**Проектная работа по модулю**

**“SQL и получение данных”**

Поздравляем, вы в финале обучения на курсе SQL! Осталось совсем немного: закрепить все пройденные занятия и полученные знания в финальной работе.

**Для выполнения работы Вам необходимо:**

1. Перейти по ссылке и ознакомиться с описанием базы данных: <https://edu.postgrespro.ru/bookings.pdf>
2. Подключиться к базе данных **avia** по одному из следующих вариантов:

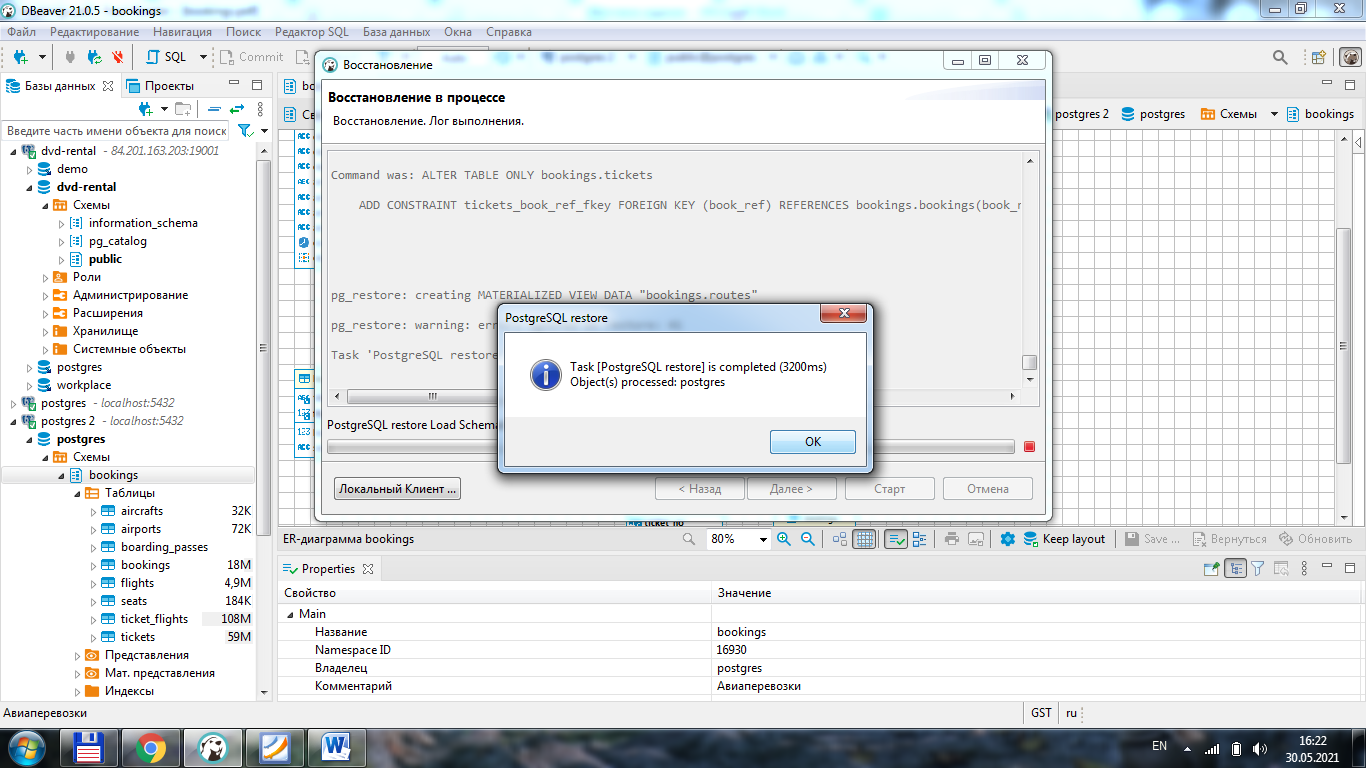
* облачное подключение, те же настройки, что и у dvd-rental, только название базы **demo**, схема **bookings**
* импорт sql запроса из sql файла, представленных на 2 странице описания базы
* восстановить базу из \*.backup файла по ссылке [avia](https://drive.google.com/file/d/1U15gYuu_ZFE2sQMN32GXK3phBLcUl6OM/view?usp=sharing)

1. Оформить работу согласно “Приложения №1” в формате \*.pdf или \*.doc
2. Создать запросы, позволяющие ответить на вопросы из “Приложения №2”, решения должны быть приложены в формате \*.sql одним файлом.
3. Отправить работу на проверку

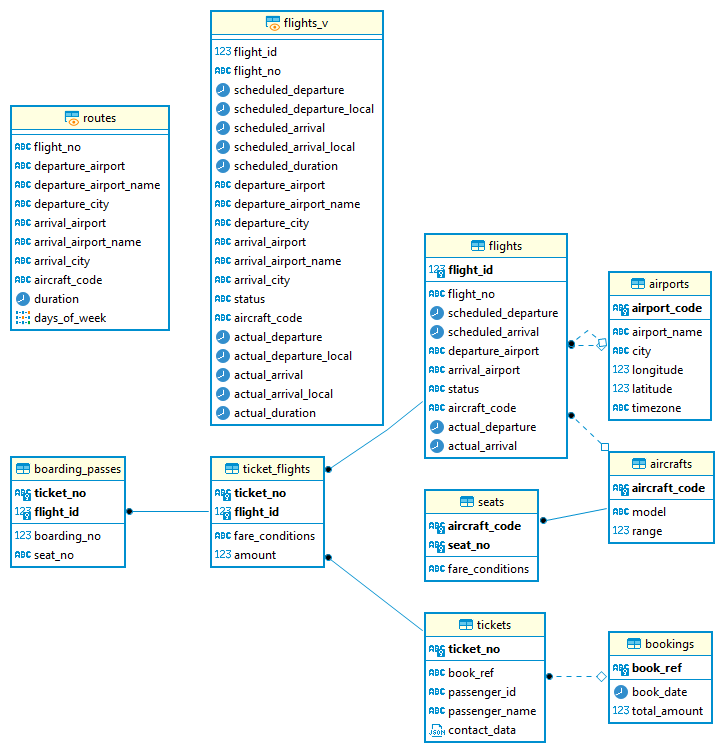
Итоговая работа

1. В работе использовался **локальный** тип подключения.

* если база была развернута из \*.sql или **\*.backup файла**, необходимо приложить скриншот успешного импорта или восстановления



1. Скриншот ER-диаграммы из DBeaver`a согласно Вашего подключения.



1. Краткое описание БД - из каких таблиц и представлений состоит.

**Таблицы**:

bookings.aircrafts

bookings.airports

bookings.boarding\_passes

bookings.bookings

bookings.flights

bookings.seats

bookings.ticket\_flights

bookings.tickets

**Представление** "bookings.flights\_v"

**Материализованное представление** bookings.routes

1. Развернутый анализ БД - описание таблиц, логики, связей и бизнес области (частично можно взять из описания базы данных, оформленной в виде анализа базы данных). Бизнес задачи, которые можно решить, используя БД.
2. Список SQL запросов из приложения №2 с описанием логики их выполнения.
3. Задача скопирована и вставлена из описания к БД
4. Ищу самолет с максимальной дальностью перелета в подзапросе. Обогащаю рейсы информацией об аэропортах. Через условие вывожу аэропорты с рейсами, содержащими самолет с максимальной дальностью полета.
5. Вычитаю в таблице рейсы из фактического время вылета плановое время вылета. Через условие убираю не состоявшиеся рейсы. Отсортировываю таблицу по задержке в обратном порядке и ограничиваю вывод.
6. Обогатил таблицу бронирования информацией о билетах, перелетах и посадочных талонах. Посадочные талоны через left join, потому что требуется увидеть выданные и не выданные талоны. Через условие вывел брони, по которым не выдавались посадочные талоны.
7. Через условие, искал только в состоявшихся полетах. В подзапросе нашел количество занятых мест (проданные билеты на каждый рейс). Обогатил рейсы информацией о всего местах и занятых местах. Используя арифметику, вывел %. Используя оконные функции, отсортировал по дате рейсы в аэропортах и вывел накапливаемую сумму.
8. За логическую основу взял <https://coderoad.ru/9161404/SQL-%D0%BE%D0%BA%D1%80%D1%83%D0%B3%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D0%BD%D1%82%D0%BE%D0%B2-%D1%87%D1%82%D0%BE%D0%B1%D1%8B-%D1%81%D0%B4%D0%B5%D0%BB%D0%B0%D1%82%D1%8C-%D1%81%D1%83%D0%BC%D0%BC%D1%83-100-1-3-%D0%BA%D0%B0%D0%BA-0-34-0-33-0-33>
9. В ОТВ посчитал стоимость для статусов Business и Economy. Через условие в рамках перелета сравнил стоимости. Дополнил таблицу информацией об аэропорте и городе.
10. Сделал таблицу «каждый с каждым» у аэропортов. Вычел из результата все прямые рейсы. Сделал материализованное представление. Во втором запросе обогатил предыдущий информацией о городах вылета и прилета.
11. Обогатил рейсы информацией о широте и долготе аэропортов вылета и прилета, а также информацией дальности полетов самолетов. Далее воспользовался предлагаемой для решения формулой и вывел результат.