SQL

INTERVIEW QUESTION AND ANSWER

Microsoft® SQL Server®

ER. AVNINDRA KANAUJIA

Table of Contents

Q1 What are the difference between clustered and a non-clustered index?
Q2 What are the different index configurations a table can have?
Q3 What are cursors?
Q4 What is the use of DBCC commands?
Q5 What is a Linked Server?
Q6 What is Collation?
Q7 What are different types of Collation Sensitivity?
Q8 What's the difference between a primary key and a unique key?
Q9 How to implement one-to-one, one-to-many and many-to many relationships while designing tables?
Q10 What is a NOLOCK?
Q11 What is difference between DELETE & TRUNCATE commands?
Q12 Difference between Function and Stored Procedure?
Q13 When is the use of UPDATE_STATISTICS command?
Q14 What types of Joins are possible with Sql Server?
Q15 What is the difference between a HAVING CLAUSE and a WHERE CLAUSE?
Q16 What is sub-query? Explain properties of sub-query.
Q17 What are types of sub-queries?
Q18 What is SQL Profiler?
Q18.1 What is User Defined Functions?
Q19 What kind of User-Defined Functions can be created?
Q20 Inline Table-Value User-Defined Function.
Q21 Multi-statement Table-Value User-Defined Function.
Q22 Which TCP/IP port does SQL Server run on? How can it be changed?
Q23 What are the authentication modes in SQL Server? How can it be changed?
Q24 Where are SQL server users names and passwords are stored in sql server?
Q25 Which command using Query Analyzer will give you the version of SQL server and operating system?
Q26 What is SQL server agent?
Q27 What is @@ERROR?
Q28 What is Raiseerror?
Q29 What is log shipping?
Q30 What is the difference between a local and a global variable?
Q31 What are the properties of the Relational tables?
Q32 What is De-normalization?

Q33 How to get @@error and @@rowcount at the same time?
Q34 What is Identity?
Q35 What is a Scheduled Jobs or What is a Scheduled Tasks?
Q36 What is a table called, if it does have neither Cluster nor Non cluster Index? What is it used for?
Q37 What is BCP? When does it used?
Q38 How do you load large data to the SQL server database?
Q39 Can we rewrite subqueries into simple select statements or with joins?
Q40 Can SQL Servers linked to other servers like Oracle?
Q41 How to know which index a table is using?
Q42 How to copy the tables, schema and views from one SQL server to another?
Q43 What is Self Join?
Q44 What is Cross Join?
Q45 Which virtual table does a trigger use?
Q46 What is Data Warehousing?
Q47 What is OLTP (OnLine Transaction Processing)?
Q48 How do SQL server 2000 and XML linked? Can XML be used to access data?
Q49 What is RDBMS?
Q50 What is normalization?
Q51 What are different normalization forms?
Q52 What is Stored Procedure?
Q53 What is Trigger?
Q54 What is View?
Q55 What is Index?
Q56 What is de-normalization and when would you go for it?
Q57 How do you implement one-to-one, one-to-many and many-to-many relationships while designing tables?
Q58 What's the difference between a primary key and a unique key?
Q59 What are user defined data types and when you should go for them?
Q60 What is bit data type and what's the information that can be stored inside a bit column?
Q61 Define candidate key, alternate key, composite key.
bound?
Q63 What is a transaction and what are ACID properties?
Q64 Explain different isolation levels.
Q65 What type of Index will get created after executing the above statement?
Q66 What is the maximum size of a row?
Q67 What is Lock Escalation?
Q68 What's the difference between DELETE TABLE and TRUNCATE TABLE

Commands?
Q69 Explain the storage models of OLAP.
Q70 What are constraints? Explain different types of constraints.

Q71 What is an index? What are the types of indexes? How many clustered indexes can be created on a table? I create a separate index on each column of a table. what are the advantages and disadvantages of this approach?

Q72 What is RAID and what are different types of RAID configurations?

Q73 What are the steps you will take to improve performance of a poor performing query?

Q74 What are the steps you will take, if you are tasked with securing an SQL Server?

Q75 What is a deadlock and what is a live lock? How will you go about resolving deadlocks?

Q76 What is blocking and how would you troubleshoot it?

Q77 Explain CREATE DATABASE syntax.

Q78 How to restart SQL Server in single user mode? How to start SQL Server in minimal configuration mode?

Q79 As a part of your job, what are the DBCC commands that you commonly use for database maintenance?

Q80 What are statistics, under what circumstances they go out of date, how do you update them?

Q81 What are the different ways of moving data/databases between servers and databases in SQL Server?

Q82 Explain different types of BACKUPs available in SQL Server? Given a particular scenario, how would you go about choosing a backup plan?

Q83 What is database replication? What are the different types of replication you can set up in SQL Server?

Q84 How to determine the service pack currently installed on SQL Server?

Q85 What are cursors? Explain different types of cursors. What are the disadvantages of cursors? How can you avoid cursors?

Q86 What is a join and explain different types of joins?

Q87 What is a self join? Explain it with an example.

Q88 Can you have a nested transaction?

Q89 What is an extended stored procedure? Can you instantiate a COM object by using T-SQL?

Q90 What is the system function to get the current user's user id?

Q91 What are triggers? How many triggers you can have on a table? How to invoke a trigger on demand?

Q92 What do you think of this implementation? Can this be implemented better?

Question 1: What are the difference between clustered and a non-clustered index?

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

A **non clustered** index is a special type of index in which the logical order of the index does not match the physical stored order of the rows on disk. The leaf node of a non clustered index does not consist of the data pages. Instead, the leaf nodes contain index rows.

Question 2: What are the different index configurations a table can have?

Answer: A table can have one of the following index configurations: **No indexes**.

A **clustered index** .A clustered index and many non clustered indexes.

A **non clustered index**, Many non clustered indexes.

Question 3: What are cursors?

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

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

- Declare cursor
- Open cursor
- Fetch row from the cursor
- Process fetched row
- Close cursor
- Deallocate cursor

Question 4: 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.

E.g. DBCC CHECKDB – Ensures that tables in the db and the indexes are correctly linked.

DBCC CHECKALLOC– To check that all pages in a db are correctly allocated.

DBCC CHECKFILEGROUP – Checks all tables file group for any damage.

Question 5: What is a Linked Server?

Answer: Linked Servers is a concept in SQL Server by which we can add other SQL Server to a Group and query both the SQL Server dbs using T-SQL Statements. With a linked server, you can create very clean, easy to follow, SQL statements that allow remote data to be retrieved, joined and combined with local data. Storped Procedure sp_addlinkedserver, sp_addlinkedsrvlogin will be used add new Linked Server.

Question 6: What is Collation?

Answer: 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 using rules that define the correct character sequence, with options for specifying case-sensitivity, accent marks, kana character types and character width.

Question 7: What are different types of Collation Sensitivity?

Answer: Case sensitivity A and a, B and b, etc. Accent sensitivity a and á, o and ó, etc. Kana Sensitivity When Japanese kana characters Hiragana and Katakana are treated differently, it is called Kana sensitive. Width sensitivity when a single-byte character (half-width) and the same character when represented as a double-byte character (full-width) are treated differently than it is width sensitive.

Question 8: What's the difference between a primary key and a unique key?

Answer: Both primary key and unique enforce uniqueness of the column on which they are defined. But by default primary key creates a clustered index on the column, where are unique creates a non clustered index by default. Another major difference is that, primary key doesn't allow NULLs, but unique key allows one NULL only.

Question 9: How to implement one-to-one, one-to-many and many-to many relationships while designing tables?

Answer: One-to-One relationship can be implemented as a single table and rarely as two tables with primary and foreign key relationships. One-to-Many relationships are implemented by splitting the data into two tables with primary key and foreign key relationships. Many-to-Many relationships are implemented using a junction table with the keys from both the tables forming the composite primary key of the junction table.

Question 10: What is a NOLOCK?

Answer: Using the NOLOCK query optimizer hint is generally considered good practice in order to improve concurrency on a busy system. When the NOLOCK hint is included in a SELECT statement, no locks are taken when data is read. The result is a Dirty Read, which means that another process could be updating the data at the exact time you are reading it. There are no guarantees that your query will retrieve the most recent data.

The advantage to performance is that your reading of data will not block updates from taking place, and updates will not block your reading of data. SELECT statements take Shared (Read) locks. This means that multiple SELECT statements are allowed simultaneous access, but other processes are blocked from modifying the data. The updates will queue until all the reads have completed, and reads requested after the update will wait for the updates to complete. The result to your system is delay(blocking).

Question 11: What is difference between DELETE & TRUNCATE commands?

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

TRUNCATE

TRUNCATE is faster and uses fewer system and transaction log resources than **DELETE.**

TRUNCATE removes the data by deallocating the data pages used to store the table's data, and only the page deallocations are recorded in the transaction log.

TRUNCATE removes all rows from a table, but the table structure and its columns, constraints, indexes and so on remain. The counter used by an identity for new rows is reset to the seed for the column.

You cannot use TRUNCATE TABLE on a table referenced by a FOREIGN KEY constraint.

Because TRUNCATE TABLE is not logged, it cannot activate a trigger.

TRUNCATE cannot be Rolled back using logs. TRUNCATE is DDL Command. TRUNCATE Resets identity of the table.

DELETE

DELETE removes rows one at a time and records an entry in the transaction log for each deleted row.

If you want to retain the identity counter, use DELETE instead. If you want to remove table definition and its data, use the DROP TABLE statement.

DELETE Can be used with or without a WHERE clause. DELETE Activates Triggers.

DELETE can be rolled back using logs. DELETE is DML Command.
DELETE does not reset identity of the table.

Question 12: Difference between Function and Stored Procedure?

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

Question 13: When is the use of UPDATE_STATISTICS command?

Answer: This command is basically used when a large processing of data has occurred. If a large amount of deletions any modification or Bulk Copy into the tables has occurred, it has to update the indexes to take these changes into account. UPDATE_STATISTICS updates the indexes on these tables accordingly.

Question 14: What types of Joins are possible with Sql Server?

Answer: Joins are used in queries to explain how different tables are related. Joins also let you select data from a table depending upon data from another table.

Types of joins: INNER JOINs, OUTER JOINs, CROSS JOINs. OUTER JOINs are further classified as LEFT OUTER JOINS, RIGHT OUTER JOINS and FULL OUTER JOINS.

Question 15: What is the difference between a HAVING CLAUSE and a WHERE CLAUSE?

Answer: Specifies a search condition for a group or an aggregate.

HAVING CLAUSE can be used only with the SELECT statement. HAVING is typically used in a GROUP BY clause. When GROUP BY is not used, HAVING behaves like a WHERE clause. Having Clause is basically used only with the GROUP BY function in a query.

WHERE CLAUSE is applied to each row before they are part of the GROUP BY function in a query. HAVING criteria is applied after the the grouping of rows has occurred.

Question 16: What is sub-query? Explain properties of sub-query.

Answer: Sub-queries are often referred to as sub-selects, as they allow a SELECT statement to be executed arbitrarily Share this: Email reedit Share Like Be the first to like this post. Within the body of another SQL statement. A sub-query is executed by enclosing it in a set of parentheses.

Sub-queries are generally used to return a single row as an atomic value, though they may be used to compare values against multiple rows with the IN keyword.

A subquery is a SELECT statement that is nested within another T-SQL statement. A subquery SELECT statement if executed independently of the T-SQL statement, in which it is nested, will return a result set.

Meaning a subquery SELECT statement can standalone and is not depended on the statement in which it is nested. A subquery SELECT statement can return any number of values, and can be found in, the column list of a SELECT statement, a FROM, GROUP BY, HAVING, and/or ORDER BY clauses of a T-SQL statement. A Subquery can also be used as a parameter to a function call. Basically a subquery can be used anywhere an expression can be used. Properties of Sub-Query A subquery must be enclosed in the parenthesis.

A subquery must be put in the right hand of the comparison operator, and A subquery cannot contain a ORDER-BY clause. A query can contain more than one sub-queries.

Question 17: What are types of sub-queries?

Answer: Single-row subquery, where the subquery returns only one row.

Multiple-row subquery, where the subquery returns multiple rows, and

Multiple column subquery, where the subquery returns multiple columns.

Question 18: What is SQL Profiler?

Answer: SQL Profiler is a graphical tool that allows system administrators to monitor events in an instance of Microsoft SQL Server. You can capture and save data about each event to a file or SQL Server table to analyze later. For example, you can monitor a production environment to see which stored procedures are hampering performances by executing too slowly. Use SQL Profiler to monitor only the events in which you are interested. If traces are becoming too large, you can filter them based on the information you want, so that only a subset of the event data is collected. Monitoring too many events adds overhead to the server and the monitoring process and can cause the trace file or trace table to grow very large, especially when the monitoring process takes place over a long period of time.

Question 18.1: What is User Defined Functions?

Answer: User-Defined Functions allow to define its own T-SQL functions that can accept 0 or more parameters and return a single scalar data value or a table data type.

Question 19: What kind of User-Defined Functions can be created?

Answer: There are three types of User-Defined functions in SQL Server 2000 and they are Scalar, Inline Table-Valued and Multi-statement Table-valued. Scalar User- Defined Function.

A Scalar user-defined function returns one of the scalar data types. Text, ntext, image and timestamp data types are not supported. These are the type of user-defined functions that most developers are used to in other programming languages. You pass in 0 to many parameters and you get a return value.

Question 20: Inline Table-Value User-Defined Function.

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

Question 21: Multi-statement Table-Value User-Defined Function.

Answer: A Multi-Statement Table-Value user-defined function returns a table and is also an exceptional alternative to a view as the function can support multiple T-SQL statements to build the final result where the view is limited to a single SELECT statement. Also, the ability to pass parameters into a T-SQL **select** command or a group of them gives us the capability to in essence create a parameterized, non-updateable view of the data in the underlying tables. Within the create function command you must define the table structure that is being returned. After creating this type of user-defined function, It can be used in the FROM clause of a T-SQL command unlike the behavior found when using a stored procedure which can also return record sets.

Question 22: Which TCP/IP port does SQL Server run on? How can it be changed?

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

Question 23: What are the authentication modes in SQL Server? How can it be changed?

Answer: Windows mode and mixed mode (SQL & Windows). To change authentication mode in SQL Server click Start, Programs, Microsoft SQL Server and click SQL Enterprise Manager to run SQL Enterprise Manager from the Microsoft SQL Server program group. Select the server then from the Tools menu select SQL Server Configuration Properties, and choose the Security page.

Question 24: Where are SQL server user's names and passwords are stored in sql server?

Answer: They get stored in master db in the sysxlogins table.

Question 25: Which command using Query Analyzer will give you the version of SQL server and operating system?

Answer: SELECT SERVERPROPERTY('productversion'), SERVERPROPERTY ('productlevel'),

Question 26: What is SQL server agent?

Answer: SQL Server agent plays an important role in the day-to-day tasks of a database administrator (DBA). It is often overlooked as one of the main tools for SQL Server management. Its purpose is to ease the implementation of tasks for the DBA, with its full-function scheduling engine, which allows you to schedule your own jobs and scripts. Can a stored procedure call itself or recursive stored procedure?

How many level SP nesting possible? Yes. Because Transact-SQL supports recursion, you can write stored procedures that call themselves. Recursion can be defined as a method of problem solving wherein the solution is arrived at by repetitively applying it to subsets of the problem.

A common application of recursive logic is to perform numeric Share this: computations that lend themselves to repetitive evaluation by the same processing steps. Stored procedures are nested when one stored procedure calls another or executes managed code by referencing a CLR routine, type, or aggregate. You can nest stored procedures and managed code references up to 32 levels.

Question 27: What is @@ERROR?

Answer: The @@ERROR automatic variable returns the error code of the last Transact-SQL statement. If there was no error, @@ERROR returns zero. Because @@ERROR is reset after each Transact-SQL statement, it must be saved to a variable if it is needed to process it further after checking it.

Question 28: What is Raiserror?

Answer: Stored procedures report errors to client applications via the RAISERROR command. RAISERROR doesn't change the flow of a procedure; it merely displays an error message, sets the @@ERROR automatic variable, and optionally writes the message to the SQL Server error log and the NT application event log.

Question 29: What is log shipping?

Answer: Log shipping is the process of automating the backup of database and transaction log files on a production SQL server, and then restoring them onto a standby server. Enterprise Editions only supports log shipping. In log shipping the transactional log file from one server is automatically updated into the backup database on the other server. If one server fails, the other server will have the same db can be used this as the Disaster Recovery plan. The key feature of log shipping is that is will automatically backup transaction logs throughout the day and automatically restores them on the standby server at defined interval.

Question 30: What is the difference between a local and a global variable?

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

A global temporary table remains in the database permanently, but the rows exist only within a given connection. When connections are closed, the data in the global temporary table disappears. However, the table definition remains with the database for access when database is opened next time.

Question 31: What are the properties of the Relational tables?

Answer: Relational tables have six properties: 1. Values are atomic. 2. Column values are of the same kind. 3. Each row is unique. 4. The sequence of columns is insignificant. 5. The sequence of rows is insignificant. 6. Each column must have a unique name. 7.

Question 32: What is De-normalization?

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

Question 33: How to get @@error and @@rowcount at the same time?

Answer: If **@@Rowcount** is checked after Error checking statement then it will have 0 as the value of **@@Recordcount** as it would have been reset. And if @@Recordcount is checked before the error-checking statement then @@Error would get reset.

To get @@error and @@rowcount at the same time do both in same statement and store them in local variable.

SELECT @RC = @@ROWCOUNT, @ER = @@ERROR

Question 34: What is Identity?

Answer: Identity (or AutoNumber) is a column that automatically generates numeric values. A start and increment value can be set, but most DBA leave these at 1. A GUID column also generates numbers, the value of this cannot be controlled. Identity/GUID columns do not need to be indexed.

Question 35: What is a Scheduled Jobs or What is a Scheduled Tasks?

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

Question 36: What is a table called, if it does have neither Cluster nor Non cluster Index? What is it used for?

Answer: Unindexed table or Heap. Microsoft Press Books and Book On Line (BOL) refers it as Heap.

A heap is a table that does not have a clustered index and, therefore, the pages are not linked by pointers. The IAM pages are the only structures that link the pages in a table together. Unindexed tables are good for fast storing of data. Many times it is better to drop all indexes from table and than do bulk of inserts and to restore those indexes after that.

Question 37: What is BCP? When does it used?

Answer: BulkCopy is a tool used to copy huge amount of data from tables and views. BCP does not copy the structures same as source to destination.

Question 38: How do you load large data to the SQL server database?

Answer: Bulk Copy is a tool used to copy huge amount of data from tables. BULK INSERT command helps to Imports a data file into a database table or view in a user-specified format.

Question 39: Can we rewrite subqueries into simple select statements or with joins?

Answer: Subqueries can often be re-written to use a standard outer join, resulting in faster performance. As we may know, an outer join uses the plus sign (+) operator to tell the database to return all non-matching rows with NULL values. Hence we combine the outer join with a NULL test in the WHERE clause to reproduce the result set without using a sub-query.

Question 40: Can SQL Servers linked to other servers like Oracle?

Answer: SQL Server can be lined to any server provided it has OLE-DB provider from Microsoft to allow a link. E.g. Oracle has a OLE-DB provider for oracle that Microsoft provides to add it as linked server to SQL Server group.

Question 41: How to know which index a table is using?

Answer: SELECT table_name,index_name FROM user_constraints

Question 42: How to copy the tables, schema and views from one SQL server to another?

Answer: Microsoft SQL Server 2000 Data Transformation Services (DTS) is a set of graphical tools and programmable objects that lets user extract, transform, and consolidate data from disparate sources into single or multiple destinations.

Question 43: What is Self Join?

Answer: This is a particular case when one table joins to itself, with one or two aliases to avoid confusion. A self join can be of any type, as long as the joined tables are the same. A self join is rather unique in that it involves a relationship with only one table. The common example is when company have a hierarchal reporting structure whereby one member of staff reports to another.

Question 44: What is Cross Join?

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

Question 45: Which virtual table does a trigger use?

Answer: Inserted and Deleted.

List few advantages of Stored Procedure. Stored procedure can reduced network traffic and latency, boosting application performance.

Stored procedure execution plans can be reused, staying cached in SQL Server’s memory, reducing server overhead. Stored procedures help promote code reuse.

Stored procedures can encapsulate logic. You can change stored procedure code without affecting clients. Stored procedures provide better security to your data.

Question 46: What is Data Warehousing?

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

Question 47: What is OLTP (OnLine Transaction Processing)?

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

Question 48: How do SQL server 2000 and XML linked? Can XML be used to access data?

Answer: FOR XML (ROW, AUTO, EXPLICIT)

You can execute SQL queries against existing relational databases to return results as XML rather than standard rowsets. These queries can be executed directly or from within stored procedures. To retrieve XML results, use the FOR XML clause of the SELECT statement and specify an XML mode of RAW, AUTO, or EXPLICIT.

Question 49: What is RDBMS?

Answer: Relational Data Base Management Systems (RDBMS) are database management systems that maintain data records and indices in tables. Relationships may be created and maintained across and among the data and tables. In a relational database, relationships between data items are expressed by means of tables. Interdependencies among these tables are expressed by data values rather than by pointers. This allows a high degree of data independence. An RDBMS has the capability to recombine the data items from different files, providing powerful tools for data usage.

Question 50: What is normalization?

Answer: Database normalization is a data design and organization process applied to data structures based on rules that help build relational databases. 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.

Question 51: What are different normalization forms?

Answer: **1NF** : Eliminate Repeating Groups Make a separate table for each set of related attributes, and give each table a primary key. Each field contains at most one value from its attribute domain.

2NF : Eliminate Redundant Data If an attribute depends on only part of a multi-valued key, remove it to a separate table.

3NF : Eliminate Columns Not Dependent On Key If attributes do not contribute to a description of the key, remove them to a separate table. All attributes must be directly dependent on the primary key

BCNF : Boyce-Codd Normal Form If there are non-trivial dependencies between candidate key attributes, separate them out into distinct tables.

4NF : Isolate Independent Multiple Relationships

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

5NF : Isolate Semantically Related Multiple Relationships

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

ONF : Optimal Normal Form

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

DKNF : Domain-Key Normal Form

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

Question 52: What is Stored Procedure?

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

Question 53: What is Trigger?

Answer: A trigger is a SQL procedure that initiates an action when an event (INSERT, DELETE or UPDATE) occurs.

Triggers are stored in and managed by the DBMS. Triggers are used to maintain the referential integrity of data by changing the data in a systematic fashion. A trigger cannot be called or executed; the DBMS automatically fires the trigger as a result of a data modification to the associated table. Triggers can be viewed as similar to stored procedures in that both consist of procedural logic that is stored at the database level. Stored procedures, however, are not event-drive and are not attached to a specific table as triggers are. Stored procedures are explicitly executed by invoking a CALL to the procedure while triggers are implicitly executed. In addition, triggers can also execute stored procedures. Nested Trigger: A trigger can also contain INSERT, UPDATE and DELETE logic within itself, so when the trigger is fired because of data modification it can also cause another data modification, thereby firing another trigger. A trigger that contains data modification logic within itself is called a nested trigger.

Question 54: What is View?

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

Question 55: What is Index?

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

Table scans are sometimes unavoidable, but on large tables, scans have a terrific impact on performance.

Clustered indexes define the physical sorting of a database table's rows in the storage media. For this reason, each database table may have only one clustered index.

Non-clustered indexes are created outside of the database table and contain a sorted list of references to the table itself.

Question 56: What is de-normalization and when would you go for it?

Answer: As the name indicates, de-normalization is the reverse process of normalization. It is the controlled introduction of redundancy in to the database design. It helps improve the query performance as the number of joins could be reduced.

Question 57: How do you implement one-to-one, one-to-many and many-to-many relationships while designing tables?

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

Many-to-Many relationships are implemented using a junction table with the keys from both the tables forming the composite primary key of the junction table. It will be a good idea to read up a database designing fundamentals text book.

Question 58: What's the difference between a primary key and a unique key?

Answer: Both primary key and unique enforce uniqueness of the column on which they are defined. But by default primary key creates a clustered index on the column, where are unique creates a non-clustered index by default. Another major difference is that, primary key does not allow NULLs, but unique key allows one NULL only.

Question 59: What are user defined data types and when you should go for them?

Answer: User defined data types let you extend the base SQL Server data types by providing a descriptive name, and format to the database. Take for example, in your database, there is a column called Flight_Num which appears in many tables. In all these tables it should be varchar(8). In this case you could create a user defined data type Home Code Samples Learn Freelance Projects Online Testing Geek Talk Job Postings Geek Blogs Complex Data Management Big Database Management Object Oriented Database called Flight_num_type of varchar(8) and use it across all your tables. See sp_addtype, sp_droptype in books online.

Question 60: What is bit data type and what's the information that can be stored inside a bit column?

Answer: Bit data type is used to store Boolean information like 1 or 0 (true or false). Until SQL Server 6.5 bit data type could hold either a 1 or 0 and there was no support for NULL. But from SQL Server 7.0 onwards, bit data type can represent a third state, which is NULL.

Question 61: Define candidate key, alternate key, composite key.

Answer: A candidate key is one that can identify each row of a table uniquely. Generally a candidate key becomes the primary key of the table. If the table has more than one candidate key, one of them will become the primary key and the rest are called alternate keys.

A key formed by combining at least two or more columns is called composite key.

Question 62: What are defaults? Is there a column to which a default cannot be bound?

Answer: A default is a value that will be used by a column, if no value is supplied to that column while inserting data.

IDENTITY columns and timestamp columns can't have defaults bound to them. See CREATE DEFAULT in books online.

Question 63: What is a transaction and what are ACID properties?

Answer: A transaction is a logical unit of work in which, all the steps must be performed or none.

ACID stands for Atomicity, Consistency, Isolation, Durability. These are the properties of a transaction.

Question 64: Explain different isolation levels.

Answer: An isolation level determines the degree of isolation of data between concurrent transactions. The default SQL Server isolation level is Read Committed. Here are the other isolation levels (in the ascending order of isolation): Read Uncommitted, Read Committed, Repeatable Read, Serializable. Be sure to read about SET TRANSACTION ISOLATION LEVEL, which lets you customize the isolation level at the connection level.

CREATE INDEX myIndex ON myTable (myColumn)

Question 65: What type of Index will get created after executing the above statement?

Answer: Non-clustered index. Important thing to note: By default a clustered index gets created on the primary key, unless specified otherwise.

Question 66: What is the maximum size of a row?

Answer: 8060 bytes. Explain Active/Active and Active/Passive cluster configurations Hopefully you have experience setting up cluster servers. But if you do not, at least be familiar with the way clustering works and the two clustering configurations Active/Active and Active/Passive. SQL Server books online has enough information on this topic and there is a good white paper available on Microsoft site. Explain the architecture of SQL Server.

Question 67: What is Lock Escalation?

Answer: Lock escalation is the process of converting a lot of low level locks (like row locks, page locks) into higher level locks (like table locks). Every lock is a memory structure too many locks would mean, more memory being occupied by locks. To prevent this from happening, SQL Server escalates the many fine-grain locks to fewer coarse-grain locks. Lock escalation threshold was definable in SQL Server 6.5, but from SQL Server 7.0 onwards it's dynamically managed by SQL Server.

Question 68: What's the difference between DELETE TABLE and TRUNCATE TABLE commands?

Answer: DELETE TABLE is a logged operation, so the deletion of each row gets logged in the transaction log, which makes it slow. TRUNCATE TABLE also deletes all the rows in a table, but it will not log the deletion of each row, instead it logs the de-allocation of the data pages of the table, which makes it faster. Of course, TRUNCATE TABLE can be rolled back.

Question 69: Explain the storage models of OLAP.

Answer: Check out MOLAP, ROLAP and HOLAP in SQL Server books online for more information.

Question 70: What are constraints? Explain different types of constraints.

Answer: Constraints enable the RDBMS enforce the integrity of the database automatically, without needing you to create triggers, rule or defaults.

Types of constraints: NOT NULL, CHECK, UNIQUE, PRIMARY KEY, FOREIGN KEY

For an explanation of these constraints see books online for the pages titled:

"Constraints" and "CREATE TABLE", "ALTER TABLE"

Question 71: What is an index? What are the types of indexes? How many clustered indexes can be created on a table? I create a separate index on each column of a table. what are the advantages and disadvantages of this approach?

Answer: Indexes in SQL Server are similar to the indexes in books. They help SQL Server retrieve the data quicker.

Indexes are of two types. Clustered indexes and non-clustered indexes. When you create a clustered index on a table, all the rows in the table are stored in the order of the clustered index key. So, there can be only one clustered index per table. Non-clustered indexes have their own storage separate from the table data storage.

Non-clustered indexes are stored as B-tree structures (so do clustered indexes), with the leaf level nodes having the index key and its row locater.

The row located could be the RID or the Clustered index key, depending up on the absence or presence of clustered index on the table.

If you create an index on each column of a table, it improves the query performance, as the query optimizer can choose from all the existing indexes to come up with an efficient execution plan. At the same time, data modification operations (such as INSERT, UPDATE, DELETE) will become slow, as every time data changes in the table, all the indexes need to be updated. Another disadvantage is that, indexes need disk space, the more indexes you have, more disk space is used.

Question 72: What is RAID and what are different types of RAID configurations?

Answer: 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. MSDN has some information about RAID levels and for detailed information, check out the RAID advisory board's homepage

Question 73: What are the steps you will take to improve performance of a poor performing query?

Answer: 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.

Download the white paper on performance tuning SQL Server from Microsoft website.

Question 74: What are the steps you will take, if you are tasked with securing an SQL Server?

Answer: Again this is another open ended question. Here are some things you could talk about: Preferring NT authentication, using server, database and application roles to control access to the data, securing the physical database files using NTFS permissions, using an un guessable SA password, restricting physical access to the SQL Server, renaming the Administrator account on the SQL Server computer, disabling the Guest account, enabling auditing, using multi-protocol encryption, setting up SSL, setting up firewalls, isolating SQL Server from the web server etc.

Question 75: What is a deadlock and what is a live lock? How will you go about resolving deadlocks?

Answer: 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 76: What is blocking and how would you troubleshoot it?

Answer: 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 77: Explain CREATE DATABASE syntax.

Answer: Many of us are used to creating databases from the Enterprise Manager or by just issuing the command:

CREATE DATABASE MyDB.

But what if you have to create a database with two file groups, one on drive C and the other on drive D with log on drive E with an initial size of 600 MB and with a growth factor of 15%? That's why being a DBA you should be familiar with the CREATE DATABASE syntax. Check out SQL Server books online for more information.

Question 78: How to restart SQL Server in single user mode? How to start SQL Server in minimal configuration mode?

Answer: 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. Check out SQL Server books online for more parameters and their explanations.

Question 79: As a part of your job, what are the DBCC commands that you commonly use for database maintenance?

Answer: 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 80: What are statistics, under what circumstances they go out of date, how do you update them?

Answer: Statistics determine the selectivity of the indexes. If an indexed column has unique values then the selectivity of that index is more, as opposed to an index with non-unique values. Query optimizer uses these indexes in determining whether to choose an index or not while executing a query.

Some situations under which you should update statistics:

If there is significant change in the key

1. values in the index

2.If a large amount of data in an indexed column has been added, changed, or removed (that is, if the distribution of key values has changed), or the table has been truncated using the TRUNCATE TABLE statement and then repopulated.

3. Database is upgraded from a previous version Look up SQL Server books online for the following commands:

UPDATE STATISTICS,

STATS_DATE,

DBCC SHOW_STATISTICS,

CREATE STATISTICS,

DROP STATISTICS,

sp_autostats,

sp_createstats,

sp_updatestats

Question 81: What are the different ways of moving data/databases between servers and databases in SQL Server?

Answer: There are lots of options available, you have to choose your option depending upon your requirements.

Some of the options you have are:

BACKUP/RESTORE,

Detaching and attaching databases,

Replication,

DTS,

BCP,

logshipping,

INSERT...SELECT,

SELECT...INTO,

creating INSERT scripts to generate data.

Question 82: Explain different types of BACKUPs available in SQL Server? Given a particular scenario, how would you go about choosing a backup plan?

Answer: Types of backups you can create in SQL Sever 7.0+ are Full database backup, differential database backup, transaction log backup, file group backup. Be prepared to write the commands in your interview. Books online also has information on detailed backup/restore architecture and when one should go for a particular kind of backup.

Question 83: What is database replication? What are the different types of replication you can set up in SQL Server?

Answer: Replication is the process of copying/moving data between databases on the

same or different servers.

SQL Server supports the following types of replication scenarios:

* Snapshot replication

* Transactional replication (with immediate updating subscribers, with queued updating subscribers)

* Merge replication

See SQL Server books online for in-depth coverage on replication. Be prepared to explain how different replication agents function, what are the main system tables used in replication etc.

Question 84: How to determine the service pack currently installed on SQL Server?

Answer: The global variable @@Version stores the build number of the sqlservr.exe, which is used to determine the service pack installed. To know more about this process visit SQL Server service packs and versions.

Question 85: What are cursors? Explain different types of cursors. What are the disadvantages of cursors? How can you avoid cursors?

Answer: Cursors allow row-by-row processing of the resultsets.

Types of cursors:

Static,

Dynamic,

Forward-only,

Keyset-driven.

See books online for more information.

Disadvantages of cursors: Each time you fetch a row from the cursor, it results in a network roundtrip, where as a normal SELECT query makes only one round trip, however large the resultset is. Cursors are also costly because they require more resources and temporary storage (results in more IO operations). Further, there are restrictions on the SELECT statements that can be used with some types of cursors.

Most of the times, set based operations can be used instead of cursors. Here is an example:

If you have to give a flat hike to your employees using the following criteria:

Salary between 30000 and 40000 -- 5000 hike

Salary between 40000 and 55000 -- 7000 hike

Salary between 55000 and 65000 -- 9000 hike

In this situation many developers tend to use a cursor, determine each employee's salary and update his salary according to the above formula. But the same can be achieved by multiple update statements or can be combined in a single UPDATE statement as shown below:

UPDATE tbl_emp SET salary =

CASE WHEN salary BETWEEN 30000 AND 40000 THEN salary + 5000

WHEN salary BETWEEN 40000 AND 55000 THEN salary + 7000

WHEN salary BETWEEN 55000 AND 65000 THEN salary + 10000

END

Another situation in which developers tend to use cursors: You need to call a stored procedure when a column in a particular row meets certain condition.

You don't have to use cursors for this. This can be achieved using WHILE loop, as long as there is a unique key to identify each row.

Write down the general syntax for a SELECT statements covering all the options.

Here's the basic syntax: (Also checkout SELECT in books online for advanced syntax).

SELECT select_list

[INTO new_table_]

FROM table_source

[WHERE search_condition]

[GROUP BY group_by__expression]

[HAVING search_condition]

[ORDER BY order__expression [ASC | DESC]]

Question 86: What is a join and explain different types of joins?

Answer: Joins are used in queries to explain how different tables are related. Joins also let you select data from a table depending upon data from another table.

Types of joins:

- INNER JOINs,
- OUTER JOINs,
- CROSS JOINs
- OUTER JOINs are further classified as
- LEFT OUTER JOINS,
- RIGHT OUTER JOINS and
- FULL OUTER JOINS.

For more information see pages from books online titled: "Join Fundamentals" and "Using Joins".

Question 87: What is a self join? Explain it with an example.

Answer: Self join is just like any other join, except that two instances of the same table will be joined in the query. Here is an example: Employees table which contains rows for normal employees as well as managers. So, to find out the managers of all the employees, you need a self join.

CREATE TABLE emp

(

empid int,

mgrid int,

empname char(10)

)

INSERT emp SELECT 1,2,'Vyas'

INSERT emp SELECT 2,3,'Mohan'

INSERT emp SELECT 3,NULL,'Shobha'

INSERT emp SELECT 4,2,'Shridhar'

INSERT emp SELECT 5,2,'Sourabh'

SELECT t1.empname [Employee], t2.empname [Manager]

FROM emp t1, emp t2

WHERE t1.mgrid = t2.empid

Here is an advanced query using a LEFT OUTER JOIN that even returns the

employees without managers (super bosses)

SELECT t1.empname [Employee], COALESCE(t2.empname, 'No manager')

[Manager]

FROM emp t1 LEFT OUTER JOIN emp t2 ON t1.mgrid = t2.empid

Question 88: Can you have a nested transaction?

Answer: Yes, very much. Check out BEGIN TRAN, COMMIT, ROLLBACK, SAVE TRAN and @@TRANCOUNT

Question 89: What is an extended stored procedure? Can you instantiate a COM object by using T-SQL?

Answer: An extended stored procedure is a function within a DLL (written in a programming language like C, C++ using Open Data Services (ODS) API) that can be called from T-SQL, just the way we call normal stored procedures using the EXEC statement. See books online to learn how to create extended stored procedures and how to add them to SQL Server.

Yes, you can instantiate a COM (written in languages like VB, VC++) object from T-SQL by using sp_OACreate stored procedure. Also see books online for sp_OAMethod, sp_OAGetProperty, sp_OASetProperty, sp_OADestroy.

Question 90: What is the system function to get the current user's user id?

Answer: USER_ID(). Also check out other system functions like

USER_NAME(),

SYSTEM_USER,

SESSION_USER,

CURRENT_USER,

USER,

SUSER_SID(),

HOST_NAME().

Question 91: What are triggers? How many triggers you can have on a table? How to invoke a trigger on demand?

Answer: 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_settriggerorderb

Triggers cannot 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.

There is a trigger defined for INSERT operations on a table, in an OLTP system. The trigger is written to instantiate a COM object and pass the newly inserted rows to it for some custom processing.

Question 92: What do you think of this implementation? Can this be implemented better?

Answer: Instantiating COM objects is a time consuming process and since you are doing it from within a trigger, it slows down the data insertion process.

Same is the case with sending emails from triggers. This scenario can be better implemented by logging all the necessary data into a separate table, and have a job which periodically checks this table and does the needful.