**Домашнее задание №7**

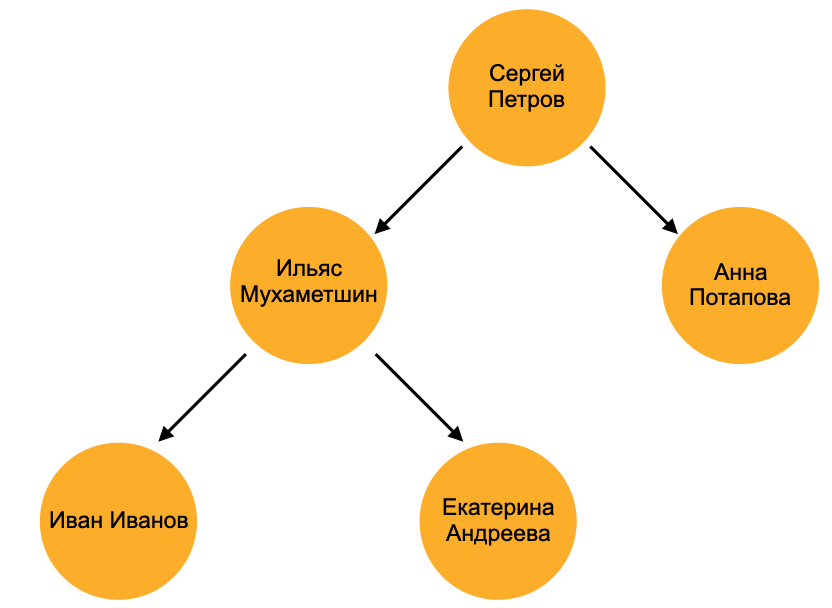
1. Примените пожалуйста [скрипт](https://drive.google.com/file/d/15CZE97IH7N9mgAvNSseu_61vxMf-wGqD/view?usp=sharing) для базы данных PostgreSQL. Если есть необходимость, то вы можете применить его заново, предварительно удалив таблицы командой

DROP TABLE *имя\_таблицы;*

Применить скрипт - это значит выполнить его через ваш используемый IDE (DataGrip, DBeaver , pgAdmin)

1. Необходимо написать SQL запросы к следующим задачам ниже. SQL запрос пишется в вашем IDE и прикладывается (как текст) вместе со скрином ответа в отдельном doc документе - который в свою очередь необходимо приложить к домашнему заданию и отправить преподавателю. Задание считается выполненным - если SQL запрос написан синтаксически корректно и возвращает ожидаемые данные на условие задачи.
2. Все зарегистрированные студенты познакомились друг с другом и решили организовать свой собственный технологический стартап. Ребята выстроили между собой следующие взаимоотношения в рамках проектной деятельности.

* Сергей Петров - лидер проекта
* Ильяс Мухаметшин - тимлид команды разработчиков
* Иван Иванов - frontend разработчик
* Екатерина Андреева - backend разработчик
* Анна Потапова - тимлид команды специалистов по тестированию

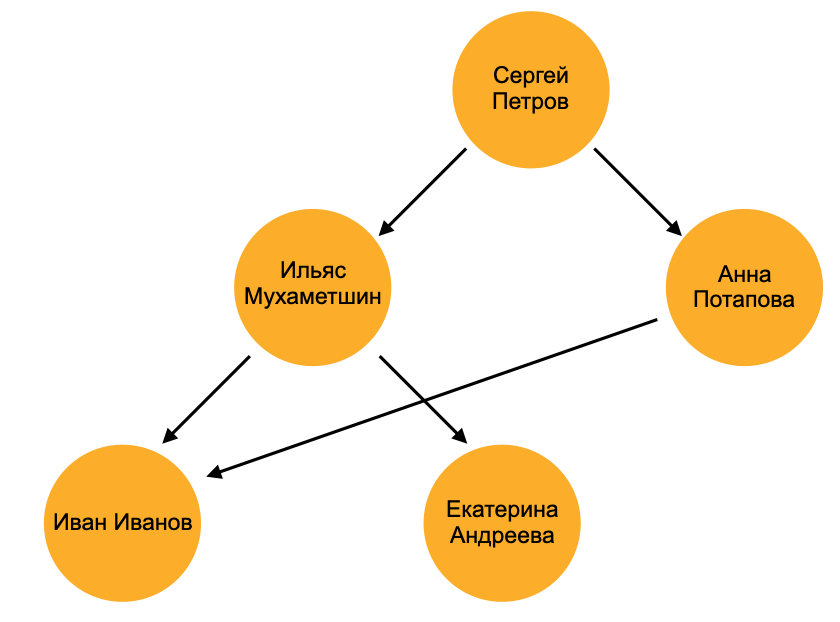


В связи с этой активностью необходимо расширить модель данных

1. Добавьте отдельную таблицу с зависимостями между участниками стартапа. Укажите внешние ключи необходимые для поддержания консистентности данных. Заполните таблицу соответствующими записями по зависимостям.
2. Добавьте таблицу-справочник , хранящую роли участников проекта и использующуюся как источник для понимания, какой студент имеет ту или иную роль. Заполните таблицу соответствующими записями.
3. Напишите SQL запрос который возвращает имена студентов, их роль на проекте и уровень вложенности от “лидера всего проекта” в следующем виде.

|  |  |  |  |
| --- | --- | --- | --- |
| **студент** | **роль** | **уровень** | **путь от лидера** |
| Сергей Петров | лидер проекта | 1 | Сергей Петров |
| Ильяс Мухаметшин | тимлид команды разработчиков | 2 | Сергей Петров -> Ильяс Мухаметшин |
| Анна Потапова | тимлид команды специалистов по тестированию | 2 | Сергей Петров ->Анна Потапова |
| Екатерина Андреева | backend разработчик | 3 | Сергей Петров -> Ильяс Мухаметшин ->Екатерина Андреева |
| Иван Иванов | frontend разработчик | 3 | Сергей Петров -> Ильяс Мухаметшин -> Иван Иванов |

1. Студент Иван Иванов решил параллельно выполнять роль “QA тестировщик” на проекте и стал подчиняться Анне Потаповой. Сделайте соответствующие изменения в данных вашей модели, чтобы выполнив запрос из пункта b) вы получили следующий результат.



|  |  |  |  |
| --- | --- | --- | --- |
| **студент** | **роль** | **уровень** | **путь от лидера** |
| Сергей Петров | лидер проекта | 1 | Сергей Петров |
| Ильяс Мухаметшин | тимлид команды разработчиков | 2 | Сергей Петров -> Ильяс Мухаметшин |
| Анна Потапова | тимлид команды специалистов по тестированию | 2 | Сергей Петров ->Анна Потапова |
| Екатерина Андреева | backend разработчик | 3 | Сергей Петров -> Ильяс Мухаметшин ->Екатерина Андреева |
| Иван Иванов | frontend разработчик | 3 | Сергей Петров -> Ильяс Мухаметшин -> Иван Иванов |
| Иван Иванов | QA тестировщик | 3 | Сергей Петров->Анна Потапова->Иван Иванов |

1. Смоделируйте ситуацию deadlock на любой таблице в вашей базе данных. Предоставьте ошибку (сделав скриншот), которая доказывает что ситуация “мертвой блокировки” случилась в рамках параллельных транзакций.
2. Смоделируйте ситуацию блокировок на уровне изоляции данных SERIALIZABLE на любой таблице в вашей базе данных. Предоставьте ошибку (сделав скриншот), которая доказывает что уровень изоляции SERIALIZABLE предотвращает обновление данных, которое случилось в параллельной транзакции с вашей.