By OnlineInterviewQuestions.com

Entity framework interview questions

Q1. Explain what is Entity Framework?

Entity Framework happens to be an open-source ORM or object-relational mapping framework utilized for ADO.Net. It is a set of mechanization that assists in the improvement of data-oriented software administration. This helps the developers to function with data in the pattern of domain-specific properties and objects like customers and client addresses, sans having to worry themselves regarding the fundamental database columns and tables where the data is collected.

With the help of Entity Framework, designers can function at an optimal level of absorption when dealing with data while creating and maintaining data-oriented applications with few codes, unlike traditional applications.

Q2. What is new in entity framework 7?

The new features in entity framework 7 include the following:

- **New platforms:** .NET along with the other distinct technologies like WPF, ASP.NET, WinForms is supported by Entity Framework 7.
- **Extensible and lightweight:** One can extend and replace the extensible.
- **Microsoft** has officially ceased EDMX in Entity Framework 7.
- **Entity Framework 7** supports non-relational and relational data collection.
- **EF7** offers a simpler SQL examination for the most usual scenarios than EF6.
- **Entity Framework 7** permits one to identify added innovative keys within one's entities in supplementation to the chief key. One can utilize then these different keys as the selection of unfamiliar key relationships. An innovative constraint is initiated for every alternate or different key in the prototype.

Q3. How to check installed version of Entity Framework?

To check, If you open up the reference file and detect system.entity.data, click the module and then inspect the runtime version numerical in the Properties explorer and you can witness the subversion also. Another alternative way to check is mentioned below:

Open PMC or Package Manager Consoles in Visual Studio while typing Get-packages at the cue. The foremost line portrays the Entity Framework while the list portrays the version of the Entity The framework has been installed.

Hence, follow the below-mentioned steps:

- Unbar the references in one's project.
- Right-click the Entity Framework.
- Select property.
- See the version row and notice the updated version of the Entity Framework.

Q4. List some pros and cons of Entity Framework?

There are several advantages and disadvantages to Entity Framework.

The pros of the Entity Framework are:

- It offers Auto Migrations and thus helps to modify and set up the database as it may permit the database to experience changes made in the model codes.
- Can shorten codes.
- It offers excellent prototypes. This enables the developers to write codes that happen to be object-oriented.

The cons of Entity Framework are:

- Decelerate performance of Entity Framework.
- Command the model form.
- It can complicate or perplex things and happens to be time-consuming.
- The syntax is complex and complicated.
- Unavailable for all RDMS.

Q5. Enlist the types of inheritance that are supported by Entity Framework?

There are three kinds of inheritance that are supported by Entity Frameworks as follows:

- **TPT or Table per Type:** This technique depicts a separate or distinct table for all and every domain class.
- **TPC or Table per Concrete Class:** This technique depicts a single table for a lone concrete class, but excluding the abstract class. So, if one inherits the abstract class in innumerable concrete classes then the features of the abstract classes become a fragment of every table of the concrete class.
- **Core Supporting Inheritance:** For mapping, there are limited patterns of class hierarchy that may be depicted in a relational database. The most favored ones include TPT or Table per Type and TPH or Table per Hierarchy.

Q6. What is the difference between ADO.Net and Entity Framework?

Entity Framework: Entity Framework happens to be the ORM model and utilizes LINQ to read databases and codes are generated. Entity Framework lowers development cost and time. It also helps the designers to visually

develop models and delineating of the database. Moreover, Entity Framework offers the power to program a conceptual prototype.

ADO.Net: ADO.Net happens to be ORM but it creates an elevated abstract object prototype over ADO.Net components. ADO.Net functions on disjunction architecture. Data happens to hold as XML. ADO.Net is extremely swift and data may be collected in data sets. ADO.Net also abstracts esoteric programming which is necessary for accessing a database.

Q7. [How to return multiple result set in Entity Framework?](#)

One can use the following pattern to return multiple result sets:

- **One** can make the database in the SQL Server or Local DB and occupy the tables by implementing the comprising script file, STOREDB.sql, utilizing SQL Server Management Studio or SSMS or Visual Studio.
- **Implementation** of the script will add 2 collected processes into the database. Choose those procedures from the functions list in the Add tab.
- **Right-click** on any blank portion on the Entity Framework developer and choose the Model browser. Now alter the **GetProductCM_Result** to **Product_SprocResult**. Save the StoreDBModel.edmx file. Discover the FunctionImportMapping nodes within **edmx:Mapping/../En**

Q8. [What is the use of .edmx file in Entity Framework?](#)

.edmx file happens to be an XML file that explains an **EDM** or **Entity Data Model**, defines delineation between database and EDM, and describes the selected database schema. It also comprises information that is utilized to supply the prototypes graphically by the **ADO.Net** Entity Data Model Designer.

Hence, it includes every delineation detail regarding the process of object mapping with **SQL tables**. It also has three divisions **CSDL**, **MSL**, and **SSDL**. This section aid you in the mapping of your RDMS data formation and explain relationship, function, and entities respectively.

Q9. [List major components of Entity Framework Architecture?](#)

The major components of **Entity Framework Architecture** are:

- **Entity SQL:** these queries happen to be internally rendered to data store supported SQL queries.
- **ADO.Net Data Provider:** utilizes standard ADO.Net to communicate with the database.
- **LINQ to Entities:** aids you to communicate opposed to the object prototype.
- **The Object Services Layer:** aids in inheritance, merging data, identity resolution, etc.
- **Entity Client Data Provider:** converts Entity SQL or LINQ-to-Entities into an SQL query.
- **The Entity Data Model:** abstracts the relational or logical schema and reveals the fundamental schema of data utilizing conceptual prototype, storage model, and mapping model.

Q10. What is Entity Data Model? How to create EDM?

EDM or the **Entity Data Model** happens to be an expanded style of the Entity-relationship prototype which states the fundamental prototype of the data utilizing several modeling procedures. It also depicts a set of fundamentals that explains data formation disregarding its collected form. EDM is hence the connection between the database and the prototype.

To create an Entity Data Model, one can

- **Right-click** on one's project in the Solution Explorer window
- **Choose Add>**New Item from the menu option.
- **Choose** the ADO.Net Entity Data Model arrangement or template.
- **Add** name and click on the Add button.

Q11. Explain lazy loading in entity framework?

Lazy Loading is a process of collection of entities or a single entity that is loaded spontaneously for the first time from the database and property that refers to those entities or entities can be accessed. In other words, lazy loading means delaying the loading of related data, until it is specifically requested. While loading the **POCO type** of entities, lazy loading is produced by generating derived proxy type instances. Lazy loading then overrides the virtual properties to insert the hook of the loading.

If the configuration is defined as the default and proper instruction is not given to the entity framework in a query, then it will return the lazy loading only.

Q12. What is the difference between Model First and code first approach?

The Difference between **Model-First** and **code first approach.**

In code first approach, the entity framework does not include any configuration file like **edmx file** for storing the schema of the database. Code first approach is used for the fast development of the codes and developers have full control of the entire entities.

In model first approach there is no existing database that offers the designer to create a conceptual data model. It **implies .edmx file** to store the information of mapping and model. In this approach, manual changes to the database cannot be done as the model defines the database. The model first approach goes well with the power pack database Generation installed.

Q13. How to connect entity framework to sql server?

The entity framework can be connected to **the SQL Server in the following way**:

- A console application is to be created
- Create a class by right click on the application
- Create the variables of the class and set the required properties
- Save the class Add one more class in the same way
- Right-click and create a new folder for generating the Framework Dynamic Link Library (DLL)
- The folder can be renamed as the developer wants
- Save the application before adding any framework
- Right-click the Program.cs file and select the required properties for copying the program's address
- Right-click on the window of the program and copy the required address for adding the DLL framework
- After copying the address, put it in the c: drive bar and search for the folder previously added
- After finding the folder, add the Entity framework DLL
- Right-click on the references and add it to browse the DLL framework
- The application configuration file is added
- The DB path and database name should be declared using the App. Config file
- In the main program, declare the context and objects
- Press F5 to execute
- In the SQL server, the database has been created successfully

Q14. How to prevent sql injection in entity framework?

The **SQL injection** is a technique for code injection that can attack data-driven applications and can destroy the database. The SQL injection is one of the most commonly used web hacking procedure that injects malicious codes in the SQL statements through the input of the web page.

LINQ helps to prevent SQL injection in the entity framework. This happens due to the passing of all data to the database through the QL parameters. LINQ queries are not susceptible to the attacks by SQL injection as they are made by implying concatenation or manipulation of the string.

Q15. What are complex types in Entity Framework?

Complex types are scalar properties that consist of different types of entities that enable the scalar elements to be organized within the required entities. Complex types do not include keys and thus exist independently. It cannot participate actively in the associations and do not have navigation properties. The properties of the complex type cannot be null. Scalar properties can be null of the complex objects.

A complex type cannot inherit anything from the other complex types. The complex type must be defined as the class. When DbContext.DetectChanges is called, Entity framework detects changes on an object of the complex type in the members.

Q16. What is Dapper?

Dapper is a microframework for a simple object mapper that maps the output of the native query to a domain or a **c# class**. Dapper is a high-performance system for data access that is built by StackOverflow and is released as an open-source. It is very easy to fire a query in SQL against the database and to map the result into the C# domain class.

It extends the **IDbconnection** interface that is available in the System. Data namespace. Dapper includes many methods for the extension of the data access and mapping the result defined under the class SqlMapper found under the namespace of the dapper.

Q17. What is fluent API in MVC?

Fluent API is a way of specifying the configuration of the model that covers everything executed by the annotations of the data. Fluent API and data annotations may be used together, but more preference is given by the Code First to **Fluent API**, then data annotations and finally to the default conventions. Fluent API is a way to configure the domain classes.

The Fluent API of the Code First is accessed by overriding the **OnModelCreating** method of the derived DbContext. Fluent API offers more configuration functionality than the DataAnnotations. Fluent API allows the user to configure the properties of the entities.

Q18. What is the proxy object?

Proxy Objects act as an intermediary between an accessible object and the client. Proxy Objects monitors the lifespan of the accessible objects and forwards the calls to the objects when it is not destroyed. Proxy objects check the availability of accessible objects whenever a client calls an **IAccessible property** to know proper information about the object.

The proxy object passes the requests of the client to the accessible object. When an accessible object is not present or is destroyed, an error is returned by the proxy object. The proxy object is always clear to the client.

Q19. How to turn off lazy loading in entity framework?

Lazy loading can be turned off for a particular context or an entity. To turn off the lazy loading for a specific property, do not convert it to the virtual. To disable the lazy loading for the entire entities in a context, the configuration property must be set to false. Now let us see the ways for turning off the lazy loading.

- Context.configuration.ProxyCreationEnabled should be true. Context.Confiuration.LazyLoadingEnabled should be made true.
- Navigation property should not be defined as virtual and public. The context will do the lazy loading if the property is defined as virtual.

Q20. What is the use of Migrations History Table in Entity Framework?

Migration History Table is utilized by Code First Migrations to collect details regarding migrations that are required to apply to the Database. By revert, the denomination of the table within the Database happens to be **_MigrationHistory** and it is made during the application of the first migration to the Database. In Entity Framework 5 it was a method table for the application being utilized in the Microsoft SQL Server database. However, this utilization of it has been altered in Entity Framework 6 and Migration History Table is no more a salient system table.

Q21. How to achieve eager loading in Entity Framework?

Eager loading happens to be a procedure where a query for a single kind of entity also charges related entities as a fragment of the query. Eager loading can be attained by the utilization of the Include process. It depicts that appealing related data to be sent back along with query outcomes from the Database. There is only a lone communication made to the data origin, a huge amount of data is sent back in the starting query. For instance, while querying students, eager-load their registration. The pupils and their registrations will be recovered in an only or single query.

Q22. What is the importance of T4 in Entity Framework?

T4 or Text Template Transformation Toolkit happens to be a template established code generation mechanism. T4 files are the core of the Entity Framework code creation. So, the T4 code templates interpret the EDMX XML files and create a C# supporting code. This C# supporting code is one's context classes and entity and nothing more. Hence, the Entity Framework aids in the manipulation and extension of the code creation utilizing the T4 template. Thus, the T4 template happens to be a blending of control logic and text blocks that can be created into a text file.

Q23. What are the advantages of Model First Approach?

The **Model First Approach** offers the choice of creating relationships and entities directly on the developing interface off the **EDMX** and later one can execute it. It is a better natural approach to operate in comparison to the Database First Approach specifically during the development of the movement of data in the starting phases of any new project. Utilizing it a designer may initiate operating with the Model of the Database and creation of entities that yield logical sense inapt regarding their process of being stored in the Database concerning the tables.

Q24. Enlist some entity states supported in Entity Framework?

An Entity may be in **one of the five states** mentioned below:

- **Unchanged:** The Entity happens to be traced by the factors and subsist in the Database. Its property worth remains unchanged from the worth of the Database.
- **Deleted:** The Entity happens to be traced by the factors and subsist in the Database but remains spotted for deletion from the Database for the later period when SaveChanges is summoned.
- **Modified:** The Entity happens to be traced by the factors and subsist in the Database and all or some of its property worth remains modified.
- **Added:** The Entity happens to be traced by the factors but doesn't subsist in the Database.
- **Detached:** The Entity happens to be untraceable by the factors.

Q25. Explain difference between Add and AddRange?

Difference between Add and Add Range in Entity Framework:

Add	Add Range
Add is used to insert one element at a time in a collection.	AddRange is used to add multiple elements.
The add method inserts the item at the end of a collection.	AddRange method is used to insert a set of records into a collection.
Add method is implemented for a new element	AddRange is employed in a collection that is already created

The **addRange** method is also used to add an array of nodes that are previously created in the collection.

Q26. What is dbcontext and dbset in Entity Framework?

The Dbcontext class happens to be an essential fragment of the Entity Framework. Dbcontext **API** in maximum cases selects or chooses to simplify one's communication with Entity Framework. It lowers the quantity of procedures and attributes that one requires to access frequently utilized tasks. Dbcontext also handles the entity articles throughout run time including occupying items with data from databases and change tracking.

The Dbset class portrays an entity set that may be utilized to **read**, **update**, **delete**, and create operations. An entity set is explained as a set of entities of a similar entity kind. All Entity kinds surely require revealing the dB set attributes to be able to engage in the **CRUD Operations**.

Q27. What are POCO classes in Entity Framework?

A POCO entity is a class that is independent of any framework-related base class. It is the same as any distinct normal **.NET CLR class**, the reason behind it being called **"Plain Old CLR Objects"**. It is also called persistence-ignorant objects. It assists the maximum of the similar LINQ queries as Entity Object obtained

entities. Dynamic Proxy Entities happens to be a runtime proxy class that envelopes the POCO entity. It permits lazy loading. However, A POCO class should not be abstract and sealed. But should be announced with communal access.

Q28. What is change tracker in Entity Framework?

By delinquency Entity Framework traces alterations of the filled entities throughout the lifespan of the context. A change tracker traces alteration when one adds **new records**, **deletes** or **updates** the existing records. It offers one all the details and information regarding updated entities being traced by the context and all the alterations are collected by the Dbcontext level.

Hence, **the change tracker** offers access to alter tracking operations and information for entity instances of the context which is being traced. Instances of these classes are usually taken from change tracker and which are not developed to be directly created in one's application codes.

Q29. What is DbContext in Entity Framework?

This class plays an integral role in the Entity Framework. An instance of **DbContext represents** a session with the database which can be used to **query** and **save** instances of your entities to a database. **DbContext** is a combination of the **Unit Of Work** and **Repository patterns**.

Q30. How can we do pessimistic locking in Entity Framework?

Pessimistic locking presumes that collision or concurrency issues will occur, so a bar is set on the records and later the data is reconditioned. One can execute pessimistic locking by identifying **"IsolationLevel"** in SQL server collected procedures, **AFO.Net level** or **via utilizing proceeding** scope object.

There are **four types** of locks one can acquire:

1. **Exclusive:** when one wants to modify data without wanting anyone else to read the proceedings.
2. **Shared:** when one wants to read-only without wanting to update any distinct transactions.
3. **Intent or demand locks:** it is utilized to lock hierarchy.
4. **Update lock:** when one wants to update procedures or operations which moves via innumerable phases before the real occurrence of updating.

Please Visit OnlineInterviewquestions.com to download more pdfs