Приложение №1

**Проектная работа**

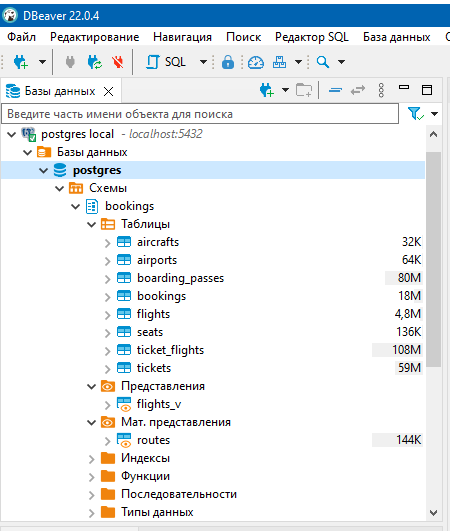
**по модулю**

**“SQL и получение данных”**

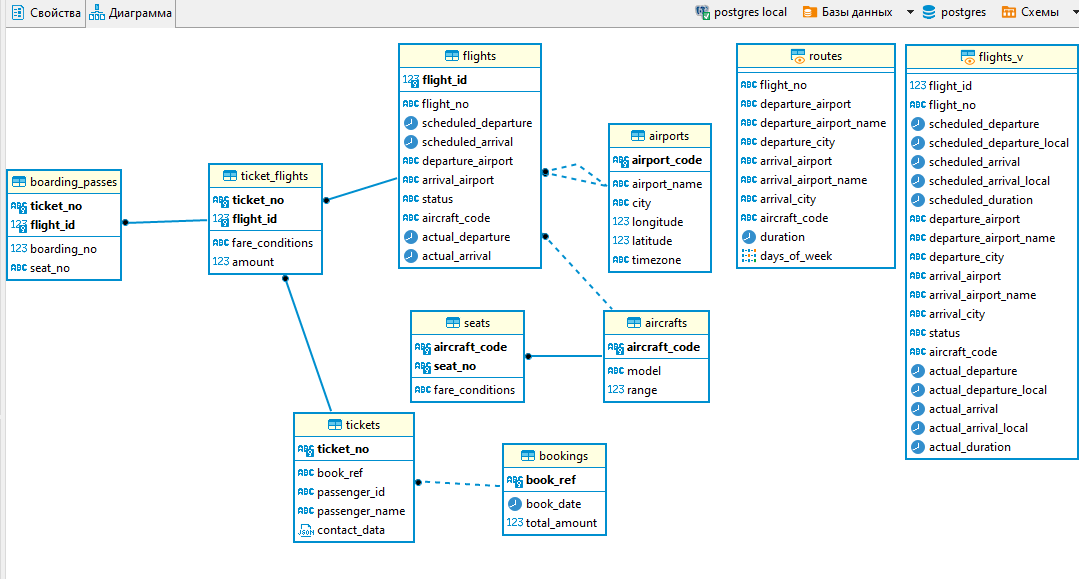
База данных **«Avia**»

В работе использовался локальный тип подключения:

* скриншот подключения:



* Скриншот ER-диаграммы:



### 

### **КРАТКОЕ ОПИСАНИЕ БАЗЫ ДАННЫХ**

База данных “Авиаперевозки” содержит данные авиаперелетов по России

Список объектов базы данных:

* **Таблицы**:

Bookings

Tickets

Tcket\_flights

Boarding\_passes

Flights

Airports

Aircrafts

Seats

* **Представление:**

Flights\_v

* **Материализованное представление:**

Routes

**АНАЛИЗ БАЗЫ ДАННЫХ**

### Описание объектов

#### **Таблица Bookings (Бронирования)**

Основная сущность.

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.

* Индексы:

PRIMARY KEY, btree (book\_ref)

* Ссылки извне:

TABLE "tickets" FOREIGN KEY (book\_ref) REFERENCES bookings(book\_ref)

#### **Таблица** **Tickets (Билеты)**

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.

* Индексы:

PRIMARY KEY, btree (ticket\_no)

* Ограничения внешнего ключа:

FOREIGN KEY (book\_ref) REFERENCES bookings(book\_ref)

* Ссылки извне:

TABLE "ticket\_flights" FOREIGN KEY (ticket\_no) REFERENCES tickets(ticket\_no)

#### **Таблица Tcket\_flights (Перелеты)**

Отношение билетов к перелетам.

Перелет соединяет билет с рейсом и идентифицируется их номерами.

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

* Индексы:

PRIMARY KEY, btree (ticket\_no, flight\_id)

* Ограничения-проверки:

CHECK (amount >= 0)

CHECK (fare\_conditions IN ('Economy', 'Comfort', 'Business'))

* Ограничения внешнего ключа:

FOREIGN KEY (flight\_id) REFERENCES flights(flight\_id)

FOREIGN KEY (ticket\_no) REFERENCES tickets(ticket\_no)

* Ссылки извне:

TABLE "boarding\_passes" FOREIGN KEY (ticket\_no, flight\_id) REFERENCES ticket\_flights(ticket\_no, flight\_id)

#### **Таблица Boarding\_passes (Посадочные талоны)**

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).

* Индексы:

PRIMARY KEY, btree (ticket\_no, flight\_id)

UNIQUE CONSTRAINT, btree (flight\_id, boarding\_no)

UNIQUE CONSTRAINT, btree (flight\_id, seat\_no)

* Ограничения внешнего ключа:

FOREIGN KEY (ticket\_no, flight\_id) REFERENCES ticket\_flights(ticket\_no, flight\_id)

#### **Таблица Flights (Рейсы)**

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.

Рейсы с одним номером имеют одинаковые пункты вылета и назначения, но будут отличаться датой отправления.

Один рейс обслуживается одним типом самолета.

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):

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

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

Delayed - 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 - отменен.

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).

* Индексы:

PRIMARY KEY, btree (flight\_id)

UNIQUE CONSTRAINT, btree (flight\_no, scheduled\_departure)

* Ограничения-проверки:

CHECK (scheduled\_arrival > scheduled\_departure)

CHECK ((actual\_arrival IS NULL) OR

((actual\_departure IS NOT NULL AND actual\_arrival IS NOT NULL) AND

(actual\_arrival > actual\_departure)))

CHECK (status IN ('On Time', 'Delayed', 'Departed', 'Arrived', 'Scheduled', 'Cancelled'))

* Ограничения внешнего ключа:

FOREIGN KEY (aircraft\_code) REFERENCES aircrafts(aircraft\_code)

FOREIGN KEY (arrival\_airport) REFERENCES airports(airport\_code)

FOREIGN KEY (departure\_airport) REFERENCES airports(airport\_code)

* Ссылки извне:

TABLE "ticket\_flights" FOREIGN KEY (flight\_id) REFERENCES flights(flight\_id)

#### **Таблица Airports (Аэропорты)**

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).

* Индексы:

PRIMARY KEY, btree (airport\_code)

* Ссылки извне:

TABLE "flights" FOREIGN KEY (arrival\_airport) REFERENCES airports(airport\_code)

TABLE "flights" FOREIGN KEY (departure\_airport) REFERENCES airports(airport\_code)

#### **Таблица Aircrafts (Самолеты)**

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

Каждая модель самолета имеет только одну компоновку салона.

* Индексы:

PRIMARY KEY, btree (aircraft\_code)

* Ограничения-проверки: CHECK (range > 0)
* Ссылки извне:

TABLE "flights" FOREIGN KEY (aircraft\_code) REFERENCES aircrafts(aircraft\_code)

TABLE "seats" FOREIGN KEY (aircraft\_code) REFERENCES aircrafts(aircraft\_code) ON DELETE CASCADE

#### **Таблица Seats (Места)**

Места определяют схему салона каждой модели. Каждое место определяется своим номером (seat\_no) и имеет закрепленный за ним класс обслуживания (fare\_conditions) — Economy, Comfort или Business.

* Индексы:

PRIMARY KEY, btree (aircraft\_code, seat\_no)

* Ограничения-проверки:

CHECK (fare\_conditions IN ('Economy', 'Comfort', 'Business'))

* Ограничения внешнего ключа:

FOREIGN KEY (aircraft\_code) REFERENCES aircrafts(aircraft\_code) ON DELETE CASCADE

**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. Именно такая информация и составляет материализованное представление routes.

1. Бизнес задачи

Информацию, которая содержится в базе данных “Авиаперевозки”, можно использовать для решения ряда бизнес задач. Комплексный анализ поможет выявить проблемы и найти пути их решения. Например:

* Анализ заполняемости рейсов ставит вопрос об эффективном использовании авиапарка, оптимальности затрат и прибыльности рейсов с низкой заполняемостью. В результате этого можно рассмотреть вопрос об отправке самолетов меньшей вместимостью на определенные направления, снизить частоту рейсов когда это невыгодно.
* Расчет расстояния между аэропортами позволит использовать маршруты с минимальным расстоянием для экономии топлива.
* Анализ количества перевезенных пассажиров позволит изучить спрос по направлениям, принять решение по сокращению или расширению количества прямых рейсов, оптимизировать рейсы с пересадками, увеличить частоту рейсов на популярных маршрутах.
* На основании анализа потока пассажиров по каждому аэропорту можно принять решение по перераспределению рейсов между аэропортами, тем самым сняв нагрузку с переполненных.
* Изучение пассажирских предпочтений и покупательской способности поможет правильно спланировать таргетированную рекламу. Привлечь новых клиентов и удержать старых тем самым увеличив прибыль.
* Анализ сроков задержки рейсов и изучение их закономерностей, могут привести к выявлению системных проблем в компании.