1: What is ADO.NET Entity Framework Network?
Answer: ADO.NET Entity Framework is an Object/Relational Mapping (ORM) framework. It allows programmers to work with relational data as domain-specific objects without writing rigorous codes for data access. Using the Entity Framework, the programmers can write queries using LINQ, then retrieve and manipulate data as strongly typed objects.

2: What do you understand by the term 'Code First Experience' in relation with the Entity Framework Network?
Answer: The Entity Framework allowing creation of entity models using code and can map to an existing database or generate a database from the model.

3: What is the difference between the ADO.NET and ADO.NET Entity Framework Network?
Answer: ADO.NET is a set of classes that provides data access services for .NET Framework programmers. The goal of present version of ADO.NET is to eliminate the mismatch between the data models and various languages faced by the programmers and thus to increase the level of abstraction for data programming. To achieve this, the present version of ADO.NET includes two techniques, namely the Language-Integrated Query (LINQ) and the ADO.NET Entity Framework. Therefore, the Entity Framework is a part of the ADO.NET family of technologiesthat allows the object oriented mapping of relational data.

4: Is it possible to carry forward the existing applications built on ADO.NET to the Entity Framework?
Answer: Yes, because it is built on the existing ADO.NET provider model.

5: Is it possible to change the mappings between the object model and the database/XML (data storage) schema without changing the application code?
Answer: Yes.

6: What is the purpose of the Entity Data Model wizard?
Answer: It creates an entity data model from a database.

7: What are the basic features of the entity data model?
Answer: The basic features of an entity data model are entities, associations and properties.

8: What would be the extension of the EDM file?
Answer: edmx

9: Is it possible to create an entity model without a pre-existing database?
Answer: Yes, it is possible to create an entity model without a pre-existing database and then generating a database from the model using the Entity Framework.

10: True or False: With Entity Framework the queried data is returned as rows and columns or data table records.
Answer: False; Data is retrieved as objects.

11: How are Changes in data saved?
Answer: As data is returned as objects, when these objects are saved, the changes are saved in the database.

12: How do entities differ from object?
Answer: Entities define the schema of an object but not the behavior of the object.

13: What is the difference between the database model and entity data model?
Answer: The entity data model reflects the business domain, whereas the database model provides the normalized schema designed by the database administrator.

14: What are the two ways of writing queries with the Entity Framework syntax?
Answer: LINQ to Queries and Entity SQL

15: How Entity Framework manages entities that do not inherit from the EntityObject?
Answer: It manages these objects as POCO (Plain Old CLR objects) entities.
16: What is the purpose of Object Services?
Answer: Object Services keep track of any entity object instantiated either as a query result or by creating a new object in code. Object services construct INSERT, UPDATE, or DELETE command for each object added, modified or deleted by comparing the original values with the current values. Object services also pass the current values to any stored procedure, if required.

17: Does Entity Framework support foreign keys?
Answer: Yes

18: What is the purpose of the EntityClient API in Entity Framework?
Answer: It provides the functionalities required for connecting with the database, executing commands, retrieving the query results and rearranging the results to match the entity data model.

19: Is it possible to use Entity Framework with web services and WCF?
Answer: Yes

20: True or False: Entity Framework moves the entity model into three XML files that are used by the EF runtime.
Answer: True

Q: What is ADO.NET Entity Framework or EF?
ANS: ADO.NET Entity Framework is an ORM Framework developed by Microsoft. It is an enhancement of ADO.NET that gives us an automated mechanism to access and store data in the database. We can access database without much more code or programming.
ORM is a tool to store data from domain object to relational database like MS SQL Server without too much coding. It has three parts: Domain Class Object, Relational Database Object and Mapping information on how domain object maps with relational database objects

Q: History of ADO.NET Entity Framework
ANS: The first version of Entity Framework released on 11 August, 2008 with .NET Framework 3.5 Service Pack 1 and Visual Studio 2008 Service Pack 1.
The second version of Entity Framework was released on 12 April 2010 with .NET Framework 4.0
A third version of Entity Framework was released on April 12, 2011. Its name was Entity Framework 4.1.
A refresh of version of Entity Framework 4.1 was released on July 25, 2011.
The version Entity Framework 4.3.1 was released on February 29, 2012.
The latest version is 5.0.0 and is targeted at .NET framework 4.5. But it is also available for .Net framework 4.
The Entity Framework 6.0 is an open source project. Its source code is hosted at CodePlex using Git and licensed under Apache License v2. Like ASP.NET MVC Framework.
What is minimum requirement for Entity Framework applications to run?
The Entity Framework is a component of the .NET Framework. Any application developed by Entity Framework, can run on any computer which has .NET Framework 3.5 SP or greater version.

Q: What are the benefits of using Entity Framework or EF?
ANS: The main and the only benefit of EF is auto-generates code for the Model (middle layer), Data Access Layer, and mapping code. It reduces a lot of development time.

Q: What is Entity Data Model (EMD)?
ANS: Entity Data Model or EMD refers to a set of concepts that describe data structure, regardless of its stored form. It is a bridge between application and data storage and helps

us to work at a conceptual level rather than the actual schema. It uses three key concepts to describe data structure (entity type, association type and property).

Q: What is .edmx file and what it contains?
ANS: An .edmx file (Entity Data Model XML) is an XML file. It defines a conceptual model, a storage model, and the mapping between these models. This file also contains the information that is used by the ADO.NET Entity Data Model Designer to render a model graphically. It contains all the mapping details of how objects map with SQL tables. It is divided into three sections: CSDL, SSDL, and MSL.
Q: What are CSDL, SSDL and MSL sections in an EDMX file?
ANS: CSDL, SSDL and MSL are actually XML files.
CSDL (Conceptual Schema Definition Language) -is the conceptual abstraction which is exposed to the application.
SSDL (Storage Schema Definition Language) - defines the mapping with our RDBMS data structure.
MSL (Mapping Schema Language) -connects the CSDL and SSDL.

Q: What is Entity SQL?
ANS: Entity SQL is a SQL-like language that is used to query in the conceptual models in the Entity Framework. The conceptual models represent data as entities and relationships. Entity SQL allows querying those entities and relationships in a format that is familiar to those who have used SQL.

Q: What is LINQ to Entities?
ANS: LINQ to Entities provides developers to write LINQ queries.

Q: What is the difference between DbContext and ObjectContext?
ANS: DbContext is a lightweight version of the ObjectContext class. DbContext requires a lot less code for the same kind of functionality. ObjectContext EF V4.0 and DbContext EF V4.1

Q: What are the main drawbacks of Entity Framework?
ANS: The main drawbacks of EF are lazy loading. It is EF default setting but we can disable it. Due to this behavior if we are loading large number of records (especially if they have foreign key relations) then it may take long time. So for better performance disable lazy loading for large number of records.

Q: How to load related Entities in EF
ANS: In Entity Framework we can load related data in three ways

Q: What is Lazy Loading?
ANS: Lazy loading is the process to delay the loading of related objects until we need it. Other hand, on demand objects loading rather than loading objects unnecessarily. When objects are returned by a query related objects are not loaded at the same time.
[Fist query will load main object and the related objects will be loaded in second queries.]

Q: What is Eager Loading?
ANS: The opposite of Lazy Loading is eager loading. Eager loading is the process of loading related entities/ objects.

Q: What is Explicit Loading?
ANS: If lazy loading disabled it is still possible to lazily load related entities. For this we need to call the Load method on the related entities.

Q: How can we turn off lazy loading?
ANS: We can turn off lazy loading by setting LazyLoadingEnabled to false.
context.ContextOptions.LazyLoadingEnabled = false;

Q: When we will use lazy Loading?
ANS: When we know that only main object will be used and related objects will not be used.

Q: What is Code First approach in Entity Framework?
ANS: In Code First approach we avoid working with the Visual Designer of Entity Framework. We can say, the EDMX file is excluded from the solution. So now, we have complete control over the context class as well as the entity classes.

Q: What is Model First Approach in Entity Framework?
ANS: In Model First approach, we create Entities, relationships directly on the design surface of EDMX. So in this approach, when you add ADO.NET Entity Data Model, we should select 'Empty Model' instead of 'Generate from database'.

Q: Entity Framework vs LINQ-to-SQL
ANS: Entity Framework and LINQ to SQL are not same. There is some difference. The difference between Entity Framework and LINQ to SQL:

Q: EF has a full provider model. It supports SQL Server, Oracle, DB2, MySQL etc.
ANS: Most of the time L2S classes must be one-to-one with database objects (Customer class can be mapped only with Customer table). Where in EF we can map our domain class with multiple tables using various inheritance strategies like table per type (class) or table per hierarchy of classes etc.
EF supports multiple modeling techniques (code first, model first or database first).
Microsoft Corporation has long term strategy to support and integrate Entity Framework with multiple Microsoft products

Q: What do you mean by Navigation Property?
ANS: A navigation property is an optional property on an entity/object that allows for navigation from one end of an association to the other end. Unlike other properties, it does not carry data.

Q: What are POCO classes in Entity Framework?
ANS: POCO means Plain Old C# Object.

Q: In POCO classes do we need EDMX files?
ANS: Yes we need. Because the context objects reads the EDMX files to do the mapping.

Q: What are T4 templates?
ANS: Text Template Transformation Toolkit or T4 is a template based code generation engine. We can go and write C# code in T4 templates (.tt is the extension) files and those C# codes execute to generate the file as per the written C# logic.
T4 C# code:
<#@ template language="C#" #>
Hello <# Write ("Mr.!") #>
C# output:
Hello
Mr. !

Q: What is the importance of T4 in Entity Framework?
ANS: T4 files are the heart of Entity Framework code generation. It read the EDMX XML file and generates C# behind code. This C# behind code is our entity and context classes. If we developed our project in VS 2012 then we can found .tt files.