Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

“Российский экономический университет имени Г.В. Плеханова”

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

Специальность: 09.02.07 Информационные системы и программирование  
Квалификация: программист

ПРАКТИЧЕСКИЕ РАБОТЫ ПО ДИСЦИПЛИНЕ

«ОСНОВЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ»

Листов:9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент  Группы: П50-4-21  Игошев Ростислав Вадимович |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_А.В. Архангельский  «\_\_\_\_\_\_» \_\_\_\_\_\_2023 года |

Москва 2023

ПРАКТИЧЕСКАЯ РАБОТА №2

Тема: «Физические и логические модели данных»

Цель: Научиться работать с физическими и логическими моделями данных. Выбрать себе предметную область и записать ее в таблицу (повторяющихся быть не должно). Создать data и info-логические модели к выбранной предметной области.

Для начала, разберем и создадим логическую модель данных, ибо по ней проще ориентироваться обычному человеку.

Логическая модель:

В данной базе данных присутствует основная таблица, от которой идут все связи к остальным, образуя связь один ко многим.

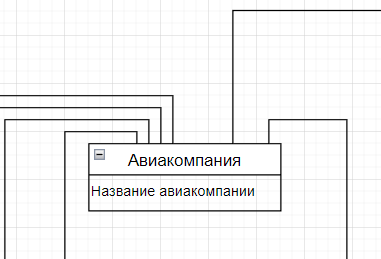


Рисунок 1 – Основная таблица

Первая ветка, которая исходит от основной таблицы – аэропорты. Тут присутствует различная информация о аэропортах, что ясно из названия.

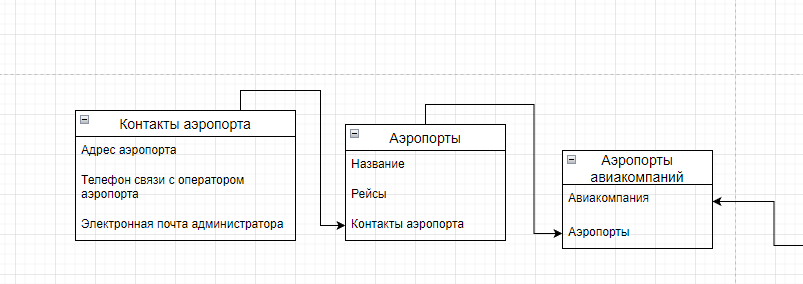


Рисунок 2 - Ветка аэропортов

Следующая ветка – самолеты. Здесь описаны характеристики каждого отдельного самолета.

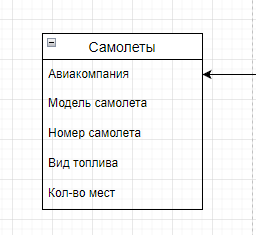


Рисунок 3 - Ветка самолетов

Третья ветка – вип-персоны. Здесь описана информация о вип-персонах в авиакомпаниях, их статус, приоритет, личная информация.

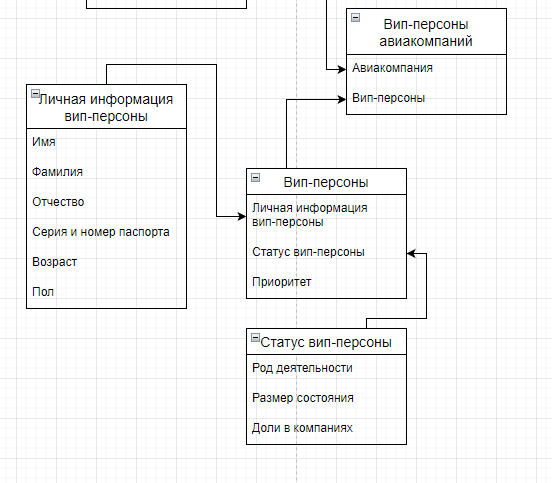


Рисунок 4 - Ветка вип-персон

Далее – ветка сотрудников. Здесь описана их личная информация, служебное положение, контактные данные.

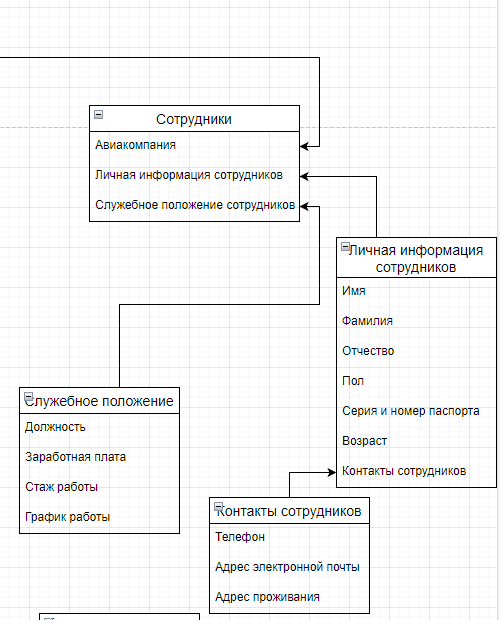


Рисунок 5 - Ветка сотрудников

Последние две ветки – ветка дочерних компаний и ветка турагентств – партнеров. Здесь описаны названия, рейсы и прочая информация о данных отделах.

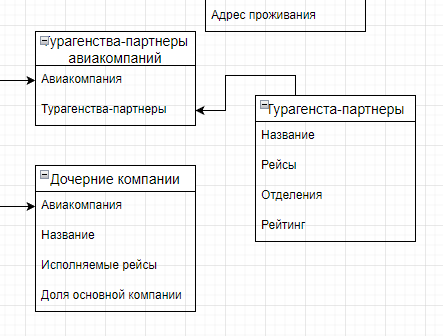


Рисунок 6 - Ветка дочерних компаний и ветка турагентств-партнеров

В целом, наша конечная модель выглядит так:

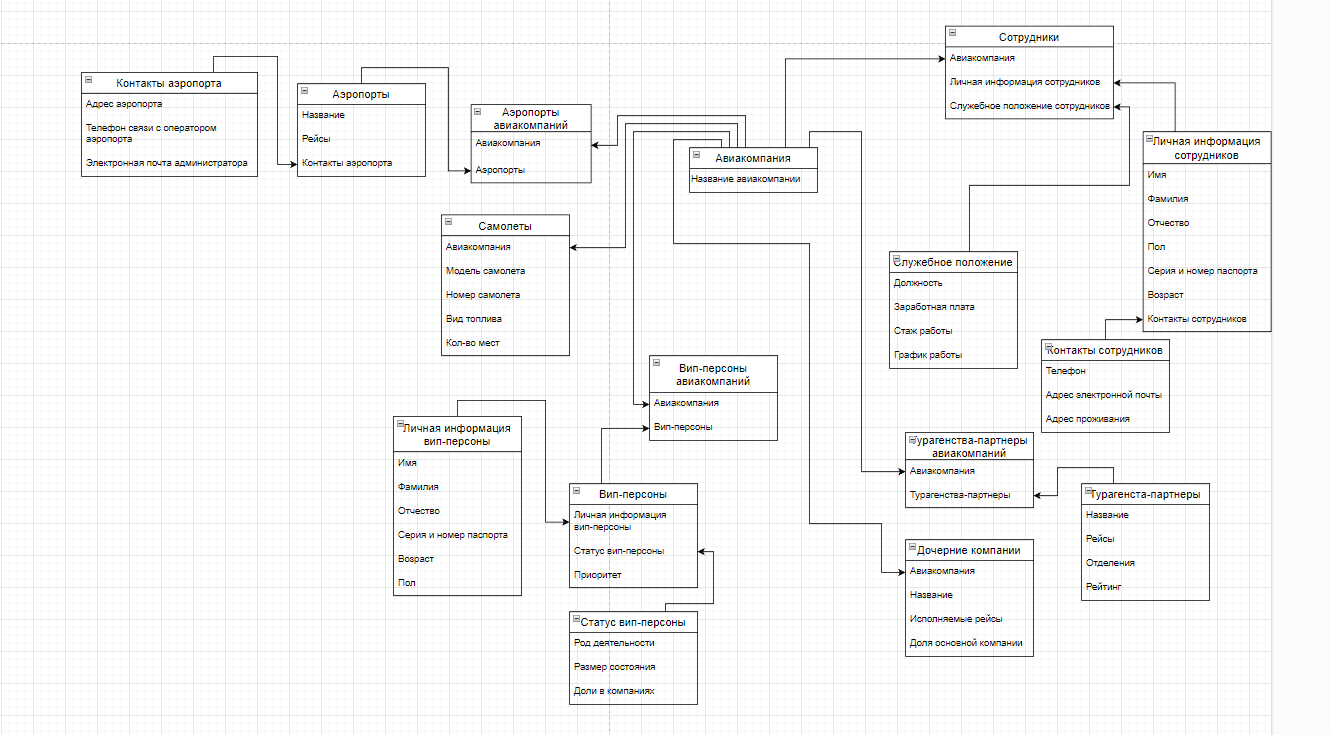


Рисунок 7 - Целая итоговая модель данных

Физическая модель:

Теперь, создадим физическую модель на основе логической. Отличается физическая тем, что названия должны быть понятны СУБД, что расшифровывается как Система Управления Базами Данных. Грубо говоря, названия должны быть на английском, применяться в стиле snake\_case и указывать в себе тип данных с ограничением и возможностью пустых ячеек, то-есть значения NULL. Кроме того, связь должна быть организована между id таблиц.

Первая таблица выглядит так же, за исключением того, что в ней появилась ячейка id и поменялся формат записи.

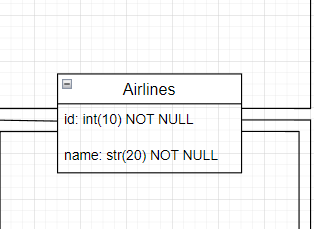


Рисунок 8 - Основная таблица физической модели

Первая ветка:

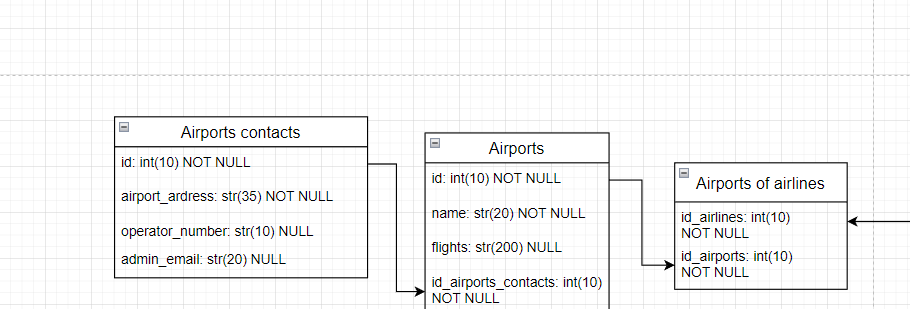


Рисунок 9 - Ветка Airports

Ветка самолетов:

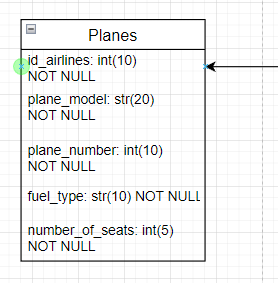


Рисунок 10 - Ветка Planes

Ветка вип-персон:

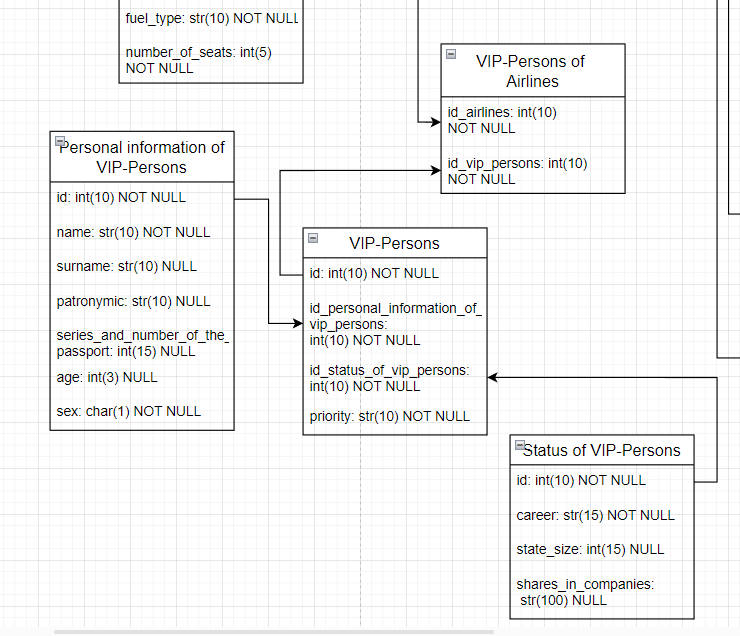


Рисунок 11 - Ветка VIP – Persons

Ветка рабочих:

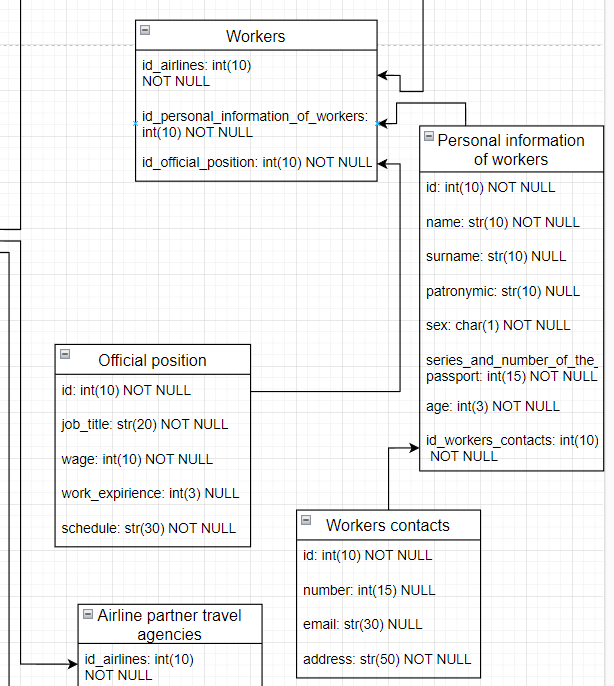


Рисунок 12 - Ветка Workers

Ветки турагентств и дочерних компаний:

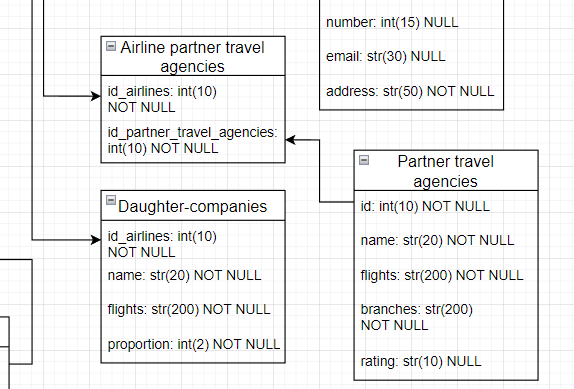


Рисунок 13 - Ветка Daughter-companies и ветка Airline partner travel agencies

После таких изменений, видим следующую картину:

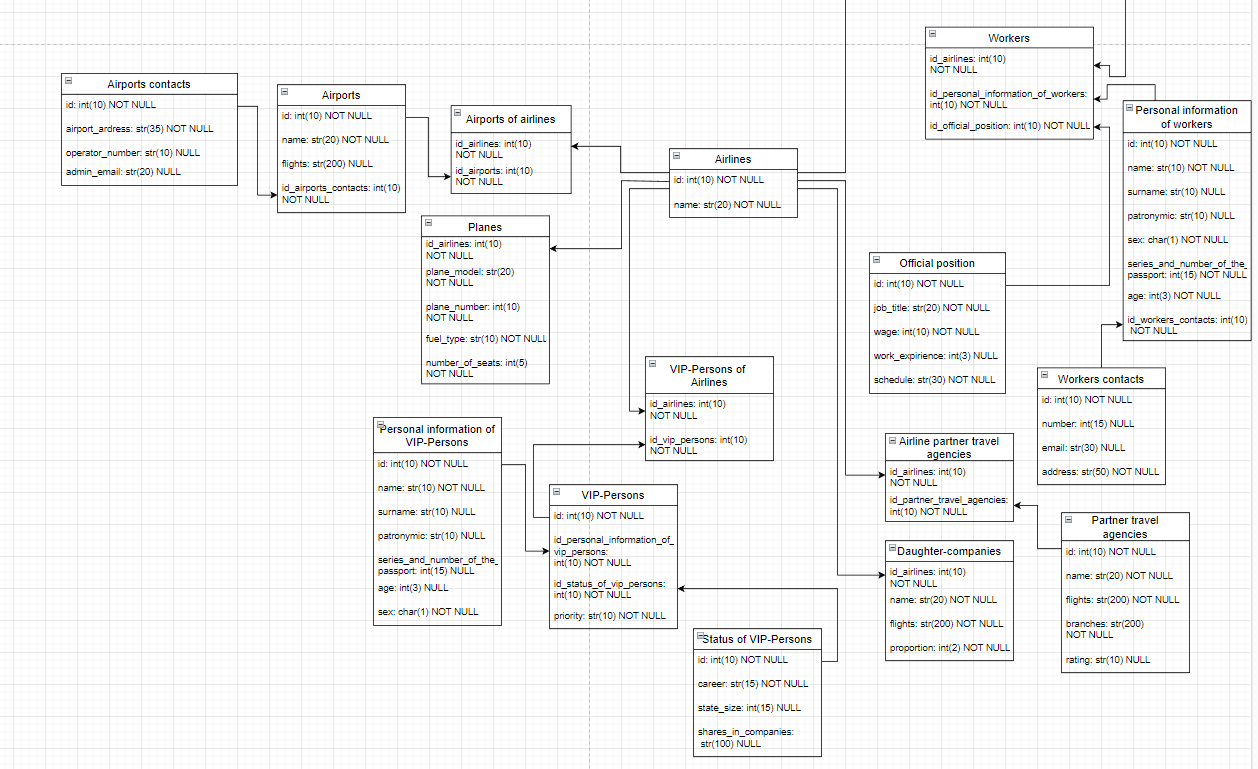


Рисунок 14 - Целая физическая модель

Вывод: Научились работать с физическими и логическими моделями данных. Выбрали себе предметную область и записали ее в таблицу (повторяющихся быть не должно). Создали data и info-логические модели к выбранной предметной области.

ПРАКТИЧЕСКАЯ РАБОТА №3

Тема: «Создание БД»

Цель работы: создать базу данных при помощи SQL запросов на основе физической и логической моделей из предыдущей практической.

Ссылка на код работы:

<https://drive.google.com/file/d/1CMfxVjPVC4FOrupjGv2R7JQHTYoTuoa/view>.

Начнем создавать нашу БД. Для взаимодействия с ней следует зарегистрировать суперпользователя. Попробуем создать БД следующей командой запросов:



Рисунок 15 – Создание БД

У БД есть несколько параметров, которые определяют ее характеристики и особенности взаимодействия с ней. Вот они:

* Logical Name: логическое имя, которое присваивается файлу базы данных.
* File Type: есть несколько типов файлов, но, как правило, основная работа ведется с файлами данных (ROWS Data) и файлом лога (LOG)
* Filegroup: обозначает группу файлов. Группа файлов может хранить множество файлов и может использоваться для разбиения базы данных на части для размещения в разных местах.
* Initial Size (MB): устанавливает начальный размер файлов при создании (фактический размер может отличаться от этого значения).
* Autogrowth/Maxsize: при достижении базой данных начального размера SQL Server использует это значение для увеличения файла.
* Path: каталог, где будут храниться базы данных.
* File Name: непосредственное имя физического файла. Если оно не указано, то применяется логическое имя.

Теперь, попробуем выделить нашу созданную БД для работы с ней.



Рисунок 16 – Использование БД

Как видим, в поле активной БД отобразилась ранее нами созданная БД.



Рисунок 17 – Демонстрация работы с БД

Для создания таблицы нам следует использовать язык определения данных (DDL - Data Definition Language). Создание, редактирование и удаление информации. В эту группу входят следующие операторы:

CREATE – используется для создания объектов базы данных;

ALTER – используется для изменения объектов базы данных;

DROP – используется для удаления объектов базы данных.

Вот типы данных которые будут использоваться в этой работе:

TINYINT: хранит числа от 0 до 255. Хорошо подходит для хранения небольших чисел.

SMALLINT: хранит числа от –32 768 до 32 767.

INT: хранит числа от –2 147 483 648 до 2 147 483 647. Наиболее используемый тип для хранения чисел.

MONEY: хранит дробные значения от -922 337 203 685 477.5808 до 922 337 203 685 477.5807. Представляет денежные величины.

VARCHAR: хранит строку. Можно указать конкретную длину для столбца - от 1 до 8 000 символов.

Теперь, попробуем создать центральную таблицу нашей БД с полем и первичным ключем. Для этого нам нужно написать следующую конструкцию в случае с ID: (name) INT IDENTITY (seed, increment) PRIMARY KEY. В случае с переменной: (name) (TYPE)(limit) (NULL/NOT NULL).

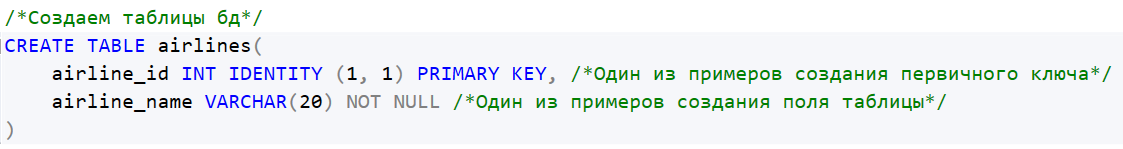


Рисунок 18 – Создание таблицы и ее полей и первичного ключа

Выделив кусок кода и запустив его, наша таблица будет создана. Чтобы регулировать связь между таблицами недостаточно лишь первичного ключа, нужен еще и вторичный. Вот пример создания одного из внешних ключей:



Рисунок 19 – Создание внешнего ключа

В следующем блоке кода используем конструкцию ALTER. Она выполняет функции изменения данных какого-либо объекта.

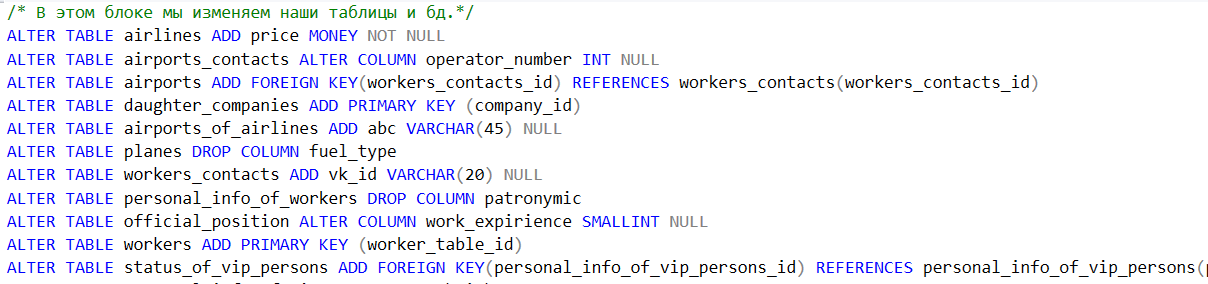


Рисунок 20 – Изменение таблиц и БД

Этот блок отвечает за удаление таблиц и баз данных.

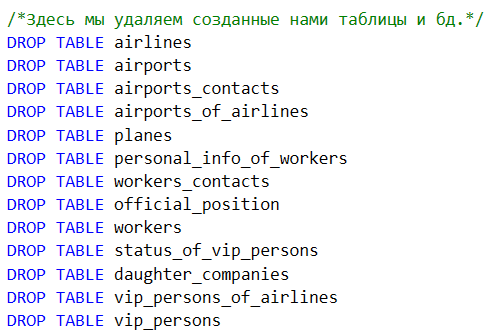


Рисунок 21 – Удаление таблиц и БД

Включив диаграмму, мы увидим следующую картину:

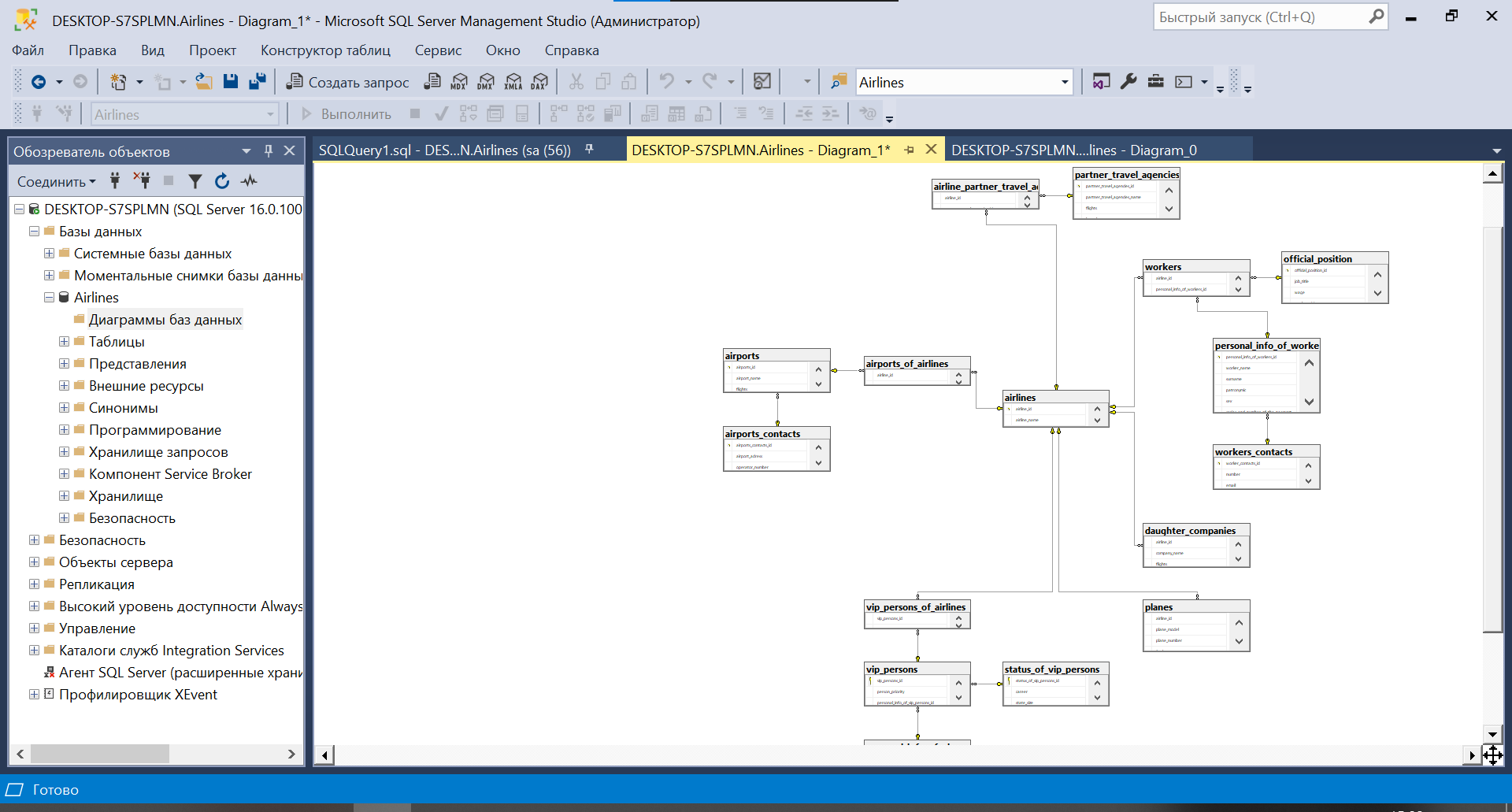


Рисунок 22 – Диаграмма БД

Как видим, наша БД была успешно создана и воспроизведена графическим образом.

Ниже представлен скрипт БД:

/\*Создаем бд\*/

CREATE DATABASE Airlines

/\*Начинаем работать с созданной бд\*/

USE Airlines

/\*Создаем таблицы бд\*/

CREATE TABLE airlines(

airline\_id INT IDENTITY (1, 1) PRIMARY KEY, /\*Один из примеров создания первичного ключа\*/

airline\_name VARCHAR(20) NOT NULL /\*Один из примеров создания поля таблицы\*/

)

CREATE TABLE airports\_contacts(

airports\_contacts\_id INT IDENTITY (1, 1) PRIMARY KEY,

airport\_adress VARCHAR(35) NOT NULL,

operator\_number VARCHAR(10) NULL,

admin\_email VARCHAR(20) NULL

)

CREATE TABLE airports(

airports\_id INT IDENTITY (1,1) PRIMARY KEY,

airport\_name VARCHAR(20) NOT NULL,

flights VARCHAR(200) NULL,

airports\_contacts\_id INT,

FOREIGN KEY (airports\_contacts\_id) /\*Указание поля, использующегося в качестве внешнего ключа\*/

REFERENCES airports\_contacts(airports\_contacts\_id) /\*Указание расположения первичного ключа для образования связи\*/

)

CREATE TABLE airports\_of\_airlines(

airline\_id INT,

airports\_id INT,

FOREIGN KEY (airline\_id)

REFERENCES airlines(airline\_id),

FOREIGN KEY (airports\_id)

REFERENCES airports(airports\_id)

)

CREATE TABLE planes(

airline\_id INT,

plane\_model VARCHAR(20) NOT NULL,

plane\_number INT NOT NULL,

fuel\_type VARCHAR(10) NOT NULL,

number\_of\_seats SMALLINT NOT NULL

FOREIGN KEY (airline\_id)

REFERENCES airlines(airline\_id),

)

CREATE TABLE workers\_contacts(

worker\_contacts\_id INT IDENTITY (1, 1) PRIMARY KEY,

number INT NULL,

email VARCHAR(30) NULL,

adress VARCHAR(50) NOT NULL

)

CREATE TABLE personal\_info\_of\_workers(

personal\_info\_of\_workers\_id INT IDENTITY (1, 1) PRIMARY KEY,

worker\_name VARCHAR(10) NOT NULL,

surname VARCHAR (10) NULL,

patronymic VARCHAR(10) NULL,

sex VARCHAR(1) NOT NULL,

series\_and\_number\_of\_the\_passport INT NOT NULL,

age INT NOT NULL,

worker\_contacts\_id INT,

FOREIGN KEY (worker\_contacts\_id)

REFERENCES workers\_contacts(worker\_contacts\_id),

)

CREATE TABLE official\_position(

official\_position\_id INT IDENTITY (1, 1) PRIMARY KEY,

job\_title VARCHAR(20) NOT NULL,

wage INT NOT NULL,

work\_expirience TINYINT NULL,

schedule VARCHAR(30) NOT NULL

)

CREATE TABLE workers(

airline\_id INT,

personal\_info\_of\_workers\_id INT,

official\_position\_id INT

FOREIGN KEY (airline\_id)

REFERENCES airlines(airline\_id),

FOREIGN KEY (personal\_info\_of\_workers\_id)

REFERENCES personal\_info\_of\_workers(personal\_info\_of\_workers\_id),

FOREIGN KEY (official\_position\_id)

REFERENCES official\_position(official\_position\_id),

)

CREATE TABLE status\_of\_vip\_persons(

status\_of\_vip\_persons\_id INT IDENTITY (1, 1) PRIMARY KEY,

career VARCHAR(15) NOT NULL,

state\_size INT NULL,

shares\_in\_companies VARCHAR(100) NULL

)

CREATE TABLE personal\_info\_of\_vip\_persons(

personal\_info\_of\_vip\_persons\_id INT IDENTITY (1, 1) PRIMARY KEY,

person\_name VARCHAR(10) NOT NULL,

surname VARCHAR(10) NULL,

patronymic VARCHAR(10) NULL,

series\_and\_number\_of\_the\_passport INT NULL,

age INT NULL,

sex VARCHAR(1) NOT NULL

)

CREATE TABLE vip\_persons(

vip\_persons\_id INT IDENTITY (1, 1) PRIMARY KEY,

person\_priority VARCHAR(10) NOT NULL,

personal\_info\_of\_vip\_persons\_id INT,

status\_of\_vip\_persons\_id INT,

FOREIGN KEY (status\_of\_vip\_persons\_id)

REFERENCES status\_of\_vip\_persons(status\_of\_vip\_persons\_id),

FOREIGN KEY (personal\_info\_of\_vip\_persons\_id)

REFERENCES personal\_info\_of\_vip\_persons(personal\_info\_of\_vip\_persons\_id),

)

CREATE TABLE vip\_persons\_of\_airlines(

vip\_persons\_id INT,

airline\_id INT,

FOREIGN KEY (vip\_persons\_id)

REFERENCES vip\_persons(vip\_persons\_id),

FOREIGN KEY (airline\_id)

REFERENCES airlines(airline\_id)

)

CREATE TABLE daughter\_companies(

airline\_id INT,

company\_name VARCHAR(20) NOT NULL,

flights VARCHAR(200) NOT NULL,

proportion INT NOT NULL,

FOREIGN KEY (airline\_id)

REFERENCES airlines(airline\_id)

)

CREATE TABLE partner\_travel\_agencies(

partner\_travel\_agencies\_id INT IDENTITY (1, 1) PRIMARY KEY,

partner\_travel\_agencies\_name VARCHAR(20) NOT NULL,

flights VARCHAR(200) NOT NULL,

branches VARCHAR(200) NOT NULL,

rating VARCHAR(10) NULL

)

CREATE TABLE airline\_partner\_travel\_agencies(

airline\_id INT,

partner\_travel\_agencies\_id INT,

FOREIGN KEY (airline\_id)

REFERENCES airlines(airline\_id),

FOREIGN KEY (partner\_travel\_agencies\_id)

REFERENCES partner\_travel\_agencies(partner\_travel\_agencies\_id)

)

/\* В этом блоке мы изменяем наши таблицы и бд.\*/

ALTER TABLE airlines ADD price MONEY NOT NULL

ALTER TABLE airports\_contacts ALTER COLUMN operator\_number INT NULL

ALTER TABLE airports ADD FOREIGN KEY(workers\_contacts\_id) REFERENCES workers\_contacts(workers\_contacts\_id)

ALTER TABLE daughter\_companies ADD PRIMARY KEY (company\_id)

ALTER TABLE airports\_of\_airlines ADD abc VARCHAR(45) NULL

ALTER TABLE planes DROP COLUMN fuel\_type

ALTER TABLE workers\_contacts ADD vk\_id VARCHAR(20) NULL

ALTER TABLE personal\_info\_of\_workers DROP COLUMN patronymic

ALTER TABLE official\_position ALTER COLUMN work\_expirience SMALLINT NULL

ALTER TABLE workers ADD PRIMARY KEY (worker\_table\_id)

ALTER TABLE status\_of\_vip\_persons ADD FOREIGN KEY(personal\_info\_of\_vip\_persons\_id) REFERENCES personal\_info\_of\_vip\_persons(personal\_info\_of\_vip\_persons\_id)

ALTER TABLE personal\_info\_of\_vip\_persons ADD height INT NULL

ALTER TABLE vip\_persons DROP COLUMN person\_priority

ALTER TABLE vip\_persons\_of\_airlines ADD PRIMARY KEY (vip\_id)

ALTER TABLE partner\_travel\_agencies ALTER COLUMN rating REAL NULL

ALTER TABLE airline\_partner\_travel\_agencies ADD idk VARCHAR(1000) NULL

ALTER DATABASE Airlines MODIFY FILE (NAME = "Airline", SIZE = 100MB, FILEGROWTH = 5%)

/\*Здесь мы удаляем созданные нами таблицы и бд.\*/

DROP TABLE airlines

DROP TABLE airports

DROP TABLE airports\_contacts

DROP TABLE airports\_of\_airlines

DROP TABLE planes

DROP TABLE personal\_info\_of\_workers

DROP TABLE workers\_contacts

DROP TABLE official\_position

DROP TABLE workers

DROP TABLE status\_of\_vip\_persons

DROP TABLE daughter\_companies

DROP TABLE vip\_persons\_of\_airlines

DROP TABLE vip\_persons

DROP TABLE airline\_partner\_travel\_agencies

DROP TABLE partner\_travel\_agencies

DROP TABLE personal\_info\_of\_vip\_persons

DROP DATABASE Airlines

Вывод: создали базу данных при помощи SQL запросов на основе физической и логической моделей из предыдущей практической.