**ИТОГОВАЯ РАБОТА**

по курсу «SQL и получение данных»

Чекуновой Анны Сергеевны



1. В работе использовался облачный тип подключения базы данных.
2. Ниже представлена ER-диаграмма из DBeaver’а.

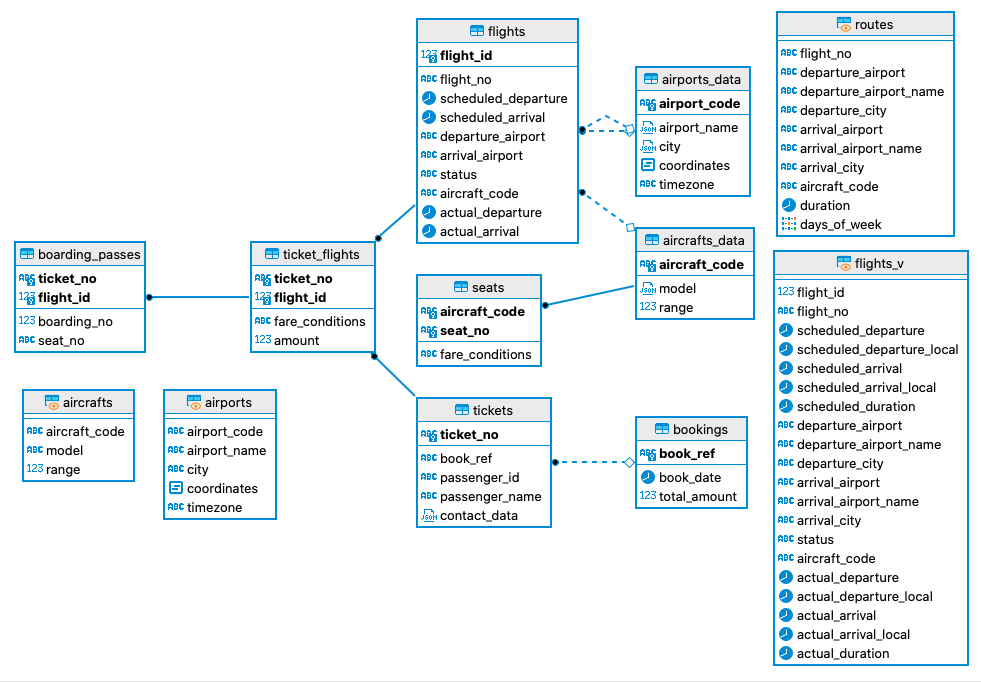


рис. 1  ER- диаграмма

Схема данных Bookings состоит из восьми таблиц и нескольких представлений. В качестве предметной области выбраны авиаперевозки по России.

1. В качестве предметной области выбраны авиаперевозки по России. Схема данных Bookings состоит из восьми таблиц и нескольких представлений.

3.1 Таблицы схемы данных Bookings:

* **boarding\_passes** - номер билета, id перелета, номер посадочного, номер места;
* **ticket\_flights** - номер билета, id перелета, класс обслуживания, стоимость перелета;
* **flights - id** перелета, номер рейса, время вылета по расписанию, время прилета по расписанию, аэропорт отправления, аэропорт прибытия, статус рейса, код самолета, фактическое время вылета, фактическое время прилета;
* **airports\_data** - код, название, город, координаты, часовой пояс аэропорта;
* **aircrafts\_data** - код, модель, максимальная дальность полета самолета;
* **seats** - код самолета, номер места, класс обслуживания;
* **bookings** - номер, дата, полная стоимость бронирования;
* **tickets** - номер билета, номер бронирования, имя и фамилия, контактные данные пассажира (e-mail, телефон);

3.2 Представления:

* routes - номер рейса, код и название аэропорта отправления, город вылета, код и название аэропорта прилета, город прилета, код самолета, продолжительность полета, дни недели, когда выполняется перелет;
* flights\_v - id перелета, номер рейса, время вылета по расписанию, местное время в городе вылета, время прилета по расписанию, местное время в городе прибытия, продолжительность перелета по расписанию, код и название аэропорта отправления, город вылета, код и название аэропорта прибытия, город прибытия, статус рейса, код самолета, фактическое время вылета, местное время в городе вылета, фактическое время прибытия, местное время в городе прибытия, фактическое время прибытия по расписанию;
* airports - копия airports\_data;
* aircrafts - копия aircrafts\_data.

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

4.1 **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)

4.2 **ticket\_flights**:

Перелет соединяет билет с рейсом и идентифицируется их номерами.

Для каждого перелета указываются его стоимость (amount) и класс обслуживания (fare\_conditions).

Ограничения-проверки:

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).

4.3 **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)

4.4 **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 (longitude), долгота (latitude) и часовой пояс (timezone).

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)

4.5 **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(aircraft\_code) TABLE "seats" FOREIGN KEY (aircraft\_code)

REFERENCES aircrafts(aircraft\_code) ON DELETE CASCADE

4.6 **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

4.7 **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)

4.8 **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).

**Бизнес задачи, которые можно решить, используя БД**

1. Можно оценить загруженность рейса исходя из размеров самолета. Если по направлению продаются все билеты на протяжении определенного периода времени, то можно рассмотреть замену борта на более вместительный.
2. Исходя из расчетного и фактического времени отправления и прибытия можно точно сказать в каком аэропорту происходят чаще всего задержки и на сколько.
3. Можно узнать самые популярные направления исходя из того по каким направлениям раньше всего бронируют билеты.
4. Приложение №1. Список SQL-запросов с описанием логики их выполнения. Exam\_SQL.sql