SQL SERVER, Big Data Interview Questions & Answers - SET 8 (10 Questions)

1. Can you tell me the difference between Name Node and Data Node?

Answer-

This question belongs to Big Data. Differences below-

NameNode

- NameNode is the centrepiece of HDFS.
- NameNode is also known as the Master
- NameNode only stores the metadata of HDFS – the directory tree of all files in the file system, and tracks the files across the cluster.
- NameNode does not store the actual data or the dataset. The data itself is actually stored in the DataNodes.
- NameNode knows the list of the blocks and its location for any given file in HDFS. With this information NameNode knows how to construct the file from blocks.
- NameNode is so critical to HDFS and when the NameNode is down, HDFS/Hadoop cluster is inaccessible and considered down.
- NameNode is a single point of failure in Hadoop cluster.
- NameNode is usually configured with a lot of memory (RAM). Because the block locations are help in main memory.

DataNode

- DataNode is responsible for storing the actual data in HDFS.
- DataNode is also known as the Slave
- NameNode and DataNode are in constant communication.

- When a DataNode starts up it announce itself to the NameNode along with the list of blocks it is responsible for.
- When a DataNode is down, it does not affect the availability of data or the cluster. NameNode will arrange for replication for the blocks managed by the DataNode that is not available.
- DataNode is usually configured with a lot of hard disk space. Because the actual data is stored in the DataNode.

2. Can you tell internal objects SQL Server Engine created inside TempDB?

Answer-

Internal Object in TempDB are

- Sorts
- Worktables
- WorkFiles
- VersionStore

Sorts-

Consider this as a workspace memory or Query memory. Whenever the operation was not fit in the memory it spills to tempDB. In case of Index sorting, it is spilled to User DB unless you specify the option "SORT_IN_TEMPDB". It is technically not a work table. They use Uniform Extents for storage.

Worktables

Worktables are temporary rowsets. SQL Engine will call an internal routine to build a table. They are same as

Temporary tables just like developers create using Create Table command. These are just pages in tempDB and they have –ve object ids. They use Mixed Extents.

Worktables are used for below operations-

- **XML Documents**
- **Spool Operators**
- **Hash Match**
- **Exchange Spill**
- **Merge Join**
- **LOB Variables**
- **Cursors**
- **Service Broker**

WorkFiles

WorkFiles are only used for one purpose that is temporary storage for Hashing. They use Uniform Extents

Version Store

They are used for Snapshot isolation including RCSI DB option. It is also used for online index rebuilding operations, triggers and MARS. Internally these extents are marked as “Version Store”.

3. How SQL Server gets SELECT Count(*) – Will it always use Table Scan every time?

Answer-

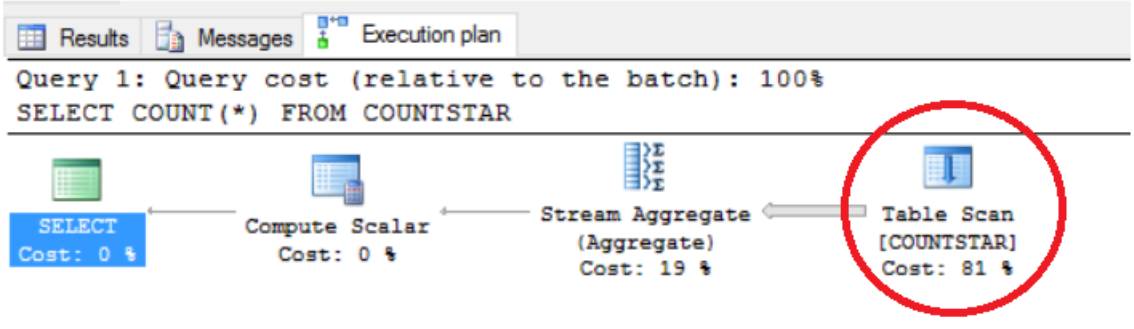
SELECT COUNT(*) will always use the smallest index structure that contains the total number of rows. The Optimizer will do a leaf level scan of the smallest nonclustered index for count(*) as well as count(not-nullable column)

Obviously if there is not index then the SQL optimizer will have to use table Scan.

Example – Table Scan

```
CREATE TABLE COUNTSTAR
(
     ID INT IDENTITY(1,1)
)
GO

INSERT INTO COUNTSTAR DEFAULT VALUES
GO 10000

SELECT COUNT(*) FROM COUNTSTAR
```

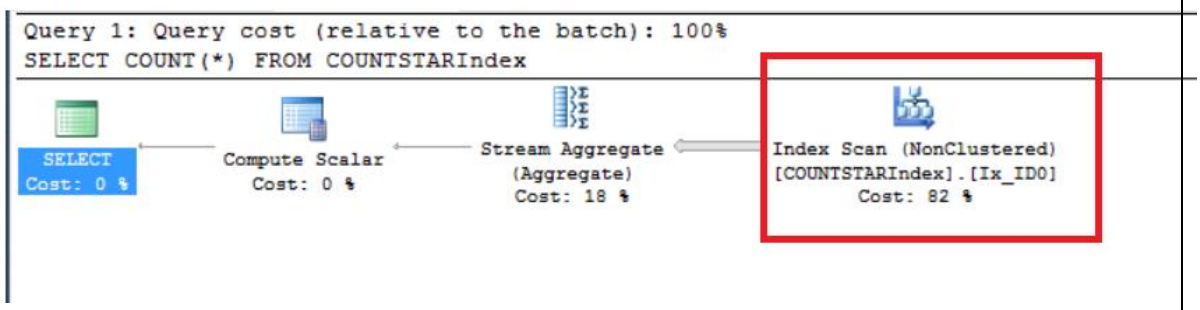
--Execution Plan



Example – Index Scan

```
CREATE TABLE COUNTSTARIndex
(
      ID  INT IDENTITY(1,1) PRIMARY KEY
     ,ID0 INT DEFAULT 0
     ,ID1 BIGINT DEFAULT 1
     ,ID2 VARCHAR(10) DEFAULT 'Pawan'
)
GO

INSERT INTO COUNTSTARIndex DEFAULT VALUES
GO 10000

CREATE NONCLUSTERED INDEX Ix_ID0 ON COUNTSTARIndex(ID0)
GO

CREATE NONCLUSTERED INDEX Ix_ID1 ON COUNTSTARIndex(ID1)
GO

CREATE NONCLUSTERED INDEX Ix_ID2 ON COUNTSTARIndex(ID2)
GO

SELECT COUNT(*) FROM COUNTSTARIndex
```

--Execution Plan



In this case it is using Ix_ID0 index which is the smallest nonclustered index structure in this case.

4. What is the Output of below Query?

```
CREATE TABLE # (a INT, b INT);
```

Answer-

The Output of the above query will be “Command Executed Successfully”.

5. What is the Output of below Queries?

```
CREATE TABLE food (a INT, b INT);

select a, b from food order by a, b;
select a, b from food order by 1, 2;

select a, b, count(*) from food group by a, b;
select a, b, count(*) from food group by 1, 2;
```

Answer-

First 3 queries will be executed successfully and provide data to the end user.

Last query will return below error.

“Msg 164, Level 15, State 1, Line 7 Each GROUP BY expression must contain at least one column that is not an outer reference.”

6. Can you tell me the purpose of NTILE ranking function in SQL Server?

Answer-

The NTILE ranking function distributes the rows in an ordered partition into a specified number of groups. The groups are numbered, starting at one. For each row, NTILE returns the number of the group to which the row belongs.

Example

```
CREATE TABLE EmpRangers
(
     EmpId INT
    ,EmpName VARCHAR(10)
    ,Salary INT
)
GO

INSERT INTO EmpRangers
VALUES
(1,'a',1000),
(2,'b',900),
(3,'c',100),
(4,'d',1100),
(5,'e',1300),
(6,'f',700),
(7,'g',330),
(8,'h',800),
(9,'i',500),
(10,'j',340),
(11,'k',600),
(12,'l',700),
(13,'m',1000)
GO


INSERT INTO EmpRangers
VALUES (14,'n',1800)
```

```
GO



-- Query with Ntile ranking function

SELECT
             Salary
            ,NTILE(3) OVER (ORDER BY Salary) Grouper
    FROM
            EmpRangers
```

7. You have got a table. You have to write a SELECT query which will fetch alternate rows from the table starting with first row?

Answer-

```
CREATE TABLE EmpRangers
(
     EmpId INT
    ,EmpName VARCHAR(10)
    ,Salary INT
)
GO

INSERT INTO EmpRangers
VALUES
(1,'a',1000),
(2,'b',900),
(3,'c',100),
(4,'d',1100),
(5,'e',1300),
(6,'f',700),
(7,'g',330),
(8,'h',800),
(9,'i',500),
(10,'j',340),
```

```
(11,'k',600),
(12,'l',700),
(13,'m',1000)
GO

SELECT EmpId,EmpName,Salary FROM EmpRangers
WHERE EmpId % 2 <> 0
GO
```

8. Can you tell me how many triggers can we create on a table?

Answer-

We can create 1 Instead of Trigger and Multiple After Triggers on a table. Also note that we can only create Instead of Triggers on a view.

9. How many kinds of database files present in SQL Server?

Answer-

There are three basic types of database files present in SQL Server Database-

- Primary data file (Extension - mdf)
- Secondary data file (Extension - ndf)
- Log file (Extension - ldf)

Primary data files

This data file is mandatory whenever you create a new database, this file contains all the startup parameter of a database when SQL Server try to bring database online or when you create a new database on SQL Server. The common/recommended extension of primary data files is

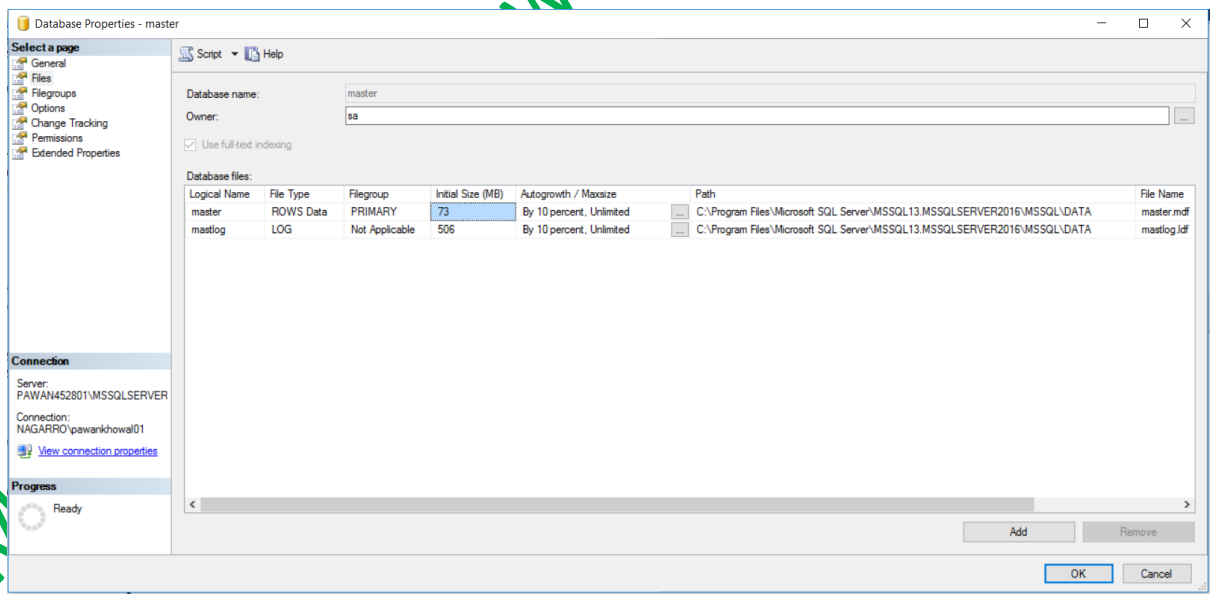
".mdf" and it holds the system objects of a database such as tables, store procedures, views and functions.

Secondary data files

These are optional user defined data files; they hold user defined database objects such as tables, store procedures, views and functions. There could be multiple secondary files of a database. The common/recommended extension used for Secondary data files is ".ndf"

Log Files

Log files contain transactional information of a database's day to day processing and are very important for database recovery process. All the transaction sequence and information is stored in these files. Every database has to have one log file in order to be operational.



10. What are Niladic functions in SQL Server?

Answer-

Niladic functions are functions that do not accept any parameters, are specified without parentheses, and return only one result.. "Nil" means nothing or null, and "adic" when is prefixed, means arguments.

Example -

```
SELECT  CURRENT_TIMESTAMP
SELECT  CURRENT_USER
SELECT  SESSION_USER
SELECT  SYSTEM_USER
SELECT  USER
```


That's all folks; I hope you've enjoyed the article and I'll see you soon with some more articles.

Thanks!

Pawan Kumar Khowal

MSBISKills.com