Комитет по образованию Правительства Санкт-Петербурга   
**Санкт-Петербургский** **Колледж информационных технологий**

**курсовой проект  
по МДК02.02 «Технология разработки и защиты баз данных»  
Специальность 09.02.03  
Программирование в компьютерных системах  
Тема: «Компании по ремонту квартир»**

Выполнил  
студент группы 474  
Черных Артём  
Проверила  
Женихова Н.В.   
Оценка:\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург 2020 г.

Содержание

[Введение 4](#_Toc57635403)

[1. Функциональные возможности системы 5](#_Toc57635404)

[2. Проектирование базы данных 6](#_Toc57635405)

[2.1. Анализ предметной области 6](#_Toc57635406)

[2.2. Концептуально-логическое проектирование 6](#_Toc57635407)

[2.3. Описание информационных объектов 8](#_Toc57635408)

[2.4. Схема данных 10](#_Toc57635409)

[3. Практическая часть 11](#_Toc57635410)

[3.1. Создание БД. Создание ограничений PK, FK 11](#_Toc57635411)

[3.2. Ввод данных с помощь процедур 14](#_Toc57635412)

[3.3. Разработка процедур с параметрами (без параметров) 14](#_Toc57635413)

[3.4. Описание пользователей и прав их доступа к объектам БД 14](#_Toc57635414)

[3.5. Создание ролей и определение их прав в БД 14](#_Toc57635415)

[3.6. Разработка стратегии резервного копирования БД 14](#_Toc57635416)

[3.7. Тестирование работы БД 14](#_Toc57635417)

[Заключение 15](#_Toc57635418)

[Список используемых источников 16](#_Toc57635419)

[Приложения 17](#_Toc57635420)

Введение

При выполнении курсовой работы решаются задачи закрепления теоретических знаний, а также практических навыков проектирования и разработки баз данных, полученных при изучении МДК02.02 «Технология разработки и защиты баз данных».

Курсовой проект представляет комплексную проектную, практическую разработку, которая в дальнейшем может стать основой выпускного дипломного проекта.

Цель курсовой работы – формирование практических навыков по проектированию и реализации базы данных для выбранной предметной области на основе полученных теоретических знаний.

В курсовой работе осуществляется проектирование и разработка баз данных для следующей предметной области:

База данных «Компании по ремонту квартир» хранит информацию о компаниях, которые предоставляют услуги по ремонту квартир. Каждая компания имеет набор услуг, сотрудников, а также в процессе рабочей деятельности приобретает заказчиков.

Заказчик по выбранной услуги и компании заключает договор, который имеет идентификатор договора, идентификатор услуги, идентификатор свой, дополнительные расходы, например за доставку материалов предоставляемая компанией, дату начала и окончания предоставляемой услуги и стоимость (общую). Услуг у заказчика может быть неограниченное количество, тем самым и договоров тоже.

1. Функциональные возможности системы
2. Занесение, просмотр информации о заказчиках.
3. Занесение, просмотр информации о договорах.
4. Занесение, просмотр информации об услугах и компаниях предоставляющие их.
5. Занесение, просмотр информации о сотрудниках.
6. Обновление или удаление информации о всём перечисленном ранее.
7. Предоставление скомпонованных данных компаний и их услугах.
8. Предоставление скомпонованных данных компаний и их сотрудников.
9. Расчёт общего дохода всех компаний.
10. Расчёт сроков предоставленных услуг в днях.
11. Группировка по названию услуги, чтобы узнать сколько похожих.
12. Группировка по названию компании, чтобы узнать сколько заказчиков у каждой из них за всё время.
13. Организация безопасности системы:
    1. Создание пользователей и ограничения их прав доступа;
    2. Настройка и разработка стратегии резервного копирования.
14. Проектирование базы данных

Процесс проектирования базы данных представляет собой последовательность переходов от словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели. В общем случае выделяют следующие этапы проектирования:

* Анализ и описание предметной области информационной системы;
* Концептуально-логическое проектирование;
* Описание информационных объектов базы данных;
* Схема данных.
  1. Анализ предметной области

Предметной областью в задании курсового проекта являются данные об заказчиках, договорах, услугах, компаниях, сотрудниках, где заказчики заказывают услугу(и) у компании(ях).

* 1. Концептуально-логическое проектирование

Даная модель даёт частично формализованное описание объектов предметной области в терминах некоторой семантической модели.

Согласно ER-модели, информационный объект изображается на диаграмме в виде прямоугольника, содержащего имя объекта. Связь изображается линией, которая связывает два информационных объекта, участвующих в отношении. Степень конца связи указывается графически, множественность связи изображается в виде «стрелки» на конце связи.

Таким образом, для связи “один-к-одному” используется линия без стрелки, для связи “один-ко-многим” – линия со стрелкой на стороне второго объекта, для связи “многие-ко-многим” – линия со стрелками на сторонах обоих объектов. При необходимости Атрибуты информационного объекта записываются внутри прямоугольника, его изображающего.

На основе анализа предметной области «Компании по ремонту квартир», были выделены следующие информационные объекты, которые необходимо хранить в базе данных: Заказчик, Договор, Услуги, Компании, Сотрудники.

Каждый из выделенных информационных объектов имеет следующие атрибуты:

Заказчик – ID заказчика, фамилия, имя, отчество, телефон, электронная почта заказчика;

Договор – ID договора, ID заказчика, ID услуги, дополнительные расходы, дата начала, дата окончания, стоимость;

Услуги – ID услуги, Название, ID компании, цена;

Компании – ID компании, название, телефон, электронная почта компании;

Сотрудники – ID сотрудника, фамилия, имя, отчество, телефон, электронная почта сотрудника.

На рисунке 1 приведена концептуально-логическая модель предметной области «Компании по ремонту квартир» в виде ER-диаграммы.

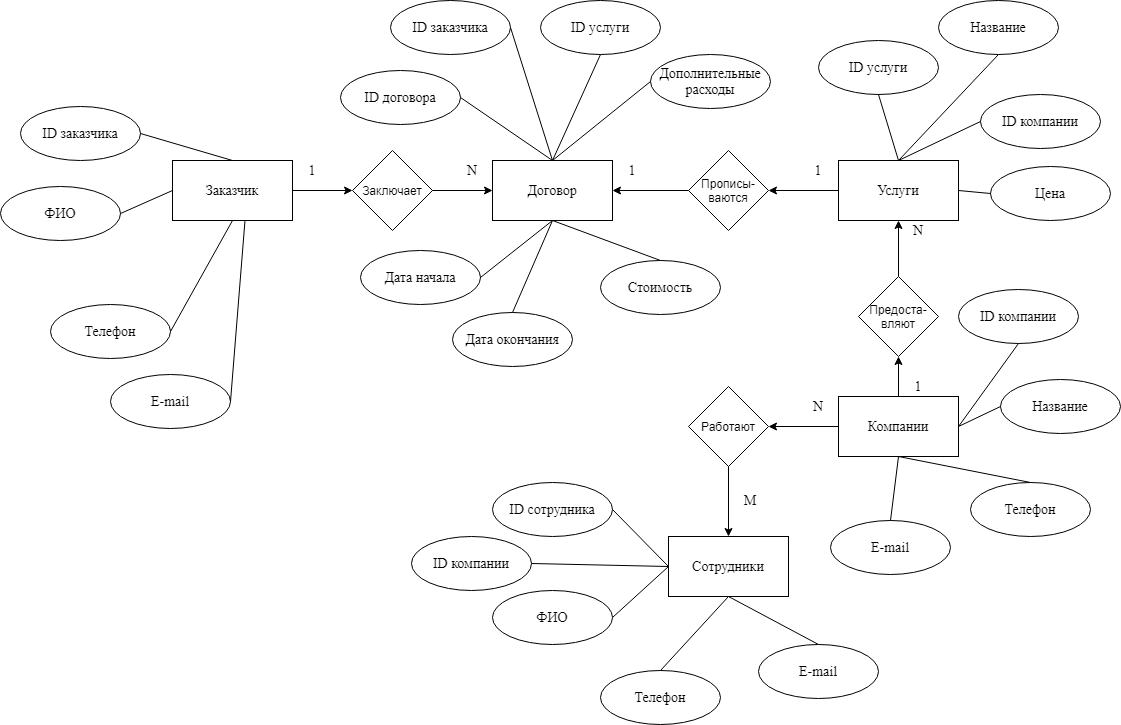


Рисунок 1 – Концептуально-логическая схема

* 1. Описание информационных объектов

В результате проектирования базы данных были получены информационные объекты, структура и характеристики которых приведены в таблицах 1-5.

Таблица 1 Заказчик (Customer)

|  |  |  |  |
| --- | --- | --- | --- |
| **Название поля** | **Тип данных** | **Свойства поля** | **Ключ** |
| ID\_заказчика | int | identity, not null | PK |
| Фамилия | nvarchar(70) | not null |  |
| Имя | nvarchar(70) | not null |  |
| Отчество | nvarchar(70) | not null |  |
| Телефон | char(20) | not null |  |
| E-mail | varchar(250) | null |  |

Таблица 2 Договор (Contract)

|  |  |  |  |
| --- | --- | --- | --- |
| **Название поля** | **Тип данных** | **Свойства поля** | **Ключ** |
| ID\_договора | int | not null | PK |
| ID\_заказчика | int | not null | FK |
| ID\_услуги | int | not null | FK |
| Дополнительные\_расходы | money | not null |  |
| Дата\_начала | date | not null |  |
| Дата\_окончания | date | not null |  |
| Стоимость | money | not null |  |

Таблица 3 Услуги (Services)

|  |  |  |  |
| --- | --- | --- | --- |
| **Название поля** | **Тип данных** | **Свойства поля** | **Ключ** |
| ID\_услуги | int | not null | PK |
| Название | nvarchar(250) | not null |  |
| ID\_компании | int | not null | FK |
| Цена | money | not null |  |

Таблица 4 Компании (Companies)

|  |  |  |  |
| --- | --- | --- | --- |
| **Название поля** | **Тип данных** | **Свойства поля** | **Ключ** |
| ID\_компании | int | identity, not null | PK |
| Название | nvarchar(250) | not null |  |
| Телефон | char(20) | not null |  |
| E-mail | varchar(250) | null |  |

Таблица 5 Сотрудники (Employees)

|  |  |  |  |
| --- | --- | --- | --- |
| **Название поля** | **Тип данных** | **Свойства поля** | **Ключ** |
| ID\_сотрудника | int | identity, not null | PK |
| Фамилия | nvarchar(70) | not null |  |
| Имя | nvarchar(70) | not null |  |
| Отчество | nvarchar(70) | not null |  |
| Телефон | char(20) | not null |  |
| E-mail | varchar(250) | null |  |

* 1. Схема данных

Разработанная схема данных в СУБД SQL Server на рисунке 2 логическая модель базы данных «Компании по ремонту квартир».

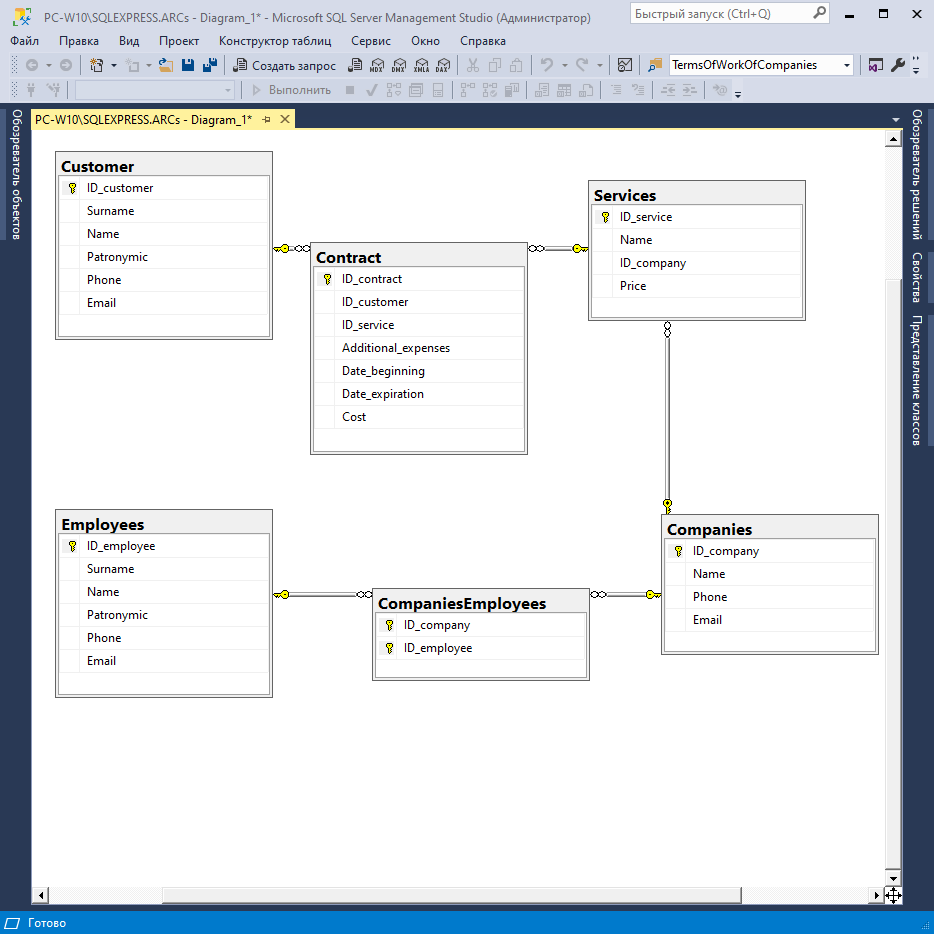


Рисунок 2 – Схема данных «Компании по ремонту квартир»

1. Практическая часть
   1. Создание БД. Создание ограничений PK, FK

База данных создана в среде Microsoft SQL Management 2017 с помощью языка запросов T-SQL. Она содержит 6 таблиц, с класторизованными (clustered) полями у главных таблиц, с каскадным обновление и удалением (update … , delete cascade).

Создание базы данных, где файл «ARCs\_dat.mdf» самой БД, а «ARCs\_log.ldf» журнала транзакций на рисунке 3.

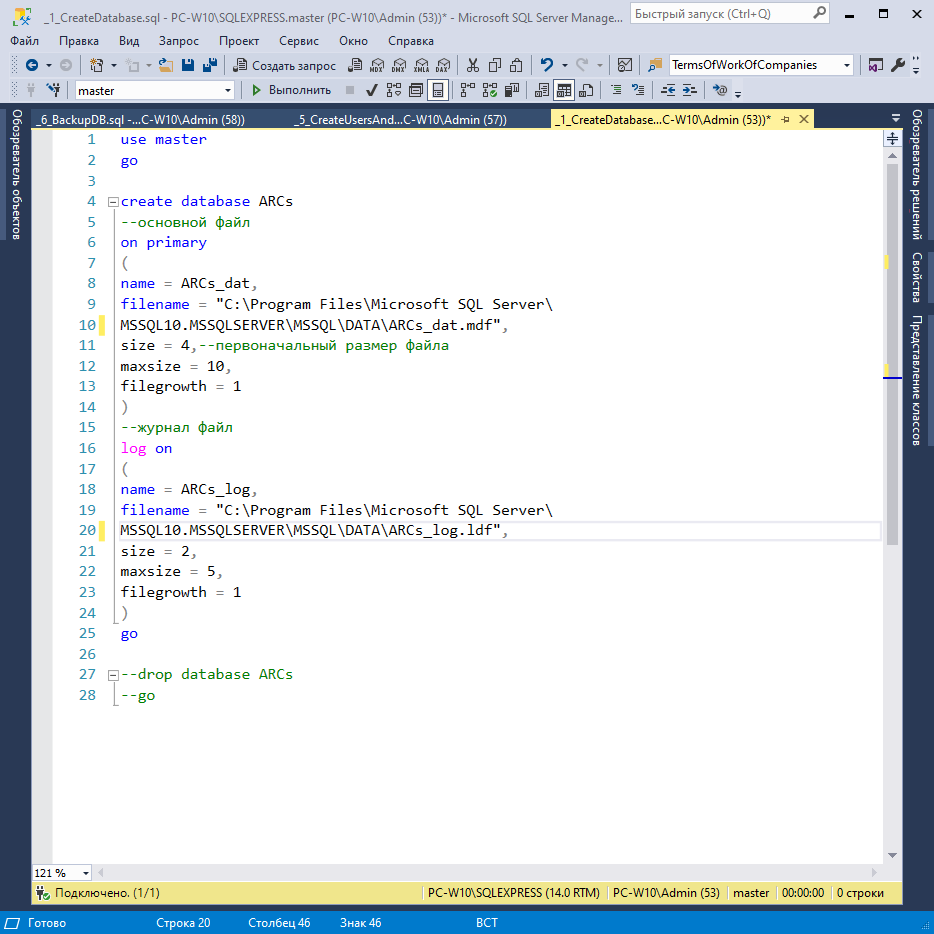


Рисунок 3 – Создание БД

На рисунках 4-6 создание таблиц: «Заказчик» (Customer), «Компании» (Companies), «Сотрудники» (Employees), «КомпанииСотрудники» (CompaniesEmployees), «Услуги» (Services), «Договор» (Contract).

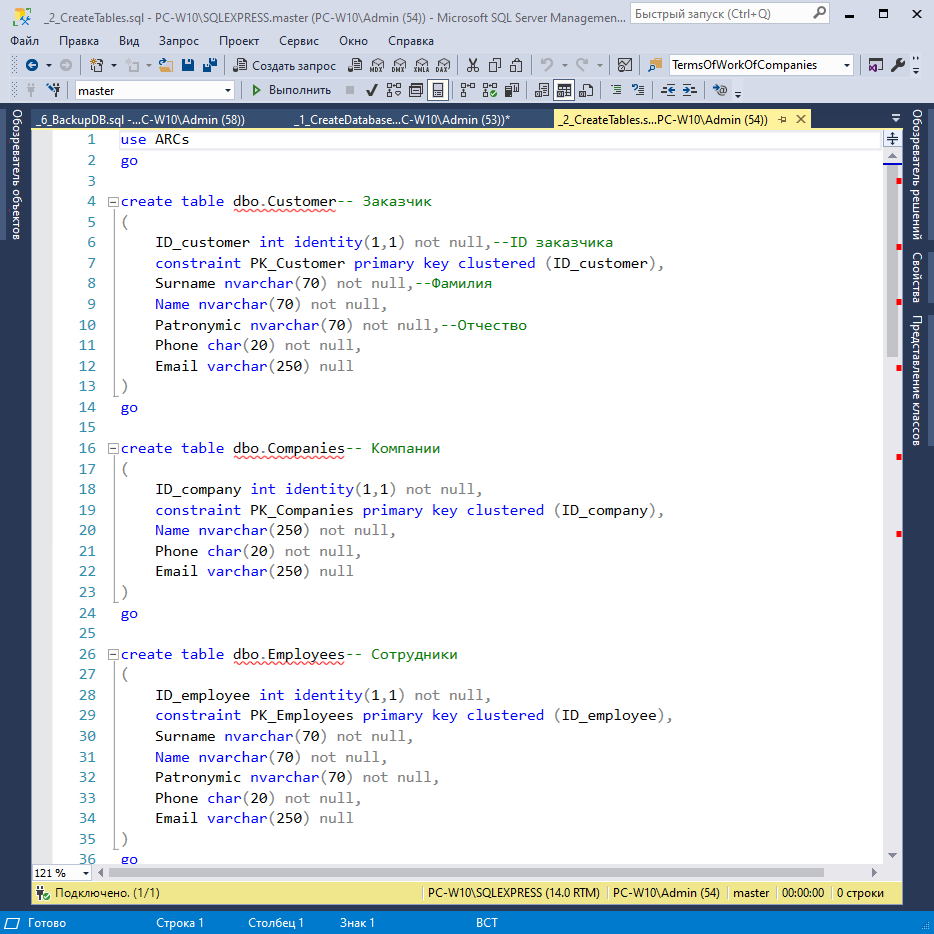


Рисунок 4 – Создание таблиц и ограничений

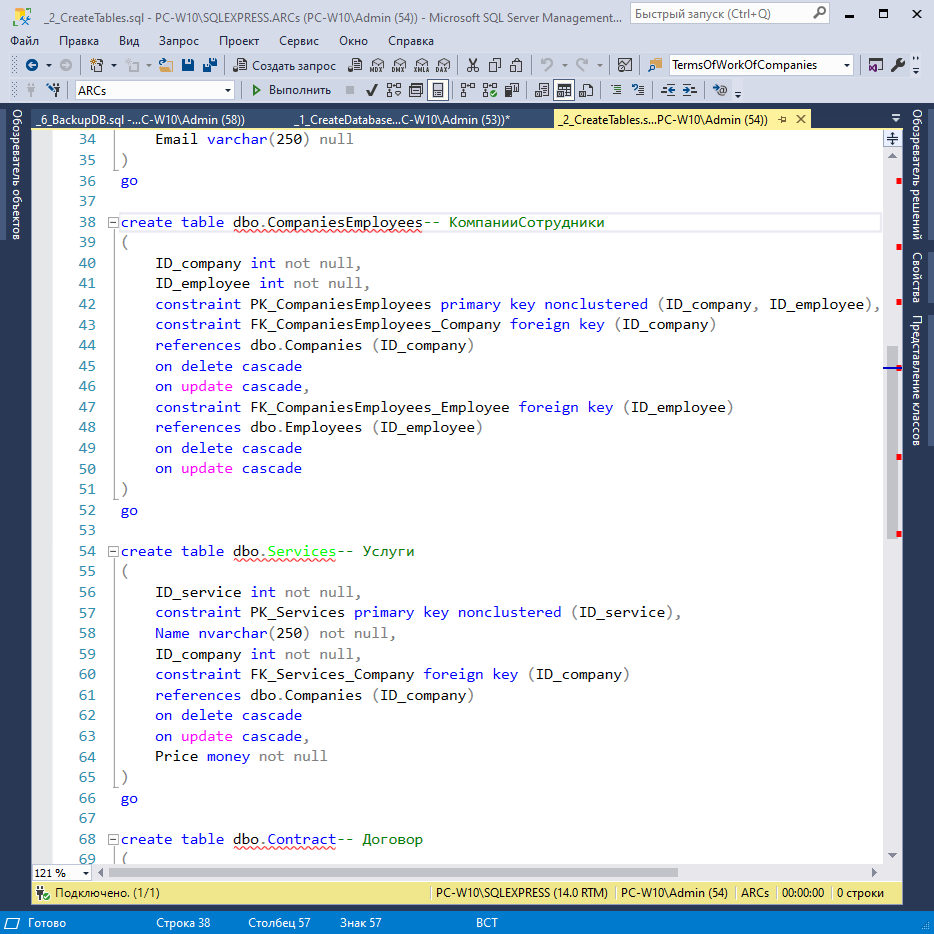


Рисунок 5 – Создание таблиц и ограничений

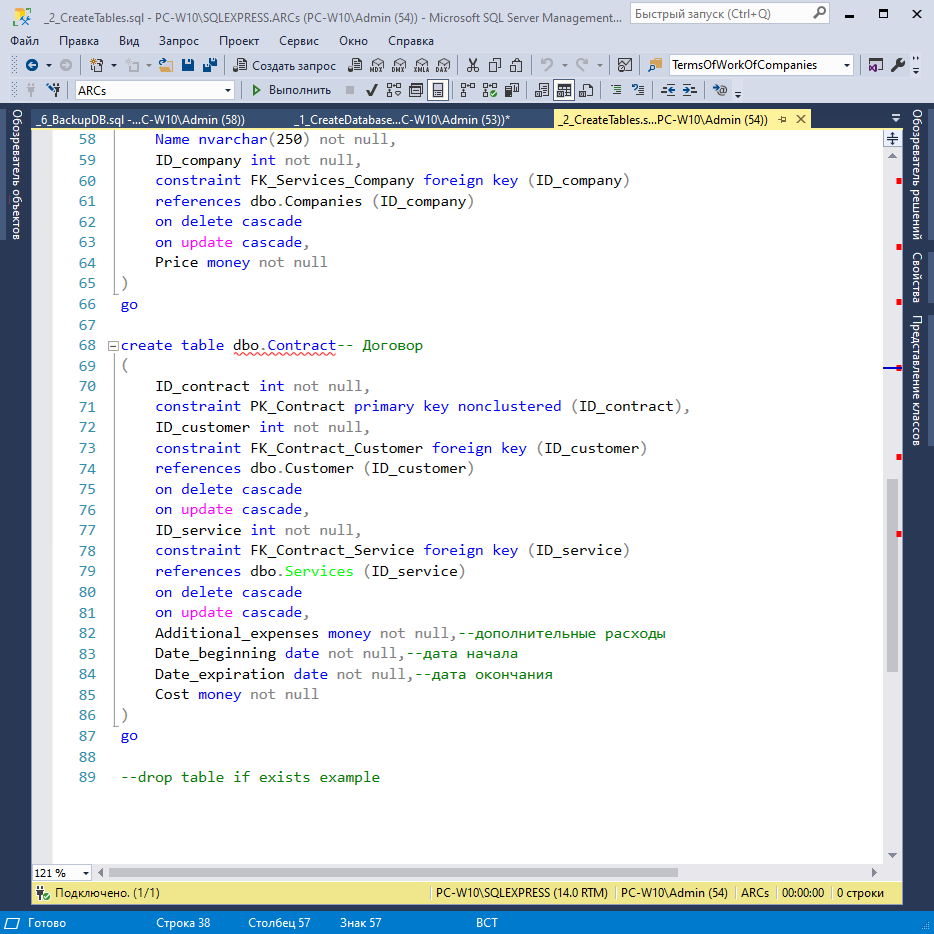


Рисунок 6 – Создание таблиц и ограничений

В обозревателе объектов БД можно наглядно видеть структуру созданных таблиц, рассмотрим на рисунке 7.

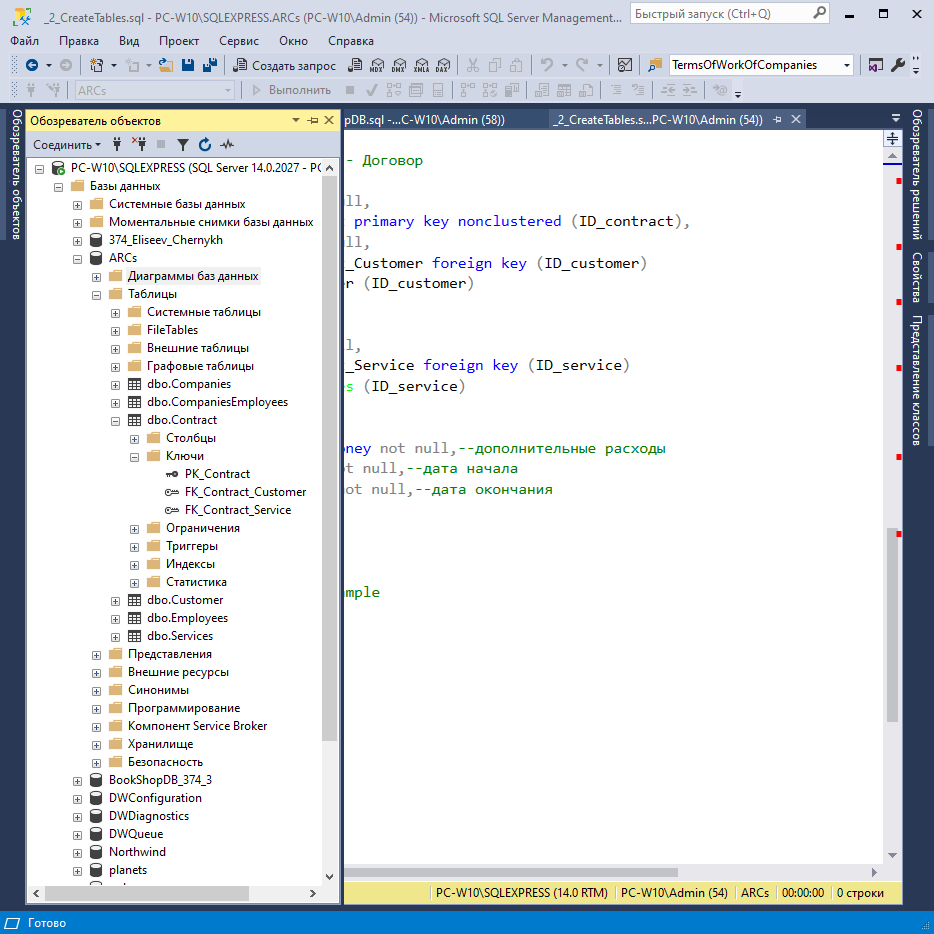


Рисунок 7 – Обозреватель объектов

* 1. Ввод данных с помощь процедур

Заполнение таблиц созданными процедурами, рассмотрим на рисунках 8-19

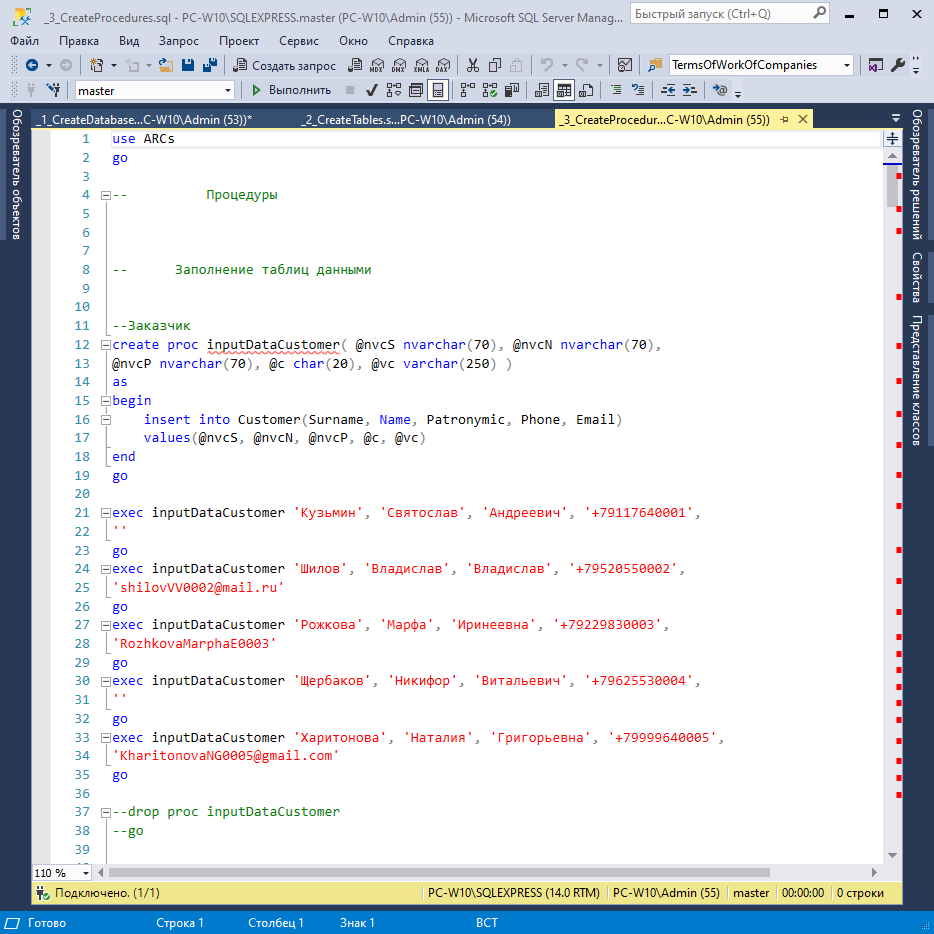


Рисунок 8 – Заполнение таблицы «Customer»

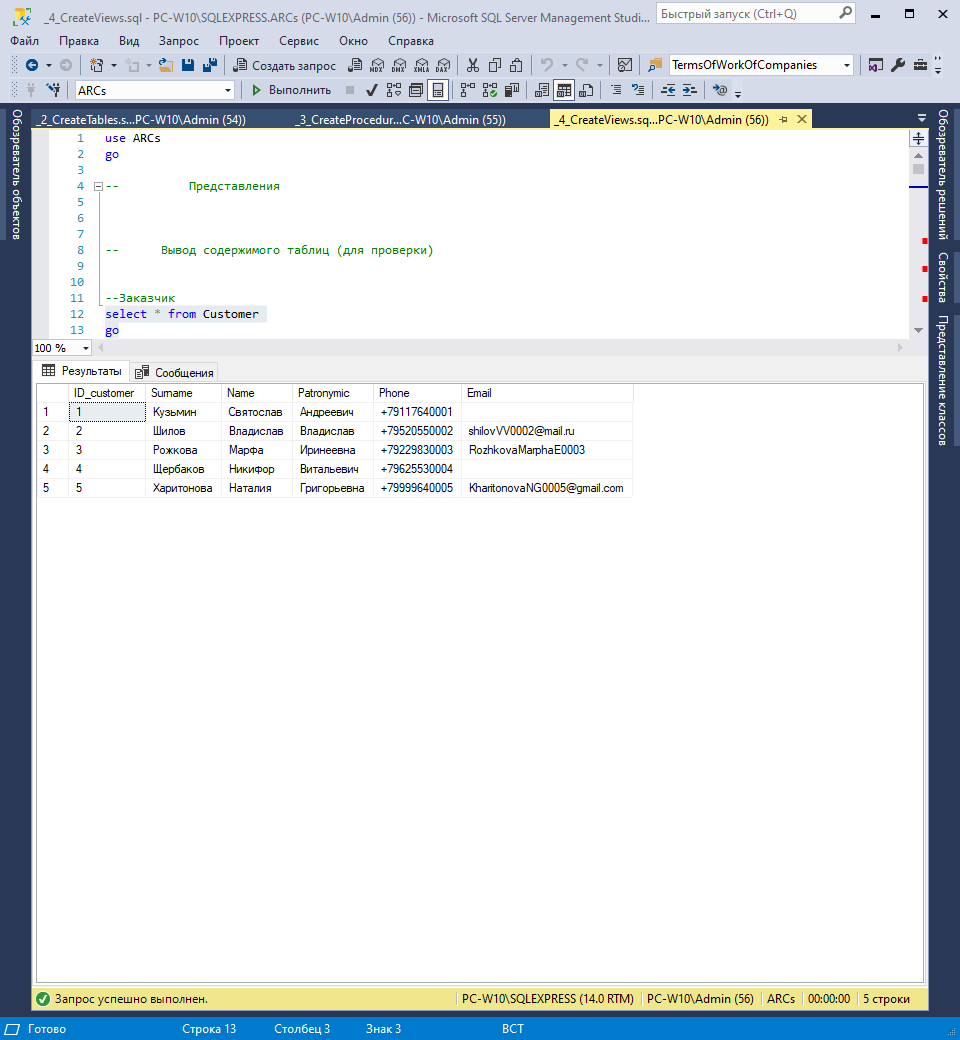


Рисунок 9 – Результат выполнения запроса

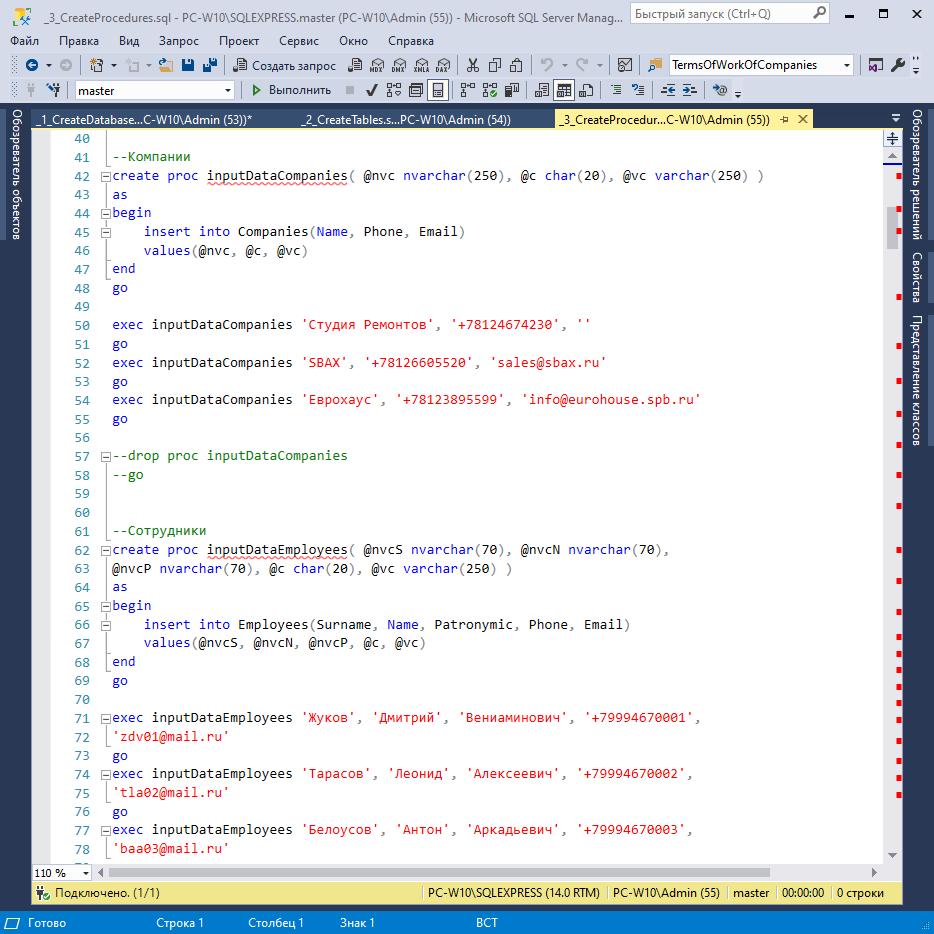


Рисунок 10 – Заполнение таблицы «Companies»

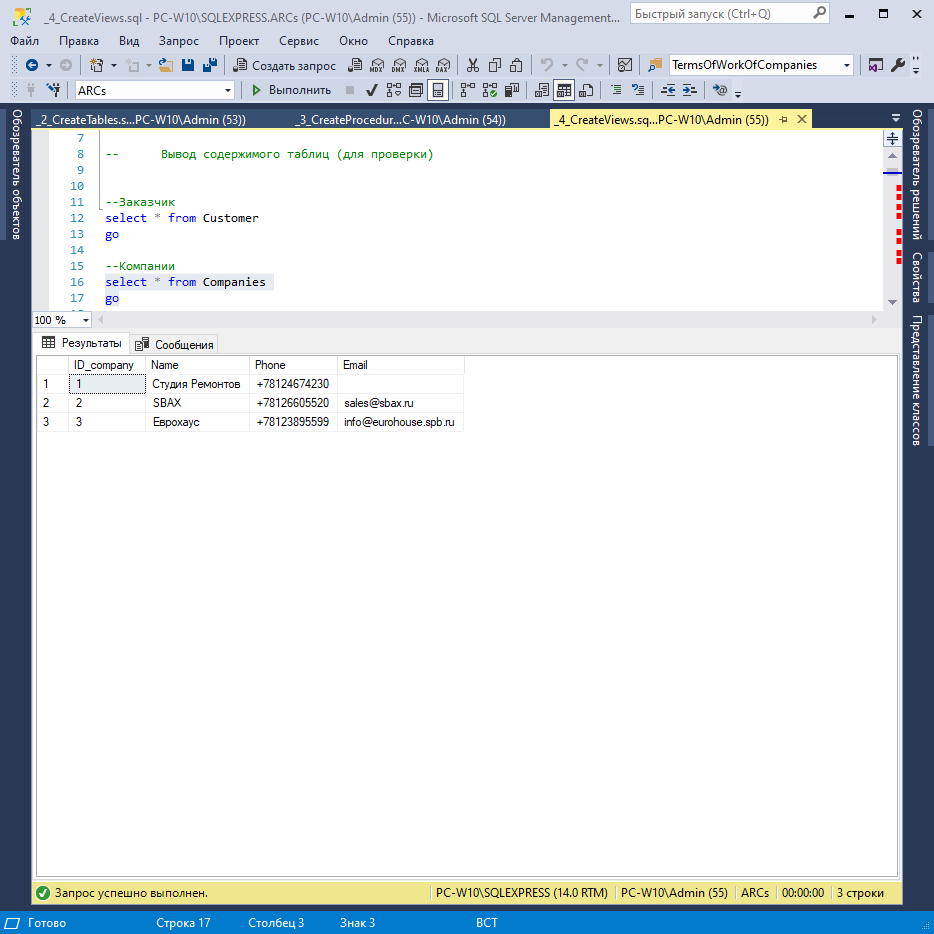


Рисунок 11 – Результат выполнения запроса

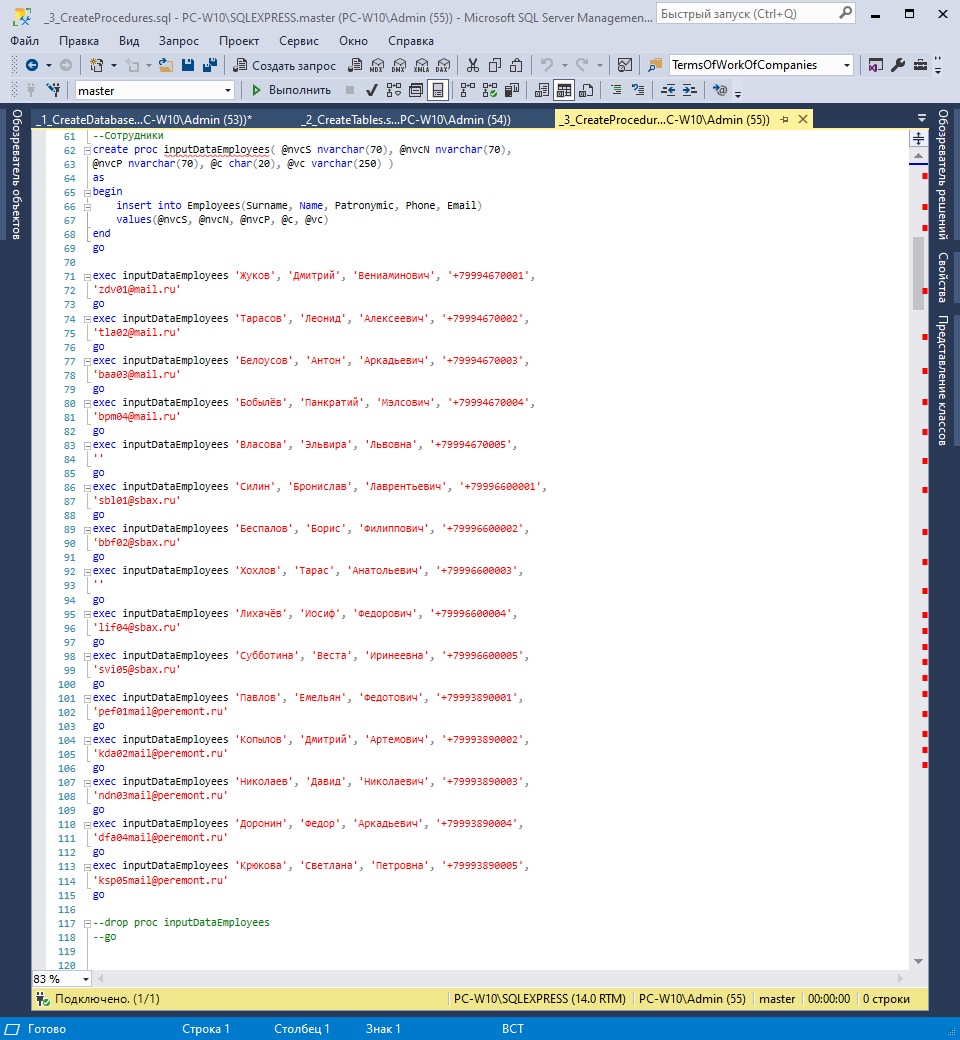


Рисунок 12 – Заполнение таблицы «Employees»

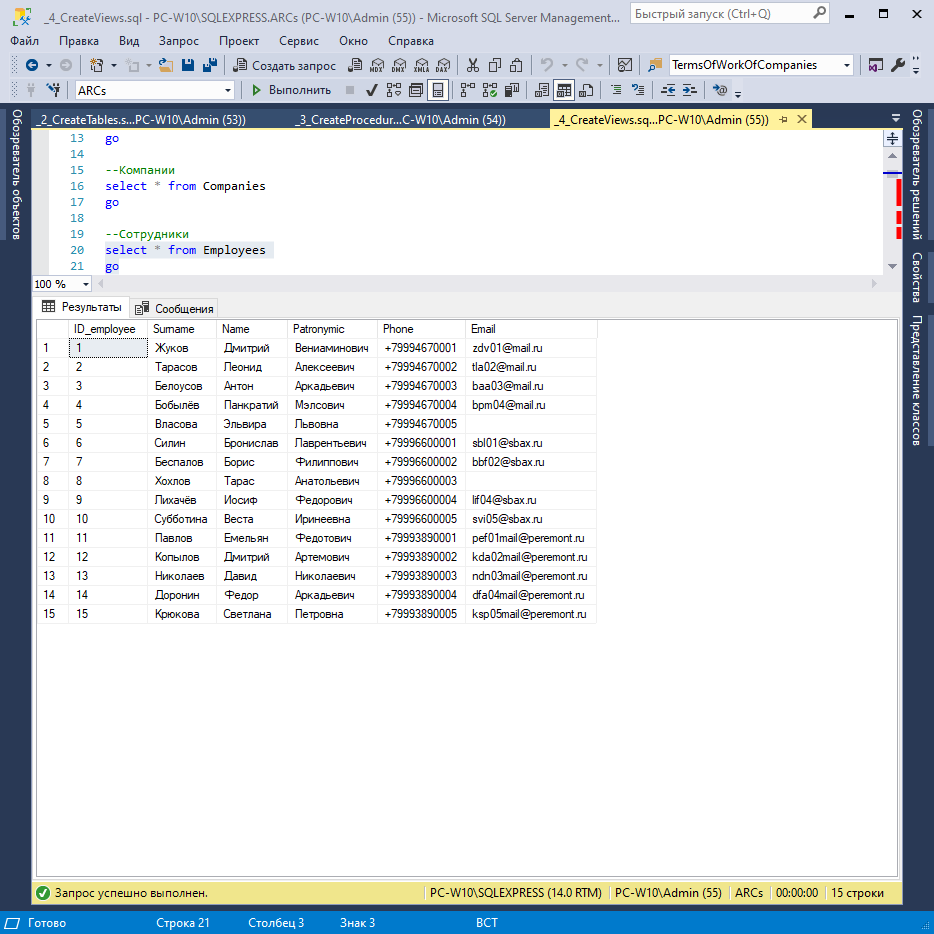


Рисунок 13 – Результат выполнения запроса

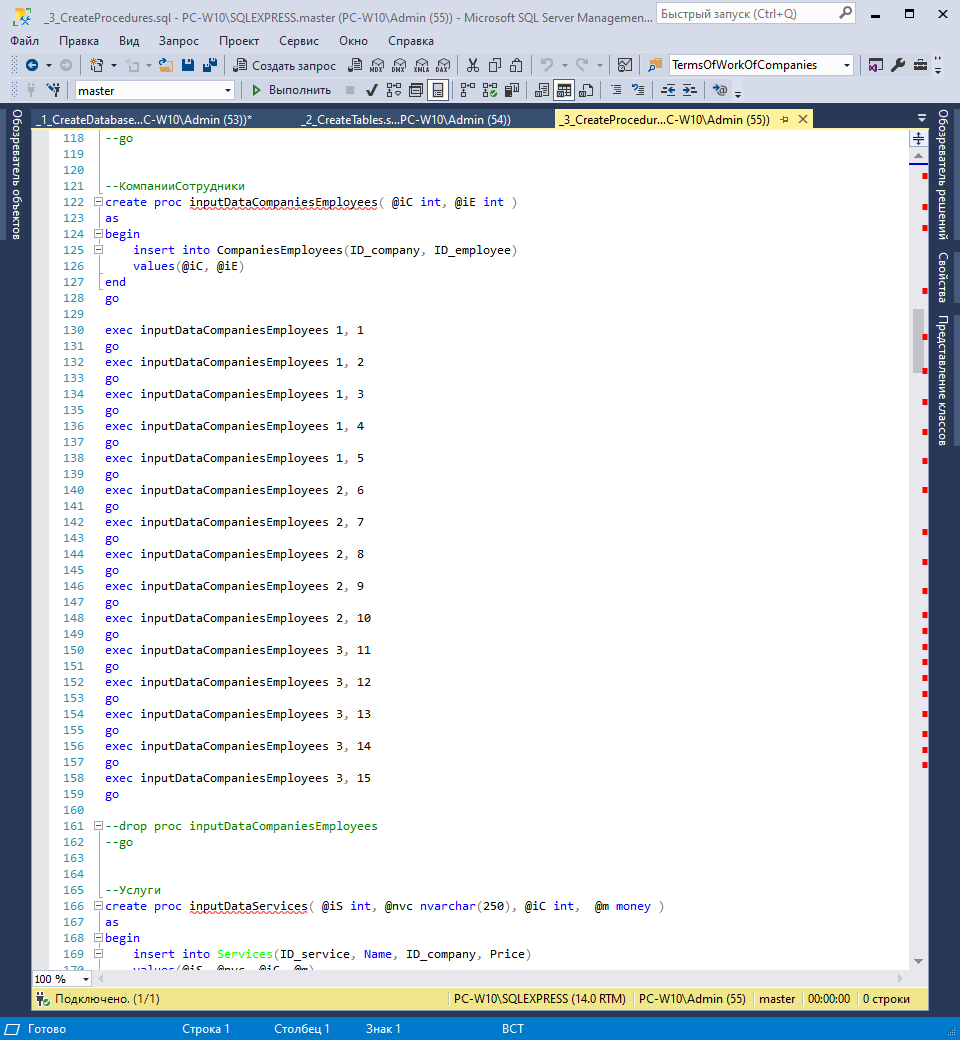


Рисунок 14 – Заполнение таблицы «CompaniesEmployees»

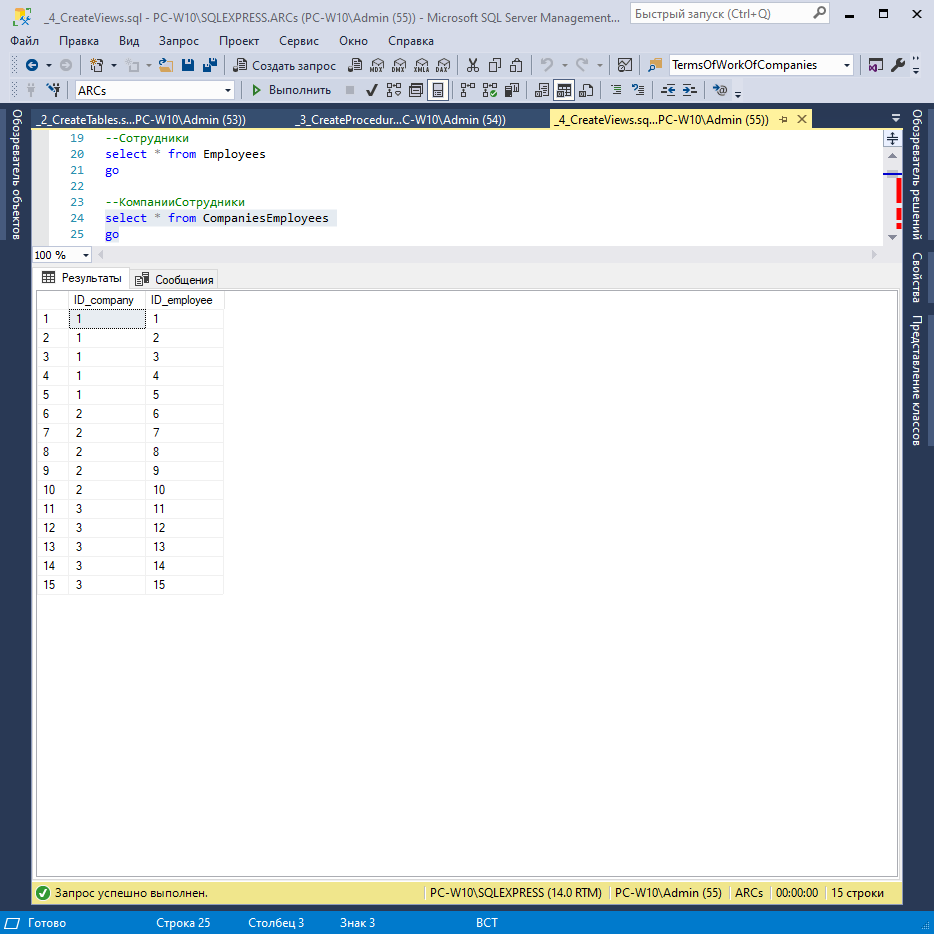


Рисунок 15 – Результат выполнения запроса

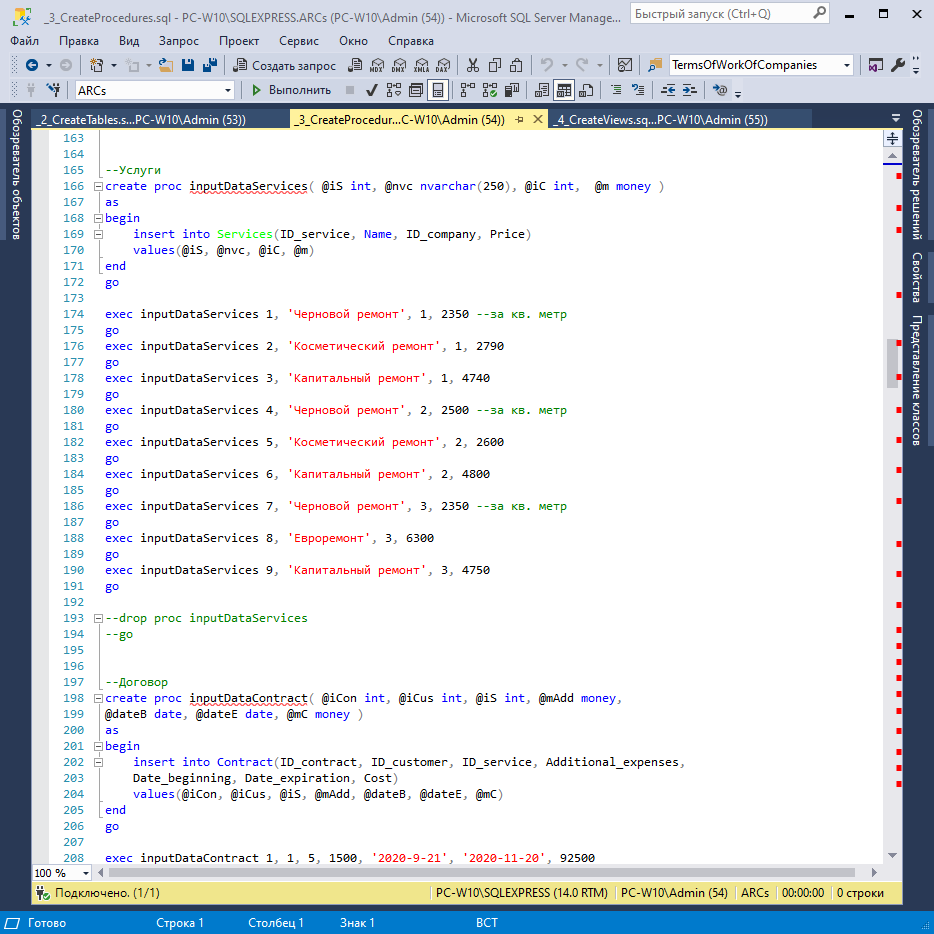


Рисунок 16 – Заполнение таблицы «Services»

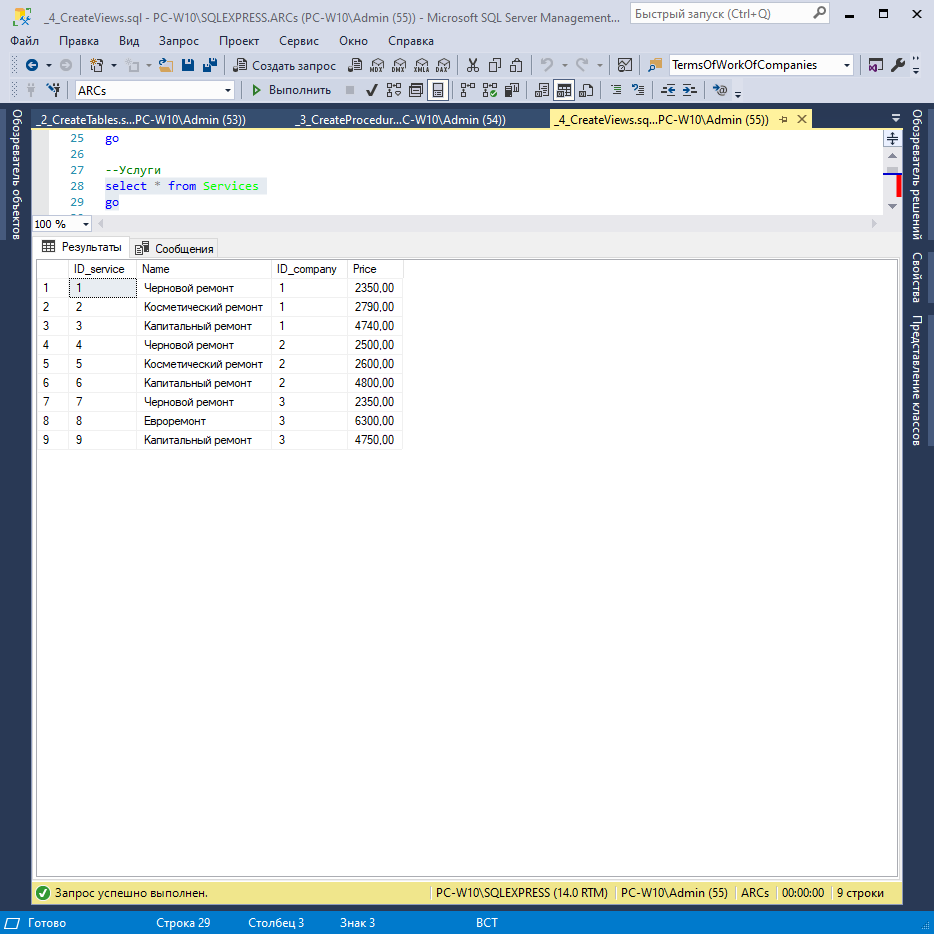


Рисунок 17 – Результат выполнения запроса

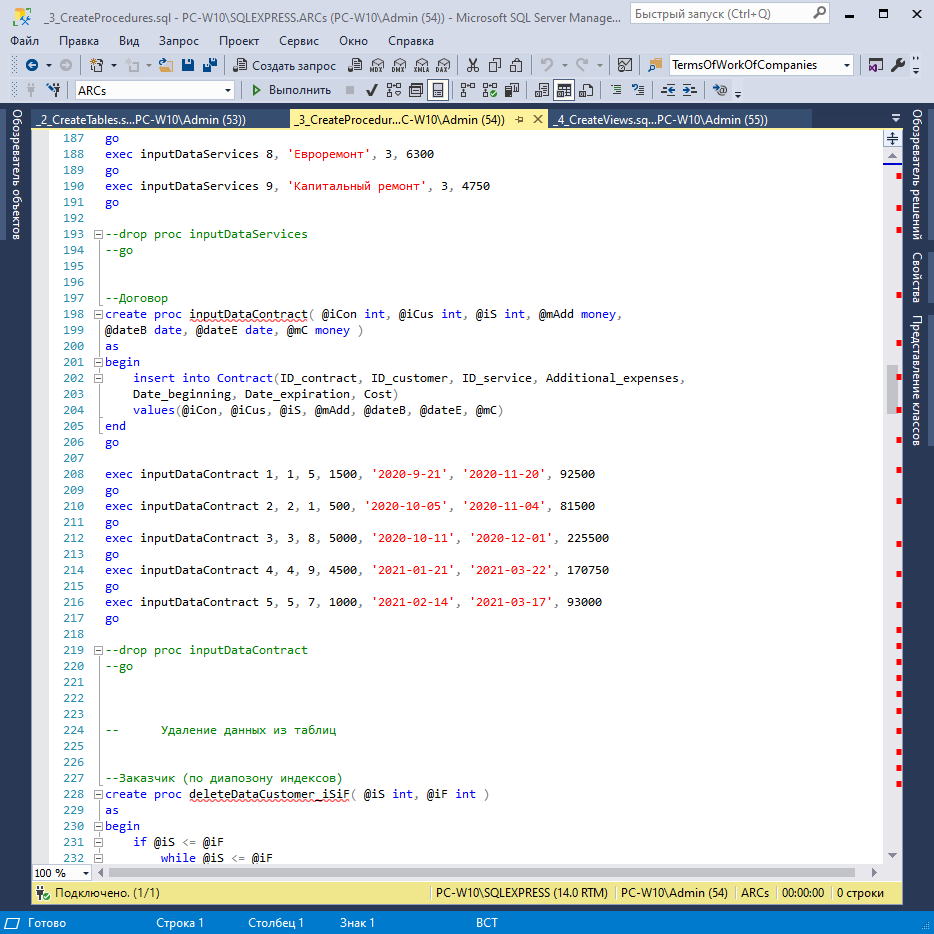


Рисунок 18 – Заполнение таблицы «Contract»

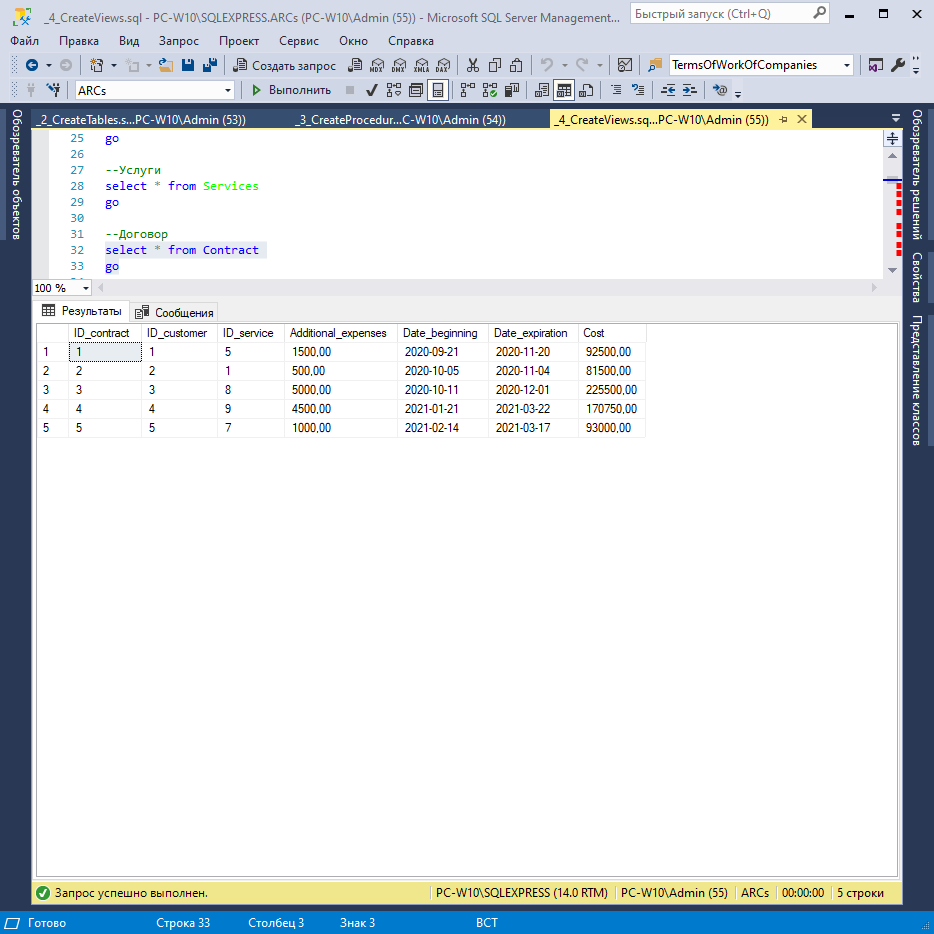


Рисунок 19 – Результат выполнения запроса

* 1. Разработка запросов и создание представлений к таблицам БД
     1. Представление на выборку данных из нескольких таблиц с использование условий отбора по нескольким полям

1. Вывод информации о компаниях и предоставляемых или услуг на рисунке 20.

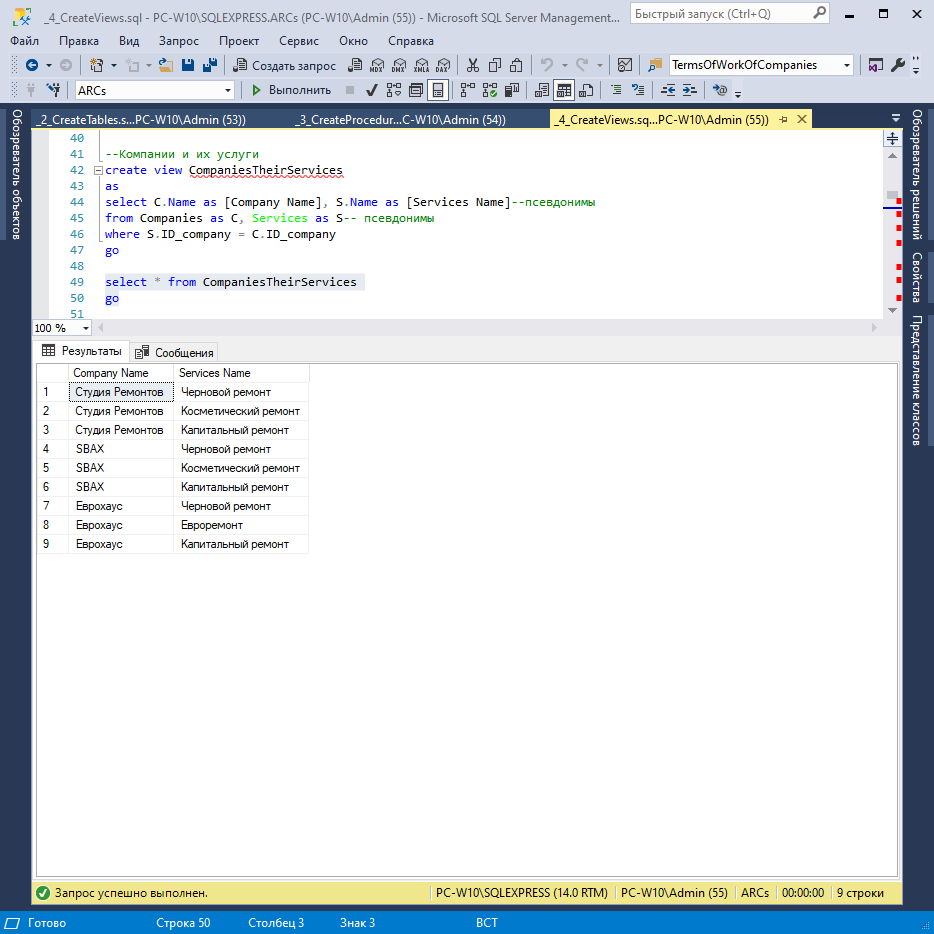


Рисунок 20 – Результат выполнения запроса 1

1. Вывод информации о заказчика без почты, также компаний на рисунке 21.

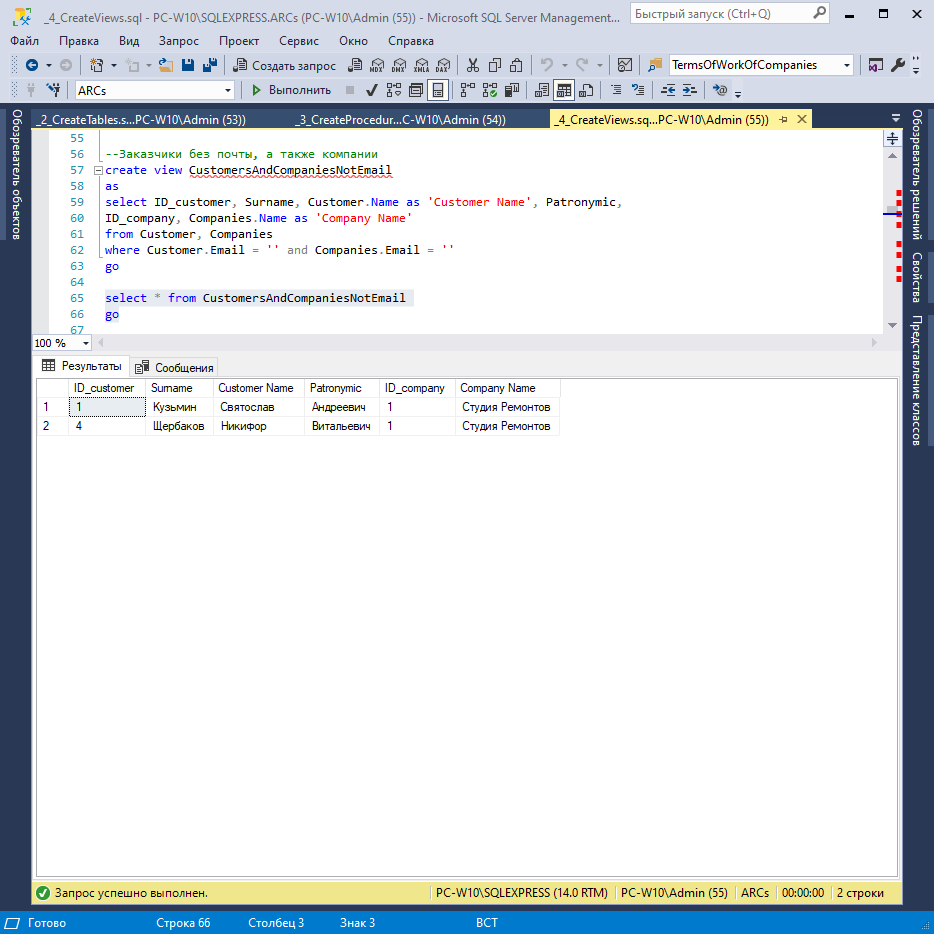


Рисунок 21 – Результат выполнения запроса 2

1. Вывод информации о компаниях и сотрудниках, которые там работают на рисунке 22.

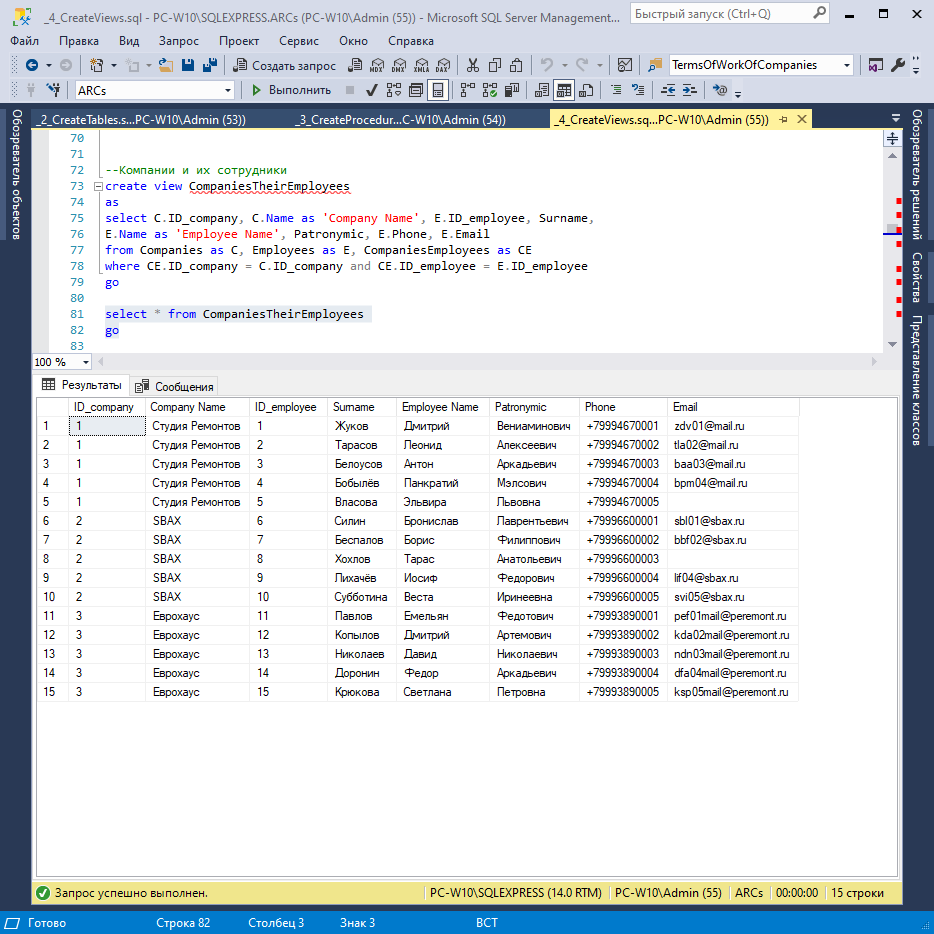


Рисунок 22 – Результат выполнения запроса 3

1. Компании с их услугами и сроками выполнения этих услуг по заказу за всё время на рисунке 23.

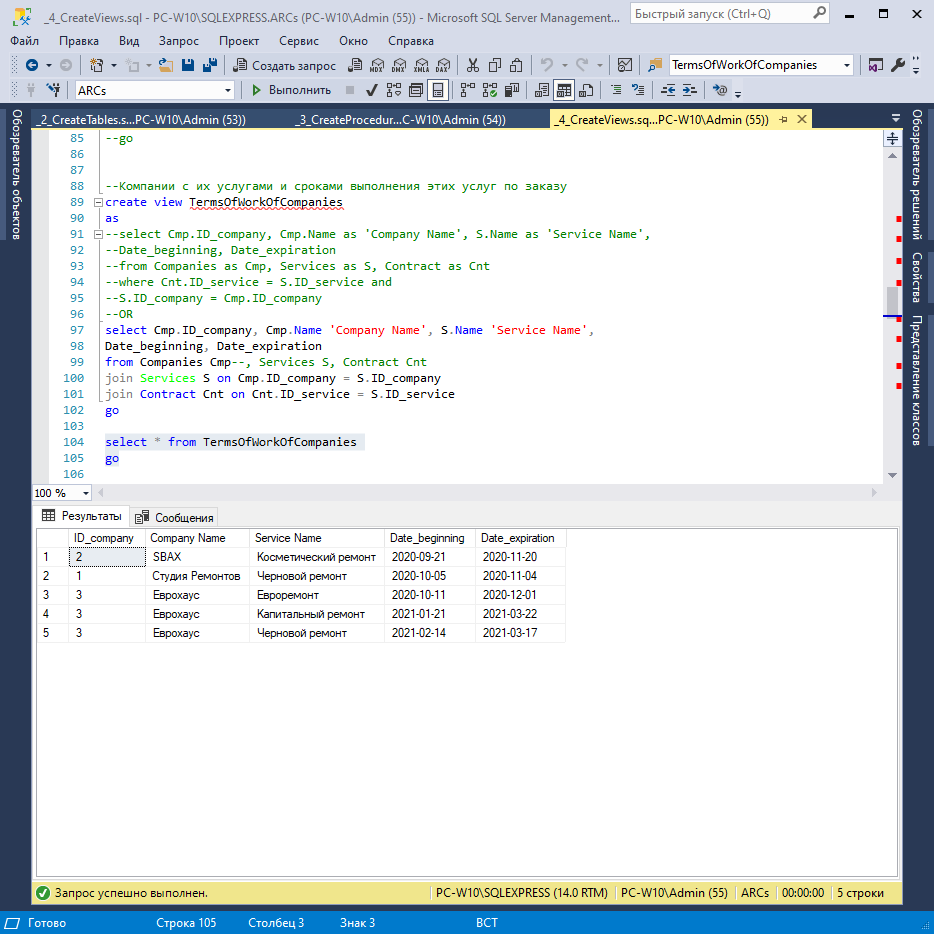


Рисунок 23 – Результат выполнения запроса 4

* + 1. Представления с использованием расчётных полей

1. Сумма дохода, если будут выполнены все заказы у всех компаний на рисунке 24.

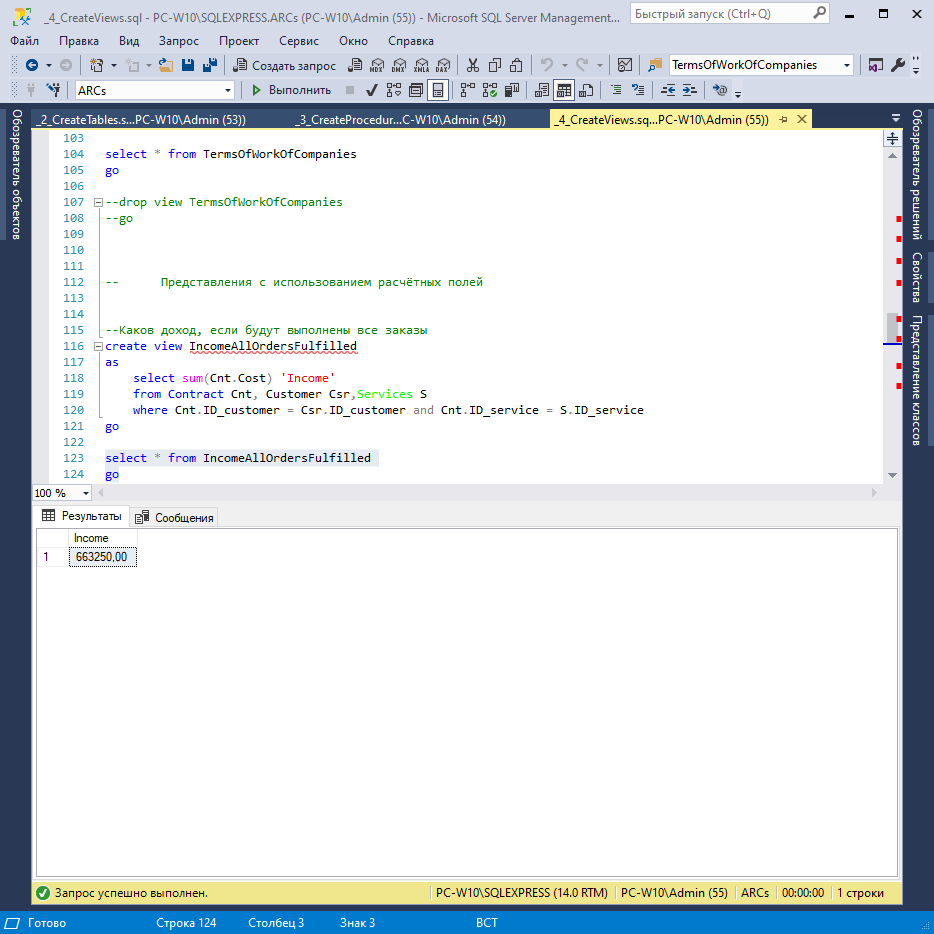


Рисунок 24 – Результат выполнения запроса 1

1. Сроки ремонтов в днях на рисунке 25.

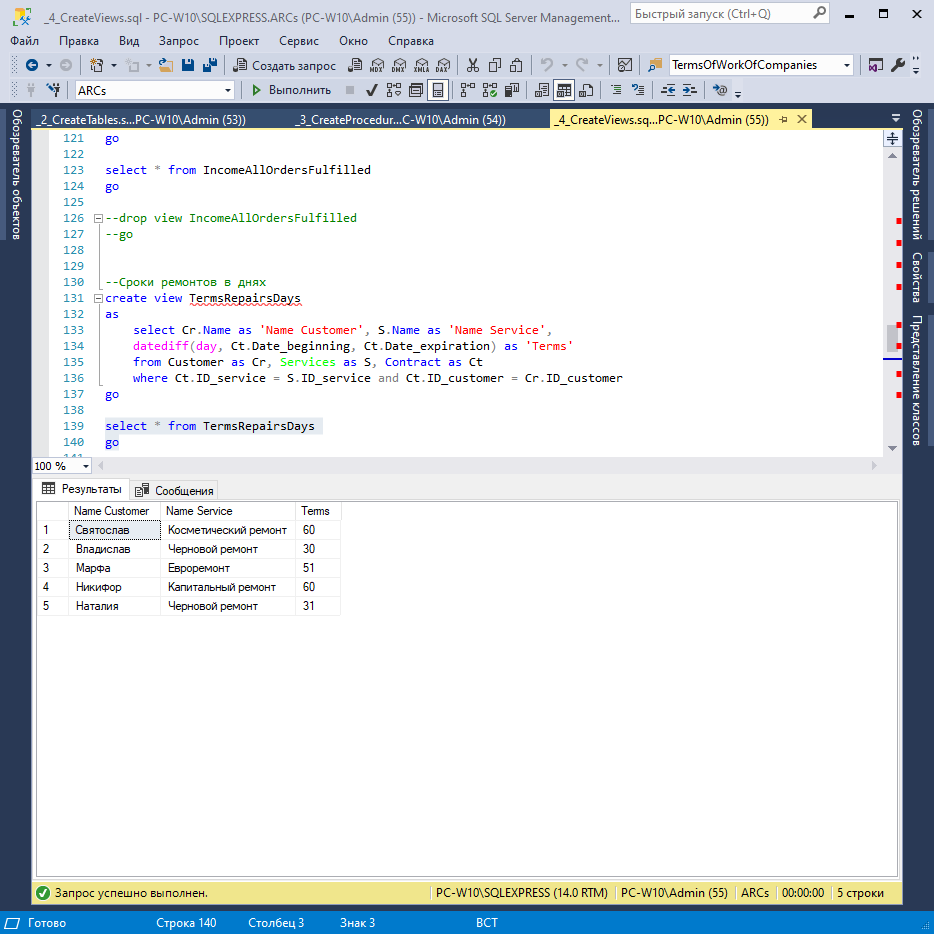


Рисунок 25 – Результат выполнения запроса 2

* + 1. Представления с использованием групповых операций

1. Количество каждой услуги на рисунке 26.

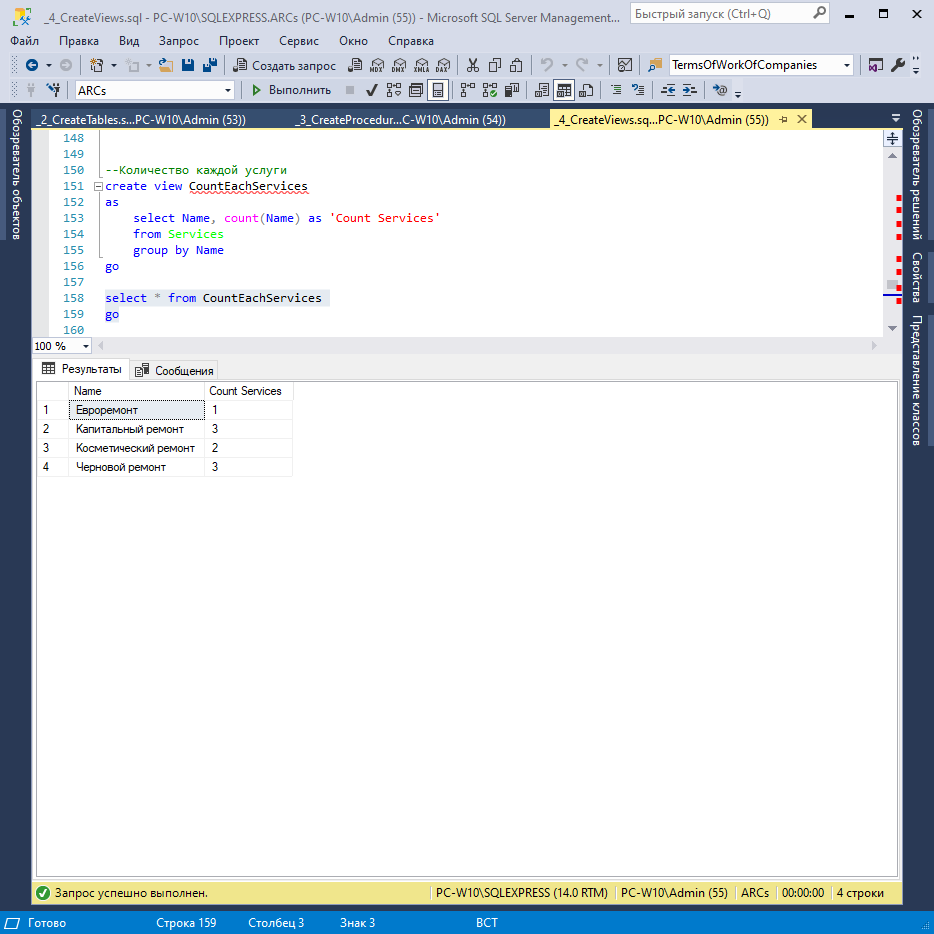


Рисунок 26 – Результат выполнения запроса 1

1. Сколько заказчиков у каждой компании на рисунке 27.

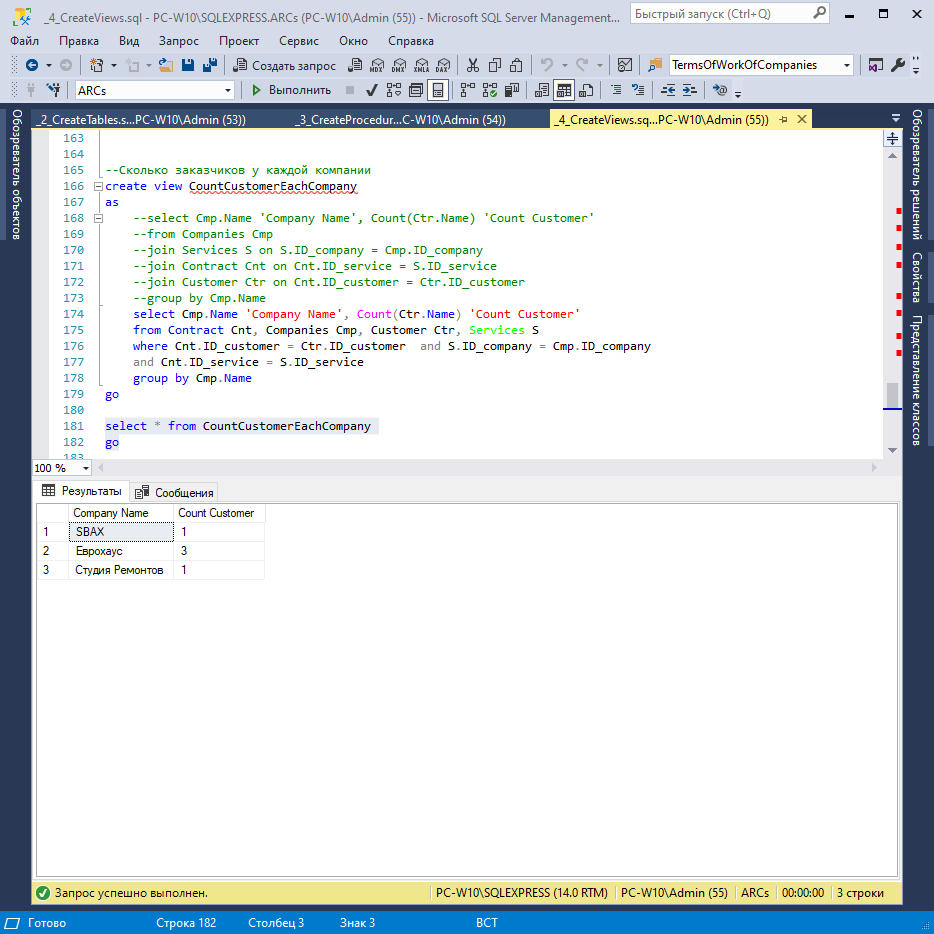


Рисунок 27 – Результат выполнения запроса 2

* 1. Разработка и создание процедур с параметрами (без параметров)
     1. Процедуры на ввод данных

Была рассмотрена в «3.2.Ввод данных с помощь процедур».

* + 1. Процедуры на обновление данных

Например, на рисунке 28 обновление данных о сотрудниках по указанному индексу и вводом данных.

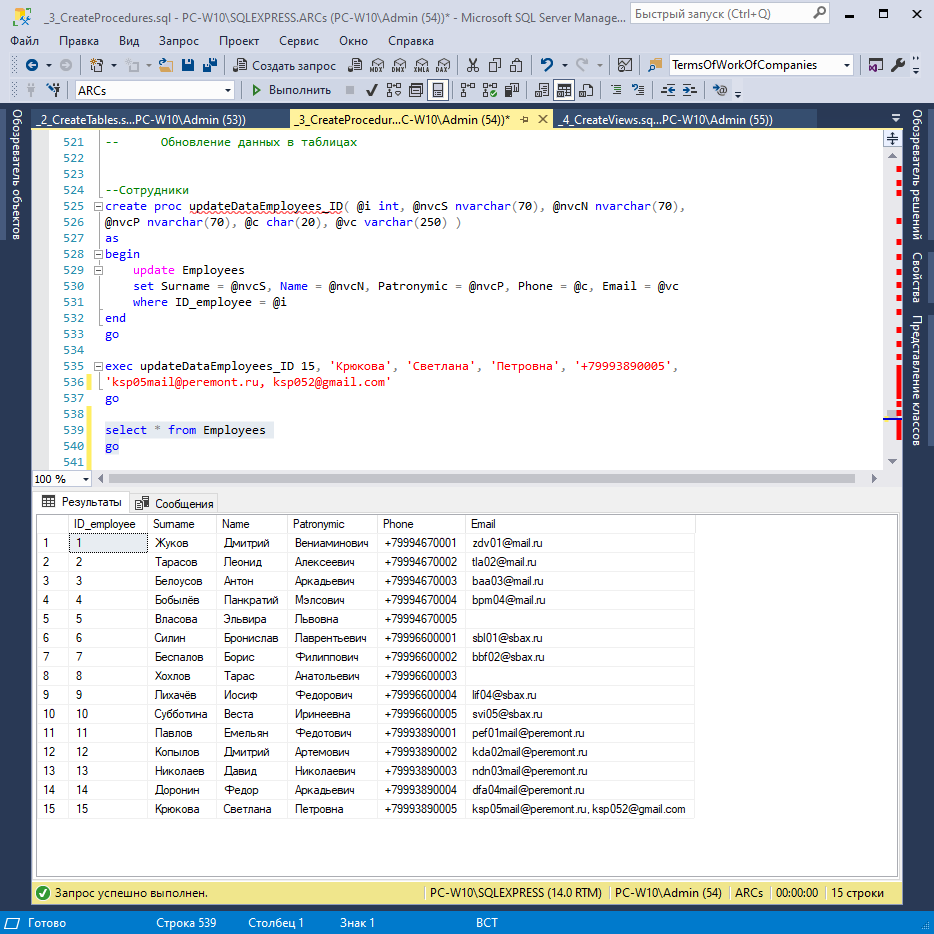


Рисунок 28 – Результат выполнения запроса на обновление

Установка нового или сброс идентификатора у таблицы «Employees» (Сотрудники) на рисунке 29.

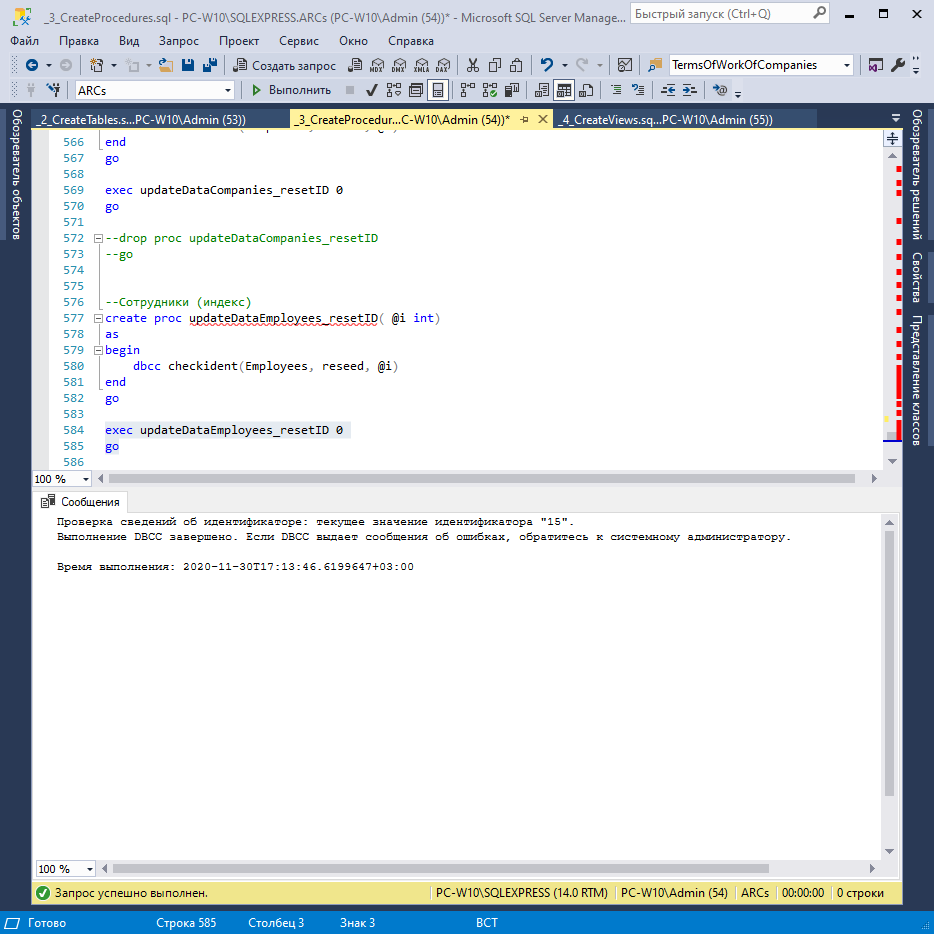


Рисунок 29 – Результат выполнения запроса на обновление

* + 1. Процедуры на удаление данных

Процедура без параметров на удаление всех записей из таблицы «Employees» (Сотрудники) на рисунке 30.

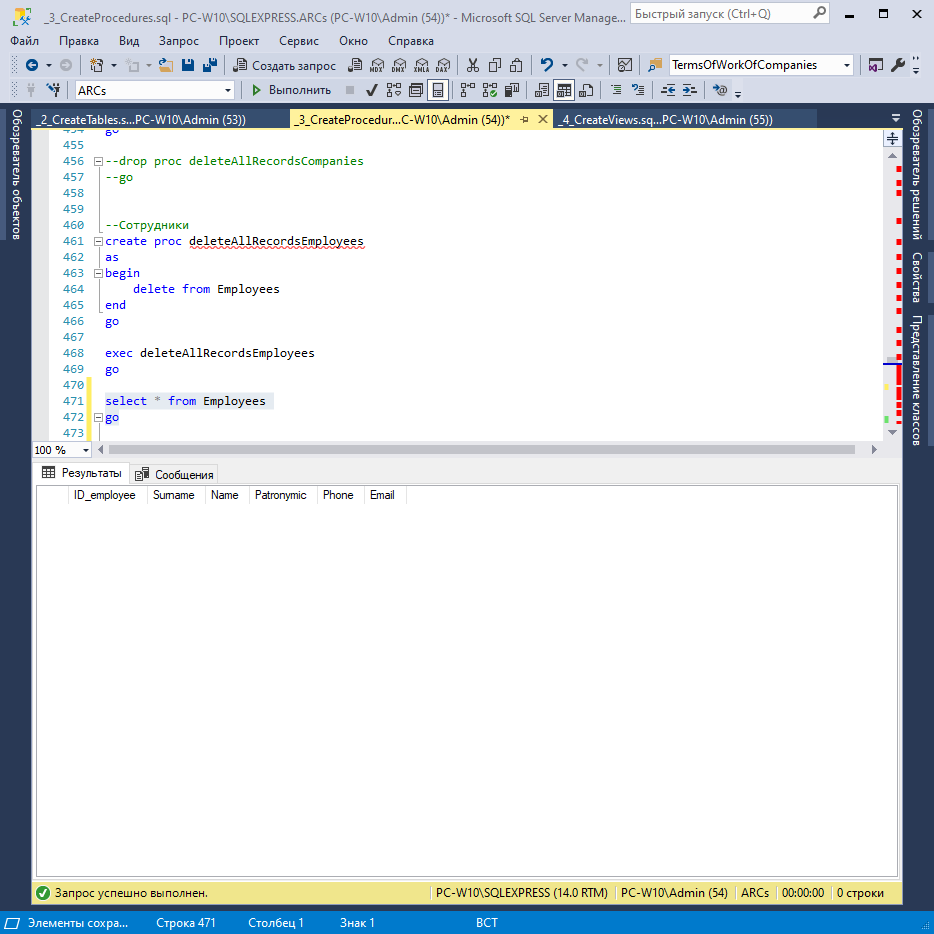


Рисунок 30 – Результат выполнения запроса на удаление

Удаление с параметрами для указания диапозона индекса начала (4) и конца удаления (9) сотрудников на рисунке 31 (процедура с исправлением ошибки индекса, если начало будет больше конца).

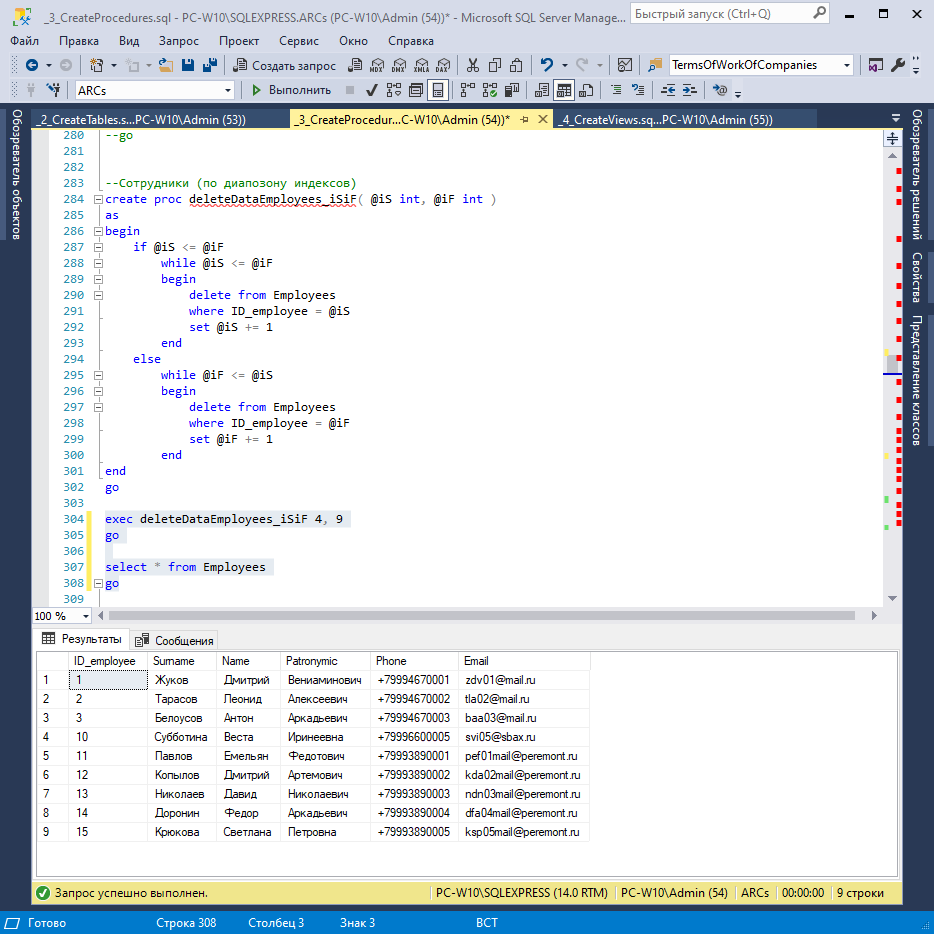


Рисунок 31 – Результат выполнения запроса на удаление

* 1. Описание пользователей и прав их доступа к объектам БД.

Создание ролей и определение их прав в БД

В базе данных «Компании по ремонту квартир» существует три типа пользователей:

1. Менеджер (Manager) – пользователи, которые отвечают за управление базой в целом, имеет доступ ко всему, могут передавать права только на резервное копирование.
2. Сотрудник (Employee) – пользователи, которые работают определёнными таблицами, представлениями, процедурами и могут осуществлять резервное копирование. По сути управление с таблицами связанными напрямую с заказчиками: «Customer» (Заказчик) и «Contract» (Договор).
3. Заказчик или клиент (Client) – пользователи, которые могут только просматривать доступные им таблицы, такие как информацию о компаниях и их услуги.

Создание учётных записей, к ним соответствующих пользователей, а также ролей с определёнными разрешениями на рисунках 32-40.

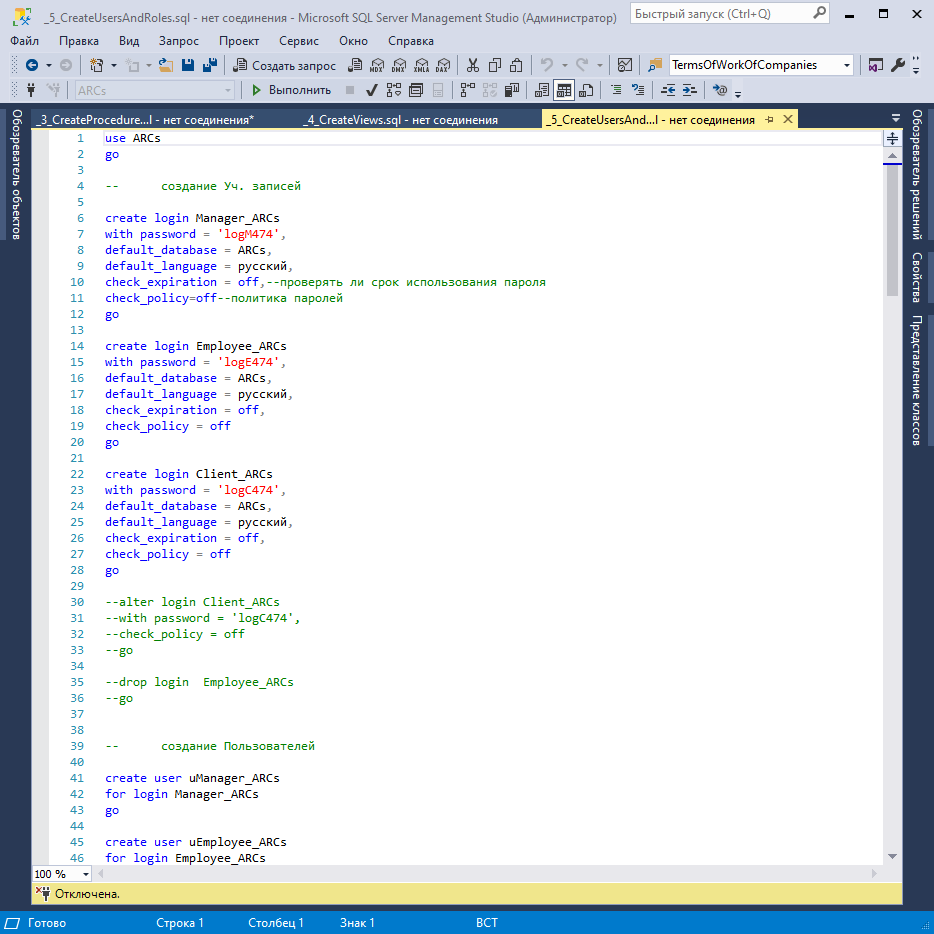


Рисунок 32 – Создание учётных записей SQL Server

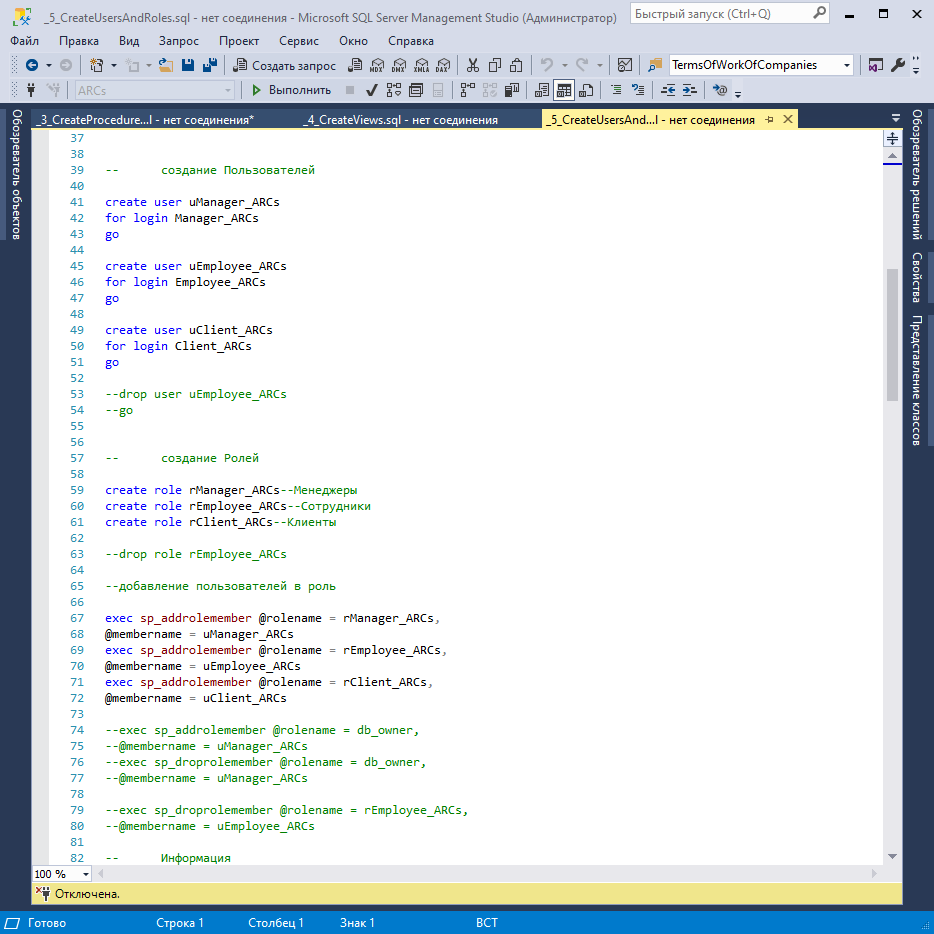


Рисунок 33 – Создание пользователей, ролей и добавление ролей пользователям

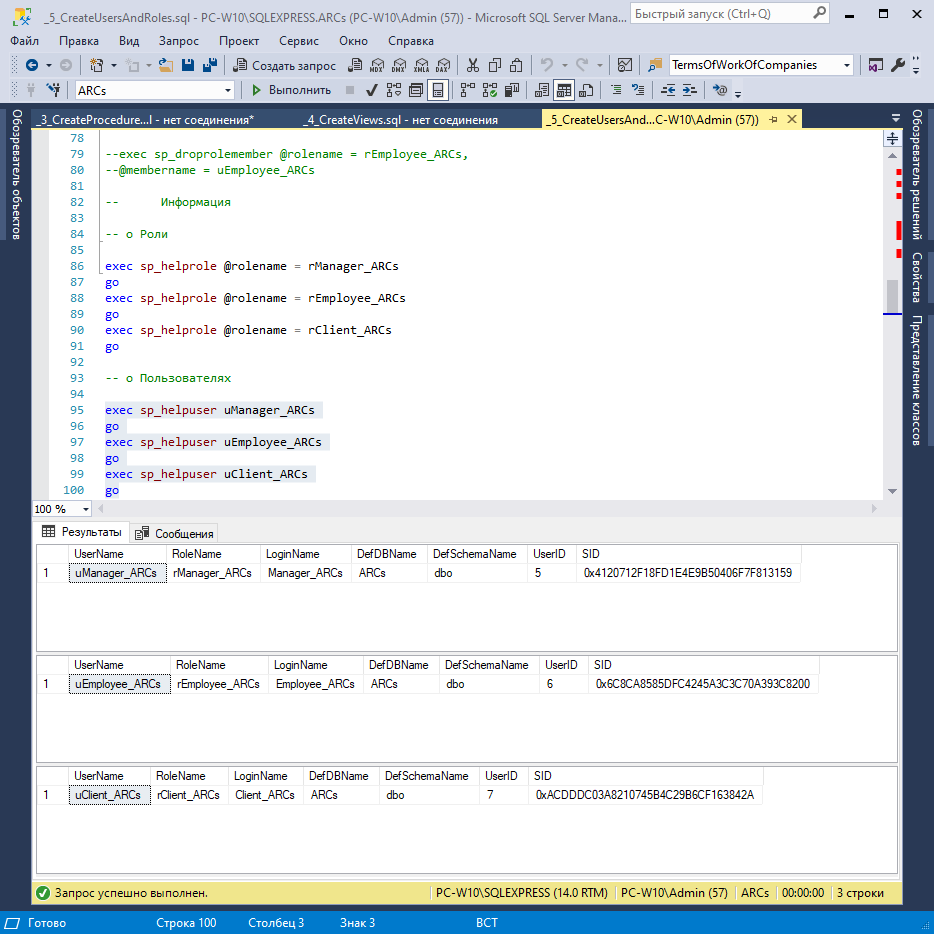


Рисунок 34 – Созданные пользователи

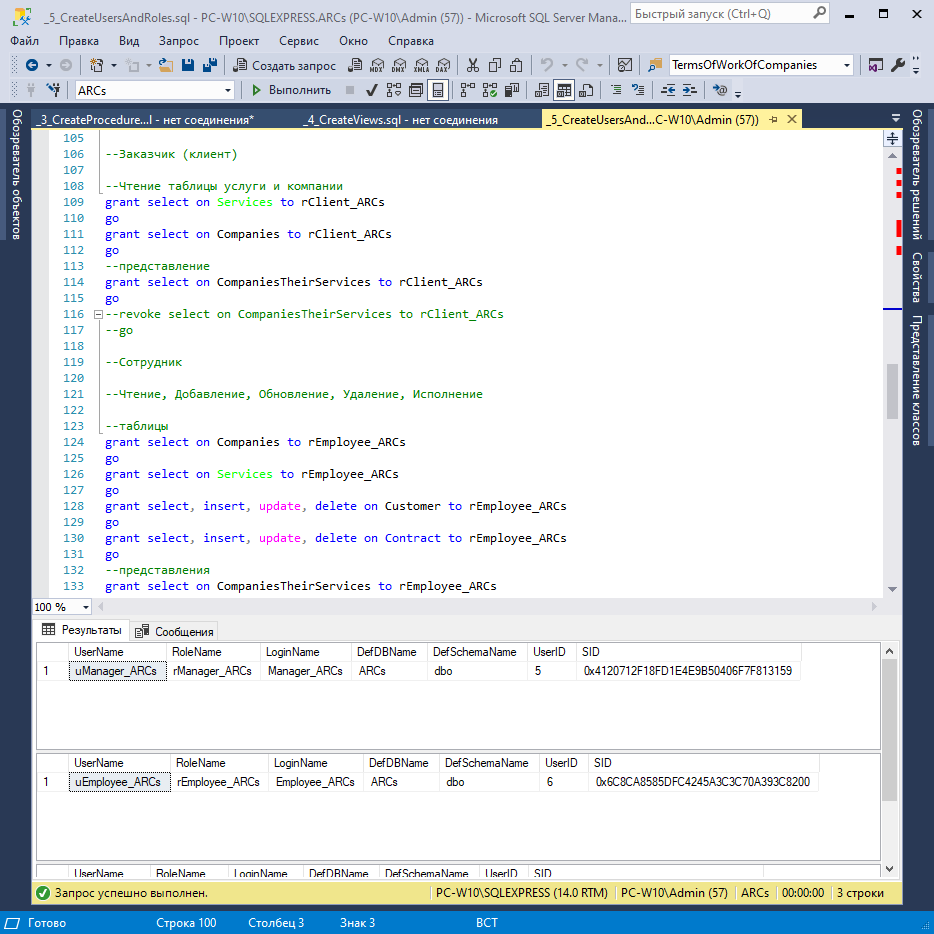


Рисунок 35 – Выдача прав роли соответствующей клиенту

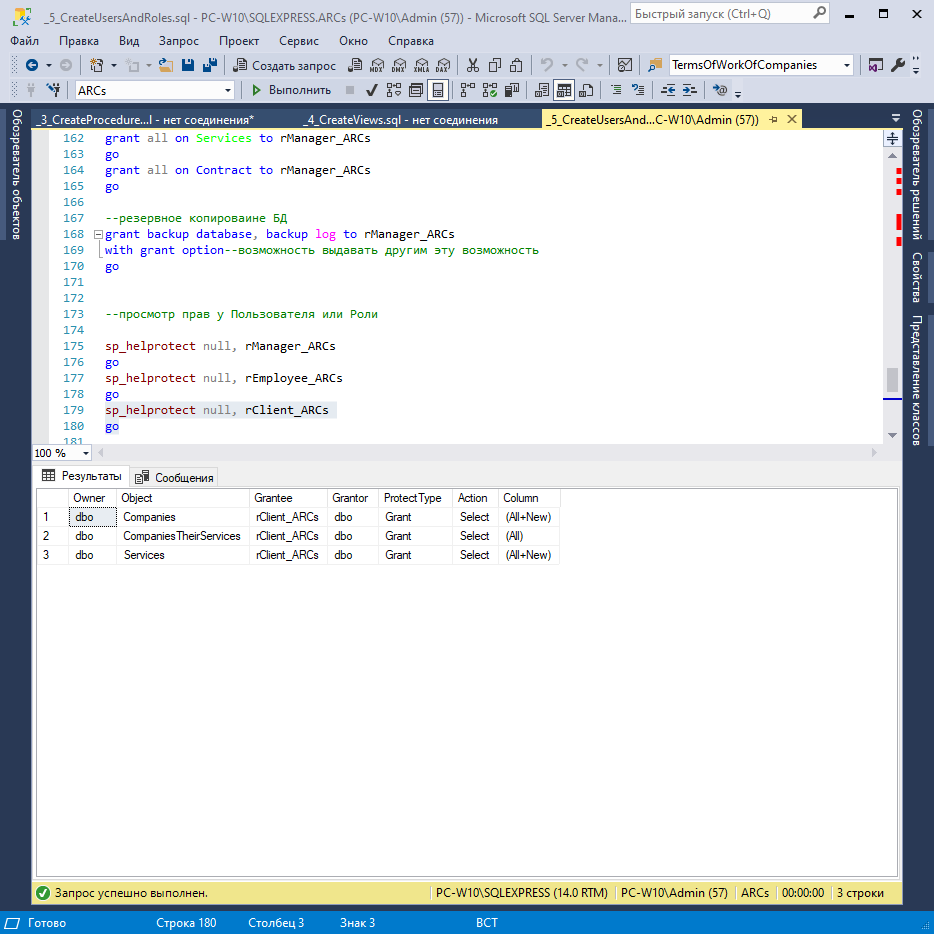


Рисунок 36 – Проверка прав у клиента

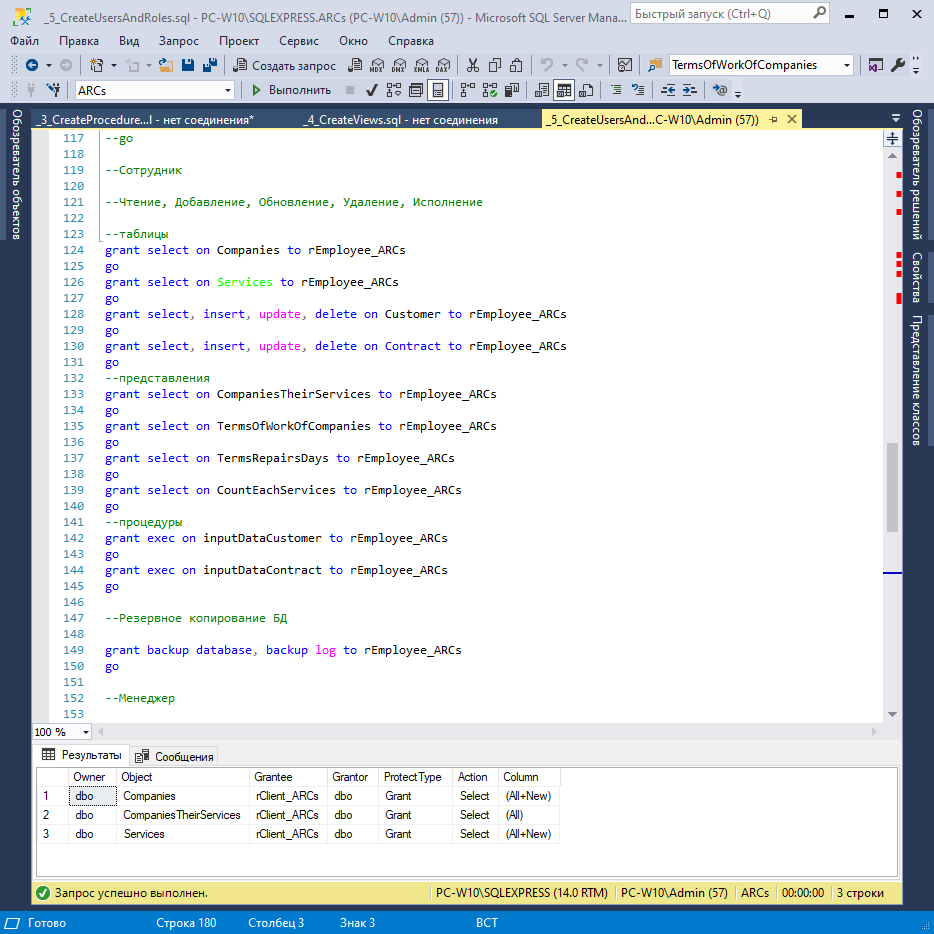


Рисунок 37 – Выдача прав роли соответствующей сотруднику

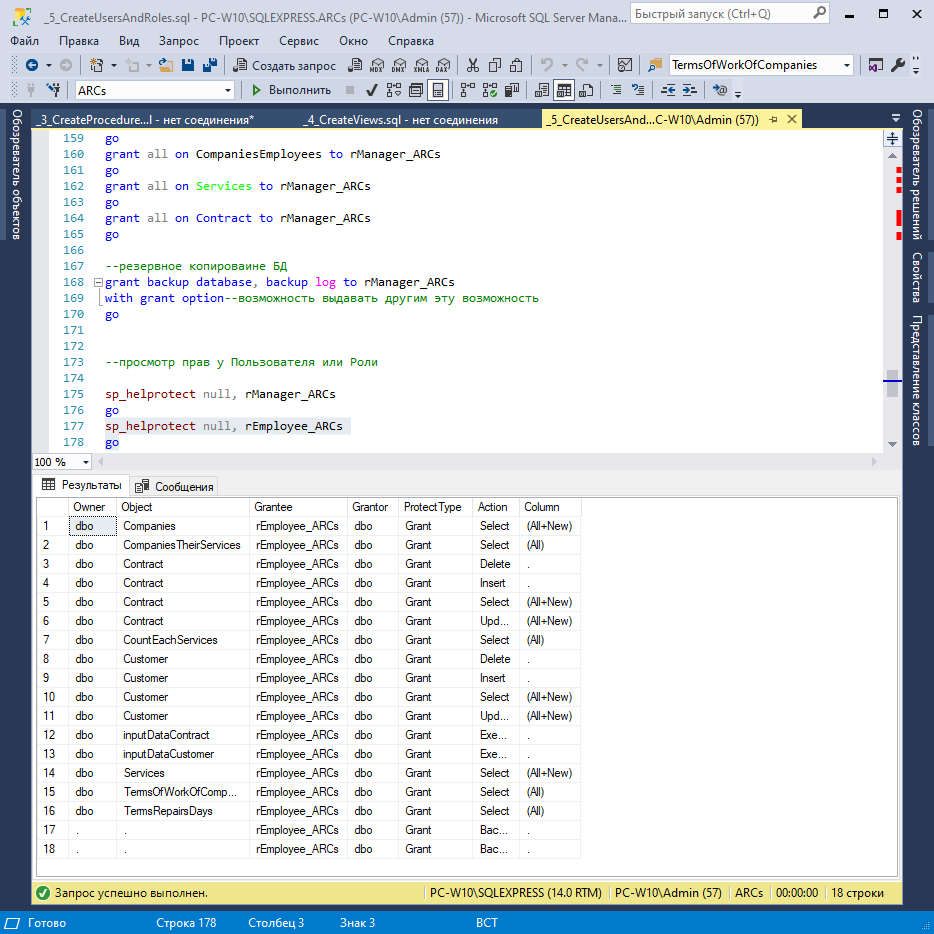


Рисунок 38 – Проверка прав у сотрудника

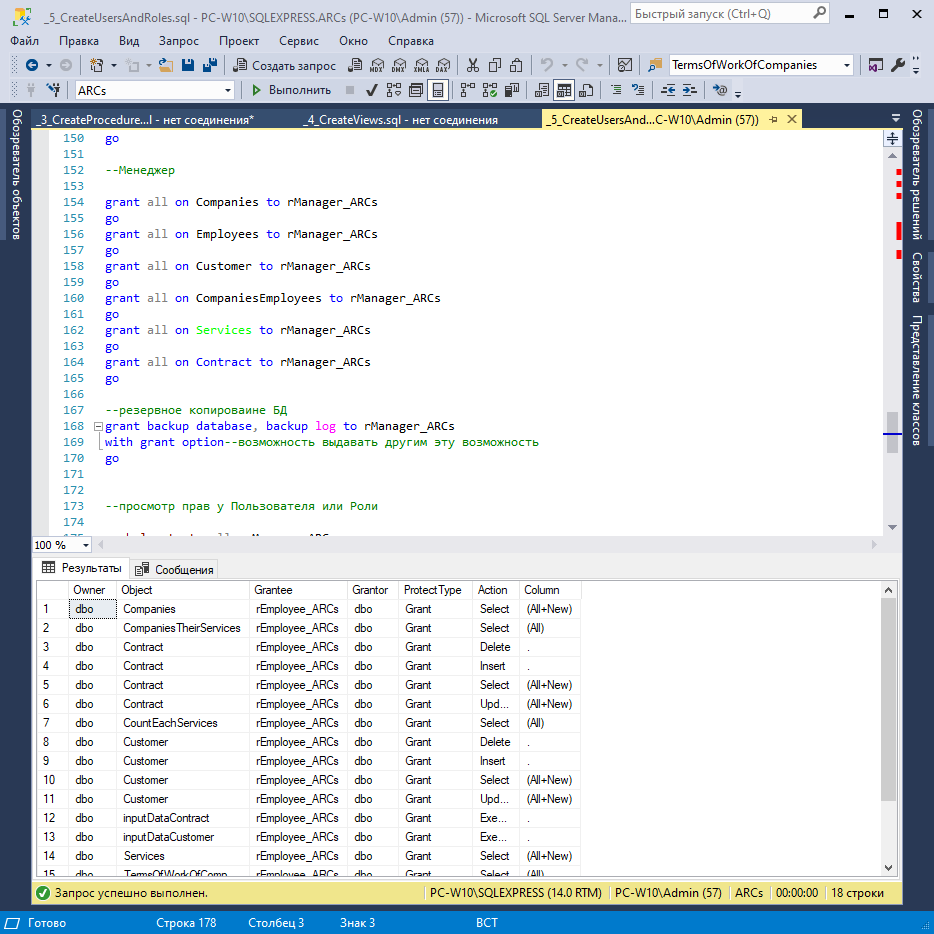


Рисунок 39 – Выдача прав роли соответствующей менеджеру

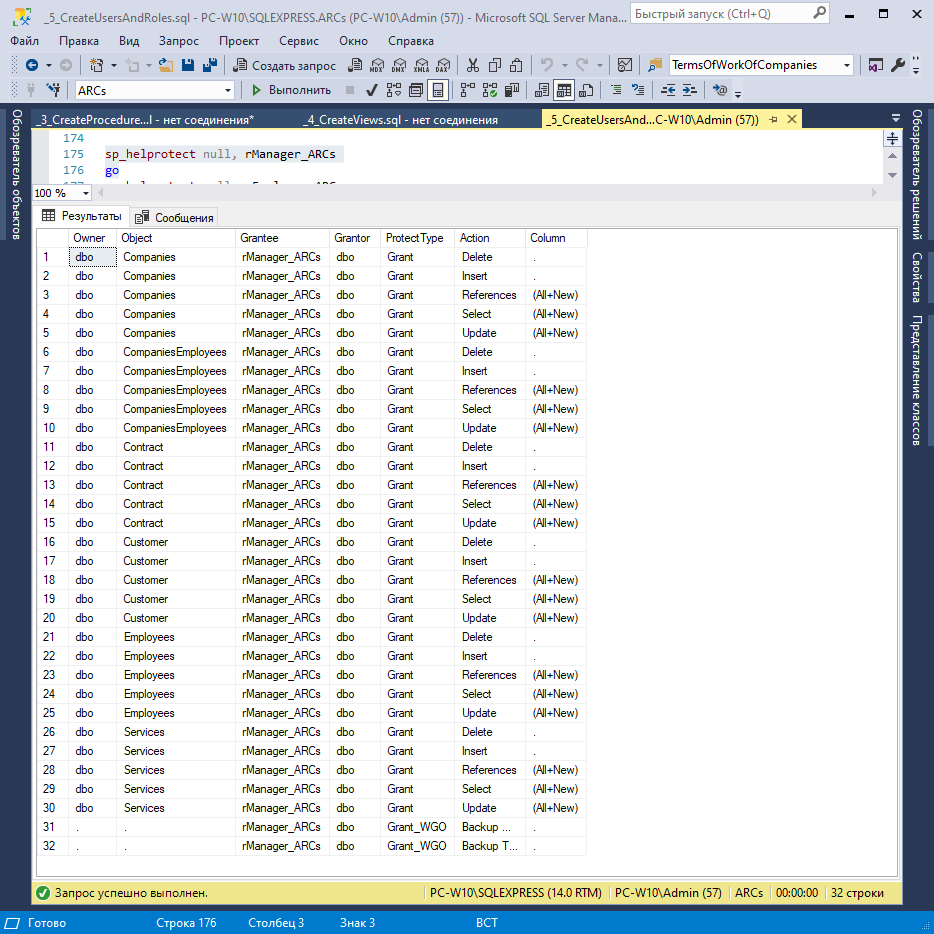


Рисунок 40 – Проверка прав у менеджера

* 1. Разработка стратегии резервного копирования БД

Для полной, разностной и журнала транзакций резервных копий, создано два устройства резервного копирования. Затем были выполнено копирование, где полная и разностная хранится на устройстве одном, а журналов транзакций на другом, смотрите ни рисунке 41-42.

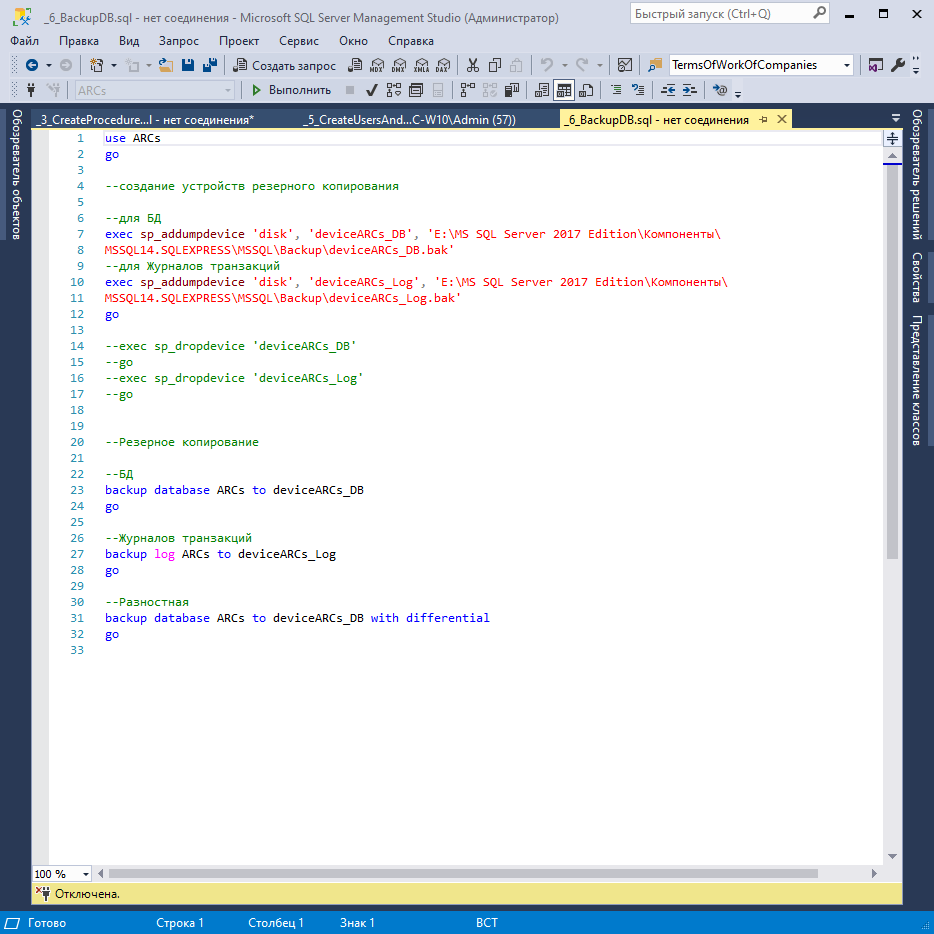


Рисунок 41 – Создание устройств и резервное копирование

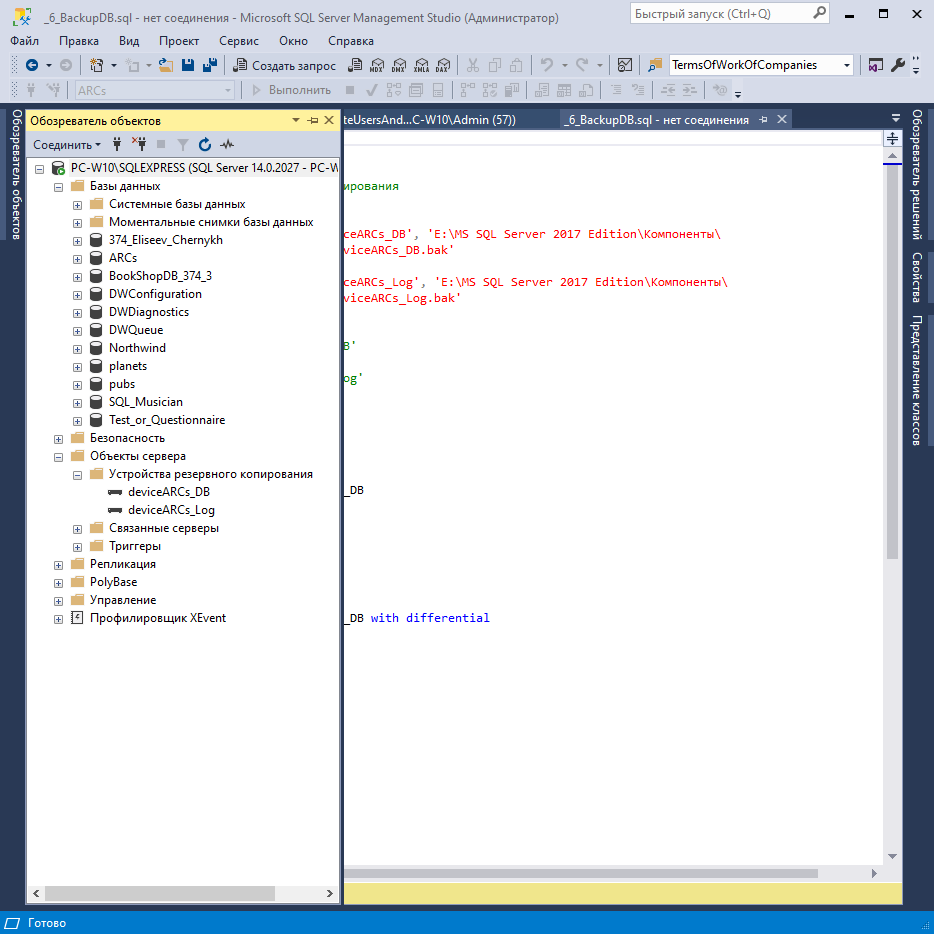


Рисунок 42 – Проверка устройств резервного копирования

Рекомендуется для данной БД делать копию журнала транзакций каждый день, а полную резервную копию раз в неделю.

* 1. Тестирование работы БД

Начнём тестирования БД с клиента у которого всё ограниченно, кроме просмотра таблиц «Companies», «Services» и некоторых процедур, смотрите на рисунках 43, 44.

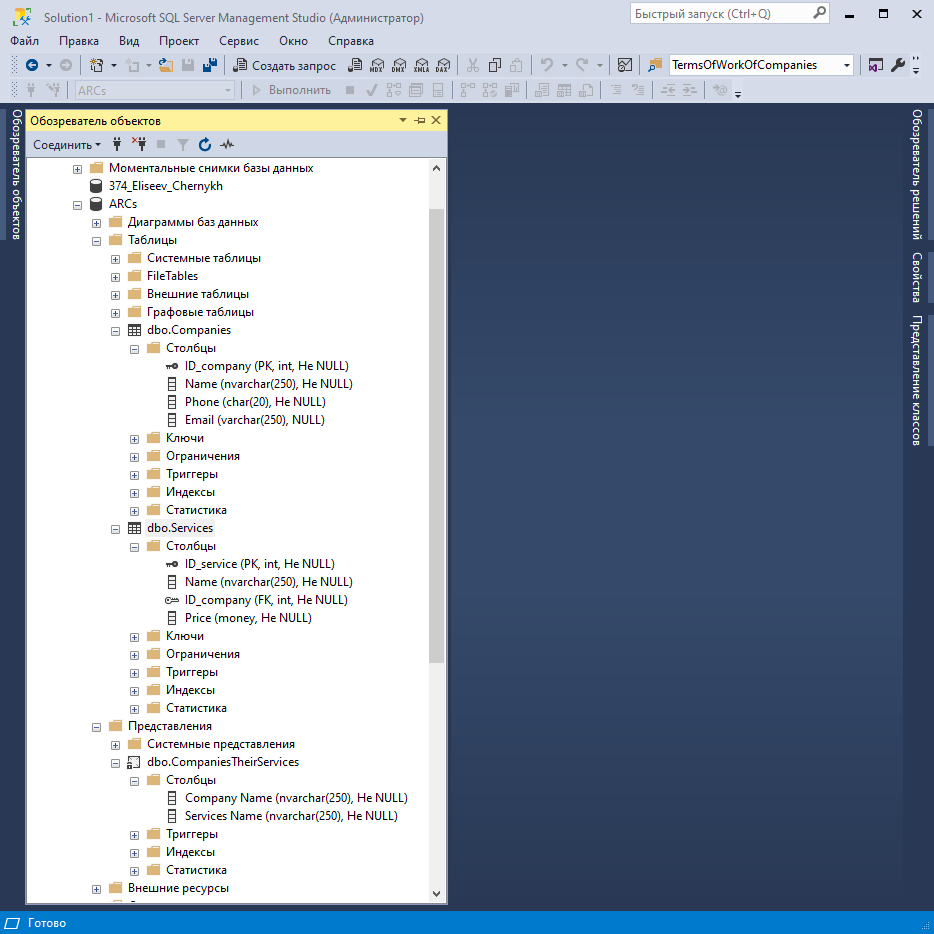


Рисунок 43 – Доступные таблицы и представления клиенту

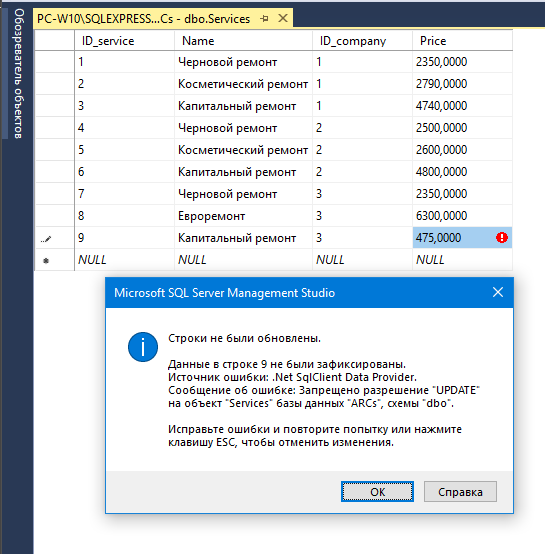


Рисунок 44 – Попытка пользователем изменить данные

Тестирование БД через пользователя сотрудник на рисунках 45-47.

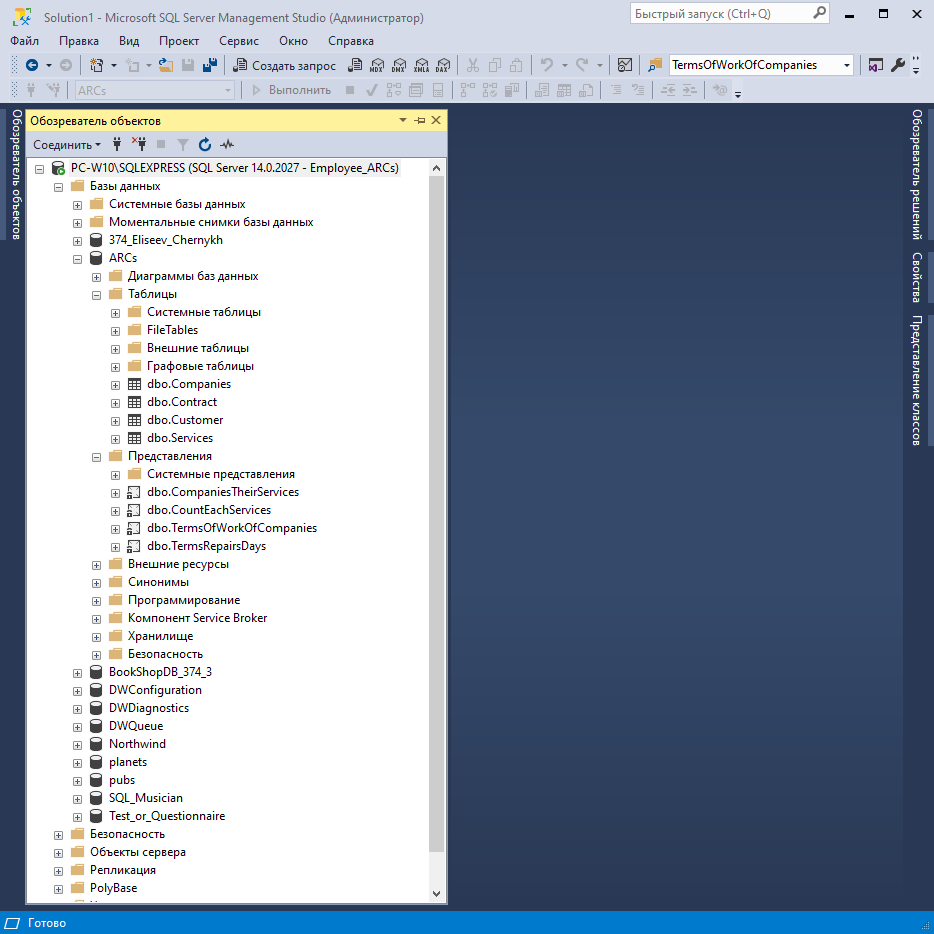


Рисунок 45 – Доступные таблицы и представления сотруднику

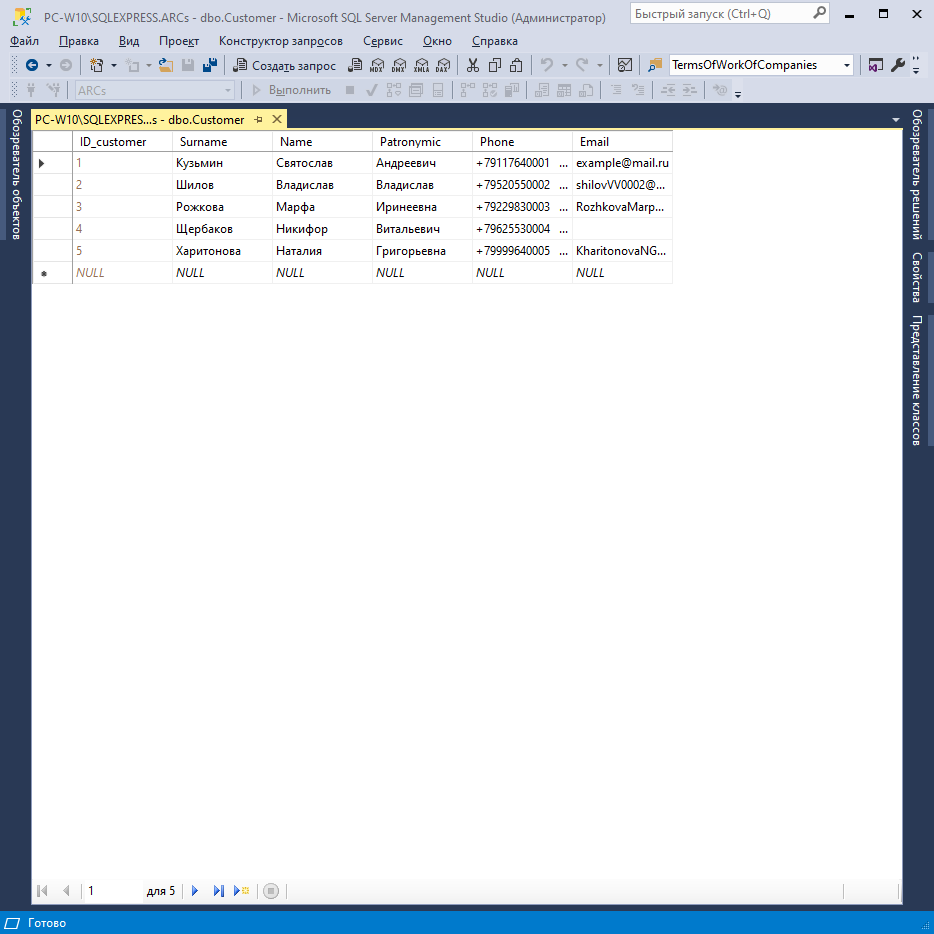


Рисунок 46 – Успешно добавлена почта «example@mail.ru»

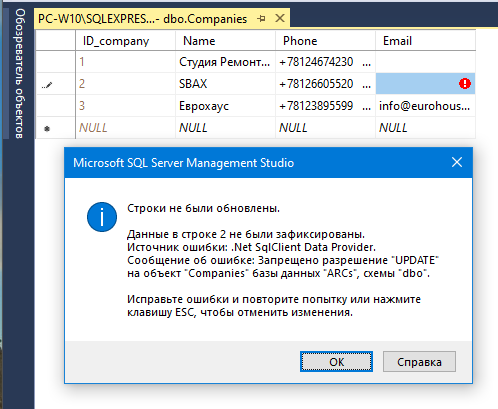


Рисунок 47 – Попытка удалить почту у компании «SBAX» не удалась

Тестирование БД через пользователя менеджера на рисунках 48-49.

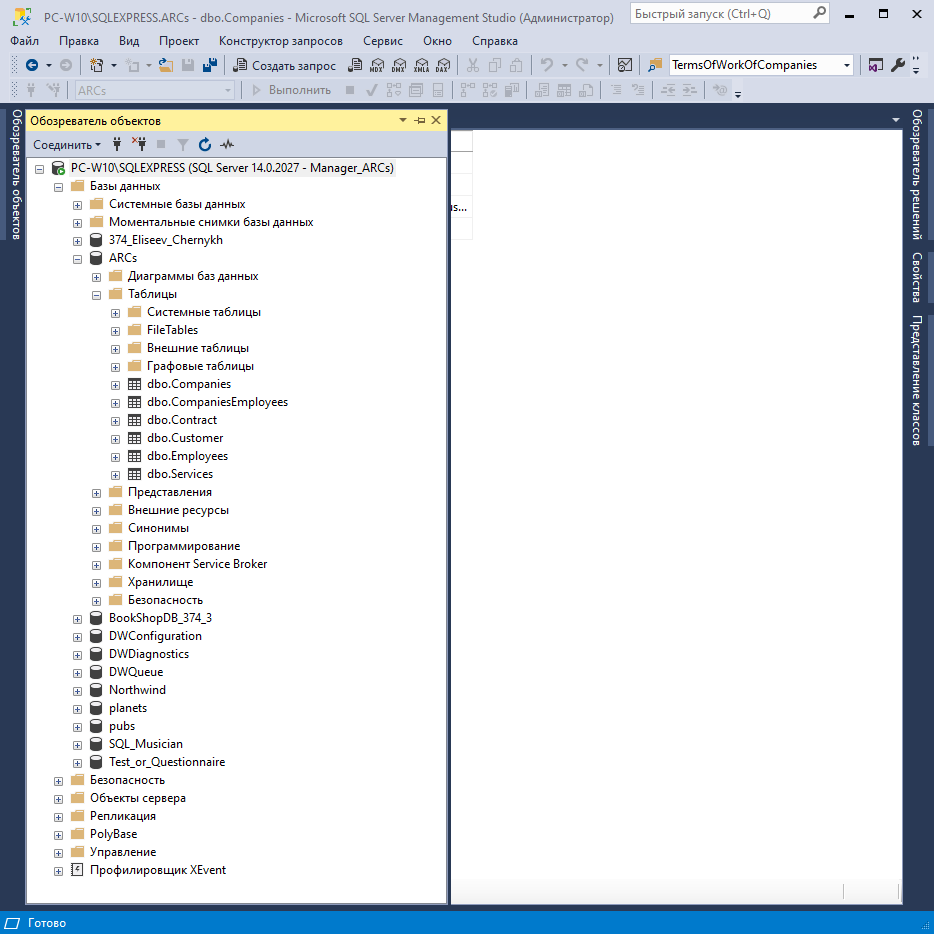


Рисунок 48 – Доступные таблицы менеджеру

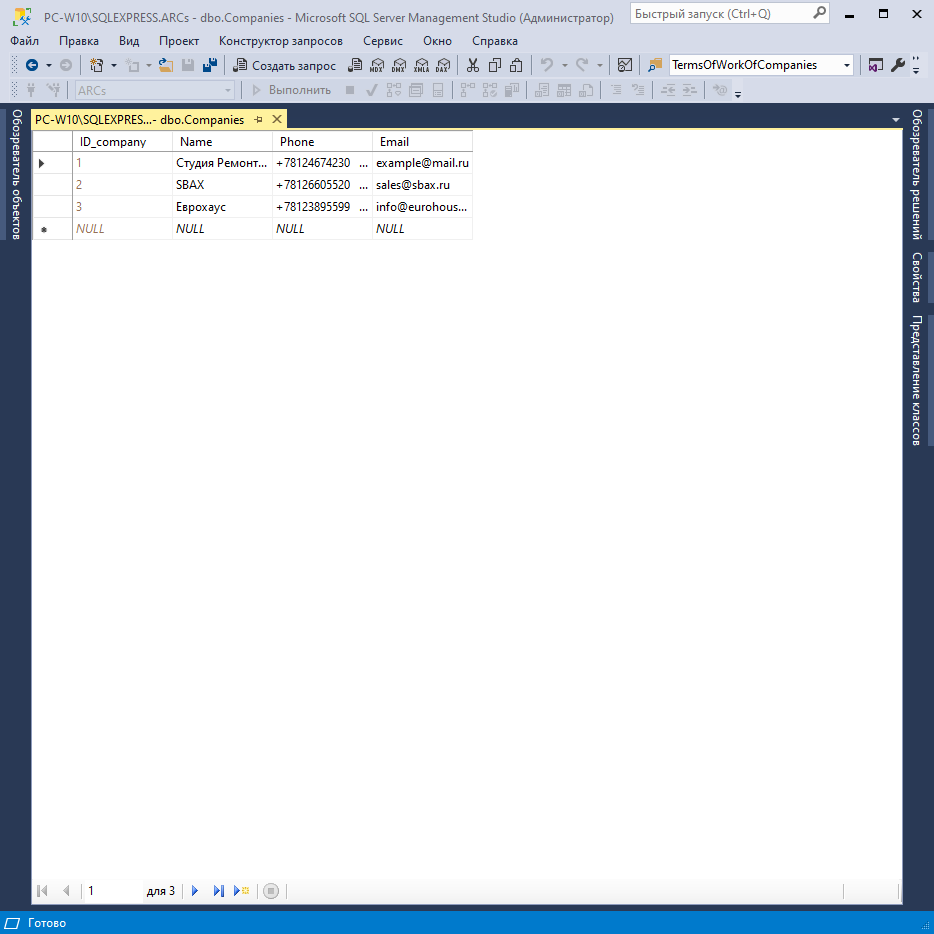


Рисунок 49 – Добавление почты у компании «Студия Ремонтов» успешно

Заключение

Список используемых источников

1. Максимов Н.В. ,Попов И.И.  Компьютерные сети : учеб. пособие / Н.В. Максимов, И.И. Попов. — 6-е изд., перераб. и доп. — М. : ФОРУМ : ИНФРА-М, 2018. — 464 с. — (Среднее профессиональное образование) —Режим доступа: <http://znanium.com/catalog/product/792685> — ЭБС «Знаниум», по паролю;
2. Кузин А.В., Кузин Д.А. Компьютерные сети : учеб. пособие / А.В. Кузин, Д.А. Кузин. — 4-е изд., перераб. и доп. — М. : ФОРУМ : ИНФРА-М, 2017. — 190 с. — (Среднее профессиональное образование) — Режим доступа: http://znanium.com/catalog/product/854772— ЭБС «Знаниум», по паролю;
3. Агальцов В.П. Базы данных. В 2-х кн. Книга 2. Распределенные и удаленные базы данных : учебник / В.П. Агальцов. — М. : ИД «ФОРУМ» : ИНФРА-М, 2017. — 271 с. : ил.— Режим доступа: http://znanium.com/ catalog/product/854772— ЭБС «Знаниум», по паролю;
4. Голицына О.Л. Основы проектирования баз данных : учеб. пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. — 2-е изд., перераб. и доп. — М. : ФОРУМ : ИНФРА-М, 2017. — 416 с. : ил. — (Cреднее профессиональное образование). .— Режим доступа: http://znanium.com/catalog.php?bookinfo=899656 — ЭБС «Знаниум», по паролю;
5. Фуфаев Э. В., Фуфаев Д.Э., Разработка и эксплуатация удаленных баз данных: учебник для студ. сред. проф. Образования,4-е изд. Стер. - М.:  Издательский центр «Академия», 2014;
6. Фуфаев Э. В., Фуфаев Д.Э., Базы данных: учебное пособие для студ. сред. проф. образования , 9-е изд. стер.- М.:  Издательский центр «Академия», 2014.

Приложения

Приложение 1

Скрипт БД.

Создание БД

use master

go

create database ARCs

--основной файл

on primary

(

name = ARCs\_dat,

filename = "C:\Program Files\Microsoft SQL Server\

MSSQL10.MSSQLSERVER\MSSQL\DATA\ARCs\_dat.mdf",

--filename = "E:\MS SQL Server 2017 Edition\Компоненты\MSSQL14.SQLEXPRESS\

--MSSQL\DATA\ARCs\_dat.mdf",

size = 4,--первоначальный размер файла

maxsize = 10,

filegrowth = 1

)

--журнал файл

log on

(

name = ARCs\_log,

filename = "C:\Program Files\Microsoft SQL Server\

MSSQL10.MSSQLSERVER\MSSQL\DATA\ARCs\_log.ldf",

--filename = "E:\MS SQL Server 2017 Edition\Компоненты\MSSQL14.SQLEXPRESS\

--MSSQL\DATA\ARCs\_log.ldf",

size = 2,

maxsize = 5,

filegrowth = 1

)

go

--drop database ARCs

--go

Создание таблиц

use ARCs

go

create table dbo.Customer-- Заказчик

(

ID\_customer int identity(1,1) not null,--ID заказчика

constraint PK\_Customer primary key clustered (ID\_customer),

Surname nvarchar(70) not null,--Фамилия

Name nvarchar(70) not null,

Patronymic nvarchar(70) not null,--Отчество

Phone char(20) not null,

Email varchar(250) null

)

go

create table dbo.Companies-- Компании

(

ID\_company int identity(1,1) not null,

constraint PK\_Companies primary key clustered (ID\_company),

Name nvarchar(250) not null,

Phone char(20) not null,

Email varchar(250) null

)

go

create table dbo.Employees-- Сотрудники

(

ID\_employee int identity(1,1) not null,

constraint PK\_Employees primary key clustered (ID\_employee),

Surname nvarchar(70) not null,

Name nvarchar(70) not null,

Patronymic nvarchar(70) not null,

Phone char(20) not null,

Email varchar(250) null

)

go

create table dbo.CompaniesEmployees-- КомпанииСотрудники

(

ID\_company int not null,

ID\_employee int not null,

constraint PK\_CompaniesEmployees primary key nonclustered (ID\_company, ID\_employee),

constraint FK\_CompaniesEmployees\_Company foreign key (ID\_company)

references dbo.Companies (ID\_company)

on delete cascade

on update cascade,

constraint FK\_CompaniesEmployees\_Employee foreign key (ID\_employee)

references dbo.Employees (ID\_employee)

on delete cascade

on update cascade

)

go

create table dbo.Services-- Услуги

(

ID\_service int not null,

constraint PK\_Services primary key nonclustered (ID\_service),

Name nvarchar(250) not null,

ID\_company int not null,

constraint FK\_Services\_Company foreign key (ID\_company)

references dbo.Companies (ID\_company)

on delete cascade

on update cascade,

Price money not null

)

go

create table dbo.Contract-- Договор

(

ID\_contract int not null,

constraint PK\_Contract primary key nonclustered (ID\_contract),

ID\_customer int not null,

constraint FK\_Contract\_Customer foreign key (ID\_customer)

references dbo.Customer (ID\_customer)

on delete cascade

on update cascade,

ID\_service int not null,

constraint FK\_Contract\_Service foreign key (ID\_service)

references dbo.Services (ID\_service)

on delete cascade

on update cascade,

Additional\_expenses money not null,--дополнительные расходы

Date\_beginning date not null,--дата начала

Date\_expiration date not null,--дата окончания

Cost money not null

)

go

--drop table if exists example

Создание процедур

use ARCs

go

-- Процедуры

-- Заполнение таблиц данными

--Заказчик

create proc inputDataCustomer( @nvcS nvarchar(70), @nvcN nvarchar(70),

@nvcP nvarchar(70), @c char(20), @vc varchar(250) )

as

begin

insert into Customer(Surname, Name, Patronymic, Phone, Email)

values(@nvcS, @nvcN, @nvcP, @c, @vc)

end

go

exec inputDataCustomer 'Кузьмин', 'Святослав', 'Андреевич', '+79117640001',

''

go

exec inputDataCustomer 'Шилов', 'Владислав', 'Владислав', '+79520550002',

'shilovVV0002@mail.ru'

go

exec inputDataCustomer 'Рожкова', 'Марфа', 'Иринеевна', '+79229830003',

'RozhkovaMarphaE0003'

go

exec inputDataCustomer 'Щербаков', 'Никифор', 'Витальевич', '+79625530004',

''

go

exec inputDataCustomer 'Харитонова', 'Наталия', 'Григорьевна', '+79999640005',

'KharitonovaNG0005@gmail.com'

go

--drop proc inputDataCustomer

--go

--Компании

create proc inputDataCompanies( @nvc nvarchar(250), @c char(20), @vc varchar(250) )

as

begin

insert into Companies(Name, Phone, Email)

values(@nvc, @c, @vc)

end

go

exec inputDataCompanies 'Студия Ремонтов', '+78124674230', ''

go

exec inputDataCompanies 'SBAX', '+78126605520', 'sales@sbax.ru'

go

exec inputDataCompanies 'Еврохаус', '+78123895599', 'info@eurohouse.spb.ru'

go

--drop proc inputDataCompanies

--go

--Сотрудники

create proc inputDataEmployees( @nvcS nvarchar(70), @nvcN nvarchar(70),

@nvcP nvarchar(70), @c char(20), @vc varchar(250) )

as

begin

insert into Employees(Surname, Name, Patronymic, Phone, Email)

values(@nvcS, @nvcN, @nvcP, @c, @vc)

end

go

exec inputDataEmployees 'Жуков', 'Дмитрий', 'Вениаминович', '+79994670001',

'zdv01@mail.ru'

go

exec inputDataEmployees 'Тарасов', 'Леонид', 'Алексеевич', '+79994670002',

'tla02@mail.ru'

go

exec inputDataEmployees 'Белоусов', 'Антон', 'Аркадьевич', '+79994670003',

'baa03@mail.ru'

go

exec inputDataEmployees 'Бобылёв', 'Панкратий', 'Мэлсович', '+79994670004',

'bpm04@mail.ru'

go

exec inputDataEmployees 'Власова', 'Эльвира', 'Львовна', '+79994670005',

''

go

exec inputDataEmployees 'Силин', 'Бронислав', 'Лаврентьевич', '+79996600001',

'sbl01@sbax.ru'

go

exec inputDataEmployees 'Беспалов', 'Борис', 'Филиппович', '+79996600002',

'bbf02@sbax.ru'

go

exec inputDataEmployees 'Хохлов', 'Тарас', 'Анатольевич', '+79996600003',

''

go

exec inputDataEmployees 'Лихачёв', 'Иосиф', 'Федорович', '+79996600004',

'lif04@sbax.ru'

go

exec inputDataEmployees 'Субботина', 'Веста', 'Иринеевна', '+79996600005',

'svi05@sbax.ru'

go

exec inputDataEmployees 'Павлов', 'Емельян', 'Федотович', '+79993890001',

'pef01mail@peremont.ru'

go

exec inputDataEmployees 'Копылов', 'Дмитрий', 'Артемович', '+79993890002',

'kda02mail@peremont.ru'

go

exec inputDataEmployees 'Николаев', 'Давид', 'Николаевич', '+79993890003',

'ndn03mail@peremont.ru'

go

exec inputDataEmployees 'Доронин', 'Федор', 'Аркадьевич', '+79993890004',

'dfa04mail@peremont.ru'

go

exec inputDataEmployees 'Крюкова', 'Светлана', 'Петровна', '+79993890005',

'ksp05mail@peremont.ru'

go

--drop proc inputDataEmployees

--go

--КомпанииСотрудники

create proc inputDataCompaniesEmployees( @iC int, @iE int )

as

begin

insert into CompaniesEmployees(ID\_company, ID\_employee)

values(@iC, @iE)

end

go

exec inputDataCompaniesEmployees 1, 1

go

exec inputDataCompaniesEmployees 1, 2

go

exec inputDataCompaniesEmployees 1, 3

go

exec inputDataCompaniesEmployees 1, 4

go

exec inputDataCompaniesEmployees 1, 5

go

exec inputDataCompaniesEmployees 2, 6

go

exec inputDataCompaniesEmployees 2, 7

go

exec inputDataCompaniesEmployees 2, 8

go

exec inputDataCompaniesEmployees 2, 9

go

exec inputDataCompaniesEmployees 2, 10

go

exec inputDataCompaniesEmployees 3, 11

go

exec inputDataCompaniesEmployees 3, 12

go

exec inputDataCompaniesEmployees 3, 13

go

exec inputDataCompaniesEmployees 3, 14

go

exec inputDataCompaniesEmployees 3, 15

go

--drop proc inputDataCompaniesEmployees

--go

--Услуги

create proc inputDataServices( @iS int, @nvc nvarchar(250), @iC int, @m money )

as

begin

insert into Services(ID\_service, Name, ID\_company, Price)

values(@iS, @nvc, @iC, @m)

end

go

exec inputDataServices 1, 'Черновой ремонт', 1, 2350 --за кв. метр

go

exec inputDataServices 2, 'Косметический ремонт', 1, 2790

go

exec inputDataServices 3, 'Капитальный ремонт', 1, 4740

go

exec inputDataServices 4, 'Черновой ремонт', 2, 2500 --за кв. метр

go

exec inputDataServices 5, 'Косметический ремонт', 2, 2600

go

exec inputDataServices 6, 'Капитальный ремонт', 2, 4800

go

exec inputDataServices 7, 'Черновой ремонт', 3, 2350 --за кв. метр

go

exec inputDataServices 8, 'Евроремонт', 3, 6300

go

exec inputDataServices 9, 'Капитальный ремонт', 3, 4750

go

--drop proc inputDataServices

--go

--Договор

create proc inputDataContract( @iCon int, @iCus int, @iS int, @mAdd money,

@dateB date, @dateE date, @mC money )

as

begin

insert into Contract(ID\_contract, ID\_customer, ID\_service, Additional\_expenses,

Date\_beginning, Date\_expiration, Cost)

values(@iCon, @iCus, @iS, @mAdd, @dateB, @dateE, @mC)

end

go

exec inputDataContract 1, 1, 5, 1500, '2020-9-21', '2020-11-20', 92500

go

exec inputDataContract 2, 2, 1, 500, '2020-10-05', '2020-11-04', 81500

go

exec inputDataContract 3, 3, 8, 5000, '2020-10-11', '2020-12-01', 225500

go

exec inputDataContract 4, 4, 9, 4500, '2021-01-21', '2021-03-22', 170750

go

exec inputDataContract 5, 5, 7, 1000, '2021-02-14', '2021-03-17', 93000

go

--drop proc inputDataContract

--go

-- Удаление данных из таблиц

--Заказчик (по диапозону индексов)

create proc deleteDataCustomer\_iSiF( @iS int, @iF int )

as

begin

if @iS <= @iF

while @iS <= @iF

begin

delete from Customer

where ID\_customer = @iS

set @iS += 1

end

else

while @iF <= @iS

begin

delete from Customer

where ID\_customer = @iF

set @iF += 1

end

end

go

exec deleteDataCustomer\_iSiF 1, 5

go

--drop proc deleteDataCustomer\_iSiF

--go

--Компании (по диапозону индексов)

create proc deleteDataCompanies\_iSiF( @iS int, @iF int )

as

begin

if @iS <= @iF

while @iS <= @iF

begin

delete from Companies

where ID\_company = @iS

set @iS += 1

end

else

while @iF <= @iS

begin

delete from Companies

where ID\_company = @iF

set @iF += 1

end

end

go

exec deleteDataCompanies\_iSiF 1, 3

go

--drop proc deleteDataCompanies\_iSiF

--go

--Сотрудники (по диапозону индексов)

create proc deleteDataEmployees\_iSiF( @iS int, @iF int )

as

begin

if @iS <= @iF

while @iS <= @iF

begin

delete from Employees

where ID\_employee = @iS

set @iS += 1

end

else

while @iF <= @iS

begin

delete from Employees

where ID\_employee = @iF

set @iF += 1

end

end

go

exec deleteDataEmployees\_iSiF 2, 16

go

--drop proc deleteDataEmployees\_iSiF

--go

--КомпанииСотрудники (по диапозону индексов)

create proc deleteDataCompaniesEmployees\_iSiF( @iCmpS int, @iCmpF int,

@iEmpS int, @iEmpF int )

as

begin

declare @x int

if @iCmpS > @iCmpF

begin

set @x = @iCmpS

set @iCmpS = @iCmpF

set @iCmpF = @x

end

if @iEmpS > @iEmpF

begin

set @x = @iEmpS

set @iEmpS = @iEmpF

set @iEmpF = @x

end

set @x = @iEmpS

while @iCmpS <= @iCmpF

begin

while @iEmpS <= @iEmpF

begin

delete from CompaniesEmployees

where ID\_company = @iCmpS and ID\_employee = @iEmpS

set @iEmpS += 1

end

set @iCmpS += 1

set @iEmpS = @x

end

end

go

exec deleteDataCompaniesEmployees\_iSiF 1, 3, 1, 15

go

--drop proc deleteDataCompaniesEmployees\_iSiF

--go

--Сервисы (по диапозону индексов)

create proc deleteDataServices\_iSiF( @iS int, @iF int )

as

begin

if @iS <= @iF

while @iS <= @iF

begin

delete from Services

where ID\_service = @iS

set @iS += 1

end

else

while @iF <= @iS

begin

delete from Services

where ID\_service = @iF

set @iF += 1

end

end

go

exec deleteDataServices\_iSiF 1, 9

go

--drop proc deleteDataServices\_iSiF

--go

--Договор (по диапозону индексов)

create proc deleteDataContract\_iSiF( @iS int, @iF int )

as

begin

if @iS <= @iF

while @iS <= @iF

begin

delete from Contract

where ID\_contract = @iS

set @iS += 1

end

else

while @iF <= @iS

begin

delete from Contract

where ID\_contract = @iF

set @iF += 1

end

end

go

exec deleteDataContract\_iSiF 1, 9

go

--drop proc deleteDataContract\_iSiF

--go

--Сотрудники (по содержимому)

create proc deleteDataEmployees\_Content( @nvcS nvarchar(70), @nvcN nvarchar(70),

@nvcP nvarchar(70), @c char(20), @vc varchar(250) )

as

begin

delete from Employees

where Surname = @nvcS and Name = @nvcN and Patronymic = @nvcP and Phone = @c

and Email = @vc

end

go

exec deleteDataEmployees\_Content 'Крюкова', 'Светлана', 'Петровна', '+79993890005',

'ksp05mail@peremont.ru'

go

--drop proc deleteDataEmployees\_Content

--go

-- удаление Всех записей из таблиц

--Заказчик

create proc deleteAllRecordsCustomer

as

begin

delete from Customer

end

go

exec deleteAllRecordsCustomer

go

--drop proc deleteAllRecordsCustomer

--go

--Компании

create proc deleteAllRecordsCompanies

as

begin

delete from Companies

end

go

exec deleteAllRecordsCompanies

go

--drop proc deleteAllRecordsCompanies

--go

--Сотрудники

create proc deleteAllRecordsEmployees

as

begin

delete from Employees

end

go

exec deleteAllRecordsEmployees

go

--drop proc deleteAllRecordsEmployees

--go

--КомпанииСотрудники

create proc deleteAllRecordsCompaniesEmployees

as

begin

delete from CompaniesEmployees

end

go

exec deleteAllRecordsCompaniesEmployees

go

--drop proc deleteAllRecordsCompaniesEmployees

--go

--Услуги

create proc deleteAllRecordsServices

as

begin

delete from Services

end

go

exec deleteAllRecordsServices

go

--drop proc deleteAllRecordsServices

--go

--Договор

create proc deleteAllRecordsContract

as

begin

delete from Contract

end

go

exec deleteAllRecordsContract

go

--drop proc deleteAllRecordsContract

--go

-- Обновление данных в таблицах

--Сотрудники

create proc updateDataEmployees\_ID( @i int, @nvcS nvarchar(70), @nvcN nvarchar(70),

@nvcP nvarchar(70), @c char(20), @vc varchar(250) )

as

begin

update Employees

set Surname = @nvcS, Name = @nvcN, Patronymic = @nvcP, Phone = @c, Email = @vc

where ID\_employee = @i

end

go

exec updateDataEmployees\_ID 15, 'Крюкова', 'Светлана', 'Петровна', '+79993890005',

'ksp05mail@peremont.ru, ksp052@gmail.com'

go

select \* from Employees

go

--drop proc updateDataEmployees\_ID

--go

--Заказчик (индекс)

create proc updateDataCustomer\_resetID( @i int)

as

begin

dbcc checkident(Customer, reseed, @i)

end

go

exec updateDataCustomer\_resetID 0

go

--drop proc updateDataCustomer\_resetID

--go

--Компанни (индекс)

create proc updateDataCompanies\_resetID( @i int )

as

begin

dbcc checkident(Companies, reseed, @i)

end

go

exec updateDataCompanies\_resetID 0

go

--drop proc updateDataCompanies\_resetID

--go

--Сотрудники (индекс)

create proc updateDataEmployees\_resetID( @i int)

as

begin

dbcc checkident(Employees, reseed, @i)

end

go

exec updateDataEmployees\_resetID 0

go

--drop proc updateDataEmployees\_resetID

--go

Создание представлений

use ARCs

go

-- Представления

-- Вывод содержимого таблиц (для проверки)

--Заказчик

select \* from Customer

go

--Компании

select \* from Companies

go

--Сотрудники

select \* from Employees

go

--КомпанииСотрудники

select \* from CompaniesEmployees

go

--Услуги

select \* from Services

go

--Договор

select \* from Contract

go

-- Представления на выборку данных из нескольких таблиц с использованием

-- условий отбора по нескольким полям

--Компании и их услуги

create view CompaniesTheirServices

as

select C.Name as [Company Name], S.Name as [Services Name]--псевдонимы

from Companies as C, Services as S-- псевдонимы

where S.ID\_company = C.ID\_company

go

select \* from CompaniesTheirServices

go

--drop view CompaniesTheirServices

--go

--Заказчики без почты, а также компании

create view CustomersAndCompaniesNotEmail

as

select ID\_customer, Surname, Customer.Name as 'Customer Name', Patronymic,

ID\_company, Companies.Name as 'Company Name'

from Customer, Companies

where Customer.Email = '' and Companies.Email = ''

go

select \* from CustomersAndCompaniesNotEmail

go

--drop view CustomersAndCompaniesNotEmail

--go

--Компании и их сотрудники

create view CompaniesTheirEmployees

as

select C.ID\_company, C.Name as 'Company Name', E.ID\_employee, Surname,

E.Name as 'Employee Name', Patronymic, E.Phone, E.Email

from Companies as C, Employees as E, CompaniesEmployees as CE

where CE.ID\_company = C.ID\_company and CE.ID\_employee = E.ID\_employee

go

select \* from CompaniesTheirEmployees

go

--drop view CompaniesTheirEmployees

--go

--Компании с их услугами и сроками выполнения этих услуг по заказу

create view TermsOfWorkOfCompanies

as

--select Cmp.ID\_company, Cmp.Name as 'Company Name', S.Name as 'Service Name',

--Date\_beginning, Date\_expiration

--from Companies as Cmp, Services as S, Contract as Cnt

--where Cnt.ID\_service = S.ID\_service and

--S.ID\_company = Cmp.ID\_company

--OR

select Cmp.ID\_company, Cmp.Name 'Company Name', S.Name 'Service Name',

Date\_beginning, Date\_expiration

from Companies Cmp--, Services S, Contract Cnt

join Services S on Cmp.ID\_company = S.ID\_company

join Contract Cnt on Cnt.ID\_service = S.ID\_service

go

select \* from TermsOfWorkOfCompanies

go

--drop view TermsOfWorkOfCompanies

--go

-- Представления с использованием расчётных полей

--Каков доход, если будут выполнены все заказы

create view IncomeAllOrdersFulfilled

as

select sum(Cnt.Cost) 'Income'

from Contract Cnt, Customer Csr,Services S

where Cnt.ID\_customer = Csr.ID\_customer and Cnt.ID\_service = S.ID\_service

go

select \* from IncomeAllOrdersFulfilled

go

--drop view IncomeAllOrdersFulfilled

--go

--Сроки ремонтов в днях

create view TermsRepairsDays

as

select Cr.Name as 'Name Customer', S.Name as 'Name Service',

datediff(day, Ct.Date\_beginning, Ct.Date\_expiration) as 'Terms'

from Customer as Cr, Services as S, Contract as Ct

where Ct.ID\_service = S.ID\_service and Ct.ID\_customer = Cr.ID\_customer

go

select \* from TermsRepairsDays

go

--drop view TermsRepairsDays

--go

-- Представление с использованием групповых операций

--Количество каждой услуги

create view CountEachServices

as

select Name, count(Name) as 'Count Services'

from Services

group by Name

go

select \* from CountEachServices

go

--drop view CountEveryServices

--go

--Сколько заказчиков у каждой компании

create view CountCustomerEachCompany

as

--select Cmp.Name 'Company Name', Count(Ctr.Name) 'Count Customer'

--from Companies Cmp

--join Services S on S.ID\_company = Cmp.ID\_company

--join Contract Cnt on Cnt.ID\_service = S.ID\_service

--join Customer Ctr on Cnt.ID\_customer = Ctr.ID\_customer

--group by Cmp.Name

select Cmp.Name 'Company Name', Count(Ctr.Name) 'Count Customer'

from Contract Cnt, Companies Cmp, Customer Ctr, Services S

where Cnt.ID\_customer = Ctr.ID\_customer and S.ID\_company = Cmp.ID\_company

and Cnt.ID\_service = S.ID\_service

group by Cmp.Name

go

select \* from CountCustomerEachCompany

go

--drop view CountCustomerEachCompany

--go

Создание ролей

use ARCs

go

-- создание Уч. записей

create login Manager\_ARCs

with password = 'logM474',

default\_database = ARCs,

default\_language = русский,

check\_expiration = off,--проверять ли срок использования пароля

check\_policy=off--политика паролей

go

create login Employee\_ARCs

with password = 'logE474',

default\_database = ARCs,

default\_language = русский,

check\_expiration = off,

check\_policy = off

go

create login Client\_ARCs

with password = 'logC474',

default\_database = ARCs,

default\_language = русский,

check\_expiration = off,

check\_policy = off

go

--alter login Client\_ARCs

--with password = 'logC474',

--check\_policy = off

--go

--drop login Employee\_ARCs

--go

-- создание Пользователей

create user uManager\_ARCs

for login Manager\_ARCs

go

create user uEmployee\_ARCs

for login Employee\_ARCs

go

create user uClient\_ARCs

for login Client\_ARCs

go

--drop user uEmployee\_ARCs

--go

-- создание Ролей

create role rManager\_ARCs--Менеджеры

create role rEmployee\_ARCs--Сотрудники

create role rClient\_ARCs--Клиенты

--drop role rEmployee\_ARCs

--добавление пользователей в роль

exec sp\_addrolemember @rolename = rManager\_ARCs,

@membername = uManager\_ARCs

exec sp\_addrolemember @rolename = rEmployee\_ARCs,

@membername = uEmployee\_ARCs

exec sp\_addrolemember @rolename = rClient\_ARCs,

@membername = uClient\_ARCs

--exec sp\_addrolemember @rolename = db\_owner,

--@membername = uManager\_ARCs

--exec sp\_droprolemember @rolename = db\_owner,

--@membername = uManager\_ARCs

--exec sp\_droprolemember @rolename = rEmployee\_ARCs,

--@membername = uEmployee\_ARCs

-- Информация

-- о Роли

exec sp\_helprole @rolename = rManager\_ARCs

go

exec sp\_helprole @rolename = rEmployee\_ARCs

go

exec sp\_helprole @rolename = rClient\_ARCs

go

-- о Пользователях

exec sp\_helpuser uManager\_ARCs

go

exec sp\_helpuser uEmployee\_ARCs

go

exec sp\_helpuser uClient\_ARCs

go

-- Разрешения

--Заказчик (клиент)

--Чтение таблицы услуги и компании

grant select on Services to rClient\_ARCs

go

grant select on Companies to rClient\_ARCs

go

--представление

grant select on CompaniesTheirServices to rClient\_ARCs

go

--revoke select on CompaniesTheirServices to rClient\_ARCs

--go

--Сотрудник

--Чтение, Добавление, Обновление, Удаление, Исполнение

--таблицы

grant select on Companies to rEmployee\_ARCs

go

grant select on Services to rEmployee\_ARCs

go

grant select, insert, update, delete on Customer to rEmployee\_ARCs

go

grant select, insert, update, delete on Contract to rEmployee\_ARCs

go

--представления

grant select on CompaniesTheirServices to rEmployee\_ARCs

go

grant select on TermsOfWorkOfCompanies to rEmployee\_ARCs

go

grant select on TermsRepairsDays to rEmployee\_ARCs

go

grant select on CountEachServices to rEmployee\_ARCs

go

--процедуры

grant exec on inputDataCustomer to rEmployee\_ARCs

go

grant exec on inputDataContract to rEmployee\_ARCs

go

--Резервное копирование БД

grant backup database, backup log to rEmployee\_ARCs

go

--Менеджер

grant all on Companies to rManager\_ARCs

go

grant all on Employees to rManager\_ARCs

go

grant all on Customer to rManager\_ARCs

go

grant all on CompaniesEmployees to rManager\_ARCs

go

grant all on Services to rManager\_ARCs

go

grant all on Contract to rManager\_ARCs

go

--резервное копироваине БД

grant backup database, backup log to rManager\_ARCs

with grant option--возможность выдавать другим эту возможность

go

--просмотр прав у Пользователя или Роли

sp\_helprotect null, rManager\_ARCs

go

sp\_helprotect null, rEmployee\_ARCs

go

sp\_helprotect null, rClient\_ARCs

go

--sp\_helprotect @name = Companies/\*null\*/, @username = null--uManager\_ARCs

--go

sp\_helprotect Services, null

go

--всё обо всех

sp\_helprotect null, null

go

Создание резервных копий

use ARCs

go

--создание устройств резерного копирования

--для БД

exec sp\_addumpdevice 'disk', 'deviceARCs\_DB', 'E:\MS SQL Server 2017 Edition\Компоненты\

MSSQL14.SQLEXPRESS\MSSQL\Backup\deviceARCs\_DB.bak'

--для Журналов транзакций

exec sp\_addumpdevice 'disk', 'deviceARCs\_Log', 'E:\MS SQL Server 2017 Edition\Компоненты\

MSSQL14.SQLEXPRESS\MSSQL\Backup\deviceARCs\_Log.bak'

go

--exec sp\_dropdevice 'deviceARCs\_DB'

--go

--exec sp\_dropdevice 'deviceARCs\_Log'

--go

--Резерное копирование

--БД

backup database ARCs to deviceARCs\_DB

go

--Журналов транзакций

backup log ARCs to deviceARCs\_Log

go

--Разностная

backup database ARCs to deviceARCs\_DB with differential

go

фываолд