Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский университет   
«Высшая школа экономики»

###### Факультет бизнеса и менеджмента, ОП Бизнес-Информатика

###### Проект

На тему: Моделирование БД Аэропорта

Студента группы № 1802

Гончарова Степана Александровича

Преподаватель

Гоменюк Кирилл Сергеевич

Москва, 2019 год

Оглавление

[Введение: 3](#_Toc29150909)

[Проектирование 3](#_Toc29150910)

[Описание предметной области: 3](#_Toc29150911)

[Цель создания базы данных: 3](#_Toc29150912)

[Возможные пользователи базы данных: 3](#_Toc29150913)

[Возможные сценарии взаимодействия пользователя с БД: 4](#_Toc29150914)

[Планируемый перечень отчетных форм: 4](#_Toc29150915)

[Возможная архитектура ПО на основе разрабатываемой БД: 4](#_Toc29150916)

[Моделирование 5](#_Toc29150917)

[Определение сущностей: 6](#_Toc29150918)

[Модели: 7](#_Toc29150919)

[Логическая модель ErWin: 7](#_Toc29150921)

[Физическая модель ErWin: 7](#_Toc29150922)

[Соответствует ли итоговая схема 4НФ? 7](#_Toc29150923)

[Демонстрация работоспособности индексов: 8](#_Toc29150924)

[Разработка SQL-запросов, функций, триггеров, процедур, представлений 9](#_Toc29150925)

[SQL-запросы, функции, триггеры, процедуры, представления: 9](#_Toc29150926)

[Реализуемые бизнес-функции: 14](#_Toc29150927)

[Алгоритмы обработки данных: 14](#_Toc29150928)

[Отчетность 14](#_Toc29150929)

[Отчет SSRS: 14](#_Toc29150930)

[Отчет Excel: 15](#_Toc29150931)

# 

# Введение:

В информационную эпоху каждый человек ежедневно сталкивается с огромными объемами информации и технологических решений, которые жизненно необходимы ему для решения не только профессиональных, но и повседневных задач. На фоне высокого спроса цифровых решений как никогда остро встает вопрос рационального хранения и эффективного управления данными. Данные задачи особенно актуальны для бизнеса, заинтересованного в оптимизации своих процессов, повышая качество обслуживания клиентов. В данной же работе будет произведен процесс проектирования и разработки базы данных объекта, без которого невозможно представить современный мир и быстрое перемещение на дальние дистанции, а именно аэропорта.

# Проектирование

## Описание предметной области:

Аэропорт — комплекс сооружений, предназначенный для приёма, отправки, базирования воздушных судов и обслуживания воздушных перевозок, имеющий для этих целей аэродром и другие наземные сооружения, а также необходимое оборудование.

В данной работе будет рассматриваться область функционирования аэропорта, связанная с организацией воздушного перевоза пассажиров, и их обслуживания.

## Цель создания базы данных:

Целью работы является проектирование базы данных аэропорта, обеспечивающей хранение информации об авиаперелетах, а также пассажирах, пользующихся услугами перелетов в интересах бизнеса:

- обработка данных с целью извлечения конкретных знаний

- накопление данных в рамках предметной области

- составление статистических моделей

- построение рекомендательных систем

## Возможные пользователи базы данных:

Потенциальными пользователями базы данных являются сотрудники компаний, работающих в границах нашей предметной области, занимающиеся обработкой, обновлением и актуализацией данных, находящихся в границах предметной области и релевантных для спроектированной БД, а так же топ-менеджеры и администраторы БД, взаимодействующие по различным сценариям с БД напрямую, либо подтверждающие перенос информации из клона в реальную БД.

## 

## Возможные сценарии взаимодействия пользователя с БД:

1. Извлечение знаний, содержащихся в данных:

Получение информации о самолетном парке конкретной авиакомпании

Получение информации о странах, в которые запрещены авиаперелеты

Выявление наиболее популярного направления (по числу перелетов в конкретную страну)

Выявление количества авиаперелетов личностей

Выявление предпочитаемого класс личностей (по наибольшему числу авиаперелетов в том или ином классе)

1. Обновление данных:

Обновление самолетного парка авиакомпании

Добавление новых членов экипажа

Обновление информации о перелетах пользователей

1. Актуализация данных:

Изменение летных параметров самолетов

Изменение контактных данных организаций и пользователей

1. Администрирование БД:

Изменение основной БД и клона

Добавление уровней доступа к БД

## Планируемый перечень отчетных форм:

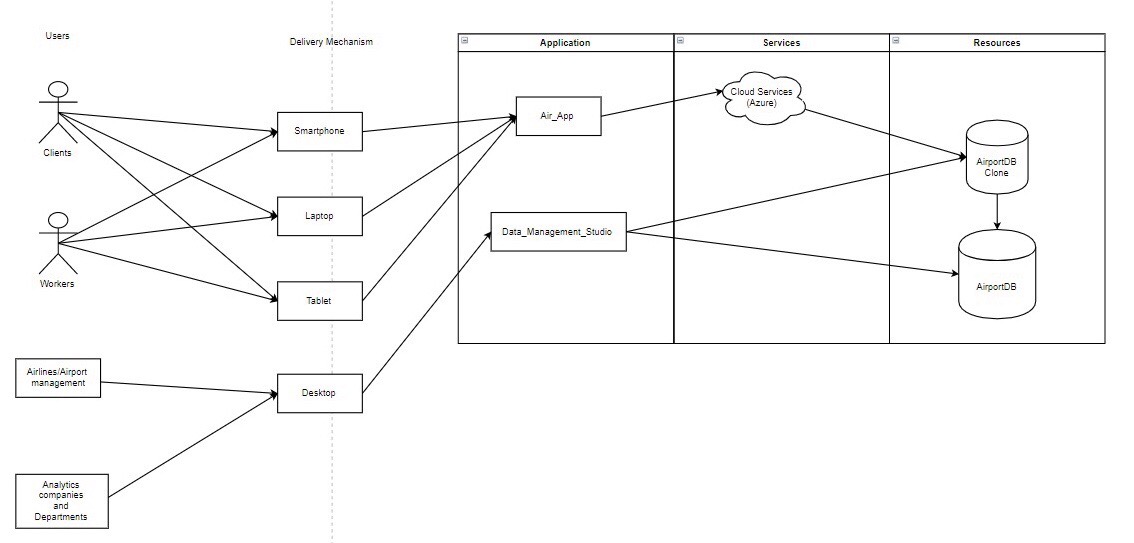
- Отчет SSRS

- Отчет Excel

- Отчет Word

- Видеоролик

## Возможная архитектура ПО на основе разрабатываемой БД:



Пользователи осуществляют взаимодействие с БД при помощи различных устройств со специальным ПО, которое в зависимости от уровня доступа и назначения может посылать запросы к БД либо напрямую, либо к клону через облачный сервис

# Моделирование

## Определение сущностей:

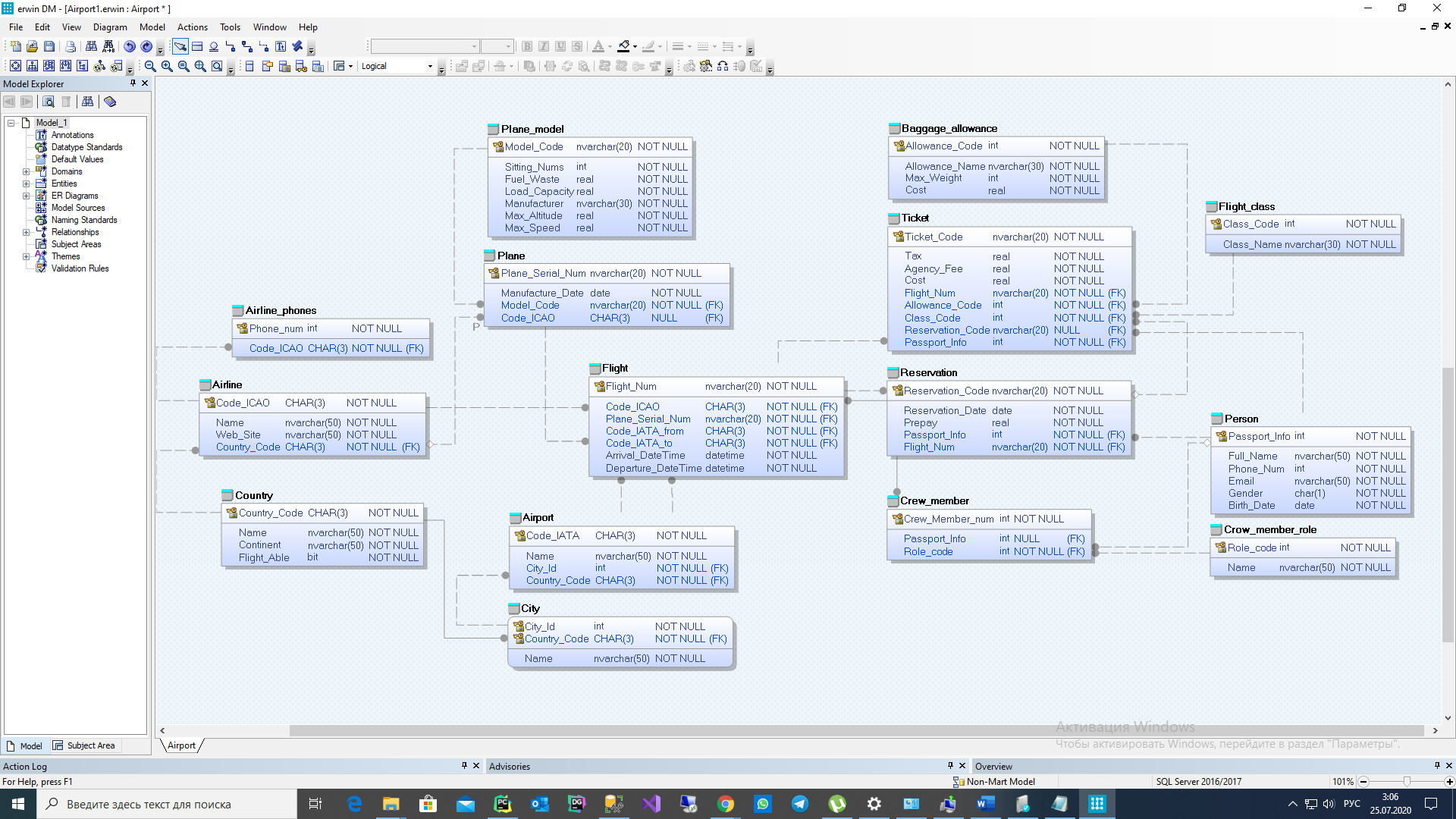
Определим основные сущности и опишем их:

|  |  |  |
| --- | --- | --- |
| Сущность | Описание | Атрибут |
| Аэропорт | Совокупность объектов типа «Аэропорт», осуществляющих гражданские авиаперелеты | Код аэропорта |
| Название (En) |
| Код города |
| Код страны |
| Город | Город базирования аэропорта | Код города(Id) |
| Название (En) |
| Код страны |
| Страна | Страна, на территории которой работает конкретный аэропорт | Код страны |
| Название (En) |
| Континент |
| Разрешение на перелет |
| Самолет | Воздушное судно, осуществляющее перелеты | Серийный номер |
| Дата выпуска |
| Номер модели |
| Код авиакомпании |
| Модель самолета | Технологическое решение, на основе которого разработан объект «Самолет» | Номер модели |
| Производитель (En) |
| Кол-во посадочных мест |
| Max Высота полета |
| Max Скорость полета |
| Грузоподъемность |
| Расход топлива |
| Авиакомпания | Организация, осуществляющая пассажирские авиаперелеты, владелец самолетного парка | Код авиакомпании |
| Название (En) |
| Страна |
| Сайт |
| Рейс | Объект, содержащий информацию о конкретном авиаперелете | Номер рейса |
| DateTime вылета |
| DateTime прилета |
| Серийный номер |
| Аэропорт отправления |
| Аэропорт прибытия |
| Код авиакомпании |
| Билет | Подтверждение права на перелет по конкретному рейсу  Содержит информацию о билетах | Код билета |
| Налоги |
| Агентский сбор |
| Цена |
| Код багажной нормы |
| Код класса |
| Код бронирования |
| Паспортные данные |
| Номер рейса |
| Багажная норма | Содержит информацию о багажных «тарифах» | Код багажной нормы |
| Название багажной нормы |
| Max вес |
| Стоимость |
| Класс места | Информация о классе места в самолёте | Код класса |
| Название Класса |
| Личность | Информация о личности, участвующей в процессе авиаперелета | Паспортные данные (серия+номер) |
| ФИО |
| Контактный телефон |
| Email |
| Пол |
| Дата рождения |
| Член экипажа | Информация о членах экипажа | Код члена экипажа |
| Код роли |
| Паспортные данные |
| Роль члена экипажа | Информация о ролях членов экипажей (должностей) | Код роли |
| Название роли |
| Член экипажа-Полет | Сущность, связывающая члена экипажа и рейс | Код члена экипажа |
| Номер рейса |

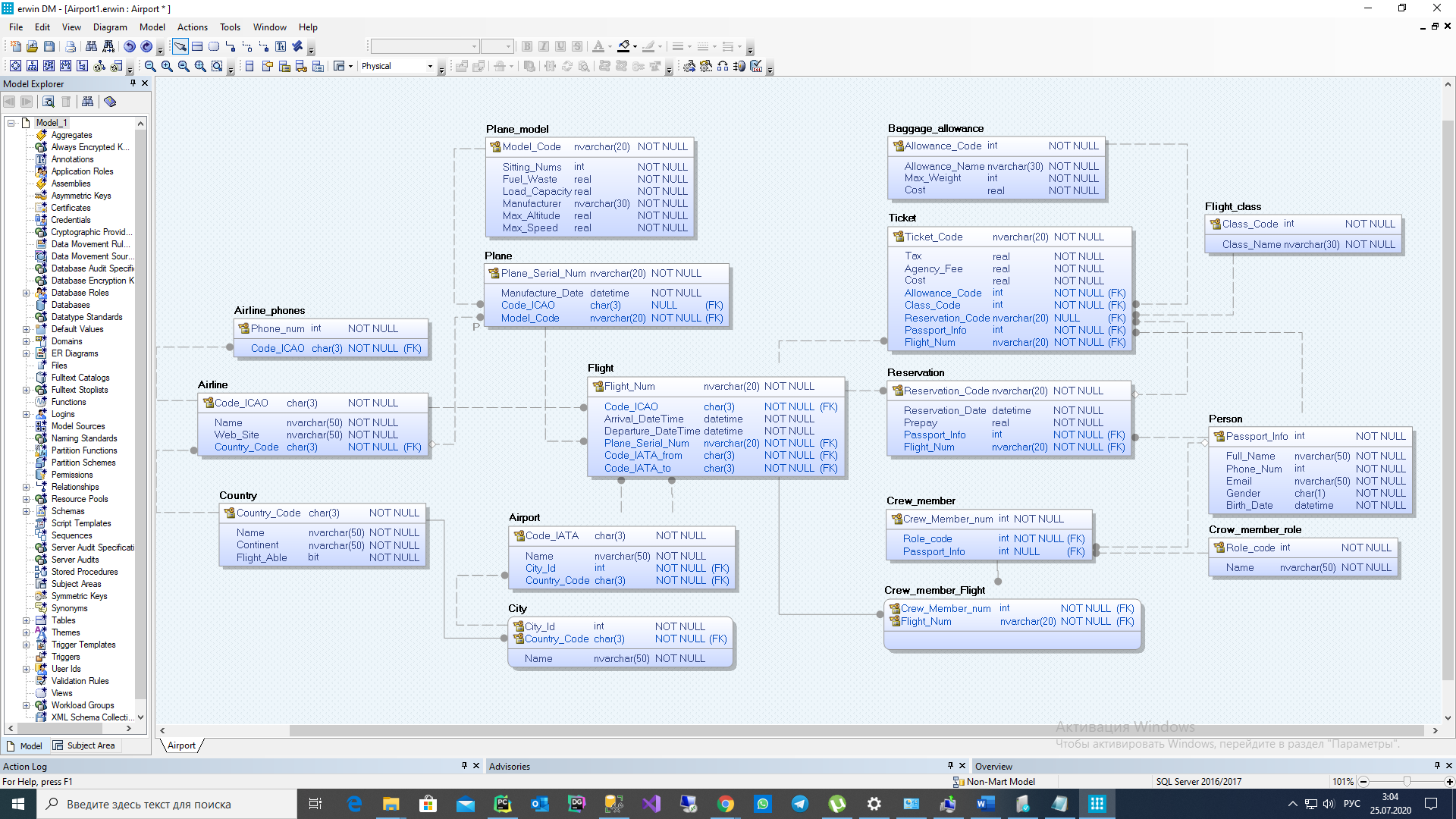
## 

## Модели:

### Логическая модель ErWin:



### Физическая модель ErWin:



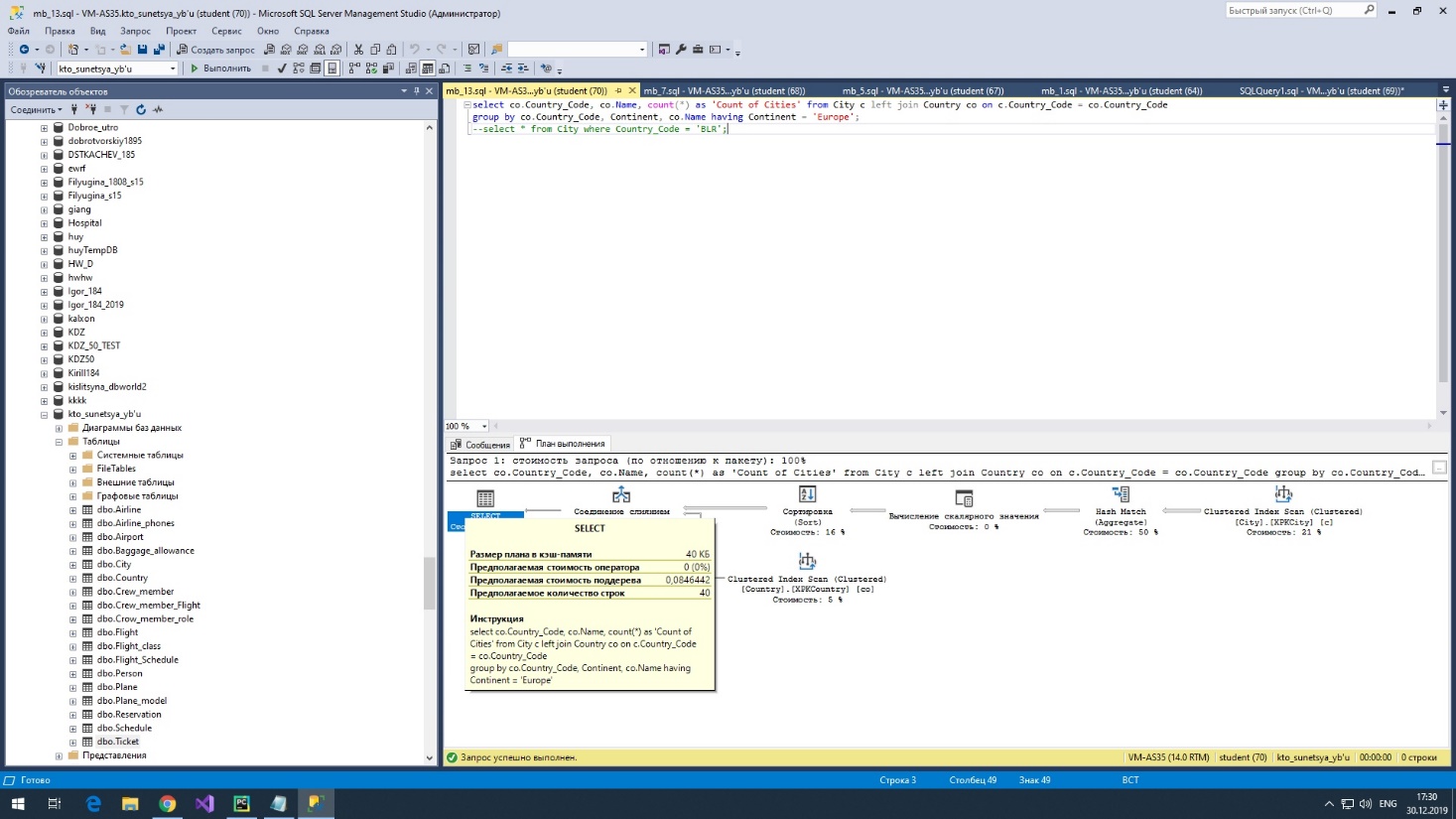
## Соответствует ли итоговая схема 4НФ?

Да, так как она находится в НФБК и все неключевые атрибуты всех таблиц зависят от их потенциальных ключей. В частности, все нетривиальные многозначные зависимости фактически являются функциональными зависимостями от их потенциальных ключей.

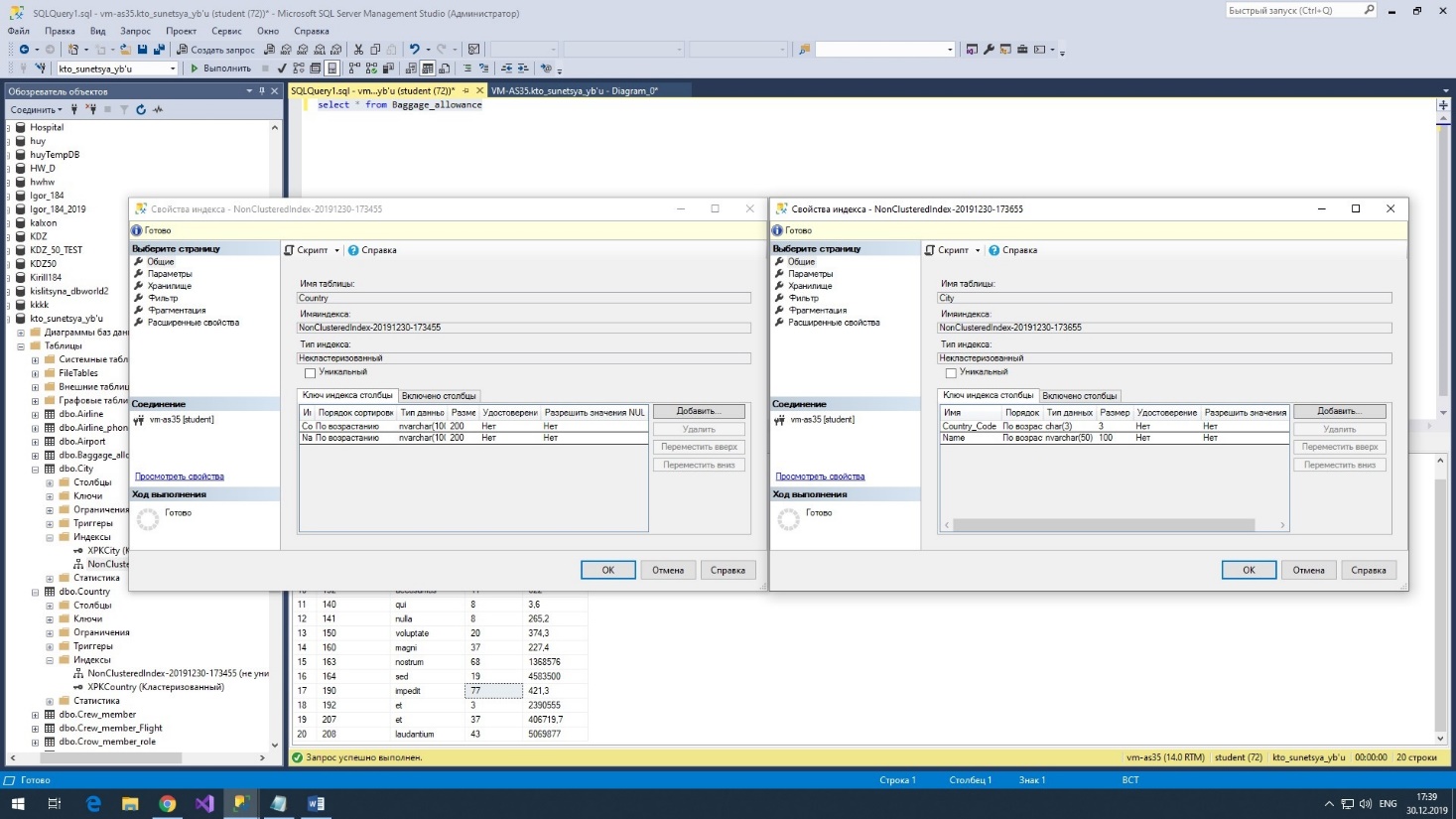
## Демонстрация работоспособности индексов:

Код запроса: select [co.Country](https://vk.com/away.php?to=http%3A%2F%2Fco.Country&cc_key=" \t "_blank)\_Code, [co.Name](https://vk.com/away.php?to=http%3A%2F%2Fco.Name&cc_key=" \t "_blank), count(\*) as 'Count of Cities' from City c left join Country co on [c.Country](https://vk.com/away.php?to=http%3A%2F%2Fc.Country&cc_key=" \t "_blank)\_Code = [co.Country](https://vk.com/away.php?to=http%3A%2F%2Fco.Country&cc_key=" \t "_blank)\_Code   
group by [co.Country](https://vk.com/away.php?to=http%3A%2F%2Fco.Country&cc_key=" \t "_blank)\_Code, Continent, [co.Name](https://vk.com/away.php?to=http%3A%2F%2Fco.Name&cc_key=) having Continent = 'Europe';

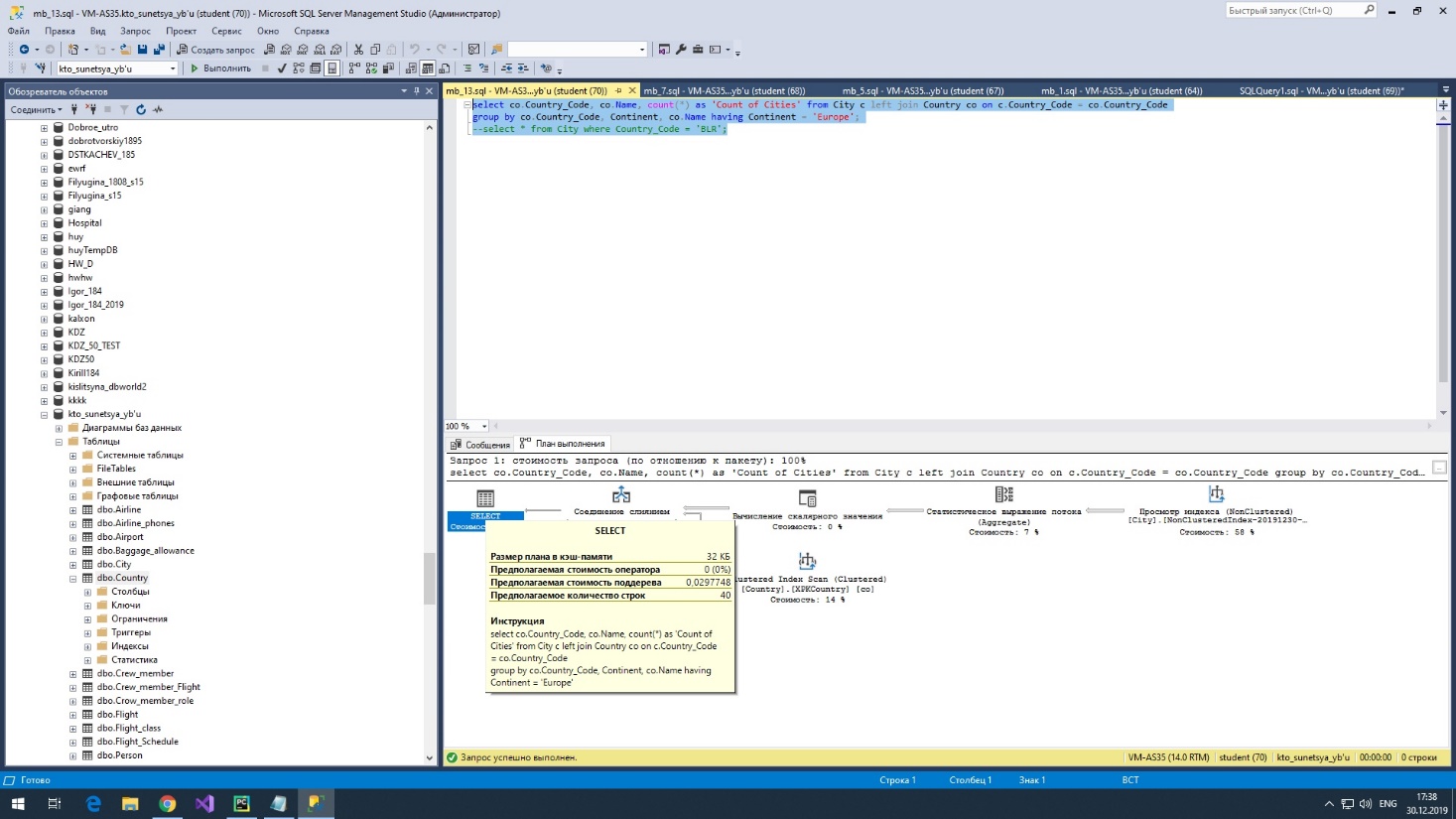
Скорость выполнения до настройки индексов: 0,0846



Настраиваем индексы:



Скорость работы после настройки индексов: 0,297



Видим увеличение скорости выполнения чуть ли не в 3 раза!

# Разработка SQL-запросов, функций, триггеров, процедур, представлений

## SQL-запросы, функции, триггеры, процедуры, представления:

1. **Представление, содержащие аналитику по приросту продаж билетов по годам**

create VIEW ticket\_sales\_analysis as

select agg\_table.[Year], agg\_table.[The total cost],

round((agg\_table.[The total cost] - LAG(agg\_table.[The total cost], 1, 0) OVER(ORDER BY agg\_table.[Year]))/LAG(agg\_table.[The total cost], 1, agg\_table.[The total cost])

OVER(ORDER BY agg\_table.[Year]) \* 100, 0)

as 'Increase in ticket sales %'

from (select YEAR(f.Arrival\_DateTime) as 'Year', SUM(t.Cost) as 'The total cost'

from Flight f left join Ticket t on f.Flight\_Num=t.Flight\_Num group by YEAR(f.Arrival\_DateTime)) as agg\_table;

1. **Представление, содержащее информацию о стоимости билетов, налогах, агентских сборах и размерах предоплат по годам**

create VIEW ticket\_analytics as

select \* from (select DISTINCT YEAR(Departure\_DateTime) as 'Year',

SUM(Cost) over(partition by YEAR(Departure\_DateTime)) as Total\_cost,

AVG(Cost) over(partition by YEAR(Departure\_DateTime)) as Average\_Cost,

MIN(Cost) over(partition by YEAR(Departure\_DateTime)) as Min\_Cost,

MAX(Cost) over(partition by YEAR(Departure\_DateTime)) as Max\_Cost,

SUM(Agency\_Fee) over(partition by YEAR(Departure\_DateTime)) as Total\_Agency\_Fee,

AVG(Agency\_Fee) over(partition by YEAR(Departure\_DateTime)) as Average\_Agency\_Fee,

MIN(Agency\_Fee) over(partition by YEAR(Departure\_DateTime)) as Min\_Agency\_Fee,

MAX(Agency\_Fee) over(partition by YEAR(Departure\_DateTime)) as Max\_Agency\_Fee,

SUM(Tax) over(partition by YEAR(Departure\_DateTime)) as Total\_Tax,

AVG(Tax) over(partition by YEAR(Departure\_DateTime)) as Average\_Tax,

MIN(Tax) over(partition by YEAR(Departure\_DateTime)) as Min\_Tax,

MAX(Prepay) over(partition by YEAR(Departure\_DateTime)) as Max\_Tax,

SUM(Prepay) over(partition by YEAR(Departure\_DateTime)) as Total\_Prepay,

AVG(Prepay) over(partition by YEAR(Departure\_DateTime)) as Average\_Prepay,

MIN(Prepay) over(partition by YEAR(Departure\_DateTime)) as Min\_Prepay,

MAX(Prepay) over(partition by YEAR(Departure\_DateTime)) as Max\_Prepay

from Ticket t inner join Flight f on t.Flight\_Num=f.Flight\_Num

inner join Reservation r on t.Reservation\_Code=t.Reservation\_Code) as tt;

1. **(Запрос) Выбираем членов экипажа, которые хоть раз летали в качестве клиентов**

select \* from Person p join (select \*

from Crew\_member cm

where Passport\_Info in

(select Passport\_Info from Reservation r where cm.Passport\_Info=r.Passport\_Info)) as member

on p.Passport\_Info=member.Passport\_Info;

1. **Выбираем клиентов, которые не являются членами экипажа**

select p.Passport\_Info, Full\_Name from Person p full outer join Crew\_member cm

on p.Passport\_Info=cm.Passport\_Info where Crew\_Member\_num is null;

1. **Информация о количестве, построенных самолётов определённых моделей за последние десять лет, по годам.**

select Model\_Code, [2009], [2010], [2011], [2012], [2013], [2014], [2015], [2016], [2017], [2018], [2019]

from (SELECT Plane\_Serial\_Num, Model\_Code, YEAR(Manufacture\_Date) as [Year] FROM Plane) p

PIVOT (COUNT(Plane\_Serial\_Num)

FOR [Year] in ([2009], [2010], [2011], [2012], [2013], [2014], [2015], [2016], [2017], [2018], [2019])) pvt;

1. **Добавление нового работника**

CREATE PROCEDURE NewCrewMemberByName

@FullName nvarchar(50),

@CrewMemberNum int,

@RoleCode int

AS

BEGIN

SET NOCOUNT ON;

begin try

insert into Crew\_member values

(@CrewMemberNum, @RoleCode, (select Passport\_Info from Person where Full\_Name = @FullName));

end try

begin catch

declare @Full\_name\_count int;

set @Full\_name\_count = (select count(\*) from Person where Full\_Name = @FullName);

if @Full\_name\_count > 0

(select 'Error in table Person')

else

select 'Unknown error'

end catch

END

1. **Вывод средней продолжительности эксплуатации тех или иных моделей самолётов от наибольшей к наименьшей**

SELECT pm.Model\_Code, avg(datediff(year, Manufacture\_Date, getdate())) AS 'average exploitation of planes by plane model'

FROM Plane p JOIN Plane\_model pm ON p.Model\_Code = pm.Model\_Code

GROUP BY pm.Model\_Code

ORDER BY 'average exploitation of planes by plane model' DESC

1. **Количество городов в стране заданного региона**

select [co.Country](https://vk.com/away.php?to=http%3A%2F%2Fco.Country&cc_key=" \t "_blank)\_Code, [co.Name](https://vk.com/away.php?to=http%3A%2F%2Fco.Name&cc_key=" \t "_blank), count(\*) as 'Count of Cities' from City c left join Country co on [c.Country](https://vk.com/away.php?to=http%3A%2F%2Fc.Country&cc_key=" \t "_blank)\_Code = [co.Country](https://vk.com/away.php?to=http%3A%2F%2Fco.Country&cc_key=" \t "_blank)\_Code   
group by [co.Country](https://vk.com/away.php?to=http%3A%2F%2Fco.Country&cc_key=" \t "_blank)\_Code, Continent, [co.Name](https://vk.com/away.php?to=http%3A%2F%2Fco.Name&cc_key=" \t "_blank) having Continent = 'Europe'

1. **Представление для отчёта в Excel**

create view for\_excel\_report  
as  
select Plane\_Serial\_Num,Manufacture\_Date, pm.Model\_Code, Sitting\_Nums, Fuel\_Waste, Load\_Capacity, Manufacturer,  
Max\_Altitude, Max\_Speed, a.Code\_ICAO, Name, Country\_Code  
from Plane p left join Airline a on p.Code\_ICAO=a.Code\_ICAO  
left join Plane\_model pm on pm.Model\_Code=p.Model\_Code;

1. **Распределение самолётов по типам в зависимости от количества мест и выбросов**

select Plane\_Serial\_Num,

case when Fuel\_Waste > 40000 and Sitting\_Nums > 2000

then 'super jet'

when Fuel\_Waste > 10000 and Fuel\_Waste <= 40000 and Sitting\_Nums <= 2000 and Sitting\_Nums > 100

then 'jet'

when Fuel\_Waste <= 10000 and Sitting\_Nums > 1 and Sitting\_Nums <= 100

then 'Private jet'

when Fuel\_Waste > 50000 and Sitting\_Nums <= 30 and Load\_Capacity > 10000

then 'Transport jet'

else 'Other jets'

end 'plane type'

from Plane p left join Plane\_model pm on p.Model\_Code=pm.Model\_Code;

1. **Триггер удаления билета**

SET ANSI\_NULLS ON   
GO   
SET QUOTED\_IDENTIFIER ON   
GO   
CREATE TRIGGER Ticket\_delete   
ON [dbo].[Ticket]   
AFTER DELETE   
AS   
BEGIN   
SET NOCOUNT ON;   
delete from Reservation where Reservation\_Code in (select Reservation\_Code from deleted);   
  
END   
GO

1. **Функция подсчёта количества людей по классу в самолёте**

CREATE FUNCTION [dbo].[Count\_Of\_People\_InPlane\_ByClass]   
(   
— Add the parameters for the function here   
@Class\_code int,   
@Flight\_num nvarchar(20)   
)   
RETURNS int   
AS   
BEGIN   
— Declare the return variable here   
DECLARE @Count\_OfPeople\_ByClass int   
  
— Add the T-SQL statements to compute the return value here   
set @Count\_OfPeople\_ByClass = (select count(\*) from Ticket where Class\_Code = @Class\_code and Flight\_Num = @Flight\_num);   
  
— Return the result of the function   
RETURN @Count\_OfPeople\_ByClass   
  
END

1. **Функция подсчёта потока пассажиров в странах по периодам времени**

CREATE FUNCTION [dbo].[Country\_Traffic\_By\_Period]   
(   
— Add the parameters for the function here   
[@Since](https://vk.com/id3258815) datetime,   
@By datetime,   
@CountryCode char(3)   
)   
RETURNS int   
AS   
BEGIN   
— Declare the return variable here   
DECLARE @Count\_of\_people int   
  
— Add the T-SQL statements to compute the return value here   
set @Count\_of\_people = (select count(\*) from Ticket   
where Flight\_Num in(select Flight\_Num from Flight   
where [@Since](https://vk.com/id3258815) <= Arrival\_DateTime and Arrival\_DateTime <= @By and Code\_IATA\_to in (select Code\_IATA from Airport   
where Country\_Code = @CountryCode)));   
  
— Return the result of the function   
RETURN @Count\_of\_people   
  
END

## Реализуемые бизнес-функции:

* Сбор общей информации о количестве перелётов по годам
* Детальное представление информации о перелётах по годам
* Мониторинг цен на билеты в разные периоды времени
* Осуществление бронирования билетов
* Сбор информации о клиентах с возможностью последующей разработки персональных предложений
* Сбор информации о средней продолжительности эксплуатации самолётов
* Управление персоналом

## Алгоритмы обработки данных:

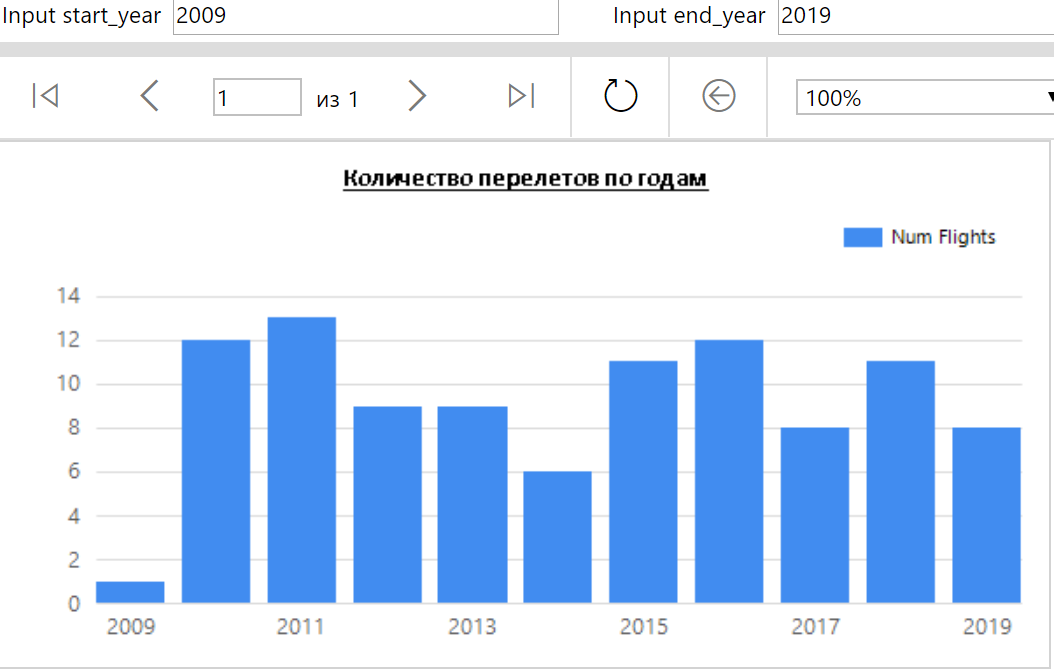
- Линейный поиск

- Поиск по индексу

# Отчетность

## Отчет SSRS:

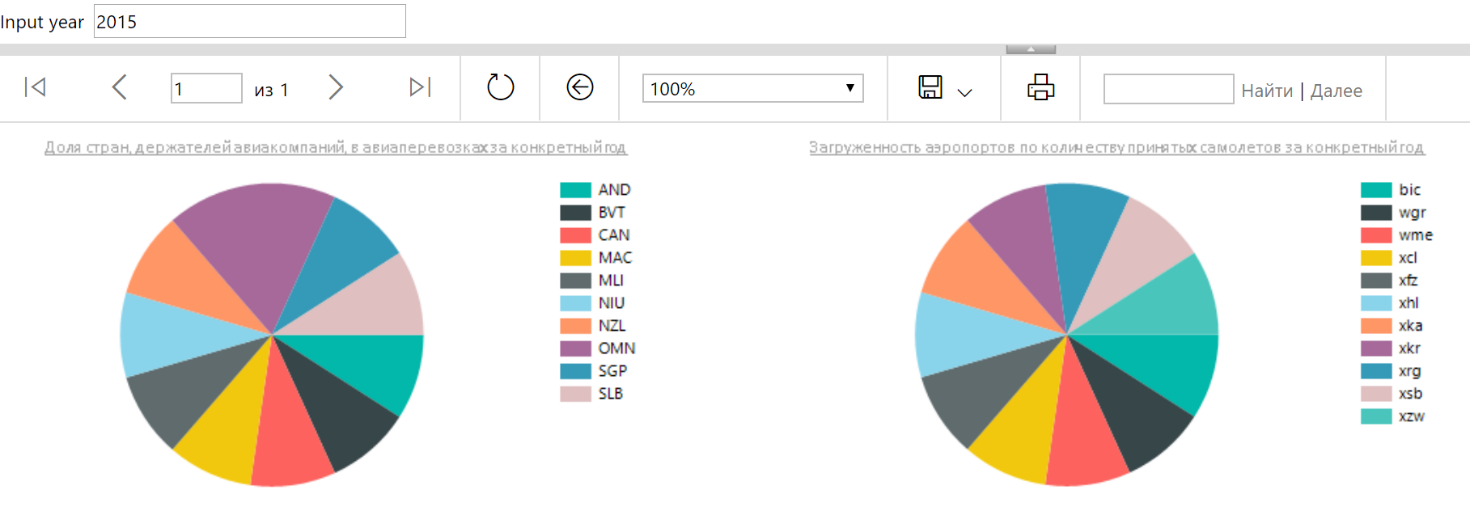
Главный отчет выполнен в виде гистограммы, содержащей информацию о количестве авиаперелетов по годам в заданном пользователей интервале, где на оси Ох отмечены года, а на оси Оу содержится информация о количестве. Пример отчета с информацией за 2009-2019 года представлен ниже:



Вложенный отчет содержит две круговые диаграммы, содержащие информацию о:

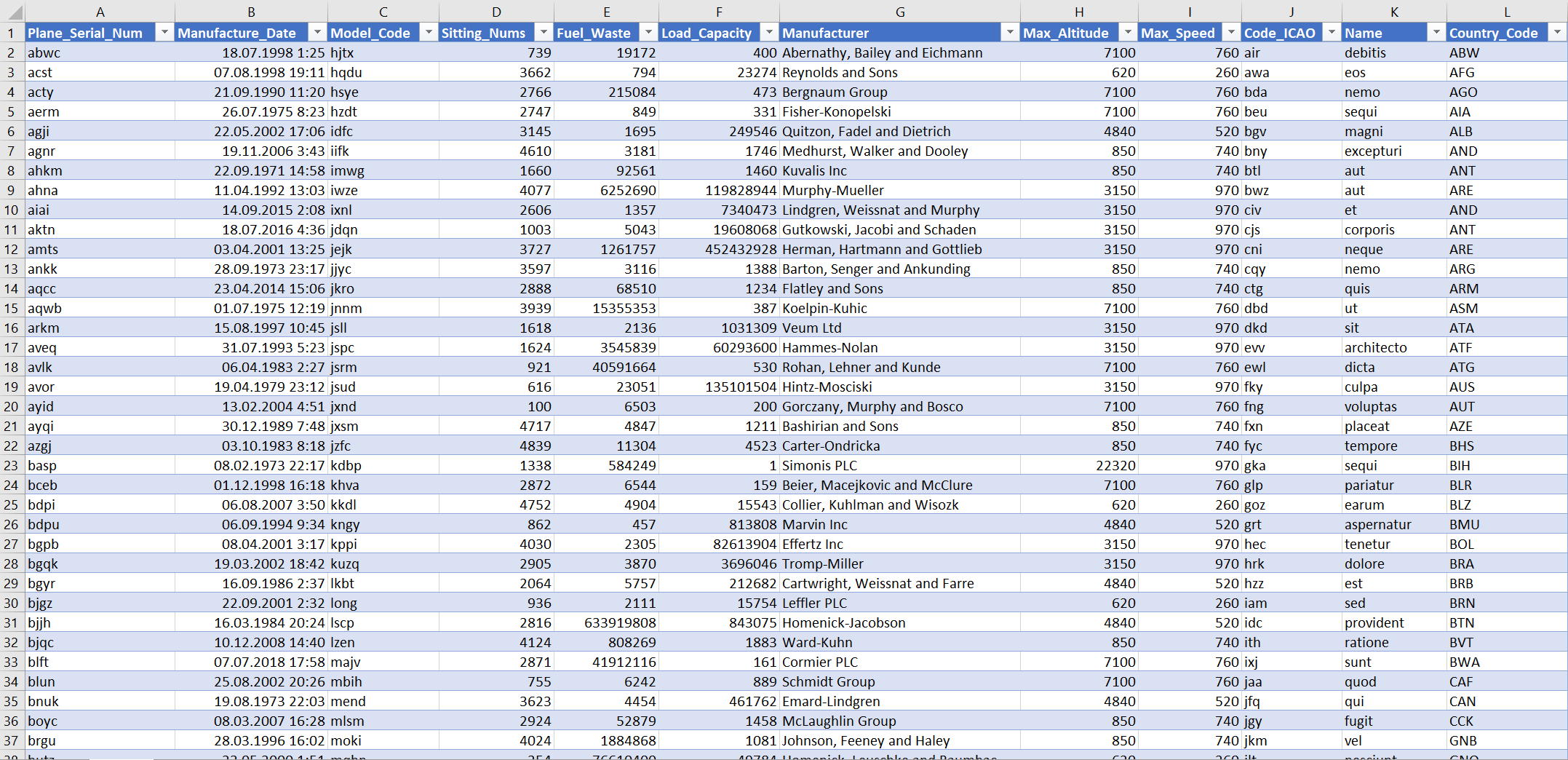
1. Доле стран, держателей авиакомпаний, в авиаперевозках
2. Загруженность аэропортов по количеству принятых самолетов

за конкретный год, выбранный в главном отчете путем нажатия на интересующий столбец. В качестве примера – 2015 год:

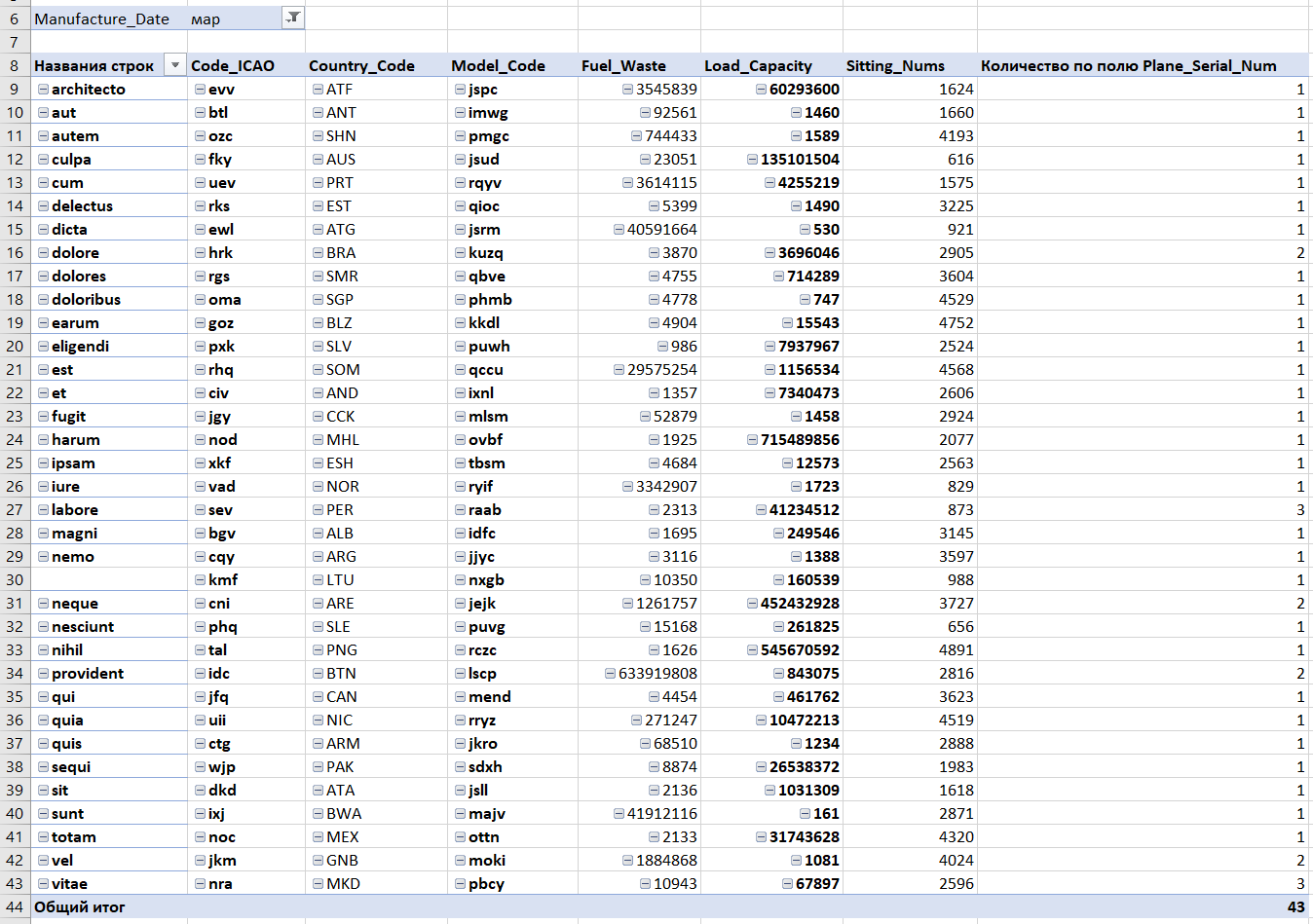


## Отчет Excel:

Лист1 показывает таблицу, содержащую все данные по всем воздушным бортам:



Лист2 показывает сводную таблицу, систематизирующую данные об авиакомпаниях через информацию о принадлежащих им самолётах. Возможна фильтрация данных по году производства этих самолётов:



Лист3 показывает диаграмму, содержащую данные о максимально возможном пассажиропотоке по авиакомпаниям, за счёт подсчёта суммы количества сидений в самолётах каждой авиакомпании. Также возможна фильтрация данных по времени (год, квартал) и по названию авиакомпании.

