

---

**Title: SQL Interview Questions**

---

**1.) What are the types of triggers?**

**Answer:** Triggers are special types of stored procedures that are defined to execute automatically in place of or after data modifications. They can be executed automatically on the INSERT, DELETE and UPDATE triggering actions. Different types of triggers are:

- After Trigger
- Multiple After Triggers
- Instead Of Triggers
- Mixing Triggers Type

**2.) Types of locks in database?**

**Answer:** There are different types of lock in SQL Server 2000 and 2005. These locks are applied in different situations. Here is the list of locks and the situation for the locks.

**SHARED** - This lock is applied for read operation where the data is not updated. A good example would be the select statement.

**UPDATE** - This locked on those resources that can be updated. This lock prevents the common form of dead lock that occurs when multiple sessions are locking the data so that they can update it later.

**EXCLUSIVE** - Used for data-modification operations, such as INSERT, UPDATE, or DELETE. Ensures that multiple updates cannot be made to the same resource at the same time.

**INTENT** - Used to establish a lock hierarchy. The different types of intent locks are: intent shared, intent exclusive, and shared with intent exclusive.

**SCHEMA** - Used when an operation is dependent on the schema of a table is executing. The different types of schema locks are: schema modification and schema

stability.

**BULK UPDATE** – This lock is applied when there is a bulk copying of data and the TABLOCK is applied

**KEY RANGE** - Protects the range of rows read by a query when using the serializable transaction isolation level. Ensures that other transactions cannot insert rows that would qualify for the queries of the serializable transaction if the queries were run again.

### 3.) Types of indexes. What is the default key created when a primary key is created in a table?

**Answer:** There are 2 types of Indexes:

- Cluster Index (Only One per table)
- Non-Cluster Index (249 per table)

By default a clustered index gets created on the primary key, unless specified otherwise

### 4.) What is clustered, non-clustered and unique index. How many indexes can be created on a table?

**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 nonclustered index does not consist of the data pages. Instead, the leaf nodes contain index rows. We can have 249 non-clustered indexes per table.

### 5.) What are the different types of backups?

**Answer:** A database may be backed up to disk or to tape. SQL Server provides several different kinds of backups including **Complete, Differential, Transaction**

**Log**, and **File(s) and Filegroup(s)** backup. A combination of these backups may be used to formulate a robust disaster recovery strategy.

### **6.) What is difference between triggers and stored procedures. And advantages of SP over triggers?**

**Answer: Triggers** are special types of stored procedures that are defined to execute automatically in place of or after data modifications. They can be executed automatically on the INSERT, DELETE and UPDATE triggering actions.

A **Stored Procedure** is a precompiled object **stored** in the database, and run as a unit to solve a specific problem or perform a set of related tasks. **Procedures** permit the caller to provide parameters that can be input only, output only, or input and output values. **Procedures** let you combine the ease and flexibility of SQL with the **procedural** functionality of a structured programming language.

#### **Differences:-**

- Trigger in action which is performed automatically before or after an event occurs whereas stored procedure is executed when it is called. Stored procedure is a precompiled object.
- When you create a trigger you have to identify event and action of your trigger but when you create a SP you don't have to identify event and action.
- Triggers are implicitly called by the DB Engine while SP has to be manually called by user.
- SP support parameters whereas triggers don't.

### **7.) What is @@ROWCOUNT used for?**

**Answer:** @@ROWCOUNT returns the number of rows that were selected, inserted or updated by the previous statement.

Eg.1.) `SELECT TOP 5 * FROM Table1`

`PRINT @@ROWCOUNT`

will print 5

Eg.2.) `SELECT TOP 5 * FROM Table1`

`PRINT 'Hello'`

`PRINT @@ROWCOUNT`

will print 0, since PRINT statement doesn't return rows.

### **8.) Explain one-to-one, one-to-many and many-to-many relationships?**

**Answer:** One-to-one relationship can be implemented as a single table and rarely as two tables with primary key 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.

### **9.) What is de-normalization? When would you use it?**

**Answer:** 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.

### **10.) Explain database normalization and normal forms briefly.**

**Answer:** Database normalization is the process of efficiently organizing data in a database. It is kind of a technique for designing relational database tables to minimize the duplication of information. **There are two purposes of this process:**

- Eliminating redundant data (for example, storing the same data in more than one table)
- Ensuring that data dependencies should make sense (only storing related data in a table)

Both of these purposes are very important since they reduce the amount of space that the database consumes and ensure that the data is logically stored.

### **The Normal Forms**

The database community has developed a series of guidelines to ensure that databases are normalized. These are referred to as "normal forms" and are numbered from one through five. In practical applications, first four of them are generally applicable. They are not rules, just guidelines. Therefore, they can be altered depending on the business requirements.

#### **First Normal Form (1NF)**

- Eliminate duplicative columns from the same table.
- Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).

#### **Second Normal Form (2NF)**

- Meet all the requirements of the first normal form.
- Remove subsets of data that apply to multiple rows of a table and place them in separate tables.
- Create relationships between these new tables and their predecessors through the use of foreign keys.

#### **Third Normal Form (3NF)**

- Meet all the requirements of the second normal form.
- Remove columns that are not dependent upon the primary key.

#### **Fourth Normal Form (4NF)**

- Meet all the requirements of the third normal form.
- A relation is in 4NF if it has no multi-valued dependencies.

### **11.) Define candidate key, alternate key and 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.

**12.) What is 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 unique creates a nonclustered index by default. Another major difference is that, primary key doesn't allow NULL, but unique key allows one NULL only.

**13.) How would you determine whether the service pack currently installed?**

**Answer:** @@Version stores the build number of the sqlservr.exe, which is used to determine the service pack installed.

**14.) What is a self join? Explain with an example.**

**Answer:** A self-join compares values within a single table. The important elements of a self-join are:

- Listing the table twice in the FROM clause and assigning a different alias to each.
- Joining the table to itself, using the table aliases to qualify the columns
- Adding a condition, often an unequal join on another column to prevent a row joining itself

**To find out the managers of all the employees, you need a self join such as:**

```
CREATE TABLE Employees (EmpID int, MgrID int, EmpName char(10))
```

```
INSERT Employees SELECT 1, 2, 'Smith'
```

```
INSERT Employees SELECT 2, 3, 'Dani'
```

```
INSERT Employees SELECT 3, NULL, 'Adam'
```

```
INSERT Employees SELECT 4, 2, 'Charles'
```

```
INSERT Employees SELECT 5, 2, 'Kate'
```

```
SELECT t1.EmpName [Employee], t2.EmpName [Manager]
```

FROM Employees t1, Employees t2

WHERE t1.MgrID = t2.EmpID

### 15.) What is SQL injection?

**Answer:** SQL injection is a technique used to take advantage of non-validated input vulnerabilities to pass SQL commands through a web application for execution by a backend database. Attackers take advantage of the fact that programmers often chain together SQL commands with user-provided parameters, and can therefore embed SQL commands inside these parameters. The result is that the attacker can execute arbitrary SQL queries and/or commands on the backend database server through the web application.

### 16.) What is heap?

**Answer:** Heap is a table without any index. Data is not stored in any particular order. Specific data can not be retrieved quickly, unless there are also non-clustered indexes.

### 17.) What is the difference between SCOPE\_IDENTITY and @@IDENTITY?

**Answer:** @@IDENTITY returns the most recently added identity for the current connection, not necessarily the identity for the recently added row in a table. Assume that there is a trigger that inserts a new record to a Logs table when the INSERT statement inserts a record to the Orders table. @@IDENTITY would yield the identity of the record added to the Logs table, not the Orders table.

**SCOPE\_IDENTITY** can be used to get the identity of the recently added row in the INSERT statement in order to avoid the possible conflicts if someone would add a trigger at a later time.

### 18.) What is Relational Database Management System? Explain.

**Answer:** RDBMS stands for **Relational Database Management System**. It is the basis for SQL and modern databases such as SQL Server, Oracle, IBM DB2, Sybase,

MySQL, etc.

A short definition of a RDBMS may be a DBMS (Database Management System) in which data is stored in the form of tables and the relationship among the data is also stored in the form of tables.

### **19.) What is a foreign key?**

**Answer:** Foreign key is a constraint that applies to a column that refers to the primary key of another table.

### **20.) Write a SQL statement that gives the 5th highest Rank from the Student Results table.**

**Answer:** SELECT TOP 1 Rank FROM (SELECT DISTINCT TOP 5 Rank FROM Results ORDER BY Rank DESC) ORDER BY Rank

### **21.) What is the Referential Integrity?**

**Answer:** Referential integrity refers to the consistency that must be maintained between primary and foreign keys, i.e. every foreign key value must have a corresponding primary key value.

### **22.) What are the differences between DELETE, TRUNCATE and DROP?**

**Answer:** The **DELETE** command is used to remove rows from a table. This is a logged activity. A **WHERE** clause is used to remove selective rows. If no WHERE condition is specified, all rows will be removed. After performing a DELETE operation you need to COMMIT or ROLLBACK the transaction to make the change permanent or to undo it. Note that this operation will cause all DELETE triggers on the table to fire. **TRUNCATE** removes all rows from a table. The operation cannot be rolled back and no triggers will be fired. As such, TRUCATE is faster and doesn't use as much undo space as a DELETE. This activity is not logged and hence it is faster.

The **DROP** command removes a table from the database. All the tables' rows, indexes and privileges will also be removed. No DML triggers will be fired. The



operation cannot be rolled back.

DROP and TRUNCATE are DDL commands, whereas DELETE is a DML command.

Therefore DELETE operations can be rolled back, while DROP and TRUNCATE operations cannot be rolled back.

### **23.) What is the difference between UNION and UNION ALL?**

**Answer:** UNION command combines the results of two SQL queries into a single table of all matching rows. When using UNION, only distinct values are selected like in SELECT DISTINCT.

UNION ALL command is equal to the UNION command, except that it selects all values where duplicate values are not removed.

### **24.) What is the difference between UNION and JOIN?**

**Answer:** A JOIN selects columns from 2 or more tables whereas UNION selects rows.

### **25.) Explain the ACID rule of thumb for transactions.**

**Answer:** A transaction must be:-

**Atomic** - It is one unit of work and does not depend on previous and following transactions.

**Consistent** - Data is either committed or roll back, no in-between case where something has been updated and something has not been.

**Isolated** - No transaction sees the intermediate results of the current transaction.

**Durable** - The values persist if the data had been committed even if the system crashes right after.

### **26.) What is the default port for SQL Server?**

**Answer:** Port 1433 is the default port for SQL Server. If SQL Server is listening on this default port of 1433, it is not required to specify the port number in the connection string.

**27.) What is the row size in SQL Server 2000?**

**Answer:** 8060 Bytes.

**28.) Where are the usernames and passwords stored in SQL Server?**

**Answer:** They are stored in the "sysxlogins" table of "master" db.

**29.) How would you add or subtract days in a date?**

**Answer:** `SELECT DATEADD(DD, -3, GETDATE()) AS [ThreeDaysAgo]`

`SELECT DATEADD(DD, 3, GETDATE()) AS [ThreeDaysFromToday]`

D or DAY can be used instead of DD inside the function.

**30.) How would you reference a table from another database?**

**Answer:** To reference a table from another database in the same server, simply prefix the table with the database name. In other words, use the 3-part naming convention of the table.

To illustrate, to select from the [dbo].[Authors] table in the pubs database from the Northwind database, the SELECT statement will be as follows:

`SELECT * FROM [pubs].[dbo].[Authors]`

**31.) What is a user defined data type in SQL Server?**

**Answer:** **User defined data types** provide a mechanism for applying a name to a data type that is more descriptive of the types of values to be held in the object.

Using user defined data types can make it easier for developers to understand the intended use of any object defined with the data type.

The **user defined data types** are based on the system data types, and can be used to predefine several attributes of a column, such as its data type, length, and whether it supports NULL values. To create a user defined data type, you can use the "**sp\_addtype**" system stored procedure or you could add one using the Enterprise Manager or Management Studio.

**32.) Explain char, nchar, varchar, nvarchar, text , ntext, binary and varbinary.**

**Answer: char:** Fixed-length character data with a maximum length of 8000 characters

**varchar:** Variable-length data with a maximum of 8000 characters

**text:** Variable-length data with a maximum length of  $2^{31} - 1$  characters

**nchar:** Fixed-length Unicode data with a maximum length of 4000 characters

**nvarchar:** Variable-length Unicode data with a maximum length of 4000 characters

**ntext:** Variable-length Unicode data with a maximum length of  $2^{30} - 1$  characters

**binary:** Fixed-length binary data with a maximum length of 8000 bytes

**varbinary:** Variable-length binary data with a maximum length of 8000 bytes

The char/varchar value uses only 1 byte to store one character whereas the nchar/nvarchar value uses 2 bytes to store one character. So, the char/varchar columns use two times less space to store data in comparison with nchar/nvarchar columns.

Using varchar/nvarchar columns instead of text/ntext columns whenever possible would be a good practice since SQL Server stores text/ntext columns on the Text/Image pages separately from the other data which are stored on the data pages. It can take more time to get the text/ntext values.

**33.) Explain briefly how SQL Server stores data.**

**Answer:** SQL Server stores data in a special structure called data pages that are 8Kb (8192 bytes) in size. Some space on the data pages is used to store system information, which leaves 8060 bytes to store user's data. So, if the table's row size is 4040 bytes, then only one row will be placed on each data page. If you can decrease the row size to 4030 bytes, you can store two rows within a single page. The less space used, the smaller the table and index, and the less the SQL Server has to perform when reading data pages from disk. So, you should design your

tables in such a way as to maximize the number of rows that can fit into one data page. To maximize the number of rows that can fit into one data page, you should specify the narrowest columns you can. The narrower the columns, the less data that is stored, and the faster SQL Server is able to read and write data.

**34.) How would you determine if a table has a primary key?**

**Answer:** OBJECTPROPERTY metadata function can be used to check this. For eg;  
`IF OBJECTPROPERTY( OBJECT_ID( '[dbo].[Customers]' ), 'TableHasPrimaryKey' ) = 1  
PRINT '[dbo].[Customers] table has a primary key.'`

**35.) How would you identify duplicate records in a table?**

**Answer:** `SELECT [FirstName], [LastName] FROM [dbo].[Accounts]  
GROUP BY [FirstName], [LastName] HAVING COUNT(*) > 1`

**36.) How would you split a full name into first and last names?**

**Answer:** `DECLARE @FullName VARCHAR(100)  
SET @FullName = 'John Smith'  
SELECT SUBSTRING(@FullName, 1, CHARINDEX(' ', @FullName) - 1) AS  
[FirstName],  
SUBSTRING(@FullName, CHARINDEX(' ', @FullName) + 1, LEN(@FullName)) AS  
[LastName]`

Substring function is used to extract the required string. You can also use LEFT & RIGHT function for the same.

**37.) How would you insert the rows of a table into another table?**

**Answer:** `INSERT INTO [dbo].[Customers] ( [ID], [FirstName], [LastName] )  
SELECT [ID], [FirstName], [LastName] FROM [dbo].[NewCustomers]`

**38.) 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. For instance; DBCC CHECKDB ensures that tables and the indexes are correctly linked.

**39.) How would you display the date in MM/DD/YYYY format?**

**Answer:** CONVERT function can be used to display a date in the MM/DD/YYYY format.

Style parameter is 101 for this format.

```
SELECT CONVERT(VARCHAR(10), GETDATE(), 101) AS [MM/DD/YYYY]
```

The output is now a VARCHAR data type and not DATETIME data type.

**40.) What is a constraint? Explain briefly.**

**Answer:** A constraint is a property assigned to a column or the set of columns in a table that prevents certain types of inconsistent data values from being placed in the column(s). Constraints are used to enforce the data integrity. This ensures the accuracy and reliability of the data in the database.

**The following categories of the data integrity exist:**

- **Entity Integrity** ensures that there are no duplicate rows in a table.
- **Domain Integrity** enforces valid entries for a given column by restricting the type, the format, or the range of possible values.
- **Referential Integrity** ensures that rows cannot be deleted, which are used by other records (for example, corresponding data values between tables will be vital).
- **User-Defined Integrity** enforces some specific business rules that do not fall into entity, domain, or referential integrity categories.

Each of these categories of the data integrity can be enforced by the appropriate constraints. Microsoft SQL Server supports the following constraints:

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

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

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

A **CHECK** constraint is used to limit the values that can be placed in a column. The check constraints are used to enforce domain integrity.

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

#### **41.) What steps will you take to improve the performance of a slow Query?**

**Answer:** It 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.

#### **42.) What is a deadlock and what is a live lock? How will you resolve 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.

#### **43.) How to log messages to a text file on the server, from your stored procedures and SQL scripts using T-SQL?**

**Answer:** This stored procedure accepts a message, filename, overwrite mode as an input parameters and logs that message to the specified file on the server

```
CREATE PROC log_to_file
@msg VARCHAR(100),
@file VARCHAR(100),
@overwrite BIT = 0
AS
```

/\* This is to log messages to text files from stored procedures/triggers/sql scripts  
Input parameters: message, file name, overwrite flag (1 to overwrite, 0 to append to file, 0 is the default)

Example: EXEC log\_to\_file 'Error Occurred','C:\logfile.txt',0

Remarks: You should have permissions required through access file system through xp\_cmdshell  
\*/

```
BEGIN
SET NOCOUNT ON
DECLARE @execstr VARCHAR(255)
SET @execstr = RTRIM('echo ' + COALESCE(LTRIM(@msg),'-') + CASE WHEN
(@overwrite = 1) THEN ' > ' ELSE ' >> ' END + RTRIM(@file))
EXEC master..xp_cmdshell @execstr
SET NOCOUNT OFF
END
```

#### **44.) How to send messages/notifications to all logged in users in Windows LAN using T-SQL?**

**Answer:** This stored procedure accepts a message as an input parameter and sends that message to all the currently logged in SQL Server users using NET SEND.

```
CREATE PROC notify_users
(
```

```
@notification VARCHAR(100) = 'SQL Server reboot in 5 mins.'
)
AS
BEGIN

/*****
*****
Purpose: To send NET SEND messages to all the connected SQL Server users in an
NT Local Area Network
Input parameters: Message to be sent
*****
*****/

SET NOCOUNT ON

DECLARE @msg VARCHAR(250)
DECLARE @hostname sysname

SELECT @hostname= min(RTRIM(hostname))
FROM
  master.dbo.sysprocesses (NOLOCK)
WHERE
  hostname <> ''

WHILE @hostname is not null
  BEGIN
    set @msg='exec master.dbo.xp_cmdshell "net send ' + RTRIM(@hostname) + ' ' +
RTRIM(@notification) + ' "'
    EXEC (@msg)
    SELECT @hostname= min(RTRIM(hostname))
    FROM
      master.dbo.sysprocesses (NOLOCK)
    WHERE
      hostname <> ''
      and hostname > @hostname
    END

SET NOCOUNT OFF

END
```

**45.) 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.

**46.) 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.



**47.) 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 SQLSERVER.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.

**48.) 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.

**49.) What are statistics, under what circumstances they go out of date, and 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.

**50.) 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