Sql server full interview questions

What is the difference between DBMS and RDBMS?

**DBMS – Data Base Management System**

**RDBMS – Relational Data Base Management System or Relational DBMS**

A DBMS has to be persistent, that is it should be accessible when the program created the data ceases to exist or even the application that created the data restarted. A DBMS also has to provide some uniform methods independent of a specific application for accessing the information that is stored.

RDBMS adds the additional condition that the system supports a tabular structure of the data, with enforced relationships between the tables. This excludes the databases that don’t support a tabular structure or don’t enforce relationships between tables.

DBMS does not impose any constraints or security with regard to data manipulation it is used or the programmer responsibility to ensure the ACID PROPERTY of the database whereas the RDBMS is more with this regards because RDBMS defines the integrity constraint for the purpose of holding ACID PROPERTY. RDBMS may be or may not be Client Server Database System.

Examples :

DBMS – File System, XML

RDBMS – SQL Server, Oracle

**What are the Properties of the Relational Tables?**

Relational tables have the following six properties:

* Values are atomic.
* Column values are of the same kind.
* Each row is unique.
* The sequence of columns is insignificant.
* The sequence of rows is insignificant.
* Each column must have a unique name.

**What is Normalization?**

Database normalization is a data design and organization process applied to data structures based on rules that help building relational databases. In relational database design, the process of organizing data to minimize redundancy is called normalization. Normalization usually involves dividing a database into two or more tables and defining relationships between the tables. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships.

**What is De-normalization?**

De-normalization is the process of attempting to optimize the performance of a database by adding redundant data. It is sometimes necessary because current DBMSs implement the relational model poorly. A true relational DBMS would allow for a fully normalized database at the logical level, while providing physical storage of data that is tuned for high performance. De-normalization is a technique to move from higher to lower normal forms of database modeling in order to speed up database access.

**How is ACID property related to Database?**

ACID (an acronym for Atomicity Consistency Isolation Durability) is a concept that Database Professionals generally look for while evaluating databases and application architectures. For a reliable database, all this four attributes should be achieved.

**Atomicity** is an all-or-none proposition.

**Consistency** guarantees that a transaction never leaves your database in a half-finished state.

**Isolation** keeps transactions separated from each other until they are finished.

**Durability** guarantees that the database will keep track of pending changes in such a way that the server can recover from an abnormal termination. ([**Read more here**](http://blog.sqlauthority.com/2007/12/09/sql-server-acid-atomicity-consistency-isolation-durability/))

**What are the Different Normalization Forms?**

**1NF: Eliminate Repeating Groups**

Make a separate table for each set of related attributes, and give each table a primary key. Each field contains at most one value from its attribute domain.

**2NF: Eliminate Redundant Data**

If an attribute depends on only part of a multi-valued key, then remove it to a separate table.

**3NF: Eliminate Columns Not Dependent On Key**

If attributes do not contribute to a description of the key, then remove them to a separate table. All attributes must be directly dependent on the primary key. ([**Read more here**](http://blog.sqlauthority.com/2007/11/26/sql-server-rules-of-third-normal-form-and-normalization-advantage-3nf/))

**BCNF: Boyce-Codd Normal Form**

If there are non-trivial dependencies between candidate key attributes, then separate them out into distinct tables.

**4NF: Isolate Independent Multiple Relationships**

No table may contain two or more 1:n or n:m relationships that are not directly related.

**5NF: Isolate Semantically Related Multiple Relationships**

There may be practical constrains on information that justify separating logically related many-to-many relationships.

**ONF: Optimal Normal Form**

A model limited to only simple (elemental) facts, as expressed in Object Role Model notation.

**DKNF: Domain-Key Normal Form**

A model free from all modification anomalies is said to be in DKNF.

Remember, these normalization guidelines are cumulative. For a database to be in 3NF, it must first fulfill all the criteria of a 2NF and 1NF database.

**What is a Stored Procedure?**

A stored procedure is a named group of SQL statements that have been previously created and stored in the server database. Stored procedures accept input parameters so that a single procedure can be used over the network by several clients using different input data. And when the procedure is modified, all clients automatically get the new version. Stored procedures reduce network traffic and improve performance. Stored procedures can be used to help ensure the integrity of the database.

e.g. sp\_helpdb, sp\_renamedb, sp\_depends etc.

**What is a Trigger?**

A trigger is a SQL procedure that initiates an action when an event (INSERT, DELETE or UPDATE) occurs. Triggers are stored in and managed by the DBMS. Triggers are used to maintain the referential integrity of data by changing the data in a systematic fashion. A trigger cannot be called or executed; DBMS automatically fires the trigger as a result of a data modification to the associated table. Triggers can be considered to be similar to stored procedures in that both consist of procedural logic that is stored at the database level. Stored procedures, however, are not event-drive and are not attached to a specific table as triggers are. Stored procedures are explicitly executed by invoking a CALL to the procedure while triggers are implicitly executed. In addition, triggers can also execute stored procedures.

*Nested Trigger*: A trigger can also contain INSERT, UPDATE and DELETE logic within itself; so when the trigger is fired because of data modification, it can also cause another data modification, thereby firing another trigger. A trigger that contains data modification logic within itself is called a nested trigger. ([**Read more here**](http://blog.sqlauthority.com/2007/07/24/sql-server-2005-server-and-database-level-ddl-triggers-examples-and-explanation/))

**What are the Different Types of Triggers?**

There are two types of Triggers.

1)      DML Trigger

There are two types of DML Triggers

1.Instead of Trigger  
Instead of Triggers are fired in place of the triggering action such as an insert, update, or delete.

2. After Trigger  
After triggers execute following the triggering action, such as an insert, update, or delete.

2)      DDL Trigger

This type of trigger is fired against Drop Table, Create Table, Alter Table or Login events. DDL Triggers are always After Triggers.

**What is a View?**

A simple view can be thought of as a subset of a table. It can be used for retrieving data as well as updating or deleting rows. Rows updated or deleted in the view are updated or deleted in the table the view was created with. It should also be noted that as data in the original table changes, so does the data in the view as views are the way to look at parts of the original table. The results of using a view are not permanently stored in the database. The data accessed through a view is actually constructed using standard T-SQL select command and can come from one to many different base tables or even other views.

**What is an Index?**

An index is a physical structure containing pointers to the data. Indices are created in an existing table to locate rows more quickly and efficiently. It is possible to create an index on one or more columns of a table, and each index is given a name. The users cannot see the indexes; they are just used to speed up queries. Effective indexes are one of the best ways to improve performance in a database application. A table scan happens when there is no index available to help a query. In a table scan, the SQL Server examines every row in the table to satisfy the query results. Table scans are sometimes unavoidable, but on large tables, scans have a terrific impact on performance.

**What is a Linked Server?**

Linked Servers is a concept in SQL Server by which we can add other SQL Server to a Group and query both the SQL Server databases using T-SQL Statements. With a linked server, you can create very clean, easy–to-follow SQL statements that allow remote data to be retrieved, joined and combined with local data. Stored Procedures sp\_addlinkedserver, sp\_addlinkedsrvlogin will be used to add new Linked Server. ([**Read more here**](http://blog.sqlauthority.com/2007/10/06/sql-server-executing-remote-stored-procedure-calling-stored-procedure-on-linked-server/))

**What is a Cursor?**

A cursor is a database object used by applications to manipulate data in a set on a row-by-row basis, instead of the typical SQL commands that operate on all the rows in the set at one time.

In order to work with a cursor, we need to perform some steps in the following order:

* Declare cursor
* Open cursor
* Fetch row from the cursor
* Process fetched row
* Close cursor
* Deallocate cursor ([**Read more here**](http://blog.sqlauthority.com/2008/04/19/sql-server-is-cursor-database-object-or-datatype/))

**What is Collation?**

Collation refers to a set of rules that determine how data is sorted and compared. Character data is sorted using rules that define the correct character sequence with options for specifying case sensitivity, accent marks, Kana character types, and character width. ([**Read more here**](http://blog.sqlauthority.com/2007/04/24/sql-server-2005-collation-explanation-and-translation/))

## What is the Difference between a Function and a Stored Procedure?

UDF can be used in the SQL statements anywhere in the WHERE/HAVING/SELECT section, whereas Stored procedures cannot be. UDFs that return tables can be treated as another rowset. This can be used in JOINs with other tables. Inline UDF’s can be thought of as views that take parameters and can be used in JOINs and other Rowset operations.

## What is subquery? Explain the Properties of a Subquery?

Subqueries are often referred to as sub-selects as they allow a SELECT statement to be executed arbitrarily within the body of another SQL statement. A subquery is executed by enclosing it in a set of parentheses. Subqueries are generally used to return a single row as an atomic value although they may be used to compare values against multiple rows with the IN keyword.

A subquery is a SELECT statement that is nested within another T-SQL statement. A subquery SELECT statement if executed independently of the T-SQL statement, in which it is nested, will return a resultset. This implies that a subquery SELECT statement can stand alone, and it does not depend on the statement in which it is nested. A subquery SELECT statement can return any number of values and can be found in the column list of a SELECT statement, and FROM, GROUP BY, HAVING, and/or ORDER BY clauses of a T-SQL statement. A subquery can also be used as a parameter to a function call. Basically, a subquery can be used anywhere an expression can be used. ([**Read more here**](http://blog.sqlauthority.com/2007/09/07/sql-server-correlated-and-noncorrelated-subquery-introduction-explanation-and-example/))

## What are Different Types of Join?

### Cross Join

A cross join that does not have a WHERE clause produces the Cartesian product of the tables involved in the join. The size of a Cartesian product result set is the number of rows in the first table multiplied by the number of rows in the second table. The common example is when company wants to combine each product with a pricing table to analyze each product at each price.

### Inner Join

A join that displays only the rows that have a match in both joined tables is known as inner Join. This is the default type of join in the Query and View Designer.

### Outer Join

A join that includes rows even if they do not have related rows in the joined table is an Outer Join. You can create three different outer join to specify the unmatched rows to be included:

* Left Outer Join: In Left Outer Join, all the rows in the first-named table, i.e. “left” table, which appears leftmost in the JOIN clause, are included. Unmatched rows in the right table do not appear.
* Right Outer Join: In Right Outer Join, all the rows in the second-named table, i.e. “right” table, which appears rightmost in the JOIN clause are included. Unmatched rows in the left table are not included.
* Full Outer Join: In Full Outer Join, all the rows in all joined tables are included, whether they are matched or not.

### Self Join

This is a particular case when one table joins to itself with one or two aliases to avoid confusion. A self join can be of any type, as long as the joined tables are the same. A self join is rather unique in that it involves a relationship with only one table. The common example is when company has a hierarchal reporting structure whereby one member of staff reports to another. Self Join can be Outer Join or Inner Join. ([**Read more here**](http://blog.sqlauthority.com/2007/06/03/sql-server-2005-explanation-and-example-self-join/))

## What are Primary Keys and Foreign Keys?

Primary keys are the unique identifiers for each row. They must contain unique values and cannot be null. Due to their importance in relational databases, Primary keys are the most fundamental aspect of all keys and constraints. A table can have only one primary key.

Foreign keys are a method of ensuring data integrity and manifestation of the relationship between tables.

## What is User-defined Functions? What are the types of User-defined Functions that can be created?

User-defined Functions allow defining its own T-SQL functions that can accept zero or more parameters and return a single scalar data value or a table data type.

Different Types of User-Defined Functions created are as follows:

**Scalar User-defined Function**

A scalar user-defined function returns one of the scalar data types. Text, ntext, image and timestamp data types are not supported. These are the type of user-defined functions that most developers are used to in other programming languages.

**Inline Table-Value User-defined Function**

An Inline table-value user-defined function returns a table data type and is an exceptional alternative to a view as the user-defined function can pass parameters into a T-SQL select command and in essence provide us with a parameterized, non-updateable view of the underlying tables.

**Multi-Statement Table-Value User-defined Function**

A multi-statement table-value user-defined function returns a table, and it is also an exceptional alternative to a view as the function can support multiple T-SQL statements to build the final result where the view is limited to a single SELECT statement. Also, the ability to pass parameters into a T-SQL select command or a group of them gives us the capability to in essence create a parameterized, non-updateable view of the data in the underlying tables. Within the create function command, you must define the table structure that is being returned. After creating this type of user-defined function, It can be used in the FROM clause of a T-SQL command unlike the behavior encountered while using a stored procedure which can also return record sets. ([**Read here for example**](http://blog.sqlauthority.com/2007/09/08/sql-server-udf-user-defined-function-get-number-of-days-in-month/))

## What is an Identity?

Identity (or AutoNumber) is a column that automatically generates numeric values. A start and increment value can be set, but most DBAs leave these at 1. A GUID column also generates unique keys. Updated based on the comment of [**Aaron Bertrand**](http://blog.sqlauthority.com/2011/07/05/sql-server-interview-questions-and-answers-frequently-asked-questions-day-5-of-31/#comment-146004). ([**Blog**](http://sqlblog.com/blogs/aaron_bertrand/))

## What is DataWarehousing?

* Subject-oriented, which means that the data in the database is organized so that all the data elements relating to the same real-world event or object are linked together;
* Time-variant, which means that the changes to the data in the database are tracked and recorded so that reports can be produced showing changes over time;
* Non-volatile, which means that data in the database is never over-written or deleted, once committed, the data is static, read-only, but retained for future reporting.
* Integrated, which means that the database contains data from most or all of an organization’s operational applications, and that this data is made consistent.

## What languages BI uses to achieve the goal?

BI uses following languages for achieve the Goal.

### MDX – Multidimensional Expressions:

This language is used for retrieving data from SSAS cubes. It looks very similar to T-SQL, but it is very different in the areas of conceptualization and implementation.

### DMX – Data Mining Extensions:

This is again used for SSAS, but rather than cubes it is used for data mining structures. This language is more complicated than MDX. Microsoft has provided many wizards in its BI tools, which further reduced number of experts for learning this language, which deals with data mining structures.

### XMLA – XML for Analysis:

This is mainly used for SSAS administrative tasks. It is quite commonly used in administration tasks such as backup or restore database, copy and move database, or for learning Meta data information. Again, MS BI tools provide a lot of wizards for the same.  
[**(Read More Here)**](http://blog.sqlauthority.com/2009/07/06/sql-server-languages-for-bi-mdx-dmx-xmla/)

## What is Standby Servers? Explain Types of Standby Servers.

Standby Server is a type of server that can be brought online in a situation when Primary Server goes offline and application needs continuous (high) availability of the server. There is always a need to set up a mechanism where data and objects from primary server are moved to secondary (standby) server. This mechanism usually involves the process of moving backup from the primary server to the secondary server using T-SQL scripts. Often, database wizards are used to set up this process.

Different types of standby servers are given as follows:

### 1) Hot Standby:

Hot Standby can be achieved in the SQL Server using SQL Server 2005 Enterprise Edition and the later enterprise versions. SQL Server 2005 has introduced Mirroring of database that can be configured for automatic failover in a disaster situation. In the case of synchronous mirroring, the database is replicated to both the servers simultaneously. This is a little expensive but provides the best high availability. In this case, both primary and standby servers have same data all the time.

### 2) Warm Standby:

In Warm Standby, automatic failover is not configured. This is usually set up using Log Shipping or asynchronous mirroring. Sometimes warm standby is lagging by a few minutes or seconds, which results into loss of few latest updates when the primary server fails and secondary server needs to come online. Sometimes a warm standby server that is lagging by a few transactions is brought back to the current state by applying the recent transaction log.

### 3) Cold Standby:

Code Standby servers need to be switched manually, and sometimes all the backups as well as the required OS need to be applied. Cold Standby just physically replaces the previous server.

[**(Read more here)**](http://blog.sqlauthority.com/2009/07/11/sql-server-standby-servers-and-types-of-standby-servers/)

## What is Dirty Read?

A dirty read occurs when two operations, say, read and write occur together giving the incorrect or unedited data. Suppose, A changed a row but did not committed the changes. B reads the uncommitted data but his view of the data may be wrong so that is Dirty Read.

## Why can’t I use Outer Join in an Indexed View?

Rows can logically disappear from an indexed view based on OUTER JOIN when you insert data into a base table. This makes incrementally updating OUTER JOIN views relatively complex to implement, and the performance of the implementation would be slower than for views based on standard (INNER) JOIN.[**(Read More Here)**](http://blog.sqlauthority.com/2009/09/25/sql-server-outer-join-in-indexed-view-question-to-readers/)

## What is the Correct Order of the Logical Query Processing Phases?

The correct order of the Logical Query Processing Phases is as follows:

1. FROM  
2. ON  
3. OUTER  
4. WHERE  
5. GROUP BY  
6. CUBE | ROLLUP  
7. HAVING  
8. SELECT  
9. DISTINCT  
10. TOP  
11. ORDER BY

[**(Read more here)**](http://blog.sqlauthority.com/2009/10/10/sql-server-download-logical-query-processing-poster/)

## Which TCP/IP port does the SQL Server run on? How can it be Changed?

SQL Server runs on port 1433. It can be changed from the Network Utility TCP/IP properties –> Port number, both on client and the server.

## What are the Difference between Clustered and a Non-clustered Index?

A clustered index is a special type of index that reorders the way records in the table are physically stored. Therefore, the table can have only one clustered index. The leaf nodes of a clustered index contain the data pages.

A non-clustered index is a special type of index in which the logical order of the index does not match the physical stored order of the rows on disk. The leaf node of a non-clustered index does not consist of the data pages. Instead, the leaf nodes contain index rows. ([**Read more here**](http://blog.sqlauthority.com/2007/04/23/sql-server-understanding-new-index-type-of-sql-server-2005-included-column-index-along-with-clustered-index-and-non-clustered-index/))

## What are the Different Index Configurations a Table can have?

A table can have one of the following index configurations:

* No indexes
* A clustered index
* A clustered index and many non-clustered indexes
* A non-clustered index
* Many non-clustered indexes

## What are Different Types of Collation Sensitivity?

Case sensitivity – A and a, B and b, etc.

Accent sensitivity – a and á, o and ó, etc.

Kana Sensitivity – When Japanese Kana characters Hiragana and Katakana are treated differently, it is called Kana sensitive.

Width sensitivity – When a single-byte character (half-width) and the same character represented as a double-byte character (full-width) are treated differently, it is width sensitive. ([**Read more here**](http://blog.sqlauthority.com/2007/04/30/case-sensitive-sql-query-search/))

## What is OLTP (Online Transaction Processing)?

In OLTP –(online transaction processing) systems, relational database design uses the discipline of data modeling and generally follows the Codd rules of data normalization in order to ensure absolute data integrity. Using these rules, complex information is broken down into its most simple structures (a table) where all of the individual atomic level elements relate to each other and satisfy the normalization rules.

## What’s the Difference between a Primary Key and a Unique Key?

Both primary key and unique key enforce uniqueness of the column on which they are defined. But by default, the primary key creates a clustered index on the column, whereas unique key creates a non-clustered index by default. Another major difference is that primary key doesn’t allow NULLs, but unique key allows one NULL only. ([**Read more here**](http://blog.sqlauthority.com/2007/02/05/sql-server-primary-key-constraints-and-unique-key-constraints/))

## What is Difference between DELETE  and TRUNCATE Commands?

Delete command removes the rows from a table on the basis of the condition that we provide with a WHERE clause. Truncate will actually remove all the rows from a table, and there will be no data in the table after we run the truncate command.

### TRUNCATE

* TRUNCATE is faster and uses fewer system and transaction log resources than DELETE. (Read all the points below)
* TRUNCATE removes the data by deallocating the data pages used to store the table’s data, and only the page deallocations are recorded in the transaction log.
* TRUNCATE removes all the rows from a table, but the table structure, its columns, constraints, indexes and so on remains. The counter used by an identity for new rows is reset to the seed for the column.
* You cannot use TRUNCATE TABLE on a table referenced by a FOREIGN KEY constraint.
* Using T-SQL – TRUNCATE cannot be rolled back unless it is used in TRANSACTION. OR TRUNCATE can be rolled back when used with BEGIN … END TRANSACTION using T-SQL.
* TRUNCATE is a DDL Command.
* TRUNCATE resets the identity of the table.

### DELETE

* DELETE removes rows one at a time and records an entry in the transaction log for each deleted row.
* DELETE does not reset Identity property of the table.
* DELETE can be used with or without a WHERE clause
* DELETE activates Triggers if defined on table.
* DELETE can be rolled back.
* DELETE is DML Command.
* DELETE does not reset the identity of the table.

## What are Different Types of Locks?

* **Shared Locks:** Used for operations that do not change or update data (read-only operations), such as a SELECT statement.
* **Update Locks:**Used on resources that can be updated. It prevents a common form of deadlock that occurs when multiple sessions are reading, locking, and potentially updating resources later.
* **Exclusive Locks:** Used for data-modification operations, such as INSERT, UPDATE, or DELETE. It ensures that multiple updates cannot be made to the same resource at the same time.
* **Intent Locks:**Used to establish a lock hierarchy. The types of intent locks are as follows: intent shared (IS), intent exclusive (IX), and shared with intent exclusive (SIX).
* **Schema Locks:**Used when an operation dependent on the schema of a table is executing. The types of schema locks are schema modification (Sch-M) and schema stability (Sch-S).
* **Bulk Update Locks:** Used when bulk-copying data into a table and the TABLOCK hint is specified.

## What are Pessimistic Lock and Optimistic Lock?

**Optimistic Locking** is a strategy where you read a record, take note of a version number and check that the version hasn’t changed before you write the record back. If the record is dirty (i.e. different version to yours), then you abort the transaction and the user can re-start it.

**Pessimistic Locking** is when you lock the record for your exclusive use until you have finished with it. It has much better integrity than optimistic locking but requires you to be careful with your application design to avoid Deadlocks.

## When is the use of UPDATE\_STATISTICS command?

This command is basically used when a large amount of data is processed. If a large amount of deletions, modifications or Bulk Copy into the tables has occurred, it has to update the indexes to take these changes into account. UPDATE\_STATISTICS updates the indexes on these tables accordingly.

## What is the Difference between a HAVING clause and a WHERE clause?

They specify a search condition for a group or an aggregate. But the difference is that HAVING can be used only with the SELECT statement. HAVING is typically used in a GROUP BY clause. When GROUP BY is not used, HAVING behaves like a WHERE clause. Having Clause is basically used only with the GROUP BY function in a query, whereas WHERE Clause is applied to each row before they are part of the GROUP BY function in a query. ([**Read more here**](http://blog.sqlauthority.com/2007/07/04/sql-server-definition-comparison-and-difference-between-having-and-where-clause/))

## What is Connection Pooling and why it is Used?

To minimize the cost of opening and closing connections, ADO.NET uses an optimization technique called connection pooling.

The pooler maintains ownership of the physical connection. It manages connections by keeping alive a set of active connections for each given connection configuration. Whenever a user calls Open on a connection, the pooler looks for an available connection in the pool. If a pooled connection is available, it returns it to the caller instead of opening a new connection. When the application calls Close on the connection, the pooler returns it to the pooled set of active connections instead of closing it. Once the connection is returned to the pool, it is ready to be reused on the next Open call.

## What are the Properties and Different Types of Sub-Queries?

### Properties of a Sub-Query

* A sub-query must be enclosed in the parenthesis.
* A sub-query must be put on the right hand of the comparison operator, and
* A sub-query cannot contain an ORDER BY clause, however sub-query can use ORDER BY when used with TOP clause. Read Comment by [**David Bridge**](http://blog.sqlauthority.com/2011/07/07/sql-server-interview-questions-and-answers-frequently-asked-questions-day-7-of-31/#comment-146597)
* [**A query can contain more than one sub-query.**](http://blog.sqlauthority.com/2011/07/07/sql-server-interview-questions-and-answers-frequently-asked-questions-day-7-of-31/#comment-146597)

### Types of Sub-query

* Single-row sub-query, where the sub-query returns only one row.
* Multiple-row sub-query, where the sub-query returns multiple rows, and
* Multiple column sub-query, where the sub-query returns multiple columns

## What is an SQL Profiler?

SQL Profiler is a graphical tool that allows system administrators to monitor events in an instance of Microsoft SQL Server. You can capture and save data about each event to a file or SQL Server table to analyze later. For example, you can monitor a production environment to see which stored procedures are hampering performances by executing very slowly.

Use SQL Profiler to monitor only the events in which you are interested. If traces are becoming too large, you can filter them based on the information you want, so that only a subset of the event data is collected. Monitoring too many events adds overhead to the server and the monitoring process and can cause the trace file or trace table to grow very large, especially when the monitoring process takes place over a long period of time.

## What are the Authentication Modes in SQL Server? How can it be Changed?

There are two authentication modes in SQL Server.

* Windows Mode
* Mixed Mode – SQL and Windows

To change authentication mode in SQL Server, go to Start -> Programs- > Microsoft SQL Server and click SQL Enterprise Manager to run SQL Enterprise Manager from the Microsoft SQL Server program group. Select the server; then from the Tools menu, select SQL Server Configuration Properties and choose the Security page.

## Which Command using Query Analyzer will give you the Version of SQL Server and Operating System?

**SELECT SERVERPROPERTY('Edition') AS Edition,  
SERVERPROPERTY('ProductLevel') AS ProductLevel,  
SERVERPROPERTY('ProductVersion') AS ProductVersion  
GO**

([**Read more here**](http://blog.sqlauthority.com/2009/06/14/sql-server-2008-ssms-feature-multi-server-queries/))

## What is an SQL Server Agent?

The SQL Server agent plays an important role in the day-to-day tasks of a database administrator (DBA). It is often overlooked as one of the main tools for SQL Server management. Its purpose is to ease the implementation of tasks for the DBA, with its full-function scheduling engine, which allows you to schedule your own jobs and scripts. ([**Read more here**](http://blog.sqlauthority.com/2007/02/26/sql-server-whats-new-in-sql-server-agent-for-microsoft-sql-server-2005/))

## Can a Stored Procedure call itself or a Recursive Stored Procedure? How many levels of SP nesting is possible?

Yes. As T-SQL supports recursion, you can write stored procedures that call themselves. Recursion can be defined as a method of problem solving wherein the solution is arrived at by repetitively applying it to subsets of the problem. A common application of recursive logic is to perform numeric computations that lend themselves to repetitive evaluation by the same processing steps. Stored procedures are nested when one stored procedure calls another or executes managed code by referencing a CLR routine, type, or aggregate. You can nest stored procedures up to 32 levels. Any reference to managed code from a Transact-SQL stored procedure counts as one level against the 32-level nesting limit. Methods invoked from within managed code do not count against this limit. ([**Read more here**](http://msdn.microsoft.com/en-us/library/ms190607.aspx)) (Courtesy: [**Vinod Kumar**](http://blogs.extremeexperts.com/))

## What is Log Shipping?

Log shipping is the process of automating the backup of database and transaction log files on a production SQL server and then restoring them onto a standby server. All Editions (except Express Edition) supports log shipping. In log shipping, the transactional log file from one server is automatically updated into the backup database on the other server. If one server fails, the other server will have the same db and can be used this as the Disaster Recovery plan. The key feature of log shipping is that it will automatically backup transaction logs throughout the day and automatically restore them on the standby server at defined intervals. (Courtney: [**Rhys**](http://blog.sqlauthority.com/2011/07/08/sql-server-interview-questions-and-answers-frequently-asked-questions-day-8-of-31/#comment-146895)**)**

## Name 3 ways to get an Accurate Count of the Number of Records in a Table?

**SELECT \* FROM table1**

**SELECT COUNT(\*) FROM table1**

**SELECT rows FROM sysindexes WHERE id = OBJECT\_ID(table1) AND indid < 2**

## What does it mean to have QUOTED\_IDENTIFIER ON? What are the Implications of having it OFF?

When SET QUOTED\_IDENTIFIER is ON, identifiers can be delimited by double quotation marks, and literals must be delimited by single quotation marks. When SET QUOTED\_IDENTIFIER is OFF, identifiers cannot be quoted and must follow all T-SQL rules for identifiers. ([**Read more here**](http://blog.sqlauthority.com/2007/03/05/sql-server-quoted_identifier-onoff-and-ansi_null-onoff-explanation/))

## What is the Difference between a Local and a Global Temporary Table?

A local temporary table exists only for the duration of a connection, or if defined inside a compound statement, for the duration of the compound statement.

A globaltemporary table remains in the database accessible across the connections. Once the connection where original global table is declared dropped this becomes unavailable.

## What is the STUFF Function and How Does it Differ from the REPLACE Function?

STUFF function is used to overwrite existing characters using this syntax: STUFF (string\_expression, start, length, replacement\_characters), where string\_expression is the string that will have characters substituted, start is the starting position, length is the number of characters in the string that are substituted, and replacement\_characters are the new characters interjected into the string. REPLACE function is used to replace existing characters of all occurrences. Using the syntax REPLACE (string\_expression, search\_string, replacement\_string), every incidence of search\_string found in the string\_expression will be replaced with replacement\_string.

## What is PRIMARY KEY?

A PRIMARY KEY constraint is a unique identifier for a row within a database table. Every table should have a primary key constraint to uniquely identify each row, and only one primary key constraint can be created for each table. The primary key constraints are used to enforce entity integrity.

## What is UNIQUE KEY Constraint?

A UNIQUE constraint enforces the uniqueness of the values in a set of columns; so no duplicate values are entered. The unique key constraints are used to enforce entity integrity as the primary key constraints.

## What is FOREIGN KEY?

A FOREIGN KEY constraint prevents any actions that would destroy links between tables with the corresponding data values. A foreign key in one table points to a primary key in another table. Foreign keys prevent actions that would leave rows with foreign key values when there are no primary keys with that value. The foreign key constraints are used to enforce referential integrity.

**What is CHECK Constraint?**

A CHECK constraint is used to limit the values that can be placed in a column. The check constraints are used to enforce domain integrity. ([**Read more here**](http://blog.sqlauthority.com/2008/06/22/sql-server-create-check-constraint-on-column/))

**What is NOT NULL Constraint?**

A NOT NULL constraint enforces that the column will not accept null values. The not null constraints are used to enforce domain integrity, as the check constraints.

([**Read more here**](http://blog.sqlauthority.com/2007/10/15/sql-server-explanation-and-understanding-not-null-constraint/))

**What is the difference between UNION and UNION ALL?**

**UNION**The UNION command is used to select related information from two tables, much like the JOIN command. However, when using the UNION command all selected columns need to be of the same data type. With UNION, only distinct values are selected.

**UNION ALL**The UNION ALL command is equal to the UNION command, except that UNION ALL selects all values.

The difference between UNION and UNION ALL is that UNION ALL will not eliminate duplicate rows, instead it just pulls all rows from all the tables fitting your query specifics and combines them into a table. ([**Read more here**](http://blog.sqlauthority.com/2007/03/10/sql-server-union-vs-union-all-which-is-better-for-performance/))

**What is B-Tree?**

The database server uses a B-tree structure to organize index information. B-Tree generally has following types of index pages or nodes:

* *Root node:* A root node contains node pointers to only one branch node.
* *Branch nodes:* A branch node contains pointers to leaf nodes or other branch nodes, which can be two or more.
* *Leaf nodes:* A leaf node contains index items and horizontal pointers to other leaf nodes, which can be many.

**How to get @@ERROR and @@ROWCOUNT at the Same Time?**

If @@Rowcount is checked after Error checking statement, then it will have 0 as the value of @@Recordcount as it would have been reset. And if @@Recordcount is checked before the error-checking statement, then @@Error would get reset. To get @@error and @@rowcount at the same time, include both in same statement and store them in a local variable. SELECT @RC = @@ROWCOUNT, @ER = @@ERROR

**What is a Scheduled Job or What is a Scheduled Task?**

Scheduled tasks let user automate processes that run on regular or predictable cycles. User can schedule administrative tasks, such as cube processing, to run during times of slow business activity. User can also determine the order in which tasks run by creating job steps within a SQL Server Agent job, e.g. back up database and update statistics of the tables. Job steps give user control over flow of execution. If one job fails, then the user can configure SQL Server Agent to continue to run the remaining tasks or to stop execution.

**What are the Advantages of Using Stored Procedures?**

* Stored procedure can reduced network traffic and latency, boosting application performance.
* Stored procedure execution plans can be reused; they staying cached in SQL Server’s memory, reducing server overhead.
* Stored procedures help promote code reuse.
* Stored procedures can encapsulate logic. You can change stored procedure code without affecting clients.
* Stored procedures provide better security to your data.

**What is a Table Called, if it has neither Cluster nor Non-cluster Index? What is it Used for?**

Unindexed table or Heap. Microsoft Press Books and Book on Line (BOL) refers it as Heap. A heap is a table that does not have a clustered index and therefore, the pages are not linked by pointers. The IAM pages are the only structures that link the pages in a table together. Unindexed tables are good for fast storing of data. Many times, it is better to drop all the indexes from table and then do bulk of INSERTs and restore those indexes after that.

**Can SQL Servers Linked to other Servers like Oracle?**

SQL Server can be linked to any server provided it has OLE-DB provider from Microsoft to allow a link, e.g. Oracle has an OLE-DB provider that Microsoft provides to add it as a linked server to the SQL Server group

**What is BCP? When is it Used?**

BCP or BulkCopy is a tool used to copy huge amounts of data from tables and views. BCP does not copy the complete structures from source to destination. BULK INSERT command helps to import a data file into a database table or view in a user-specified format.

## What Command do we Use to Rename a db, a Table and a Column?

### To Rename db

**sp\_renamedb ‘oldname’ , ‘newname**

If someone is using db it will not accept sp\_renmaedb. In that case, first bring db to single user mode using sp\_dboptions. Use sp\_renamedb to rename the database. Use sp\_dboptions to bring the database to multi-user mode.

e.g.

**USE MASTER;  
GO  
EXEC sp\_dboption AdventureWorks, 'Single User', True  
GO  
EXEC sp\_renamedb 'AdventureWorks', 'AdventureWorks\_New'  
GO  
EXEC sp\_dboption AdventureWorks, 'Single User', False  
GO**

### To Rename Table

We can change the table name using sp\_rename as follows:

**sp\_rename 'oldTableName' 'newTableName'**

e.g.

**sp\_RENAME 'Table\_First', 'Table\_Last'  
GO**

### To rename Column

The script for renaming any column is as follows:

**sp\_rename 'TableName.[OldcolumnName]', 'NewColumnName', 'Column'**

e.g.

**sp\_RENAME 'Table\_First.Name', 'NameChange' , 'COLUMN'  
GO**

## What are sp\_configure Commands and SET Commands?

Use sp\_configure to display or change server-level settings. To change the database-level settings, use ALTER DATABASE. To change settings that affect only the current user session, use the SET statement.

e.g.

**sp\_CONFIGURE 'show advanced', 0  
GO  
RECONFIGURE  
GO  
sp\_CONFIGURE  
GO**

You can run the following command and check the advanced global configuration settings.  
**sp\_CONFIGURE 'show advanced', 1  
GO  
RECONFIGURE  
GO  
sp\_CONFIGURE  
GO**

([**Read more here**](http://blog.sqlauthority.com/2008/07/13/sql-server-sp_configure-displays-or-changes-global-configuration-settings/))

## How to Implement One-to-One, One-to-Many and Many-to-Many Relationships while Designing Tables?

One-to-One relationship can be implemented as a single table and rarely as two tables with primary and foreign key relationships. One-to-Many relationships are implemented by splitting the data into two tables with primary key and foreign key relationships.

Many-to-Many relationships are implemented using a junction table with the keys from both the tables forming the composite primary key of the junction table.

## What is Difference between Commit and Rollback when Used in Transactions?

The usual structure of the TRANSACTION is as follows:

BEGIN TRANSACTION

Operations

COMMIT TRANSACTION or ROLLBACK TRANSACTION

When Commit is executed, every statement between BEGIN and COMMIT becomes persistent to database. When Rollback is executed, every statement between BEGIN and ROLLBACK are reverted to the state when BEGIN was executed.

## What is an Execution Plan? When would you Use it? How would you View the Execution Plan?

An execution plan is basically a road map that graphically or textually shows the data retrieval methods chosen by the SQL Server query optimizer for a stored procedure or ad-hoc query, and it is a very useful tool for a developer to understand the performance characteristics of a query or stored procedure since the plan is the one that SQL Server will place in its cache and use to execute the stored procedure or query. Within the Query Analyzer, there is an option called “Show Execution Plan” (in the Query drop-down menu). If this option is turned on, it will display query execution plan in a separate window when the query is ran again.

**What is Difference between Table Aliases and Column Aliases? Do they Affect Performance?**

Usually, when the name of the table or column is very long or complicated to write, aliases are used to refer them.

e.g.

SELECT VeryLongColumnName **col1**  
FROM VeryLongTableName **tab1**

In the above example, col1 and tab1 are the column alias and table alias, respectively. They do not affect the performance at all.

**What is the difference between CHAR and VARCHAR Datatypes?**

VARCHARS are variable length strings with a specified maximum length. If a string is less than the maximum length, then it is stored verbatim without any extra characters, e.g. names and emails. CHARS are fixed-length strings with a specified set length. If a string is less than the set length, then it is padded with extra characters, e.g. phone number and zip codes. For instance, for a column which is declared as VARCHAR(30) and populated with the word ‘SQL Server,’ only 10 bytes will be stored in it. However, if we have declared the column as CHAR(30) and populated with the word ‘SQL Server,’ it will still occupy 30 bytes in database.

**What is the Difference between VARCHAR and VARCHAR(MAX) Datatypes?**

VARCHAR stores variable-length character data whose range varies up to 8000 bytes; varchar(MAX) stores variable-length character data whose range may vary beyond 8000 bytes and till 2 GB. TEXT datatype is going to be deprecated in future versions, and the usage of VARCHAR(MAX) is strongly recommended instead of TEXT datatypes.

**What is the Difference between VARCHAR and NVARCHAR datatypes?**

In principle, they are the same and are handled in the same way by your application. The only difference is that NVARCHAR can handle unicode characters, allowing you to use multiple languages in the database (Arabian, Chinese, etc.). NVARCHAR takes twice as much space when compared to VARCHAR. Use NVARCHAR only if you are using foreign languages.

**Which are the Important Points to Note when Multilanguage Data is Stored in a Table?**

There are two things to keep in mind while storing unicode data. First, the column must be of unicode data type (nchar, nvarchar, ntext). Second, the value must be prefixed with N while insertion. For example,

INSERT INTO table (Hindi\_col) values (N’hindi data’)

**How to Optimize Stored Procedure Optimization?**

There are many tips and tricks for the same. Here are few:

* Include SET NOCOUNT ON statement.
* Use schema name with object name.
* Do not use the prefix “sp\_” in the stored procedure name.
* Use IF EXISTS (SELECT 1) instead of (SELECT \*).
* Use the sp\_executesql stored procedure instead of the EXECUTE statement.
* Try to avoid using SQL Server cursors whenever possible.
* Keep the Transaction as short as possible.
* Use TRY-Catch for error handling.

[**(Read more here)**](http://blog.sqlauthority.com/2010/02/16/sql-server-stored-procedure-optimization-tips-best-practices/)

**What is SQL Injection? How to Protect Against SQL Injection Attack?**

SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.

Here are few methods which can be used to protect again SQL Injection attack:

* Use Type-Safe SQL Parameters
* Use Parameterized Input with Stored Procedures
* Use the Parameters Collection with Dynamic SQL
* Filtering Input parameters
* Use the escape character in LIKE clause
* Wrapping Parameters with QUOTENAME() and REPLACE()

**How to Find Out the List Schema Name and Table Name for the Database?**

We can use following script:

**SELECT '['+SCHEMA\_NAME(schema\_id)+'].['+name+']' AS SchemaTable  
FROM sys.tables**

[**(Read more here)**](http://blog.sqlauthority.com/2009/06/17/sql-server-list-schema-name-and-table-name-for-database/)

**What is CHECKPOINT Process in the SQL Server?**

CHECKPOINT process writes all dirty pages for the current database to disk. Dirty pages are data pages that have been entered into the buffer cache and modified, but not yet written to disk.