1. Что такое ORM?

ORM (Object-Relational Mapping) - это технология, которая позволяет разработчикам работать с базами данных, используя объектно-ориентированный подход. ORM связывает объекты в коде с таблицами в базе данных, обеспечивая прозрачный доступ к данным.

1. Зачем используются ORM-решения при работе с базами данных? Опишите их достоинства и недостатки.

ORM-решения используются для облегчения и ускорения работы с базами данных. Они позволяют разработчикам работать с базами данных, используя объектно-ориентированный подход, без необходимости писать SQL-запросы вручную. Основными преимуществами ORM являются повышение производительности, упрощение кода, повышение безопасности и уменьшение вероятности ошибок. Недостатком ORM является увеличение сложности и размера кода.

1. Что такое Entity Framework?

Entity Framework (EF) - это ORM-решение для работы с базами данных в Microsoft .NET Framework. EF позволяет разработчикам работать с данными, используя объектно-ориентированный подход, и скрывает детали работы с базами данных.

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

EDM (Entity Data Model) - это модель данных, которая описывает структуру базы данных в Entity Framework. EDM состоит из трех уровней: концептуальный уровень, логический уровень и физический уровень. Концептуальный уровень описывает данные в терминах объектов, логический уровень описывает связь между объектами и таблицами в базе данных, а физический уровень описывает структуру таблиц в базе данных.

1. Перечислите способы создания моделей. Опишите их действия.

Существует несколько способов создания моделей в Entity Framework:

* Database First - модель создается на основе существующей базы данных.
* Model First - модель создается с помощью дизайнера моделей Entity Framework.
* Code First - модель создается с помощью кода.

1. Назовите основные отличия EF от других технологий работы с базами данных.

Основные отличия EF от других технологий работы с базами данных заключаются в том, что EF позволяет работать с данными, используя объектно-ориентированный подход, и скрывает детали работы с базами данных. Это упрощает код и повышает безопасность

1. Для чего и как используется DbContext?

DbContext - это класс, который представляет контекст базы данных в Entity Framework. DbContext используется для управления сессиями работы с базой данных, управления состоянием объектов и выполнения запросов.

1. Как и для чего используются аннотации при настройке конфигураций Code First?

Аннотации в Entity Framework используются для управления маппингом классов на таблицы базы данных. Они позволяют задавать свойства и ограничения для полей класса, указывать названия таблиц и столбцов в базе данных, а также настраивать связи между таблицами. Аннотации удобны в использовании, когда нужно быстро настроить модель без создания отдельных классов конфигураций.

1. Как и для чего используются Fluent API при настройке конфигураций Code First?

Fluent API в Entity Framework также используется для настройки маппинга классов на таблицы базы данных, но позволяет более гибко и точно задавать конфигурации, чем аннотации. Fluent API позволяет настраивать связи между таблицами, индексы, ограничения и другие параметры. Использование Fluent API позволяет более гибко управлять маппингом, но требует написания дополнительного кода.

1. Какие существуют способы загрузки связанных данных?

Существует несколько способов загрузки связанных данных в Entity Framework: жадная загрузка (eager loading), ленивая загрузка (lazy loading) и явная загрузка (explicit loading).

* Жадная загрузка подразумевает загрузку связанных данных сразу вместе с основными данными при запросе.
* Ленивая загрузка позволяет загружать связанные данные по мере их запроса в коде приложения.
* Явная загрузка позволяет явно загрузить связанные данные, когда это необходимо.

1. В чем разница между концепциями Code First и Database First?

Концепция Code First в Entity Framework подразумевает создание модели объектов в коде и генерацию базы данных на основе этой модели. В то время как концепция Database First предполагает создание базы данных вручную и генерацию модели объектов на основе этой базы данных.

1. Зачем EF использует миграции? Как создать новую миграцию?

Миграции в Entity Framework используются для автоматического обновления структуры базы данных при изменении модели объектов. Миграции позволяют создавать и применять изменения в базе данных, сохраняя данные в уже существующих таблицах. Для создания новой миграции необходимо выполнить команду Add-Migration в консоли диспетчера пакетов (Package Manager Console) и задать имя миграции.

13. Расскажите алгоритм выполнения CRUD-запросов при работе с БД через библиотеку Entity Framework.

1. Алгоритм выполнения CRUD-запросов (Create, Read, Update, Delete) в Entity Framework следующий:

* Создание контекста данных DbContext и определение объекта DbSet, представляющего таблицу базы данных.
* Создание нового объекта класса, соответствующего записи в таблице базы данных.
* Добавление объекта в контекст данных с помощью метод
* Add() объекта DbSet. Вызов метода SaveChanges() контекста данных для сохранения изменений в базе данных. При выполнении операции чтения (Read) в Entity Framework, необходимо выполнить следующие шаги:
* Создание контекста данных DbContext и определение объекта DbSet, представляющего таблицу базы данных. Вызов метода DbSet, который возвращает коллекцию объектов класса, соответствующих записям в таблице базы данных. Выполнение необходимых операций с данными, таких как фильтрация, сортировка, выборка определенных полей и т.д. При выполнении операции обновления (Update) в Entity Framework, необходимо выполнить следующие шаги:
* Создание контекста данных DbContext и определение объекта DbSet, представляющего таблицу базы данных. Изменение нужных свойств объекта класса, соответствующего записи в таблице базы данных. Вызов метода SaveChanges() контекста данных для сохранения изменений в базе данных. При выполнении операции удаления (Delete) в Entity Framework, необходимо выполнить следующие шаги:
* Создание контекста данных DbContext и определение объекта DbSet, представляющего таблицу базы данных. Выбор объекта класса, соответствующего удаляемой записи в таблице базы данных. Удаление объекта из контекста данных с помощью метода Remove() объекта DbSet. Вызов метода SaveChanges() контекста данных для сохранения изменений в базе данных.