

Курсовая работа по предмету
"Распределенные информационные системы"
на тему
"Проектирование инфологической
и концептуальной моделей"
Вариант 6

Выполнил: студент гр. 4381

Иванов Д.В. _____

Проверил:

Вдовичев Н.М. _____

Казань 2018 г.

Оглавление

1. Задание	3
2. Проектирование инфологической модели предметной области	3
2.1. Составление перечня атрибутов предметной области.....	3
2.2. Составление перечня атрибутов предметной области	3
2.2.1 Агрегация атрибутов в сущности.....	5
2.2.2 Определение первичных ключей	6
2.2.3 Нормализация сущностей	6
1.2.4. Внешнее кодирование.....	7
1.2.5. Графическое представление сущностей	7
1.3. Определение связей.	7
2.3.1. Графическое изображение инфологической модели предметной области.	8
3. Проектирования концептуальной модели	9
3.1. Преобразование сущностей в отношения.	9
3.2. Определение свойств атрибутов отношений	10
3.3. Реализация связей типа «многие-ко-многим»	11
3.3. Определение внешних ключей	11
3.5. Графическое изображение концептуальной модели предметной области	15
4. Состав и связь интерфейсов интерактивного взаимодействия	16
4.1. Главная страница базы данных льгот и их потребителей.....	18
4.2. Форма «Document».....	15
4.3. Форма «LgotaVid»	19
4.4. Форма «Lgota»	16
4.5. Форма «Lgotnik».....	18
3.5. Заключение.....	19
Приложение 1. Листинг скрипта создания базы данных.....	20
Приложение 2. Листинг кода приложения.	26

Задание

Вариант №6.

База данных нормативных документов по льготам содержит следующую информацию:

- наименование законодательного документа;
- дата начала и конца действия нормативного документа;
- категории граждан, группа инвалидности (в случае инвалидности), место боевых действий (в случае участия в боевых действиях);
- вид и размер льготы;
- в случае жилищно-коммунальной услуги - наименование, процент льготы, дата начала и конца действия льготы;
- источник финансирования (бюджет РФ и так далее);

Требуется выполнить следующие запросы:

- 1) каков нормативный документ, в котором описываются указанные пользователем льготы для указанной категории граждан;
- 2) выбрать все категории граждан и льготы, указанные в заданном нормативном документе и подсчитать их;
- 3) выбрать все льготы, указанные в заданном нормативном документе, для заданной категории граждан, которые действовали в указанный период времени;
- 4) определить перечень категорий граждан, для которых действуют максимальные жилищно-коммунальные льготы;

Задание: для выбранной предметной области спроектировать в соответствии с изложенными методиками инфологическую и концептуальную модели.

2. Проектирование инфологической модели предметной области.

2.1 Составление перечня атрибутов предметной области

Составим первоначальный перечень атрибутов:

1. наименование законодательного документа
2. дата начала действия нормативного документа
3. дата конца действия нормативного документа
4. **категории граждан**
5. **группа инвалидности (в случае инвалидности)**
6. **место боевых действий (в случае участия в боевых действиях)**
7. *вид льготы*
8. *размер льготы*
9. наименование льготы ЖКХ
10. процент льготы ЖКХ

11. дата начала льготы ЖКХ
12. дата конца льготы ЖКХ
13. источник финансирования

Исключим все синонимы, это *вид льготы* и *наименование льготы ЖКХ*, а также *размер льготы* и *процент льготы ЖКХ* (по проценту можно вычислить размер) Атрибутов ононимов в данном списке не найдено.

Обозначение атрибута	Имя атрибута	Примечание
doc_name	наименование законодательного документа	1 и более слов
doc_data_begin	дата начала действия нормативного документа	дата
doc_data_end	дата конца действия нормативного документа	дата
category	категории граждан	целое без знака
inv_group	группа инвалидности (в случае инвалидности)	целое без знака (м.б. NULL)
war_place	место боевых действий (в случае участия в боевых действиях)	1 и более слов (м.б. NULL)
lgota_name	вид льготы	1 и более слов (м.б. NULL)
lgota_proc	процент льготы	целое без знака (м.б. NULL)
lgota_data_begin_JKH	дата начала льготы ЖКХ	дата (м.б. NULL)
lgota_data_end_JKH	дата конца льготы ЖКХ	дата (м.б. NULL)
finance	источник финансирования	1 и более слов

Таблица 1.1. Перечень атрибутов предметной области

2.2 Определение сущностей

2.2.1 Агрегация атрибутов в сущности

Шаг 1.

Атрибуты **doc_name** и **doc_data_begin** имеют тип соответствия 1:1 - объединим их в сущность **DOCUMENT { doc_name , doc_data_begin }**

Шаг 2.

Атрибут **doc_data_end** и **finance** относится к сущности **DOCUMENT** как 1:1 - включаем его в сущность **DOCUMENT{ doc_name , doc_data_begin, doc_data_end, finance }**

Атрибут **lgota_name** относится к сущности **DOCUMENT** как M:1 (в документе будет описываться много льгот) - вынесем атрибут **lgota** в отдельную сущность **VID_LGOTI { lgota_name }**. Атрибут **lgota_proc** относится к сущности **VID_LGOTI** 1:1:

VID_LGOTI { lgota_name, lgota_proc, lgota_data_begin_JKH, lgota_data_end_JKH }

Атрибут **category** относится к обоим сущностям как M:1 - поэтому выделим его в отдельную сущность **GRAJDANIN { category }**. Атрибуты **inv_group** и **war_place** относятся к сущности **GRAJDANIN** как 1:1:

GRAJDANIN { category, inv_group, war_place }

Обозначение сущности	Имя сущности	Список атрибутов
DOCUMENT	Документ	1. наименование законодательного документа (doc_name) 2. дата начала действия нормативного документа (doc_data_begin) 3. дата конца действия нормативного документа(doc_data_end) 4. источник финансирования (finance)
VID_LGOTI	Вид льготы	1. вид льготы (lgota_name) 2. процент льготы (lgota_proc) 3. дата начала льготы ЖКХ (lgota_data_begin) 4. дата конца льготы ЖКХ (lgota_data_end_JKH)
GRAJDANIN	Гражданин	1. категория гражданина (category) 2. группа инвалидности (в случае инвалидности) (inv_group) 3. место боевых действий (в случае участия в боевых действиях)(war_place)

2.2.2 Определение первичных ключей

Сущность	Первичный ключ	Примечание
Документ	наименование законодательного документа	
Вид льготы	вид льготы	
Гражданин	номер паспорта	НОВЫЙ

2.2.3 Нормализация сущностей

Сущности уже находятся в первой нормальной форме.(были преведены в п. 1.1):

-дата начала и конца действия нормативного документа:

-дата начала

-дата конца

- начала и конца действия льготы:

-дата начала

-дата конца

Остальные сущности не имеют сложных атрибутов.

Полученные отношения уже находятся во 2 нормальной форме, т.к. каждый неключевой

атрибут функционально полно зависит от составного ключа. Приведём отношения к 3 НФ:

Обозначение сущности	Имя сущности	Список атрибутов
DOCUMENT	Документ	1 . наименование законодательного документа (doc_name) 2. дата начала действия нормативного документа (doc_data_begin) 3. дата конца действия нормативного документа(doc_data_end) 4. источник финансирования (finance)
VID_LGOTI	Вид льготы	1. вид льготы (lgota_name) 2. процент льготы (lgota_proc) 3. дата начала льготы ЖКХ (lgota_data_begin) 4. дата конца льготы ЖКХ (lgota_data_end_JKH)
GRAJDANIN	Гражданин	1. номер паспорта(passport) 2. категория гражданина (category) 3. группа инвалидности (в случае инвалидности) (inv_group) 4. место боевых действий (в случае участия в боевых действиях)(war_place)

2.2.4. Внешнее кодирование

Во внешнем кодировании нет нужды.

2.2.5. Графическое представление сущностей

Документ
1. наименование законодательного документа (doc_name)
2. дата начала действия нормативного документа (doc_data_begin)
3. дата конца действия нормативного документа (doc_data_end)
4. источник финансирования (finance)

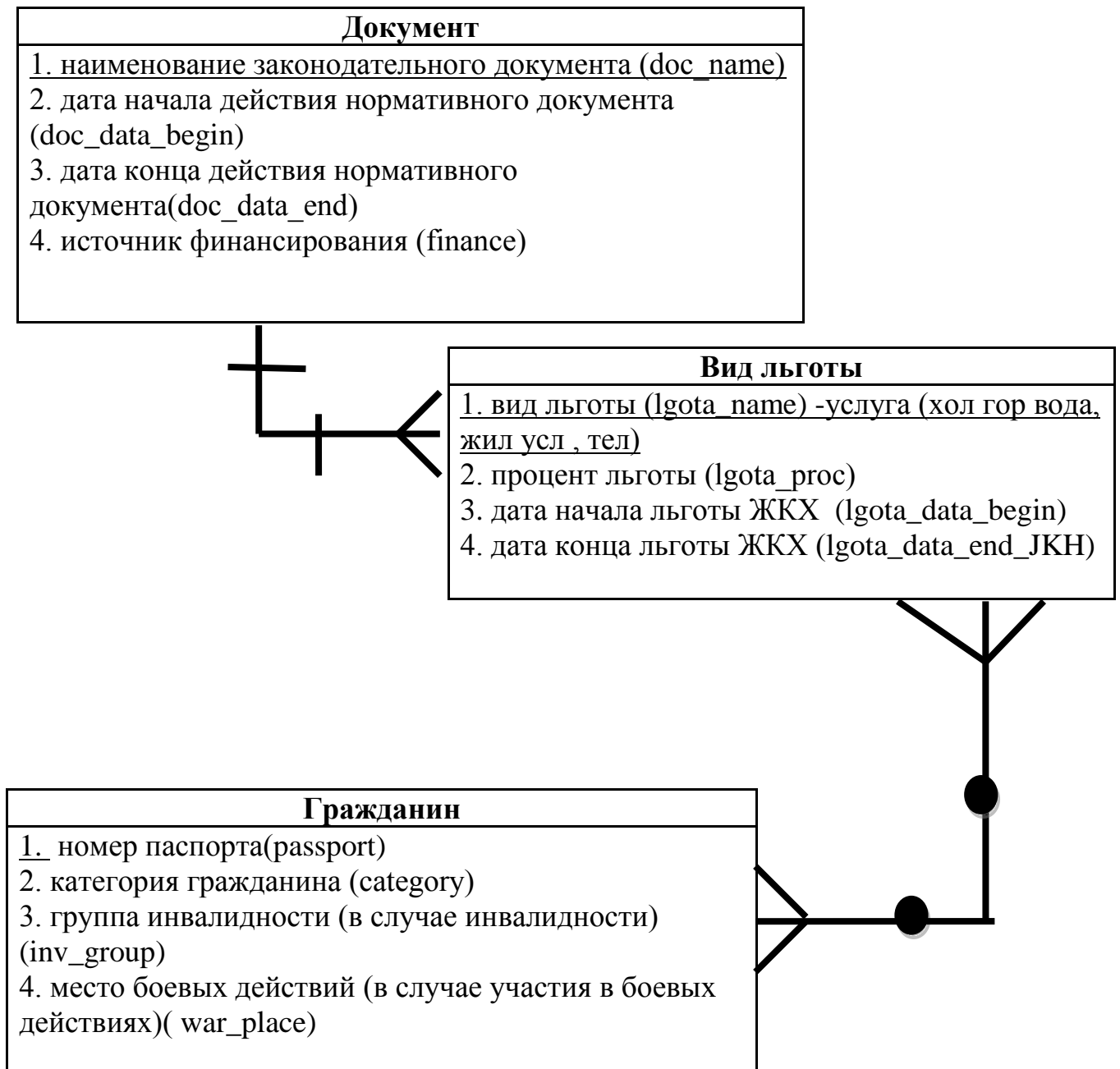
Вид льготы
1. вид льготы (lgota_name)
2. процент льготы (lgota_proc)
3. дата начала льготы ЖКХ (lgota_data_begin)
4. дата конца льготы ЖКХ (lgota_data_end_JKH)

Гражданин
1. номер паспорта (passport)
2. категория гражданина (category)
3. группа инвалидности (в случае инвалидности) (inv_group)
4. место боевых действий (в случае участия в боевых действиях) (war_place)

2.3. Определение связей

Сущность 1	Сущность 2	Тип связи	Обязательность
Документ	Вид льготы	1:M	1-1
Гражданин	Вид льготы	M:M	0-0

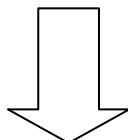
2.3.1. Графическое изображение инфологической модели предметной области



3. Проектирования концептуальной модели

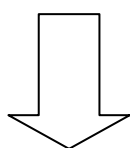
3.1. Преобразование сущностей в отношения.

Документ
1. <u>наименование законодательного документа</u>
2. дата начала действия нормативного документа
3. дата конца действия нормативного документа
4. источник финансирования



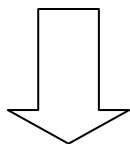
DOCUMENT
1. <u>doc_name</u>
2. doc_data_begin
3. doc_data_end
4. finance

Вид льготы
1. <u>вид льготы (lgota_name)</u>
2. процент льготы (lgota_proc)
3. дата начала льготы ЖКХ (lgota_data_begin)
4. дата конца льготы ЖКХ (lgota_data_end_JKH)



VID_LGOTI
1. <u>lgota_name</u>
2. lgota_proc
3. lgota_data_begin
4. lgota_data_end_JKH

Гражданин
1. номер паспорта(passport) 2. категория гражданина (category) 3. группа инвалидности (в случае инвалидности) (inv_group) 4. место боевых действий (в случае участия в боевых действиях)(war_place)



GRAJDANIN
1. passport 2. category 3. inv_group 4. war_place

3.2. *Определение свойств атрибутов отношений*

Свойства атрибутов отношения **DOCUMENT**:

Атрибут	Тип (размер)	Null
1. <u>doc_name</u>	Varchar(256)	Not Null
2. doc_data_begin	Date	Null
3. doc_data_end	Date	Null
4. finance	Varchar(256)	Null

Свойства атрибутов отношения **VID_LGOTI**:

Атрибут	Тип (размер)	Null
1. <u>lgota_name</u>	Varchar(256)	Not Null
2. lgota_proc	Int	Null
3. lgota_data_begin	Date	Null
4. lgota_data_end_JKH	Date	Null

Свойства атрибутов отношения **GRAJDANIN**:

Атрибут	Тип (размер)	Null
1. passport	Int	Not Null
2. category	Int	Null
3. inv_group	Int	Null
4. war_place	Varchar(256)	Null

3.3. Реализация связей типа «многие-ко-многим»

Рассмотрим отношение M:M (GRAJDANIN- VID_LGOTI):

введем новое отношение Льгота (Lgota)

Введем два новых атрибута: lgota_name и passport . Эти атрибуты являются внешними ключами, ссылающимися на первичные ключи связываемых отношений.

Оба внешних ключа объявляются *обязательными* («NOT NULL») и в совокупности образуют *первичный ключ* отношения связи.

Имя атрибута	Тип (размер)	NULL	Ссылка
passport	Int	Not Null	GRAJDANIN :passport
lgota_name	Varchar(256)	Not Null	VID_LGOTI : <u>lgota_name</u>

3.3. Определение внешних ключей

Правило целостности внешних ключей. Внешние ключи не должны быть несогласованными, т.е. для каждого значения внешнего ключа должно существовать соответствующее значение ключевого атрибута в родительском отношении

Рассмотрим отношение 1:M (DOCUMENT - VID_LGOTI):

Тут дочерним отношением будет VID_LGOTI. Среди атрибутов дочернего отношения нет атрибутов, ссылающийся на значения первичного ключа родительской таблицы. Добавим в дочернее отношение внешний ключ , ссылающийся на первичный ключ сущности DOCUMENT: **doc_name_**

Т.к. дочернее отношение является обязательным участником связи (если есть документ, то и должны быть льготы, описанные в нём) - то внешний ключ NOT NULL.

Рассмотрим отношение 1:M (VID_LGOTI-LGOTA):

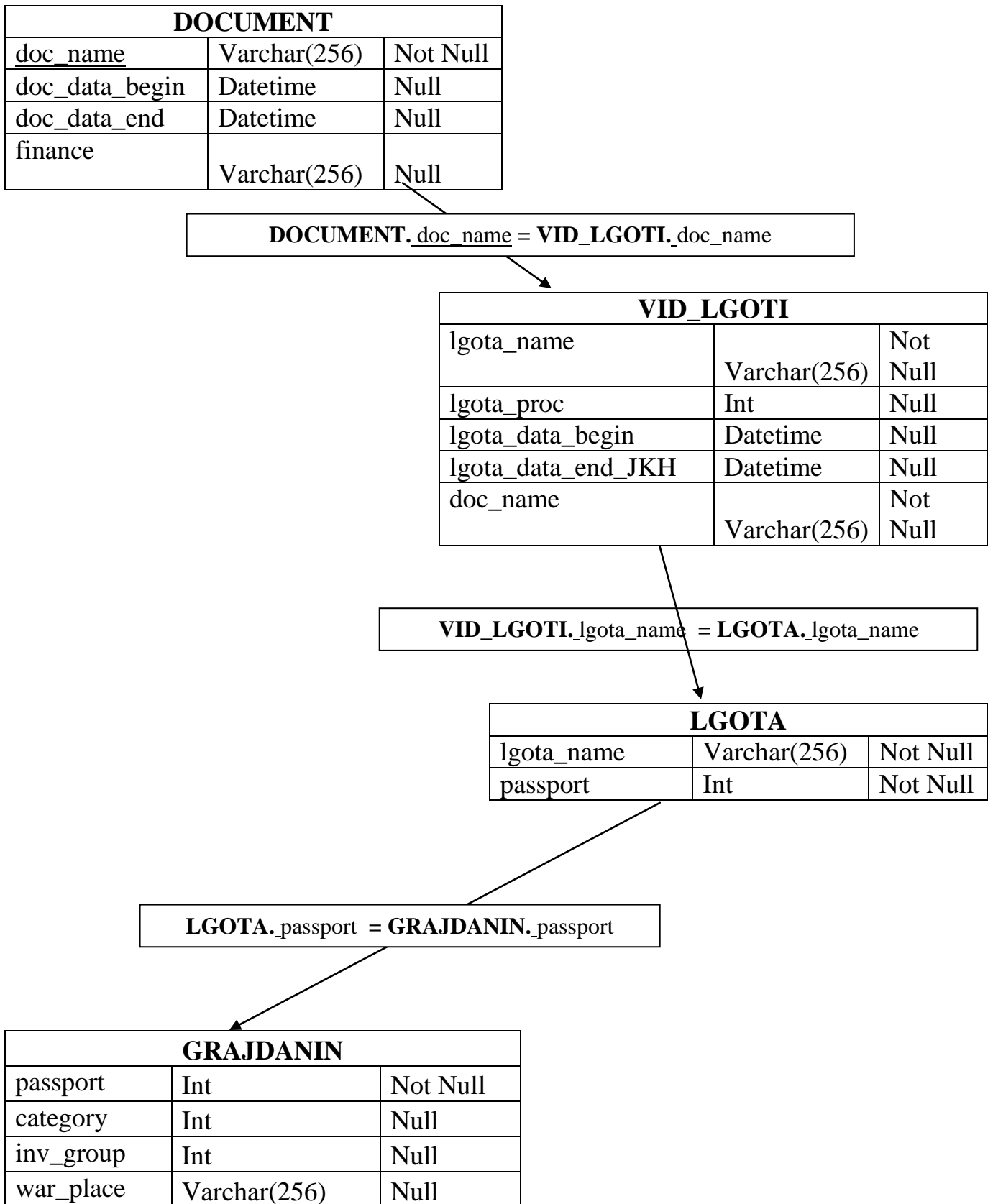
Тут дочерним отношением будет LGOTA. Среди атрибутов дочернего отношения есть атрибут LGOTA: **lgota_name** , ссылающийся на значения первичного ключа родительской таблицы.

Рассмотрим отношение 1:M (LGOTA-GRAJDANIN):

Тут дочерним отношением будет GRAJDANIN. Среди атрибутов дочернего отношения есть атрибут GRAJDANIN: **passport** , ссылающийся на значения первичного ключа родительской таблицы.

Дочернее отношение	Внешний ключ	Тип (размер)	NULL	Ссылка	Примечание
VID_LGOTI	doc_name	Varchar(256)	NOT NULL	DOCUMENT: doc_name	ДОБАВЛЕНИЕ
LGOTA	lgota_name	Varchar(256)	NOT NULL	VID_LGOTI: lgota_name	ДОБАВЛЕНИЕ
GRAJDANIN	passport	Int	NOT NULL	LGOTA: passport	ДОБАВЛЕНИЕ

3.5. Графическое изображение концептуальной модели предметной области



4. Состав и связь интерфейсов интерактивного взаимодействия.

4.1. Главная страница базы данных льгот и их потребителей.

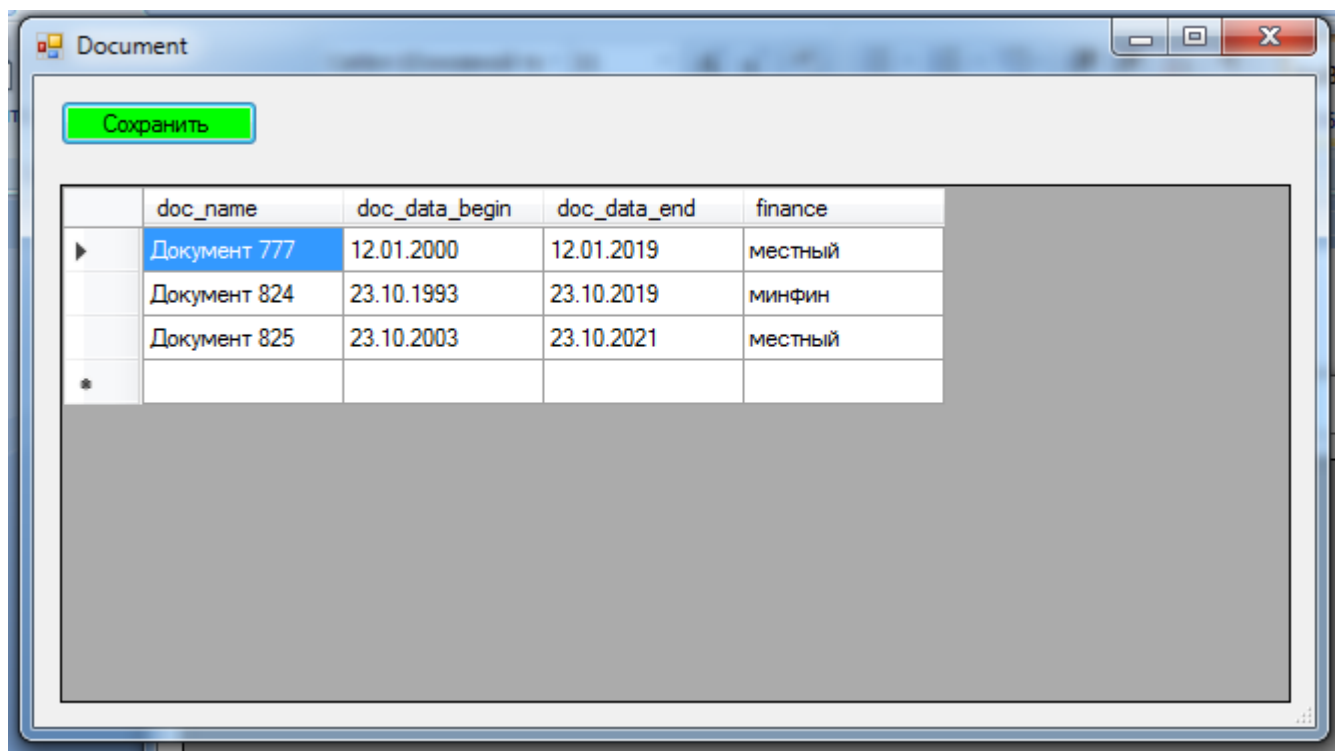
На главной форме (Рисунок 1.1.) размещены элементы управления, наиболее связанные с задачей курсовой работы..

Рисунок 1.1. Главная форма.

В этой форме можно создавать сложные запросы к удаленной БД., результаты которых будут выводиться на экран. Также там расположена кнопка "Заполнить таблицы" , при нажатии на которую происходит выполнение скрипта, заполняющего таблицы.

4.2. Форма «Document»

Экранная форма «Document» (Рисунок 1.2.) открывается по клику на кнопку "Документы"



	doc_name	doc_data_begin	doc_data_end	finance
▶	Документ 777	12.01.2000	12.01.2019	местный
	Документ 824	23.10.1993	23.10.2019	минфин
	Документ 825	23.10.2003	23.10.2021	местный
*				

Рисунок 1.2. Форма «Document»

В этой форме можно добавлять новые записи и сохранять их в таблицу на локальном сервере MS SQL.

4.3. Форма «LgotaVid»

Форма «LgotaVid» (Рисунок 1.3.) вызывается по клику на кнопку "Виды льгот" , расположенной в главной форме. В этой форме можно также добавлять и сохранять новые записи на удаленном сервере БД.

	lgota_name	lgota_proc	lgota_data_begin	lgota_data_end_JK	doc_name
▶	Другие	12	12.02.2000	12.03.2020	Документ 825
	ЖКХ	20	23.10.1995	23.10.2019	Документ 824
	Здравоохранение	50	23.10.2003	23.10.2021	Документ 825
	Проезд	5	23.10.2003	23.10.2021	Документ 825
	Телефон	17	23.10.2003	23.10.2019	Документ 824
*					

Рисунок 1.3. Форма « LgotaVid »

4.4. Форма «Lgota»

Форма «Lgota» (Рисунок 1.4.) вызывается по клику на кнопку "Льготники" , расположенной в главной форме. В этой форме можно также добавлять и сохранять новые записи на удаленном сервере БД.

	passport	lgota_name
▶	789465678	ЖКХ
	789475678	ЖКХ
	789495670	ЖКХ
	789495678	ЖКХ
	879555780	Здравоохранение
	879945780	Здравоохранение
	889045780	ЖКХ
	889945680	Проезд
	889945780	Телефон
	995945780	Другие
	999945780	Здравоохранение

Рисунок 1.4. Форма «Lgota»

4.5. Форма «Lgotnik»

Форма «Льготники» (Рисунок 1.5.) вызывается по клику на кнопку "Льготники" , расположенной в главной форме. В этой форме можно также добавлять и сохранять новые записи на удаленном сервере БД.

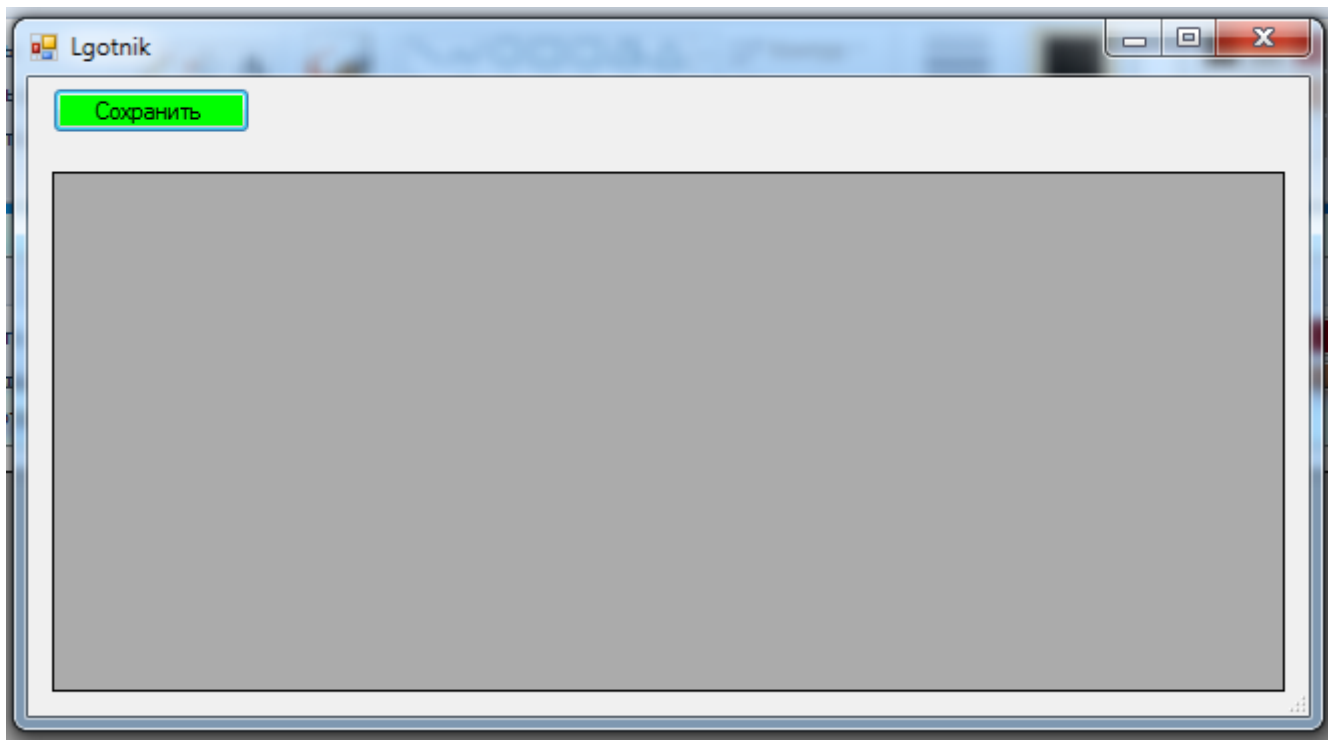


Рисунок 1.5. «Lgotnik»

4. Заключение

В данной работе была спроектирована и реализована система ведения учета льготников и сопутствующих документов.

При проектировании применялась технология «клиент–сервер», что позволяет использовать эту систему в сети, размещать базу данных на сервере и получать доступ к ней одновременно нескольким пользователям.

Для разработки базы использовался SQL-сервер MS SQL.

Также в данной работе были изучены и раскрыты такие возможности современных БД как триггеры и процедуры,. Были применены алгоритмы автоматического создания базы данных и таблиц, а также автоматического их заполнения.

Приложение 1. Листинг скрипта создания базы данных.

База данных создавалась из самой программы:

```
conn = new SqlConnection(@"Data Source=localhost; Integrated
Security=SSPI;");
conn.Open();

//создаем БД
SqlCommand command = new SqlCommand(@"USE master;
IF DB_ID('DocumentLgoty') IS NULL CREATE
DATABASE DocumentLgoty;", conn);
command.ExecuteNonQuery();
//создаем таблицы
command = new SqlCommand(@"USE DocumentLgoty;
IF OBJECT_ID('dbo.Document') IS NULL CREATE
TABLE Document(
        doc_name Varchar(256) Not Null,
        doc_data_begin Datetime Null,
        doc_data_end Datetime Null,
        finance Varchar(256) Null,
        PRIMARY KEY(doc_name),
        CHECK (YEAR(doc_data_begin)<=YEAR(getdate()))
AND YEAR(doc_data_begin)>1990 AND
YEAR(doc_data_end)>=YEAR(getdate()) AND
doc_data_begin<doc_data_end)
);
IF OBJECT_ID('dbo.VidLgoty') IS NULL CREATE
TABLE VidLgoty(
        lgota_name Varchar(256) Not Null,
        lgota_proc Int Null,
        lgota_data_begin Datetime Null,
        lgota_data_end JKH Datetime Null,
        doc_name Varchar(256) Not Null,
        PRIMARY KEY(lgota_name),
        FOREIGN KEY (doc_name) REFERENCES
Document(doc_name),
        CHECK
(YEAR(lgota_data_begin)<=YEAR(getdate())) AND
YEAR(lgota_data_begin)>1990 AND
YEAR(lgota_data_end_JKH)>=YEAR(getdate()) AND
lgota_data_begin<lgota_data_end_JKH)
);
IF OBJECT_ID('dbo.Lgota') IS NULL CREATE TABLE
Lgota(
        passport Int Not Null,
        lgota_name Varchar(256) Not Null,
        PRIMARY KEY(passport),
        FOREIGN KEY (lgota_name) REFERENCES
VidLgoty(lgota_name),
        CHECK (passport > 100000000)
);
```

```

IF OBJECT_ID('dbo.Grajdantin') IS NULL CREATE
TABLE Grajdantin(
    passport Int Not Null,
    category Int Null,
    inv_group Int Null,
    war_place Varchar(256) Null,
    PRIMARY KEY(passport),
    FOREIGN KEY (passport) REFERENCES
Lgota(passport),
    CHECK (passport > 100000000 AND category<4
AND category>0)
);", conn);

command.ExecuteNonQuery();

```

Создание пользователя:

```

USE master
GO
-- создадим логин Ivanov с паролем 123
CREATE LOGIN Ivanov
    WITH PASSWORD = '123';
GO

--создание самого пользователя
CREATE USER Ivanov FOR LOGIN Ivanov;
GO
GRANT CONNECT SQL TO Ivanov
GO

```

Триггеры и процедуры создавались непосредственно из среды MS SQL Server из-за невозможности настроить привелегии к БД:

--процедура 1

```

USE DocumentLgoty
GO

CREATE PROC dbo.Zapros01
@var varchar(256)
AS
SELECT doc_name FROM Document WHERE Document.doc_name IN
(SELECT doc_name FROM VidLgoty WHERE lgota_name=@var)
GO

CREATE PROC dbo.Zapros1
@var varchar(256)

```

```

AS
CREATE TABLE dbo.Table1 ([name] sysname NOT NULL)
INSERT INTO dbo.Table1 ([name])
EXEC dbo.Zapros01 @var
SELECT * FROM dbo.Table1
DROP TABLE dbo.Table1
GO

--ВЫЗОВ
USE DocumentLgoty
GO
EXECUTE dbo.Zapros1 'ЖКХ'
GO

```

--процедура 2

```

USE DocumentLgoty
GO

CREATE PROC dbo.Zapros02
@var varchar(256)
AS
    SELECT Grajdantin.category AS cotegory, Lgota.lgota_name AS
lgota,
    COUNT(Lgota.lgota_name) AS
LgotaCount, COUNT(Grajdantin.category) CategoryCount
    FROM Grajdantin, Lgota WHERE Lgota.lgota_name IN (SELECT
lgota_name FROM VidLgoty AS V WHERE
    V.doc_name IN (SELECT doc_name FROM Document WHERE
doc_name=@var)) GROUP BY
    Grajdantin.category, Lgota.lgota_name
GO

CREATE PROC dbo.Zapros2
@var varchar(256)
AS

```

```

CREATE TABLE dbo.Table2 ([name] sysname NOT NULL,
[name2] sysname NOT NULL,
[name3] sysname NOT NULL,
[name4] sysname NOT NULL
)
INSERT INTO dbo.Table2 ([name],[name2],[name3],[name4])
EXEC dbo.Zapros02 @var
SELECT * FROM dbo.Table2
DROP TABLE dbo.Table2
GO

```

```

--вызов процедуры
USE DocumentLgoty
GO
EXECUTE dbo.Zapros2 'Документ 825'
GO

```

--процедура 3

```

USE DocumentLgoty
GO

CREATE PROC dbo.Zapros03
@cat int,
@docname varchar(256),
@d_before datetime,
@d_end datetime
AS
SELECT lgota_name FROM VidLgoty AS Vlgota WHERE
Vlgota.lgota_name
IN (SELECT Lgota.lgota_name FROM Lgota AS lgota WHERE
lgota.passport
IN (SELECT Grajdanin.passport FROM Grajdanin WHERE
category=@cat)) AND Vlgota.doc_name IN
(SELECT doc_name FROM Document AS D WHERE
D.doc_name=@docname

```

```
        AND D.doc_data_begin BETWEEN @d_before AND @d_end)
GO
```

```
CREATE PROC dbo.Zapros3
@cat int,
@docname varchar(256),
@d_before datetime,
@d_end datetime
AS
CREATE TABLE dbo.Table3 ([name] sysname NOT NULL)
INSERT INTO dbo.Table3 ([name])
EXEC dbo.Zapros03 @cat,@docname,@d_before,@d_end
SELECT * FROM dbo.Table3
DROP TABLE dbo.Table3
GO
```

```
--вызов процедуры
USE DocumentLgoty
GO
DECLARE @cat int,
@docname varchar(256),
@d_before datetime,
@d_end datetime
SET @cat=1
SET @docname='Документ 824'
SET @d_before='03/04/1993'
SET @d_end='07/05/2018'
EXECUTE dbo.Zapros3 @cat,@docname,@d_before,@d_end
GO
```


***--триггер на событие INSERT на таблицу Lgota, он будет
автоматически добавлять записи в таблицу Grajdanin***

```
USE DocumentLgoty

IF OBJECT_ID ('auto_fill_grajdanin') IS NOT NULL DROP
TRIGGER auto_fill_grajdanin

GO

CREATE TRIGGER auto_fill_grajdanin ON Lgota
FOR INSERT AS DECLARE
    @passport Varchar(256),
    @lgota_name int
BEGIN
    SELECT @passport = (SELECT passport FROM inserted)

    INSERT INTO Grajdanin VALUES (@passport, NULL, NULL,
NULL)
END
GO

--проверка работы триггера
USE DocumentLgoty
INSERT INTO Lgota VALUES (789490670,'ЖКХ')
GO
SELECT * FROM Grajdanin
GO
```

Приложение 2. Листинг кода приложения.

Главная форма

```
namespace Kursovaya_Ivanov
{
    public partial class Form1 : Form
    {
        SqlConnection conn;
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            //коннектимся к локалке Microsoft SQL Server
            conn = new SqlConnection(@"Data Source=localhost;
Integrated Security=SSPI;");
            conn.Open();
            //создаем БД
            SqlCommand command = new SqlCommand(@"USE master;
            IF DB_ID('DocumentLgoty') IS NULL CREATE
DATABASE DocumentLgoty;", conn);
            command.ExecuteNonQuery();
            //создаем таблицы
            command = new SqlCommand(@"USE DocumentLgoty;
            IF OBJECT_ID('dbo.Document') IS NULL CREATE
TABLE Document(
                doc_name Varchar(256) Not Null,
                doc_data_begin Datetime Null,
                doc_data_end Datetime Null,
                finance Varchar(256) Null,
                PRIMARY KEY(doc_name),
                CHECK (YEAR(doc_data_begin)<=YEAR(getdate()))
AND YEAR(doc_data_begin)>1990 AND
YEAR(doc_data_end)>=YEAR(getdate()) AND
doc_data_begin<doc_data_end)
            );
            IF OBJECT_ID('dbo.VidLgoty') IS NULL CREATE
TABLE VidLgoty(
                lgota_name Varchar(256) Not Null,
                lgota_proc Int Null,
                lgota_data_begin Datetime Null,
                lgota_data_end_JKH Datetime Null,
                doc_name Varchar(256) Not Null,
                PRIMARY KEY(lgota_name),
                FOREIGN KEY (doc_name) REFERENCES
Document(doc_name),
                CHECK
                (YEAR(lgota_data_begin)<=YEAR(getdate()) AND
YEAR(lgota_data_begin)>1990 AND
YEAR(lgota_data_end_JKH)>=YEAR(getdate()) AND
lgota_data_begin<lgota_data_end_JKH)
```

```

    );
    IF OBJECT_ID('dbo.Lgota') IS NULL CREATE TABLE
Lgota (
        passport Int    Not Null,
        lgota_name    Varchar(256)    Not Null,
        PRIMARY KEY(passport),
        FOREIGN KEY (lgota_name) REFERENCES
VidLgoty(lgota_name),
        CHECK (passport > 100000000)
    );
    IF OBJECT_ID('dbo.Grajdantin') IS NULL CREATE
TABLE Grajdantin(
        passport Int    Not Null,
        category Int    Null,
        inv_group    Int    Null,
        war_place    Varchar(256)    Null,
        PRIMARY KEY(passport),
        FOREIGN KEY (passport) REFERENCES
Lgota(passport),
        CHECK (passport > 100000000 AND category<4
AND category>0)
    );", conn);
    command.ExecuteNonQuery();
    comboBox1.Items.Add("Найти документ по названию
льготы");
    comboBox1.Items.Add("Найти всех граждан и льготы и
их количество по документу");
    comboBox1.Items.Add("Выбрать все льготы, указанные в
заданном нормативном документе, для заданной категории граждан,
которые действовали в указанный период времени");
    comboBox1.Items.Add("Определить перечень категорий
граждан, для которых действуют максимальные жилищно-коммунальные
льготы");
    for (int i = 1; i < 4; i++)
    {
        comboBox2.Items.Add(Convert.ToString(i));
    }
    dateTimePicker2.CustomFormat = "MM/dd/yyyy";
    dateTimePicker2.Format =
DateTimePickerFormat.Custom;
    dateTimePicker1.CustomFormat = "MM/dd/yyyy";
    dateTimePicker1.Format =
DateTimePickerFormat.Custom;
    conn.Close();
    refresh_data();
}

private async void refresh_data(){
    SqlDataReader sqlReader = null;
    await conn.OpenAsync();
    SqlCommand command = new SqlCommand("USE
DocumentLgoty;SELECT doc_name FROM Document", conn);
    try

```

```

        {
            sqlReader = await command.ExecuteReaderAsync();
            while (sqlReader.Read())
            {
                comboBox3.Items.Add(Convert.ToString(sqlReader["doc_name"]));
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message.ToString(),
                ex.Source.ToString(), MessageBoxButtons.OK,
                MessageBoxIcon.Error);
        }
        finally
        {
            if (sqlReader != null) sqlReader.Close();
        }
        sqlReader = null;
        command = new SqlCommand("USE DocumentLgoty;SELECT
lgota_name FROM VidLgoty", conn);
        try
        {
            sqlReader = await command.ExecuteReaderAsync();
            while (sqlReader.Read())
            {
                comboBox4.Items.Add(Convert.ToString(sqlReader["lgota_name"]));
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message.ToString(),
                ex.Source.ToString(), MessageBoxButtons.OK,
                MessageBoxIcon.Error);
        }
        finally
        {
            if (sqlReader != null) sqlReader.Close();
        }
        conn.Close();
        //conn.Dispose();
    }
    private async void button1_Click(object sender,
EventArgs e)
    {
        await conn.OpenAsync();
        try
        {
            SqlCommand command = new SqlCommand(@"USE
DocumentLgoty;INSERT INTO Document VALUES ('Документ

```

```

824',convert(datetime, '10/23/1993', 101),convert(datetime,
'10/23/2019', 101),'МИНФИН'),
        ('Документ 825',convert(datetime, '10/23/2003',
101),convert(datetime, '10/23/2021', 101),'местный');
        INSERT INTO VidLgoty
VALUES('ЖКХ','20',convert(datetime, '10/23/1995',
101),convert(datetime, '10/23/2019', 101),'Документ 824'),
        ('Телефон','17',convert(datetime, '10/23/2003',
101),convert(datetime, '10/23/2019', 101),'Документ 824'),
        ('Проезд','5',convert(datetime, '10/23/2003',
101),convert(datetime, '10/23/2021', 101),'Документ 825'),
        ('Здравоохранение','50',convert(datetime,
'10/23/2003', 101),convert(datetime, '10/23/2021',
101),'Документ 825');
        INSERT INTO Lgota VALUES(889945780,'Телефон'),
        (889045780,'ЖКХ'),
        (889945680,'Проезд'),
        (879945780,'Здравоохранение'),
        (879555780,'Здравоохранение'),
        (999945780,'Здравоохранение');
        INSERT INTO Grajdanin VALUES(889945780,1,2,NULL),
        (889045780,3,NULL,NULL),
        (889945680,1,NULL,NULL),
        (879945780,2,NULL,NULL);", conn);
        await command.ExecuteNonQueryAsync();
        this.button1.BackColor =
System.Drawing.Color.Red;
        this.button1.Text = "Выполнено";

    }
    catch (System.Data.SqlClient.SqlException ex)
    {
        MessageBox.Show(ex.Message.ToString(),
ex.Source.ToString(), MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
    finally
    {
        conn.Close();
        refresh_data();
    }
}

private void comboBox1_SelectedIndexChanged(object
sender, EventArgs e)
{
    comboBox4.Enabled = false; comboBox3.Enabled =
false; comboBox2.Enabled = false; dateTimePicker2.Enabled =
false; dateTimePicker1.Enabled = false;
    int i = comboBox1.SelectedIndex;
    switch (i)
    {
        case 0:

```

```

        comboBox4.Enabled = true;
        ;break;
    case 1:
        comboBox3.Enabled = true;
        ;break;
    case 2:
        comboBox3.Enabled = true; comboBox2.Enabled
= true; dateTimePicker2.Enabled = true; dateTimePicker1.Enabled
= true;
        ;break;
    case 3:
        ;break;
    }
}

private async void button2_Click(object sender,
EventArgs e)
{
    int i = comboBox1.SelectedIndex;
    String str = "USE DocumentLgoty;";
    switch (i)
    {
        case 0: str += @"SELECT doc_name FROM Document
WHERE Document.doc_name IN
        (SELECT doc_name FROM VidLgoty WHERE
lgota_name='" + comboBox4.Text + "')"; break;
        //"EXEC dbo.Zapros01 '" + comboBox4.Text +
        "'";break;
        case 1: str += @"SELECT Grajdanin.category AS
cotegory,Lgota.lgota_name AS lgota,
        COUNT(Lgota.lgota_name) AS
LgotaCount,COUNT(Grajdanin.category) CategoryCount
        FROM Grajdanin,Lgota WHERE Lgota.lgota_name
IN (SELECT lgota_name FROM VidLgoty AS V WHERE
        V.doc_name IN (SELECT doc_name FROM Document
WHERE doc_name='" + comboBox3.Text + @')) GROUP BY
        Grajdanin.category,Lgota.lgota_name"; break;
        case 2: str += @"SELECT lgota_name FROM VidLgoty
AS Vlgota WHERE Vlgota.lgota_name
        IN (SELECT Lgota.lgota_name FROM Lgota AS
lgota WHERE lgota.passport
        IN (SELECT Grajdanin.passport FROM Grajdanin
WHERE category=" + comboBox2.Text + @")) AND Vlgota.doc_name IN
        (SELECT doc_name FROM Document AS D WHERE
D.doc_name='" + comboBox3.Text + @'"
        AND D.doc_data_begin BETWEEN
        '"+dateTimePicker2.Text+"' AND '"+dateTimePicker1.Text+"')";
break;
        case 3: str += @"SELECT G.category FROM
Grajdanin AS G WHERE G.passport IN (
        SELECT L.passport FROM Lgota AS L WHERE
L.lgota_name IN (
        SELECT VL.lgota_name FROM VidLgoty AS VL

```

```

        WHERE (VL.lgota_name='ЖКХ' OR
VL.lgota_name='Телефон') AND VL.lgota_proc=(SELECT
MAX(VL2.lgota_proc)
        FROM VidLgoty AS VL2 WHERE
VL2.lgota_name='ЖКХ' OR VL2.lgota_name='Телефон'))"; break;
    }

    await conn.OpenAsync();
    SqlDataAdapter da = new SqlDataAdapter(str, conn);
    //da.SelectCommand.CommandType =
CommandType.StoredProcedure;

    SqlCommandBuilder cb = new SqlCommandBuilder(da);

    try
    {
        DataSet ds = new DataSet();
        da.Fill(ds);
        dataGridView1.DataSource = ds.Tables[0];
    }
    catch (Exception ex)
    {
        MessageBox.Show("Запрос не вернул результат:" +
ex.Data);
    }
    finally
    {
        conn.Close();
    }
}

private void button3_Click(object sender, EventArgs e)
{
    Document form_doc=new Document(this, conn);
    form_doc.ShowDialog();
}

private void button4_Click(object sender, EventArgs e)
{
    LgotaVid form = new LgotaVid(this, conn);
    form.ShowDialog();
}

private void button5_Click(object sender, EventArgs e)
{
    Lgota form =new Lgota(this,conn);
    form.ShowDialog();
}

private void button6_Click(object sender, EventArgs e)
{
    Lgotnik form = new Lgotnik(this, conn);
    form.ShowDialog();
}

```

```

    }
}

```

Форма "Lgota":

```

namespace Kursovaya_Ivanov
{
    public partial class Lgota : Form
    {
        SqlConnection conn;
        SqlDataAdapter da;
        DataSet ds;
        int rowcount = 0;
        public Lgota(Form f, SqlConnection c)
        {
            InitializeComponent();
            conn = c;
        }

        private void Lgota_Load(object sender, EventArgs e)
        {
            da = new SqlDataAdapter("USE DocumentLgoty;SELECT *
FROM Lgota", conn);
            ds = new DataSet();
            conn.Open();
            da.Fill(ds, "Lgota");
            conn.Close();
            dataGridView1.DataSource = ds;
            dataGridView1.DataMember = "Lgota";
            rowcount = dataGridView1.RowCount;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            using (var bulkCopy = new SqlBulkCopy("Data
Source=localhost; Initial Catalog=DocumentLgoty;Integrated
Security=SSPI;"))
            {
                bulkCopy.BatchSize = 500;
                bulkCopy.NotifyAfter = 1000;
                bulkCopy.DestinationTableName = "Lgota";

                DataTable dt = new DataTable();

                dt.Columns.Add("passport", typeof(int));
                dt.Columns.Add("lgota_name", typeof(string));

                //dataGridView1.AllowUserToAddRows = false;
                for (int i = rowcount - 1; i <
dataGridView1.RowCount - 1; i++)

```



```

        {
            dt.Rows.Add(dataGridView1.Rows[i].Cells[0].Value,
dataGridView1.Rows[i].Cells[1].Value);

            }
            rowcount = dataGridView1.RowCount;
            bulkCopy.WriteToServer(dt);
            //dataGridView1.AllowUserToAddRows = true;
        }
    }
}

```

Остальные формы аналогичны.