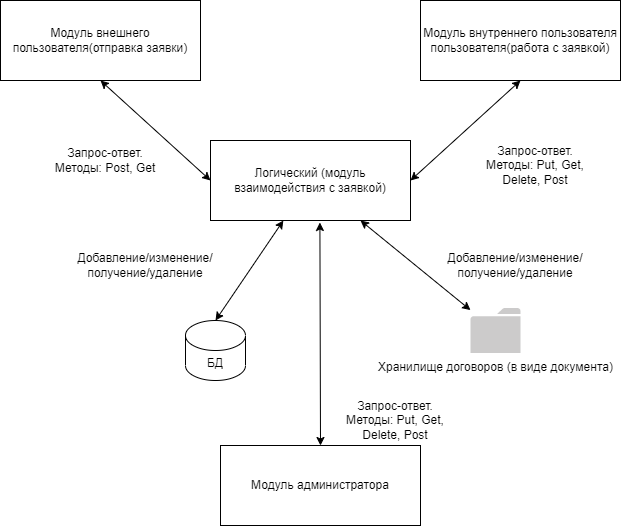
Архитектурная схема модулей приложения



Состав таблиц базы данных

Примечание.

1.Так как не известно вероятное число знаков у атрибутов со строковыми значениями, принимаем число знаков за максимальноe.

2. При заполнении формы предлагается сделать все поля обязательными, для этого необходимо организовать валидацию форм ввода.

3. Так как не уточнено бизнес-требования касаемо даты и времени, формат даты и времени принимаем за datetime2(7).

4. Стоимость договоров принята за целое числа, так как не известно будут ли использоваться десятичные числа.

5. Заказчиков и исполнителей можно реализовать в формате ФИО.

Связь сущностей, а также их атрибуты представлены в виде диаграммы базы данных.

**Заявки(Requests)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| ID | Guid | Идентификатор 128 бит | да | Первичный ключ |
| Number | nvarchar(max) | Строки символов переменной длины | да | Номер заявки |
| RegistrationDate | datetime2(7) | Дата и время | да | Дата регистрации |
| SourceID | Guid | Идентификатор 128 бит | да | Внешний ключ к таблице «Источник» |
| ActivityTypeID | Guid | Идентификатор 128 бит | да | Внешний ключ к таблице «Вид работ» |
| CustomerID | Guid | Идентификатор 128 бит | да | Внешний ключ к таблице «Заказчик» |
| ObjectName | nvarchar(max) | Строки символов переменной длины | да | Имя объекта |
| ObjectAddress | nvarchar(max) | Строки символов переменной длины | да | Адрес объекта |
| ObjectTypeID | Guid | Идентификатор 128 бит | да | Внешний ключ к таблице «Тип объекта» |
| RequestStatusID | Guid | Идентификатор 128 бит | да | Внешний ключ к таблице «Статус заявки» |
| ContractID | Guid | Идентификатор 128 бит | да | Внешний ключ к таблице «Договор» |

**Договора(Contracts)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| ID | Guid | Идентификатор 128 бит | да | Первичный ключ |
| Number | nvarchar(max) | Строки символов переменной длины | да | Номер |
| DischargeDate | datetime2(7) | Дата и время | да | Дата выписки |
| ContractSubjectID | Guid | Идентификатор 128 бит | да | Внешний ключ к таблице «Предмет договора» |
| ContractAmount | Int | Целые числа | да | Сумма договора |
| StartDate | datetime2(7) | Дата и время | да | Дата начала договора |
| EndDate | datetime2(7) | Дата и время | да | Дата окончания договора |
| Specification | nvarchar(max) | Строки символов переменной длины | да | Описание |

**Предметы договора(ContractSubjects)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| ID | Guid | Идентификатор 128 бит | да | Первичный ключ |
| Value | nvarchar(max) | Строки символов переменной длины | да | Значение |

**Статусы заявки (RequestStatuses)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| ID | Guid | Идентификатор 128 бит | да | Первичный ключ |
| Value | nvarchar(max) | Строки символов переменной длины | да | Значение |

**Типы объектов (ObjectTypes)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| ID | Guid | Идентификатор 128 бит | да | Первичный ключ |
| Value | nvarchar(max) | Строки символов переменной длины | да | Значение |

**Заказчики (Customets)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| ID | Guid | Идентификатор 128 бит | да | Первичный ключ |
| Name | nvarchar(max) | Строки символов переменной длины | да | Имя |

**Виды работ (ActivityTypes)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| ID | Guid | Идентификатор 128 бит | да | Первичный ключ |
| Value | nvarchar(max) | Строки символов переменной длины | да | Значение |

**Источники (Sources)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| ID | Guid | Идентификатор 128 бит | да | Первичный ключ |
| Value | nvarchar(max) | Строки символов переменной длины | да | Значение |

**Исполнители (Executors)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| ID | Guid | Идентификатор 128 бит | да | Первичный ключ |
| Name | nvarchar(max) | Строки символов переменной длины | да | Имя |

**ЗаявкиИсполнители (RequestsExecutors)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| RequestID | Guid | Идентификатор 128 бит | да | Внешний ключ к таблице «Заявки» |
| ExecutorID | Guid | Идентификатор 128 бит | да | Внешний ключ к таблице «Исполнители» |
| RequestID, ExecutorID |  |  |  | Составной внешний ключ |

**Стороны (Sides)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| ID | Guid | Идентификатор 128 бит | да | Первичный ключ |
| Value | nvarchar(max) | Строки символов переменной длины | да | Значение |

**ЗаказчикиДоговора (ContractsSides)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Тип данных** | **Формат** | **Обязательное** | **Особые отметки** |
| SideID | Guid | Идентификатор 128 бит | да | Внешний ключ к таблице «Стороны» |
| ContractID | Guid | Идентификатор 128 бит | да | Внешний ключ к таблице «Договора» |
| SideID,ContractID |  |  |  | Составной внешний ключ |

Скрипт для создания таблиц в базе данных.

Use Task

--Исполнители. Можно добавить в формате ФИО (First Name, Patronymic, FamilyName)

Create Table dbo.Executors(

ID UNIQUEIDENTIFIER Primary key default NEWID(),

Name nvarchar(max) Not Null,

)

--Источники

Create Table dbo.Sources(

ID UNIQUEIDENTIFIER Primary key default NEWID(),

Value nvarchar(max) Not Null,)

--Виды работ

Create Table dbo.ActivityTypes(

ID UNIQUEIDENTIFIER Primary key default NEWID(),

Value nvarchar(max) Not Null,)

--Заказчики Можно добавить в формате ФИО (First Name, Patronymic, FamilyName)

--Так же, вероятно является стороной в договоре.

Create Table dbo.Customers(

ID UNIQUEIDENTIFIER Primary key default NEWID(),

Name nvarchar(max) Not Null,)

--Типы объекта

Create Table dbo.ObjectTypes(

ID UNIQUEIDENTIFIER Primary key default NEWID(),

Value nvarchar(max) Not Null,)

--Статусы заявки

Create Table dbo.RequestStatuses(

ID UNIQUEIDENTIFIER Primary key default NEWID(),

Value nvarchar(max) Not Null,)

--Предметы договора

Create Table dbo.ContractSubjects(

ID UNIQUEIDENTIFIER Primary key default NEWID(),

Value nvarchar(max) Not Null,)

--Договора

Create Table dbo.Contracts(

ID UNIQUEIDENTIFIER Primary key default NEWID(),

Number nvarchar(max) Not Null, --Номер может быть не только цифровой

DischargeDate datetime2(7) Not Null, --дата выписки

ContractSubjectID Uniqueidentifier Foreign key references dbo.ContractSubjects(ID) Not Null, --one to many предмет договора

ContractAmount int Not Null, --Сумма договора

StartDate datetime2(7) Not Null, -- Начало действия

EndDate datetime2(7) Not Null, --Окончание действия

Specification nvarchar(max) Not Null, --Описание

)

--Заявки

Create Table dbo.Requests(

ID UNIQUEIDENTIFIER Primary key default NEWID(),

Number nvarchar(max) Not Null, --Номер может быть не только цифровой

RegistrationDate datetime2(7) Not Null,

SourceID Uniqueidentifier Foreign key references dbo.Sources(ID) Not Null, --one to many

ActivityTypeID Uniqueidentifier Foreign key references dbo.ActivityTypes(ID) Not Null, --one to many

CustomerID Uniqueidentifier Foreign key references dbo.Customers(ID) Not Null, --one to many

ObjectName nvarchar(max) Not Null,

ObjectAddress nvarchar(max) Not Null,

ObjectTypeID Uniqueidentifier Foreign key references dbo.ObjectTypes(ID) Not Null, --one to many

RequestStatusID Uniqueidentifier Foreign key references dbo.RequestStatuses(ID) Not Null, --one to many

ContractID Uniqueidentifier Foreign key references dbo.Contracts(ID) Not Null, --one to many

)

--Промежуточная таблица связь для заявки и исполнителя many to many

Create Table dbo.RequestsExecutors(

RequestID Uniqueidentifier Foreign key references dbo.Requests(ID) Not Null,

ExecutorID Uniqueidentifier Foreign key references dbo.Executors(ID) Not Null,

Primary key (RequestID,ExecutorID)

)

--Стороны

Create Table dbo.Sides(

ID UNIQUEIDENTIFIER Primary key default NEWID(),

Value nvarchar(max) Not Null,

)

--Many to Many

--Стороны. Промежуточная таблица связь для договора и сторонц many to many

Create Table dbo.ContractsSides(

SideID Uniqueidentifier Foreign key references dbo.Sides(ID) Not Null,

ContractID Uniqueidentifier Foreign key references dbo.Contracts(ID) Not Null,

Primary key (SideID,ContractID)

)

**Скрип на получение запроса из 3 пункта**

Use Task

Select Requests.Number as "Номер заявки", Requests.RegistrationDate as "Дата регистрации", Requests.ObjectName as "Имя объекта",

Customers.Name as "Заказчик", RequestStatuses.Value as "Статус заявки",

Contracts.Number as "Номер договора", Contracts.DischargeDate as "Дата выписки договора",

Executors.Name as "Исполнители"

From Requests, Customers, RequestStatuses, Contracts, Executors, RequestsExecutors

Where Requests.CustomerID=Customers.ID AND Requests.RequestStatusID=RequestStatuses.ID

AND Requests.ContractID=Contracts.ID

AND Requests.ID=RequestsExecutors.RequestID AND Executors.ID=RequestsExecutors.ExecutorID

**Скрип на получение запроса из 4 пункта**

Use Task

Select Count(\*) as "Количество заявок"

From Requests r

Where Exists (Select ID From Contracts c

Where r.ContractID=c.ID And c.EndDate>=GETDATE())

**Скрип на получение запроса из 5 пункта**

Use Task

Select SUM(c.ContractAmount) as "Сумма контрактов"

From Contracts c

Where Exists (Select ID From Requests r

Where r.ContractID=c.ID And r.RegistrationDate<=GETDATE())

Для работы внешнего пользователя с формами заявки предлагается создать контроллеры с методами GET и POST.

Для работы внутреннего пользователя с заявкой предлагается создать контроллеры с методами GET, PUT, DELETE (для заявки) и GET, PUT, POST, DELETE (для договора).

Для работы администратора так же необходимо создать контроллеры с методами GET, PUT, DELETE, POST для управления заявками, договорами и пользователями.

Состав классов определяется исходя из выбора стэка разработки.

К примеру, в ORM Entity Framework логический модуль работы с заявкой и договором можно реализовать с помощью следующих основных классов:

-Модели данных;

-DTO (объекты преобразования моделей данных);

-Сервисы (логика работы);

-Интерфейсы;

-Контроллеры (вызов интерфейсов);