Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«Национальный исследовательский университет

“Высшая школа экономики”»

###### Факультет бизнеса и менеджмента

###### Программа «Бизнес-Информатика»

Морев Евгений Александрович

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ АЭРОПОРТА**

Курсовая работа студента 2 курса бакалавриата группы № ББИ187

**Научный руководитель:**

Доцент кафедры Бизнес-Аналитики

Ямпольский Сергей Михайлович

Москва, 2020

Оглавление

[Введение 3](#_Toc42988161)

[Глава1 3](#_Toc42988162)

[Постановка задач 3](#_Toc42988163)

[Описание предметной области 4](#_Toc42988164)

[Основные Бизнес-процессы 5](#_Toc42988165)

[Глава 2 5](#_Toc42988166)

[Моделирование базы данных 5](#_Toc42988167)

[Инфологическая модель 6](#_Toc42988168)

[Описание таблиц и атрибутов сущностей 6](#_Toc42988169)

[Описание связей 16](#_Toc42988170)

[Логическая модель базы данных 21](#_Toc42988171)

[Глава 3 22](#_Toc42988172)

[Реализация базы данных 22](#_Toc42988173)

[Диаграмма базы данных 22](#_Toc42988174)

[Примеры заполнения данных 23](#_Toc42988175)

[Примеры разработки запросов в базе данных 29](#_Toc42988176)

[Примеры разработки процедур 32](#_Toc42988177)

[Примеры разработки триггеров 34](#_Toc42988178)

[Примеры разработки функций 38](#_Toc42988179)

[Примеры разработки представлений 40](#_Toc42988180)

[Глава 4 42](#_Toc42988181)

[Приложение для базы данных «Аэропорт» 42](#_Toc42988182)

[Заключение 44](#_Toc42988183)

[Список используемой литературы 1](#_Toc42988184)

# Введение

###### Мы живём в информационном мире, когда грамотное использование информации, ее хранение и управление - стало основным фактором успеха любой корпорации. Эффективное управление данными - не простая задача, особенно, в случаях работы с серьезными и крупными сферами деятельности бизнеса. Однако, следует отметить, чем больше компания становится, тем больше она нуждается в автоматизации бизнес-процессов.

Для исследования была выбрана предметная область «Аэропорт». В данной работе будет проведен анализ процесса проектирования и разработки базыданных аэропорта**.** Объектом исследования являются способы хранения информации в реляционной базе данных аэропорта. Предметом исследования служит проектирование и разработка базы данных, для организации бизнес-процессов и повышение автоматизации деятельности аэропорта. Сама база служит для сбора, хранения и обработки информации, связанной со всеми этапами работы аэропорта.

# Глава1

# Постановка задач

###### Целью работы является изучение предметной области, ее проектирование и разработка. Для этого будут выполнены поэтапные задачи.

###### Этапы работы : изучение и анализ предметной области для создания правдоподобной модели базы данных аэропорта, создание инфологической модели для наглядного и полноценного представления предметной области аэропорта, создание даталогической модели(машинное проектирование) для нормализации сущностей базы данных, создание физической модели базы данных на основе даталогической для определения способа размещения данных в среде хранения и доступа к этим данным, формирование полезных запросов, триггеров и т.д, визуализация процесса с помощью реализации графического приложения под созданную базу данных.

## Описание предметной области

В наше время воздушный транспорт является наиболее быстрым и востребованным способом перевозки пассажиров, багажа и грузов на большие расстояния. При данных задачах имеет большое значение грамотная организация служб аэропорта, правильное составление графика рейсов. Для каждого рейса нужно выделить самолет, экипаж, и внести в график взлетов на взлетно-посадочной полосе. Структура аэропорта состоит из большого числа пунктов, взаимодействующих между собой. Очень удобно все пункты структурировать в виде таблицы с помощью баз данных. Что позволит, составлять графики полётов, отслеживать размеры багажа, зарегистрированных пассажиров на рейс, даст информацию о задержках/отменах рейсов, о авиалиниях с которыми данный аэропорт сотрудничает, позволит отслеживать и менять рабочий штаб.

Для чего нужна автоматизация бизнес-процессов. Например, верно составленный график полётов позволяет эффективно использовать время занятости полосы и загруженность рейсов. Грамотное отслеживания параметров багажа поможет избежать возможных проблем с перевесом. Так, структурированная информация об авиакомпаниях и их флотах поможет избежать «нежданных гостей». Ну и наконец, отслеживание покупок билетов по конкретному тарифу с выбранным местом поможет избежать переполненности самолёта, либо проданного дважды места.

Подобная база данных может быть полезна для любого аэропорта, либо авиакомпании, либо компании, которая занимается грузоперевозками. Однако, для грузоперевозок ее придётся немного модифицировать. Кроме того, владельцы компаний всегда должны отслеживать работу своих сотрудников, отслеживать взаимодействие различных пунктов деятельности компании. Для удобного хранения информации, ее организации необходима база данных, включающая всю необходимую информацию. Для работы с базами данных необходимо быть специалистом. Но, в таком случае она становится недоступным ресурсом для обычного работника, либо пользователя. Поэтому нужно создать программу, которая будет понятна для обычного человека и автоматизирует работу с базой данных.

## Основные Бизнес-процессы

Для аэропорта характерны следующие сферы деятельности : обеспечение взлетов, посадок и стоянки; техническое обслуживание авиационной̆ техники; обеспечение обслуживания пассажиров, багажа, почты и грузов при внутренних и международных воздушных перевозках; аэродромное обеспечение; регистрация пассажиров, проверка багажа, наличия билетов, ориентирование людей в аэропорте; отслеживание состояния рейса; систематическая обработка входящей информации, ее автоматизация и вывод обработанных данных на различные способы отображения. С помощью объединения оперативной базы данных и внешних источников информации создается единое информационное пространство для аэропорта.

# Глава 2

# Моделирование базы данных

Инфологическая модель отображена через ER-диаграмму в нотации Чена.

Используются классические знаки и фигуры. Cущности изображены в виде подписанных прямоугольников. У сущности, которая определяется другой сущностью есть двойная граница на прямоугольнике(индикатор не дублируется). Атрибуты сущностей — в виде подписанных овалов: ключевые названия атрибутов(идентификаторы)) подчеркиваются. Связи, соединяющие сущности проводятся линиями, от одной сущности к другой, в середине связей есть подписанный ромб, который описывает связь. Вдобавок, идентифицирующие связи отображаются в виде двойного ромба. Далее, в связи нужно указывать её мощность. Мощности бывают 3х видов : один к одному, один ко многим, многие ко многим.

## Инфологическая модель

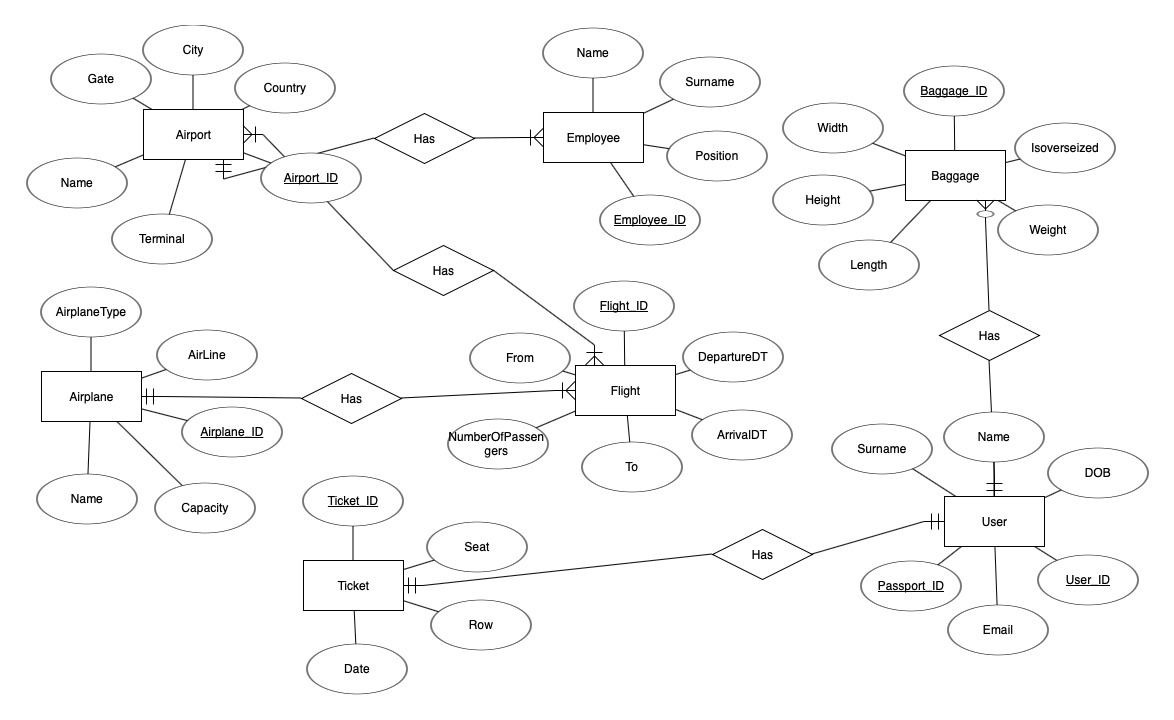


Рисунок . Инфологическая модель

## Описание таблиц и атрибутов сущностей

При разработке базы данных были выявлены следующие сущности и атрибуты. Некоторые из них появились лишь на даталогическом уровне для нормализации таблиц:

1)Airport - сущность, предназначенная для приёма, отправки, базирования самолётов, обслуживания воздушных перевозок. Хранит информацию о расположении, терминалах и выходах, имени аэропорта.

* Airport\_ID - уникальный номер аэропорта

Тип данных(INT)

Primary Key

* Name - название аэропорта

Тип данных(VARCHAR(30))

Не ключевой атрибут

* CityID - уникальный номер города, принадлежащего стране, в котором находится аэропорт

Тип данных(INT)

Foreign Key

2)Airplane - сущность, предназначенная для перевоза пассажиров. Хранит информацию о типе самолёта(Боинг, Эирбус, Региональные), Авиакомпании и вместимости самолёта, имени самолёта.

* Airplane\_ID - уникальный номер самолёта

Тип данных(INT)

PrimaryKey

* AirplaneTypeID - уникальный номер типа самолёта.

Тип данных(ID)

Foreign Key

* AirlineID - уникальный номер авиалинии.

Тип данных(ID)

Foreign Key

* Name - имя самолёта

Тип данных(VARCHAR(30))

Не ключевой атрибут

3)Ticket - сущность, для посадки пассажиров в самолёт. Хранит информацию о пассажире и его посадочном месте, дате рейса, багажа, цене билета.

* TicketID - уникальный номер билета

Тип данных(INT)

Primary Key

* SeatID - уникальный номер посадочного места( с уже сформированной ценой за конкретный тариф)

Тип данных(INT)

Foreign Key

* PassengerID - уникальный номер пассажира

Тип данных(INT)

Foreign Key

* FlightID - уникальный номер рейса

Тип данных(INT)

Foreign Key

* BaggageID - уникальный номер багажа

Тип данных(INT)

Foreign Key

* Date - дата полёта

Тип данных(DATETIME)

Не ключевой атрибут

4)Flight - сущность рейс(полёт). Хранит информацию о времени вылета, прилета, количестве пассажиров на рейсе, выполнении рейса(задержка,отмена), маршруте следования.

* FlightID - уникальный номер рейса

Тип данных(INT)

Primary Key

* DTDeparture - дата вылета

Тип данных(DATETIME)

Не ключевой атрибут

* DTArrival - дата прилёта

Тип данных(DATETIME)

Не ключевой атрибут

* NumberOfPassengers

Тип данных(INT)

Не ключевой атрибут

* DelayMinutes - время задержки

Тип данных(INT)

Не ключевой атрибут

* GateID - уникальный номер выхода

Тип данных(INT)

Foreign Key

* [From] - город вылета

Тип данных(VARCHAR(30))

Не ключевой атрибут

* [To] - город прилёта

Тип данных(VARCHAR(30))

Не ключевой атрибут

* IsCancelled - отменен рейс или нет.

Тип данных(BOOLEAN)

Не ключевой атрибут

* AirplaneID - уникальный номер самолёта

Тип данных(INT)

Foreign Key

5)Passenger - сущность физического лица, пассажир, совершающий покупку билета. Хранит информацию об имени и фамилии, дате рождения, паспортных данных, адреса почты.

* PassengerID - уникальный номер пользователя

Тип данных(INT)

Primary Key

* Name - имя пассажира

Тип данных(VARCHAR(30))

Не ключевой атрибут

* Surname - фамилия пассажира

Тип данных(VARCHAR(40))

Не ключевой атрибут

* DOB - дата рождения пассажира

Тип данных(DATE)

Не ключевой атрибут

* PassportID - уникальная серия и номер паспорта пассажира

Тип данных(INT)

Не ключевой атрибут

* Mail - электронная почта пользователя

Тип данных(VARCHAR(45))

Не ключевой атрибут

6)Baggage - сущность, описывающая параметры багажа. Хранит информацию о его весе и размерах.

* BaggageID - уникальный номер багажа

Тип данных(INT)

Primary Key

* Weight - вес багажа

Тип данных(FLOAT)

Не ключевой атрибут

* Height - высота багажа

Тип данных(FLOAT)

Не ключевой атрибут

* Width - ширина багажа

Тип данных(FLOAT)

Не ключевой атрибут

* Length - длина багажа

Тип данных(FLOAT)

Не ключевой атрибут

* IsOversized - есть ли перевес багажа.

Тип данных(BOOLEAN)

Не ключевой атрибут

7)OversizedPrice - сущность, предназначенная для формирования стоимости нестандартного багажа.

* BaggageID - уникальный номер багажа.

Тип данных(INT)

Foreign Key

* Price - цена провоза багажа

Тип данных(DECIMAL)

Не ключевой атрибут

* PermittedWeight - разрешенный вес

Тип данных(INT)

Не ключевой атрибут

* Difference- разница между разрешенным весом и весом багажа.

Тип данных(INT)

Не ключевой атрибут

8)Gate - сущность расположения выхода на посадку. Хранит информацию о типе посадки(“рукав” или автобус), номере выхода

* GateID - уникальный номер выхода

Тип данных(INT)

Primary Key

* Floor - этаж, на котором расположен выход

Тип данных(INT)

Не ключевой атрибут

* TID - уникальный номер терминала

Тип данных(INT)

Foreign Key

* TypeOfGateID - уникальный номер типа выхода

Тип данных(INT)

Foreign Key

* Number - номер выхода

Тип данных(INT)

Не ключевой атрибут

9) TypeOfGate - сущность, определяющая способ доставки пассажиров на борт самолёта. Хранит информацию о типе выхода.

* TypeOfGateID - уникальный номер типа выхода

Тип данных(INT)

Primary Key

* Name - тип выхода(рукав/автобус)

Тип данных(VARCHAR(20))

Не ключевой атрибут

10) TicketPrice - сущность формирования цены билета. Хранит информацию о цене.

* SeatID - уникальный номер тарифа(бизнес/эконом, с багажом/без, впереди/ сзади)

Тип данных(INT)

Foreign Key

* Price - цена билета

Тип данных(DECIMAL)

Не ключевой атрибут

11) Revenue - сущность счета. Хранит информацию о цене билета и всех параметрах билета.

* TicketID - уникальный номер билета, по параметрам которого формируется итоговая стоимость билета

Тип данных(INT)

Foreign Key

* Sum - общая стоимость билета

Тип данных(DECIMAL)

Не ключевой атрибут

12)Terminal - сущность, предназначенная для обслуживания пассажиров воздушного транспорта и операций с багажом. Хранит информацию о площади здания и название аэропорта, к которому принадлежит.

* TID - уникальный номер терминала

Тип данных(INT)

Primary Key

* Square - площадь терминала

Тип данных(Float)

Не ключевой атрибут

* Number - номер турминала

Тип данных(INT)

Не ключевой атрибут

* AirportID - уникальный номер аэропорта

Тип данных(INT)

Foreign Key

13)City - сущность, предназначенная для определения локации аэропорта. Хранит информацию о название города и о том, в какой стране город находится.

* CityID - уникальный номер города

Тип данных(INT)

Primary Key

* Country ID - уникальный номер страны

Тип данных(INT)

Foreign Key

* CityName - название города

Тип данных(VARCHAR(30))

Не ключевой атрибут

14)Country - сущность страны, в которой находятся города, в которых находится аэропорт. Хранит информацию о название страны.

* CoutryID - уникальный номер страны

Тип данных(INT)

Primary Key

* CountryName - название страны

Тип данных(VARCHAR(30))

Не ключевой атрибут

15)Employee - сущность индивида, работающего в аэропорте. Хранит информацию об имени, фамилии и занимающей должности.

* EmployeeID - уникальный номер сотрудника аэропорта.

Тип данных(INT)

Primary Key

* Name - имя работника

Тип данных(VARCHAR(30))

Не ключевой атрибут

* Surname - фамилия работника

Тип данных(VARCHAR(30))

Не ключевой атрибут

* PositionID - уникальный номер занимаемой должности.

Тип данных(VARCHAR(50))

Foreign Key

* AirportID - в какому аэропорту принадлежит сотрудник

16) Position - должность сотрудника аэропорта.

* PositionID - уникальный номер должности

Тип данных(INT)

Primary Key

* Name - название должности

Тип данных(VARCHAR(50))

Не ключевой атрибут

17)Airport\_Airline - сущность связи Airport и Airline. На даталогическом уровне связь - многие ко многим происходит с помощью перекрёстной таблицы.

* AirportID - уникальный номер аэропорта

Тип данных(INT)

Foreign Key

* AirlineID - уникальный номер авиалинии

Тип данных(INT)

Foreign Key

18) Airline - сущность авиакомпании, которая осуществляет перевозку пассажиров. Хранит информацию о названии и количестве самолётов.

* AirlineID - уникальный номер должности

Тип данных(INT)

Primary Key

* Name - название авиакомпании

Тип данных(VARCHAR(40))

Не ключевой атрибут

* Capacity - вместимость флота

Тип данных(INT)

Не ключевой атрибут

19)Maker - сущность предприятия по производству самолётов. Хранит информацию о названии предприятия и стране производства.

* MakerID - уникальный номер производителя

Тип данных(INT)

Primary Key

* Name - кто производит

Тип данных(VARCHAR(30))

Не ключевой атрибут

20)AirplaneType - сущность модели самолёта. Хранит информацию об основных параметрах модлели.

* AirplaneTypeID - уникальный номер самолёта

Тип данных(INT)

Primary Key

* Capacity - вместимость самолёта

Тип данных(INT)

Не ключевой атрибут

* Name - название типа

Тип данных(VARCHAR(15))

Не ключевой атрибут

* MakerID - уникальный номер производителя

Тип данных(INT)

Foreign Key

21)AirplaneType\_Seat - сущность, описывающая тип размещения. Хранит информацию о месте, ряде, классе салона и наличии выбранного места.

* SeatID - уникальный номер места

Тип данных(INT)

Foreign Key

* ClassID - уникальный номер класса билета

Тип данных(INT)

Foreign Key

* IsBooking - забранировано место или нет

Тип данных(BOOLEAN)

Не ключевой атрибут

22)Class - сущность, предназначенная для выбора класса полёта(эконом, бизнес). Хранит информацию о типе класса.

* ClassID - уникальный номер класса

Тип данных(INT)

Primary Key

* Name - название типа класса

Тип данных(VARCHAR(25))

Не ключевой атрибут

23)Seat - сущность места. Хранит информации о положение места(ABCDEF…) в ряду.

* SeatID - уникальный номер места

Тип данных(INT)

Primary Key

* Letter - расположение места в ряде

Тип данных(VARCHAR(1))

Не ключевой атрибут

* RowID - уникальный номер ряда

Тип данных(INT)

Foreign Key

24)Row - сущность ряда

* RowID - уникальный номер ряда

Тип данных(INT)

Primary Key

* Number - расположение места в ряду

Тип данных(INT)

Не ключевой атрибут

## Описание связей

При разработке базы данных были построены следующие связи между таблицами:

1. Связь «Country-City». Таблица Country является главной(родительской), а таблица City - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Counrty предается индикатор Country ID в City. Отношение один ко многим, так как у города только одна страна, но у одной страны много городов.
2. Связь «City-Airport». Таблица City является главной(родительской), а таблица Airport - подчиненной(дочерняя). Не идентифицирующая связь. Неможет быть NULL значение. Из City передается индикатор CityID. Отношение один ко многим, так как один аэропорт может находится только в одном городе, но в одном городе может быть несколько аэропортов.
3. Связь «Airport-Terminal». Таблица Airport является главной(родительской), а таблица Terminal - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Airport передается индикатор AirportID. Отношение один ко многим, так как у одного аэропорта может быть несколько терминалов, но у одного терминала только один аэропорт.
4. Связь «Airport-Employee». Таблица Airport является главной(родительской), а таблица Employee - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Airport передается индикатор AirportID. Отношение один ко многим, так как у одного аэропорта может быть несколько сотрудников, но один сотрудник может работать только в одном аэропорте.
5. Связь «Airport-Airline». На даталогическом уровни не бывает связи многие ко многим. Для такой связи используется перекрестная таблица. Связываются все возможные аэропорты со всеми возможными авиакомпаниями. Из Airport передается идентифицирующая связь в Airport\_Airline, где Airport является главной таблицей, а Airport\_Airline - подчиненной. Из Airport передается индикатор AirportID. Такая же процедура реализуется в связи «Airline-Airport\_Airline», где Airline является главной таблицей, а Airline\_Airport - подчиненной.
6. Связь «Position-Employee». Таблица Position является главной (родительской), а таблица Employee - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Position передается индикатор PositionID. Отношение один ко многим, так как одну должность могут занимать сотрудников, но один сотрудник может работать только на одной должности.
7. Связь «Airline-Airplane». Таблица Airline является главной (родительской), а таблица Airplane - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Airline передается индикатор AirlineID. Отношение один ко многим, так как у одной компании несколько самолётов, но один самолёт принадлежит только одной авиакомпании.
8. Связь «Maker-AirplaneType». Таблица Maker является главной (родительской), а таблица AirplaneType - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Maker передается индикатор MakerID. Отношение один ко многим, так как один производитель может выпустить разные модели самолётов, но конкретную модель выпускает только один производитель.
9. Связь «AirplaneType-Airplane». Таблица AirplaneType является главной (родительской), а таблица Airplane - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из AirplaneType передается индикатор AirplaneTypeID. Отношение один ко многим, так как может быть выпущено много самолетов этой модели, но самолет принадлежит только к одной, конкретной модели.
10. Связь «Airplane-Flight». Таблица Airplane является главной (родительской), а таблица Flight- подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Airplane передается индикатор AirplaneID. Отношение один ко многим, так как один рейс реализуется только засчет одного самолёта, но у один самолёт осуществляет много рейсов.
11. Связь «Row-Seat». Таблица Row является главной (родительской), а таблица Seat подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Row передается индикатор RowID. Отношение один ко многим, так как в одном ряду есть несколько, но одно место находится в одном ряду.
12. Связь «Seat-AirplaneType\_Seat». Таблица Seat является главной (родительской), а таблица AirplaneType\_Seat подчиненной(дочерняя). Идентифицирующая связь. Из Seat передается индикатор SeatID. Отношение один ко многим, так как одному типу рассадки принадлежат несколько мест, но одному месту принадлежит только один тип рассадки.
13. Связь «Class-AirplaneType\_Seat». Таблица Class является главной (родительской), а таблица AirplaneType\_Seat подчиненной(дочерняя). Не идентифицирующая связь. Из Class передается индикатор ClassID. Отношение один ко многим, так как в одном классе есть несколько вариантов рассадки, но конкретному типу рассадки принадлежит конкретный класс.
14. Связь «Terminal-Gate». Таблица Terminal является главной(родительской), а таблица Gate - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Terminal передается индикатор TerminalID. Отношение один ко многим, так как у одного терминала может быть несколько выходов, но один выход принадлежит только к одному терминалу.
15. Связь «TypeOfGate-Gate». Таблица TypeOfGate является главной(родительской), а таблица Gate - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из TypeOFGate передается индикатор TypeOfGateID. Отношение один ко многим, так как у одного выхода какой-то один, конкретный вариант транспортировки, но могут быть несколько выходов с одинаковым вариантом транспортировки.
16. Связь «Gate-Flight». Таблица Gate является главной(родительской), а таблица Flight - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Gate передается индикатор GateID. Отношение один ко многим, так как у одного выхода за день могут быть несколько рейсов, но одному рейсу принадлежит только один выход.
17. Связь «Flight-Ticket». Таблица Flight является главной(родительской), а таблица Ticket - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Flight передается индикатор FlightID. Отношение один ко многим, так как в одном рейсе много пассажиров и, соответственно, много билетов, но с помощью одного билета можно попасть только на один рейс.
18. Связь «AirplaneType\_Seat-TicketPrice». Таблица AirplaneType\_Seat является главной(родительской), а таблица TicketPrice - подчиненной(дочерняя). Идентифицирующая связь. Не может быть NULL значение. Из AirplaneType\_Seat передается индикатор SeatID. Отношение один к одному, так как для одного типа рассадки(тарифа) есть только одна, общая цена.
19. Связь «TicketPrice-Ticket». Таблица TicketPrice является главной(родительской), а таблица Ticket - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из TicketPrice передается индикатор SeatID. Отношение один ко многим, так как одна цена может быть для нескольких билетов, но у одного билета только одна цена.
20. Связь «Passenger-Ticket». Таблица Passenger является главной(родительской), а таблица Ticket - подчиненной(дочерняя). Не идентифицирующая связь. Не может быть NULL значение. Из Passenger передается индикатор PassengerID. Отношение один к одному, так как одному пассажиру для полёта нужен только один билет, и один билет может принадлежать только одному пассажиру.
21. Связь «Baggage-OversizedPrice». Таблица Baggage является главной(родительской), а таблица OversizedPrice - подчиненной(дочерняя). Идентифицирующая связь. Не может быть NULL значение. Из Baggage передается индикатор BaggageID. Отношение один к одному, так как у каждого багажа определенный перевес и стоимость за перевес для каждого багажа своя, в зависимости от значения перевеса.
22. Связь «OversizedPrice-Ticket». Таблица OversizedPrice является главной(родительской), а таблица Ticket - подчиненной(дочерняя). Не идентифицирующая связь. Может быть NULL значение(если у пассажира не было багажа). Из OversizedPrice передается индикатор BaggageID. Отношение один к одному, так как у каждого багажа определенный перевес, либо его отсутствие и стоимость за перевес для каждого багажа своя, в зависимости от значения перевеса.
23. Связь «Ticket - Revenue». Таблица Ticket является главной(родительской), а таблица Revenue - подчиненной(дочерняя). Идентифицирующая связь. Не может быть NULL значение. Из Ticket передается индикатор TicketID. Отношение один к одному, так как один билет оплачивается один раз, и конкретный счет принадлежит только одному билету.

## Логическая модель базы данных

Изображение выглядит как снимок экрана, внутренний, компьютер, ноутбук

Автоматически созданное описание

Рисунок . Даталогическая модель

# Глава 3

# Реализация базы данных

С помощью MS SQL была построена диаграмма базы данных. Она заполнялась при помощи поиска достоверной информации в интернете и

генерации данных с помощью Jupiter Notebook. Jupiter часто используется для работы с данными и статистическим моделированием.

## Диаграмма базы данных

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3. Физическая модель

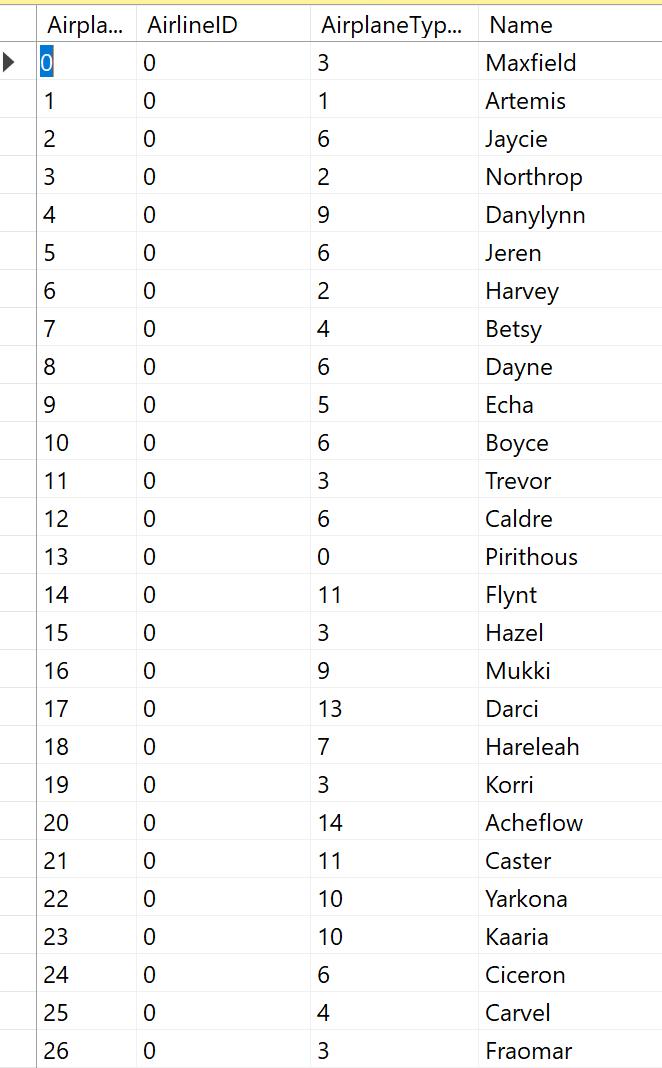
## Примеры заполнения данных

Для заполнения данных были использованы ресурсы интернета, сайты со списками стран, городов, аэропортов, авиалиний. Так же была произведена генерация данных для таких таблиц, как человек, работник и т.д с помощью Notebook Jupiter. Примеры заполнения нескольких таблиц начальными данными).

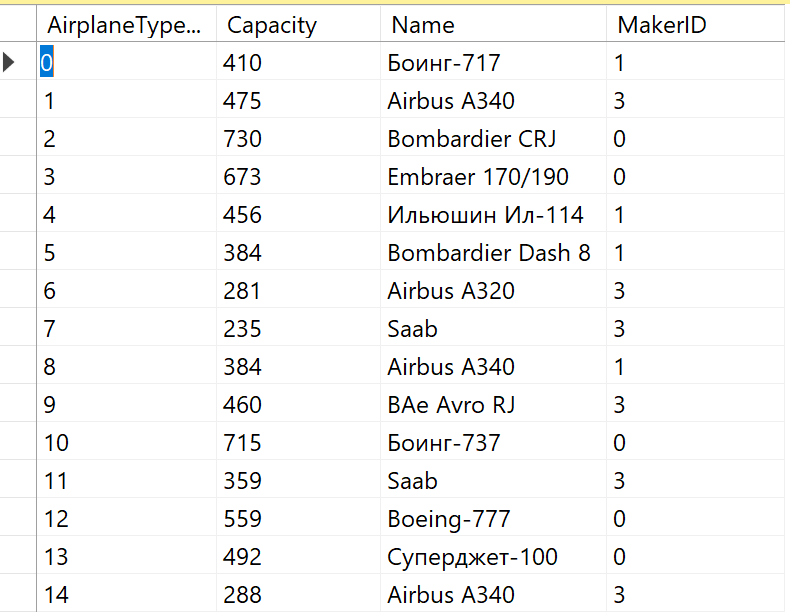
1)Airline



2)Airplane



3)AirplaneType



4)Airport

Изображение выглядит как снимок экрана, компьютер

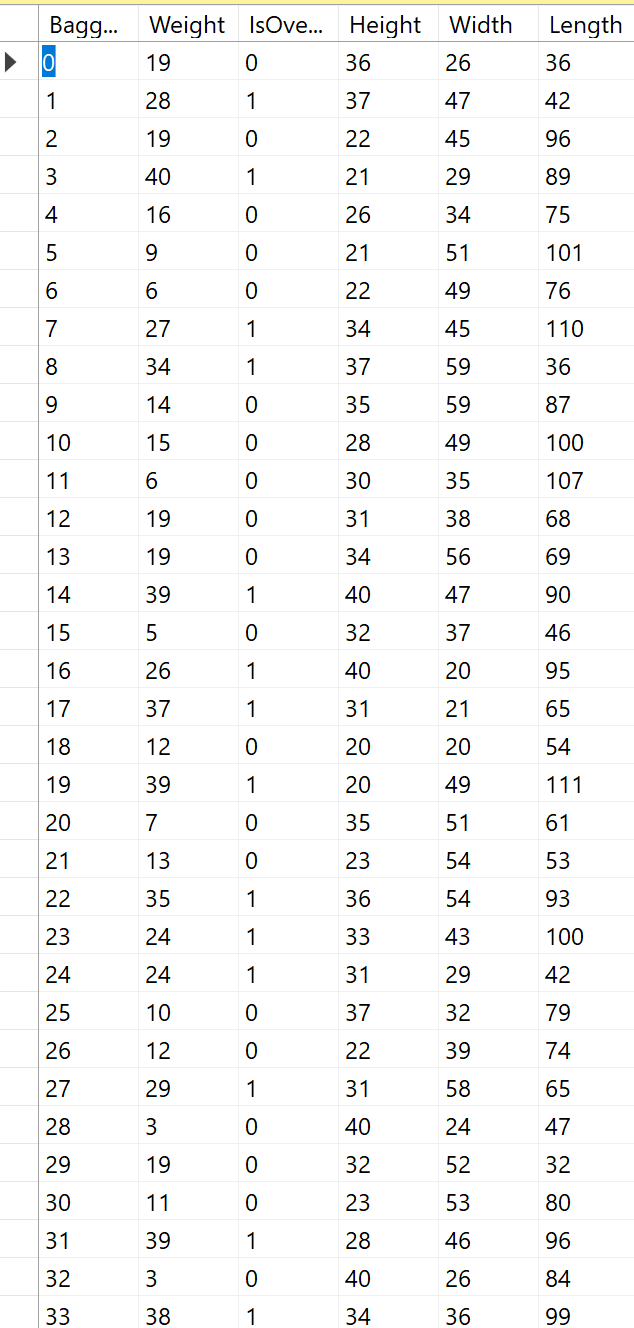
Автоматически созданное описание

5)City

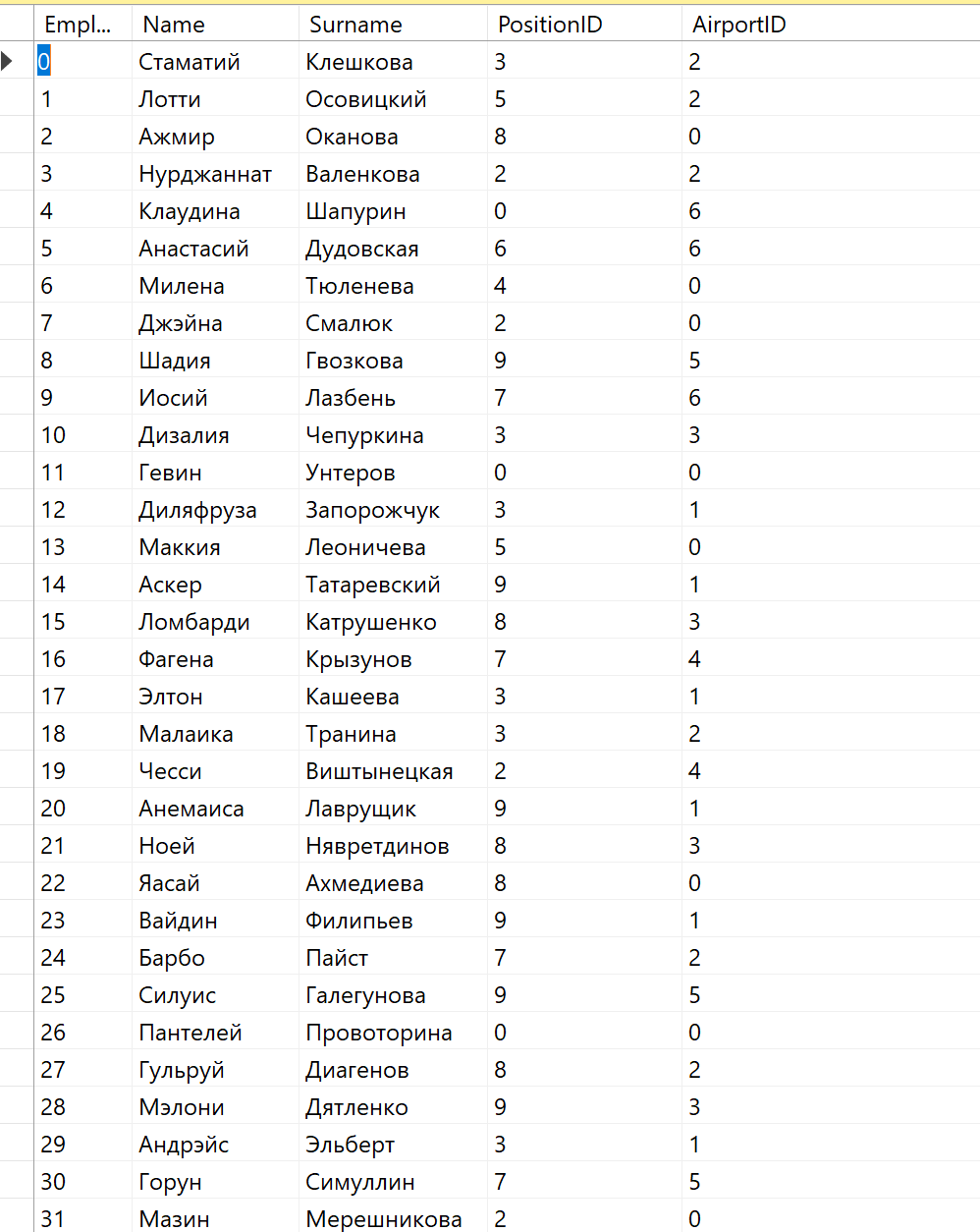
Изображение выглядит как компьютер, стол, белый, рабочий стол

Автоматически созданное описание

6)Baggage



7)Employee

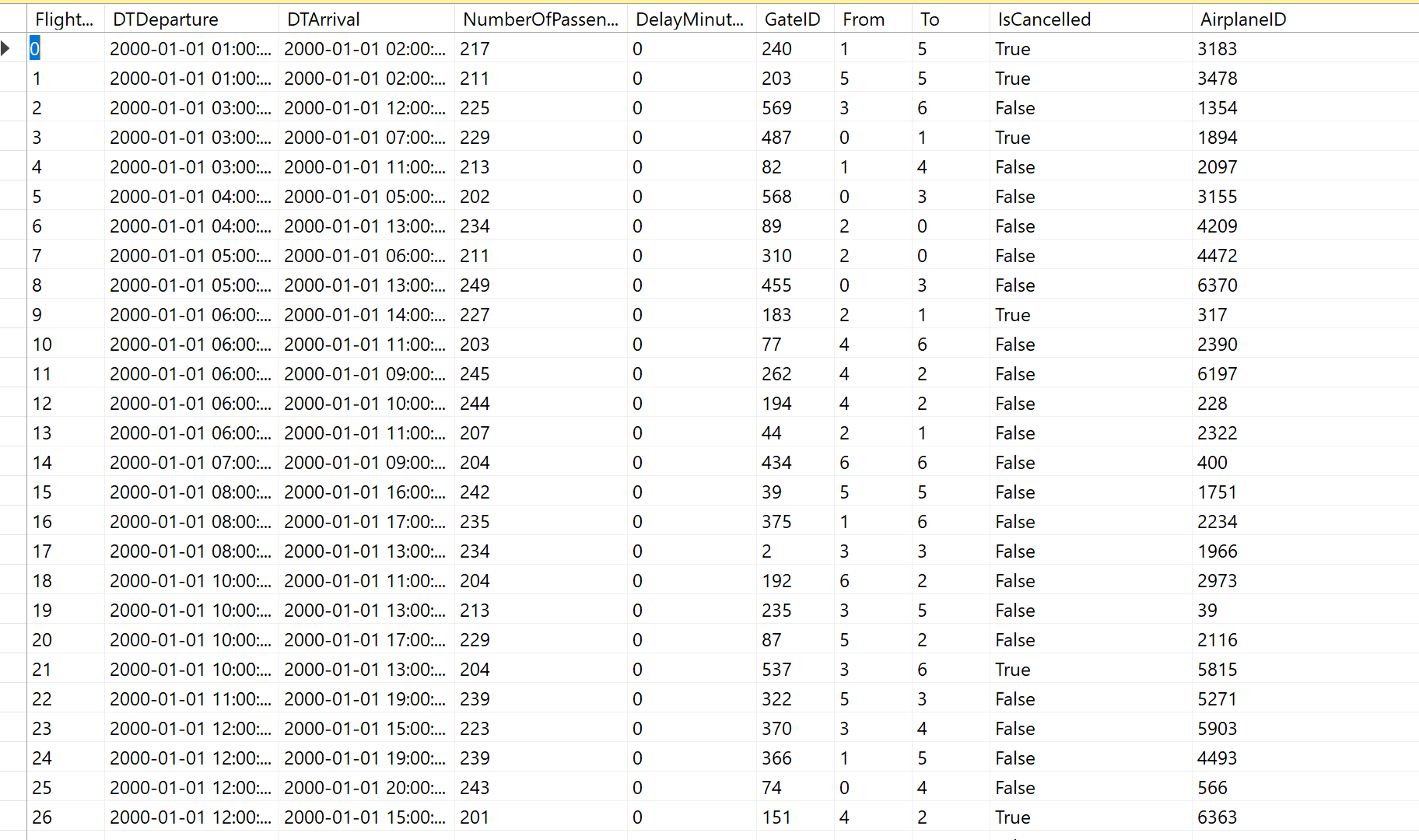


8)Country

Изображение выглядит как снимок экрана, белый, город, улица

Автоматически созданное описание

9)Flight

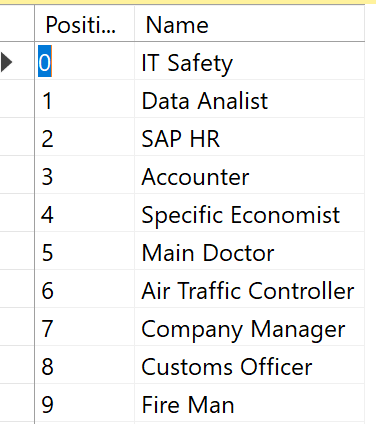


10)Passenger

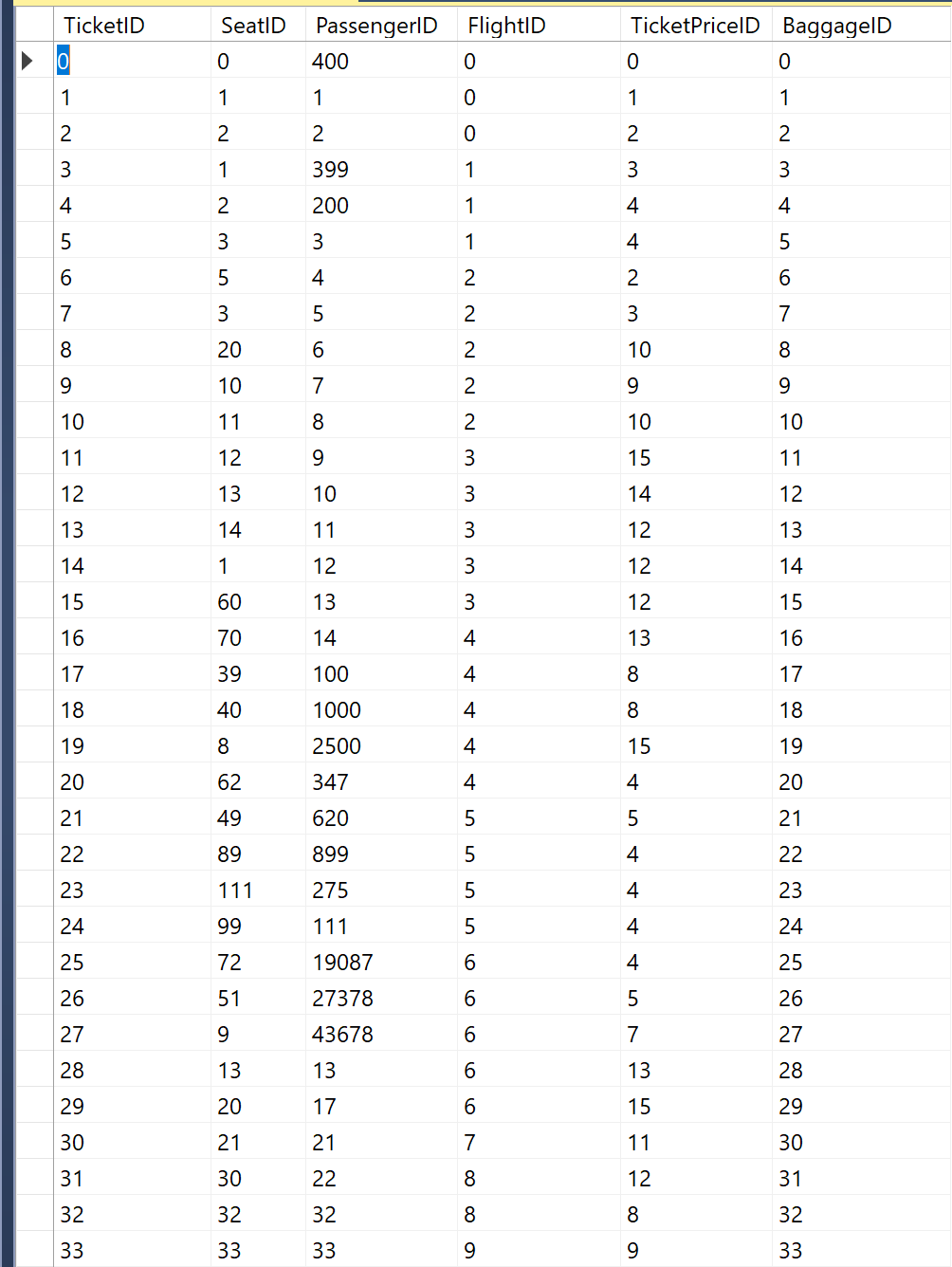
Изображение выглядит как текст

Автоматически созданное описание

11)Position



12)Ticket



13)TypeOfGate

Изображение выглядит как снимок экрана

Автоматически созданное описание

14)Revenue

Изображение выглядит как белый, деревянный, компьютер, люди

Автоматически созданное описание

Всего было создано и заполнено 24 таблицы. В некоторых данные были сгенерированы, в других взяты из реальных данных, в нескольких заполнены вручную.

## Примеры разработки запросов в базе данных

1. Количество сверхгабаритного багажа у пассажира (2 примера для визуализации большого количества данных)

Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как снимок экрана

Автоматически созданное описание

2)Количество билетов в каждом классе

Изображение выглядит как снимок экрана

Автоматически созданное описание

3)Количество сотрудников, занимающие одинаковые должности

Изображение выглядит как снимок экрана

Автоматически созданное описание

4)Количество состоявшихся рейсов из аэропортов.

Изображение выглядит как снимок экрана

Автоматически созданное описание

5)Выручка авиакомпаний за продажу билетов на различные рейсы

Изображение выглядит как снимок экрана

Автоматически созданное описание

## Примеры разработки процедур

1)Процедура, созданная с целью изменить время задержки рейса на указанное число минут.

Тело процедуры :

Изображение выглядит как снимок экрана

Автоматически созданное описание

Результат выполнения процедуры на примере 3х рейсов :

Изображение выглядит как снимок экрана, компьютер

Автоматически созданное описание

2)Процедура изменения и обновления разрешенного веса багажа

Тело процедуры:

Изображение выглядит как снимок экрана

Автоматически созданное описание

Исходные данные:

Изображение выглядит как снимок экрана

Автоматически созданное описание

Результат выполнения процедуры на примере багажа 3х пассажиров :

Изображение выглядит как снимок экрана

Автоматически созданное описание

## Примеры разработки триггеров

1)Триггер, удаляющий билеты из таблицы Revenue при удалении рейса.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Исходные данные таблиц Flight :

Изображение выглядит как снимок экрана, стол, компьютер

Автоматически созданное описание

Данные таблиц Flight и Revenue после удаления одного рейса:

Изображение выглядит как снимок экрана, компьютер

Автоматически созданное описание

Изображение выглядит как черный, белый, люди, город

Автоматически созданное описание

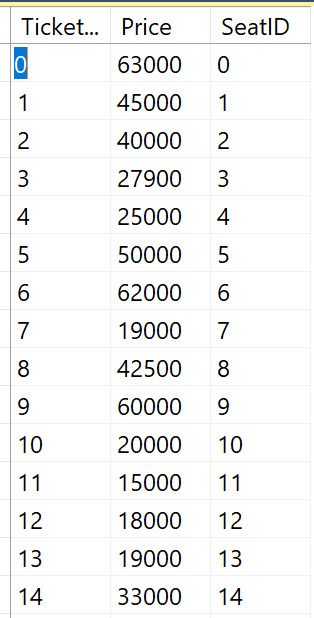
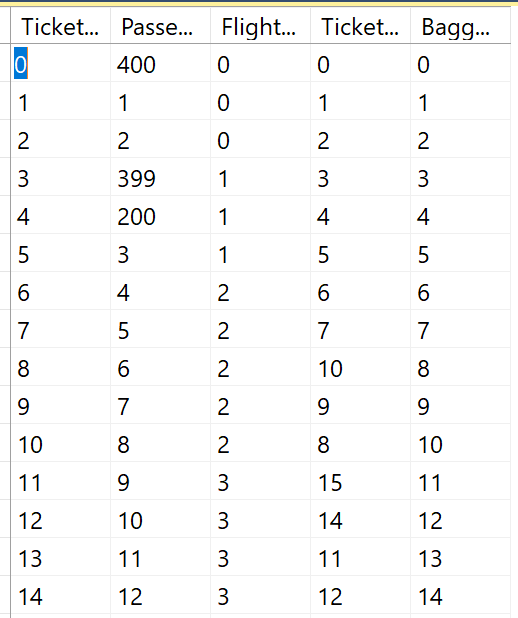
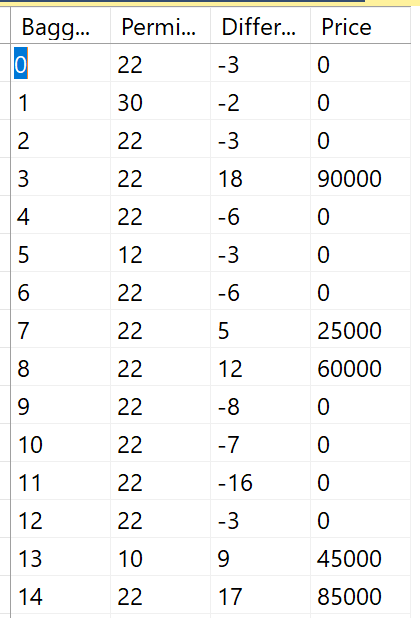
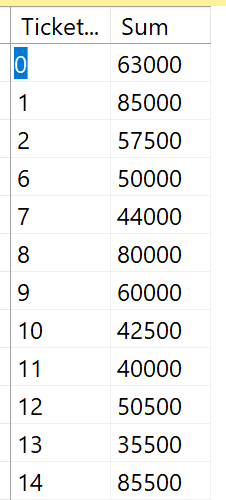
Примечание: При удалении FlightID = 1 были удалены все 3 билета, проданные на этот рейс.

2)Триггер, меняющий значение sum в таблице Revenue после изменения цены билета.

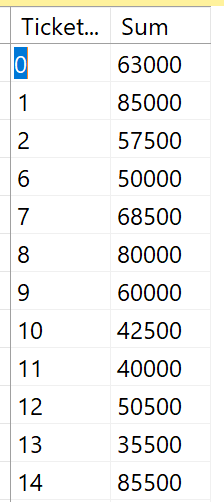
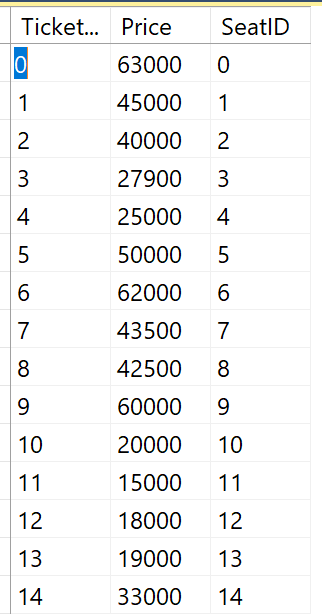
Изображение выглядит как снимок экрана

Автоматически созданное описание

Исходные данные таблицы Revenue и Ticket и TicketPrice:

Результат выполнения триггера: Изменили TicketPriceID = 7



Примечание: Как мы видим, после изменения Ticket Price - значение Sum в Revenue автоматически изменилось, как TicketPrice.Price + OversizedPrice.Price

3)Триггер, меняющий цену за перевес багажа при изменении разрешенного веса

Изображение выглядит как снимок экрана

Автоматически созданное описание

Исходные данные:

Изображение выглядит как снимок экрана

Автоматически созданное описание

Результат работы триггера на примере багажа 3х пассажиров:

Изображение выглядит как снимок экрана

Автоматически созданное описание

## Примеры разработки функций

1. Функция считает выручку авиакомпании с полета

Тело функции:

Изображение выглядит как снимок экрана

Автоматически созданное описание

Результат выполнения функции:

Изображение выглядит как снимок экрана

Автоматически созданное описание

Примечание: FlightID = 1 имеет нулевое значение, так как рейс был удалён.

2)Функция, которая считает количество пассажиров на рейсе

Тело функции:

Изображение выглядит как снимок экрана

Автоматически созданное описание

Результат выполнения функции:

Изображение выглядит как снимок экрана

Автоматически созданное описание

## Примеры разработки представлений

1. Представление с количеством перелетов между аэропортами

Изображение выглядит как снимок экрана

Автоматически созданное описание

Результат выполнения представления:

Изображение выглядит как снимок экрана

Автоматически созданное описание

2)Представление считает выручку по авиакомпании

Изображение выглядит как снимок экрана

Автоматически созданное описание

Результат представления представления:

Изображение выглядит как снимок экрана

Автоматически созданное описание

# Глава 4

# Приложение для базы данных «Аэропорт»

Приложение было создано с целью облегчить работу с базой данных со стороны пользователя. Пользователь имеет возможность посмотреть актуальную информацию о прилётах и вылетах, о расписании рейсов, найти необходимую информацию об авиалиниях, с которыми работает данный аэропорт.

Главное окно, в котором пользователь выбирает необходимый ему вид информации:

Изображение выглядит как снимок экрана, телефон, монитор, экран

Автоматически созданное описание

Окно расписания рейсов:

Изображение выглядит как снимок экрана

Автоматически созданное описание

Табло вылета:

Изображение выглядит как снимок экрана, монитор, экран

Автоматически созданное описание

Табло прилёта:

Изображение выглядит как снимок экрана, монитор, экран, держит

Автоматически созданное описание

Окно выбора рейса в расписании:

Изображение выглядит как снимок экрана, монитор, черный, сидит

Автоматически созданное описание

Полный функционал приложения и его взаимодействие с базой данных аэропорт будет продемонстрировано непосредственно на презентации.

# Заключение

Подводя итоги проделанной работы, можно выделить следующие основные моменты курсовой работы:

1. Выбрана и описана предметная область Аэропорт;
2. Разработана и реализована инфологическая модель данных;
3. Разработана и реализована даталогическая модель данных;
4. Выявлены и описаны сущности базы данных, атрибуты сущностей, а также описаны связи между таблицами базы данных;
5. Построена даталогическая модель в ERwin;
6. Создана физическая модель в MS SQL Server на основе даталогической модели;
7. Заполнение данных таблиц с помощью их генерации в Notebook Jupiter, найдены реальные данные на ресурсе Kaggle, а так же некоторые таблицы заполнялись в ручную.
8. Созданы запросы, представления, хранимые процедуры, функции и триггеры;
9. Разработано web-приложение на языке С# на основе базы данных базы данных «Аэропорт».

# Список используемой литературы

1. Грофф, Джеймс; Вайнберг, Пол SQL: полное руководство; Киев: BHV, 2005. - 608 c.
2. Ишкова, Э. А. Самоучитель С#. Начала программирования / Э.А. Ишкова. - М.: Наука и техника, 2013. - 496 c.
3. Иванов, Ю.П. BPwin и ERwin. CASE-средства проектирования информационных систем: учебное пособие / Ю.П. Иванов, Е.В. Федоренко. – М.: Находка, 2008. – 80 с.
4. Грубер М. Понимание SQL .1993.
5. Базы данных. Проектирование и создание. Диго С.М . М.: ЕАОИ, 2008.
6. Молинаро Э. SQL. Сборник рецептов. – Пер. с англ. – СПб: Символ!Плюс, 2009. – 672 с., ил.