SQL Server Architecture - DBA Related

<https://www.ourtechideas.com/blog/sql-server-dba-its-architecture-part-1/>

# Database Administration

A database administrator is a person responsible for the database design, implementation, maintenance and repair of the database. The main goal of DBA is to keep the database server always up and make it available to users. In case of any failures, DBA should minimize the Database by implementing powerful backup and restoring technique.

## DBA Responsibilities

**As a DBA we have to perform these tasks,**

1. Maintaining the availability of database by minimizing the down time.
2. Data recovery, we have to minimize the data loss in case of failures by implementing high availability.
3. Provide high security in accessing the databases externally.
4. Need to monitor the performance of server, implement various techniques to increases the performance
5. Regularly monitor database growth, disk space sql server logs, Event viewer logs to avoid issues and to identify any bottlenecks.

## DBA Roles / Daily Activities

1. As part of the DBA team need to provide 24/7 production support to client and users.
2. Work on user requirements and problems that comes in the form of tickets.
3. Responding to alerts that we receive in the form of an email from third-party monitoring tools.
4. Make sure all the maintenance jobs are running successfully.
5. Make sure all backup jobs executed successfully without any issues an all servers.
6. Checking SQL server logs to identify bottlenecks.
7. Checking drive spaces on critical servers to ensure that there is ample amount of space
8. Regularly monitor data file growth, log file growth, database growth as part of capacity planning.
9. Maintain documentation of all the tasks and issues that you encounter for future reference.
10. Check whether all SQL services are running (or) not.

# SQL Server Architecture

SQL Server follows client-server architecture. Whenever the user performs any action on client machine, it converts in the form of a query. This query moves from client to server in the form of network packets using protocols for connection and communication between the source and destination servers.

1. Relational Engine
2. Storage Engine

**Relational Engine (Query Processor)** prepares the execution plan and hand over to the storage Engine.

**Storage Engine** It is a central repository, responsible in the execution of query using execution plan, the response sent to the user.

**Buffer pool** is another important component contains plan cache and data cache which is used for query execution.

**SQL OS** is a core to SQL Server architecture, used for scheduling, I/O completion, Memory Management and resource management. It is a thin layer between windows OS and SQL server.

## Components of SQL Server

### **SQL Server Network Interface (SNI)**

SNI is a Protocol layer that establishes the network connection between the client and the server. It uses TCP/IP protocol to send queries in the form of TOS packets.

### **Command Parser**

Command Parser first checks for syntax errors, then it generates query plan (or) find an existing plan query plan contains detail steps how query is going to execute. Command parser checks whether a plan already exists in plan cache of buffer pool. If finds plan passes to query executor for execution. If it does not find then query passes to optimizer.

### **Optimizer**

Optimizer prepares query plans for one query in that SQL server select best plan based on response time, the query plan passes to query executor for execution.

### **Query Executor**

Query executor requires data to read the query plan it passes to access methods of storage engine.

### **Access Methods**

Access methods require data to complete the query it asks buffer managers to provide data page. Once it receives required data, the query results passes back to the relational engine and there to the user.

### **Buffer Manager**

Buffer manager checks in the data cache of the buffer pool to see if it has the page already in cache memory. If page exists, it passes results to Access methods. If not exists it pulls required pages from mdf data file put in data cache and passes it back to Access methods.

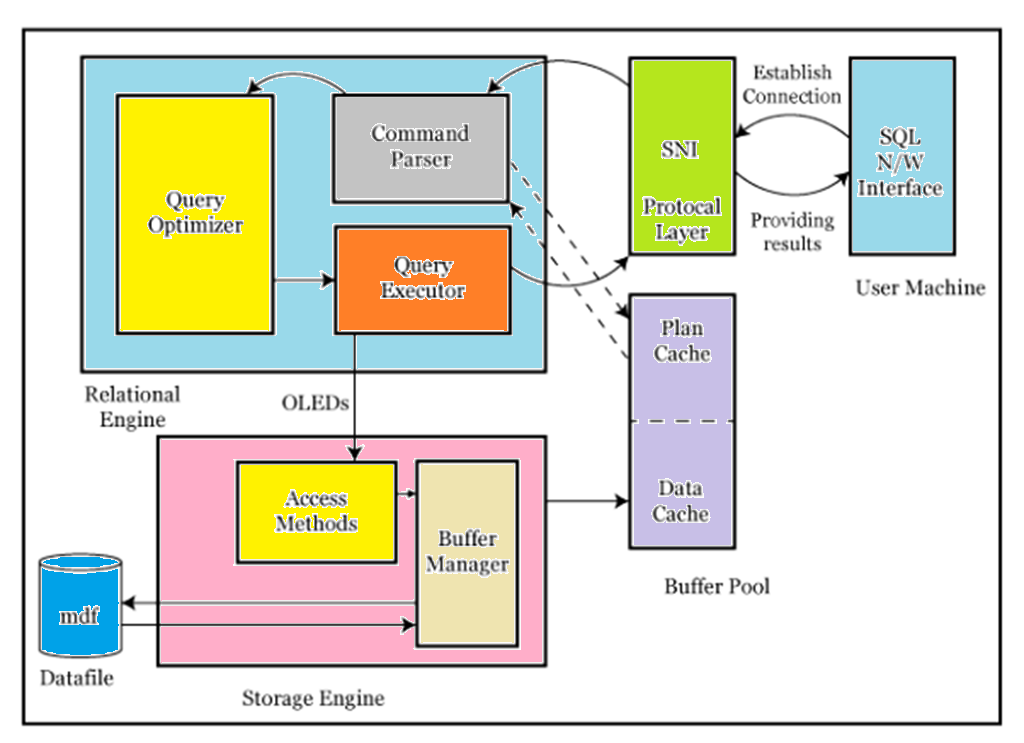
### **Plan Cache**

Part of SQL servers buffer pool used to store previously executed execution plans in case they are needed later.

### **Data Cache**

Data cache is largest part of buffer pool. Every data page that is read from disk is written a copy here before using.Under memory pressure these pages are flushed from cache using LRU (Least recently used) policy.

### **SQL Server Architecture**



### **Steps in executing a query**

1. Server Network Interface (SNI) of the user establishes the connection between client and server using TCP/IP protocol, sends a query in TDS packets.
2. Query at command parser checks syntax errors then checks plan in plan cache of the buffer pool. If the plan not exists, pass the query to the optimizer.
3. The optimizer generates the best plan and passes to the query executor, it reads the plan and passes to access method of storage engine through OLEDB.
4. The access method requests the buffer manager to provide the data.
5. Buffer manager checks in the data cache of the buffer pool for an existing page. If the page not exists it pulls the required pages from the data (MDF) file,  put in the data cache and passes to access method.
6. Finally, the Access method passes the results back to relational engine, from there it sent back to the user who executed the query.

## Protocols available in SQL Server

SQL Server Network Interface (SNI) is a protocol layer that establishes the network connection between the client and the server. SQL Server supports 4 protocols.

1. Shared memory
2. Named pipes
3. TCP/IP
4. VIA

**Shared Memory**: It is default protocol used to connect client and SQL Server on the same machine

**Named Pipes:** Client and server will connect within a LAN. It has certain limitations.

**TCP/IP**: TCP/IP is the most used protocol for SQL Server client establishes a connection with SQL server using an IP Address and a port number 1433 we can access the databases using the internet hence there are no boundaries for this protocol.

**VIA (Virtual Interface Adapter):** VIA is a wireless internet protocol for connecting clients and servers within a certain range.

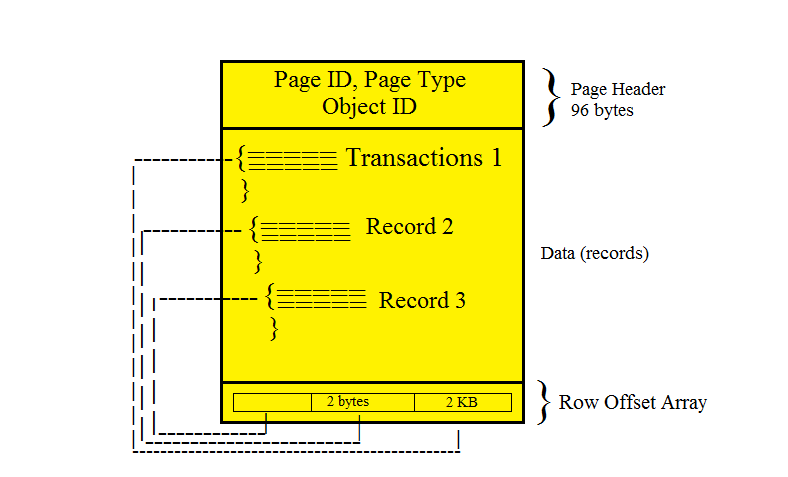
To establish a secure SQL connection we need a port number along with the protocol. The default port number for TCP/IP protocol is 1433 we can change the port number from configuration manager — SQL server network configure — protocols we can change.

<https://www.ourtechideas.com/blog/sql-server-storage-architecture-part-2/>

# Introduction

In SQL server all the data will be stored in the form of records, these records also called row data. All these records further grouped into a page. The page is a default storage unit of the SQL server. The size of the page is 8kb.

# Page Architecture



## Page consists of 3 sections

1. Page Header
2. Actual Data
3. Row offset array

**Page Header –**It consists of Page ID, Page Type, Object ID Header version.

**Page ID** – To identify particular page using unique page ID.

**Page Type** – What type of page it is either data page or Index page.

In Row offset location of record will be stored (2 bytes).

### **Types of Pages**

1. **Data Page** – stores data entered by user.
2. **Index Page** – Indexes are pointer which store address of original pages for quickly locating data
3. **Free space page** – It stores page allocation information and unused space available on pages.
4. **Text/Image** – It stores large object data (LOB) like Text, Image and XML Data.
5. **GAM (Global Allocation Map)** or SGAM (Shared Global Allocation Map) – It stores extent allocation information.
6. **BCM (Bulk Changed Map)** – Stores extents information in a Bulk Operation
7. **DCM (Differential Change Map)** – It stores modified extents information after Full BackUp.
8. **I AM (Index Allocation Map)** – Stores extents information that are used by a table (or) Index.
9. These are important types of pages. All these pages are further grouped into an Extent.

## SQL Server Extents

Extent is a storage structure consists of 8 consecutive SQL Server pages. Pages in a Extent can be one table (or) upto Eight tables.

**There are 2 types of Extents**

1. Uniform Extent: If all pages are going to store same table data
2. Mixed Extent: If the pages shared by 2 (or) more tables.

When a table is created and a row is inserted table gets 1 page in mixed extent, when a table grows then these tables moved to a uniform extent. This is to manage space efficiently.

## SQL Server File

All the extents further group into a File. A file we will have better control in SQL Server.

There are 2 types of files mainly,

1. MDF (Master Data File)
2. LDF (Log Data File)

**MDF** – Stores Permanent Data

**LDF –**Stores changes information will be recorded later these changes apply on MDF Data.

## Database

Files combine to form the database. We require a minimum 2 files 1 MF and 1 LDF to create a database. Maximum we can ‘n’ number of files means No limit.

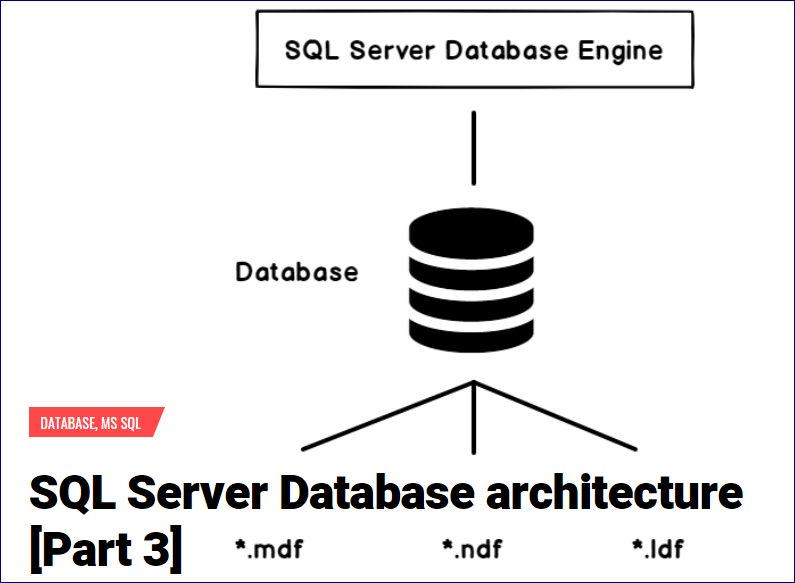
## File Groups

Some files stored system data and some store user database data. Logically dividing databases into groups called File Groups.

## Storage Structure

Records > Pages > Extents > Files > Database

<https://www.ourtechideas.com/blog/sql-server-database-architecture-part-3/>



# Database architecture

SQL Server data mainly in 2 types of files,

1. Data File (MDF)
2. Log File (LDF)

Data file stores actual data with .mdf extension. It stores permanent data. Log files stores modified recorded information with .ldf extension. We have another file called secondary data file .ndf file extension. A database may or may not have these secondary data files.

## Transactions

A Transaction is a set of T-SQL statements that read and wt=rite data into the database. There are 2 types of Transactions.

* Implicit Transactions
* Explicit Transactions

**Implicit Transaction:** Implicit transactions are these without begin transaction

**Explicit Transaction:**Explicit transactions are started using begin transaction and are controlled by using T-SQL Command commit transaction (or) Rollback transaction. Any transaction should process ACID properties then only changes more from LDF to MDF.

## ACID Properties in SQL Server

Atomicity means all the statements of a transaction must complete successfully or rolled back completely means either all (or) None to updated.

**Consistency**

Consistency means a transaction never leaves database in half-finished state, whenever any change happen on parent object it should automatically reflect on dependent child object to ensure that database in a consistent state.

**Isolation**

Isolation keeps changes of incomplete transactions independent from one another.

**Durability**

Once a transaction is committed, it must be permanent even if there is a system failure means it cannot be rolled back.

## Database Architecture

SQL Server stores data mainly in two types of files.

1. **Data File (MDF)**
2. **Log File (LDF**)

## 

**MDF  –**It contains Permanent Data

**LDF** – LDF contains whatever changes we are performing on the database all the change-related information will be recorded in the LDF file.

## SQL Server Buffer

Buffer is a ram to perform modifications on a copy of the permanent page. Once it commits record the information will record in LDF and the same changes apply on MDF when checkpoint runs.

## How Buffer Works

SQL Server will not allow doing modifications directly to MDF. SQL Server will make a copy of pages from MDF to buffer. Once the transaction is fully committed it records the information that what type of data he is inserting, Number of pages affecting, what he is performing all these change-related information will record in the same sequential way in Log File. Pages will stay some time in the buffer for faster retrieval read and write operations from the buffer will be very faster comparing to operations from MDF Data. Using recorded information whenever checkpoint runs on the log file. It applies the same changes permanently on the MDF file.

## SQL Server Checkpoint Process

The checkpoint is an internal mechanism performs regular based on the number of transaction (or) number of pages there is no time interval for running this. Checkpoint scans log file, checks how many committed transactions are there, how many failed and how many still running committed transactions more to MDF, failed transactions will be rolled back. Currently, running transactions will not be touched by the checkpoint.

### **Advantages of the checkpoint in SQL Server,**

1. Checkpoint help in the speeding recovery process
2. Checkpoint helping in committing data permanently

### **SQL Server Recovery Process**

Whenever SQL server restarts checkpoint verifies pending transactions before the restart, the SQL server will perform the recovery process. This process will analyze what is the state of the log file and perform 2 properties.

1. **Redo (or) Roll forward** –  committed changes will be moved from LDF to MDF permanently.
2. **Undo (or) Roll Back** – failed transactions and running transactions will be deleted from log file.

Once this recovery process complete then only users can able to access the database.

## Lazy Writer in SQL Server

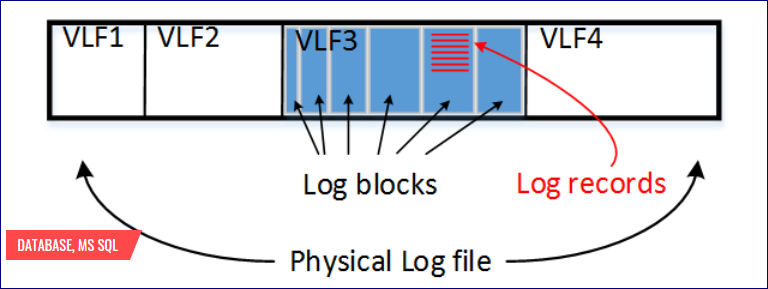
Modified pages will be in buffer some time, whenever the buffer is about to fill with these modified pages, Lazy writer is another internal mechanism usually in sleep mode invokes and clear the buffer pages.

It uses the LRU algorithm in the clearing, LRU stands for L. Recently used pages, on page header of the page there will e reference counter means how many times this page is used, based on counter least used pages will be deleted in the buffer.

## Dirty Pages

Pages commit in the log file and waiting for checkpoint to more MDF, those called dirty pages.

<https://www.ourtechideas.com/blog/sql-server-transaction-log-architecture-part-4/>



**Transaction Log Architecture**

SQL Server uses LSN (Log Sequential Number) in identifying the transaction. Each and every transaction that comes to the log file will associate with an LSN number. Roll forward and the rollback will be done internally using these LSN numbers only.

**WAL (Write Ahead Logging)**

Before committing in MDF every transaction should be written an entry in a log file is called WAL. Transactions never come to MDF directly.

**Logfile divided into 2 parts.**

* Active portion (or) Physical Log
* Inactive Portion (or) Virtual Log

**Active Log Portion:** whenever performs transactions it will have 3 states.

1. Committed to the log file and waiting for the checkpoint.
2. Failed in the middle
3. Transactions still running

All these 3 states’ transactions will be in Active Portion of Log file. When checkpoint runs committed transactions make a copy in inactive portion and moves to MDF.

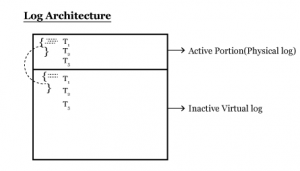
**Inactive Log**

SQL Server maintains fully committed transactions in these Inactive portions. This portion only used for taking the backup of the log. Whenever we take log backup it copies the inactive portion and truncates the inactive portions.

**We have 2 types of backup for log portions.**

Full backup takes the backup of MDF and Active log portion log backup takes the backup of an inactive log portion. This portion we call as a virtual log. SQL Server does not use these records that’s why it calls as Inactive virtual logs.

**SQL Server Log Architecture**



Inactive portion further divided into more virtual logs we have a property called log reusability. Log backup copy inactive portion to a file and truncates the log data. The same space can be used multiple times called log reusability concept.

The transaction log is a cyclic process of writing log records into virtual log file by the SQL server. Whenever one virtual log is filled up it will go to the next virtual log. If all virtual logs files are filled up the inactive portion will grow further and creates more virtual logs, till we have log space allocated. If it cannot grow further it will throw an error “ Transaction log for database is full and the transaction will fail”.

The only way to clear the inactive virtual log is to take log backup released logs. After truncation, this space will be released. The backup will not active portion.

**Advantages of T-Log**

1. IT provides Transactional consistency.
2. It provides transactional recoverability
3. It provides log reusability.

**Enhancement in SQL Server 2012 (Indirect checkpoint)**

A checkpoint occurs automatically based on work hard (or) by a certain operation internally. We can set SQL server level recovery interval to run checkpoint by using SP\_configure (or) manually issue a checkpoint using a checkpoint T-SQL command.

In SQL Server 2012, we have enhancement on checkpoint at database level TARGET T- RECOVERY-TIME option. Increasing time of recovery to seconds (or) minutes.

Alter database sql test 2012 set Target-recovery-time = 5 seconds.

Here every 5 seconds checkpoint will occur.