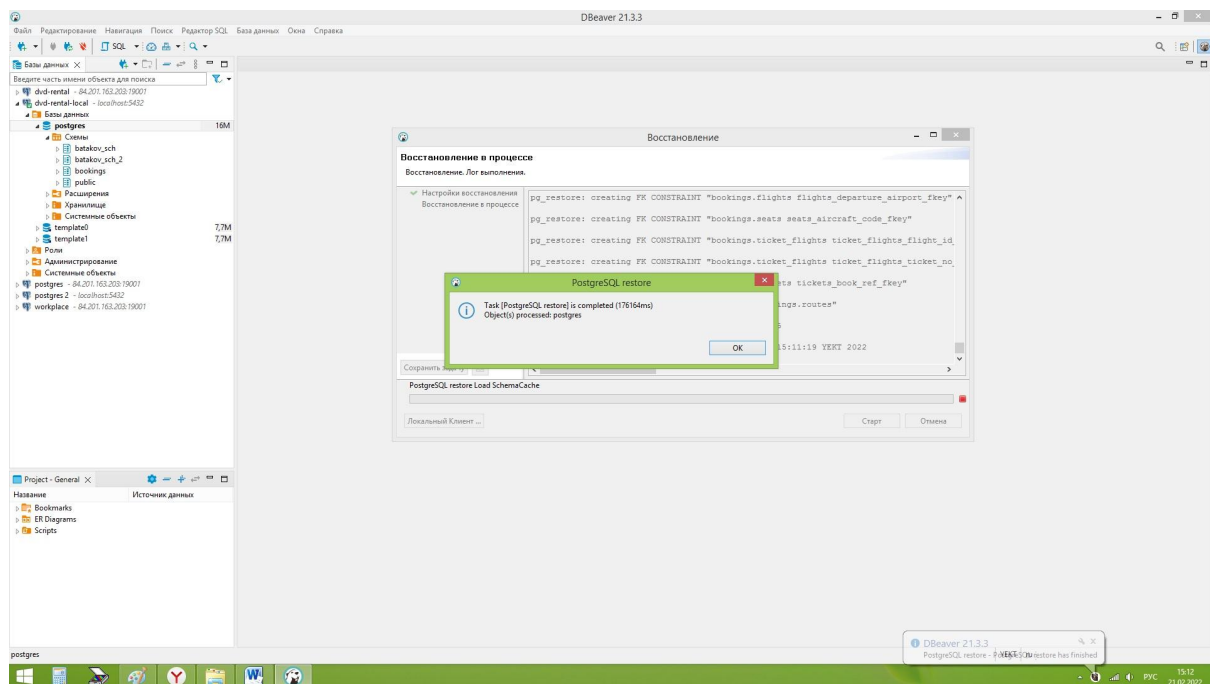


ИТОГОВАЯ РАБОТА
по модулю
“SQL и получение данных”

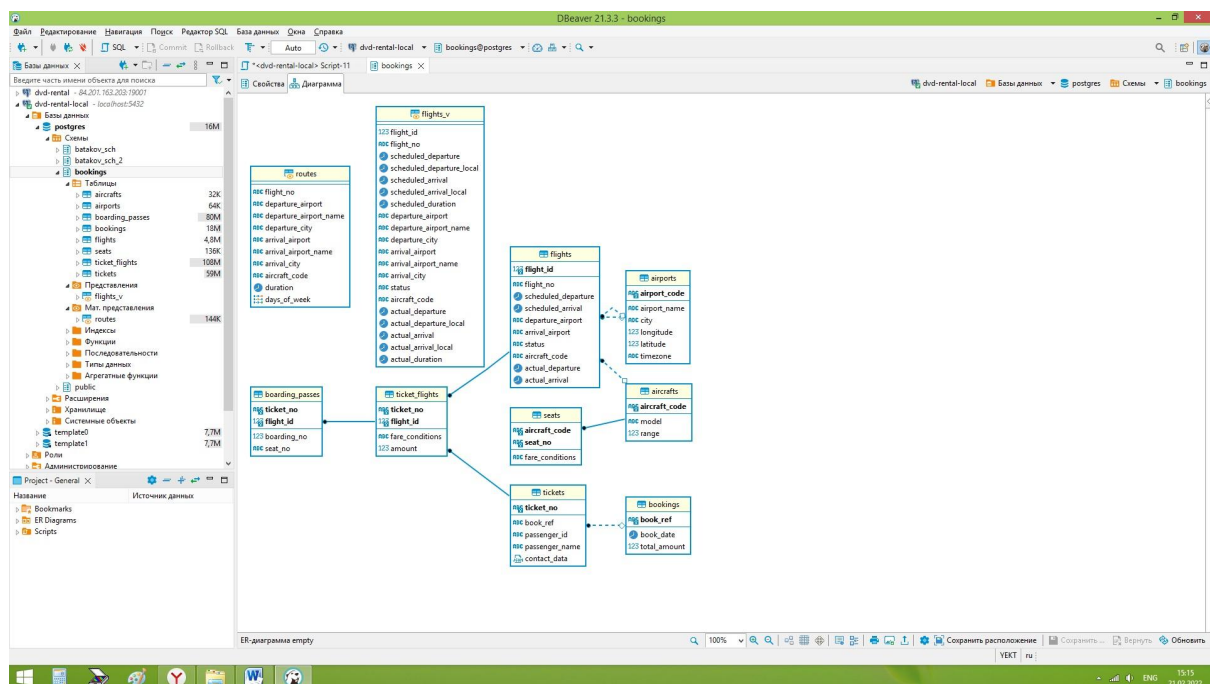
Михаил Батаков, DSU-21

г. Челябинск, 2022

1. В работе использовался локальный тип подключения. Скриншот прилагаю.



2. Скриншот ER-диаграммы прилагаю.



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

База данных состоит из 8 таблиц: aircrafts, airports, boarding_passes, bookings, flights, seats, ticket_flights, tickets, одного обычного представления flights_v и одного материализованного представления routes.

4. Развернутый анализ БД - описание таблиц, логики, связей и бизнес области. Основной сущностью является таблица 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. Как таковой пассажир не является отдельной сущностью. Как имя, так и номер документа пассажира могут меняться с течением времени, так что невозможно однозначно найти все билеты одного человека; для простоты можно считать, что все пассажиры уникальны.

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. Схема данных не контролирует, что места в посадочных талонах соответствуют имеющимся в самолете (такая проверка может быть сделана с использованием табличных триггеров или в приложении).

Таблица aircrafts.

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

Первичным ключом является aircraft_code, имеется ограничения внешнего ключа на таблицы flights и seats.

Ссылки из других таблиц TABLE "flights" FOREIGN KEY (aircraft_code) REFERENCES aircrafts(aircraft_code), TABLE "seats" FOREIGN KEY (aircraft_code) REFERENCES aircrafts(aircraft_code) ON DELETE CASCADE.

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).

Первичным ключом является airport_code.

Ссылки из других таблиц TABLE "flights" FOREIGN KEY (arrival_airport) REFERENCES airports(airport_code), TABLE "flights" FOREIGN KEY (departure_airport) REFERENCES airports(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).

Имеется составной первичный ключ ticket_no, flight_id, ограничение 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.

Первичный ключ book_ref, ссылки из других таблиц 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.

Первичный ключ flight_id, 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 из других таблиц 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.

Составной первичный ключ aircraft_code, seat_no, ограничение 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).

Имеется составной первичный ключ ticket_no, flight_id. Ограничения внешнего ключа FOREIGN KEY (flight_id) REFERENCES flights (flight_id), FOREIGN KEY (ticket_no) REFERENCES tickets (ticket_no).

Ссылка из других таблиц 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.

Первичный ключ ticket_no. Ограничения внешнего ключа FOREIGN KEY (book_ref) REFERENCES bookings(book_ref). Ссылка из внешних таблиц TABLE "ticket_flights" FOREIGN KEY (ticket_no) REFERENCES tickets (ticket_no).

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 **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.

Бизнес задачи, которые можно решить, используя БД:

- данная база данных позволяет проанализировать перелеты по направлениям на предмет заполняемости рейсов, можно сделать выводы о том, по каким направлениям возможно стоит добавить количество рейсов (где самолеты летают полные), а какие-то рейсы можно сократить (там где самолеты летают пустые);
- эту же информацию стоит использовать при анализе стоимости билетов, на более популярных направлениях можно рассмотреть возможность увеличения стоимости билетов, на менее популярных снизить ее или привнести какие-то дополнительные бонусы, плюшки, чтобы повысить спрос;
- можно проанализировать “системность” задержки вылетов по определенным направлениям - если есть системные задержки в одних и тех же направлениях, нужно проанализировать их причины и составить перечень мероприятий, направленных на минимизацию этих задержек;
- также по заполняемости самолетов и максимальной дальности их перелёта возможно проанализировать целесообразность применения тех или иных моделей на конкретных перелетах, возможно где-то используются самолеты с избыточными характеристиками по вместимости и дальности полета;
- также можно проанализировать загрузку и финансовый оборот в течение года, чтобы определить, какие месяцы являются наиболее прибыльными, а в какие наименее (если будут таковые закономерности), чтобы также можно было разработать какие-то мероприятия для стимулирования спроса в месяцы с малым финансовым оборотом.

5. Список SQL запросов с описанием логики их выполнения приведен в приложении 2.