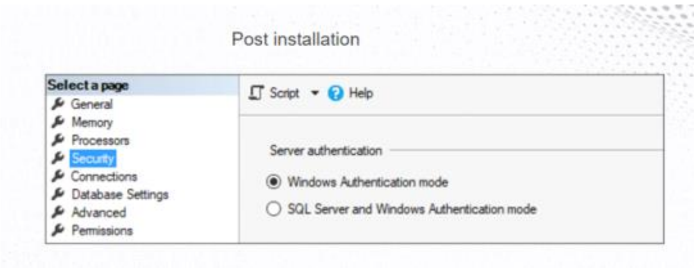
    Specify the password for the SQL Server system administrator (sa) account.

    ✓ Enter password: ••••••••••••

    ✓ Confirm password: ••••••••••••

    Specify SQL Server administrators

    SQL Server administrators have unrestricted access to the Database Engine.

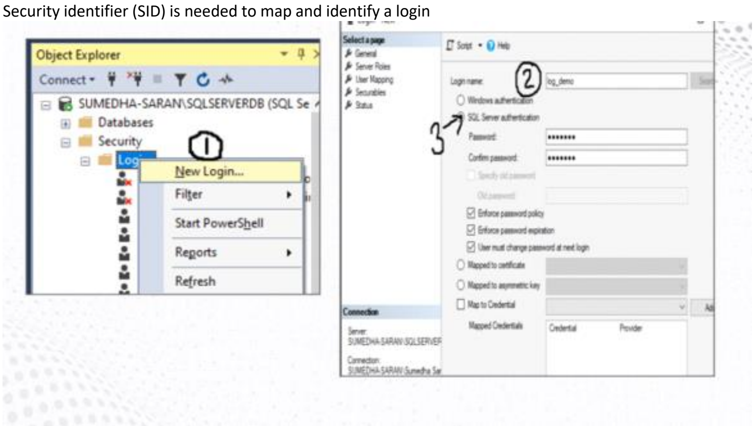
  - Post installation
  - 

    Post installation

    Server authentication

    ● Windows Authentication mode

    ○ SQL Server and Windows Authentication mode

- Sql server login
  - Security identifier (SID) is needed to map and identify a login
  - 

    1 New Login...

    2 Login name: log_demo

    ○ Windows authentication

    3 ● SQL Server authentication

    Password: •••••••

    Confirm password: •••••••

    □ Specify old password

    Old password:

    ☑ Enforce password policy

    ☑ Enforce password expiration

    ☑ User must change password at next login

    ○ Mapped to certificate

    ○ Mapped to asymmetric key

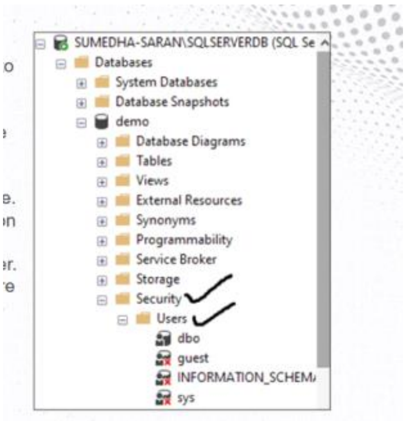
    □ Map to Credential

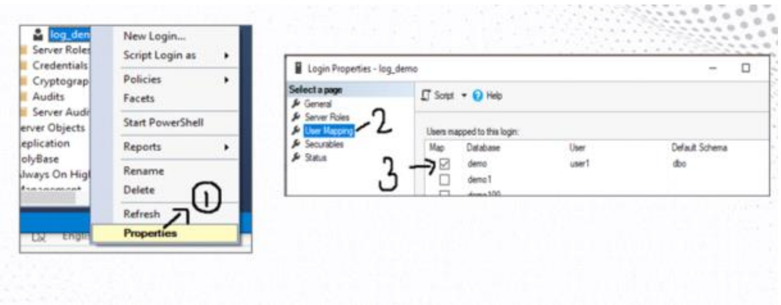
    Mapped Credentials

    Server: SUMEDHA-SARAN\SQLSERVER

    Connection: SUMEDHA-SARAN\Sumedha Sa

  - Server user is different and databse user is different

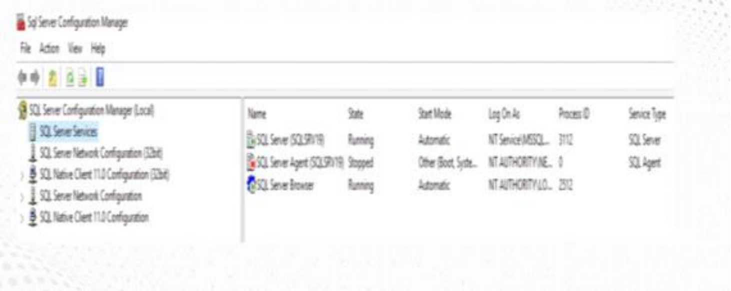
- 

- 

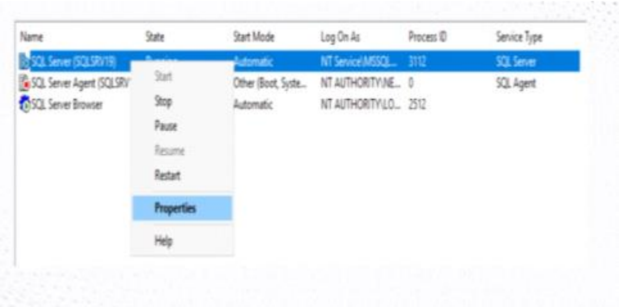
Map	Database	User	Default Schema
☑	demo	user1	dbo
□	demo1		
□	demo100		

    - ○ Sql server permissions
        - Server level permissions
            - ○ It is assigned to login and server roles
        - Database level
            - ○ Is is assigned to databse user and database roles

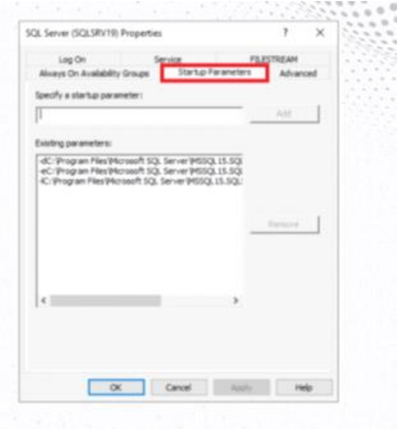
➢ MSSQL CONFIGURATION MANAGER

- ○ It is used for
    - Managing network access for sql server client computers
    - Configuring the network protocols used by sql server
    - Managing the services connected to sql server
    - 
Sql Server Configuration Manager
File
Action
View
Help
SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
SQL Server Network Configuration
SQL Native Client 11.0 Configuration

Name	State	Start Mode	Log On As	Process ID	Service Type
SQL Server (SQLSRV19)	Running	Automatic	NT Service\MSSQL...	3112	SQL Server
SQL Server Agent (SQLSRV19)	Stopped	Other (Boot, Syste...	NT AUTHORITY\NE...	0	SQL Agent
SQL Server Browser	Running	Automatic	NT AUTHORITY\LO...	2512	

    - 

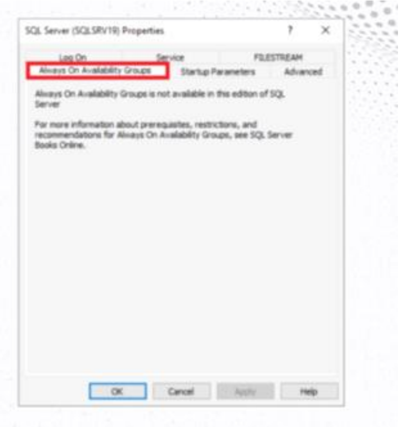
Name	State	Start Mode	Log On As	Process ID	Service Type
SQL Server (SQLSRV19)	Running	Automatic	NT Service\MSSQL...	3112	SQL Server
SQL Server Agent (SQLSRV		Other (Boot, Syste...	NT AUTHORITY\NE...	0	SQL Agent
SQL Server Browser		Automatic	NT AUTHORITY\LO...	2512	

    - To change sql server startup parameters
        - ○ 

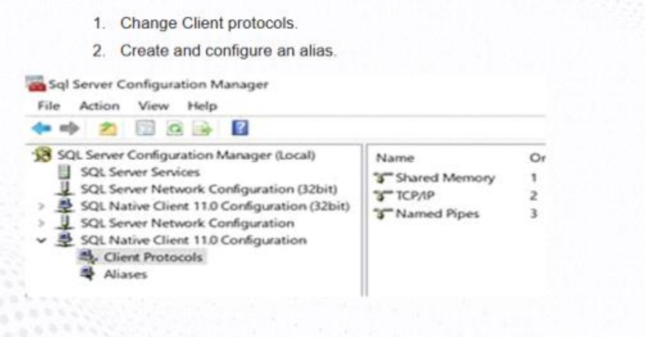
Specify a startup parameter:

Existing parameters:

```
-dC:\Program Files\Microsoft SQL Server\MSSQL15.SQ
-eC:\Program Files\Microsoft SQL Server\MSSQL15.SQ
-lC:\Program Files\Microsoft SQL Server\MSSQL15.SQL
```

- To enable always on
  - 

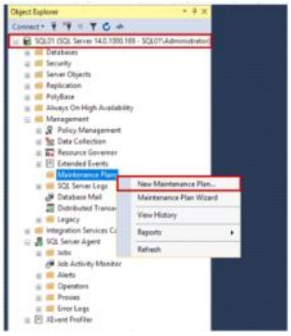
    Always On Availability Groups is not available in this edition of SQL Server

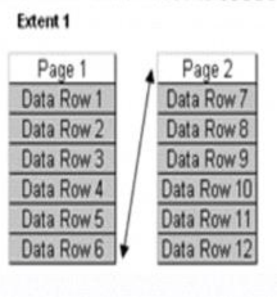
    For more information about prerequisites, restrictions, and recommendations for Always On Availability Groups, see SQL Server Books Online.
- Native client configuration
  - 

    1. Change Client protocols.
    2. Create and configure an alias.

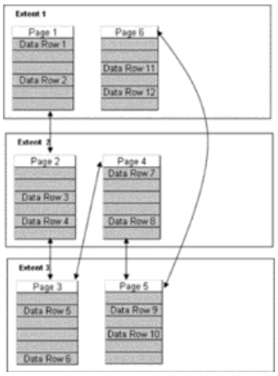
    Sql Server Configuration Manager
File
Action
View
Help
SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
SQL Server Network Configuration
SQL Native Client 11.0 Configuration
Client Protocols
Aliases

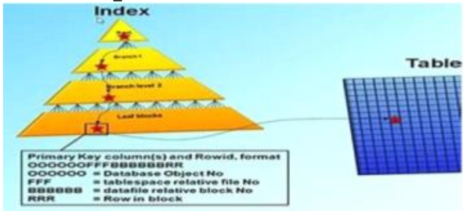
Name	Or
Shared Memory	1
TCP/IP	2
Named Pipes	3
  - Clients can connect to the sql server uing below
    - Tcp/ip
    - Named pipes
      - These are used for inter-process communication and is configured for a local area network
    - Shared memory
      - Clients use the shared memory protocol to connect to sql server instance on the same server
      - Basic protocol and no configuration options

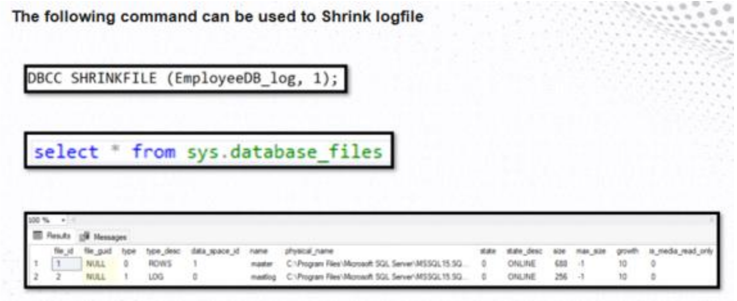
- MSSQL DATABASE MAINTENANCE
  - Database maintenance
    - A set of tasks that are all run with the intention to improve your database , these are routines meant to help performance ,free up disk space,check for data errors, check for hardware faults, update internal statistics
    - Data maintenance have four main categories
      - 

      - Index defragmentation
        - A data page in both clustered or non-clustered indexes contains pointers to previous and next page
        - 

        - Logical fragmentation - the logical order of the pages does not correspond their physical order
        - Internal fragmentation - the data pages in the index contain free space. This led to an increase in the number of logical reads during the query execution

- 

- Detecting fragmentation :sys.dm_db_index_physical_stats data management function analyse fragmentation
    - Avg_page_space_used_in_percent shows the average percentage of data storage space used on the page
    - Avg_fragmentation_in_percent - external index fragmentation
    - For tables with clustered indexes fragment_count indicates how many continuous data fragments the index has
- Removing index fragmentation
    - Alter index all on tablename reorganize
    - Alter index idex_name on tablename rebuild
    - 
Index
Table
Primary Key column(s) and Rowid, format
OOOOOOFFFBBBBBBRR
OOOOOO = Database Object No
FFF = tablespace relative file No
BBBBBB = datafile relative block No
RRR = Row in block

- Log file maintenance
    - When maintaining a databses's transaction log we'll often want to retrieve information about the log so we can verify its settings or track how much log space is being used
    - Select name,size,max_size, growth, is_percent_growth from sys.database_files where type_desc='LOG'
    - Backing up the log file
        - DBCC SQLPERF - check log file details
        - BACKUP LOG employeeDB TO DISK ='E:\ogBackup\employeeDB_log.bak';
    - 

The following command can be used to Shrink logfile

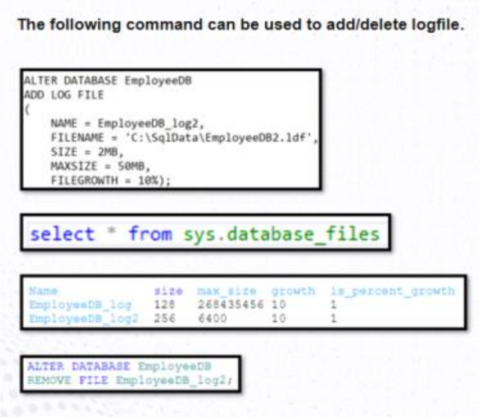
```
DBCC SHRINKFILE (EmployeeDB_log, 1);
```

```
select * from sys.database_files
```

100 %

Results Messages

	file_id	file_guid	type	type_desc	data_space_id	name	physical_name	state	state_desc	size	max_size	growth	is_media_read_only
1	1	NULL	0	ROWS	1	master	C:\Program Files\Microsoft SQL Server\MSSQL15.SQ...	0	ONLINE	688	-1	10	0
2	2	NULL	1	LOG	0	mastlog	C:\Program Files\Microsoft SQL Server\MSSQL15.SQ...	0	ONLINE	256	-1	10	0

    - 

The following command can be used to add/delete logfile.

```
ALTER DATABASE EmployeeDB
ADD LOG FILE
(
    NAME = EmployeeDB_log2,
    FILENAME = 'C:\SqlData\EmployeeDB2.ldf',
    SIZE = 2MB,
    MAXSIZE = 50MB,
    FILEGROWTH = 10%);
```

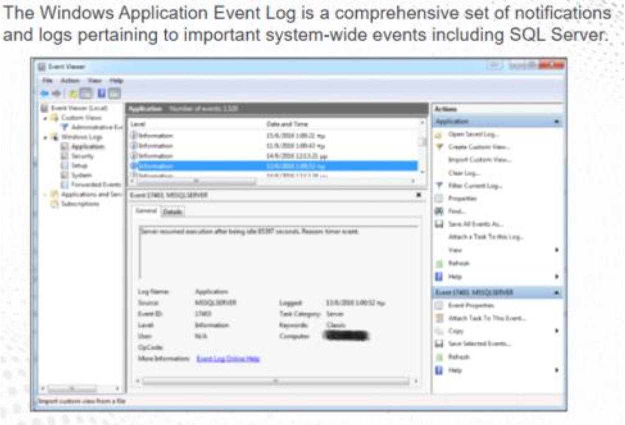
```
select * from sys.database_files
```

```
Name             size  max_size  growth  is_percent_growth
EmployeeDB_log   128   268435456 10      1
EmployeeDB_log2  256   6400      10      1
```

```
ALTER DATABASE EmployeeDB
REMOVE FILE EmployeeDB_log2;
```

- File/data compaction
    - When we work with sql db, the file size both grows and shrinks. Each time the file needs more space, it grows by a certain increment.And as data is added to the db (as a chunk), it is saved into any spot that has enough memory to store that chunk . Unfortunately , the data is not always stored close to other data from the same table. The data itself becomes fragmented , just like the indexes
    - To resolve this, we run what is called a ' data compaction' , which reorganizes the file by grouping all associated data together. This not only groups data together, but it can also free up space inside the file as well, which can then be reclaimed by the os as free disk on hard disc
- Integrity check
    - For checking logical and physical integrity of the sql server database /to correct errors run - DBCC CHECKDB
    - Below operations are performed by dbcc checkdb command
        - DBCC CHECKALOOC
        - DBCC CHECKTABLE
        - DBCC CHECKCATALOG
        - DBCC CHECKDB
    - The commands follows below steps
        - Verify the allocated structures such as GAM,SGAM,PFS,IAM
        - Verify the table consistency and its all indexes
        - Finally, verify all the system catalog
- Components of sql server
    - HELP
    - FLUCHAUTHCACHE
    - TRACEOFF

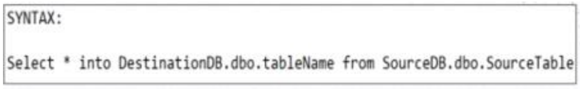
        - FREESESSIONCACHE
        - TRACEON
        - FREESYSTEMCACHE
        - CLONEDATABASE

- MSSQL DATABASE MONITORING
    - Assessing server performance is the main aim of database monitoring
    - Inorder to avoid processes that producing tasks and follow performsnce trends, effective monitoring entails taking frequent snapshots of current performance
    - Server monitor
        - The process of monitoring server's system resources , such as CPU usage ,memory consumption ,I/O ,network ,disc usage, and processes
        - Helps in capacity planning
        - For monitoring server availability and data loss
        - Responsiveness of server
        - To understand the server capacity, user load and speed of the server
        - To detect and prevent any issues that might effect the servee proactively
        - Reason to use sql server instance and db monitoring
            - Capacity planning
            - Monitor current resource usage
        - A good sql server monitoring plan can help you stay on top of performance, growth, security.
    - Sql server monitoring key areas
        - Disk activity
        - Processor utilization
        - Memory
    - Sql server monitoring tools
        - Sql server profiler
        - Sql server management studio reports
        - Activity monitor
        - Sql logs,system monitor
        - Database engine stores procedures
    - Window application log
        - Realted to windows event viewer
        - System log
        - Security log
        - Application log
        - 

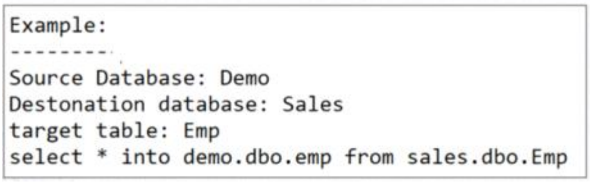
The Windows Application Event Log is a comprehensive set of notifications and logs pertaining to important system-wide events including SQL Server.

    - DBCC(Database concole command)
        - The dbcc commands listed below act on read-only database snapshot that the database creates internally
            - DBCC CHECKALLOC
            - DBCC CHECKATALOG
            - DBCC CHECKDB
            - DBCC CHECKFILEGROUP
            - DBCC CHECKTABLE

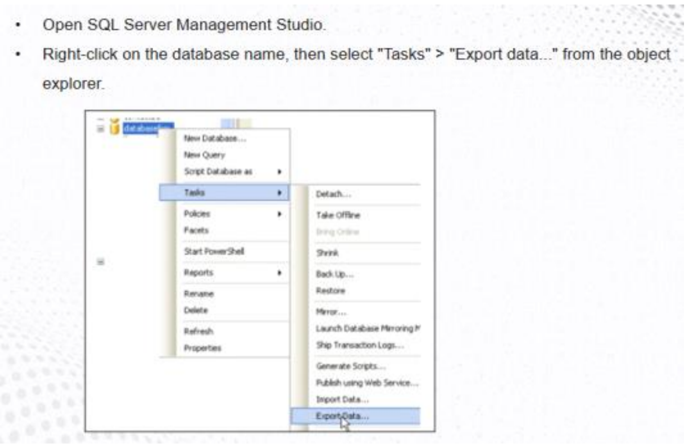
- MSSQL MOVING DATA
    - Transferring data from one db to another db



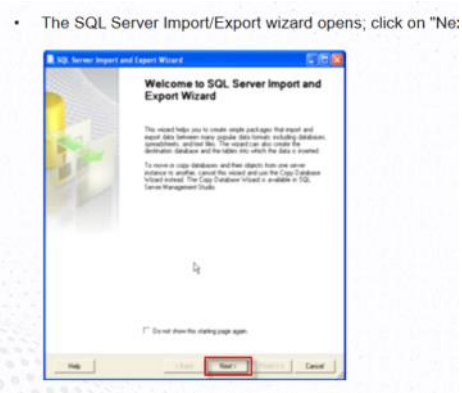
```
SYNTAX:

Select * into DestinationDB.dbo.tableName from SourceDB.dbo.SourceTable
```

    - 

```
Example:
--------
Source Database: Demo
Destonation database: Sales
target table: Emp
select * into demo.dbo.emp from sales.dbo.Emp
```

- 

  - Open SQL Server Management Studio.
  - Right-click on the database name, then select "Tasks" > "Export data..." from the object explorer.

- 

  - The SQL Server Import/Export wizard opens; click on "Ne:

  Welcome to SQL Server Import and Export Wizard

  [illegible]

- Data transfer using script
  - Right-click on databse and select task
  - Click on generate scripts
  - Select the database you want to script
  - Choose the ocject types
  - Select the tables
  - Select the output option for the script
  - Change/edit the db name to the name you wan to execute the script for