**Question:** What is COMMIT & ROLLBACK statement in SQL?

**Answer:** Commit statement helps in termination of the current transaction and does all the changes that occur in transaction persistent and this also commits all the changes to the database. COMMIT we can also use in store procedure.

ROLLBACK do the same thing just terminate the current transaction but one another thing is that the changes made to database are ROLLBACK to the database.

**Question:** What is difference between OSQL and Query Analyzer?

**Answer:** Both are the same but there is little difference OSQL is command line tool which is execute query and display the result same a query analyzer but query analyzer is graphical and OSQL is a command line tool. OSQL have not ability like query analyzer to analyze queries and show static on speed of execution and other useful thing about OSQL is that its helps in scheduling.

**Question:** What is SQL?

**Answer:** The Structured Query Language (SQL) is foundation for all relational database systems. Most of the large-scale databases use the SQL to define all user and administrator interactions. SQL is Non-Procedural language. It allows the user to concentrate on specifying what data is required rather than concentrating on the how to get it.

The DML component of SQL comprises four basic statements:

* SELECT to get rows from tables
* UPDATE to update the rows of tables
* DELETE to remove rows from tables
* INSERT to add new rows to tables

**Question:** What is DTS in SQL Server?

**Answer:** If a organization is big then it is also there that there is multiple option to store data some people are using EXCEL some are using ACCESS and some of they are using SQL SERVER and in some other format also but there a problem is arise that how to merge that data into one format there is different tool are there for doing this function. One of product of SQL SERVER-2000 DTS helps in this problem it provides a set of tool from that tool we can customize are database according to our need DTSRun is a command-prompt utility used to execute existing DTS packages.

**Question:** What is the difference between SQL and Pl/Sql?

**Answer:** We can get modify, Retrieve by single command or statement in SQL but PL/SQL process all SQL statements one at a time. With PL/SQL, an entire block of statements process in a single command line. SQL is structured query language, various queries are used to handle the database in a simplified manner. While pl/sql is procedural language contains various types of variable, functions and procedures and other major difference is Sql as the name suggest it is
just structured query language whereas PLSQL is a combination of Programming language & SQL.

**Question: Can You explain integration between SQL Server 2005 and Visual Studio 2005?**
**Answer:** This integration provide wider range of development with the help of CLR for database server. Because CLR helps developers to get flexibility for developing database applications and also provides language interoperability just like Visual C++, Visual Basic .Net and Visual C# .Net. The CLR helps developers to get the arrays, classes and exception handling available through programming languages such as Visual C++ or Visual C# which is use in stored procedures, functions and triggers for creating database application dynamically and also provide more efficient reuse of code and faster execution of complex tasks. We particularly liked the error-checking powers of the CLR environment, which reduces run-time errors.

**Question: What is Checkpoint in SQL Server?**
**Answer:** When we done operation on SQL SERVER that is not committed directly to the database. All operation must be logged in to Transaction Log files after that they should be done on to the main database. Checkpoint are the point which alert Sql Server to save all the data to main database if no check point is there then log files get full we can use Checkpoint command to commit all data in the SQL SERVER. When we stop the SQL Server it will take long time because Checkpoint is also fired.

**Question: What is the difference between UNION ALL Statement and UNION?**
**Answer:**- The main difference between UNION ALL statement and UNION is UNION All statement is much faster than UNION, the reason behind this is that because UNION ALL statement does not look for duplicate rows, but on the other hand UNION statement does look for duplicate rows, whether or not they exist.

**Question: Write some disadvantage of Cursor?**
**Answer:**- Cursor plays there row quite nicely but although there are some disadvantage of Cursor. Because we know cursor doing roundtrip it will make network line busy and also make time consuming methods. First of all select query generate output and after that cursor goes one by one so roundtrip happen. Another disadvange of cursor are too costly because they require lot of resources and temporary storage so network is quite busy.

**Question: What is Log Shipping and its purpose?**
**Answer:** In Log Shipping the transactional log file from one server is automatically updated in backup database on the other server and in the case when one server fails the other server will have the same DB and we can use this as the DDR(disaster recovery) plan.

**Question: What are the null values in SQL SERVER?**
**Answer:** Before understand the null values we have some overview about what the value is.
Value is the actual data stored in a particular field of particular record. But what is done when there is no value in the field. That value is something like <null>. Nulls present missing information. We can also call null propagation.

**Question: What is difference between OSQL and Query Analyzer?**

**Answer:** Both are same for functioning but there is a little difference OSQL is command line tool which execute query and display the result same a Query Analyzer do but Query Analyzer is graphical. OSQL have not ability like Query Analyzer to analyze queries and show statistics on speed of execution. And other useful thing about OSQL is that its helps in scheduling which is done in Query Analyzer with the help of JOB.

**Question: Write a Role of Sql Server 2005 in XML Web Services?**

**Answer:** SQL Server 2005 create a standard method for getting the database engine using SOAP via HTTP. By this method, we can send SOAP/HTTP requests to SQL Server for executing T-SQL batch statements, stored procedures, extended stored procedures, and scalar-valued user-defined functions may be with or without parameters.

**Question: What are the different types of Locks ?**

**Answer:** There are three main types of locks that SQL Server

1. Shared locks are used for operations that does not allow to change or update data, such as a SELECT statement.
2. Update locks are used when SQL Server intends to modify a page, and later promotes the update page lock to an exclusive page lock before actually making the changes.
3. Exclusive locks are used for the data modification operations, such as UPDATE, INSERT, or DELETE.

**Question: What is ‘Write-ahead log’ in Sql Server?**

**Answer:** Before understanding it we must have an idea about the transaction log files. These files are the files which hold the data for change in database. Now we explain when we are doing some Sql Server 2000 query or any Sql query like Sql insert query, delete sql query, update sql query and change the data in sql server database it cannot change the database directly to table. Sql server extracts the data that is modified by sql server 2000 query or by sql query and places it in memory. Once data is stores in memory user can make changes to that a log file is generated this log file is generated in every five minutes of transaction is done. After this sql server writes changes to database with the help of transaction log files. This is called Write-ahead log.

**Question: What do u mean by Extents and types of Extents ?**

**Answer:** An Extent is a collection of 8 sequential pages to hold database from becoming fragmented. Fragment means these pages relates to same table of database these also holds in indexing. To avoid for fragmentation Sql Server assign space to table in extents. So that the Sql
Server keep up to date data in extents. Because these pages are continuously one after another. There are usually two types of extents: Uniform and Mixed.

**Uniform** means when extent is own by a single object means all collection of 8 ages hold by a single extent is called uniform.

**Mixed** means when more than one object is comes in extents is known as mixed extents.

**Question:** What is different in Rules and Constraints?

**Answer:** Rules and Constraints are similar in functionality but there is little difference between them. Rules are used for backward compatibility. One the most exclusive difference is that we can bind rules to a data types whereas constraints are bound only to columns. So we can create our own data type with the help of Rules and get the input according to that.

**Question:** What Is Database?

**Answer:** A database is similar to a data file in that it is a storage place for data. Like a data file, a database does not present information directly to a user; the user runs an application that accesses data from the database and presents it to the user in an understandable format. Database systems are more powerful than data files in that data is more highly organized. In a well-designed database, there are no duplicate pieces of data that the user or application must update at the same time. Related pieces of data are grouped together in a single structure or record, and relationships can be defined between these structures and records. When working with data files, an application must be coded to work with the specific structure of each data file. In contrast, a database contains a catalog that applications use to determine how data is organized. Generic database applications can use the catalog to present users with data from different databases dynamically, without being tied to a specific data format. A database typically has two main parts: first, the files holding the physical database and second, the database management system (DBMS) software that applications use to access data. The DBMS is responsible for enforcing the database structure, including: maintaining relationships between data in the database. Ensuring that data is stored correctly and that the rules defining data relationships are not violated. Recovering all data to a point of known consistency in case of system failures.

**Question:** what is Relational Database?

**Answer:** Although there are different ways to organize data in a database, relational databases are one of the most effective. Relational database systems are an application of mathematical set theory to the problem of effectively organizing data. In a relational database, data is collected into tables (called relations in relational theory). A table represents some class of objects that are important to an organization. For example, a company may have a database with a table for employees, another table for customers, and another for stores. Each table is built of columns and rows (called attributes and tuples in relational theory). Each column represents some attribute of the object represented by the table. For example, an Employee table would typically have columns for attributes such as first name, last name, employee ID,
department, pay grade, and job title. Each row represents an instance of the object represented by the table. For example, one row in the Employee table represents the employee who has employee ID 12345. When organizing data into tables, you can usually find many different ways to define tables. Relational database theory defines a process called normalization, which ensures that the set of tables you define will organize your data effectively.

**Question: What are Data Integrity and its categories?**

**Answer:** Enforcing data integrity ensures the quality of the data in the database. For example, if an employee is entered with an employee_id value of 123, the database should not allow another employee to have an ID with the same value. If you have an employee_rating column intended to have values ranging from 1 to 5, the database should not accept a value of 6. If the table has a dept_id column that stores the department number for the employee, the database should allow only values that are valid for the department numbers in the company. Two important steps in planning tables are to identify valid values for a column and to decide how to enforce the integrity of the data in the column. Data integrity falls into these categories:

1) Entity integrity
2) Domain integrity
3) Referential integrity
4) User-defined integrity

**Entity Integrity:** Entity integrity defines a row as a unique entity for a particular table. Entity integrity enforces the integrity of the identifier column(s) or the primary key of a table (through indexes, UNIQUE constraints, PRIMARY KEY constraints, or IDENTITY properties).

**Domain Integrity:** Domain integrity is the validity of entries for a given column. You can enforce domain integrity by restricting the type (through data types), the format (through CHECK constraints and rules), or the range of possible values (through FOREIGN KEY constraints, CHECK constraints, DEFAULT definitions, NOT NULL definitions, and rules).

**Referential Integrity:** Referential integrity preserves the defined relationships between tables when records are entered or deleted. In Microsoft® SQL Server™ 2000, referential integrity is based on relationships between foreign keys and primary keys or between foreign keys and unique keys (through FOREIGN KEY and CHECK constraints). Referential integrity ensures that key values are consistent across tables. Such consistency requires that there be no references to nonexistent values and that if a key value changes, all references to it change consistently throughout the database. When you enforce referential integrity, SQL Server prevents users from:

- Adding records to a related table if there is no associated record in the primary table.
- Changing values in a primary table that result in orphaned records in a related table.
- Deleting records from a primary table if there are matching related records.

For example, with the sales and titles tables in the pubs database, referential integrity is based on the relationship between the foreign key (title_id) in the sales table and the primary key (title_id) in the titles table.
User-Defined: Integrity User-defined integrity allows you to define specific business rules that do not fall into one of the other integrity categories. All of the integrity categories support user-defined integrity (all column- and table-level constraints in CREATE TABLE, stored procedures, and triggers).

Question: SQL Server runs on which TCP/IP port and From where can you change the default port?
Answer: SQL Server runs on port 1433 but we can also change it for better security and From the network Utility TCP/IP properties –>Port number.Both on client and the server.

Question: What is the use of DBCC commands?
Answer: DBCC stands for database consistency checker. We use these commands to check the consistency of the databases, i.e., maintenance, validation task and status checks. DBCC CHECKDB – Ensures that tables in the db and the indexes are correctly linked and DBCC CHECKALLOC To check that all pages in a db are correctly allocated. DBCC SQLPERF – It gives report on current usage of transaction log in percentage. DBCC CHECKFILEGROUP – Checks all tables file group for any damage.

Question: What is the difference between a HAVING CLAUSE and a WHERE CLAUSE?
Answer: Having Clause is basically used only with the GROUP BY function in a query. WHERE Clause is applied to each row before they are part of the GROUP BY function in a query.

Question: When do you use SQL Profiler?
Answer: SQL Profiler utility allows us to basically track Connections to the SQL Server and also determine activities such as which SQL Scripts are running, failed jobs etc.

Question: Can you explain the role of each service?
Answer: SQL SERVER – is for running the databases SQL AGENT – is for automation such as Jobs, DB Maintenance, Backups DTC – Is for linking and connecting to other SQL Servers.

Question: What is Normalization?
Answer: The logical design of the database, including the tables and the relationships between them, is the core of an optimized relational database. A good logical database design can lay the foundation for optimal database and application performance. A poor logical database design can impair the performance of the entire system.

Normalizing a logical database design involves using formal methods to separate the data into multiple, related tables. A greater number of narrow tables (with fewer columns) is characteristic of a normalized database. A few wide tables (with more columns) is characteristic of an non-normalized database. Reasonable normalization often improves performance. When
useful indexes are available, the Microsoft® SQL Server™ 2000 query optimizer is efficient at selecting rapid, efficient joins between tables.

Some of the benefits of normalization include:
· Faster sorting and index creation.
· A larger number of clustered indexes. For more information, Narrower and more compact indexes.
· Fewer indexes per table, which improves the performance of INSERT, UPDATE, and DELETE statements.
· Fewer null values and less opportunity for inconsistency, which increase database compactness.

As normalization increases, so do the number and complexity of joins required to retrieve data. Too many complex relational joins between too many tables can hinder performance. Reasonable normalization often includes few regularly executed queries that use joins involving more than four tables.

Sometimes the logical database design is already fixed and total redesign is not feasible. Even then, however, it might be possible to normalize a large table selectively into several smaller tables. If the database is accessed through stored procedures, this schema change could take place without affecting applications. If not, it might be possible to create a view that hides the schema change from the applications.

**Question:** Can you explain what View is in SQL?
**Answer:** View is just a virtual table nothing else which is based or we can say develop with SQL SELECT query. So we can say that it's a real database table (it has columns and rows just like a regular table), but one difference is that real tables store data, but views can’t. View data is generated dynamically when the view is referenced. And view can also reference one or more existing database tables or other views. We can say that it is filter of database.

**Question:** How to get which Process is Blocked in SQL SERVER?
**Answer:** There are two ways to get this: `sp_who` and `sp_who2`. You cannot get any detail about the `sp_who2` but it provides more information than the `sp_who`. And another option from which we can find which process is blocked by other process is by using Enterprise Manager or Management Studio, these two commands work much faster and more efficiently than these GUI-based front-ends.

**Question:** If I want to see what fields a table is made of, and what the sizes of the fields are, what option do I have to look for?
`Sp_Columns 'TableName'`

**Question:** What is a query?
A request for information from a database. There are three general methods for posing queries:
# Choosing parameters from a menu: In this method, the database system presents a list of parameters from which you can choose. This is perhaps the easiest way to pose a query because the menus guide you, but it is also the least flexible.

# Query by example (QBE): In this method, the system presents a blank record and lets you specify the fields and values that define the query.

# Query language: Many database systems require you to make requests for information in the form of a stylized query that must be written in a special query language. This is the most complex method because it forces you to learn a specialized language, but it is also the most powerful.

**Question: What is the purpose of the model database?**
It works as Template Database for the Create Database Syntax

**Question: What is the purpose of the master database?**
Master database keeps the information about sql server configuration, databases users etc

**Question: What is the purpose of the tempdb database?**
Tempdb database keeps the information about the temporary objects (#TableName, #Procedure). Also the sorting, DBCC operations are performed in the TempDB

**Question: What is the purpose of the USE command?**
Use command is used for to select the database. For i.e Use Database Name

**Question: If you delete a table in the database, will the data in the table be deleted too?**
Yes

**Question: What is the Parse Query button used for? How does this help you?**
Parse query button is used to check the SQL Query Syntax

**Question: Tables are created in a ________________ in SQL Server 2005.**
resouce database(System Tables)

**Question: What is usually the first word in a SQL query?**
SELECT

**Question: Does a SQL Server 2005 SELECT statement require a FROM?**
NO

**Question: Can a SELECT statement in SQL Server 2005 be used to make an assignment? Explain with examples.**
Yes. Select @MyDate = GetDate()
Question: What is the ORDER BY used for?
Order By clause is used for sorting records in Ascending or Descending order.

Question: Does ORDER BY actually change the order of the data in the tables or does it just change the output?
Order By clause change only the output of the data.

Question: What is the default order of an ORDER BY clause?
Ascending Order

Question: What kind of comparison operators can be used in a WHERE clause?

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>= (Equals)</td>
<td>Equal to</td>
</tr>
<tr>
<td>&gt; (Greater Than)</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt; (Less Than)</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;= (Greater Than or Equal To)</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;= (Less Than or Equal To)</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>&lt;&gt; (Not Equal To)</td>
<td>Not equal to</td>
</tr>
<tr>
<td>!= (Not Equal To)</td>
<td>Not equal to (not SQL-92 standard)</td>
</tr>
<tr>
<td>!&lt; (Not Less Than)</td>
<td>Not less than (not SQL-92 standard)</td>
</tr>
<tr>
<td>!&gt; (Not Greater Than)</td>
<td>Not greater than (not SQL-92 standard)</td>
</tr>
</tbody>
</table>

Question: What are four major operators that can be used to combine conditions on a WHERE clause?
OR, AND, IN and BETWEEN

Question: What are the logical operators?

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>TRUE if all of a set of comparisons are TRUE.</td>
</tr>
<tr>
<td>AND</td>
<td>TRUE if both Boolean expressions are TRUE.</td>
</tr>
<tr>
<td>ANY</td>
<td>TRUE if any one of a set of comparisons are TRUE.</td>
</tr>
<tr>
<td>BETWEEN</td>
<td>TRUE if the operand is within a range.</td>
</tr>
<tr>
<td>EXISTS</td>
<td>TRUE if a subquery contains any rows.</td>
</tr>
<tr>
<td>IN</td>
<td>TRUE if the operand is equal to one of a list of expressions.</td>
</tr>
<tr>
<td>LIKE</td>
<td>TRUE if the operand matches a pattern.</td>
</tr>
<tr>
<td><strong>NOT</strong></td>
<td>Reverses the value of any other Boolean operator.</td>
</tr>
<tr>
<td><strong>OR</strong></td>
<td>TRUE if either Boolean expression is TRUE.</td>
</tr>
<tr>
<td><strong>SOME</strong></td>
<td>TRUE if some of a set of comparisons are TRUE.</td>
</tr>
</tbody>
</table>

**Question:** In a WHERE clause, do you need to enclose a text column in quotes? Do you need to enclose a numeric column in quotes?

Enclose Text in Quotes (Yes)
Enclose Number in Quotes (NO)

**Question:** Is a null value equal to anything? Can a space in a column be considered a null value? Why or why not?

No NULL value means nothing. We can’t consider space as NULL value.

**Question:** Will COUNT(column) include columns with null values in its count?

Yes, it will include the null column in count.

**Question:** What are column aliases? Why would you want to use column aliases? How can you embed blanks in column aliases?

You can create aliases for column names to make it easier to work with column names, calculations, and summary values. For example, you can create a column alias to:

* Create a column name, such as “Total Amount,” for an expression such as (quantity * unit_price) or for an aggregate function.
* Create a shortened form of a column name, such as “d_id” for “discounts.stor_id.”

After you have defined a column alias, you can use the alias in a Select query to specify query output.

**Question:** What are table aliases?

Aliases can make it easier to work with table names. Using aliases is helpful when:

* You want to make the statement in the SQL Pane shorter and easier to read.
* You refer to the table name often in your query — such as in qualifying column names — and want to be sure you stay within a specific character-length limit for your query. (Some databases impose a maximum length for queries.)
* You are working with multiple instances of the same table (such as in a self-join) and need a way to refer to one instance or the other.

**Question:** What are table qualifiers? When should table qualifiers be used?

[@table_qualifier =] qualifier

Is the name of the table or view qualifier. qualifier is sysname, with a default of NULL. Various DBMS products support three-part naming for tables (qualifier.owner.name). In SQL Server, this
column represents the database name. In some products, it represents the server name of the table’s database environment.

**Question:** Are semicolons required at the end of SQL statements in SQL Server 2005?
No it is not required

**Question:** Do comments need to go in a special place in SQL Server 2005?
No its not necessary

**Question:** When would you use the ROWCOUNT function versus using the WHERE clause?
Returns the number of rows affected by the last statement. If the number of rows is more than 2 billion, use ROWCOUNT_BIG.
Transact-SQL statements can set the value in @@ROWCOUNT in the following ways:
* Set @@ROWCOUNT to the number of rows affected or read. Rows may or may not be sent to the client.
* Preserve @@ROWCOUNT from the previous statement execution.
* Reset @@ROWCOUNT to 0 but do not return the value to the client.
Statements that make a simple assignment always set the @@ROWCOUNT value to 1.

**Question:** Is SQL case-sensitive? Is SQL Server 2005 case-sensitive?
No both are not case-sensitive. Case sensitivity depends on the collation you choose.
If you installed SQL Server with the default collation options, you might find that the following queries return the same results:

```sql
CREATE TABLE mytable
(
    mycolumn VARCHAR(10)
)
GO
SET NOCOUNT ON
INSERT mytable VALUES('Case')
GO
SELECT mycolumn FROM mytable WHERE mycolumn='Case'
SELECT mycolumn FROM mytable WHERE mycolumn='caSE'
SELECT mycolumn FROM mytable WHERE mycolumn='case'
You can alter your query by forcing collation at the column level:
SELECT myColumn FROM myTable
WHERE myColumn COLLATE Latin1_General_CS_AS = ‘caSE’
SELECT myColumn FROM myTable
WHERE myColumn COLLATE Latin1_General_CS_AS = ‘case’
SELECT myColumn FROM myTable
WHERE myColumn COLLATE Latin1_General_CS_AS = ‘Case’
```
– if myColumn has an index, you will likely benefit by adding
– AND myColumn = ‘case’

**Question:** What is a synonym? Why would you want to create a synonym?
SYNONYM is a single-part name that can replace a two, three or four-part name in many SQL statements. Using SYNONYMS in RDBMS cuts down on typing.
SYNONYMS can be created for the following objects:

* Table
* View
* Assembly (CLR) Stored Procedure
* Assembly (CLR) Table-valued Function
* Assembly (CLR) Scalar Function
* Assembly Aggregate (CLR) Aggregate Functions
* Replication-filter-procedure
* Extended Stored Procedure
* SQL Scalar Function
* SQL Table-valued Function
* SQL Inline-table-valued Function
* SQL Stored Procedure

**Syntax**
CREATE SYNONYM [ schema_name_1. ] synonym_name FOR < object >
< object > ::= =
{ [ server_name.[ database_name ] . [ schema_name_2 ].| database_name . [ schema_name_2 ] | schema_name_2. ] object_name }

**Question:** Can a synonym name of a table be used instead of a table name in a SELECT statement?
Yes

**Question:** Can a synonym of a table be used when you are trying to alter the definition of a table?
Not Sure will try

**Question:** Can you type more than one query in the query editor screen at the same time?
Yes we can.

**Question:** While you are inserting values into a table with the INSERT INTO .. VALUES option, does the order of the columns in the INSERT statement have to be the same as the order of
the columns in the table?
Not Necessary

Question: While you are inserting values into a table with the INSERT INTO .. SELECT option, does the order of the columns in the INSERT statement have to be the same as the order of the columns in the table?
Yes if you are not specifying the column names in the insert clause, you need to maintain the column order in SELECT statement

Question: When would you use an INSERT INTO .. SELECT option versus an INSERT INTO .. VALUES option? Give an example of each.
INSERT INTO .. SELECT is used insert data in to table from different tables or condition based insert. INSERT INTO .. VALUES you have to specify the insert values

Question: What does the UPDATE command do?
Update command will modify the existing record

Question: Can you change the data type of a column in a table after the table has been created? If so, which command would you use?
Yes we can. Alter Table Modify Column

Question: Will SQL Server 2005 allow you to reduce the size of a column?
Yes it allows

Question: What integer data types are available in SQL Server 2005?
Exact-number data types that use integer data.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Range</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigint</td>
<td>(-2^{63} (-9,223,372,036,854,775,808) to 2^{63}-1 (9,223,372,036,854,775,807))</td>
<td>8 Bytes</td>
</tr>
<tr>
<td>int</td>
<td>(-2^{31} (-2,147,483,648) to 2^{31}-1 (2,147,483,647))</td>
<td>4 Bytes</td>
</tr>
<tr>
<td>smallint</td>
<td>(-2^{15} (-32,768) to 2^{15}-1 (32,767))</td>
<td>2 Bytes</td>
</tr>
<tr>
<td>tinyint</td>
<td>0 to 255</td>
<td>1 Byte</td>
</tr>
</tbody>
</table>

Question: What is the default value of an integer data type in SQL Server 2005?
NULL

Question: What is the difference between a CHAR and a VARCHAR datatpe?
CHAR and VARCHAR data types are both non-Unicode character data types with a maximum length of 8,000 characters. The main difference between these 2 data types is that a CHAR data type is fixed-length while a VARCHAR is variable-length. If the number of characters entered in
a **CHAR** data type column is less than the declared column length, spaces are appended to it to fill up the whole length.

Another difference is in the storage size wherein the storage size for **CHAR** is \( n \) bytes while for **VARCHAR** is the actual length in bytes of the data entered (and not \( n \) bytes).

You should use **CHAR** data type when the data values in a column are expected to be consistently close to the same size. On the other hand, you should use **VARCHAR** when the data values in a column are expected to vary considerably in size.

**Question:** Does Server SQL treat **CHAR** as a variable-length or fixed-length column?

SQL Server treats **CHAR** as fixed length column

**Question:** If you are going to have too many nulls in a column, what would be the best data type to use?

Variable length columns only use a very small amount of space to store a NULL so **VARCHAR** datatype is the good option for null values

**Question:** When columns are added to existing tables, what do they initially contain?

The column initially contains the NULL values

**Question:** What command would you use to add a column to a table in SQL Server?

```
ALTER TABLE tablename ADD column_name DATATYPE
```

**Question:** Does an index slow down updates on indexed columns?

Yes

**Question:** What is a constraint?

Constraints in Microsoft SQL Server 2000/2005 allow us to define the ways in which we can automatically enforce the integrity of a database. Constraints define rules regarding permissible values allowed in columns and are the standard mechanism for enforcing integrity. Using constraints is preferred to using triggers, stored procedures, rules, and defaults, as a method of implementing data integrity rules. The query optimizer also uses constraint definitions to build high-performance query execution plans.

**Question:** How many indexes does SQL Server 2005 allow you to have on a table?

250 indices per table

**Question:** What command would you use to create an index?

```
CREATE INDEX INDEXNAME ON TABLE(COLUMN NAME)
```

**Question:** What is the default ordering that will be created by an index (ascending or descending)?
Clustered indexes can be created in SQL Server databases. In such cases the logical order of the index key values will be the same as the physical order of rows in the table. By default it is ascending order, we can also specify the index order while index creation.

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ] INDEX index_name
ON { table | view } ( column [ ASC | DESC ] [ ,...n ] )
```

**Question:** How do you delete an index?

DROP INDEX authors.au_id_ind

**Question:** What does the NOT NULL constraint do?

Constrain will not allow NULL values in the column

**Question:** What command must you use to include the NOT NULL constraint after a table has already been created?

DEFAULT, WITH CHECK or WITH NOCHECK

**Question:** When a PRIMARY KEY constraint is included in a table, what other constraints does this imply?

Unique + NOT NULL

**Question:** What is a concatenated primary key?

Each table has one and only one primary key, which can consist of one or many columns. A concatenated primary key comprises two or more columns. In a single table, you might find several columns, or groups of columns, that might serve as a primary key and are called candidate keys. A table can have more than one candidate key, but only one candidate key can become the primary key for that table

**Question:** How are the UNIQUE and PRIMARY KEY constraints different?

A UNIQUE constraint is similar to PRIMARY key, but you can have more than one UNIQUE constraint per table.

When you declare a UNIQUE constraint, SQL Server creates a UNIQUE index to speed up the process of searching for duplicates. In this case the index defaults to NONCLUSTERED index, because you can have only one CLUSTERED index per table.

* The number of UNIQUE constraints per table is limited by the number of indexes on the table i.e 249 NONCLUSTERED index and one possible CLUSTERED index.

Contrary to PRIMARY key UNIQUE constraints can accept NULL but just once. If the constraint is defined in a combination of fields, then every field can accept NULL and can have some values on them, as long as the combination values is unique.
Question: What is a referential integrity constraint? What two keys does the referential integrity constraint usually include?

Referential integrity in a relational database is consistency between coupled tables. Referential integrity is usually enforced by the combination of a primary key or candidate key (alternate key) and a foreign key. For referential integrity to hold, any field in a table that is declared a foreign key can contain only values from a parent table’s primary key or a candidate key. For instance, deleting a record that contains a value referred to by a foreign key in another table would break referential integrity. The relational database management system (RDBMS) enforces referential integrity, normally either by deleting the foreign key rows as well to maintain integrity, or by returning an error and not performing the delete. Which method is used would be determined by the referential integrity constraint, as defined in the data dictionary.

Question: What is a foreign key?

FOREIGN KEY constraints identify the relationships between tables. A foreign key in one table points to a candidate key in another table. Foreign keys prevent actions that would leave rows with foreign key values when there are no candidate keys with that value. In the following sample, the order_part table establishes a foreign key referencing the part_sample table defined earlier. Usually, order_part would also have a foreign key against an order table, but this is a simple example.

CREATE TABLE order_part
(order_nmbr int,
 part_nmbr int
FOREIGN KEY REFERENCES part_sample(part_nmbr)
ON DELETE NO ACTION,
qty_ordered int)
GO
You cannot insert a row with a foreign key value (except NULL) if there is no candidate key with that value. The ON DELETE clause controls what actions are taken if you attempt to delete a row to which existing foreign keys point. The ON DELETE clause has two options: NO ACTION specifies that the deletion fails with an error. CASCADE specifies that all the rows with foreign keys pointing to the deleted row are also deleted.

The ON UPDATE clause defines the actions that are taken if you attempt to update a candidate key value to which existing foreign keys point. It also supports the NO ACTION and CASCADE options.

Question: What does the ON DELETE CASCADE option do?

ON DELETE CASCADE
Specifies that if an attempt is made to delete a row with a key referenced by foreign keys in existing rows in other tables, all rows containing those foreign keys are also deleted. If
cascading referential actions have also been defined on the target tables, the specified cascading actions are also taken for the rows deleted from those tables.

ON UPDATE CASCADE
Specifies that if an attempt is made to update a key value in a row, where the key value is referenced by foreign keys in existing rows in other tables, all of the foreign key values are also updated to the new value specified for the key. If cascading referential actions have also been defined on the target tables, the specified cascading actions are also taken for the key values updated in those tables.

**Question:** What does the ON UPDATE NO ACTION do?

ON DELETE NO ACTION
Specifies that if an attempt is made to delete a row with a key referenced by foreign keys in existing rows in other tables, an error is raised and the DELETE is rolled back.

ON UPDATE NO ACTION
Specifies that if an attempt is made to update a key value in a row whose key is referenced by foreign keys in existing rows in other tables, an error is raised and the UPDATE is rolled back.

**Question:** Can you use the ON DELETE and ON UPDATE in the same constraint?

Yes we can.

```
CREATE TABLE part_sample
(part_nmbr int PRIMARY KEY,
 part_name char(30),
 part_weight decimal(6,2),
 part_color char(15) )
CREATE TABLE order_part
(order_nmbr int,
 part_nmbr int
FOREIGN KEY REFERENCES part_sample(part_nmbr)
ON DELETE NO ACTION ON UPDATE NO ACTION,
 qty_ordered int)
GO
```

**Question:** How do you test proper TCP/IP configuration Windows machine?

Windows NT: IPCONFIG/ALL, Windows 95: WINIPCFG, Ping or ping ip.add.re.ss

**Question:** What is RAID and what are different types of RAID configurations?

RAID stands for Redundant Array of Inexpensive Disks, used to provide fault tolerance to database servers. There are six RAID levels 0 through 5 offering different levels of performance, fault tolerance.

**Question:** How do you define testing of network layers?
Reviewing with your developers to identify the layers of the Network layered architecture, your Web client and Web server application interact with. Determine the hardware and software configuration dependencies for the application under test.

**Question: What are the steps you will take to improve performance of a poor performing query?**

This is a very open ended question and there could be a lot of reasons behind the poor performance of a query. But some general issues that you could talk about would be: No indexes, table scans, missing or out of date statistics, blocking, excess recompilations of stored procedures, procedures and triggers without SET NOCOUNT ON, poorly written query with unnecessarily complicated joins, too much normalization, excess usage of cursors and temporary tables.

Some of the tools/ways that help you troubleshooting performance problems are: SET SHOWPLAN_ALL ON, SET SHOWPLAN_TEXT ON, SET STATISTICS IO ON, SQL Server Profiler, Windows NT /2000 Performance monitor, Graphical execution plan in Query Analyzer.

**Question: How many layers of TCP/IP protocol combined of?**

Five. (Application, Transport, Internet, Data link, Physical).

**Question: How many bits IP Address consist of?**

An IP Address is a 32-bit number.

**Question: What is a deadlock and what is a live lock? How will you go about resolving deadlocks?**

Deadlock is a situation when two processes, each having a lock on one piece of data, attempt to acquire a lock on the other’s piece. Each process would wait indefinitely for the other to release the lock, unless one of the user processes is terminated. SQL Server detects deadlocks and terminates one user’s process.

A livelock is one, where a request for an exclusive lock is repeatedly denied because a series of overlapping shared locks keeps interfering. SQL Server detects the situation after four denials and refuses further shared locks. A livelock also occurs when read transactions monopolize a table or page, forcing a write transaction to wait indefinitely.

**Question: What is blocking and how would you troubleshoot it?**

Blocking happens when one connection from an application holds a lock and a second connection requires a conflicting lock type. This forces the second connection to wait, blocked on the first.

**Question: How to restart SQL Server in single user mode? How to start SQL Server in minimal configuration mode?**

SQL Server can be started from command line, using the SQLSERVR.EXE. This EXE has some very important parameters with which a DBA should be familiar with. -m is used for starting SQL
Server in single user mode and -f is used to start the SQL Server in minimal configuration mode.

**Question:** As a part of your job, what are the DBCC commands that you commonly use for database maintenance?
DBCC CHECKDB, DBCC CHECKTABLE, DBCC CHECKCATALOG, DBCC CHECKALLOC, DBCC SHOWCONTIG, DBCC SHRINKDATABASE, DBCC SHRINKFILE etc. But there are a whole load of DBCC commands which are very useful for DBAs.

**Question:** What is the difference between them (Ethernet networks and token ring networks)?
With Ethernet, any devices on the network can send data in a packet to any location on the network at any time. With Token Ring, data is transmitted in ‘tokens’ from computer to computer in a ring or star configuration. Token ring speed is 4/16 Mbit/sec, Ethernet – 10/100 Mbit/sec.

**Question:** What are triggers? How many triggers you can have on a table? How to invoke a trigger on demand?
Triggers are special kind of stored procedures that get executed automatically when an INSERT, UPDATE or DELETE operation takes place on a table.
In SQL Server 6.5 you could define only 3 triggers per table, one for INSERT, one for UPDATE and one for DELETE. From SQL Server 7.0 onwards, this restriction is gone, and you could create multiple triggers per each action. But in 7.0 there’s no way to control the order in which the triggers fire. In SQL Server 2000 you could specify which trigger fires first or fires last using sp_settriggerorder
Triggers can't be invoked on demand. They get triggered only when an associated action (INSERT, UPDATE, DELETE) happens on the table on which they are defined.
Triggers are generally used to implement business rules, auditing. Triggers can also be used to extend the referential integrity checks, but wherever possible, use constraints for this purpose, instead of triggers, as constraints are much faster.
Till SQL Server 7.0, triggers fire only after the data modification operation happens. So in a way, they are called post triggers. But in SQL Server 2000 you could create pre triggers also.

**Question:** What is the system function to get the current user’s user id?
USER_ID(). Also check out other system functions like USER_NAME(), SYSTEM_USER, SESSION_USER, CURRENT_USER, USER, SUSER_SID(), HOST_NAME().

**Question:** What is a traditional Network Library for SQL Servers?
Named Pipes.

**Question:** What is Execution Context?
Whenever User log on or connects to SQL Server, it will create the User Session. Whenever that user executes the statement SQL Server uses that session’s user id, permission, password to execute the query. That session is called execution context.

Execution context is represented by a pair of security tokens: a login token and a user token. The tokens identify the primary and secondary principals against which permissions are checked and the source used to authenticate the token. A login connecting to an instance of SQL Server has one login token and one or more user tokens, depending on the number of databases to which the account has access.

**Question: What is Principals in SQL Server 2005?**
Principals are the individuals, groups, and processes that can request SQL Server resources. Principals are categorized by their scope. Every principal has security identifier

- -> Windows level
- -> SQL Server level
- -> Database level

**Question: Windows-level principals**

- Windows Domain Login
- Windows Local Login

**Question: SQL Server-level principal**

- SQL Server Login

**Question: Database-level principals**

- Database User
- Database Role
- Application Role

**Question: What are SQL Server Securables?**
SQL Server Securables contains three scopes, which are used to assign permissions to users. The securables are nested and each securable contains various other securables. The securable scopes are as follows:

- **Server**: It includes server roles, logins, etc.
- **Database**: It includes database users, application roles, database roles, etc.
- **Schema**: It includes various database objects such as tables, views, procedures, etc.
The securables are used to assign permissions to the users based on scope and the tasks assigned. The issues related to the connectivity to databases, accessing database objects, etc., can be resolved by granting or denying the permissions to the users.

**Question: What is the use of the Public database role in SQL Server?**
Every database user belongs to the **public** database role. When a user has not been granted or denied specific permissions on a securable, the user inherits the permissions granted to **public** on that securable.

**Question: Explain Certificate based SQL Server Logins/Principals?**
Server principals with names enclosed by double hash marks (##) are for internal system use only. The following principals are created from certificates when SQL Server is installed, and should not be deleted.

- ##MS_SQLResourceSigningCertificate##
- ##MS_SQLReplicationSigningCertificate##
- ##MS_SQLAuthenticatorCertificate##
- ##MS_AgentSigningCertificate##
- ##MS_PolicyEventProcessingLogin##
- ##MS_PolicySigningCertificate##
- ##MS_PolicyTsqlExecutionLogin##

**Question: What will you do if you lost rights of your SQL Server instance?**
We can use the below options
1. Dedicated Administrator Connection
2. BUILTIN\Administrator Group (Incase its rights are not revoked)
3. Final Option is to change the registry value
4. You can change authentication mode via registry

**Question: What is SQL Injection?**
SQL Injection is developed where unhandled/unexpected SQL commands are passed to SQL Server in a malicious manner. It is a problem because unknowingly data can be stolen, deleted, updated, inserted or corrupted.

**Question: What is the Guest user account in SQL Server? What login is it mapped to it?**
The Guest user account is created by default in all databases and is used when explicit permissions are not granted to access an object. It is not mapped directly to any login, but can be used by any login. Depending on your security needs, it may make sense to drop the Guest user account, in all databases except Master and TempDB

**Question: What is the use of BUILTIN\Administrators Group in SQL Server?**
Any Windows login in BUILTIN\Administrators group is by default a SQL Server system administrator. This single group can be used to manage administrators from a Windows and SQL Server perspective.

**Question:** What are steps to load a .NET code in SQL SERVER 2005?

Write the managed code and compile it to a DLL/Assembly.

After the DLL is compiled using the “CREATE ASSEMBLY” command you can load the assembly into SQL SERVER. Below is the create command which is loading “mycode.dll” into SQL SERVER using the “CREATE ASSEMBLY” command

**Syntax**

```sql
CREATE ASSEMBLY AssemblyName FROM 'C:/MyAssembly.dll'
```

**Question:** How can we drop an assembly from SQL SERVER?

```sql
DROP ASSEMBLY AssemblyName
```

**Question:** Are changes made to assembly updated automatically in database?

No, it will not synchronize the code automatically. For that you have to drop the assembly (Using DROP ASSEMBLY) and create (Using the CREATE ASSEMBLY again)

**Question:** Why do we need to drop assembly for updating changes?

When we load the assembly into SQL SERVER, it stores into the sys.assemblies system table. So any changes after that to the external DLL/ASSEMBLY will not reflect in SQL SERVER. So we have to DROP and CREATE assembly again in SQL SERVER.

**Question:** How to see assemblies loaded in SQL SERVER?

```sql
SELECT * FROM sys.assemblies_files
```

**Question:** If I want to see which files are linked with which assemblies?

Use sys.Assemblies_files system tables have the track about which files are associated with what assemblies.

```sql
SELECT * FROM sys.assemblies_files
```

**Question:** Does .NET CLR and SQL SERVER run in different process?

.NET CLR engine (hence all the .NET applications) and SQL SERVER run in the same process or address space. This “Same address space architecture” is implemented so that there no speed issues. If the architecture was implemented the other way (i.e. SQL SERVER and .NET CLR engine running in different memory process area) there would have been reasonable speed issue.

**Question:** Does .NET controls SQL SERVER or is it vice-versa?

SQL SERVER controls the way .NET application will run. Normally .NET framework controls the
way application should run. But in order that we have high stability and good security SQL SERVER will control the way .NET framework works with SQL SERVER environment. So lot of things will be controlled through SQL SERVER example: threads, memory allocations, security etc.

SQL SERVER can control .NET framework by “Host Control” mechanism provided by .NET Framework 2.0. Using the “Host Control” framework external application’s can control the way memory management is done, thread allocation’s are done and lot more. SQL SERVER uses “host Control” mechanism exposed by .NET 2.0 and controls the framework.

**Question: Is SQLCLR configured by default?**
SQLCLR is not configured by default. If Developers want to use the CLR integration feature of SQL SERVER it has to be enabled by DBA (From the Surface Area Configuration).

**Question: How to configure CLR for SQL SERVER?**
It’s a advanced option you will need to run the following query Analyzer

```
EXEC sp_configure 'show advanced option', '1'; go reconfigure go
EXEC sp_configure 'clr enabled', '1' go reconfigure; go
EXEC sp_configure 'show advanced options', '1'; reconfigure
EXEC sp_configure 'clr enabled' '1' reconfigure
```

**Question: Is .NET feature loaded by default in SQL SERVER?**
NO it will not be loaded, CLR is lazy loaded that means its only loaded when needed. It goes one step ahead. where the database Administrator has to turn the feature and using the “SP_Configure”.

NOTE: Loading .NET programming consumes some memory resources around 20 to 30 MB(it may vary depending on lot of situations). So if you really need .NET Integration then only go for this option.

**Question: How does SQL Server control .NET at run-time?**
.NET CLR exposes interfaces by which an external host can control the way .NET run time runs. In previous versions of .NET it was done via COM interface “ICorRuntimeHost”.
In previous version you can only do the following with COM interface.

* Specify that whether its server or work station DLL
* Specify version of the CLR (e.g.version 1.1 or 2.0)
* Specify garbage collection behavior
* Specify whether or not jitted code may be shared across AppDomains.
In .NET 2.0 it’s done by “ICLRRuntimeHOST” But in .NET 2.0 you can do much above that was provided by the previous COM interface

* Exceptional conditions
* Code loading
* Class loading
* Security particulars
* Resource allocation

SQL Server uses the “ICLRRuntimeHOST” to control .NET run time as the flexibility provided by the interface is far beyond what is given by the previous .NET version, and that ‘s what exactly SQL Server needs, a full control of the .NET run time.

**Question: What’s a “SAND BOX” in SQL SERVER 2005?**

“Sand box is a safe place for running semi-trusted programs or scripts, often originationg from the a third party”

Now for SQL SERVER it’s .NET the external third party which is running and SQL SERVER has to be sure that .NET runtime crashes does not affect his working. So in order that SQL Server runs properly there are three sandboxes that user code can run:

Safe Access sandbox: This will be the favorite setting of DBA’s if they are compelled to run CLR-Safe access Safe means you have only access to in-proc data access functionalities. So you can create stored procedures, triggers, functions, data types, triggers etc. But you can not access memory, disk, create files etc. In short you ca not hang the SQL Server.

External access sandbox: In External access you can use some real cool features of .NET like accessing file systems outside box, you can leverage your classes etc. But here you are not allowed to play around with threading, memory allocation etc.

Unsafe access sandbox:In Unsafe access you have access to memory management, threading,etc. So here developers can write unreliable and unsafe code which destabilizes SQL Server. In the first two access levels of sand box its difficult to write unreliable and unsafe code.

**Question: Explain transaction log backup**

In Full or Bulk Logged recovery models, it is very important that we have scheduled periodic Transaction Log backups so it will help us to maintain the the size of the transaction log within reasonable limits and will allow for the recovery of data with the least amount of data loss in case of any failure.

**Transaction Log backups come in three forms:**
**Pure Log Backup:** — A Pure Log backup contains only transactions and is completed when the database is in Full recovery model or Bulk Logged recovery model, but no bulk operations have been executed. In case of Bulk Logged recovery Bulk Operations are minimally logged.

**Bulk Log Backup:** — Bulk Log backups contain both transactional data and any physical extents modified by bulk operations while the database was in Bulk Logged recovery.

**Tail Log Backup:** — Tail Log backups are completed when the database is in Full or Bulk Logged recovery prior to a database restoration to capture all transaction log records that have not yet been backed up. It is possible in some instances to execute a Tail Log backup even if the database is damaged.

**Pure or Bulk Log Backup Example**
BACKUP LOG SQLDBPool
TO DISK = 'D:\SQLBackups\SQLDBPool.TRN'

**Tail Log Backup Example**
BACKUP LOG SQLDBPool
TO DISK = 'D:\SQLBackups\SQLDBPoolTailLog.TRN'
WITH NO_TRUNCATE

**Question: What is the use of SQLBrowser Service and how it work?**
The SQLBrowser Service is used by SQL Server for named instance name resolution and server name enumeration over TCP/IP and VIA networks.

The default instance of SQL Server is assigned the TCP Port 1433 by default to support client incoming requests. However, because more than one application/SQL Server Instances cannot share a port assignment, any named instances are given a random port number when the service is started. This random port assignment makes it difficult for clients to connect to it, because the client applications don’t know what port the server is listening on. To meet this need, the SQLBrowser Service was created.

On start-up, the SQLBrowser Service queries the registry to discover all the names and port numbers of installed servers and reserves **UDP Port 1434**. It then listens on UDP Port 1434 for **SQL Server Resolution Protocol (SSRP)** requests and responds to the requests with the list of instances and their respective port assignments so that clients can connect without knowing the port number assignment.

It is very important that **no unauthenticated traffic on UDP Port 1434 be allowed on the network**, because the service will respond to any request on that port.
If the SQLBrowser Service is disabled, it will be necessary to specify a static port number for all named instances of the SQL Server Service and to configure all client applications that connect to those instances with the appropriate connection information.

There will be only one SQL Browser Service for all the instances on same machine.

**Question: What is Orphaned User?**
An Orphaned User in SQL Server is a database user for which a valid SQL Server Login is not available or it is wrongly defined with the different SID in the SQL Server instance, thereby not allowing the user to get connect to the database to perform activities.

Below scenarios are mostly responsible for Orphan Users
1. A SQL Server Login was accidentally dropped
2. A database is restored with a copy of database from another SQL Server Instance
3. SID of the login is different in sys.server_principals and sys.sysusers

**Question: What are the steps SQL Server performs internally at the time of FULL backup?**
SQL Server follow the below steps once you execute the BACKUP command
1. Backup Process will lock the database and block all the transaction
2. Place a mark in the transaction log
3. Release the database lock
4. Extract all the pages in the data files and write them to the backup device
5. Lock the database and block all the transactions
6. Place a mark in the transaction log
7. Release the database lock
8. Extract the portion of the log between the marks and append it to backup

**Question: Which operations do not allowed during the full backup?**
1. Adding and removing database files
2. Shrinking the database

**Question: Explain SQL Server Protocols**

**TCP/IP and Named Pipes**
By default, clients have TCP and Named Pipes as available protocols on most of client computer. You can manipulate the protocol ordering by using the SQL Server Client utility. The client application uses the protocols in the order specified on the client computer. If you are using SQL Server 2005, the protocol order is stored in the ProtocolOrder registry entry under the following registry subkey:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SNI9.0
**IPX/SPX**
SQL Server 2005 does not support IPX/SPX. The newer versions of NetWare, such as NetWare 6.5, support TCP/IP which makes it the common protocol for all clients.

**VIA**
The Virtual Interface Adapter (VIA) can be used only by VIA hardware.

**Shared Memory**
Shared Memory can only be used on the local computer and cannot be used as a network protocol.

**Question:** What is the difference between database mirroring and log shipping?

<table>
<thead>
<tr>
<th>Database Mirroring</th>
<th>Log-shipping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database mirroring is functionality in the SQL Server engine that reads from the transaction log and copies transactions from the principal server instance to the mirror server instance. Database mirroring can operate synchronously or asynchronously.</td>
<td>Log shipping is based on SQL Server Agent jobs that periodically take log backups of the primary database, copy the backup files to one or more secondary server instances, and restore the backups into the secondary database(s). Log shipping supports an unlimited number of secondary’s for each primary database.</td>
</tr>
<tr>
<td>Database mirroring can operate synchronously or asynchronously. If configured to operate synchronously, the transaction on the principal will not be committed until it is hardened to disk on the mirror.</td>
<td>Log shipping is always asynchrony. Log shipping totally depends on the log backup and restore schedule</td>
</tr>
<tr>
<td>Database mirroring supports only one mirror for each principal database. That means DB mirroring is at database level</td>
<td>Log-shipping can work on database and server level. You can configure multiple databases in logshipping</td>
</tr>
</tbody>
</table>

**Data Transfer:**  Individual T-Log records are transferred using TCP endpoints  
**Transactional Consistency:** Only committed transactions are transferred  
**Server Limitation:** Can be applied to only one mirror server  
**Failover:** Automatic  
**Failover Duration:** Failover is fast, sometimes < 3 seconds but not more than 10 seconds  
**Role Change:** Role change is fully automatic

**With Log Shipping:**  
**Data Transfer:** T-Logs are backed up and transferred to secondary server  
**Transactional Consistency:** All committed and un-committed are transferred  
**Server Limitation:** Can be applied to multiple
<table>
<thead>
<tr>
<th><strong>Client Re-direction:</strong> Fully automatic as it uses .NET 2.0/.Net 3.0</th>
<th><strong>stand-by servers</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Failover:</strong> Manual</td>
<td><strong>Failover Duration:</strong> Can take more than 30 mins</td>
</tr>
<tr>
<td><strong>Role Change:</strong> Role change is manual</td>
<td><strong>Client Re-direction:</strong> Manual changes required</td>
</tr>
<tr>
<td>Support only full recovery model</td>
<td>Supports full and bulk-logged recovery model</td>
</tr>
<tr>
<td>Mirror database is always in recovery mode. To read it you have use database snapshot.</td>
<td>You can use the stand-by option to read the database on standby server</td>
</tr>
<tr>
<td>Auto Page Recovery introduced with SQL SERVER 2008 so it will recover the damaged pages.</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

**Question:** Explain difference between Index Rebuild and Index Reorganize?

<table>
<thead>
<tr>
<th><strong>Index Rebuild</strong></th>
<th><strong>Index Reorganize</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>It is offline operation</td>
<td>It is online operation</td>
</tr>
<tr>
<td>Option is available in all SQL Server 2005 edition</td>
<td>Option is available in SQL Server 2005 Enterprise and Developer edition only</td>
</tr>
<tr>
<td>Index rebuilds works by re-creating the index internally again and when that has been achieved, it drops the existing index.</td>
<td>Index reorganize is the process of physically re-organizing the leaf nodes of the index</td>
</tr>
<tr>
<td>Index rebuild need more log space so it is advisable to change the database recovery model Simple or Bulk-logged</td>
<td>Reorganize swaps one page with another and thus does not require free space for this operation like rebuild does. Infect, reorganize can free up some pages as it does the reorganize in two phases – compaction and defrag.</td>
</tr>
<tr>
<td>During the index rebuild process, It will also re-compute index statistics</td>
<td>A reorganize can remove almost all of the logical fragmentation but it cannot necessarily fix extent fragmentation in which the previous and the next extents are physically contiguous.</td>
</tr>
<tr>
<td>Can be done on any data type column because it is offline operation.</td>
<td>Another point to note is that an index (clustered or non-clustered) cannot be built online if it contains LOB data (text, ntext,</td>
</tr>
</tbody>
</table>
If you want to rebuild an index with multiple partitions in an online environment, you need to rebuild the entire index which means rebuilding all the partitions.

Regarding partitions of an index, if an index has multiple partitions, then you cannot rebuild a single partition online. You can reorganize a single index partition online.

USE AdventureWorks;
GO
ALTER INDEX ALL ON Production.Product REBUILD
GO

Index should be rebuild when index fragmentation is great than 40%.

Index should be reorganized when index fragmentation is between 10% to 40%

ONLINE option will not keep index available during the rebuilding.

ONLINE option will keep index available during the rebuilding.

**Question: How to improve TempDB performance?**

1) TempDB file size and physical placement on harddrive can affect the performance of a system
2) Set the tempdb recovery model to simple
3) Do the capacity planning of the TempDB database and pre allocate the appropriate size to MDF files, which will improve performance for allocating size
4) Put TempDB on fast I/O System
5) Create as many tempdb data files by considering the number of cpu
6) make all the tempdb data file size same

**Question: How to troubleshoot suspect database problem? How to bring it back online? What are the do you need to perform once the database is online?**

**Solution**

Step 1: Bring the database online using below script
USE Master
GO

– Determine the original database status
SELECT [Name], DBID, Status
FROM master.dbo.sysdatabases
GO

– Enable system changes
sp_configure ‘allow updates’,1
GO
RECONFIGURE WITH OVERRIDE
GO

– Update the database status
UPDATE master.dbo.sysdatabases
SET Status = 24
WHERE [Name] = ‘SuspectedDatabaseName’
GO

– Disable system changes
sp_configure ‘allow updates’,0
GO
RECONFIGURE WITH OVERRIDE
GO

– Determine the final database status
SELECT [Name], DBID, Status
FROM master.dbo.sysdatabases
GO

Step 2: Check for database corruption. This is very important step please execute it.
  • DBCC CHECKDB – Validate the overall database integrity
  • DBCC CHECKCATALOG – Validate the system catalog integrity
  • DBCC CHECKTABLE – Validate the integrity for a single table

Step 3: To resolve the corruption issue, please execute below commands
  • Drop and Recreate Index(es)
  • Move the recoverable data from an existing table to a new table
  • Update statistics
  • DBCC UPDATEUSAGE
  • sp_recompile

Step 4: Repeat Step 2 to validate all the corruption occurred

What is Schema in SQL Server 2005? Explain its properties with example?
A schema is nothing more than a named, logical container in which you can create database objects. A new schema is created using the CREATE SCHEMA DDL statement.

Properties
  • Ownership of schemas and schema-scoped securables is transferable.
  • Objects can be moved between schemas
  • A single schema can contain objects owned by multiple database users.
  • Multiple database users can share a single default schema.
  • Permissions on schemas and schema-contained securables can be managed with greater precision than in earlier releases.
- A schema can be owned by any database principal. This includes roles and application roles.
- A database user can be dropped without dropping objects in a corresponding schema.

Create database SQL2k5
Use SQL2k5

– Created Schema Employee –
Create Schema Employee

– Created table in Employee schema –
Create Table Employee.EmpInfo
  (EmpNo int Primary Key identity(1,1),
   EmpName varchar(20)
  )

– data insertion –

Insert Into Employee.Empinfo Values(‘Jshah-3’)

– Data Selection –
Select * From Employee.Empinfo

– Created another schema HR –
Create Schema HR

– Transfer Objects between Schemas –
ALTER SCHEMA HR
TRANSFER Employee.Empinfo

– Assigning Permission to Schema –
GRANT SELECT ON SCHEMA::HR TO Jshah

**Question: What is Service Pack and Hot fixes in SQL Server?**
With respect to SQL Server, if SQL Server has some in house error/bugs. We can’t resolve it. To remove product level error Microsoft has implemented two ways.

1. Service Packs
2. Hot fixes
Question: What is the difference between a Service Pack and a Hotfix?
Service Packs normally have three properties:

- Provide security fixes
- Correct software errors
- Enhance performance

Service Packs don’t add new functionality or change the interface dramatically. Service Packs are bundled into a programmed delivery method, and are cumulative. That means that you can install Service Pack three without applying Service Pack two, or even one. They are for general use — pretty much everyone should install the Service Pack.

A Hotfix is usually a specific security or software flaw that is addressed in code. There may or may not be a packaged delivery method — some Hotfixes just come with instructions of how and where to copy the patch. Hotfixes are normally not for everyone — Microsoft states that you should only apply the patch if you’re having the specific problem it addresses. Even then, some Hotfixes are only available from a Microsoft support representative.

Question: What does integration of .NET Framework mean for SQL Server 2005?
This feature enables us to execute C# or VB.NET code in the DBMS to take advantage of the .NET functionality. This feature gives more flexibility in writing complex stored procedures, functions, and triggers that can be written in .net compatible language.

Question: What is SSIS?
According to Microsoft SQL Server Integration Services, “(SSIS) is an effective set of tools for both the traditional demands of ETL operations, as well as for the evolving needs of general purpose data integration.” In short, it is the next version of DTS (Data Transformation Services). ETL stands for Extract, Transform and Loading. In short it is a data migration tool that is flexible, fast, and has scalable architecture that enables effective data integration in current business environments.

Question: What is MARS?
In previous versions of SQL Server, applications had to process or cancel all result sets from one batch before it could execute any other batch on that connection. SQL Server 2005 introduces a new connection attribute that allows applications to have more than one pending request per connection, and in particular, to have more than one active default result set per connection. Multiple Active Result Sets (MARS) is the ability to have more than one pending request under a given SQL Server connection. MARS is a programming model enhancement that allows multiple requests to interleave in the server. We need to note that it is not a parallel execution in the server. However, it may benefit us with some performance benefits if used correctly. By default, this feature is not set in SQL Server 2005.

Question: What are the Security Enhancements in SQL Server 2005?
SQL Server 2005 enables administrators to manage permissions at a granular level.

-> In the new SQL Server 2005, we can specify a context under which statements in a module can execute.


-> Administrators can specify Microsoft Windows-style policies on standard logins so that a consistent policy is applied across all accounts in the domain.

-> SQL Server 2005 supports encryption capabilities within the database itself, fully integrated with a key management infrastructure. By default, client-server communications are encrypted.

**Question: What is new with the Reporting services in SQL server 2005?**

SQL Server 2005 Reporting Services is a key component of SQL Server 2005 that provides customers with an enterprise-capable reporting platform. This comprehensive environment is used for authoring, managing, and delivering reports to the entire organization. SQL Server 2005 reporting services have some major changes when compared with the previous version.

-> Changes to the core functionality of the Reporting services in the design of the report, processing, and interactivity

-> Better Integration with other components – Enhanced integration with other components within SQL Server 2005 like SSIS, SSAS and SQL Server Management studio

-> Report Builder – A new reporting tool that enables business users to create their own reports

**Question: What is OLAP?**

Online Analytical Processing (OLAP) allows us to access aggregated and organized data from business data sources, such as data warehouses, in a multidimensional structure called a cube. The arrangement of data into cubes avoids a limitation of relational databases which are not well suited for near instantaneous analysis of large amounts of data. OLAP cubes can be thought of as extensions to the two-dimensional array of a spreadsheet.

**Question: What is Data Mining?**

According to MSDN Data, mining is “the process of extracting valid, authentic, and actionable information from large databases.” Microsoft data mining tools are different from traditional data mining applications in significant ways. Data Mining is a platform for developing intelligent applications, not a stand-alone application. You can build custom applications that are intelligent because the data mining models are easily accessible to the outside world. Further, the model is extensible so that third parties can add custom algorithms to support particular mining needs.

**Question: What is new with the Analysis Services (SSAS) in SQL Server 2005?**

SQL Server 2005 Analysis Services (SSAS) delivers online analytical processing (OLAP) and data mining functionality through a combination of server and client technologies, further reinforced through the use of a specialized development and management environment coupled with a
well-defined object model for designing, creating, deploying, and maintaining business intelligence applications. The server component of Analysis Services is implemented as a Microsoft Windows service. Clients communicate with Analysis Services using the public standard XML for Analysis (XMLA), a SOAP-based protocol. Let us see the enhancements of made to SSAS.

- Supports up to 16 instances of Analysis Services Service.
- As discussed above, the Analysis Services service fully implements the XML for Analysis (XMLA) 1.1 specification. All communication with an instance of Analysis Services is handled through XMLA commands in SOAP messages.
- Uses the Proactive caching.

**Question:** What is Information Schema in SQL Sever 2005?

Information Schema is the part of the SQL- 92 standard which exposes the metadata of the database. In SQL server, a set of views are created in each of the databases which exposes the metadata of the database. The information schema is kept in a separate schema – information schema – which exists in all databases, but which is not included in the search path by default. For more information regarding Information schema please read this article.

**Question:** What is Full Text Search? How does it get implemented in SQL server 2005?

Full-text search allows fast and flexible indexing for keyword-based query of text data stored in a Microsoft SQL Server database. In contrast to the LIKE predicate which only works on character patterns, full-text queries perform linguistic searches against this data, by operating on words and phrases based on rules of a particular language.

**Question:** What is integration of Microsoft Office System mean?

The integration with Microsoft Office system means the following.
- Table Analysis Tools for Excel: Provides an easy-to-use add-in that leverages SQL Server 2005 Data Mining behind the scenes to perform powerful end user analysis on spreadsheet data.
- Data Mining Client for Excel: Offers a full data mining model development lifecycle directly within Excel 2007.
- Data Mining Templates for Visio: Enables powerful rendering and sharing of mining models as annotatable Visio 2007 drawings.

**Question:** What is the support of Web Services in SQL Server 2005?

With this feature the database engine can be directly exposed as a web service without a middle tier or even an IIS. This will enable the user to directly call a stored procedure by calling a web method. This feature is designed with well-known standards such as SOAP 1.2, WSDL 1.1, and HTTP. With this new feature we can now connect to SQL Server not only with TDS- Tabular data stream (a binary protocol for connecting to SQL Server 2005) but also over SOAP/ HTTP.

**Question:** What is OLTP?
Online Transaction Processing (OLTP) relational databases are optimal for managing changing data. When several users are performing transactions at the same time, OLTP databases are designed to let transactional applications write only the data needed to handle a single transaction as quickly as possible.

**Question: What is Snapshot in SQL Server 2005?**
A database snapshot is a read-only, static view of a database, the source database. Each database snapshot is transaction-consistent with the source database as it existed at the time of the snapshot’s creation.

**Question: What is snapshot isolation in SQL Server 2005?**
SQL Server 2005 introduces a new “snapshot” isolation level that is intended to enhance concurrency for online transaction processing (OLTP) applications. In prior versions of SQL Server, concurrency was based solely on locking, which can cause blocking and deadlocking problems for some applications. Snapshot isolation depends on enhancements to row versioning and is intended to improve performance by avoiding reader-writer blocking scenarios.

**Question: What is Database Partitioning in SQL Server 2005?**
SQL Server 2005 provides a new capability for the partitioning of tables across file groups in a database. Partitioning a database improves performance and simplifies maintenance. By splitting a large table into smaller, individual tables, queries accessing only a fraction of the data can run faster because there is less data to scan.

**Question: What is SQL Server Agent?**
SQL Server Agent is a Microsoft Windows service that executes scheduled administrative tasks called jobs. SQL Server Agent uses SQL Server to store job information. Jobs contain one or more job steps. We generally schedule the backups on the production databases using the SQL server agent. In SQL Server 2005 we have roles created for using SQL Server agents.

- SQLAgentUserRole
- SQLAgentReaderRole
- SQLAgentOperatorRole

SQL Server Agent for SQL Server 2005 provides a more robust security design than earlier versions of SQL Server. This improved design gives system administrators the flexibility they need to manage their Agent service.

**Question: What is Replication? What is the need to have the replication? What are the enhancements made to SQL Server 2005 related to the replication?**
“Replication is a set of technologies for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency.” In short, replication is all about having multiple copies of the same database. We need
replication when we need to distribute data to and from different locations. Generally we have a master copy of data. There will be multiple slaves (Clients) located at various locations which need to be replicated. We use replication for a variety of reasons. Load balancing is sharing the data among a number of servers and distributing the query load. Offline processing is one of the main reasons. In this scenario we need to modify the data on the database that is not connected to the network. The last reason may be to have a back-up to the database in case of failure to the existing database. Let us see the enhancements of SQL server 2005 database related to replication.

- **Database Mirroring** – Database Mirroring is moving the transactions of database from one SQL Server database to another SQL server database on a different SQL Server.
- **Replication Management topology (RMO)** – RMO is a new construct in SQL Server 2005. It is a .NET Framework library that provides a set of common language runtime classes for configuring, managing, and scripting replication, and for synchronizing Subscribers.

**Question:** What are Business Logic Handlers?
Business logic handlers are written in managed code and allow us to execute custom business logic during the merge synchronization. We can invoke the business logic handler in case of non-conflicting data changes. Business logic handler can perform one of the following three actions.
- Reject Data
- Accept Data
- Apply Custom Data

**Question:** What are different variants of SQL Server 2005?
There are different variants of SQL Server 2005 commercially available.
- Express – Free and only for one user
- Enterprise – 5 users apart from server
- Workgroup – 10 users apart from server
- Standard – 25 users apart from server

**Question:** What are Various Service packs available for SQL Server 2005?
As of now there are two service packs available for the SQL Server 2005.
- Service Pack 1 – Has major changes or enhancements to SQL Server 2005 in Analysis Services, Data Programmability, SSIS, and reporting services.
- Service Pack 2 – Unlike Service Pack 2, this service pack enables SQL Server 2005 customers to take advantage of the enhancements within Windows Vista and the 2007 Office system.

**Question:** What are the New Data types introduced in SQL Server 2005?
SQL Server 2005 has added some new data types to its existing data types.
- XML Data type
- VARCHAR (MAX)
As we can see, the new term MAX has been introduced in SQL Server 2005. This new specifier expands the storage capabilities of the varchar, nvarchar, and varbinary data types. Varchar(max), nvarchar(max), and varbinary(max) are collectively called large-value data types.

**Question:** Does SQL Server 2005 support SMTP?
SQL Server 2005 now supports sending E-mail from the database. It is called as database mail and it uses DatabaseMail90.exe. Gone are the days when we were using a third party component for this. Receiving an e-mail was not supported in the previous versions of SQL Server.

**Question:** What is SQL Management Object is SQL Server 2005?
These are collection of objects that are made for programming all aspects of managing Microsoft SQL Server 2005. SMO is a .NET based object model. It comes with SQL Server 2005 as a .Net assembly named Microsoft.SqlServer.Smo.dll. We can use these objects for connecting to a database, calling methods of the database that returns a table, using transactions, transferring data, scheduling administrative tasks, etc. The best part about SMO is that most of it can also be used with SQL Server 2000.

**Question:** What is SQL Service Broker in SQL Server 2005?
SQL Service broker is a new technology introduced in SQL Server 2005 for building database-intensive distributed applications. Basically, service broker has been built for developing applications that consist of individual components which are loosely coupled. Service broker supports asynchronous yet reliable messages that are passed between the components. These messages are called conversations.

**Question:** What is database snapshot?
A database snapshot provides a read-only, static view of a source database as it existed at snapshot creation, minus any uncommitted transactions. Uncommitted transactions are rolled back in a newly created database snapshot because the Database Engine runs recovery after the snapshot has been created.

**Question:** Is database snapshot transitionally consistent?
Yes.
Each database snapshot is transitionally consistent with the source database at the moment of the snapshot’s creation. When we create a database snapshot, the source database will typically have number open transactions. Before the snapshot becomes available, the open transactions are rolled back to make the database snapshot transitionally consistent. Just like it follows the recovery interval step and it will not affect the source database.
Question: What are the uses of database snapshot?
Reporting Purpose
we can recover damaged database using database snapshot
also useful if we are planning to do major change in the source database
Mirroring database is always in recovering mode, to read that database we can use database snapshot.

Question: Is Developer edition of SQL Server 2005 supports database snapshot?
No.

Question: What is copy-on-write operation in database snapshot?
Database snapshots operate at the data-page level. Before a page of the source database is modified for the first time, the original page is copied from the source database to the snapshot. This process is called a copy-on-write operation. The snapshot stores the original page, preserving the data records as they existed when the snapshot was created.

Question: What is Sparse file in Database Snapshot?
To store the copied original pages, the snapshot uses one or more sparse files. Initially, a sparse file is an essentially empty file that contains no user data and has not yet been allocated disk space for user data. As more and more pages are updated in the source database, the size of the file grows. When a snapshot is taken, the sparse file takes up little disk space. As the database is updated over time, however, a sparse file can grow into a very large file.

Question: What is NTFS File System?
The NTFS acronym stands for New Technology File System. The name derives from the implementation of very innovative data storage techniques that were refined in NTFS. While none of the techniques are unique to NTFS, it is the first time that so many innovations were released at once on a production file system. The FAT file system had long been criticized for not including some of the more obvious improvements such as journaling, disk quotas, and file compression. However, these improvements made NTFS incompatible with previous versions of Windows, and also with hard disk tools designed for FAT file systems. For example, data recovery tools such as GetDataBack and partitioning tools such as PartitionMagic would run on Windows NT, yet could not function on the newer file system. This led to much frustration with users who had purchased licenses for these products before upgrading to Windows NT.

Question: Explain Sparse file Size in database snapshot
At the creation time sparse file will take very little space, but as the data changes occurred into the parent database, data page before the changes copied into the sparse file. Thus sparse file grows. Sparse files are the feature of NTFS file system. As the sparse file grows NTFS will allocate the space to sparse file gradually.
Question: Why does the size of a sparse file slightly exceed than the space actually filled by pages in it?
Sparse files grow in 64-kilobyte (KB) increments; thus, the size of a sparse file on disk is always a multiple of 64 KB. The latest 64-KB increment holds from one to eight 8-KB pages, depending on how many pages have been copied from the source database. This means that, on the average, the size of a sparse file slightly exceeds the space actually filled by pages.

Question: How to create database snapshot?
1. Based on the current size of the source database, ensure that you have sufficient disk space to hold the database snapshot. The maximum size of a database snapshot is the size of the source database at snapshot creation.
2. Issue a CREATE DATABASE statement on the files using the AS SNAPSHOT OF clause. Creating a snapshot requires specifying the logical name of every database file of the source database.

Syntax
CREATE DATABASE AdventureWorks_SS ON
(NAME = AdventureWorks_Data,
FILENAME = ‘D:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\AdventureWorks_Data.ss’
)
AS SNAPSHOT OF AdventureWorks

Question: Explain SQL Server System Databases?

Master
- Purpose – Core system database to manage the SQL Server instance. In SQL Server 2005, the Master database is the logical repository for the system objects residing in the sys schema. In SQL Server 2000 and previous editions of SQL Server, the Master database physically stored all of the system objects.
- Prominent Functionality
  - Per instance configurations
  - Databases residing on the instance
  - Files for each database
  - Logins
  - Linked\Remote servers
  - Endpoints
- Additional Information
  - The first database in the SQL Server startup process
In SQL Server 2005, needs to reside in the same directory as the Resource database

Resource
- Purpose – The Resource database is responsible for physically storing all of the SQL Server 2005 system objects. This database has been created to improve the upgrade and rollback of SQL Server system objects with the ability to overwrite only this database.
- Prominent Functionality
  - System object definition
- Additional Information
  - Introduced in SQL Server 2005 to help manage the upgrade and rollback of system objects
  - Prior to SQL Server 2005 the system related data was stored in the master database
  - Read-only database that is not accessible via the SQL Server 2005 tool set
  - The database ID for the Resource database is 32767
  - The Resource database does not have an entry in master.sys.databases

TempDB
- Purpose – Temporary database to store temporary tables (#temptable or ##temptale), table variables, cursors, work tables, row versioning, create or rebuild indexes sorted in TempDB, etc. Each time the SQL Server instance is restarted all objects in this database are destroyed, so permanent objects cannot be created in this database.
- Prominent Functionality
  - Manage temporary objects listed in the purpose above
- Additional Information
  - Each time a SQL Server instance is rebooted, the TempDB database is reset to its original state

Model
- Purpose – Template database for all user defined databases
- Prominent Functionality
  - Objects
  - Columns
  - Users
- Additional Information
  - User defined tables, stored procedures, user defined data types, etc can be created in the Model database and will exist in all future user defined databases
  - The database configurations such as the recovery model for the Model database are applied to future user defined databases

MSDB
- Purpose – Primary database to manage the SQL Server Agent configurations
- Prominent Functionality
- SQL Server Agent Jobs, Operators and Alerts
- DTS Package storage in SQL Server 7.0 and 2000
- SSIS Package storage in SQL Server 2005

**Additional Information**
- Provides some of the configurations for the SQL Server Agent service
- For the SQL Server 2005 Express edition installations, even though the SQL Server Agent service does not exist, the instance still has the MSDB database

**Distribution**
- **Purpose** – Primary data to support SQL Server replication
- **Prominent Functionality**
  - Database responsible for the replication meta data
  - Supports the data for transaction replication between the publisher and subscriber(s)

**ReportServer**
- **Purpose** – Primary database for Reporting Services to store the meta data and object definitions
- **Prominent Functionality**
  - Reports security
  - Job schedules and running jobs
  - Report notifications
  - Report execution history

**ReportServerTempDB**
- **Purpose** – Temporary storage for Reporting Services
- **Prominent Functionality**
  - Session information
  - Cache

**Question: What is CTE (Common Table Expression)?**
**Ans:** CTE is a temporary Table created from a simple SQL query. You can say it’s a view. Below is a sample CTE created “PurchaseOrderHeaderCTE” from “PurchaseOrderHeader”

```sql
WITH PurchaseOrderHeaderCTE (Orderdate, status) As
    (SELECT order date, status
    FROM Purchasing.PurchasingOrderHeader
    )

SELECT * FROM PurchasingOrderHeader
```

The WITH Statement define the CTE and later using the CTE name I have display the CTE data.
Question: Why would you use CTE rather than simple View?
With CTE you can use a recursive query with CTE itself. That’s not possible with view.

Question: What is TRY/CATCH block in T-SQL?
NO I am referring to .NET TRY/CATCH block, this is a new way of handling error in SQL Server. For instance in the below T-SQL code any error during delete statement is caught and the neccessary error information is displayed:

BEGIN TRY
DELETE table 1 WHERE id=122
END TRY

BEGIN CATCH
SELECT
ERROR_NUMBER As ErrNum,
ERROR_SEVERITY ( ) As ErrSev,
ERROR_STATE ( ) As ErrSt,
ERROR_MESSAGE ( ) As ErrMsg;
END CATCH

Question: What is PIVOT feature in SQL Server?
PIVOT feature convert data row to column for better analytical view. Below is a sample PIVOT filed using CTE. OK the first section is the CTE which is the input and later PIVOT is applied Over it:

WITH PURCHASEORDERHEADERCTE(Orderdate, Status, Subtotal) As
(
select year (orderdate), status, isnull(SubTotal, 0) from purchasing. PURCHASEORDER HEADER
)

Select Status as OrderStatus, isnull ([2006], 0) as ‘YR 2007’, isnull([2007], 0) as ‘Yr 2007’ from PURCHASEORDERHEADERCTE Pivot (sum(subtotal) for Orderdate in ([2006], [2007] ) ) as pivoted
You can see from the above SQL the top WITH statement is the CTE supplied to the PIVOT. After that PIVOT is applied on subtotal and orderdate. You have to secify in what you want the pivot (here it is 2006 and 2007). So below is the output of CTE table.

<table>
<thead>
<tr>
<th>Status</th>
<th>OrderStatus</th>
<th>YR 2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2006</td>
<td>4 201.4</td>
</tr>
<tr>
<td>2</td>
<td>2006</td>
<td>1 272.1015</td>
</tr>
</tbody>
</table>
CTE output
After the PIVOT is applied you can see the rows now grouped column wise with the subtotal assigned to each. You can summarize that PIVOT summarizes your data in cross tab format.

Order Status Yr 2006 Yr 2007
1 3 171.0765 383552.904
2 1 272.1015 0.00
3 4 103452.643 3842580.126

Question: What is UNPIVOT?
It’s exactly the vice versa of PIVOT. That means you have a PIVOTED data and you want to UNPIVOTED.

Question: What are RANKING functions?
They add columns that are calculated based on a ranking algorithm. These functions include ROW_NUMBER(), RANK(), DENSE_RANK(), and NTILE().

Question: What is ROW_NUMBER()?
The ROW_NUMBER() function adds a column that display a number corresponding the row’s position is the query result. If the column that you specify in the OVER clause is not unique, it still produces an incrementing column based on the column specified in the OVER Clause. You can see in the figure below I have applied ROW_NUMBER function over column col2 and you notice the incrementing numbers generated.

Select col1, col2, row_number(), over (order by col2) as ROW_NUMBER from table _1

<table>
<thead>
<tr>
<th>col1</th>
<th>col2</th>
<th>RowNumber</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
**Question: What is RANK()?**
The `RANK()` function works much like `ROW_NUMBER()` function in that it numbers records in order. When the column specified by the `ORDER BY` clause contains unique values, then `ROW_NUMBER()` and `RANK()` produce identical results. They differ in the way they work when duplicate values are contained in the `ORDER BY` expression. `ROW_NUMBER` will increment the number by one every record, regardless of duplicates. `RANK()` produces a single number for each value in the result set. You can see duplicate value it does not increment the TOP number.

```
SELECT col1, col2, RANK(), over (order by col2) as ROW_NUMBER from table _1
```

```
col1 col2 ROW NUMBER
1 2 1
2 3 2
4 3 3
4 3 4
5 6 5
```

**Question: What is DENSE_RANK()?**
`DENSE_RANK()` works the same way as `RANK()` does but eliminates the gaps in the numbering. When I say GAPS you can see in previous results it has eliminated 4 and 5 from the count because of the gap in between COL@. BUT for `dense_rank` it overlooks the gap.

```
SELECT col1, col2, Dense_rank(), over (order by col2) as ROW_NUMBER from table _1
```

```
col1 col2 RowNumber
1 2 1
2 3 2
4 3 2
4 3 2
5 6 3
5 6 3
```

**Question: What is NTILE()?**
`NTILES()` breaks the result set into a specified number of groups and assigns the same number to each record in a group. OK `NTILE` just groups depending on the number given or you can say divide the data. For instance I have said to `NTILE` it to 3. It has 6 total rows so it grouped in number of 2.

```
SELECT col1, col2,NTILE(3) over (order by col2) as ROW_NUMBER from table _1
```
Question: What is SQL Injection?
It is a form of attack on a database-driven Web site in which the attacker executes unauthorized SQL Commands by taking advantage of insecure code on a system connected to the Internet by passing the firewall. SQL injection attacks are used to steal information from the database from which the data would normally not be available and gain access to an organization’s host computers through the computer that is hosting the database.

SQL injection attacks typically are easy to avoid by ensuring that a system has strong input validation.
As name suggest’s we inject SQL which can be relatively dangerous for the database.
Example:

SELECT Email, passwd, login_id, Full_name
FROM Member
WHERE email = ‘X’

Now some body does not put “x” as the input but put “x; DROP TABLE members;”. So the actual SQL will execute is:

SELECT Email, passwd, login_id, Full_name
FROM Member
WHERE email = ‘X’ ; DROP TABLE members

Question: What is a “Database Transactions”?
A database transaction is a unit of work performed against a database management system or similar system that is treated in a coherent and reliable way independent of other transactions. A database transaction, by definition, must be atomic, consistent, isolated and durable. These properties of database transactions are often referred to by the acronym ACID.

Transactions provide an “all-or-nothing” proposition stating that work units performed in a database must be completed in their entirety or take no effect whatsoever. Further, transactions must be isolated from other transactions, results must conform to existing constraints in the database and transactions that complete successfully must be committed to durable storage.
In some systems, transactions are also called LUWs for Logical Units of Work.

**Question: What is ACID?**
The ACID model is one of the oldest and most important concepts of database theory. It sets forward four goals that every database management system must strive to achieve: atomicity, consistency, isolation and durability. No database that fails to meet any of these four goals can be considered reliable.

Let’s take a moment to examine each one of these characteristics in detail:

**Atomicity** states that database modifications must follow an “all or nothing” rule. Each transaction is said to be “atomic.” If one part of the transaction fails, the entire transaction fails. It is critical that the database management system maintain the atomic nature of transactions in spite of any DBMS, operating system or hardware failure.

**Consistency** states that only valid data will be written to the database. If, for some reason, a transaction is executed that violates the database’s consistency rules, the entire transaction will be rolled back and the database will be restored to a state consistent with those rules. On the other hand, if a transaction successfully executes, it will take the database from one state that is consistent with the rules to another state that is also consistent with the rules.

**Isolation** requires that multiple transactions occurring at the same time not impact each other’s execution. For example, if Joe issues a transaction against a database at the same time that Mary issues a different transaction, both transactions should operate on the database in an isolated manner. The database should either perform Joe’s entire transaction before executing Mary’s or vice-versa. This prevents Joe’s transaction from reading intermediate data produced as a side effect of part of Mary’s transaction that will not eventually be committed to the database. Note that the isolation property does not ensure which transaction will execute first, merely that they will not interfere with each other.

**Durability** ensures that any transaction committed to the database will not be lost. Durability is ensured through the use of database backups and transaction logs that facilitate the restoration of committed transactions in spite of any subsequent software or hardware failures.

**Question: What is “Begin Trans”, “Commit Tran”, “Rollback Tran” and “Save Tran”?**
Transactions group a set of tasks into a single execution unit. Each transaction begins with a specific task and ends when all the tasks in the group successfully complete. If any of the tasks fails, the transaction fails. Therefore, a transaction has only two results: success or failure. Incomplete steps result in the failure of the transaction.

Users can group two or more Transact-SQL statements into a single transaction using the following statements:
* Begin Transaction
* Rollback Transaction
* Commit Transaction

**Begin Transaction**
Marks the starting point of an explicit, local transaction. BEGIN TRANSACTION increments @@TRANCOUNT by 1.

**Rollback Transaction**
If anything goes wrong with any of the grouped statements, all changes need to be aborted. The process of reversing changes is called rollback in SQL Server terminology. A ROLLBACK, on the other hand, works regardless of the level at which it is issued, but rolls back all transactions, regardless of the nesting level.

**Commit Transaction**
If everything is in order with all statements within a single transaction, all changes are recorded together in the database. In SQL Server terminology, we say that these changes are committed to the database. A COMMIT issued against any transaction except the outermost one doesn’t commit any changes to disk – it merely decrements the @@TRANCOUNT automatic variable.

**Save Tran**
Savepoints offer a mechanism to roll back portions of transactions. A user can set a savepoint, or marker, within a transaction. The savepoint defines a location to which a transaction can return if part of the transaction is conditionally canceled. SQL Server allows you to use savepoints via the SAVE TRAN statement, which doesn’t affect the @@TRANCOUNT value. A rollback to a savepoint (not a transaction) doesn’t affect the value returned by @@TRANCOUNT, either. However, the rollback must explicitly name the savepoint: using ROLLBACK TRAN without a specific name will always roll back the entire transaction.

**Question: What are “Checkpoint’s” in SQL Server?**
Forces all dirty pages for the current database to be written to disk. Dirty pages are data or log pages modified after entered into the buffer cache, but the modifications have not yet been written to disk.

**Syntax**
CHECKPOINT

**Question: What are “Implicit Transactions”?**
Microsoft SQL Server operates in three transaction modes:

**Autocommit transactions**
Each individual statement is a transaction.
Explicit transactions
Each transaction is explicitly started with the BEGIN TRANSACTION statement and explicitly ended with a COMMIT or ROLLBACK statement.

Implicit transactions
A new transaction is implicitly started when the prior transaction completes, but each transaction is explicitly completed with a COMMIT or ROLLBACK statement.

Question: Is it good to use “Implicit Transactions”?
If you want all your commands to require an explicit COMMIT or ROLLBACK in order to finish, you can issue the command SET IMPLICIT_TRANSACTIONS ON. By default, SQL Server operates in the autocommit mode; it does not operate with implicit transactions. Any time you issue a data modification command such as INSERT, UPDATE, or DELETE, SQL Server automatically commits the transaction. However, if you use the SET IMPLICIT_TRANSACTIONS ON command, you can override the automatic commitment so that SQL Server will wait for you to issue an explicit COMMIT or ROLLBACK statement to do anything with the transaction. This can be handy when you issue commands interactively, mimicking the behavior of other databases such as Oracle.

What’s distinctive about implicit transactions is that reissuing SET IMPLICIT_TRANSACTIONS ON does not increase the value of @@TRANCOUNT. Also, neither COMMIT nor ROLLBACK reduce the value of @@TRANCOUNT until after you issue the command SET IMPLICIT_TRANSACTIONS OFF. Developers do not often use implicit transactions; however, there is an interesting exception in ADO. See the sidebar, Implicit Transactions and ADO Classic.

Question: What is Concurrency?
When many people attempt to modify data in a database at the same time, a system of controls must be implemented so that modifications made by one person do not adversely affect those of another person. This is called concurrency control.

Concurrency control theory has two classifications for the methods of instituting concurrency control:

Pessimistic concurrency control
A system of locks prevents users from modifying data in a way that affects other users. After a user performs an action that causes a lock to be applied, other users cannot perform actions that would conflict with the lock until the owner releases it. This is called pessimistic control because it is mainly used in environments where there is high contention for data, where the cost of protecting data with locks is less than the cost of rolling back transactions if concurrency conflicts occur.

Optimistic concurrency control
In optimistic concurrency control, users do not lock data when they read it. When an update is performed, the system checks to see if another user changed the data after it was read. If another user updated the data, an error is raised. Typically, the user receiving the error rolls
back the transaction and starts over. This is called optimistic because it is mainly used in environments where there is low contention for data, and where the cost of occasionally rolling back a transaction outweighs the costs of locking data when read.

**Question: What are “Dirty reads”?**
Uncommitted dependency occurs when a second transaction selects a row that is being updated by another transaction. The second transaction is reading data that has not been committed yet and may be changed by the transaction updating the row.

**Question: What are “Unrepeatable reads”?**
Inconsistent Analysis (Nonrepeatable Read)
Inconsistent analysis occurs when a second transaction accesses the same row several times and reads different data each time. Inconsistent analysis is similar to uncommitted dependency in that another transaction is changing the data that a second transaction is reading. However, in inconsistent analysis, the data read by the second transaction was committed by the transaction that made the change. Also, inconsistent analysis involves multiple reads (two or more) of the same row and each time the information is changed by another transaction; thus, the term nonrepeatable read.

**Question: What is “Phantom rows”?**
Phantom reads occur when an insert or delete action is performed against a row that belongs to a range of rows being read by a transaction. The transaction’s first read of the range of rows shows a row that no longer exists in the second or succeeding read, as a result of a deletion by a different transaction. Similarly, as the result of an insert by a different transaction, the transaction’s second or succeeding read shows a row that did not exist in the original read.

For example, an editor makes changes to a document submitted by a writer, but when the changes are incorporated into the master copy of the document by the production department, they find that new unedited material has been added to the document by the author. This problem could be avoided if no one could add new material to the document until the editor and production department finish working with the original document.

**Question: What is “Lost Updates”?**
Lost updates occur when two or more transactions select the same row and then update the row based on the value originally selected. Each transaction is unaware of other transactions. The last update overwrites updates made by the other transactions, which results in lost data.

**Question: What are different levels of granularity of locking resources?**
Microsoft SQL Server 2000 has multigranular locking that allows different types of resources to be locked by a transaction. To minimize the cost of locking, SQL Server locks resources automatically at a level appropriate to the task. Locking at a smaller granularity, such as rows, increases concurrency, but has a higher overhead because more locks must be held if many
rows are locked. Locking at a larger granularity, such as tables, are expensive in terms of concurrency because locking an entire table restricts access to any part of the table by other transactions, but has a lower overhead because fewer locks are being maintained.

SQL Server can lock these resources (listed in order of increasing granularity).

**RID:** Row identifier. Used to lock a single row within a table.

**Key:** Row lock within an index. Used to protect key ranges in serializable transactions.

**Page:** 8 kilobyte –(KB) data page or index page.

**Extent:** Contiguous group of eight data pages or index pages.

**Table:** Entire table, including all data and indexes.

**DB:** Database.

**Question:** What are different types of Isolation levels in SQL Server?

**READ COMMITTED**

Specifies that shared locks are held while the data is being read to avoid dirty reads, but the data can be changed before the end of the transaction, resulting in nonrepeatable reads or phantom data. This option is the SQL Server default.

**READ UNCOMMITTED**

Implements dirty read, or isolation level 0 locking, which means that no shared locks are issued and no exclusive locks are honored. When this option is set, it is possible to read uncommitted or dirty data; values in the data can be changed and rows can appear or disappear in the data set before the end of the transaction. This option has the same effect as setting NOLOCK on all tables in all SELECT statements in a transaction. This is the least restrictive of the four isolation levels.

**REPEATABLE READ**

Locks are placed on all data that is used in a query, preventing other users from updating the data, but new phantom rows can be inserted into the data set by another user and are included in later reads in the current transaction. Because concurrency is lower than the default isolation level, use this option only when necessary.

**SERIALIZABLE**

Places a range lock on the data set, preventing other users from updating or inserting rows into the data set until the transaction is complete. This is the most restrictive of the four isolation levels. Because concurrency is lower, use this option only when necessary. This option has the same effect as setting HOLDLOCK on all tables in all SELECT statements in a transaction.

**Question:** If you are using COM+, what “Isolation” level is set by default?

SERIALIZABLE transaction isolation level is the default isolation level for the COM+ application.
Question: What are “Lock” hints?
A range of table-level locking hints can be specified using the SELECT, INSERT, UPDATE, and DELETE statements to direct Microsoft SQL Server 2000 to the type of locks to be used. Table-level locking hints can be used when a finer control of the types of locks acquired on an object is required. These locking hints override the current transaction isolation level for the session.

Question: What is a “Deadlock”?
Deadlocking occurs when two user processes have locks on separate objects and each process is trying to acquire a lock on the object that the other process has. When this happens, SQL Server identifies the problem and ends the deadlock by automatically choosing one process and aborting the other process, allowing the other process to continue. The aborted transaction is rolled back and an error message is sent to the user of the aborted process. Generally, the transaction that requires the least amount of overhead to rollback is the transaction that is aborted.

Question: What are the steps you can take to avoid “Deadlocks”?
Here are some tips on how to avoid deadlocking on your SQL Server:
* Ensure the database design is properly normalized.
* Have the application access server objects in the same order each time.
* During transactions, don’t allow any user input. Collect it before the transaction begins.
* Avoid cursors.
* Keep transactions as short as possible. One way to help accomplish this is to reduce the number of round trips between your application and SQL Server by using stored procedures or keeping transactions with a single batch. Another way of reducing the time a transaction takes to complete is to make sure you are not performing the same reads over and over again. If your application does need to read the same data more than once, cache it by storing it in a variable or an array, and then re-reading it from there, not from SQL Server.
* Reduce lock time. Try to develop your application so that it grabs locks at the latest possible time, and then releases them at the very earliest time.
* If appropriate, reduce lock escalation by using the ROWLOCK or PAGLOCK.
* Consider using the NOLOCK hint to prevent locking if the data being locked is not modified often.
* If appropriate, use as low of an isolation level as possible for the user connection running the transaction.
* Consider using bound connections.

Question: What is Bound Connection?
Bound connections allow two or more connections to share the same transaction and locks. Bound connections can work on the same data without lock conflicts. Bound connections can be created from multiple connections within the same application, or from multiple applications with separate connections. Bound connections make coordinating actions across multiple connections easier.
To participate in a bound connection, a connection calls `sp_getbindtoken` or `srv_getbindtoken` (Open Data Services) to get a bind token. A bind token is a character string that uniquely identifies each bound transaction. The bind token is then sent to the other connections participating in the bound connection. The other connections bind to the transaction by calling `sp_bindsession`, using the bind token received from the first connection.

**Question: Specify the types of Bound Connections**

**Local bound connection**
Allows bound connections to share the transaction space of a single transaction on a single server.

**Distributed bound connection**
Allows bound connections to share the same transaction across two or more servers until the entire transaction is either committed or rolled back by using Microsoft Distributed Transaction Coordinator (MS DTC).

**Question: How can I know what locks are running on which resource?**
Use `SP_Locks` system stored procedure

**Question: What is database or database management systems (DBMS)?**
A collection of programs that enables you to store, modify, and extract information from a database. There are many different types of DBMSs, ranging from small systems that run on personal computers to huge systems that run on mainframes.
The following are examples of database applications:
* computerized library systems
* automated teller machines
* flight reservation systems
* computerized parts inventory systems

**Question: What is difference between DBMS and RDBMS?**
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 is a Relational Data Base Management System Relational DBMS. This adds the additional condition that the system supports a tabular structure for 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.

Many DBA’s think that RDBMS is a Client Server Database system but that’s not the case with RDBMS.
Yes you can say DBMS does not impose any constraints or security with regard to data manipulation it is user or the programmer responsibility to ensure the ACID PROPERTY of the database whereas the rdbms is more with this regard bcz rdbms define the integrity constraint for the purpose of holding ACID PROPERTY.

**Question: Explain CODD rules?**

A relational DBMS must use its relational facilities exclusively to manage and interact with the database.

The rules:

These rules were defined by Codd in a paper published in 1985. They specify what a relational database must support in order to be relational. These rules have been considerably extended in reference [1].

1. **Information rule**
   - Data are represented only one way: as values within columns within rows.
   - Simple, consistent and versatile.
   - The basic requirement of the relational model.

2. **Guaranteed access rule**
   - Every value can be accessed by providing table name, column name and key.
   - All data are uniquely identified and accessible via this identity.

3. **Systematic treatment of null values**
   - Separate handling of missing and/or non applicable data.
   - This is distinct to zero or empty strings
   - Codd would further like several types of null to be handled.

4. **Relational online catalog**
   - Catalog (data dictionary) can be queried by authorized users as part of the database.
   - The catalog is part of the database.

5. **Comprehensive data sublanguage**
   - Used interactively and embedded within programs
   - Supports data definition, data manipulation, security, integrity constraints and transaction processing
   - Today means: must support SQL.

6. **View updating rule**
   - All theoretically possible view updates should be possible.
   - Views are virtual tables. They appear to behave as conventional tables except that they are built dynamically when the query is run. This means that a view is always up to date. It is not always theoretically possible to update views. Codd himself, did not completely understand this. One problem exists when a view relates to part of a table not including a candidate key. This means that potential updates would violate the entity integrity rule.

7. **High-level insert, update and delete**
   - Must support set-at-a-time updates.
   - ie. Transactions
* eg: UPDATE mytable SET mycol = value WHERE condition;
Many rows may be updated with this single statement.
8. Physical data independence
* Physical layer of the architecture is mapped onto the logical layer.
* Users and programs are not dependent on the physical structure of the database.
* (Physical layer implementation is dependent on the DBMS.)
9. Logical data independence
* Users and programs are independent of the logical structure of the database.
* i.e.: the logical structure of the data can evolve with minimal impact on the programs.
10. Integrity independence
* Integrity constraints are to be stored in the catalog not the programs.
* Alterations to integrity constraints should not affect application programs.
* This simplifies the programs.
* It is not always possible to do this.
11. Distribution independence
* Applications should still work in a distributed database (DDB).
12. Nonsubversion rule
* If there is a record-at-a-time interface (eg via 3GL), security and integrity of the database must not be violated.
* There should be no backdoor to bypass the security imposed by the DBMS.

**Question: Is access database a RDBMS?**
Yes Access is RDBMS

**Question: Explain page splits?**
A page is 8Kbytes of data which can be index related, data related, large object binary (lob’s) etc...

When you insert rows into a table they go on a page, into ‘slots’, your row will have a row length and you can only so many rows on the 8Kbyte page. What happens when that row’s length increases because you entered a bigger product name in your varchar column for instance, well, SQL Server needs to move the other rows along in order to make room for your modification, if the combined new length of all the rows on the page will no longer fit on that page then SQL Server grabs a new page and moves rows to the right or left of your modification onto it – that is called a ‘page split’.

**Question: What are E-R diagrams?**
Definition: An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.
Also Known As: ER Diagram, E-R Diagram, entity-relationship model
Question: 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.

Case sensitivity
If A and a, B and b, etc. are treated in the same way then it is case-insensitive. A computer treats A and a differently because it uses ASCII code to differentiate the input. The ASCII value of A is 65, while a is 97. The ASCII value of B is 66 and b is 98.

Accent sensitivity
If a and á, o and ó are treated in the same way, then it is accent-insensitive. A computer treats a and á differently because it uses ASCII code for differentiating the input. The ASCII value of a is 97 and á is 225. The ASCII value of o is 111 and ó is 243.

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 when represented as a double-byte character (full-width) are treated differently then it is width sensitive. Database, Tables and columns with different collation

SQL Server 2000 allows the users to create databases, tables and columns in different collations.

Question: What is Extent and Page?
The fundamental unit of data storage in Microsoft SQL Server™ is the page. In SQL Server 2000, the page size is 8 KB. This means SQL Server 2000 databases have 128 pages per megabyte.

The start of each page is a 96-byte header used to store system information, such as the type of page, the amount of free space on the page, and the object ID of the object owning the page.

Types of pages in SQL Server
Data
Index
Text/Image
Global Allocation Map
Secondary Global Allocation Map
Index Allocation Map
Question: What is normalization? What are different types of normalization?
In relational database design, the process of organizing data to minimize redundancy. 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.

There are three main normal forms, each with increasing levels of normalization:

# First Normal Form (1NF): Each field in a table contains different information. For example, in an employee list, each table would contain only one birthdate field.

# Second Normal Form (2NF): Each field in a table that is not a determiner of the contents of another field must itself be a function of the other fields in the table.

# Third Normal Form (3NF): No duplicate information is permitted. So, for example, if two tables both require a birthdate field, the birthdate information would be separated into a separate table, and the two other tables would then access the birthdate information via an index field in the birthdate table. Any change to a birthdate would automatically be reflect in all tables that link to the birthdate table.

There are additional normalization levels, such as Boyce Codd Normal Form (BCNF), fourth normal form (4NF) and fifth normal form (5NF). While normalization makes databases more efficient to maintain, they can also make them more complex because data is separated into so many different tables.

Question: What is denormalization?
As the name indicates, denormalization is the reverse process of normalization. It’s the controlled introduction of redundancy in to the database design. It helps improve the query performance as the number of joins could be reduced.