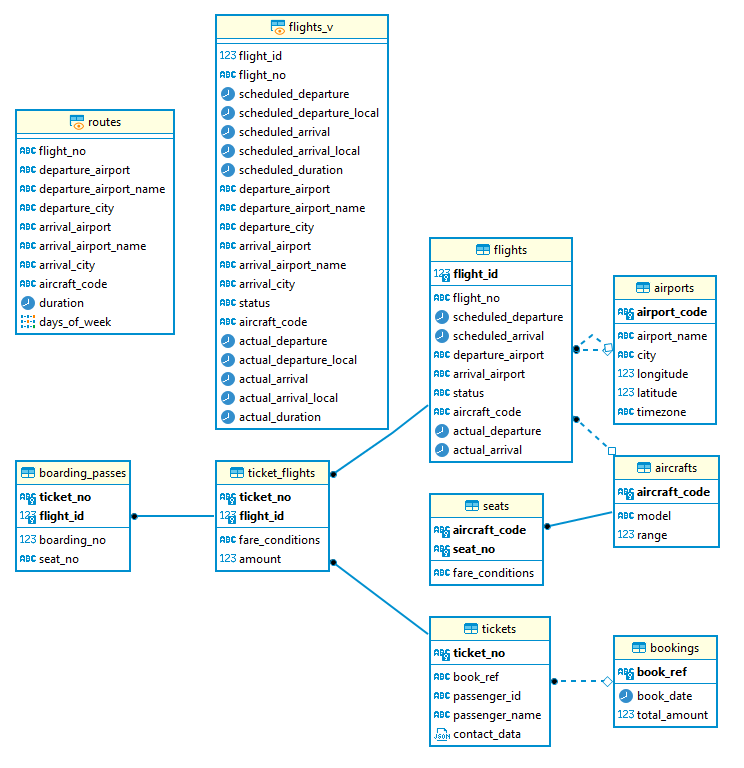
**Учебный SQL проект**

по работе с базой данных бронирования авиабилетов

**Описание данных:**

База данных состоит из 8-ми таблиц и 2-х представлений (именованный запрос и материализованное представление).

Схема содержит данные о пассажирских авиаперевозках по России, распределенные по объектам в виде таблиц с фактами и заданными признаками.



**Таблицы:**

1. **aircrafts**

* aircraft\_code - код воздушного судна
* model - модель воздушного судна
* range - максимальная дальность полета в км

1. **airports**

* airport\_code - код аэропорта
* airport\_name - имя аэропорта
* city - город расположения аэропорта
* longitude - широта
* latitude - долгота
* timezone - часовой пояс

1. **boarding\_passes**

* ticket\_no - номер билета
* flight\_id - идентификатор рейса
* boarding\_no - номер посадочного талона
* seat\_no - номер места

1. **bookings**

* book\_ref - номер бронирования
* book\_date - дата бронирования
* total amount - общая стоимость бронирования

1. **flights**

* flight\_id - идентификатор рейса
* flight\_no - номера рейса
* scheduled\_departure - дата отправления
* scheduled\_arrival - дата прибытия
* departure\_airport - аэропорт вылета
* arrival\_airport - аэропорт прибытия
* status - статус рейса
* aircraft\_code - код воздушного судна
* actual\_departure - реальное время вылета
* actual\_arrival - реальное время прибытия

1. **seats**

* aircraft\_code - код воздушного судна
* seat\_no - номер места
* fare\_conditions - класс обслуживания

1. **ticket\_flights**

* ticket\_no - номер билета
* flight\_id - идентификатор рейса
* fare\_conditions - класс обслуживания
* amount - стоимость перелета

1. **tickets**

* ticket\_no - номер билета
* book\_ref - номер бронирования
* passenger\_id - номер документа пассажира
* passenger\_name - фамилия и имя пассажира
* contact\_data - контактная информация пассажира

**Представления:**

1. **flights\_v** (запрос)

* flight\_id - Идентификатор рейса
* flight\_no - Номер рейса
* scheduled\_departure - Время вылета по расписанию
* scheduled\_departure\_local - Время вылета по расписанию,

  местное время в пункте отправления

* scheduled\_arrival - Время прилёта по расписанию
* scheduled\_arrival\_local - Время прилёта по расписанию,

  местное время в пункте прибытия

* scheduled\_duration - Планируемая продолжительность полета
* departure\_airport - Код аэропорта отправления
* departure\_airport\_name - Название аэропорта отправления
* departure\_city - Город отправления
* arrival\_airport - Код аэропорта прибытия
* arrival\_airport\_name - Название аэропорта прибытия
* arrival\_city - Город прибытия
* status - Статус рейса
* aircraft\_code - Код самолета, IATA
* actual\_departure - Фактическое время вылета
* actual\_departure\_local - Фактическое время вылета,

  местное время в пункте отправления

* actual\_arrival - Фактическое время прилёта
* actual\_arrival\_local - Фактическое время прилёта,

  местное время в пункте прибытия

* actual\_duration - Фактическая продолжительность полета

1. **routes**  (мат. представление)

* flight\_no - Номер рейса
* departure\_airport - Код аэропорта отправления
* departure\_airport\_name - Название аэропорта отправления
* departure\_city - Город отправления
* arrival\_airport - Код аэропорта прибытия
* arrival\_airport\_name - Название аэропорта прибытия
* arrival\_city - Город прибытия
* aircraft\_code - Код самолета, IATA
* duration - Продолжительность полета
* days\_of\_week - Дни недели, когда выполняются рейсы

**Развернутое описание БД:**

1. **aircrafts**

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

1. **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).

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)

1. **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)

1. **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)

1. **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(airport\_code)

FOREIGN KEY (departure\_airport)

REFERENCES airports(airport\_code)

Ссылки извне:

TABLE "ticket\_flights"

FOREIGN KEY (flight\_id)

REFERENCES flights(flight\_id)

1. **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

1. **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)

1. **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)

**Анализируя данные, представленные в Базе**

**можно решать следующие задачи:**

* Скорректировать текущее расписание, исходя из экономической целесообразности наличия рейсов в определенные даты;
* Учитывая загрузку определенных рейсов, заменить модель самолета на более (или менее) вместительный;
* Применять динамическое ценообразование, учитывая загрузку каждого рейса.

Например, увеличивать цену билета при заполнении воздушного судна на каждые 10%. Помимо получения дополнительной прибыли, данная мера также позволит быстрее привлекать средства и планировать загрузку задолго до отправления.

* Внедрение программы лояльности для часто летающих пассажиров.

Например, разделение маршрутов на сегменты и подсчет “бонусных миль” для каждого пассажира.

* Строить прогнозы и модели:
* Исходя из динамики пассажиропотока, планировать инвестиции в основные средства в виде воздушных судов определенной дальности и вместительности с оснащением их местами нужного класса.
* Прогнозировать загрузку бортперсонала и оснащенность рейсов товарами в зависимости от модели поведения пассажиров определенного пола и возраста, а также времени, дня недели и направления вылета. Например: питание, алкоголь и другие доп услуги на борту.

**Список SQL запросов**

Ниже приведена ссылка на файл со списком запросов.

Описание логики их выполнения дано в текстовом комментарии перед каждым запросом.

[SQL requests to airports DB](https://github.com/graff1te/Progects/blob/9372d36727b80e082210a3b53e630ee1da439d0c/SQL%20requests%20to%20airports%20DB.sql)