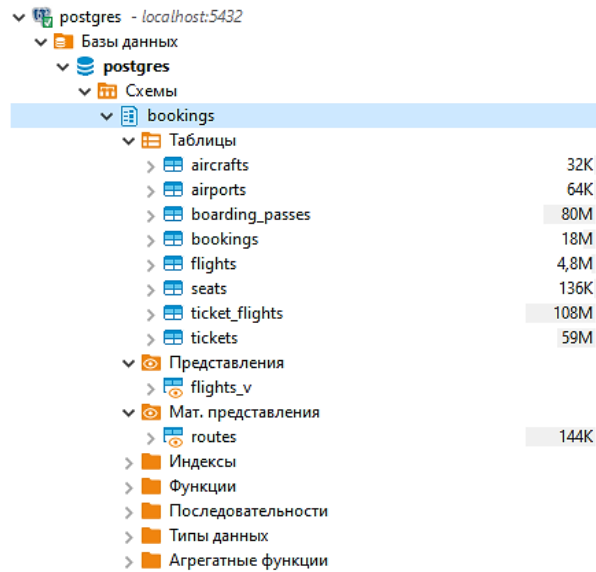


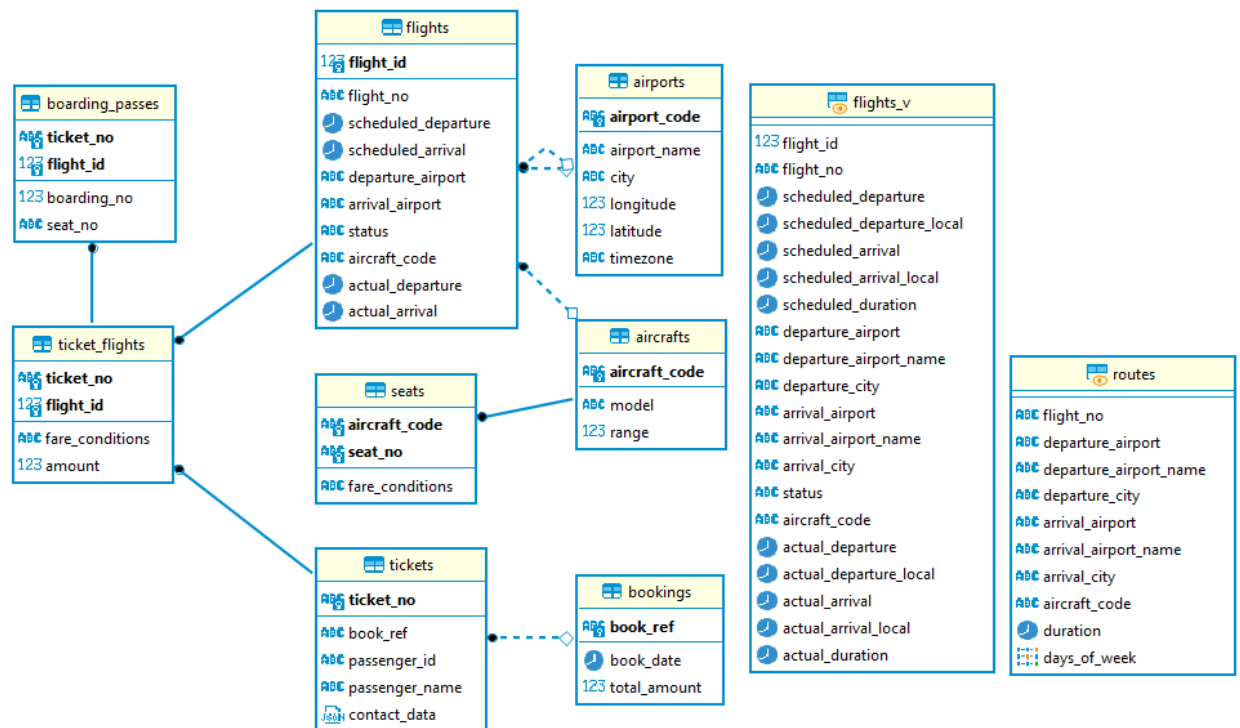
# Отчёт по итоговой работе

БЛОК-МОДУЛЬ «SQL И ПОЛУЧЕНИЕ ДАННЫХ»  
ВЫПОЛНИЛ: ТРОФИМОВ П.В.

В работе использовался **ЛОКАЛЬНЫЙ** тип подключения



## ER-Диаграмма схемы **bookings**



## Краткое описание базы данных **bookings**

База данных **bookings** имеет следующие отношения:

- **aircrafts** – информация по самолётам  
(код, модель, максимальная дальность полёта в километрах)
- **airports** – информация по аэропортам  
(код, название, город, долгота, широта, временная зона)
- **boarding\_passes** – информация по посадочным талонам  
(номер билета, идентификатор рейса, номер посадочного талона, номер места)
- **bookings** – информация по бронированию  
(номер, дата, общая стоимость)
- **flights** – информация по рейсам  
(Идентификатор рейса, номер рейса, время вылета и прилёта по расписанию, аэропорты вылета и прибытия, статус рейса, код самолёта, фактическое время вылета и прилёта)
- **seats** – информация по местам  
(код самолёта, номер места, класс обслуживания)
- **ticket\_flights** – информация по билетам и рейсам  
(номер билета, идентификатор рейса, класс обслуживания, стоимость перелёта)
- **tickets** - информация по билетам  
(номер билета, номер бронирования, идентификатор пассажира, фамилия и имя, контактные данные)

Также в базе данных **bookings** имеются два представления:

- **flights\_v** – информация по рейсам и аэропортам с дополнениями  
(дополнения: время вылета и прилёта местное, планируемая и фактическая продолжительность полёта, код самолёта)
- **routes** – материализованное представление с информацией по маршрутам (по рейсам и аэропортам с дополнениями)  
(дополнения: продолжительность полёта, дни недели, когда выполняется рейс)

## Aircrafts

Каждая модель самолёта идентифицируется своим трёхзначным кодом (**aircraft\_code**). Также указывается название модели (**model**) и максимальная дальность полёта 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
```

## Airports

Аэропорт идентифицируется трёхбуквенным кодом (**airport\_code**) и имеет своё 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)
```

# Boarding\_passes

При регистрации на рейс, которая возможна за сутки до плановой даты отправления, пассажиру выдаётся посадочный талон. Он идентифицируется также, как и перелёт — 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)

# 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 перелётов всех пассажиров.

## Индексы:

PRIMARY KEY, btree (book\_ref)

## Ссылки извне:

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

# 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**). Реальное 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)

# Seats

Места определяют схему салона каждой модели. Каждое место определяется своим номером (**seat\_no**) и имеет закреплённый 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

# Ticket\_flights

Перелёт 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)

# 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\_data**).

## 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)

Анализ и использование данных, имеющихся в базе данных **bookings**, дают возможность оценить целесообразность рейсов, подталкивают к подбору оптимальных типов самолётов для уже имеющихся рейсов, пересмотру классов обслуживания и правил бронирования.

К примеру, классом «Комфорт» можно слетать только в пять городов. Возможно, уменьшение количества мест класса «Эконом» в сторону повышения количества мест класса «Комфорт» повысит рентабельность некоторых перелётов. Процент свободных мест в салоне самолёта может наоборот говорить о необходимости уменьшения количества мест класса «Бизнес» и увеличении количества мест другого класса. Низкий процент занятых мест в салоне может говорить также о неудобном времени вылета или прилёта.

Также следует обратить внимание на задержки рейсов. Около половины рейсов были задержаны, а, если говорить точнее, то почти 16 тысяч рейсов из всех 33 тысяч. Может быть, это связано с техническим состоянием самолётов, халатным отношением работников, неправильным подбором самолётов для конкретных погодных условий или расстояний и т. д.

Описание логики SQL-запросов, где это виделось необходимым, можно найти в виде комментариев в файле с запросами.