**Проектная работа по модулю**

**“DWH”**

**Что необходимо сделать?**

**Источником данных будет база данных bookings, знакомая вам по курсу SQL**

1. подключаемся к yandex облаку. Host - 84.201.153.170, порт - 19001, база данных - demo, пользователь - netology, пароль - NetoSQL2019

***Комментарий.*** *Если у Вас после курса по SQL осталась развернутая база по авиаперевозкам - можно работать с ней.*

2. Описание базы данных: <https://edu.postgrespro.ru/bookings.pdf>

3. Работаем в схеме **bookings**

В рамках итоговой работы будет необходимо:

1. Сохранить таблицы с фактами и размерностями в csv файлы (каждая таблица - отдельный csv-файл).

В качестве альтернативы можно запустить чистую базу данных в докере и создать там таблицы с фактами и размерностями. \*

1. Наполнить базу данными из бд bookings при помощи ETL

**Требуемые таблицы:**

Fact\_Flights - содержит совершенные перелеты. Если в рамках билета был сложный маршрут с пересадками - каждый сегмент учитываем независимо

* Пассажир
* Дата и время вылета (факт)
* Дата и время прилета (факт)
* Задержка вылета (разница между фактической и запланированной датой в секундах)
* Задержка прилета (разница между фактической и запланированной датой в секундах)
* Самолет
* Аэропорт вылета
* Аэропорт прилета
* Класс обслуживания
* Стоимость

Dim\_Calendar - справочник дат

Dim\_Passengers - справочник пассажиров

Dim\_Aircrafts - справочник самолетов

Dim\_Airports - справочник аэропортов

Dim\_Tariff - справочник тарифов (Эконом/бизнес и тд)

**Правила оценивания:**

| **#** | **Описание** | **Баллы** |
| --- | --- | --- |
| **1** | SQL-скрипт создания таблицы фактов и всех справочников | 40 |
| **2** | Реализован ETL загрузки таблицы фактов и всех справочников (кроме DIM\_Calendar - его можно прям в SQL сгенерировать) | 40 |
| **3** | Реализованы проверки качества данных.  Для каждой таблицы в хранилище необходимо придумать и реализовать как минимум 3-4 проверки качества данных. Например, дата фактического вылета не может быть больше сегодняшнего дня. Описание каждой проверки должно быть в документации (см пункт 4)  Некачественные строки записываются в отдельные rejected-таблицы | 30 |
| **4** | Предоставлена документация, описывающая процедуру ETL | 30 |
| **5** | SQL-скрипты, трансформации и документация выложены на github | 10 |

Итого - максимум можно набрать 150 баллов. Для зачета необходимо набрать минимум 90 баллов.

\* Если вы сделали какое-то задание не для всех таблиц (например, реализовали процедуру качества данных только для таблицы фактов, и не реализовали для справочников) - в 3 и 4 задании таблица фактов оценивается в 10 баллов, каждая из таблиц-справочников (кроме Dim\_Calendar) - в 5. Сделали только для таблицы фактов без ошибок - 10 баллов из 30.

**В качестве решения надо прислать:**

1. Ссылку на github или архив с материалами (SQL-скрипты, трансформации, документация)
2. ER-диаграмму созданной вами базы данных хранилища
3. Скриншоты ETL-процессов