1. What is meant by the Entity Framework?

An Entity Framework (EF) is an open-source ORM (Object-Relational Mapper) from Microsoft. It allows developers to work with .NET applications and other domain-specific objects. It helps streamline mapping between different objects within the software by providing tables and columns.

A framework uses objects of domain-specific classes and also provides an option to store and access data in the database automatically. This concept is derived from MVC. However, the Entity framework does not focus on the original database columns and tables used to store data. This makes it easier to operate an abstraction of a higher level while creating new applications or maintaining the existing ones.

2. Why should we use the Entity Framework?

It is a kind of a problematic task when it comes to writing and managing ADO.NET codes. Therefore, Microsoft introduced the Entity Framework to make this tedious task more manageable. Entity Framework reduces a significant amount of code-based tasks by providing relational data in the form of domain-specific objects.

3. What is ADO.NET EF?

ADO.NET Entity Framework is an ORM framework that allows us to work with different relational databases, such as Oracle, MYSQL, SQL Server, DB2, etc. It enables us to work with the data either as objects or entities.

Apart from this, ADO.NET also helps create senior abstract object model on the top of ADO.NET components. This eventually enables us to use advanced level domain objects, such as customers, suppliers, etc.

4. How will you differentiate ADO.NET from Entity Framework?

ADO.NET is fast., It creates several data layer codes., It does not create codes for the data access layers, intermediate layers, and mapping codes by itself. Entity Framework is comparatively slower, It does not create any data layer codes. It automatically creates codes for the data access layers, intermediate layers, and mapping codes. This eventually helps developers cut down the development of work and time.

5. What are the main advantages and disadvantages of the Entity Framework?

Advantages:

It provides a wide range of prototypes that helps write object-oriented codes.
It includes auto migration support that helps configure or manage a database easily and quickly.
It provides many alternate commands that help shorten codes and make a coding job a lot easier.
Disadvantages:

It is considered a slower form of ORM.
If a user does not use raw SQL codes, things might become difficult.
It directs the shape of the entire model when it comes to difficult projects. Additionally, cleaning a model is not possible without an ORM technique.

6. Illustrate the main components of the Entity Framework Architecture.

The following are the main components of the Entity Framework Architecture:

Entity Data Model (EDM): It consists of the three parts, such as storage model, conceptual model, and mapping.

Entity SQL: It is an alternate query language used in Entity Frameworks along with L2E. However, L2E is comparatively more straightforward.

LINQ to Entities (L2E): It is a query language that helps write queries against the object, which further helps to retrieve the entities based on the definitions specifies in the conceptual model.

Entity Client Data Provider: It is defined as the layer that helps convert the L2E queries to SQL queries to be easily understood by the database. Additionally, it can interact with the ADO.NET data provider to transfer or retrieve data from different databases.

Net Data Provider: It is another layer that helps interact with the database by using standard ADO.NET.

Object Service: It is an entry point into the database used to access and send back the data when needed. It helps convert the data coming from an entity client data provider into an entity object structure.

7. What are the primary functions of the Entity Framework?

The following are the primary functions of EF:

It helps map domain classes to the database schema translates.
It keeps tracks of changes in the entities.
It helps execute LINQ queries to SQL.
It stores the changes stats to the database

8. What is defined by the migration in context with the Entity Framework? Also, name its types.

Entity Framework contains a migration tool that automatically helps users update the entire database whenever there is a change in the model. The tool updates the database without losing the stored data or any database objects.

There are mainly two types of migration in Entity Framework:

Automated Migration
Code-based Migration

9. What are the processes used to load related entities in the Entity Framework?

We can use any of the following processes to load related entities in the Entity Framework:

Lazy Loading: This process delays the loading of related objects until there is a requirement of them. Lazy loading only returns objects needed by the user, and all other related objects are only returned when required in the process.

Eager Loading: This process mainly takes place when we query for an object. Eager loading returns all the related objects. Additionally, all the related objects are automatically loaded with the parent object.

Explicit Loading: This process only occurs when we want to use lazy loading, even when we have already disabled lazy loading. To process explicit loading, we are required to call the relevant load method on the related entities explicitly.

10. What are the different types of inheritance supported in Entity Framework?

There are mainly three types of inheritance supported in Entity Framework, such as:

Table per Hierarchy (TPH): This type of inheritance depicts a single table for the entire inheritance hierarchy class. Besides, the table consists of a discriminator column that helps distinguish different inheritance classes. It is the default inheritance mapping technique in Entity Framework.

Table per Type (TPT): This type of inheritance depicts a separate or specific table for all and each domain class.

Table per Concrete Class (TPC): This type of inheritance depicts a single table for a single concrete class, but it does not include the abstract class. Therefore, if an abstract class is inherited in various concrete classes, then every concrete class table will have the same properties as the abstract class.

11.) What are the primary parts of the Entity Data Model?

There are mainly three parts of the Entity Data Model, such as:

Storage Model
Conceptual Model
Mapping

12. What is meant by a model in context to Entity Framework?

A model is nothing but a class mainly used to represent the data. In context to EF, a model represents the data from a table inside the existing database.

Example: The following codes display the basic customer model:

```
public class Customer
{
  public int ID { get; set; }
  public string Name { get; set; }
  public DateTime JoinDate { get; set; }
}
```

13. How will you define the Conceptual Model?

The conceptual model is usually defined as the model class that consists of relationships. This type of model remains independent of the database structure.

14. How will you explain the Storage Model?

The storage model is usually explained as the database design model that consists of database tables, stored procs, views, and keys with relationships.

15. What do you understand by the EDM? Also, list the process for creating EDM.

EDM in EF is a short form of 'Entity Data Model'. It is defined as the entity-relationship prototype that helps assign basic prototypes for the data utilizing various modeling procedures. Additionally, it is referred to as a collection of core principles that define data that disregards its aggregated form. In short, it is nothing but a simple connection formed between the database and the prototype.

The following are the steps that help create an Entity Data Model:

First, we need to right-click on the project's name given in the solution explorer tab.

Next, we are required to click on 'Add a new item from the menu'.

After that, we are required to select the ADO.NET Entity Data Model arrangement or any template.

Finally, we must specify a name for the model and click on the 'ADD' button.

That is how we can create EDM in Entity Framework

16. What is meant by dbcontext and dbset?

DbContext is referred to as a class in the Entity Framework API that helps create a communication between the database and the domain/entity class. In simple terms, it is a class mainly used for communicating with the database.

DbSet is also referred to as a class that helps represent an entity set for different operations, such as creating, updating, reading, and deleting. The context class in DbContext must include the DbSet type properties for all such entities that help join database tables and views.

17. What are the different types of approaches used in Entity Framework?

There are mainly three types of approaches used in Entity Framework, such as:

Model First Approach
Code First Approach
Database First Approach

18. How will you define the Code First approach and the Model First approach? Also, enlist their advantages.

Code First Approach: In Entity Framework, the Code First approach is mainly used to create a model and its relationships using classes, which further help create a database. This enables developers to work in an object-oriented manner without thinking about the structure of the database. In this approach, the developers first write POCO classes and then create the database with the help of these POCO classes. Most developers following Domain-Driven Design (DDD) technique use the Code First approach.

Advantages of Code First Approach:

It allows developers to decide the database structure according to business objects, making it more beneficial for smaller applications.
It enables developers to decide which classes should be serialized. It also allows us to specify the collection to eager load.
Model First Approach: On the other side, the Model First approach is used to create model classes and their relationships using ORM. Once the model classes and relationships are created successfully, the physical database is created with the help of these models.

Advantages of Model First Approach:

It offers flexibility for designing the Entity Models separately and provides options to improve them in further stages.

It does not use many databases because we can create model classes by drawing them using the EDMX designer.

19. Define the database first approach.

In terms of the Entity Framework, the Database First approach is the method of generating an entity model from the available database. The primary motive of this approach is to lower the number of codes to be written. This approach mainly helps create domain and context classes based on the existing classes.

20. Which, according to you, is the best approach in the Entity Framework?

There is no particular approach that can be referred to as the best approach in Entity Framework. The selection of the development approach primarily depends on the project requirements and the project's types. If there is the database's existence, then it is good to use the Database First approach. If there is no database and the model classes, then the Model First approach is the best selection. If there is the availability of the domain classes, the Code First approach is the most suitable choice.

21. Define the .edmx file in the context of Entity Framework.

In the context of Entity Framework, a .edmx file is a simple XML file that may contain storage models, conceptual models, and the mapping between them. It contains all the mapping information of how object maps with SQL tables. Besides, it can include information required by ADO.NET Entity Data Designer to render a model graphically.

22. What is meant by database concurrency and how to handle it?

EF's database concurrency is defined as the scenario when multiple users access and modify the same data in the same database simultaneously. The systems that protect the consistency of data in such situations are known as concurrency controls.

The database concurrency is usually handled by implementing optimistic locking. To implement the locking, we must first right-click on the EDMX designer and then set the concurrency mode to Fixed. After making this change, we will get a positive concurrency exception error if there is a concurrency issue.

23. What are the different types of properties supported in Entity Framework?

There are mainly three types of properties supported in Entity Framework, such as:

Navigational Property
Complex Property
Scalar Property

24. How will you define Mapping in Entity Framework?

The mapping in Entity Framework is defined as the information that explains how the conceptual models are mapped to the storage models.

25. What do you understand by LINQ to Entities?

LINQ to Entities (L2E) is defined as one of the popular query languages in Entity Framework. L2E mainly helps write queries against the objects to retrieve entities based on the conceptual model's definitions.

26. What is the role of Pluralize and Singularize in Entity Framework?

In Entity Framework, Pluralize and Singularize are primarily responsible for assigning meaningful naming conventions to objects. We can access this feature while adding a .edmx file. While using this feature, the Entity Framework will assign to Singular or Plural coding conventions. An extra 's' is added while giving convention names if there is more than one record within the object.

27. What are the methods used for executing plain SQL in Entity Framework?

The following are the methods used for executing plain SQL in Entity Framework:

SqlQuery()

Database.SqlQuery()
Database.ExecuteSqlCommand()

28. What do you understand by the SQL injection attack?

SQL injection attacks involve attackers stealing confidential information from an organization's database. This type of attack is usually done by adding SQL commands to the existing applications' coding based on SQL statements. This is an application-layer attack, meaning front-end attack, as attackers use and modify existing code, which is improperly coded.

SQL injection attack is usually inserted using the fields that are available for user input. It occurs as these fields allow SQL statements to be pass through it and query the database directly. The SQL injection problem is a common problem with ADO.NET data service queries. Besides, Entity Framework is an injection safe because it creates parameterized SQL commands every time to secure the database from SQL injection. Also, the developers should never combine user inputs with Entity SQL command text. This will add extra layers of security.

29. What is meant by a Dapper?

Dapper is a simple micro ORM that helps map the native query's output to a domain or any c# class. It is a powerful system used for data access in the .NET world. Dapper was developed by the StackOverflow team and released as an open-source project. It consists of a NuGet library that can be easily inserted into any.NET project based on database operations.

30. What can we do to improve the performance of the Entity Framework?

We can use the following ways to improve the performance of the Entity Framework:

We can use compiled queries whenever required.
We must avoid the use of Views and Contains.
We can disable and alter tracking for the entity when it is not needed.
We can Debug and Optimize the LINQ query.
If not obligatory, we must try to evade fetching all the fields.
We should retrieve only the desired number of records when binding data to the grid.

31. What other O/RMs you can use with .NET based applications?

The following O/RMs, you can use with .NET based applications:

Entity Framework 6.x

Entity Framework Core

Dapper

N Hibernate

32. What is Micro O/RMs?

A Micro ORM is architected to focus on the most important task of working with database tables instead of creating, modifying the database schema, tracking changes etc. EF 6.x and EF Core both are O/RMs since they provide a full set of features.

33. What are POCO classes?

The term POCO does not mean to imply that your classes are either plain or old. The term POCO simply specifies that the POCO classes don’t contain any reference that is specific to the entity framework or .NET framework.

Basically, POCO (Plain Old CLR Object) entities are existing domain objects within your application that you use with Entity Framework.

34. What is the proxy object?

An object that is created from a POCO class or entities generated by the Entity Framework to support change tracking and lazy loading, is known as a proxy object.

There are some rules for creating a proxy class object:

The class must be public and not sealed.

Each property must be marked as virtual.

Each property must have a public getter and setter.

Any collection navigation properties must be typed as ICollection <T>.

35. What are the various Entity States in EF?

Each and every entity has a state during its lifecycle which is defined by an enum (EntityState) that have the following values:

Added

Modified

Deleted

Unchanged

Detached

36. What are different types of inheritance in Entity Framework?

Inheritance in the Entity Framework is similar to inheritance for classes in C#. In Entity Framework, you can map an inheritance hierarchy to single or multiple database tables based on your requirements. EF supports three types of inheritances:

Table-per-Hierarchy (TPH)

Table-per-Type (TPT)

Table-per-Concrete-Type (TPC)

37. What is Migrations History Table?

In EF6, Migrations history table (__MigrationHistory) is a part of the application database and used by Code First Migrations to store details about migrations applied to a database. This table is created when you apply the first migration to the database. This table stores metadata describing the schema version history of one or more EF Code First models within a given database. In EF 5, this table was a system table when you use the Microsoft SQL Server database.

In EF 5, this table was a system table when you use the Microsoft SQL Server database.

38. What is automatic migration?

lEntity Framework supports automatic migration so that you don't need to migrate model changes manually. So, when you will run the application, it will be handled by the EF.

39. What is ObjectSet?

ObjectSet a typed entity set which is used to perform create, read, update, and delete operations on a particular entity. ObjectSet is can only be created from an ObjectContext instance. ObjectSet does not support the Entity SQL methods.