**1. Что такое ORM?**

ORM (Object-Relational Mapping) - это технология, которая позволяет работать с базой данных в терминах объектов и классов.

**2. Зачем используются ORM-решения при работе с базами данных?**

**Опишите их достоинства и недостатки.**

ORM-решения используются для упрощения процесса взаимодействия с базой данных и сокращения количества кода, необходимого для доступа к данным. Они позволяют уменьшить время на разработку приложения и повысить его переносимость. Однако, некоторые ORM-решения могут снижать производительность и сложность настройки.

**3. Что такое Entity Framework?**

Entity Framework - это ORM-решение для работы с базами данных в .NET.

**4. Что такое EDM в Entity Framework и из каких уровней она состоит?**

EDM (Entity Data Model) в Entity Framework - это модель данных, которая описывает сущности и их взаимосвязи в базе данных. Она состоит из трех уровней: концептуальный уровень, логический уровень и физический уровень.

**5. Перечислите способы создания моделей. Опишите их действия.**

Способы создания моделей в Entity Framework: Database First, Model First, Code First. Database First используется для создания модели на основе существующей базы данных, Model First - для создания модели через графический интерфейс, а Code First - для создания модели с помощью кода.

**6. Назовите основные отличия EF от других технологий работы с базами**

**данных.**

Основные отличия EF от других технологий работы с базами данных - это возможность работы с базой данных в терминах объектов и классов, автоматическое создание SQL-запросов, поддержка ленивой загрузки данных и миграций.

**7. Для чего и как используется DbContext?**

DbContext - это класс в Entity Framework, который предоставляет доступ к базе данных и управляет жизненным циклом объектов.

**8. Как и для чего используются аннотации при настройке конфигураций**

**Code First?**

Аннотации используются для настройки конфигураций Code First и указания атрибутов, связанных с таблицами в базе данных, таких как имя таблицы и ее столбцов.

**9. Как и для чего используются Fluent API при настройке конфигураций**

**Code First?**

Fluent API используется для настройки конфигураций Code First и представляет собой набор методов для конфигурации сущностей и их свойств.

**10.Какие существуют способы загрузки связанных данных?**

Существуют два способа загрузки связанных данных: явная загрузка (Eager Loading) и ленивая загрузка (Lazy Loading). В явной загрузке связанные данные извлекаются из базы данных вместе с главными данными с помощью метода Include. В ленивой загрузке связанные данные извлекаются только при обращении к связанному свойству.

**11.В чем разница между концепциями Code First и Database First?**

Code First и Database First являются разными подходами к разработке баз данных в Entity Framework. При использовании Code First модель базы данных создается на основе классов сущностей, а при использовании Database First модель базы данных создается на основе существующей базы данных.

**12.Зачем EF использует миграции? Как создать новую миграцию?**

Миграции в Entity Framework используются для обновления базы данных с помощью изменений в классах сущностей. Чтобы создать новую миграцию, нужно выполнить команду Add-Migration в консоли диспетчера пакетов и указать имя миграции.

**13.Расскажите алгоритм выполнения CRUD-запросов при работе с БД**

**через библиотеку Entity Framework.**

Алгоритм выполнения CRUD-запросов при работе с БД через библиотеку Entity Framework состоит из следующих шагов: создание объекта контекста данных (DbContext), создание объекта сущности, добавление объекта в контекст данных (Add), выполнение запроса на сохранение изменений (SaveChanges), выполнение запроса на получение данных (ToList, FirstOrDefault и т. д.), выполнение запроса на изменение данных (Update), выполнение запроса на удаление данных (Remove).

**Lab 10**

**1. Для чего используют многоуровневые архитектуры?**

Многоуровневые архитектуры используются для создания приложений, которые являются модульными, легко расширяемыми, поддерживают принцип единственности.

**2. Опишите назначение слоев многоуровневой архитектуре. Business layer**

**(уровень бизнес-логики), Data Access layer (уровень доступа к данным).**

Уровень бизнес-логики обычно содержит код, который управляет бизнес-логикой приложения. Он может содержать правила бизнес-логики, сервисы и другие классы, которые управляют поведением приложения. Уровень доступа к данным (Data Access layer) содержит код, который обеспечивает доступ к данным, таким как запросы к базе данных, чтение и запись данных, и т. д.

**3. Поясните назначение и принцип использования паттерна Repository?**

Шаблон проектирования Repository позволяет скрыть детали работы с базой данных, предоставляя удобный интерфейс для взаимодействия с данными. В основе шаблона лежит идея разделения ответственности: объекты бизнес-логики не должны иметь прямого доступа к базе данных, а должны взаимодействовать только с объектами репозитория. Репозиторий, в свою очередь, должен реализовывать все необходимые операции для работы с данными: чтение, запись, удаление и т.д.

**4. Поясните назначение и принцип использования паттерна Unit of Work?**

Шаблон Unit of Work (единица работы) предназначен для управления транзакциями и сохранения изменений в базе данных. Он позволяет группировать несколько операций в одну транзакцию и сохранять все изменения в базе данных только после успешного завершения всех операций. Также шаблон Unit of Work позволяет отслеживать изменения объектов и сохранять их в базу данных при необходимости.

Обычно эти два шаблона используются вместе: репозитории предоставляют удобный интерфейс для взаимодействия с данными, а Unit of Work управляет транзакциями и сохранением изменений в базе данных. Это позволяет сделать код более гибким, удобным и легко поддерживаемым.

**5. Опишите основные особенности библиотеки Entity Framework?**

Библиотека Entity Framework - это ORM-фреймворк для работы с базами данных. Она предоставляет удобный способ работы с базами данных, позволяет разработчикам работать с данными как с объектами в памяти, а не с SQL-запросами, и предоставляет удобный API для работы с различными типами баз данных.

**6. Какие преимущества обеспечивает использование слабосвязанного**

**кода?**

Использование слабосвязанного кода может обеспечить лучшую гибкость и возможность повторного использования кода. Когда код слабосвязан, его можно легко заменить или модифицировать без влияния на другие части системы. Это может упростить тестирование и развертывание, а также сделать код более модульным и легко читаемым.