

Виктор Лоханько
15 октября 2021 г.
Нетология

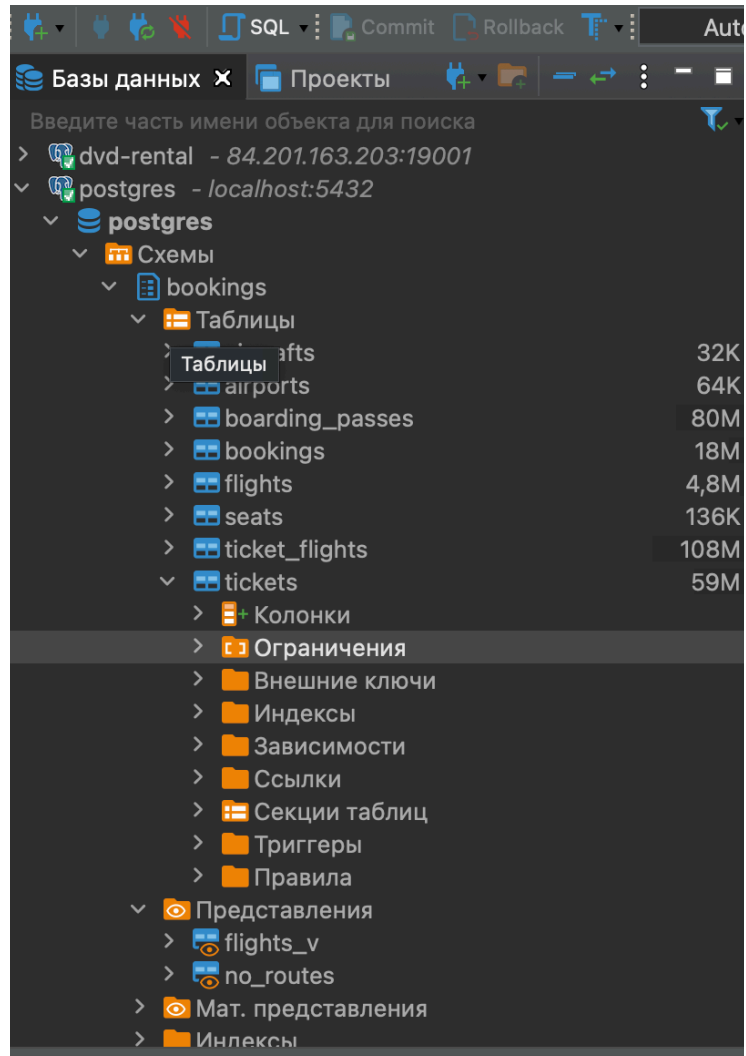
Проектная работа по модулю “SQL и получение данных”

Авиаперевозки

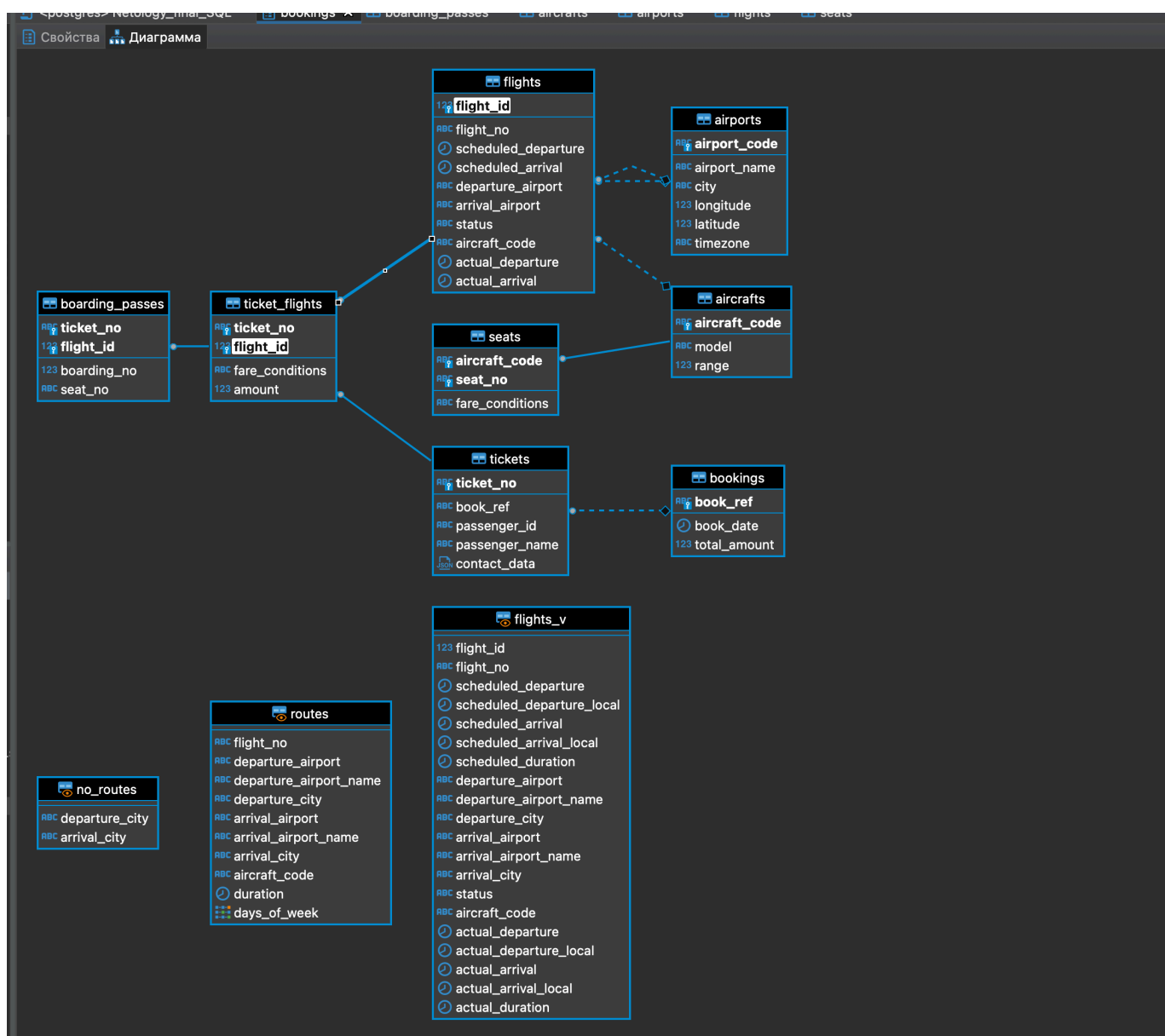


Итоговая работа

1. В работе использовался локальный тип подключения.



2. Скриншот ER-диаграммы из DBeaver'a согласно локального подключения



3. Краткое описание БД. Таблицы и представления

Таблицы

aircrafts: код самолета IATA, модель самолета, максимальная дальность полета, км

airports: код аэропорта, название аэропорта, город, координаты аэропорта: долгота, координаты аэропорта: широта, координаты аэропорта: широта

boarding_passes: номер билета, идентификатор рейса, номер посадочного талона, номер места

bookings: номер бронирования, дата бронирования, полная сумма бронирования

flights: идентификатор рейса, номер рейса, время вылета по расписанию, время прилёта по расписанию, аэропорт отправления, аэропорт прибытия, статус рейса, код самолета IATA, фактическое время вылета, фактическое время прилёта

seats: код самолета IATA, номер места, класс обслуживания

ticket_flights: номер билета, идентификатор рейса, класс обслуживания, стоимость перелета

tickets: номер билета, номер бронирования, идентификатор пассажира, имя пассажира, контактные данные пассажира

Представления

Представление flights_v: идентификатор рейса, номер рейса, время вылета по расписанию, время вылета по расписанию, местное время в пункте отправления, время прилёта по расписанию, время прилёта по расписанию, местное время в пункте прибытия, планируемая продолжительность полета, код аэропорта отправления, название аэропорта отправления, город отправления, код аэропорта прибытия, название аэропорта прибытия, город прибытия, статус рейса, код самолета IATA, фактическое время вылета, фактическое время вылета, местное время в пункте отправления, фактическое время прилёта, фактическое время прилёта, местное время в пункте прибытия, фактическая продолжительность полета

Материализованное представление bookings.routes: номер рейса, код аэропорта отправления, название аэропорта отправления, город отправления, код аэропорта прибытия, название аэропорта прибытия, город прибытия, код самолета IATA, продолжительность полета, дни недели, когда выполняются рейсы

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

Таблицы

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

bookings.aircrafts	
Самолеты	
ABC aircraft_code - Код самолета, IATA	bpchar(3) NOT NULL
ABC model - Модель самолета	text NOT NULL
123 range - Максимальная дальность полета, км	int4 NOT NULL

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*)



bookings.airports	
Аэропорты	
ABC airport_code - Код аэропорта	bpchar(3) NOT NULL
ABC airport_name - Название аэропорта	text NOT NULL
ABC city - Город	text NOT NULL
123 longitude - Координаты аэропорта: долгота	float8 NOT NULL
123 latitude - Координаты аэропорта: широта	float8 NOT NULL
ABC timezone - Временная зона аэропорта	text NOT NULL

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)

bookings.ticket_flights	
Перелеты	
ABC  ticket_no - Номер билета	bpchar(13) NOT NULL
123  flight_id - Идентификатор рейса	int4 NOT NULL
ABC fare_conditions - Класс обслуживания	varchar(10) NOT NULL
123 amount - Стоимость перелета	numeric(10, 2) NOT NULL

Ограничения внешнего ключа:



FOREIGN KEY (ticket_no, flight_id) REFERENCES ticket_flights(ticket_no, flight_id)

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)

bookings.bookings	
Бронирования	
ABC  book_ref - Номер бронирования	bpchar(6) NOT NULL
 book_date - Дата бронирования	timestampz NOT NULL
123 total_amount - Полная сумма бронирования	numeric(10, 2) NOT NULL

Ссылки извне:

TABLE "tickets" FOREIGN KEY (book_ref) REFERENCES bookings(book_ref)

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'))

bookings.flights Рейсы	
 flight_id - Идентификатор рейса	serial4 NOT NULL
 <i>flight_no</i> - Номер рейса	bpchar(6) NOT NULL
 <i>scheduled_departure</i> - Время вылета по расписанию	timestampz NOT NULL
 <i>scheduled_arrival</i> - Время прилёта по расписанию	timestampz NOT NULL
 <i>departure_airport</i> - Аэропорт отправления	bpchar(3) NOT NULL
 <i>arrival_airport</i> - Аэропорт прибытия	bpchar(3) NOT NULL
 <i>status</i> - Статус рейса	varchar(20) NOT NULL
 <i>aircraft_code</i> - Код самолета, IATA	bpchar(3) NOT NULL
 <i>actual_departure</i> - Фактическое время вылета	timestampz
 <i>actual_arrival</i> - Фактическое время прилёта	timestampz

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
```

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
```

bookings.seats Места	
ABC  aircraft_code - Код самолета, IATA	bpchar(3) NOT NULL
ABC  seat_no - Номер места	varchar(4) NOT NULL
ABC  fare_conditions - Класс обслуживания	varchar(10) NOT NULL

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'))
```

bookings.ticket_flights Перелеты	
ABC  ticket_no - Номер билета	bpchar(13) NOT NULL
123  flight_id - Идентификатор рейса	int4 NOT NULL
ABC  fare_conditions - Класс обслуживания	varchar(10) NOT NULL
123  amount - Стоимость перелета	numeric(10, 2) NOT NULL

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)
```

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_data*). 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)
```

bookings.tickets Билеты	
 ticket_no - Номер билета	bpchar(13) NOT NULL
 book_ref - Номер бронирования	bpchar(6) NOT NULL
 passenger_id - Идентификатор пассажира	varchar(20) NOT NULL
 passenger_name - Имя пассажира	text NOT NULL
 contact_data - Контактные данные пассажира	jsonb

Ограничения внешнего ключа:

```
FOREIGN KEY (book_ref) REFERENCES bookings(book_ref)
```

Ссылки извне:

```
TABLE "ticket_flights" FOREIGN KEY (ticket_no) REFERENCES tickets(ticket_no)
```

Представления

Представление **flights_v**












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*).

bookings.flights_v Рейсы		
123	flight_id - Идентификатор рейса	int4
ABC	flight_no - Номер рейса	bpchar(6)
🕒	scheduled_departure - Время вылета по расписанию	timestampz
🕒	scheduled_departure_local - Время вылета по расписанию, местное время в пункте отправления	timestamp
🕒	scheduled_arrival - Время прилёта по расписанию	timestampz
🕒	scheduled_arrival_local - Время прилёта по расписанию, местное время в пункте прибытия	timestamp
🕒	scheduled_duration - Планируемая продолжительность полета	interval
ABC	departure_airport - Код аэропорта отправления	bpchar(3)
ABC	departure_airport_name - Название аэропорта отправления	text
ABC	departure_city - Город отправления	text
ABC	arrival_airport - Код аэропорта прибытия	bpchar(3)
ABC	arrival_airport_name - Название аэропорта прибытия	text
ABC	arrival_city - Город прибытия	text
ABC	status - Статус рейса	varchar(20)
ABC	aircraft_code - Код самолета, IATA	bpchar(3)
🕒	actual_departure - Фактическое время вылета	timestampz
🕒	actual_departure_local - Фактическое время вылета, местное время в пункте отправления	timestamp
🕒	actual_arrival - Фактическое время прилёта	timestampz
🕒	actual_arrival_local - Фактическое время прилёта, местное время в пункте прибытия	timestamp
🕒	actual_duration - Фактическая продолжительность полета	interval

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. Именно такая информация и составляет материализованное представление routes.

 bookings.routes		
Маршруты		
 flight_no - Номер рейса		bpchar(6)
 departure_airport - Код аэропорта отправления		bpchar(3)
 departure_airport_name - Название аэропорта отправления		text
 departure_city - Город отправления		text
 arrival_airport - Код аэропорта прибытия		bpchar(3)
 arrival_airport_name - Название аэропорта прибытия		text
 arrival_city - Город прибытия		text
 aircraft_code - Код самолета, IATA		bpchar(3)
 duration - Продолжительность полета		interval
 days_of_week - Дни недели, когда выполняются рейсы		_int4

Какие бизнес-задачи можно решить, используя эти данные

1 Основные бизнес-метрики для авиакомпаний:

Вместимость - количество доступных мест (ASM). Количество мест рассчитывается путем умножения количества миль, которые пролетит данный самолет, на количество мест, доступных для данного рейса.

Трафик - Доходные пассажирские мили (RPM). Доходные пассажирские мили рассчитываются путем умножения количества заплативших пассажиров на пройденное расстояние.

Коэффициент загрузки - это процент, который отражает, насколько эффективно авиакомпания получает доход. Чтобы рассчитать коэффициент загрузки авиакомпании, необходимо разделить доходные пассажирские мили (RPM) на количество доступных мест (ASM).

Выручка - Выручка на доступную место (пустое или полное) на милю полета (RASM). Расчет дохода на доступное место-милю (RASM) представляет собой общую операционную выручку, деленную на количество доступных мест.

Стоимость - Стоимость за доступное место-милю (CASM)

Доход от пассажиров на одну милю доступного места и учитывает все места, включая пустые (PRASM)

Исходя из методологии расчета вышеуказанных метрик и доступных нам данных, мы можем посчитать только вместимость, трафик и коэффициент загрузки. Сделав дополнительные вычисления, получаем следующие данные (принимая во внимание, что у нас данные по полетам за 1 месяц):

Вместимость - количество доступных мест (ASM)- 1 145 118 307 место-миль

Трафик - Доходные пассажирские мили (RPM) - 579 254 762 пассажиро-миль

Коэффициент загрузки (RPM/ASM) - 0,51

Посмотрим какая в этот период была статистика по коэффициенту статистики на рынке. В открытом доступе есть данные Бюро транспортной статистики США. В рассматриваемый нами период (сентябрь- ноябрь 2016 года) в секторе внутренних пассажирских авиаперевозок коэффициент загрузки составил 0,85. (*Bureau of Transportation Statistics*). Это в 1,66 раза больше, чем в рассматриваемой нами авиакомпании.

Это говорит о том, что вероятнее всего, данная авиакомпания является убыточной на внутренних пассажирский авиаперевозках.

Что мы можем посоветовать для увеличения коэффициента загрузки :

1. Уменьшение периодичности вылетов на низкозагруженных направлениях
2. Увеличение периодичности вылетов на высокозагруженных направлениях
3. Анализ правильности времени вылета/прилета (утро, день, вечер)
4. Замена на самолеты меньшей вместительности (на высокозагруженных направлениях - на самолеты большей вместительности).
5. Увеличение цен, тарифов
6. На потенциально перспективных направлениях, которые могут «выстрелить»
-проведение маркетинговых мероприятий

Само собой, все эти изменения нужно проводить, проведя огромную аналитическую работу с анализом множества дополнительных данных.

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

Логика выполнения запросов прописана в виде комментариев в SQL запросе.