**Преподаватель:** Николай Хащанов

**Студент:** Леонард Черепанов

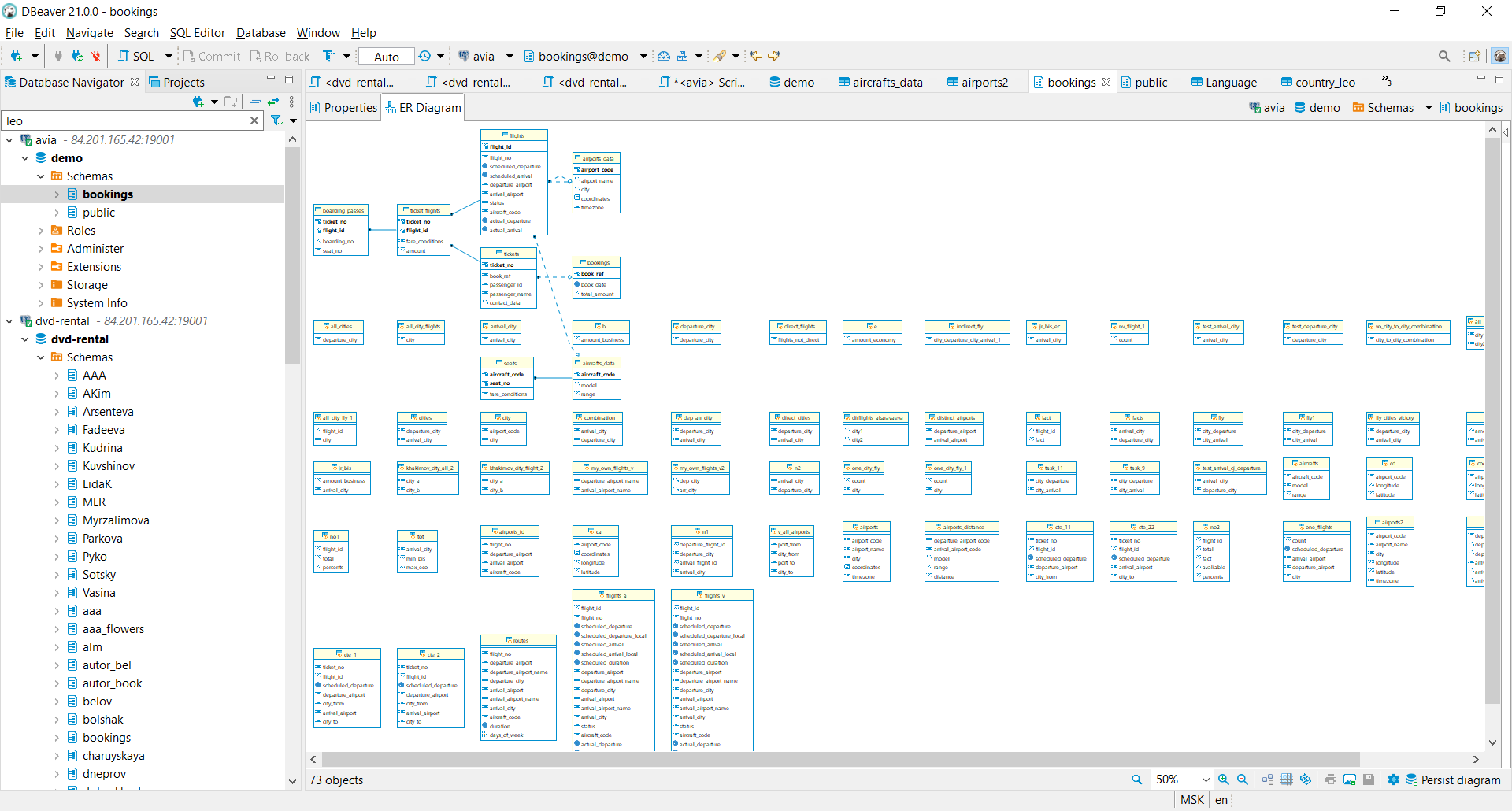
**Дипломный проект по модулю:**

**«SQL и получение данных»**



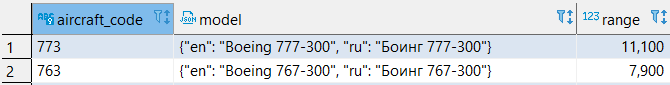
В работе использовался облачный тип подключения.

Скриншот ER-диаграммы из DBeaver`a согласно подключения.

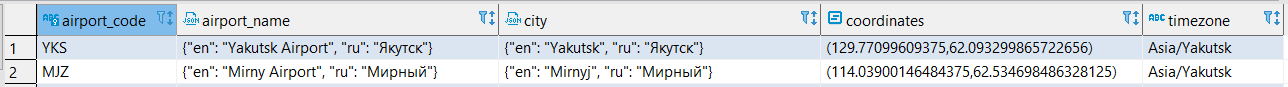


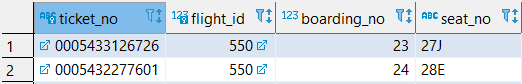
Краткое описание БД:

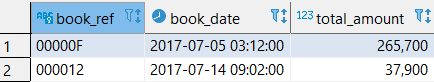
## 8 таблиц:

aircrafts\_data – код воздушного судна (ВС), модель ВС, максимальная дальность полёта (км)

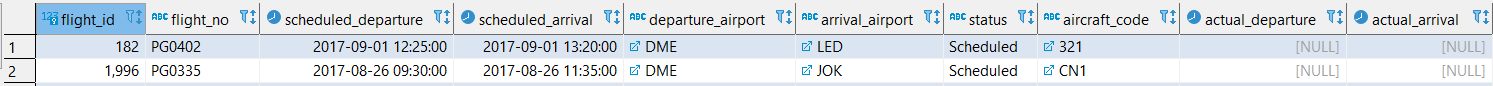
airports\_data –код аэропорта, название аэропорта, город, координаты (долгота/широта), временная зона аэропорта



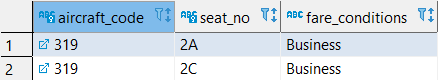
boarding\_passes – номер билета, id рейса, номер посадочного, номер места

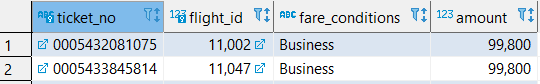
bookings – номер бронирования, дата бронирования, полная сумма бронирования

flights – id рейса, номер рейса, время вылета и прилета по расписанию, аэропорты отправления и прибытия, статус рейса, код ВС, фактическое время вылета и прилета

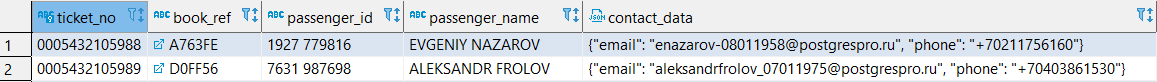


seats – код ВС, номер места, класс обслуживания



ticket\_flights – номер билета, id рейса, класс обслуживания, стоимость перелета

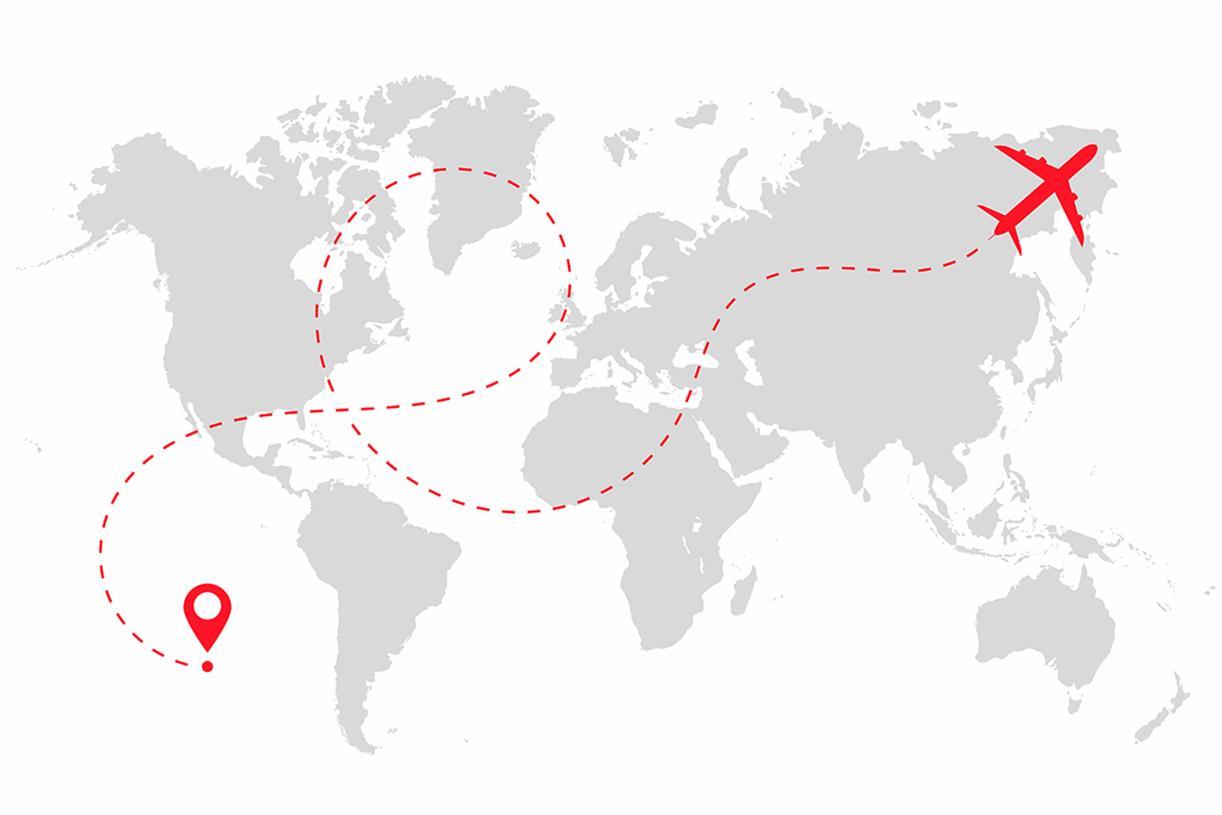
tickets – номер билета, номер бронирования, id пассажира, ФИО пассажира, контактные данные пассажира



## 2 представления:

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

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



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

## Таблица aircrafts\_data

* 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

## Таблица airports\_data

* 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)

## 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)

## 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) bookings(book\_ref)

## 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.
* 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 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

## 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))

## 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).
* 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)

## Бизнес задачи:

* Рентабельность рейсов
* Загруженность рейсов – пересмотр парка ВС.
* Анализ потребительской активности. Предпочтение эконом/бизнес.

Список SQL запросов из приложения с описанием логики их выполнения. Комментарии с описанием логики находятся в самих решениях.