1. Поясните понятие «ORM»

**Object-Relational Mapping** - технология программирования, которая позволяет работать с SQL-базой данных, как с набором программных объектов. **Mapping**: база данных – объект contextDB, таблица – коллекция объектов, строка в таблице – объект, структура таблицы – класс.

1. Какой дополнительный пакет необходимо установить для работы sequelize c Microsoft SQL Server? tedious
2. Поясните понятие «sequelize-схема».

это описание структуры модели (таблицы) базы данных, включая поля, их типы, ограничения и связи с другими моделями. Она задаётся с помощью методов sequelize.define или классов Model.

1. Какая функция sequelize позволяет выполнить SELECT-запрос?

Для SELECT-запросов используется функция **findAll** (получить все записи) или **findOne** (получить одну запись), а также findByPk для поиска по первичному ключу.

1. Какая функция sequelize позволяет выполнить INSERT-запрос? create
2. Какая функция sequelize позволяет выполнить UPDATE-запрос? update
3. Какая функция sequelize позволяет выполнить DELETE-запрос? destroy
4. Поясните назначение функции sequelize *Схема*.**hasMany**.

Метод hasMany задаёт связь «один-ко-многим» между моделями. Это означает, что один объект модели А может иметь множество связанных объектов модели B. Например, один пользователь может иметь много заказов.

1. Поясните понятие «raw query».

Raw query — это выполнение SQL-запроса напрямую, минуя ORM-интерфейс. В Sequelize используется метод sequelize.query('RAW SQL')

1. Поясните понятие «code first».

Code First — это подход, при котором сначала создаются модели в коде, а затем на их основе генерируются таблицы в базе данных. Это противоположно Database First, где сначала создаётся БД.

В Model First синхронизируется и код, и база данных.

1. Поясните понятие «транзакция».

Транзакция — это последовательность операций с базой данных, которые выполняются как единое целое. В случае ошибки все операции откатываются. Sequelize поддерживает как управляемые, так и неуправляемые транзакции

Опции:

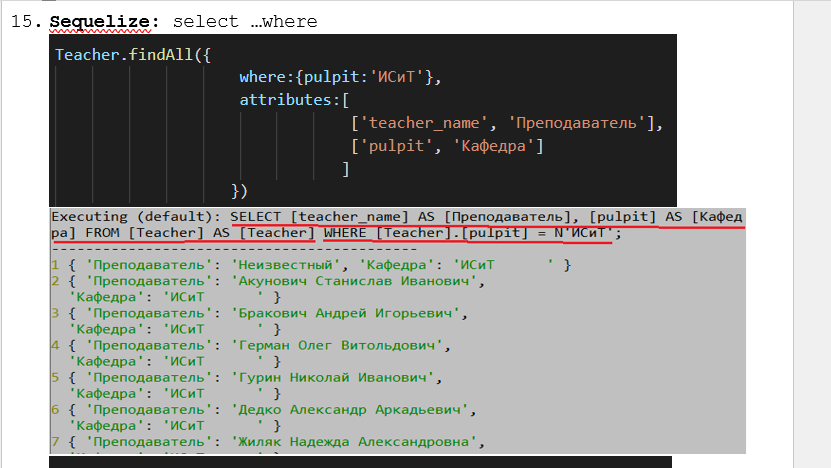
sequelize.define('ModelName', {...}, {...}) — функция определения модели. Первый аргумент — имя модели, второй — структура (поля), третий — опции.

Свойства полей (внутри модели):  
type: DataTypes.STRING(n) / DataTypes.INTEGER — указывает тип данных в БД (строка с ограничением длины или целое число).  
allowNull: false — поле не может содержать NULL, обязательно для заполнения.  
primaryKey: true — обозначает первичный ключ таблицы.  
references: { model, key } — создаёт внешний ключ, указывающий на другую таблицу (для связей между моделями).

Свойства модели (в опциях):  
tableName — явно указывает имя таблицы в БД, если оно отличается от имени модели.  
timestamps: false — отключает автоматическое создание полей createdAt и updatedAt, которые Sequelize добавляет по умолчанию.

Связи:

hasMany — устанавливает сторону "один".  
belongsTo — устанавливает сторону "много", ссылающуюся на "один".  
foreignKey — поле, по которому связываются модели (оно уже указано в references, но здесь явно задаётся связь на уровне Sequelize API).



Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Содержимое, созданное искусственным интеллектом, может быть неверным.