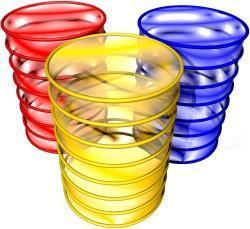
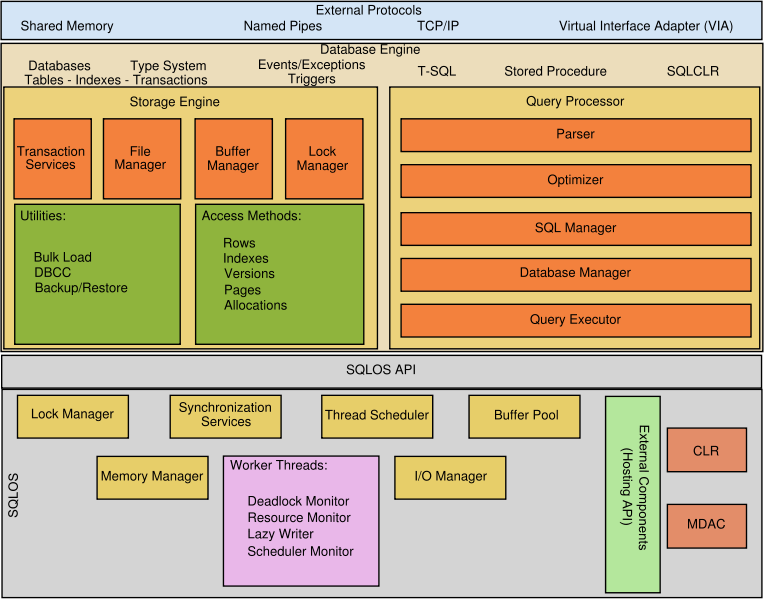
**SQL Server Architecture**



**A Complete Look – SQL Server Architecture**

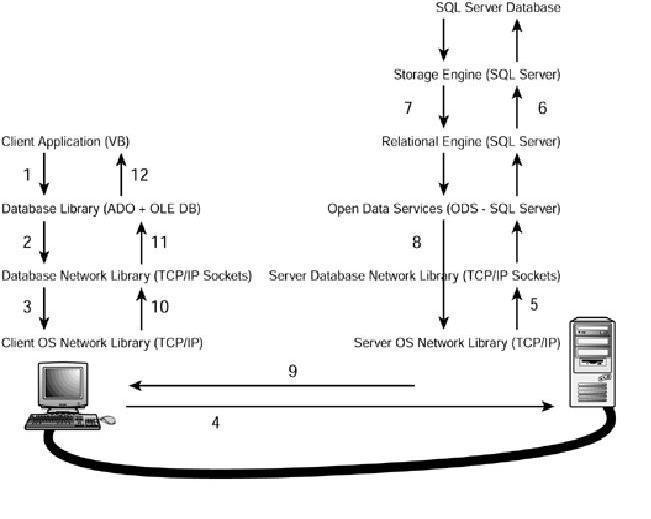
**Q. Can you draw SQL Server architectural diagram with all the components?**

**Ans:**

****

**Network Flow – SQL Server Architecture**

**Q. Can you explain the network flow in SQL Server Architecture?**

**Ans:**

**These steps are defined as follows:**

1. The user selects an option in a client application. This option calls a function in the client application that generates a query that is sent to SQL Server. The application uses a database access library to send the query in a way SQL Server can understand.

2. The database library transforms the original request into a sequence of one or more Transact-SQL statements to be sent to SQL Server. These statements are encapsulated in one or more Tabular Data Stream (TDS) packets and passed to the database network library to be transferred to the server computer.

3. The database network library uses the network library available in the client computer to repackage the TDS packets as network protocol packets.

4. The network protocol packets are sent to the server computer network library across the network, where they are unwrapped from their network protocol.

5. The extracted TDS packets are sent to Open Data Services (ODS), where the original query is extracted.

6. ODS sends the query to the relational engine, where the query is optimized and executed in collaboration with the storage engine.

7. The relational engine creates a result set with the final data and sends it to ODS.

8. ODS builds one or more TDS packets to be sent to the client application, and sends them to the server database network library.

9. The server database network library repackages the TDS packets as network protocol packets and sends them across the network to the client computer.

10. The client computer receives the network protocol packets and forwards them to the network libraries where the TDS packets are extracted.

11. The network library sends the TDS packets to the database access library, where these packets are reassembled and exposed as a client result set to the client application.

12. The client application displays information contained in the result sets to the user.

**Pages and Extents**

**Q. What is a page in database systems and tell me something about pages?**

**Ans:**

The fundamental unit of data storage in SQL Server is the page.

Page size is 8kb means 128 pages = 1 MB.

Page starts with the header of 96 bytes that is used to store page number, page type, the amount of free space on the page, and the object id that owns the page. The maximum size of a single row on a page is 8060 bytes. But this restriction is relaxed for tables which are having varchar, nvarchar, Varbinary, Taxt or Image columns.

**Q. Any idea what is “ROW\_OVERFLOW\_DATA allocation unit”?**

**Ans:**

When the total row size of all fixed and variable columns in a table exceeds the 8,060 byte limitation, SQL Server dynamically moves one or more variable length columns to pages in the ROW\_OVERFLOW\_DATA allocation unit.

This is done whenever an insert or update operation increases the total size of the row beyond the 8060 byte limit. When a column is moved to a page in the ROW\_OVERFLOW\_DATA allocation unit, a 24-byte pointer on the original page in the IN\_ROW\_DATA allocation unit is maintained. If a subsequent operation reduces the row size, SQL Server dynamically moves the columns back to the original data page.

**Q. Can you briefly describe about EXTENT?**

**Ans:**

An extent is eight physically contiguous pages, or 64 KB means 16 Extents= 1 MB.

There are two types of Extents. Uniform Extents and Mixed Extents.

Uniform extents are owned by a single object;

Mixed extents are shared by up to eight objects

**Q. What are the different Types of Pages available?**

**Ans:**

**GAM and SGAM (Global Allocation Map & Shared GAM):**

**GAM:** Extents have been allocated: 1 – Free space 0 – No space

**SGAM:** Mixed Extents have been allocated: 1 – Free Space + Mixed Extent and 0 – No space

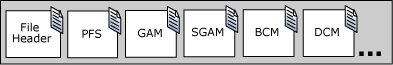
Each GAM / SGAM covers 64000 extents – 4 GB

**PFS (Page Free Space):** Percentage of free space available in each page in an extent.

**DCM (Differential Changed Map):** This tracks the extents that have changed since the last BACKUP DATABASE statement. 1 – Modified, 0 – Not modified

**BCM (Bulk Changed Map):** This tracks the extents that have been modified by bulk logged operations since the last BACKUP LOG statement. 1 – Modified, 0 – Not modified (Used only in bulk logged Recovery model)

In each data file pages are arranged like below



Along with that we have three different data pages

**Data**

**Index**

**Text/ Image (LOB, ROW\_OVERFLOE, XML)**

**Files and File Groups**

**Q. Have you ever heard the word “Files” or “File Groups” in SQL Server?**

**Ans:**

**Files:** There are three types, Primary- .mdf, Secondary – .ndf and Log files – .ldf

**File Groups:** There are two, Primary File Group – All system tables and User Defined – Depends

All secondary files and user defined file groups are created to optimize the data access and for partitioning the tables.

**Q. What are the advantages and disadvantages over using filegroups?**

**Ans:**

**Advantages:**

* Using filegroups, you can explicitly place database objects into a particular set of database files. For example, you can separate tables and their nonclustered indexes into separate filegroups. This can improve performance, because modifications to the table can be written to both the table and the index at the same time. This can be especially useful if you are not using striping with parity (RAID-5).
* Another advantage of filegroups is the ability to back up only a single filegroup at a time. This can be extremely useful for a VLDB, because the sheer size of the database could make backing up an extremely time-consuming process.
* Yet another advantage is the ability to mark the filegroup and all data in the files that are part of it as either read-only or read-write.

**Disadvantages**

* The first is the administration that is involved in keeping track of the files in the filegroup and the database objects that are placed in them.
* The other is that if you are working with a smaller database and have RAID-5 implemented, you may not be improving performance.

**Transaction log Architecture**

**Q. Can you talk about “Transactionlog” Logical and Physical architecture?**

**Ans:**

**“Transactionlog” Logical Architecture:**

Each log record is identified by a unique number called LSN (Log Sequence Number). A log record contains the LSN, TransactionID to which it belongs and data modification record.

Data modification record: It’s either operation performed or before and after data image

* When recorded “Operation Performed”
* Transaction committed – Logical Operation is permanently applied to the data
* Transaction rollback – Reverse Logical Operation is applied to the data.
* When Recorded “Before and After Data Image”
* Transaction committed – Applied the after transaction image
* Transaction rollback – Applied the before transaction image

**“Transactionlog” Physical Architecture:**

The transaction log is used to guarantee the data integrity of the database and for data recovery.

The SQL Server Database Engine divides each physical log file internally into a number of virtual log files. Virtual log files have no fixed size, and there is no fixed number of virtual log files for a physical log file.

The only time virtual log files affect system performance is if the log files are defined by small size and growth\_increment values. If these log files grow to a large size because of many small increments, they will have lots of virtual log files. This can slow down database startup and also log backup and restore operations.

**Q. What are the CheckPoints in SQL Server database?**

**Ans:**

Checkpoints flush dirty data pages from the buffer cache of the current database to disk. This minimizes the active portion of the log that must be processed during a full recovery of a database. During a full recovery, the following types of actions are performed:

* The log records of modifications not flushed to disk before the system stopped are rolled forward.
* All modifications associated with incomplete transactions, such as transactions for which there is no COMMIT or ROLLBACK log record, are rolled back.

Before a database backup, the Database Engine automatically performs a checkpoint so that all changes to the database pages are contained in the backup. Also, stopping a server issues a checkpoint in each database on the server.

**Q. What is an Active Log?**

**Ans:**

The section of the log file from the MinLSN to the last-written log record is called the active portion of the log, or the active log. This is the section of the log required to do a full recovery of the database. No part of the active log can ever be truncated. All log records must be truncated from the parts of the log before the MinLSN.

**Q. What is “Write Ahead Transaction Log”?**

**Ans:**

SQL Server uses a write-ahead log (WAL), which guarantees that no data modifications are written to disk before the associated log record is written to disk.

Data modifications are not made directly to disk, but are made to the copy of the page in the buffer cache. The modification is not written to disk until a checkpoint occurs in the database. A page modified in the cache, but not yet written to disk, is called a dirty page. The internal process that actually goes on:

1. Copy of the data pages are pulled and placed in buffer cache
2. Applied the operation on the pages that are on buffer cache
3. Write the log record details (Pages modified) to Disk
4. Write / flush /apply the page to the disk

If step 4 happens before the step 3 then rollback is impossible. SQL Server takes the responsibility of writing the log details to disk before flushing the dirty pages.

**Memory Management Architecture**

**Q. Can you describe SQL Server Memory Architecture?**

**Ans:**

SQL Server dynamically acquires and frees memory as required. Typically, an administrator need not have to specify how much memory should be allocated to SQL Server, although the option still exists and is required in some environments.

SQL Server supports Address Windowing Extensions (AWE) allowing use of physical memory over 4 gigabytes (GB) on 32-bit versions of Microsoft Windows operating systems. This feature is deprecated from Dinali 2012.

SQL Server tries to reach a balance between two goals:

* Keep the buffer pool from becoming so big that the entire system is low on memory.
* Minimize physical I/O to the database files by maximizing the size of the buffer pool.

**Q. Do you have any idea about Buffer Management?**

**Ans:**

A buffer is a 8kb size in memory. To reduce the I/O operations from database to disk buffer manager use the buffer cache. BM gets the data from database to buffer cache and modifies the data and the modified page is sent back to the disk

The buffer manager only performs reads and writes to the database. Other file and database operations such as open, close, extend, and shrink are performed by the database manager and file manager components.

**Q. Explain effects of Min and Max memory configuration options**

**Ans:**

The min server memory and max server memory configuration options establish upper and lower limits to the amount of memory used by the buffer pool of the Microsoft SQL Server Database Engine. The buffer pool starts with only the memory required to initialize. As the Database Engine workload increases, it keeps acquiring the memory required to support the workload. The buffer pool does not free any of the acquired memory until it reaches the amount specified in min server memory. Once min server memory is reached, the buffer pool then uses the standard algorithm to acquire and free memory as needed. The only difference is that the buffer pool never drops its memory allocation below the level specified in min server memory, and never acquires more memory than the level specified in max server memory.

**Thread and Task Architecture**

**Q. Can you describe how SQL Server handles Batch or Task Scheduling?**

**Ans:**

Each instance must handle potentially thousands of concurrent requests from users. Instances of SQL Server use Microsoft Windows threads, or if configured, they use fibers, to manage these concurrent tasks efficiently. This includes one or more threads for each server Net-Library, a network thread to handle network I/O, and a signal thread for communicating with the Service Control Manager.

**Understanding Scheduling:** Each instance of SQL Server has an internal layer (SQL OS/Kernel) that implements an environment similar to an operating system. This internal layer is used for scheduling and synchronizing concurrent tasks without having to call the Windows kernel.

**Connection:** A connection is established when the user is successfully logged in. The user can then submit one or more Transact-SQL statements for execution. A connection is closed when the user explicitly logs out, or the connection is terminated.

**Batch:** An SQL batch is a set of one or more Transact-SQL statements sent from a client to an instance of SQL Server for execution.

**Task:** A task represents a unit of work that is scheduled by SQL Server. A batch can map to one or more tasks.

**Windows thread:** Each Windows thread represents an independent execution mechanism.

**Fiber:** A fiber is a lightweight thread that requires fewer resources than a Windows thread. One Windows thread can be mapped to many fibers.

**Worker thread:** The worker thread represents a logical thread (Task) in SQL Server that is internally mapped (1:1) to either a Windows thread or, if lightweight pooling is turned ON, to a fiber. The mapping can be done till the free worker threads available. (Parameter: Max worker Threads)

**Thread and Fiber Execution:** Microsoft Windows uses a numeric priority system that ranges from 1 through 31 to schedule threads for execution. Zero is reserved for operating system use. When several threads are waiting to execute, Windows dispatches the thread with the highest priority.

By default, each instance of SQL Server is a priority of 7, which is referred to as the normal priority. The priority boost configuration option can be used to increase the priority of the threads from an instance of SQL Server to 13. This is referred to as high priority.

The performance of any instances running at normal priority can be adversely affected. Also, the performance of other applications and components on the server can decline if priority boost is turned on.

**Query Processing Architecture**

**Q. I have submitted a Query to SQL Server from an Application and I got the reply as “data inserted successfully”. Can you demonstrate what the processing done inside?**

**Ans:**

When you submit a query to a SQL Server database, a number of processes on the server go to work on that query. The purpose of all these processes is to manage the system such that it will provide your data back to you, or store it, in as timely a manner as possible, whilst maintaining the integrity of the data.

All these processes go through two stages:

1. Relational Engine

2. Storage Engine

**At Client:**

1. User enter data and click on submit

2. The client database library transforms the original request into a sequence of one or more Transact-SQL statements to be sent to SQL Server. These statements are encapsulated in one or more Tabular Data Stream (TDS) packets and passed to the database network library

3. The database network library uses the network library available in the client computer to repackage the TDS packets as network protocol packets.

4. The network protocol packets are sent to the server computer network library across the network

At Server:

5. The extracted TDS packets are sent to Open Data Services (ODS), where the original query is extracted.

6. ODS sends the query to the relational engine

7. A connection established to the relational engine and assign a SID to the connection

**At Relational Engine:**

8. Check permissions and determines if the query can be executed by the user associated with the request

9. Query sends to Query Parser

* It checks that the T-SQL is written correctly
* Build a Parse Tree \ Sequence Tree

10. Parse Tree sends to Algebrizer

* Verifies all the columns, objects and data types
* Aggregate Binding (determines the location of aggregates such as GROUP BY, and MAX)
* Builds a Query Processor Tree in Binary Format

11. Query Processor Tree sends to Optimizer

* Based on the query processor tree and Histogram (Statistics) builds an optimized execution plan
* Stores the execution plan into cache and send it to the database engine

**At Database Engine:**

12. Database engine map a batch into different tasks

13. Each task associated with a process

14. Each process assigned with a Windows Thread or a Windows Fiber. The worker thread takes care of this.

15. The Thread/Fiber send to the execution queue and wait for the CPU time.

16. The Thread/Fiber identifies the table location where the data need to be stored

17. Go to the file header, checks the PFS, GAM and GSAM and go to the correct page

18. Verifies the page is not corrupted using Torn page Detection / Check SUM and writes the data

19. If require allocates new pages and stores data on it. Once the data is stored/updated/added in a page, it updates the below locations

* PFS
* Page Header – Checksum / Torn Page Detection (Sector info)
* BCM
* DCM

20. In this process the

* Memory manager take care of allocating buffers, new pages etc,
* Lock manager take care of allocating appropriate locks on the objects/pages and releasing them when task completed
* Thread Scheduler: schedules the threads for CPU time
* I/O manager: Establish memory bus for read/write operations from memory to disk and vice versa
* Deadlock\Resource\Scheduler Monitor: Monitors the processes

21. If that is a DML operation, it picks the appropriate page from disk and put the page in Memory.

22. While the page is available on Memory based on the ISOLATION LEVEL an shared / exclusive / update / Schema lock issued on that page.

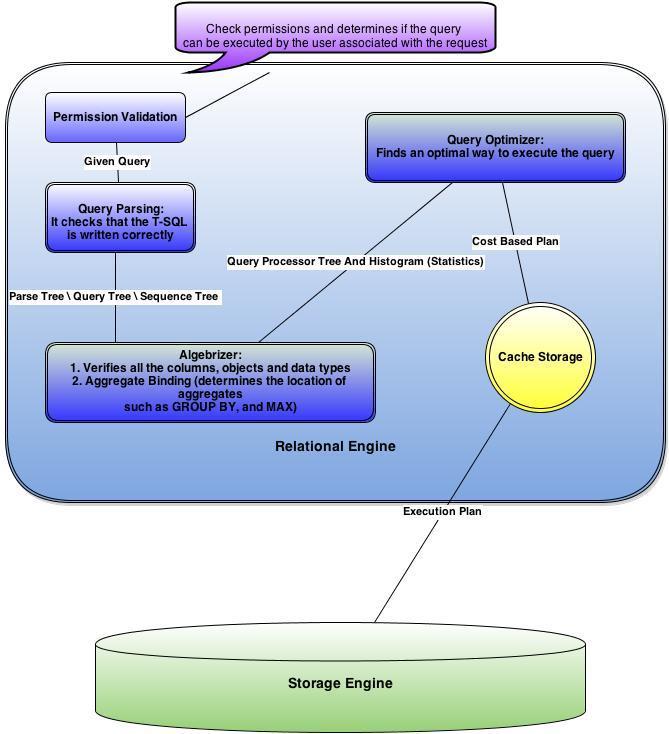
23. Once the page is modified at Memory, that means once the transaction completed the transactional operation logged into log file (.ldf) to the concerned VLF.

24. Here we should understand that only the operation (T-SQL statements) logged into ldf file. The modified page waits in memory till check point happens. These pages are know as dirty pages as the page data is differ in between the page on Disk and Memory.

25. Once the checkpoint happens the page will be written back to the disk.

26. Once the process is completed the result set is submitted to the relational engine and follow the same process for sending back the result set to client application.

27. The connection will be closed and the SID is removed



**Network Protocols**

**Q. What are the different types of network protocols? Explain each of them in detail**

**Ans:**

**Shared Memory:**

Clients using the shared memory protocol can only connect to a SQL Server instance running on the same computer; it is not useful for most database activity. Use the shared memory protocol for troubleshooting when you suspect the other protocols are configured incorrectly.

Server – Machine 1

Clients – Machine 1

**TCP/IP:**

TCP/IP is a common protocol widely used over the Internet. It communicates across interconnected networks of computers that have diverse hardware architectures and various operating systems. TCP/IP includes standards for routing network traffic and offers advanced security features. It is the most popular protocol that is used in business today.

Server – Machine 1

Clients – WAN (Any machine from any network)

**Named Pipes:**

Named Pipes is a protocol developed for local area networks. A part of memory is used by one process to pass information to another process, so that the output of one is the input of the other. The second process can be local (on the same computer as the first) or remote (on a networked computer).

Server – Machine 1

Clients – LAN (Any machine from LAN)

**VIA:**

Virtual Interface Adapter (VIA) protocol works with VIA hardware. This feature will be deprecated in future releases.

**Miscellaneous**

**Q. What are the memory capacity specifications for SQL Server 2008 R2?**

**Ans:**

SQL Server edition Maximum memory supported

Datacenter Operating system maximum

Enterprise 2 TB

Developer Operating system maximum

Standard 64 GB

Web 64 GB

Workgroup 4 GB (64-bit), OS maximum (32-bit)

Express 1 GB

Express with Tools 1 GB

Express with

Advanced Services 1 GB and 4 GB for Reporting Services

**Q. What are the maximum capacity specifications for SQL Server 2008 R2?**

**Ans:**

Bytes per Index Key 900

Bytes per Primary Key 900

Bytes per Foreign Key 900

Bytes per Row 8060

Columns per Index Key 16

Columns per Primary Key 16

Columns per Foreign Key 16

Columns in GROUP BY, ORDER BY 900 Bytes

Columns for Insert Statement 4096

Columns for Delete Statement 4096

Columns for Update Statement 4096

Database Size 524,272 TB

Databases per Instance 32,767

File Groups per Database 32,767

Files per Database 32,767

File Size (Data) 16 TB

File Size (Log) 2 TB

Instances per Computer 50

Instances per Cluster 25

Nested Stored Procedures 32

Nested Sub quires 32

Nested Triggers 32

Clustered Indexes per Table 1

Non Clustered Indexes per Table 999

XML Indexes 249

Parameters per SP’s 2100

Parameters per UDF’s 2100

User Connections 32767

Articles – Merge Publication 256

Articles – Tran and Snapshot Repl 32767