Университет Нетология

г.Москва

**Проектная работа по модулю   
«SQL и получение данных»**

Подготовлена студентом Павловым Н.Г.

Курс: DS-27

Апрель

2021

**Оглавление**

[Задание для описательной части проектной работы 3](#_Toc70669679)

[Описание 4](#_Toc70669680)

[Краткое описание БД (таблицы и представления) 5](#_Toc70669681)

[Развернутый анализ БД - описание таблиц, логики, связей и бизнес области 6](#_Toc70669682)

[Бизнес задачи, которые можно решить, используя БД 8](#_Toc70669683)

# Задание для описательной части проектной работы

1. В работе использовался \_\_\_\_ тип подключения.

* если база была развернута из \*.sql или \*.backup файла, необходимо приложить скриншот успешного импорта или восстановления

1. Скриншот ER-диаграммы из DBeaver`a согласно Вашего подключения.
2. Краткое описание БД - из каких таблиц и представлений состоит.
3. Развернутый анализ БД - описание таблиц, логики, связей и бизнес области (частично можно взять из описания базы данных, оформленной в виде анализа базы данных). Бизнес задачи, которые можно решить, используя БД.
4. Список SQL запросов из приложения №2 с описанием логики их выполнения. (см. sql – файл с комментариями)

# Описание

В работе использовался локальный тип подключения. База была развёрнута из или \*.backup файла размером 22 МБ (см. рис.1). ER-диаграмму согласно локальному подключению можно наблюдать на рисунке 2.

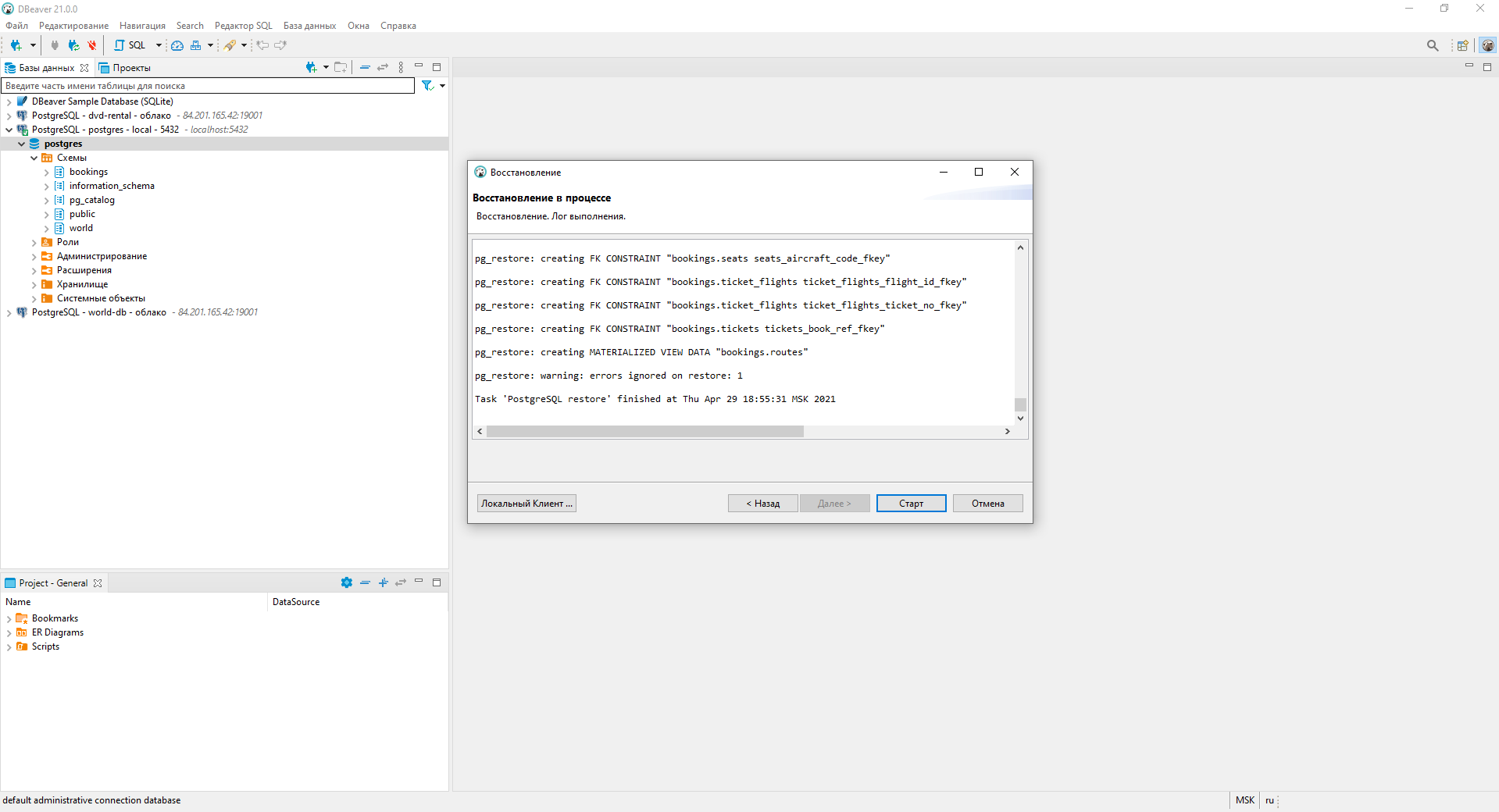


Рис. 1. Восстановление БД в DBeaver

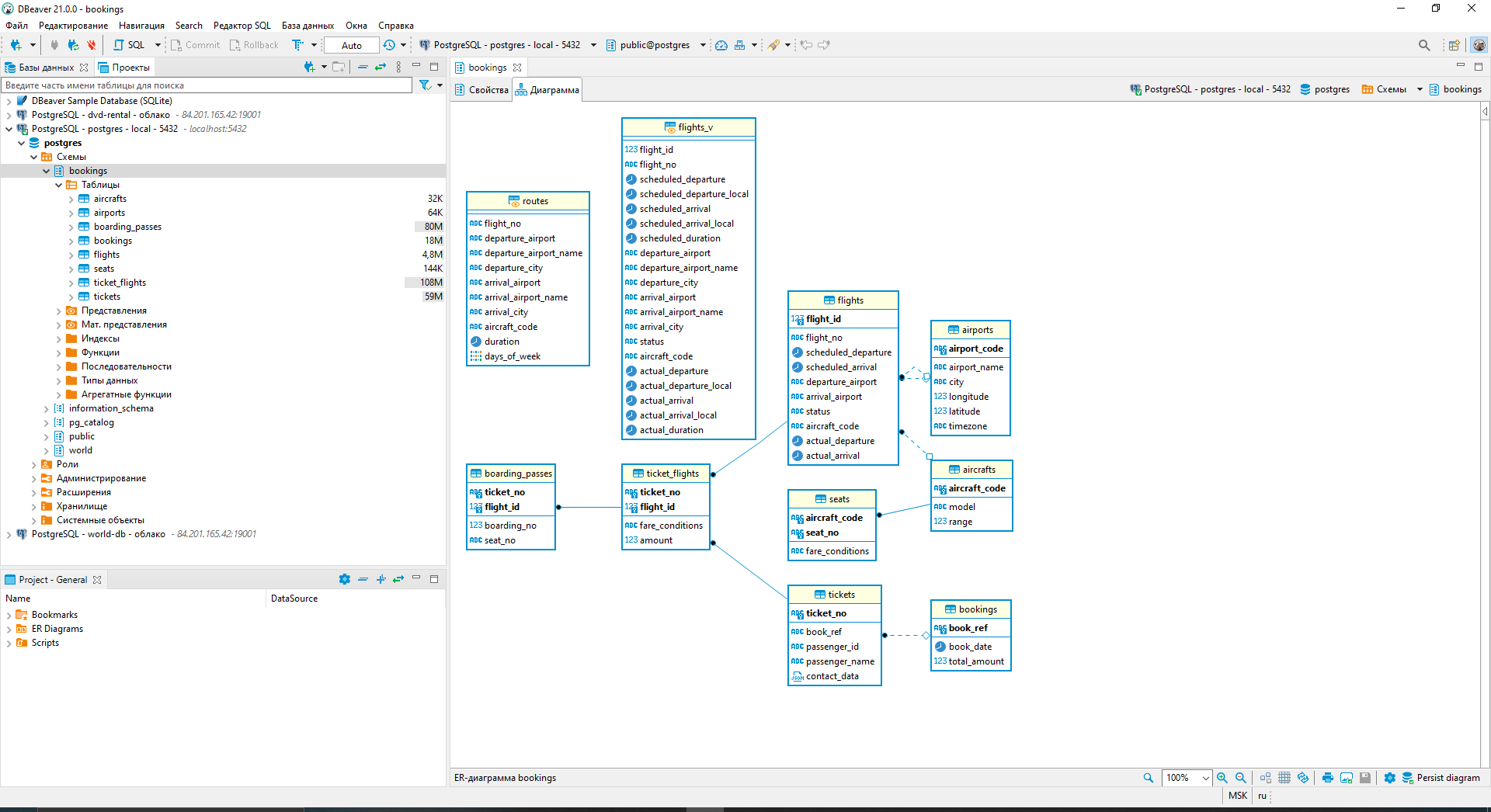


Рис. 2. ER диаграмма БД в DBeaver

## **Краткое описание БД (таблицы и представления)**

В данной работе была восстановлена СУБД PostgreSQL, которая состоит из 8 таблиц и 2-х представлений. Тематика данной базы – авиаперевозки по России.

База данных Bookings включает в себя следующие таблицы:

1. Aircrafts – самолеты
2. Airports – аэропорты
3. Boarding\_passes - посадочные талоны
4. Bookings – бронирования – основная сущность
5. Flights – рейсы
6. Seats – места
7. Ticket\_flights – перелёты
8. Tickets – билеты

Также включает представления:

1. Flights\_v – дополнительная информация для flights
2. Routes – материализованное представление – рейсы

## **Развернутый анализ БД - описание таблиц, логики, связей и бизнес области**

Таблица 1. Развёрнутый анализ БД

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Сущность** | **Столбцы** | **Описание** | **Ограничения** | **Индексы** | **Ссылки извне** |
| **Aircrafts**  **Отношение** | 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). | CHECK (range > 0) | PRIMARY KEY, btree (aircraft\_code) | TABLE "flights" FOREIGN KEY (aircraft\_code)  REFERENCES aircrafts(aircraft\_code)  TABLE "seats" FOREIGN KEY (aircraft\_code)  REFERENCES aircrafts(aircraft\_code) ON DELETE CASCADE |
| **Airports**  **Отношение** | 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). |  | PRIMARY KEY, btree (airport\_code) | TABLE "flights" FOREIGN KEY (arrival\_airport)  REFERENCES airports(airport\_code)  TABLE "flights" FOREIGN KEY (departure\_airport)  REFERENCES airports(airport\_code) |
| **Boarding\_passes**  **Отношение** | 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). | FOREIGN KEY (ticket\_no, flight\_id)  REFERENCES ticket\_flights(ticket\_no, flight\_id) | PRIMARY KEY, btree (ticket\_no, flight\_id)  UNIQUE CONSTRAINT, btree (flight\_id, boarding\_no)  UNIQUE CONSTRAINT, btree (flight\_id, seat\_no) |  |
| **Bookings**  **Отношение** | 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. |  | PRIMARY KEY, btree (book\_ref) | TABLE "tickets" FOREIGN KEY (book\_ref) REFERENCES bookings(book\_ref) |
| **Flights**  **Отношение** | 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. | 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'))  FOREIGN KEY (aircraft\_code) REFERENCES aircrafts(aircraft\_code) FOREIGN KEY (arrival\_airport) REFERENCES airports(airport\_code) FOREIGN KEY (departure\_airport) REFERENCES airports(airport\_code) | PRIMARY KEY, btree (flight\_id) UNIQUE CONSTRAINT, btree (flight\_no, scheduled\_departure) | TABLE "ticket\_flights" FOREIGN KEY (flight\_id) REFERENCES flights(flight\_id) |
| **Seats**  **Отношение** | 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. | CHECK (fare\_conditions IN ('Economy', 'Comfort', 'Business'))  FOREIGN KEY (aircraft\_code)  REFERENCES aircrafts(aircraft\_code) ON DELETE CASCADE | PRIMARY KEY, btree (aircraft\_code, seat\_no) |  |
| **Ticket\_flights**  **Отношение** | 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). | CHECK (amount >= 0)  CHECK (fare\_conditions IN ('Economy', 'Comfort', 'Business'))  FOREIGN KEY (flight\_id) REFERENCES flights(flight\_id)  FOREIGN KEY (ticket\_no) REFERENCES tickets(ticket\_no) | PRIMARY KEY, btree (ticket\_no, flight\_id) | TABLE "boarding\_passes" FOREIGN KEY (ticket\_no, flight\_id)  REFERENCES ticket\_flights(ticket\_no, flight\_id) |
| **Tickets**  **Отношение** | 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. | FOREIGN KEY (book\_ref) REFERENCES bookings(book\_ref) | PRIMARY KEY, btree (ticket\_no) | TABLE "ticket\_flights" FOREIGN KEY (ticket\_no) REFERENCES tickets(ticket\_no) |
| **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). |  |  |  |
| **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. |  |  |  |

# Бизнес задачи, которые можно решить, используя БД

1. Загрузка рейса в конкретный день – при помощи анализа величины, можно либо задержать рейс, чтобы вместить всех пассажиров на следующий рейс и не прожигать топливо, либо отменить рейс на конкретную дату.
2. Рентабельность рейса – если в течение длительного времени видно, что рейс не пользуется спросом, т.е. заполняемость салона мала, то для бизнеса это потеря денег. Следовательно, можно убрать этот рейс.
3. Запуск новых рейсов – при анализе пассажиропотока на определённом маршруте мы можем сделать вывод, в какие города и в какие сезоны требуются рейсы. В случае чего, запустить рейсы через самые «пассажирополные» остановки.
4. Загрузка аэропорта – наполняемость самолётами для определения посадочных мест, мест для парковки, для организации заправки и дополнительного обслуживания самолётов аэропортом.
5. Диспетчеризация самолётов – зная, количество рейсов, организацию их вылетов, можно безошибочно и безопасно организовать их отправления и прибытия.
6. Организация выдачи посадочных талонов – зная, сколько людей забронировали билеты, можно подготовить определённое количество людей/касс для выдачи посадочных талонов – это всё человекочасы/деньги.
7. Реагирование на внешние ситуации – зная, сколько самолётов находится в аэропорту и какой они дальности, можно в случае экстренных ситуаций организовать эвакуацию из определённых мест по поручению, или перенести людей с самолёта одной дальности на другой другой дальности. Опять же при поломке самолёта, возможно, реально заменить на другой.
8. Продумывание скидочной системы – к примеру, по наполняемости салона, или в несезон рейса.