

- Практическое задание: Работа с PostgreSQL и Node.js
 - Цель задания:
 - Задачи:
 - Дополнительные задания (по желанию):
 - Требования к выполнению:

Практическое задание: Работа с PostgreSQL и Node.js

Цель задания:

Получить практические навыки работы с PostgreSQL через Node.js, создав простое приложение для управления данными сотрудников.

Задачи:

1. Настройка базы данных:

- Создайте новую базу данных `company_db` в PostgreSQL.
- В базе данных создайте таблицу `employees` со следующими полями:
 - `id` (первичный ключ, автоматически инкрементируемый),
 - `name` (имя сотрудника, строковый тип),
 - `position` (должность сотрудника, строковый тип),
 - `email` (email сотрудника, строковый тип, уникальный),
 - `salary` (зарплата, числовой тип),
 - `is_active` (флаг активности, булевый тип, по умолчанию `true`).

2. Настройка Node.js проекта:

- Инициализируйте новый проект Node.js и установите пакет `pg` для работы с PostgreSQL.
- Создайте файл `db.js` для подключения к базе данных `company_db`.

3. Реализация CRUD операций:

- Создайте файлы `createEmployee.js`, `readEmployees.js`, `updateEmployee.js`, и `deleteEmployee.js` для выполнения CRUD операций с таблицей `employees`.

4. Задания:

- **Создание сотрудника:** В файле `createEmployee.js` реализуйте функционал добавления нового сотрудника в таблицу `employees`.
- **Чтение списка сотрудников:** В файле `readEmployees.js` реализуйте функционал чтения и вывода в консоль всех записей из таблицы `employees`.
- **Обновление данных сотрудника:** В файле `updateEmployee.js` реализуйте функционал обновления данных существующего сотрудника (например, изменение зарплаты).
- **Удаление сотрудника:** В файле `deleteEmployee.js` реализуйте функционал удаления сотрудника из таблицы по `id`.

Дополнительные задания (по желанию):

- Реализуйте дополнительные проверки в каждом скрипте (например, проверка на существование сотрудника перед обновлением или удалением).
- Создайте скрипт `index.js`, который будет предоставлять пользовательское меню для выполнения всех операций через консоль.

Требования к выполнению:

- Каждый скрипт должен корректно выполнять свою задачу.
- Код должен быть чистым, хорошо структурированным и комментированным.
- Для каждой операции предусмотреть обработку ошибок.



МАССОВОЕ БЕЗДЕЛЬЕ