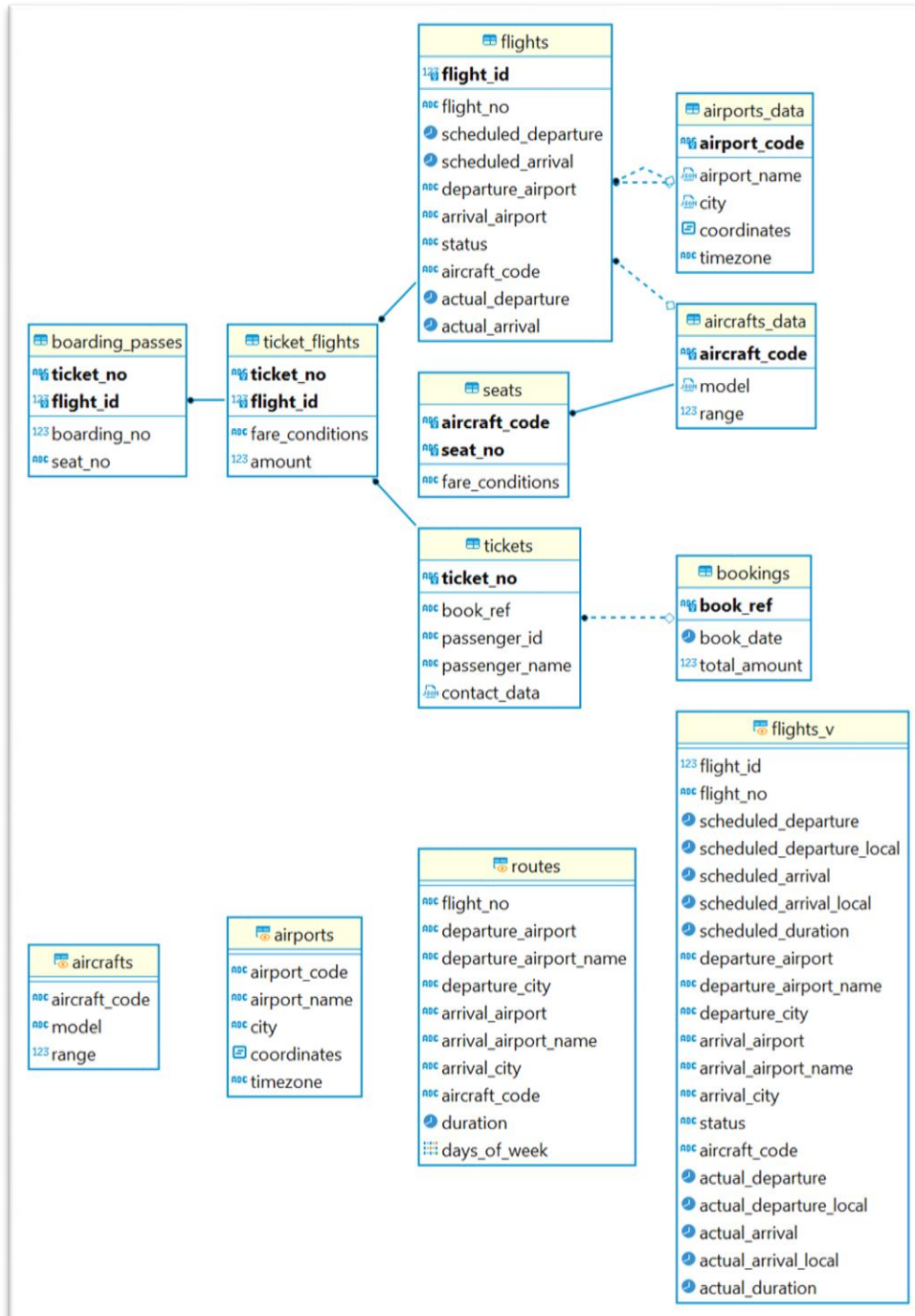


Итоговая работа

1. В работе использовался облачный тип подключения.
2. Скриншот ER-диаграммы из DBeaver'a.



Краткое описание БД - из каких таблиц и представлений состоит.

Таблицы:

1.aircrafts

aircraft_code - Код самолета, IATA первичный ключ

model - Модель самолета

range - Максимальная дальность полета, км

2.airports

airport_code - Код аэропорта первичный ключ

airport_name - Название аэропорта

city - Город

coordinates- Координаты аэропорта: широта и долгота

timezone - Временная зона аэропорта

3.boarding_passes

ticket_no - Номер билета первичный ключ

flight_id - Идентификатор рейса

boarding_no - Номер посадочного талона

seat_no - Номер места

4.bookings

book_ref - Номер бронирования первичный ключ

book_date - Дата бронирования

total_amount - Полная сумма бронирования

5.flights

flight_id - Идентификатор рейса первичный ключ

flight_no - Номер рейса

scheduled_departure - Время вылета по расписанию

scheduled_arrival - Время прилёта по расписанию

departure_airport - Аэропорт отправления

arrival_airport - Аэропорт прибытия

status - Статус рейса

aircraft_code - Код самолета, IATA

actual_departure - Фактическое время вылета

actual_arrival - Фактическое время прилёта

6.seats

aircraft_code - Код самолета, IATA

seat_no - Номер места

fare_conditions - Класс обслуживания

7.ticket_flights

ticket_no - Номер билета

flight_id - Идентификатор рейса

fare_conditions - Класс обслуживания

amount - Стоимость перелета

8.tickets

ticket_no - Номер билета первичный ключ

book_ref - Номер бронирования

passenger_id - Идентификатор пассажира

passenger_name - Имя пассажира

contact_data - Контактные данные пассажира

Представления:

1.flights_v

flight_id

flight_no

scheduled_departure

scheduled_departure_local

scheduled_arrival

scheduled_arrival_local

scheduled_duration

departure_airport

departure_airport_name

departure_city
arrival_airport
arrival_airport_name
arrival_city
status
aircraft_code
actual_departure
actual_departure_local
actual_arrival
actual_arrival_local
actual_duration

Мат.представления:

1.routes

flight_no
departure_airport
departure_airport_name
departure_city
arrival_airport
arrival_airport_name
arrival_city
aircraft_code
duration
days_of_week

Развернутый анализ БД - описание таблиц, логики, связей и бизнес области. Бизнес задачи, которые можно решить, используя БД.

Таблица aircrafts_data

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

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) REFERENCES aircrafts_data(aircraft_code) ON DELETE CASCADE

Таблица airports_data

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 и долгота (coordinates) и часовой пояс (timezone).

Индексы:

- PRIMARY KEY, btree (airport_code)

Ссылки извне:

- TABLE "flights"
 - FOREIGN KEY (arrival_airport) REFERENCES airports_data(airport_code)
 - FOREIGN KEY (departure_airport) REFERENCES airports_data(airport_code)

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).

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

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.

Индексы:

- PRIMARY KEY, btree (book_ref)

Ссылки извне:

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

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.

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_data(airport_code)
- FOREIGN KEY (departure_airport) REFERENCES airports_data(airport_code)

Ссылки извне:

- TABLE "ticket_flights" FOREIGN KEY (flight_id) REFERENCES flights(flight_id)

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.

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 ticket_flights

Перелет 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)

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_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.

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 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).

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.

Именно такая информация и составляет материализованное представление routes.

Бизнес задачи, которые можно решить, используя БД

1. Множество вариантов анализа наполненности рейсов:

- по направлениям,
- по дням недели,
- по классу обслуживания,

и т.п.

Данный пункт позволяет предложить варианты оптимизации расходов с помощью объединения или отмены некоторых рейсов.

2. Получение ресурсов с задержкой вылета для последующего изучения причин.
3. Получение данных для возврата денег за неиспользованные билеты.
4. Возможность развития новых направлений перелетов с помощью получения городов, между которыми нет прямых рейсов.
5. Анализ нагрузки и подсчет полётных часов

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

№1. В каких городах больше одного аэропорта?

```
with cte1 as (select city, count(airp) as count from airports group by city)
```

city	count
Ульяновск	2
Москва	2

1)С помощью оконной функции, используя таблицу airports производим подсчет аэропортов в каждом городе.

2)В основном запросе выводим список городов, где их количество больше 1.

№2 В каких аэропортах есть рейсы, выполняемые самолетом с максимальной дальностью перелета?(Подзапрос)

```
select DISTINCT departure_airport, departure_airport_name from flights_v where (select max(distance) from flights_v where departure_airport = departure_airport)
```

departure_airport_name
Внуково
Пермь
Толмачёво
Шереметьево
Кольцово
Сочи
Домодедово

1)Используем представление flights_v, выводим поле departure_airport_name

2)В условии Where используем подзапрос в котором проводим сортировку по максимальной дальности самолета, и выводим первую строку из списка.

№3 Вывести 10 рейсов с максимальным временем задержки вылета(Оператор LIMIT)

```
select * from flights order by (scheduled_departure - actual_departure) desc limit 10
```

flight_id	flight_no	scheduled_departure	actual_departure	departure_airport	arrival_airport	status	aircraft_code	actual_c
17368	PG0132	7-05-22 12:25:00.000 +0300	7-05-22 13:55:00.000 +0300	LED	VOZ	Arrived	CR2	7-05-22 17
31478	PG0531	7-05-17 09:15:00.000 +0300	7-05-17 11:45:00.000 +0300	YKS	UUD	Arrived	CR2	7-05-17 13
29148	PG0589	7-07-29 15:30:00.000 +0300	7-07-29 15:55:00.000 +0300	PEE	SVX	Arrived	SU9	7-07-29 20
2702	PG0164	7-07-29 15:25:00.000 +0300	7-07-29 18:35:00.000 +0300	DME	NLX	Arrived	SU9	7-07-29 15
44711	PG0544	7-07-06 09:55:00.000 +0300	7-07-06 12:05:00.000 +0300	OVS	DME	Arrived	SU9	7-07-06 14
48079	PG0364	7-07-19 11:45:00.000 +0300	7-07-19 13:20:00.000 +0300	KRR	ULV	Arrived	763	7-07-19 16
53560	PG0151	7-07-03 14:00:00.000 +0300	7-07-03 14:30:00.000 +0300	MCX	MRV	Arrived	SU9	7-07-03 18
6310	PG0337	7-06-01 12:45:00.000 +0300	7-06-01 14:10:00.000 +0300	DME	ARH	Arrived	CR2	7-06-01 17
55271	PG0354	7-07-03 08:45:00.000 +0300	7-07-03 10:10:00.000 +0300	NOZ	KJA	Arrived	CN1	7-07-03 13
45487	PG0444	7-05-25 12:30:00.000 +0300	7-05-25 14:00:00.000 +0300	OVS	TJM	Arrived	CN1	7-05-25 16

1)Используя оконную функцию выводим все строки из flights в которых актуальное время вылета не пустое.

2)В основном запросе выведем дополнительный столбец в котором высчитали разность между временем отправки по графику и реальным временем отправки

3)делаем сортировку от большего к меньшему по дополнительному столбцу и выводим первые 10 значений.

№4 Были ли брони, по которым не были получены посадочные талоны?(Верный тип JOIN)

Результат 1 ×

select case when count(b.book_ref) > 0 then 'Введите SQL в

Таблица	have_armor	amount
1	Да	127 899

- 1)Основная таблица bookings - бронирования, к ней присоединяем таблицы: tickets(Билеты) и boarding_passes(Посадочные талоны)
- 2)Нам нужны все поля в которых boarding_no(Номер посадочного места) пуст
- 3) производим подсчет сколько было бронирований без посадочных талонов через count

№5 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 или более ранних рейсах в течении дня(Оконная функция,Подзапросы или/и cte)

with cte1 as (select f.flight_id,f.flight_no,f.aircraft_id,f.departure_airport,f.actual_departure,f.boarded_count,f.free_seats,f.free_seats_percent,f.accumulated_passengers

Таблица	flight_no	departure_airport	actual_departure	boarded_count	free_seats	free_seats_percent	Накопительно пассажиров
1	PG0480	AAQ	-05-17 12:26:0	4	93	96	4
2	PG0252	AAQ	-05-17 13:09:0	49	81	62	53
3	PG0480	AAQ	-05-18 12:25:0	7	90	93	7
4	PG0252	AAQ	-05-18 13:08:0	47	83	64	54
5	PG0480	AAQ	-05-19 12:27:0	2	95	98	2
6	PG0252	AAQ	-05-19 13:07:0	50	80	62	52
7	PG0480	AAQ	-05-20 12:26:0	5	92	95	5
8	PG0252	AAQ	-05-20 13:09:0	49	81	62	54
9	PG0480	AAQ	-05-21 12:28:0	1	96	99	1
10	PG0252	AAQ	-05-21 13:06:0	51	79	61	52
11	PG0480	AAQ	-05-22 12:30:0	5	92	95	5
12	PG0252	AAQ	-05-22 13:07:0	60	70	54	65
13	PG0480	AAQ	-05-23 12:32:0	8	89	92	8

- 1)Через первую оконную функцию получаем количество выданных посадочных талонов по каждому рейсу
- 2) С помощью ограничения actual_departure is not null отслеживаем уже вылетевшие рейсы
- 3)Через вторую оконную функцию получаем количество мест в самолёте
- 4)В итоговом запросе оба СТЕ соединяем
- 5)Для подсчета накопительной суммы используется оконная функция с разделением по аэропорту отправления и времени вылета приведенному к формату date.

№6. Найдите процентное соотношение перелетов по типам самолетов от общего количества.(Подзапрос или окно,Оператор ROUND)

Результат 1 ×

select distinct a.city,a2.city from airports a, air

city	city
Новосибирск	Йошкар-Ола
Норильск	Череповец
Нижневартовск	Архангельск
Новокузнецк	Омск
Новокузнецк	Ханты-Мансийск
Анапа	Ставрополь
Калининград	Иваново
Новосибирск	Чебоксары
Чита	Москва
Хабаровск	Архангельск
Ижевск	Саратов
Ставрополь	Екатеринбург
Петрозаводск	Уфа

Save Cancel Script 200 9 584 Rows MSK ru Запись Инт

- 1)С помощью декартового произведения находим уникальное количество пар город в которые существуют рейсы
- 2)Далее находим разность между уникальными значениями городов и общим списков рейсов, итог этой разности и будут города в которые нет прямых рейсов

№9. Вычислите расстояние между аэропортами, связанными прямыми рейсами, сравните с допустимой максимальной дальностью перелетов в самолетах, обслуживающих эти рейсы(Оператор RADIANS или использование sind/cosd,CASE)

airports(+) 1 ×

select distinct a1.airport_name as dispatch_city,

dispatch_city_	arrival_city_	aircraft_range_	between_cities	result
Абакан	Архангельск	6 700	5 631,35	Долетит
Абакан	Грозный	4 200	5 068,1	Не долетит
Абакан	Кызыл	1 200	335,47	Долетит
Абакан	Москва	6 700	5 946,5	Долетит
Абакан	Новосибирск	1 200	971,18	Долетит
Абакан	Томск	1 200	686,69	Долетит
Анадырь	Москва	6 700	15 446,94	Не долетит
Анадырь	Москва	6 700	15 511,94	Не долетит
Анадырь	Москва	6 700	15 520,39	Не долетит
Анадырь	Хабаровск	6 700	4 991,37	Долетит
Астрахань	Барнаул	2 700	3 957,29	Не долетит
Астрахань	Москва	2 700	1 344,29	Долетит
Астрахань	Москва	2 700	1 416,6	Долетит

Save Cancel Script 200 618 Rows: 1 Инт. встав

- 1)Соединяем две таблицы airports и aircrafts_data
- 2)По условию того, что если рассчитанная дальность между городами меньше дальности самолета, то самолет долететь до конечного пункта на одном баке, иначе нет.3)Расстояние между городами высчитываем по формуле из задания.