

**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 и создание базы данных «Гостиница»**

Студент гр. 9308

\_\_\_\_\_

Дубенков С.А.

Преподаватель

\_\_\_\_\_

Новакова Н. Е.

## **АННОТАЦИЯ**

В процессе выполнения индивидуального домашнего задания была спроектирована и создана база данных «Гостиница». При проектировании и создании были использованы знания, полученные в процессе изучения курса «Базы данных». Были созданы и использованы объекты промежуточного слоя (представления, хранимые процедуры, UDF-ы), построена диаграмма базы данных и выбрана стратегия резервного копирования.

## **SUMMARY**

In the process of completing the individual homework, the Hotel database was designed. When designing, the knowledge gained in the course of studying the course "Databases" was used. Middleware objects (views, stored procedures, UDFs) were created and used, a database diagram was built, and a backup strategy was selected.

<b>Оглавление</b>	
<b>Введение .....</b>	<b>4</b>
<b>Цель работы .....</b>	<b>5</b>
<b>Формулировка задания .....</b>	<b>5</b>
<b>Формулировка задачи .....</b>	<b>5</b>
<b>Создание БД.....</b>	<b>5</b>
<b>Создание ограничений целостности.....</b>	<b>6</b>
<b>Диаграмма .....</b>	<b>7</b>
<b>Спецификация базы данных.....</b>	<b>8</b>
<b>Заполнение таблицы данными .....</b>	<b>10</b>
<b>Разработка объектов промежуточного слоя .....</b>	<b>13</b>
Представления .....	13
Хранимые процедуры.....	14
UDF .....	16
<b>Резервное копирование .....</b>	<b>17</b>
<b>Выводы .....</b>	<b>18</b>
<b>Список используемых источников.....</b>	<b>19</b>

## **ВВЕДЕНИЕ**

В данной работе была спроектирована и создана база данных, реализующая работу гостиницы.

База данных включает в себя данные о проживающих клиентах и служащих гостиницы, убирающих в номерах.

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

## **Цель работы**

Закрепить теоретические знания, полученные на курсе «Базы данных», и получить практические навыки в проектировании и создании базы данных. Необходимо создать объекты промежуточного слоя и определить стратегию резервного копирования.

## **Формулировка задания**

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

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

## **Формализация задачи**

Необходимо спроектировать базу данных, содержащую следующие таблицы:

1. Guests – таблица, хранящая в себе информацию о проживающих
2. Rooms – таблица, хранящая в себе информацию о номерах
3. Employees – таблица, хранящая в себе информацию о служащих
4. Schedule – таблица, хранящая в себе информацию о расписании уборки комнат

## **Создание БД**

Создание БД «Гостиница»:

```
create database Hotel
on primary
(name = N'Hotel',
```

```

filename = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Hotel.mdf',
size = 8192 KB, filegrowth = 65536 KB)
log on
(name = N'Hotel_log',
filename = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Hotel_log.ldf',
size = 8192 KB, filegrowth = 65536 KB)
go

```

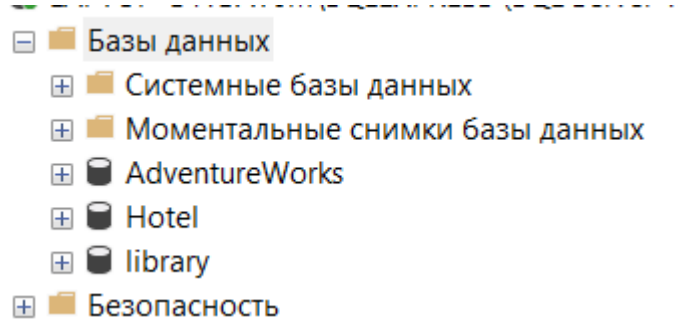


Рисунок 1 – Созданная база данных

## Создание таблиц и ограничений целостности

Создание таблицы Rooms:

```

create table Rooms(
room int identity(1,1) primary key, --autoinc
type int default 1 check(type in (1, 2, 3)),
capacity int
--check(Занятость <= Тип)
default 0 check(capacity in (0, 1, 2, 3)),
price int check(price >= 2000) not null,
telephone varchar(20) not null unique
);

```

Создание таблицы Guests:

```

create table Guests(
passport bigint primary key check(passport > 0),
lastname nvarchar(50) not null,
name nvarchar(50) not null,
city nvarchar(50) not null,
settlement_date date not null,
numberOfPaidDays int not null check(numberOfPaidDays >=
0),
comment nvarchar(100),
room int references Rooms

```

```
);
```

Создание таблицы Employees:

```
create table Employees(  
personnel_number int identity(700000,1) primary key,  
name nvarchar(80) not null,  
dayOfWeek nchar(2),  
status char(1) default '+' check(status in ('+', '-'))  
);
```

Создание таблицы Schedule:

```
create table Schedule(  
personnel_number int references Employees on delete  
cascade,  
room int references Rooms on delete cascade,  
cleaning_day date,  
constraint PK_Schedule primary key(personnel_number,  
room, cleaning_day)  
);
```

На рис. 2 представлены созданные таблицы в базе данных Hotel

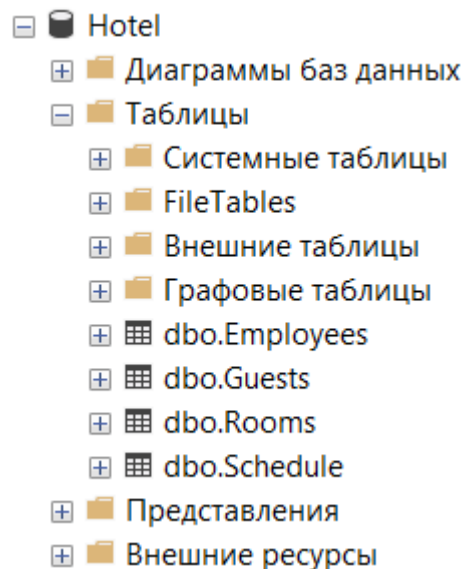


Рисунок 2 – Созданные таблицы

## Диаграмма

На рис. 3 представлена диаграмма связей между таблицами.

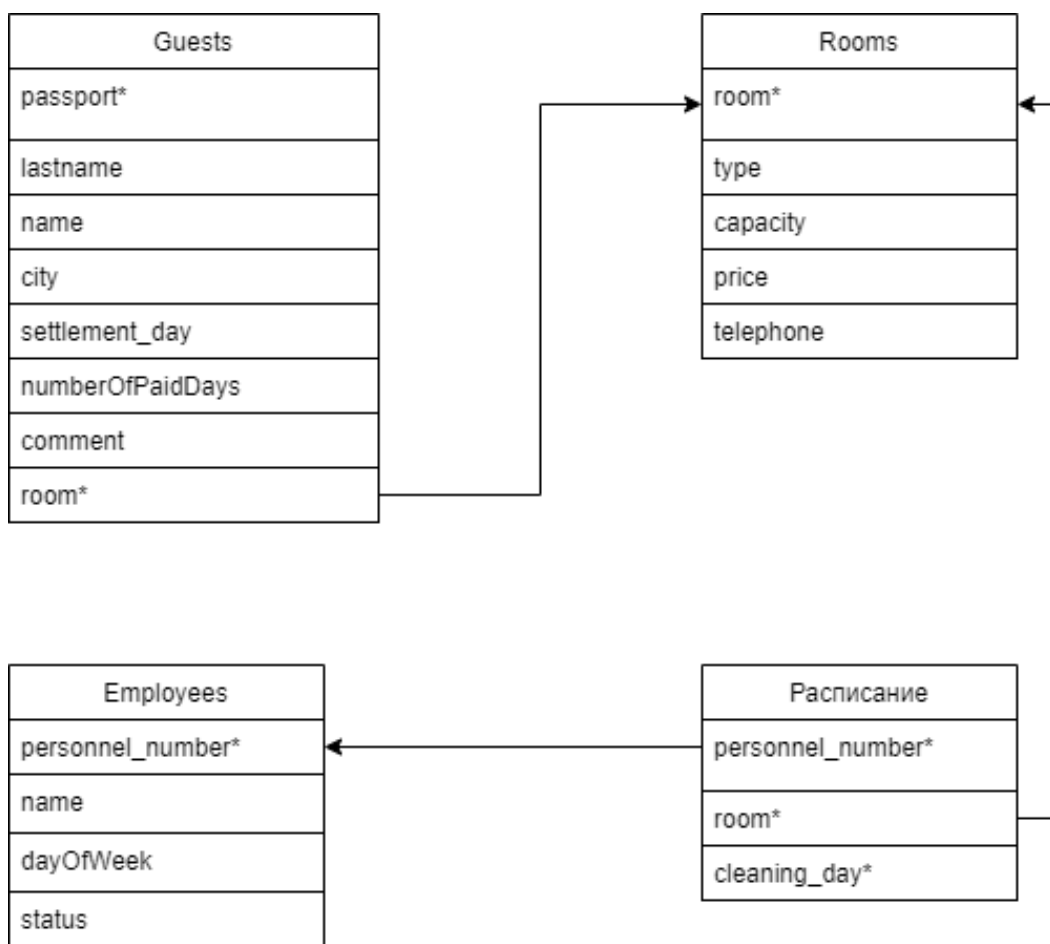


Рисунок 3 – Связи между таблицами

## Спецификация базы данных

Таблица 1 – Спецификация таблицы Rooms

Описание структуры таблицы БД		Наименование таблицы БД: Таблица информации о номерах		Имя таблицы: Rooms	
Дата разработки: 10.11.2021					
Порядковый номер таблицы: 1					
№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничения целостности
1	Номер комнаты	room	int	P	
2	Тип	type	int		default 1, [type] in (1, 2, 3)
3	Занятость	capacity	int		default 0, [capacity] in (0, 1, 2, 3)



4	Стоимость в сутки	price	int		[price] >= 2000, not null
5	Номер телефона	telephone	[varchar](20)		not null, unique

Таблица 2 – Спецификация таблицы Guests

Описание структуры таблицы БД		Наименование таблицы БД: Таблица информации о гостях		Имя таблицы: Guests	
Дата разработки: 10.11.2021					
Порядковый номер таблицы: 2					
№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничения целостности
1	Номер паспорта	passport	bigint	P	[passport] > 0
2	Фамилия	lastname	[nvarchar](50)		not null
3	Имя + иниц.	name	[nvarchar](50)		not null
4	Город	city	[nvarchar](50)		not null
5	Дата поселения	settlement_day	date		not null
6	Кол-во оплаченных дней	numberOfPaidDays	int		not null, [numberOfP aidDays] >= 0
7	Примечание о госте	comment	[nvarchar](100)		
8	Комната	room	int	F	

Таблица 3 – Спецификация таблицы Employees

Описание структуры таблицы БД		Наименование таблицы БД: <b>Таблица информации о служащих</b>	Имя таблицы: Employees
Дата разработки: 10.11.2021			
Порядковый номер таблицы: <b>3</b>			
		Спецификация данных	

№ п/п	Наименование поля	Имя поля	Тип данных	Ключ	Ограничения целостности
1	Табельный номер	personell_number	int	P	
2	Фамилия + имя	name	[nvarchar](80)		not null
3	День недели начала уборки	dayOfWeek	[nchar](2)		
4	Статус служащего (работает или нет)	status	[char](1)		default '+', [status] in ('+', '-')

Таблица 4 – Спецификация таблицы Schedule

Описание структуры таблицы БД		Наименование таблицы БД: <b>Таблица информации о расписании</b>		Имя таблицы: Schedule	
Дата разработки: 10.11.2021					
Порядковый номер таблицы: <b>4</b>					

№ п/п	Наименование поля	Спецификация данных			
		Имя поля	Тип данных	Ключ	Ограничения целостности
1	Табельный номер	personell_number	int	P	
2	Убираемая комната	room	int	P	
3	Дата уборки	cleaning_day	date	P	

### Заполнение таблицы данными

Заполнение таблицы Rooms:

```

insert into Rooms values (3, 2, 6000, '7(904) 895 88-33');
insert into Rooms values (3, 1, 5500, '7(904) 704 36-64');
insert into Rooms values (3, 0, 5000, '7(904) 704 36-65');
insert into Rooms values (2, 2, 4000, '7(904) 704 36-66');
insert into Rooms values (2, 2, 4100, '7(904) 704 36-67');
insert into Rooms values (1, 0, 3000, '7(904) 704 36-68');
insert into Rooms values (1, 1, 3200, '7(904) 704 36-69');
insert into Rooms values (2, 0, 3900, '7(904) 704 36-60');
insert into Rooms values (2, 1, 4500, '7(904) 704 37-64');
insert into Rooms values (3, 1, 5100, '7(904) 704 38-64');
insert into Rooms values (3, 2, 7000, '7(904) 704 39-64');
insert into Rooms values (3, 3, 4300, '7(904) 704 30-64');
insert into Rooms values (3, 3, 6000, '7(904) 705 36-64');

```

Результаты		Сообщения			
	room	type	capacity	price	telephone
1	1	3	2	6000	7(904) 895 88-33
2	2	3	1	5500	7(904) 704 36-64
3	3	3	0	5000	7(904) 704 36-65
4	4	2	2	4000	7(904) 704 36-66
5	5	2	2	4100	7(904) 704 36-67
6	6	1	0	3000	7(904) 704 36-68
7	7	1	1	3200	7(904) 704 36-69
8	8	2	0	3900	7(904) 704 36-60
9	9	2	1	4500	7(904) 704 37-64
10	10	3	1	5100	7(904) 704 38-64
11	11	3	2	7000	7(904) 704 39-64
12	12	3	3	4300	7(904) 704 30-64
13	13	3	3	6000	7(904) 705 36-64

Рисунок 4 – Результат заполнения таблицы Rooms

Заполнение таблицы Guests:

```

insert into Guests values (1215985326, 'Гаврилов', 'Михаил',
'Краснокаменск', '2020-04-15', 6, null, 1);
insert into Guests values (1215985327, 'Чичваркин', 'Антон',
'Красноярск', '2020-04-15', 3, null, 1);
insert into Guests values (1215985356, 'Евдокимов',
'Андрей', 'Москва', '2020-04-15', 2, 'Избил пожилую
женщину', 2);
insert into Guests values (1215985426, 'Губин', 'Евгений',
'Санкт-Петербург', '2020-04-16', 7, null, 4);
insert into Guests values (1215983326, 'Павлова', 'Анна',
'Санкт-Петербург', '2020-04-16', 0, null, 4);
insert into Guests values (1215955326, 'Иванов', 'Иван',
'Екатеринбург', '2020-04-17', 3, 'Разбил вазу, разорвал
шторы', 7);
insert into Guests values (1215885326, 'Иванов', 'Иван',
'Вологда', '2020-04-19', 6, null, 5);
insert into Guests values (1215185326, 'Иванов', 'Иван',
'Вологда', '2020-04-19', 6, null, 5);
insert into Guests values (1215965326, 'Конобеев', 'Илья',
'Владивосток', '2020-04-19', 2, null, 9);
insert into Guests values (1235985326, 'Гаврилова', 'Анна',
'Краснокаменск', '2020-04-20', 2, 'Не оставила чаевых', 10);
insert into Guests values (1515985326, 'Чемоданова',
'Анастасия', 'Владимир', '2020-04-20', 5, null, 11);

```

```
insert into Guests values (1015985326, 'Мельник',
'Ольга', 'Краснокаменск', '2020-04-20', 0, null, 11);
```

	passport	lastname	name	city	settlement_date	numberOfPaidDays	comment	room
1	1015985326	Мельник	Ольга	Краснокаменск	2020-04-20	0	NULL	11
2	1215185326	Иванов	Иван	Вологда	2020-04-19	6	NULL	5
3	1215885326	Иванов	Иван	Вологда	2020-04-19	6	NULL	5
4	1215985326	Иванов	Иван	Екатеринбург	2020-04-17	3	Разбил вазу, разорвал шторы	7
5	1215965326	Конобеев	Илья	Владивосток	2020-04-19	2	NULL	9
6	1215983326	Павлова	Анна	Санкт-Петербург	2020-04-16	0	NULL	4
7	1215985326	Гаврилов	Михаил	Краснокаменск	2020-04-15	6	NULL	1
8	1215985327	Чичваркин	Антон	Красноярск	2020-04-15	3	NULL	1
9	1215985356	Евдокимов	Андрей	Москва	2020-04-15	2	Избил пожилую женщину	2
10	1215985426	Губин	Евгений	Санкт-Петербург	2020-04-16	7	NULL	4
11	1235985326	Гаврилова	Анна	Краснокаменск	2020-04-20	2	Не оставила чаевых	10
12	1515985326	Чемоданова	Анастасия	Владимир	2020-04-20	5	NULL	11

Рисунок 5 – Результат заполнения таблицы Guests

Заполнение таблицы Employees:

```
insert into Employees values('Жмышенко Пётр', 'пт', '+');
insert into Employees values('Андреева Анна', 'пт', '+');
insert into Employees values('Антонов Михаил', 'сб', '+');
insert into Employees values('Василенко Кирилл', 'пн', '-');
insert into Employees values('Придыбайло Кирилл', 'пн', '-');
insert into Employees values('Владимир Ольга', 'вт', '+');
insert into Employees values('Путин Антон', 'ср', '+');
insert into Employees values('Николаенко Константин', 'чт', '+');
insert into Employees values('Епифанцева Елизавета', 'чт', '-');
insert into Employees values('Мельник Николай', 'чт', '+');
```

	personnel_number	name	dayOfWeek	status
1	700000	Жмышенко Пётр	пт	+
2	700001	Андреева Анна	пт	+
3	700002	Антонов Михаил	сб	+
4	700003	Василенко Кирилл	пн	-
5	700004	Придыбайло Кирилл	пн	-
6	700005	Владимир Ольга	вт	+
7	700006	Путин Антон	ср	+
8	700007	Николаенко Константин	чт	+
9	700008	Епифанцева Елизавета	чт	-
10	700009	Мельник Николай	чт	+

Рисунок 6 – Результат заполнения таблицы Employees

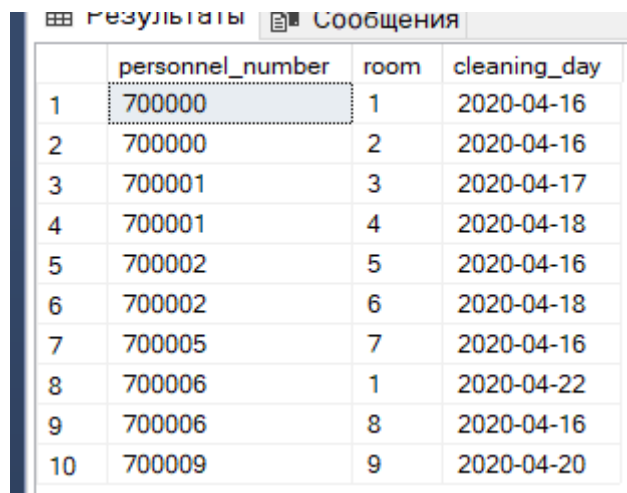
Заполнение таблицы Schedule:

```
insert into Schedule values(700000, 1, '2020-04-16');
insert into Schedule values(700000, 2, '2020-04-16');
```

```

insert into Schedule values(700001, 3, '2020-04-17');
insert into Schedule values(700001, 4, '2020-04-18');
insert into Schedule values(700002, 5, '2020-04-16');
insert into Schedule values(700002, 6, '2020-04-18');
insert into Schedule values(700005, 7, '2020-04-16');
insert into Schedule values(700006, 8, '2020-04-16');
insert into Schedule values(700006, 1, '2020-04-22');
insert into Schedule values(700009, 9, '2020-04-20');

```



	personnel_number	room	cleaning_day
1	700000	1	2020-04-16
2	700000	2	2020-04-16
3	700001	3	2020-04-17
4	700001	4	2020-04-18
5	700002	5	2020-04-16
6	700002	6	2020-04-18
7	700005	7	2020-04-16
8	700006	1	2020-04-22
9	700006	8	2020-04-16
10	700009	9	2020-04-20

Рисунок 7 – Результат заполнения таблицы Schedule

## Разработка объектов промежуточного слоя

### Представления

Создание представление GuestsCommentsView:

```

create view GuestsCommentsView as
select Guests.lastname + ' ' + Guests.name as Name,
Guests.settlement_date as Date,
Guests.numberOfPaidDays as NumberOfPaidDays,
Guests.comment as Comment,
Guests.room as Room
from Guests
where Comment is not null

```

Данное представление выводит информацию о постояльцах, у которых имеются замечания:

100 %					
Результаты		Сообщения			
	Name	Date	NumberOfPaidDays	Comment	Room
1	Иванов Иван	2020-04-17	3	Разбил вазу, разорвал шторы	7
2	Евдокимов Андрей	2020-04-15	2	Избил пожилую женщину	2
3	Гаврилова Анна	2020-04-20	2	Не оставила чаевых	10

Рисунок 8 – Результат создания представления GuestsCommentsView

Создание представления CleaningDaysView:

```
create view CleaningDaysView as
select Employees.personnel_number,
Employees.name,
Schedule.room,
Schedule.cleaning_day as date
from Employees
right join Schedule on
Employees.personnel_number = Schedule.personnel_number
```

Данное представление выводит информацию о работающих служащих: когда и какую комнату они убирают:

100 %				
Результаты		Сообщения		
	personnel_number	name	room	date
1	700000	Жмышенко Пётр	1	2020-04-16
2	700000	Жмышенко Пётр	2	2020-04-16
3	700001	Андреева Анна	3	2020-04-17
4	700001	Андреева Анна	4	2020-04-18
5	700002	Антонов Михаил	5	2020-04-16
6	700002	Антонов Михаил	6	2020-04-18
7	700005	Владимир Ольга	7	2020-04-16
8	700006	Путин Антон	1	2020-04-22
9	700006	Путин Антон	8	2020-04-16
10	700009	Мельник Николай	9	2020-04-20

Рисунок 9 – Результат создания представления CleaningDaysView

## Хранимые процедуры

Создание хранимой процедуры AddRoom. Эта процедура добавляет в таблицу Rooms новые данные. При попытке поставить номер с неправильным кол-вом мест, заполняемостью, ценой или форматом номера телефона, процедура бросит исключение:

```
create procedure AddRoom
```

```

@type int,
@capacity int,
@price int,
@telephone varchar(20) as
begin

if (@type not in (1, 2, 3))
throw 51000, 'Rooms should be single, double or triple
placed', 1;

if (@capacity > @type or @capacity < 0)
throw 51000, 'Capacity shouldn't be larger than type of
room or less than 0', 1;

if (@price < 2000 or @price > 10000)
throw 51000, 'Price should be more 2000 and less than
10000 rubles per day', 1;

if (@telephone not like '7(____) ____ -__')
throw 51000, 'Telephone should have next format: 7(xxx)
yyy zz-zz', 1;

insert into Rooms values(@type, @capacity, @price,
@telephone);

end;

```

Вызов процедуры:

```
exec AddRoom 3, 0, 7500, '7(904) 555 43-13';
```

Результаты		Сообщения			
	room	type	capacity	price	telephone
1	1	3	2	6000	7(904) 895 88-33
2	2	3	1	5500	7(904) 704 36-64
3	3	3	0	5000	7(904) 704 36-65
4	4	2	2	4000	7(904) 704 36-66
5	5	2	2	4100	7(904) 704 36-67
6	6	1	0	3000	7(904) 704 36-68
7	7	1	1	3200	7(904) 704 36-69
8	8	2	0	3900	7(904) 704 36-60
9	9	2	1	4500	7(904) 704 37-64
10	10	3	1	5100	7(904) 704 38-64
11	11	3	2	7000	7(904) 704 39-64
12	12	3	3	4300	7(904) 704 30-64
13	13	3	3	6000	7(904) 705 36-64
14	14	2	0	4300	7(904) 555 43-12
15	15	3	0	7500	7(904) 555 43-13

Рисунок 10 – Результат выполнения процедуры

Создание хранимой процедуры DeleteEmployee. Эта процедура удаляет служащего из таблицы Employee, а его обязанности в расписании удаляются:

```
create procedure DeleteEmployee (@EmployeesId int)
as begin
delete from Employees where personnel_number =
@EmployeesId;
end
```

Вызов хранимой процедуры:

```
exec DeleteEmployee @EmployeesId = 700006
```

	personnel_number	name	dayOfWeek	status
1	700000	Жмышенко Пётр	пт	+
2	700001	Андреева Анна	пт	+
3	700002	Антонов Михаил	сб	+
4	700003	Василенко Кирилл	пн	-
5	700004	Придыбайло Кирилл	пн	-
6	700005	Владимир Ольга	вт	+
7	700008	Епифанцева Елизавета	чт	-
8	700009	Мельник Николай	чт	+

	personnel_number	room	cleaning_day
1	700000	1	2020-04-16
2	700000	2	2020-04-16
3	700001	3	2020-04-17
4	700001	4	2020-04-18
5	700002	5	2020-04-16
6	700002	6	2020-04-18
7	700005	7	2020-04-16
8	700009	9	2020-04-20

Рисунок 11 – Результат выполнения процедуры

## UDF

Создание функции, которая возвращает информацию о постояльцах, которые прибыли в назначенную дату:

```
create function GuestsInfo (@Date date)
returns table
as return
select * from Guests where settlement_date = @Date;
```

Вызов UDF:

```
select * from GuestsInfo('2020-04-19')
```

	passport	lastname	name	city	settlement_date	numberOfPaidDays	comment	room
1	1215185326	Иванов	Иван	Вологда	2020-04-19	6	NULL	5
2	1215885326	Иванов	Иван	Вологда	2020-04-19	6	NULL	5
3	1215965326	Конобеев	Илья	Владивосток	2020-04-19	2	NULL	9

Рисунок 12 – Результат выполнения функции



Создание функции, которая возвращает кол-во номеров, чья цена ниже той, что указано в параметрах:

```
create function NumberOfRooms (@price int)
returns int
as begin
declare @count int;
select @count = count(*)
from Rooms where price > @price;
if (@count is null) set @count = 0;
return @count;
end
```

Вызов UDF:

```
select dbo.NumberOfRooms(5500);
```

### Резервное копирование

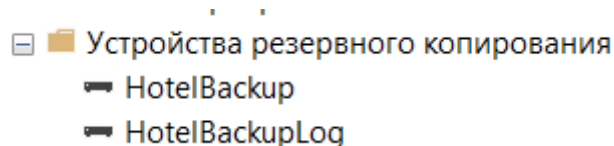


Рисунок 13 – Устройства резервного копирования

Теперь будет произведено полное резервное копирование базы данных Hotel в устройство HotelBackup и резервное копирование журнала в HotelBackupLog.

Для того чтобы копирование прошло успешно, необходимо изменить модель восстановления базы данных на full вместо simple.

Создание резервного копирования:

```
BACKUP DATABASE Hotel
TO DISK = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\Backup\HotelBackup.bak'
WITH INIT
GO
BACKUP LOG Hotel
TO HotelBackupLog;
```

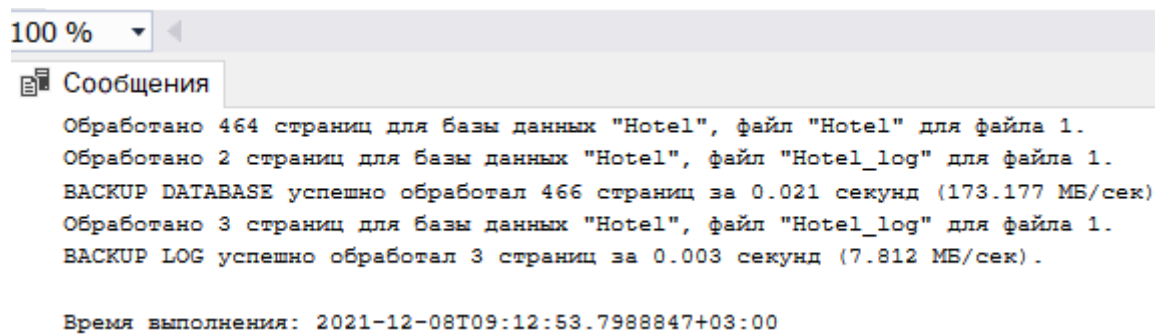


Рисунок 14 – Результат

### Выводы

В результате проделанного индивидуального домашнего задания были приобретены практические навыки проектирования базы данных. Были установлены связи между таблицами, созданы объекты промежуточного слоя, а также установлена стратегия резервного копирования.

## Список используемых источников

1. Дунаев В.В. Базы данных. Язык SQL: [учеб. пособие для студентов и программистов]. – 2-е изд., [доп. и перераб.]. – СПб.: БХВ-Петербург, 2007. – 302 с.: ил. – (Для студента). – ISBN 978-5-9775-0113-2 (в пер.).
2. Дэвид Крёнке Теория и практика построения баз данных – Database processing: монография. – 9-е изд. – СПб.: Питер, 2005. – 859 с.: ил., табл. – (Классика computer science). – Парал. тит. л.: англ. – ISBN 5-94723-583-8 (в пер.).
3. Г. Гарсиа-Молина, Дж. Д. Ульман, Дж. Уидом Системы баз данных. Полный курс – Database systems. The Complete Book: монография [Пер. с англ. и ред. А.С. Варакина]. – М.: Вильямс, 2003. – 1083 с.: ил. – Парал. тит. л.: англ. – ISBN 5-8459-0384-X (в пер.).