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.

**ORM** – это методика или техника программирования, предназначенная для преобразования между **несовместимыми типами данных в объектно-ориентированных языках программирования.** Суть заключается в том, что создается некая абстракция - “**виртуальная объектная база**”, запросы к которой, преобразуются в **SQL команды,** т.е. больше не нужно писать **SQL-запросы к базе данных вручную**.

**ORM фреймворк** может быть написан на каком-либо объектно-ориентированном языке и представлять обертку над некой реляционной базой данных. **Классы будут соответствовать таблицам в базе, а экземпляры этих классов – конкретным строкам таблицы**.

1. **Какой дополнительный пакет необходимо установить для работы sequelize c Microsoft SQL Server?**

Npm i tedious.Модуль **tedious** — **это** реализация протокола TDS на языке JavaScript с открытым кодом. -Табличный поток данных - это протокол прикладного уровня, используемый для передачи данных между сервером базы данных и клиентом.

1. Поясните понятие «sequelize-схема».

Множество классов и их объектов, описывающих контекст и таблицы базы данных. Описание схемы позволяет проецировать структуру БД на программный код в виде классов-моделей.

1. Какая функция sequelize позволяет выполнить SELECT-запрос?

**findAll** генерирует стандартный SELECT запрос, который извлекает все записи из таблицы (если только они не ограничены чем-то вроде whereпредложения, например).

**findByPk** получает только одну запись из таблицы, используя предоставленный первичный ключ.

**findOne** получает первую найденную запись (которая удовлетворяет необязательным параметрам запроса, если они предусмотрены).

**findOrCreate** создаст запись в таблице, если он не может найти ее, выполняющую параметры запроса. В обоих случаях он возвращает экземпляр (найденный экземпляр или созданный экземпляр) и логическое значение, указывающее, был ли этот экземпляр создан или уже существовал.

1. Какая функция sequelize позволяет выполнить INSERT-запрос?

create(), findOrCreate()

1. Какая функция sequelize позволяет выполнить UPDATE-запрос?

Update()

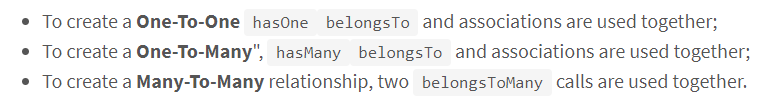
1. Какая функция sequelize позволяет выполнить DELETE-запрос?

Destroy()

1. Поясните назначение функции sequelize *Схема*.**hasMany**.

Создание связей между таблицами осуществляется с использованием моделей. Рассмотрим установление следующих типов связей:

* один к одному;
* один ко многим;
* многие ко многим.



Эта A.hasOne(B) связь означает, что существует взаимно однозначное отношение между A и B, причем внешний ключ определяется в целевой модели (B).

Эта A.belongsTo(B) связь означает, что существует взаимно однозначное отношение между A и B, причем внешний ключ определяется в исходной модели (A).

A.hasMany(B) Ассоциация означает, что существует отношение "один ко многим" Между A и B, причем внешний ключ определяется в целевой модели (B).

1. Поясните понятие «raw query».

Поскольку часто встречаются случаи использования, в которых просто проще выполнять необработанные / уже подготовленные SQL-запросы, вы можете использовать этот sequelize.query метод.

По умолчанию функция возвращает два аргумента - массив результатов и объект, содержащий метаданные (например, количество затронутых строк и т. д.).

Задаем запрос вручную, самописный запрос, когда очень сложно реализовать через методы.

1. Поясните понятие «code first».

Сперва определяем модели. Метод sync() синхронизирует структуру базы данных с определением моделей. Например, если для какой-то модели отстуствует соответствующая таблица в БД, то эта таблица создается.

Если в бд есть подобная таблица, но она не соответствует определению модели, то мы можем использовать параметр {force: true}, чтобы удалить таблицы и создать их заново, но уже с нужной нам структурой

1. Поясните понятие «транзакция».

Sequelize по умолчанию не использует транзакции. Однако для готового к производству использования Sequelize вам определенно следует настроить Sequelize для использования транзакций.

Sequelize поддерживает два способа использования транзакций:

* Неуправляемые транзакции: фиксация и откат транзакции должны выполняться пользователем вручную (путем вызова соответствующих методов Sequelize).
* Управляемые транзакции: Sequelize автоматически откатит транзакцию, если возникнет какая-либо ошибка, или зафиксирует транзакцию в противном случае. Кроме того, если CLS (локальное хранилище продолжения) включен, все запросы в обратном вызове транзакции автоматически получат объект транзакции.