**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ**

**по дисциплине «Базы данных»**

**Тема: Проектирование базы данных**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1308 |  | Мальцев А.П. |
| Преподаватель |  | Новакова Н.Е. |

Санкт-Петербург

2023

**ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ**

|  |  |  |
| --- | --- | --- |
| Студент Мальцев А.П. | | |
| Группа 1308 | | |
| Тема работы: Проектирование базы данных | | |
| Исходные данные:  Разработать БД сотрудников организации/предприятия. | | |
| Содержание пояснительной записки: «Содержание», «Введение», «Задание», «Предметная область», «Проектирование базы данных», «Содержание базы данных», «Создание таблиц», «Заполнение таблиц данными», «Разработка объектов промежуточного слоя», «Разработка стратегии резервного копирования», «Заключение», «Список использованных источников». | | |
| Предполагаемый объем пояснительной записки:  Не менее 20 страниц. | | |
| Дата выдачи задания: 10.10.2023 | | |
| Дата сдачи задания: 00.12.2023 | | |
| Дата защиты задания: 00.12.2023 | | |
| Студент |  | Мальцев А.П. |
| Преподаватель |  | Новакова Н.Е. |

**АННОТАЦИЯ**

Целью данного индивидуального домашнего задания является проектирование и создание базы данных в рамках выбранной предметной области. Также частью проекта является разработка объектов промежуточного слоя (хранимые процедуры, UDF, представления). Также в рамках работы предлагается стратегия резервного копирования базы данных.

**SUMMARY**

The purpose of this individual homework is designing and creating a database within the chosen subject area. Second part of the project is the development of intermediate layer objects (stored procedures, UDFs, views). Also it is suggesting a database backup strategy.

Оглавление

[1. ВВЕДЕНИЕ 5](#_Toc153408131)

[2. Задание 6](#_Toc153408132)

[3. Краткое описание предметной области 7](#_Toc153408133)

[4. Проектирование базы данных 8](#_Toc153408134)

[5. Создание базы данных 17](#_Toc153408135)

[6. Создание таблиц базы данных 17](#_Toc153408136)

[7. Заполнение таблиц данными 20](#_Toc153408137)

[8. Разработка объектов промежуточного слоя 23](#_Toc153408138)

[9. Резервное копирование 32](#_Toc153408139)

[10. Вывод 35](#_Toc153408140)

[11. Список использованных источников 36](#_Toc153408141)

# ВВЕДЕНИЕ

В данной работе была спроектирована и создана база данных (БД) «Строительный магазин», предназначенная для хранения и управления сведений о магазинах, отделах, товарах, складах.

В БД предусмотрено, что отдел имеет состояние работы (открытый или закрытый), каждый товар имеет дату поступления в магазин, каждый магазин закреплен за одним складом, одинаковые товары могут продаваться в разных отделах.

Также в процессе выполнения работы были созданы объекты промежуточного слоя (представления, хранимые процедуры, UDF-ы), а также выбрана стратегия резервного копирования.

# Задание

**Цель работы**

Закрепить теоретические знания, полученные на курсе «Базы данных», и получить практические навыки в проектировании и создании базы данных.

Задачи

Для работы была выбрана база данных «Строительный магазин». Необходимо выполнить следующие задания:

1. Краткое описание предметной области;
2. Спроектировать БД (структура данных);
3. Создать БД;
4. Создать таблицы и ограничения целостности;
5. Заполнить таблицы данными;
6. Создать объекты промежуточного слоя (представления, хранимые процедуры, UDF-ы);
7. Разработка стратегии резервного копирования;

**Формализация задания**

Спроектировать базу данных для строительного магазина, используя метод ER-диаграмм, создать базу данных и необходимые таблицы, задать ограничения для таблиц и ключи, заполнить базу данных и создать в базе данных объекты промежуточного слоя.

**Формализация задачи**

Необходимо спроектировать базу данных, содержащую таблицы со сведениями о магазинах, отделах, товарах и торговых базах.

# Краткое описание предметной области

В БД должны храниться сведения о строительных магазинах города; товарах, имеющихся в магазинах; о складах и товарах, хранящихся на складах. Каждый магазин закреплён за одной торговой базой. Магазин характеризуется номером и имеет несколько отделов. Отдел характеризуется товаром. Каждый товар в каждом магазине продаётся, по крайней мере, в одном отделе. Товары, имеющиеся в магазине, характеризуются ценой, сортом и количеством. Розничные цены в магазине зависят от класса магазина и сорта товара и могут изменяться. Магазин может открыть новый отдел или закрыть старый. В этом случае товар передаётся в другие отделы.

БД «Строительный магазин» проектируем при условии, что:

– каждый магазин закреплён за одной торговой базой;

– одинаковые товары могут продаваться в разных отделах;

– каждый отдел имеет состояние (закрыт или открыт);

– у каждого товара есть дата поступления в магазин.

# Проектирование базы данных

В базе данных представлены следующие таблицы:

1. Warehouse — таблица, хранящая основную информацию о складах. Таблица хранит информацию об номере склада, ее название, ФИО управляющего и примечание. Первичный ключ — warehouse \_number.
2. Shop — таблица, хранящая основную информацию о каждом магазине, а именно: номер магазина, номер торговой базы за которой он закреплен, название, адрес, фио директора и примечания. Первичный ключ — shop\_number. Внешним ключом является base\_number.
3. Departments — таблица, содержащая информацию о отделах, номер магазина в котором находится отдел, класс, состояние и примечание. Первичный ключ — department\_number. Внешним ключом является shop\_number.
4. Product\_d — таблица, содержащая информацию о товарах в отделах. Содержит столбцы с информацией о шифре, название, сорт, дату поступления, цену и примечание. Первичный ключ — code.
5. Product\_w— таблица, содержащая информацию о товарах на складах. Содержит столбцы с информацией о шифре, название, сорт, дату поступления, цену и примечание. Первичный ключ — code.
6. Product\_warehouse — таблица содержит информацию о связи товаров и складов. Содержит столбцы с информацией о номере склада и шифр и количество товара. Первичный ключ — ID.
7. Product\_department — таблица содержит информацию о связи товаров и отделов. Содержит столбцы с информацией о номере отдела и шифр и количество товара. Первичный ключ — ID.

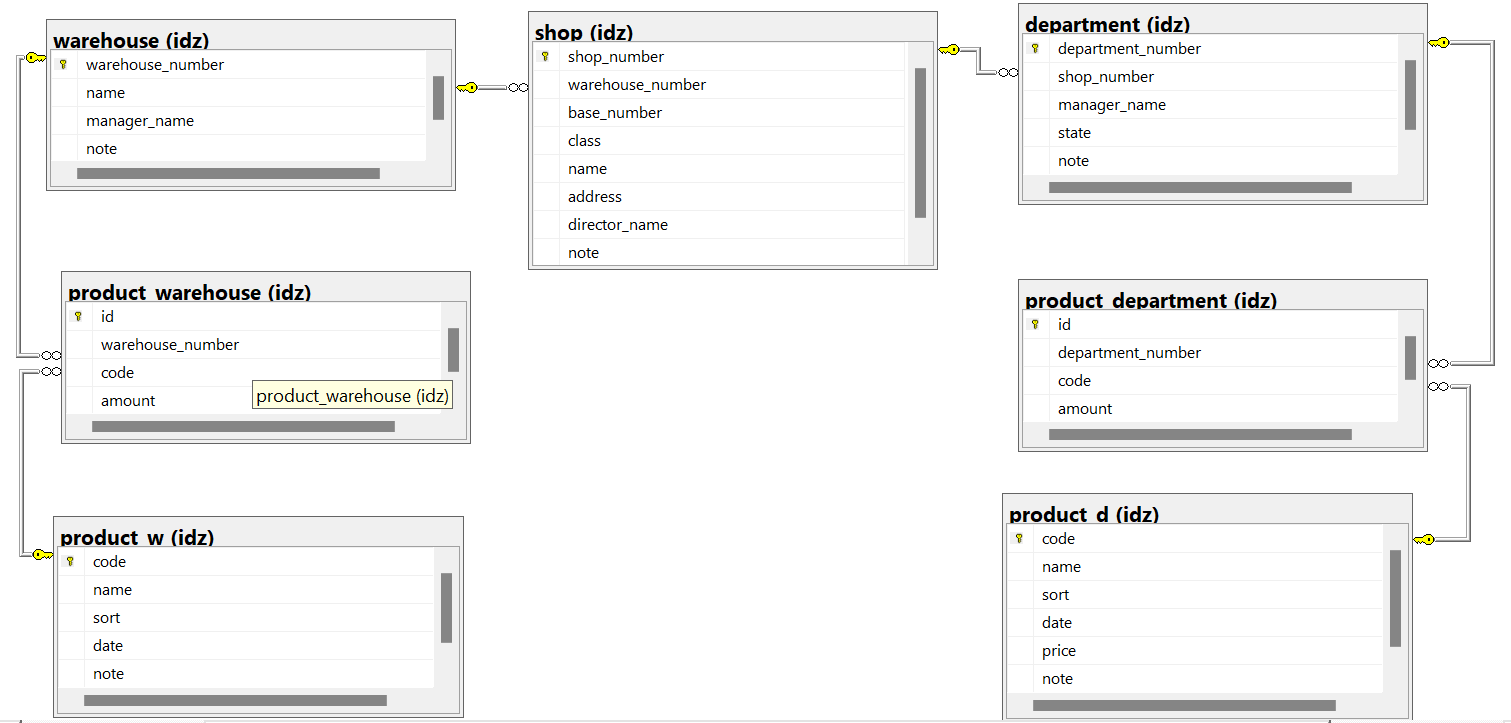


Рисунок 1

**Физическая модель данных**

Опишем данные в терминах T-SQL.

Таблица 1 – Спецификация таблицы bases

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица складов** | | | | Имя таблицы:  warehouse | | | |
| Дата разработки:  22.10.2023 | | |
| Порядковый номер таблицы:  **1** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Номер склада | | warehouse\_number | int | | PК | notNull | |  |
|  | 2 | Название | | name | nchar(15) | |  | notNull | |  |
|  | 3 | ФИО Управляющего | | manager\_name | nchar(50) | |  | notNull | |  |
|  | 4 | Примечание | | note | nchar(30) | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 2 – Спецификация таблицы shops

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица магазинов** | | | | Имя таблицы:  shop | | | |
| Дата разработки:  22.10.2023 | | |
| Порядковый номер таблицы:  **2** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Номер магазина | | shop\_number | int | | PK | notNull | |  |
|  | 2 | Номер склада | | warehouse\_number | int | | FK | notNull | |  |
|  | 3 | Адрес | | address | nchar(50) | |  | notNull | |  |
|  | 4 | ФИО директора | | director\_name | nchar(50) | |  | notNull | |  |
|  | 5 | Примечание | | note | nchar(30) | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 3 – Спецификация таблицы departments

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица отделов** | | | | Имя таблицы:  department | | | |
| Дата разработки:  22.10.2023 | | |
| Порядковый номер таблицы:  **3** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Номер отдела | | department\_number | int | | PK | notNull | |  |
|  | 2 | Номер магазина | | shop\_number | int | | FK | notNull | |  |
|  | 3 | Класс | | class | nchar(50) | |  | notNull | |  |
|  | 4 | Состояние | | state | bit | |  | notNull | |  |
|  | 5 | Примечание | | note | nchar(30) | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 4 – Спецификация таблицы products\_d

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица товаров отдела** | | | | Имя таблицы:  product\_d | | | |
| Дата разработки:  22.10.2023 | | |
| Порядковый номер таблицы:  **4** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Шифр | | code | int | | PK | notNull | |  |
|  | 2 | Название | | name | nchar(30) | |  | notNull | |  |
|  | 3 | Сорт | | sort | nchar(30) | |  | notNull | |  |
|  | 4 | Дата поступления | | date | datetime | |  | notNull | |  |
|  | 5 | Цена | | price | real | |  | price > 0 | |  |
|  | 6 | Примечание | | note | nchar(30) | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 5 – Спецификация таблицы products\_departments

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица связи продуктов и магазинов** | | | | Имя таблицы:  products\_department | | | |
| Дата разработки:  22.10.2023 | | |
| Порядковый номер  таблицы:  **5** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Номер отдела | | department\_number | int | | FK | notNull | |  |
|  | 2 | Шифр | | code | int | | FK | notNull | |  |
|  | 3 | Количество | | amount | int | |  | amount >0 | |  |
|  |  |  | |  |  | |  |  | |  |

# 

Таблица 6 – Спецификация таблицы product\_w

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица товаров базы** | | | | Имя таблицы:  product\_w | | | |
| Дата разработки:  17.11.2021 | | |
| Порядковый номер таблицы:  **6** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Шифр | | code | int | | P | notNull | |  |
|  | 2 | Название | | name | nchar(30) | |  | notNull | |  |
|  | 3 | Сорт | | sort | nchar(30) | |  | notNull | |  |
|  | 4 | Дата поступления | | date | datetime | |  | notNull | |  |
|  | 5 | Примечание | | note | nchar(30) | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 7 – Спецификация таблицы products\_bases

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица связи продуктов и складов** | | | | Имя таблицы:  products\_warehouse | | | |
| Дата разработки:  17.11.2021 | | |
| Порядковый номер  таблицы:  **7** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Номер базы | | base\_number | int | | Fk | notNull | |  |
|  | 2 | Шифр | | code | int | | Fk | notNull | |  |
|  | 3 | Количество | | amount | int | |  | amount >0 | |  |
|  |  |  | |  |  | |  |  | |  |

# Создание базы данных

Создание базы данных «Управление торговли»:

CREATE DATABASE Trade ON

(

name = EmployeesOfCompany\_dat,

filename = 'C:\DataBases\Trade.mdf',

size = 5MB,

maxsize = 50MB,

filegrowth = 5MB

)

log ON

(

name = EmployeesOfCompany\_log,

filename = 'C:\DataBases\Trade.ldf',

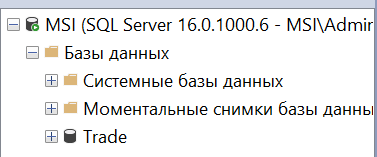
size = 5MB,

maxsize = 50MB,

filegrowth = 5MB

)

На рис. 2 представлена созданная база данных Trade.

Рисунок 2

Создадим отдельную схему — idz.

USE Trade

GO

CREATE SCHEMA idz

# 

# Создание таблиц базы данных

Создание таблицы warehouse:

CREATE TABLE [idz].[warehouse](

[waregouse\_number] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,

[name] [nchar](15) NOT NULL,

[manager\_name] [nchar](50) NOT NULL,

[note] [nchar](30) NULL,

)

Создание таблицы shop:

CREATE TABLE [idz].[shop](

[shop\_number] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,

[warehouse\_number] [int] REFERENCES idz.warehouse(warehouse\_number) NOT NULL,

[address] [nchar](50) NOT NULL,

[director\_name] [nchar](50) NOT NULL,

[note] [nchar](30) NULL,

)

Создание таблицы department:

CREATE TABLE [idz].[department](

[department\_number] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,

[shop\_number] [int] REFERENCES idz.shop(shop\_number) NOT NULL,

[class] [nchar](50) NOT NULL,

[state] [bit] NOT NULL,

)

Создание таблицы products\_d:

CREATE TABLE [idz].[products\_d](

[сode] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY ,

[name] [nchar](30) NOT NULL,

[sort] [nchar](30) NOT NULL,

[date] [datetime] NOT NULL,

[price] [real] NOT NULL CHECK (([price]>(0))),

[note] [nchar](30) NULL,

)

Создание таблицы products\_department:

CREATE TABLE [idz].[product\_department](

id INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

[department\_number] [int]REFERENCES idz.department ([department\_number]) NOT NULL,

[code] [int] REFERENCES idz.products\_d ([сode]) NOT NULL,

[amount] [int] NOT NULL CHECK ([amount]>(0)),

)

Создание таблицы product\_w:

CREATE TABLE [idz].[product\_w](

[сode] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY ,

[name] [nchar](30) NOT NULL,

[sort] [nchar](30) NOT NULL,

[date] [datetime] NOT NULL,

[note] [nchar](30) NULL,

)

Создание таблицы product\_warehouse:

CREATE TABLE [idz].[product\_warehouse](

id INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

[warehouse\_number] [int]REFERENCES idz.warehouse ([warehouse\_number]) NOT NULL,

[code] [int] REFERENCES idz.product\_w ([сode]) NOT NULL,

[amount] [int] NOT NULL CHECK ([amount]>(0)),

)

На рис. 3 представлены созданные таблицы в базе данных Trade.

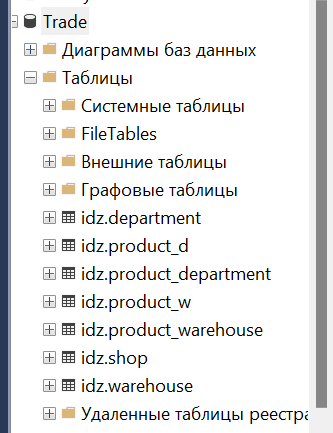


Рисунок 3

# **Заполнение таблиц данными**

Заполнение таблицы warehouse:

INSERT INTO [Trade].[idz].[warehouse](warehouse\_number,name,manager\_name,note) VALUES

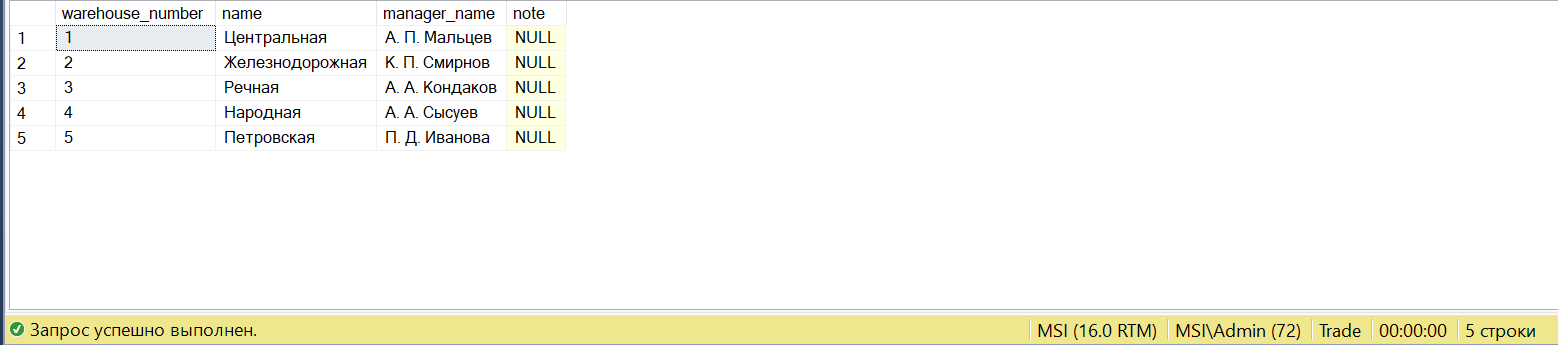
(1, 'Центральная', 'А. П. Мальцев', NULL),

(2, 'Железнодорожная', 'К. П. Смирнов', NULL),

(3, 'Речная', 'А. А. Кондаков', NULL),

(4, 'Народная', 'А. А. Сысуев', NULL),

(5, 'Петровская', 'П. Д. Иванова', NULL);

Рисунок 4

Заполнение таблицы shop:

INSERT INTO [Trade].[idz].[shop](shop\_number,warehouse\_number,address,director\_name,note)

VALUES

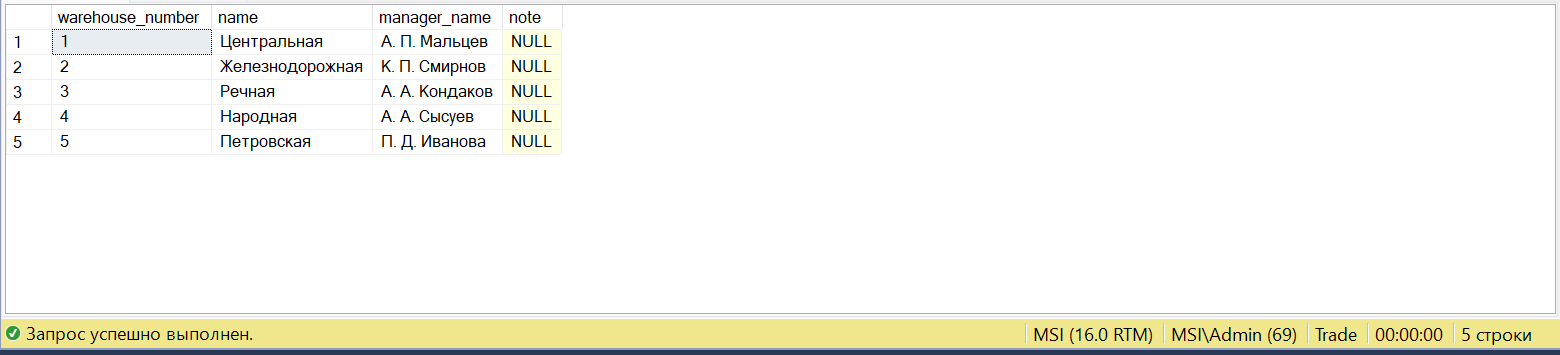
(1, 1, 'ул. Пушкина, 95', 'А. Г. Гусев', NULL),

(2, 1, 'ул Достоевского, 17','А. В. Игнатова', NULL),

(3, 2, 'ул. Тургенева, 31','А. И. Смирнов', NULL),

(4, 2, 'ул. Чернышевского, 49','Д. П. Балашова', NULL),

(5, 3, 'ул. Ленина, 99','Д. В. Крючкова', NULL);

Рисунок 5

Заполнение таблицы department:

INSERT INTO [Trade].[idz].[department](department\_number,shop\_number,class,state)

VALUES

(1, 1, 'Окна', 1),

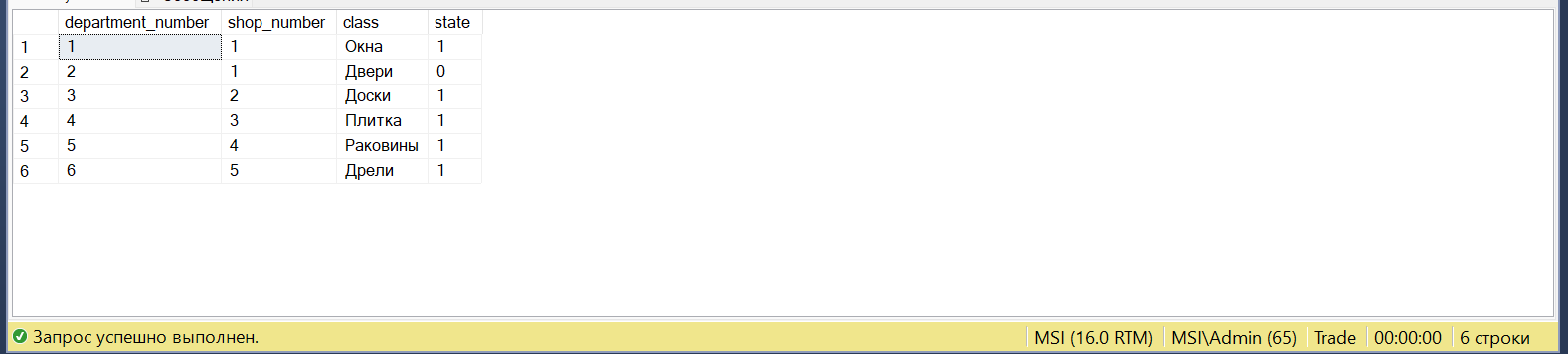
(2, 1, 'Двери', 0),

(3, 2, 'Доски', 1),

(4, 3, 'Плитка', 1),

(5, 4, 'Раковины', 1),

(6, 5, 'Дрели', 1);

Рисунок 6

Заполнение таблицы product\_d:

INSERT INTO

[Trade].[idz].[product\_d](сode,name,sort,date,price,note)

VALUES

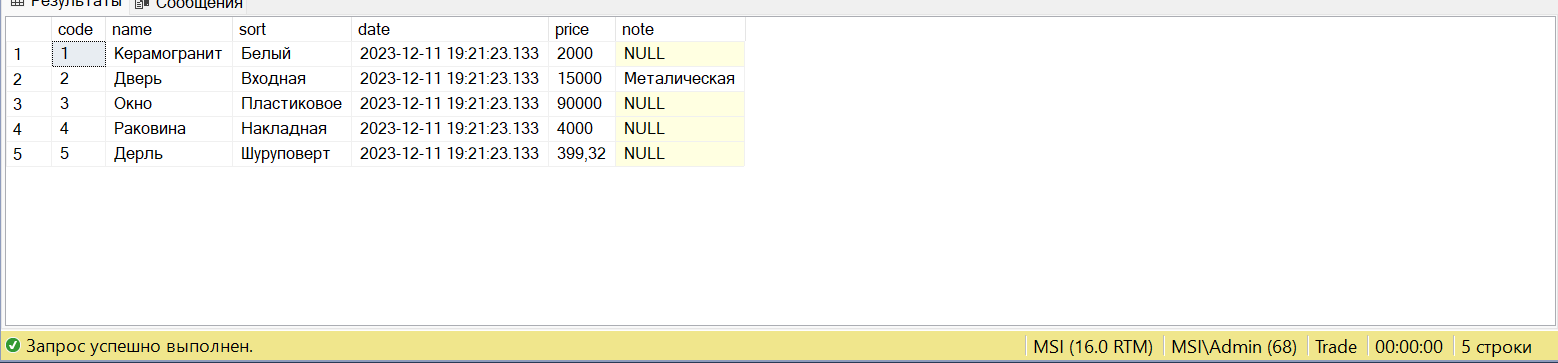
(1, 'Керамогранит', 'Белый', GETDATE(), 2000.00, NULL),

(2, 'Дверь', 'Входная', GETDATE(), 15000.00, 'Металическая'),

(3, 'Окно', 'Пластиковое', GETDATE(), 90000.00, NULL),

(4, 'Раковина', 'Накладная', GETDATE(), 4000.00, NULL),

(5, 'Дерль', 'Шуруповерт', GETDATE(), 399.32, NULL);

Рисунок 8

Заполнение таблицы product\_department:

INSERT INTO

[Trade].[idz].[product\_department](id,department\_number,code,amount)

VALUES

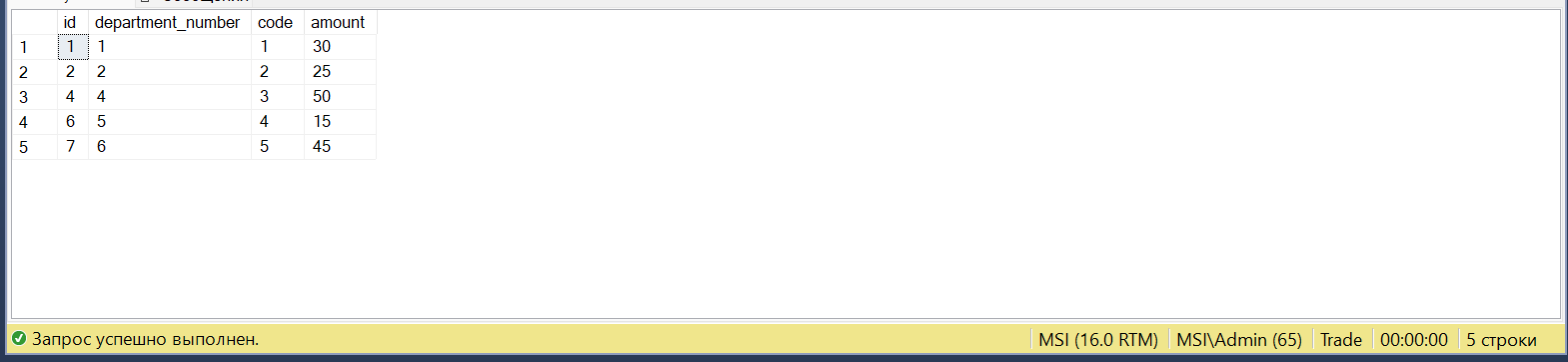
(1,1,1, 30),

(2,2,2, 25),

(4,4,3, 50),

(6,5,4, 15),

(7,6,5,45);

Рисунок 9

Заполнение таблицы product\_w:

INSERT INTO

[Trade].[idz].[product\_w](сode,name,sort,date,note)

VALUES

(1, 'Керамогранит', 'Белый', GETDATE(), NULL),

(2, 'Дверь', 'Входная', GETDATE(), 'Металическая'),

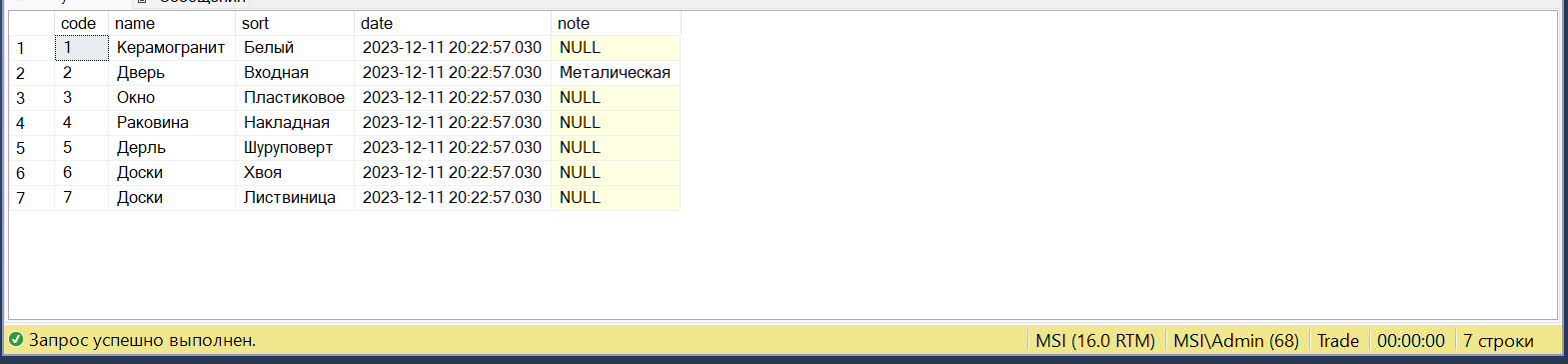
(3, 'Окно', 'Пластиковое', GETDATE(), NULL),

(4, 'Раковина', 'Накладная', GETDATE(), NULL),

(5, 'Дерль', 'Шуруповерт', GETDATE(), NULL),

(6, 'Доски', 'Хвоя',GETDATE(), NULL),

(7, 'Доски', 'Листвиница',GETDATE(), NULL);

Рисунок 10

Заполнение таблицы product\_warehouse:

INSERT INTO

[Trade].[idz].[product\_warehouse](id,warehouse\_number,code,amount)

VALUES

(1,1,1, 30),

(2,2,2, 25),

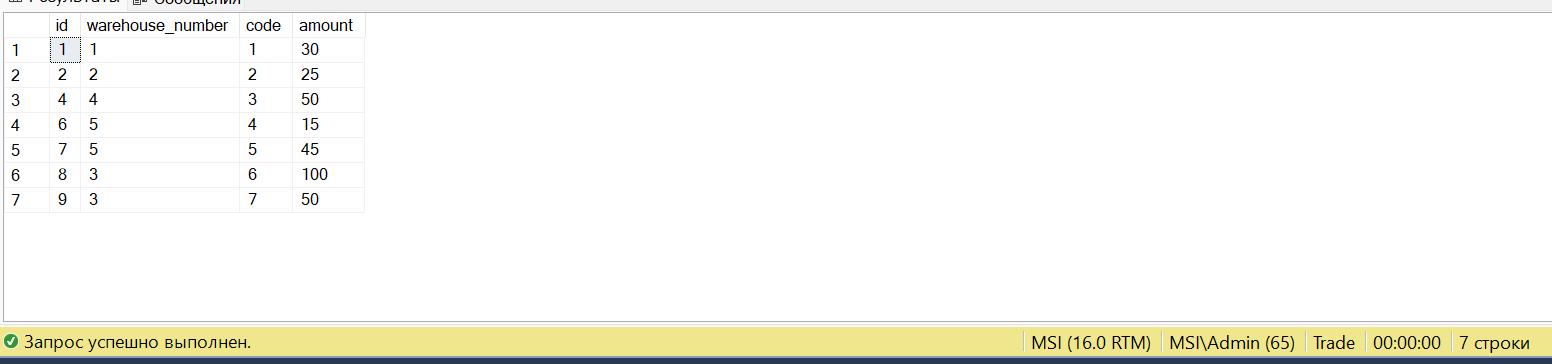
(4,4,3, 50),

(6,5,4, 15),

(7,5,5,45),

(8,3,6,100),

(9,3,7,50);

Рисунок 10

# Разработка объектов промежуточного слоя

**Хранимые процедуры**

Создание хранимой процедуры CloseDepartment. Эта процедура закрывает отдел магазина, при этом переносит товары из этого отдела в открытый. При попытке поставить id, которое отсутствует в таблицах, произойдет исключение.

GO

CREATE PROCEDURE CloseDepartment

@department int,

@new\_department int,

@shop int

AS

BEGIN

IF (NOT EXISTS (SELECT department\_number FROM departments Where shop\_number = @shop and department\_number = @department))

THROW 51000, 'This department dont exist in the database', 1;

IF (NOT EXISTS (SELECT department\_number FROM departments Where shop\_number = @shop and department\_number = @new\_department and state = 1))

THROW 51000, 'This new\_department dont exist in the database', 1;

UPDATE departments

SET state = 0

WHERE department\_number = @department AND shop\_number = @shop

UPDATE products\_departments

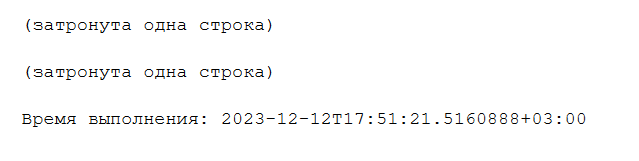
SET department\_number = @new\_department

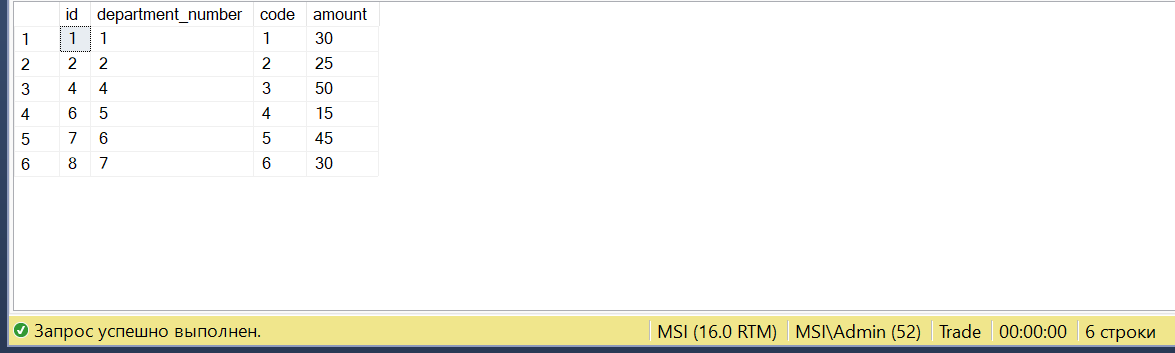
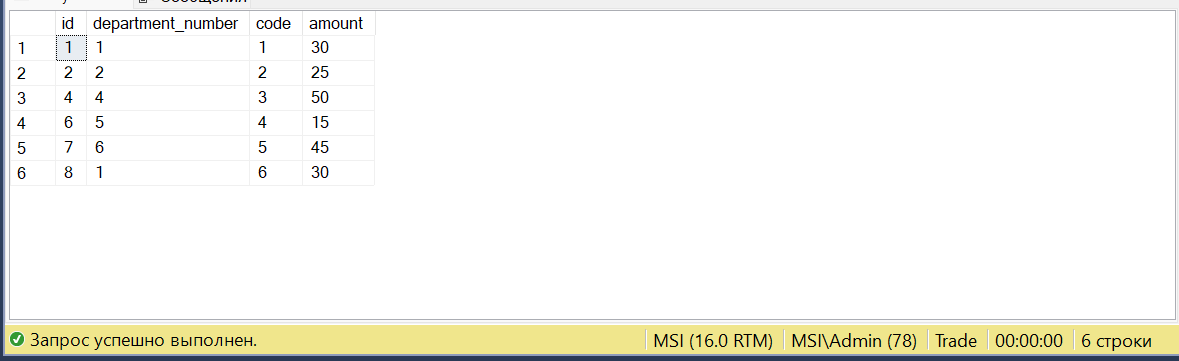
WHERE department\_number = @department

END

Проверим:

EXEC idz.CloseDepartment 7, 1, 1

Рисунок 16

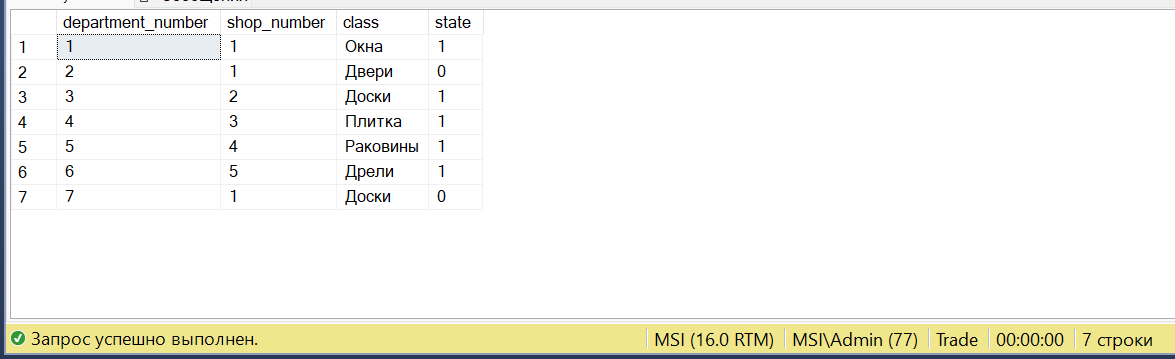
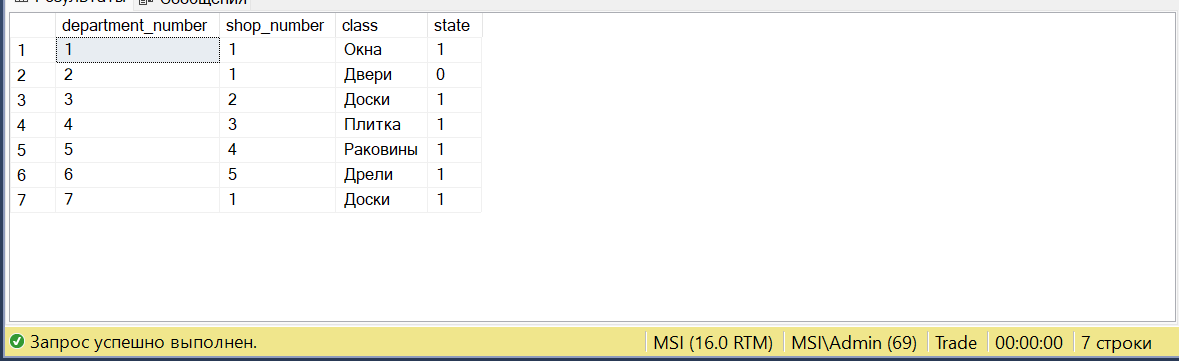
Рисунок 17

Рисунок 18

Создание хранимой процедуры MoveFromBaseToDepartment. Эта процедура переводит товар из склада в отдел. При этом следует указать цену товара в отделе. При попытке поставить id, которое отсутствует в таблицах, произойдет исключение.

GO

CREATE PROCEDURE idz.MoveFromBaseToDepartment

@old\_code int,

@new\_code int,

@warehouse int,

@department int,

@price int

AS

BEGIN

IF (NOT EXISTS (SELECT сode FROM product\_w t Where сode = @old\_code))

THROW 51000, 'Thes product dont exist in the database', 1;

Set IDENTITY\_INSERT product\_d ON

INSERT INTO product\_d(сode,name,sort,date,price,note) VALUES (

@new\_code,

(SELECT name FROM product\_w t WHERE сode = @old\_code),

(SELECT sort FROM product\_w t WHERE сode = @old\_code),

GETDATE(),

@price,

NULL);

DELETE FROM product\_w WHERE сode = @old\_code;

Set IDENTITY\_INSERT product\_department ON

INSERT INTO product\_department VALUES (

@department,

@new\_code,

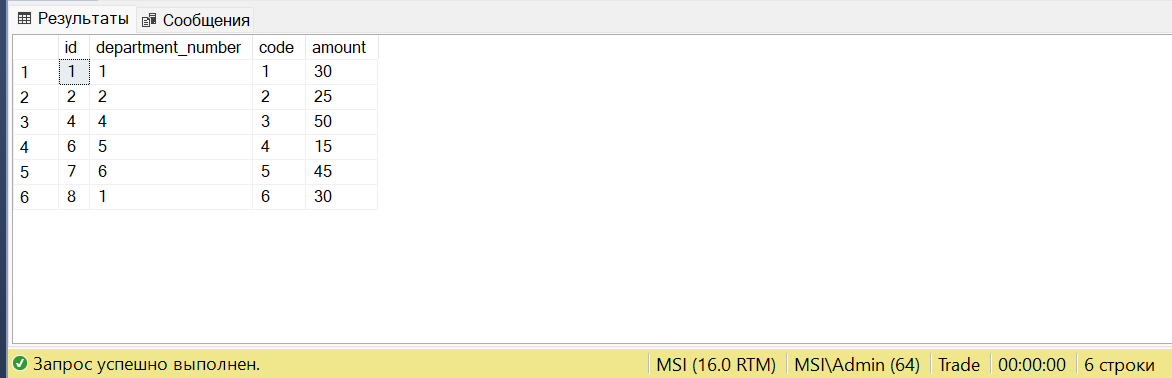
(SELECT amount FROM product\_warehouse WHERE code = @old\_code));

DELETE FROM product\_warehouse WHERE code = @old\_code and warehouse\_number = @warehouse;

END

Проверка:

EXEC idz.FromWarehousetoDepartment 13, 7, 3, 3, 1000



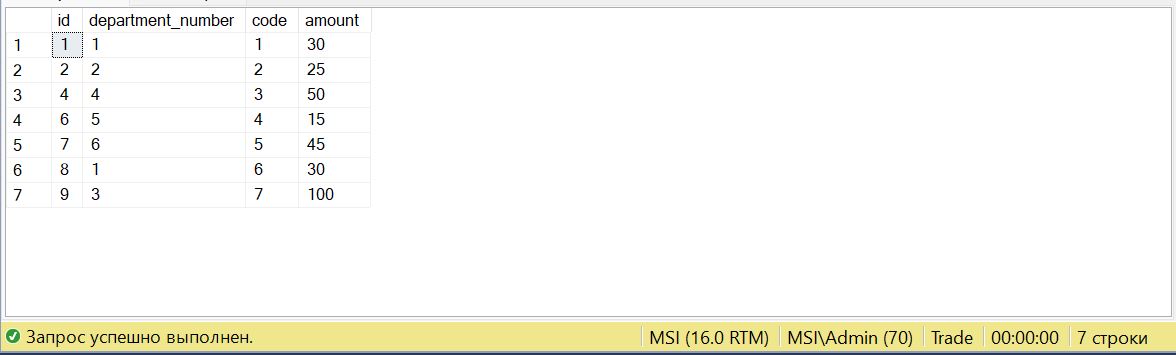
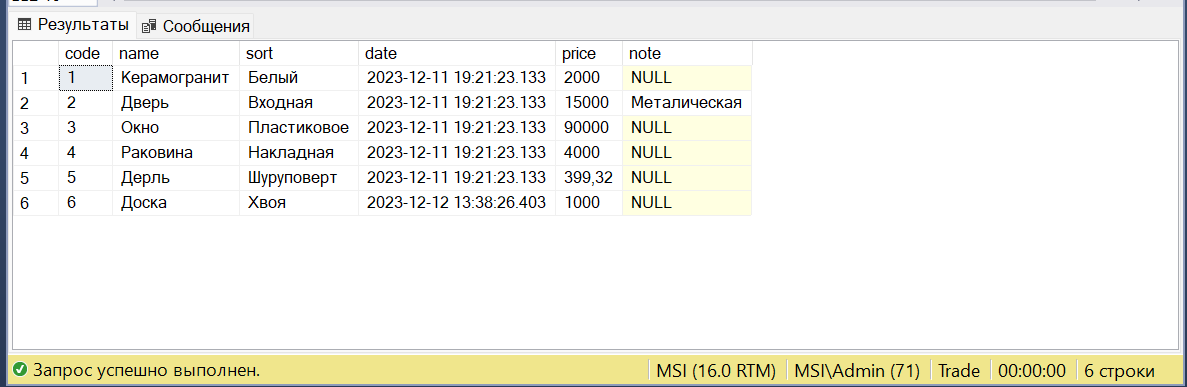


Рисунок 19



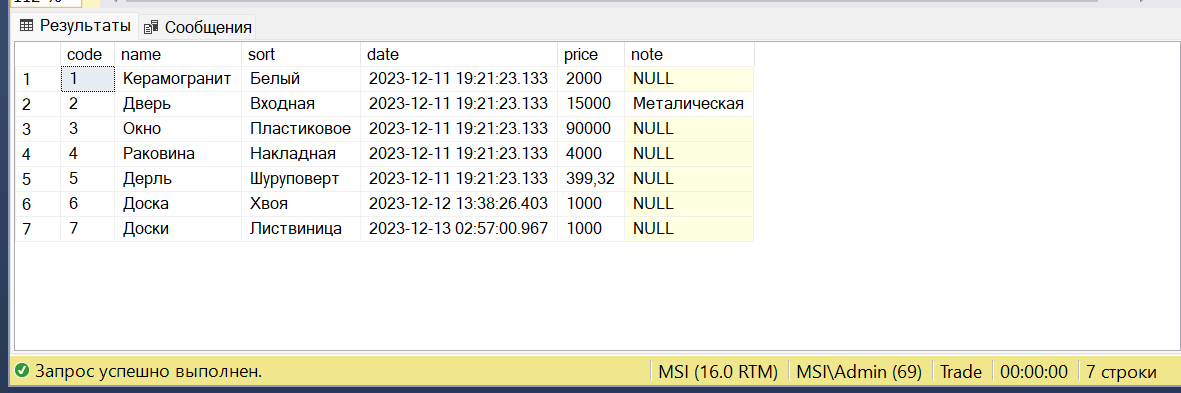
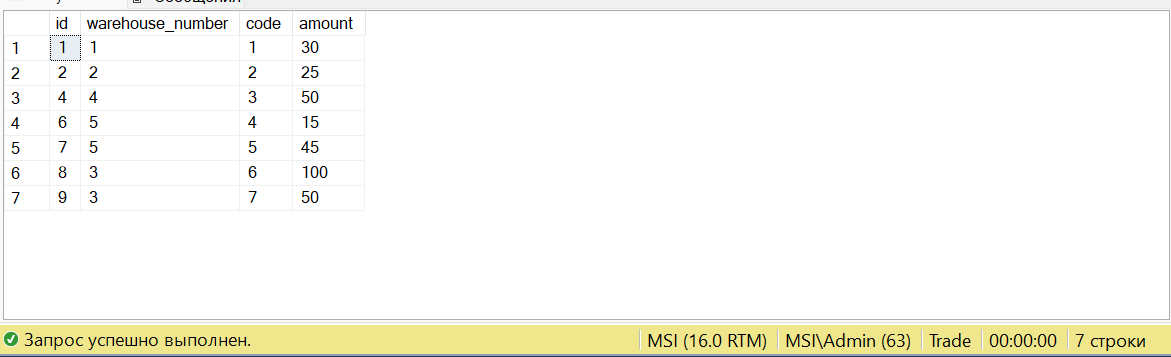


Рисунок 20



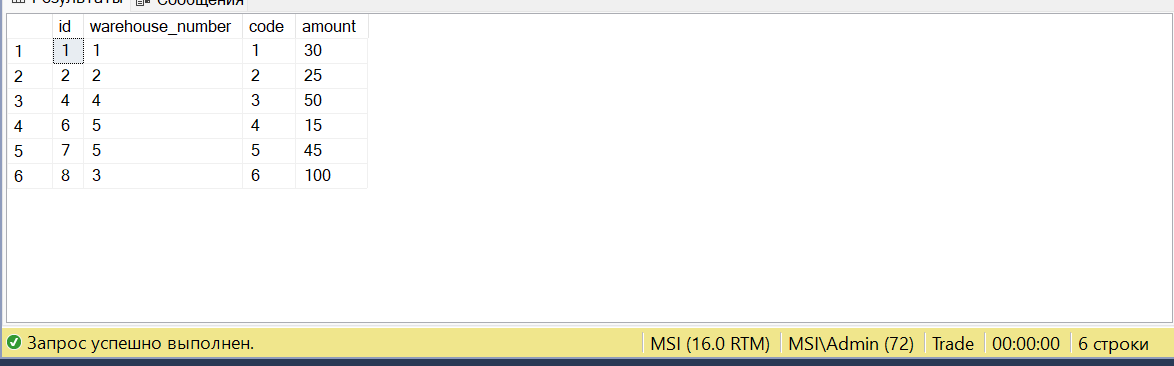
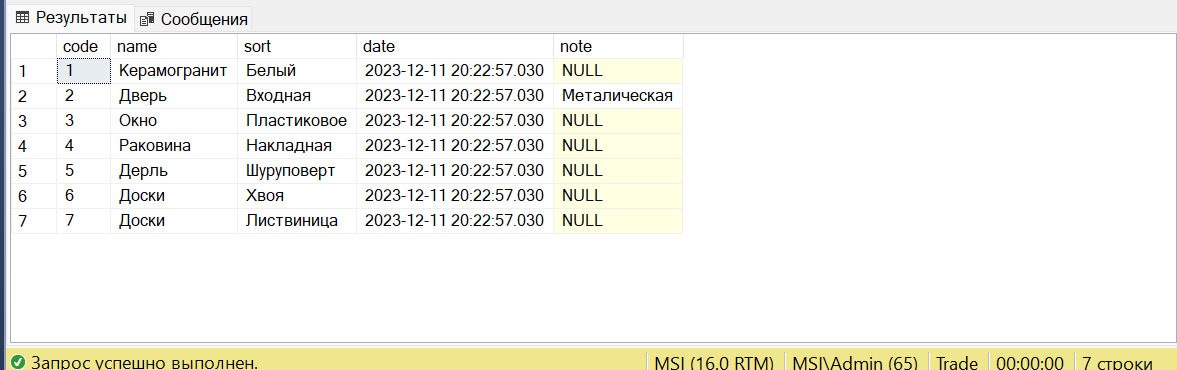


Рисунок 21



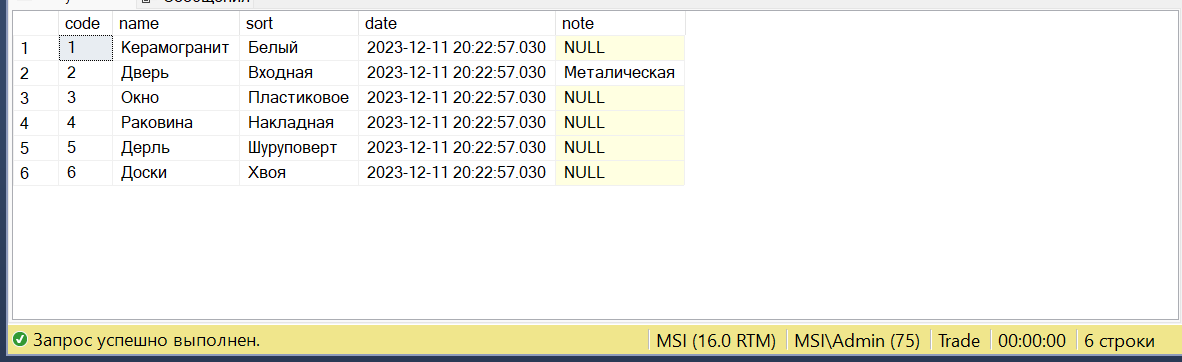


Рисунок 22

**Представления**

Создание представление ProductOnlyWarehouse. Это представление выводит на экран товары которые имеются только на складах.

CREATE VIEW idz.ProductOnlyWarehouse

AS

select w.\* from idz.product\_w as w

full join idz.product\_d as d on w.name=d.name and w.sort=d.sort

where d.сode is null

Проверка:

SELECT [сode],[name],[sort]

FROM [Trade].[idz].[ProductOnlyWarehouse]

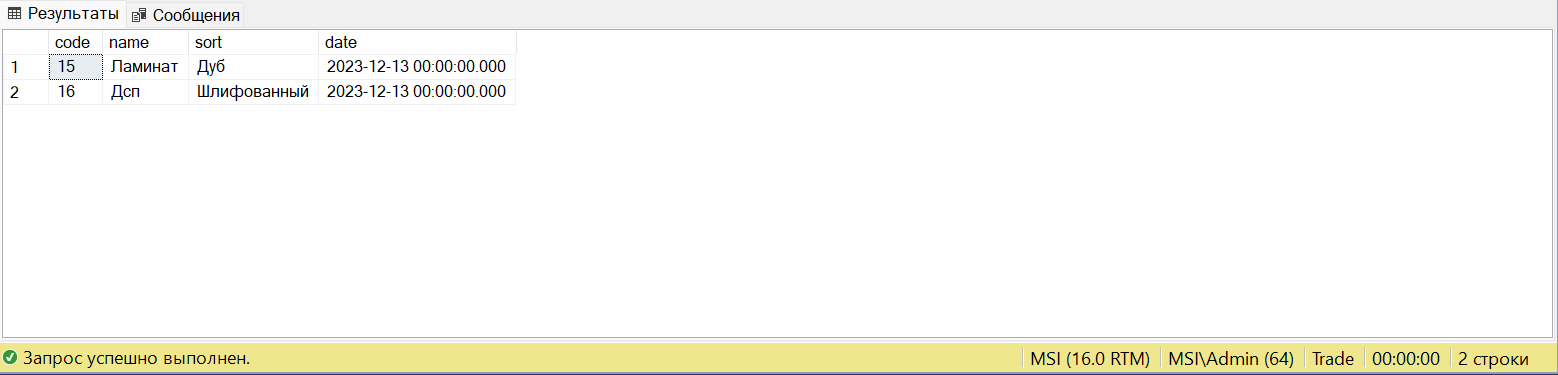
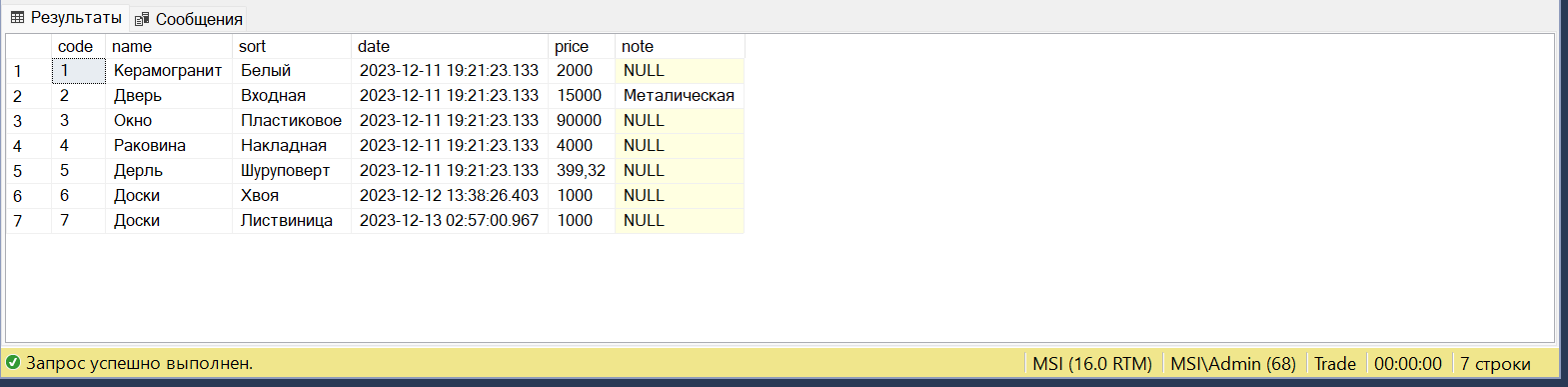
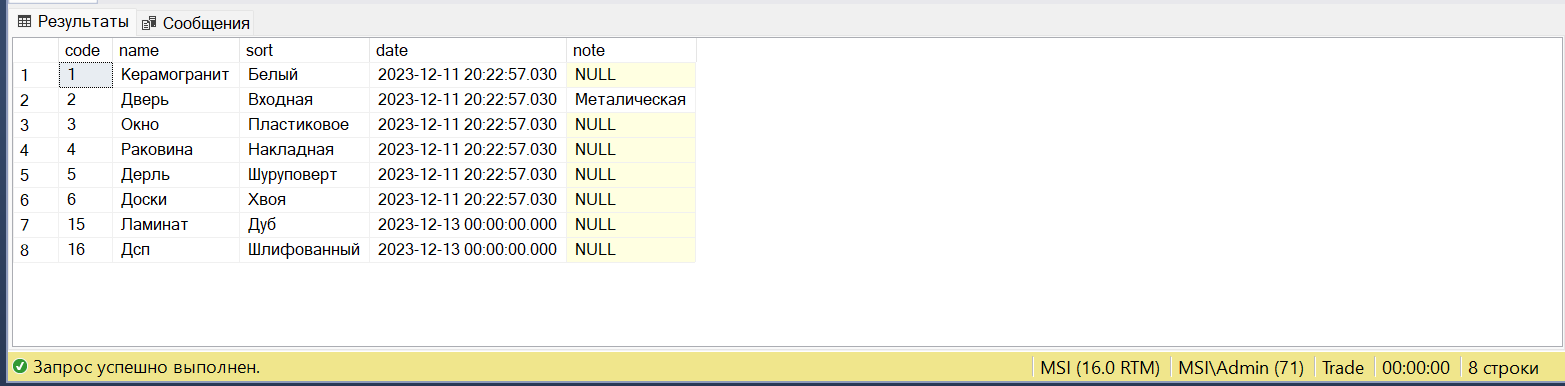


Рисунок 23

Создание представление CountingDepartments. Это представление выводит на экран сколько отделов определенного класса открыто во всех отделах.

CREATE VIEW idz.CountingDepartments

AS

SELECT

[class], count(shop\_number) as amount

FROM [Trade].[idz].[department] as d

where d.state=1 group by class

Проверка:

SELECT \*

FROM [Trade].[idz].[CountingDepartments]

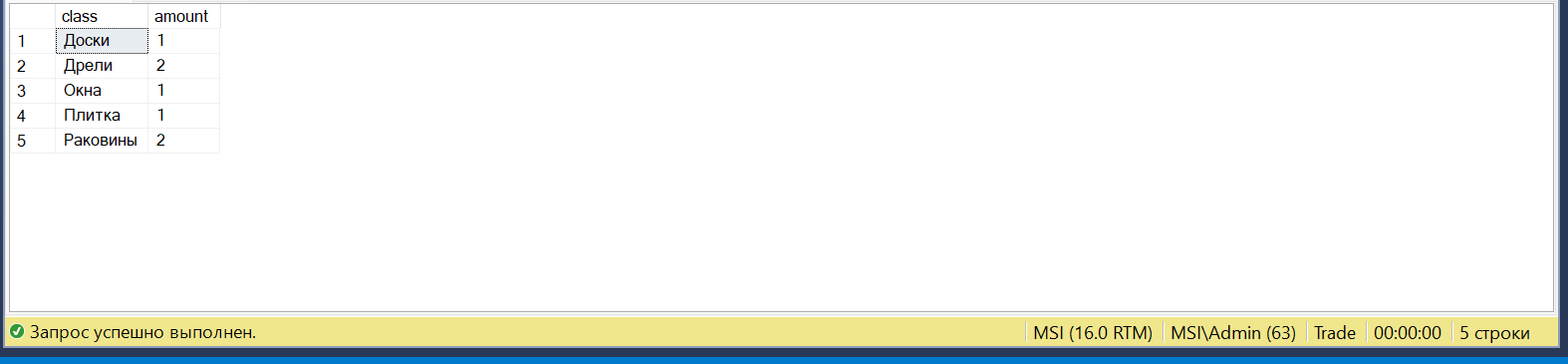
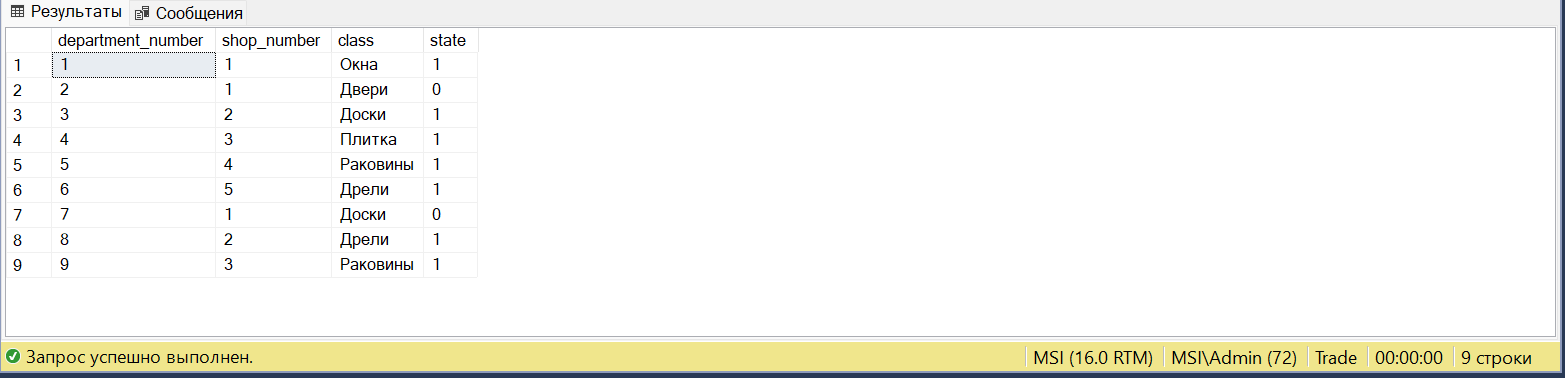


Рисунок 24

**UDF**

Создание определяемой пользователем функции Revenue\_Departmen, которая будет возвращать информацию о выручке отдела, если он открыт:

CREATE FUNCTION idz.Revenue\_Departmen(@DepartmentNumber INT)

RETURNS INT

AS

BEGIN

DECLARE @Revenue INT

IF EXISTS (SELECT \* FROM idz.department WHERE state =1 and department\_number=@DepartmentNumber)

Begin

Set @Revenue = (SELECT sum(pd.amount\*p.price) as revenue

FROM [Trade].[idz].[product\_department] as pd

join idz.product\_d as p on pd.code=p.сode

join idz.department as d on pd.department\_number=d.department\_number

where d.state =1 and pd.department\_number=@DepartmentNumber)

RETURN @Revenue

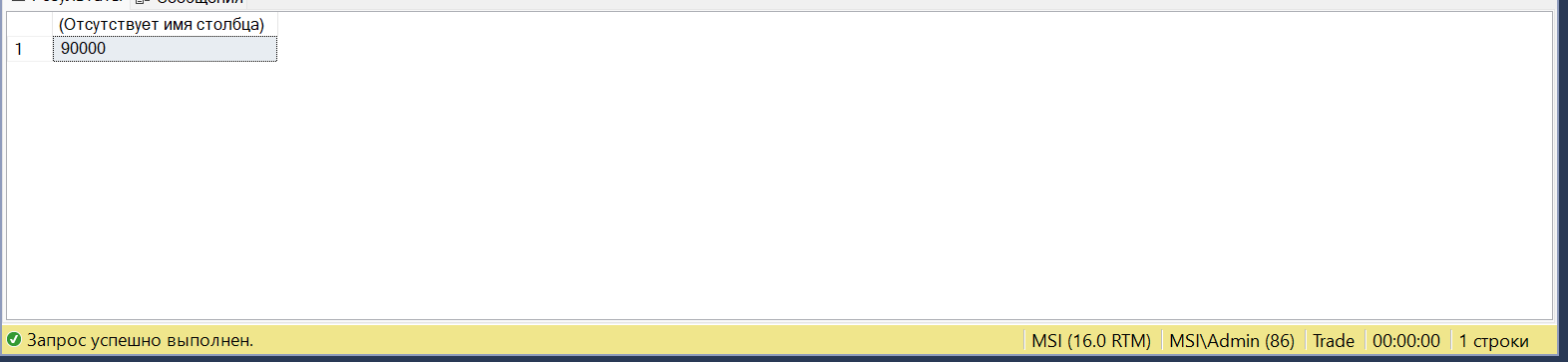
END

return NUll

End

Вызов определяемой пользователем функции:

SELECT idz.Revenue\_Departmen(1)

Рисунок 25

Создание определяемой пользователем функции Revenue\_Shop, которая будет возвращать информацию общую выручку магазина:

CREATE FUNCTION idz.Revenue\_Shop(@ShopNumber INT)

RETURNS INT

AS

BEGIN

DECLARE @Revenue INT

IF EXISTS (SELECT \* FROM idz.department WHERE state =1 and shop\_number=@ShopNumber)

Begin

Set @Revenue = (select sum(idz.Revenue\_Departmen(department\_number))

from idz.department as d

where state=1 and shop\_number=@ShopNumber)

RETURN @Revenue

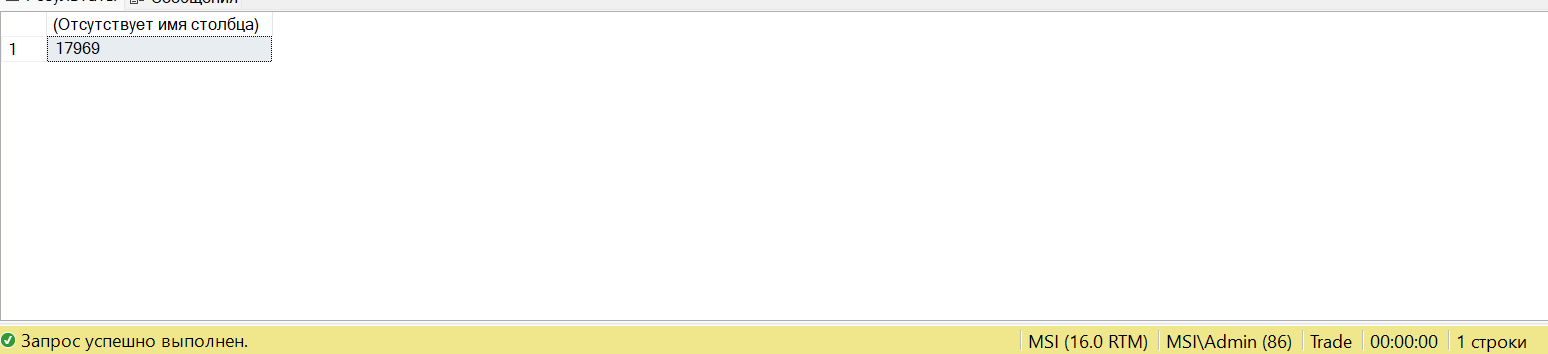
END

return NUll

End

Вызов определяемой пользователем функции:

SELECT idz.Revenue\_Shop(5)

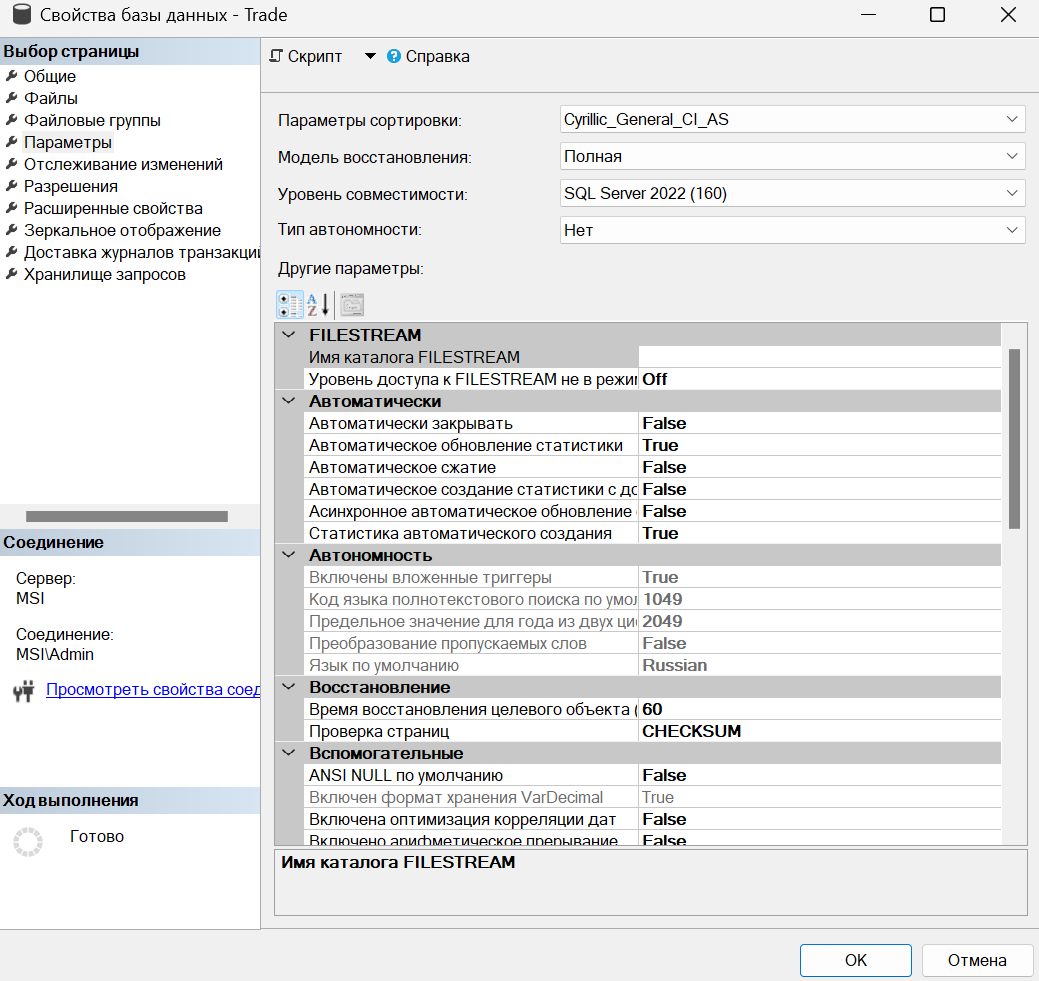


*Рисунок 26*

# Резервное копирование

Для обеспечения целостности и сохранности базы данных на случай  
возникновения сбоев в системе, хакерских атак, физического повреждения  
носителя базы данных и случайного изменения (удаления) данных необходимо  
разработать стратегию резервного копирования базы данных.  
В работе были сделаны: полная резервная копия самой базы данных,  
дифференциальная копия базы данных, которая учитывает изменения с  
последнего копирования, и журнал транзакций базы данных.

Перед началом изменим модель восстановления базы данных на «Полную».



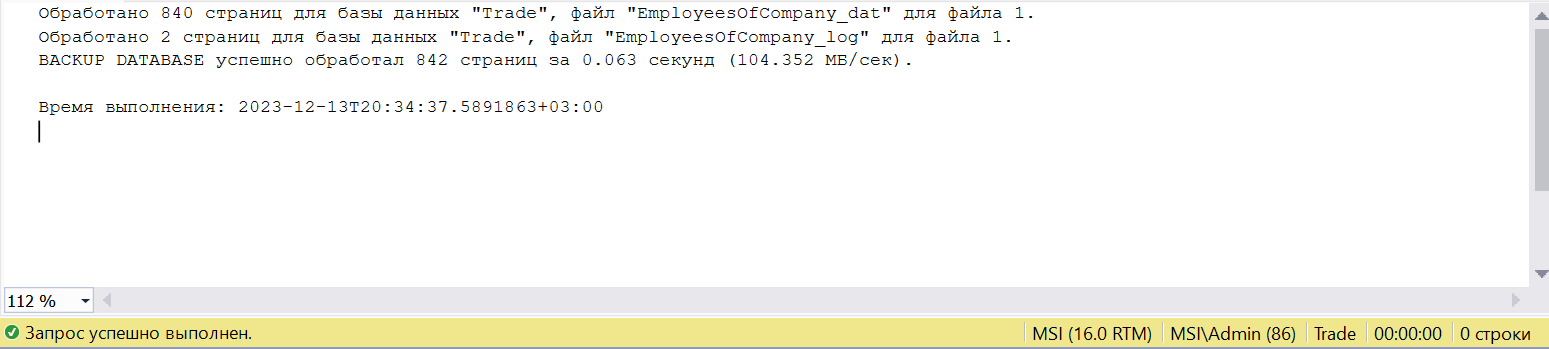
*Рисунок 27*

SQL-скрипт с созданием полной резервной копии БД :

BACKUP DATABASE Trade

TO DISK = 'C:\DataBases\rersrvtrade.bak'

WITH INIT



*Рисунок 28*

Создадим разностную резервную копию

BACKUP DATABASE Trade

TO DISK = 'C:\DataBases\rersrvtradeRAS.bak'

WITH

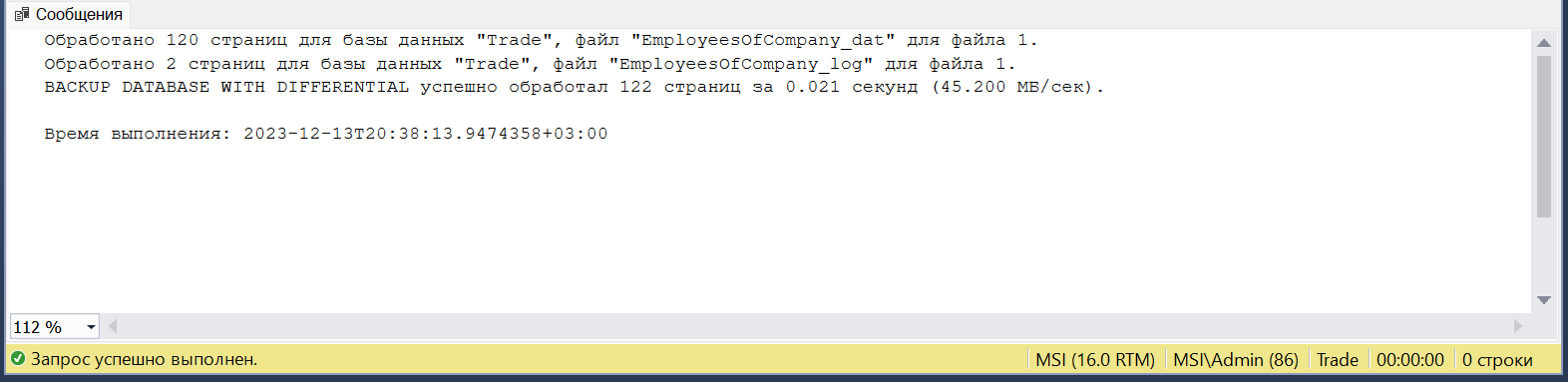
DIFFERENTIAL,INIT  


Рисунок 29

Создадим резервную копию журнала транзакций

BACKUP LOG Trade

TO DISK = 'C:\DataBases\rersrvtradeLOG.bak'

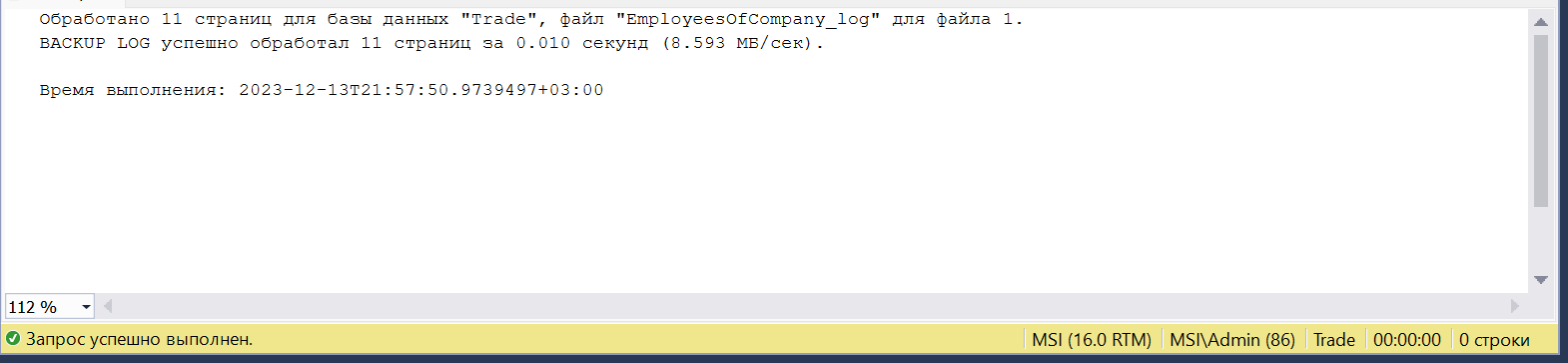


Рисунок 30

Для восстановления резервной копии базы данных и одного или нескольких журналов транзакций или в других случаях, когда необходимо применить несколько инструкций RESTORE, необходимо указать параметр WITH NORECOVERY для всех инструкций, кроме последней.

RESTORE DATABASE Trade

FROM DISK='C:\DataBases\rersrvtrade.bak'

WITH NORECOVERY

RESTORE DATABASE Trade

FROM DISK='C:\DataBases\rersrvtradeRAS.bak'

WITH NORECOVERY

RESTORE LOG Trade

FROM DISK='C:\DataBases\rersrvtradeLOG.bak'

WITH RECOVERY

# Вывод

В ходе выполнения данного индивидуального задания была спроектирована база данных «Строительный магазин».

В ходе работы было разработано несколько объектов промежуточного слоя:

1. Хранимые процедуры для закрытия отдела и переноса товаров в открытый и переноса товаров со склада в отдел

2. Функции для просмотра выручки с магазина и отдела.

3. Представления для определения товаров хранящихся только на складе и подсчета открытых отделов магазинов.

Также была предложена стратегия резервного копирования базы данных.

Выполнение задания позволило отточить навыки работы с базами данных, закрепить навыки, полученные на лекционной и практической частях курса «Базы данных».

# Список использованных источников

1. Распределенные базы данных: Методические указания к лабораторным работам / Сост.: А. В. Горячев, Н. Е. Новакова. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2008. 32 с
2. Курс «Базы данных» / Moodle — Open-source learning platform [Электронный ресурс] Режим доступа: https://vec.etu.ru/moodle/course/view.php?id=9558 (дата обращения 03.12.2022).
3. Справочник по Transact-SQL / Microsoft Learn [Электронный ресурс] Режим доступа: https://learn.microsoft.com/ru-ru/sql/t-sql/ (дата обращения 08.12.2022).