



350 Interview Questions & Answers on .Net Framework, C#, SQL, OOPS, ASP.Net, WCF

By Karthikeyan Anbarasan

Copyright

This is a preliminary document and may be changed substantially. The information contained in this document represents the current view of the authors as of the date of publication. This EBook is for informational purposes only. WE MAKE NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of its authors.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this document are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

©2012 Karthikeyan Anbarasan, (www.F5debug.Net) All rights reserved.

I dedicate this eBook to my Parents and my Wife, who make it all worthwhile.

— *Karthikeyan Anbarasan, Microsoft MVP www.f5debug.net*

ABOUT THE AUTHOR



Karthikeyan Anbarasan, is a **Microsoft MVP** (Most Valuable Professional) in ASP.NET/IIS Architecture and a **Microsoft India Rockstar**. Recently he has also been awarded by Nokia as "**Nokia Developer Champion**" in India on Windows Phone. He is the Founder and Chief Editor of the www.f5debug.net website and has authored couple of books on Microsoft technologies. He has over 9 years of industry experience in design, development and architecting products for Enterprise.

He is passionate about Microsoft Technologies and has worked on most of the Microsoft product from its initial release. He is also a Speaker and a Blogger in most of the Microsoft communities in India. He has been certified by Microsoft and has been awarded by the largest community groups around the globe (**CSharpCorner MVP, DNF MVP, and DZone MVB**). In his leisure time he enjoys watching TV and spending time with family.

Author can be reached @

Facebook – fb.com/f5debug

Twitter – [@f5debug](https://twitter.com/f5debug)

Email – Karthikeyan@f5debug.net

ACKNOWLEDGEMENT

I was always been a big believer and advocate of free knowledge and education for all. To continue this belief, I have personally begun writing free distributable books for the community. Please feel free to share this book with your friends and co-workers.

Also, do not forget to share your knowledge and spread the word around about the blog www.F5Debug.net. I should also mention about my website www.f5debug.net, which has always inspired me to write more on .NET and related technologies.

Thanks to Mohit Chabra for consolidating, reviewing and checking the technical accuracy of the eBook.

A lot of thanks to my wife, for all her support and encouragement. Without her it would have been impossible in accomplishing this.

DISCLAIMER

The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this eBook. The strategies contained herein may not be suitable for every situation. Neither the publisher nor the author shall be liable for damages arising here from. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

Question 1 - What Is CLR?

CLR is Common Language Runtime is the runtime that converts a MSIL code into the host machine language code. It is the execution engine for .NET Framework applications. It provides a number of services, including:

- Code management (loading and execution)
- Memory Management
- Thread Management
- Conversion of IL to native code.
- Access to metadata (enhanced type information)
- Managing memory for managed objects (Garbage collection)
- Enforcement of code access security (Security Management)
- Exception handling, including cross-language exceptions
- Interoperation btw managed code, COM objects, and pre-existing DLL's (unmanaged code and data)
- Support for developer services (profiling, debugging, and so on).
- Type safety.

Question 2 - What is CLR HOST?

A CLR host is an application that is responsible for loading the CLR into a process, creating application domains within the process, and executing user code within the application domains.

Examples of hosts that ship with the .NET Framework include:

- ASP.Net - An ISAPI filter that ships with ASP.NET loads the CLR and does the initialization necessary to handle web requests.
- Internet Explorer: - A MIME filter hooks into IE versions 5.01 and higher to execute managed controls referenced from HTML pages.

Question 3 - What is CTS?

CTS is Common Type System, which describes how types are declared, used and managed. CTS facilitate cross-language integration, type safety, and high performance code execution.

Example in VB you have “Integer” and in C++ you have “long” these data types are not compatible so the interfacing between them is very complicated. In order that these two different languages communicate Microsoft introduced Common Type System. So “Integer” data type in VB and “int” data type in C++ will convert it to System.int32, which is data type of CTS.

Question 4 - What is CLS?

CLS Is Common Language Specification, is a specification that defines the rules to support language integration. This is done in such a way, that programs written in any language (.NET compliant) can interoperate with one another. This also can take full advantage of inheritance, polymorphism, exceptions, and other features. It was always a dream of Microsoft to unite all different languages in to one umbrella and CLS is one-step towards that.

Question 5 - What is an Intermediate Language?

(IL) Intermediate Language Or (CIL) Common Intermediate Language Or (MSIL) Microsoft Intermediate Language, is one of the Core component of the .NET Framework. Any .NET source codes written in any .net supportive language (C#,VB.net etc), when compiled are converted to MSIL. This MSIL, when installed or at the Runtime, gets converted to machine code. The Runtime conversion of MSIL code to the machine code is handled by a component called as the Just In Time (JIT) Compiler.

Question 6 - What is Just In Time Compiler?

It is a compiler which converts MS IL (Microsoft Intermediate Language) code to Native Code (i.e. CPU-specific code that runs on the same computer architecture). Just-In-Time compiler- it converts the language that you write in .Net into machine language that a computer can understand. There are 3 types of JITs

- Pre-JIT compiler (Compiles entire code into native code completely in a single Operation)
- Econo JIT compiler (Compiles only methods(Code) called at Runtime)
- Normal JIT compiler (Compiles only that part of code called at Runtime and places in cache)

Question 7 - What is Portable executable (PE)?

The file format defining the structure that all executable files (EXE) and Dynamic Link Libraries (DLL) must use to allow them to be loaded and executed by Windows. Windows PE is divided into 2 main sections.

- The first section includes the PE/COFF headers that references the contents within the PE file.
- The second section is the native image section which contains .data, .rdata, .rsrc, and .text sections.

Question 8 - What is Managed Code?

Managed code runs inside the environment of CLR i.e. .NET runtime. In short, all IL are managed code. However, if you are using some third party software example VB6 or VC++ component they are unmanaged code, as .NET runtime (CLR) does not have control over the source code execution of these languages. A runtime-aware compiler compiles the IL into native executable code within a managed execution environment that ensures type safety, array bound and index checking, exception handling, and garbage collection. Also, many unproductive programming tasks are automatically taken care of, such as type safety checking, memory management, and destruction of unneeded objects. You can therefore focus on the business logic of your applications and write them using fewer lines of code. The result is shorter development time and more secure and stable applications.

Question 9 - What is UnManaged Code?

As mentioned above the Unmanaged code will be one where

- The CLR cannot able to understand the code.
- The CLR cannot instruct the code.
- The second time compilation is unmanaged code. It is understood only by the machine not by the user.

Question 10 - What is Garbage Collector?

Garbage collection is a process of releasing the memory used by the objects, which are no longer referenced. This is done in different ways and different manners in various platforms and languages.

When an program is loaded in the memory there will be a bunch of memory allocated for that particular program alone and loaded with memory. This bunch of memory is called Managed Heap(is nothing but a bunch of memory allocated for the program at run time). This amount of memory will only be used when an object is to be loaded in to the memory for that particular program.

This memory is separated in to three parts:

- Generation Zero – Smaller size
- Generation One – Medium size
- Generation Two – Larger size

When we try to create an object by using NEW keyword the system will,

- Calculate the number of bytes required for the object or type to be loaded in to the managed heap.

- The CLR then checks that the bytes required to allocate the object are available in the reserved region. If the object fits, it is allocated at the address pointed to by NextObjPtr.
- These processes will happen at the Generation zero level.

Question 11 - What is a Strong Name?

A strong name is a .NET assembly name combined with its version number and other information to uniquely identify the assembly. This allows multiple versions of the same assembly to peacefully co-exist in the global assembly cache, where shared assemblies are typically stored. It consists of five parts as mentioned below

- **Simple Name** – Usually the name of the file (without the extension) that contains the assembly
- **Public Key** – RSA cryptographic public key that helps verify the assembly's authenticity
- **Version** – Four-part version number, in the form of Major.Minor.Build.Revision
- **Culture** – Target audience for the assembly, such as "neutral" (default audience), "en-us" (English – United States) or "fr" (France) etc.
- **Processor Architecture** – Defines the assembly's format, such as MSIL (intermediate language) or x86 (binary for Intel x86 processors)

Question 12 - What are the steps to create Strong Name?

We have a set of steps that should be followed to create a strong name as shown below.

- Open .net command prompt.
- Go to the folder containing DLL.
- Type `sn -k test.snk`, This will create test .snk file in that folder.
- Open the assemblyinfo.cs file of project.
- Type file path in this tag `[assembly:AssemblyKeyFile@"E:\Code\practice\HP\bin\Debug\HP.snk"]`
- Build application, finally your strong name created for your DLL.

Question 13 - What are the Problems faced using Strong Name?

- Requires Exact Match. If you use strong names, your application or library must load the assembly with the exact strong name that you specify, including version and culture.
- Cannot Lose Private Key. If your private key is lost or stolen, the security of your assembly is compromised. You will be forced to re-issue a new assembly signed with a new public-private key pair.

Question 14 - What is Program Database?

A program database file holds debugging and project state information that allows incremental linking of debug configuration of your program. A PDB file can be up to 2GB.

Question 15 - What is Delay Signing?

It is signing an assembly with its strong name public key, which is freely distributable, instead of using the private key as usual. This allows developers to use and test a strong-named assembly without access to the private key. Then at a later stage (typically just before shipping the assembly), a manager or trusted key holder must sign the assembly with the corresponding private key.

Question 16 - What is an Assembly?

Assemblies are the building blocks of .NET Framework applications; they form the fundamental unit of deployment, version control, reuse, activation scoping, and security permissions. An assembly is a collection of types and resources that are built to work together and form a logical unit of functionality. An assembly provides the common language runtime with the information it needs to be aware of type implementations.

Question 17 - What are the Contents of an Assembly

- Type metadata.
- The assembly manifest, which contains assembly metadata.
- Microsoft intermediate language (MSIL) code that implements the types.
- A set of resources

Question 18 - What are the Types of an Assemblies?

Assemblies are of four types

- Private - The assembly is intended only for one application
- Shared - If the assembly is to be made into a Shared Assembly(GAC)
- Static – These are the .NET PE files that you create at compile time.
- Dynamic - These are PE-formatted, in-memory assemblies that you dynamically create at runtime

Question 19 - What is a Satellite assembly?

- A .NET Framework assembly containing resources specific to a given language.
- Using satellite assemblies, you can place the resources for different languages in different assemblies.
- The correct assembly is loaded into memory only if the user elects to view in that language.
- Culture is maintained in a text file which acts like a resource to the assembly.

Question 20 - What are the Steps to Create Satellite Assembly?

Below are the steps to create a satellite assembly

- Set the paths for resgen and al.exe:
- Create a .resources file.
- Create the satellite assembly.
- The assembly should have the naming convention for .NET to be able to search for it.
- Specify the settings for culture.
- Put the satellite assembly in the appropriate folder.
- Once the satellite assembly is created, physically copy it to the appropriate directory.
- Repeat the process for each language in which you are creating an assembly.

Question 21 - What is an Assembly Loader?

- Checks if the assembly is Strongly signed.
- If yes it will search in the GAC
- Loader will search the policy file name in the format of
- Policy.AssemblyMajorVersion.AssemblyMinorVersion.AssemblyName

- Eg. MyPolicy.1.2.Assembly1
- If such a file exists it will look inside of it if the version of the assembly that we are trying to load matches the version/versions range written in the policy file. If it does, it will try to load the assembly with the version specified there. If no such policy file exists, it will try to load assembly from the GAC.
- If it will fail to find it in the GAC, it will start to search in the system's search path.
In web applications it will also include the application's Bin directory in the search path.

Question 22 - What is Multi Module Assembly or Assembly Linker?

We like to combine Hello.dll with GoodBye.dll and put them into a Private Assembly we call GreetAssembly.dll.

```
DotNet> al /t:library /out:bin\Hellowroldasassembly.dll bin\GoodDay.dll bin\GoodMorning.dll
```

For this purpose we use the Assembly Linker. As /t (target) we generate here a library referencing the two other DLLs. This is also called a Multi-Module Assembly. Again, we store all the binaries in a bin folder.

Question 23 - What is an Assembly Manifest?

The assembly manifest contains this assembly metadata. An assembly manifest contains all the metadata needed to specify the assembly's version requirements and security identity, and all metadata needed to define the scope of the assembly and resolve references to resources and classes. The assembly manifest can be stored in either a PE file (an .exe or .dll) with Microsoft intermediate language (MSIL) code or in a standalone PE file that contains only assembly manifest information.

Question 24 - What is a Metadata?

Metadata is data that describes the state of the assembly and a detailed description of each type, attribute within the assembly. Metadata stores the following information:

- Description of the assembly.
 - Identity (name, version, culture, public key).
 - The types that are exported.
 - Other assemblies that this assembly depends on.
 - Security permissions needed to run.
- Description of types.

- Name, visibility, base class, and interfaces implemented.
- Members (methods, fields, properties, events, nested types).
- Attributes.
 - Additional descriptive elements that modify types and members.

Question 25 - What is a Base class in .Net?

Base class is the one from which the object and references are being inherited in .net

- System.object is for .Net
- System.Web.UI is for asp.net

Question 26 - What is Full Assembly Reference?

A full assembly reference includes the assembly's text name, version, culture, and public key token (if the assembly has a strong name). A full assembly reference is required if you reference any assembly that is part of the common language runtime or any assembly located in the global assembly cache.

Question 27 - What is Partial Assembly Reference?

We can dynamically reference an assembly by providing only partial information, such as specifying only the assembly name. When you specify a partial assembly reference, the runtime looks for the assembly only in the application directory.

Question 28 - What is an Assembly Qualified Name?

An assembly qualified name isn't the filename of the assembly; it's the internal name of the assembly combined with the assembly version, culture, and public key, thus making it unique.

Question 29 - What is ILDASM (Intermediate Language Disassembler)?

It is a tool provided in C# to view and read the assembly content in manifest view. This tool is supplied along with the Visual Studio .NET you are using. It is also available along with .NET SDK. To access this tool, you have to run the ildasm.exe

Question 30 - What is Global Assembly Cache?

- The GAC stores assemblies specifically designated to be shared by several applications on the computer.
- Assemblies deployed in the GAC must have a strong name.
- When an assembly is added to the GAC, integrity checks are performed on all files that make up the assembly. The cache performs these integrity checks to ensure that an assembly has not been tampered with, for example, when a file has changed but the manifest does not reflect the change.

Use a developer tool called the Global Assembly Cache tool (**Gacutil.exe**), provided by the .NET Framework SDK or Use Windows Explorer to drag assemblies into the cache.

Question 41 - What is Custom Serialization?

- In some cases, the default serialization techniques provided by .NET may not be sufficient in real life.
- This is when we require implementing custom serialization.
- It is possible to implement custom serialization in .NET by implementing the ISerializable interface.
- This interface allows an object to take control of its own serialization and de-serialization process.
- It gives us a great deal of flexibility in the way we can save and restore objects.

Question 42 - What is a Namespace?

- Containers of objects which contain classes, unions, structures, interfaces, enumerators, delegates.
- Main goal is for creating hierarchical organization of program.
- Developers does not need to worry about the naming conflicts of classes, functions, variables etc.

Question 43 - What is GUID?

It is Short form of Globally Unique Identifier,

A unique 128-bit number that is produced by the Windows OS or Windows app to identify a particular component, application, file, database entry, and/or user.

Question 44 - What is a Formatter?

- A formatter is used to determine the serialization format for objects.
- In other words, it is used to control the serialization of an object to and from a stream.

- They are the objects that are used to encode and serialize data into an appropriate format before they are transmitted over the network.
- They expose an interface called the IFormatter interface. IFormatter's significant methods are Serialize and De-serialize which perform the actual serialization and de-serialization.
- There are two formatter classes provided within .NET, the BinaryFormatter and the SoapFormatter. Both these classes extend the IFormatter interface.

Question 45 - What is a Binary Formatter?

The Binary formatter provides support for serialization using binary encoding. The BinaryFormatter class is responsible for binary serialization and is used commonly in .NET's Remoting technology. This class is not appropriate when the data is supposed to be transmitted through a firewall.

Question 46 - What is a SOAP Formatter?

The SOAP formatter provides formatting that can be used to serialize objects using the SOAP protocol. It is used to create a Soap envelop and it uses an object graph to generate the result. It is responsible for serializing objects into SOAP messages or parsing the SOAP messages and extracting these serialized objects from the SOAP messages. SOAP formatters in .NET are widely used by the Web Services.

Question 47 - What is Reflection?

- It is a collection of classes which allow u to query assembly (class/object) metadata at runtime.
- Using reflection we can also create new types and their instances at runtime and invoke methods on these new instances.
- At runtime, the Reflection mechanism uses the PE file to read information about the assembly.
- We can dynamically invoke methods using System.Type.InvokeMember
- We can dynamically create types at runtime using System.Reflection.Emit.TypeBuilder
- With reflection we can do the below
 - we can dynamically create an instance of a type
 - bind the type to an existing object
 - get the type from an existing object
 - invoke its methods or access its fields and properties

Question 48 - What is Thread and Process?

- Threads are basically light weight processes responsible for multitasking within a single application.
- The base class used for threading is System.Threading.
- Threads are implemented when situations in which you want to perform more than one task at a time.
 - A Process is an instance of a running application.
 - A thread is the Execution stream of the Process.
 - A process can have multiple Threads.
- **Example:** A Microsoft Word is an Application. When you open a word file, an instance of the Word starts and a process is allocated to this instance which has one thread.
 - **Create** Thread – use System.Thread() class and create an instance.
 - **Join** Thread - use object.Join() to join threads.
 - **Suspend** Thread - use object.Sleep(<No of seconds>) to suspend a thread.
 - **Kill** Thread - use object.Abort() to abort a thread.

Question 49 - What are the difference between a Dll and an Exe?

DLL	EXE
Create an object of dll	Not in exe
In Process Component	Out Process Component
Multiple Uses	Single use
Cannot be started as stand alone	Can be started as stand alone

Question 50 - What are Globalization and Localization?

To implementing a multilingual user interface, you design the user interface to open in the default UI language and offer the option to change to other languages. Globalization is the first step in the process. A globalized application supports localized user interfaces and regional data for all users. Truly global applications should be culture-neutral and language-neutral. A globalized application can correctly accept,

process, and display a worldwide assortment of scripts, data formats, and languages. Accommodating these cultural differences in an application is called localization.

Question 51 - What is a Resource File?

Resource files are the files containing data that is logically deployed with an application. These files can contain data in a number of formats including strings, images and persisted objects. It has the main advantage of If we store data in these files then we don't need to compile these if the data get changed. In .NET we basically require them storing culture specific information's by localizing application's resources. You can deploy your resources using satellite assemblies.

Question 52 - What is Code Access Security(CAS)?

CLR allows code to perform only those operations that the code has permission to perform. So CAS is the CLR's security system that enforces security policies by preventing unauthorized access to protected resources and operations. Using the Code Access Security, you can do the following:

- Restrict what your code can do
- Restrict which code can call your code
- Identify code

Code access security consists of the following elements:

- Permissions - represent access to a protected resource or the ability to perform a protected operation.
- Permission sets - A permission set is a collection of permissions.
- Code groups - logical grouping of code that has a specified condition for membership Evidence
- Policy - Security policy is the configurable set of rules that the CLR follows when determining the permissions to grant to code.
- There are four policy levels - Enterprise, Machine, User and Application Domain, each operating independently from each other.

Syntax

- See CAS objects -- Run '**caspol -lg**' from command line.
- Add CAS objects -- **caspol -ag** 1.3 -site www.mydomain.com FullTrust

- Change CAS obj -- **caspol -cg** 1.3 FullTrust
- Turn Off -- **caspol -s** off

Question 53 - What is difference between Code Based Security and Role Based Security?

CAS is the approach of using permissions and permission sets for a given code to run. Example, Admin can disable running executables off the Internet or restrict access to corporate database to only few applications.

- Role security most of the time involves the code running with the privileges of the current user. This way the code cannot supposedly do more harm than mess up a single user account.
- Neither is better. It depends on the nature of the application; both code-based and role-based security could be implemented to an extent.

Question 54 - What is difference between Invoke and Begin Invoke

- Delegate.Invoke: Executes synchronously, on the same thread.
- Delegate.BeginInvoke: Executes asynchronously, on a threadpool thread.
- Control.Invoke: Executes on the UI thread, but calling thread waits for completion before continuing.
- Control.BeginInvoke: Executes on the UI thread, and calling thread doesn't wait for completion.
- BeginInvoke is asynchronous. When BeginInvoke is called from the UI thread the request will be executed in parallel with the UI thread. Which means it may not execute until after the currently executing method has returned. So in this case the text box will never appear to update because the for loop will not be interrupted, as the calling thread will not wait for this event to be completed before continuing.
- Alternatively, Invoke is synchronous. The text box will be updated because the calling thread will wait for the call to complete before continuing execution.

Question 55 - What is the difference between Debug and Trace?

Tracing is actually the process of collecting information about the program's execution. Debugging is the process of finding & fixing errors in our program. Tracing is the ability of an application to generate information about its own execution.

Trace and Debug work in a similar way, the difference is that tracing from the Debug class only works in builds that have the DEBUG symbol defined, whereas tracing from the Trace class only works in builds that have the TRACE symbol defined.

- Use `System.Diagnostics.Trace.WriteLine` for tracing that you want to work in debug and release builds
- Use `System.Diagnostics.Debug.WriteLine` for tracing that you want to work only in debug builds.

Question 56 - What is a Debug version of a code?

- Preprocessor(Debugging Diagnostic) macro `_DEBUG` is enabled.
- More memory size.
- Support files required. (MFC Dll's)
- No Code Optimization
- Uses MFC Debug Library
- ASSERT is enabled.
- Execution takes more time

Question 57 - What is a Release version of a code?

- Preprocessor(Debugging Diagnostic) macro `NDEBUG` is enabled.
- Less memory size.
- Support files not required. (MFC Dll's)
- Code Optimization
- Uses MFC Release Library
- ASSERT is disabled anything inside of ASSERT will not be executed.
- Execution takes less time

Question 58 - What is an IDisposable Interface?

- The primary use of this interface is to release unmanaged resources. The garbage collector automatically releases the memory allocated to a managed object when that object is no longer used.
- However, it is not possible to predict when garbage collection will occur.

- Furthermore, the garbage collector has no knowledge of unmanaged resources such as window handles, or open files and streams.
- The consumer of an object can call this method when the object is no longer needed.

Question 59 - What is Finalize block in .net?

- Finalize() is called by the runtime
- Is a C# equivalent of destructor, called by Garbage Collector when the object goes out of scope.
- Implement it when you have unmanaged resources in your code, and want to make sure that these resources are freed when the Garbage collection happens.
- Finalize() can NOT be overridden or called in C#.
- Since, Finalize() is called by the Garbage Collector, it is non-deterministic.

Question 60 - What is Dispose block in .net?

- Dispose() is called by the user
- Same purpose as finalize, to free unmanaged resources. However, implement this when you are writing a custom class, that will be used by other users.
- Overriding Dispose() provides a way for user code to free the unmanaged objects in your custom class.
- Dispose() has to be implemented in classes implementing IDisposable interface.
- Dispose() method is called explicitly in the code itself.

Question 61 - What is Runtime Host?

Ranging from Windows applications, Web applications to Mobile applications, CLR is designed to support various types of applications. .NET Framework provides different types of runtime hosts to manage the execution of application code (to load runtime in to process, create application domain within process, load user code in to application domain) and provides various services to the application. Runtime hosts included in .Net framework are : ASP.NET, Microsoft Internet Explorer and windows shell.

Question 62 - What is Connection Pooling?

- A Connection Pool is a container of open and reusable connections. A Connection Pool is released from the memory when the last connection to the database is closed.
- The Data Providers in ADO.NET have Connection Pooling turned on by default; if you need to turn it off, specify Pooling = false in the connection string being used.
- Connection Pooling gives you an idle, open, reusable connection instead of opening a new one every time a connection request to the database is made. When the connection is closed or disposed, it is returned to the pool and remains idle until a request for a new connection comes in.
- The pool can house connections up to the maximum limit as specified in the connection string that was used to connect to the database.
- Advantage of using Connection Pooling is an improvement of performance and scalability
- Disadvantage is that one or more database connections, even if they are currently not used, are kept open.

Question 63 - What are the main parameters used by Connection Pooling?

Connection Pooling is controlled and the parameters passed to a connection string comprises the following:

- Connect Timeout
- Min Pool Size
- Max Pool Size
- Pooling

Question 64 - What is Connection Pool Manager?

A Connection Pool is maintained internally by the Connection Pool Manager. When a request for a subsequent connection comes in, the Connection Pool Manager searches the pool for the availability of a free connection and returns it to the application if one is available. Connection Pools works as below

- If any unused connection is available, it returns one.
- If all connections are used up, a new connection is created and added to the pool.
- If the number of connections reaches the maximum number of connections in the pool, the requests are queued until a connection becomes free for reuse.

Question 65 - What is Object Pooling?

It is something that tries to keep a pool of objects in memory to be re-used later and hence it will reduce the load of object creation to a great extent. Whenever there is a request for a new object, the pool manager will take the request and it will be served by allocating an object from the pool. Pooling basically means utilizing the resources efficiently, by limiting access of the objects to only the period the client requires it.

Question 66 - What are the Advantages of Object Pooling?

It minimizes the consumption of memory and the system's resources by recycling and re-using objects as and when it is needed and serving the request for new objects from the pool of ready-to-be-used objects. The objects that the application is done with (the objects are no longer needed) are sent back to the pool rather than destroying them from the memory.

According to MSDN, "Once an application is up and running, memory utilization is affected by the number and size of objects the system requires. Object pooling reduces the number of allocations, and therefore the number of garbage collections, required by an application.

Question 67 - What is the Difference between Connection Pooling and Object Pooling?

Object Pooling is great in the sense that it can optimize access to expensive resources (like file handles or network connections) by pooling them in memory and reusing them as and when they are needed.

According to MSDN, "Object pooling lets you control the number of connections you use, as opposed to connection pooling, where you control the maximum number reached."

Question 68 - What is an Indexer?

- Indexers permit instances of a class or struct to be indexed in the same way as arrays.
- Indexers are similar to properties except that their accessors take parameters.
- The indexers are usually known as smart arrays in C#.
- An indexer, also called an indexed property, is a class property that allows you to access a member variable of a class using the features of an array.
- Defining an indexer allows you to create classes that act like virtual arrays. Instances of that class can be accessed using the [] array access operator.

Question 69 - What are the important points to remember on indexers?

- Indexers are always created with this keyword.
- Parameterized property are called indexer.
- Indexers are implemented through get and set accessors for the [] operator.
- ref and out parameter modifiers are not permitted in indexer.
- Indexer is an instance member so can't be static but property can be static.
- Indexers are used on group of elements.
- Indexer can be overloaded.

Question 70 - What is the Difference between Indexers and Properties?

Indexers	Properties
Indexers are created with this keyword	Properties don't require this keyword
Indexers are identified by signature	Properties are identified by their names
Indexers are accessed using indexes	Properties are accessed by their names
Indexer are instance member, so can't be static	Properties can be static as well as instance members

Question 81 - What are the Types of Inheritance?

- **Implementation inheritance** refers to the ability to use a base class's properties and methods with no additional coding.
- **Interface inheritance** refers to the ability to use just the names of the properties and methods, but the child class must provide the implementation.
- **Visual inheritance** refers to the ability for a child form (class) to use the base forms (class) visual representation as well as the implemented code.

Question 82 - What is Multiple Inheritance?

C# does not support multiple implementation inheritance. A class cannot be derived from more than one class, however a class can be derived from multiple interfaces.

Question 83 - What are the examples of Multiple Inheritance?

Imagine a class named TransmitData, whose function is to transmit data, and another class named ReceiveData, whose function is to receive data. Now imagine that you want to create a class named SocketPort, whose function is to transmit and receive data. In order to accomplish this, you would want to derive SocketPort from both TransmitData and ReceiveData.

Question 84 - What are the Advantages of Inheritance?

- Once a behavior (method) or property is defined in a super class(base class), that behavior or property is automatically inherited by all subclasses (derived class).
- Code reusability increased through inheritance.
- Inheritance provide a clear model structure which is easy to understand without much complexity Using inheritance, classes become grouped together in a hierarchical tree structure Code are easy to manage and divided into parent and child classes.

Question 85 - What is an Encapsulation?

- Encapsulation is a process of hiding all the internal details of an object from the outside world.
- Encapsulation is the ability to hide its data and methods from outside the world and only expose data and methods that are required
- Encapsulation gives us maintainability, flexibility and extensibility to our code.
- Encapsulation makes implementation inaccessible to other parts of the program and protect from whatever actions might be taken outside the function or class.
- Encapsulation provides a way to protect data from accidental corruption
- Encapsulation hides information within an object
- Encapsulation is technique or process of making fields in a class private and providing access to the fields using public methods

- Encapsulation allows us to create a "black box" and protects an objects internal state from corruption by its clients.
- The idea of encapsulation comes from the need to cleanly distinguish between the specification and the implementation of an operation and the need for modularity.

Question 86 - What are the examples of Encapsulation?

- Let's say you have an object named Bike and this object has a method named start(). When you create an instance of a Bike object and call its start() method you are not worried about what happens to accomplish this, you just want to make sure the state of the bike is changed to 'running' afterwards. This kind of behavior hiding is encapsulation and it makes programming much easier.
- Video Recorder, which has a record, play, pause buttons is another example of encapsulation, so VCR is encapsulated into a single object where the internals can change but stays the same for users interface point of view.
- Medical Capsules i.e. one drug is stored in bottom layer and another drug is stored in Upper layer these two layers are combined in single capsule.

Question 87 - What is an Abstraction?

- Abstraction means to show only the necessary details to the client of the object.
- Abstraction is about paying attention to the details that are relevant and ignoring the rest.
- It refers to act of representing essential features without including background details / explanations.

Question 88 - What are the examples of Abstraction?

- Do you know the inner details of the Monitor of your PC? What happen when you switch ON Monitor? No Right, Important thing for you is weather Monitor is ON or NOT.
- When you change the gear of your vehicle are you really concern about the inner details of your vehicle engine? No but what matter to you is that Gear must get changed that's it!!
- Let's say you have a method "CalculateSalary" in your Employee class, which takes EmployeeId as parameter and returns the salary of the employee for the current month as an integer value. Now if someone wants to use that method. He does not need to care about how Employee object calculates the

salary? An only thing he needs to be concern is name of the method, its input parameters and format of resulting member. This is abstraction; show only the details which matter to the user.

- TV Remote Button in that number format and power buttons and other buttons there just we are seeing the buttons, we don't see the button circuits .i.e buttons circuits and wirings all are hidden.

Question 89 - Difference between Encapsulation and Abstraction ?

Encapsulation	Abstraction
Hiding the internal details or mechanics of how an object does something.	It focus on what the object does instead of how it does it.
Binding data and member functions together inside a single unit.	Hiding the complexities of your type from outside world.
Eg: VCR Example	Eg: Monitor Example

Question 90 - What is an Abstract Class?

- It is a class that cannot be instantiated, it exists extensively for inheritance and it must be inherited.
- Abstract classes cannot be used to instantiate objects; because abstract classes are incomplete
- Abstract classes may contain only definition of the properties or methods.
- Derived classes that inherit the abstract class needs to implements it's properties or methods.
- An abstract class is essentially a blueprint for a class without any implementation.
- An abstract class is a class that must be inherited and have the methods overridden.
 - An abstract class cannot be a sealed class.
 - An abstract method cannot be private.
 - An abstract member cannot be static.
- An abstract method cannot have the modifier virtual. Because an abstract method is implicitly virtual.
- The access modifier of the abstract method should be same in both the abstract class and its derived class. If you declare an abstract method as protected, it should be protected in its derived class.

Otherwise, the compiler will raise an error.

Question 91 - What is an Interface?

- An interface looks like a class, but has no implementation.
- An interface is a named set of method signatures.
- An Interface is a reference type and it contains only abstract members.
- An interface is an array of related function that must be implemented in derived type.
- Members of an interface are implicitly public & abstract.
- It can contain definitions of events, indexers, methods parameter less and parameterful properties.
- The interface can't contain constants, data fields, constructors, destructors and static members.
- All the member declarations inside interface are implicitly public.
- Interfaces are great for putting together plug-n-play like architectures where components can be interchanged at will. Since all interchangeable components implement the same interface, they can be used without any extra programming.

Question 92 - What is a difference between Abstract Class and Interface?

Abstract Class	Interface
Cannot be instantiated, means one cannot make a object of this class.	We can only define method definition and no implementation.
Access modifiers are allowed	Access modifiers are not allowed
Some methods can be concrete	All methods are abstract
A class can inherit only one abstract class	A class can inherit many interface.
Can have any access modifiers	By default its public static final
Can have constructor and destructor	Cannot have constructor and destructor
Only one abstract class can be derived	Class can have multiple interfaces
Requires more time to find actual method in class	Faster
Abstract class provides 0 to 100% generalization	Interface provides 100% generalization
It provides both generalization and specialization	Interface provides only Generalization

Question 93 - What is a Constructor?

- Constructor is used to initialize an object (instance) of a class.
- Constructor is a like a method without any return type.

- Constructor has same name as class name.
- Constructor follows the access scope (Can be private, protected, public, Internal and external).
- Constructor can be overloaded, means we can have constructors with different set of parameters.
- We can always make the call to one constructor from within the other constructor.
- Only this and base keywords allowed in initializing constructors, other method calls will raise error.

Question 94 - What is a Constructor chaining?

Overloading the constructor using the this and base keywords so that it overload is called constructor chaining

Question 95 - What are the Types of constructors?

- Static Constructor
- Default Constructor
- Private Constructor
- Copy Constructor
- Parameterized Constructor

Question 96 - What is a Private Constructor?

- Used to prevent the user to instantiate the class directly.
- Used to prevent the creation of instances of a class when there are no instance fields or methods
- A private constructor is a special instance constructor.
- It is commonly used in classes that contain static members only.
- If a class has one or more private constructors and no public constructors, then other classes (except nested classes) are not allowed to create instances of this class.
- Note that if you don't use an access modifier with the constructor it will still be private by default.
- Private constructors are used to restrict the instantiation of object using 'new' operator.
- This type of constructors is mainly used for creating singleton object.
- Can use nested class (Inner Class) or static method to initialize a class having private constructor.
- Example of Private Constructor - Math class

Question 97 - What is a Static Constructors?

- Special constructor and gets called before the first object is created of the class.
- The time of execution cannot be determined, but it is definitely before the first object creation - could be at the time of loading the assembly.
- Static constructors might be convenient, but they are slow. The runtime is not smart enough to optimize them in the same way it can optimize inline assignments.
- The static constructor for a class executes before any of the static members for the class are referenced.
- The static constructor for a class executes after the static field initializers (if any) for the class.
- A static constructor cannot be called directly.
- The user has no control on when the static constructor is executed in the program.
- Example - When the class is using a log file and the constructor is used to write entries to this file.

Question 98 - What are the features of Static Constructor?

- **Only one Static constructor** - Overloading needs the two methods to be different in terms to methods definition, so you can have at the most one static constructor
- **Without parameters** - It is going to be called by CLR, who can pass the parameters to it, if required, No one, so we cannot have parameterized static constructor.
- **Access only static members** - If allowed to work on non-static members, will reflect the changes in all the object instances, which is impractical. Non-static members in the class are specific to the object instance
- **No access modifier** - The call to the static method is made by the CLR and not by the object, so we do not need to have the access modifier to it.

Question 99 - What is a Default Constructor?

- A default constructor is a constructor in both that has no parameters or where it has parameters they are all defaulted.
- If no constructor is supplied then the compiler will supply a default constructor.
- This default constructor is a parameter less constructor with no body, which calls the parameter less constructor of the base class.

Question 100 - What is a COPY Constructor?

- C# does not provide a copy constructor.
- A copy constructor is a special constructor used to create a new object as a copy of an existing object.
- This constructor takes a single argument: a reference to the object to be copied.
- It is a great convenience to create copy constructor for C# classes using Reflection.
- If you create a new object and want to copy the values from an existing object, you have to write the appropriate method yourself.

Question 101 - What is a Parameterized constructor?

Constructor that accepts arguments is known as parameterized constructor. There may be situations, where it is necessary to initialize various data members of different objects with different values when they are created. Parameterized constructors help in doing that task.

Question 102 - What is a Singleton Class?

- A singleton class is such kind of class in which only one object is created throughout the life time of the class.
- A Singleton class is used when you wish to restrict instantiation of a class to only one object.

Question 103 - What is a Partial Class?

- It is possible to split the definition of a class or a struct, or an interface over two or more source files
- Each source file contains a section of class definition, and all parts are combined at compile time.
- All the partial definitions must proceeded with the key word "Partial".
- All the partial types must be defined within a same assembly and module.
- Method signatures (return type, name of the method, and parameters) must be unique
- The partial types must have the same accessibility.
- If any part is sealed, then the entire class is sealed.
- If any part is abstract, the entire class is abstract.
- Inheritance at any partial type applies to the entire class.

Question 104 - What is a Partial Method?

- A partial method is like a usual method in a class except that the user may or may not implement it.
- A partial method gets executed only when it has an implementation.
- Definition of a partial method is in one part of the partial class and implementation in another, but it is legal to have both in the same part of the partial class. Also you can use a partial method in a partial structure but not in partial interface.
- Partial methods are indicated by the partial modifier.
- Partial methods must be private.
- Partial methods must return void.
- Partial methods must only be declared within partial classes.
- Partial methods do not always have an implementation.
- Partial methods can be static and generic.
- Partial methods can have arguments including ref but not out.
- You cannot make a delegate to a partial method.

Question 105 - What is a Delegate?

- Delegate in C# is similar to a function pointer in C or C++.
- Delegate is type which holds the method(s) reference in an object. It is also referred as a type safe function pointers.
- Delegate allows the programmer to encapsulate a reference to a method inside a delegate object.
- The delegate object can then be passed to code which can call the referenced method, without having to know at compile time which method will be invoked.
- An interesting and useful property of a delegate is that it does not know or care about the class of the object that it references.

Question 106 - What is a Syntax of Single class delegate?

- Syntax - delegate result-type identifier ([parameters]);
 - result-type: The result type, which matches the return type of the function.
 - identifier: The delegate name.
 - parameters: The Parameters, that the function takes.

Question 107 - What are the advantages of Delegates?

- Encapsulating the method's call from caller.
- Effective use of Delegate improves the performance of application.
- Used to call a method asynchronously.
- A delegate type maintains three important pieces of information :
 - The name of the method on which it make calls.
 - Any argument (if any) of this method.
 - The return value (if any) of this method.

Question 108 - What is a Multicast Delegate?

- Delegate wraps a method. Calling delegate results in calling the method. It is possible to wrap more than one method in a delegate. This is known as a multicast delegate.
- If you make a call to a multicast delegate it will call all the functions it wraps in the order specified. Please note that functions in this case should not return any values.

Question 109 - What is an Event?

- An event in C# is a way for a class to provide notifications to clients of that class when some interesting thing happens to an object.
- An event is a way for a class to allow clients to give it delegates to methods that should be called when the event occurs. The most familiar use for events is in graphical user interfaces.
- Event Handlers in the .NET Framework return void and take two parameters.
- The first parameter is the source of the event; that is the publishing object.
- The second parameter is an object derived from EventArgs.
- Events are properties of the class publishing the event.
- The keyword event controls how the event property is accessed by the subscribing classes.

Question 110 - What is a Hash Table?

- Stored in two dimensional array and have a link format of Key and Value.
- The Hashtable object contains items in key/value pairs. The keys are used as indexes.

- The data type of Hashtable is object and the default size of a Hashtable is 16.
- The keys are used as indexes. We can search value by using their corresponding key.
- Items are added to the Hashtable with the Add() method.
- A close friend of Hashtable is Array (came from C/C++), it is an instance of System.Array class.

Question 121 – What is a Shallow Copy?

A shallow copy means the contents (each array element) contains references to the same object as the elements in the original array.

Question 122 – What is a Deep Copy?

A deep copy (which neither of these methods performs) would create a new instance of each element's object, resulting in a different, yet identical object. Basically it copies the source objects as well. So that there will be two different copies of the sources and objects.

Question 123 – What are the different String Compare options available?

String Functions	Description
String.Compare	Compares the values of two strings. Returns an integer value
String.CompareOrdinal	Compares two strings without regard to local culture. Returns an integer value
String.CompareTo	Compares the current string object to another string. Returns an integer value
String.StartsWith	Determines whether a string begins with the string passed. Returns a Boolean value
String.EndsWith	Determines whether a string ends with the string passed. Returns a Boolean value
String.Equals	Determines whether two strings are the same. Returns a Boolean value
String.ReferenceEquals	Returns true if both objects point to the same location in memory

Question 124 – What is a Statics Class

- It is not possible to create instances of a static class using the new keyword.
- A class can be declared static, indicating that it contains only static members.

- If a class is declared as static then the variables and methods should compulsorily declared as static.
- Static classes are loaded automatically by the .NET Framework common language runtime (CLR) when the program or namespace containing the class is loaded.
- Static classes cannot contain a constructor, although it is still possible to declare a static constructor to assign initial values or set up some static state.

Question 125 – What are the advantages of using Static Class?

- The advantage of using a static class is that the compiler can check to make sure that no instance members are accidentally added.
- The compiler will guarantee that instances of this class cannot be created.

Question 126 – List some of the main features of a Static Class?

The main features of a static class are:

- They only contain static members.
- They cannot be instantiated.
- They are sealed.
- They cannot contain Instance Constructors

Question 127 – What is a Static Member?

- A static method, field, property, or event is callable on a class even when no instance of the class has been created.
- Static members are often used to represent data or calculations that do not change in response to object state; for instance, a math library might contain static methods for calculating sine and cosine.
- Static class members are declared using the STATIC keyword.
- Static members are preloaded in the memory.

Question 128 – What is a Static Variable?

- A static variable is a variable that will be the same in all instances of a class.
- The initial value of a static variable is the default value of the variable's type.
- A field declared with the static modifier is called a static variable. A static variable comes into existence before execution of the static constructor.
- Normally each class has its own copy of all variables. If you declare some as static, then they will have the same value in each instance of a class.

Question 129 – What is a Static Method?

- Static methods show have static keyword as the method definition
- Static methods are accessed without creating an instance of the class object
- Static methods will not have “this” functionality to access the method
- Static methods cannot access non static members of the class.

Question 130 – What is a Nested Class?

- A nested class is one that is created inside another class.
- Nested classes have access to the private members of the outer class. So a scenario where this is the right way would be when creating a Comparer (ie. implementing the IComparer interface).
- Nested classes are very useful for implementing internal details that should not be exposed.

Question 131 – What is Shadowing or Hiding?

- When global and local variable are in the same name, the local variable in a method which use to override the global is called the shadowing. ie the Global variable is being shadowed by local variable.
- When two elements in a program have the same name, one of them can hide and shadow the other one. So in such cases the element which shadowed the main element is referenced.
- This is a VB.Net Concept by which you can provide a new implementation for the base class member without overriding the member.

Hiding is the C# concept equivalent for shadowing.

- You can shadow a base class member in the derived class by using the keyword Shadows.

- Shadowing is bad programming practice according to OOPs concepts.
- In shadowing signature could be different.
- In Shadowing both Derived class and Base Class methods are available for use.(So it's a bad practice)
- The access level signature and the return type can only be changed when you are shadowing with VB.NET. Hiding and overriding demands the these parameters as same in C#.

Question 132 – What are Out and Ref parameters?

- The out parameter - return the values in the same variable passed as a parameter of the method. Any changes made to the parameter will be reflected in the variable.
- The ref keyword on a method parameter causes a method to refer to the same variable that was passed as an input parameter for the same method. If you do any changes to the variable, they will be reflected in the variable.
- The implementation of ref and out parameter in IL Code(in CLR) is same, there is no difference whether you use ref or out parameters.

Question 133 – What are the differences of Out and Ref Parameters?

REF	OUT
Ref must be initialized before it is passed to the method	Out its is not necessary, but after method call it should get initialized
Extra performance cost.	Faster
The called method can read the argument at anytime	The called method must initialize the parameterbefore reading it
ref parameters are used to get a value and return a change to that value(Secondary value)	out parameters are used when you just need to geta secondary return value

Question 134 – What are the differences between String and String Builder?

String	String Builder
String objects are immutable, which means that once they are created they cannot be changed.	StringBuilder are mutable class, which means whenconcatenate any text to the variable it allows

When we use one of the methods of the String class, we create a new string object.	us to modify the original value without creating a new object.
Here concatenation is used to combine two strings.	Here Append method is used.
String object is used to concatenate two strings.	StringBuilder object is used.
The first string is combined to the other string by creating a new copy in the memory as a string object, and then the old string is deleted	Insertion is done on the existing string.
Less efficient	StringBuilder is more efficient for large amounts of string manipulations

Question 135 – Why C# is strongly typed language?

C# is called Strongly Typed Language because its type rules are very strict. For example you can't call a function that is designed to call Integer with a string or decimal. If you want to do so then you will have to explicitly convert them to integer.

Question 136 – What are Imperative and Interrogative function?

- Imperative methods return values or provide information back to the calling code. It returns a value.
- Interrogative methods, just perform a service and return nothing to the calling code. It does not return a value.

Question 137 – What is a Collection Class?

- A collection is a set of the same type of objects that are grouped together and that is able to supply a reference to an enumerator. Part of the System.Collections or System.Collections.Generic namespace
- An enumerator is an object that iterates through its associated collection. It can be thought of as a movable pointer pointing to any element in the collection. In order to provide an enumerator, a class must implement the IEnumerable interface.
- Most collection classes derive from the interfaces ICollection, IComparer, IEnumerable, IList, IDictionary, and DictionaryEnumerator and their generic equivalents.
- Using generic collection classes provides increased type-safety and in some cases can provide better performance, especially when storing value types.

Question 138 – What are the differences between Const & Readonly?

Const	Read Only
Must be initialized at the time of its creation	Assigned in constructor and called at Runtime
Used if we want to define something at compile time.	At Runtime you can make use of the value
Protects from accidentally changing value of the field	
Can't be static.	Can be either instance-level or static.
Value is evaluated at compile time.	Value is evaluated at run time.
Initialized at declaration only.	Initialized in declaration or by code in the constructor

Question 139 – What is a Stack?

- Stack is responsible of keeping track of running memory needed in your application.
- As the name says stack it stacks this memory allocation on the top of the first memory allocation. You can think about stack as series of compartment or boxes put on top of each other.
- Memory allocation and de-allocation is done using LIFO (Last in first out) logic. In other words memory is allocated and de-allocated at only one end of the memory i.e. top of the stack.
- Reference pointers are allocated on stack.
- At the end event clears all the memory variables which are assigned on stack
- Static Memory – Stack
- Details stored in stack are Name, Data type and Value of the variable.

Question 140 – What is a Heap?

- It creates a pointer on the stack and the actual object is stored in a different type of memory location called as 'Heap'.
- 'Heap' does not track running memory it's just pile of objects which can be reached at any moment of time.
- Heap is used for dynamic memory allocation. FIFO ('First In First Out')
- Instance created using new keyword it will be in HEAP.
- It did not de-allocate the heap memory. This memory will be later de-allocated by "Garbage Collector".
- Dynamic Memory – Heap

Question 141 – What are Value Types?

- Value types are types which hold both data and the memory on the same location.
- When we assign the 'int' value to the other 'int' value it creates a complete different copy. These kinds of data types are called as 'Value types'.
- Value Types in .Net are
 - Double, Float, Integer types (sByte, byte, short, ushort, int, uint, long, ulong, char)
 - Decimal, Bool, Enumeration types

Question 142 – What are Reference Types?

- Reference type has a pointer which points to the memory location.
- When we create an object and when we assign one object to the other object, they both point to the same memory location. In other words if we change one of them the other object is also affected this is termed as 'Reference types'.
- Reference Types in .Net
 - String and Object

Question 143 – What are Boxing and UnBoxing means?

- Move a value type to reference type -- stack to the heap – BOXING
- Move reference type to a value type -- heap to the stack -- UNBOXING

Question 144 – What is Early Binding?

- Early binding is nothing but declaring the Object of specific type. With this kind of object its cant be used to hold any other type of the Object/class.
- Early bound just means the target method is found at compile time.
- If the method doesn't exist the compiler will fail to compile the code.
- Most script languages use late binding, and compiled languages use early binding.
- The compiler guarantees that the function takes the right number of arguments, correct type and return value is of the correct type.

Question 145 – What is Late Binding?

- Declaring an object of generic type and that object be used to hold the instance of the any object/Class.
- Late bound means the target method is looked up at run time.
- If the method isn't there, then program will crash or go to some exception handling scheme at run time.
- Most script languages use late binding, and compiled languages use early binding.
- The target function may not accept the arguments passed to it, and may have a return value of the wrong type.

Question 146 – What are the different WCF binding available?

- **BasicHttpBinding** Basic Web service communication. No security by default
- **WSHttpBinding** Web services with WS-* support. Supports transactions
- **WSDualHttpBinding** Web services with duplex contract and transaction support
- **WSFederationHttpBinding** Web services with federated security. Supports transactions
- **MsmqIntegrationBinding** Communication directly with MSMQ applications. Supports transactions
- **NetMsmqBinding** Communication between WCF apps by using queuing. Supports transactions
- **NetNamedPipeBinding** Communication between WCF apps on same computer. Supports duplex contracts and transactions
- **NetPeerTcpBinding** Communication between computers across peer-to-peer services. Supports duplex contracts
- **NetTcpBinding** Communication between WCF apps across computers. Supports duplex contracts and transactions

Question 147 – What is a BasicHttpBinding?

This binding is used when we need to use SOAP over HTTP.

Question 148 – What is a WSHttpBinding?

It is same like BasicHttpBinding. In short, it uses SOAP over HTTP. But with it also supports reliable message transfer, security and transaction.

Question 149 – What is a NetTcpBinding?

This binding sends binary-encoded SOAP, including support for reliable message transfer, security, and transactions, directly over TCP. The biggest disadvantage of NetTcpBinding is that both server and client should be also made in .NET language.

Question 150 – What is a WSDualHttpBinding?

- The WSDualHttpBinding class is used to provide a secure and interoperable binding.
- It is used with duplex service contracts to let both services and clients send/receive messages.
- It assists Web Service protocols as does the WSHttpBinding class but for duplex contracts.
- The WSDualHttpBinding class only supports SOAP security and requires reliable messaging.
 - Advantage: can get immediate response w/o waiting on polling timer
 - Disadvantage: less scalable than WsHttpBinding
 - Disadvantage: less firewall friendly
 - Disadvantage: slower than WSHttpBinding

Question 151 – What is an ASP.NET Application and Page Life Cycle?

ASP.NET creates an environment which can process the request. i.e. creates application object, request, response and context objects

Once environment is created, the request is processed through a series of events which is processed by using modules, handlers and page objects. MHPM (Module, handler, page and Module event)

Question 152 – What are the Steps for ASP.net environment creation?

Step 1 IIS checks which ISAPI extension can server and pass to respective. e.g. aspx page to aspnet_isapi.dll

Step 2 If first request then ApplicationManager creates a Application Domain.

Step 3 App Domain will create hosting environment (HttpRuntime) object, then core objects like HttpContext, HttpRequest and HttpResponse will be created.

Step 4 Once core objects created, HttpApplication is created, if global.aspx is available then object for that will be created. global.asax file inherits from 'HttpApplication' class

Step 5 The HttpApplication object is then assigned to the core ASP.NET objects(HttpContext, HttpRequest and HttpResponse) to process the page.

Step 6 HttpApplication then starts processing the request by the following format HTTP module events,

handlers and page events. (MHPM)

Step 7 (M: HttpModule): Client request processing starts. Used when we want to inject logic in events of ASP.NET pipeline. It is an event based processor. 6 important events we can utilize before page object is created BeginRequest, AuthenticateRequest, AuthorizeRequest, ResolveRequestCache, AcquireRequestState and PreRequestHandlerExecute.

Step 8 (H: 'HttpHandler'): Used when we want to inject logic based in file extensions like '.ASPX', '.HTML'. It is an extension based processor. Once above 6 events fired, ASP.NET engine will invoke ProcessRequest event.

Step 9 (P: ASP.NET page): ASP.NET page object will be created. There are 6 important events - Init, Load, validate, event, render and unload. SILVER to remember the events S – Start , I – (Init) , L (Load) , V (Validate), E (Event) and R (Render).

Step 10 (M: HttpModule): Once the page object is executed and unloaded from memory, HttpModule provides post page execution events which can be used to inject custom post-processing logic. There are 4 important post-processing events PostRequestHandlerExecute, ReleaseRequestState, UpdateRequestCache and EndRequest.

Question 153 – What is the ASP.NET Page Life Cycle?

Section	Event	Description
HttpModule	BeginRequest	Signals a new request; guaranteed to be raised on each request.
HttpModule	AuthenticateRequest	Signals Asp.Net runtime is ready to authenticate the user.
HttpModule	AuthorizeRequest	Signals Asp.Net runtime is ready to authorize the user.
HttpModule	ResolveRequestCache	Asp.Net runtime determines Page can be loaded from Cache or to load completely. Caching logic can be implemented here.
HttpModule	AcquireRequestState	Signals Asp.Net runtime is ready to acquire session variables.
HttpModule	PreRequestHandlerExecute	Before handling over the control to handler if we needed to perform any tasks.
HttpHandler	ProcessRequest	Write logic which needs to be executed as per page extensions.
Page	Init	Creating controls dynamically, in case you have controls to be created on runtime. Any setting initialization. Master pages and the settings. We do not have access to viewstate, postedvalues and neither the controls are initialized.
Page	Load	Asp.Net controls are fully loaded and write UI manipulation logic.

Page	Validate	Any validators on your page, If we like to check the same here.
	Render	Like to make some changes to the final HTML which is going out to the browser, you can enter your HTML logic here.
Page	Unload	Page object is unloaded from the memory.
HttpModule	PostRequestHandlerExecute	Any logic you would like to inject after the handlers are executed.
HttpModule	ReleaseRequestState	Like to save update some state variables like session variables.
HttpModule	UpdateRequestCache	Before you end, if you want to update your cache.
HttpModule	EndRequest	Last stage before your output is sent to the client browser.

Question 154 – What is a Directive in ASP.Net?

Directives specify settings that are used by the page and user-control compilers when the compilers process ASP.NET Web Forms pages and user control files. There are 11 types of directives in asp.net and are as follows

- @Page - Enables you to specify attributes and values for an Asp.Net Page to be used when the page is parsed and compiled.
- @Master - It is quite similar to the Page directive. The Master directive belongs to Master Pages that is .master files.
- @Control - It is used when we build an Asp.Net user controls. The @Control directive helps us to define the properties to be inherited by the user control. These values are assigned to the user control as the page is parsed and compiled.
- @Register - Associates aliases with namespaces and classes, which allow user controls and custom server controls to be rendered when included in a requested page or user control.
- @Reference - Links a page, user control, or COM control to the current page or user control declaratively.
- @PreviousPageType - Creates a strongly typed reference to the source page from the target of a cross-page posting.
- @OutputCache - It controls the output caching policies of the Asp.Net page or user control.

- @Import - Imports a namespace into a page or user control explicitly.
- @Implements - It gets the Asp.Net page to implement a specified .NET framework interface.
- @Assembly - It is used to make your ASP.NET page aware of external components.
- @MasterType - To access members of a specific master page from a content page, you can create a strongly typed reference to the master page by creating a @MasterType directive.

Question 155 – What are the different Validation Controls in ASP.Net?

- Required Field Validator – Makes an input control as a required field.
- Range Validator – Check that the users input falls under a specific range.
- Compare Validator – Compare the value of 1 input control to other input control or to fixed value.
- Regular Expression Validator – Ensure the value of 1 input control matches a specified pattern.
- Custom Validator – we can write our own validation logic, e.g. we can check for values at run time.
- Validation Summary – Displays report of all the validation errors of current page.

Question 156 – What is the Difference between User Control and Custom Control?

User Controls	Custom Controls
Easier to create	Harder to create
Compiled at runtime	Pre-compiled control
Cannot be added to tool box	Can be added to toolbox
Good for Static layout	Good for dynamic layout
Not compiled in to dll	Compiled into dll
HTML Design(Visual Design possible)	No Visual Design, HTML needs to be declared programmatically
One user control cannot be used in different projects	one custom control can be used in different projects

Question 157 – What is the Difference between Client Side and Server Side Code?

Client Side	Server Side
Code is written in a scripting language such as JavaScript and HTML	Code is written in VB, C# or other compiled languages
The browser itself executes the code in response to a user action and no server round trip is involved	Code is executed by the server during a roundtrip in response to a user request or action
Client browser executes code to dynamically modify the HTML. This is called Dynamic HTML	The server executes server side code and returns HTML code to the client browser for display
Code is script and therefore it is interpreted	Code is either compiled dynamically or precompiled into assemblies

Question 158 – What is the Difference between Server.Transfer and Response.Redirect?

Server.Transfer	Response.Redirect
There is no Round trip	There is a Round trip
Previous page values are accessible	Previous page values are not accessible
Performance wise its better	Less performance
Information can be preserved in Server transfer through a parameter called preserveForm	A state can be maintained in Response redirect but has a lot of drawbacks

Question 159 – What are the different IIS Isolation Levels in ASP.Net?

IIS5 supports three isolation levels

- Low (IIS Process)
 - ASP pages run in INetInfo.Exe, the main IIS process
 - ASP crashes, IIS crashes
- Medium (Pooled)
 - ASP runs in a different process, which makes this setting more reliable
 - If ASP crashes IIS won't.
- High (Isolated)
 - Each ASP application runs out-process in its own process space
 - If an ASP application crashes, neither IIS nor any other ASP application will be affected

Question 160 – What are the different Authentication Modes available in ASP.Net?

- Windows
- Form
- Passport
- None

Question 161 – What is Windows Authentication Mode in ASP.Net?

Windows Authentication mode relies upon IIS to perform the required authentication of a client. After IIS authenticates a client, it passes a security token to ASP.NET.

Question 162 - What are the advantages and disadvantages of Windows Authentication Mode?

Advantages - Authenticates using Windows accounts, so you do not need to write any custom authentication code.

Disadvantages - May require the use and management of individual Windows user accounts.

Question 163 – What is Form Authentication Mode in ASP.Net?

Form Authentication is an authentication scheme that makes it possible for the application to collect credentials using an HTML form directly from the client. If your application authenticates the client, it issues a cookie to the client

Question 164 - What are the advantages and disadvantages of Form Authentication Mode?

Advantages - Makes it possible for custom authentication schemes using arbitrary criteria. Can be used for authentication or personalization. It does not require corresponding Windows accounts.

Disadvantages - Is subject to replay attacks for the lifetime of the cookie, unless using SSL/TLS. Is only applicable for resources mapped to AspNet_isapi.dll.

Question 165 – What is Passport Authentication Mode in ASP.Net?

Passport Authentication is a centralized authentication service provided by Microsoft that offers a single logon. When register with Passport, the Passport service grants a site-specific key. The server uses this key to encrypt and decrypt the query strings passed between the site and the Passport logon server.

Question 166 - What are the advantages and disadvantages of Passport Authentication Mode?

Advantages - Supports single sign-in across multiple domains. It is Compatible with all browsers.

Disadvantages - Places an external dependency for the authentication process.

Question 167 – What is None Authentication Mode in ASP.Net?

None Authentication Mode is used when users are not authenticated at all or if you plan to develop custom authentication code.

Question 168 - What are the advantages and disadvantages of None Authentication Mode?

Advantages - Offers total control of the authentication process providing the greatest flexibility. It also provides the highest performance if you do not implement an authentication method.

Disadvantages - Custom-built authentication schemes are seldom as secure as those provided by the operating system. It requires extra work to custom-build an authentication scheme.

Question 169 – What are the different IIS authentications available?

IIS provides a variety of authentication schemes:

- Anonymous (enabled by default) - Anonymous authentication gives users access to the public areas of your Web site without prompting them for a user name or password.
- Basic - username and password are transmitted in clear text
- Digest - username and password are transmitted with encrypted format
- Integrated Windows authentication (enabled by default) - NTLM authentication/ Kerberos V5
- Client Certificate Mapping - A certificate is a digitally signed statement that contains information about an entity and the entity's public key, thus binding these two pieces of information together.

Question 170 – What is an Event Bubbling?

Server Controls like DataGrid, DataGridView, DataList etc have other controls inside them.

A control can participate in event bubbling through two methods that it inherits from the base class

System.Web.UI.Control. These methods are OnBubbleEvent and RaiseBubbleEvent.

Example an DataGridView can have an Textbox or an button inside it. These Child Controls cannot raise events by themselves, but they pass the event to the parent control (DataGridView), which is passed to the page as "ItemCommand" event. This process is known as Event Bubbling.

Question 171 – What are the differences between Machine.Config and a Web.Config files in Asp.Net?

Web.config	Machine.config
Machine level configuration	Application/folder level configuration
Can have more than one Web.config	Only one Machine.config
Web.config overrides the machine.config file	Same
Automatically installed when installing Visual Studio.	Created when creating an ASP.Net web application

Question 172 – What is an Authentication?

Authentication is a process of identifying a user based on their credentials (means user id and password). There are 5 types of authentication and are as below.

- Password based authentication
- Device based authentication
- Biometric Authentication
- Retina Scanners
- Hand Scanners

Question 173 – What is an Authorization?

Authorization is a process of determining whether an authenticated user is allowed to access a specific resource or not.

Question 174 – What is an Impersonation?

- It is the process of executing code in the context of another user identity.
- It is the mechanism in the ASP.NET which decide under which account web application should be run.
- By default Impersonation is off, we need to enable it in Web.config file.

- By default all the ASP.NET applications run under the ASP.NET user account. If you set the impersonation to TRUE in the Web.config file then that application runs under the current user's account who has logged in that machine. `<identity impersonate="true" />`
- We can use the impersonation in this two scenarios:
 - To give each web application different permissions.
 - To use existing Windows user permission.
- These two scenario are fundamentally different. In the first one, impersonation defines a single, specific account. In this case, no matter what user access the application, and no matter what type of user-level security you use, the code will run under the account you've set. In the second one, the user must be authenticated by IIS. The web-page code will then execute under the identity of the appropriate user.

Question 175 – What is Autopostback event in ASP.Net?

It is the mechanism, by which the page will be posted Back to the server automatically based on some events in the web controls.

Example - Dropdown Box web control has the property autopostback, if we set the property to true, when ever user selects a different value in the combo box, and event will be fired in the server. i.e. a request will be send to the server.

Autopostback - Property of the control

IsPostBack - Property of the Page class

IsPostBack is a property which returns Boolean value. It checks weather the page is posted back or not.

Question 176 – What is Tracing in ASP.Net?

Tracing is a way to monitor the execution of your ASP.NET application. You can record exception details and program flow in a way that doesn't affect the program's output.

Page level Tracing - Enabled on a page-by-page basis by adding "Trace=true" to the Page directive

Application Tracing - You can enable tracing for the entire application by adding tracing settings in web.config. In below example, pageOutput="false" and requestLimit="20"

Question 177 – What is Scavenging?

It is the process of deleting items from the cache when memory is scarce. Items are removed when they have not been accessed in some time or when items are marked as low priority when they are added to the

cache. ASP.NET uses the CacheItemPriority object to determine which items to scavenge first. CacheItemPriority.High assigns a priority level to an item so that the item is least likely to be deleted from the cache.

Question 178 – What are Trace Listeners?

- It is a class that listens for trace and debug messages from .NET applications.
- Display messages to users by writing messages to files.
- It can modify messages before displaying them, or even we can filter out the messages
- It is common practice to simply log the message with possibly some formatting for easy readability.
- It is derived from the System.Diagnostics.TraceListener base class.
- The .NET Framework ships with three trace listeners:
 - EventLogTraceListener - logging messages to event logs.
 - DefaultTraceListener - logging messages to standard debug output such as debuggers.
 - TextWriterTraceListener - logging messages to text files.

Question 179 – What is the difference between Response.Write and Response.Output.Write?

Response.Output.Write	Response.Write
Formatted output will be displayed.	unformatted output will be displayed.
It gives String.Format-style formatted output.	It never gives like that.
It writes the HTTP Output Stream.	It writes the text stream
As per specified options it formats the string and then write to web page.	It just output a string to web page.

Question 180 – What is SmartNavigation?

- It is the property that enables the individual controls to maintain focus betweenPostBack
- The features of SmartNavigation property are namely:
- Maintaining element focus between post backs
- Eliminate page flash caused by page post back
- Prevents each post back from being saved in the browser history
- The scroll position of a Web page is maintained after PostBack.
- Only the most recent Web page state is retained in the Web browser history folder.

Question 181 - What is a State Management in ASP.Net?

State management is the process by which you maintain state and page information over multiple requests for the same or different pages. There are 2 types State Management:

- Client – Side State Management
- Server – Side State Management

Question 182 - What are the different management objects available with Client & Server Side State Management?

Client – Side State Management

- **View State** – Asp.Net uses View State to track the values in the Controls. You can add custom values to the view state. It is used by the Asp.net page framework to automatically save the values of the page and of each control just prior to rendering to the page. When the page is posted, one of the first tasks performed by page processing is to restore view state.
- **Control State** – If you create a custom control that requires view state to work properly, you should use control state to ensure other developers don't break your control by disabling view state.
- **Hidden fields** – Like view state, hidden fields store data in an HTML form without displaying it in the user's browser. The data is available only when the form is processed.
- **Cookies** – Cookies store a value in the user's browser that the browser sends with every page request to the same server. Cookies are the best way to store state data that must be available for multiple Web pages on a web site.
- **Query Strings** - Query strings store values in the URL that are visible to the user. Use query strings when you want a user to be able to e-mail or instant message state data with a URL.

Server – Side State Management

- **Application State** - Application State information is available to all pages, regardless of which user requests a page.
- **Session State** –Information is available to all pages opened by a user during a single visit. Both application state and session state information is lost when the application restarts. To persist user data between application restarts, you can store it using profile properties.

Question 183 - How to Pass values between pages?

We can use one of the below options to pass value between 2 pages in ASP.Net.

- If the value is a String then we can use either one of the following: QueryString, PostBack, Cookies, Session, Cache, Application, Database
- If the value is a Dataset or an object then we can use either one of the following: (Assuming DTO object not serialized) - Session, Application, Cache

Question 184 - What is a View State?

- It is a built-in structure for automatically retaining values amongst multiple requests for the same page.
- The viewstate is internally maintained as a hidden field on the page but is hashed.
- View State is used to retain the state of server-side objects between post backs.
- Controls perform well - Label, TextBox, CheckBox, RadioButton, and HyperLink
- Controls perform less - DropDownList, ListBox, DataGrid, and DataList because of size and large data.
- Making roundtrips to the server.
- Item stored in Viewstate exist for the life of the current page. This includes PostBack (to the same page).

Question 185 - What are the Benefits of ViewState?

- No server resources are required.
- The values in view state are hashed, compressed, and encoded, thus representing a higher state of security than hidden fields
- It is good for caching data in Web frame configurations because the data is cached on the client
- Simple to use

Question 186 - What are the Limitations of ViewState?

- Page loading and posting performance decreases when large values are stored.
- Although view state stores data in a hashed format, it can still be tampered because it is stored in a hidden field on the page.

Question 187 - What is an EnableViewState in ASP.Net?

- EnableViewState turns on the automatic state management feature that enables server controls to re-populate their values on a round trip.
- Through EnableViewState=true /false we can define whether we have to maintain values of control in hidden fields when page is posted back again and again
- For example, if you are binding a control to data on every round trip then you do not need the control to maintain its view state, since you will wipe out any re-populated data in any case.
- Viewstate is enabled for all server controls by default. To disable it, set the EnableViewState property of the control to false.

Question 188 - What are Hidden Fields in ASP.Net?

- These are html input control with hidden type that store hidden data in the html.
- The data stored in hidden field is available when form processed on the server or using JavaScript.
- Syntax - `<input type="hidden" name="EVENTTARGET" id="EVENTTARGETID" value="" />`
- Can store only one value in their value property. The value is saved as a string so we need to Typecast.
- There are two types of server control hidden fields -
 - `System.Web.UI.WebControls.HiddenField`
 - `System.Web.UI.HtmlControls.HtmlInputHidden.`

Question 189 - What are the Benefits of Hidden Fields in ASP.Net?

- Easy to implement, Hidden fields are supported by all browsers
- Faster access of information because data is stored on client side
- No server resources are required.

Question 190 - What are the Limitations of Hidden Fields in ASP.Net?

- Not secure because the data is stored on Client side so can be tampered creating a security hole
- Decreases page performance if too many hidden fields(large data's)
- Only support single value and do not support rich structures as HTML.
- Have to work around with delimiters etc to handle complex structures.

Question 191 - What are Hidden Frames in ASP.Net?

- A hidden frame is a frame with 0 width and height, so that no one can see it.
- You can run programs in there when you need to make a round trip to the server so that your main frame does not refresh.
- To create one, you just use a frameset tag
- A hidden frame is a complete document, e.g. IFRAME, that is not visible.

Question 192 - What are the Benefits of Hidden Frames in ASP.Net?

- Hidden frames allow you to cache more than one data field.
- The ability to cache and access data items stored in different hidden forms.

Question 193 - What are the Limitations of Hidden Frames in ASP.Net?

- Not supported by all browsers
- Stored data is not secured as it is stored on client's machine in the form of hidden fields.

Question 194 - What is a Cookie in ASP.Net?

- It is a small bit of text that accompanies requests and pages as they go between the server and browser.
- The cookie contains information the Web application can read whenever the user visits the site.
- It provide a useful means in Web applications to store user-specific information.
- Cookies are associated with a Web site, not with a specific page, so the browser and server will exchange cookie information no matter what page the user requests from your site.
- Most browsers support cookies of up to 4096 bytes.
- Most browsers allow only 20 cookies per site, so if you try to store more, the old cookies are discarded.
- Maximum 300 cookies can be stored that will be accepted from all sites combined.

Question 195 - What are the examples of Cookie usage in ASP.Net?

- Cookies are used for all sorts of purposes, all relating to helping the Web site remember you.

- For example, a site conducting a poll might use a cookie simply as a Boolean value to indicate whether your browser has already participated in voting so that you don't vote twice.
- A site that asks you to log on might use a cookie to tell itself that you've already logged on so that you don't have to keep entering your credentials.

Question 196 - What are the Types of Cookies in ASP.Net?

- Persistent - Means the cookie will be expired as soon as the application is closed
- Non Persistent - Even the application is closed the data will be remained as per the cookie timeout value.

Question 197 - What are the Benefits of Cookies in ASP.Net?

- They are simple to use.
- Light in size, thus occupy less memory.
- Stores server information on client side
- Data need not to be sent back to server.
- Cookies persist for much longer period of time than a session.

Question 198 - What are the Limitations of Cookies in ASP.Net?

- Since they are stored on the client side in plain text, they are not secure.
- Number of cookies that can be stored and their size is limited.
- They don't work if security level is set too high in browser.
- Some people might just disable cookies on their browsers.

Question 199 - What is the relation between Cookies and Session State?

- By default, ASP.NET uses a non-persistent cookie to store the session state.
- If a user has disabled cookies on the browser, Session state cannot use a cookie to store the session ID and Session state doesn't work. - No cookie, No Session state.

Question 200 - What is a Cookieless Session in ASP.Net?

You can configure your application to store Session IDs not in a cookie, but in the URLs of pages in your site. By keeping the Session ID in the URL, ASP.NET stores the ID in the browser, in a manner of speaking, and can get it back when the user requests another page. Cookieless sessions can get around the problem of a browser that refuses cookies and allow you to work with Session state.

However, under some limited circumstances, if the user shares the URL with someone else — perhaps to email the URL to a colleague while the user's session is still active — then both users might end up sharing the same session, with unpredictable results.

Question 201 - What are the Advantages of Session?

- It helps to maintain user states and data to all over the application.
- It can easily be implemented and we can store any kind of object.
- Stores every client data separately.
- Session is secure and transparent from user.

Question 202 - What are the Disadvantages of Session?

- Performance overhead in case of large volume of user, because of session data stored in server memory.
- Overhead involved in serializing and De-Serializing session Data, because In case of State Server and SQL Server session mode we need to serialize the object before store.

Question 203 - What is an In Proc Mode of storing sessions?

In this mode Session, state is stored in the memory space of the Aspnet_wp.exe process. This is the default setting. If the IIS reboots or web application restarts then session state is lost.

Question 204 - What are the Advantages of InProc Sessions?

- It store Session data in memory object of current application domain. So accessing data is very fast and data is easily available.
- There is not requirements of serialization to store data in InProc Session Mode.
- Implementation is very easy, just similar to using View State.

Question 205 - What are the Disadvantages of InProc Sessions?

- If the worker Process or application domain recycles all session data will be lost.
- Though its fastest, but more session data and more users can affects performance, because of memory.
- we can't use it in web Garden scenarios .
- This session mode is not suitable for web farm scenarios also.

Question 206 - What is a State Server Mode of Storing Sessions?

In this mode Session state is serialized and stored in a separate process (Aspnet_state.exe); therefore, the state can be stored on a separate computer (a state server).

Question 207 - What are the Advantages of State Server Session?

- Its keeps the data separate from IIS so, any Issue with IIS does not hamper Session data.
- It is useful in web farm and web garden scenarios.

Question 208 - What are the Disadvantages of State Server Session?

- Process is slow due to Serialization and De-Serialization
- State Server always need to be up and running.

Question 209 - What is a SQL Server Mode of Storing Session?

- In this mode Session, state is serialized and stored in a SQL Server database.
- SQL Server Session mode is more reliable and secure session state management.
- Its keeps data in a centralized location (database).
- We should use SQL server session mode when we need to implement Session with some more security.
- If there happens to be frequent server Restart we can implement SQL server.
- This is perfect mode that fits in web farm and web garden scenarios (I have explained in details later).
- we can use SQL server Session mode when we need to share session between two different application.

Question 210 - What are the Advantages of SQL Server mode Session?

- Session data do not affected if we restart the IIS.
- It is the most reliable and secure session management.
- It keeps data located centrally, It can be easily accessible from other application.
- It is very useful in web farm and web garden scenarios.

Question 211 - What are the Disadvantages of SQL Server modes?

- Processing is very slow in nature.
- Object serialization and de-serialization creates overhead for application.
- As the session data is handled in different server, so we have to take care of SQL server. It should be always up and running.

Question 212 - What is Custom Session Mode in ASP.Net?

- We can use custom session mode in following of the cases,
- We want to store session data rather than SQL Server.
- When we have to use some Existing table to store session data.
- When we need to create our own session ID.

Question 213 - What are the Advantages of using Custom Session Mode in ASP.Net?

- It is useful when we have to use some old database rather than SQL Server.
- It's not depending on IIS , So Restarting web server does not make any effects on session data.
- We can create our own algorithm for generating Session ID.

Question 214 - What are the Disadvantages of using Custom Session Mode in ASP.Net?

- Processing of Data is very slow.
- Creating a custom state provider is a low-level task that needs to be handled carefully to ensure security.

Question 215 - What is a Query String in ASP.Net?

- QueryString is way to transfer information from one page to another through the URL.
- QueryString is attached to the URL with "?".

Question 216 - What are the Benefits of a Query String in ASP.Net?

- Supported by all the browsers
- No extra effort is needed to code.
- Easy to use.

Question 217 - What are the Limitations of a Query String in ASP.Net?

- All the attributes and values are visible to the end user. Therefore, they are not secure.
- There is a limit to URL length of 255 characters.

Question 218 - What is Cross Page Posting in ASP.Net?

- Sometimes, other than round-trip cycle of asp.net web page we need to post one page to another page by configuring controls to be posted to different target page. This is referenced as cross page posting in ASP.NET.
- Cross page posting is a new feature introduced in ASP.NET 2.0, which eases the life of developers previously they have to use Server.Transfer which has its own advantages and disadvantages but now this is a part of ASP.NET which results flexibility and efficiency.

Question 219 - What is SQL Cache Dependency in ASP.Net?

It is a new technique in ASP.NET 2.0 which can automatically invalidate a cached data object just like a Dataset. when the related data is modified in the database. So for instance if you have a dataset which is tied up to a database tables any changes in the database table will invalidate the cached data object which can be a dataset or a data source.

To enable this we need a syntax that is as follows:-

aspnet_regsql -ed -E -d Northwind

-ed :- command-line switch

-E: - Use trusted connection

- S: - Specify server name it other than the current computer you are working on
- d: - Database Name

Question 220 - What is Global.asax in ASP.Net?

- It used to implement session level and application level events raised by ASP.net or by HTTP modules.
- It is also known as ASP.net application file and resides in the root directory of the ASP.NET application.
- Events available with global.asax are as follows

Question 221 - What are the Event available in Global.asax?

Events	Description
Application_Init	Fired when an application initializes or is first called. It's invoked for all HttpApplication object instances.
Application_Start	Fired when first instance of the HttpApplication class is created. It allows to create objects that are accessible by all HttpApplication instances.
Session_Start	Fired when a new user visits the application Web site.
Application_BeginRequest	Fired when an application request is received. It's the first event fired for a request, which is often a page request (URL) that a user enters.
Application_EndRequest	The last event fired for an application request.
Application_AuthenticateRequest	Fired when the security module has established the current user's identity as valid. Here, the user's credentials have been validated.
Application_Error	Fired when an unhandled exception is encountered within the application.
Session_End	Fired when a user's session times out, ends, or they leave the application Web site.
Application_End	Fired when the last instance of an HttpApplication class is destroyed. It's fired only once during an application's lifetime.

Question 222 - What is Caching in ASP.Net?

- Caching is a technique where we can store frequently used data and Web pages are stored temporarily on local hard disks for later retrieval.
- This technique improves the access time when multiple users access a Web site simultaneously or a single user accesses a Web site multiple times.
- Caching can occur on the client (browser caching), on a server between the client and the Web server (proxy / Reverse Proxy Caching), and on the Web server itself (page or data caching).
- We can classified caching location in 4 way
 - Client Caching (Client Browser)
 - Proxy Caching (In between Client and Server)
 - Reverse Proxy Caching (In between Client and Server)
 - Web Server Caching (Data Caching/ Page Output Caching)

Question 223 - What are the Types of Caching in ASP.Net?

- **Page Output caching [Output caching]** - Is used to fetch information or data at page level. It is best used when the site is mainly static. Used by declaring the output page directive
- **Fragment caching [Output caching]** - Is used to cache the structure level information. It is used when parts of pages change. For example: user control
- **Data Caching** - Is used to fetch the information of an application quickly based on the requirements.

Question 224 - How to cache different versions of the same page in ASP.Net?

- The ways to cache different versions on the same page using ASP.NET cache object is using OutputCache object. Three parameters of OutputCache, has 3 parameters:
- VarByParam: based on input parameters passed through HTTP POST/GET.
- VarByHeader: based on the page header content.
- VayByCustom: based on declaration of an attribute and overriding GetVaryByCustomString handler.
- VarByControl: varies with the value of properties of ASP objects.

Question 225 - What is a Fragment Cache in ASP.Net?

It is used to store user controls individually within a web form in cache instead of the whole web form as such. The idea is to simply have different cache parameters for different user controls.

control1: <%@ OutputCache Duration="40" VaryByParam="none"%>

control2: <%@ OutputCache Duration="60" VaryByParam="none"%>

If control1 and control2 exist on a single web form together, the values of control1 and control2 would be cached for different durations based on @OutputCache directive.

Question 226 - What are Resource Files in ASP.Net?

- A resource file is a XML file that contains the strings that we want to Translate into different languages.
- Can be updated dynamically so that user themselves can modify values in resource files once the application is deployed on the server without re-compiling the entire application itself.
- The resource file contains key / value pairs.
 - Each pair is an individual resource.
 - Key names are not case sensitive.

Question 227 - What are the Types of Resource Files available in ASP.Net?

There are two types of resources

- Local Resources
- Global Resources

Question 228 - What is a Local Resource File in ASP.Net?

- It is specific to a single Web page and used for providing versions of a Web page in different languages.
- It must be stored in App_LocalResources sub folder.
- It must be named in format <WebPageName> [.language / language and culture].resx.

Question 229 - What is a Global Resource File in ASP.Net?

- Global resource can be read from any page or code that is in the application.
- Global resource must be stored in App_GlobalResources at the root of the application.

Question 230 - What are the main tags in Web.Config file of ASP.Net application?

<authentication> - This element is used to verify the client's identity when the client requests a page from the server. This is set at the application level. We have four types of authentication modes: "None", "Windows", "Forms", and "Passport".

<authorization> - The <authorization> tag controls client access to web page resources. This element can be declared at any level (machine, site, application, subdirectory, or page).

<authorization>

<allow users="comma-separated list of users"

roles="comma-separated list of roles"

verbs="comma-separated list of verbs"/>

<deny users="comma-separated list of users"

roles="comma-separated list of roles"

verbs="comma-separated list of verbs"/>

</authorization>

<allow>: Using this tag, we can control access to resources on the basis of the following verbs. In these attributes, we use symbols: ? and *.? means for anonymous users/resources, and * means for all users.

users: This contains the list of user names (comma separated) that are allowed to access the resources.

roles: This contains the list of roles (comma separated) that are allowed to access the resources.

verbs: This contains the list of HTTP verbs to which the action applies (comma separated). It is used to create a rule that applies to a specific type of HTTP request (GET, POST, HEAD, OR DEBUG).

<deny>: Using this tag, we can control access to resources on the basis of the following verbs:

users: This contains the list of users names (comma separated) that are denied access to the resources.

roles: This contains the list of roles (comma separated) that are denied access to the resources.

verbs: This contains the list of HTTP verbs to which the action applies (comma separated). It is used to create a rule that applies to a specific type of HTTP request (GET, POST, HEAD, OR DEBUG).

Question 231 - What is the use of <compilation> tag in ASP.Net Web.Config File?

- Compilation tag configure the settings of the compiler.
- debug and defaultLanguage are the common attributes used.
- Setting debug to true means we want the debugging information in the browser, but it has a performance tradeoff, so normally, it is set as false.

- The defaultLanguage attribute tells ASP.NET which language compiler to use: VB or C#.

Question 232 - What is the use of <customErrors> tag in ASP.Net Web.Config File?

- Custom Error tags includes the error settings for the application, and is used to give custom error pages (user-friendly error pages) to end users.
- If an error occurs then, the website is redirected to the default URL.
- For enabling and disabling custom errors, we need to specify the mode attribute.

- Example of <customErrors> tag:

```
<customErrors defaultRedirect="url" mode="Off">  
  <error statusCode="403" redirect="/accesdenied.html" />  
  <error statusCode="404" redirect="/pagenotfound.html" />  
</customErrors>
```

- "On" means if there is any error, the website is redirected to the default URL.

"Off" means the custom errors are disabled.

"RemoteOnly" shows that custom errors will be shown to remote clients only.

Question 233 - What is the use of <globalization> tag in ASP.Net Web.Config File?

- Globalization tag is used when we want to use encoding or specify a culture for the application.
- We need to define the character set for the server to send the response to the client, which is by default is UTF-8, and the settings of which the server should use to interpret and display culturally specific strings, such as numbers and dates.

- Example of <globalization> tag

```
<globalization requestEncoding="utf-8" responseEncoding="utf-8" />
```

Question 234 - What is the use of <httpRuntime> tag in ASP.Net Web.Config File?

- Httpruntime tag can be used to configure the general runtime settings of the application.

The 2 main settings are: <httpRuntime appRequestQueueLimit="50" executionTimeout="600" />

- The `appRequestQueueLimit` attribute defines the number of requests that can be queued up on the server for processing. If there are 51 or more requests, then server would return the 503 error ("Server too busy").
- The `executionTimeout` attribute defines the number of minutes ASP.NET will process a request before it gets timeout.

Question 235 - What is the use of <trace> tag in ASP.Net Web.Config File?

- Trace tag is used for tracing the execution of an application.
- We have here two levels of tracing: page level and application level.
- Application level enables the trace log of the execution of every page available in the application. If `pageOutput="true"`, trace information will be displayed at the bottom of each page. Else, we can view the trace log in the application root folder, under the name `trace.axd`.
- Example of <trace> tag

```
<trace enabled="false" requestLimit="10" pageOutput="false" traceMode="SortByTime" localOnly="true" />
```

Set the attribute `localOnly` to false for not viewing the trace information from the client. For enabling trace at page level, set `Trace="True"` in the Page tag (on the top of the page).

Question 236 - What is the use of <identity> tag in ASP.Net Web.Config File?

Identity tag is used to control the identity of the application. By default, Impersonation is disabled. Using Impersonation, an ASP.NET application can execute optionally with the identity of a client on whose behalf they are operating.

```
<identity impersonate="false" userName="domain\username" password="password" />
```

Question 237 - What is the use of <sessionState> tag in ASP.Net Web.Config File?

- Sessionstate tag is used to tell the ASP.NET framework where to store the session. By default, it's `inproc` which means storing the session values on the server. But we have four options:
- "Off" means session is not enabled for the application.
- "inproc" means storing the session values on the server.

"StateServer" means session states are stored in a remote server.

"SQLServer" means session states are stored in a SQL Server database. For this, we need to install the InstallSQLState.sql script in the SQL Server database. It is mainly used when we use web farms (an application deployed on multiple servers), but it makes the performance slow as compared to "inproc".

- Other Basic Settings:

"cookieless": when it is true, it means the session used is without cookies.

"timeout" specifies after how much time the session would expire

"stateConnectionString" needs to be specified when the session mode is StateServer.

"sqlConnectionString" is connection string of the SQL Server database if the session mode is SQLServer.

"stateNetworkTimeout" attribute, when using the StateServer mode to store session state, specifies the number of seconds the TCP/IP network connection between the web server and the state server can be idle before the session is abandoned. The default is 10.

- Example

```
<sessionState mode="Off" cookieless="true"
    timeout="100"
    stateConnectionString="tcpip=server:port"
    sqlConnectionString="sql connection string"
    stateNetworkTimeout="number of seconds"/>
```

Question 238 - What is the use of <appSettings> tag in ASP.Net Web.Config File?

- AppSetting tag is used to store custom application configuration like database connection strings, file paths etc.
- This also can be used for custom application-wide constants to store information over multiple pages. It is based on the requirements of the application.

- Example:

```
<appSettings>
    <add key="Emailto" value="me@microsoft.com" />
    <add key="cssFile" value="CSS/text.css" />
</appSettings>
```

It can be accessed from code like:

`ConfigurationSettings.AppSettings("Emailto");` All the values returned from it are strings.

Question 239 - What is HTTP GET in ASP.Net?

- In HTTP GET method data passed through url QueryString using name value pair. It's simpler and you can troubleshoot any problems simply by looking at the address bar in your browser because all values passed are displayed there.
- This is also the primary weakness of this method. The data being passed is visible and is limited in size to the maximum length of a request string.

Question 240 - What is HTTP POST in ASP.Net?

- In HTTP POST request data are embedded in a HTTP HEADER. So data are NOT visible to end user while you can see the data passed in HTTP GET method.
- If you want to pass sensitive information/data, you should have to use HTTP POST request. Another advantage is that you can send larger amounts of information compare to HTTP GET method.

Question 241 – What is Normalization in SQL Server?

- In relational database design, the process of organizing data to minimize redundancy is called normalization.
- It usually involves dividing a database into 2 or more tables and defining relationships between tables.
- Objective is to isolate data so that additions, deletions, and modifications can be made in just one table.

Question 242 – What is a De-Normalization in SQL Server?

It is a technique to move from higher to lower normal forms of database modeling in order to speed up database access.

Question 243 – What are the different Normalization Forms in SQL Server?

- 1NF: Eliminate Repeating Groups

- 2NF: Eliminate Redundant Data
- 3NF: Eliminate Columns Not Dependent On Key
- 4NF: Isolate Independent Multiple Relationships
- 5NF: Isolate Semantically Related Multiple Relationships

Question 244 – What is a Stored Procedure in SQL Server?

- It is a named group of SQL statements created and stored in the database.
- Accept input parameters so that single procedure can be used by several clients using different inputs.
- When the procedure is modified, all clients automatically get the new version.
- It reduces network traffic and improve performance.
- It can be used to help ensure the integrity of the database.

Question 245 – What are the Advantages of Stored Procedure in SQL Server?

- Execution plan retention and reuse
- Encapsulation of business rules and policies
- Sharing of application logic between applications
- Access to database objects that is both secure and uniform
- Consistent, safe data modification
- Network bandwidth conservation
- Improved security
- Reduced development cost and increased reliability
- Centralized security, administration, and maintenance for common routines.

Question 246 – What are the DisAdvantages of Stored Procedure in SQL Server?

- Source control can be a pain.
- Debugging is hard.
- Developing stored procedures can be a fairly specialized task, especially as they get more complex.

Question 247 – What is a User Defined Functions in SQL Server?

A user-defined function (UDF) is a prepared code segment that can accept parameters, process some logic, and then return some data. UDFs in SQL Server 2000 can accept anywhere from 0 to 1024 parameters

Question 248 – What is a Scalar value-returning User Defined Function in SQL Server?

- Similar to functions in other languages.
- Returns a single value of a scalar data type.
- Can return user-defined data types.
- Useful when the same segment of T-SQL code is used in several places
- Can be used in several stored procedures and batch SQL statements.
- Used to make the code more maintainable, reusable, and less complex.

Question 249 – What is an In Line Table User Defined Function in SQL Server?

- Returns a row set of the SQL Server table data type.
- Takes the structure of the row set from the SELECT statement itself.
- It is an exceptional alternative to a view
- Can pass parameters and in essence provide us with a parameterized, non-updateable view

Question 250 – What is a Multistatement Table User Defined Function in SQL Server?

- Returns a table and is also an exceptional alternative to a view
- Defines column names and data types in the RETURNS clause.
- Able to house more complicated and numerous T-SQL logic blocks.
- It can be used it in the FROM clause of a T-SQL command

Question 251 – What are the Advantages of User Defined Functions?

- UDF can be used in a Select, Where, or Case statement.
 - UDF can be used in join T-SQL Statements
 - It can act like a table, so the ability to break out complex logic into shorter and shorter code blocks
- Additional benefit of making the code less complex and easier to write and maintain.

- We can pass parameters to customize and gets the return based on our requirement
- Simpler to invoke in sql statements than a stored procedure.

Question 252 – What are the DisAdvantages of User Defined Functions?

- UDF that return non-deterministic values are not allowed to be called from inside UDF
- GETDATE is an example of a non-deterministic function. Every time the function is called, a different value is returned.
- Not every SQL statement or operation is valid within a function.
- The following lists enumerate the valid and invalid function operations:
 - Valid:
 - Assignment statements
 - Control-flow statements
 - Variable declarations
 - SELECT statements that modify local variables
 - Cursor operations that fetch into local variables
 - INSERT, UPDATE, DELETE statement that act upon local table variables
 -
 - Invalid:
 - Built-in, nondeterministic functions such as GetDate()
 - Statements that update, insert, or delete tables or views
 - Cursor fetch operations that return data to the client

Question 253 – What is the difference between a Stored Procedure and Functions?

Stored Procedure	Functions
Called independently, using the EXEC command	Called from within another SQL statement
May return a scalar value, a table value or Nothing	Always return a value (either a scalar value or a table)
Allow you to enhance application security by GRANT and Deny access	Here we can't use this

Question 254 – What is a Trigger?

- A trigger is a SQL procedure that initiates an action when an event (INSERT, DELETE, UPDATE) occurs.
- Triggers can restrict access to specific data, perform logging, or audit data modifications.
- A trigger cannot be called or executed.

Question 255 – What is a DML Trigger?

- Fired when a Data Manipulation Language (DML) event takes place
- AFTER - Executed after the action of the INSERT, UPDATE, or DELETE statement is performed.
- INSTEAD OF - Executed in place of the usual triggering action. INSTEAD OF triggers can also be defined on views with one or more base tables, where they can extend the types of updates a view can support.

Question 256 – What is a DDL Trigger?

- DDL triggers are new to SQL Server 2005.
- This type of triggers, like regular triggers, fire stored procedures in response to an event.
- They fire in response to a variety of Data Definition Language (DDL) events.
- These events are specified by the T-SQL statements that start with the keywords CREATE, ALTER, and DROP.
- Certain stored procedures that perform DDL-like operations can also fire this.
- These are used for administrative tasks like auditing and regulating database operations.

Question 257 – What is a CLR Trigger?

- CLR triggers can be a DDL or DML one or can also be an AFTER or INSTEAD OF trigger.
- Here we need to execute one or more methods written in managed codes that are members of an assembly created in the .Net framework.
- Again, that assembly must be deployed in SQL Server 2005 using CREATE assembly statement.
- The Microsoft.SqlServer.Server Namespace contains the required classes and enumerations for this objective.
- Question 258 – What are the Advantages of Triggers?
- Audit a table for security

- Automatic updating of one or more tables whenever a DML/DDL statement is executed on that table.
- Triggers can be used to enforce constraints. For e.g. : Any insert/update/ Delete statements should not be allowed on a table after office hours. For enforcing this constraint Triggers should be used.
- Used to publish information about database events to subscribers. Events like Database startup or shutdown or a user even like User login in or user logoff.

Question 259 – What are the DisAdvantages of Triggers?

- It is easy to view constraints, indexes, Sp's in database but triggers are difficult to view.
- Execute invisible to application. They are not visible or can be traced in debugging code.
- Hard to follow their logic as it they can be fired before or after the database insert/update happens.
- Easy to forget about triggers and if there is no documentation it will be difficult to figure out.
- Run every time when the db fields are updated; it is overhead on system, it makes system run slower.

Question 260 – What are Views in SQL Server?

- A view is an "Virtual Table".
- It does not contain any data directly, it is a set of query that are applied to one or more tables as object.
- It can be thought of as a subset of a table.
- It can be used for retrieving data, as well as updating or deleting rows.
- The results of using a view are not permanently stored in the database.
- A view serves as a security mechanism. This ensures that users are able to retrieve and modify only the data seen by them.
- A view also serves as a mechanism to simplify query execution. Complex queries can be stored in the form as a view, and data from the view can be extracted using simple queries.

Question 261 – What are the Types of VIEW?

- **Standard Views** - Combining data from one or more tables through a standard view. Focus on specific data and simplifying data manipulation.
- **Indexed Views** - It has been computed and stored. You index a view by creating a unique clustered index on it. Indexed views dramatically improve the performance of some types of queries. Indexed views work

best for queries that aggregate many rows. They are not well-suited for underlying data sets that are frequently updated.

- **Partitioned Views** - Joins horizontally partitioned data from a set of member tables across one or more servers. This makes the data appear as if from one table. A view that joins member tables on the same instance of SQL Server is a local partitioned view.

Question 262 – How Views ensure security of data?

- Specific rows of the tables.
- Specific columns of the tables.
- Specific rows and columns of the tables.
- Rows fetched by using joins.
- Statistical summary of data in a given tables.
- Subsets of another view or a subset of views and tables.

Question 263 – List some of the common examples of views?

- A subset of rows or columns of a base table.
- A union of two or more tables.
- A join of two or more tables.
- A statistical summary of base tables.
- A subset of another view, or some combination of views and base table.

Question 264 – What are the Limitations of views?

- A view can be created only in the current database.
- The name of a view must not be the same as that of the base table.
- A view can be created only if there is a SELECT permission on its base table.
- A SELECT INTO statement cannot be used in view declaration statement.
- A trigger or an index cannot be defined on a view.
- The CREATE VIEW statement cannot be combined with other SQL statements in a single batch.

Question 265 – What are Linked Servers in SQL Server?

- A linked server configuration enables SQL Server to execute commands against OLE DB data sources on remote servers.
- It is a concept in SQL by which we can add other SQL Server to a Group and query both the Server.
- Stored Procedure sp_addlinkedserver, sp_addlinkedsrvlogin will be used add new Linked Server.
- 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.

Question 266 – What are the Advantages of Linked Servers?

- Remote server access.
- The ability to issue distributed queries, updates, commands, and transactions across the enterprise.
- The ability to address diverse data sources similarly.

Question 267 – What is a Cursor in SQL Server?

It 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.

Question 268 – What are the steps to Create Cursor in SQL Server?

Below are the steps by step approach to create a Cursor.

- Declare cursor
- Open cursor
- Fetch row from the cursor
- Process fetched row
- Close cursor
- De-allocate cursor

Question 269 – What are the Types of Cursors in SQL Server?

There are 4 types of cursors in SQL Server and are as below.

- Static
- Dynamic
- Forward only
- Keyset Driven

Question 270 – What are the Cursor Optimization Tips available?

- Avoid using SQL Server cursors, whenever possible.
- Do not forget to close SQL Server cursor when its result set is not needed.
- Do not forget to de-allocate SQL Server cursor when the cursors are not needed.
- Reduce the number of records to process in the cursor.
- Reduce the number of columns to process in the cursor.
- Use READ ONLY cursors, whenever possible, instead of updatable cursors.
- Try avoid using insensitive, static and keyset cursors, whenever possible.
- Use FAST_FORWARD cursors, whenever possible.
- Use FORWARD_ONLY cursors, if you need updatable cursor and the FETCH NEXT will be the only used fetch option.

Question 271 – What are the Cursor Alternatives?

- Set based logic
- SQL Server Integration Services (SSIS) or Data Transformation Services (DTS)
- WHILE loop
- COALSCE
- sp_MSforeachdb - Run The Same SQL Command Against All SQL Server Databases
- sp_MSforeachtable
- CASE expression - Using the CASE expression instead of dynamic SQL in SQL Server

Question 272 – What are the Limitations of Cursors?

- Cursor requires a network roundtrip each time it fetches a record, thus consume network resources.
- While data processing, it issues locks on part of the table, or on the whole table.

Question 273 – What is a SubQuery?

- Subquery or Inner query or Nested query is a query in a query.
- A Subquery is usually added in the WHERE Clause of the sql statement.
- Used when we know how to search a value using SELECT statement, but don't know the exact value.
- Subqueries are an alternate way of returning data from multiple tables.
- Subqueries can be used with the following sql statements along with the comparison operators like =, <, >, >=, <= etc. SELECT, INSERT, UPDATE, DELETE

Question 274 – What are the Properties of Subquery?

- It must be enclosed in the parenthesis.
- It must be put in the right hand of the comparison operator.
- It cannot contain an ORDER-BY clause.
- A query can contain more than one sub-query.

Question 275 – What are the Types of Subquery?

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

Question 276 – What is a Correlated Subquery?

- A query is called correlated sub query when both the inner query and the outer query are interdependent.
- For every row processed by the inner query, the outer query is processed as well.
- The inner query depends on the outer query before it can be processed.
- Correlated SubQuery can be said to be dependent on the outer query.

- `SELECT p.product_name FROM product p WHERE p.product_id = (SELECT o.product_id FROM order_items o WHERE o.product_id = p.product_id);`

Question 277 – What are the Properties of Correlated Subquery?

- Can nest many queries you want but it is recommended not to nest more than 16 Subqueries in Oracle.
- If a Subquery is not dependent on the outer query it is called a non-correlated Subquery.

Question 278 – What are the different Authentication options and Authentication Modes in SQL Server?

- Authentication Types - Windows and SQL Server Authentication are the two types available.
- Authentication Mode Types - Windows Authentication mode and Mixed Mode are two modes available

Question 279 – What is a Windows Authentication Mode?

- It allows a user to connect through a Microsoft Windows NT® 4.0 or Windows® 2000 user account.
- When using 'Windows authentication mode' you can only use Windows authentication to connect to SQL Server.

Question 280 – What is a Mixed Authentication Mode(Windows Authentication and SQL Server Authentication)?

- Mixed Mode allows users to connect to an instance of SQL Server using either Windows Authentication or SQL Server Authentication.
- When using 'Mixed mode' you can use either 'Windows authentication' or 'SQL Server authentication' to connect to SQL Server 2005

Question 281 – How to Select an Authentication Mode in SQL Server?

- Microsoft's best practice recommendation is that you use Windows authentication mode whenever possible.
- The main benefit is that the use of this mode allows you to centralize account administration for your entire enterprise in a single place: Active Directory.

- This dramatically reduces the chances of error or oversight.

Question 282 – What is an Identity?

- Identity (or AutoNumber) is a column that automatically generates numeric values.
- A start and increment value can be set, but most 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.
- `SELECT @@IDENTITY` - returns the last IDENTITY value produced on a connection
- `SELECT IDENT_CURRENT('tablename')` - returns the last IDENTITY value produced in a table
- `SELECT SCOPE_IDENTITY()` - returns the last IDENTITY value produced on a connection

Question 283 – What are INNER JOINS in SQL?

- This join returns rows when there is at least one match in both the tables.
- `Select * From Table1 Inner Join Table2 ON table1.ColumnName = Table2.ColumnName`

Question 284 – What is a LEFT OUTER JOIN in SQL?

- This join returns all the rows from the left table in conjunction with the matching rows from the right table.
- If there are no columns matching in the right table, it returns NULL values.
- Example - `Select * From Table1 LEFT Join Table2 ON table1.ColumnName = Table2.ColumnName`

Question 285 – What is a RIGHT OUTER JOIN in SQL?

- This join returns all the rows from the right table in conjunction with the matching rows from the left table.
- If there are no columns matching in the left table, it returns NULL values.
- Example - `Select * From Table1 RIGHT Join Table2 ON table1.ColumnName = Table2.ColumnName`

Question 286 – What is a FULL OUTER JOIN in SQL?

- This join combines left outer join and right after join. It returns row from either table when the conditions are met and returns null value when there is no match.
- Example - `Select * From Table1 FULL Join Table2 ON table1.ColumnName = Table2.ColumnName`

Question 287 – What is an EQUI JOIN in SQL?

It is a specific type of comparator-based join, or the join that uses only equality(only =) comparisons in the join-predicate.

Question 288 – What is a CROSS JOIN in SQL?

- It produces the Cartesian product of the tables involved in the join.
- A cross join that does not have a WHERE clause
- The size of a result set is the no of rows in the 1st table multiplied by the no of rows in the 2nd table.
- Example: is when company wants to combine each product with a pricing table to analyze each product at each price.

Question 289 – What is a SELF JOIN in SQL?

Same table is specified twice with two different aliases in order to match the data within the same table.

Question 290 – What is a NATURAL JOIN in SQL?

- It offers a further specialization of equi-joins.
- The join predicate arises implicitly by comparing all columns in both tables that have the same column-name in the joined tables.
- The resulting joined table contains only one column for each pair of equally-named columns.
- Example - `SELECT * FROM Vendor NATURAL JOIN advance`

Question 291 – What is Log shipping?

- It 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 1 server fails, the other 1 will have the same db and can be used as the Disaster Recovery plan.
- The key feature of log shipping is that it will automatically backup transaction logs throughout the day and automatically restore them on the standby server at defined interval.

- This in effect keeps the two SQL Servers in "synch". Should the production server fail, all you have to do is point the users to the new server, and you are all set.

Question 292 – What is an Index?

- An index is a physical structure containing pointers to the data.
- Indices are created in an existing table to locate rows more quickly and efficiently.
- It's possible to create 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.
- Table scans are sometimes unavoidable, but on large tables, it has a terrific impact on performance.
- A table can have one of the below indexes combinations
 - No Indexes
 - A clustered index
 - A clustered index and many non-clustered indexes
 - A non-clustered index
 - Many non-clustered indexes

Question 293 – How to Create an Effective Index?

- Use the following guidelines to help create efficient indexes
- Create indexes based on use.
- Keep clustered index keys as small as possible.
- Consider range data for clustered indexes.
- Create an index on all foreign keys.
- Create highly selective indexes.
- Consider a covering index for often-used, high-impact queries.
- Use multiple narrow indexes rather than a few wide indexes.
- Create composite indexes with the most restrictive column first.
- Consider indexes on columns used in WHERE, ORDER BY, GROUP BY, and DISTINCT clauses.

- Remove unused indexes.
- Use the Index Tuning Wizard.

Question 294 – What are the Types of Indexes available?

- Clustered Index
- Non Clustered index
- Primary Key index
- Unique index
- Bitmap index
- Hash index
- Function Based index
- B-Tree index
- Virtual index
- Composite index
- Covering index

Question 295 – What is the difference between Primary Key and a Unique Key?

Primary Key	Unique Key
Primary key creates clustered index	Unique key creates non clustered index
Can have only 1 Primary key in a table	Can have many Unique key in a table
It cannot contain NULL values	Can have NULL values, even more than 1 null values

Question 296 – What is the difference between Delete and a Truncate?

Delete	Truncate
Does not reset the identity of the table	Resets identity of the table
Its DML Command	Its DDL Command
It can be rolled back	It cannot be rolled back
Its Slower (Removes row one by one)	Its Faster (Uses Fewer systems)
WHERE Condition can be used	WHERE Condition can't be used

Records entry in transaction log for each deleted row	Removes the data by de-allocating the data pages
Trigger can be activated	Trigger can't be activated

Question 297 – What is a Clustered Index?

A clustered index is a special type of index that reorders the way records in the table are physically stored. The leaf nodes of a clustered index contain the data pages. Clustered index is unique for any given table

Question 298 – What is a Non Clustered Index?

A nonclustered 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.

Question 299 – What is the difference between Clustered and Non Clustered Index?

Clustered Index	Non Clustered Index
can have only one clustered index on a table	we can have as many non-clustered indexes(255)
The leaf level of a clustered index is the actual data	non-clustered index the leaf level is actually a pointer to the data in rows
The RowLocator in Clustered Index is the clusteredIndex key	The row locator in Non Clustered Index is a pointer to the row. ROW ID (RowLocator)= file identifier + page number + row number on the page
It doesn't allow null values	It allow one null values
Assigned for primary key	Assigned for unique key
Cluster index exists on the physical level	They are not created on the physical level but at the logical level
It sorts the data at physical level	It does not sort the data at physical level
A clustered index requires no separate storage than the table storage	requires separate storage than the table storage to store the index information

Question 300 – What is BCP used in SQL?

- BCP (Bulk Copy Program) is a command line utility by which you can import and export large amounts of data in and out of SQL SERVER database.

- To import or export a set of columns WHERE clause can be used with bcp commands and all the conditions can be mentioned in the query to generate the set of rows you want to copy.

Question 301 - What is the use of Having and Where Clause in SQL?

- HAVING is just an additional filter to 'Where' clause.
- First SQL server filters the rows using WHERE conditions and then performs group by on remaining rows and then filters the rows again with HAVING.
- If a condition refers to an aggregate function, put that condition in the HAVING clause. Otherwise, use the WHERE clause.
- You can't use HAVING unless you also use GROUP BY.
- HAVING is typically used in a GROUP BY clause. When GROUP BY is not used, HAVING behaves like a WHERE clause.

Question 302 - What is the difference between Having and Where Clause in SQL?

Having	Where
It applies to summarized rows (summarized 'with GROUP BY)	It applies to rows
Having can used only with the select statement, typically used with group by clause	WHERE clause is used to impose condition on SELECT statement as well as single row function
Used after GROUP BY clause, when it is used without group by it work like where clause	Used before GROUP BY clause

Question 303 - What is the Local Temporary Table?

- They are created using same syntax as CREATE TABLE except table name is preceded by '#' sign.
- When table is preceded by single '#' sign, it is defined as local temporary table and its scope is limited to session in which it is created.

Question 304 - What is the Global Temporary Table?

- Syntax difference between global and local temporary table is of an extra '#' sign. Global temporary tables are preceded with two '#' (##) sign. Following is the definition.
- In contrast of local temporary tables, global temporary tables are visible across entire instance.
- Global temporary tables are dropped when the session that created it ends, and all other sessions have stopped referencing it.

Question 305 - What is the difference between @@Error and @@Rowcount ?

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 306 - What is a Service Broker in SQL?

- It is a unique feature in Microsoft SQL Server 2005 which is used to by internal or external processes to send and receive guaranteed, asynchronous messages by using extensions to Transact-SQL Data Manipulation Language (DML).
- Service Broker not only brings asynchronous, queued messaging to database applications but significantly expands the state of the art for reliable messaging.
- The key components of SQL Server Service broker are
- Queue stores the messages for a particular service.
- Dialog is a conversation between two services.
- Conversation group is a group of related conversations.
- Every conversation belongs to exactly one conversation group.
- Activation specifies a sp that will handle messages destined for a particular service.

Question 307 - What is Database Mirroring in SQL?

- Database mirroring involves two copies of a single database that typically reside on different computers.
- One copy of the database is currently available to clients which are known as the principal database.
- Updates made by clients to the principal database are applied on the other copy of the database, known as the mirror database.

- Mirroring involves applying the transaction log from every insertion, update, or deletion made on the principal database onto the mirror database.
- Database mirroring Increases data protection.
- Database mirroring Increases availability of a database.
- Database mirroring improves the availability of the production database during upgrades.
- Database mirroring cannot be used on system databases.

Question 308 - What is Extended Stored Procedure in SQL?

- An extended stored procedure is a function within a DLL that can be called from T-SQL, just the way we call normal stored procedures using the EXEC statement.
- Question 309 – How to increase the Performance of a Query in SQL?
- Know the performance and scalability characteristics of queries.
- Write correctly formed queries.
- Return only the rows and columns needed.
- Avoid expensive operators such as NOT LIKE.
- Avoid explicit or implicit functions in WHERE clauses.
- Use locking and isolation level hints to minimize locking.
- Use stored procedures or parameterized queries.
- Minimize cursor use.
- Avoid long actions in triggers.
- Use temporary tables and table variables appropriately.
- Limit query and index hints use.
- Fully qualify database objects.

Question 310 - What is an Execution Plans in SQL?

- It helps to check how the query runs background to fetch the data's
- Guidelines considered for execution plan
- Evaluate the query execution plan.
- Avoid table and index scans.

- Evaluate hash joins.
- Evaluate bookmarks.
- Evaluate sorts and filters.
- Compare actual versus estimated rows and executions.

Question 311 – What is Dead Lock?

- A deadlock is a situation where in two transactions wait for each other to give up their respective locks.
- When this happens, the SQL Server ends the deadlock by automatically choosing one and aborting the 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.
- SQL Server detects deadlocks and terminates one user's process.
- Deadlocking can occur with locks, parallelism, threads, and application events.
- Most frequent source of deadlocking is resource locking where the resources are table or index objects.

Question 312 – What is an example of Deadlock?

Transaction A attempts to update table 1 and subsequently read/update data from table 2, whereas transaction B attempts to update table 2 and subsequently read/update data from table 1. In such situations, transaction A holds locks that transaction B needs to complete its task and vice versa; neither transaction can complete until the other transaction releases locks.

Question 313 – How to Resolve Deadlock?

- In SQL Server 2000, the Lock Monitor thread detects the deadlock.
- It uses a periodic detection system, inspecting processes about every 5 seconds to determine if there are any deadlock cycles.
- When it finds one, it automatically chooses one thread as the deadlock victim.
- It then rolls back the victim thread's transaction, cancels its query, and returns error 1205 to its client.
- The Lock Monitor generally chooses the least expensive transaction to roll back.
- You can override this somewhat using SET DEADLOCK_PRIORITY to LOW for a session.

- Whenever both threads have the same DEADLOCK_PRIORITY setting, the Lock Monitor will have to choose one of them as the victim.

Question 314 – What is Live Lock?

- 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.
- This is different than deadlock as in deadlock both the processes wait on each other.

Question 315 – What is an example of Live Lock?

A human example of live lock would be two people who meet face-to-face in a corridor and each moves aside to let the other pass, but they end up moving from side to side without making any progress because they always move the same way at the same time and never cross each other. This is good example of live lock.

Question 316 – What are the options available to Move Database between servers?

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,
- Log shipping,
- INSERT...SELECT,
- SELECT...INTO,

- Creating INSERT scripts to generate data.

Question 317 – What is Replication?

- SQL replication allows not only for copying data between databases, but also copying any database objects as well. Essentially replication performs synchronization between databases. By utilizing SQL replication, you can distribute data to as many remote network locations you need, and you can do that over different types of networks including LAN, WAN, and Internet to name a few.
- Database replication can be done in at least three different ways:
- Snapshot replication: Data on one server is simply copied to another server, or to another database on the same server.
- Merging replication: Data from two or more databases is combined into a single database.
- Transactional replication: Users receive full initial copies of the database and then receive periodic updates as data changes.

Question 318 – What is Union?

- UNION only selects distinct values.
- When using the UNION command all selected columns need to be of the same data type
- A UNION statement effectively does a SELECT DISTINCT on the results set.

Question 319 – What is Union ALL?

- UNION ALL selects all values.
- Union all will not eliminate duplicate rows, instead it just pulls all rows from all tables fitting your query specifics and combines them into a table.
- If you know that all the records returned are unique, use UNION ALL instead, it gives faster results.

Question 320 – List some of the DBCC (Database Console Commands for SQL Server) Commands?

- DBCC CACHESTATS displays information about the objects currently in the buffer cache, such as hit rates, compiled objects and plans, etc.

DBCC DROPCLEANBUFFERS: Use this command to remove all the data from SQL Server's data cache

(buffer) between performance tests to ensure fair testing. Keep in mind that this command only removes clean buffers, not dirty buffers.

- DBCC ERRORLOG: If you rarely restart the mssqlserver service, you may find that your server log gets very large and takes a long time to load and view.

Question 321 - What is a Full Database Backup type in SQL?

With this backup you are backing up the entire database and will be able to restore the entire database from just this backup.

Question 322 - What is a Differential Database Backup type in SQL?

- A differential backup will record all of the data that has changed since the last database backup.
- You must have a database backup in place to use a starting point for your differential backup.
- Differential backups have several limitations including the following:
 - They do not provide point-in-time restore capabilities
 - They may only be restored after a complete database backup is restored
 - They may not be performed on the master database

Question 323 - What is a Transactional Log Database Backup type in SQL?

Transaction log backups record all transactions that have been recorded against the database since the last transaction log backup.

Question 324 - What is a Log Database Backup type in SQL?

- File backups involve backing up individual files within a database.
- This can be the quickest way to restore, but it also has a lot of overhead associated with it.
- You must keep track of your file backups as well as use these file backups in conjunction with transaction log backups.

Question 325 - What are the advantages of NoLock in SQL?

Applying nolock in select statement will increase concurrency and performance in fetching.

Question 326 - What are the disadvantages of NoLock in SQL?

It doesn't consider the transaction scenario, even if the transaction is under process not yet committed or rollback it will fetch the current record.

Question 327 - What is GRANT command in SQL?

- It is a command used to provide access or privileges on the database objects to the users.
- Syntax - GRANT privilege_name ON object_name TO {user_name |PUBLIC |role_name}
- Example - GRANT SELECT ON Table1 TO User1

Question 328 - What is REVOKE command in SQL?

- It removes user access rights or privileges to the database objects.
- Syntax - REVOKE privilege_name ON object_name FROM {user_name |PUBLIC |role_name}
- Example - REVOKE SELECT ON Table1 TO User1

Question 329 - What is Privileges command in SQL?

- It defines the access rights provided to a user on a database object. There are of two types.
- System privileges - This allows the user to CREATE, ALTER, or DROP database objects.
- Object privileges - This allows the user to EXECUTE, SELECT, INSERT, UPDATE, or DELETE data.

Question 330 - What is Cascade command in SQL?

- While dropping a parent table , it is required to drop the corresponding child table also.
- Parent table cannot be deleted if there is any child for it. So the cascaded drop or delete helps in deleting the child and then the parent.
- When you give a "drop table" command, it will not drop the table that has dependencies unless you include the "cascade" command in the statement.

Question 331 – What is the use of RESTRICT Keyword in SQL?

- It specifies that table should not be dropped if any dependencies (i.e. triggers, stored procedure, primary key, foreign key etc) exist.

- Therefore, if there are dependencies then error is generated and the object is not dropped.

Question 332 – What is the use of DML in SQL?

It is abbreviation of Data Manipulation Language. It is used to retrieve, store, modify, delete, insert and update in database. Examples: SELECT, INSERT, UPDATE, DELETE, MERGE, CALL, EXPLAIN PLAN, LOCK TABLE

Question 333 – What is the use of DDL in SQL?

It is abbreviation of Data Definition Language. It is used to create and modify the structure of database objects in database. Examples: CREATE, ALTER, DROP, TRUNCATE, COMMENT, RENAME

Question 334 – What is the use of DCL in SQL?

It is abbreviation of Data Control Language. It is used to create roles, permissions, and referential integrity as well it is used to control access to database by securing it. Examples: GRANT, REVOKE statements

Question 335 – What is the use of TCL in SQL?

It is abbreviation of Transactional Control Language. It is used to manage different transactions occurring within a database. Examples: COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION statements.

Question 336 – What is the use of Wildcards in SQL?

- It can substitute for one or more characters when searching for data in a database.
- SQL wildcards must be used with the SQL LIKE operator.
- With SQL, the following wildcards can be used %, -, [charlist] etc.

Question 337 – What is the use of Aggregate functions?

- It perform a calculation on a set of values and return a single value.
- Except for COUNT, aggregate functions ignore null values.
- Aggregate functions are frequently used with the GROUP BY clause of the SELECT statement.

- Aggregate functions can be used as expressions only in the following:
- The select list of a SELECT statement (either a SubQuery or an outer query).
- A COMPUTE or COMPUTE BY clause.
- A HAVING clause.

Question 338 – What is the use of ROLLUP in SQL?

- ROLLUP clause is used to do aggregate operation on multiple levels in hierarchy.
- It work with the "Group By " clause its main functioning comes into existence when we use Group by.
We can get sub-total of row by using the Rollup function.
- When result is return by Group By class first row display grand total or we can say that the main total.
- Syntax:- select firstcolumn, secondcolumn, sum(thirdcolumn) from tablename group by firstcolumn, secondcolumn with rollup order by firstcolumn.

Question 339 – What is the use of CUBE in SQL?

- The CUBE operator generates a result set that is a multidimensional cube.
- It is a result set that contains a cross tabulation of all the possible combinations of the dimensions.
- The CUBE operator is specified in the GROUP BY clause of a SELECT statement.
- The select list contains the dimension columns and aggregate function expressions.
- The GROUP BY specifies the dimension columns and the keywords WITH CUBE.

Question 340 – What are the Differences between CUBE and ROLLUP?

- CUBE generates a result set that shows aggregates for all combinations of values in the selected columns.
- ROLLUP generates a result set that shows aggregates for a hierarchy of values in the selected columns.

Question 341 – What is the use of CUBE Operator in SQL?

The CUBE Operator is useful in generating reports that contain subtotals and totals. There are extensions of the GROUP BY clause.

Question 342 – What is the Difference between ROLLUP and COMPUTE?

- ROLLUP returns a single result set while COMPUTE BY returns multiple result sets that increase the complexity of application code.
- ROLLUP can be used in a server cursor while COMPUTE BY cannot.
- The query optimizer can sometimes generate more efficient execution plans for ROLLUP than it can for COMPUTE BY.

Question 343 – What is the use of COMPUTE in SQL?

- The optional BY keyword. This calculates the specified row aggregate on a per column basis.
- A row aggregate function name. This includes SUM, AVG, MIN, MAX, or COUNT.
- A column upon which to perform the row aggregate function.

Question 344 – What is the use of COMPUTE BY in SQL?

- It allows you to see both detail and summary rows with one SELECT statement.
- You can calculate summary values for subgroups, or a summary value for the whole result set.

Question 345 – What is the use of With TIES in SQL?

- The SELECT TOP N query always return exactly N records, and arbitrarily drops any record that have the same value as the last record in the group.
- The SELECT TOP N WITH TIES query always return N records with any record having the same value as the last record.
- The TOP N PERCENT clause also do the same WITH TIES and Without TIES

Question 346 – What is the use of ALL & ANY operator in SQL?

- Comparison operators that introduce a SubQuery can be modified by the keywords ALL or ANY
Using the > comparison operator as an example, >ALL means greater than every value, In other words, it means greater than the maximum value.
- For example, >ALL (1, 2, 3) means greater than 3. >ANY means greater than at least one value, that is, greater than the minimum. So >ANY (1, 2, 3) means greater than 1.

- Similarly, >ANY means that for a row to satisfy the condition specified in the outer query, the value in the column that introduces the SubQuery must be greater than at least one of the values in the list of values returned by the SubQuery

Question 347 – What is the use of Master Database in SQL?

This database holds information for all databases located on the SQL Server instance. SQL Server cannot start without a functioning master database.

Question 348 – What is the use of MSDB Database in SQL?

This database stores information regarding database backups, SQL Agent information, DTS packages, SQL Server jobs, and some replication information such as for log shipping.

Question 349 – What is the use of TEMPDB Database in SQL?

This holds temporary objects such as global and local temporary tables and stored procedures.

Question 350 – What is the use of MODEL Database in SQL?

This is essentially a template database used in the creation of any new user database created in the instance.