**Most commonly asked T-SQL interview questions…**

1. **A. What are statistics? Index vs Column Statistics?**

# SQL Server Query Optimizer uses *statistics* to estimate the distribution of values in one or more columns of a table or index views, and the number of rows (called cardinality) to create a high-quality query execution plan

One major difference between ***indexes and column statistics*** is that indexes are permanent objects that are updated when changes to the underlying table occur. Column statistics are not updated. If your data is constantly changing, the statistics manager might need to rely on stale column statistics.

1. **A.Clustered vs Non-Clustered Indexes?**

* SQL Index Types

A ***clustered index*** alters the way that the rows are physically stored. When you create a clustered index on a column (or a number of columns), the SQL server sorts the table's rows by that column(s). ... A ***non****-****clustered index***, on the other hand, does not alter the way the rows are stored in the table.

1. **What is Filtered index, Unique Index, Index with Included columns, covering index?**

* ***Filtered Index***is a new feature in SQL SERVER. Filtered Index is used to index a portion of rows in a table that means it applies a filter on INDEX which improves query performance, reduces index maintenance costs, and reduce index storage costs compared with full-table

***A unique index***ensures the index key columns do not contain any duplicate values. A unique index may consist of one or many columns. If a unique index has one column, the values in this column will be unique. ... A unique index can be clustered or non-clustered.

***Indexes with included columns*** provide the greatest benefit when covering the query. This means that the index includes all columns referenced by your query, as you can add columns with data types, number or size not allowed as index key columns.

1. **What is index fragmentation and how to fix it?**

* ***Fragmentation*** happens when the logical order of pages in an index does not match the physical order in the data file. Because fragmentation can affect the performance of some queries, you need to monitor the fragmentation level of your indexes and, if required, perform re-organize or rebuild operations on them
* You can***fix index fragmentation***by rebuilding or defragmenting the index. If the fragmentation level is low, you can defragment the index. If it's high, then you should rebuild the index. You can use SQL Server Management Studio (SSMS) or T-SQL to get started managing index fragmentation.

1. **Explain Columnstore indexes and types**.

* ***Columnstore index*** is a new type of index introduced in SQL Server 2012. It is a column-based non-clustered index geared toward increasing query performance for workloads that involve large amounts of data, typically found in data warehouse fact tables

1. **Name all 6 types of Integrity constraints**

Constraints are used to make sure that the integrity of data is maintained in the database. Following are the most used constraints that can be applied to a table.

* NOT NULL
* UNIQUE
* PRIMARY KEY
* FOREIGN KEY
* CHECK
* DEFAULT

1. **Difference between Primary Key and Unique Constraint**

* The difference between a ***UNIQUE constraint***and a***Primary Key*** is that per table you may only have one Primary Key but you may define more than one UNIQUE constraints. Primary Key constraints are not nullable. UNIQUE constraints may be nullable. ... A UNIQUE constraint will generate a unique NON-CLUSTERED INDEX

1. **Difference between Unique Index vs Unique Constraint**

* There is no difference between ***Unique Index*** and ***Unique Constraint****.* ... Unique Constraint creates Unique Index to maintain the constraint to prevent duplicate keys. Unique Index or Primary Key Index are physical structure that maintain uniqueness over some combination of columns across all rows of a table

1. **What is Index Scan?**

* An***index scan*** or ***table scan*** is when SQL Server has to scan the data or index pages to find the appropriate records. A scan is the opposite of a seek, where a seek uses the index to pinpoint the records that are needed to satisfy the query.

1. **What is Heap? B-Tree?**

* A***heap*** is a table without a clustered index. One or more nonclustered indexes can be created on tables stored as a heap. Data is stored in the heap without specifying an order.
* In SQL Server, indexes are organized as***B****-****trees****.* Each page in an index***B****-****tree*** is called an index node. The top node of the***B****-****tree***is called the root node. The bottom nodes in the index are called the leaf nodes. In a clustered index, the leaf nodes contain the data pages of the underlying table

1. **When is the benefit of indexes? What are disadvantages?**

* ***Advantage of Indexes in SQL***
  + The database engine can use indexes to boost performance in a number of different queries. Sometimes these performance improvements are dramatic. An important feature of SQL Server 2000 is a component known as the query optimizer.

***Disadvantages of SQL:***

* Difficulty in Interfacing: Interfacing a SQL database is more difficult than including a couple of lines of code. ...
* Partial Control: The developers who use SQL does not have a full control over the database as a result of the concealed business rules.
* Implementation: ...
* Cost:

1. **What is the maximum number of index per table? Clustered vs Nonclustred?**

* ***999***

SQL Server allows us to create multiple Non-clustered indexes, up to 999 Non-clustered indexes, on each table, with index IDs values assigned to each index starting from 2 for each partition used by the index, as you can find in the sys. partitions table

1. **What is Execution Plan, Difference between Scan and Seek?**

* An ***execution plan***is a visual representation of the operations performed by the database engine in order to return the data required by your query. The execution plan for a query is your view into the SQL Server query optimizer and query engine.
* DIFFERENCES BETWEEN SQL SERVER CLUSTERED ***INDEX SCAN AND INDEX SEEK***. Index scan means it retrieves all the rows from the table and index seek means it retrieves selective rows from the table. INDEX SCAN: ... Thus, a scan is an efficient strategy if the table is small or most of the rows qualify for the predicate.

1. **How to get rid of Key LookUP/BookMark LookUp?**

# If table has clustered index, it is called bookmark lookup (or key lookup); if the table does not have clustered index, but a non-clustered index, it is called RID lookup. You Must remove the lookup from the Execution plan to improve your Performance

1. What are red flags of execution plan?

There are some flags in execution plan which normally reduces the performance of the query. Some of them are given below.

* High Percentage Operations
* Table Scans, Index Scans, Clustered Index Scans
* Spools
* Parallelism operations
* Warnings
* Thick Arrows
* Hash Joins
* Bookmark Lookups
* Sorting

1. **When to use Set Statistics IO/OFF? What are logical Reads?**

* When ***STATISTICS IO*** is ON, statistical information is displayed, and when OFF, the information isn't displayed. After this option is set ON, all Transact-SQL statements return the statistical information until the option is set to OFF.
* The***Logical Reads*** is how many pages of data would be needed to be read from disk. It is the number to focus on when doing I/O related performance tuning.

1. **Can Primary Key have non-clustered index?**

* ***Primary Key*** is defined as a***Non****-****clustered Index***

In this case we will explicitly defined Primary Key as a non-clustered index and it will create it as a non-clustered index. It proves that Primary Key can be non-clustered index.

1. **Differences between Delete and Truncate**

* ***TRUNCATE*** is the DDL statement whereas ***DELETE***is a DML statement. Below are the ***differences*** between the two: ... ***TRUNCATE*** always removes all the rows from a table, leaving the table empty and the table structure intact whereas ***DELETE***may remove conditionally if the where clause is used

1. **Difference between Stor Proc and UDF?**

* ***Procedure*** can return zero or n values whereas function can return one value which is mandatory. Procedures can have input/output parameters for it whereas functions can have only input parameters. ...***UDF*** can be used in the SQL statements anywhere in the WHERE/HAVING/SELECT section where as ***Stored procedures*** cannot be.

1. **Can we call a procedure from a function?**

* Technically, ***calling a stored procedure from a function*** is possible. But remember the purpose of the stored procedure and functions. Purpose of Stored procedure: The stored procedure is used to execute business logic and hence may or may not return a value

1. **Can we call UDF in stor proc?**

* ***UDF*** can have upto 1023 input parameters,***Stored Procedure*** can have upto 2100 input parameters. ***UDF can not Call Stored Procedure*** Only access to Extended **St**ored Procedure. UDF can not run dynamic SQL which are dynamically build in UDF. Temporary Tables can not be used in UDF as well

1. **Can we write DML inside a UDF?**

* ***Directly DML is not possible in UDF it is for sure.*** "Functions have only READ-ONLY Database Access" If DML operations would be allowed in functions then function would be prety similar to stored Procedure. ***No, you can not do Insert/Update/Delete***

1. **What is Subquery?**

* ***A Subquery or Inner query or a Nested query is a query within another SQL query*** and embedded within the WHERE clause. A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.
* ***Correlated Subquery?***

In a SQL database query, a correlated subquery (also known as a synchronized subquery) is a subquery (a query nested inside another query) that uses values from the outer query. Because the subquery may be evaluated once for each row processed by the outer query, it can be slow.

1. **View vs Indexed View?**

* ***Index View Vs Normal View – Learn more on the SQLServerCentral forums.*** ... When you add an index to a view, then SQL Server will internally execute the ... in a normal view, there are lots of restrictions on indexed views.

1. **Can we put order by inside view?**

* ***The ORDER BY clause is invalid in views***, inline functions, derived tables, subqueries, and common table expressions, unless TOP, OFFSET or FOR XML is also specified

1. **What is CTE? Recursive CTE? What are benefits of it? When CTE should not be used?**

* A ***CTE (Common Table Expression)*** is a temporary result set that you can reference within another SELECT, INSERT, UPDATE, or DELETE statement.
* ***Recursive CTEs***. A recursive CTE is a CTE that references itself. In doing so, the initial CTE is repeatedly executed, returning subsets of data, until the complete result is returned. Being able to reference itself is a unique feature and benefit.
* ***CTE*** be used to replace a view which stores the metadata. CTEs help improve readability of the code without compromising performance. They help improve maintainability of the code without compromising performance. They make writing **recursive** code in T-SQL significantly easier than the previous SQL Server versions.
* A ***CTE can reference itself and previously defined CTEs in the same WITH clause. Forward referencing is not allowed***. Specifying more than one ***WITH clause in a CTE is not allowed***. For example, if a CTE\_query\_definition contains a subquery, that subquery cannot contain a nested WITH clause that defines another CTE

1. **What is MAXRECURSION? What is default MAXRECURSION for recursive CTE?**

* The **MAXRECURSION** value specifies the number of times that the CTE can recur before throwing an error and terminating. Advertisement. You can provide the **MAXRECURSION** hint with any value between 0 and 32,767 within your T-**SQL** query, with **MAXRECURSION** value equal to 0 means that no limit is applied to the recursion level

1. What is the limit of OPTION (MAXRECURSION 0)?

* **MAXRECURSION** hint value can be between **0** to 32,767. Specifying it's value as **0** means no **limit**.

1. In Option (MAXRECURSION N)? Can we specify N number > 32767?

* By **default** maximum **recursion** level supported by **CTE** is 100. But **CTE** provides an option to change it by means of the **MAXRECURSION** hint. **MAXRECURSION** hint value can be between 0 to 32,767

1. What are Window Functions?

* **SQL Server Window Functions**. **SQL Server Window Functions** calculate an aggregate value based on a group of rows and return multiple rows for each group. Assign a rank value to each row within a partition of a result, with no gaps in rank values.

1. What are different types of Window functions? Give examples...

* Aggregate Window Functions. SUM(), MAX(), MIN(), AVG(). COUNT()
* Ranking Window Functions. RANK(), DENSE\_RANK(), ROW\_NUMBER(), NTILE()
* Value Window Functions. LAG(), LEAD(), FIRST\_VALUE(), LAST\_VALUE()

1. Difference between Row\_Number vs Rank vs Dense\_Rank?

* The only **difference between RANK**, **DENSE\_RANK** and **ROW\_NUMBER** function **is** when there are duplicate values **in the** column being used in ORDER BY Clause. ... On the other hand, the **DENSE\_RANK** function does not skip ranks if there **is** a tie **between** ranks. Finally, the **ROW\_NUMBER** function has no concern with **ranking**

1. What are benefits of stor procs? Benefits of views?

* By grouping **SQL** statements, a **stored procedure** allows them to be executed with a single call. This minimizes the use of slow networks, reduces network traffic, and improves round-trip response time. OLTP applications, in particular, **benefit** because result set processing eliminates network bottlenecks
* **Views can provide advantages over tables:**

Views can represent a subset of the data contained in a table. ...

Views can join and simplify multiple tables into a single virtual table.

Views can act as aggregated tables, where the database engine aggregates data (sum, average, etc.) ...

Views can hide the complexity of data.

1. What are different types of UDF?

* **There are three types of User-Defined functions in SQL Server:**

Scalar Function.

Inline Function.

Multi-statement Table-valued Function.

1. What are different types of joins? Difference between Inner/Outer(left/right/full)/Cross Joins?

* There are four basic **types** of **SQL joins**: inner, left, right, and full. The easiest and most intuitive way to explain the difference between these four **types** is by using a Venn diagram, which shows all possible logical relations between data sets.
* The main **difference between RIGHT OUTER join** and **LEFT OUTER join**, as there name suggest, is the inclusion of non-matched rows. ... In short result of **LEFT outer join** is **INNER JOIN** + unmatched rows from **LEFT** table and **RIGHT OUTER join** is **INNER JOIN** + unmatched rows from the **right**-hand side table.

1. What is Self Join?

* A **self join** allows you to **join** a table to itself. It is useful for querying hierarchical data or comparing rows within the same table. A **self join** uses the inner **join** or left **join** clause.

1. Joins vs Union? Union vs Union all?

* Both **joins** and **unions** can be used to combine data from one or more tables into a single results. ... Whereas a **join** is used to combine columns from different tables, the **union** is used to combine rows.
* The only difference between **Union** and **Union All** is that **Union All** will not removes duplicate rows or records, instead, it just selects **all** the rows from **all** the tables which meets the conditions of your specifics query and combines them into the result table. ... Whereas, **UNION ALL** works with **all** data type columns

1. What are different types of Set Operators? Union vs Intersect vs Except?
2. **Union**. This set operator is used to combine the outputs of two or more queries into a single set of rows and columns having different records. ...
3. **Union All.** This set operator is used to join the outputs of two or more queries into a single set of rows and columns without the removal of any duplicates. ...
4. **INTERSECT.**

This set operator is availed to retrieve the information which is common in both tables. The number of columns and data type must be same in intersect set operator.

1. **Except**

This set operator is availed to retrieve the information of one table which is not available in another table.

* **UNION**: Combine two or more result sets into a single set, without duplicates. ... **INTERSECT**: Takes the data from both result sets which are in common. **EXCEPT**: Takes the data from the first result set, but not in the second result set (i.e. no matching to each other)

1. Cross Appy vs Outer Apply

* The **APPLY** operator can take one of two forms: **CROSS APPLY** or **OUTER APPLY**. The **CROSS APPLY** operator returns rows from the primary (**outer**) table only if the table-value function produces a result set. ... The **OUTER APPLY** form, on the other hand, returns all rows from the **outer** table, even if the function produces no results.

1. What is Merge statement?

* The **MERGE statement** is used to make changes in one table based on values matched from anther. It can be used to combine insert, update, and delete operations into one **statement**.

1. Joins vs Exist vs In

* **EXISTS** is only used to test if a subquery returns results, **and** short circuits as soon as it does. **JOIN** is used to extend a result set by combining it with additional fields from another table to which there is a relation. In your example, the queries are symantically equivalent.

1. Explain system databases Master/Model/TempDb/Msdb/ResourceDB

* An Introduction to **SQL Server** System **Databases**. **SQL Server** 2008 (and 2005) contain five special **databases**: master, model, tempdb, msdb, and mssqlsystemresource (aka Resource). These **databases** are used by **SQL Server for** its own maintenance and management

1. Difference between .mdf, .ndf and .ldf

* **mdf** is the data file where you place your database. it is the file extension use in the **sql server**. **ndf** is the filegroup in the **sql server**. **ldf** is the log file in the **sql server**. to see the data

1. Temp table vs Table Variable? Can we index Temp table? Can table variable have indexes?
2. Temporary **Tables** are physically created in the tempdb database. These **tables** act as the normal **table** and also can have constraints, index like normal **tables**. **Table Variable** acts like a **variable** and exists for a particular batch of query execution. ... It is created in the memory database but may be pushed out to tempdb
3. **we can** only add a (non PK) **index** to a #**temp table**. If **you** simply reference #tableName in the stored procedure (without creating it), it **will** look into the caller's scope
4. **Table Variable**. You **can**'t add **Indexes** explicitly to the **Table Variable**. But as **Table variables** support Primary and Unique Key constraints, so specifying these constraints during **Table variable** declaration internally creates **Indexes** on the **Table Variable** columns
5. **How do you do performance tuning on slow running query/stor proc?**

* If a developer can diagnose and fix a *slow query*, then there's no reason ... to find slow SQL **queries** and **do performance tuning** in SQL Server. ... **tuning** consists of making **queries** of a relation database **run** as fast as possible.

1. **What are transactions? What are ACID properties of transactions?**

* In the context of **transaction** processing, the acronym **ACID** refers to the four key **properties** of a **transaction**: atomicity, consistency, isolation, and durability. ... After a **transaction** successfully completes, changes to data persist and are not undone, even in the event of a system failure

1. What is sp\_who2? Sp\_lock?

* The **sp\_who2** is a undocumented thus unsupported stroed procedure in SQL server, but widely used inststed of sp\_who to list processes currently active in SQL Server. ... **sp\_who2** usually helps me to track / kill deadlocks in my database.
* The **sp\_lock** system stored procedure is a great tool for checking the amount of locking that occurs on your database system. It returns the number and types of locks that are being held by current active SQL Server sessions.

1. I have opened a nested transaction inside an outer transaction. if I do rollback, which transaction will roll back?

* If the rollback is present in the outermost transaction then everything will be rolled back irrespective of what is written in the inner transactions. If the rollback is present in the inner transaction then we will not have any transaction in the outer most transaction.

1. What are new features of SQL Server 2019?

**Top 10 New Features of SQL Server 2019**

* Intelligent Query Processing Enhancements. ...
* Accelerated Database Recovery (ADR) ...
* AlwaysEncrypted with Secure Enclaves. ...
* Memory-optimized Tempdb Metadata. ...
* Query Store Custom Capture Policies. ...
* Verbose Truncation Warnings. ...
* Resumable Index Build. ...
* Data Virtualization with Polybase.

### Last Actual Execution Plan DMF

### Multiple Internal Performance Improvements

1. What are physical joins in SQL Server? Difference between Nested Loop, Merge and Hash Match?

* **Physical Joins**: These are the **joins** that users don't use/write in their **SQL** queries. Instead these are implemented inside **SQL Server** engine as operators or algorithms to implement the Logical **Joins**. Their types are Nested Loop, Merge and Hash
* An index **nested loops** join performs better than a **merge** join or **hash** join if a small set **of** rows are involved. Whereas, if a large set **of** rows are involved the **Nested Loops** join might not be an optimal choice. ... It does not apply the same **for** the inner table rows though.

1. I have a table with millions of rows. I want to retain only last 5% of the rows? How to do it?

* This is a huge performance improvement if you know for certain that your **rows** are in the most recent TOP **5**% of the **table** for example. Such that even if there are indexes on the **Tables**, it further limits the possibilites to **only 5**% of **rows** in **tables** which **have** 100+ **million** or 1+ **billion rows**.

1. Why loops, cursors and other row by row operations bad? What is Set-Based operation?

* **Disadvantages** of using **Cursor**: So occupies more resources and temporary storage. Each time when a **row** is fetched from the **cursor** it may result in a network round trip. This uses much more network bandwidth than the execution of a single SQL statement like SELECT or DELETE etc that makes only one round trip.
* **Set based** means that the **operations** are perform on a **set** of values but i am not able to relate this defination with **operations** being perform with SQL Server

1. Which is better a CTE or a subquery? Why?

* Both CTEs and Sub Queries have pretty much the same performance and function. **CTE's** have an advantage over using a **subquery** in that you can use recursion in a **CTE**. The biggest advantage of using **CTE** is readability. CTEs can be reference multiple times in the same statement where as **sub query** cannot

1. What is linked server?

* **Linked Servers** allows you to connect to other **database** instances on the same **server** or on another machine or remote **servers**. It allows **SQL Server** to execute **SQL** scripts against OLE DB data sources on remote **servers** using OLE DB providers

1. Can we use an ORDER BY clause in a CTE?

* You **can** not **use ORDER** BY inside the **cte**. However, you **can** have **ORDER** BY in the query which is selecting from **cte**, e.g. This is a huge conceptual error! A **CTE**, or VIEW or derived table, etc

1. What is identity column?

* An **identity column** is a **column** (also known as a **field**) in a database table that is made up of values generated by the database. This is much like an AutoNumber **field** in Microsoft Access or a sequence in Oracle. ... In Microsoft **SQL Server** you have options for both the seed (starting value) and the increment.

1. ISNULL() vs COALESCE()?

* One apparent advantage that **COALESCE** has over **ISNULL** is that it supports more than two inputs, whereas **ISNULL** supports only two. Another advantage of **COALESCE** is that it's a standard function (namely, defined by the ISO/ANSI SQL standards), whereas **ISNULL** is T-SQL–specific

1. How to delete duplicate rows in SQL Server?

* **To delete the duplicate rows from the table in SQL Server, you follow these steps:**

1.Find duplicate rows using GROUP BY clause or ROW\_NUMBER() function.

2.Use DELETE statement to remove the duplicate rows.

1. Different ways of calculating 3rd Highest salary?

* By default ORDER BY clause print rows in ascending order, since we need the **highest salary** at the top, we have used ORDER BY DESC, which will display **salaries** in descending order. Again DISTINCT is used to remove duplicates. The outer query will then pick the top most **salary**, which would be your **Nth highest salary**

1. Different ways of calculating running total?

* There are several **ways** to **calculate** a **running total** in SQL. In this article, we will cover two **methods**: Joins, and Window Functions.

**Calculate A Running Total in SQL using an INNER JOIN**

1. Get rows for the **running total**.
2. Setup details for the **running total** using inner joins.
3. **Calculate** the **running total** by summarizing data.
4. A string, with more than hundreds of characters, has a special character X. X is repeated many times in this string. Write a select statement to count the number of Xs in this string..

# Repeat a string:repeat("ha", 5) => "hahahahaha"

1. What is database normalization? Why important? What are different forms?

* **Data normalization** helps you design new **databases** to meet these goals or to test **databases** to see whether they meet the goals. Sometimes **database** designers refer to these goals in terms such as **data** integrity, referential integrity, or keyed **data** access. Ideally, you **normalize data** before you create **database** tables.

**The total normalization process includes 8 normal forms.**

* First Normal Form (1NF)
* Second Normal Form (2NF)
* Third Normal Form (3NF)
* Boyce-Codd Normal Form (BCNF)
* Fourth Normal Form (4NF)
* Fifth Normal Form (5NF)
* Domain/Key Normal Form(DKNF)
* Sixth Normal Form(6NF)

1. What is referential integrity? Cascading referential integrity?

* The **Referential Integrity** constraint requires that values in a foreign key column must either be present in the primary key that is referenced by the foreign key or they must be null
* The **Cascading Referential Integrity** Constraints in SQL Server are the foreign key constraints which tell SQL Server to perform certain actions whenever a user attempts to delete or update a primary key to which an existing foreign keys point.

1. What is the use of DBCC commands?

* **DBCC commands** act as database console **commands** which means that they are **used** to check the consistency of SQL server database. They are **use** for maintenance of database, tables, filegroups, indexes. They also can be **used** to perform the validation operation on tables, filegroups, indexes, catalogs.

1. What is Database Collation? Case sensitive vs case incensitive?

* In **database** systems, **Collation** specifies how data is sorted and compared in a **database**. **Collation** provides the sorting rules, case, and accent sensitivity properties for the data in the **database**. ... **Collation** is also used to determine how accents are treated, as well as character width and Japanese kana characters
* A **case**-**sensitive** program that expects you to enter all commands in uppercase will not respond correctly if you enter one or more characters in lowercase. ... Programs that do not **distinguish between** uppercase and lowercase are said to be **case**-**insensitive**.

1. Please give examples for one-to-one, one-to-many and many-to-many relationships?

* In a **one-to-many relationship**, the local table has **one** row that may be associated with **many** rows in another table. ... In the opposite **many**-to-**one relationship**, the local table may have **many** rows that are associated with **one** row in another table. In our example, **many** Order s may be associated to **one** Customer

1. What is the difference between a HAVING CLAUSE and a WHERE CLAUSE?

* The **difference between** WHERE and **HAVING clause** are: The WHERE **clause** is used to filter rows before the grouping is performed. The **HAVING clause** is used to filter rows after the grouping is performed. It often includes the result of aggregate functions and is used with GROUP BY

1. STUFF function vs REPLACE function.

* **Difference between STUFF and REPLACE in SQL Server**. **STUFF** is used to **replace** the part of string with some other string. **REPLACE** is used to **replace** all the occurances of the given pattern **in a** string

1. What is Try.. Catch block in Tsql?

* "**Try**" and "**catch**" are keywords that represent the handling of exceptions due to data or coding errors during program execution. A **try block** is the **block** of code in which exceptions occur. A **catch block** catches and handles **try block** exceptions.

1. Can we rewrite subqueries as joins?

* ﻿A **subquery can** be used with **JOIN** operation. ... The temporary table from the **subquery** is given an alias so that **we can** refer to it in the outer select statement. Note that the left and right table of the **join** keyword must both return a common key that **can** be used for the **join**.

1. How to get random 100 rows from a table?

* **select** top(20) \* from Orders order by newid() ...
* TABLESAMPLE [SYSTEM] (sample\_number [ PERCENT | **ROWS** ] ) [ REPEATABLE (repeat\_seed) ] ...
* **Select** \* from Orders TABLESAMPLE(20 **rows**) ...
* **Select** top(500) \* from Orders TABLESAMPLE(1000 **rows**) ...
* **select** \* from Orders TABLESAMPLE(30 **rows**) repeatable(55)

1. What are Magic Tables in SQL Server?

* **Magic Tables** are invisible **tables** or virtual **tables**. You can see them only with the help Triggers in **SQL Server**. **Magic Tables** are those **tables** which allow you to hold inserted, deleted and updated values during insert, delete and update DML operations on a **table in SQL Server**

1. What is TABLESAMPLE?

* **TABLESAMPLE** is a clause for a query which can be used to select a pseudo-random number of rows from a table, based upon a percentage or a number of rows and an optional seed number – if a repeatable result is required.

1. What is SQL injection?

* **SQL Injection** (SQLi) is a type of an **injection** attack that makes it possible to execute malicious **SQL** statements. .

1. What is BulkCopy?

* Microsoft **SQL** Server includes a popular command-line utility named bcp for quickly bulk copying large files into tables or views in **SQL** Server databases. The **SqlBulkCopy** class allows you to write managed code solutions that provide similar functionality. ... A single **bulk copy** operation. Multiple **bulk copy** operations

1. What are DMV? (Dynamic Management Views)

* **Dynamic management views** and functions return server state information that can be used to monitor the health of a server instance, diagnose problems, and tune performance.

1. Difference between char/varchar/nchar/nvharchar?

* **nchar** and **char** pretty much operate in exactly the same way as each other, as do **nvarchar** and **varchar**. The only **difference between** them is that **nchar**/**nvarchar** store Unicode characters (essential if you require the use of extended character sets) whilst **varchar** does not

1. What lock, block, deadlock? what is KILL command?

* **Deadlock** occurs when one process is blocked and waiting for a second process to complete its work and release **locks**, while the second process at the same time is blocked and waiting for the first process to release the **lock**
* **SQL Server Kill Command**

Use this list of processes to confirm the SPID you wish to kill. To kill a process, type the following TSQL statement in the query window and execute it: KILL <SPID> GO. This will end the process and all uncompleted transactions will begin the rollback process.

1. Sql Profiler vs Extended Events?

* **Extended Events** has less resource overhead compared to SQL Trace/**Profiler**. Anyone who has used SQL **Profiler** has no doubt run into the performance issues it can cause due to the resources it requires, especially when running it locally.

1. What are different SQL Server High Availability (HA) solutions? Difference between Replication/Log Shipping/Mirroring/Clustering/Always ON Availability Groups..

* **High Availability** (HA) is the **solution**\process\technology to make the application\database available 24x7 under either planned or un-planned outages. Mainly, there are five **options** in MS **SQL Server** to achieve\setup **high availability solution** for the databases.
* **Log Shipping**::Both committed and uncommitted transactions are transferred to the secondary database. **Mirroring**::Only committed transactions are transferred to the **mirror** database. Replication::Only committed transactions are transferred to the subscriber database.

1. SQL HINTS. NO Lock