Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский экономический университет им. Г.В. Плеханова»

Московский приборостроительный техникум

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

Общепрофессиональная дисциплина: ОП 08 Основы проектирования баз данных

МПТ.09.02.07-П.ОП.08.ОПБД.П50-7-20.10.22

Отчёт к практической работе № 8

«Реализация таблиц и ограничений в базе данных»

Тема: «Разработка базы данных на примере предметной области: ««Ресторан». Реализация

подсистем: учёт и обработка заказов, учёт и реализация меню».

|  |  |
| --- | --- |
| Проверил: | Выполнил: |
| Щаников И.М.\_\_\_\_\_\_\_\_\_\_ | Студент группы |
| (\_\_\_\_\_\_\_\_)\_\_\_\_\_\_\_\_\_\_\_\_\_ | П50-7-20 |
| «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2022г. | \_\_\_\_\_\_\_\_\_\_\_\_Огурцов А.А. |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2022г. |

2022

СОДЕРЖАНИЕ

[1. ЦЕЛИ ПРАКТИЧЕСКОЙ РАБОТЫ 3](#_Toc92804752)

[2. ЭТАПЫ ВЫПОЛНЕНИЯ 4](#_Toc92804753)

[3. ВЫВОД 5](#_Toc92804754)

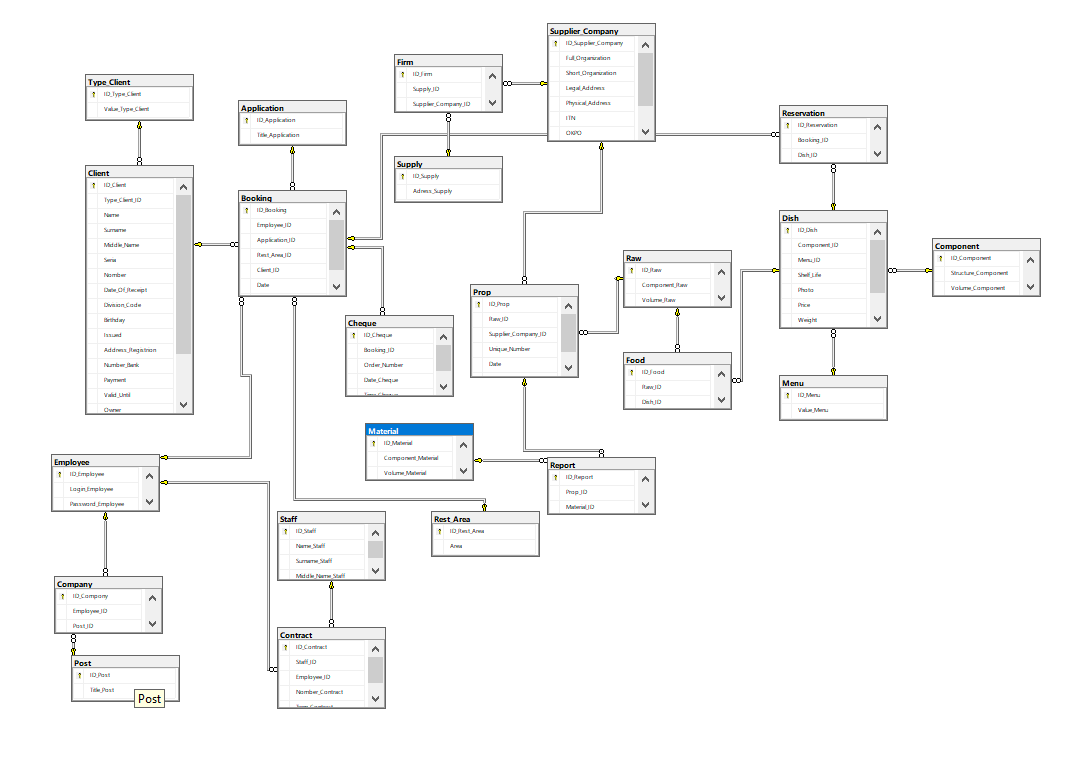
[4. СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ 6](#_Toc92804755)

1. ЦЕЛИ ПРАКТИЧЕСКОЙ РАБОТЫ

На основании даталогической модели данных и словаря данных, разработать структуру таблиц базы данных, на сервере СУБД.

ЭТАПЫ ВЫПОЛНЕНИЯ

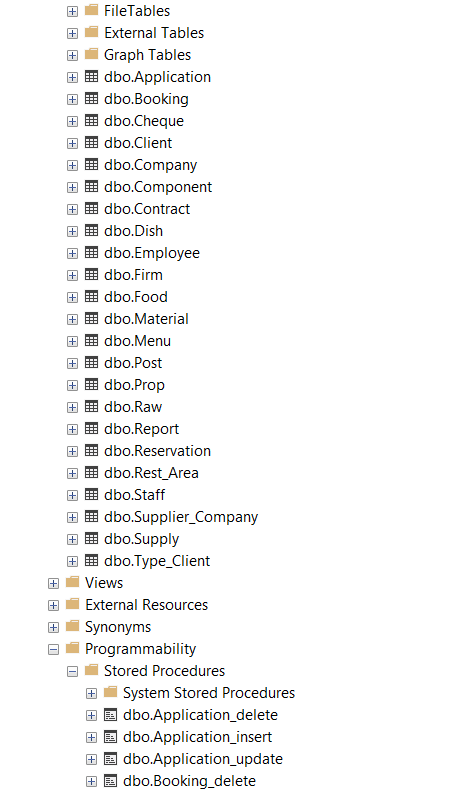
1. Иллюстрация диагнраммы базы данных;

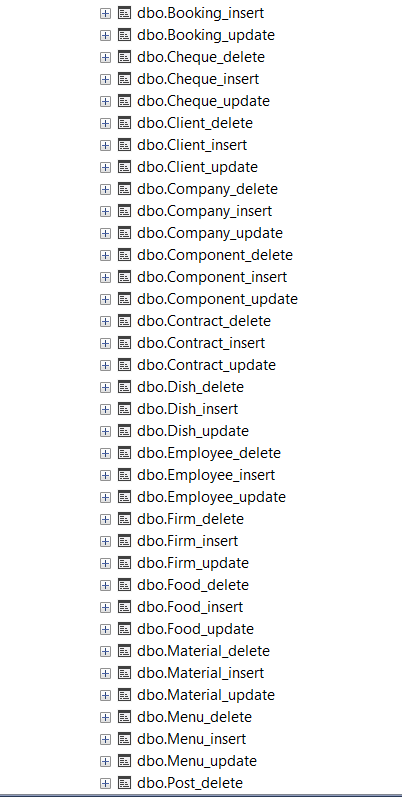


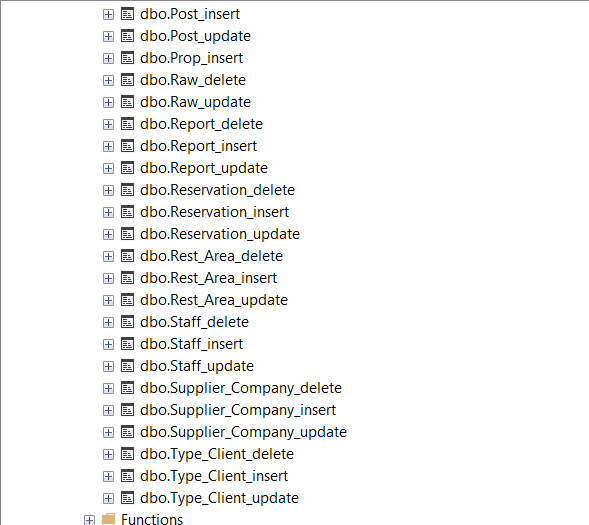
1. К каждой таблице базы данных, необходимо реализовать минимум по 3 – и хранимые процедуры, на добавление, изменение, удаление данных (при этом одна функция может включать в себя сразу несколько действий, к разным таблицам).

|  |  |
| --- | --- |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Type\_Client]  (  [ID\_Type\_Client] [int] not null identity(1,1),  [Value\_Type\_Client] [varchar] (10) not null  constraint [PK\_Type\_Client] primary key clustered  ([ID\_Type\_Client] ASC) on [PRIMARY],  constraint [UQ\_Value\_Type\_Client\_Type\_Client] unique ([Value\_Type\_Client])  )  go | create or alter procedure [dbo].[Type\_Client\_insert]  @Value\_Type\_Client [varchar] (10)  as  begin try  insert into [dbo].[Type\_Client] ([Value\_Type\_Client])  values (@Value\_Type\_Client)  end try  begin catch  print('Данный вид клиента уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Type\_Client\_update]  @ID\_Type\_Client [int], @Value\_Type\_Client [varchar] (10)  as  begin try  update [dbo].[Type\_Client] set  [Value\_Type\_Client] = @Value\_Type\_Client  where  [ID\_Type\_Client] = @ID\_Type\_Client  end try  begin catch  print('Данный вид клиента уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Type\_Client\_delete]  @ID\_Type\_Client [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Client]  where [Type\_Client\_ID] = @ID\_Type\_Client)  if (@any\_child\_record > 0)  print('Тип клиента не может удалён, так как в таблице "Тип клиента", есть связанные данные!');  else  delete from [dbo].[Type\_Client]  where  [ID\_Type\_Client] = @ID\_Type\_Client  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Post]  (  [ID\_Post] [int] not null identity(1,1),  [Title\_Post] [varchar] (32) not null  constraint [PK\_Post] primary key clustered  ([ID\_Post] ASC) on [PRIMARY],  constraint [UQ\_Title\_Post\_Post] unique ([Title\_Post])  )  go | create or alter procedure [dbo].[Post\_insert]  @Title\_Post [varchar] (32)  as  begin try  insert into [dbo].[Post] ([Title\_Post])  values (@Title\_Post)  end try  begin catch  print('Данная должность уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Post\_update]  @ID\_Post [int], @Title\_Post [varchar] (32)  as  begin try  update [dbo].[Post] set  [Title\_Post] = @Title\_Post  where  [ID\_Post] = @ID\_Post  end try  begin catch  print('Данная должность уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Post\_delete]  @ID\_Post [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Company]  where [Post\_ID] = @ID\_Post)  if (@any\_child\_record > 0)  print('Должность не может быть удалена удалён, так как в таблице "Должность", есть связанные данные!');  else  delete from [dbo].[Post]  where  [ID\_Post] = @ID\_Post  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Menu]  (  [ID\_Menu] [int] not null identity(1,1),  [Value\_Menu] [varchar] (max) not null  constraint [PK\_Menu] primary key clustered  ([ID\_Menu] ASC) on [PRIMARY]  )  go | create or alter procedure [dbo].[Menu\_insert]  @Value\_Menu [varchar] (max)  as  begin try  insert into [dbo].[Menu] ([Value\_Menu])  values (@Value\_Menu)  end try  begin catch  print('Данное название блюда уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Menu\_update]  @ID\_Menu [int], @Value\_Menu [varchar] (max)  as  begin try  update [dbo].[Menu] set  [Value\_Menu] = @Value\_Menu  where  [ID\_Menu] = @ID\_Menu  end try  begin catch  print('Данное название блюда уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Menu\_delete]  @ID\_Menu [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Dish]  where [Menu\_ID] = @ID\_Menu)  if (@any\_child\_record > 0)  print('Название блюда не может быть удалена удалён, так как в таблице "Меню", есть связанные данные!');  else  delete from [dbo].[Menu]  where  [ID\_Menu] = @ID\_Menu  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Component]  (  [ID\_Component] [int] not null identity(1,1),  [Structure\_Component] [varchar] (32) not null,  [Volume\_Component] [int] not null  constraint [PK\_Component] primary key clustered  ([ID\_Component] ASC) on [PRIMARY],  constraint [UQ\_Structure\_Component\_Component] unique ([Structure\_Component]),  constraint [CH\_Volume\_Component\_Component] check ([Volume\_Component] > 0)  )  go | create or alter procedure [dbo].[Component\_insert]  @Structure\_Component [varchar] (32),@Volume\_Component [int]  as  begin try  insert into [dbo].[Component] ([Structure\_Component], [Volume\_Component])  values (@Structure\_Component, @Volume\_Component)  end try  begin catch  print('Данный компонент уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Component\_update]  @ID\_Component [int],@Structure\_Component [varchar] (32),@Volume\_Component [int]  as  begin try  update [dbo].[Component] set  [Structure\_Component] = @Structure\_Component,  [Volume\_Component] = @Volume\_Component  where  [ID\_Component] = @ID\_Component  end try  begin catch  print('Данный компонент уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Component\_delete]  @ID\_Component [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Dish]  where [Component\_ID] = @ID\_Component)  if (@any\_child\_record > 0)  print('Компонент не может быть удалена удалён, так как в таблице "Компонент", есть связанные данные!');  else  delete from [dbo].[Component]  where  [ID\_Component] = @ID\_Component  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Raw]  (  [ID\_Raw] [int] not null identity(1,1),  [Component\_Raw] [varchar] (32) not null,  [Volume\_Raw] [int] not null  constraint [PK\_Raw] primary key clustered  ([ID\_Raw] ASC) on [PRIMARY],  constraint [UQ\_Component\_Raw\_Raw] unique ([Component\_Raw]),  constraint [CH\_Volume\_Raw\_Raw] check ([Volume\_Raw] > 0)  )  go | create or alter procedure [dbo].[Raw\_insert]  @Component\_Raw [varchar] (32),@Volume\_Raw [int]  as  begin try  insert into [dbo].[Raw] ([Component\_Raw], [Volume\_Raw])  values (@Component\_Raw, @Volume\_Raw)  end try  begin catch  print('Данное сырьё уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Raw\_update]  @ID\_Raw [int],@Component\_Raw [varchar] (32),@Volume\_Raw [int]  as  begin try  update [dbo].[Raw] set  [Component\_Raw] = @Component\_Raw,  [Volume\_Raw] = @Volume\_Raw  where  [ID\_Raw] = @ID\_Raw  end try  begin catch  print('Данное сырьё уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Raw\_delete]  @ID\_Raw [int]  as  declare @any\_child\_record1 [int] = (select count(\*) from [dbo].[Prop]  where [Raw\_ID] = @ID\_Raw)  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Food]  where [Raw\_ID] = @ID\_Raw)  if (@any\_child\_record >0 or @any\_child\_record1 > 0)  print('Компонент не может быть удалена удалён, так как в таблице "Компонент", есть связанные данные!');  else  delete from [dbo].[Raw]  where  [ID\_Raw] = @ID\_Raw  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Material]  (  [ID\_Material] [int] not null identity(1,1),  [Component\_Material] [varchar] (32) not null,  [Volume\_Material] [int] not null  constraint [PK\_Material] primary key clustered  ([ID\_Material] ASC) on [PRIMARY],  constraint [UQ\_Component\_Material\_Material] unique ([Component\_Material]),  constraint [CH\_Volume\_Material\_Material] check ([Volume\_Material] > 0)  )  go | create or alter procedure [dbo].[Material\_insert]  @Component\_Material [varchar] (32),@Volume\_Material [int]  as  begin try  insert into [dbo].[Material] ([Component\_Material], [Volume\_Material])  values (@Component\_Material, @Volume\_Material)  end try  begin catch  print('Данный материал уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Material\_update]  @ID\_Material [int],@Component\_Material [varchar] (32),@Volume\_Material [int]  as  begin try  update [dbo].[Material] set  [Component\_Material] = @Component\_Material,  [Volume\_Material] = @Volume\_Material  where  [ID\_Material] = @ID\_Material  end try  begin catch  print('Данный материал уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Material\_delete]  @ID\_Material [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Report]  where [Material\_ID] = @ID\_Material)  if (@any\_child\_record > 0)  print('Материал не может быть удалена удалён, так как в таблице "Компонент", есть связанные данные!');  else  delete from [dbo].[Material]  where  [ID\_Material] = @ID\_Material  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Application]  (  [ID\_Application] [int] not null identity(1,1),  [Title\_Application] [varchar] (25) not null  constraint [PK\_Application] primary key clustered  ([ID\_Application] ASC) on [PRIMARY],  constraint [UQ\_Title\_Application\_Application] unique ([Title\_Application])  )  go | create or alter procedure [dbo].[Application\_insert]  @Title\_Application [varchar] (25)  as  begin try  insert into [dbo].[Application] ([Title\_Application])  values (@Title\_Application)  end try  begin catch  print('Данная заявка уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Application\_update]  @ID\_Application [int],@Title\_Application [varchar] (25)  as  begin try  update [dbo].[Application] set  [Title\_Application] = @Title\_Application  where  [ID\_Application] = @ID\_Application  end try  begin catch  print('Данная заявка уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Application\_delete]  @ID\_Application [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Booking]  where [Application\_ID] = @ID\_Application)  if (@any\_child\_record > 0)  print('Заявка не может быть удалена удалён, так как в таблице "Компонент", есть связанные данные!');  else  delete from [dbo].[Application]  where  [ID\_Application] = @ID\_Application  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Employee]  (  [ID\_Employee] [int] not null identity(1,1),  [Login\_Employee] [varchar] (32) not null,  [Password\_Employee] [varchar] (32) not null  constraint [PK\_Employee] primary key clustered  ([ID\_Employee] ASC) on [PRIMARY],  constraint [UQ\_Login\_Employee\_Employee] unique ([Login\_Employee]),  constraint [CH\_Login\_Employee\_Employee] check (len([Login\_Employee])>=8),  constraint [CH\_Password\_Employee\_Employee\_Upper] check ([Password\_Employee] like ('%[A-Z]%')),  constraint [CH\_Password\_Employee\_Employee\_Letter\_Lower] check ([Password\_Employee] like ('%[a-z]%')),  constraint [CH\_Password\_Employee\_Employee\_Symbols] check ([Password\_Employee] like ('%[!@#$%^&\*()]%'))  )  go | create or alter procedure [dbo].[Employee\_insert]  @Login\_Employee [varchar] (32),@Password\_Employee [varchar] (32)  as  begin try  insert into [dbo].[Employee] ([Login\_Employee], [Password\_Employee])  values (@Login\_Employee, @Password\_Employee)  end try  begin catch  print('Данный Работник уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Employee\_update]  @ID\_Employee [int],@Login\_Employee [varchar] (32),@Password\_Employee [varchar] (32)  as  begin try  update [dbo].[Employee] set  [Login\_Employee] = @Login\_Employee,  [Password\_Employee] = @Password\_Employee  where  [ID\_Employee] = @ID\_Employee  end try  begin catch  print('Данный Работник уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Employee\_delete]  @ID\_Employee [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Booking]  where [Employee\_ID] = @ID\_Employee)  declare @any\_child\_record1 [int] = (select count(\*) from [dbo].[Contract]  where [Employee\_ID] = @ID\_Employee)  declare @any\_child\_record2 [int] = (select count(\*) from [dbo].[Company]  where [Employee\_ID] = @ID\_Employee)  if (@any\_child\_record > 0 or @any\_child\_record1 > 0 or @any\_child\_record2 > 0)  print('Работник не может быть удалён, так как в таблице "Работник", есть связанные данные!');  else  delete from [dbo].[Employee]  where  [ID\_Employee] = @ID\_Employee  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Staff]  (  [ID\_Staff] [int] not null identity(1,1),  [Name\_Staff] [varchar] (30) not null,  [Surname\_Staff] [varchar] (30) not null,  [Middle\_Name\_Staff] [varchar] (30) not null  constraint [PK\_Staff] primary key clustered  ([ID\_Staff] ASC) on [PRIMARY]  )  go | create or alter procedure [dbo].[Staff\_insert]  @Name\_Staff [varchar] (30),@Surname\_Staff [varchar] (30)  as  begin try  insert into [dbo].[Staff] ([Name\_Staff], [Surname\_Staff])  values (@Name\_Staff, @Surname\_Staff)  end try  begin catch  print('Данный Персонал уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Staff\_update]  @ID\_Staff [int],@Name\_Staff [varchar] (32),@Surname\_Staff [varchar] (32)  as  begin try  update [dbo].[Staff] set  [Name\_Staff] = @Name\_Staff,  [Surname\_Staff] = @Surname\_Staff  where  [ID\_Staff] = @ID\_Staff  end try  begin catch  print('Данный Персонал уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Staff\_delete]  @ID\_Staff [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Contract]  where [Staff\_ID] = @ID\_Staff)  if (@any\_child\_record > 0)  print('Персонал не может быть удалён, так как в таблице "Персонал", есть связанные данные!');  else  delete from [dbo].[Staff]  where  [ID\_Staff] = @ID\_Staff  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Rest\_Area]  (  [ID\_Rest\_Area] [int] not null identity(1,1),  [Area] [varchar] (30) not null  constraint [PK\_Rest\_Area] primary key clustered  ([ID\_Rest\_Area] ASC) on [PRIMARY],  constraint [UQ\_Area\_Rest\_Area] unique ([Area])  )  go | create or alter procedure [dbo].[Rest\_Area\_insert]  @Area [varchar] (30)  as  begin try  insert into [dbo].[Rest\_Area] ([Area])  values (@Area)  end try  begin catch  print('Данная зона уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Rest\_Area\_update]  @ID\_Rest\_Area [int],@Area [varchar] (30)  as  begin try  update [dbo].[Rest\_Area] set  [Area] = @Area  where  [ID\_Rest\_Area] = @ID\_Rest\_Area  end try  begin catch  print('Данная зона уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Rest\_Area\_delete]  @ID\_Rest\_Area [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Booking]  where [Rest\_Area\_ID] = @ID\_Rest\_Area)  if (@any\_child\_record > 0)  print('Зона не может быть удалён, так как в таблице "Персонал", есть связанные данные!');  else  delete from [dbo].[Rest\_Area]  where  [ID\_Rest\_Area] = @ID\_Rest\_Area  go |
| Скрипт таблицы | Хранимая процедура на удаление данных |
| create table [dbo].[Supplier\_Company]  (  [ID\_Supplier\_Company] [int] not null identity(1,1),  [Full\_Organization] [varchar] (32) not null,  [Short\_Organization] [varchar] (32) not null,  [Legal\_Address] [varchar] (max) not null,  [Physical\_Address] [varchar] (max) not null,  [ITN] [varchar] (12) not null,  [OKPO] [varchar] (8) not null,  [BIC] [varchar] (9) not null  constraint [PK\_Supplier\_Company] primary key clustered  ([ID\_Supplier\_Company] ASC) on [PRIMARY],  constraint [UQ\_Full\_Organization\_Supplier\_Company] unique ([Full\_Organization]),  constraint [CH\_ITN\_Supplier\_Company] check ([ITN] like ('[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')),  constraint [CH\_OKPO\_Supplier\_Company] check ([OKPO] like ('[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')),  constraint [CH\_BIC\_Supplier\_Company] check ([BIC] like ('[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))  )  go | create or alter procedure [dbo].[Supplier\_Company\_insert]  @Full\_Organization [varchar] (32),@Short\_Organization [varchar] (32),  @Legal\_Address [varchar] (max),@Physical\_Address [varchar] (max),@ITN [varchar] (12),  @OKPO [varchar] (8),@BIC [varchar] (9)  as  begin try  insert into [dbo].[Supplier\_Company] ([Full\_Organization], [Short\_Organization],[Legal\_Address], [Physical\_Address], [ITN], [OKPO], [BIC])  values (@Full\_Organization,@Short\_Organization,@Legal\_Address,@Physical\_Address,@ITN,@OKPO,@BIC)  end try  begin catch  print('Данная компания уже есть в таблице!')  end catch  go |
| Хранимая процедура на добавление данных |
| create or alter procedure [dbo].[Supplier\_Company\_update]  @ID\_Supplier\_Company [int],@Full\_Organization [varchar] (32),@Short\_Organization [varchar] (32),  @Legal\_Address [varchar] (max),@Physical\_Address [varchar] (max),@ITN [varchar] (12),  @OKPO [varchar] (8),@BIC [varchar] (9)  as  begin try  update [dbo].[Supplier\_Company] set  [Full\_Organization] = @Full\_Organization,  [Short\_Organization] = @Short\_Organization,  [Legal\_Address] = @Legal\_Address,  [Physical\_Address] = @Physical\_Address,  [ITN] = @ITN,  [OKPO] = @OKPO,  [BIC] = @BIC  where  [ID\_Supplier\_Company] = @ID\_Supplier\_Company  end try  begin catch  print('Данная компания уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Supplier\_Company\_delete]  @ID\_Supplier\_Company [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Prop]  where [Supplier\_Company\_ID] = @ID\_Supplier\_Company)  declare @any\_child\_record1 [int] = (select count(\*) from [dbo].[Firm]  where [Supplier\_Company\_ID] = @ID\_Supplier\_Company)  if (@any\_child\_record > 0 or @any\_child\_record > 0)  print('Компания не может быть удалён, так как в таблице "Компания", есть связанные данные!');  else  delete from [dbo].[Supplier\_Company]  where  [ID\_Supplier\_Company] = @ID\_Supplier\_Company  go |
| Скрипт таблицы | Хранимая процедура на удаление данных |
| create table [dbo].[Client]  (  [ID\_Client] [int] not null identity(1,1),  [Type\_Client\_ID] [INT] not null,  [Name] [VARCHAR] (30) not null,  [Surname] [VARCHAR] (30) not null,  [Middle\_Name] [VARCHAR] (30) not null,  [Seria] [VARCHAR] (4) not null,  [Nomber] [VARCHAR] (6) not null,  [Date\_Of\_Receipt] [DATE] not null,  [Division\_Code] [VARCHAR] (7) not null,  [Birthday] [DATE] not null,  [Issued] [VARCHAR] (MAX) not null,  [Address\_Registrion] [VARCHAR] (MAX) not null,  [Number\_Bank] [VARCHAR] (16) not null,  [Payment] [VARCHAR] (30) not null,  [Valid\_Until] [VARCHAR] (5) not null,  [Owner] [VARCHAR] (MAX) not null,  [CVC] [VARCHAR] (3) not null,  [Login] [VARCHAR] (32) not null,  [Password] [VARCHAR] (32) not null  constraint [PK\_Client] primary key clustered  ([ID\_Client] ASC) on [PRIMARY],  constraint [UQ\_Login\_Client] unique ([Login]),  constraint [UQ\_Number\_Bank\_Client] unique ([Number\_Bank]),  constraint [CH\_Valid\_Until\_Client] check ([Valid\_Until] like '[0-9][0-9][/][0-9][0-9]'),  constraint [CH\_Login\_Client] check (len([Login])>=8),  constraint [CH\_Password\_Client\_Upper] check ([Password] like ('%[A-Z]%')),  constraint [CH\_Password\_Client\_Letter\_Lower] check ([Password] like ('%[a-z]%')),  constraint [CH\_Password\_Client\_Symbols] check ([Password] like ('%[!@#$%^&\*()\_-]%')),  constraint [CH\_CVC\_Client\_INT] check ([CVC] like '[0-9][0-9][0-9]'),  constraint [CH\_Number\_Bank\_Client\_INT] check ([Number\_Bank] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Seria\_Client\_INT] check ([Seria] like '[0-9][0-9][0-9][0-9]'),  constraint [CH\_Division\_Code\_Client] check ([Division\_Code] like '[0-9][0-9][0-9][-][0-9][0-9][0-9]'),  constraint [CH\_Nomber\_Client\_INT] check ([Nomber] like '[0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [FK\_Type\_Client] foreign key ([Type\_Client\_ID]) references [dbo].[Type\_Client] ([ID\_Type\_Client]),  constraint [UQ\_Seria\_Nomber\_Client] unique ([Nomber], [Seria]),  constraint [CH\_Birthday\_Client] check ([Birthday] < Getdate())  )  go | create or alter procedure [dbo].[Client\_insert]  @Type\_Client\_ID [INT],@Name [VARCHAR] (30),@Surname [VARCHAR] (30),  @Middle\_Name [VARCHAR] (30),@Seria [VARCHAR] (4),@Nomber [VARCHAR] (6),@Date\_Of\_Receipt [DATE],  @Division\_Code [VARCHAR] (7),@Birthday [DATE],@Issued [VARCHAR] (MAX),  @Address\_Registrion [VARCHAR] (MAX),@Number\_Bank [VARCHAR] (16),@Payment [VARCHAR] (30),  @Valid\_Until [VARCHAR] (5),@Owner [VARCHAR] (MAX),@CVC [VARCHAR] (3),@Login [VARCHAR] (32),  @Password [VARCHAR] (32)  as  begin try  insert into [dbo].[Client] ([Type\_Client\_ID], [Surname], [Name], [Middle\_Name], [Login], [Password], [Seria], [Nomber], [Date\_Of\_Receipt], [Division\_Code], [Birthday], [Issued], [Address\_Registrion], [Number\_Bank], [Payment], [Valid\_Until], [Owner], [CVC])  values (@Type\_Client\_ID,@Name,@Surname,@Middle\_Name,@Seria,@Nomber,@Date\_Of\_Receipt,  @Division\_Code,@Birthday,@Issued,@Address\_Registrion,@Number\_Bank,@Payment,@Valid\_Until,  @Owner,@CVC,@Login,@Password)  end try  begin catch  print('Данный клиент уже есть в таблице!')  end catch  go |
| Хранимая процедура на добавление данных |
| create or alter procedure [dbo].[Client\_update]  @ID\_Client [int],@Type\_Client\_ID [INT],@Name [VARCHAR] (30),@Surname [VARCHAR] (30),  @Middle\_Name [VARCHAR] (30),@Seria [VARCHAR] (4),@Nomber [VARCHAR] (6),@Date\_Of\_Receipt [DATE],  @Division\_Code [VARCHAR] (7),@Birthday [DATE],@Issued [VARCHAR] (MAX),  @Address\_Registrion [VARCHAR] (MAX),@Number\_Bank [VARCHAR] (16),@Payment [VARCHAR] (30),  @Valid\_Until [VARCHAR] (5),@Owner [VARCHAR] (MAX),@CVC [VARCHAR] (3),@Login [VARCHAR] (32),  @Password [VARCHAR] (32)  as  begin try  update [dbo].[Client] set  [Type\_Client\_ID] = @Type\_Client\_ID,  [Name] = @Name,  [Surname] = @Surname,  [Middle\_Name] = @Middle\_Name,  [Seria] = @Seria,  [Nomber] = @Nomber,  [Date\_Of\_Receipt] = @Date\_Of\_Receipt,  [Division\_Code] = @Division\_Code,  [Birthday] = @Birthday,  [Issued] = @Issued,  [Address\_Registrion] = @Address\_Registrion,  [Number\_Bank] = @Number\_Bank,  [Payment] = @Payment,  [Valid\_Until] = @Valid\_Until,  [Owner] = @Owner,  [CVC] = @CVC,  [Login] = @Login,  [Password] = @Password  where  [ID\_Client] = @ID\_Client  end try  begin catch  print('Данный клиент уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Client\_delete]  @ID\_Client [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Booking]  where [Client\_ID] = @ID\_Client)  if (@any\_child\_record > 0)  print('Зона не может быть удалён, так как в таблице "Персонал", есть связанные данные!');  else  delete from [dbo].[Client]  where  [ID\_Client] = @ID\_Client  Go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Dish]  (  [ID\_Dish] [int] not null identity(1,1),  [Component\_ID] [int] not null,  [Menu\_ID] [int] not null,  [Shelf\_Life] [int] not null,  [Photo] [image] not null,  [Price] [int] not null,  [Weight] [int] not null,  [Short\_Recital] [varchar] (max) not null  constraint [PK\_Dish] primary key clustered  ([ID\_Dish] ASC) on [PRIMARY],  constraint [FK\_Component] foreign key ([Component\_ID])  references [dbo].[Component] ([ID\_Component]),  constraint [FK\_Menu] foreign key ([Menu\_ID])  references [dbo].[Menu] ([ID\_Menu]),  constraint [CH\_Shelf\_Life\_Dish] check ([Shelf\_Life] > 0)  )  go | create or alter procedure [dbo].[Dish\_insert]  @Component\_ID [int], @Menu\_ID [int], @Shelf\_Life [int], @Photo [image], @Price [int], @Weight [int], @Short\_Recital [varchar] (max)  as  begin try  insert into [dbo].[Dish] ([Component\_ID], [Menu\_ID], [Shelf\_Life], [Photo], [Price], [Weight], [Short\_Recital])  values (@Component\_ID,@Menu\_ID,@Shelf\_Life,@Photo,@Price,@Weight,@Short\_Recital)  end try  begin catch  print('Данное блюдо уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Dish\_update]  @ID\_Dish [int],@Component\_ID [int], @Menu\_ID [int], @Shelf\_Life [int], @Photo [image], @Price [int], @Weight [int], @Short\_Recital [varchar] (max)  as  begin try  update [dbo].[Dish] set  [Component\_ID] = @Component\_ID,  [Menu\_ID] = @Menu\_ID,  [Shelf\_Life] = @Shelf\_Life,  [Photo] = @Photo,  [Price] = @Price,  [Weight] = @Weight,  [Short\_Recital] = @Short\_Recital  where  [ID\_Dish] = @ID\_Dish  end try  begin catch  print('Данное блюдо уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Dish\_delete]  @ID\_Dish [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Reservation]  where [Dish\_ID] = @ID\_Dish)  declare @any\_child\_record1 [int] = (select count(\*) from [dbo].[Food]  where [Dish\_ID] = @ID\_Dish)  if (@any\_child\_record > 0 or @any\_child\_record > 0)  print('Блюдо не может быть удалён, так как в таблице "Блюдо", есть связанные данные!');  else  delete from [dbo].[Dish]  where  [ID\_Dish] = @ID\_Dish  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Contract]  (  [ID\_Contract] [int] not null identity(1,1),  [Staff\_ID] [int] not null,  [Employee\_ID] [int] not null,  [Nomber\_Contract] [VARCHAR] (10) not null,  [Term\_Contract] [int] not null  constraint [PK\_Contract] primary key clustered  ([ID\_Contract] ASC) on [PRIMARY],  constraint [FK\_Staff] foreign key ([Staff\_ID])  references [dbo].[Staff] ([ID\_Staff]),  constraint [FK\_Employee\_Contract] foreign key ([Employee\_ID])  references [dbo].[Employee] ([ID\_Employee]),  constraint [CH\_Nomber\_Contract] check ([Nomber\_Contract] like '[0-9][0-9][0-9][0-9][0-9][0-9][-][С][Г][П]'),  constraint [UQ\_Nomber\_Contract] unique ([Nomber\_Contract]),  constraint [CH\_Term\_Contract] check ([Term\_Contract] > 0)  )  go | create or alter procedure [dbo].[Contract\_insert]  @Staff\_ID [int],@Employee\_ID [int],@Nomber\_Contract [VARCHAR] (10),@Term\_Contract [int]  as  begin try  insert into [dbo].[Contract] ([Staff\_ID], [Employee\_ID],[Nomber\_Contract],[Term\_Contract])  values (@Staff\_ID,@Employee\_ID,@Nomber\_Contract,@Term\_Contract)  end try  begin catch  print('Данный контракт уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Contract\_update]  @ID\_Contract [int],@Staff\_ID [int],@Employee\_ID [int],@Nomber\_Contract [VARCHAR] (10),@Term\_Contract [int]  as  begin try  update [dbo].[Contract] set  [Staff\_ID] = @Staff\_ID,  [Employee\_ID] = @Employee\_ID,  [Nomber\_Contract] = @Nomber\_Contract,  [Term\_Contract] = @Term\_Contract  where  [ID\_Contract] = @ID\_Contract  end try  begin catch  print('Данный контракт уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Contract\_delete]  @ID\_Contract [int]  as  delete from [dbo].[Contract]  where  [ID\_Contract] = @ID\_Contract  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Company]  (  [ID\_Compony] [int] not null identity(1,1),  [Employee\_ID] [int] not null,  [Post\_ID] [int] not null  constraint [PK\_Company] primary key clustered  ([ID\_Compony] ASC) on [PRIMARY],  constraint [FK\_Post] foreign key ([Post\_ID])  references [dbo].[Post] ([ID\_Post]),  constraint [FK\_Employee\_Company] foreign key ([Employee\_ID])  references [dbo].[Employee] ([ID\_Employee]),  )  go | create or alter procedure [dbo].[Company\_insert]  @Employee\_ID [int],@Post\_ID [int]  as  begin try  insert into [dbo].[Company] ([Employee\_ID], [Post\_ID])  values (@Employee\_ID,@Post\_ID)  end try  begin catch  print('Данная компания уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Company\_update]  @ID\_Company [int],@Employee\_ID [int],@Post\_ID [int]  as  begin try  update [dbo].[Company] set  [Employee\_ID] = @Employee\_ID,  [Post\_ID] = @Employee\_ID  where  [ID\_Compony] = @ID\_Company  end try  begin catch  print('Данная компания уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Company\_delete]  @ID\_Company [int]  as  delete from [dbo].[Company]  where  [ID\_Compony] = @ID\_Company  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Booking]  (  [ID\_Booking] [int] not null identity(1,1),  [Employee\_ID] [int] not null,  [Application\_ID] [int] not null,  [Rest\_Area\_ID] [int] not null,  [Client\_ID] [int] not null,  [Date] [date] not null,  [Time] [time] not null  constraint [PK\_Booking] primary key clustered  ([ID\_Booking] ASC) on [PRIMARY],  constraint [FK\_Employee\_Booking] foreign key ([Employee\_ID])  references [dbo].[Employee] ([ID\_Employee]),  constraint [FK\_Application\_Booking] foreign key ([Application\_ID])  references [dbo].[Application] ([ID\_Application]),  constraint [FK\_Rest\_Area\_Booking] foreign key ([Rest\_Area\_ID])  references [dbo].[Rest\_Area] ([ID\_Rest\_Area]),  constraint [FK\_Client\_Booking] foreign key ([Client\_ID])  references [dbo].[Client] ([ID\_Client]),  constraint [CH\_Date\_Booking] check ([Date] > getdate())  )  go | create or alter procedure [dbo].[Booking\_insert]  @Employee\_ID [int],@Application\_ID [int],@Rest\_Area\_ID [int],@Client\_ID [int],@Date [Date],@Time [Time]  as  begin try  insert into [dbo].[Booking] ([Employee\_ID], [Application\_ID],[Rest\_Area\_ID],  [Client\_ID],[Date],[Time])  values (@Employee\_ID,@Application\_ID,@Rest\_Area\_ID,@Client\_ID,@Date,@Time)  end try  begin catch  print('Данная бронь уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Booking\_update]  @ID\_Booking [int],@Employee\_ID [int],@Application\_ID [int],@Rest\_Area\_ID [int],@Client\_ID [int],@Date [Date],@Time [Time]  as  begin try  update [dbo].[Booking] set  [Employee\_ID] = @Employee\_ID,  [Application\_ID] = @Application\_ID,  [Rest\_Area\_ID] = @Rest\_Area\_ID,  [Client\_ID] = @Client\_ID,  [Date] = @Date,  [Time] = @Time  where  [ID\_Booking] = @ID\_Booking  end try  begin catch  print('Данная бронь уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Booking\_delete]  @ID\_Booking [int]  as  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Cheque]  where [Booking\_ID] = @ID\_Booking)  declare @any\_child\_record1 [int] = (select count(\*) from [dbo].[Reservation]  where [Booking\_ID] = @ID\_Booking)  if (@any\_child\_record > 0 or @any\_child\_record1 > 0)  print('Бронь не может быть удалён, так как в таблице "Персонал", есть связанные данные!');  else  delete from [dbo].[Booking]  where  [ID\_Booking] = @ID\_Booking  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Cheque]  (  [ID\_Cheque] [int] not null identity(1,1),  [Booking\_ID] [int] not null,  [Order\_Number] [int] not null,  [Date\_Cheque] [date] not null,  [Time\_Cheque] [time] not null  constraint [PK\_Cheque] primary key clustered  ([ID\_Cheque] ASC) on [PRIMARY],  constraint [FK\_Booking\_Cheque] foreign key ([Booking\_ID])  references [dbo].[Booking] ([ID\_Booking]),  constraint [UQ\_Order\_Number\_Cheque] unique ([Order\_Number])  )  go | create or alter procedure [dbo].[Cheque\_insert]  @Booking\_ID [int],@Order\_Number [int],@Date\_Cheque [Date],@Time\_Cheque [Time]  as  begin try  insert into [dbo].[Cheque] ([Booking\_ID],[Order\_Number], [Date\_Cheque], [Time\_Cheque])  values (@Booking\_ID,@Order\_Number,@Date\_Cheque,@Time\_Cheque)  end try  begin catch  print('Данный чек уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Cheque\_update]  @ID\_Cheque [int],@Booking\_ID [int],@Order\_Number [int],@Date\_Cheque [Date],@Time\_Cheque [Time]  as  begin try  update [dbo].[Cheque] set  [Booking\_ID] = @Booking\_ID,  [Order\_Number] = @Order\_Number,  [Date\_Cheque] = @Date\_Cheque,  [Time\_Cheque] = @Time\_Cheque  where  [ID\_Cheque] = @ID\_Cheque  end try  begin catch  print('Данный чек уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Cheque\_delete]  @ID\_Cheque [int]  as  delete from [dbo].[Cheque]  where  [ID\_Cheque] = @ID\_Cheque  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Prop]  (  [ID\_Prop] [int] not null identity(1,1),  [Raw\_ID] [int] not null,  [Supplier\_Company\_ID] [int] not null,  [Unique\_Number] [BIGINT] not null,  [Date] [date] not null,  [Time] [time] null default(Convert(time, SYSDATETIME()))  constraint [PK\_Prop] primary key clustered  ([ID\_Prop] ASC) on [PRIMARY],  constraint [FK\_Raw\_Prop] foreign key ([Raw\_ID])  references [dbo].[Raw] ([ID\_Raw]),  constraint [FK\_Supplier\_Company\_Prop] foreign key ([Supplier\_Company\_ID])  references [dbo].[Supplier\_Company] ([ID\_Supplier\_Company]),  constraint [CH\_Unique\_Number\_Prop] check ([Unique\_Number] > 0)  )  go | create or alter procedure [dbo].[Prop\_insert]  @Raw\_ID [int],@Supplier\_Company\_ID [int],@Unique\_Number [BIGINT],@Date [Date],@Time [Time]  as  begin try  insert into [dbo].[Prop] ([Raw\_ID],[Supplier\_Company\_ID],[Unique\_Number],[Date],[Time])  values (@Raw\_ID,@Supplier\_Company\_ID,@Unique\_Number,@Date,@Time)  end try  begin catch  print('Данный реквизит уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Prop\_update]  @ID\_Prop [int],@Raw\_ID [int],@Supplier\_Company\_ID [int],@Unique\_Number [BIGINT],@Date [Date],@Time [Time]  as  begin try  update [dbo].[Prop] set  [Raw\_ID] = @Raw\_ID,  [Supplier\_Company\_ID] = @Supplier\_Company\_ID,  [Unique\_Number] = @Unique\_Number,  [Date] = @Date,  [Time] = @Time  where  [ID\_Prop] = @ID\_Prop  end try  begin catch  print('Данный реквизит уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Prop\_delete]  @ID\_Prop [int]  as  delete from [dbo].[Prop]  where  [ID\_Prop] = @ID\_Prop  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Firm]  (  [ID\_Firm] [int] not null identity(1,1),  [Supply\_ID] [int] not null,  [Supplier\_Company\_ID] [int] not null  constraint [PK\_Firm] primary key clustered  ([ID\_Firm] ASC) on [PRIMARY],  constraint [FK\_Supply\_Firm] foreign key ([Supply\_ID])  references [dbo].[Supply] ([ID\_Supply]),  constraint [FK\_Supplier\_Company\_Firm] foreign key ([Supplier\_Company\_ID])  references [dbo].[Supplier\_Company] ([ID\_Supplier\_Company])  )  go | create or alter procedure [dbo].[Firm\_insert]  @Supply\_ID [int],@Supplier\_Company\_ID [int]  as  begin try  insert into [dbo].[Firm] ([Supply\_ID],[Supplier\_Company\_ID])  values (@Supply\_ID,@Supplier\_Company\_ID)  end try  begin catch  print('Данная фирма уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Firm\_update]  @ID\_Firm [int],@Supply\_ID [int],@Supplier\_Company\_ID [int]  as  begin try  update [dbo].[Firm] set  [Supply\_ID] = @Supply\_ID,  [Supplier\_Company\_ID] = @Supplier\_Company\_ID  where  [ID\_Firm] = @ID\_Firm  end try  begin catch  print('Данная фирма уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Firm\_delete]  @ID\_Firm [int]  as  delete from [dbo].[Firm]  where  [ID\_Firm] = @ID\_Firm  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Report]  (  [ID\_Report] [int] not null identity(1,1),  [Prop\_ID] [int] not null,  [Material\_ID] [int] not null  constraint [PK\_Report] primary key clustered  ([ID\_Report] ASC) on [PRIMARY],  constraint [FK\_Prop\_Report] foreign key ([Prop\_ID])  references [dbo].[Prop] ([ID\_Prop]),  constraint [FK\_Material\_Report] foreign key ([Material\_ID])  references [dbo].[Material] ([ID\_Material])  )  go | create or alter procedure [dbo].[Report\_insert]  @Prop\_ID [int],@Material\_ID [int]  as  begin try  insert into [dbo].[Report] ([Prop\_ID],[Material\_ID])  values (@Prop\_ID,@Material\_ID)  end try  begin catch  print('Данный доклад уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Report\_update]  @ID\_Report [int],@Prop\_ID [int],@Material\_ID [int]  as  begin try  update [dbo].[Report] set  [Prop\_ID] = @Prop\_ID,  [Material\_ID] = @Material\_ID  where  [ID\_Report] = @ID\_Report  end try  begin catch  print('Данный доклад уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Report\_delete]  @ID\_Report[int]  as  delete from [dbo].[Report]  where  [ID\_Report] = @ID\_Report  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Reservation]  (  [ID\_Reservation] [int] not null identity(1,1),  [Booking\_ID] [int] not null,  [Dish\_ID] [int] not null  constraint [PK\_Reservation] primary key clustered  ([ID\_Reservation] ASC) on [PRIMARY],  constraint [FK\_Booking\_Reservation] foreign key ([Booking\_ID])  references [dbo].[Booking] ([ID\_Booking]),  constraint [FK\_Dish\_Reservation] foreign key ([Dish\_ID])  references [dbo].[Dish] ([ID\_Dish])  )  go | create or alter procedure [dbo].[Reservation\_insert]  @Booking\_ID [int],@Dish\_ID [int]  as  begin try  insert into [dbo].[Reservation] ([Booking\_ID],[Dish\_ID])  values (@Booking\_ID,@Dish\_ID)  end try  begin catch  print('Данное бронирование уже есть в таблице!')  end catch  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Reservation\_update]  @ID\_Reservation [int],@Booking\_ID [int],@Dish\_ID [int]  as  begin try  update [dbo].[Reservation] set  [Booking\_ID] = @Booking\_ID,  [Dish\_ID] = @Dish\_ID  where  [ID\_Reservation] = @ID\_Reservation  end try  begin catch  print('Данное бронирование уже есть в таблице!')  end catch  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Reservation\_delete]  @ID\_Reservation [int]  as  delete from [dbo].[Reservation]  where  [ID\_Reservation] = @ID\_Reservation  go |

1. Иллюстрация обозревателя объекта







ВЫВОД

Я получил навыки, на основании проведённого анализа предметной области, а именно в проведении нормализации, приведение ненормализованной формы к 3НФ, применение свойств данных, которые подлежат хранению в базе данных к нормализации отношений, приведении отношений к структурированному виду, применение принципов и механизмов межтабличных связей.

1. СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

[Пример этапов практической](file:///\\ACER-AN\Users\logge\Desktop\база%20данных\3\Практическая%20работа%20№%202.docx) (Word, [задание](https://drive.google.com/file/d/1zYxKsKm5Ab-BnWJW_s2Mf3UT1uGUmMb6/view?usp=drive_web&authuser=1), [шаблон отчёта](https://drive.google.com/file/d/1Kc04FsjOh3HGFh_ZDjBzIs9-qh3YjYNP/view?usp=drive_web&authuser=1).