

Ноябрь 2022г.

# ИТОГОВАЯ РАБОТА

курс «SQL и получение  
данных»



Белов Константин  
DAU-40

# Содержание

1. Используемый тип подключение, восстановление бэкапа базы данных.	3
2. ER-диаграмма базы данных.	3
3. Краткое описание базы данных - из каких таблиц и представлений состоит.	4
4. Развернутый анализ базы данных.	5
4.1 Описание таблиц, логики, связей и бизнес области.	5
4.2 Бизнес задачи, которые можно решить, используя базу данных.	10
5. Список SQL запросов с описанием логики их выполнения.	10

## 1. Используемый тип подключения, восстановление бэкапа базы данных.

В работе использовался локальный тип подключения:

Рис. 1 Восстановление базы из backup - файла

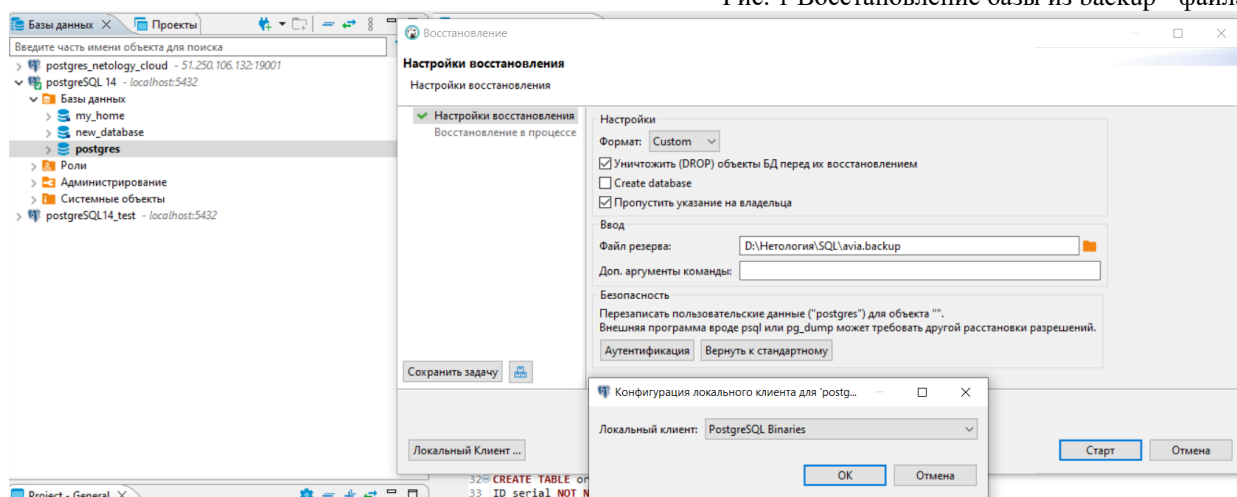
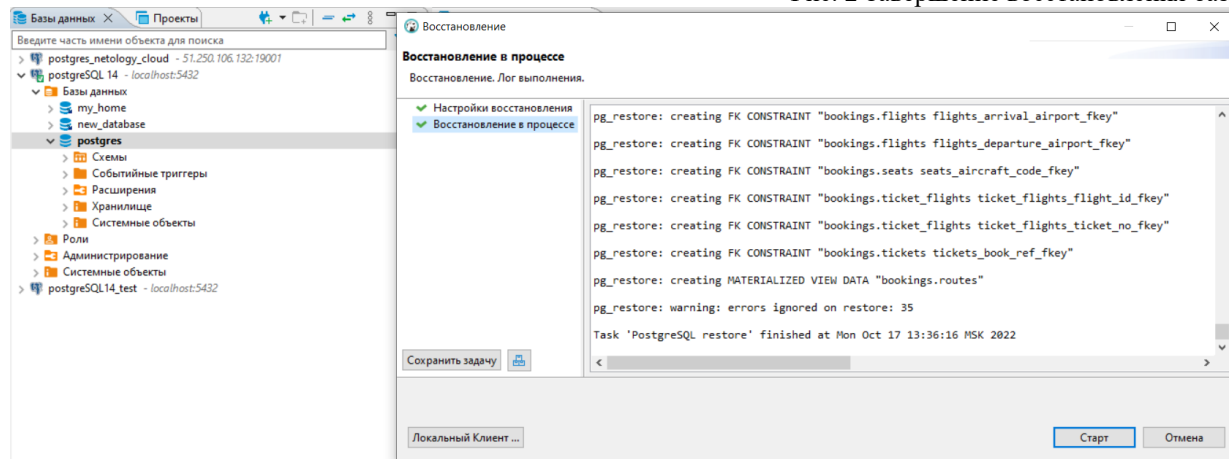
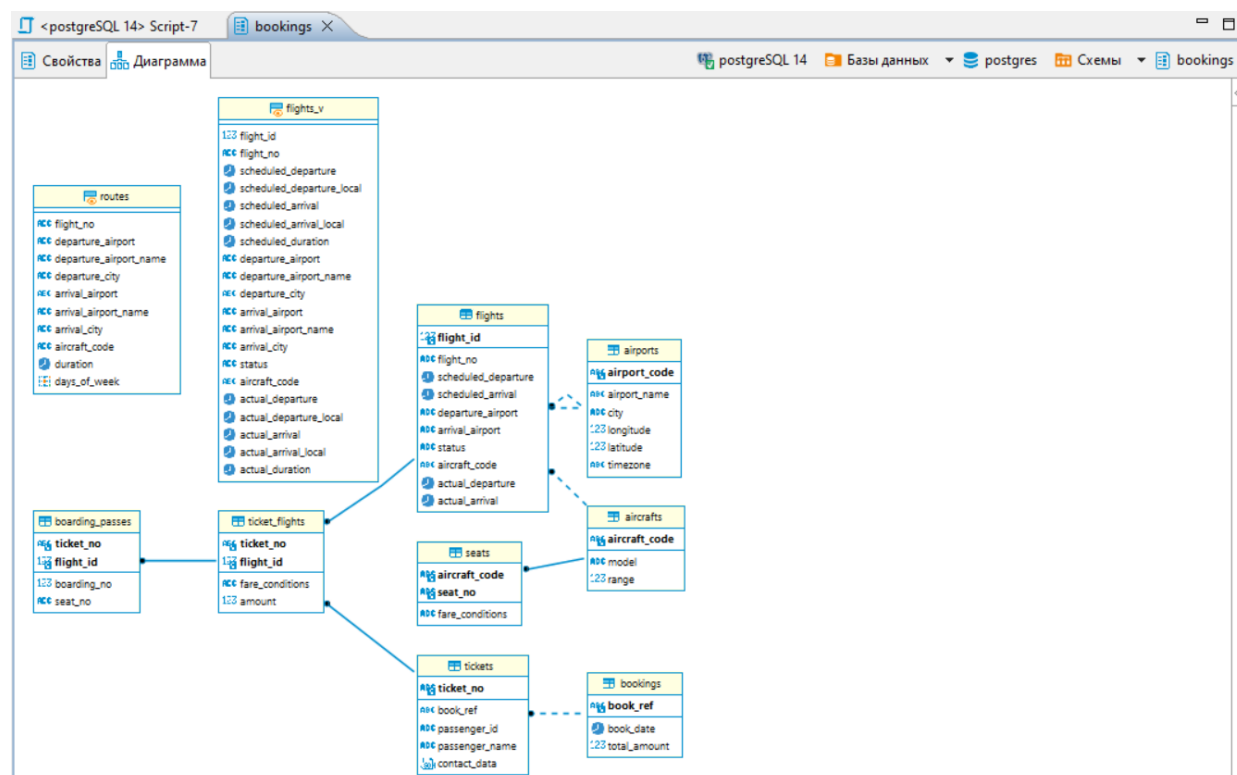


Рис. 2 Завершение восстановления базы



## 2. ER-диаграмма базы данных.



### 3. Краткое описание базы данных - из каких таблиц и представлений состоит.

База данных состоит из следующих таблиц:

- *Aircrafts*

Поля:

aircraft\_code - Код самолета, IATA – первичный ключ

model – Модель самолета

range – Максимальная дальность полета, км

- *Airports*

Поля:

airport\_code – Код аэропорта – первичный ключ

airport\_name – Название аэропорта

city – Город, рядом с которым находится аэропорт

longitude – Координаты аэропорта: долгота

latitude – Координаты аэропорта: широта

timezone – Временная зона аэропорта

- *Boarding\_passes*

Поля:

ticket\_no - Номер билета

flight\_id - Идентификатор рейса

*ticket\_no и flight\_id образуют составной первичный ключ данной таблицы*

boarding\_no – номер посадочного талона

seat\_no – номер места

- *Bookings*

Поля:

book\_ref – Номер бронирования – первичный ключ

book\_date – Дата бронирования

total\_amount – Полная сумма бронирования

- *Flights*

Поля:

flight\_id - Идентификатор рейса – первичный ключ

flight\_no – Номер рейса

scheduled\_departure – Время вылета по расписанию

scheduled\_arrival – Время прилета по расписанию

departure\_airport – Аэропорт отправления – внешний ключ к таблице Airports

arrival\_airport – Аэропорт прибытия - внешний ключ к таблице Airports

status - Статус рейса

aircraft\_code - Код самолета, IATA – внешний ключ к таблице Aircrafts

actual\_departure - Фактическое время вылета

actual\_arrival - Фактическое время прилёта

- *Seats*

Поля:

aircraft\_code - Код самолета, IATA – первичный ключ, одновременно является внешним ключем к таблице Aircrafts

seat\_no – Номер места

fare\_conditions – Класс обслуживания

- *Ticket\_flights*

Поля:

ticket\_no – Номер билета

flight\_id - Идентификатор рейса

*ticket\_no и flight\_id образуют составной первичный ключ данной таблицы. При этом они являются внешним ключом к таблице Boarding\_passes. Столбец ticket\_no является внешним ключом к таблице Tickets.*

fare\_conditions – Класс обслуживания  
amount – Стоимость перелета

- *Tickets*

Поля:

ticket\_no – Номер билета – первичный ключ  
book\_ref – Номер бронирования – внешний ключ к таблице Bookings  
passenger\_id – Идентификатор пассажира  
passenger\_name – Имя пассажира  
contact\_data – Контактные данные пассажира

Так же в базе данных есть следующие представления:

- *Flights\_v* – нематериализованное представление

Поля:

flight\_id - 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 полет

- *Routes* - материализованное представление

Поля:

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

## 4. Развернутый анализ базы данных.

### 4.1 Описание таблиц, логики, связей и бизнес области.

#### 4.1.1 Таблица bookings.aircrafts

Каждая модель воздушного судна идентифицируется своим трехзначным кодом (aircraft\_code). Указывается также название модели (model) и максимальная дальность полета в километрах (range).

Столбец	Тип данных	Модификаторы	Описание
aircraft_code	char(3)	NOT NULL	Код самолета, IATA
model	text	NOT NULL	Модель самолета
range	integer	NOT NULL	Максимальная дальность полета, км

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
```

#### 4.1.2 Таблица *bookings.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).

Столбец	Тип данных	Модификаторы	Описание
airport_code	char(3)	NOT NULL	Код аэропорта
airport_name	text	NOT NULL	Название аэропорта
city	integer	NOT NULL	Город
longitude	float	NOT NULL	Координаты аэропорта: долгота
latitude	float	NOT NULL	Координаты аэропорта: широта
timezone	text	NOT NULL	Временная зона аэропорта

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)
```

#### 4.1.3 Таблица *bookings.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).

Столбец	Тип данных	Модификаторы	Описание
ticket_no	char(13)	NOT NULL	Номер билета
flight_id	integer	NOT NULL	Идентификатор рейса
boarding_no	integer	NOT NULL	Номер посадочного талона
seat_no	varchar(4)	NOT NULL	Номер места

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)
```

#### 4.1.4 Таблица *bookings.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.

Столбец	Тип данных	Модификаторы	Описание
book_ref	char(6)	NOT NULL	Номер бронирования

book_date	timestampz	NOT NULL	Дата бронирования
total_amount	numeric(10,2)	NOT NULL	Полная сумма бронирования

Индексы:

PRIMARY KEY, btree (book\_ref)

Ссылки извне:

TABLE "tickets" FOREIGN KEY (book\_ref) REFERENCES bookings(book\_ref)

#### 4.1.5 Таблица bookings.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.

Столбец	Тип данных	Модификаторы	Описание
flight_id	serial	NOT NULL	Идентификатор рейса
flight_no	char(6)	NOT NULL	Номер рейса
scheduled_departure	timestampz	NOT NULL	Время вылета по расписанию
scheduled_arrival	timestampz	NOT NULL	Время прилёта по расписанию
departure_airport	char(3)	NOT NULL	Аэропорт отправления
arrival_airport	char(3)	NOT NULL	Аэропорт прибытия
status	varchar(20)	NOT NULL	Статус рейса
aircraft_code	char(3)	NOT NULL	Код самолета, IATA
actual_departure	timestampz		Фактическое время вылета
actual_arrival	timestampz		Фактическое время прилёта

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)

```

#### 4.1.6 Таблица bookings.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.

Столбец	Тип данных	Модификаторы	Описание
aircraft_code	char(3)	NOT NULL	Код самолета, IATA
seat_no	varchar(4)	NOT NULL	Номер места
fare_conditions	varchar(10)	NOT NULL	Класс обслуживания

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

```

#### 4.1.7 Таблица bookings.ticket\_flights

Перелет соединяет билет с рейсом и идентифицируется их номерами. Для каждого перелета указываются его стоимость (amount) и класс обслуживания (fare\_conditions).

Столбец	Тип данных	Модификаторы	Описание
ticket_no	char(13)	NOT NULL	Номер билета
flight_id	integer	NOT NULL	Идентификатор рейса
fare_conditions	varchar(10)	NOT NULL	Класс обслуживания
amount	numeric(10,2)	NOT NULL	Стоимость перелета

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)

```

#### 4.1.8 Таблица bookings.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.

Столбец	Тип данных	Модификаторы	Описание
ticket_no	char(13)	NOT NULL	Номер билета
book_ref	char(6)	NOT NULL	Номер бронирования



passenger_id	varchar(20)	NOT NULL	Идентификатор пассажира
passenger_name	text	NOT NULL	Имя пассажира
contact_data	jsonb		Контактные данные пассажира

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)

#### 4.1.9 Представление "bookings.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).

Столбец	Тип данных	Описание
flight_id	integer	Идентификатор рейса
flight_no	char(6)	Номер рейса
scheduled_departure	timestamptz	Время вылета по расписанию
scheduled_departure_local	timestamp	Время вылета по расписанию, местное время в пункте отправления
scheduled_arrival	timestamptz	Время прилёта по расписанию
scheduled_arrival_local	timestamp	Время прилёта по расписанию, местное время в пункте прибытия
scheduled_duration	interval	Планируемая продолжительность полета
departure_airport	char(3)	Код аэропорта отправления
departure_airport_name	text	Название аэропорта отправления
departure_city	text	Город отправления
arrival_airport	char(3)	Код аэропорта прибытия
arrival_airport_name	text	Название аэропорта прибытия
arrival_city	text	Город прибытия
status	varchar(20)	Статус рейса
aircraft_code	char(3)	Код самолета, IATA
actual_departure	timestamptz	Фактическое время вылета
actual_departure_local	timestamp	Фактическое время вылета, местное время в пункте отправления
actual_arrival	timestamptz	Фактическое время прилёта, местное время в пункте прибытия
actual_duration	interval	Фактическая продолжительность полета

#### 4.1.10. 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.

Столбец	Тип данных	Описание
flight_no	char(6)	Номер рейса
departure_airport	char(3)	Код аэропорта отправления
departure_airport_name	text	Название аэропорта отправления
departure_city	text	Город отправления
arrival_airport	char(3)	Код аэропорта прибытия
arrival_airport_name	text	Название аэропорта прибытия
arrival_city	text	Город прибытия
aircraft_code	char(3)	Код самолета, IATA
duration	interval	Продолжительность полета
days_of_week	integer	Дни недели, когда выполняются рейсы

## 4.2. Бизнес задачи, которые можно решить, используя базу данных.

4.2.1 Проведя анализ заполняемости рейсов по каждому из направлений можно решить следующие задачи:

- Рейсы, с маленькой заполняемостью можно совершать реже, дабы увеличить заполняемость, либо вовсе сделать чартерными.
- При малой заполняемости рейса можно изменить тип самолета с большей загрузкой на тип самолета с меньшей загрузкой, тем самым повысить рентабельность рейса.
- Анализ заполняемости рейсов в зависимости от сезона так же позволит более эффективно управлять парком самолетов – перебрасывать свободные борты на популярные направления, а также планировать график прохождения капитального технического обслуживания бортов, не выводя их из эксплуатации в высокий сезон.
- Для руководства аэропорта: анализ сезонности перелетов так же позволит получить информацию о загруженности аэропортов, а значит, позволит лучше спланировать график проведения ремонта, технического обслуживания и модернизации наземного оборудования, поддерживающего функциональность аэропорта.
- Корректировать стоимость билетов в зависимости от популярности маршрута и % заполняемости рейса.

4.2.2. Данные о количестве отмененных броней могут служить материалом для дополнительного исследования о причинах отмены брони, и как следствие, об эффективности службы бронирования.

4.2.3. На основании данных о задержках рейсов можно провести анализ эффективности работы как сотрудников авиакомпании, так и наземных служб, и инфраструктуры аэропорта. Возможно, причина задержек связана с недостаточным количеством взлетных полос, и для нормального функционирования аэропорта необходима его модернизация.

4.2.4. По каждой паре городов, между которыми нет прямых рейсов можно узнать информацию о количестве пассажиров, которые в данный момент совершают перелет из одного города в другой с пересадками. Если количество таких пассажиров велико – имеет смысл организовать прямой рейс между двумя городами.

4.2.5. Данные о пассажирах, воспользовавшихся услугами авиакомпании позволяют реализовать следующие мероприятия:

- Отследить сезонные перемещения пассажира и на основании этих данных организовать отправку сообщений о доступности рейсов в зависимости от сезона. Это возможно реализовать, даже при условии непостоянности идентификатора пассажира. Идентифицировать его можно по контактными данным (при смене фамилии контактный телефон и электронная почта может не измениться)
- Разработать программу лояльности, предоставляя скидки и преференции постоянным клиентам.
- Как следствие реализации предыдущего пункта: в случае, если места эконом-класса на рейс выкуплены, а места в бизнес-классе свободны, можно предложить пассажиру эконом-класса, который является постоянным клиентом, пересесть в бизнес-класс бесплатно, либо за символическую плату. Освободившееся место в эконом-классе мы можем продать другому пассажиру, повысив тем самым заполняемость рейса.

## 5. Список SQL запросов с описанием логики их выполнения.

Данный список представлен в отдельном sql-файле, отправленным вместе с данной Итоговой работой.