**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
3. Отправить работу на проверку

Приложение №1

Итоговая работа

1. В работе использовался **облачный** 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.

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Описание** |
| aircrafts | таблица | Самолеты |
| airports | таблица | Аэропорты |
| boarding\_passes | таблица | Посадочные талоны |
| bookings | таблица | Бронирования |
| flights | таблица | Рейсы |
| flights\_v | представление | Рейсы |
| routes | мат. предст. | Маршруты |
| seats | таблица | Места |
| ticket\_flights | таблица | Перелеты |
| tickets | таблица | Билеты |

1. Развернутый анализ БД - описание таблиц, логики, связей и бизнес области (частично можно взять из описания базы данных, оформленной в виде анализа базы данных)

*Данная база данных содержит в себе информацию об авиаперевозках по России.*

*Таблица Aircrafts (Самолеты) - справочник по моделям самолетов. Состоит из:*

* *aircraft\_code (код самолета) - первичный ключ. Уникальный трехзначный для каждой модели. Ссылки извне из таблиц flights (перелеты) и seats (места).*
* *Model (модель самолета)*
* *Range (максимальная дальность полета, км) - должна быть > 0.*

*Таблица Airports (аэропорты) - справочник по аэропортам. Состоит из:*

* *airport\_code (Код аэропорта) - трехбуквенный код. Является первичным ключом. Ссылки извне из таблицы flights (перелеты) из столбцов arrival\_airport и departure\_airport*
* *airport\_name (Название аэропорта)*
* *city (Город) - может служить для определения аэропортов одного города.*
* *longitude (Координаты аэропорта: долгота)*
* *latitude (Координаты аэропорта: широта)*
* *timezone (Временная зона аэропорта)*

*Таблица Boarding\_passes (посадочные талоны). Состоит из:*

* *ticket\_no (Номер билета)*
* *flight\_id (Идентификатор рейса)*
* *boarding\_no (Номер посадочного талона) - последовательные номера в порядке регистрации пассажиров на рейс. Уникальны в рамках одного рейса (flight\_id).*
* *seat\_no (Номер места) - Уникальны в рамках одного рейса (flight\_id), чтобы не было двух пассажиров на одно место.*

*Посадочные талоны появляются при регистрации на рейс, т.е. за сутки до планируемой даты вылета. Идентифицируются номером билета и идентификатором рейса (составной первичный ключ, должен быть в таблице ticket\_flights.*

*Таблица Bookings (бронирования) - основная сущность. Состоит из:*

* *book\_ref (номер бронирования) - первичный ключ и внешний ключ из таблицы tickets - шестизначная комбинация букв и цифр*
* *Book\_date – дата бронирования - открывается за месяц до даты рейса, можно забронировать билет себе и другим пассажирам*
* *total\_amount – полная сумма бронирования - хранит общую стоимость всех перелетов всех пассажиров в бронировании*

*Таблица Flights (рейсы). Состоит из:*

* *flight\_id (Идентификатор рейса) - первичный суррогатный ключ*
* *flight\_no (Номер рейса) - часть естественного составного ключа*
* *scheduled\_departure (Время вылета по расписанию) - часть естественного составного ключа*
* *scheduled\_arrival (Время прилёта по расписанию) - должно быть больше scheduled\_departure*
* *departure\_airport (Аэропорт отправления) - должен быть в таблице Airports*
* *arrival\_airport (Аэропорт прибытия) - должен быть в таблице Airports*
* *status (Статус рейса)*
* *aircraft\_code (Код самолета) - должен быть в таблице Aircrafts*
* *actual\_departure (Фактическое время вылета)*
* *actual\_arrival (Фактическое время прилёта)*

*Рейс всегда соединяет только 2 точки - аэропорт вылета и прилета. Коды аэропортов ссылаются на таблицу Аэропорты. Рейсы с одним номером имеют одинаковые пункты вылета и прилета, но будут отличаться датой отправления.*

*Каждый рейс имеет запланированные и реальные дату и время вылета и прилета, а также Статус (scheduled, on time, delayed, departured, arrived, cancelled).*

*Также таблица Рейсы содержит Код самолета (ссылается на таблицу Самолеты)*

*Таблица Seats (места). Состоит из:*

* *aircraft\_code (код самолета) - внешний ключ из таблицы aircrafts (самолеты)*
* *Seat\_no (номер места) - определяет схему салона каждой модели самолета. Вместе с aircraft\_code (код самолета) образуют первичный составной ключ, чтобы в одном самолете не было повторяющихся мест*
* *Fare\_conditions (класс обслуживания) - закреплен за каждым местом. Может быть Economy, Comfort или Business.*

*Таблица ticket\_flights (перелеты). Состоит из:*

* *Ticket\_no (номер билета) - часть первичного составного ключа. Должен быть в таблице tickets (билеты)*
* *Flight\_id (идентификатор рейса) - часть первичного составного ключа. Должен быть в таблице flights (рейсы)*
* *Fare\_conditions (класс обслуживания) - может быть Economy, Comfort или Business.*
* *amount (стоимоть перелета) - должен быть >0*

*Перелеты входят в билет. Может быть один или несколько перелетов. Несколько перелетов может быть если нет прямого рейса или билет взят “туда-обратно”. Предполагается что в одном бронировании все билеты имеют одинаковый набор перелетов.*

*Таблица Tickets (билеты) - каждому пассажиру выписывается отдельный билет. Билет имеет уникальный номер и содержит данные о пассажире. Из-за того, что имя и номер паспорта можно поменять каждый пассажир считается уникальным. Состоит из:*

* *Ticket\_no (номер билета) - первичный ключ*
* *Book\_ref (номер бронирования) - внешний ключ на bookings (бронирования)*
* *Passenger\_id, passenger\_name, contact\_data – 3 столбца информации о пассажире*

*Представление flight\_v (рейсы) - содержит дополнительную информацию:*

* *расшифровку данных об аэропорте вылета (departure\_airport, departure\_airport\_name, departure\_city)*
* *расшифровку данных об аэропорте прибытия (arrival\_airport, arrival\_airport\_name, arrival\_city)*
* *местное время вылета (scheduled\_departure\_local, actual\_departure\_local)*
* *местное время прибытия (scheduled\_arrival\_local, actual\_arrival\_local)*
* *продолжительность полета (scheduled\_duration, actual\_duration).*

*Материализованное представление routes (маршруты). Содержит информацию о маршрутах, без привязки к конкретным датам и состоит из:*

* *flight\_no (Номер рейса)*
* *departure\_airport (Код аэропорта отправления)*
* *departure\_airport\_name (Название аэропорта отправления)*
* *departure\_city (Город отправления)*
* *arrival\_airport (Код аэропорта прибытия)*
* *arrival\_airport\_name (Название аэропорта прибытия)*
* *arrival\_city (Город прибытия)*
* *aircraft\_code (Код самолета)*
* *duration (Продолжительность полета)*
* *days\_of\_week (Дни недели, когда выполняются рейсы)*

1. Список SQL запросов с описанием логики их выполнения - в sql файле

Баллы за оформление:

1. 0 - облачная база, 10 - локальная база
2. 5
3. 10
4. 20
5. 15

Итого: максимум 60 баллов.

Для зачета необходимо набрать **минимум 30 баллов**.

Приложение №2

|  |  |  |
| --- | --- | --- |
| № | Вопрос | В решении обязательно должно быть использовано |
| 1 | В каких городах больше одного аэропорта? |  |
| 2 | В каких аэропортах есть рейсы, выполняемые самолетом с максимальной дальностью перелета? | - Подзапрос |
| 3 | Вывести 10 рейсов с максимальным временем задержки вылета | - Оператор LIMIT |
| 4 | Были ли брони, по которым не были получены посадочные талоны? | - Верный тип JOIN |
| 5 | Найдите свободные места для каждого рейса, их % отношение к общему количеству мест в самолете.  Добавьте столбец с накопительным итогом - суммарное количество вывезенных пассажиров из аэропорта за день. Т.е. в этом столбце должна отражаться сумма - сколько человек уже вылетело из данного аэропорта на этом или более ранних рейсах за сегодняшний день | - Оконная функция  - Подзапрос |
| 6 | Найдите процентное соотношение перелетов по типам самолетов от общего количества. | - Подзапрос  - Оператор ROUND |
| 7 | Были ли города, в которые можно добраться бизнес - классом дешевле, чем эконом-классом в рамках перелета? | - CTE |
| 8 | Между какими городами нет прямых рейсов? | - Декартово произведение в предложении FROM - Представления  - Оператор EXCEPT |
| 9 | Вычислите расстояние между аэропортами, связанными прямыми рейсами, сравните с допустимой максимальной дальностью перелетов в самолетах, обслуживающих эти рейсы \* | - Оператор RADIANS или использование sind/cosd |

\* - В облачной базе координаты находятся в столбце airports\_data.coordinates - работаете, как с массивом. В локальной базе координаты находятся в столбцах airports.longitude и airports.latitude.

Кратчайшее расстояние между двумя точками A и B на земной поверхности (если принять ее за сферу) определяется зависимостью:

d = arccos {sin(latitude\_a)·sin(latitude\_b) + cos(latitude\_a)·cos(latitude\_b)·cos(longitude\_a - longitude\_b)}, где latitude\_a и latitude\_b — широты, longitude\_a, longitude\_b — долготы данных пунктов, d — расстояние между пунктами, измеряемое в радианах длиной дуги большого круга земного шара.

Расстояние между пунктами, измеряемое в километрах, определяется по формуле:

L = d·R, где R = 6371 км — средний радиус земного шара.

Баллы за запросы:

1. 10
2. 15
3. 15
4. 15
5. 25
6. 25
7. 25
8. 35
9. 35

Итого: максимум 200 баллов.

Для зачета необходимо набрать **минимум 130 баллов**.