



InterviewBit

Entity Framework Interview Questions



To view the live version of the page, [click here](#).

© Copyright by Interviewbit

Contents

Entity Framework Interview Questions for Freshers

1. Explain the advantages of the Entity Framework.
2. Describe some of the disadvantages of the Entity Framework.
3. What are the features of the Entity Framework?
4. What are the main components of Entity Framework Architecture?
5. Explain different parts of the entity data model.
6. Explain what the .edmx file contains.
7. What do you mean by migration? Write its type.
8. What are different types of Entity framework approaches?
9. Which according to you is considered the best approach in Entity Framework?
10. What do you mean by the term navigation property in the entity framework?
11. What are different entity states in EF?
12. Write the importance of the T4 entity in Entity Framework.
13. Explain CSDL, SSDL, and MSL sections in an Edmx file?
14. Explain the ways to increase the performance of EF.
15. Write some XML generation methods provided by the dataset object.
16. What do you mean by the migration history table in Entity Framework?
17. Explain how EF supports transactions.
18. What do you mean by Deferred Execution in EF?

Entity Framework Interview Questions for Experienced

Entity Framework Interview Questions for Experienced

(.....Continued)

19. Write difference between LINQ and Entity Framework.
20. Write the steps to retrieve data from database using Entity Framework in MVC.
21. Explain the term dbcontext and dbset.
22. Difference between ADO.Net and Entity Framework.
23. Explain the role of Pluralize and Singularize in the entity framework.
24. What is the difference between Dapper and Entity Framework?
25. Explain POCO Classes in EF.
26. In Entity Framework, what are the ways to use stored procedures?
27. Explain database concurrency and the way to handle it.
28. What are different types of loading available to load related entities in EF?
29. What do you mean by lazy loading, eager loading and explicit loading?
30. What are the pros and cons of different types of loading?
31. Write different types of inheritance supported by Entity Framework.
32. Explain Complex Type in Entity Framework.
33. What do you mean by Micro ORM?
34. Explain EF Data access Architecture.
35. What do you mean by SQL injection attack?
36. What is the best way to handle SQL injection attacks in Entity Framework?
37. Explain the ObjectSet in EF.
38. Write the namespace that is used to include .NET Data provider for SQL server in .NET code.

Entity Framework Interview Questions for Experienced

(.....Continued)

39. Explain EDM and write the process to create it.
40. What do you mean by DbEntityEntry Class in EF?

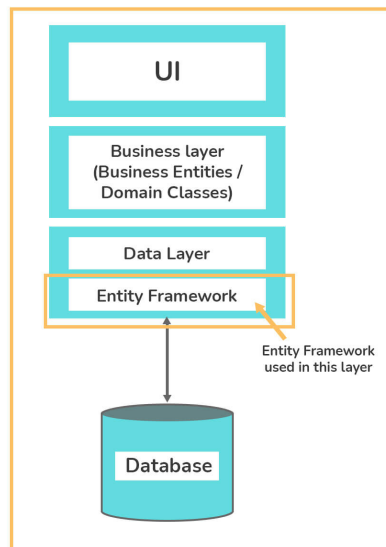


Let's get Started

What is Entity Framework?

In [.NET](#) applications (desktop and web), ADO.NET is an old and powerful method for accessing databases and requires a lot of coding. Simply put, all of the work is done manually. Microsoft has therefore introduced Entity Framework to automate the database technique.

Entity Framework (EF), a Microsoft-supported open-source ORM (Object-Relational Mapper) for .NET applications, is a set of utilities and mechanisms that works towards optimizing data-driven applications. This is considered one of the most important concepts in the MVC model. With the help of tables and columns, developers are able to streamline mapping between various objects in software applications. It makes use of objects of domain-specific classes but ignores the database tables and columns that are used to store the data. Designers can utilize EF in order to maintain and develop data-oriented applications with little coding and a high level of absorption when dealing with data, unlike traditional applications.



The EF fits between the business entities (domain classes) and the database, as shown in the above figure. Additionally, it retrieves data from the database and converts it to business entity objects automatically, and also saves data stored in business entity properties.

Entity Framework Interview Questions for Freshers

1. Explain the advantages of the Entity Framework.

Entity Framework has the following advantages:

- With its excellent prototypes, it is possible to write object-oriented programs.
- By allowing auto-migration, it is simple to create a database or modify it.
- It simplifies the developer's job by reducing the code length with the help of alternate commands.
- It reduces development time, development cost, and provides auto-generated code.
- A unique syntax (LINQ / Yoda) is provided for all object queries, whether they are databases or not.
- It enables the mapping of multiple conceptual models to a single storage schema.
- Business objects can be mapped easily (with drag & drop tables).

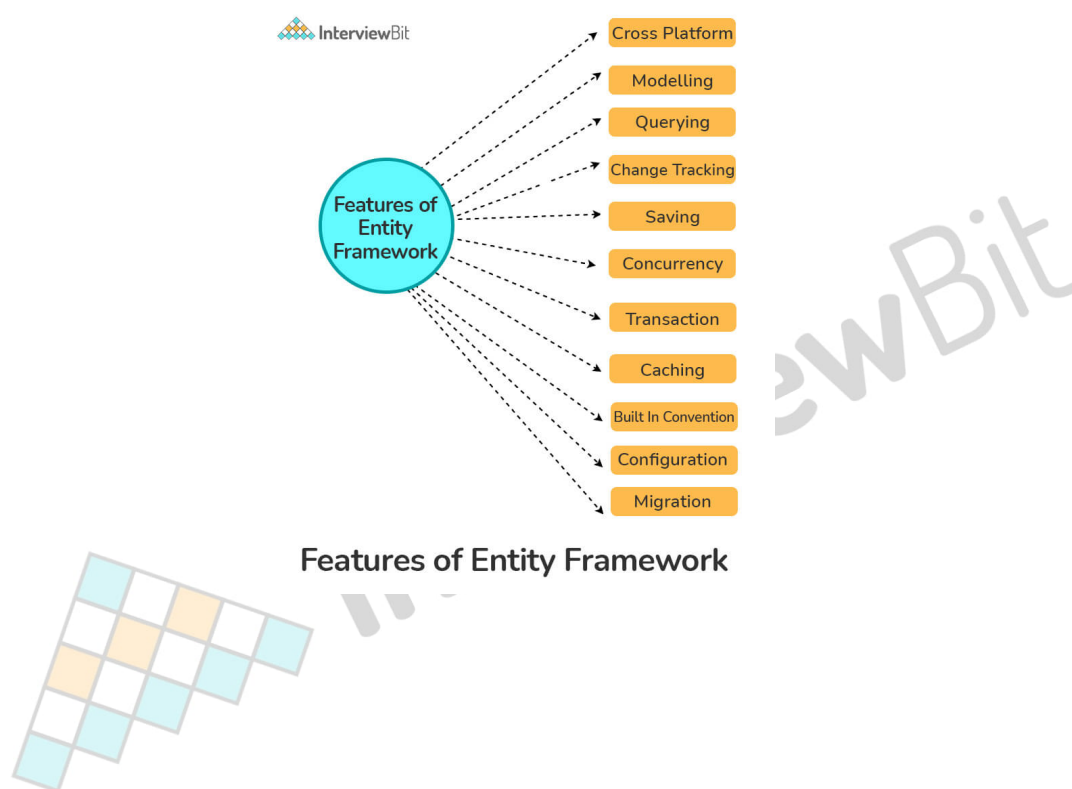
2. Describe some of the disadvantages of the Entity Framework.

Entity Framework has the following disadvantages:

- If the developer does not use raw SQL codes, things can become complicated sometimes.
- It is a slower form of the Object Relational Mapper.
- For a big domain model, it's not ideal.
- Some RDMS do not offer this feature.
- EF's main drawback is its lazy loading
- This requires a non-traditional approach to handling data that isn't available for every database.
- Since the data migration functionality is weak, it isn't fully effective in practice.

3. What are the features of the Entity Framework?

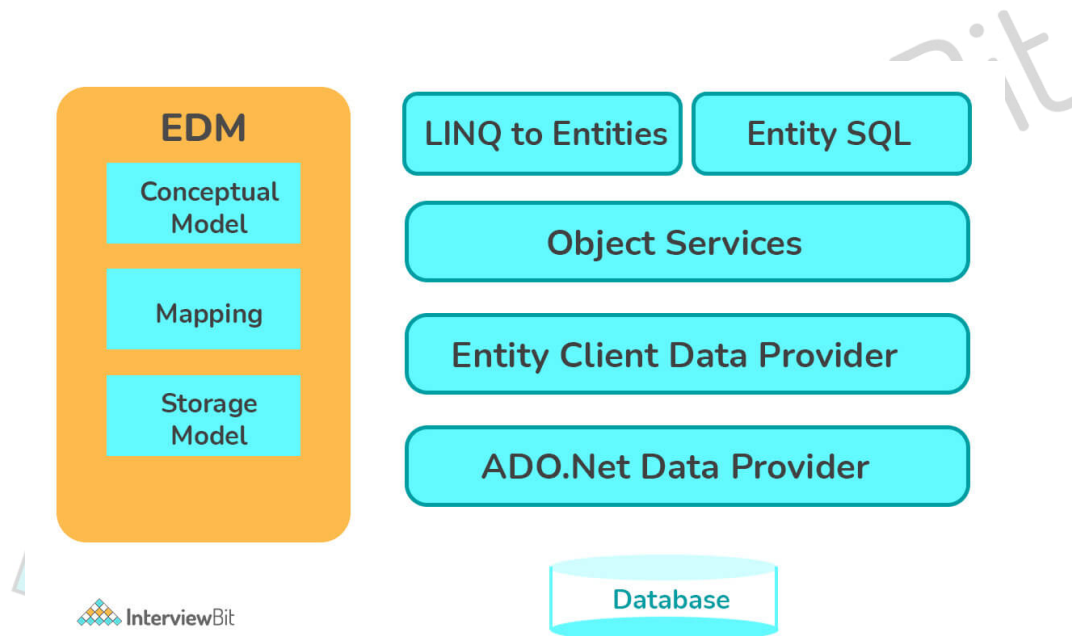
Below are some of Entity Framework's basic features:



- **Cross-Platform:** It is lightweight, extensible, open-source, and can be used on Windows, Linux, and Mac.
- **Querying:** It allows us to retrieve data from underlying databases using LINQ queries, which are then transformed into database-specific query languages.
- **Modeling:** EDMs (Entity Data Models) are typically created based on POCOs (Plain Old CLR Objects), which are entities with get/set properties of different types. This model is used when querying and saving entity data to the underlying database.
- **Change Tracking:** By using the SaveChanges method of the context, EF tracks changes to entities and their relationships and ensures the correct updates are performed on the database. Change tracking is enabled by default in EF but can be disabled by setting the AutoDetectChangesEnabled property of DbContext to false.
- **Saving:** Upon calling the "SaveChanges()" method, EF executes the INSERT, UPDATE, and DELETE commands to the database based on the changes made to entities. "SaveChangesAsync()" is another asynchronous method provided by EF.
- **Concurrency:** EF provides built-in support for Optimistic Concurrency to prevent an unknown user from overwriting data from the database.
- **Transaction:** EF's transaction management capabilities automate the querying and saving of data. Furthermore, you can customize the way that transactions are managed.
- **Caching:** First-level caching of entities is supported out of the box in the EF. Repeated queries will retrieve data from the cache rather than the database in this case.
- **Built-in Conventions:** EF conforms to the conventions of configuration programming and has a set of default settings that automatically configure the model.
- **Configuration:** By using the data annotation attribute or Fluent API, we can configure the EF model and override the default conventions.
- **Migrations:** EF provides migration commands that are executable on the command-line interface or NuGet Package Manager Console to incrementally update the database schema to keep it in sync with the application's data model.

4. What are the main components of Entity Framework Architecture?

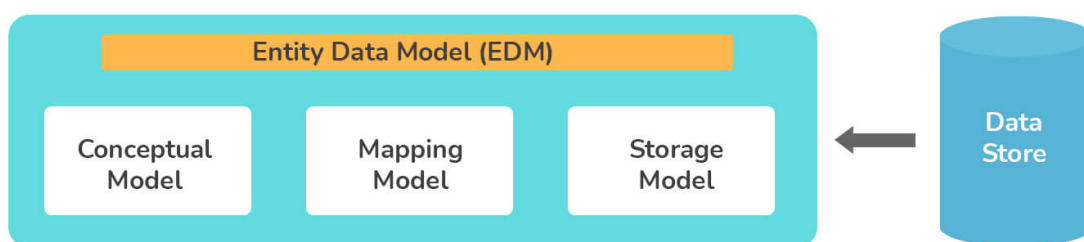
Entity Framework Architecture consists of the following components:



- **Entity Data Model (EDM):** EDMs abstract logical or relational schema and expose conceptual schema of data with a three-layered model, i.e., Conceptual (C-Space), Mapping (C-S Space), and Storage (S - Space).
- **LINQ to Entities (L2E):** L2E is basically a query language generally used to write queries against the object model. The entities defined in the conceptual model are returned by L2E.
- **Entity SQL (E-SQL):** Similar to L2E, E-SQL is another query language (for EF6 only). The developer must however learn it separately since it is more difficult than L2E. Internally, E-SQL queries are translated or converted to data store-dependent SQL queries. EF is used for converting E-SQL queries to their respective datastore queries, such as T-SQL.
- **Entity Client Data Provider:** This layer's main task is to convert E-SQL or L2E queries into SQL queries that the database understands. In turn, the ADO.Net data provider sends and retrieves data from the database.
- **Net Data Provider:** It uses standard ADO.NET to enable interaction with the database.
- **Object Service:** It is a service that facilitates access to a database, and returns data for analysis when necessary. By using it, you are able to translate data coming from entity clients into entity object structures.

5. Explain different parts of the entity data model.

The Entity Data Model consists of 3 core components that form the basis for Entity Framework. The three main components of EDM are as follows:



- **Conceptual Model:** It is also referred to as the Conceptual Data Definition Language Layer (C-Space). Typically, it consists of model classes (also known as entities) and their relationships. Your database table design will not be affected by this. It makes sure that business objects and relationships are defined in XML files.
- **Mapping Model:** It is also referred to as the Mapping Schema Definition Language layer (C-S Space). Information about how the conceptual model is mapped to the storage model is usually included in this model. In other words, this model enables the business objects and relationships defined at the conceptual layer to be mapped to tables and relationships defined at a logical layer.
- **Storage Model:** It is also referred to as the Store Space Definition Language Layer (S-Space). Schematically, it represents the storage area in the backend. Therefore, the storage model is also known as a database design model that is composed of tables, keys, stored procedures, views, and related relationships.

6. Explain what the .edmx file contains.

First of all, a database lets you reverse engineer a model from an existing database. Entity Framework Designer is used to view and edit models stored and created in EDMX files (.edmx extensions). Using the EDMX file, you automatically generate classes that you can interact with within your application.

EDMX files represent conceptual models, storage models, and their mappings. This file contains all the mapping information between SQL tables and objects. In addition, it also includes essential information required for rendering models graphically with ADO.NET Entity Data Designer. Furthermore, it is divided into three divisions, CSDL, MSL, and SSDL.

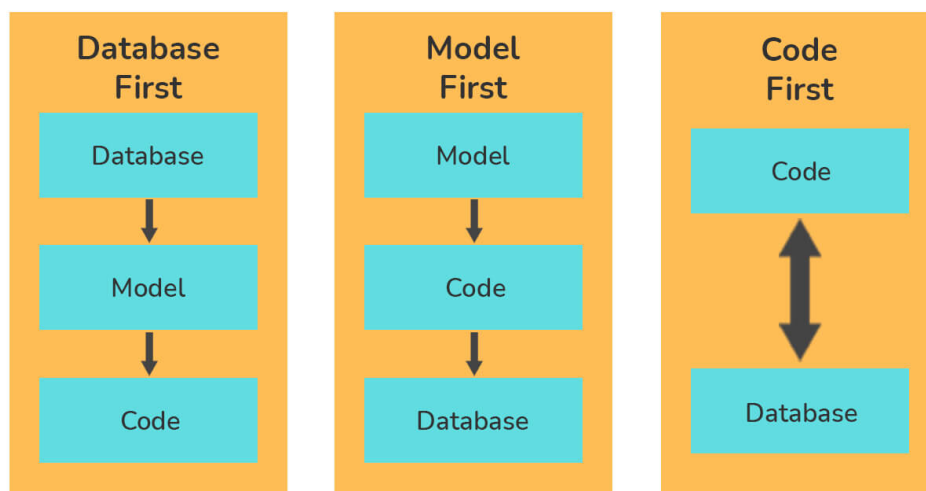
7. What do you mean by migration? Write its type.

Migration is a tool that was introduced in EF to update the database schema automatically when a model is modified without losing any data or other objects. Migrate Database To Latest Version is a new database initializer used by it. Entity Framework offers two types of migration:

- **Automated Migration:** Entity Framework 4.3 was the first to introduce automated migration so you don't have to manually migrate databases every time you alter a domain class. For example, you must also change the domain classes for each time you make a change, but with automated migration, you can simply run a command through the Package Manager Console.
- **Code-based Migration:** When you use a code-based migration, you can configure additional aspects of the migration, like setting the default value of a column, configuring a computed column, etc.

8. What are different types of Entity framework approaches?

Three different approaches to implement Entity Framework are as follows:



- **Code First Approach:** The Code First approach primarily uses classes to create the model and its relations, which are then used to create a database. This way, developers can work in an object-oriented manner without considering the database structure. By following this model, developers first write POCO classes and then use these classes to create the database. Code First is the method used by most developers using Domain-Driven Design (DDD).
- **Model First Approach:** In contrast, the Model First approach uses ORM to build model classes and their relationships. Following the successful creation of the model classes and relationships, the physical database is created using these models.
- **Database-First Approach:** In Entity Framework, Database First approach is used to build entity models based on existing databases and reduce the amount of code required. By using this approach, domain and context classes can be created based on existing classes.

9. Which according to you is considered the best approach in Entity Framework?

It is impossible to define one approach as the optimal approach when using the Entity Framework. Project requirements and the type of project determine which development approach should be used. Database First is a good approach if there is a database present. Model First is the optimal choice if no database and model classes exist. As long as the domain classes are available, the Code First method is the best choice.

10. What do you mean by the term navigation property in the entity framework?

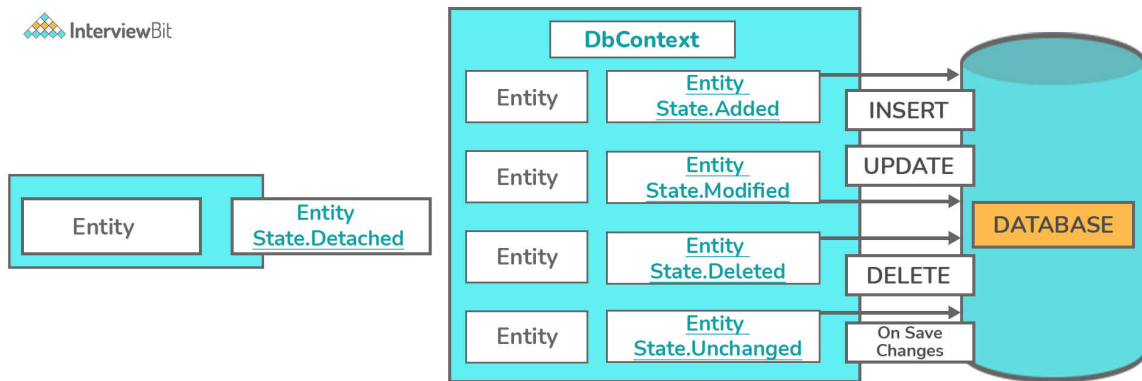
A foreign key relationship in the database is represented by the navigation property supported by the Entity Framework. It is possible to specify relationships between entities in a database using this property type. Relationships are defined in a way as to remain coherent in object-oriented code.

11. What are different entity states in EF?

There are five possible states where an entity can exist:

- **Added:** It is a state in which an entity exists within the context but does not exist within the database. When the user invokes the SaveChanges method, DbContext usually generates an INSERT SQL query to insert the data into the database. Upon successful completion of the SaveChanges method, the entity's state changes to unchanged.
- **Deleted:** This state indicates that the entity is marked for deletion has not been removed from the database. Also, it indicates the existence of the entity in the database. When the user invokes the SaveChanges method, DbContext usually generates a DELETE SQL query to delete or remove the entity from the database. Upon successful completion of the delete operation, DbContext removes the entity.
- **Modified:** When the entity is modified, its state becomes Modified. Also, it indicates the existence of the entity in the database. When the user invokes the SaveChanges method, DbContext usually generates an UPDATE SQL query to update the entity from the database. Upon successful completion of the SaveChanges method, the entity's state changes to unchanged.
- **Unchanged:** Since the context retrieved the entity's property values from the database, the values have not changed. This entity is ignored by SaveChanges.
- **Detached:** This state indicates that the entity is not tracked by the DbContext. If an entity was created or retrieved outside the domain of the current instance of DbContext, then its entity state will be Detached.

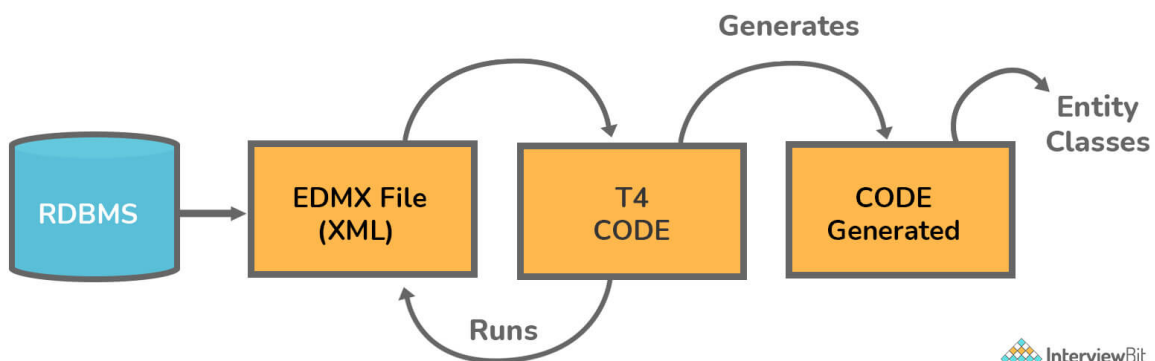
The following diagram represents the different entity states in Entity Framework:



Entity State in Entity Framework

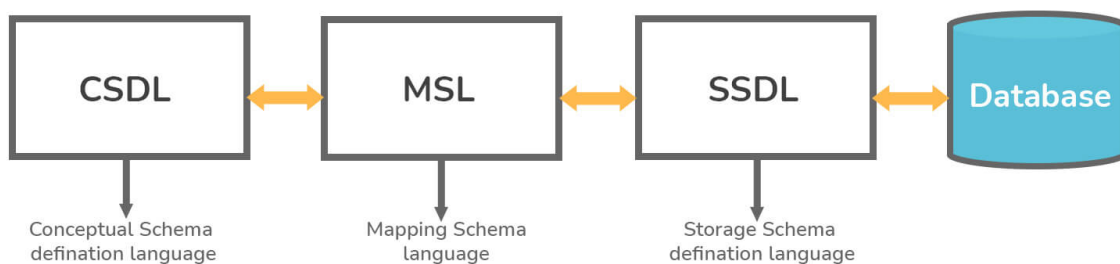
12. Write the importance of the T4 entity in Entity Framework.

In Entity Framework code generation, T4 files are crucial. EDMX XML files are read by T4 code templates, which generate C# behind code. The generated C# behind code consists only of your entity and context classes.



13. Explain CSDL, SSDL, and MSL sections in an Edmx file?

- **CSDL:** This stands for Conceptual Schema Definition Language. Basically, it's a conceptual abstraction that is exposed to the application. In this file, you will find a description of the model object.
- **SSDL:** This stands for Storage Schema Definition Language. In this section, we define the mapping to our RDBMS data structure.
- **MSL:** This stands for Mapping Schema Language. SSDL and CSDL are connected by it. It bridges the gap between the CSDL and SSDL or maps the model and the storage.



14. Explain the ways to increase the performance of EF.

Entity Framework's performance is enhanced by following these steps:

- Choose the right collection for data manipulation.
- Do not put all DB objects into one entity model.
- When the entity is no longer required, its tracking should be disabled and altered.
- Use pre-generating Views to reduce response time for the first request.
- Don't fetch all fields unless needed.
- Whenever possible, avoid using Views and Contains.
- Bind data to a grid or paging only by retrieving the number of records needed.
- Optimize and debug LINQ queries.
- Whenever possible, use compiled queries.

15. Write some XML generation methods provided by the dataset object.

DataSet objects provide the following methods for generating XML:

- **ReadXml():** This method reads an XML document into a DataSet object.
- **GetXml():** This method returns a string containing an XML document.
- **WriteXml():** This method writes XML data to disk.

16. What do you mean by the migration history table in Entity Framework?

EF6's Migration's history table (`__MigrationHistory`) is basically a database table that is used to store data about migrations applied to a database by Code First Migrations. A table like this is created when the first migration is applied to the database. Within a given database, this table contains meta-data describing the EF Code First models' schema versions. When you used the Microsoft SQL Server database, this table was considered a system table in EF5.

17. Explain how EF supports transactions.

The `SaveChanges()` method in EF always wraps any operation involving inserting, updating, or deleting data into a transaction. Hence, you do not have to explicitly open the transaction scope.

18. What do you mean by Deferred Execution in EF?

Deferred Execution refers to the process of delaying the evaluation of an expression until its realized value is actually required. As a result, performance is greatly improved since unnecessary execution is avoided. Queries are deferred until the query variable or query object is iterated over a loop.

Entity Framework Interview Questions for Experienced

19. Write difference between LINQ and Entity Framework.

LINQ	Entity Framework
In order to operate, LINQ relies only on SQL Server Databases.	In order to operate, the entity framework relies on several databases including SQL Server, Oracle, MYSQL, DB2, etc.
It generates a .dbml to maintain the relationship.	In this case, an .edmx file is generated first, then an .edmx file is maintained using three separate files- .csdl, .msl, and .ssdl.
DataContext enables you to query data.	ObjectContext, DbContext, and EntitySQL can all be used to query data.
Complex types are not supported.	Complex types are supported.
A database is not created from the model.	A database can be created from the model.
Application is developed more quickly using SQL Server.	Applications are developed more quickly using SQL Server and other databases like MYSQL, Oracle, DB2, etc.
It consists of a tightly coupled mechanism.	It consists of a loosely coupled mechanism.
Only one-to-one mappings are allowed.	One-to-one, one-to-many & many-to-many mappings are allowed.

20. Write the steps to retrieve data from database using Entity Framework in MVC.

The following steps will show you how to retrieve data from a database in MVC (Model View Controller) using Entity Framework:

- As a first step, we must create a new project.
- The next step is to add an Entity Framework reference from the NuGet package manager.
- Then, a new class has to be created within the model inside the table structure.
- After that, we are required to add a connection string in the web.config.connection. It should be matched with the context.
- The next step is to open the Global.asax.cs class and add the new namespace of EF. We must then initialize the database.
- You will now need to right-click on the Controller folder and add a new controller, followed by a model reference in the section namespace.
- Last but not least, right-click on the controller's name and add the sections you want to retrieve.

21. Explain the term dbcontext and dbset.

DbSet: An entity set is represented by a DbSet class that can be used for creating, reading, updating, and deleting operations on it. Those DbSet type properties, which map to database tables and views, must be included in the context class (derived from DbContext).

DbContext: It is considered an essential class in EF API that bridges the gap between an entity or domain class and the database. Communication with the database is its primary responsibility.

22. Difference between ADO.Net and Entity Framework.

Below are the differences between Aadonet and Entity Framework:

- A few data layer codes are created by ADO.Net that Entity Framework doesn't create.
- Entity Framework, unlike ADO.Net, generates code for intermediate layers, data access layers, and mappings automatically. This results in a reduction in development time.
- On a performance basis, ADO.Net is more efficient and faster than Entity Framework.

23. Explain the role of Pluralize and Singularize in the entity framework.

Objects in Entity Framework are primarily assigned names using Pluralize and Singularize. This feature is available when adding a .edmx file. Entity Framework automatically assigns the Singular or Plural coding conventions when using this feature. In convention names, an additional 's' is added if there is more than one record in the object.

24. What is the difference between Dapper and Entity Framework?

.NET developers are allowed to work with relational data using domain-specific objects by object-relational mappers such as Entity Framework (EF) and Dapper. Performance-wise, Dapper is the King of Micro ORMs.

- **Dapper:** A simple micro ORM, Dapper is considered a powerful system used for data access in the .NET world. As a means to address and open-source their issues, the Stack Overflow team created Dapper. Adding this NuGet library to your .NET project allows you to perform database operations. In terms of speed, it is the king of Micro ORMs and is almost as fast as using raw ADO.NET data readers.
- **Entity Framework:** It is a set of .NET APIs used in software development for performing data access. It is [Microsoft's official tool](#) for accessing data.

Comparison

- According to NuGet downloads and performance, Dapper is the world's most popular Micro ORM. In contrast, Entity Framework is significantly slower than Dapper.
- In comparison to other ORMs, such as the Entity Framework, Dapper does not generate as much SQL, but it does an excellent job mapping from database columns to CLR properties.
- Since Dapper uses RAW SQL, it can be difficult to code, especially when multiple relationships are involved, but when a lot of data is involved and performance matters, it is worth the effort.
- Since Dapper uses IDbConnection, developers can execute SQL queries to the database directly rather than put data in other objects as they do in Entity Framework.

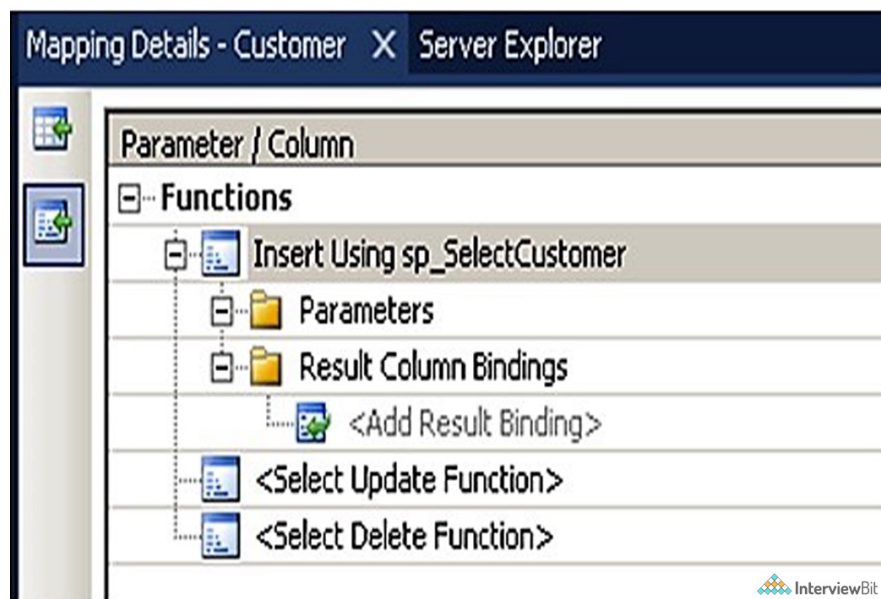
25. Explain POCO Classes in EF.

POCO stands for 'Plain Old CLR Objects'. Yet, it does not mean these classes are plain or old. A POCO class is defined as a class that contains no reference to the EF Framework or the .NET Framework at all. In EF applications, Poco entities are known as available domain objects.

POCO class is just like other normal .NET classes as these classes don't depend on any framework-specific base class, unlike the standard .NET class. Persistence-ignorant objects, or POCOs, support LINQ queries, which are supported by entities derived from the Entity Object itself. Both EF 6 and EF Core support POCO entities.

26. In Entity Framework, what are the ways to use stored procedures?

This figure shows how stored procedure mapping details can be used in EDMX:



27. Explain database concurrency and the way to handle it.

Database concurrency in EF means that multiple users can simultaneously modify the same data in one database. Concurrency controls help safeguard data consistency in situations like these.

Optimistic locking is usually used to handle database concurrency. We must first right-click on the EDMX designer and then change the concurrency mode to Fixed in order to implement locking. With this change, if there is a concurrency issue, we will receive a positive concurrency exception error.

28. What are different types of loading available to load related entities in EF?

Entity Framework offers the following types of loading:

- Eager Loading
- Lazy Loading
- Explicit Loading

29. What do you mean by lazy loading, eager loading and explicit loading?

- **Lazy Loading:** This process delays the loading of related objects until they are needed. During lazy loading, only the objects needed by the user are returned, whereas all other related objects are only returned when needed.
- **Eager Loading:** This process occurs when you query for an object and all of its related objects are returned as well. Aside from that, all related objects will load with the parent object automatically. When the Include method is used, eager loading can be achieved in EF6.
- **Explicit Loading:** Explicit loading occurs only when lazy loading is desired, even when lazy loading is disabled. We must explicitly call the relevant load method on the related entities to process explicit loading. When the Load method is used, explicit loading can be achieved in EF6.

30. What are the pros and cons of different types of loading?

1. Lazy Loading

Pros

- When the relationships are not too high, use Eager Loading. So you can reduce further queries on the server by using Eager Loading.
- If you know that related entities will be used everywhere with the main entity, use Eager Loading.

Cons

- Adding the extra lines of code to implement lazy load makes the code more complicated.
- It can affect a website's search engine ranking sometimes because the unloaded content is not properly indexed.

2. Eager Loading

Pros

- Upon executing the code, the system initializes or loads the resource.
- Additionally, related entities that are referenced by a resource must be pre-loaded.
- It is advantageous when resources need to be loaded in the background.
- It saves you time by avoiding the need to execute extra SQL queries.

Cons

- Since everything must be loaded to begin running, starting the application takes a longer time.

Choosing the right tool

- When you know you will use related entities with your main entity everywhere, use Eager Loading.
- You should use Lazy Loading whenever you have one-to-many collections.
- Use lazy loading only if you are sure you won't need related entities right away.
- When you are unsure about whether or not an entity will be used, use explicit loading after you have turned off Lazy Loading.

31. Write different types of inheritance supported by Entity Framework.

In Entity Framework, inheritance is primarily divided into three types:

- **Table per Hierarchy (TPH):** The TPH inheritance representation shows one table per inheritance hierarchy class. A discriminator column also aids in distinguishing between inheritance classes. This is Entity Framework's default inheritance mapping technique.
- **Table per Type (TPT):** In this inheritance method, each domain class has its own table.
- **Table per Concrete Class (TPC):** TPC demonstrates a single table per concrete class, but does not include the abstract class. Because of this, if an abstract class is inherited by many concrete classes, then the tables in all those concrete classes will have the same properties as that of an abstract class.

32. Explain Complex Type in Entity Framework.

Complex types are defined as the non-scalar properties of entity types that assist in organizing scalar properties within entities. In addition to scalar properties, complex types may also have other complex type properties. Instances of complex types are complex objects.

33. What do you mean by Micro ORM?

Rather than creating database schemas, modifying database schemas, tracking changes, etc., Micro ORMs focus on working with database tables. EF 6.x and EF Core provide a full set of capabilities and features, making them ORMs.

34. Explain EF Data access Architecture.

There are two types of Data Access Architecture supported by the ADO.NET Framework:

- **Disconnected data access:** Disconnected data access is possible with the Data Adapter object. Datasets work independently of databases, and the data can be edited.
- **Connected data access:** A Data Reader object of a Data Provider allows you to access linked data. Data can be accessed quickly, but editing is not permitted.

35. What do you mean by SQL injection attack?

SQL injection is a method that hackers use to access sensitive information from an organization's database. This application-layer attack is the result of inappropriate coding in our applications, allowing hackers to inject SQL statements into your SQL code.

The most common cause of SQL Injection is that user input fields allow SQL statements to pass through and directly query the database. ADO.NET Data Services queries are commonly affected by SQL Injection issues.

36. What is the best way to handle SQL injection attacks in Entity Framework?

The injection-proof nature of Entity Framework lies in the fact that it generates parameterized SQL commands that help prevent our database from SQL injections.

By inserting some malicious inputs into queries and parameter names, one can generate a SQL injection attack in Entity SQL syntax. It is best to never combine user inputs with Entity SQL commands text to prevent or avoid this problem.

37. Explain the ObjectSet in EF.

ObjectSet is generally considered as a specific type of data set that is commonly used to read, update, create, and remove operations from existing entities. Only theObjectContext instance can be used to create it. No Entity SQL method is supported by it.

38. Write the namespace that is used to include .NET Data provider for SQL server in .NET code.

NET Data Provider for SQL Server is included in .NET code by using the namespace System.Data.SqlClient.

39. Explain EDM and write the process to create it.

In Entity Framework, EDM refers to the '**Entity Data Model**'. It is considered as an entity-relationship prototype that assigns some basic prototypes for the data using various modeling procedures. Moreover, it is defined as a set of principles pertaining to the formation of data, regardless of how it is collected. Shortly, it's just a simple link or connection created between the database and the prototype. The steps for creating an Entity Data Model are as follows:

- Right-click on a project in the Solution Explorer.
- Select the Add>New Item option from the menu.
- Select the ADO.Net Entity Data Model arrangement or template.
- Please enter a name and click the 'Add' button.

40. What do you mean by DbEntityEntry Class in EF?

An important class, DbEntityEntry helps you retrieve a variety of information about an entity. DbContext offers the Entry method for retrieving an instance of DbEntityEntry of a specific entity.

Example:

```
DbEntityEntry studentEntry = dbcontext.Entry(entity);
```

You can access the entity state, as well as the current and original values of all properties of an entity using the DbEntityEntry. EntityState can be set using the DbEntityEntry, as shown below.

```
context.Entry(student).State = System.Data.Entity.EntityState.Modified;
```

Conclusion:

With .NET Framework 3.5, Microsoft introduced Entity Framework in 2008. Since then, the company has released a number of Entity Framework versions. EF 6 and EF Core are the two latest versions of Entity Framework. With Entity Framework, [.NET developers](#) can work with a database using .NET objects. By doing this, the traditional method of accessing data through code is eliminated. Basically, it allows you to read and write data from a database. We have compiled a list of the top Entity Framework interview questions to assist freshers and experienced developers who are seeking new career opportunities and skills that require Entity Framework knowledge and skills.

Links to More Interview Questions

[C Interview Questions](#)

[Php Interview Questions](#)

[C Sharp Interview Questions](#)

[Web Api Interview Questions](#)

[Hibernate Interview Questions](#)

[Node Js Interview Questions](#)

[Cpp Interview Questions](#)

[Oops Interview Questions](#)

[Devops Interview Questions](#)

[Machine Learning Interview Questions](#)

[Docker Interview Questions](#)

[Mysql Interview Questions](#)

[Css Interview Questions](#)

[Laravel Interview Questions](#)

[Asp Net Interview Questions](#)

[Django Interview Questions](#)

[Dot Net Interview Questions](#)

[Kubernetes Interview Questions](#)

[Operating System Interview Questions](#)

[React Native Interview Questions](#)

[Aws Interview Questions](#)

[Git Interview Questions](#)

[Java 8 Interview Questions](#)

[Mongodb Interview Questions](#)

[Dbms Interview Questions](#)

[Spring Boot Interview Questions](#)

[Power Bi Interview Questions](#)

[Pl Sql Interview Questions](#)

[Tableau Interview Questions](#)

[Linux Interview Questions](#)

[Ansible Interview Questions](#)

[Java Interview Questions](#)

[Jenkins Interview Questions](#)