Задача приложения, если коротко)

1. Некий «Оператор» обзванивает потенциальных клиентов, и тех, кто заинтересовался заносит данные в таблицу «Leads».
2. По данным таблицы «Leads» «Менеджер» делает детальное предложение о сотрудничестве. Тех, кто отказался отмечается статусом «спросить позже», а тех, кто согласился, на основе их информации в таблице «Leads» создает компанию для заказчика в таблице «Companies» (разбиваю их по разным таблицам, чтобы таблица «компании» не засорялась теми, кто не проявил интерес и в таблице компании были только компании, с которыми начали работу). Так же новую компанию может самостоятельно создать и юзер с ролью «Заказчик-Пользователь». Автоматически такой компании назначается: общая информация по компании, в поле «Менеджер» - текущий менеджер создающий данную компанию, в поле «Оператор» - оператор, который записал ранее лида на основе, которого создается данная компания (так же возможно в поле «Работники» рандомно записывается пользователь с ролью «Работник» из списка «Работников» данного менеджер, но возможно это не нужно и можно назначать «Работника» не компании, а «Заказу»). (в случае создания компании заказчиком – все назначается рандомно)).
3. Далее «Менеджер» выбирает эту компанию из списка своих компаний и делает для нее «Заказ» или список «Заказов», в нашем случае продуктом является только «написание отзыва» (потом будут разные). При появлении заказа (ему назначается «Работник, если мы его не добавили ранее на уровне компании) и «Работник» начинает работать с ним.
4. По выполнению заказа, компании выставляется счет и после оплаты статус заказа меняется на «выполнено» и уходит в архив или вниз списка заказов компании

Все юзеры хранятся в таблице users. Юзерам присваивается одна из ролей «Админ», «Оператор», «Менеджер», «Работник», «Заказчик или Пользователь».

У каждого юзера может быть список подчиненных юзеров по категориям. Например, у админа должен быть список вообще всех «Операторов», «Менеджеров» и «Работников», но не всех пользователей. У заказчика список его «Менеджеров», но «Оператор» и «Работники» должны быть пустыми. У пользователя с ролью «Менеджер», должен быть список его «Операторов» и «Работников», именно тех, которых ему назначили, а также список компаний, которые он будет вести (или список «Заказчиков», у которых есть компании в списке принадлежащих им компаний). И у пользователя «Работник» должен быть один или несколько «Менеджеров», но не должно быть «Операторов», видимо этот список должен быть пустым. У меня таблица получилась следующей:

|  |
| --- |
| Логин |
| Пароль |
| ФИО |
| Роль |
| Email |
| Номер телефона |
| Код активации |
| Активен или нет |
| Дата создания |
| Корзина |
| Список компаний |
| Список операторов |
| Список менеджеров |
| Список работников  И  Список пользователей под вопросом |

Главный вопрос: Как реализовать у юзера список «Операторов», «Менеджеров» и «Работников», так, чтобы текущий юзер могу получать доступ к полям каждого из них, но при этом не не нагружать систему? Я вижу 3 варианта: **1 вариант:** Сделать поля в таблице, которые будут ссылаться на эту же самую таблицу. Но мне кажется это невозможно. ведь если мы будем делать внешний ключ на id юзеров, привязанных к определенному менеджеру, то получится, что мы ссылаемся на одну и туже таблицу users, с внешним ключом user\_id на user\_id (например, user\_id\_1 будет иметь поле user\_id и ссылаться (или иметь связь) с [user\_id\_2, user\_id\_3, user\_id\_4], и я не уверен, что можно ссылаться таблице на саму же себя. **2 вариант:** это создать отдельные сущности «Операторов», «Менеджеров» и «Работников», куда будут прописываться id пользователей с соответствующими ролями, и потом уже user будет ссылаться на id этих «Операторов», «Менеджеров» и «Работников» и через который можно будет получить доступ к полям этих user-ов. **3 вариант:** Не создавать лишние сущности и не ссылаться на туже таблицу, а формировать список внутри сервиса программы, делая запрос к БД по нужным аргументам «, например, взять все логины юзеров, у которых Менеджер такой-то такой-то) и положить просто в списки «Операторов», «Менеджеров» и «Работников» в таблице users, эти логины в виде списка строк и при необходимости по этим спискам логинов находить нужных пользователей в БД и вытаскивать их при необходимости, например, формируя DTO?

Задача приложения, если коротко)

1. Некий «Оператор» обзванивает потенциальных клиентов, и тех, кто заинтересовался заносит данные в таблицу «Leads».
2. По данным таблицы «Leads» «Менеджер» делает детальное предложение о сотрудничестве. Тех, кто отказался отмечается статусом «спросить позже», а тех, кто согласился, на основе их информации в таблице «Leads» создает компанию для заказчика в таблице «Companies» (разбиваю их по разным таблицам, чтобы таблица «компании» не засорялась теми, кто не проявил интерес и в таблице компании были только компании, с которыми начали работу). Так же новую компанию может самостоятельно создать и юзер с ролью «Заказчик-Пользователь». Автоматически такой компании назначается: общая информация по компании, в поле «Менеджер» - текущий менеджер создающий данную компанию, в поле «Оператор» - оператор, который записал ранее лида на основе, которого создается данная компания (так же возможно в поле «Работники» рандомно записывается пользователь с ролью «Работник» из списка «Работников» данного менеджер, но возможно это не нужно и можно назначать «Работника» не компании, а «Заказу»). (в случае создания компании заказчиком – все назначается рандомно)).
3. Далее «Менеджер» выбирает эту компанию из списка своих компаний и делает для нее «Заказ» или список «Заказов», в нашем случае продуктом является только «написание отзыва» (потом будут разные). При появлении заказа (ему назначается «Работник, если мы его не добавили ранее на уровне компании) и «Работник» начинает работать с ним.
4. По выполнению заказа, компании выставляется счет и после оплаты статус заказа меняется на «выполнено» и уходит в архив или вниз списка заказов компании

Все юзеры хранятся в таблице users. Юзерам присваивается одна из ролей «Админ», «Оператор», «Менеджер», «Работник», «Заказчик или Пользователь».

У каждого юзера может быть список подчиненных юзеров по категориям. Например, у админа должен быть список вообще всех «Операторов», «Менеджеров» и «Работников», но не всех пользователей. У заказчика список его «Менеджеров», но «Оператор» и «Работники» должны быть пустыми. У пользователя с ролью «Менеджер», должен быть список его «Операторов» и «Работников», именно тех, которых ему назначили, а также список компаний, которые он будет вести (или список «Заказчиков», у которых есть компании в списке принадлежащих им компаний). И у пользователя «Работник» должен быть один или несколько «Менеджеров», но не должно быть «Операторов», видимо этот список должен быть пустым. У меня таблица получилась следующей:

-- Создание схемы

CREATE SCHEMA IF NOT EXISTS `otziv` DEFAULT CHARACTER SET utf8 ;

USE `otziv` ;

-- Таблица пользователей

CREATE TABLE IF NOT EXISTS `otziv`.`users` (

`id` INT NOT NULL AUTO\_INCREMENT,

`username` VARCHAR(16) NOT NULL,

`password` VARCHAR(32) NOT NULL,

`fio` VARCHAR(30) NOT NULL,

`email` VARCHAR(255) NOT NULL,

`phone\_number` VARCHAR(50) NOT NULL,

`activate\_code` VARCHAR(255) DEFAULT NULL,

`active` BIT(1) DEFAULT 0,

`create\_date` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP,

`operator\_id` INT NULL,

`manager\_id` INT NULL,

`worker\_id` INT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `email\_UNIQUE` (`email` ASC),

UNIQUE INDEX `phone\_number\_UNIQUE` (`phone\_number` ASC),

UNIQUE INDEX `username\_UNIQUE` (`username` ASC),

INDEX `operator\_id\_idx` (`operator\_id` ASC),

INDEX `manager\_id\_idx` (`manager\_id` ASC),

INDEX `worker\_id\_idx` (`worker\_id` ASC),

CONSTRAINT `operator\_fk` FOREIGN KEY (`operator\_id`) REFERENCES `users` (`id`) ON DELETE SET NULL,

CONSTRAINT `manager\_fk` FOREIGN KEY (`manager\_id`) REFERENCES `users` (`id`) ON DELETE SET NULL,

CONSTRAINT `worker\_fk` FOREIGN KEY (`worker\_id`) REFERENCES `users` (`id`) ON DELETE SET NULL

);

-- Таблица ролей

CREATE TABLE IF NOT EXISTS `otziv`.`roles` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(20) NOT NULL,

PRIMARY KEY (`id`)

);

-- Таблица связей между пользователями и ролями

CREATE TABLE IF NOT EXISTS `otziv`.`user\_roles` (

`user\_id` INT NOT NULL,

`role\_id` INT NOT NULL,

PRIMARY KEY (`user\_id`, `role\_id`),

CONSTRAINT `user\_roles\_user` FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`) ON DELETE CASCADE,

CONSTRAINT `user\_roles\_role` FOREIGN KEY (`role\_id`) REFERENCES `roles` (`id`) ON DELETE CASCADE

);

-- Таблица компаний

CREATE TABLE IF NOT EXISTS `otziv`.`companies` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(100) NOT NULL,

PRIMARY KEY (`id`)

);

-- Таблица связей между пользователями и компаниями

CREATE TABLE IF NOT EXISTS `otziv`.`user\_companies` (

`user\_id` INT NOT NULL,

`company\_id` INT NOT NULL,

PRIMARY KEY (`user\_id`, `company\_id`),

CONSTRAINT `user\_companies\_user` FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`) ON DELETE CASCADE,

CONSTRAINT `user\_companies\_company` FOREIGN KEY (`company\_id`) REFERENCES `companies` (`id`) ON DELETE CASCADE

);