

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

ОТЧЕТ
по индивидуальному домашнему заданию
по дисциплине «Базы данных»
Тема: Проектирование базы данных салона красоты»

Студент гр. 0308

Сабурова Е.А.

Преподаватель

Новакова Н. Е.

ЗАДАНИЕ НА ИНДИВИДУАЛЬНУЮ РАБОТУ

Студент: Сабурова Е.А.

Группа: 0308

Тема работы: Разработка базы данных

Исходные данные: Поставленное задание (стр. 6)

СУБД: Microsoft SQL Server 2019 Express Edition

Содержание пояснительной записки: «Аннотация», «Содержание», «Введение», «Первый раздел», «Заключение», «Список использованных источников».

Предполагаемый объем пояснительной записки:

Не менее 30 страниц.

Дата выдачи задания: 11.09.2022

Дата сдачи задания: 2.12.2022

Дата защиты задания: 21.12.2022

Студент гр. 0308

Сабурова Е.А.

Преподаватель

Новакова Н. Е.

АННОТАЦИЯ

В процессе выполнения индивидуального домашнего задания была спроектирована база данных для салона красоты. При проектировании и создании были использованы знания, полученные в процессе изучения курса «Базы данных». Была описана предметная область, построены концептуальная, логическая и физическая модели данных. Была создана сама база данных и таблицы, которые затем были заполнены данными. Созданы объекты промежуточного слоя: представления, UDF-ы, хранимые процедуры, была разработана и реализована стратегия резервного копирования базы данных.

SUMMARY

In the process of completing individual homework assignments, a database was designed for the beauty saloon. During the design and creation, the knowledge gained in the course of studying the course "Databases" was used. The subject area was described, conceptual, logical and physical data models were built. The database itself and the tables were created, which were then populated with data. Middleware objects were created: views, UDFs, stored procedures, a database backup strategy was developed and implemented.

Содержание	
ВВЕДЕНИЕ	5
Цель работы	6
Задачи	6
Формулировка задания	6
Формулировка задачи	6
1. Краткое описание предметной области	7
2. Проектирование БД	7
3. Создание Базы Данных	15
4. Создание таблиц	16
5. Заполнение таблиц данными	19
6. Разработка объектов промежуточного слоя	21
7. Разработка стратегии резервного копирования	34
ЗАКЛЮЧЕНИЕ	36
Список использованных источников	37

ВВЕДЕНИЕ

В данной работе была спроектирована и создана база данных для администраторов салона красоты. База данных включает в себя сведения о работниках, клиентах, а также информацию о записях в салон.

Реализация каждой задачи производилась с помощью SQL-запросов. Процесс разработки базы данных включал в себя создание и заполнение таблиц данными, написание хранимых процедур, пользовательских функций и представлений для удобной работы с базой данных.

Цель работы

Целью работы является закрепление теоретических знаний и получение практических навыков в проектировании базы данных.

Задачи

- Краткое описание предметной области.
- Проектирование БД (структура данных)
- Создание БД
- Создание таблиц и ограничений целостности
- Заполнение таблиц данным
- Разработка объектов промежуточного слоя (представлений, хранимых процедур, UDF-ов)
- Разработка стратегии резервного копирования

Формулировка задания

В данной работе требуется выполнить следующие задания: разработать структуру базы данных, изобразить ее с помощью ER-диаграммы, спроектировать базу данных относительно заданной диаграммы, создать базу данных и необходимые таблицы, задать ограничения для таблиц и ключи, заполнить базу данных, создать в базе данных объекты промежуточного слоя – представления, хранимые процедуры, UDF, выбрать стратегию резервного копирования.

Формулировка задачи

Спроектировать базу данных (БД) для администраторов салона красоты, используя метод ER-диаграмм.

В БД должны храниться сведения о работниках: ФИ, номер телефона, специальность; сведения о клиентах: ФИ, номер телефона; информация о записях на процедуры: цена, дата и время, название процедуры, а также клиент и работник, которые связаны процедурой. Возможно появление новых клиентов, работников, специализаций и записей в салон.

1. Краткое описание предметной области

Предметной областью в данной работе является салон красоты, работники которой оказывают различные услуги пациентам непосредственно в самом салоне красоты.

Работа салона красоты построена по принципу разделения услуг между работниками в зависимости от их специализации. К каждой из этих специализаций прикрепляется работник, оказывающий услуги красоты клиентам.

Также при разработке базы данных предусмотрено, что клиент может воспользоваться не одной услугой, один работник может быть несколько выполнять несколько процедур и только по одной специализации, каждая специализация может иметь несколько работников.

2. Проектирование БД

Концептуальная модель данных.

Исходя из предметной области выделим следующие сущности:

1. Сущность Client:

Содержит информацию о клиентах, записанных в этот салон. Сущность содержит следующие атрибуты: персональный номер клиента, ФИ клиента и его номер телефона.

Потенциальным ключом может являться персональный номер клиента в базе данных.

2. Сущность Employee:

Эта сущность хранит в себе информацию о работниках этого салона красоты. Сущность обладает следующими атрибутами: персональный номер работника, ФИ работника, его номер телефона и номер его специализации.

Потенциальным ключом может являться персональный номер работника в базе данных.

3. Сущность Specials:

Данная сущность содержит информацию о всех специализациях,

имеющихся в данный момент в салоне. Имеются следующие атрибуты: Шифр специализации и название.

Потенциальным ключом может являться шифр специализации.

4. Сущность Appointment:

Данная сущность содержит информацию о всех записях в салон. Сущность обладает следующими атрибутами: персональный номер записи, цена за услугу, дата и время, название услуги, персональный номер клиента, персональный номер работника.

Потенциальным ключом может являться персональный номер записи.

На основе всех данных построим ER-диаграмму сущностей и связей между ними:

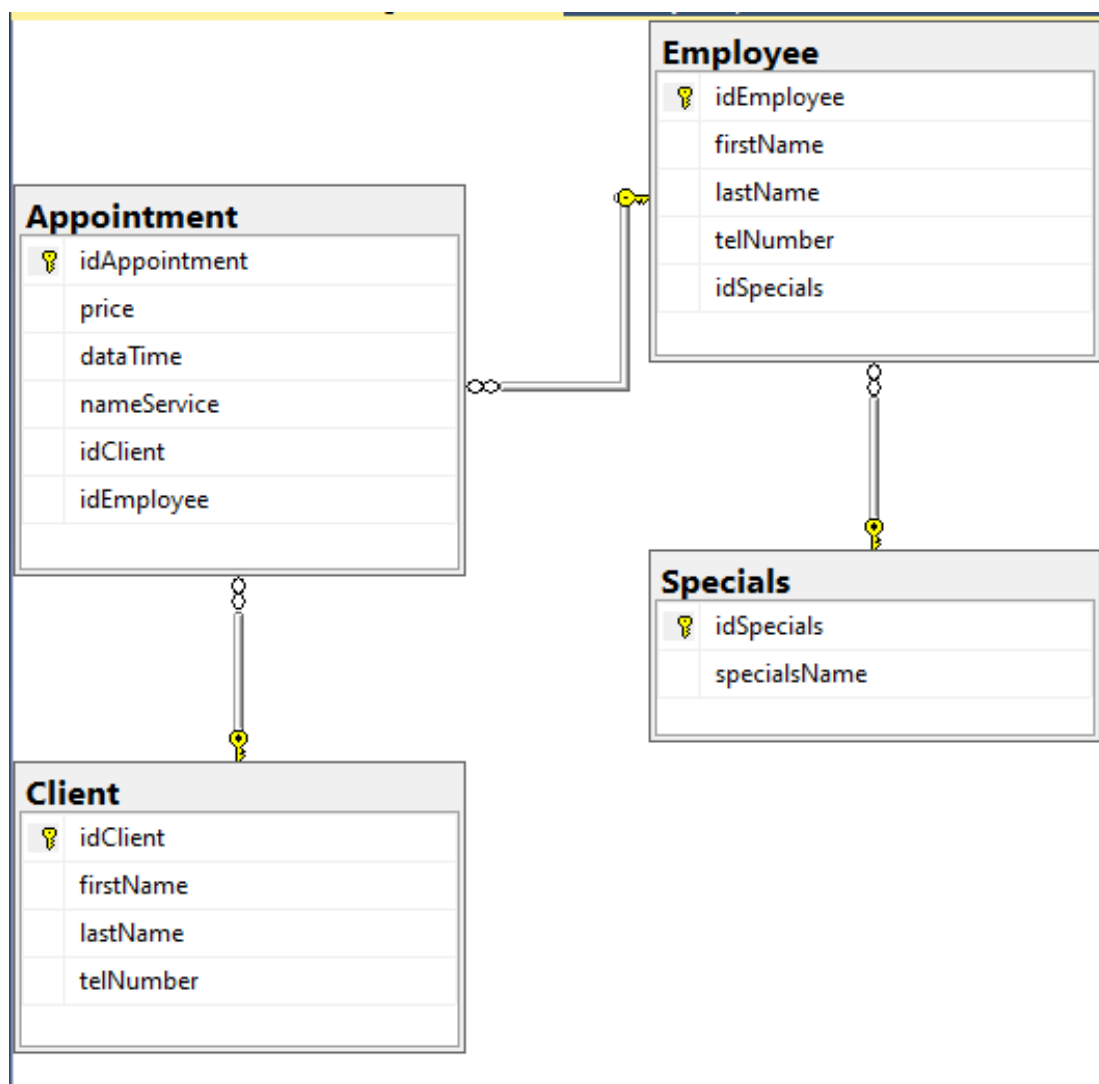


Рисунок 1. ER-диаграмма базы данных в SSMS

Логическая модель данных.

Исходя из определенных в концептуальной модели сущностей и связей между ними, создадим реляционную модель будущей базы данных.

Применим правила создания реляционной модели для ER-схемы и получим следующую схему:

1. Данные о клиенте (персональный номер клиента, фамилия, имя, номер телефона)
2. Данные о работнике (персональный номер работника, фамилия, имя, номер телефона, шифр специализации)
3. Специализация (шифр специализации, название специализации)
4. Записи в салон (персональный номер записи, цена, дата и время, название услуги, персональный номер клиента, персональный номер работника)

Таким образом, полученная реляционная модель нормализована.

Физическая модель данных.

Опишем данные в терминах T-SQL:

Описание структуры таблицы БД		Наименование таблицы БД: Специализация		Имя таблицы: Specials	
Дата разработки: 15.12.2022					
Порядковый номер таблицы: 1					

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничения целостности
1	Шифр специализации	idSpecials	tinyint	PK	
2	Название специализации	specialsName	var- char(50)		NOT NULL

Описание структуры таблицы БД	Наименование таблицы БД: Работник	Имя таблицы: Employee
Дата разработки: 15.12.2022		
Порядковый номер таблицы: 2		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничения целостности
1	Персональный номер работника	idEmployee	int	PK	
2	Имя	firstName	varchar(50)		NOT NULL
3	Фамилия	lastName	varchar(50)		NOT NULL
4	Номер телефона	telNumber	varchar(11)		NOT NULL
5	Шифр специализации	idSpecials	int	FK	NOT NULL

Описание структуры таблицы БД	Наименование таблицы БД: Клиент	Имя таблицы: Client
Дата разработки: 15.12.2022		
Порядковый номер таблицы: 3		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничения целостности
1	Персональный номер клиента	idClient	int	PK	
2	Имя	firstName	varchar(50)		NOT NULL
3	Фамилия	lastName	varchar(50)		NOT NULL
4	Номер телефона	telNumber	varchar(11)		NOT NULL

Описание структуры таблицы БД	Наименование таблицы БД: Записи	Имя таблицы: Appointment
Дата разработки: 15.12.2022		
Порядковый номер таблицы: 4		

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничения целостности
1	Персональный номер записи	idAppointment	int	PK	
2	Цена	price	int		NOT NULL
3	Дата и время	dateTime	datetime		NOT NULL
4	Название услуги	nameService	varchar(50)		NOT NULL
5	Персональный номер клиента	idClient	int	FK	NOT NULL
6	Персональный номер работника	idEmployee	int	FK	NOT NULL

3. Создание Базы Данных

Создадим БД BeautySaloon, для этого создадим новый запрос, в котором укажем имя файла и расположение файла.

Запрос:

```
CREATE DATABASE BeautySaloon
CONTAINMENT = NONE
ON PRIMARY ( --Первичный файл
    NAME = N'Beauty Saloon', --Логическое имя файла БД
    FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\BeautySaloon_log.mdf',
    SIZE = 8192KB,
    FILEGROWTH = 65536KB
)
LOG ON ( --Явно указываем файлы журналов
    NAME = N'Beauty_Saloon_log', --Логическое имя файла журнала
    FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\BeautySaloon_log.ldf',
    SIZE = 8192KB,
    FILEGROWTH = 65536KB
)
```

Результаты выполнения запроса:

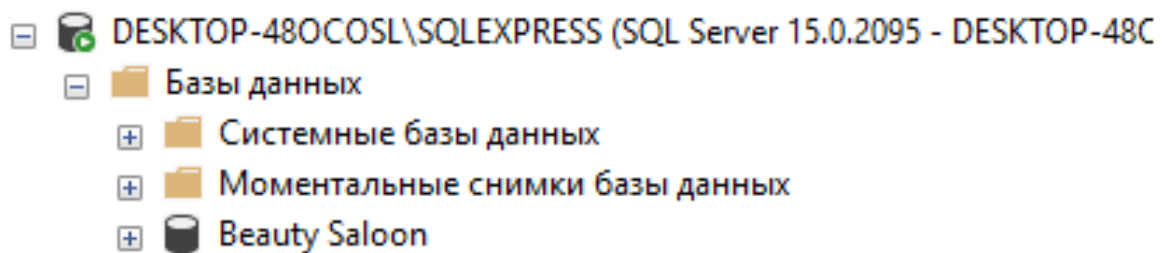


Рисунок 2. Созданная база данных BeautySaloon

4. Создание таблиц

Создадим все таблицы и ограничения целостности, которые были определены в физической модели данных.

1. Создание таблицы «Специализация»:

Запрос:

```
USE BeautySaloon
CREATE TABLE Specials(
idSpecials int identity primary key,
specialsName varchar(50) NOT NULL
);
```

Результаты выполнения запроса:

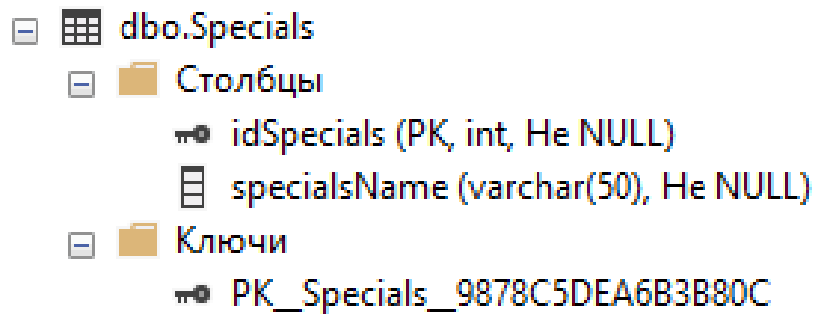


Рисунок 3. Созданная таблица Specials

2. Создание таблицы «Работник»:

Запрос:

```
USE BeautySaloon
CREATE TABLE Employee(
    idEmployee int identity primary key,
    firstName varchar(50) NOT NULL,
    lastName varchar(50) NOT NULL,
    telNumber varchar(11) NOT NULL,
    idSpecials int NOT NULL REFERENCES Specials(idSpecials)
);
```


Результаты выполнения запроса:

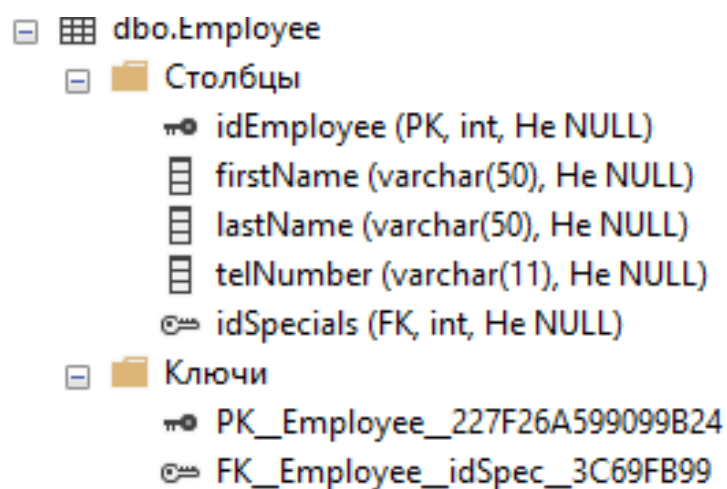


Рисунок 4. Созданная таблица Employee

3. Создание таблицы «Клиент»:

Запрос:

```
USE BeautySaloon
CREATE TABLE Client(
    idClient int identity primary key,
    firstName varchar(50) NOT NULL,
    lastName varchar(50) NOT NULL,
    telNumber varchar(11) NOT NULL,
);
```

Результаты выполнения запроса:

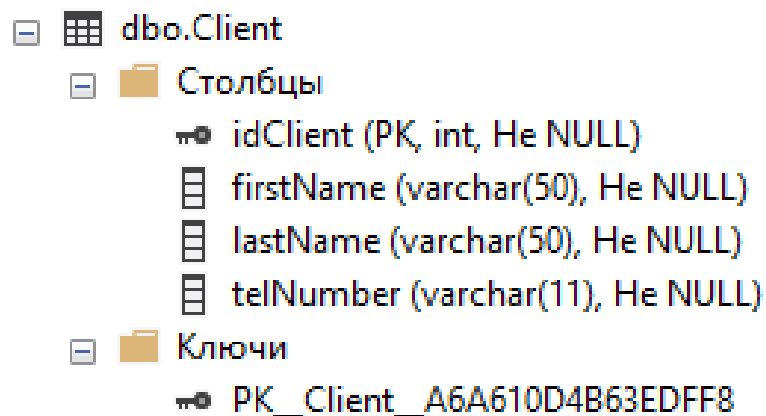


Рисунок 5. Созданная таблица Client

4. Создание таблицы «Записи»:

Запрос:

```
USE BeautySaloon
```

```
CREATE TABLE Appointment(  
    idAppointment int identity primary key,  
    price int NOT NULL,  
    dateTime datetime,  
    nameService varchar(50),  
    idClient int NOT NULL REFERENCES Client (idClient),  
    idEmployee int NOT NULL REFERENCES Employee (idEmployee),  
);
```

Результаты выполнения запроса:

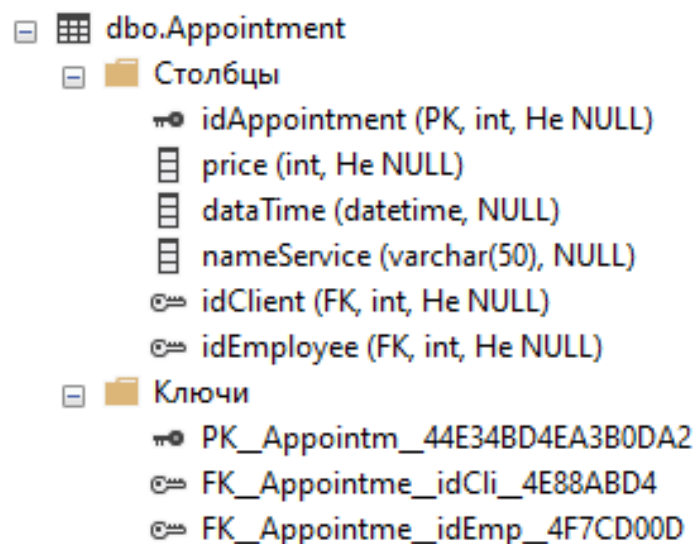


Рисунок 6. Созданная таблица Appointment

5. Заполнение таблиц данными

После создания таблиц произведем заполнение их данными.

1. Заполнение данными таблицы «Специализация».

Запрос:

```
USE BeautySaloon
INSERT INTO Specials
VALUES
    ( 'Мастер маникюра' ),
    ( 'Парикмахер' );
```

Результаты выполнения запроса:

	idSpecials	specialsName
1	1	Мастер маникюра
2	2	Парикмахер

Рисунок 7. Данные из таблицы Specials

2. Заполнение данными таблицы «Работник».

Запрос:

```
USE BeautySaloon
INSERT INTO Employee
VALUES
    ( 'Екатерина', 'Петрова', '+7962417820', '1' ),
    ( 'Валерия', 'Захарова', '+7034528642', '1' ),
    ( 'Мария', 'Лопатина', '+7954652585', '2' ),
    ( 'Регина', 'Соколова', '+7954200214', '2' ),
    ( 'Мария', 'Некрасова', '+7936304078', '2' );
```

Результаты выполнения запроса:

	idEmployee	firstName	lastName	telNumber	idSpecials
1	1	Екатерина	Петрова	+7962417820	1
2	2	Валерия	Захарова	+7034528642	1
3	3	Мария	Лопатина	+7954652585	2
4	4	Регина	Соколова	+7954200214	2
5	5	Мария	Некрасова	+7936304078	2

Рисунок 8. Данные из таблицы Employee

3. Заполнение данными таблицы «Клиент».

Запрос:

```
USE BeautySaloon
```

```
INSERT INTO Client
```

```
VALUES
```

```
( 'Екатерина', 'Елисеева', '+7965627820' ),  
( 'Карина', 'Соболева', '+7034857842' ),  
( 'Мария', 'Золотова', '+7954965285' ),  
( 'Регина', 'Курочкина', '+7953021214' ),  
( 'Марина', 'Селезнёва', '+7936698548' );
```

Результаты выполнения запроса:

	idClient	firstName	lastName	telNumber
1	1	Екатерина	Елисеева	+7965627820
2	2	Карина	Соболева	+7034857842
3	3	Мария	Золотова	+7954965285
4	4	Регина	Курочкина	+7953021214
5	5	Марина	Селезнёва	+7936698548

Рисунок 9. Данные из таблицы Client

4. Заполнение данными таблицы «Записи».

Запрос:

```
USE BeautySaloon
```

```
INSERT INTO Appointment
```

```
VALUES
```

```
( '2000', '2022-12-15T15:00:00', 'Маникюр с покрытием', '1', '1' ),  
( '3000', '2022-12-15T12:00:00', 'Маникюр с укреплением гелем', '2', '2' ),  
( '4000', '2022-12-21T15:00:00', 'Окрашивание', '1', '3' ),  
( '2000', '2022-12-21T17:00:00', 'Стрижка каре женская', '4', '4' ),  
( '2000', '2022-12-15T10:00:00', 'Маникюр с покрытием', '3', '1' ),  
( '2500', '2022-12-19T19:00:00', 'Стрижка каре женская', '5', '5' ),  
( '6000', '2022-12-24T13:00:00', 'Окрашивание', '2', '3' );
```

Результаты выполнения запроса:

	idAppointment	price	dateTime	nameService	idClient	idEmployee
1	1	2000	2022-12-15 15:00:00.000	Маникюр с покрытием	1	1
2	2	3000	2022-12-15 12:00:00.000	Маникюр с укреплением гелем	2	2
3	3	4000	2022-12-21 15:00:00.000	Окрашивание	1	3
4	4	2000	2022-12-21 17:00:00.000	Стрижка каре женская	4	4
5	5	2000	2022-12-15 10:00:00.000	Маникюр с покрытием	3	1
6	6	2500	2022-12-19 19:00:00.000	Стрижка каре женская	5	5
7	7	6000	2022-12-24 13:00:00.000	Окрашивание	2	3

Рисунок 10. Данные из таблицы Appointment

6. Разработка объектов промежуточного слоя

1. Процедура AddClient.

Данная процедура добавляет в таблицу Client нового клиента, записавшегося в салон красоты.

Запрос 1:

```
CREATE PROCEDURE AddClient
@FirstName nvarchar(50),
@LastName nvarchar(50),
@TelNumber nvarchar(11)
AS BEGIN
INSERT INTO Client (
    firstName,
    lastName,
    telNumber
)
VALUES (
    @FirstName,
    @LastName,
    @TelNumber)
END;
```

Результаты выполнения запроса:

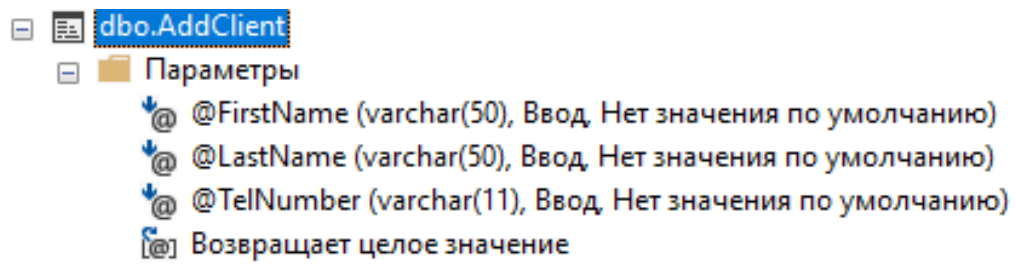


Рисунок 11. Процедура AddClient в обозревателе объектов

Проверим работу данной процедуры.

Запрос 2:

```
EXEC AddClient 'Анастасия', 'Абрамова', '+7911451474';
```

Результаты выполнения запроса:

	idClient	firstName	lastName	telNumber
1	1	Екатерина	Елисеева	+7965627820
2	2	Карина	Соболева	+7034857842
3	3	Мария	Золотова	+7954965285
4	4	Регина	Курочкина	+7953021214
5	5	Марина	Селезнёва	+7936698548
6	6	Анастасия	Абрамова	+7911451474

Рисунок 12. Результат выполнения процедуры AddClient

2. Процедура AddSpecial.

Данная процедура добавляет в таблицу Specials информацию о специализациях работников в салоне красоты.

Запрос 1:

```
CREATE PROCEDURE AddSpecial
@SpecialsName nvarchar(50)
AS BEGIN
INSERT INTO Specials (
    specialsName
)
VALUES (
    @SpecialsName
)
END;
```

Результаты выполнения запроса:

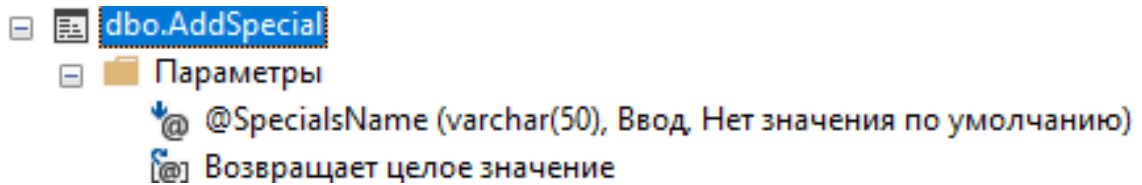


Рисунок 13. Процедура AddSpecial в обозревателе объектов

Проверим работу данной процедуры.

Запрос 2:

```
EXEC AddSpecial 'Бровист';
```

Результаты выполнения запроса:

	idSpecials	specialsName
1	1	Мастер маникюра
2	2	Парикмахер
3	3	Бровист

Рисунок 14. Результат выполнения процедуры AddSpecial

3. Процедура AddEmployee

Данная процедура добавляет в таблицу Employee нового работника, устроившегося в салон красоты. При попытке добавить работника с несуществующим шифром специализации, произойдет исключение.

Запрос 1:

```
CREATE PROCEDURE AddEmployee
@FirstName nvarchar(50),
@LastName nvarchar(50),
@TelNumber nvarchar(11),
@IdSpecials int
AS BEGIN
IF (NOT EXISTS (SELECT * FROM Specials
where idSpecials = @IdSpecials))
THROW 51000, 'No specialization found with such ID!', 1;
INSERT INTO Employee (
    firstName,
    lastName,
```

```

telNumber,
idSpecials
)
VALUES (
    @FirstName,
    @LastName,
    @TelNumber,
    @IdSpecials)
END;

```

Результаты выполнения запроса:

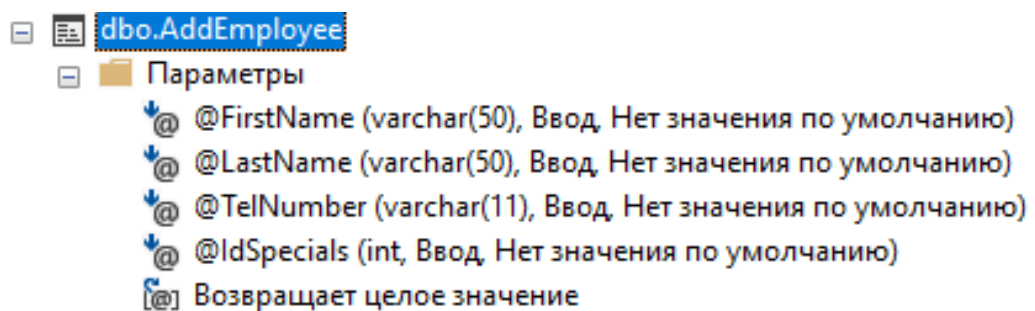


Рисунок 15. Процедура AddEmployee в обозревателе объектов

Проверим работу данной процедуры.

Запрос 2:

```
EXEC AddEmployee 'Анастасия', 'Власова', '+7965638578', 3;
```

Результаты выполнения запроса

	idEmployee	firstName	lastName	telNumber	idSpecials
1	1	Екатерина	Петрова	+7962417820	1
2	2	Валерия	Захарова	+7034528642	1
3	3	Мария	Лопатина	+7954652585	2
4	4	Регина	Соколова	+7954200214	2
5	5	Мария	Некрасова	+7936304078	2
6	6	Анастасия	Власова	+7965638578	3

Рисунок 16. Результат выполнения процедуры AddEmployee

Запрос 3:

Попытка записать работнику несуществующую специализацию:

```
EXEC AddEmployee 'Анастасия', 'Власова', '+7965638578', 7;
```


Результаты выполнения запроса

```
сообщение: 51000, уровень: 16, состояние: 1, процедура: AddEmployee, строка: 10 [строка начала пакета: 0]  
No specialization found with such ID!
```

Рисунок 17. Результат выполнения процедуры AddEmployee

4. Процедура AddAppointment.

Данная процедура добавляет в таблицу Appointment информацию о записи клиента на услугу. При попытке добавить запись к несуществующим работнику или клиенту, произойдет исключение.

Запрос 1:

```
CREATE PROCEDURE AddAppointment  
@Price int,  
@DateTime datetime,  
@NameService varchar(50),  
@IdClient int,  
@IdEmployee int  
AS BEGIN  
IF (NOT EXISTS (SELECT * FROM Employee  
where idEmployee = @IdEmployee))  
THROW 51000, 'No employee found with such ID!', 1;  
IF (NOT EXISTS (SELECT * FROM Client  
where idClient = @IdClient))  
THROW 51000, 'No client found with such ID!', 1;  
INSERT INTO Appointment (  
    price,  
    dateTime,  
    nameService,  
    idClient,  
    idEmployee  
)  
VALUES (  
    @Price,  
    @DateTime,  
    @NameService,  
    @IdClient,
```

@IdEmployee)

END;

Результаты выполнения запроса:

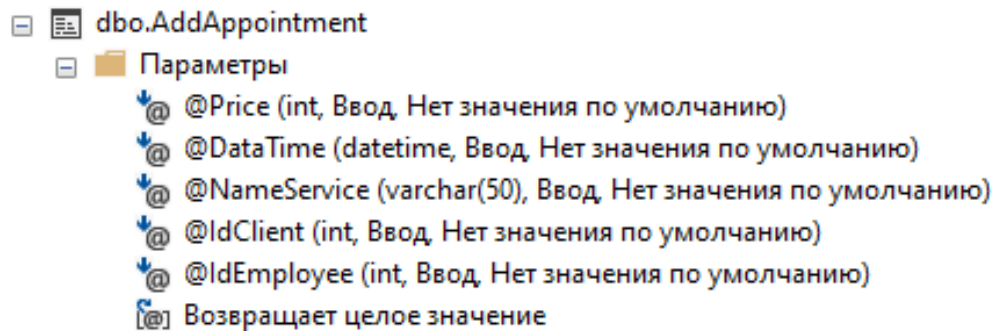


Рисунок 18. Процедура AddAppointment в обозревателе объектов

Проверим работу данной процедуры.

Запрос 2:

EXEC AddAppointment 700, '2022-12-27T14:30:00', 'Ламинирование бровей', 6,
6;

Результаты выполнения запроса:

	idAppointment	price	dateTime	nameService	idClient	idEmployee
1	1	2000	2022-12-15 15:00:00.000	Маникюр с покрытием	1	1
2	2	3000	2022-12-15 12:00:00.000	Маникюр с укреплением гелем	2	2
3	3	4000	2022-12-21 15:00:00.000	Окрашивание	1	3
4	4	2000	2022-12-21 17:00:00.000	Стрижка каре женская	4	4
5	5	2000	2022-12-15 10:00:00.000	Маникюр с покрытием	3	1
6	6	2500	2022-12-19 19:00:00.000	Стрижка каре женская	5	5
7	7	6000	2022-12-24 13:00:00.000	Окрашивание	2	3
8	8	700	2022-12-27 14:30:00.000	Ламинирование бровей	6	6

Рисунок 19. Результат выполнения процедуры AddAppointment

Запрос 3:

Попытка записать работнику несуществующую специализацию:

EXEC AddAppointment 700, '2023-01-14T14:30:00', 'Маникюр с покрытием', 5,
9;

Результаты выполнения запроса

сообщение: 51000, уровень: 16, состояние: 1, процедура: AddAppointment, строка: 12 [строка начала пакет
No employee found with such ID!]

Рисунок 20. Результат выполнения процедуры AddAppointment

5. Функция ServicesByEmployee.

Функция, возвращающая таблицу записей к выбранному работнику салона.

Запрос 1:

```
CREATE FUNCTION dbo.ServicesByEmployee(@IdEmployee int)
RETURNS TABLE
AS
RETURN (
    SELECT CONCAT(Client.firstName, ' ', Client.lastName) AS Client_FIO,
    Appointment.dateTime, Appointment.nameService
    FROM Appointment JOIN Employee ON
    Appointment.idEmployee=Employee.idEmployee
    JOIN Client ON
    Appointment.idClient=Client.idClient
    WHERE Appointment.idEmployee=@IdEmployee
);
```

Результаты выполнения запроса:

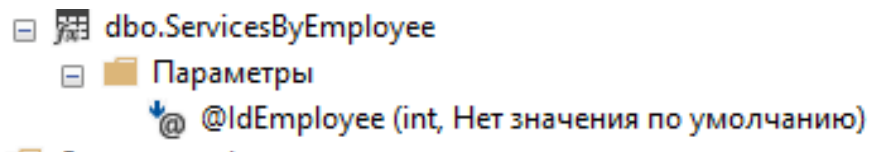


Рисунок 21. Функция ServicesByEmployee в обозревателе объектов

Проверим работу данной функции.

Запрос 2:

```
SELECT * FROM dbo.ServicesByEmployee (1);
```

Результаты выполнения запроса:

	Client_FIO	dateTime	nameService
1	Екатерина Елисеева	2022-12-15 15:00:00.000	Маникюр с покрытием
2	Мария Золотова	2022-12-15 10:00:00.000	Маникюр с покрытием

Рисунок 22. Результат работы функции ServicesByEmployee

6. Функция ServicesOfClient.

Функция, возвращающая таблицу записей выбранного клиента.

Запрос 1:

```
CREATE FUNCTION dbo.ServicesOfClient(@IdClient int)
RETURNS TABLE
```

AS

RETURN (

```
SELECT CONCAT(Client.firstName, ' ', Client.lastName) AS Client_FIO,  
CONCAT(Employee.firstName, ' ', Employee.lastName) AS Employee_FIO,  
Appointment.dateTime,  
Appointment.nameService,  
Appointment.price  
FROM Appointment JOIN Client ON  
Appointment.idClient=Client.idClient  
JOIN Employee ON  
Appointment.idEmployee=Employee.idEmployee  
WHERE Appointment.idClient=@IdClient
```

);

Результаты выполнения запроса:

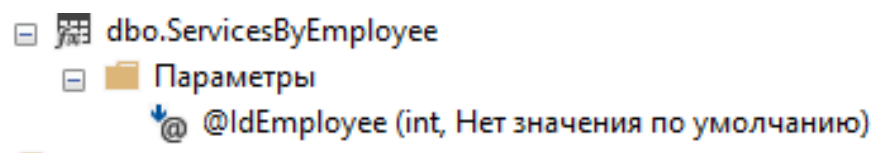


Рисунок 23. Функция ServicesOfClient в обозревателе объектов

Проверим работу данной функции.

Запрос 2:

```
SELECT * FROM dbo.ServicesOfClient (2);
```

Результаты выполнения запроса:

	Client_FIO	Employee_FIO	dateTime	nameService	price
1	Карина Соболева	Валерия Захарова	2022-12-15 12:00:00.000	Маникюр с укреплением гелем	3000
2	Карина Соболева	Мария Попатина	2022-12-24 13:00:00.000	Окрашивание	6000

Рисунок 24. Результат работы функции ServicesOfClient

7. Представление Bookings

Представление для вывода всех записей клиентов.

Запрос 1:

```
CREATE VIEW Bookings AS
```

```
SELECT CONCAT(Client.firstName, ' ', Client.lastName) AS Client_FIO,  
CONCAT(Employee.firstName, ' ', Employee.lastName) AS Employee_FIO,  
Client.telNumber,
```

```

Appointment.nameService,
Appointment.dateTime,
Appointment.price
FROM Client JOIN Appointment ON
Client.idClient=Appointment.idClient JOIN
Employee ON Employee.idEmployee=Appointment.idEmployee;

```

Результаты выполнения запроса:

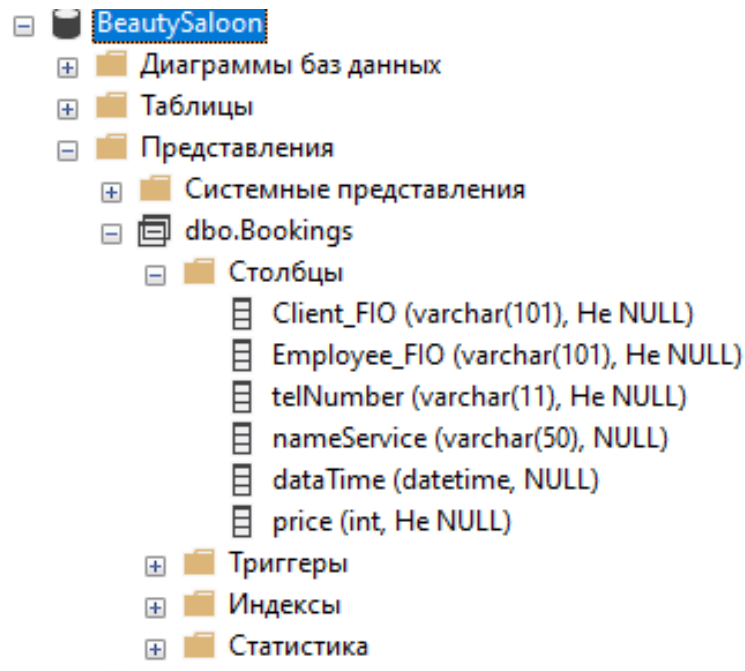


Рисунок 25. Представление Bookings в обозревателе объектов

Проверим работу данной функции.

Запрос 2:

```
SELECT * FROM dbo.Bookings;
```

Результаты выполнения запроса:

	Client_FIO	Employee_FIO	telNumber	nameService	dateTime	price
1	Екатерина Елисеева	Екатерина Петрова	+7965627820	Маникюр с покрытием	2022-12-15 15:00:00.000	2000
2	Карина Соболева	Валерия Захарова	+7034857842	Маникюр с укреплением гелем	2022-12-15 12:00:00.000	3000
3	Екатерина Елисеева	Мария Лопатина	+7965627820	Окрашивание	2022-12-21 15:00:00.000	4000
4	Регина Курочкина	Регина Соколова	+7953021214	Стрижка каре женская	2022-12-21 17:00:00.000	2000
5	Мария Золотова	Екатерина Петрова	+7954965285	Маникюр с покрытием	2022-12-15 10:00:00.000	2000
6	Марина Селезнёва	Мария Некрасова	+7936698548	Стрижка каре женская	2022-12-19 19:00:00.000	2500
7	Карина Соболева	Мария Лопатина	+7034857842	Окрашивание	2022-12-24 13:00:00.000	6000
8	Анастасия Абрамова	Анастасия Власова	+7911451474	Ламинирование бровей	2022-12-27 14:30:00.000	700

Рисунок 26. Результат работы представления Bookings

8. Процедура RemoveClient

Процедура, удаляющая данные клиента из таблицы Client. При попытке удаления клиента с несуществующим номером работника, произойдет исключение.

Запрос 1:

```
CREATE PROC RemoveClient
@IdClient int
AS BEGIN
IF (NOT EXISTS (SELECT * FROM Client where
idClient = @IdClient))
THROW 51000, 'No client found with such ID!', 1;
DELETE FROM Appointment
WHERE idClient = @IdClient;
DELETE FROM Client
WHERE idClient = @IdClient;
END;
```

Результаты выполнения запроса:

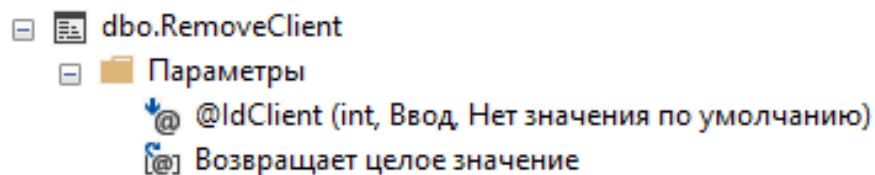


Рисунок 27. Функция RemoveClient в обозревателе объектов

Проверим работу данной функции.

Запрос 2:

```
EXEC RemoveClient 3;
```

Результаты выполнения запроса:

	idClient	firstName	lastName	telNumber			
1	1	Екатерина	Елисеева	+7965627820			
2	2	Карина	Соболева	+7034857842			
3	4	Регина	Курочкина	+7953021214			
4	5	Марина	Селезнёва	+7936698548			
5	6	Анастасия	Абрамова	+7911451474			

	idAppointment	price	dateTime	nameService	idClient	idEmployee
1	1	2000	2022-12-15 15:00:00.000	Маникюр с покрытием	1	1
2	2	3000	2022-12-15 12:00:00.000	Маникюр с укреплением гелем	2	2
3	3	4000	2022-12-21 15:00:00.000	Окрашивание	1	3
4	4	2000	2022-12-21 17:00:00.000	Стрижка каре женская	4	4
5	6	2500	2022-12-19 19:00:00.000	Стрижка каре женская	5	5
6	7	6000	2022-12-24 13:00:00.000	Окрашивание	2	3
7	8	700	2022-12-27 14:30:00.000	Ламинирование бровей	6	6

Рисунок 28. Результат работы функции RemoveClient

9. Процедура RemoveEmployee

Процедура, удаляющая данные работника из таблицы Employee. При попытке удалить работника с несуществующим персональным номером работника, произойдет исключение.

Запрос 1:

```
CREATE PROC RemoveEmployee
@IdEmployee int
AS BEGIN
IF (NOT EXISTS (SELECT * FROM Employee where
idEmployee = @IdEmployee))
THROW 51000, 'No employee found with such ID!', 1;
DELETE FROM Appointment
WHERE idEmployee = @IdEmployee;
DELETE FROM Employee
WHERE idEmployee = @IdEmployee;
END;
```

Результаты выполнения запроса:

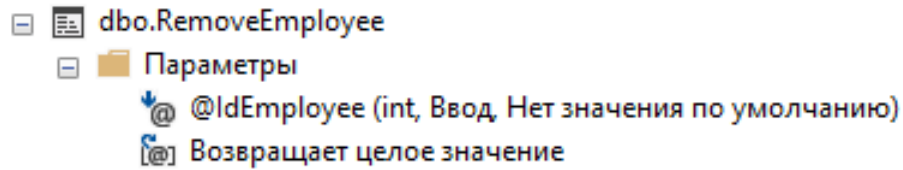


Рисунок 29. Функция RemoveEmployee в обозревателе объектов

Проверим работу данной функции.

Запрос 2:

```
EXEC RemoveEmployee 7;
```

сообщение: 51000, уровень: 16, состояние: 1, процедура: RemoveEmployee, строка: 6 [строка начала пакета: 0]
No employee found with such ID!

Рисунок 30. Результат работы функции RemoveEmployee

Запрос 3:

```
EXEC RemoveEmployee 6;
```

Результаты выполнения запроса:

	idEmployee	firstName	lastName	telNumber	idSpecials
1	1	Екатерина	Петрова	+7962417820	1
2	2	Валерия	Захарова	+7034528642	1
3	3	Мария	Попатина	+7954652585	2
4	4	Регина	Соколова	+7954200214	2
5	5	Мария	Некрасова	+7936304078	2

	idAppointment	price	dateTime	nameService	idClient	idEmployee
1	1	2000	2022-12-15 15:00:00.000	Маникюр с покрытием	1	1
2	2	3000	2022-12-15 12:00:00.000	Маникюр с укреплением гелем	2	2
3	3	4000	2022-12-21 15:00:00.000	Окрашивание	1	3
4	4	2000	2022-12-21 17:00:00.000	Стрижка каре женская	4	4
5	6	2500	2022-12-19 19:00:00.000	Стрижка каре женская	5	5
6	7	6000	2022-12-24 13:00:00.000	Окрашивание	2	3

Рисунок 31. Результат работы функции RemoveEmployee

10. Процедура RemoveAppointment

Процедура, удаляющая запись из таблицы Appointment. При попытке удалить запись с несуществующим персональным номером записи, произойдет исключение.

Запрос 1:

```
CREATE PROC RemoveAppointment  
@IdAppointment int  
AS BEGIN
```



```

IF (NOT EXISTS (SELECT * FROM Appointment where
idAppointment = @IdAppointment))
THROW 51000, 'No appointment found with such ID!', 1;
DELETE FROM Appointment
WHERE idAppointment = @IdAppointment;
END;

```

Результаты выполнения запроса:

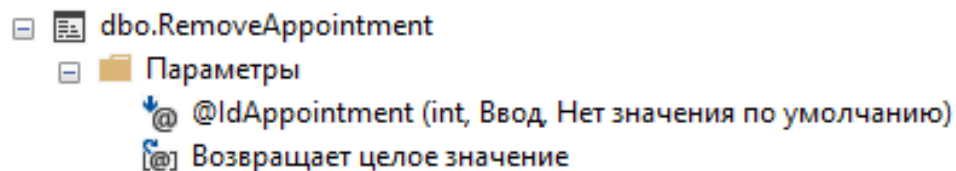


Рисунок 32. Функция RemoveAppointment в обозревателе объектов

Проверим работу данной функции.

Запрос 2:

```
EXEC RemoveAppointment 6;
```

Результаты выполнения запроса:

	idAppointment	price	dateTime	nameService	idClient	idEmployee
1	1	2000	2022-12-15 15:00:00.000	Маникюр с покрытием	1	1
2	2	3000	2022-12-15 12:00:00.000	Маникюр с укреплением гелем	2	2
3	3	4000	2022-12-21 15:00:00.000	Окрашивание	1	3
4	4	2000	2022-12-21 17:00:00.000	Стрижка каре женская	4	4
5	7	6000	2022-12-24 13:00:00.000	Окрашивание	2	3

Рисунок 33. Результат работы функции RemoveAppointment

Запрос 3:

```
EXEC RemoveAppointment 42;
```

Результаты выполнения запроса:

```

сообщение: 51000, уровень: 16, состояние: 1, процедура: RemoveAppointment, строка: 6 [строка начала пакета: 0]
No appointment found with such ID!

```

Рисунок 34. Результат работы функции RemoveAppointment

7. Разработка стратегии резервного копирования

Для начала в свойствах базы данных BeautySaloon изменим модель восстановления с простой на полную для обеспечения максимальных возможностей восстановления

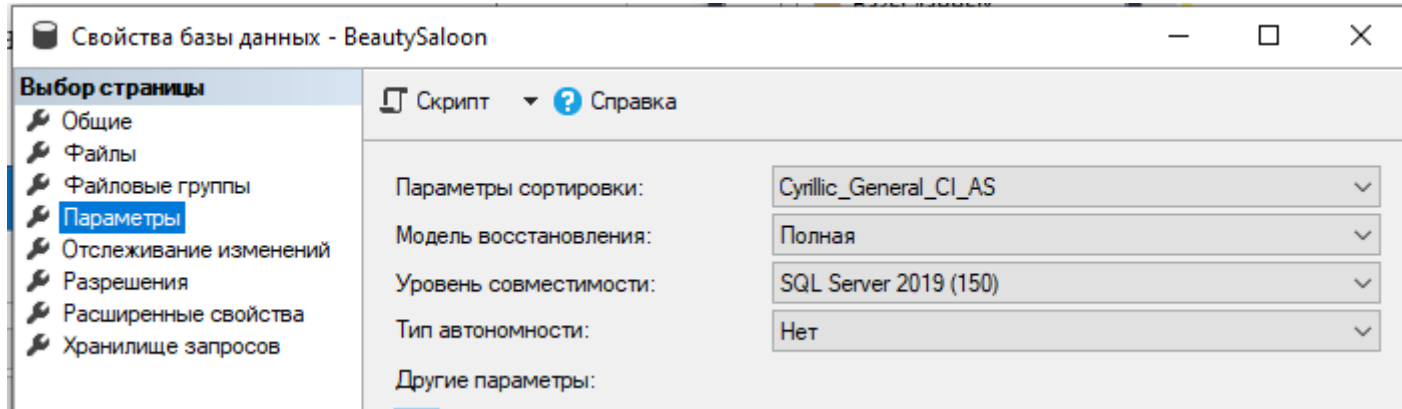


Рисунок 35. Изменение модели восстановления базы данных

В качестве стратегии резервного копирования для нашей базы данных выберем дифференциальное копирование. При дифференциальном резервном копировании для начала единожды создаётся полная резервная копия базы, а затем, при каждом следующем копировании, сохраняются только данные, изменившиеся по сравнению с полной копией.

Создадим полную резервную копию базы данных.

Запрос 1:

```
BACKUP DATABASE BeautySaloon
TO DISK = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\Backup\BeautySaloon.bak'
WITH INIT;
```

Результаты выполнения запроса:

```
Обработано 480 страниц для базы данных "BeautySaloon", файл "Beauty Saloon" для файла 1.
Обработано 1 страниц для базы данных "BeautySaloon", файл "Beauty_Saloon_log" для файла 1.
BACKUP DATABASE успешно обработал 481 страниц за 0.051 секунд (73.558 МБ/сек).

Время выполнения: 2022-12-20T23:11:22.0038364+03:00
```

Рисунок 36. Результат полного резервного копирования базы данных

Создадим разностную резервную копию базы данных.

Запрос 2:

```
BACKUP DATABASE BeautySaloon
TO DISK = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\Backup\BeautySaloonDiff.bak'
WITH DIFFERENTIAL, INIT;
```

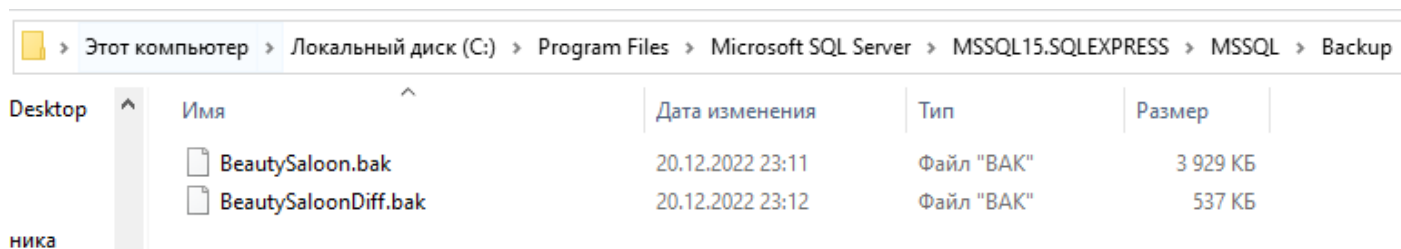
Результаты выполнения запроса:

```
Обработано 56 страниц для базы данных "BeautySaloon", файл "Beauty Saloon" для файла 1.
Обработано 1 страниц для базы данных "BeautySaloon", файл "Beauty_Saloon_log" для файла 1.
BACKUP DATABASE WITH DIFFERENTIAL успешно обработал 57 страниц за 0.037 секунд (11.863 МБ/сек).

Время выполнения: 2022-12-20T23:12:39.8960745+03:00
```

Рисунок 37. Результат создания разностной резервной копии

После выполнения двух предыдущих запросов у нас появились следующие файлы:



The screenshot shows a Windows File Explorer window with the address bar set to 'Этот компьютер > Локальный диск (C:) > Program Files > Microsoft SQL Server > MSSQL15.SQLEXPRESS > MSSQL > Backup'. The left sidebar shows 'Desktop' and 'ника'. The main area displays a table of files:

Имя	Дата изменения	Тип	Размер
BeautySaloon.bak	20.12.2022 23:11	Файл "BAK"	3 929 КБ
BeautySaloonDiff.bak	20.12.2022 23:12	Файл "BAK"	537 КБ

Рисунок 33. Резервные копии

Для восстановления базы данных следует выполнить запрос ниже.

Запрос 3:

```
USE BeautySaloon
RESTORE DATABASE BeautySaloon
FROM DISK = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\Backup\BeautySaloon.bak'
WITH NORECOVERY

RESTORE DATABASE BeautySaloon
FROM DISK = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\Backup\BeautySaloonDiff.bak'
WITH NORECOVERY

RESTORE DATABASE BeautySaloon WITH RECOVERY
```

ЗАКЛЮЧЕНИЕ

В результате выполнения индивидуального домашнего задания была спроектирована и создана база данных для салона красоты. В процессе разработки базы данных были применены полученные знания о проектировании и создании базы данных, таблиц с ограничениями целостности и о заполнении таблиц данными. Также были созданы объекты промежуточного слоя, такие как процедуры, пользовательские функции и представления.

Кроме того, была разработана стратегия дифференциального резервного копирования, при которой копируются только данные, изменившиеся по сравнению с полной копией. Преимуществами такой стратегии являются экономия места и времени выполнения, а также быстрота восстановления данных.

Таким образом, были закреплены на практике навыки разработки баз данных и работы с ними.

Список использованных источников

1) Распределенные базы данных: Методические указания к лаб. работам / Сост.: А.В. Горячев, Н.Е. Новакова. – СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2008. – 32 с.

2) Домашняя страница документации и учебных ресурсов Майкрософт для разработчиков и технических специалистов // URL: <https://docs.microsoft.com/ru-ru/>

3) Lucidchart – интеллектуальное построение диаграмм // URL: <https://www.lucidchart.com/pages/ru/erd-%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0>