**Дипломный проект**

**SQL и получение данных**

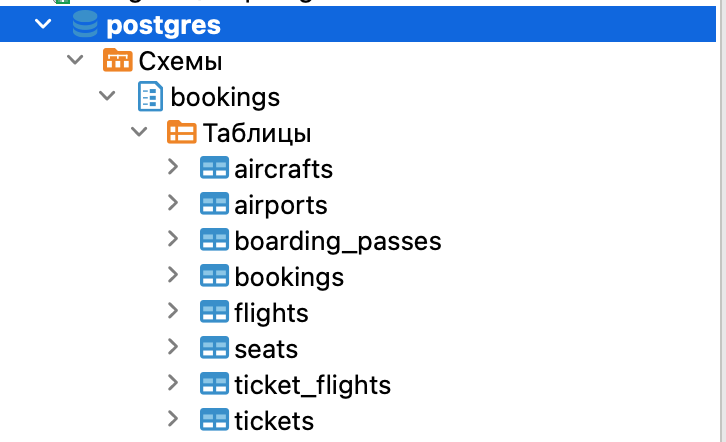


Полуэктова Татьяна

март 2021

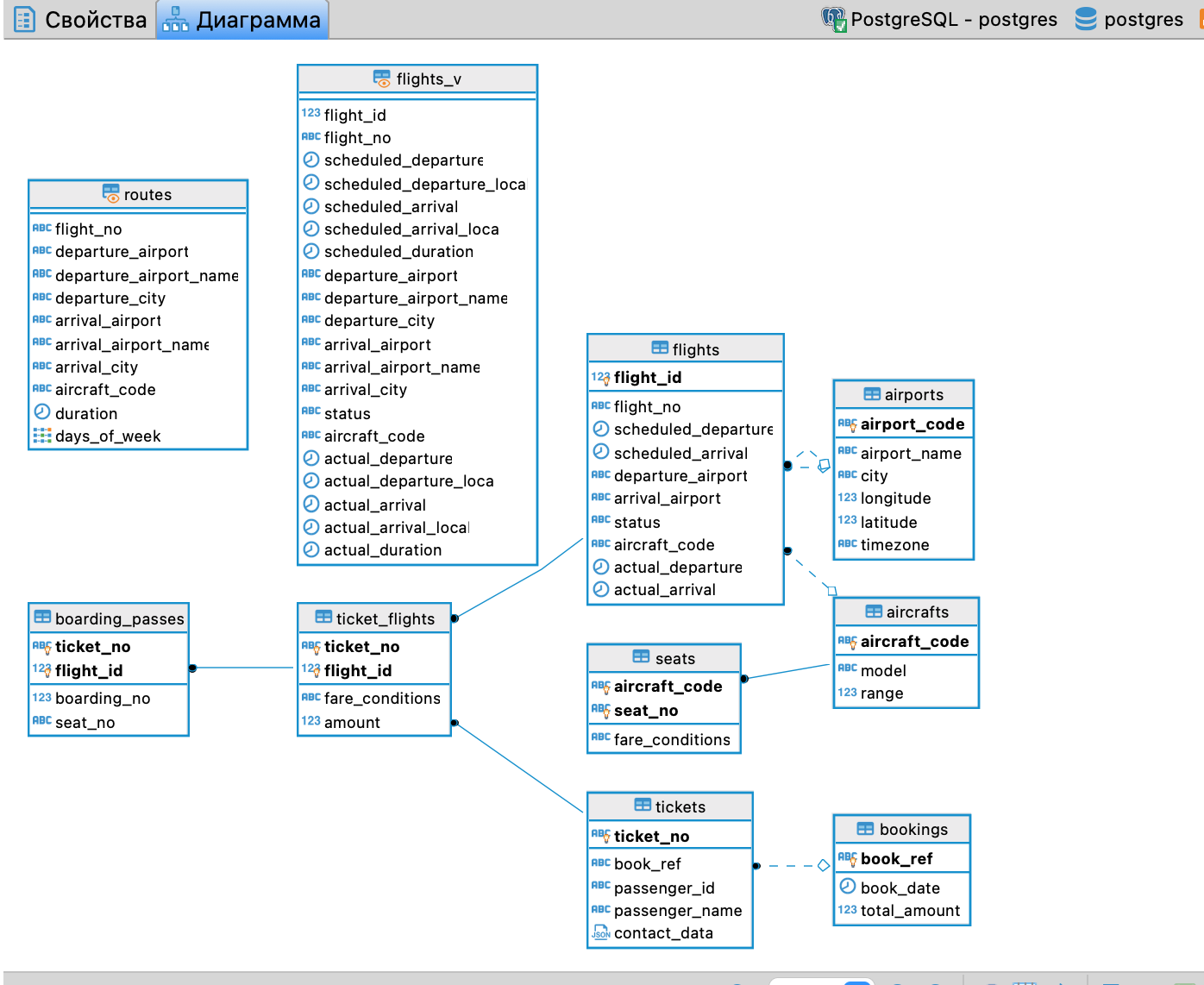
1. **Тип подключения**

В работе использовался локальный тип подключения.



1. **ER-диаграмма**

Скриншот ER-диаграммы из DBeaver`a согласно моего подключения:



1. **Краткое описание БД (таблицы и представления)**
   1. **Таблицы**
   * Aircrafts – код воздушного судна (aircraft\_code), модель (model), максимальная дальность полёта, км (range).
   * Airports – код аэропорта (airport\_code), название аэропорта (airport\_name), город (city), координаты (longitude/ latitude), временная зона аэропорта (timezone)
   * boarding\_passes – номер билета (ticket\_no), id рейса (flight\_id), номер посадочного (boarding\_no), номер места (seat\_no)
   * bookings – номер бронирования (book\_ref), дата бронирования (book\_date), полная сумма бронирования (total\_amount)
   * flights – id рейса (flight\_id), номер рейса (flight\_no), время вылета и прилета по расписанию (scheduled\_departure/ scheduled\_arrival), аэропорты отправления и прибытия (departure\_airport/ arrival\_airport), статус рейса (status), код ВС (aircraft\_code), фактическое время вылета и прилета (actual\_departure/ actual\_arrival)
   * seats – код ВС (aircraft\_code), номер места (seat\_no), класс обслуживания (fare\_conditions)
   * ticket\_flights – номер билета (ticket\_no), id рейса (flight\_id), класс обслуживания (fare\_conditions), стоимость перелета (amount)
   * tickets – номер билета (ticket\_no), номер бронирования (book\_ref), id пассажира (passenger\_id), ФИ пассажира (passenger\_name), контактные данные пассажира (contact\_data)
   1. **Представления**
   * flights\_v – данные об аэропорте вылета (departure\_airport, departure\_airport\_name, departure\_city), данные 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)
   * routes (материализованное представление) - номер рейса, код аэропорта отправления, название аэропорта отправления, город отправления, код аэропорта прибытия, название аэропорта прибытия, город прибытия, код самолёта, продолжительность полета, дни недели, когда выполняется рейс
2. **Развернутый анализ БД - описание таблиц, логики, связей и бизнес области**
   1. **Таблица aircrafts**

* Каждая модель воздушного судна идентифицируется своим трехзначным кодом (aircraft\_code). Указывается также название модели (model) и максимальная дальность полета в километрах (range).
* 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
  1. **Таблица 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).
* 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)
  1. **Таблица 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).
* 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)
  1. **Таблица 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.
* Индексы: PRIMARY KEY, btree (book\_ref)
* Ссылки извне: TABLE "tickets" FOREIGN KEY (book\_ref) REFERENCES bookings(book\_ref)
  1. **Таблица 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.

* 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 Tim e', '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)
  1. **Таблица seats**
* 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
  1. **Таблица ticket\_flights**
* 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)
  1. Таблица 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.
* 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)

1. **Бизнес-задачи, которые могут быть решены с помощью БД**

Данные, которые хранятся в базе данных bookings, могут помочь дать ответ на следующие бизнес-задачи:

1. Оценка прибыльности перелетов по рейсу/направлению на основании средней загрузки кресел и стоимости билетов
2. Оценка сезонности загрузки кресел для планирования ценовой политики
3. Оценка оптимальности подобранных самолетов на определенные направления на основе дальности полета, загруженности рейсов и вместительности самолета
4. Оценка отклонения планового времени прилета/вылета от фактического для дальнейшей оптимизации
5. Выявление паттернов формирования заказа (чаще заказывают билеты только для себя/для нескольких людей, на какую продолжительность чаще берут билеты) для дальнейшей оптимизации интерфейса покупки билетов
6. **Список SQL запросов из приложения №2 с описанием логики их выполнения.**

Список SQL запросов находится в файле diplom.sql