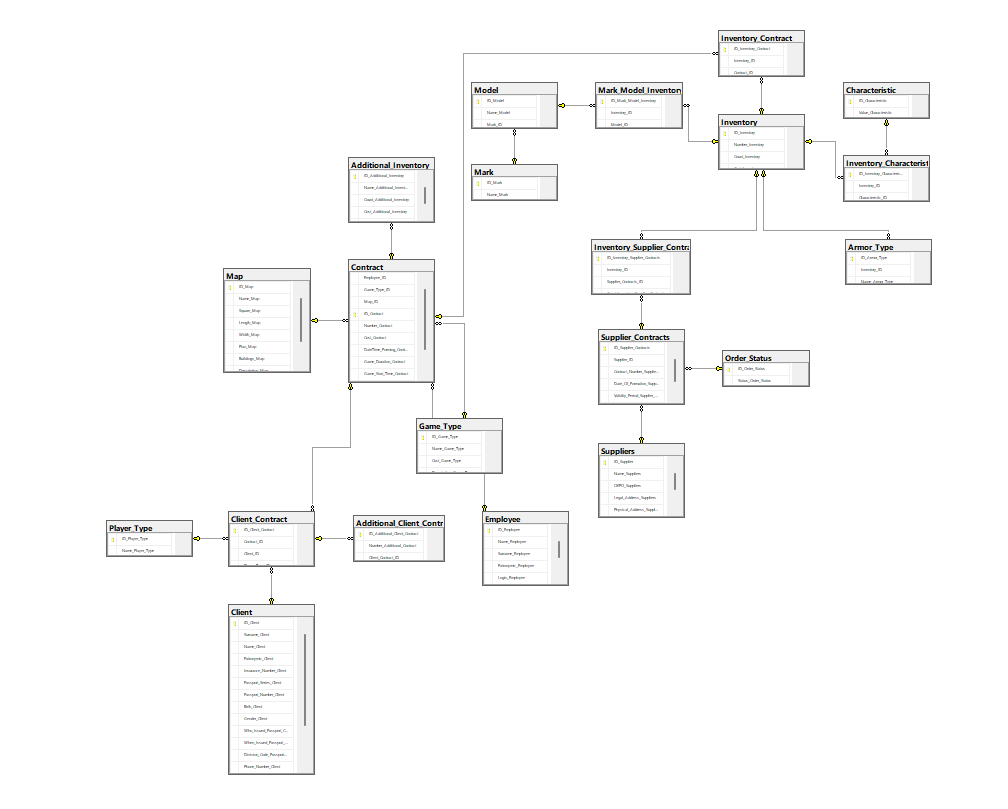
1. Иллюстрация диагнраммы базы данных;



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].[Employee]  (  [ID\_Employee] [int] not null identity(1,1),  [Name\_Employee] [varchar](50) not null,  [Surname\_Employee] [varchar](50) not null,  [Patronymic\_Employee] [varchar](50) null default 'Нет данных',  [Login\_Employee] [varchar](255) not null,  [Password\_Employee] [varchar](36) not null,  constraint [PK\_Employee] primary key clustered  ([ID\_Employee] ASC) on [PRIMARY],  constraint [UQ\_Login\_Employee] unique ([Login\_Employee]),  constraint [CH\_Login\_Employee] check (len([Login\_Employee]) >= 6),  constraint [CH\_Password\_Employee] check (len([Password\_Employee]) >= 6 and [Password\_Employee] like '%[a-zA-Z]%' and [Password\_Employee] like '%[0-9]%' and [Password\_Employee] like '%[!@#$%^&\*()]%')  )  go | create or alter procedure [dbo].[Employee\_insert]  @Name\_Employee [varchar](50),  @Surname\_Employee [varchar](50),  @Patronymic\_Employee [varchar](50) = 'Нет данных',  @Login\_Employee [varchar](255),  @Password\_Employee [varchar](36)  as  begin  if (len(@Login\_Employee) < 6)  begin  print('Логин должен содержать не менее 6 символов!');  return;  end  if (len(@Password\_Employee) < 6 or not @Password\_Employee like '%[a-zA-Z]%' or not @Password\_Employee like '%[0-9]%' or not @Password\_Employee like '%[!@#$%^&\*()]%')  begin  print('Пароль должен содержать не менее 6 символов, включая буквы, цифры и специальные символы!');  return;  end  insert into [dbo].[Employee] (Name\_Employee, Surname\_Employee, Patronymic\_Employee, Login\_Employee, Password\_Employee)  values (@Name\_Employee, @Surname\_Employee, @Patronymic\_Employee, @Login\_Employee, @Password\_Employee);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Employee\_update]  @ID\_Employee [int],  @Name\_Employee [varchar](50),  @Surname\_Employee [varchar](50),  @Patronymic\_Employee [varchar](50) = 'Нет данных',  @Login\_Employee [varchar](255),  @Password\_Employee [varchar](36)  as  begin  if (len(@Login\_Employee) < 6)  begin  print('Логин должен содержать не менее 6 символов!');  return;  end  if (len(@Password\_Employee) < 6 or not @Password\_Employee like '%[a-zA-Z]%' or not @Password\_Employee like '%[0-9]%' or not @Password\_Employee like '%[!@#$%^&\*()]%')  begin  print('Пароль должен содержать не менее 6 символов, включая буквы, цифры и специальные символы!');  return;  end  update [dbo].[Employee]  set Name\_Employee = @Name\_Employee,  Surname\_Employee = @Surname\_Employee,  Patronymic\_Employee = @Patronymic\_Employee,  Login\_Employee = @Login\_Employee,  Password\_Employee = @Password\_Employee  where ID\_Employee = @ID\_Employee;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Employee\_delete]  @ID\_Employee [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Contract] where Employee\_ID = @ID\_Employee)  if (@any\_child\_record > 0)  begin  print('Сотрудник не может быть удалён, так как в таблице "Contract", есть связанные данные!');  return;  end  delete from [dbo].[Employee]  where ID\_Employee = @ID\_Employee;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Client]  (  [ID\_Client] [int] not null identity(1,1),  [Surname\_Client] [varchar](50) not null,  [Name\_Client] [varchar](50) not null,  [Patronymic\_Client] [varchar](50) null default 'Нет данных',  [Insurance\_Number\_Client] [varchar](19) not null,  [Passport\_Series\_Client] [varchar](5) not null,  [Passport\_Number\_Client] [varchar](6) not null,  [Birth\_Client] [datetime] not null,  [Gender\_Client] [varchar](3) not null,  [Who\_Issued\_Passport\_Client] [varchar](max) not null,  [When\_Issued\_Passport\_Client] [date] not null,  [Division\_Code\_Passport\_Client] [varchar](7) not null,  [Phone\_Number\_Client] [varchar](16) not null,  [Email\_Client] [varchar](255) not null,  [Login\_Client] [varchar](255) not null,  [Password\_Client] [varchar](36) not null,  constraint [PK\_Client] primary key clustered  ([ID\_Client] ASC) on [PRIMARY],  constraint [UQ\_Insurance\_Number\_Client] unique ([Insurance\_Number\_Client]),  constraint [UQ\_Phone\_Number\_Client] unique ([Phone\_Number\_Client]),  constraint [UQ\_Email\_Client] unique ([Email\_Client]),  constraint [UQ\_Login\_Client] unique ([Login\_Client]),  constraint [CH\_Insurance\_Number\_Client] check ([Insurance\_Number\_Client] 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\_Passport\_Series\_Client] check ([Passport\_Series\_Client] like '[0-9][0-9] [0-9][0-9]'),  constraint [CH\_Passport\_Number\_Client] check ([Passport\_Number\_Client] like '[0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Birth\_Client] check ([Birth\_Client] <= dateadd(year, -17, getdate())),  constraint [CH\_Gender\_Client] check ([Gender\_Client] in ('Муж', 'Жен')),  constraint [CH\_When\_Issued\_Passport\_Client] check ([When\_Issued\_Passport\_Client] >= dateadd(year, -14, getdate())),  constraint [CH\_Division\_Code\_Passport\_Client] check ([Division\_Code\_Passport\_Client] like '[0-9][0-9][0-9]-[0-9][0-9][0-9]'),  constraint [CH\_Phone\_Number\_Client] check ([Phone\_Number\_Client] like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]'),  constraint [CH\_Email\_Client] check ([Email\_Client] like '%\_@\_\_%.\_\_%'),  constraint [CH\_Login\_Client] check (len([Login\_Client]) >= 6 and [Login\_Client] like '%[a-zA-Z]%'),  constraint [CH\_Password\_Client] check (len([Password\_Client]) >= 6 and [Password\_Client] like '%[a-zA-Z]%' and [Password\_Client] like '%[0-9]%' and [Password\_Client] like '%[!@#$%^&\*()]%')  )  go | create or alter procedure [dbo].[Client\_insert]  @Surname\_Client [varchar](50),  @Name\_Client [varchar](50),  @Patronymic\_Client [varchar](50) = 'Нет данных',  @Insurance\_Number\_Client [varchar](19),  @Passport\_Series\_Client [varchar](5),  @Passport\_Number\_Client [varchar](6),  @Birth\_Client [datetime],  @Gender\_Client [varchar](3),  @Who\_Issued\_Passport\_Client [varchar](max),  @When\_Issued\_Passport\_Client [date],  @Division\_Code\_Passport\_Client [varchar](7),  @Phone\_Number\_Client [varchar](16),  @Email\_Client [varchar](255),  @Login\_Client [varchar](255),  @Password\_Client [varchar](36)  as  begin  if (len(@Login\_Client) < 6 or not @Login\_Client like '%[a-zA-Z]%')  begin  print('Логин должен содержать не менее 6 символов и включать буквы!');  return;  end  if (len(@Password\_Client) < 6 or not @Password\_Client like '%[a-zA-Z]%' or not @Password\_Client like '%[0-9]%' or not @Password\_Client like '%[!@#$%^&\*()]%')  begin  print('Пароль должен содержать не менее 6 символов, включая буквы, цифры и специальные символы!');  return;  end  if (@Insurance\_Number\_Client not 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]')  begin  print('Неверный формат номера страхового полиса!');  return;  end  if (@Passport\_Series\_Client not like '[0-9][0-9] [0-9][0-9]')  begin  print('Неверный формат серии паспорта!');  return;  end  if (@Passport\_Number\_Client not like '[0-9][0-9][0-9][0-9][0-9][0-9]')  begin  print('Неверный формат номера паспорта!');  return;  end  if (@Birth\_Client > dateadd(year, -17, getdate()))  begin  print('Клиент должен быть старше 17 лет!');  return;  end  if (@Gender\_Client not in ('Муж', 'Жен'))  begin  print('Пол клиента должен быть "Муж" или "Жен"!');  return;  end  if (@When\_Issued\_Passport\_Client < dateadd(year, -14, getdate()))  begin  print('Паспорт должен быть выдан не более 14 лет назад!');  return;  end  if (@Division\_Code\_Passport\_Client not like '[0-9][0-9][0-9]-[0-9][0-9][0-9]')  begin  print('Неверный формат кода подразделения!');  return;  end  if (@Phone\_Number\_Client not like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]')  begin  print('Неверный формат номера телефона!');  return;  end  if (@Email\_Client not like '%\_@\_\_%.\_\_%')  begin  print('Неверный формат email!');  return;  end  insert into [dbo].[Client] (Surname\_Client, Name\_Client, Patronymic\_Client, Insurance\_Number\_Client, Passport\_Series\_Client, Passport\_Number\_Client, Birth\_Client, Gender\_Client, Who\_Issued\_Passport\_Client, When\_Issued\_Passport\_Client, Division\_Code\_Passport\_Client, Phone\_Number\_Client, Email\_Client, Login\_Client, Password\_Client)  values (@Surname\_Client, @Name\_Client, @Patronymic\_Client, @Insurance\_Number\_Client, @Passport\_Series\_Client, @Passport\_Number\_Client, @Birth\_Client, @Gender\_Client, @Who\_Issued\_Passport\_Client, @When\_Issued\_Passport\_Client, @Division\_Code\_Passport\_Client, @Phone\_Number\_Client, @Email\_Client, @Login\_Client, @Password\_Client);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Client\_update]  @ID\_Client [int],  @Surname\_Client [varchar](50),  @Name\_Client [varchar](50),  @Patronymic\_Client [varchar](50) = 'Нет данных',  @Insurance\_Number\_Client [varchar](19),  @Passport\_Series\_Client [varchar](5),  @Passport\_Number\_Client [varchar](6),  @Birth\_Client [datetime],  @Gender\_Client [varchar](3),  @Who\_Issued\_Passport\_Client [varchar](max),  @When\_Issued\_Passport\_Client [date],  @Division\_Code\_Passport\_Client [varchar](7),  @Phone\_Number\_Client [varchar](16),  @Email\_Client [varchar](255),  @Login\_Client [varchar](255),  @Password\_Client [varchar](36)  as  begin  if (len(@Login\_Client) < 6 or not @Login\_Client like '%[a-zA-Z]%')  begin  print('Логин должен содержать не менее 6 символов и включать буквы!');  return;  end  if (len(@Password\_Client) < 6 or not @Password\_Client like '%[a-zA-Z]%' or not @Password\_Client like '%[0-9]%' or not @Password\_Client like '%[!@#$%^&\*()]%')  begin  print('Пароль должен содержать не менее 6 символов, включая буквы, цифры и специальные символы!');  return;  end  if (@Insurance\_Number\_Client not 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]')  begin  print('Неверный формат номера страхового полиса!');  return;  end  if (@Passport\_Series\_Client not like '[0-9][0-9] [0-9][0-9]')  begin  print('Неверный формат серии паспорта!');  return;  end  if (@Passport\_Number\_Client not like '[0-9][0-9][0-9][0-9][0-9][0-9]')  begin  print('Неверный формат номера паспорта!');  return;  end  if (@Birth\_Client > dateadd(year, -17, getdate()))  begin  print('Клиент должен быть старше 17 лет!');  return;  end  if (@Gender\_Client not in ('Муж', 'Жен'))  begin  print('Пол клиента должен быть "Муж" или "Жен"!');  return;  end  if (@When\_Issued\_Passport\_Client < dateadd(year, -14, getdate()))  begin  print('Паспорт должен быть выдан не более 14 лет назад!');  return;  end  if (@Division\_Code\_Passport\_Client not like '[0-9][0-9][0-9]-[0-9][0-9][0-9]')  begin  print('Неверный формат кода подразделения!');  return;  end  if (@Phone\_Number\_Client not like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]')  begin  print('Неверный формат номера телефона!');  return;  end  if (@Email\_Client not like '%\_@\_\_%.\_\_%')  begin  print('Неверный формат email!');  return;  end  update [dbo].[Client]  set Surname\_Client = @Surname\_Client,  Name\_Client = @Name\_Client,  Patronymic\_Client = @Patronymic\_Client,  Insurance\_Number\_Client = @Insurance\_Number\_Client,  Passport\_Series\_Client = @Passport\_Series\_Client,  Passport\_Number\_Client = @Passport\_Number\_Client,  Birth\_Client = @Birth\_Client,  Gender\_Client = @Gender\_Client,  Who\_Issued\_Passport\_Client = @Who\_Issued\_Passport\_Client,  When\_Issued\_Passport\_Client = @When\_Issued\_Passport\_Client,  Division\_Code\_Passport\_Client = @Division\_Code\_Passport\_Client,  Phone\_Number\_Client = @Phone\_Number\_Client,  Email\_Client = @Email\_Client,  Login\_Client = @Login\_Client,  Password\_Client = @Password\_Client  where ID\_Client = @ID\_Client;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Client\_delete]  @ID\_Client [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Client\_Contract] where Client\_ID = @ID\_Client)  if (@any\_child\_record > 0)  begin  print('Клиент не может быть удалён, так как в таблице "Client\_Contract", есть связанные данные!');  return;  end  delete from [dbo].[Client]  where ID\_Client = @ID\_Client;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Map]  (  [ID\_Map] [int] not null identity(1,1),  [Name\_Map] [varchar](20) not null,  [Square\_Map] [decimal](5,2) not null,  [Length\_Map] [decimal](5,2) not null,  [Width\_Map] [decimal](5,2) not null,  [Plan\_Map] [varchar](max) not null,  [Buildings\_Map] [varchar](max) null default 'Отсутствует',  [Description\_Map] [varchar](max) not null,  [Cost\_Map] [decimal](10,3) not null,  constraint [PK\_Map] primary key clustered  ([ID\_Map] ASC) on [PRIMARY],  constraint [UQ\_Name\_Map] unique ([Name\_Map]),  constraint [CH\_Square\_Map] check ([Square\_Map] > 0),  constraint [CH\_Length\_Map] check ([Length\_Map] > 0),  constraint [CH\_Width\_Map] check ([Width\_Map] > 0),  constraint [CH\_Cost\_Map] check ([Cost\_Map] >= 0)  )  go | create or alter procedure [dbo].[Map\_insert]  @Name\_Map [varchar](20),  @Square\_Map [decimal](5,2),  @Length\_Map [decimal](5,2),  @Width\_Map [decimal](5,2),  @Plan\_Map [varchar](max),  @Buildings\_Map [varchar](max) = 'Отсутствует',  @Description\_Map [varchar](max),  @Cost\_Map [decimal](10,3)  as  begin  if (@Square\_Map <= 0)  begin  print('Площадь карты должна быть больше 0!');  return;  end  if (@Length\_Map <= 0)  begin  print('Длина карты должна быть больше 0!');  return;  end  if (@Width\_Map <= 0)  begin  print('Ширина карты должна быть больше 0!');  return;  end  if (@Cost\_Map < 0)  begin  print('Стоимость карты не может быть отрицательной!');  return;  end  insert into [dbo].[Map] (Name\_Map, Square\_Map, Length\_Map, Width\_Map, Plan\_Map, Buildings\_Map, Description\_Map, Cost\_Map)  values (@Name\_Map, @Square\_Map, @Length\_Map, @Width\_Map, @Plan\_Map, @Buildings\_Map, @Description\_Map, @Cost\_Map);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Map\_update]  @ID\_Map [int],  @Name\_Map [varchar](20),  @Square\_Map [decimal](5,2),  @Length\_Map [decimal](5,2),  @Width\_Map [decimal](5,2),  @Plan\_Map [varchar](max),  @Buildings\_Map [varchar](max) = 'Отсутствует',  @Description\_Map [varchar](max),  @Cost\_Map [decimal](10,3)  as  begin  if (@Square\_Map <= 0)  begin  print('Площадь карты должна быть больше 0!');  return;  end  if (@Length\_Map <= 0)  begin  print('Длина карты должна быть больше 0!');  return;  end  if (@Width\_Map <= 0)  begin  print('Ширина карты должна быть больше 0!');  return;  end  if (@Cost\_Map < 0)  begin  print('Стоимость карты не может быть отрицательной!');  return;  end  update [dbo].[Map]  set Name\_Map = @Name\_Map,  Square\_Map = @Square\_Map,  Length\_Map = @Length\_Map,  Width\_Map = @Width\_Map,  Plan\_Map = @Plan\_Map,  Buildings\_Map = @Buildings\_Map,  Description\_Map = @Description\_Map,  Cost\_Map = @Cost\_Map  where ID\_Map = @ID\_Map;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Map\_delete]  @ID\_Map [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Contract] where Map\_ID = @ID\_Map)  if (@any\_child\_record > 0)  begin  print('Карта не может быть удалена, так как в таблице "Contract", есть связанные данные!');  return;  end  delete from [dbo].[Map]  where ID\_Map = @ID\_Map;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Game\_Type]  (  [ID\_Game\_Type] [int] not null identity(1,1),  [Name\_Game\_Type] [varchar](20) not null,  [Cost\_Game\_Type] [decimal](10,3) not null,  [Description\_Game\_Type] [varchar](max) not null,  constraint [PK\_Game\_Type] primary key clustered  ([ID\_Game\_Type] ASC) on [PRIMARY],  constraint [UQ\_Name\_Game\_Type] unique ([Name\_Game\_Type]),  constraint [CH\_Cost\_Game\_Type] check ([Cost\_Game\_Type] >= 0)  )  go | create or alter procedure [dbo].[Game\_Type\_insert]  @Name\_Game\_Type [varchar](20),  @Cost\_Game\_Type [decimal](10,3),  @Description\_Game\_Type [varchar](max)  as  begin  if (@Cost\_Game\_Type < 0)  begin  print('Стоимость типа игры не может быть отрицательной!');  return;  end  insert into [dbo].[Game\_Type] (Name\_Game\_Type, Cost\_Game\_Type, Description\_Game\_Type)  values (@Name\_Game\_Type, @Cost\_Game\_Type, @Description\_Game\_Type);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Game\_Type\_update]  @ID\_Game\_Type [int],  @Name\_Game\_Type [varchar](20),  @Cost\_Game\_Type [decimal](10,3),  @Description\_Game\_Type [varchar](max)  as  begin  if (@Cost\_Game\_Type < 0)  begin  print('Стоимость типа игры не может быть отрицательной!');  return;  end  update [dbo].[Game\_Type]  set Name\_Game\_Type = @Name\_Game\_Type,  Cost\_Game\_Type = @Cost\_Game\_Type,  Description\_Game\_Type = @Description\_Game\_Type  where ID\_Game\_Type = @ID\_Game\_Type;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Game\_Type\_delete]  @ID\_Game\_Type [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Contract] where Game\_Type\_ID = @ID\_Game\_Type)  if (@any\_child\_record > 0)  begin  print('Тип игры не может быть удален, так как в таблице "Contract", есть связанные данные!');  return;  end  delete from [dbo].[Game\_Type]  where ID\_Game\_Type = @ID\_Game\_Type;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Inventory]  (  [ID\_Inventory] [int] not null identity(1,1),  [Number\_Inventory] [varchar](16) not null,  [Count\_Inventory] [int] not null,  [Cost\_Inventory] [decimal](10,3) not null,  constraint [PK\_Inventory] primary key clustered  ([ID\_Inventory] ASC) on [PRIMARY],  constraint [UQ\_Number\_Inventory] unique ([Number\_Inventory]),  constraint [CH\_Number\_Inventory] check ([Number\_Inventory] like 'ИнВ-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Count\_Inventory] check ([Count\_Inventory] >= 0),  constraint [CH\_Cost\_Inventory] check ([Cost\_Inventory] >= 0)  )  go | create or alter procedure [dbo].[Inventory\_insert]  @Number\_Inventory [varchar](16),  @Count\_Inventory [int],  @Cost\_Inventory [decimal](10,3)  as  begin  if (@Number\_Inventory not like 'ИнВ-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')  begin  print('Неверный формат номера инвентаря!');  return;  end  if (@Count\_Inventory < 0)  begin  print('Количество инвентаря не может быть отрицательным!');  return;  end  if (@Cost\_Inventory < 0)  begin  print('Стоимость инвентаря не может быть отрицательной!');  return;  end  insert into [dbo].[Inventory] (Number\_Inventory, Count\_Inventory, Cost\_Inventory)  values (@Number\_Inventory, @Count\_Inventory, @Cost\_Inventory);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Inventory\_update]  @ID\_Inventory [int],  @Number\_Inventory [varchar](16),  @Count\_Inventory [int],  @Cost\_Inventory [decimal](10,3)  as  begin  if (@Number\_Inventory not like 'ИнВ-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')  begin  print('Неверный формат номера инвентаря!');  return;  end  if (@Count\_Inventory < 0)  begin  print('Количество инвентаря не может быть отрицательным!');  return;  end  if (@Cost\_Inventory < 0)  begin  print('Стоимость инвентаря не может быть отрицательной!');  return;  end  update [dbo].[Inventory]  set Number\_Inventory = @Number\_Inventory,  Count\_Inventory = @Count\_Inventory,  Cost\_Inventory = @Cost\_Inventory  where ID\_Inventory = @ID\_Inventory;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Inventory\_delete]  @ID\_Inventory [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Inventory\_Contract] where Inventory\_ID = @ID\_Inventory)  if (@any\_child\_record > 0)  begin  print('Инвентарь не может быть удален, так как в таблице "Inventory\_Contract", есть связанные данные!');  return;  end  delete from [dbo].[Inventory]  where ID\_Inventory = @ID\_Inventory;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Characteristic]  (  [ID\_Characteristic] [int] not null identity(1,1),  [Value\_Characteristic] [varchar](50) not null,  constraint [PK\_Characteristic] primary key clustered  ([ID\_Characteristic] ASC) on [PRIMARY],  constraint [UQ\_Value\_Characteristic] unique ([Value\_Characteristic])  )  go | create or alter procedure [dbo].[Characteristic\_insert]  @Value\_Characteristic [varchar](50)  as  begin  insert into [dbo].[Characteristic] (Value\_Characteristic)  values (@Value\_Characteristic);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Characteristic\_update]  @ID\_Characteristic [int],  @Value\_Characteristic [varchar](50)  as  begin  update [dbo].[Characteristic]  set Value\_Characteristic = @Value\_Characteristic  where ID\_Characteristic = @ID\_Characteristic;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Characteristic\_delete]  @ID\_Characteristic [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Inventory\_Characteristic] where Characteristic\_ID = @ID\_Characteristic)  if (@any\_child\_record > 0)  begin  print('Характеристика не может быть удалена, так как в таблице "Inventory\_Characteristic", есть связанные данные!');  return;  end  delete from [dbo].[Characteristic]  where ID\_Characteristic = @ID\_Characteristic;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Contract]  (  [Employee\_ID] [int] not null,  [Game\_Type\_ID] [int] not null,  [Map\_ID] [int] not null,  [ID\_Contract] [int] not null identity(1,1),  [Number\_Contract] [varchar](10) not null,  [Cost\_Contract] [decimal](10,3) not null,  [DateTime\_Forming\_Contract] [datetime] not null,  [Game\_Duration\_Contract] [decimal](5,2) not null,  [Game\_Start\_Time\_Contract] [time] not null,  [Game\_End\_Time\_Contract] [time] not null,  [Final\_Cost\_Contract] [decimal](10,3) not null,  constraint [PK\_Contract] primary key clustered  ([ID\_Contract] ASC) on [PRIMARY],  constraint [UQ\_Number\_Contract] unique ([Number\_Contract]),  constraint [CH\_Number\_Contract] check ([Number\_Contract] like 'GM-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Cost\_Contract] check ([Cost\_Contract] >= 0),  constraint [CH\_Game\_Duration\_Contract] check ([Game\_Duration\_Contract] >= 0),  constraint [CH\_Game\_Start\_Time\_Contract] check ([Game\_Start\_Time\_Contract] > cast([DateTime\_Forming\_Contract] as time)),  constraint [CH\_Game\_End\_Time\_Contract] check ([Game\_End\_Time\_Contract] > [Game\_Start\_Time\_Contract] and [Game\_End\_Time\_Contract] > cast([DateTime\_Forming\_Contract] as time)),  constraint [CH\_Final\_Cost\_Contract] check ([Final\_Cost\_Contract] > 0),  constraint [FK\_Employee\_Contract] foreign key ([Employee\_ID])  references [dbo].[Employee] ([ID\_Employee]),  constraint [FK\_Game\_Type\_Contract] foreign key ([Game\_Type\_ID])  references [dbo].[Game\_Type] ([ID\_Game\_Type]),  constraint [FK\_Map\_Contract] foreign key ([Map\_ID])  references [dbo].[Map] ([ID\_Map])  )  go | create or alter procedure [dbo].[Contract\_insert]  @Employee\_ID [int],  @Game\_Type\_ID [int],  @Map\_ID [int],  @Number\_Contract [varchar](10),  @Cost\_Contract [decimal](10,3),  @DateTime\_Forming\_Contract [datetime],  @Game\_Duration\_Contract [decimal](5,2),  @Game\_Start\_Time\_Contract [time],  @Game\_End\_Time\_Contract [time],  @Final\_Cost\_Contract [decimal](10,3)  as  begin  if (@Number\_Contract not like 'GM-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]')  begin  print('Неверный формат номера контракта!');  return;  end  if (@Cost\_Contract < 0)  begin  print('Стоимость контракта не может быть отрицательной!');  return;  end  if (@Game\_Duration\_Contract < 0)  begin  print('Продолжительность игры не может быть отрицательной!');  return;  end  if (@Game\_Start\_Time\_Contract > @Game\_End\_Time\_Contract)  begin  print('Время начала игры не может быть позже времени окончания игры!');  return;  end  if (@Final\_Cost\_Contract <= 0)  begin  print('Итоговая стоимость контракта должна быть больше 0!');  return;  end  insert into [dbo].[Contract] (Employee\_ID, Game\_Type\_ID, Map\_ID, Number\_Contract, Cost\_Contract, DateTime\_Forming\_Contract, Game\_Duration\_Contract, Game\_Start\_Time\_Contract, Game\_End\_Time\_Contract, Final\_Cost\_Contract)  values (@Employee\_ID, @Game\_Type\_ID, @Map\_ID, @Number\_Contract, @Cost\_Contract, @DateTime\_Forming\_Contract, @Game\_Duration\_Contract, @Game\_Start\_Time\_Contract, @Game\_End\_Time\_Contract, @Final\_Cost\_Contract);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Contract\_update]  @ID\_Contract [int],  @Employee\_ID [int],  @Game\_Type\_ID [int],  @Map\_ID [int],  @Number\_Contract [varchar](10),  @Cost\_Contract [decimal](10,3),  @DateTime\_Forming\_Contract [datetime],  @Game\_Duration\_Contract [decimal](5,2),  @Game\_Start\_Time\_Contract [time],  @Game\_End\_Time\_Contract [time],  @Final\_Cost\_Contract [decimal](10,3)  as  begin  if (@Number\_Contract not like 'GM-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]')  begin  print('Неверный формат номера контракта!');  return;  end  if (@Cost\_Contract < 0)  begin  print('Стоимость контракта не может быть отрицательной!');  return;  end  if (@Game\_Duration\_Contract < 0)  begin  print('Продолжительность игры не может быть отрицательной!');  return;  end  if (@Game\_Start\_Time\_Contract > @Game\_End\_Time\_Contract)  begin  print('Время начала игры не может быть позже времени окончания игры!');  return;  end  if (@Final\_Cost\_Contract <= 0)  begin  print('Итоговая стоимость контракта должна быть больше 0!');  return;  end  update [dbo].[Contract]  set Employee\_ID = @Employee\_ID,  Game\_Type\_ID = @Game\_Type\_ID,  Map\_ID = @Map\_ID,  Number\_Contract = @Number\_Contract,  Cost\_Contract = @Cost\_Contract,  DateTime\_Forming\_Contract = @DateTime\_Forming\_Contract,  Game\_Duration\_Contract = @Game\_Duration\_Contract,  Game\_Start\_Time\_Contract = @Game\_Start\_Time\_Contract,  Game\_End\_Time\_Contract = @Game\_End\_Time\_Contract,  Final\_Cost\_Contract = @Final\_Cost\_Contract  where ID\_Contract = @ID\_Contract;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Contract\_delete]  @ID\_Contract [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Client\_Contract] where Contract\_ID = @ID\_Contract)  if (@any\_child\_record > 0)  begin  print('Контракт не может быть удален, так как в таблице "Client\_Contract", есть связанные данные!');  return;  end  delete from [dbo].[Contract]  where ID\_Contract = @ID\_Contract;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Player\_Type]  (  [ID\_Player\_Type] [int] not null identity(1,1),  [Name\_Player\_Type] [varchar](20) not null,  constraint [PK\_Player\_Type] primary key clustered  ([ID\_Player\_Type] ASC) on [PRIMARY],  constraint [UQ\_Name\_Player\_Type] unique ([Name\_Player\_Type]),  constraint [CH\_Name\_Player\_Type] check ([Name\_Player\_Type] in ('Ответственный', 'Участник'))  )  go | create or alter procedure [dbo].[Player\_Type\_insert]  @Name\_Player\_Type [varchar](20)  as  begin  if (@Name\_Player\_Type not in ('Ответственный', 'Участник'))  begin  print('Тип игрока должен быть "Ответственный" или "Участник"!');  return;  end  insert into [dbo].[Player\_Type] (Name\_Player\_Type)  values (@Name\_Player\_Type);  end  go |
| Хранимая процедура на изменение данных |
| Не рационально |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Player\_Type\_delete]  @ID\_Player\_Type [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Client\_Contract] where Player\_Type\_ID = @ID\_Player\_Type)  if (@any\_child\_record > 0)  begin  print('Тип игрока не может быть удален, так как в таблице "Client\_Contract", есть связанные данные!');  return;  end  delete from [dbo].[Player\_Type]  where ID\_Player\_Type = @ID\_Player\_Type;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Client\_Contract]  (  [ID\_Client\_Contract] [int] not null identity(1,1),  [Contract\_ID] [int] not null,  [Client\_ID] [int] not null,  [Player\_Type\_ID] [int] not null,  constraint [PK\_Client\_Contract] primary key clustered  ([ID\_Client\_Contract] ASC) on [PRIMARY],  constraint [FK\_Contract\_Client\_Contract] foreign key ([Contract\_ID])  references [dbo].[Contract] ([ID\_Contract]),  constraint [FK\_Client\_Client\_Contract] foreign key ([Client\_ID])  references [dbo].[Client] ([ID\_Client]),  constraint [FK\_Player\_Type\_Client\_Contract] foreign key ([Player\_Type\_ID])  references [dbo].[Player\_Type] ([ID\_Player\_Type])  )  go | create or alter procedure [dbo].[Client\_Contract\_insert]  @Contract\_ID [int],  @Client\_ID [int],  @Player\_Type\_ID [int]  as  begin  insert into [dbo].[Client\_Contract] (Contract\_ID, Client\_ID, Player\_Type\_ID)  values (@Contract\_ID, @Client\_ID, @Player\_Type\_ID);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Client\_Contract\_update]  @ID\_Client\_Contract [int],  @Contract\_ID [int],  @Client\_ID [int],  @Player\_Type\_ID [int]  as  begin  update [dbo].[Client\_Contract]  set Contract\_ID = @Contract\_ID,  Client\_ID = @Client\_ID,  Player\_Type\_ID = @Player\_Type\_ID  where ID\_Client\_Contract = @ID\_Client\_Contract;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Client\_Contract\_delete]  @ID\_Client\_Contract [int]  as  begin  delete from [dbo].[Client\_Contract]  where ID\_Client\_Contract = @ID\_Client\_Contract;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Inventory\_Characteristic]  (  [ID\_Inventory\_Characteristic] [int] not null identity(1,1),  [Inventory\_ID] [int] not null,  [Characteristic\_ID] [int] not null,  constraint [PK\_Inventory\_Characteristic] primary key clustered  ([ID\_Inventory\_Characteristic] ASC) on [PRIMARY],  constraint [FK\_Inventory\_Inventory\_Characteristic] foreign key ([Inventory\_ID])  references [dbo].[Inventory] ([ID\_Inventory]),  constraint [FK\_Characteristic\_Inventory\_Characteristic] foreign key ([Characteristic\_ID])  references [dbo].[Characteristic] ([ID\_Characteristic])  )  go | create or alter procedure [dbo].[Inventory\_Characteristic\_insert]  @Inventory\_ID [int],  @Characteristic\_ID [int]  as  begin  insert into [dbo].[Inventory\_Characteristic] (Inventory\_ID, Characteristic\_ID)  values (@Inventory\_ID, @Characteristic\_ID);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Inventory\_Characteristic\_update]  @ID\_Inventory\_Characteristic [int],  @Inventory\_ID [int],  @Characteristic\_ID [int]  as  begin  update [dbo].[Inventory\_Characteristic]  set Inventory\_ID = @Inventory\_ID,  Characteristic\_ID = @Characteristic\_ID  where ID\_Inventory\_Characteristic = @ID\_Inventory\_Characteristic;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Inventory\_Characteristic\_delete]  @ID\_Inventory\_Characteristic [int]  as  begin  delete from [dbo].[Inventory\_Characteristic]  where ID\_Inventory\_Characteristic = @ID\_Inventory\_Characteristic;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Inventory\_Contract]  (  [ID\_Inventory\_Contract] [int] not null identity(1,1),  [Inventory\_ID] [int] not null,  [Contract\_ID] [int] not null,  constraint [PK\_Inventory\_Contract] primary key clustered  ([ID\_Inventory\_Contract] ASC) on [PRIMARY],  constraint [FK\_Inventory\_Inventory\_Contract] foreign key ([Inventory\_ID])  references [dbo].[Inventory] ([ID\_Inventory]),  constraint [FK\_Contract\_Inventory\_Contract] foreign key ([Contract\_ID])  references [dbo].[Contract] ([ID\_Contract])  )  go | create or alter procedure [dbo].[Inventory\_Contract\_insert]  @Inventory\_ID [int],  @Contract\_ID [int]  as  begin  insert into [dbo].[Inventory\_Contract] (Inventory\_ID, Contract\_ID)  values (@Inventory\_ID, @Contract\_ID);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Inventory\_Contract\_update]  @ID\_Inventory\_Contract [int],  @Inventory\_ID [int],  @Contract\_ID [int]  as  begin  update [dbo].[Inventory\_Contract]  set Inventory\_ID = @Inventory\_ID,  Contract\_ID = @Contract\_ID  where ID\_Inventory\_Contract = @ID\_Inventory\_Contract;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Inventory\_Contract\_delete]  @ID\_Inventory\_Contract [int]  as  begin  delete from [dbo].[Inventory\_Contract]  where ID\_Inventory\_Contract = @ID\_Inventory\_Contract;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Additional\_Client\_Contract]  (  [ID\_Additional\_Client\_Contract] [int] not null identity(1,1),  [Number\_Additional\_Contract] [varchar](12) not null,  [Client\_Contract\_ID] [int] not null,  constraint [PK\_Additional\_Client\_Contract] primary key clustered  ([ID\_Additional\_Client\_Contract] ASC) on [PRIMARY],  constraint [UQ\_Number\_Additional\_Contract] unique ([Number\_Additional\_Contract]),  constraint [CH\_Number\_Additional\_Contract] check ([Number\_Additional\_Contract] like 'GM-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]/[0-9]'),  constraint [FK\_Client\_Contract\_Additional\_Client\_Contract] foreign key ([Client\_Contract\_ID])  references [dbo].[Client\_Contract] ([ID\_Client\_Contract])  )  go | create or alter procedure [dbo].[Additional\_Client\_Contract\_insert]  @Number\_Additional\_Contract [varchar](12),  @Client\_Contract\_ID [int]  as  begin  if (@Number\_Additional\_Contract not like 'GM-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]/[0-9]')  begin  print('Неверный формат номера дополнительного контракта!');  return;  end  insert into [dbo].[Additional\_Client\_Contract] (Number\_Additional\_Contract, Client\_Contract\_ID)  values (@Number\_Additional\_Contract, @Client\_Contract\_ID);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Additional\_Client\_Contract\_update]  @ID\_Additional\_Client\_Contract [int],  @Number\_Additional\_Contract [varchar](12),  @Client\_Contract\_ID [int]  as  begin  if (@Number\_Additional\_Contract not like 'GM-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]/[0-9]')  begin  print('Неверный формат номера дополнительного контракта!');  return;  end  update [dbo].[Additional\_Client\_Contract]  set Number\_Additional\_Contract = @Number\_Additional\_Contract,  Client\_Contract\_ID = @Client\_Contract\_ID  where ID\_Additional\_Client\_Contract = @ID\_Additional\_Client\_Contract;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Additional\_Client\_Contract\_delete]  @ID\_Additional\_Client\_Contract [int]  as  begin  delete from [dbo].[Additional\_Client\_Contract]  where ID\_Additional\_Client\_Contract = @ID\_Additional\_Client\_Contract;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Armor\_Type]  (  [ID\_Armor\_Type] [int] not null identity(1,1),  [Inventory\_ID] [int] not null,  [Name\_Armor\_Type] [varchar](15) not null,  constraint [PK\_Armor\_Type] primary key clustered  ([ID\_Armor\_Type] ASC) on [PRIMARY],  constraint [UQ\_Name\_Armor\_Type] unique ([Name\_Armor\_Type]),  constraint [FK\_Inventory\_Armor\_Type] foreign key ([Inventory\_ID])  references [dbo].[Inventory] ([ID\_Inventory])  )  go | create or alter procedure [dbo].[Armor\_Type\_insert]  @Inventory\_ID [int],  @Name\_Armor\_Type [varchar](15)  as  begin  insert into [dbo].[Armor\_Type] (Inventory\_ID, Name\_Armor\_Type)  values (@Inventory\_ID, @Name\_Armor\_Type);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Armor\_Type\_update]  @ID\_Armor\_Type [int],  @Inventory\_ID [int],  @Name\_Armor\_Type [varchar](15)  as  begin  update [dbo].[Armor\_Type]  set Inventory\_ID = @Inventory\_ID,  Name\_Armor\_Type = @Name\_Armor\_Type  where ID\_Armor\_Type = @ID\_Armor\_Type;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Armor\_Type\_delete]  @ID\_Armor\_Type [int]  as  begin  delete from [dbo].[Armor\_Type]  where ID\_Armor\_Type = @ID\_Armor\_Type;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Additional\_Inventory]  (  [ID\_Additional\_Inventory] [int] not null identity(1,1),  [Name\_Additional\_Inventory] [varchar](30) not null,  [Count\_Additional\_Inventory] [int] not null,  [Cost\_Additional\_Inventory] [decimal](10,3) not null,  [Contract\_ID] [int] not null,  constraint [PK\_Additional\_Inventory] primary key clustered  ([ID\_Additional\_Inventory] ASC) on [PRIMARY],  constraint [CH\_Count\_Additional\_Inventory] check ([Count\_Additional\_Inventory] >= 0),  constraint [CH\_Cost\_Additional\_Inventory] check ([Cost\_Additional\_Inventory] >= 0),  constraint [FK\_Contract\_Additional\_Inventory] foreign key ([Contract\_ID])  references [dbo].[Contract] ([ID\_Contract])  )  go | create or alter procedure [dbo].[Additional\_Inventory\_insert]  @Name\_Additional\_Inventory [varchar](30),  @Count\_Additional\_Inventory [int],  @Cost\_Additional\_Inventory [decimal](10,3),  @Contract\_ID [int]  as  begin  if (@Count\_Additional\_Inventory < 0)  begin  print('Количество дополнительного инвентаря не может быть отрицательным!');  return;  end  if (@Cost\_Additional\_Inventory < 0)  begin  print('Стоимость дополнительного инвентаря не может быть отрицательной!');  return;  end  insert into [dbo].[Additional\_Inventory] (Name\_Additional\_Inventory, Count\_Additional\_Inventory, Cost\_Additional\_Inventory, Contract\_ID)  values (@Name\_Additional\_Inventory, @Count\_Additional\_Inventory, @Cost\_Additional\_Inventory, @Contract\_ID);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Additional\_Inventory\_update]  @ID\_Additional\_Inventory [int],  @Name\_Additional\_Inventory [varchar](30),  @Count\_Additional\_Inventory [int],  @Cost\_Additional\_Inventory [decimal](10,3),  @Contract\_ID [int]  as  begin  if (@Count\_Additional\_Inventory < 0)  begin  print('Количество дополнительного инвентаря не может быть отрицательным!');  return;  end  if (@Cost\_Additional\_Inventory < 0)  begin  print('Стоимость дополнительного инвентаря не может быть отрицательной!');  return;  end  update [dbo].[Additional\_Inventory]  set Name\_Additional\_Inventory = @Name\_Additional\_Inventory,  Count\_Additional\_Inventory = @Count\_Additional\_Inventory,  Cost\_Additional\_Inventory = @Cost\_Additional\_Inventory,  Contract\_ID = @Contract\_ID  where ID\_Additional\_Inventory = @ID\_Additional\_Inventory;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Additional\_Inventory\_delete]  @ID\_Additional\_Inventory [int]  as  begin  delete from [dbo].[Additional\_Inventory]  where ID\_Additional\_Inventory = @ID\_Additional\_Inventory;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Mark]  (  [ID\_Mark] [int] not null identity(1,1),  [Name\_Mark] [varchar](15) not null,  constraint [PK\_Mark] primary key clustered  ([ID\_Mark] ASC) on [PRIMARY],  constraint [UQ\_Name\_Mark] unique ([Name\_Mark])  )  go | create or alter procedure [dbo].[Mark\_insert]  @Name\_Mark [varchar](15)  as  begin  insert into [dbo].[Mark] (Name\_Mark)  values (@Name\_Mark);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Mark\_update]  @ID\_Mark [int],  @Name\_Mark [varchar](15)  as  begin  update [dbo].[Mark]  set Name\_Mark = @Name\_Mark  where ID\_Mark = @ID\_Mark;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Mark\_delete]  @ID\_Mark [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Model] where Mark\_ID = @ID\_Mark)  if (@any\_child\_record > 0)  begin  print('Марка не может быть удалена, так как в таблице "Model", есть связанные данные!');  return;  end  delete from [dbo].[Mark]  where ID\_Mark = @ID\_Mark;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Model]  (  [ID\_Model] [int] not null identity(1,1),  [Name\_Model] [varchar](15) not null,  [Mark\_ID] [int] not null,  constraint [PK\_Model] primary key clustered  ([ID\_Model] ASC) on [PRIMARY],  constraint [UQ\_Name\_Model] unique ([Name\_Model]),  constraint [FK\_Mark\_Model] foreign key ([Mark\_ID])  references [dbo].[Mark] ([ID\_Mark])  )  go | create or alter procedure [dbo].[Model\_insert]  @Name\_Model [varchar](15),  @Mark\_ID [int]  as  begin  insert into [dbo].[Model] (Name\_Model, Mark\_ID)  values (@Name\_Model, @Mark\_ID);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Model\_update]  @ID\_Model [int],  @Name\_Model [varchar](15),  @Mark\_ID [int]  as  begin  update [dbo].[Model]  set Name\_Model = @Name\_Model,  Mark\_ID = @Mark\_ID  where ID\_Model = @ID\_Model;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Model\_delete]  @ID\_Model [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Mark\_Model\_Inventory] where Model\_ID = @ID\_Model)  if (@any\_child\_record > 0)  begin  print('Модель не может быть удалена, так как в таблице "Mark\_Model\_Inventory", есть связанные данные!');  return;  end  delete from [dbo].[Model]  where ID\_Model = @ID\_Model;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Mark\_Model\_Inventory]  (  [ID\_Mark\_Model\_Inventory] [int] not null identity(1,1),  [Inventory\_ID] [int] not null,  [Model\_ID] [int] not null,  constraint [PK\_Mark\_Model\_Inventory] primary key clustered  ([ID\_Mark\_Model\_Inventory] ASC) on [PRIMARY],  constraint [FK\_Inventory\_Mark\_Model\_Inventory] foreign key ([Inventory\_ID])  references [dbo].[Inventory] ([ID\_Inventory]),  constraint [FK\_Model\_Mark\_Model\_Inventory] foreign key ([Model\_ID])  references [dbo].[Model] ([ID\_Model])  )  go | create or alter procedure [dbo].[Mark\_Model\_Inventory\_insert]  @Inventory\_ID [int],  @Model\_ID [int]  as  begin  insert into [dbo].[Mark\_Model\_Inventory] (Inventory\_ID, Model\_ID)  values (@Inventory\_ID, @Model\_ID);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Mark\_Model\_Inventory\_update]  @ID\_Mark\_Model\_Inventory [int],  @Inventory\_ID [int],  @Model\_ID [int]  as  begin  update [dbo].[Mark\_Model\_Inventory]  set Inventory\_ID = @Inventory\_ID,  Model\_ID = @Model\_ID  where ID\_Mark\_Model\_Inventory = @ID\_Mark\_Model\_Inventory;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Mark\_Model\_Inventory\_delete]  @ID\_Mark\_Model\_Inventory [int]  as  begin  delete from [dbo].[Mark\_Model\_Inventory]  where ID\_Mark\_Model\_Inventory = @ID\_Mark\_Model\_Inventory;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Suppliers]  (  [ID\_Supplier] [int] not null identity(1,1),  [Name\_Suppliers] [varchar](50) not null,  [OKPO\_Suppliers] [varchar](10) not null,  [Legal\_Address\_Suppliers] [varchar](max) not null,  [Physical\_Address\_Suppliers] [varchar](max) not null,  [Phone\_Number\_Suppliers] [varchar](16) not null,  constraint [PK\_Suppliers] primary key clustered  ([ID\_Supplier] ASC) on [PRIMARY],  constraint [UQ\_Name\_Suppliers] unique ([Name\_Suppliers]),  constraint [UQ\_OKPO\_Suppliers] unique ([OKPO\_Suppliers]),  constraint [UQ\_Phone\_Number\_Suppliers] unique ([Phone\_Number\_Suppliers]),  constraint [CH\_OKPO\_Suppliers] check ([OKPO\_Suppliers] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Phone\_Number\_Suppliers] check ([Phone\_Number\_Suppliers] like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]')  )  go | create or alter procedure [dbo].[Suppliers\_insert]  @Name\_Suppliers [varchar](50),  @OKPO\_Suppliers [varchar](10),  @Legal\_Address\_Suppliers [varchar](max),  @Physical\_Address\_Suppliers [varchar](max),  @Phone\_Number\_Suppliers [varchar](16)  as  begin  if (@OKPO\_Suppliers not like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')  begin  print('Неверный формат ОКПО!');  return;  end  if (@Phone\_Number\_Suppliers not like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]')  begin  print('Неверный формат номера телефона!');  return;  end  insert into [dbo].[Suppliers] (Name\_Suppliers, OKPO\_Suppliers, Legal\_Address\_Suppliers, Physical\_Address\_Suppliers, Phone\_Number\_Suppliers)  values (@Name\_Suppliers, @OKPO\_Suppliers, @Legal\_Address\_Suppliers, @Physical\_Address\_Suppliers, @Phone\_Number\_Suppliers);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Suppliers\_update]  @ID\_Supplier [int],  @Name\_Suppliers [varchar](50),  @OKPO\_Suppliers [varchar](10),  @Legal\_Address\_Suppliers [varchar](max),  @Physical\_Address\_Suppliers [varchar](max),  @Phone\_Number\_Suppliers [varchar](16)  as  begin  if (@OKPO\_Suppliers not like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')  begin  print('Неверный формат ОКПО!');  return;  end  if (@Phone\_Number\_Suppliers not like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]')  begin  print('Неверный формат номера телефона!');  return;  end  update [dbo].[Suppliers]  set Name\_Suppliers = @Name\_Suppliers,  OKPO\_Suppliers = @OKPO\_Suppliers,  Legal\_Address\_Suppliers = @Legal\_Address\_Suppliers,  Physical\_Address\_Suppliers = @Physical\_Address\_Suppliers,  Phone\_Number\_Suppliers = @Phone\_Number\_Suppliers  where ID\_Supplier = @ID\_Supplier;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Suppliers\_delete]  @ID\_Supplier [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Supplier\_Contracts] where Supplier\_ID = @ID\_Supplier)  if (@any\_child\_record > 0)  begin  print('Поставщик не может быть удален, так как в таблице "Supplier\_Contracts", есть связанные данные!');  return;  end  delete from [dbo].[Suppliers]  where ID\_Supplier = @ID\_Supplier;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Order\_Status]  (  [ID\_Order\_Status] [int] not null identity(1,1),  [Status\_Order\_Status] [varchar](8) not null,  constraint [PK\_Order\_Status] primary key clustered  ([ID\_Order\_Status] ASC) on [PRIMARY],  constraint [UQ\_Status\_Order\_Status] unique ([Status\_Order\_Status]),  constraint [CH\_Status\_Order\_Status] check ([Status\_Order\_Status] in ('открыт', 'завершён'))  )  go | create or alter procedure [dbo].[Order\_Status\_insert]  @Status\_Order\_Status [varchar](8)  as  begin  if (@Status\_Order\_Status not in ('открыт', 'завершён'))  begin  print('Статус заказа должен быть "открыт" или "завершён"!');  return;  end  insert into [dbo].[Order\_Status] (Status\_Order\_Status)  values (@Status\_Order\_Status);  end  go |
| Хранимая процедура на изменение данных |
| Не рационально |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Order\_Status\_delete]  @ID\_Order\_Status [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Supplier\_Contracts] where Order\_Status\_ID = @ID\_Order\_Status)  if (@any\_child\_record > 0)  begin  print('Статус заказа не может быть удален, так как в таблице "Supplier\_Contracts", есть связанные данные!');  return;  end  delete from [dbo].[Order\_Status]  where ID\_Order\_Status = @ID\_Order\_Status;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Supplier\_Contracts]  (  [ID\_Supplier\_Contracts] [int] not null identity(1,1),  [Supplier\_ID] [int] not null,  [Contract\_Number\_Supplier\_Contracts] [varchar](17) not null,  [Date\_Of\_Formation\_Supplier\_Contracts] [datetime] not null,  [Validity\_Period\_Supplier\_Contracts] [varchar](30) not null,  [Order\_Status\_ID] [int] not null,  constraint [PK\_Supplier\_Contracts] primary key clustered  ([ID\_Supplier\_Contracts] ASC) on [PRIMARY],  constraint [UQ\_Contract\_Number\_Supplier\_Contracts] unique ([Contract\_Number\_Supplier\_Contracts]),  constraint [CH\_Contract\_Number\_Supplier\_Contracts] check ([Contract\_Number\_Supplier\_Contracts] like 'Д-П/[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\_Date\_Of\_Formation\_Supplier\_Contracts] check ([Date\_Of\_Formation\_Supplier\_Contracts] <= getdate()),  constraint [CH\_Validity\_Period\_Supplier\_Contracts] check ([Validity\_Period\_Supplier\_Contracts] like '%[0-9]%'),  constraint [FK\_Supplier\_Supplier\_Contracts] foreign key ([Supplier\_ID])  references [dbo].[Suppliers] ([ID\_Supplier]),  constraint [FK\_Order\_Status\_Supplier\_Contracts] foreign key ([Order\_Status\_ID])  references [dbo].[Order\_Status] ([ID\_Order\_Status])  )  go | create or alter procedure [dbo].[Supplier\_Contracts\_insert]  @Supplier\_ID [int],  @Contract\_Number\_Supplier\_Contracts [varchar](17),  @Date\_Of\_Formation\_Supplier\_Contracts [datetime],  @Validity\_Period\_Supplier\_Contracts [varchar](30),  @Order\_Status\_ID [int]  as  begin  if (@Contract\_Number\_Supplier\_Contracts not like 'Д-П/[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]/[0-9][0-9]')  begin  print('Неверный формат номера контракта поставщика!');  return;  end  if (@Date\_Of\_Formation\_Supplier\_Contracts > getdate())  begin  print('Дата формирования контракта не может быть в будущем!');  return;  end  if (@Validity\_Period\_Supplier\_Contracts not like '%[0-9]%')  begin  print('Срок действия контракта должен содержать цифры!');  return;  end  insert into [dbo].[Supplier\_Contracts] (Supplier\_ID, Contract\_Number\_Supplier\_Contracts, Date\_Of\_Formation\_Supplier\_Contracts, Validity\_Period\_Supplier\_Contracts, Order\_Status\_ID)  values (@Supplier\_ID, @Contract\_Number\_Supplier\_Contracts, @Date\_Of\_Formation\_Supplier\_Contracts, @Validity\_Period\_Supplier\_Contracts, @Order\_Status\_ID);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Supplier\_Contracts\_update]  @ID\_Supplier\_Contracts [int],  @Supplier\_ID [int],  @Contract\_Number\_Supplier\_Contracts [varchar](17),  @Date\_Of\_Formation\_Supplier\_Contracts [datetime],  @Validity\_Period\_Supplier\_Contracts [varchar](30),  @Order\_Status\_ID [int]  as  begin  if (@Contract\_Number\_Supplier\_Contracts not like 'Д-П/[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]/[0-9][0-9]')  begin  print('Неверный формат номера контракта поставщика!');  return;  end  if (@Date\_Of\_Formation\_Supplier\_Contracts > getdate())  begin  print('Дата формирования контракта не может быть в будущем!');  return;  end  if (@Validity\_Period\_Supplier\_Contracts not like '%[0-9]%')  begin  print('Срок действия контракта должен содержать цифры!');  return;  end  update [dbo].[Supplier\_Contracts]  set Supplier\_ID = @Supplier\_ID,  Contract\_Number\_Supplier\_Contracts = @Contract\_Number\_Supplier\_Contracts,  Date\_Of\_Formation\_Supplier\_Contracts = @Date\_Of\_Formation\_Supplier\_Contracts,  Validity\_Period\_Supplier\_Contracts = @Validity\_Period\_Supplier\_Contracts,  Order\_Status\_ID = @Order\_Status\_ID  where ID\_Supplier\_Contracts = @ID\_Supplier\_Contracts;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Supplier\_Contracts\_delete]  @ID\_Supplier\_Contracts [int]  as  begin  declare @any\_child\_record [int] = (select count(\*) from [dbo].[Inventory\_Supplier\_Contracts] where Supplier\_Contracts\_ID = @ID\_Supplier\_Contracts)  if (@any\_child\_record > 0)  begin  print('Контракт поставщика не может быть удален, так как в таблице "Inventory\_Supplier\_Contracts", есть связанные данные!');  return;  end  delete from [dbo].[Supplier\_Contracts]  where ID\_Supplier\_Contracts = @ID\_Supplier\_Contracts;  end  go |
| Скрипт таблицы | Хранимая процедура на добавление данных |
| create table [dbo].[Inventory\_Supplier\_Contracts]  (  [ID\_Inventory\_Supplier\_Contracts] [int] not null identity(1,1),  [Inventory\_ID] [int] not null,  [Supplier\_Contracts\_ID] [int] not null,  [Count\_Inventory\_Supplier\_Contracts] [int] not null,  constraint [PK\_Inventory\_Supplier\_Contracts] primary key clustered  ([ID\_Inventory\_Supplier\_Contracts] ASC) on [PRIMARY],  constraint [CH\_Count\_Inventory\_Supplier\_Contracts] check ([Count\_Inventory\_Supplier\_Contracts] > 0),  constraint [FK\_Inventory\_Inventory\_Supplier\_Contracts] foreign key ([Inventory\_ID])  references [dbo].[Inventory] ([ID\_Inventory]),  constraint [FK\_Supplier\_Contracts\_Inventory\_Supplier\_Contracts] foreign key ([Supplier\_Contracts\_ID])  references [dbo].[Supplier\_Contracts] ([ID\_Supplier\_Contracts])  )  go | create or alter procedure [dbo].[Inventory\_Supplier\_Contracts\_insert]  @Inventory\_ID [int],  @Supplier\_Contracts\_ID [int],  @Count\_Inventory\_Supplier\_Contracts [int]  as  begin  if (@Count\_Inventory\_Supplier\_Contracts <= 0)  begin  print('Количество инвентаря должно быть больше 0!');  return;  end  insert into [dbo].[Inventory\_Supplier\_Contracts] (Inventory\_ID, Supplier\_Contracts\_ID, Count\_Inventory\_Supplier\_Contracts)  values (@Inventory\_ID, @Supplier\_Contracts\_ID, @Count\_Inventory\_Supplier\_Contracts);  end  go |
| Хранимая процедура на изменение данных |
| create or alter procedure [dbo].[Inventory\_Supplier\_Contracts\_update]  @ID\_Inventory\_Supplier\_Contracts [int],  @Inventory\_ID [int],  @Supplier\_Contracts\_ID [int],  @Count\_Inventory\_Supplier\_Contracts [int]  as  begin  if (@Count\_Inventory\_Supplier\_Contracts <= 0)  begin  print('Количество инвентаря должно быть больше 0!');  return;  end  update [dbo].[Inventory\_Supplier\_Contracts]  set Inventory\_ID = @Inventory\_ID,  Supplier\_Contracts\_ID = @Supplier\_Contracts\_ID,  Count\_Inventory\_Supplier\_Contracts = @Count\_Inventory\_Supplier\_Contracts  where ID\_Inventory\_Supplier\_Contracts = @ID\_Inventory\_Supplier\_Contracts;  end  go |
| Хранимая процедура на удаление данных |
| create or alter procedure [dbo].[Inventory\_Supplier\_Contracts\_delete]  @ID\_Inventory\_Supplier\_Contracts [int]  as  begin  delete from [dbo].[Inventory\_Supplier\_Contracts]  where ID\_Inventory\_Supplier\_Contracts = @ID\_Inventory\_Supplier\_Contracts;  end  go |

1. Иллюстрация обозревателя объекта

