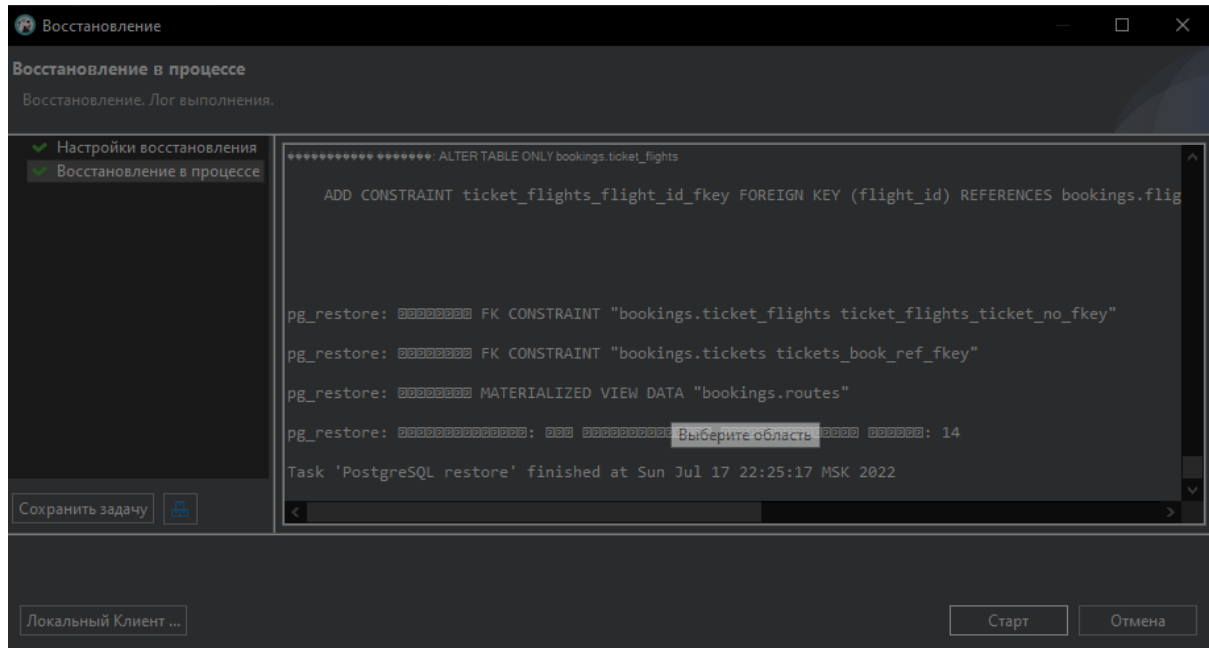


Итоговая работа

1. В работе использовался локальный тип подключения.

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



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



3. Краткое описание БД - из каких таблиц и представлений состоит.

Список отношений

Имя	Тип	Размер	Описание
aircrafts	таблица	16 kB	Самолеты
airports	таблица	48 kB	Аэропорты
boarding_passes	таблица	31 MB	Посадочные талоны
bookings	таблица	13 MB	Бронирования
flights	таблица	3 MB	Рейсы
flights_v	представление	0 kb	Рейсы
routes	мат. предст.	136 kB	Маршруты
seats	таблица	88 kB	Места
ticket_flights	таблица	64 MB	Перелеты
tickets	таблица	47 MB	Билеты

4. Развернутый анализ БД - описание таблиц, логики, связей и бизнес области (частично можно взять из описания базы данных, оформленной в виде анализа базы данных). Бизнес задачи, которые можно решить, используя БД.

Таблица *bookings.aircrafts*

Каждая модель воздушного судна идентифицируется своим трехзначным кодом (aircraft_code). Указывается также название модели (model) и максимальная дальность полета в километрах (range).

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(aircraft_code)
```

```
TABLE "seats" FOREIGN KEY (aircraft_code)
```

```
REFERENCES aircrafts(aircraft_code) ON DELETE CASCADE
```

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 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
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	timestampz	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
<code>flight_id</code>	<code>serial</code>	<code>NOT NULL</code>	Идентификатор рейса
<code>flight_no</code>	<code>char(6)</code>	<code>NOT NULL</code>	Номер рейса
<code>scheduled_departure</code>	<code>timestampz</code>	<code>NOT NULL</code>	Время вылета по расписанию
<code>scheduled_arrival</code>	<code>timestampz</code>	<code>NOT NULL</code>	Время прилёта по расписанию
<code>departure_airport</code>	<code>char(3)</code>	<code>NOT NULL</code>	Аэропорт отправления
<code>arrival_airport</code>	<code>char(3)</code>	<code>NOT NULL</code>	Аэропорт прибытия
<code>status</code>	<code>varchar(20)</code>	<code>NOT NULL</code>	Статус рейса
<code>aircraft_code</code>	<code>char(3)</code>	<code>NOT NULL</code>	Код самолета, IATA
<code>actual_departure</code>	<code>timestampz</code>		Фактическое время вылета
<code>actual_arrival</code>	<code>timestampz</code>		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_data). 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	timestampz	Время вылета по расписанию
scheduled_departure_local	timestamp	Время вылета по расписанию, местное время в пункте отправления
scheduled_arrival	timestampz	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	timestampz	Фактическое время вылета
actual_departure_local	timestamp	Фактическое время вылета, местное время в пункте отправления
actual_arrival	timestampz	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

Функция pow Демонстрационная база содержит временной «срез» данных — так, как будто в некоторый момент была сделана резервная копия реальной системы. Например, если некоторый рейс имеет статус Departed, это означает,

что в момент резервного копирования самолет вылетел и находился в воздухе. Позиция «среза» сохранена в функции `bookings.now()`. Ей можно пользоваться в запросах там, где в обычной жизни использовалась бы функция `now()`. Кроме того, значение этой функции определяет версию демонстрационной базы данных. Актуальная версия на текущий момент — от 13.10.2016.

Используя данную БД, можно решить следующие бизнес задачи:

- 1)Повышение эффективности перевозок
- 2)Перераспределение рейсов
- 3)Корректировка ценовой политики
- 4)Разработка программ лояльности
- 5)Создание рекомендательной модели

5. Список SQL запросов из приложения №2 с описанием логики их выполнения.

Выполнено в файле с запросами.

Приложение №2

Вопрос	В решении обязательно должно быть использовано	Баллы за запросы
В каких городах больше одного аэропорта?	-	10
В каких аэропортах есть рейсы, выполняемые самолетом с максимальной дальностью перелета?	Подзапрос	15
Вывести 10 рейсов с максимальным временем задержки вылета	Оператор LIMIT	15
Были ли брони, по которым не были получены посадочные талоны?	Верный тип JOIN	15
Найдите количество свободных мест для каждого рейса, их % отношение к общему количеству мест в самолете. Добавьте столбец с накопительным итогом - суммарное накопление количества вывезенных пассажиров из каждого аэропорта на каждый день. Т.е. в этом столбце должна отражаться накопительная сумма - сколько человек уже вылетело из данного аэропорта на этом	Оконная функция; подзапросы или/и сte	35

или более ранних рейсах в течении дня		
Найдите процентное соотношение перелетов по типам самолетов от общего количества	Подзапрос или окно; оператор ROUND	25
Были ли города, в которые можно добраться бизнес - классом дешевле, чем эконом-классом в рамках перелета?	СТЕ	25
Между какими городами нет прямых рейсов?	Декартово произведение в предложении FROM; самостоятельны о созданные представления (если облачное подключение, то без представления) ; оператор EXCEPT	25
Вычислите расстояние между аэропортами, связанными прямыми рейсами, сравните с допустимой максимальной дальностью перелетов в самолетах, обслуживающих эти рейс*	Оператор RADIANS или использование sind/cosd; CASE	35