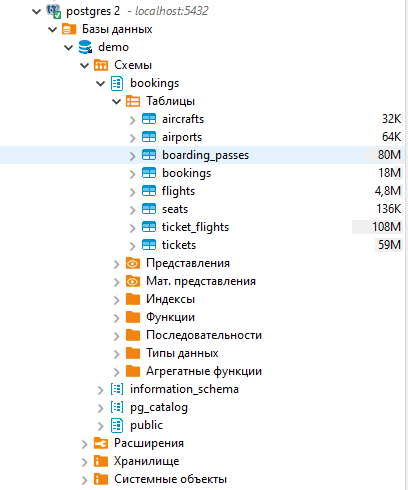
**Проектная работа по модулю**

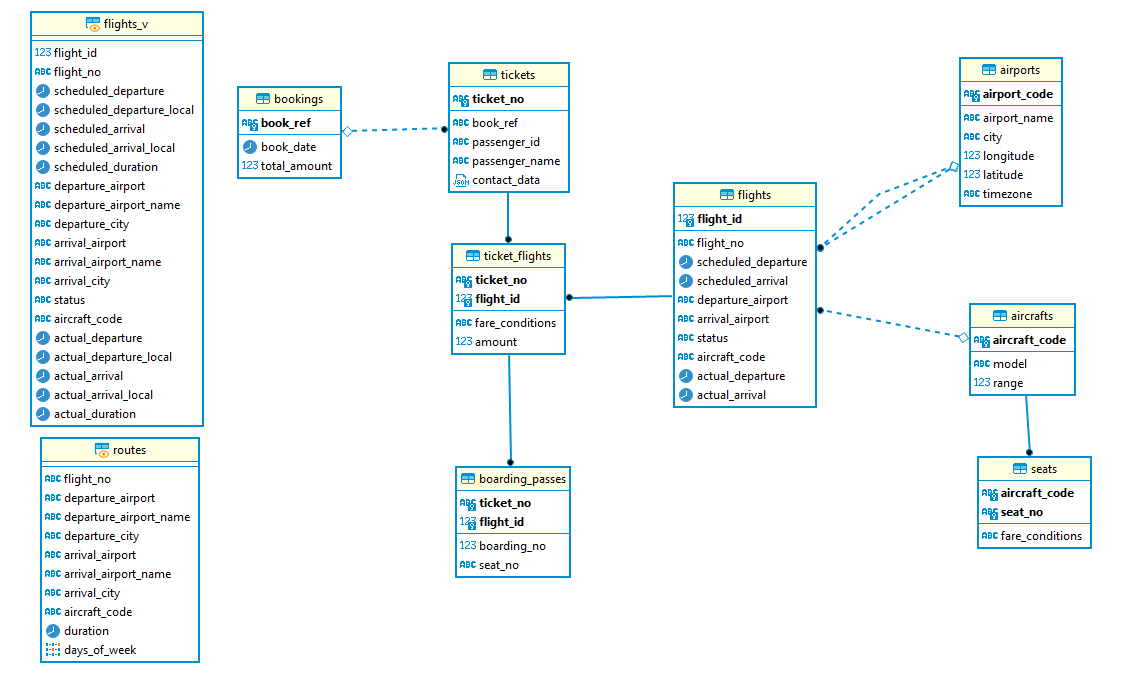
**“SQL и получение данных”**

(Группа SQL-36 Королев Валерий)

1. В работе использовался локальный тип подключения (восстановление из \*.backup файла).



1. Скриншот ER-диаграммы из DBeaver`a.



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

База данных состоит из следующих объектов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Описание |
| aircrafts  airports  boarding\_passes  bookings  flights  flights\_v  routes  seats  ticket\_flights  tickets | таблица  таблица  таблица  таблица  таблица  представление  мат. представление  таблица  таблица  таблица | Самолеты  Аэропорты  Посадочные талоны  Бронирования  Рейсы  Рейсы  Маршруты  Места  Перелеты  Билеты |

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

Бизнес – область базы данных – авиаперевозки по России.

Основной сущностью является бронирование (bookings). Бронь (book\_ref) может быть выписана на нескольких пассажиров, для каждого выписывается отдельный билет (ticket\_no), содержащий информацию о пассажире. В билете может быть несколько перелетов (flight\_id), если между аэропортами маршрута нет прямых рейсов, либо билет «туда/обратно».

Вся информация о перелетах содержится в таблице flights (номер рейса, аэропорт вылета/прилета, время вылета/прилета). Данные по аэропортам и городам их нахождения - таблица airports.

При регистрации на рейс пассажиру выдается посадочный талон (boarding\_no), в котором указывается место в самолете (seat\_no). Пассажир может зарегистрироваться только на тот рейс, который есть у него в билете.

Данные о самолетах, количестве мест и их классах обслуживания содержатся в таблицах aircrafts и seats.

4.1 aircrafts

* Код воздушного судна (ВС), модель ВС, максимальная дальность полёта (км)
* 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(aircraft\_code)

TABLE "seats" FOREIGN KEY (aircraft\_code) REFERENCES aircrafts(aircraft\_code) ON DELETE CASCADE

4.2 airports

* Код аэропорта, название аэропорта, город, координаты (долгота/широта), временная зона аэропорта
* 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).
* 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)

4.3 boarding\_passes

* Номер билета, id рейса, номер посадочного, номер места
* 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)

4.4 bookings

* Номер бронирования, дата бронирования, полная сумма бронирования
* 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)

4.5 flights

* id рейса, номер рейса, время вылета и прилета по расписанию, аэропорты отправления и прибытия, статус рейса, код ВС, фактическое время вылета и прилета
* 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(airport\_code)

FOREIGN KEY (departure\_airport) REFERENCES airports(airport\_code)

* Ссылки извне:

TABLE "ticket\_flights" FOREIGN KEY (flight\_id) REFERENCES flights(flight\_id)

4.6 flights\_v

* Представление: идентификатор рейса, номер рейса, время вылета по расписанию + местное, время прилета по расписанию + местное, планируемая продолжительность полета, код аэропорта отправления, название аэропорта отправления, город отправления, код аэропорта прибытия, название аэропорта прибытия, город прибытия, статус рейса, код самолета, фактическое время вылета + местное, фактическое время прилета + местное, фактическая продолжительность полета
* 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).

4.7 routes

* Материализованное представление: номер рейса, код аэропорта отправления, название аэропорта отправления, город отправления, код аэропорта прибытия, название аэропорта прибытия, город прибытия, код самолёта, продолжительность полета, дни недели, когда выполняется рейс
* 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.

4.8 seats

* код ВС, номер места, класс обслуживания
* 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

4.9 ticket\_flights

* номер билета, id рейса, класс обслуживания, стоимость перелета
* 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)

4.10 tickets

* номер билета, номер бронирования, id пассажира, ФИ пассажира, контактные данные пассажира
* 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)

Бизнес задачи, которые можно решить, используя БД.

Авиаперевозчик, как любая коммерческая организация заинтересован в рентабельности бизнеса и увеличении прибыли. Ключевыми факторами доходной части компании являются стоимость билетов и заполнение рейсов.

БД позволяет проанализировать заполняемость самолетов в зависимости от направлений полетов, сезонных изменений пассажиропотока. На заполняемость могут влиять дни недели. Исходя из этого перевозчик может принимать меры по увеличению заполняемости, а именно:

* Распределение летного парка по маршрутам, исходя из вместимости самолетов и пассажиропотока.
* Изменение расписания полетов (сезонное, увеличение рейсов в одних направлениях и снижение в других.
* Объединение рейсов
* Объединение пассажиропотоков в рамках альянсов авиакомпаний
* Скидки, если позволяет рентабельность.

Информация БД позволить планировать закупки новых самолетов, исходя из предполагаемых маршрутов их использования.

На более длинных временных интервалах можно прогнозировать (в комплексе с другими социально-экономическими, политическими и т.д. факторами) увеличение/снижение пассажиропотока на каких-то направлениях. Это позволит вовремя уйти с теряющих рентабельность маршрутов и подготовиться и увеличить присутствие на перспективных направлениях.

Безусловно надо использовать клиентскую базу пассажиров, хотя информации о них у компании не так много. На мой взгляд, можно сделать некий частотный анализ полетов пассажиров. Будет определенный поток пассажиров, часто использующих авиаперелеты. Например, очень много нижегородцев работают в Москве и на выходные приезжают в Нижний Новгород. Многие летают самолетом, потому что удобное расписание и приемлемая цена. Изменив расписание, можно потерять стабильный, постоянный поток пассажиров, который уйдет на железную дорогу.

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

Логика выполнения заданий содержится в практической части.