

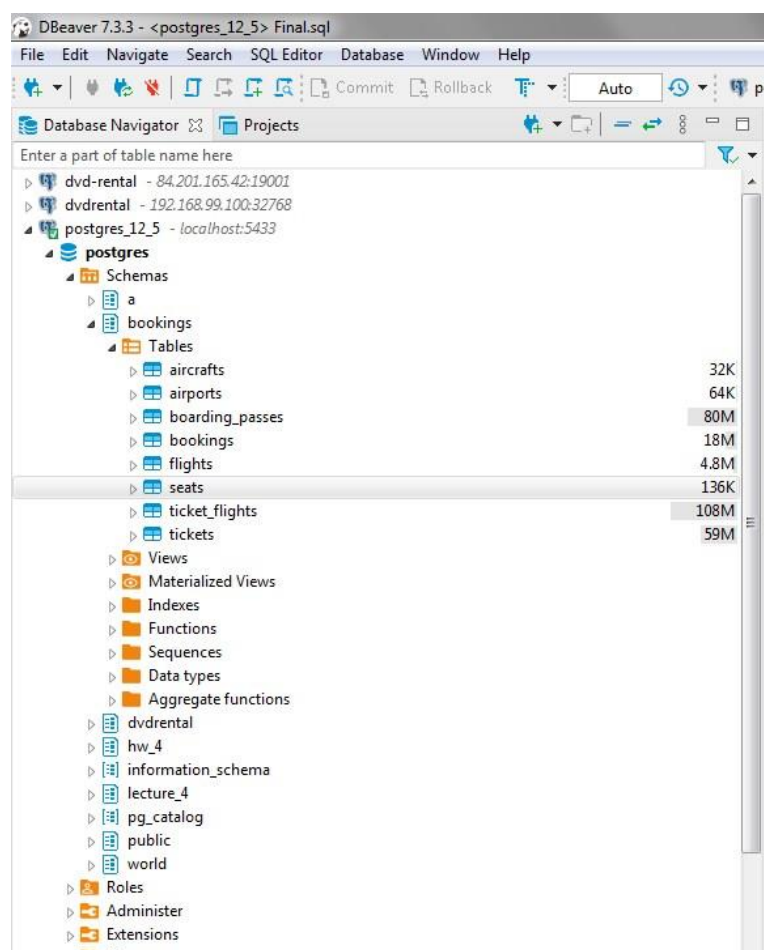
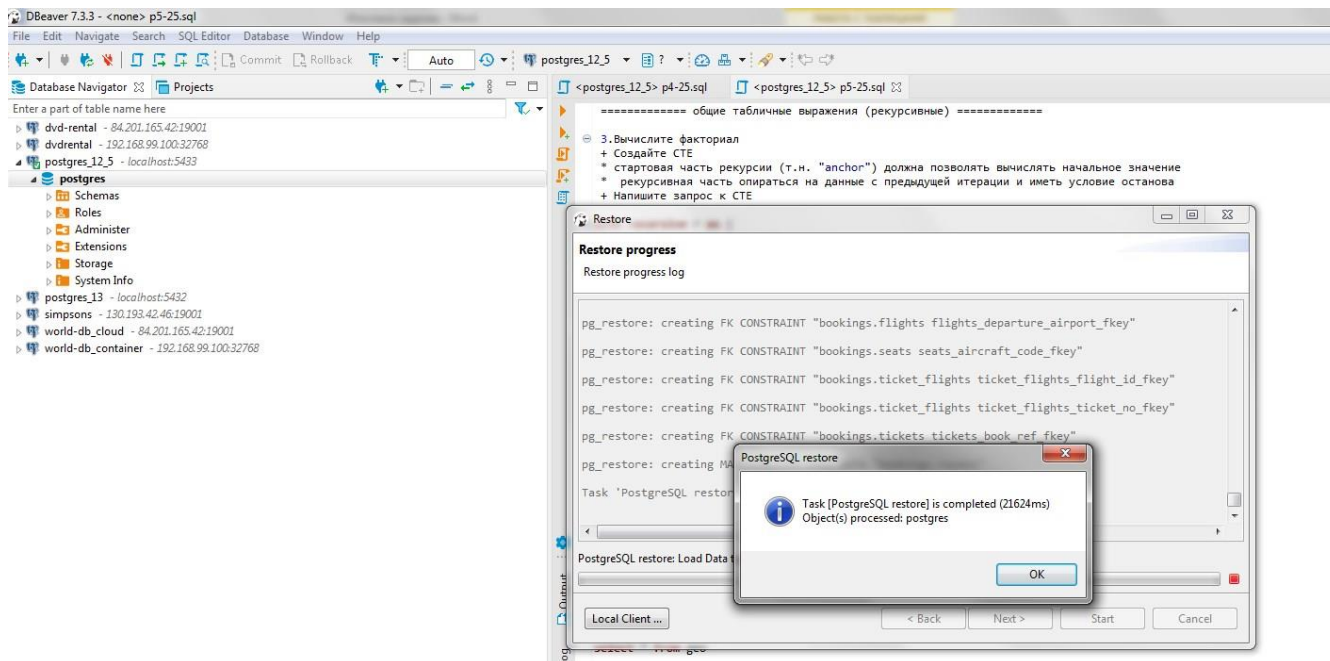


Проектная работа по модулю “SQL и получение данных”

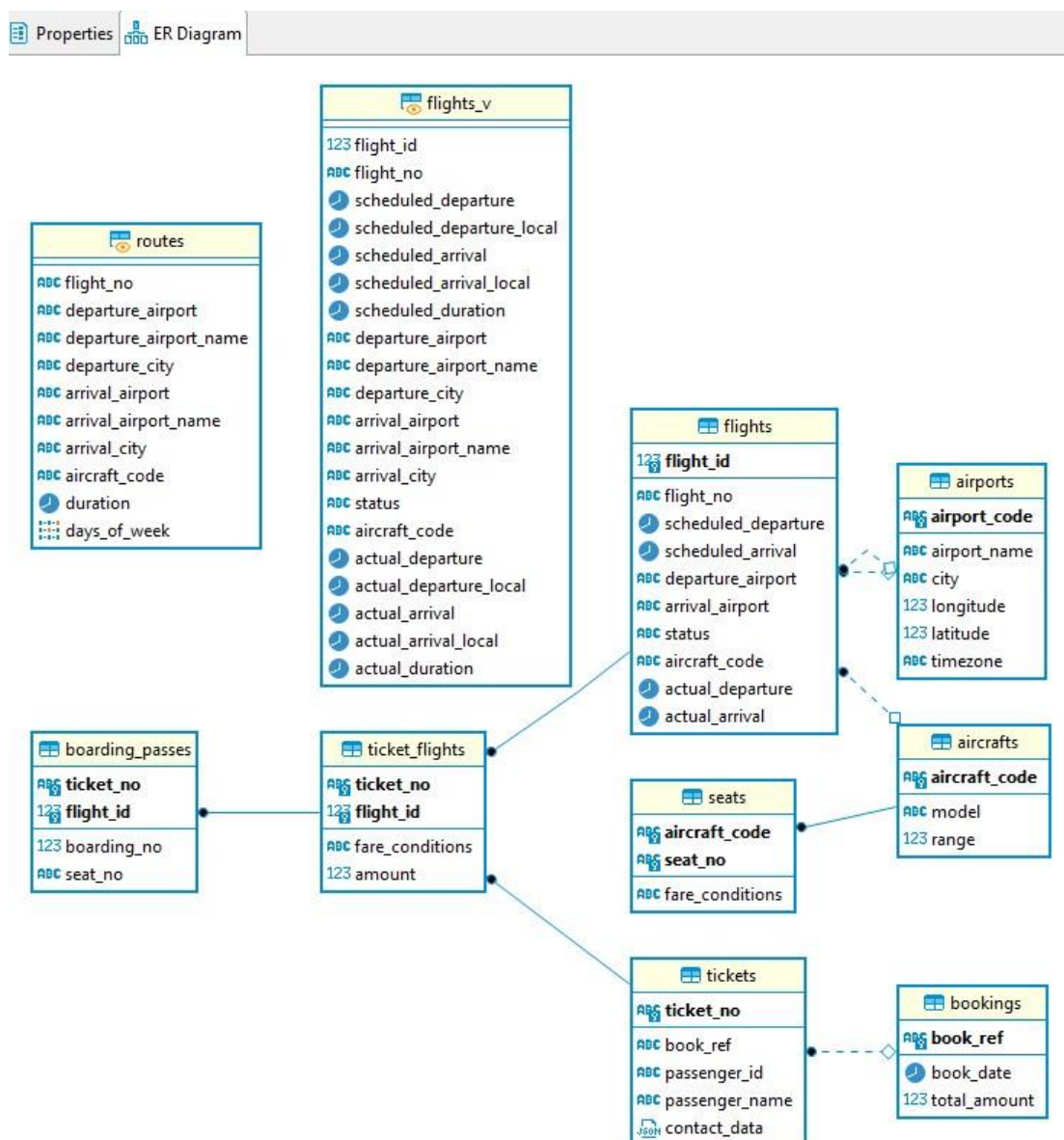
ГРУППА SQL-25

Итоговая работа

1. В работе использовался локальный тип подключения.
2. Скриншот успешного восстановления:



3. Скриншот ER-диаграммы из DBeaver'а согласно подключения.



4. Краткое описание БД.

Имя	Тип	Описание
aircrafts	таблица	Самолеты
airports	таблица	Аэропорты
boarding_passes	таблица	Посадочные талоны
bookings	таблица	Бронирования
flights	таблица	Рейсы
flights_v	представление	Рейсы
routes	мат. предст.	Маршруты
seats	таблица	Места
ticket_flights	таблица	Перелеты
tickets	таблица	Билеты

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

5.1 Таблица 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

5.2 Таблица 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)

5.3 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).

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)

5.4 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.

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)

5.5 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.

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)

5.6 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.

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

5.7 Таблица 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)

5.8 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_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)

5.9 Представление " flights_v".

Над таблицей flights создано представление flights_v, содержащее дополнительную информацию.

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

5.10 Материализованное представление 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.

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

5.11 Бизнес задачи, которые можно решить, используя БД.



В рассматриваемой базе данных, на мой взгляд, есть два направления работы с информацией:

Во-первых - финансовая составляющая. Данные могут быть использованы:

- для оценки и анализа выручки от продажи полетов в разрезе
 - маршрутов,
 - дат полетов,
- расчета коэффициента сезонности,
- фондоотдачи (заполняемость самолета),
- обоснованности цен по классам обслуживания (популярности тех или иных классов в зависимости от коэффициента, применяемого к цене эконом. класса, либо в зависимости от стоимости и дальности перелета).

Во-вторых – техническая составляющая. Данные могут быть использованы:

- для определения наибольшей загруженности и частоты использования конкретных бортов для составления графика обслуживания и плановых ремонтов, а также своевременной закупки расходуемых элементов,
- для контроля соблюдения расписания полетов путем сравнения фактических временных данных с расчетными по расписанию и анализа причин наиболее частых расхождений в том числе в разрезе аэропортов и бортов.
- для определения более подходящих моделей самолетов с учетом фактора дальности перелета и популярности направления.

В-третьих, возможно следует оценить возможность развития бизнеса с точки зрения поиска вариантов кооперации с другими авиакомпаниями по тем рейсам, где загрузка меньше чем необходима для достижения точки безубыточности, и наоборот, возможно стоит рассмотреть введение новых рейсов и тарифов на популярные направления.

6. Список SQL запросов с описанием логики их выполнения.

1. В каких городах больше одного аэропорта?
Группируем таблицу airports по городам и считаем количество аэропортов в каждом городе, если количество больше одного, то выводим соответствующую строку.
2. В каких аэропортах есть рейсы, выполняемые самолетом с максимальной дальностью перелета?
В подзапросе находим самолеты с максимальной дальностью перелета (select max(a."range") from aircrafts a). В основном запросе в таблице flights находим уникальные аэропорты где aircraft_code в списке самолетов, выбранных подзапросом.
3. Вывести 10 рейсов с максимальным временем задержки вылета.
Находим разницу между актуальным временем вылета и плановым и если она актуальна (не null), то сортируем в порядке убывания и находим первые десять строк.
4. Были ли брони, по которым не были получены посадочные талоны?
Берем таблицу tickets, где есть брони и соединяем с таблицей boarding_passes чтобы выявить каким билетам из tickets нет соответствия в boarding_passes. Выводим такие брони, где нет соответствия и группируем по ним, чтобы убрать дубли.
5. Найдите свободные места 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 или более ранних рейсах за день.
Сначала находим количество пассажиров, которым выданы посадочные талоны на рейс, затем общее количество мест в самолете, который осуществляет полет, соединяем результирующие таблицы, добавляем к ним flights, чтобы получить аэропорт и время отправления. Выводим расчет по свободным местам, их проценту, и оконную функцию для расчета нарастающего итога по вывезенным пассажирам в разрезе даты и аэропорта.
6. Найдите процентное соотношение перелетов по типам самолетов от общего количества.
В подзапросе определяем общее количество перелетов всеми самолетами, в таблице flights группируем по видам самолетов и находим количество перелетов каждым самолетом, добавляем таблицу aircrafts, чтобы вывести название самолета и в выводе прописываем вычисления и округления.
7. Были ли города, в которые можно добраться бизнес - классом дешевле, чем эконом-классом в рамках перелета?
Во-первых, находим уникальную стоимость Эконом и Бизнес классов по каждому уникальному перелету из таблицы ticket_flights в двух отдельных cte. Во-вторых, соединяем их по flight_id и проверяем, есть ли такие перелеты, где эконом тариф больше бизнес. Во втором варианте можно добавить таблицы flights и airports, чтобы вывести название города прибытия и обернуть основной запрос в подзапрос, для вывода уникального города, соответствующего условию.
8. Между какими городами нет прямых рейсов?
Выводим все возможные маршруты с помощью декартова произведения, и убираем дубли и строки, где города отправления и прибытия одинаковые. Создаем материализованное представление, т.к. данные в нем не оперативные и поменяются только тогда, когда в обслуживании появятся новые города.

Выводим с помощью таблицы `flights` все имеющиеся маршруты, соединяем с таблицей `airports`, чтобы получить данные как по городу вылета, так и по городу прилета.

Группируем, чтобы убрать дубли, создаем обычное представление на случай если внезапно появятся новые маршруты между уже обслуживаемыми городами - информация будет оперативной.

Используем EXCEPT, чтобы убрать из всех возможных маршрутов имеющиеся и получаем не существующие маршруты между городами.

9. Вычислите расстояние между аэропортами, связанными прямыми рейсами, сравните с допустимой максимальной дальностью перелетов в самолетах, обслуживающих эти рейсы

Выводим с помощью таблицы `flights` все имеющиеся маршруты, соединяем с таблицей `airports`, чтобы получить данные как по аэропорту вылета, так и по аэропорту прилета.

Соединяем с таблицей `aircrafts`, чтобы получить максимальную дистанцию, на которую рассчитана эта модель самолета.

По формуле вычисляем расстояние до городов и сравниваем его с максимальной дистанцией. Сортируем, чтобы узнать наиболее критические в этом отношении маршруты.