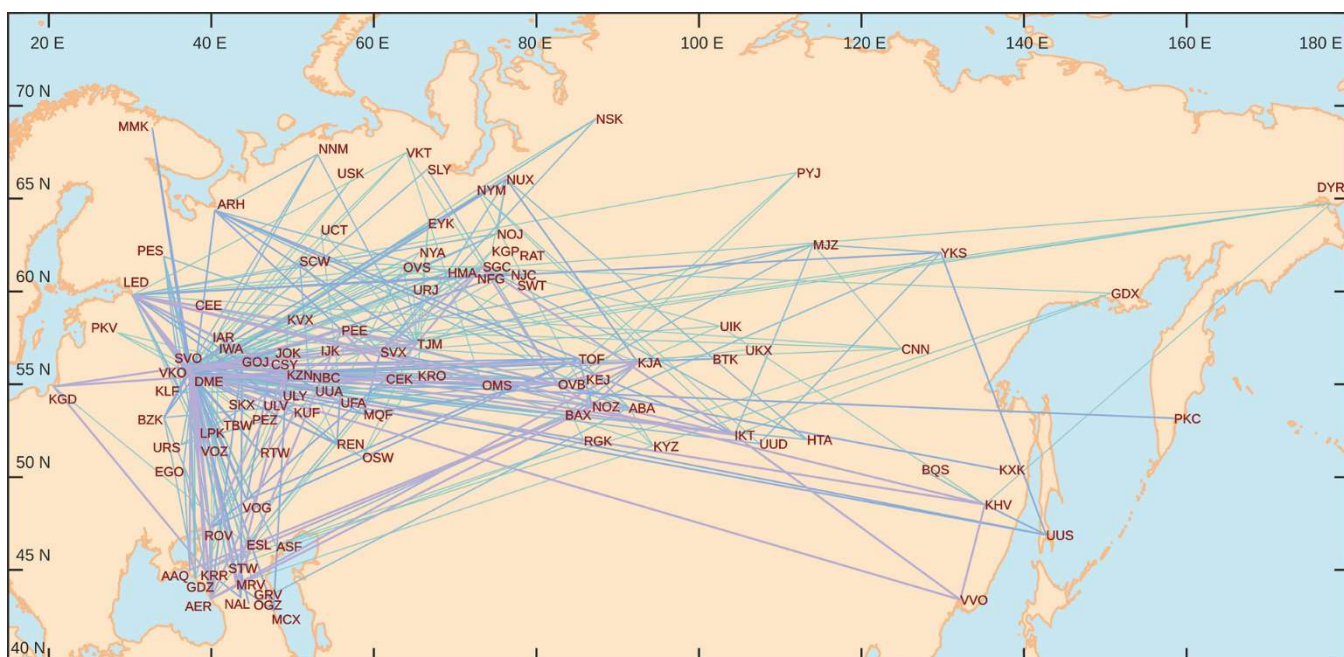


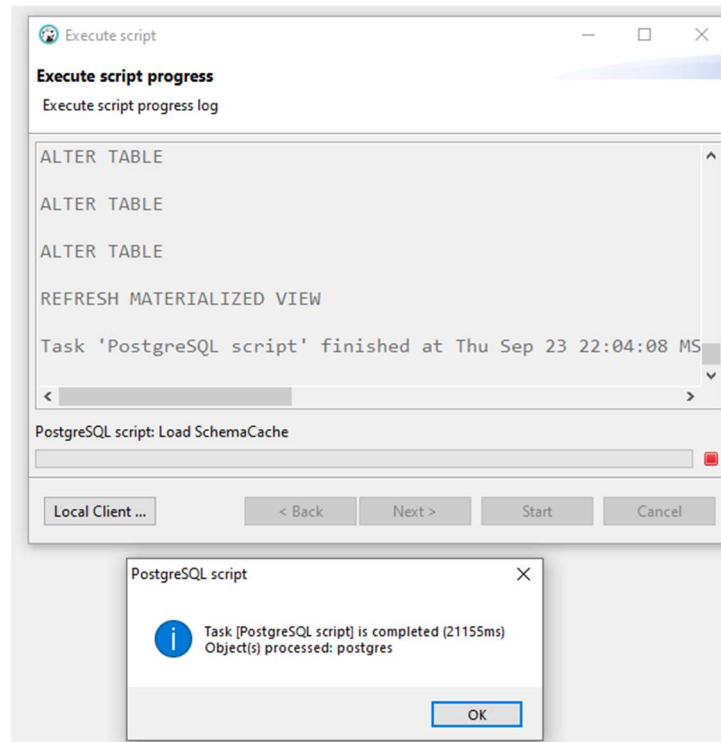
Итоговая работа по модулю «SQL и получение данных»



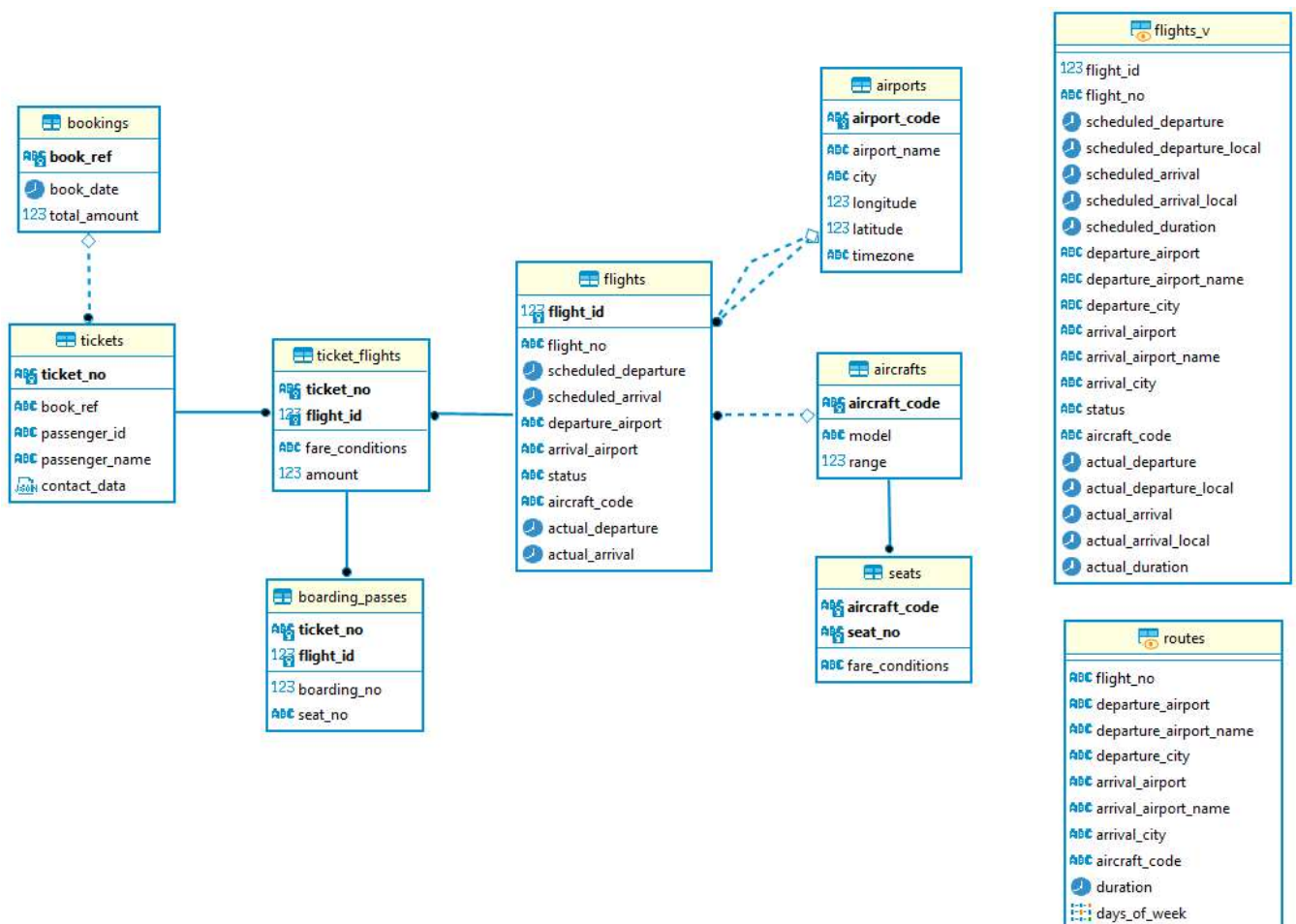
Выполнила:
Хошина Наталья
группа DS-36

сентябрь 2021

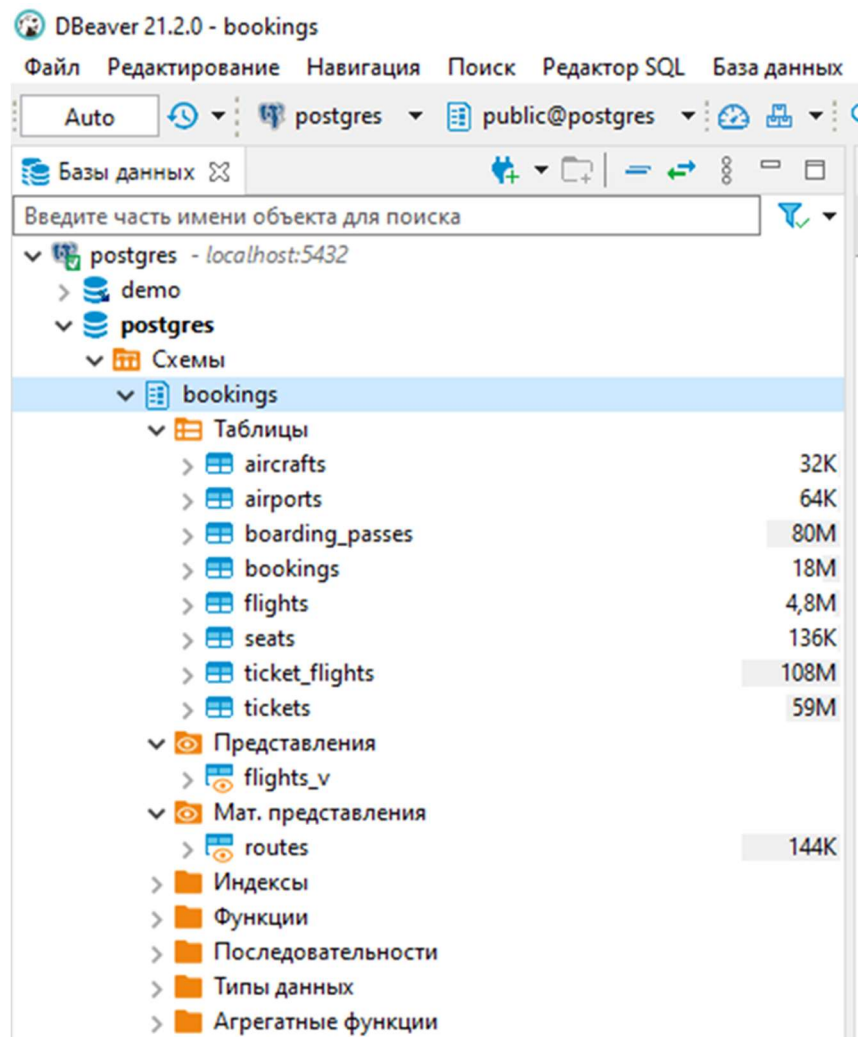
1. В работе использовался локальный тип подключения.



2. Скриншот ER-диаграммы БД bookings.



3. Краткое описание БД bookings.



База данных Авиаперевозок состоит из 8 таблиц, 1 представления и 1 материализованного представления.

Список таблиц и представлений БД:

Наименование таблицы, представления	Тип	Описание
aircrafts	таблица	Самолеты
airports	таблица	Аэропорты
boarding_passes	таблица	Посадочные талоны
bookings	таблица	Бронирования
flights	таблица	Рейсы
seats	таблица	Места
ticket_flight	таблица	Перелеты
tickets	таблица	Билеты
flights_v	представление	Рейсы
routes	мат. представление	Маршруты

4. Развернутый анализ БД bookings.

Основной сущностью является бронирование (bookings).

В одно бронирование можно включить нескольких пассажиров, каждому из которых выписывается отдельный билет (tickets). Билет имеет уникальный номер и содержит информацию о пассажире.

Билет включает один или несколько перелетов (ticket_flights). Несколько перелетов могут включаться в билет в случаях, когда нет прямого рейса, соединяющего пункты отправления и назначения (полет с пересадками), либо, когда билет взят «туда и обратно».

В схеме данных предполагается, что все билеты в одном бронировании имеют одинаковый набор перелетов.

Каждый рейс (flights) следует из одного аэропорта (airports) в другой. Рейсы с одним номером имеют одинаковые пункты вылета и назначения, но будут отличаться датой отправления.

При регистрации на рейс пассажиру выдается посадочный талон (boarding_passes), в котором указано место в самолете. Пассажир может зарегистрироваться только на тот рейс, который есть у него в билете. Комбинация рейса и места в самолете должна быть уникальной.

Количество мест (seats) в самолете и их распределение по классам обслуживания зависит от модели самолета (aircrafts), выполняющего рейс. Предполагается, что каждая модель самолета имеет только одну компоновку салона.

✂ Aircrafts

Таблица aircrafts является справочником типов самолетов и дальности их полетов.

Каждая модель воздушного судна идентифицируется своим трехзначным кодом (aircraft_code).

Столбец	Тип	Модификаторы	Описание
aircraft_code	char(3)	NOT NULL	Код самолета, IATA
model	text	NOT NULL	Модель самолета
range	integer	NOT NULL	Максимальная дальность полета, км

Индексы:

PRIMARY KEY, btree (aircraft_code)

Ограничения-проверки:

CHECK (range > 0)

Ссылки извне:

TABLE "flights" FOREIGN KEY (aircraft_code)

REFERENCES aircrafts_data(aircraft_code)

TABLE "seats" FOREIGN KEY (aircraft_code)

✂ Airports

Таблица airports является справочником аэропортов, с указанием международного кода аэропорта, названия аэропорта, города, координат и временной зоны.

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.

Столбец	Тип	Модификаторы	Описание
airport_code	char(3)	NOT NULL	Код аэропорта
airport_name	text	NOT NULL	Название аэропорта
city	ctext	NOT NULL	Город
longitude	float	NOT NULL	Координаты аэропорта: долгота
latitude	float	NOT NULL	Координаты аэропорта: широта
timezone	text	NOT NULL	Временная зона аэропорта

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

Таблица `boarding_passes` – данные по выданным посадочным талонам.

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`).

Столбец	Тип	Модификаторы	Описание
<code>ticket_no</code>	<code>char(13)</code>	<code>NOT NULL</code>	Номер билета
<code>flight_id</code>	<code>integer</code>	<code>NOT NULL</code>	Идентификатор рейса
<code>boarding_no</code>	<code>integer</code>	<code>NOT NULL</code>	Номер посадочного талона
<code>seat_no</code>	<code>varchar(4)</code>	<code>NOT NULL</code>	Номер места

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

Таблица `bookings` – данные по бронированию билетов. Номер бронирования может включать несколько пассажиров, каждому из которых выписывается отдельный билет.

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.

Столбец	Тип	Модификаторы	Описание
<code>book_ref</code>	<code>char(6)</code>	<code>NOT NULL</code>	Номер бронирования
<code>book_date</code>	<code>timestampz</code>	<code>NOT NULL</code>	Дата бронирования
<code>total_amount</code>	<code>numeric(10,2)</code>	<code>NOT NULL</code>	Полная сумма бронирования

Индексы:

```
PRIMARY KEY, btree (book_ref)
```

Ссылки извне:

```
TABLE "tickets" FOREIGN KEY (book_ref) REFERENCES bookings(book_ref)
```

✕ Flights

Таблица `flights` содержит данные, которые используются для большого количества запросов из БД. Это расписание рейсов со статусом их фактического выполнения: номер рейса, время вылета и прибытия по расписанию и по факту, текущий статус рейса, коды аэропортов вылета и прибытия, вид самолета.

Имеет наибольшее число связей с другими таблицами БД.

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.

Столбец	Тип	Модификаторы	Описание
flight_id	serial	NOT NULL	Идентификатор рейса
flight_no	char(6)	NOT NULL	Номер рейса
scheduled_departure	timestampz	NOT NULL	Время вылета по расписанию
scheduled_arrival	timestampz	NOT NULL	Время прилёта по расписанию
departure_airport	char(3)	NOT NULL	Аэропорт отправления
arrival_airport	char(3)	NOT NULL	Аэропорт прибытия
status	varchar(20)	NOT NULL	Статус рейса
aircraft_code	char(3)	NOT NULL	Код самолета, IATA
actual_departure	timestampz		Фактическое время вылета
actual_arrival	timestampz		Фактическое время прилёта

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

Таблица `seats` является расширением справочника самолетов с указанием номеров мест и класса каждого 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.

Столбец	Тип	Модификаторы	Описание
<code>aircraft_code</code>	<code>char(3)</code>	NOT NULL	Код самолета, IATA
<code>seat_no</code>	<code>varchar(4)</code>	NOT NULL	Номер места
<code>fare_conditions</code>	<code>varchar(10)</code>	NOT NULL	Класс обслуживания

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_flight

Таблица `ticket_flights` связывает рейсы, билеты и посадочные талоны.

Перелет соединяет билет с рейсом и идентифицируется их номерами. Для каждого перелета указываются его стоимость (`amount`) и класс обслуживания (`fare_conditions`).

Столбец	Тип	Модификаторы	Описание
<code>ticket_no</code>	<code>char(13)</code>	NOT NULL	Номер билета
<code>flight_id</code>	<code>integer</code>	NOT NULL	Идентификатор рейса
<code>fare_conditions</code>	<code>varchar(10)</code>	NOT NULL	Класс обслуживания
<code>amount</code>	<code>numeric(10,2)</code>	NOT NULL	Стоимость перелета

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

Таблица `tickets` является справочником по купленным билетам.

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.

Столбец	Тип	Модификаторы	Описание
ticket_no	char(13)	NOT NULL	Номер билета
book_ref	char(6)	NOT NULL	Номер бронирования
passenger_id	varchar(20)	NOT NULL	Идентификатор пассажира
passenger_name	text	NOT NULL	Имя пассажира
contact_data	jsonb		Контактные данные пассажира

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)

✕ 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).

Столбец	Тип	Описание
flight_id	integer	Идентификатор рейса
flight_no	char(6)	Номер рейса
scheduled_departure	timestamptz	Время вылета по расписанию
scheduled_departure_local	timestamp	Время вылета по расписанию, местное время в пункте отправления
scheduled_arrival	timestamptz	Время прилёта по расписанию
scheduled_arrival_local	timestamp	Время прилёта по расписанию, местное время в пункте прибытия
scheduled_duration	interval	Планируемая продолжительность полета
departure_airport	char(3)	Код аэропорта отправления
departure_airport_name	text	Название аэропорта отправления
departure_city	text	Город отправления
arrival_airport	char(3)	Код аэропорта прибытия
arrival_airport_name	text	Название аэропорта прибытия
arrival_city	text	Город прибытия
status	varchar(20)	Статус рейса
aircraft_code	char(3)	Код самолета, IATA
actual_departure	timestamptz	Фактическое время вылета
actual_departure_local	timestamp	Фактическое время вылета, местное время в пункте отправления
actual_arrival	timestamptz	Фактическое время прилёта
actual_arrival_local	timestamp	Фактическое время прилёта, местное время в пункте прибытия
actual_duration	interval	Фактическая продолжительность полета

✂ 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.

Столбец	Тип	Описание
flight_no	char(6)	Номер рейса
departure_airport	char(3)	Код аэропорта отправления
departure_airport_name	text	Название аэропорта отправления
departure_city	text	Город отправления
arrival_airport	char(3)	Код аэропорта прибытия
arrival_airport_name	text	Название аэропорта прибытия
arrival_city	text	Город прибытия
aircraft_code	char(3)	Код самолета, IATA
duration	interval	Продолжительность полета
days_of_week	integer[]	Дни недели, когда выполняются рейсы

Бизнес-задачи, которые можно решить с использованием БД

При помощи этой базы данных можно решить ряд бизнес-задач, связанных с организацией пассажирских авиаперевозок и аналитикой совершенных перелетов.

Основными задачами бизнеса, являются увеличение прибыли и сокращения операционных затрат. Этого можно достичь за счет, грамотного распределения имеющихся ресурсов посредством оптимизации маршрутов и загрузки рейсов.

В частности, можно создавать и отслеживать бронирования перелетов, контролировать наличие мест в самолетах, проверять соответствие транспортного судна требованиям к рейсу (дальность, наличие класса обслуживания и т.д.). Анализируя данные, можно проверять заполняемость самолетов, выявлять наиболее популярные маршруты, минимизировать количество незаполненных, неэффективных рейсов на менее востребованных маршрутах и быть готовыми выделить дополнительные мощности для востребованных.

Оценка загруженности рейсов позволит рассчитать прямые и косвенные расходы, связанные с осуществлением перелетов. Зная количество рейсов, можно рассчитывать затраты на топливо, на тех. обслуживание самолетов, на оплату труда экипажей.

На основе имеющихся записей так же можно отслеживать активность отдельных пассажиров и предлагать им различные скидки и бонусы в зависимости от их активности.