Итоговая работа

1. В работе использовался облачный тип подключения.
2. Скриншот ER-диаграммы из DBeaver`a согласно Вашего подключения.
3. Краткое описание БД - из каких таблиц и представлений состоит.
4. 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.
5. Список SQL запросов из приложения №2 с описанием логики их выполнения.

Решение  
  
Описание

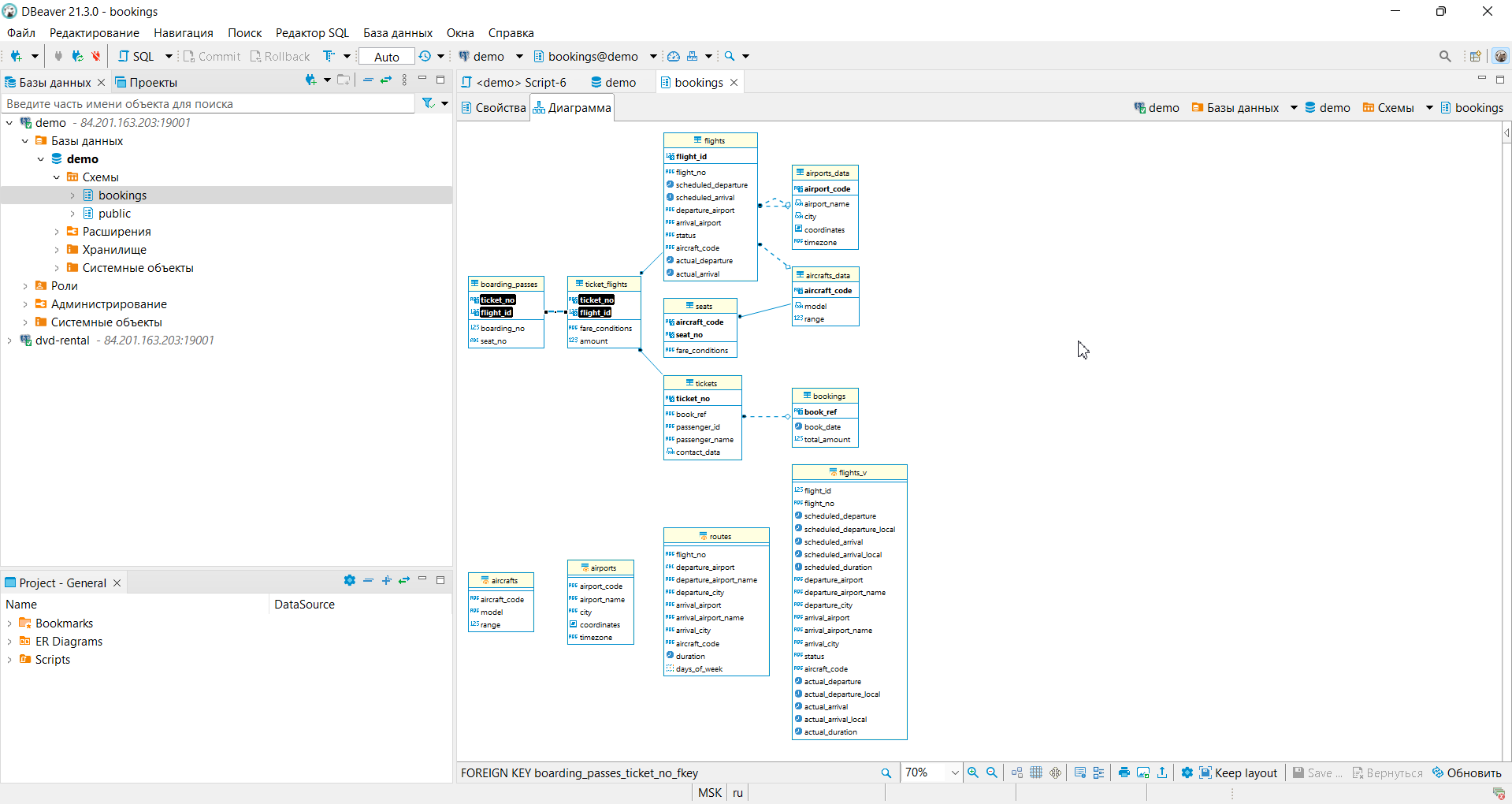
 В облачной базе координаты находятся в столбце airports\_data.coordinates - работаете, как с массивом. В локальной базе координаты находятся в столбцах airports.longitude и airports.latitude.

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 измеряется 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.



Список отношений

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | 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 |

**Таблица bookings.aircrafts**

Каждая модель воздушного судна идентифицируется своим трехзначным кодом (aircraft\_code). Указывается также название модели (model) и максимальная дальность полета в километрах (range).

**Таблица bookings.airports**

Аэропорт идентифицируется трехбуквенным кодом (airport\_code) и имеет свое имя (airport\_name).

Для города не предусмотрено отдельной сущности, но название (city) указывается и может служить для того, чтобы определить аэропорты одного города. Также указывается широта (longitude), долгота (latitude) и часовой пояс (timezone).

**Таблица bookings.boarding\_passes**

При регистрации на рейс, которая возможна за сутки до плановой даты отправления, пассажиру выдается посадочный талон.

Он идентифицируется также, как и перелет — номером билета и номером рейса. Посадочным талонам присваиваются последовательные номера (boarding\_no) в порядке регистрации пассажиров на рейс (этот номер будет уникальным только в пределах данного рейса). В посадочном талоне указывается номер места (seat\_no).

**Таблица bookings.bookings**

Пассажир заранее (book\_date, максимум за месяц до рейса) бронирует билет себе и, возможно, нескольким другим пассажирам.

Бронирование идентифицируется номером (book\_ref, шестизначная комбинация букв и цифр). Поле total\_amount хранит общую стоимость включенных в бронирование перелетов всех пассажиров.

**Таблица bookings.flights**

Естественный ключ таблицы рейсов состоит из двух полей — номера рейса (flight\_no) и даты отправления (scheduled\_departure). Чтобы сделать внешние ключи на эту таблицу компактнее, в качестве первичного используется суррогатный ключ (flight\_id).

Рейс всегда соединяет две точки — аэропорты вылета (departure\_airport) и прибытия (arrival\_airport). Такое понятие, как «рейс с пересадками» отсутствует: если из одного аэропорта до другого нет прямого рейса, в билет просто включаются несколько необходимых рейсов.

У каждого рейса есть запланированные дата и время вылета (scheduled\_departure) и прибытия (scheduled\_arrival). Реальные время вылета (actual\_departure) и прибытия (actual\_arrival) могут отличаться: обычно не сильно, но иногда и на несколько часов, если рейс задержан.

Статус рейса (status) может принимать одно из следующих значений:

• Scheduled Рейс доступен для бронирования. Это происходит за месяц до плановой даты вылета; до этого запись о рейсе не существует в базе данных.

• On Time Рейс доступен для регистрации (за сутки до плановой даты вылета) и не задержан.

• Delayed Рейс доступен для регистрации (за сутки до плановой даты вылета), но задержан. • Departed Самолет уже вылетел и находится в воздухе.

• Arrived Самолет прибыл в пункт назначения.

• Cancelled Рейс отменен.

**Таблица bookings.ticket\_flights**

Перелет соединяет билет с рейсом и идентифицируется их номерами.

Для каждого перелета указываются его стоимость (amount) и класс обслуживания (fare\_conditions).

**Таблица bookings.tickets**

Билет имеет уникальный номер (ticket\_no), состоящий из 13 цифр.

Билет содержит идентификатор пассажира (passenger\_id) — номер документа, удостоверяющего личность, — его фамилию и имя (passenger\_name) и контактную информацию (contact\_date).

Ни идентификатор пассажира, ни имя не являются постоянными (можно поменять паспорт, можно сменить фамилию), поэтому однозначно найти все билеты одного и того же пассажира невозможно

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

Над таблицей flights создано представление flights\_v, содержащее 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 bookings.routes**

Таблица рейсов содержит избыточность: из нее можно было бы выделить информацию о маршруте (номер рейса, аэропорты отправления и назначения), которая не зависит от конкретных дат рейсов.

Бизнес-задачи, которые можно решить с помощью данной БД

1. Эффективность частоты перелётов
2. Узнать какой приток новых пользователей за нужный период
3. Узнать какой отток пользователей за нужный период
4. Какие маршруты пользуются особой популярностью у пользователей
5. Собрать статистику об отношении времени года и популярных маршрутов
6. Узнать какие маршруты больше загружены и в какой период
7. Отследить клиентопоток
8. Узнать сколько клиентов осуществляют бронь и потом и как долго планируют поездку
9. Узнать самую популярную авиакомпанию
10. Узнать люди с какого региона пользуются чаще всего услугами авиакомпаний