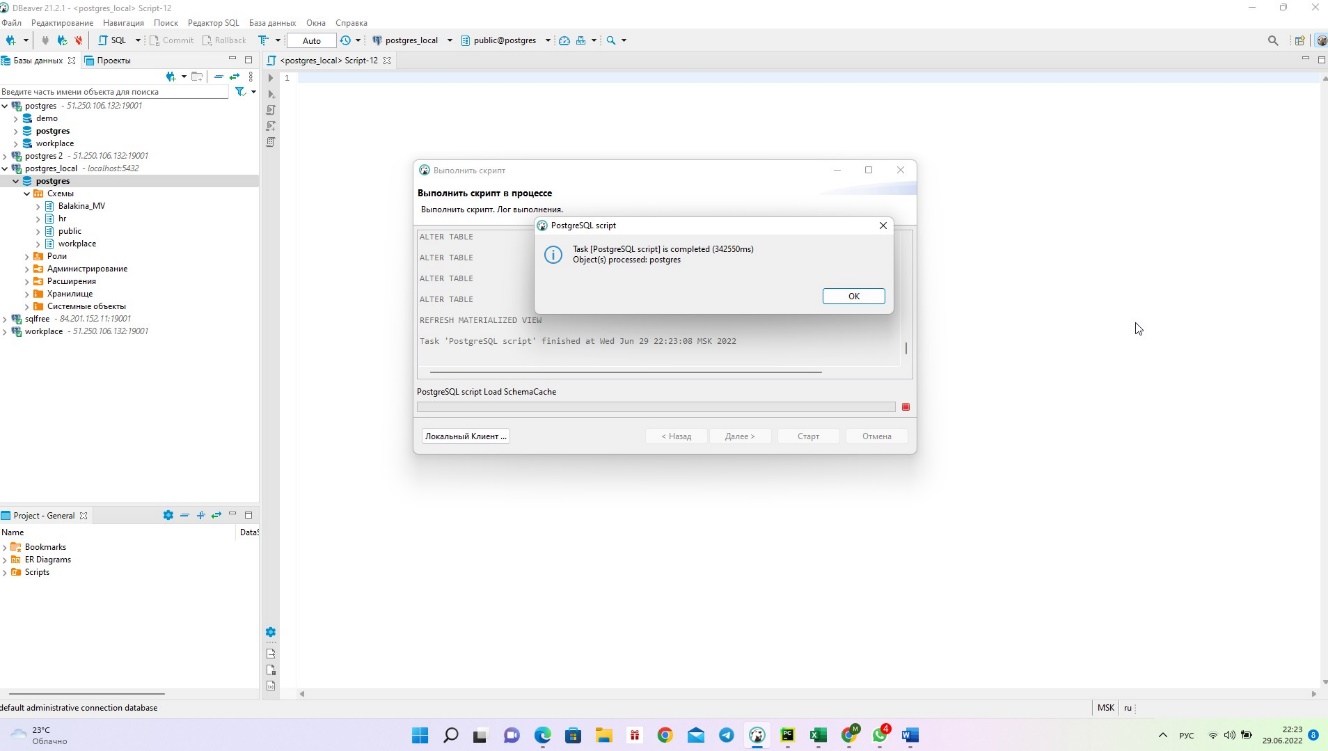
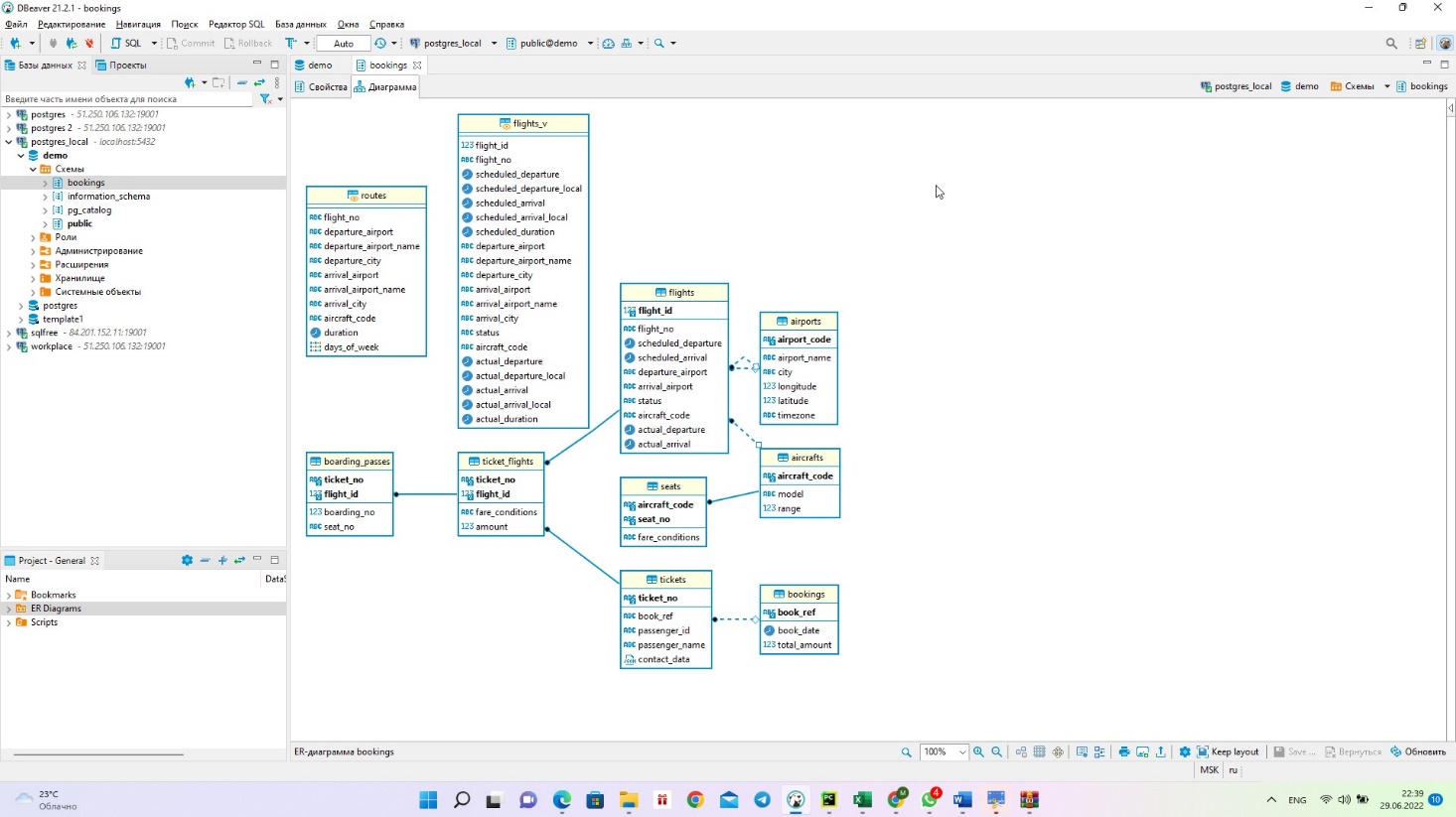
**Итоговая работа**

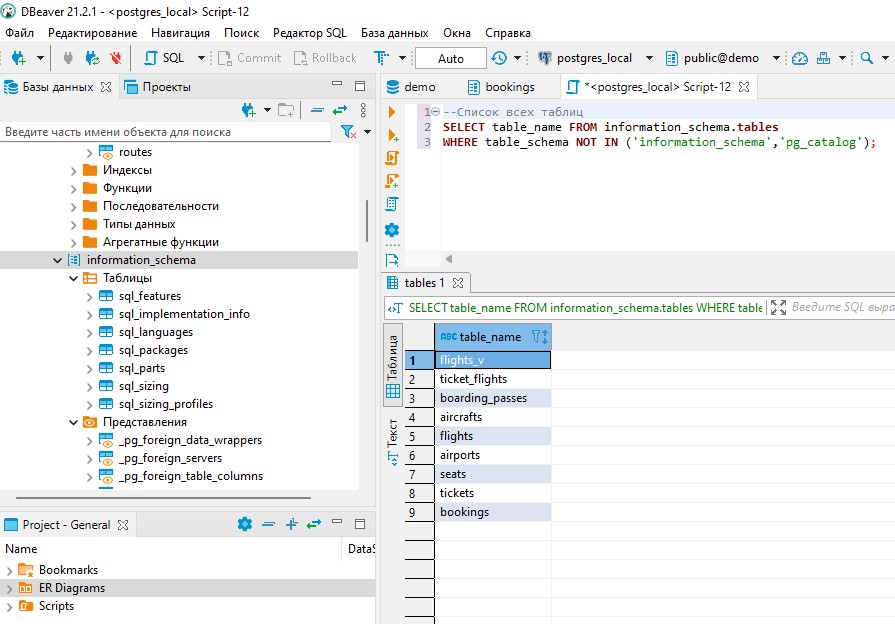
1 **Тип подключения**

В работе использовался локальный тип подключения. База развернута из demo\_big.sql

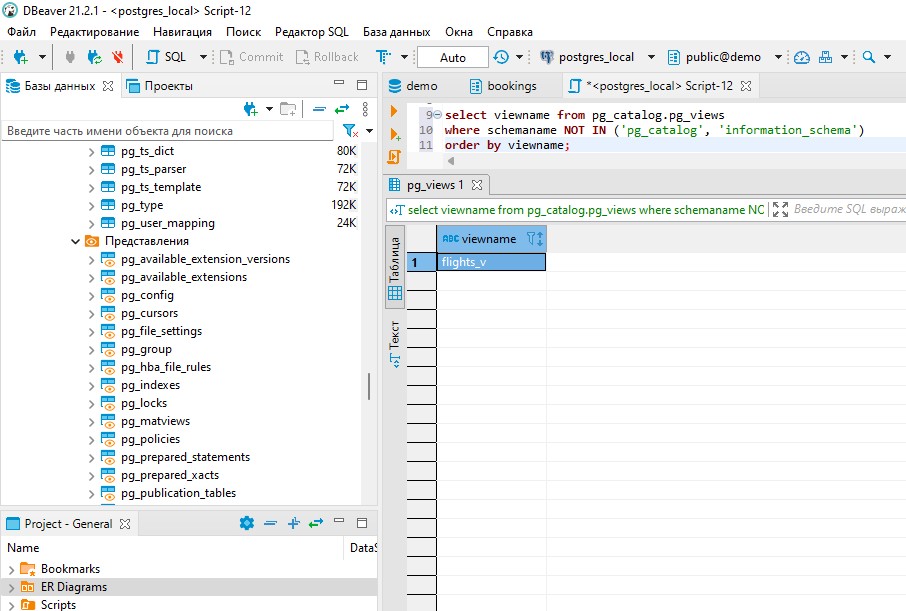
* если база была развернута из \*.sql или \*.backup файла, необходимо приложить скриншот успешного импорта или восстановления
* 

1. **Скриншот ER-диаграммы из DBeaver`a согласно подключению**.
2. **Краткое описание БД - из каких таблиц и представлений состоит.**

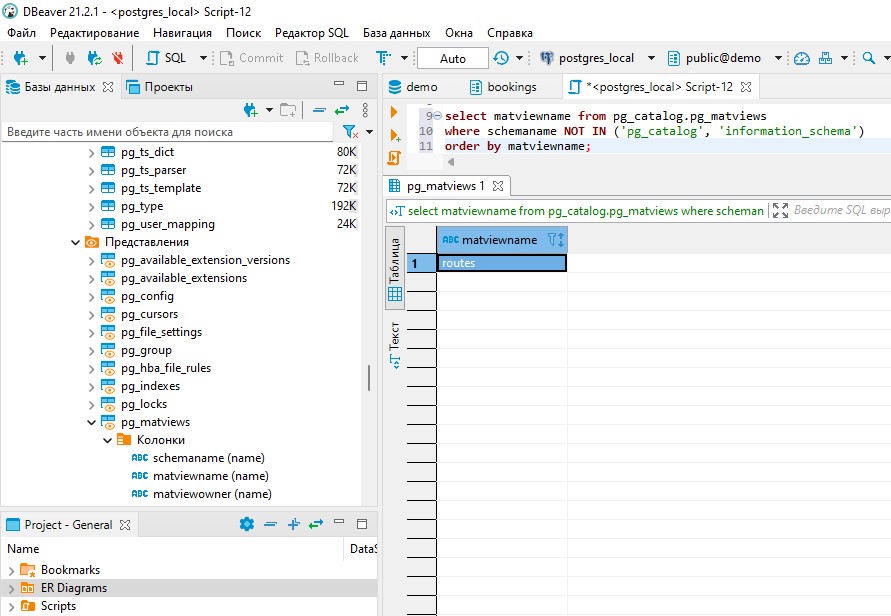
БД состоит из 8 таблиц: aircrafts, airports, boarding\_passes, booking, flights, seats,ticket\_flights,tickets



, , представления flights\_v



и материализованного представления routes



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

Основной сущностью является бронирование (bookings). В одно бронирование можно включить несколько 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 нет прямого рейса, соединяющего пункты отправления и назначения (полет с пересадками), либо когда билет взят «туда и обратно». В схеме данных нет жесткого ограничения, но 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, чтобы не допустить выдачу двух посадочных талонов на одно место.

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.

*Список отношений*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Тип | Размер (Big) | Описание |
| aircrafts | таблица | 16 kB | самолеты |
| airports | таблица | 48 kB | аэропорты |
| boarding\_passes | таблица | 427 MB | посадочные талоны |
| bookings | таблица | 105 MB | бронирования |
| flights | таблица | 19 MB | Рейсы |
| flights\_v | представление | 0 kB | Рейсы |
| routes | мат. предст. | 136 kB | Маршруты |
| seats | таблица | 88 kB | Места |
| ticket\_flights | таблица | 516 MB | Перелеты |
| tickets | таблица | 381 MB | Билеты |

**Таблица bookings.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).

|  |  |  |  |
| --- | --- | --- | --- |
| Столбец | Тип | Модификаторы | Описание |
| airport\_code | char(3) | NOT NULL | Код аэропорта |
| airport\_name | text | NOT NULL | Название аэропорта |
| city | text | 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)

**Таблица bookings.boarding\_passes**

При регистрации на рейс, которая возможна за сутки до плановой даты отправления, пассажиру выдается посадочный талон. Он идентифицируется так же, 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 |
| ticket\_no | char(13) | NOT NULL | Номер билета |
| flight\_id | integer | NOT NULL | Идентификатор рейса |
| boarding\_no | integer | NOT NULL | Номер посадочного талона |
| seat\_no | varchar(4) | NOT NULL | 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)

**Таблица bookings.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.

|  |  |  |  |
| --- | --- | --- | --- |
| Столбец | Тип | Модификаторы | Описание |
| book\_ref | char(6) | NOT NULL | Номер бронирования |
| book\_date | timestamptz | NOT NULL | Дата бронирования |
| total\_amount | numeric(10,2) | NOT NULL | Полная сумма бронирования |

*Индексы*: PRIMARY KEY, btree (book\_ref)

*Ссылки извне*: TABLE "tickets" FOREIGN KEY (book\_ref) REFERENCES bookings(book\_ref)

**Таблица bookings.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.

|  |  |  |  |
| --- | --- | --- | --- |
| Столбец | Тип | Модификаторы | Описание |
| flight\_id | serial | NOT NULL | Идентификатор рейса |
| flight\_no | char(6) | NOT NULL | Номер рейса |
| scheduled\_departure | timestamptz | NOT NULL | Время вылета по расписанию |
| scheduled\_arrival | timestamptz | 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 | timestamptz |  | Фактическое время вылета |
| actual\_arrival | timestamptz |  | 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)

**Таблица bookings.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.

|  |  |  |  |
| --- | --- | --- | --- |
| Столбец | Тип | Модификаторы | Описание |
| aircraft\_code | char(3) | NOT NULL | Код самолета, IATA |
| seat\_no | varchar(4) | NOT NULL | Номер места |
| fare\_conditions | varchar(10) | 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

**Таблица bookings.ticket\_flights**

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 |
| ticket\_no | char(13) | NOT NULL | Номер билета |
| flight\_id | integer | NOT NULL | Идентификатор рейса |
| fare\_conditions | varchar(10) | NOT NULL | Класс обслуживания |
| amount | numeric(10,2) | 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)

**Таблица bookings.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.

|  |  |  |  |
| --- | --- | --- | --- |
| 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)

**Представление "bookings.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 |

**Материализованное представление bookings.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, используя БД.**

1. Количество посадочных, свободных, занятых мест по каждой группе самолетов и по классам обслуживания в динамике
2. Перечень посадочных мест, разбиение по классам обслуживания и их общее количество для каждого типа самолета
3. Список аэропортов по городам России
4. Перечень аэропортов, куда, в какие дни недели и за какое время можно долететь из указанного аэропорта
5. Ближайший рейс из заданного аэропорта вылета в заданный аэропорт прилета
6. Поиск бронирования по коду, ФИО пассажира, номеру билета
7. Состав перелетов для билета с заданным номером
8. Создание нового бронирования, редактирование, удаление, просмотр бронирования
9. Проверка регистрации на забронированный рейс
10. Объем пассажироперевозок по городам, аэропортам, рейсам в динамике
11. Выручка по аэропортам в разбивке по направлениям вылетов, по классам обслуживания, по типам самолетов
12. Статистика по отмененным и задержанным рейсам в динамике с разбивкой по аэропортам и тапам самолетов
13. Рейтинг аэропортов по числу отмененных рейсов, по числу задержанных рейсов
14. Рейтинг аэропортов по количеству ежедневных прилетов/вылетов
15. **Список SQL запросов из приложения №2 с описанием логики их выполнения** – cм. в скрипте SQL**.**