1. Взять назначенную Jira Task в работу

2. Установить docker

3. Зарегистрироваться на github: https://github.com/

4. В github сделать fork репозитория: https://github.com/smith1984/sde\_test\_db

5. Сделать репозиторий приватным

6. Выдать права на доступ к репозиторию пользователю smith1984

7. Установить клиент git

8. Склонировать созданный репозиторий на локальную машину

9. Установить среду разработки IntelliJ IDEA

10. Открыть склонированный проект в среде разработки

11. В директории bash реализовать в скрипте run\_init\_db.sh следующее:

- скачать docker образ postgres

- запустить образ postgres со следующими параметрами:

POSTGRES\_PASSWORD="@sde\_password012"

POSTGRES\_USER="test\_sde"

POSTGRES\_DB="demo"

примонтировать директорию проекта sql в поднятом контейнере

- заполнить БД данными через выполнение скрипта ./sql/init\_db/demo.sql

12. Установить DBeaver

13. Подключиться к поднятой базе через клиента DBeaver

14. Описание БД находится в репозитории проекта в файле description.pdf (первоисточник: https://postgrespro.ru/education/demodb)

15. Написать и протестировать скрипт (sql запросы) на локальной базе:

Создать таблицу results c атрибутами id (INT), response (TEXT), где

* id – номер запроса из списка ниже
* response – результат запроса

Если запись содержит несколько атрибутов, то их значения должны объединяться в одно, через конкатенацию с разделителем “|”

Если результат возвращает несколько записей, то все записи записываются в результирующую таблицу, с id номера запроса и отсортированные по возрастанию по всем выводимым атрибутам.

1. Вывести максимальное количество человек в одном бронировании
2. Вывести количество бронирований с количеством людей больше среднего значения людей на одно бронирование
3. Вывести количество бронирований, у которых состав пассажиров повторялся два и более раза, среди бронирований с максимальным количеством людей (п.1)?
4. Вывести номера брони и контактную информацию по пассажирам в брони (passenger\_id, passenger\_name, contact\_data) с количеством людей в брони = 3
5. Вывести максимальное количество перелётов на бронь
6. Вывести максимальное количество перелётов на пассажира в одной брони
7. Вывести максимальное количество перелётов на пассажира
8. Вывести контактную информацию по пассажиру(ам) (passenger\_id, passenger\_name, contact\_data) и общие траты на билеты, для пассажира потратившему минимальное количество денег на перелеты
9. Вывести контактную информацию по пассажиру(ам) (passenger\_id, passenger\_name, contact\_data) и общее время в полётах, для пассажира, который провёл максимальное время в полётах
10. Вывести город(а) с количеством аэропортов больше одного
11. Вывести город(а), у которого самое меньшее количество городов прямого сообщения
12. Вывести пары городов, у которых нет прямых сообщений исключив реверсные дубликаты
13. Вывести города, до которых нельзя добраться без пересадок из Москвы?
14. Вывести модель самолета, который выполнил больше всего рейсов
15. Вывести модель самолета, который перевез больше всего пассажиров
16. Вывести отклонение в минутах суммы запланированного времени перелета от фактического по всем перелётам
17. Вывести города, в которые осуществлялся перелёт из Санкт-Петербурга 2016-09-13
18. Вывести перелёт(ы) с максимальной стоимостью всех билетов
19. Выбрать дни в которых было осуществлено минимальное количество перелётов
20. Вывести среднее количество вылетов в день из Москвы за 09 месяц 2016 года
21. Вывести топ 5 городов у которых среднее время перелета до пункта назначения больше 3 часов

16. Перенести скрипт в репозиторий в ./sql/main/calc.sql

17. Закоммитить изменения в репозитории (./bash/run\_init\_db.sh, ./sql/main/calc.sql)

18. Запушить изменения с локального репозитория в удаленный

19. Прикрепить ссылку на репозиторий в Jira Task

20. Закрыть Jira Task до срока сдачи решения