Проектная работа по модулю “SQL и получение данных”

1.b.1.Описание предметной области и проблемы, которую вы решаете своей работой

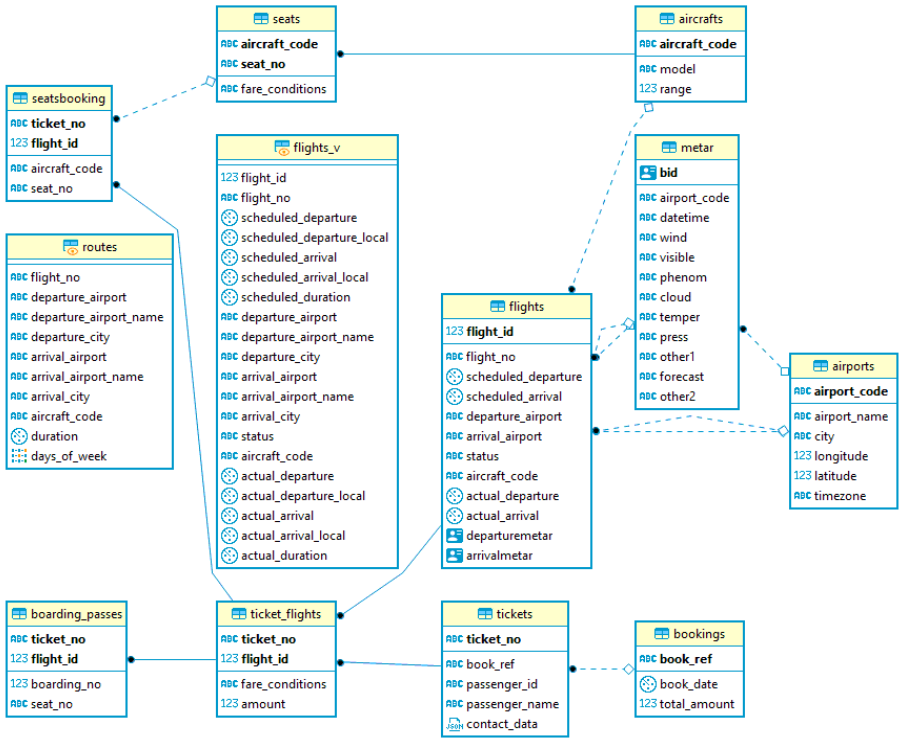
Выбранная область – выборка данных для анализа корреляции параметров перелётов и задержек прибытия. Результаты подобного анализа могут быть использованы для выявления системных организационных проблем, вызывающих задержки рейсов, а также разработки системы рекомендаций запаса времени между стыковочными рейсами, учитывающей риски задержки рейсов. Перечнем из 7 запросов (запросы 5.1-5.7) выбираются данные о распределении средних задержек прибытия рейса в зависимости от конкретного рейса, города отправления, аэропорта отправления, модели самолёта, количества пассажиров, дня недели, часа суток. Цель работы – создание скриптов выгрузки данных, результаты анализа целью работы не являются, так как данные являются тестовыми.

1.b.2-3 *Текстовое описание таблиц и логики их связи*

seatsbooking - таблица с выбранными пассажирами местами, для модуля продажи/самостоятельного выбора мест в самолёте. Связана с таблицами seats и tickets_flights по их составным ключам. Эти внешние ключи выбраны как ссылающиеся на первичные данные. В качестве первичного ключа выбран составной ключ tickets_flights, так как он должен быть уникальным, в отличие от места самолёта, которое будет повторяться от полёта к полёту в разных билетах.

Metar - таблица для погодных кодов METAR. Содержит связи с таблицей airports по коду аэропорта кода METAR и двойную связь с таблицей flights по добавленным в эту таблицу полям ID кода METAR для вылета и прилёта. В качестве первичного ключа выбран суррогатный ключ (в качестве естественного первичного ключа может выступать только составной ключ из кода аэропорта и даты METAR, что неудобно в качестве связи с таблицей flights) в формате UUID (как лучшая практика).

1.b.4 ER-диаграма:



2. Ответы на вопросы пункта 3 (скрипты с соответствующими номерами в приложении и отдельным файлом):

(2.)3.1. В каких городах больше одного аэропорта? Ульяновск и Москва:

(2.)3.2. Были ли брони, по которым не совершались перелеты? Да, перечень кодов бронирования (127901 бронирование):

*(2.)*3.3. В каких аэропортах(!) есть рейсы, которые обслуживаются самолетами с максимальной дальностью перелетов? Пермь, Кольцово, Толмачёво, Домодедово, Сочи, Шереметьево, Внуково:

(2.)3.4. Между какими городами(!) пассажиры делали пересадки? Запрос выводит список пар городов и количество пересадок (626 городов)

(2.)3.4.1 При полетах между какими городами делают пересадки чаще? Запрос выводит пар городов и количество пересадок (самая частая пара городов - Москва и Новосибирск)

(2.)3.4.2 Какие города используют для пересадок чаще? Запрос выводит перечень городов (91) с частотой пересадок. Лидер - Москва.*

(2.)3.4.3 В каких городах пересадки самые длительные?Запрос выводит список городов со средней продолжительностью пересадки. Лидеры: Горно-Алтайск, Усинск, Нягань:*

(2.)3.5. Между какими городами(!) нет прямых рейсов? Запрос выводит список пар направлений в виде пар городов, между которыми нет прямых рейсов (9584)*

Скрипты

```
-- 3.1. В каких городах больше одного аэропорта? Ульяновск и Москва:
SELECT city, COUNT(city) AS "airport count" FROM airports GROUP BY city HAVING COUNT(city)>1;
-- 3.2. Были ли брони, по которым не совершались перелеты? Да, перечень кодов бронирования (127901 бронирование):
SELECT book_ref FROM tickets t LEFT JOIN boarding_passes b ON t.ticket_no=b.ticket_no WHERE b.ticket_no IS NULL;
-- 3.3. В каких аэропортах(!) есть рейсы, которые обслуживаются самолетами с максимальной дальностью перелетов? Пермь, Кольцово, Толмачёво, Домодедово, Сочи, Шереметьево, Внуково:
SELECT DISTINCT airport_code, airport_name, city FROM airports a LEFT JOIN flights f ON ((a.airport_code=f.arrival_airport) OR (a.airport_code=f.departure_airport)) WHERE aircraft_code IN (SELECT aircraft_code FROM aircrafts WHERE range>=11000);

-- 3.4. Между какими городами(!) пассажиры делали пересадки? Запрос выводит список пар городов и количество пересадок (626 городов)
-- 3.4.1 При полетах между какими городами делают пересадки чаще? Запрос выводит пар городов и количество пересадок (самая частая пара городов - Москва и Новосибирск)

WITH flights_new AS -- таблица полётов, с номерами билетов
(SELECT tf.ticket_no, f.departure_airport, f.arrival_airport, --выбираем из двух таблиц номер билета, аэропорт вылета, аэропорт прилёта
        ROW_NUMBER() OVER (partition BY tf.ticket_no ORDER BY tf.ticket_no, f.actual_departure) AS num, -- добавляем порядковый номер перелёта в билете
        COUNT(tf.ticket_no) OVER (partition BY tf.ticket_no) AS flightcount -- и общее количество полётов по билету
        FROM ticket_flights tf LEFT JOIN flights f ON tf.flight_id=f.flight_id)
,
firstflits AS -- таблица первых перелётов с пересадками в билетах, в которых нас интересует аэропорт вылета, к которому сразу подбираем город
(SELECT fn.ticket_no, a.city AS departure_city FROM flights_new fn LEFT JOIN airports a ON fn.departure_airport=a.airport_code WHERE (flightcount>1) AND (num=1))
,
lastflights AS -- таблица последних перелётов с пересадками в билетах, в которых нас интересует аэропорт прибытия, к которому сразу подбираем город
(SELECT fn.ticket_no, a.city AS arrival_city FROM flights_new fn LEFT JOIN airports a ON fn.departure_airport=a.airport_code WHERE (flightcount>1) AND (num=flightcount))
-- в основном запросе объединяем таблицы первых и последних перелётов, сразу группируем парами и считаем количество
SELECT ff.departure_city, lf.arrival_city, COUNT (ff.departure_city) AS countflight FROM firstflits ff LEFT JOIN lastflights lf ON ff.ticket_no=lf.ticket_no GROUP BY ff.departure_city, lf.arrival_city ORDER BY countflight DESC
;
--3.4.2* Какие города используют для пересадок чаще? Запрос выводит перечень городов (91) с частотой пересадок. Лидер - Москва.
WITH flights_new AS -- таблица полётов, с номерами билетов
(SELECT tf.ticket_no, f.departure_airport, f.arrival_airport, --выбираем из двух таблиц номер билета, аэропорт вылета, аэропорт прилёта
        ROW_NUMBER() OVER (partition BY tf.ticket_no ORDER BY tf.ticket_no, f.actual_departure) AS num, -- добавляем порядковый номер перелёта в билете
        COUNT(tf.ticket_no) OVER (partition BY tf.ticket_no) AS flightcount -- и общее количество полётов по билету
        FROM ticket_flights tf LEFT JOIN flights f ON tf.flight_id=f.flight_id)
,
transitflights AS -- таблица промежуточных перелётов в билетах, в которых нас интересует аэропорт вылета, к которому сразу подбираем город
(SELECT a.city AS transit_city FROM flights_new fn LEFT JOIN airports a ON fn.departure_airport=a.airport_code WHERE (flightcount>1) AND (num>1))
-- в основном запросе считаем количество пересадок в транзитных городах
SELECT transit_city, COUNT (transit_city) AS counttransits FROM transitflights GROUP BY transit_city ORDER BY counttransits DESC
;
-- 3.4.3* В каких городах пересадки самые длительные?Запрос выводит список городов со средней продолжительностью пересадки. Лидеры: Горно-Алтайск, Усинск, Нягань:
WITH flights_new_time AS -- таблица полётов, с номерами билетов и датами вылета и прибытия
(SELECT tf.ticket_no, f.departure_airport, f.arrival_airport, scheduled_departure, scheduled_arrival, --выбираем из двух таблиц номер билета, аэропорт вылета, аэропорт прилёта, время вылета, время прибытия
        ROW_NUMBER() OVER (partition BY tf.ticket_no ORDER BY tf.ticket_no, f.actual_departure) AS num, -- добавляем порядковый номер перелёта в билете
        COUNT(tf.ticket_no) OVER (partition BY tf.ticket_no) AS flightcount -- и общее количество полётов по билету
        FROM ticket_flights tf LEFT JOIN flights f ON tf.flight_id=f.flight_id)
,
flights_new AS -- таблица полётов спродолжительностью
(SELECT a.ticket_no, a.departure_airport, a.arrival_airport, a.scheduled_departure-b.scheduled_arrival AS duration, a.num, a.flightcount
```

```
        FROM flights_new_time a LEFT JOIN flights_new_time b ON ((a.ticket_no=b.ticket_no) AND (b.num=a.num-1)))
,
transitflights AS -- таблица промежуточных перелётов в билетах, в которых нас интересует аэропорт вылета и продолжительность, к которому сразу подбираем город
(SELECT a.city AS transit_city, fn.duration FROM flights_new fn LEFT JOIN airports a ON fn.departure_airport=a.airport_code WHERE (flightcount>1) AND (num>1))
-- в основном запросе считаем количество пересадок в транзитных городах
SELECT transit_city, avg (duration) AS avgduration FROM transitflights GROUP BY transit_city ORDER BY avgduration DESC -- группируем города и считаем среднюю продолжительность пересадки
;
-- 3.5*. Между какими городами(!) нет прямых рейсов? Запрос выводит список пар направлений в виде пар городов, между которыми нет прямых рейсов (9584)
WITH recursive
discity AS --пронумерованный и отсортированный список городов - 101 город
(SELECT ROW_NUMBER() OVER () AS num, city FROM (SELECT DISTINCT city FROM airports ORDER BY city) AS t),
st AS -- таблица пар НОМЕРОВ городов во всех возможных комбинациях - 101^2=10201 комбинация
(SELECT 1 AS i, CAST (1 AS BIGINT) AS k, CAST (1 AS BIGINT) AS t
UNION
SELECT i+1 AS i, i/(SELECT COUNT(city) FROM discity)+1 AS k,  i % (SELECT COUNT(city) FROM discity)+1 AS t FROM st
WHERE i<(SELECT COUNT(city) FROM discity)*(SELECT COUNT(city) FROM discity)),
allvectors AS --таблица пар НАЗВАНИЙ городов во всех возможных комбинациях  - 10201 комбинация минус 101 комбинация, где пара состоит из одинаковых городов, итого - 10100 пар
(SELECT b.city AS city1, c.city AS city2 FROM st a LEFT JOIN discity b ON a.k=b.num LEFT JOIN discity c ON a.t=c.num WHERE b.city<>c.city ORDER BY a.i),
flights_new AS --таблица всех пар городов, между которыми есть рейсы (направлений) - 516 направлений
(SELECT DISTINCT a.city AS departure_airport, a2.city AS arrival_airport FROM flights f
        LEFT JOIN airports a ON f.departure_airport=a.airport_code
        LEFT JOIN airports a2 ON f.arrival_airport=a2.airport_code)
--следующим запросом выбираем те пары городов, между которыми нет прямых рейсов - 10100 минус 516=9584
SELECT a.city1, a.city2 FROM allvectors a
      LEFT JOIN flights_new b ON ((a.city1=b.departure_airport) AND (a.city2=b.arrival_airport))
      WHERE b.departure_airport IS NULL ORDER BY a.city1, a.city2;

-- 4.1 - создание таблиц
CREATE TABLE seatsbooking -- таблица с выбранными пассажирами местами, для модуля продажи/смостоятельного выбора мест в самолёте
(
ticket_no CHAR(13), -- номер билета
flight_id INTEGER, -- номер полёта
CONSTRAINT seatbooking_key FOREIGN KEY (ticket_no, flight_id) REFERENCES ticket_flights(ticket_no, flight_id),
aircraft_code bpchar(3),
seat_no VARCHAR (4),
CONSTRAINT seatbooking_key2 FOREIGN KEY (aircraft_code, seat_no) REFERENCES seats(aircraft_code, seat_no),
PRIMARY KEY (ticket_no, flight_id)
);
COMMENT ON TABLE seatsbooking IS 'Забронированные места';

INSERT INTO seatsbooking-- заполнение данных о выбранных местах на основе уже совершённых рейсов
(ticket_no, flight_id, aircraft_code, seat_no)
SELECT a.ticket_no, a.flight_id, b.aircraft_code, c.seat_no FROM ticket_flights a
       LEFT JOIN flights b ON a.flight_id=b.flight_id LEFT JOIN boarding_passes c ON ((a.ticket_no=c.ticket_no) AND (a.flight_id=c.flight_id)) WHERE c.seat_no IS NOT NULL;

CREATE TABLE metar -- таблица с погодными кодами METAR
(
bID uuid PRIMARY KEY DEFAULT uuid_generate_v1(),
airport_code bpchar(3),
datetime bpchar (7),
wind bpchar (20),
visible bpchar (20),
phenom bpchar (20),
cloud bpchar (20),
temper bpchar (20),
press bpchar (20),
other1 bpchar (20),
forecast bpchar (20),
other2 bpchar (20),
FOREIGN KEY (airport_code) REFERENCES airports(airport_code)
);

COMMENT ON TABLE metar IS 'Коды METAR';
ALTER TABLE flights ADD DepartureMETAR uuid;
ALTER TABLE flights ADD ArrivalMETAR uuid;
ALTER TABLE flights ADD FOREIGN KEY (DepartureMETAR) REFERENCES metar(bID);
```

```sql
ALTER TABLE flights ADD FOREIGN KEY (ArrivalMETAR) REFERENCES metar(bID);

-- 5. запросы для анализа БД

-- 5.1. средняя задержка прибытия по рейсам
SELECT flight_no, avg (actual_arrival-scheduled_arrival) AS delay FROM flights
      WHERE actual_arrival IS NOT NULL GROUP BY flight_no ORDER BY delay DESC;

 -- 5.2. средняя задержка прибытия по городам отправления
 SELECT a.city, avg (f.actual_arrival-f.scheduled_arrival) AS delay FROM flights f
      LEFT JOIN airports a ON f.departure_airport=a.airport_code
      WHERE f.actual_arrival IS NOT NULL GROUP BY a.city ORDER BY delay DESC;

 -- 5.3. средняя задержка прибытия по аэропортам
 SELECT f.departure_airport, a.city, avg (f.actual_arrival-f.scheduled_arrival) AS delay FROM
flights f
      LEFT JOIN airports a ON f.departure_airport=a.airport_code
      WHERE f.actual_arrival IS NOT NULL GROUP BY f.departure_airport, a.city ORDER BY delay DESC;

 -- 5.4. средняя задержка прибытия по моделям самолётов с количеством полётов
 SELECT a.model, COUNT (flight_id) AS count_flights,avg (f.actual_arrival-f.scheduled_arrival) AS
delay FROM flights f
      LEFT JOIN aircrafts a ON f.aircraft_code=a.aircraft_code
      WHERE f.actual_arrival IS NOT NULL GROUP BY a.model ORDER BY delay DESC;

 -- 5.5. средняя задержка прибытия по количеству пассажиров
 SELECT CASE -- диапазоны количества пассажиров взяты таким образом, чтобы содержать близкое
количество перелётов
      WHEN passangers>=0 AND passangers<5 THEN '000-005'
      WHEN passangers>=5 AND passangers<10 THEN '005-010'
      WHEN passangers>=10 AND passangers<17 THEN '010-017'
      WHEN passangers>=17 AND passangers<25 THEN '017-025'
      WHEN passangers>=25 AND passangers<36 THEN '025-036'
      WHEN passangers>=36 AND passangers<50 THEN '036-050'
      WHEN passangers>=50 AND passangers<75 THEN '050-075'
      WHEN passangers>=75 AND passangers<100 THEN '075-100'
      WHEN passangers>=100 THEN '100+'  END
      passangerscount
      , COUNT (passangers) AS countflights, avg (f.actual_arrival-f.scheduled_arrival) AS delay
FROM flights f
      INNER JOIN (SELECT flight_id, COUNT(ticket_no) passangers FROM ticket_flights GROUP BY
flight_id) tf ON f.flight_id=tf.flight_id
      WHERE f.actual_arrival IS NOT NULL GROUP BY passangerscount ORDER BY delay DESC;

 -- 5.6. средняя задержка прибытия по дням недели
 SELECT (EXTRACT(ISODOW FROM scheduled_departure::TIMESTAMP))::INTEGER  AS weekday, avg
(actual_arrival-scheduled_arrival) AS delay FROM flights
      WHERE actual_arrival IS NOT NULL GROUP BY weekday ORDER BY delay DESC;


 -- 5.7. средняя задержка прибытия по часу суток
 SELECT (EXTRACT(HOUR FROM scheduled_departure::TIMESTAMP))::INTEGER  AS weekday, avg
(actual_arrival-scheduled_arrival) AS delay FROM flights
      WHERE actual_arrival IS NOT NULL GROUP BY weekday ORDER BY delay DESC;
```