

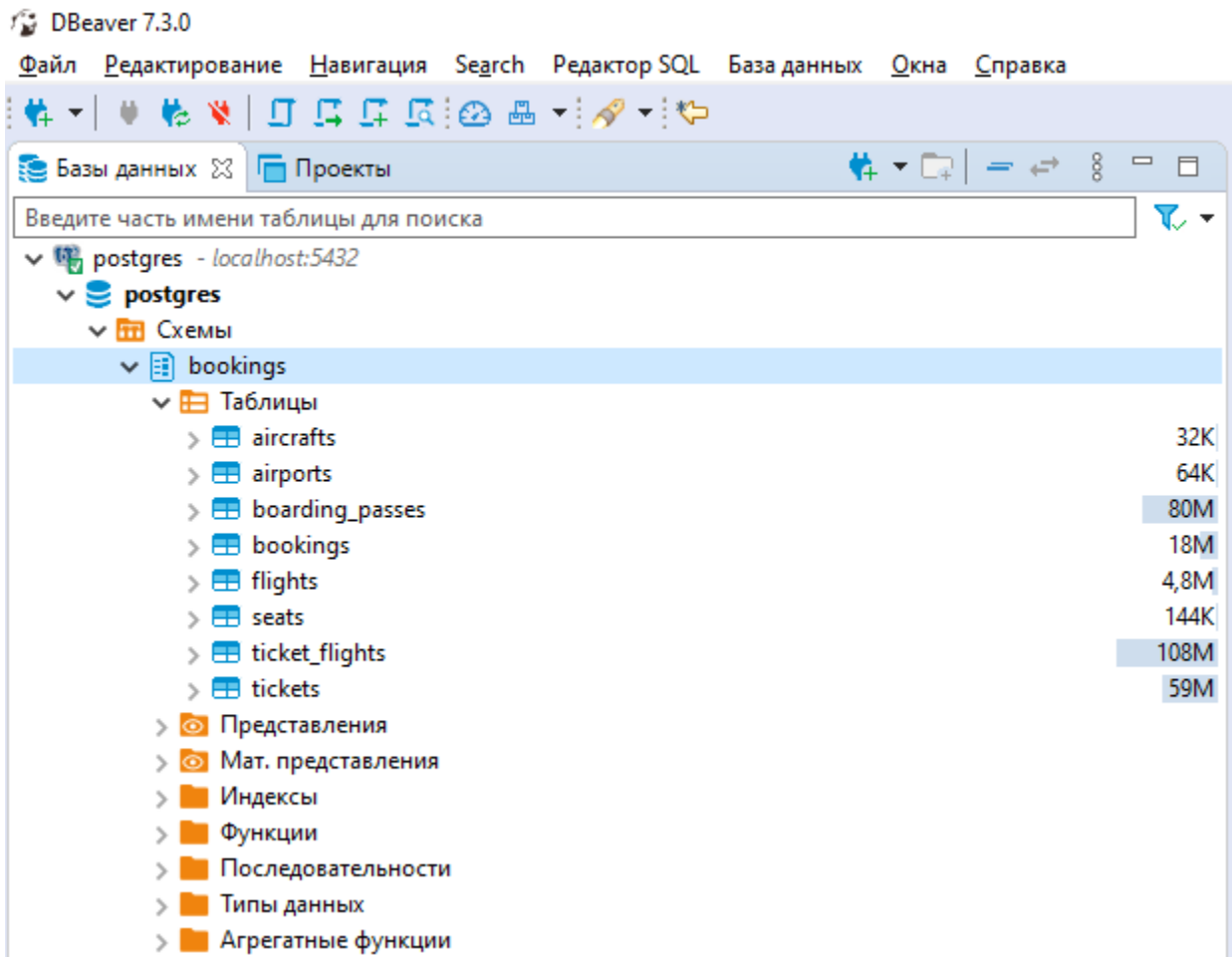
# Проектная работа по модулю “SQL и получение данных”



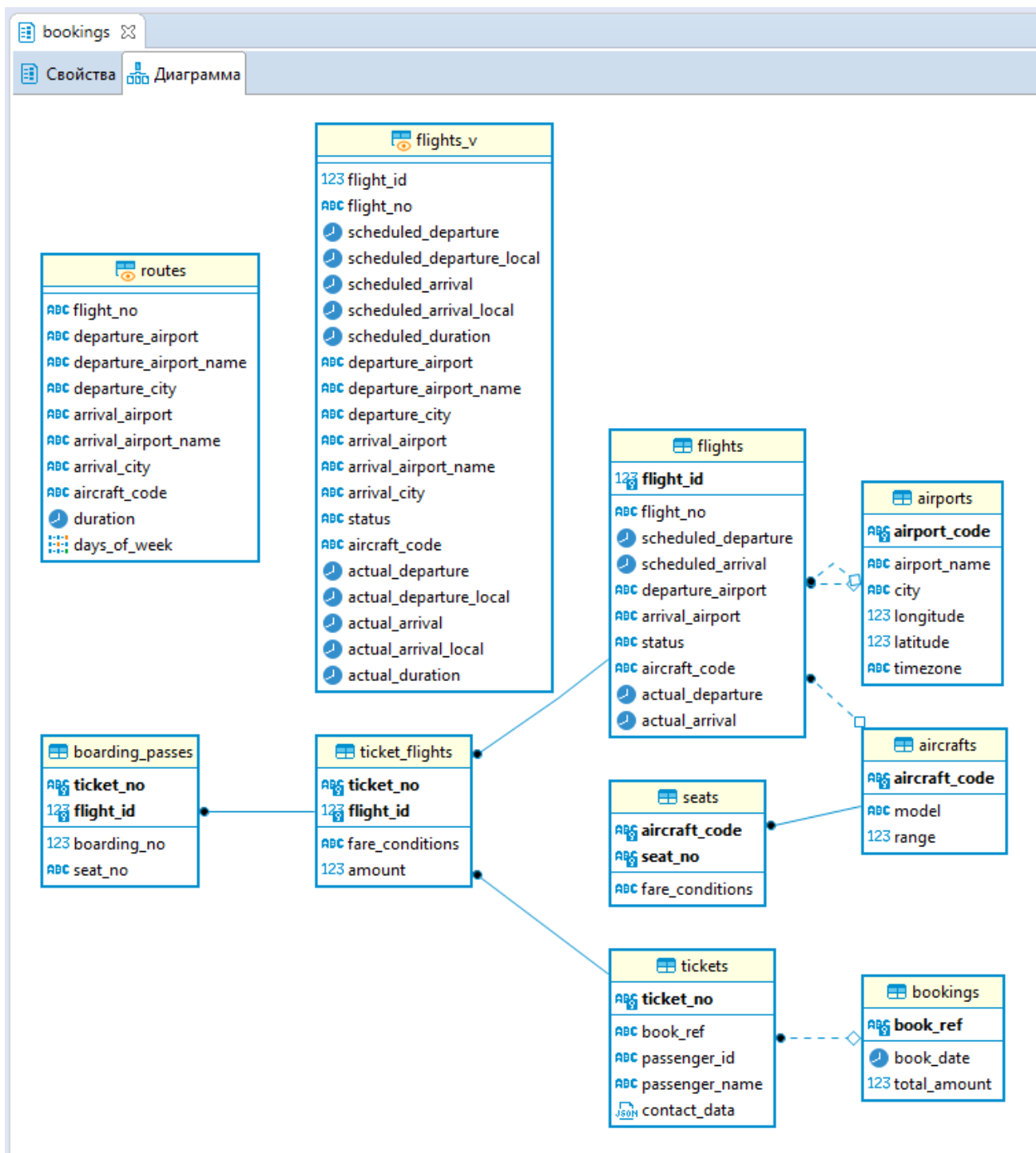
Чернецов Александр

январь, 2021

1. В работе использовался локальный тип подключения.



2. Скриншот ER-диаграммы из DBeaver`а согласно моего подключения:



### 3. Краткое описание БД (таблицы и представления):



#### а. Таблицы:

- i. Aircrafts – код воздушного судна (ВС), модель ВС, максимальная дальность полёта (км)
- ii. Airports – код аэропорта, название аэропорта, город, координаты (долгота/широта), временная зона аэропорта
- iii. boarding\_passes – номер билета, id рейса, номер посадочного, номер места
- iv. bookings – номер бронирования, дата бронирования, полная сумма бронирования
- v. flights – id рейса, номер рейса, время вылета и прилета по расписанию, аэропорты отправления и прибытия, статус рейса, код ВС, фактическое время вылета и прилета
- vi. seats – код ВС, номер места, класс обслуживания
- vii. ticket\_flights – номер билета, id рейса, класс обслуживания, стоимость перелета
- viii. tickets – номер билета, номер бронирования, id пассажира, ФИ пассажира, контактные данные пассажира

#### б. Представления

- i. flights\_v - идентификатор рейса, номер рейса, время вылета по расписанию + местное, время прилета по расписанию + местное, планируемая продолжительность полета, код аэропорта отправления, название аэропорта отправления, город отправления, код аэропорта прибытия, название аэропорта прибытия, город прибытия, статус рейса, код самолета, фактическое время вылета + местное, фактическое время прилета + местное, фактическая продолжительность полета
- ii. routes – материализованное – 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

## 4. Развернутый анализ БД - описание таблиц, логики, связей и бизнес-области:

### 4.1 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

### 4.2 Airports:

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

### 4.3 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)

### 4.4 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)

### 4.5 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.
- 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)

#### 4.6 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

#### 4.7 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)

#### 4.8 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).
- 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)





## Мои мысли о предоставленной базе и её практическом использовании в принятии управленческих решений:

Как человек с высшим экономическим образованием по специальности «менеджмент на воздушном транспорте» (МГТУ ГА, 1995-2000 годы обучения, красный диплом), правда ни дня ни работавшего по данной специальности могу резюмировать следующее:

1. На основании средней загрузки кресел как по перелету, так и рейсу/направлению можно судить об убыточности как этих самых перелетов, так и рейсов/направлений в общем. Как утверждает В. Савельев в данной статье <https://www.kommersant.ru/doc/3246832> ниже 75% и рейс нерентабелен. Мы, в упомянутые выше годы обучения, закладывали еще меньшую маржинальность и рейсы до 80-85% загрузки кресел уже шли убыточными. Поэтому, считая кресла по перелетам и рейсам/направлениям можно уже примерно понимать экономический результат деятельности перевозчика, не имея на руках больше никаких дополнительных данных.

2. Помимо этого, можно сделать предварительные выводы об оптимальности подобранных самолетов на отдельные перелёты и рейсы/направления, исходя не только из максимальной дальности полета выбранных самолетов, но и принимая в расчет загруженность рейсов. Как следствие этого (из данной базы, конечно, это уже не вытянуть – просто развивая мысль дальнейших шагов повышения эффективности работы) подбор альтернативных ВС (воздушных судов) исходя из соотношения вместимость/дальность/загрузка, так и внесение корректировок в само производство полётов:

- объединение рейсов внутри компании (производить полеты реже, но с большей загрузкой кресел)
- кооперации с членами альянса, в который входит данный перевозчик (при условии пересекающихся направлений полетов). это позволит продавать совместные перевозки, не теряя в периодичности, выбирая при этом наиболее оптимальные самолеты с точки зрения общей вместимости и дальности полета с планируемым общим пассажиропотоком членов альянса.



5. Список SQL запросов с описанием логики их выполнения:

Комментарии с описанием логики даны в самих решениях, прилагаемых к данному отчёту.

