

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине: «Разработка приложений баз данных для информационных систем»

на тему: «Разработка серверной части информационной системы  
в СУБД *MS SQL Server*»

Выполнил: студент гр. ИТП-31

Ермолин Д.К.

Принял: ректор

Асенчик О.Д.

Гомель 2023

**Цель работы:** разработать серверную часть клиент-серверной информационной системы, основанной на базе данных в заданной предметной области средствами СУБД *MS SQL Server*.

**Задание:**

1. Разработать логическую модель реляционной базы данных, моделирующую предметную область согласно своему варианту задания. Структура БД должна быть нормализована – таблицы должны удовлетворять требованиям третьей нормальной формы.

2. Создать базу данных и таблицы в СУБД *MS SQL Server* и заполнить их тестовым набором данных, для этого написать *Transact SQL* скрипт:

2.1. Создания базы данных и ее таблиц.

При создании таблиц должны быть назначены первичные и внешние ключи и установлены необходимые ограничения целостности данных. Наименования таблиц и полей давать в соответствии с соглашением об именовании этих объектов.

2.2. Заполнения не менее чем трех таблиц БД данными (см. пример внутри этого электронного курса).

При выборе таблиц для заполнения тестовыми наборами руководствоваться следующим:

выбранные таблицы должны содержать основную информацию, касающуюся предметной области приложения;

таблицы должны быть связаны непосредственно;

следует воздерживаться от выбора таблиц, характеризующих кадровую подсистему приложения (данные о сотрудниках и их должностях);

не менее, чем одна таблица должна находиться в схеме базы данных на стороне отношения «многие».

При этом заполнение осуществлять в следующем порядке: сначала генерируются данные для таблиц на стороне отношения «один» (таблицы-«справочники»), потом – в таблицы на стороне отношения «многие» («оперативные таблицы»).

БД заполнить записями в количестве, необходимом для отладки и демонстрации возможностей приложения. Таблицы на стороне отношения один должны содержать не менее 500 записей, таблицы на стороне отношения многие должны содержать не менее 20000 записей.

3. Создать с использованием средств *Transact SQL* представления, позволяющие отображать данные в удобном для пользователя виде, и заменяющие часто используемые запросы на выборку из двух и более связанных таблиц.

4. Написать не менее трех хранимых процедур с параметрами для вставки и (или) обновление данных в таблицы базы данных.

Вариант задания указан на рисунке 1.

1. Виды отдыха (наименование, описание, ограничения).
2. Отели (наименование, страна, город, адрес, телефон, количество звёзд, контактное лицо, фото отеля и номеров).
3. Дополнительные услуги (наименование, описание, цена).
4. Клиенты (ФИО, дата рождения, пол, адрес, телефон, паспортные данные, скидка).
5. Путёвки (дата начала, дата окончания, отель, наименование вида отдыха, перечень дополнительных услуг, клиент, сотрудник, отметка о бронировании, отметка об оплате).

Рисунок 1 – Вариант задания

### Ход работы

В начале процесса проектирования базы данных была создана таблица с названием "*AdditionalServices*", которая разработана с целью добавления дополнительных услуг для клиента в отеле. Пример таблицы указан на рисунке 2. Листинг создания этой таблицы указана в приложении А.

51	Аренда	Аренда машины	155,2200
52	Лодка	Аренда лодки с выездом в море	144,0000

Рисунок 2 – Пример таблицы *AdditionalServices*

Далее была создана таблица с названием "*Hotels*", в которой содержится информация о названии отеля, страна, улица, адрес, телефон, количество звёзд, контактное лицо и фото. Пример таблицы указан на рисунке 3. Листинг создания этой таблицы указана в приложении А.

Id	Name	Country	City	Address	Phone	Stars
1	Отель 1	Страна 1	Город 1	ул. Улица 1, д. 1	12345678901	2
2	Отель 2	Страна 2	Город 2	ул. Улица 2, д. 2	12345678902	3
3	Отель 3	Страна 3	Город 3	ул. Улица 3, д. 3	12345678903	4
4	Отель 4	Страна 4	Город 4	ул. Улица 4, д. 4	12345678904	5
5	Отель 5	Страна 5	Город 5	ул. Улица 5, д. 5	12345678905	1

Рисунок 3 – Пример таблицы *Hotels*

После была разработана таблица "*Clients*", в которой хранилась информация о ФИО, дата рождения, пол, адрес, телефон, паспортные данные, скидка. Пример таблицы указан на рисунке 4. Листинг создания этой таблицы указана в приложении А.

Id	FIO	DateOfBirth	Sex	Address	Series	Number	Discount
1	Дмитрий Петров	1939-05-07	Мужской	Адрес 0	BD	1447502791	1
2	Екатерина Козлов	1983-04-21	Женский	Адрес 1	56	134500248	5
3	Анна Иванов	1984-01-09	Мужской	Адрес 2	FB	107904553	4
4	Мария Сидоров	1941-11-07	Женский	Адрес 3	E5	37164407	17
5	Иван Смирнов	1998-01-06	Мужской	Адрес 4	9A	1091371489	5
6	Александр Петров	1972-04-16	Женский	Адрес 5	E4	1541915436	15
7	Мария Козлов	1968-05-07	Мужской	Адрес 6	82	860388735	15
8	Иван Сидоров	1926-02-06	Женский	Адрес 7	71	1540736665	12
9	Иван Иванов	1982-07-11	Мужской	Адрес 8	13	391892029	0

Рисунок 4 – Пример таблицы *Clients*

Далее была создана таблица с названием "*Employees*", в которой содержится информация о ФИО, должности и возрасте. Пример таблицы указан на рисунке 5. Листинг создания этой таблицы указана в приложении А.

	id	name	middlename	sumame	address_id	mobile_phone	email
1	1	Иван	Петрович	Смирнов	50	+375661035312	CCA528DB-7@gmail.com
2	2	Иван	Иванович	Сидоров	77	+375635249163	B9E6F521-4@gmail.com
3	3	Александр	Александрович	Антонова	50	+375169649821	0CB11227-5@gmail.com
4	4	Евгений	Андреевич	Смирнов	66	+375652379774	F61A6CB4-1@gmail.com
5	5	Иван	Сергеевна	Иванов	50	+375536456322	BB448D9C-5@gmail.com
6	6	Евгений	Андреевич	Антонова	31	+375707283833	AD0EB7D1-F@gmail.com
7	7	Петр	Андреевич	Петров	58	+375892958381	C45997A2-3@gmail.com
8	8	Евгений	Петрович	Петров	14	+375145098802	A17B58AD-1@gmail.com
9	9	Александр	Петрович	Иванов	41	+375374132174	D6479D0B-A@gmail.com
10	10	Кирил	Александрович	Петров	18	+375910414843	02ABA095-C@gmail.com
11	11	Петр	Сергеевна	Сидоров	27	+375899458025	8B66CC7E-C@gmail.com
12	12	Евгений	Александрович	Смирнов	92	+375301389832	1D10CD4C-3@gmail.com

Рисунок 5 – Пример таблицы *Employees*

В дальнейшей работе была создана таблица "*TypesOfRecreation*". Эта таблица содержит информацию о наименовании, описании, ограничениях. Пример таблицы указан на рисунке 6. Листинг создания этой таблицы указана в приложении А.

Id	Name	Description	Restrictions
1	Пляжный отдых	Отдых на берегу моря	Нет особых ограничений
2	Горнолыжный отдых	Катание на горных лыжах	Требуется опыт вождения лыж
3	Экскурсии	Посещение туристических мест и достопримечательностей	Возможны долгие поездки
4	Активный отдых	Спортивные мероприятия и приключения	Физическая подготовка необходима
5	Культурный отдых	Посещение музеев и исторических мест	Без специальных ограничений
6	Гастрономический туризм	Путешествие для гурманов	Возможны ограничения по диете
7	Спа-отдых	Посещение спа-центров и процедур	Могут потребоваться предварительные...
8	Экологический туризм	Путешествие по экологически чистым местам	Соблюдение правил охраны природы
9	Круизы	Путешествие на круизном лайнере	Важно учитывать стихийные бедствия
10	Альпинизм	Покорение вершин и скал	Высокая физическая подготовка
11	Путешествие с детьми	Семейный отдых с детьми	Удобства и развлечения для детей
12	Фестивали и мероприятия	Посещение фестивалей и культурных событий	Сезонность мероприятий

Рисунок 6 – Пример таблицы *TypesOfRecreation*

Далее, в ходе проектирования базы данных, была разработана таблица с названием *"Vouchers"*, предназначенная для хранения информации о дате начала, дате окончания, отель, наименование вида отдыха, перечень дополнительных услуг, клиент, сотрудник, отметка о бронировании, отметка об оплате. Пример таблицы указан на рисунке 7. Листинг создания этой таблицы указана в приложении А.

Id	StartDate	ExpirationDate	HotelId	TypeOfRecreat...	AdditionalServ...	ClientId	EmployessId	Reservation	Payment
1	2023-02-12	2023-02-13	50	47	5	25	21	False	False
2	2022-11-03	2022-11-09	26	7	4	27	14	True	False
3	2023-06-14	2023-07-04	31	19	7	14	39	False	True
4	2023-08-21	2023-09-20	23	11	1	2	25	True	False
5	2023-07-13	2023-08-08	5	1	1	15	50	False	False
6	2023-08-19	2023-09-12	35	19	6	4	50	True	True
7	2023-07-02	2023-07-04	30	38	5	19	32	False	False
8	2022-09-27	2022-10-12	41	16	8	50	31	True	False
9	2022-12-03	2022-12-06	9	31	4	35	20	False	True
10	2023-02-03	2023-02-10	8	46	7	17	6	True	False
11	2022-10-04	2022-10-07	28	41	10	28	37	False	False

Рисунок 7 – Пример таблицы *Vouchers*

После создания базы данных, была создана диаграмма, которая иллюстрирует зависимости между таблицами. Пример диаграммы базы данных указан на рисунке 8.

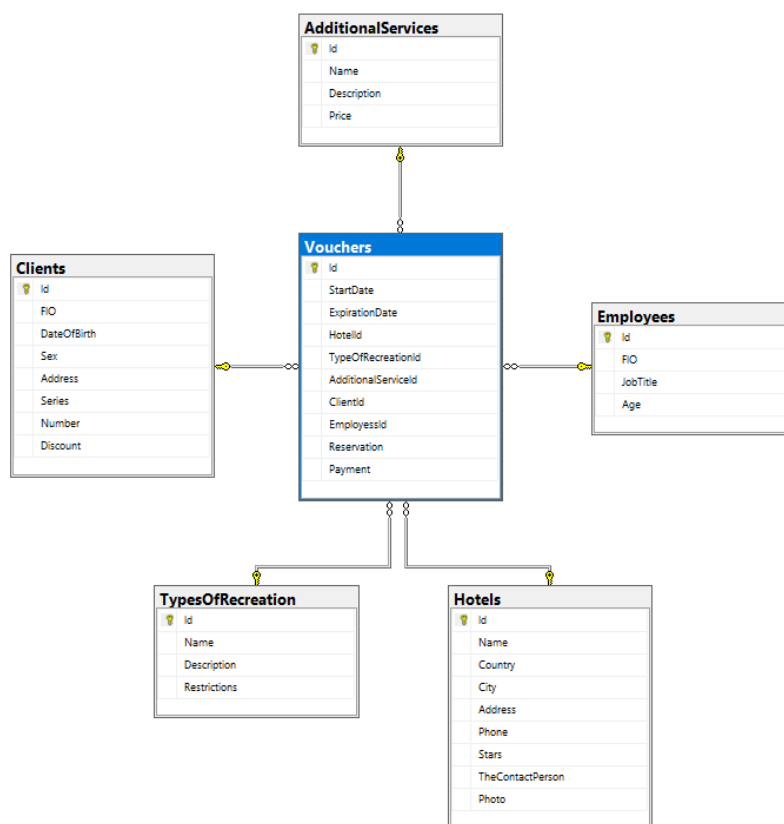


Рисунок 8 – Пример диаграммы базы данных

После написания всех скриптов для создания таблиц был написан скрипт для заполнения таблиц. Пример данного скрипта указан в приложение А. Далее были созданы три представления для удобного получения доступа к данным из таблицы.

Первым было создано представления для получения всех данных из таблицы "Hotels". Пример работы данного представления указан на рисунке 9. Листинг кода по созданию этого представления указан в приложении А.

Id	Name	Country	City	Address	Phone	Stars	TheContactPer...
1	Отель 1	Страна 1	Город 1	ул. Улица 1, д. 1	12345678901	2	Контактное ли...
2	Отель 2	Страна 2	Город 2	ул. Улица 2, д. 2	12345678902	3	Контактное ли...
3	Отель 3	Страна 3	Город 3	ул. Улица 3, д. 3	12345678903	4	Контактное ли...
4	Отель 4	Страна 4	Город 4	ул. Улица 4, д. 4	12345678904	5	Контактное ли...
5	Отель 5	Страна 5	Город 5	ул. Улица 5, д. 5	12345678905	1	Контактное ли...
6	Отель 6	Страна 6	Город 6	ул. Улица 6, д. 6	12345678906	2	Контактное ли...
7	Отель 7	Страна 7	Город 7	ул. Улица 7, д. 7	12345678907	3	Контактное ли...
8	Отель 8	Страна 8	Город 8	ул. Улица 8, д. 8	12345678908	4	Контактное ли...
9	Отель 9	Страна 9	Город 9	ул. Улица 9, д. 9	12345678909	5	Контактное ли...
10	Отель 10	Страна 10	Город 10	ул. Улица 10, д. ....	123456789010	1	Контактное ли...
11	Отель 11	Страна 11	Город 11	ул. Улица 11, д. ....	123456789011	2	Контактное ли...

Рисунок 9 – Пример работы представления *HotelView*

Вторым было создано представления для получения данных из таблицы "Employees", где находило работников старше 30 лет. Пример работы данного представления указан на рисунке 10. Листинг кода по созданию этого представления указан в приложении А.

	Id	FIO	JobTitle	Age
1	2	Robert Smith	Employee	32
2	3	Sarah Williams	Employee	38
3	4	Robert Brown	Employee	41
4	5	Sarah Davis	Employee	37
5	6	Robert Davis	Manager	42
6	7	Sarah Brown	Employee	38
7	8	Jane Brown	Employee	37
8	10	Jane Johnson	Employee	40
9	12	John Johnson	Employee	39

Рисунок 10 – Пример работы представления *EmployeesOver30*

Третьим было создано представления для получения всех данных о путёвках. Пример работы данного представления указан на рисунке 11. Листинг кода по созданию этого представления указан в приложении А.

VoucherId	StartDate	ExpirationDate	HotelName	ClientName	EmployeeName	RecreationType	AdditionalS...	Reservation	Payment
1	2023-06-12	2023-07-08	Отель 46	Екатерина Петров	Jane Johnson	Культурный отдых	Услуга 10	False	False
2	2022-12-08	2023-01-07	Отель 30	Александр Смирнов	Emily Davis	Спортивные мероприя...	Услуга 1	True	False
3	2023-01-19	2023-02-10	Отель 26	Александр Петров	John Johnson	Спортивные мероприя...	Услуга 1	False	True
4	2022-10-28	2022-11-27	Отель 20	Александр Сидоров	Jane Williams	Экстремальный спорт	Услуга 8	True	False
5	2023-04-27	2023-05-25	Отель 4	Екатерина Иванов	Robert Williams	Экскурсии	Услуга 7	False	False
6	2023-05-15	2023-05-26	Отель 35	Иван Козлов	Jane Williams	Лошади и верховая езда	Услуга 7	True	True
7	2022-10-19	2022-10-30	Отель 38	Мария Петров	Sarah Johnson	Зоотуризм	Услуга 1	False	False
8	2023-05-03	2023-05-18	Отель 40	Дмитрий Сидоров	Michael Brown	Путешествие в лес	Услуга 7	True	False
9	2023-03-27	2023-04-18	Отель 33	Иван Смирнов	Jane Davis	Сафари	Услуга 7	False	True
10	2022-11-24	2022-12-10	Отель 1	Екатерина Петров	Robert Smith	Горнолыжный отдых	Услуга 8	True	False
11	2023-07-05	2023-08-02	Отель 33	Анна Иванов	Robert Williams	Альпинизм	Услуга 2	False	False

Рисунок 11 – Пример работы представления *VoucherView*

Далее были реализованы пять хранимых процедур для добавления новых записей в таблицу *Hotels*, обновления записей в таблицу *Clients* и удаления данных по *Id* в таблице *AdditionalServices*. Листинг данных хранимых процедур указан в приложении А.

**Вывод:** в ходе выполнения лабораторной работы была успешно разработана и настроена база данных в среде *MS SQL Server*, предназначенная для использования в серверной части клиент-серверного приложения. Эта база данных включает в себя необходимые таблицы, которые были заполнены данными, а также были созданы хранимые процедуры, обеспечивающие возможность вставки новых записей в эти таблицы.

## ПРИЛОЖЕНИЕ А

### Листинг скрипта для генерации базы данных

```
USE master;
GO

IF NOT EXISTS (SELECT name FROM sys.databases WHERE name = 'TouristAgency1')
BEGIN
    CREATE DATABASE TouristAgency1;
END
GO

USE TouristAgency1;
GO

-- Таблица для дополнительных услуг
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'dbo' AND TABLE_NAME =
'AdditionalServices')
BEGIN
    CREATE TABLE [dbo].[AdditionalServices](
        [Id] [int] IDENTITY(1,1) NOT NULL,
        [Name] [nvarchar](50) NOT NULL,
        [Description] [nvarchar](200) NOT NULL,
        [Price] [money] NOT NULL,
        PRIMARY KEY CLUSTERED ([Id] ASC)
    );
END

-- Таблица для сотрудников
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'dbo' AND TABLE_NAME =
'Employees')
BEGIN
    CREATE TABLE [dbo].[Employees](
        [Id] [int] IDENTITY(1,1) NOT NULL,
        [FIO] [nvarchar](150) NOT NULL,
        [JobTitle] [nvarchar](50) NOT NULL,
        [Age] [int] NOT NULL,
        PRIMARY KEY CLUSTERED ([Id] ASC)
    );
END

-- Таблица для видов отдыха
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'dbo' AND TABLE_NAME =
'TypesOfRecreation')
BEGIN
    CREATE TABLE [dbo].[TypesOfRecreation](
        [Id] [int] IDENTITY(1,1) NOT NULL,
        [Name] [nvarchar](50) NOT NULL,
        [Description] [nvarchar](100) NOT NULL,
        [Restrictions] [nvarchar](50) NOT NULL,
        PRIMARY KEY CLUSTERED ([Id] ASC)
    );
END

-- Таблица для клиентов
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'dbo' AND TABLE_NAME = 'Clients')
```



BEGIN

```
CREATE TABLE [dbo].[Clients](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [FIO] [nvarchar](150) NOT NULL,
    [DateOfBirth] [date] NOT NULL,
    [Sex] [nvarchar](50) NOT NULL,
    [Address] [nvarchar](100) NOT NULL,
    [Series] [nvarchar](50) NOT NULL,
    [Number] [bigint] NOT NULL,
    [Discount] [bigint] NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

END

-- Таблица для отелей

IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Hotels')

BEGIN

```
CREATE TABLE [dbo].[Hotels](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](50) NOT NULL,
    [Country] [nvarchar](50) NOT NULL,
    [City] [nvarchar](50) NOT NULL,
    [Address] [nvarchar](100) NOT NULL,
    [Phone] [nvarchar](20) NOT NULL,
    [Stars] [int] NOT NULL,
    [TheContactPerson] [nvarchar](100) NOT NULL,
    [Photo] [image] NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

END

-- Таблица для ваучеров

IF NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_SCHEMA = 'dbo' AND TABLE\_NAME = 'Vouchers')

BEGIN

```
CREATE TABLE [dbo].[Vouchers](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [StartDate] [date] NOT NULL,
    [ExpirationDate] [date] NOT NULL,
    [HotelId] [int] NOT NULL,
    [TypeOfRecreationId] [int] NOT NULL,
    [AdditionalServiceId] [int] NOT NULL,
    [ClientId] [int] NOT NULL,
    [EmployessId] [int] NOT NULL,
    [Reservation] [bit] NOT NULL,
    [Payment] [bit] NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

```
ALTER TABLE [dbo].[Vouchers] WITH CHECK ADD CONSTRAINT [FK_Vouchers_AdditionalServices] FOREIGN
KEY([AdditionalServiceId])
REFERENCES [dbo].[AdditionalServices] ([Id]);
```

```
ALTER TABLE [dbo].[Vouchers] WITH CHECK ADD CONSTRAINT [FK_Vouchers_Clients] FOREIGN KEY([ClientId])
REFERENCES [dbo].[Clients] ([Id]);
```

```

ALTER TABLE [dbo].[Vouchers] WITH CHECK ADD CONSTRAINT [FK_Vouchers_Employees] FOREIGN KEY([EmployeeId])
REFERENCES [dbo].[Employees] ([Id]);

ALTER TABLE [dbo].[Vouchers] WITH CHECK ADD CONSTRAINT [FK_Vouchers_Hotels] FOREIGN KEY([HotelId])
REFERENCES [dbo].[Hotels] ([Id]);

ALTER TABLE [dbo].[Vouchers] WITH CHECK ADD CONSTRAINT [FK_Vouchers_TypesOfRecreation] FOREIGN
KEY([TypeOfRecreationId])
REFERENCES [dbo].[TypesOfRecreation] ([Id]);
END

-- Проверяем, пуста ли таблица Hotels
IF NOT EXISTS (SELECT * FROM [dbo].[Hotels])
BEGIN
    -- Заполняем таблицу 50 примерами данных
    DECLARE @HotelIndex INT;
    SET @HotelIndex = 1;

    WHILE @HotelIndex <= 50
    BEGIN
        INSERT INTO [dbo].[Hotels] ([Name], [Country], [City], [Address], [Phone], [Stars], [TheContactPerson], [Photo])
        VALUES (
            'Отель ' + CAST(@HotelIndex AS NVARCHAR(10)), -- Уникальное название отеля
            'Страна ' + CAST(@HotelIndex AS NVARCHAR(10)), -- Уникальная страна
            'Город ' + CAST(@HotelIndex AS NVARCHAR(10)), -- Уникальный город
            'ул. Улица ' + CAST(@HotelIndex AS NVARCHAR(10)) + ', д. ' + CAST(@HotelIndex AS NVARCHAR(10)), -- Уникальный
адрес
            '1234567890' + CAST(@HotelIndex AS NVARCHAR(10)), -- Уникальный телефон
            @HotelIndex % 5 + 1, -- Рейтинг отеля (звезды)
            'Контактное лицо ' + CAST(@HotelIndex AS NVARCHAR(10)), -- Уникальное контактное лицо
            0x -- Пустое изображение (для столбца Photo)
        );

        SET @HotelIndex = @HotelIndex + 1;
    END
END

-- Проверяем, пуста ли таблица AdditionalServices
IF NOT EXISTS (SELECT * FROM [dbo].[AdditionalServices])
BEGIN
    -- Проверяем, пуста ли таблица AdditionalServices
    IF (SELECT COUNT(*) FROM [dbo].[AdditionalServices]) = 0
    BEGIN
        -- Заполняем таблицу 50 примерами данных
        INSERT INTO [dbo].[AdditionalServices] ([Name], [Description], [Price])
        VALUES
            ('Услуга 1', 'Описание услуги 1', 10.99),
            ('Услуга 2', 'Описание услуги 2', 25.50),
            ('Услуга 3', 'Описание услуги 3', 5.75),
            ('Услуга 4', 'Описание услуги 4', 30.00),
            ('Услуга 5', 'Описание услуги 5', 15.25),
            ('Услуга 6', 'Описание услуги 6', 12.99),
            ('Услуга 7', 'Описание услуги 7', 8.50),
            ('Услуга 8', 'Описание услуги 8', 40.00),
            ('Услуга 9', 'Описание услуги 9', 18.75),
            ('Услуга 10', 'Описание услуги 10', 22.50),

```

('Услуга 11', 'Описание услуги 11', 13.99),  
('Услуга 12', 'Описание услуги 12', 7.25),  
('Услуга 13', 'Описание услуги 13', 9.99),  
('Услуга 14', 'Описание услуги 14', 11.50),  
('Услуга 15', 'Описание услуги 15', 35.00),  
('Услуга 16', 'Описание услуги 16', 28.75),  
('Услуга 17', 'Описание услуги 17', 17.50),  
('Услуга 18', 'Описание услуги 18', 21.99),  
('Услуга 19', 'Описание услуги 19', 6.25),  
('Услуга 20', 'Описание услуги 20', 14.50),  
('Услуга 21', 'Описание услуги 21', 31.00),  
('Услуга 22', 'Описание услуги 22', 19.75),  
('Услуга 23', 'Описание услуги 23', 27.99),  
('Услуга 24', 'Описание услуги 24', 23.50),  
('Услуга 25', 'Описание услуги 25', 20.25),  
('Услуга 26', 'Описание услуги 26', 45.00),  
('Услуга 27', 'Описание услуги 27', 33.75),  
('Услуга 28', 'Описание услуги 28', 12.50),  
('Услуга 29', 'Описание услуги 29', 15.99),  
('Услуга 30', 'Описание услуги 30', 8.25),  
('Услуга 31', 'Описание услуги 31', 24.50),  
('Услуга 32', 'Описание услуги 32', 29.00),  
('Услуга 33', 'Описание услуги 33', 9.75),  
('Услуга 34', 'Описание услуги 34', 10.99),  
('Услуга 35', 'Описание услуги 35', 22.50),  
('Услуга 36', 'Описание услуги 36', 13.25),  
('Услуга 37', 'Описание услуги 37', 16.50),  
('Услуга 38', 'Описание услуги 38', 32.00),  
('Услуга 39', 'Описание услуги 39', 18.75),  
('Услуга 40', 'Описание услуги 40', 25.99),  
('Услуга 41', 'Описание услуги 41', 7.25),  
('Услуга 42', 'Описание услуги 42', 11.50),  
('Услуга 43', 'Описание услуги 43', 35.00),  
('Услуга 44', 'Описание услуги 44', 28.75),  
('Услуга 45', 'Описание услуги 45', 17.50),  
('Услуга 46', 'Описание услуги 46', 21.99),  
('Услуга 47', 'Описание услуги 47', 6.25),  
('Услуга 48', 'Описание услуги 48', 14.50),  
('Услуга 49', 'Описание услуги 49', 31.00),  
('Услуга 50', 'Описание услуги 50', 19.75);

END

END

-- Проверяем, пуста ли таблица Clients

IF NOT EXISTS (SELECT \* FROM [dbo].[Clients])

BEGIN

-- Переменные для генерации реалистичных данных

DECLARE @FirstNames TABLE (Name NVARCHAR(50));

DECLARE @LastNames TABLE (Name NVARCHAR(50));

INSERT INTO @FirstNames (Name) VALUES

('John'), ('Jane'), ('Robert'), ('Emily'), ('Michael'), ('Sarah');

INSERT INTO @LastNames (Name) VALUES

('Smith'), ('Johnson'), ('Williams'), ('Brown'), ('Davis');

-- Проверяем, пуста ли таблица Clients

IF NOT EXISTS (SELECT \* FROM [dbo].[Clients])

BEGIN

```

-- Переменные для генерации реалистичных данных
DECLARE @CustomFirstNames TABLE (Name NVARCHAR(50));
DECLARE @CustomLastNames TABLE (Name NVARCHAR(50));
INSERT INTO @CustomFirstNames (Name) VALUES
    ('Иван'), ('Мария'), ('Александр'), ('Екатерина'), ('Дмитрий'), ('Анна');
INSERT INTO @CustomLastNames (Name) VALUES
    ('Иванов'), ('Петров'), ('Сидоров'), ('Козлов'), ('Смирнов');

-- Генерируем случайные данные для клиентов
WHILE (SELECT COUNT(*) FROM [dbo].[Clients]) < 50
BEGIN
    DECLARE @CustomFirstName NVARCHAR(50);
    DECLARE @CustomLastName NVARCHAR(50);
    DECLARE @DateOfBirth DATE;
    DECLARE @Sex NVARCHAR(50);
    DECLARE @Address NVARCHAR(100);
    DECLARE @Series NVARCHAR(50);
    DECLARE @Number BIGINT;
    DECLARE @Discount BIGINT;

    SELECT TOP 1 @CustomFirstName = Name FROM @CustomFirstNames ORDER BY NEWID();
    SELECT TOP 1 @CustomLastName = Name FROM @CustomLastNames ORDER BY NEWID();
    SET @DateOfBirth = DATEADD(DAY, -1 * (ABS(CHECKSUM(NEWID())) % 36525), GETDATE()); -- Генерируем дату
    рождения в пределах 100 лет
    SET @Sex = CASE WHEN (SELECT COUNT(*) FROM [dbo].[Clients]) % 2 = 0 THEN 'Мужской' ELSE 'Женский' END;
    SET @Address = 'Адрес ' + CAST((SELECT COUNT(*) FROM [dbo].[Clients]) AS NVARCHAR(10));
    SET @Series = LEFT(CAST(NEWID() AS NVARCHAR(36)), 2); -- Случайная серия (первые два символа GUID)
    SET @Number = ABS(CAST(CHECKSUM(NEWID()) AS BIGINT)) % 10000000000; -- Случайный номер в пределах 10
    миллиардов
    SET @Discount = ABS(CAST(CHECKSUM(NEWID()) AS INT)) % 20; -- Случайная скидка в пределах 20

    -- Вставляем данные в таблицу
    INSERT INTO [dbo].[Clients] ([FIO], [DateOfBirth], [Sex], [Address], [Series], [Number], [Discount])
    VALUES (CONCAT(@CustomFirstName, ' ', @CustomLastName), @DateOfBirth, @Sex, @Address, @Series, @Number,
    @Discount);
END
END

-- Проверяем, пуста ли таблица Employees
IF NOT EXISTS (SELECT * FROM [dbo].[Employees])
BEGIN
    -- Переменные для генерации реалистичных данных
    DECLARE @CusFirstNames TABLE (Name NVARCHAR(50));
    DECLARE @CusLastNames TABLE (Name NVARCHAR(50));
    INSERT INTO @CusFirstNames (Name) VALUES
        ('John'), ('Jane'), ('Robert'), ('Emily'), ('Michael'), ('Sarah');
    INSERT INTO @CusLastNames (Name) VALUES
        ('Smith'), ('Johnson'), ('Williams'), ('Brown'), ('Davis');

    -- Генерируем случайные данные для сотрудника
    WHILE (SELECT COUNT(*) FROM [dbo].[Employees]) < 50
    BEGIN
        DECLARE @FirstName NVARCHAR(50);
        DECLARE @LastName NVARCHAR(50);
        DECLARE @JobTitle NVARCHAR(50);
        DECLARE @Age INT;

```

```

SELECT TOP 1 @FirstName = Name FROM @CusFirstNames ORDER BY NEWID();
SELECT TOP 1 @LastName = Name FROM @CusLastNames ORDER BY NEWID();
SET @JobTitle = CASE WHEN (SELECT COUNT(*) FROM [dbo].[Employees]) % 5 = 0 THEN 'Manager' ELSE 'Employee' END;
SET @Age = RAND() * 20 + 25; -- Генерируем возраст от 25 до 45 лет

-- Вставляем данные в таблицу
INSERT INTO [dbo].[Employees] ([FIO], [JobTitle], [Age])
VALUES (CONCAT(@FirstName, ' ', @LastName), @JobTitle, @Age);
END
END

-- Проверяем, пуста ли таблица TypesOfRecreation
IF NOT EXISTS (SELECT * FROM [dbo].[TypesOfRecreation])
BEGIN
    -- Заполняем таблицу 50 примерами данных
    INSERT INTO [dbo].[TypesOfRecreation] ([Name], [Description], [Restrictions])
    VALUES
        ('Пляжный отдых', 'Отдых на берегу моря', 'Нет особых ограничений'),
        ('Горнолыжный отдых', 'Катание на горных лыжах', 'Требуется опыт вождения лыж'),
        ('Экскурсии', 'Посещение туристических мест и достопримечательностей', 'Возможны долгие
поездки'),
        ('Активный отдых', 'Спортивные мероприятия и приключения', 'Физическая подготовка
необходима'),
        ('Культурный отдых', 'Посещение музеев и исторических мест', 'Без специальных
ограничений'),
        ('Гастрономический туризм', 'Путешествие для гурманов', 'Возможны ограничения по диете'),
        ('Спа-отдых', 'Посещение спа-центров и процедур', 'Могут потребоваться предварительные
записи'),
        ('Экологический туризм', 'Путешествие по экологически чистым местам', 'Соблюдение
правил охраны природы'),
        ('Круизы', 'Путешествие на круизном лайнере', 'Важно учитывать стихийные бедствия'),
        ('Альпинизм', 'Покорение вершин и скал', 'Высокая физическая подготовка'),
        ('Путешествие с детьми', 'Семейный отдых с детьми', 'Удобства и развлечения для детей'),
        ('Фестивали и мероприятия', 'Посещение фестивалей и культурных событий', 'Сезонность
мероприятий'),
        ('Путешествие в исторические эпохи', 'Посещение исторических реконструкций',
'Соблюдение исторической аутентичности'),
        ('Паломничество', 'Религиозное путешествие к святым местам', 'Духовное вдохновение и
покаяние'),
        ('Треккинг', 'Пеший туризм по тропам и маршрутам', 'Специальное снаряжение и физическая
подготовка'),
        ('Велотуризм', 'Путешествие на велосипеде', 'Безопасность на дороге'),
        ('Дайвинг', 'Подводное плавание и исследование морских глубин', 'Сертификация и
оборудование'),
        ('Сафари', 'Путешествие на африканских сафари', 'Безопасность при встрече с дикой
природой'),
        ('Городские прогулки', 'Ознакомление с историей города', 'Пешеходные маршруты и гиды'),
        ('Путешествие в маленькие деревни', 'Знакомство с местной культурой', 'Простые условия и
общение с местными жителями'),
        ('Водные виды отдыха', 'Активности на воде: каякинг, серфинг и др.', 'Уровень подготовки и
безопасность'),
        ('Гаражный туризм', 'Путешествие на автомобиле или мотоцикле', 'Транспорт и маршруты'),
        ('Экстремальный спорт', 'Путешествие для адреналин-зависимых', 'Экстремальные условия и
безопасность'),
        ('Путешествие в лес', 'Оздоровление и медитация в природе', 'Выживание и безопасность'),
        ('Морская охота', 'Подводная охота и добыча морских биоресурсов', 'Законы и лицензии'),

```

('Археологические раскопки', 'Исследование археологических памятников', 'Соблюдение законов и методологии'),  
 ('Искусство и ремесла', 'Мастер-классы и творческие занятия', 'Творческий потенциал и интересы'),  
 ('Астрономические наблюдения', 'Изучение звезд и планет', 'Телескоп и наблюдательные условия'),  
 ('Походы с палатками', 'Переезды и ночевки в палатках', 'Подготовка и оборудование'),  
 ('Термальные источники', 'Посещение горячих источников и курортов', 'Гигиенические меры и релаксация'),  
 ('Фотосафари', 'Съемка дикой природы и животных', 'Фототехника и охрана природы'),  
 ('Рыбалка', 'Охота на рыбу и ловля', 'Рыболовные удочки и законы'),  
 ('Каякинг', 'Путешествие на байдарках и каяках', 'Безопасность на воде и маршруты'),  
 ('Серфинг', 'Катание на волнах и серфбордах', 'Оборудование и безопасность'),  
 ('Йога и здоровье', 'Занятия йогой и здоровый образ жизни', 'Инструкторы и практика'),  
 ('Лошади и верховая езда', 'Путешествие на лошадях и тренировки', 'Лошади и седла'),  
 ('Походы в горы', 'Покорение вершин и альпинизм', 'Высота и экстремальные условия'),  
 ('Медицинский туризм', 'Путешествие для получения медицинских услуг', 'Медицинская документация'),  
 ('Путешествия на воздушных шарах', 'Воздушные прогулки и экскурсии', 'Безопасность и погодные условия'),  
 ('Семейный отдых', 'Отдых с семьей и детьми', 'Удобства для детей и развлечения'),  
 ('Спортивные мероприятия', 'Посещение спортивных событий и соревнований', 'Билеты и график мероприятий'),  
 ('Зоотуризм', 'Посещение зоопарков и заповедников', 'Экологические меры и безопасность'),  
 ('Архитектурный туризм', 'Исследование архитектурных памятников и стилей', 'Архитектурные особенности'),  
 ('Гастрономические туры', 'Путешествие для гурманов и дегустации', 'Местная кухня и рестораны'),  
 ('Кино и культура', 'Посещение киносъемок и культурных мероприятий', 'Расписание и билеты'),  
 ('Походы на водопады', 'Путешествие к водопадам и каскадам', 'Тропы и маршруты к водопадам'),  
 ('Посещение термальных источников', 'Оздоровление в термальных источниках', 'Температурные режимы и лечебные свойства'),  
 ('Путешествия на морских круизных лайнерах', 'Плавание на круизных кораблях', 'Маршруты и каюты на корабле'),  
 ('Фотографический туризм', 'Съемка пейзажей и архитектуры', 'Фотоаппаратура и композиция'),  
 ('Спелеотуризм', 'Исследование подземных пещер и карстов', 'Снаряжение и опасности в пещерах'),  
 ('Индивидуальные туры', 'Организация индивидуальных маршрутов', 'Персональное обслуживание'),  
 ('Исторические путешествия', 'Посещение исторических мест и памятников', 'Экскурсии и исторические факты'),  
 ('Садоводство и ботанические сады', 'Уход за садами и изучение растений', 'Ботанические особенности');  
 END  
 END

```

-- Проверяем, пуста ли таблица Vouchers
IF NOT EXISTS (SELECT * FROM [dbo].[Vouchers])
BEGIN
    DECLARE @VoucherIndex INT;
    SET @VoucherIndex = 1;

    WHILE @VoucherIndex <= 50
    BEGIN

```

```

DECLARE @StartDate DATE;
DECLARE @ExpirationDate DATE;
DECLARE @HotelId INT;
DECLARE @TypeOfRecreationId INT;
DECLARE @AdditionalServiceId INT;
DECLARE @ClientId INT;
DECLARE @EmployessId INT;
DECLARE @Reservation BIT;
DECLARE @Payment BIT;

-- Генерация случайных значений для записи в таблицу Vouchers
SET @StartDate = DATEADD(DAY, -1 * (ABS(CHECKSUM(NEWID())) % 365), GETDATE());
SET @ExpirationDate = DATEADD(DAY, (ABS(CHECKSUM(NEWID())) % 30) + 1, @StartDate);
SET @HotelId = (ABS(CHECKSUM(NEWID())) % 50) + 1; -- Предполагается, что у вас есть 50 отелей
SET @TypeOfRecreationId = (ABS(CHECKSUM(NEWID())) % 50) + 1; -- Предполагается, что у вас есть 50 типов отдыха
SET @AdditionalServiceId = (ABS(CHECKSUM(NEWID())) % 10) + 1; -- Предполагается, что у вас есть 10 дополнительных
услуг
SET @ClientId = (ABS(CHECKSUM(NEWID())) % 50) + 1; -- Предполагается, что у вас есть 50 клиентов
SET @EmployessId = (ABS(CHECKSUM(NEWID())) % 50) + 1; -- Предполагается, что у вас есть 50 сотрудников
SET @Reservation = CASE WHEN @VoucherIndex % 2 = 0 THEN 1 ELSE 0 END; -- Альтернативно, меняйте значение бита
для резервации
SET @Payment = CASE WHEN @VoucherIndex % 3 = 0 THEN 1 ELSE 0 END; -- Альтернативно, меняйте значение бита для
оплаты

-- Вставляем данные в таблицу Vouchers
INSERT INTO [dbo].[Vouchers] ([StartDate], [ExpirationDate], [HotelId], [TypeOfRecreationId], [AdditionalServiceId],
[ClientId], [EmployessId], [Reservation], [Payment])
VALUES (@StartDate, @ExpirationDate, @HotelId, @TypeOfRecreationId, @AdditionalServiceId, @ClientId, @EmployessId,
@Reservation, @Payment);

SET @VoucherIndex = @VoucherIndex + 1;
END
END

-- Просмотр данных таблиц
GO
CREATE VIEW VoucherView AS
SELECT
    v.[Id] AS [VoucherId],
    v.[StartDate],
    v.[ExpirationDate],
    h.[Name] AS [HotelName],
    c.[FIO] AS [ClientName],
    e.[FIO] AS [EmployeeName],
    tr.[Name] AS [RecreationType],
    s.[Name] AS [AdditionalService],
    v.[Reservation],
    v.[Payment]
FROM [dbo].[Vouchers] v
JOIN [dbo].[Hotels] h ON v.[HotelId] = h.[Id]
JOIN [dbo].[Clients] c ON v.[ClientId] = c.[Id]
JOIN [dbo].[Employees] e ON v.[EmployessId] = e.[Id]
JOIN [dbo].[TypesOfRecreation] tr ON v.[TypeOfRecreationId] = tr.[Id]
JOIN [dbo].[AdditionalServices] s ON v.[AdditionalServiceId] = s.[Id];

-- Создаем представление для таблицы Hotels
GO

```

```

CREATE VIEW HotelView AS
SELECT * FROM [dbo].[Hotels];

-- Представление для таблицы TypesOfRecreation
GO
CREATE VIEW RecreationView AS
SELECT * FROM [dbo].[TypesOfRecreation];

-- Создаем представление с определенными данными
GO
CREATE VIEW EmployeesOver30 AS
SELECT * FROM [dbo].[Employees]
WHERE [Age] > 30;

-- Создание хранимых процедур
IF OBJECT_ID('dbo.InsertHotel', 'P') IS NOT NULL
    DROP PROCEDURE dbo.InsertHotel;
GO

CREATE PROCEDURE InsertHotel
    @Name NVARCHAR(100),
    @Country NVARCHAR(100),
    @City NVARCHAR(100),
    @Address NVARCHAR(200),
    @Phone NVARCHAR(20),
    @Stars INT,
    @TheContactPerson NVARCHAR(100)
AS
BEGIN
    INSERT INTO [dbo].[Hotels] ([Name], [Country], [City], [Address], [Phone], [Stars], [TheContactPerson])
    VALUES (@Name, @Country, @City, @Address, @Phone, @Stars, @TheContactPerson);
END
GO

IF OBJECT_ID('dbo.UpdateClient', 'P') IS NOT NULL
    DROP PROCEDURE dbo.UpdateClient;
GO

CREATE PROCEDURE UpdateClient
    @ClientId INT,
    @FIO NVARCHAR(150),
    @DateOfBirth DATE,
    @Sex NVARCHAR(50),
    @Address NVARCHAR(100),
    @Series NVARCHAR(50),
    @Number BIGINT,
    @Discount BIGINT
AS
BEGIN
    UPDATE [dbo].[Clients]
    SET
        [FIO] = @FIO,
        [DateOfBirth] = @DateOfBirth,
        [Sex] = @Sex,
        [Address] = @Address,
        [Series] = @Series,
        [Number] = @Number,
        [Discount] = @Discount
    WHERE

```



```
                [Id] = @ClientId;

            END

        GO

    IF OBJECT_ID('dbo.DeleteAdditionalServiceById', 'P') IS NOT NULL
        DROP PROCEDURE dbo.DeleteAdditionalServiceById;
    GO

    CREATE PROCEDURE DeleteAdditionalServiceById
        @ServiceId INT
    AS
    BEGIN
        DELETE FROM [dbo].[AdditionalServices]
        WHERE [Id] = @ServiceId;
    END

    GO
```