ГБПОУ МО Физтех-колледж

ОТЧЕТ БД ПО ТЕМЕ:

Ресторан

Студент группы ИСП 3-1

Смирнов Игорь Юрьевич

Преподаватель:

Глянцев Денис Яковлевич

Долгопрудный 2021

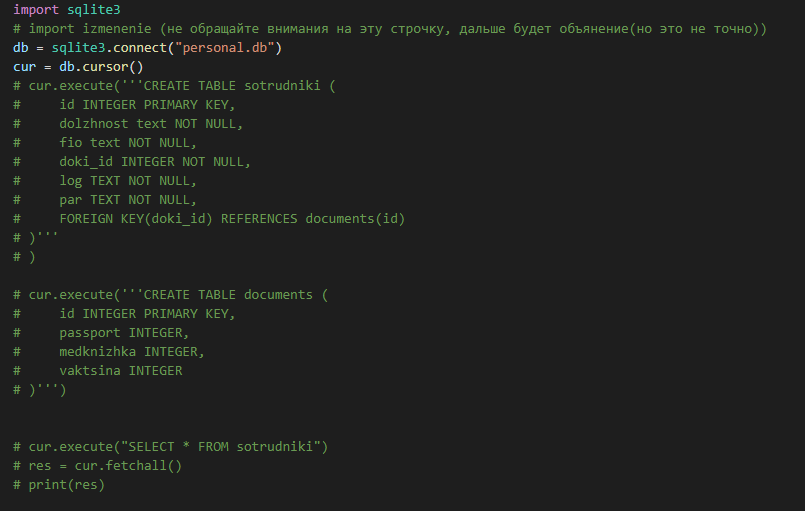
СОДЕРЖАНИЕ:

1. Предисловие
2. О самой программе
3. Итоги

ВВЕДЕНИЕ

Данный проект представляет из себя БД и функциональность, созданную в python. В этом комплексном задании есть множество расписанных функций, каждая из которых исправно выполняет свою поставленную задачу. Проект удался на славу и процесс его создания оказался весьма интересным и занятным. При редактировании кода было замечено множество недочетов, в ходе которых функции выполнялись некорректно. В последствии все было исправлено и допилено.

Данная программа имеет достаточно много реализованных функции, каждая из них выполняет свою роль. Ниже будет расписаны начало, середина и конец.



На данном скриншотеизображены две БД, которые связаны между собой. То, что они связаны, доказывает эта строка: FOREIGN KEY(doki\_id) REFERENCES documents(id)

Реализовать связь было решено только потому, что это казалось наиболее оптимальным способом работы с данными, но, как потом окажется, это не совсем так...

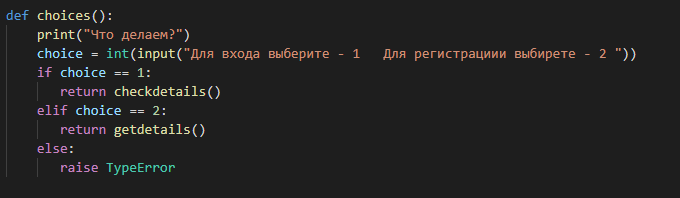
Самые последние три сточки - это просто проверка БД на функциональность, т.е. проверка корректные ли данные выведет или нет.



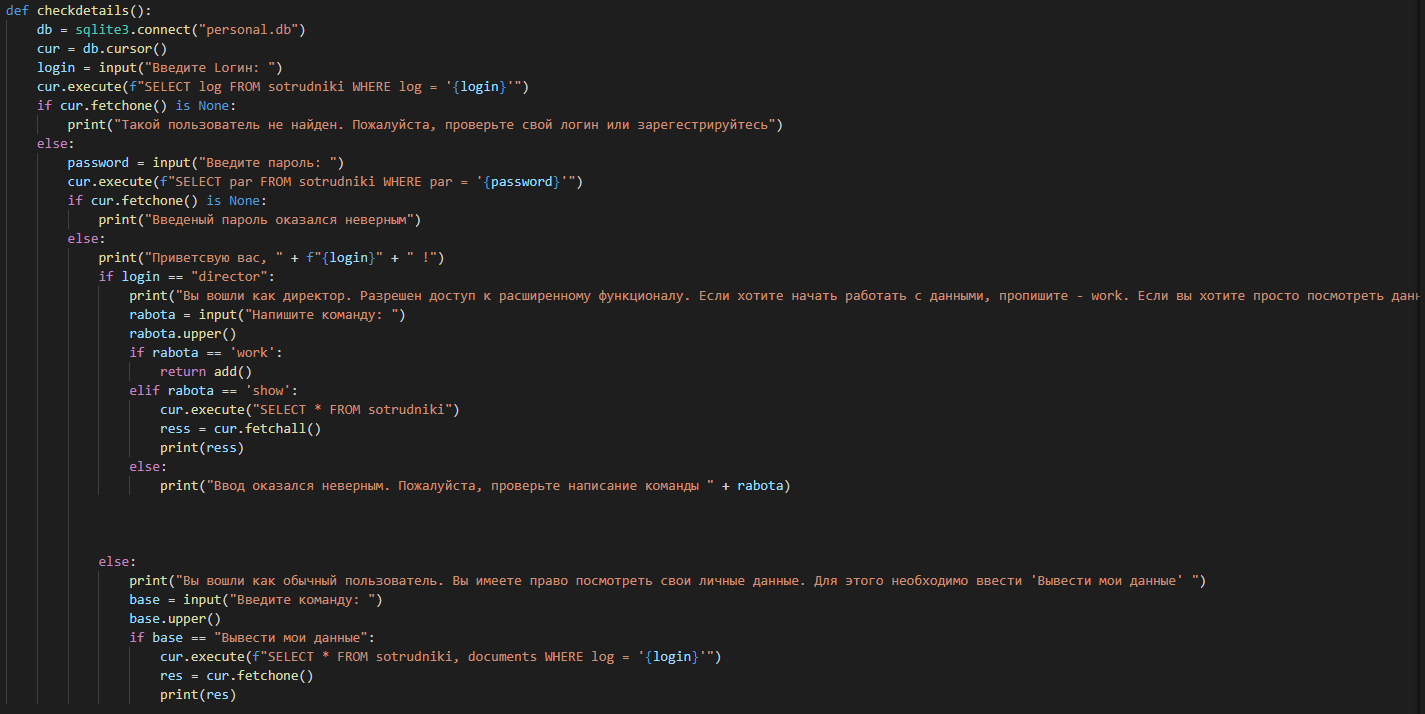
И, как оказалось, эти три строчки вывели корекктные данные по каждому сотруднику.

Следующим ходом стала реализация авторизации/регистрации в БД. Здесь возникли проблемы. Так как моих знаний по python на тот момент хоть и хватало, но моё долгое отсутствие в программировании дало о себе знать. Я даже не мог продумать у себя в голове код, но после долгих моих копаний в интернете и советов от друзей я пришел к вот такому коду:

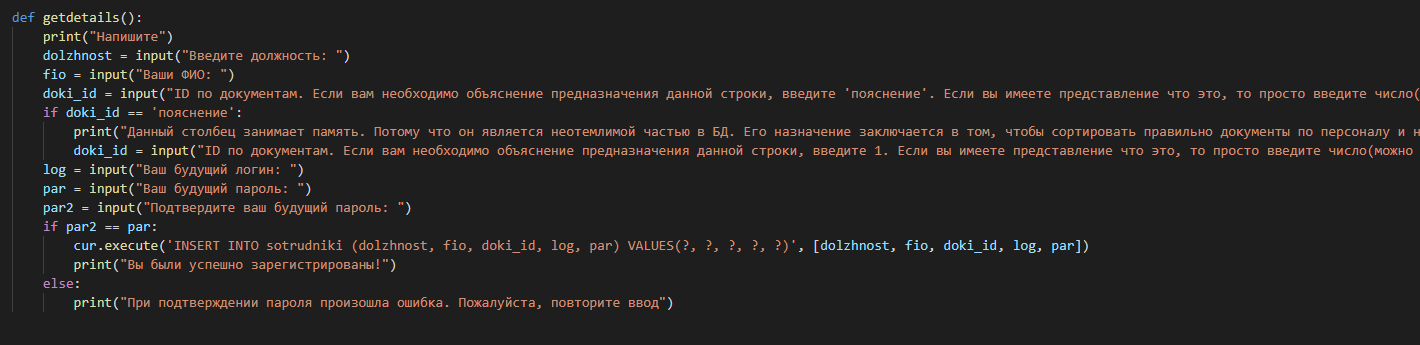
1 часть:



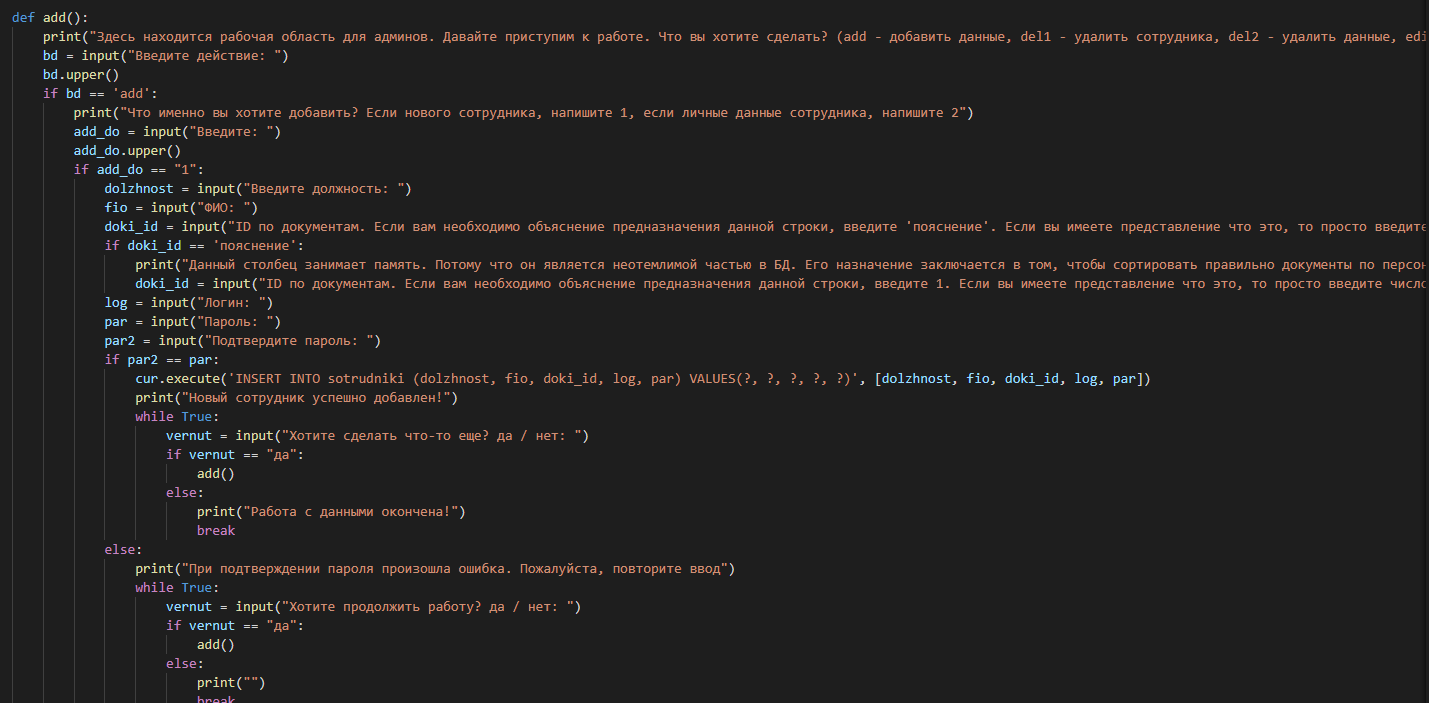
2 часть:

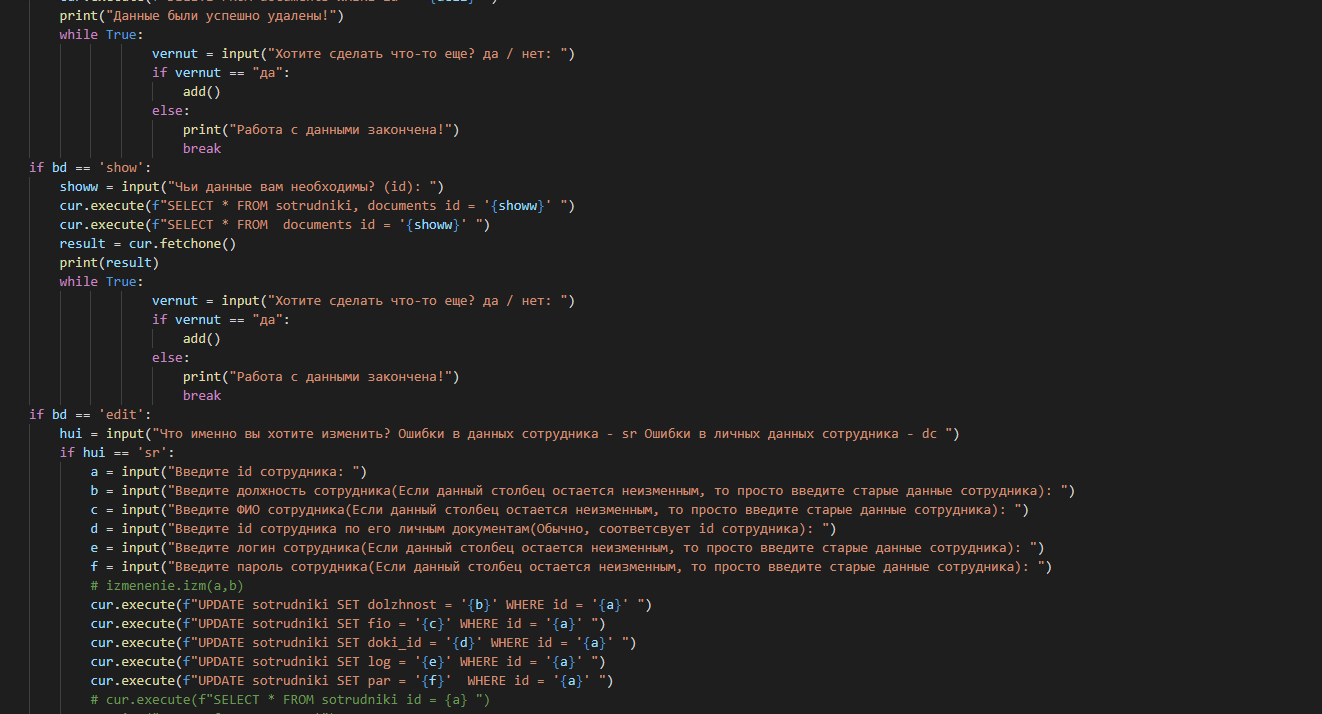


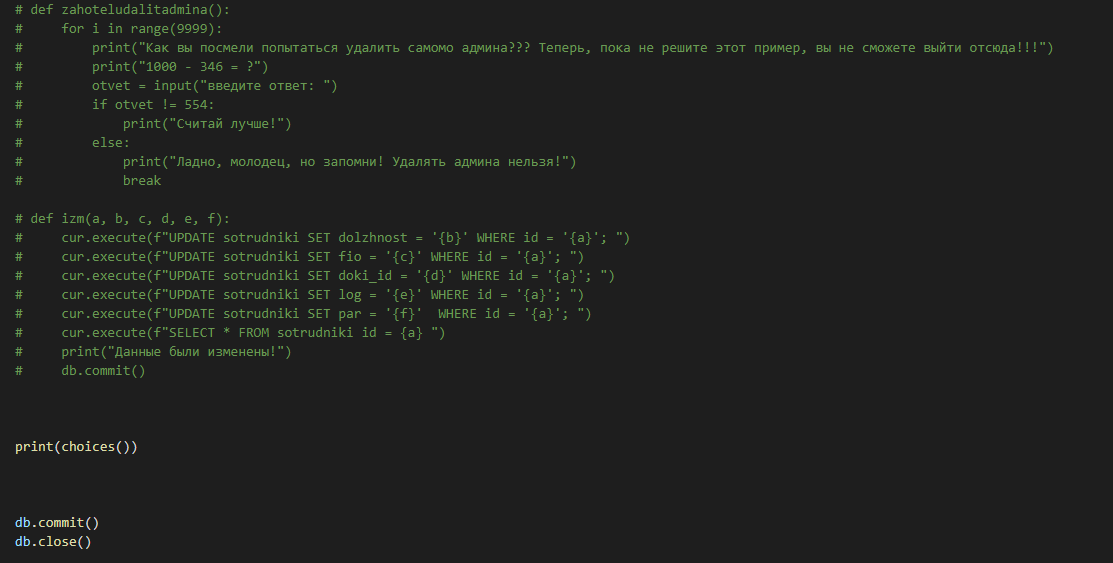
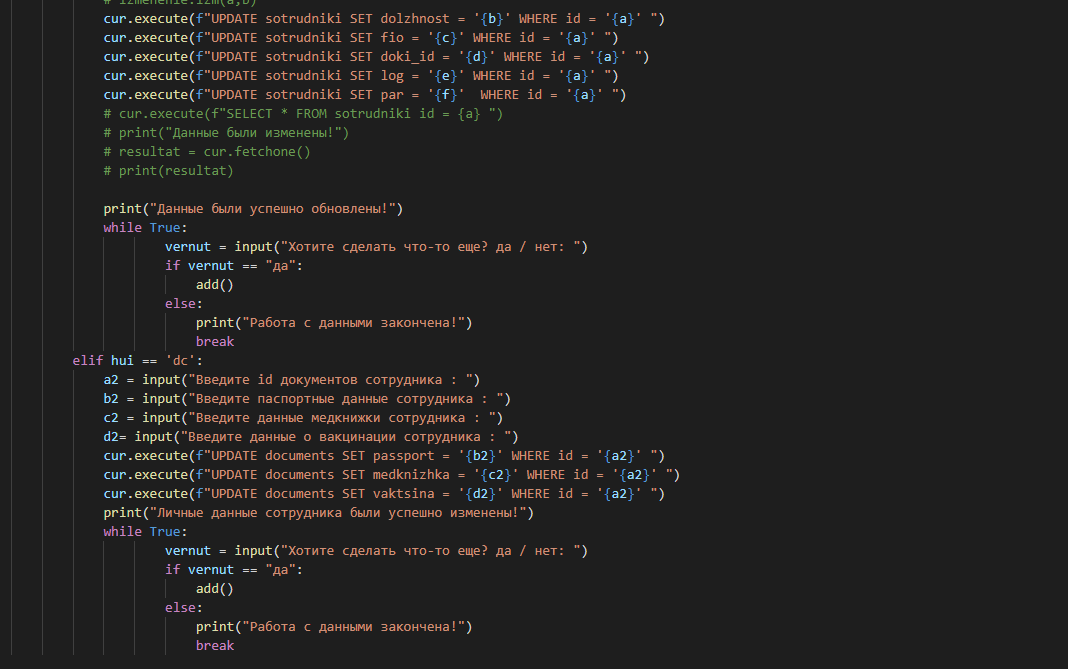
3 часть:



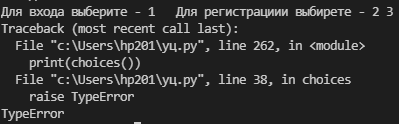
4 часть:





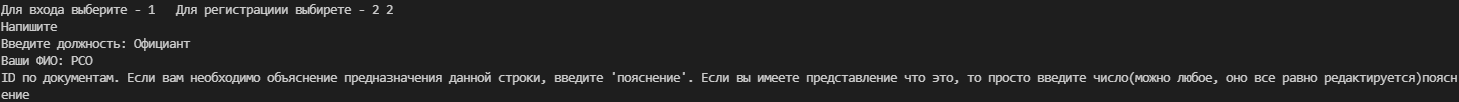
Как вы поняли, в 4 части весь основной код. Задокументированные части - мои попытки реализации каких-либо функций в программе.

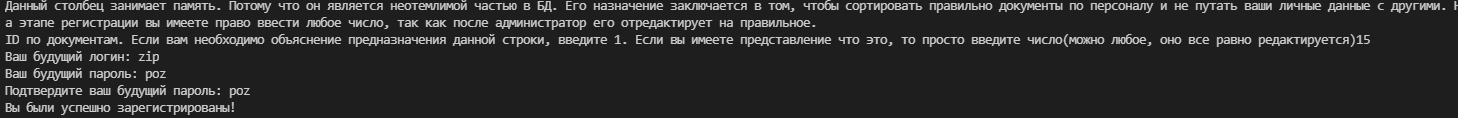
В 1 части идет, так сказать, обобщение и куда, чему ссылаться. На данном этапе избирается первое действие: войти или зарегистрироваться. Если ввести что-то помимо 1 ил 2, то программа выдаст ошибку:



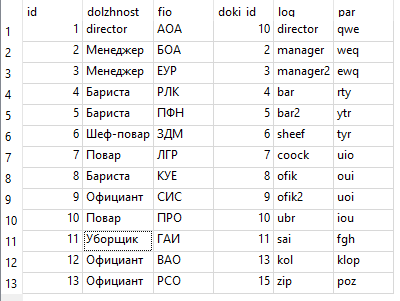
Если выбрать 1 вариант, то вы выберете действие такое как “вход”. В этом случае начнет свое применение функция def checkdetails():

, то есть вторая часть. Идет процесс авторизации. Думаю, нам сейчас нету смысла авторизовываться. Поэтому давайте для начала зарегистрируем обычный аккаунт.



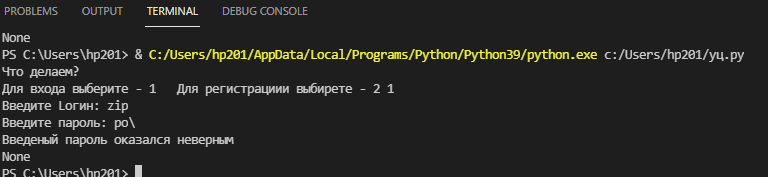


Это весь процесс регистрации. Т.е. нам необходимо ввести данные, которые будут, по сути, открытые или, по-другому, ваш статус в ресторане. Личные данные каждого сотрудника заполняет сам директор (он же админ). Наш аккаунт был успешно зарегистрирован. Давайте взглянем, появился ли он в БД:

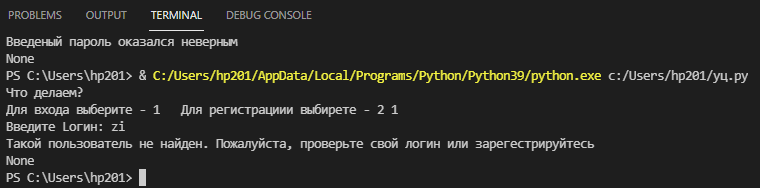


Да, под 13 id. Это наш аккаунт. Производим авторизацию. Так как это обычный аккаунт,редактировать данные не является возможным, есть право только на просмотр. Об этом вас предупредят.

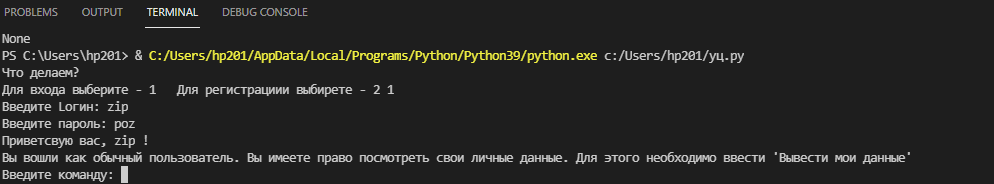
При вводе пароля случайно произошел миссклик и вот что выдала программа:



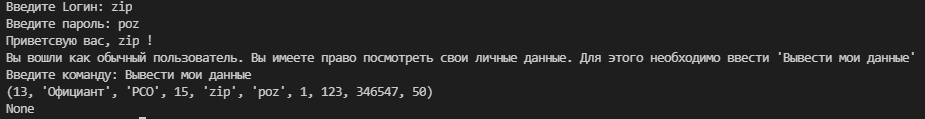
“Пароль оказался неверным”. То есть авторизация работает хорошо. Спросите: “А если я введу неправильный логин? Все равно пустит?”. Нет, конечно нет:



Если вы введете неверный логин, то вам выведут, что такого логина не существует и программа перестанет работать. Так, здесь реализованы малюсенькие SQL-инъекции. Теперь давайте, все-таки, войдем в аккаунт...



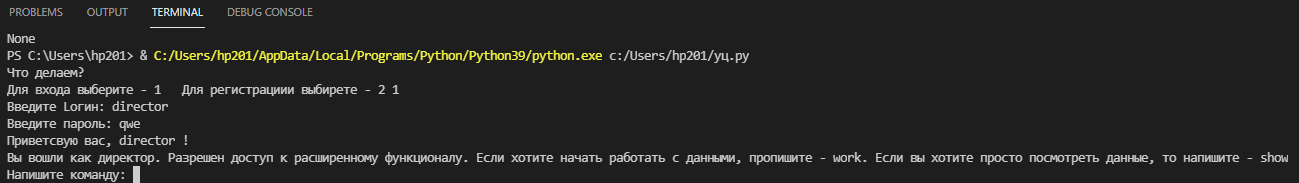
Когда вы вошли в аккаунт, у есть возможность только посмотреть свои личные данные:



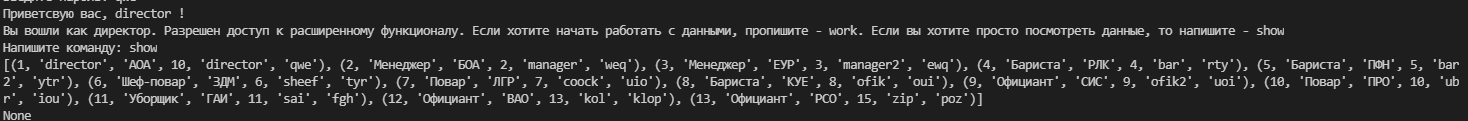
Кажется, админ успел заполнить личные данные сотрудника. Ладно, позже можно продемонстрировать функцию изменения данных.

Теперь перейдем в входу в админский аккаунт и его возможностям.

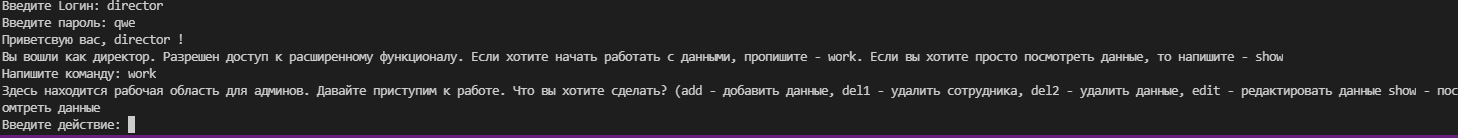
Админ - директор, которые может вертеть данными как захочет. У него есть доступ ко всему функционалу.



Мы вошли как директор и сходу нам предлагают либо перейти в рабочую область, либо просто вывести все данные.



Вывод данных работает прекрасно. Теперь давайте перейдем непосредственно к воздействию над данными.

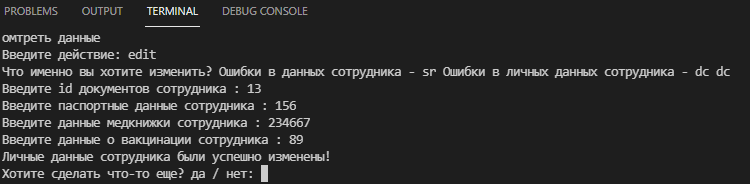


Как вы напишите команду “work”, вас перекинет в рабочую область, где вы получите расширенный функционал по БД. Прошу подметить! Если вы введете команду неверно, вас попросят перепроверить написание и выведут ее на экран (небольшое эстетическое удовольствие).

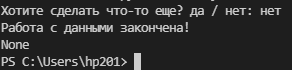
Дальше имеем право выбрать наши следующие действия по БД. Давайте для начала, как и обещал, испытаем изменение данных. Кстати, эта функция вызвала у меня наибольшие затруднения. Потому что как бы я не пытался написать этот код, она не работала. Писал и отдельным файлом, и через отдельную функция и пытался использовать другое написание. Все бестолку. В итоге все не работало из-за одной фальшивой строчки с выводом из изменений данных. Убрав ее, все заработало как надо.

Как только мы введем “edit”, нас сразу предложат изменить данные либо в статусе сотрудника, либо в его личных данных. Так как тут набедокурили с личными данными сотрудника, давайте измених их, т.е. “dc”.

Это наши изначальные данные



Данные были успешно изменены. Дальше нам предлагают сделать что-то еще. То есть, если вы не закончили работу с БД, то можно не выходя из аккаунта поработать еще. Допустим вы не хотим еще работать с данными и напишем просто “нет”.



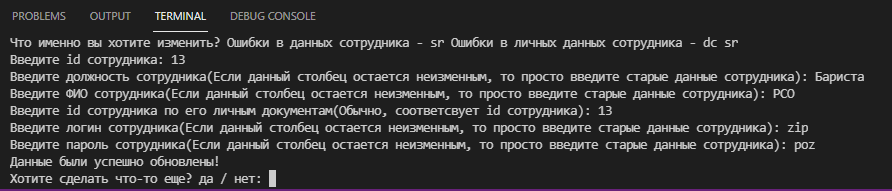
И, как мы видим, работа окончена. Так же видно и изменение в данных сотрудника:



Получается, что изменение работает как швейцарские часы! Точно также можно и менять статус сотрудника:



Давайте сделаем его Бариста:

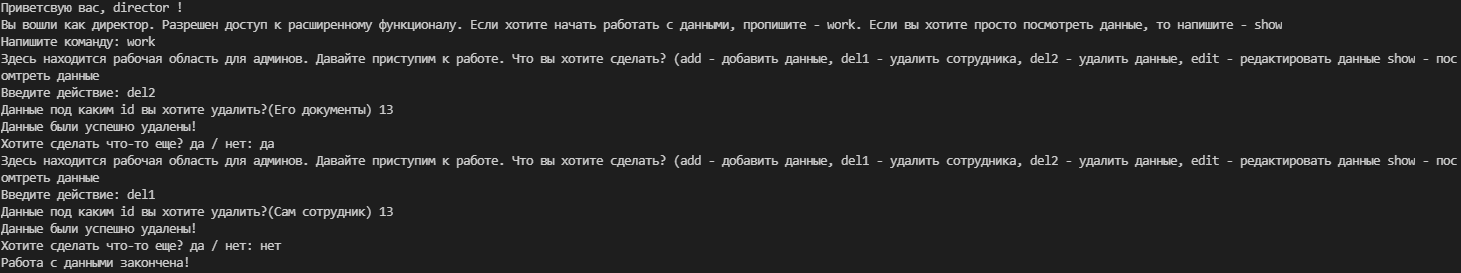


Данные были успешно обновлены. Давайте это проверим.

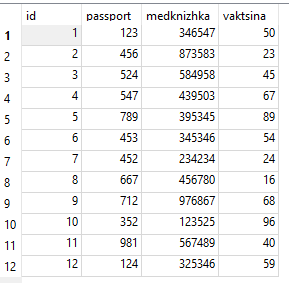


Теперь нам следует опробовать оставшиеся функции: удаление, добавление и просмотр данных по сотрудникам.

Начнем с удаления. Нам предложат либо del1, либо del2. То есть, удаление и тех, и тех данных сотрудника. Давайте проверим как это работает:



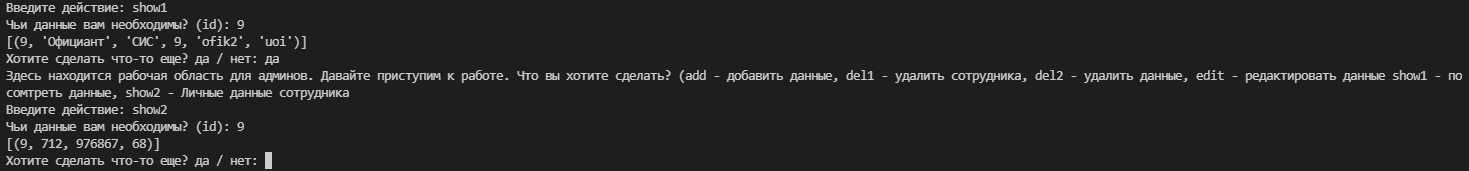
В программе все работает, а как обстоят дела с БД?





Как мы видим, данные были успешно удалены. Данная функция тоже работает безотказно!

Теперь давайте опробуем показ данных по каждому сотруднику:



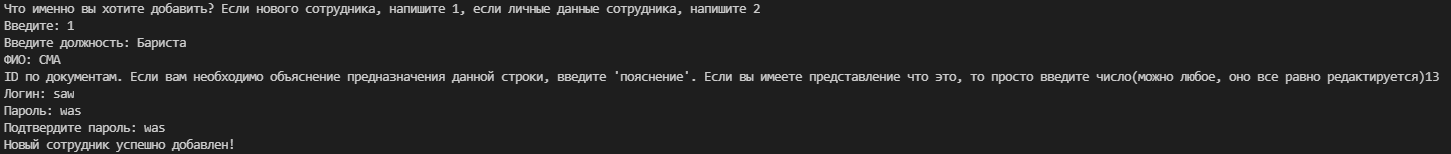
Здесь можно лицезреть функцию вывода данных по сотрудникам. Прошу заметить, что разделение личных данных и статуса было сделано специально для удобства работы с данными.

Теперь остался процесс добавления через админа. По своей сути, все проходит почти также, как и в регистрации. Только теперь можно добавить еще и личные данные каждого сотрудника.

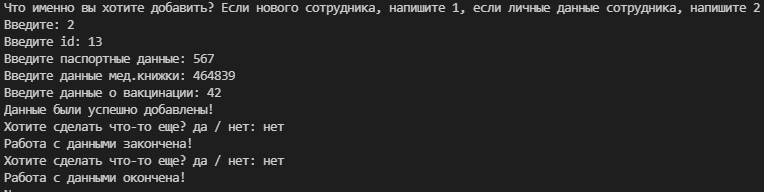
После введения функции “add” нам предложит выполнить добавление либо сотрудника, либо его личных данных.



Давайте снова добавим сотрудника и после его личные данные.

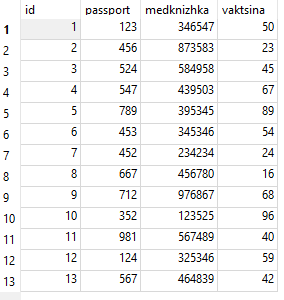


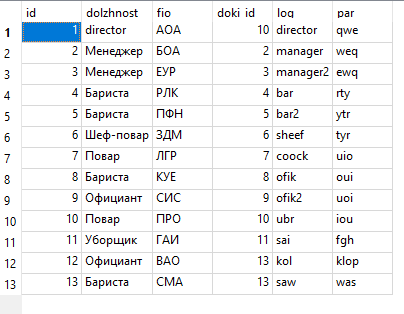
Здесь идет процесс добавления самого сотрудника.



А здесь идет процесс добавления его личных данных. Все функции работают исправно, без каких-либо нареканий. Все изменяется, добавляется, удаляется и регистрируется.

Ниже будут приведены скриншоты/доказательства о том, что данные были успешно добавлены в БД и теперь там хранятся, занимая свое почетное место





Под ведем итоги: процесс создания такой БД был весьма себе интересным. При создании я столкнулся с множеством ошибок, которые мне предстояло исправить. В конечном итоге, я потратил небольшое количество нервных клеток и смог сделать вполне хорошую программу, которую в будущем можно будет доработать лучше и сделать полноценной.

Ссылка на скачивание БД и программы:

https://disk.yandex.ru/d/dRQWTSIcusI5xw