

A Study on Microsoft SQL Server Architecture

SQL and Page Extent Architecture

Prithviraj J. Mane

Database Administrator – Microsoft SQL Server

Abstract- Databases are used to collect and store data in a proper format. For that database management system (DBMS) is necessary. Microsoft SQL server is the most known database management system. This paper will give you a brief introduction about Microsoft SQL Server, how it works, what are the features of Microsoft SQL Server and how would the data be being stored in the database.

Index Terms- Microsoft SQL Server, Relational Database, DBMS, Architecture, SQL Server.

I. INTRODUCTION

SQL stands for structured query language. Microsoft SQL Server is a relational database management system that supports a wide variety of transaction processing business intelligence and analytical applications in corporate environment. It's a software product with the primary function of storing and retrieving data as required by another software application, which is maybe some computer or on another computer across the network. MS SQL server also has T SQL or transact SQL and the focus of T SQL is to handle the transaction. Microsoft SQL Server is designed to run on a central server so that multiple users can access the same data simultaneously. generally, users access the database through an application.

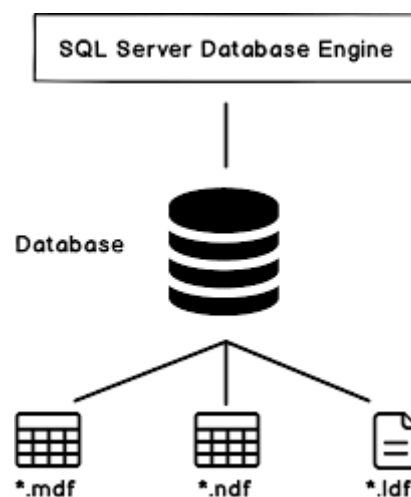
II. DATABASE

Database is nothing but connection of data which is arranged in the proper format.

We can create 32,000 databases in a SQL instance, because of heavy load we cannot create this many databases.

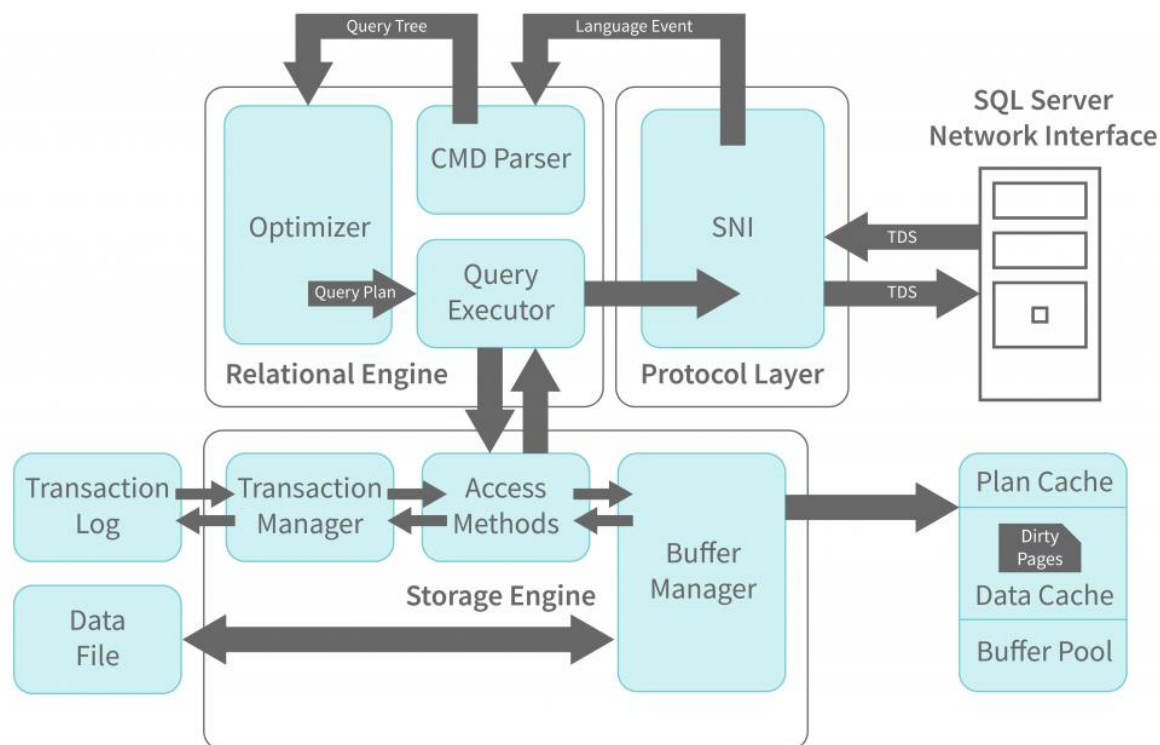
Whenever we create a database two files will be automatically generated one is .mdf, the second is .ldf.

- .mdf stands for primary data file or master data file which stores all your data.
- .ldf Stands for low data files which contain all the change related information. Whatever you are performing changes like inserting, updating all changes will be logged in the .ldf file before moving to .mdf file.



- If your database size is very huge like 5 TB, 10 TB, 15TB. In this case we suggest keeping the database in a separate data file to improve performance. We create .ndf file called secondary data file to store data, it's like splitting the data files. We can create n number of secondary data files to store data.

III. SQL ARCHITECTURE



As we can see in the diagram this total SQL Server is divided into two parts, the first one is relation engine and second one is storage engine.

Relational engine is divided into two parts, one is client side another is a server side. As we see in the diagram, on both sides we have SNI (SQL Server network interface). So, whenever user tries to insert a query, it will fetch by SNI, and it will send through TDS to server side SNI. TDS is a type of communication protocol. In protocol layer the query will be encrypted, and we will get an SQL query. The default port for SQL is 1433.

Then protocol layer sends the query to CMD parser true language event. The CMD parser first checks whether the query has any syntax error. If the query has some syntax error, then it will revert back and it will show error to user. If the query is OK, then it will pass to optimizer through query tree.

Now it's going into the optimizer the optimizer will choose the best execution plan for the query. When the optimizer is done with the task it sends to the query executor true query plan.

The query executor is executing that exact query and it again sent to access method for the next process. So, after the query executor task relational engine task is complete here. Now this query is in storage engine.

In the storage engine it will mainly check two things whether it is select query or non-select query.

If it is a select query, then it will go to buffer manager. After buffer manager it goes to plan cache. The buffer manager has a temporary area called buffer pool, in the buffer pool we have plan cache and data cache. Consider example if the same queries triggered by another user some time ago and each page data from the data file will store a plan in the plan cache. Now you are trying to run the same query now, you don't need to go to data file and fetch the data, data is already present in the plan cache. It simply takes execution plan from the plan cache, and it will fetch the data from the data cache, and it could simply revert back to user.

If the query is not available in the plan cache, then it will fetch the data from the data file. If the plan is available in the plan cache and it's the fetch the data from the data cache and it reverted to the user, so no need to fetch the data from the data file it is known that soft parsing. If that required query has to fetch data from the data file, then it's called hard parsing.

Let's consider it's a non-select query like update query, at that time it will fetch the data from the transaction log file through transaction manager. In transaction manager there are also two components. The first is log manager and second is lock manager.

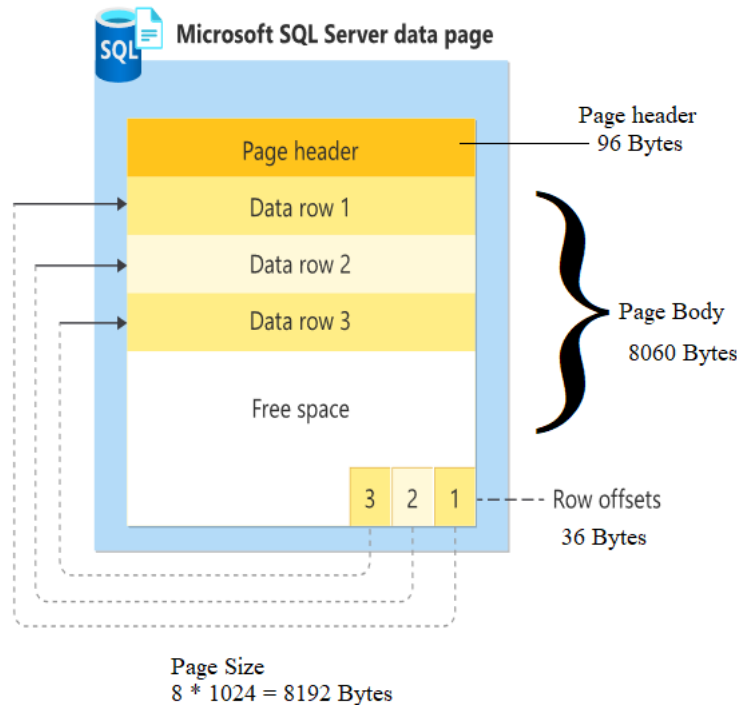
First the access method is sending the query to the transaction manager and then it transferred to transaction log. transaction log will assign a log sequence number sequential number. it will keep a lock also. It assigns a lock to avoid data loss and for data consistency.

The transaction will go from transaction log to transaction manager to buffer manager through access method and eat write the data in the data cache and through buffer pull it will fetch the data from data file once checkpoint occurred.

And it will send to client side SNI and So through this way the user will get output.

IV. PAGE ARCHITECTURE

- Whenever we are storing data into .mdf file it will store data in the form of pages. Just like how we write something in our notebook, a similar concept used in the database to store data.
- Page content three sections header, page body and footer. This page size is 8 KB, and one more thing we cannot increase or decrease the page size it is fixed value.



- Total page size is 8192 bytes, in that page header contains 96 bytes, each body contains 8060 bytes of data and page footer contains 36 bytes of data.
- Page Header size is 96 bytes which will store what is data page, how many rows or slots are there in the page, how much free space is there, how much data is filled, which object is present as well as current page ID, previous page ID, next page ID, object ID.
- Page footer nothing but whenever you are trying to insert or update some data in your page, data is entering from footer through row offset it is told in the end of the page after data row session.
- Whenever you are trying to insert but I called it will first come to row offset. The aim of row offset is to show current data is going in which row. each row having its own unique ID.
- Whenever the first page is full of data, automatically it's allocated another page and the data is inserted in that second page. This process is continuously going on until the data ends.

Page Type

1. Data page.
2. global allocation map.
3. shared global allocation map.
4. Page free space.
5. boot page.
6. text / Image page.

1. Data page:

When is the actual data each storing in a page is nothing but data page.

2. Global allocation map (GAM):

One gram page size is 4GB, it contains 64,000 extents. The GAM size directly does not go to 4GB once we create a database a normal GAM page will create and whenever we are inserting or updating any data in the database automatically the GAM page will increase. Mostly inserting data will increase the GAM page size till it reaches 4GB, after one GAM page full it will create another GAM page and insert data into another GAM page.

Let's consider example if we are inserting 1,00,000 records, to fill 1,00,000 record it occupied it occupied 10,000 extents, so now out of 64,000 extents 10,000 already occupied so remaining 54,000 extents is empty.

3. Shared global allocation map:

Let's take an example, If I'm having 8 page (One Extent) and out of 8 page 4 are empty. So, this information is collected by SGAM page.

SGAM page size is also 4GB and contains 64,000 extents. So, our allocation will be done by GAM page and SGAM will cover whether your extents it's free or it's filled with any data.

So, if the bit is 1 then this extent is used adds mixed extent and having free space.

If bit is 0 then extent is not used as mixed extents and all the pages are being used.

GAM content allocation information and SGAM will content how much free space is there an extent.

4. Page free space:

It is tracking how much free space there is in every single data page. It is tracking for all the individual data pages. Like how much data page allocated how much data pages are free and how much are filled with data.

SGAM page will track how much space free space is there in extents and PFS is tracking how much free space in every individual page.

Like 1% to 50% free space in a page,
 51% to 80% free space in a page,
 81% to 90% free space in a page,
 91% to 99% free space in a page.

5. Boot page:

Boot page content metadata related information about the database. Like what is your database name, database ID, location of database any external operations Bing performed by the user on the database, check DB, checkpoint.

All the metadata information will be stored on a single page.

Only one page will be available for the whole database and this page will be always the 9th page.

6. Text / Image page:

Whenever you are storing any document or image this page will come into use.

There's a concept called file stream, whenever we are installing SQL Server at that time, we get a window to enable or disable the file stream. If we enable the file stream, then only we will be able to create text / image page.

In the file stream users are able to store file / image / document in database.

Let's consider for example: whenever we are filling in any form in the bank, the bank asks for your photo and signature, so this document is going to store in the text / image page.

V. EXTENTS

Extent is collection of eight continuous data pages. Which having a size of 64 KB.

Of actual data is going to be stored in a single page but when we combine 8 pages it becomes an extent.

In SQL Server there are majorly two types of extent.

- Mixed extents
- Uniform extents

Mixed extents

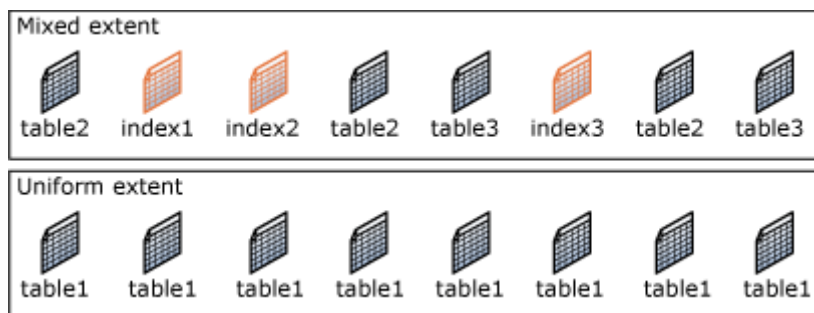
If it continues pages contain multiple object data, then it will become a mixed extent.

Example: if we take a page in which we write some data as well as we store some drawings are photographs then it contains information also and photographs also so it will become a mixed extent.

Uniform extent

If it continues pages contained same object of data, then is considered as uniform extent.

Example: if we are adding only a single object data like SQL Server data or information about sequel server in a paragraph way then it will become a uniform extent.



CONCLUSION

The purpose of this case study is just to clarify how SQL Server stores the data in the database. MSSQL is used in the small business as well as large scale business also such important to understand how sequel itself storing a data and fetching the data from databases through pages. SQL Server manages multiple users simultaneously and ensure that transactions observe the properties of the chosen isolation level. Here we try to explain how SQL server internally works how it fetched data as well as what is the architecture for SQL services.

REFERENCES:

1. Ms SQL Server Research, by Noradlirazman Bin Othman (references)
2. Securing Relational Databases against Security Vulnerabilities: A Case of Microsoft SQL Server and PostgreSQL , Hassan Kilavo, Salehe I. Mrutu &Robert G. Dudu.
3. A case study for MySQL and Microsoft SQL server database reporting, Shivangi Thakkar
4. Comparative Performance Analysis of MySQL and SQL Server Relational Database Management Systems in Windows Environment, Amlanjyoti Saikia¹, Sherin Joy², Dhondup Dolma³, Roseline Mary. R⁴