**Задание 07.** Ответьте на следующие вопросы

1. Поясните понятие «ORM»

ORM (Object-Relational Mapping) - это технология, которая позволяет связать объектно-ориентированный подход программирования с реляционной базой данных. ORM является слоем абстракции между приложением и базой данных, предоставляя удобный интерфейс для работы с данными в базе через объекты.

ORM-фреймворки позволяют программистам работать с базой данных, используя объекты и методы, а не SQL-запросы. ORM-фреймворк автоматически генерирует SQL-запросы, выполняет их и возвращает результаты в виде объектов, скрывая от программиста детали реализации запросов и работы с базой данных.

ORM упрощает и ускоряет разработку приложений, позволяя сосредоточиться на бизнес-логике и архитектуре приложения, а не на деталях работы с базой данных. ORM также облегчает поддержку приложения и сокращает время разработки, так как позволяет избежать ошибок, связанных с неправильным использованием SQL-запросов.

ORM-фреймворки предоставляют различные возможности, такие как:

* Отображение таблиц базы данных на классы и объекты в языке программирования
* Создание, изменение и удаление записей в базе данных, используя методы объектов
* Отображение связей между таблицами базы данных на связи между классами и объектами в языке программирования
* Поддержка миграций базы данных для изменения структуры базы данных без необходимости написания SQL-запросов
* Поддержка кэширования данных, чтобы уменьшить количество запросов к базе данных
* Поддержка транзакций для обеспечения целостности данных.

Некоторые из популярных ORM-фреймворков включают в себя: Hibernate для Java, Entity Framework для .NET, SQLAlchemy для Python и Sequelize для Node.js.

1. Поясните понятие «модель».

Модель – это абстракция данных, представляющая сущности или объекты из предметной области, которые необходимы приложению для выполнения своих функций. Модель описывает структуру и свойства объектов, а также правила взаимодействия между ними.

Модель может быть представлена в виде класса или структуры данных, которые содержат поля, методы и связи с другими объектами. Модель может также содержать правила валидации, которые определяют правильность данных, вводимых пользователем.

1. Поясните понятие «пул соединений» и его назначение.

Пул соединений (Connection Pool) - это механизм управления соединениями с базой данных, который позволяет приложению повторно использовать уже установленные соединения вместо создания новых каждый раз, когда требуется выполнить запрос к базе данных.

Пул соединений представляет собой пул (набор) соединений, которые создаются заранее и хранятся в памяти приложения. Когда приложение нуждается в выполнении запроса к базе данных, оно запрашивает свободное соединение из пула соединений, выполняет запрос и возвращает соединение в пул.

Назначение пула соединений заключается в уменьшении времени, затрачиваемого на установление соединения с базой данных. Установление соединения с базой данных - это дорогостоящая операция, так как требуется установить соединение через сеть, аутентифицироваться и т.д.

1. Что такое Sequelize и Prisma?

Sequelize и Prisma - это два популярных ORM-фреймворка для работы с базами данных в Node.js.

1. Поясните, что такое «scope» в Sequelize?

"Scope" - это один из механизмов Sequelize, который позволяет определять набор условий для выборки данных из базы данных и повторно использовать их в различных запросах.

Scope позволяет определить набор условий выборки в модели Sequelize и использовать их внутри других методов модели, например, findAll, findOne, count и других. Это позволяет создавать более гибкие и модульные запросы к базе данных.

Scope также может быть объединен с другими scope и условиями выборки, чтобы создавать более сложные запросы к базе данных.

Scope позволяет улучшить читаемость и поддерживаемость кода, а также уменьшить дублирование кода при работе с базами данных в Sequelize.

1. Поясните, что такое «хуки» в Sequelize? Какие они бывают?

Хуки (hooks) - это функции, которые позволяют выполнять пользовательский код в определенных точках жизненного цикла модели Sequelize.

Хуки позволяют расширять функциональность модели Sequelize и выполнять дополнительные действия при сохранении, обновлении или удалении записей в базе данных. Хуки могут быть определены для каждой операции базы данных, такой как beforeCreate, afterCreate, beforeUpdate, afterUpdate, beforeDestroy и afterDestroy.

Хуки могут быть определены внутри модели, как методы экземпляра или класса, или в отдельном файле, который затем импортируется в модель.

1. Поясните назначение функции Sequelize *Модель*.**hasMany**.

Функция Model.hasMany() в Sequelize - это метод, который позволяет установить связь «один-ко-многим» между двумя моделями данных.

Связь «один-ко-многим» означает, что у одного объекта в первой модели может быть много связанных объектов во второй модели, но у каждого объекта во второй модели может быть только один связанный объект в первой модели.

1. Поясните понятие «raw query» в Sequelize.

"Raw query" (сырой запрос) в Sequelize - это SQL-запрос, который напрямую передается в базу данных без использования ORM-методов Sequelize. Это означает, что такой запрос не будет обработан или преобразован Sequelize, и результаты будут возвращены в их исходном виде.

Для выполнения "raw query" в Sequelize можно использовать метод sequelize.query(), который принимает SQL-запрос в виде строки и возвращает Promise с результатами запроса. Например, следующий код выполнит "raw query" для выборки всех пользователей из таблицы users:

1. Назовите отличия Prisma от других фреймворков ORM?

Prisma - это современный ORM-фреймворк для работы с базами данных, который отличается от других фреймворков ORM следующими особенностями:

* Отсутствие генерации кода: Prisma не генерирует код на основе схемы базы данных, как это делают другие ORM-фреймворки. Вместо этого, Prisma использует декларативный язык моделирования данных для описания схемы базы данных и автоматически генерирует типобезопасные API для доступа к данным в базе данных.
* Типобезопасность: Prisma обеспечивает типобезопасность для доступа к данным в базе данных, что позволяет выявлять ошибки на этапе компиляции, а не на этапе выполнения. Это достигается за счет генерации типобезопасных API на основе схемы данных, которые автоматически проверяют типы данных при компиляции.
* Поддержка нескольких базданных: Prisma поддерживает несколько баз данных, включая MySQL, PostgreSQL и SQLite. Это позволяет разработчикам использовать один и тот же ORM-фреймворк для работы с различными базами данных.
* Масштабируемость: Prisma обеспечивает высокую масштабируемость за счет оптимизации запросов к базе данных и возможности распределения нагрузки между несколькими серверами баз данных.
* Асинхронность: Prisma предоставляет асинхронные API для доступа к данным в базе данных, что позволяет выполнять запросы к базе данных без блокировки основного потока выполнения приложения.
* Интуитивный API: Prisma предоставляет интуитивный API для доступа к данным в базе данных, который позволяет выполнять запросы к базе данных с помощью простых и понятных методов, таких как findMany, create, update и delete.

1. Можно ли использовать Prisma с базами данных SQL и NoSQL?

Prisma является ORM-фреймворком для работы с базами данных, который поддерживает работу только с реляционными базами данных. На сегодняшний день Prisma поддерживает работу с следующими реляционными базами данных:

* PostgreSQL
* MySQL
* MariaDB
* SQLite

Таким образом, Prisma не может быть использован с NoSQL базами данных, такими как MongoDB или Cassandra, поскольку они используют не реляционную модель данных.

1. Какие языки программирования совместимы с Prisma?

Несмотря на то, что Prisma создавался в первую очередь для работы с Node.js, он может использоваться с любым языком программирования, который может взаимодействовать с базой данных через протоколы, такие как TCP/IP, HTTP или WebSockets.

1. Расскажите, где и как можно использовать ***every***, ***none*** и ***some*** вPrisma?
2. Для чего нужен метод ***upsert*** вPrisma?

Метод upsert() в Prisma предназначен для выполнения операции вставки или обновления данных в базе данных. Название метода upsert происходит от сочетания английских слов "update" (обновление) и "insert" (вставка), что отражает его функциональность.

1. Как включить регистронезависимый поиск вPrisma?

В Prisma можно включить регистронезависимый (case-insensitive) поиск, используя функцию lower() в комбинации с методами findMany() или findFirst(). Функция lower() преобразует все символы в строке к нижнему регистру, что позволяет выполнить поиск без учета регистра символов.

1. Объясните назначение методов ***$queryRaw***, ***$executeRaw*** вPrisma.

Методы $queryRaw и $executeRaw в Prisma предназначены для выполнения сырых SQL-запросов в базу данных, к которой подключен Prisma.

Метод $queryRaw используется для выполнения запросов, которые возвращают данные из базы данных. Он принимает SQL-запрос в виде строки и возвращает результат запроса в виде объекта, который содержит массив строк, представляющих каждую строку результата запроса, и метаданные, такие как количество строк, затронутых запросом.