

# SQL SERVER Interview Questions and Answers (Set 4)

<http://msbiskills.com/>

## Question – What are column store indexes? How they are better for performance?

- In SQL Server 2012, Microsoft introduced the columnstore index, a memory-optimized column-oriented index. They are used to handle data warehouse workloads.
- An xVelocity memory optimized columnstore index, groups and stores data for each column and then joins all the columns to complete the whole index. This differs from traditional indexes which group and store data for each row and then join all the rows to complete the whole index.
- For some types of queries, the SQL Server query processor can take advantage of the columnstore layout to significantly improve query execution times. The explosive growth of data warehousing, decision support and BI applications has generated an urgency to read and process very large data sets quickly and accurately into useful information and knowledge. This growth of data volumes and escalating expectations are challenging to maintain or improve through incremental performance improvements.
- SQL Server columnstore index technology is especially appropriate for typical data warehousing data sets. Columnstore indexes can transform the data warehousing experience for users by enabling faster performance for common data warehousing queries such as filtering, aggregating, grouping, and star-join queries.
- SQL Server 2014 adds support for updateable clustered columnstore indexes. With SQL Server 2014, the clustered columnstore index can be used in place of a traditional rowstore

clustered index. The clustered columnstore index permits data modifications and bulk load operations. Although it has limitations, the clustered columnstore index still represents a significant step forward in providing an indexing strategy better suited to the enterprise data warehouse.

If you need further details click below –

[https://technet.microsoft.com/en-us/library/gg492088\(v=sql.110\).aspx#HT\\_create](https://technet.microsoft.com/en-us/library/gg492088(v=sql.110).aspx#HT_create)

[https://technet.microsoft.com/en-us/library/gg492153\(v=sql.110\).aspx](https://technet.microsoft.com/en-us/library/gg492153(v=sql.110).aspx)

## [Question – How SQL Server handles concurrency in SQL 2014?](#)

In-memory OLTP the Concurrency model is a version-based supporting multiple versions of the rows with different lifetime. SQL Server stores in tempdb copies (versions) of all the previously committed versions of any data rows, since the beginning of the oldest open transaction (i.e. it keeps those copies as long as there are any transactions that might need to access them). The space in tempdb used to store previous versions of changed rows is the version store. When using this row versioning, readers do not block writers, and writers do not block readers (though writers do still take locks and will block other writers).

I suggests please go through the link below –

<https://www.simple-talk.com/sql/t-sql-programming/row-versioning-concurrency-in-sql-server/>

**Question – How you can achieve high availability in SQL SERVER, Your database should be up all the time, 0% downtime is required.**

SQL Server provides several options for creating high availability for a server or database. The high-availability options include the following:

- AlwaysOn Failover Cluster Instances
- AlwaysOn Availability Groups
- Database mirroring
- Log shipping
- Also consider VMware Vsphere – It is used to minimize Downtime from Server and OS Failures. Explained in detail in the next question.

For details follow this link – <https://msdn.microsoft.com/en-us/library/ms190202.aspx>

**Question – How you can achieve high availability in applications, Your application should be up all the time, 0 downtime is required.**

You can use VMware Vsphere – It is used to minimize Downtime from Server and OS Failures

VMware vSphere® High Availability (HA) delivers the availability required by most applications running in virtual machines, independent of the operating system and application running in it. HA provides uniform, cost-effective failover protection against hardware and operating system outages within your virtualized IT

environment. HA provides the foundation for a highly available environment by monitoring virtual machines and the hosts upon which they run. HA is a mature solution that provides scalability, reliability and usability.

### **Scalability**

- Master-slave node relationship – This new relationship model between nodes in a cluster replaces primary and secondary nodes. Availability actions are coordinated by a single master node, which communicates all activities and states to VMware® vCenter Server™. This eliminates much of the planning required to design a highly available environment—administrators no longer need to worry about which hosts are primary nodes and where they are located. This is especially significant when deploying HA on blade chassis and in stretched cluster environments.
- Support for IPv6 networking – Enables IT departments in need of a larger “address” space to fully leverage their network infrastructure.
- Simple deployment mechanism – Fast and easy completion of routine tasks such as deploying vSphere HA agent and configuring HA functionality.

### **Reliability**

When an outage occurs, the last thing administrators need is to worry about whether their high availability solution will work. Guided by real-world customer feedback, VMware has added capabilities to maximize your confidence in HA:

- No external component dependencies – HA has no dependency upon DNS resolution. This reduces the odds that an external component outage will impact HA operations.
- Multiple communication paths – HA nodes within a cluster can communicate through the storage subsystem as well as over the management network. Multiple communication paths increase

redundancy and enable better assessment of the health of a vSphere host and its virtual machines.

- VM-VM anti-affinity rules – HA has been enhanced to respect VM-VM anti-affinity rules defined in VMware vSphere® Distributed Resource Scheduler™, eliminating the need for VMware vSphere® vMotion® migrations after failover.

### **Usability**

HA's enhanced interface lets users quickly identify the role and state of each node in a cluster. Error condition messages are also easier to understand and act upon. In the rare case that an issue does occur with HA, only one log file needs to be reviewed, speeding time to resolution.

For more details please click here –

<http://www.vmware.com/in/products/vsphere/features/high-availability>

### **Question – You have given a requirement? How do you start DB design for it?**

You can start with some points mentioned below.

- First try and understand the purpose of your database.
- In the second step find the Entities (Tables)
- Understand your data – Duplicate and non-duplicate
- Find out fields for each entity (Table), Decide on their data types and size
- Decide your primary keys or clustered indexes based on the business, They are not required but highly recommended
- Check normalization based on OLAP or OLTP
- Find out the relationships between these entities

- Create extra tables if you have M-N relationship between tables
- Decide on the foreign keys, Add foreign keys in respective tables
- Decide on the fill factor of the tables (OLAP/OLTP)
- Find out the Non clustered indexes you want to put on each tables
- Also decide on whether you want to put all the file groups on the same server or on different servers
- Check hardware based on OLTP/OLAP, if you need high availability go for VMware vSphere, EMC XtremIO, etc.

### Question –Write a SQL script to implement Fibonacci series in SQL Server up to 200.

This should not be complex. We can easily achieve this with or without recursion or using any loop. Two sample methods are given below. IF you not aware of Fibonacci series, please visit below URLs-

<https://www.mathsisfun.com/numbers/fibonacci-sequence.html>

<https://www.math.hmc.edu/funfacts/ffiles/10002.4-5.shtml>

```

/***** WITH RECURSION *****/
--
;WITH FibonacciCTE
AS
    (
        SELECT  0 Levels, 0 FibNum, 1 FibNex
        UNION ALL
        SELECT  a.Levels + 1 Levels, a.FibNex  FibNum, a.FibNum +
a.FibNex AS FibNex
        FROM FibonacciCTE a
        WHERE a.FibNex &lt; 200
    )
SELECT FibNum FibonacciNumbers

```

```

FROM FibonacciCTE f

GO

/***** WITHOUT RECURSION *****/

DECLARE @phi FLOAT
SELECT @phi = (1 + Sqrt(5))/2

;WITH CTE AS
(
    SELECT DISTINCT Number
    FROM master..spt_values
    WHERE number BETWEEN 0 AND 200
)
SELECT
ROUND((POWER(((1 + Sqrt(5))/2), Number) - POWER(-1/((1 +
Sqrt(5))/2), Number )) / Sqrt(5) ,1) FibonacciNumbers
FROM CTE
WHERE
ROUND((POWER(((1 + Sqrt(5))/2), Number) - POWER(-1/((1 +
Sqrt(5))/2), Number )) / Sqrt(5) ,1) &&&< 200

--

```

**Question – How to find 2nd highest number from a integer array in C#/C language?( Yes it's not related to SQL :))**

One method is sort the array using sort method and get the element present at Array length- 1

**Question – What are Spools in Execution Plan? What are the types of Spools in execution plans?**

## **So what is a Spool?**

1. Spool is a physical operator that shows up in the execution plan.
2. Spool operator effectively saved an intermediate result to the tempdb for further usage.
3. Consider Spools are a cache in the query processor.

## **How they are implemented?**

Spools are implemented as hidden tables in tempdb. Means you get all of your data in tempdb and then queries it. Query optimizer uses spools when it thinks that it's better to put data in the temp table rather than hitting a table again and again using seeks and scans. So what it does is, it picks data from the table and saves that in the temp table as a hidden table and uses it whenever query optimizer requires it.

Note – Please note that this hidden table is not same as temp table. Consider this as a hidden table (Work table) for a particular query.

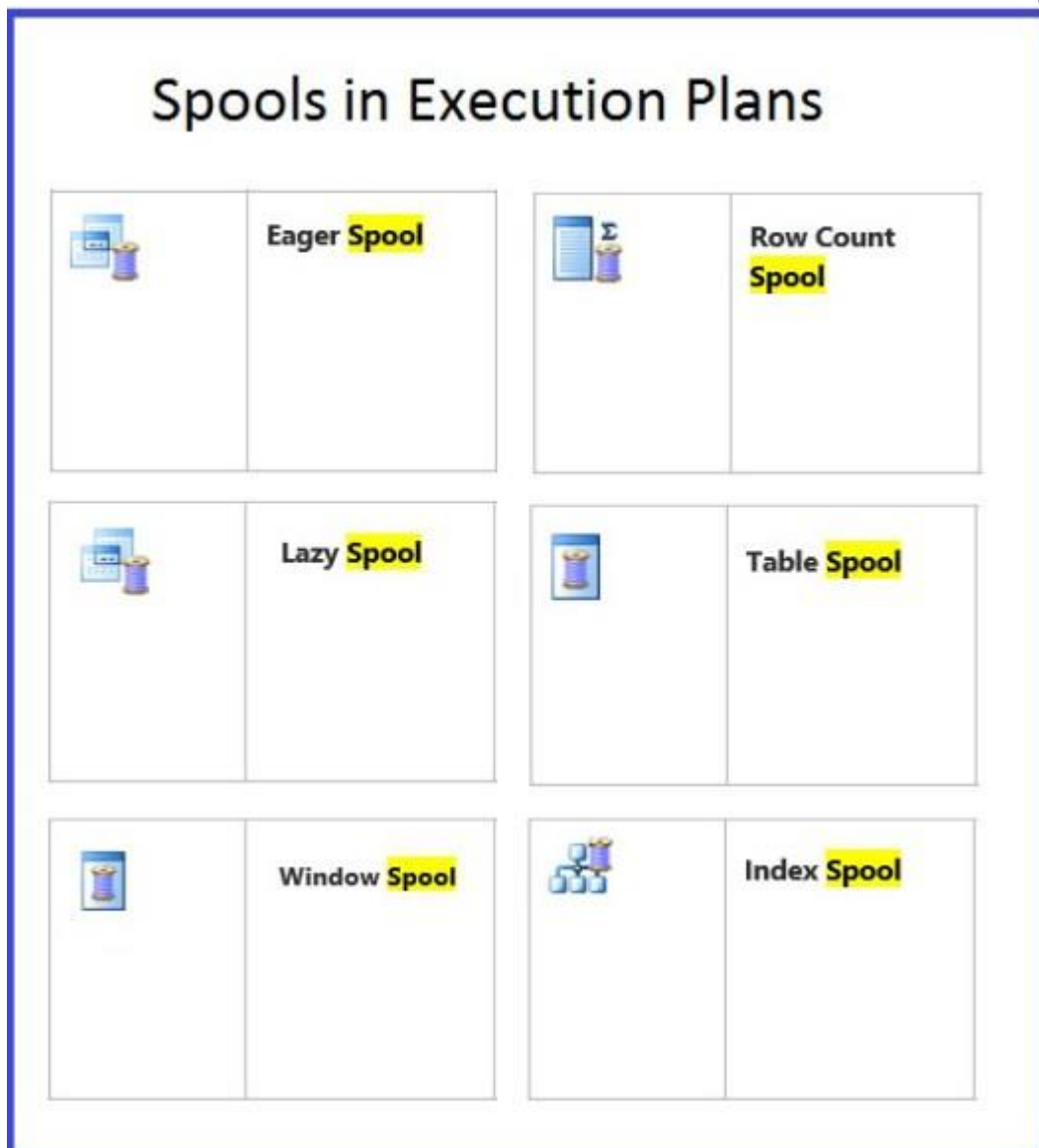
## **Why we have spools in execution plan?**

- This is because you didn't create proper indexes perhaps. Example Unique key, Primary Key, etc
- you didn't provide correct information to the query processor about the uniqueness of your data.
- Sometimes even you have unique key or primary key on your table, optimizer losses it along the way due to the behaviour of various iterators in the execution plan. We can bring this case by using keyword distinct to tell the optimizer that the data is unique and it doesn't have to cache.
- Recursion while updating data (The Halloween problem)

## **Are spools bad?**



1. Well Optimizer uses spools as a performance optimization technique. So effectively we cannot say it a bad or good.
2. Consider a scenario where you have pick all data from a physical table and put that into tempdb and then optimizer queries it, and again pick another table put that in the tempdb and then queries, Now will that a fast query? Probably not. So spools are not good in most of the cases. I will explain scenarios like how we can identify bad things like spools and how to avoid them in the upcoming posts.



**Question – What is a Partition Column in SQL Server?**

Partition column is a concept used in Table Partitioning. Table partitioning is a way to divide a large table into smaller, more manageable parts without creating separate tables for each part.

Data in a partitioned table is partitioned based on a single column, that column is called partition column, often called the partition key. Note that only one column can be used as the partition column, but it is possible to use a computed column.

Most of the time a date column or a key column is used as the partition column. SQL Server places rows in the correct partition based on the values in the date column. It is important to choose a partition column that is almost always used as a predicate in queries.

When the partition column is used as a filter in queries, SQL Server can approach only the relevant partitions and we will get excellent performance.

**Question – If the partition column value is NULL, then in which partition rows are inserted or will the insert statement will give error on insertion?**

It will not throw any error. If the partition column value is NULL then rows are placed in the first partition.

**Question – What is Range right and range left in SQL Server?**

Partition functions are created as either range left or range right to specify whether the boundary values belong to their left or right partitions:

1. Range left means that the actual boundary value belongs to its left partition, it is the last value in the left partition.
2. Range right means that the actual boundary value belongs to its right partition, it is the first value in the right partition.

```
--  
--Example Partition Function  
CREATE PARTITION FUNCTION Sales (DATE)  
AS RANGE RIGHT FOR VALUES ('2011-01-01', '2012-01-01', '2013-01-01',  
'2014-01-01', '2015-01-01');  
--
```

## Question – What is the Partition Scheme?

The partition scheme is used to map the logical partitions to physical filegroups. We can map each partition to its own filegroup or all partitions to one filegroup.