ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ

Курсовая работа

по дисциплине «Базы данных»

по теме

Разработка БД для отдела аспирантуры ВУЗа

Ремизов Дмитрий

Группа А-05-20

Преподаватель Сидорова Н.П

Москва 2023

Оглавление

[Введение 3](#_Toc153401493)

[Цель работы: 3](#_Toc153401494)

[Уточнения предметной области: 3](#_Toc153401495)

[Этапы выполнения курсовой работы 3](#_Toc153401496)

[Проектирование базы данных 4](#_Toc153401497)

[Инфологическая модель: 4](#_Toc153401498)

[Логическая модель БД 6](#_Toc153401499)

[Реляционная модель БД: 7](#_Toc153401500)

[Функциональные зависимости: 9](#_Toc153401501)

[Проектирование правил целостности 11](#_Toc153401502)

[Реализация БД 12](#_Toc153401503)

[Создание таблиц 12](#_Toc153401504)

[Реализация правил ссылочной целостности 14](#_Toc153401505)

[Хранимые процедуры 17](#_Toc153401506)

[Программная реализация 21](#_Toc153401507)

[Окна регистрации и авторизации 21](#_Toc153401508)

[Главное окно 22](#_Toc153401509)

[Окно запросов 23](#_Toc153401510)

[Окно отчетов 24](#_Toc153401511)

[Изменение данных аспирантов 25](#_Toc153401512)

[Заключение 28](#_Toc153401513)

# Введение

Цель работы: разработка БД для отдела аспирантуры ВУЗа. Каждый аспирант имеет

одного научного руководителя и обучается по одному научному направлению.

БД должна поддерживать выполнение следующих функций:

• учет сведений об аспирантах ВУЗа различных категорий (очные, заочные, соискатели, докторанты);

• формирование списка аспирантов по кафедрам и по научным руководителям;

• изменение данных об аспирантах (научный руководитель, научное направление, персональные данные, дипломы, награды);

• учет научных публикаций аспирантов (у одной публикации только один автор);

• учет научных публикаций научного руководителя аспиранта;

• учет работы научных советов.

• составление отчетов о состоявшихся защитах (по различным категориям, научным направлениям, кафедрам, научным руководителям и т.п.)

## Уточнения предметной области:

* По одному направлению может быть несколько научных советов, один научный совет может принимать защиты по нескольким направлениям
* Аспирант может обучаться только на одном направлении, на одном направлении может обучаться несколько аспирантов
* У одного аспиранта может быть только одна защита (успешная или неуспешная), одну защиту может выполнять только один аспирант.
* У публикации может быть только 1 автор.
* Один совет может принимать много защит, у одной защиты может быть только один совет.
* У аспиранта может быть только один научный руководитель, у научного руководителя может быть несколько аспирантов.
* Один преподаватель может преподавать на нескольких направлениях, на одном направлении могут преподавать несколько преподавателей

## Этапы выполнения курсовой работы

1. Анализ предметной области
2. Разработка ER-модели данных в Erwin Data Modeler
3. Разработка и нормализация БД
4. Реализация базы данных и интерфейса в среде программирования.

# Проектирование базы данных

В результате анализа предметной области были выделены следующие сущности:

* Аспирант
* Научный руководитель
* Направление обучения
* Публикация
* Защита
* Научный совет

## Инфологическая модель:

|  |
| --- |
| **Main Subject Area/Display1** |



|  |
| --- |
| **Entity** |

|  |  |
| --- | --- |
| **Entity** | |
| **Name** | **Definition** |
| Defending | Защита диссертации |
| Direction | Научное направление |
| Graduate\_student | Аспирант |
| Publication | Публикация |
| Science\_council | Научный совет |
| Scientific\_director | Научный руководитель |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute(s) of "Defending" Entity** | | | | |
| **Name** | **Datatype** | **Is PK** | **Is FK** | **Definition** |
| defending\_id | INTEGER | Yes | No | Идентификатор защиты |
| council\_id | INTEGER | No | Yes | Идентификатор научного совета, оценивавшего защиту |
| student\_id | INTEGER | No | Yes | Идентификатор аспиранта |
| date | DATE | No | No | Дата защиты |
| council\_decision | VARCHAR(20) | No | No | Решение совета по защите (результат защиты) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute(s) of "Direction" Entity** | | | | |
| **Name** | **Datatype** | **Is PK** | **Is FK** | **Definition** |
| direction\_id | INTEGER | Yes | No | Идентификатор направления |
| name | VARCHAR(20) | No | No | Название направления |
| category | VARCHAR(20) | No | No | Категория направления |
| department | VARCHAR(20) | No | No | Кафедра направления |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute(s) of "Graduate\_student" Entity** | | | | |
| **Name** | **Datatype** | **Is PK** | **Is FK** | **Definition** |
| student\_id | INTEGER | Yes | No | Идентификатор аспиранта |
| director\_id | INTEGER | No | Yes | Идентификатор научного руководителя |
| direction\_id | INTEGER | No | Yes | Идентификатор направления |
| name | VARCHAR(20) | No | No | ФИО аспиранта |
| date\_of\_birth | DATE | No | No | Дата рождения |
| awards | INTEGER | No | No | Количество наград |
| diploms | VARCHAR(20) | No | No | Дипломы аспиранта |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute(s) of "Publication" Entity** | | | | |
| **Name** | **Datatype** | **Is PK** | **Is FK** | **Definition** |
| publication\_id | INTEGER | Yes | No | Идентификатор публикации |
| student\_id | INTEGER | No | Yes | Идентификатор аспиранта |
| publication\_name | VARCHAR(20) | No | No | Название публикации |
| publication\_resource | VARCHAR(20) | No | No | Ресурс, на котором была опубликована |
| publication\_date | DATE | No | No | Дата публикации |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute(s) of "Science\_council" Entity** | | | | |
| **Name** | **Datatype** | **Is PK** | **Is FK** | **Definition** |
| council\_id | INTEGER | Yes | No | Идентифиикатор совета |
| composition | VARCHAR(20) | No | No | Состав совета |
| number\_of\_successfull\_defendings | INTEGER | No | No | Количество успешных защит перед советом |
| total\_number\_of\_defendings | INTEGER | No | No | Общее число защит перед советом |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute(s) of "Scientific\_director" Entity** | | | | |
| **Name** | **Datatype** | **Is PK** | **Is FK** | **Definition** |
| director\_id | INTEGER | Yes | No | Идентификатор научного руководителя |
| name | VARCHAR(20) | No | No | ФИО научного руководителя |
| department | VARCHAR(20) | No | No | Кафедра |

|  |  |  |
| --- | --- | --- |
| **Relationship** | | |
| **Relationship** | | |
| **FK Name** | **Comment** | **Cardinality** |
| Совет\_\_\_\_направление | Соотносит совет с направлениями, по которым совет может принимать защиту. По одному направлению может быть несколько советов, один совет может принимать защиту по нескольким направлениям | Many-to-Many |
| Аспирант\_\_\_\_направление | Соотносит аспиранта и его направление. Аспирант может обучаться только на одном направлении, на одном направлении может обучаться несколько аспирантов | One-to-One-or-More (P) |
| Защищает | Соотносит аспиранта с защитой его диссертации. У аспиранта может быть только 1 защита, одну защиту может выполнять только один аспирант | One-to-Exactly-1 |
| Опубликовал | Соотносит аспиранта с его публикациями. У публикации может быть только 1 автор. | One-to-One-or-More (P) |
| Принимает\_защиту | Соотносит защиту с научным советом, который ее принимает | One-to-One-or-More (P) |
| Научный\_руководитель\_\_\_\_аспирант | Соотносит научного руководителя с его аспирантом. У аспиранта может быть только один научный руководитель, у научного руководителя может быть несколько аспирантов | One-to-One-or-More (P) |
| Преподает\_на | Соотносит научного руководителя с направлением, к которому он относится. Один научный руководитель может относиться к нескольким направлениям | Many-to-Many |

## Логическая модель БД

На основе построенной ER-модели для разработки БД создаётся реляционная модель базы данных. Для её построения необходимо выполнить несколько этапов:

* Преобразовать сущности ER-модели в отношения реляционной модели базы данных
* Привести отношения к 3-ей нормальной форме
* Определить правила целостности для отношений в  реляционной модели базы данных

## Реляционная модель БД:

|  |
| --- |
| **Main Subject Area/Display1** |



|  |  |
| --- | --- |
| **Name** | Graduate\_student |
| **Comment** | Аспирант |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column(s) of "Graduate\_student" Table** | | | | |
| **Name** | **Datatype** | **Comment** | **Is PK** | **Is FK** |
| student\_id | integer | Идентификатор аспиранта | Yes | No |
| director\_id | integer | Идентификатор научного руководителя | No | Yes |
| direction\_id | integer | Идентификатор направления | No | Yes |
| name | varchar(20) | ФИО аспиранта | No | No |
| date\_of\_birth | datetime | Дата рождения | No | No |
| awards | integer | Количество наград | No | No |
| diploms | varchar(20) | Дипломы аспиранта | No | No |

|  |  |
| --- | --- |
| **Name** | Scientific\_director |
| **Comment** | Научный руководитель |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column(s) of "Scientific\_director" Table** | | | | |
| **Name** | **Datatype** | **Comment** | **Is PK** | **Is FK** |
| director\_id | integer | Идентификатор научного руководителя | Yes | No |
| name | varchar(20) | ФИО научного руководителя | No | No |
| department | varchar(20) | Кафедра | No | No |

|  |  |
| --- | --- |
| **Name** | Direction |
| **Comment** | Научное направление |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column(s) of "Direction" Table** | | | | |
| **Name** | **Datatype** | **Comment** | **Is PK** | **Is FK** |
| direction\_id | integer | Идентификатор направления | Yes | No |
| category | varchar(20) | Категория направления | No | No |
| name | varchar(20) | Название направления | No | No |
| department | varchar(20) | Кафедра направления | No | No |

|  |  |
| --- | --- |
| **Name** | Publication |
| **Comment** | Публикация |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column(s) of "Publication" Table** | | | | |
| **Name** | **Datatype** | **Comment** | **Is PK** | **Is FK** |
| publication\_id | integer | Идентификатор публикации | Yes | No |
| student\_id | integer | Идентификатор аспиранта | No | Yes |
| publication\_name | varchar(20) | Название публикации | No | No |
| publication\_resource | varchar(20) | Ресурс, на котором была опубликована | No | No |
| publication\_date | datetime | Дата публикации | No | No |

|  |  |
| --- | --- |
| **Name** | Defending |
| **Comment** | Защита диссертации |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column(s) of "Defending" Table** | | | | |
| **Name** | **Datatype** | **Comment** | **Is PK** | **Is FK** |
| defending\_id | integer | Идентификатор защиты | Yes | No |
| council\_id | integer | Идентификатор совета | No | Yes |
| student\_id | integer | Идентификатор аспиранта | No | Yes |
| date | datetime | Дата защиты | No | No |
| council\_decision | varchar(20) | Решение совета | No | No |

|  |  |
| --- | --- |
| **Name** | Science\_council |
| **Comment** | Научный совет |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column(s) of "Science\_council" Table** | | | | |
| **Name** | **Datatype** | **Comment** | **Is PK** | **Is FK** |
| council\_id | integer | Идентификатор совета | Yes | No |
| composition | varchar(20) | Состав совета | No | No |
| number\_of\_successfull\_defendings | integer | Количество успешных защит перед советом | No | No |
| total\_number\_of\_defendings | integer | Общее число защит перед советом | No | No |

|  |  |
| --- | --- |
| **Name** | Scientific\_director\_Direction |
| **Comment** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column(s) of "Scientific\_director\_Direction" Table** | | | | |
| **Name** | **Datatype** | **Comment** | **Is PK** | **Is FK** |
| Scientific\_director\_Direction\_id | integer |  | Yes | No |
| director\_id | integer | Идентификатор научного руководителя | No | Yes |
| direction\_id | integer | Идентификатор направления | No | Yes |

|  |  |
| --- | --- |
| **Name** | Direction\_Science\_council |
| **Comment** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column(s) of "Direction\_Science\_council" Table** | | | | |
| **Name** | **Datatype** | **Comment** | **Is PK** | **Is FK** |
| Direction\_Science\_council\_id | integer |  | Yes | No |
| direction\_id | integer | Идентификатор направления | No | Yes |
| council\_id | integer | Идентификатор совета | No | Yes |

Для обоснования нахождения БД в 3НФ необходимо провести анализ функциональных зависимостей (ФЗ) спроектированных отношений.

## **Функциональные зависимости**:

**director\_id**

director\_id => name

director\_id => department

**direction\_id**

direction\_id => category

direction\_id => name

direction\_id => department

**council\_id**

council\_id => composition

council\_id => number\_of\_successfull\_defendings

council\_id => total\_number\_of\_defendings

**Scientific\_director\_Direction\_id(director\_id, direction\_id)**

**Direction\_Science\_council\_id(direction\_id, council\_id)**

**student\_id (director\_id, direction\_id)**

student\_id => name

student\_id => date\_of\_birth

student\_id => awards

student\_id => diploms

**publication\_id(student\_id)**

publication\_id => publication\_name

publication\_id => publication\_resource

publication\_id => publication\_date

**defending\_id(council\_id, student\_id)**

defending\_id => date

defending\_id => council\_decision

Обоснование нахождения РБД в третьей нормальной форме.

Таблица находится в первой нормальной форме (1НФ) тогда и только тогда, когда каждое поле отношение содержит атомарное значение.

Для всех полей таблиц это условие автоматически выполняется благодаря их типам.

Для остальных полей атомарность значений полей предполагается по умолчанию.

(поле name в department может содержать только одно название конкретного отделения. Если же у отделения изменилось название, устаревшее значение соответствующего поля заменяется новым.)

Отношение находится во 2НФ, если оно находится в 1НФ и каждый неключевой атрибут функционально полно зависит от ключа.

Рассмотрев все функциональные зависимости, представленные выше, можно увидеть что атрибуты функционально полно зависят от ключа. Таким образом, показали, что каждый неключевой атрибут функционально полно зависит от ключа.

Отношение находится в 3НФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Рассмотрим функциональные зависимости, представленные выше. В качестве примера рассмотрим таблицу направлений: функциональные зависимости между атрибутами name, category и department отсутствуют, следовательно, каждый неключевой атрибут нетранзитивно зависит от первичного ключа direction\_id. Аналогично можем показать нетранзитивные зависимости атрибутов от первичных ключей для остальных отношений.

## Проектирование правил целостности

* Атрибуты идентификаторов, количества успешных защит совета, общее число защит совета, награды аспиранта соответствуют формату INT;
* Атрибуты даты защиты, даты рождения аспиранта, даты публикации соответствуют формату DATE;
* Атрибуты названий, имен, категории направления, кафедры, состав совета дипломы аспиранта, ресурс публикации, решения совета соответствуют формату VARCHAR(50)
* Идентификаторы являются ненулевыми значениями

С целью обеспечения целостности данных, содержащихся в разных таблицах, будем использовать ограничения FOREIGN KEY (ограничения ссылочной целостности – referential integrity constraint), задаваемые при описании структуры таблицы. Ограничение ссылочной целостности не позволяет значениям из столбца одной таблицы принимать значения, кроме как из присутствующих в столбце другой таблицы. Это делается при помощи ограничителей FOREIGN KEY (внешний ключ) и REFERENCES (указатель ссылки). Таблица, содержащая FOREIGN KEY, считается родительской таблицей. Таблица, содержащая REFERENCES, считается дочерней таблицей. Внешний ключ и указатель ссылки могут находиться в одной таблице, т.е. родительская таблица одновременно является дочерней. Для обеспечения целостности данных, содержащихся в разных таблицах, можно использовать ограничения FOREIGN KEY, задаваемые при описании структуры таблицы, и триггеры. Ограничения FOREIGN KEY поддерживают связи между таблицами. Ограничения ссылочной целостности используются при каскадном удалении, т.е. при удалении записи в родительской таблице удаляются все записи с указанным ключом из дочерних таблиц, и, наоборот, при запрете удаления/модификации, т.е. при наличии зависимых записей в дочерних таблицах, значение ключа записи в родительской таблице нельзя удалить или модифицировать. Описатель внешнего ключа применяется для принудительного установления связи между данными в двух таблицах.

Будем использовать каскадную стратегию поддержания ссылочной целостности:

*CASCADE (КАСКАДИРОВАТЬ)*- разрешить выполнение требуемой операции, но внести при этом необходимые поправки в других отношениях так, чтобы не допустить нарушения ссылочной целостности и сохранить все имеющиеся связи. Изменение начинается в родительском отношении и каскадно выполняется в дочернем отношении. В реализации этой стратегии имеется одна тонкость, заключающаяся в том, что дочернее отношение само может быть родительским для некоторого третьего отношения. При этом может дополнительно потребоваться выполнение какой-либо стратегии и для этой связи и т.д. Если при этом какая-либо из каскадных операций (любого уровня) не может быть выполнена, то необходимо отказаться от первоначальной операции и вернуть базу данных в исходное состояние. Это самая сложная стратегия, но она хороша тем, что при этом не нарушается связь между кортежами родительского и дочернего отношений.

Для таблиц:

* Scientific\_director\_Direction по director\_id из Scientific\_director
* Scientific\_director\_Direction по direction\_id из Direction
* Science\_council\_Direction по council\_id из Science\_council
* Science\_council\_Direction по direction\_id из Direction,
* Graduate\_student по director\_id из Scientific\_director,
* Graduate\_student по direction\_id из Direction,
* Publication по student\_id из Graduate\_Student,
* Defending по council\_id из Science\_council,
* Defending по student\_id из Graduate\_Student,

# Реализация БД

В качестве системы управления реляционными базами данных выбрана СУБД Microsoft SQL Server.  Данный продукт является разработкой корпорации Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с небольшими и средними по размеру базами данных до крупных баз данных масштаба предприятия**.**

## Создание таблиц

CREATE TABLE Direction

(

direction\_id INT IDENTITY (1,1) NOT NULL PRIMARY KEY,

direction\_name VARCHAR (50),

category VARCHAR (50),

direction\_department VARCHAR(50)

)

CREATE TABLE Scientific\_director

(

director\_id INT IDENTITY (1,1) NOT NULL PRIMARY KEY,

director\_name VARCHAR (50),

director\_department VARCHAR(50)

)

CREATE TABLE Scientific\_director\_Direction

(

Scientific\_director\_Direction\_id INT IDENTITY (1,1) NOT NULL PRIMARY KEY,

director\_id INT NOT NULL FOREIGN KEY REFERENCES Scientific\_director,

direction\_id INT NOT NULL FOREIGN KEY REFERENCES Direction

)

CREATE TABLE Science\_council

(

council\_id INT IDENTITY (1,1) NOT NULL PRIMARY KEY,

composition VARCHAR (50),

number\_of\_successfull\_defendings INT,

total\_number\_of\_defendings INT

)

CREATE TABLE Science\_council\_Direction

(

Direction\_Science\_council\_id INT IDENTITY (1,1) NOT NULL PRIMARY KEY,

council\_id INT NOT NULL FOREIGN KEY REFERENCES Science\_council,

direction\_id INT NOT NULL FOREIGN KEY REFERENCES Direction,

)

CREATE TABLE Graduate\_student

(

student\_id INT IDENTITY (1,1) NOT NULL PRIMARY KEY,

director\_id INT NOT NULL FOREIGN KEY REFERENCES Scientific\_director,

direction\_id INT NOT NULL FOREIGN KEY REFERENCES Direction,

student\_name VARCHAR (50),

date\_of\_birth DATE,

awards INT,

diploms VARCHAR (50)

)

CREATE TABLE Publication

(

publication\_id INT IDENTITY (1,1) NOT NULL PRIMARY KEY,

student\_id INT NOT NULL FOREIGN KEY REFERENCES Graduate\_Student,

publication\_name VARCHAR (50),

publication\_resource VARCHAR (50),

publication\_date DATE

)

CREATE TABLE Defending

(

defending\_id INT IDENTITY (1,1) NOT NULL PRIMARY KEY,

council\_id INT NOT NULL FOREIGN KEY REFERENCES Science\_council,

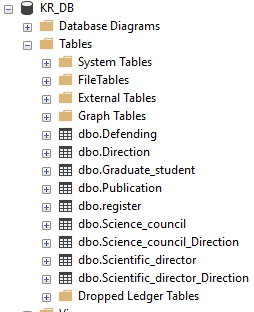
student\_id INT NOT NULL FOREIGN KEY REFERENCES Graduate\_Student,

defending\_date DATE,

council\_decision VARCHAR(50)

)

**Представление таблиц:**



## Реализация правил ссылочной целостности

Триггеры, которые реализуют каскадное изменение полей в связанных записях в таблицах:

* ***Scientific\_director\_Direction по director\_id из Scientific\_director***

CREATE TRIGGER director\_direction

ON Scientific\_director

FOR UPDATE AS

BEGIN

DECLARE @told int

DECLARE @tnew int

SELECT @told = director\_id FROM deleted

SELECT @tnew = director\_id FROM inserted

UPDATE Scientific\_director\_Direction

SET Scientific\_director\_Direction.director\_id = @tnew

WHERE Scientific\_director\_Direction.director\_id = @told

END

* ***Scientific\_director\_Direction по direction\_id из Direction***

CREATE TRIGGER direction\_director

ON Direction

FOR UPDATE AS

BEGIN

DECLARE @told int

DECLARE @tnew int

SELECT @told = direction\_id FROM deleted

SELECT @tnew = direction\_id FROM inserted

UPDATE Scientific\_director\_Direction

SET Scientific\_director\_Direction.direction\_id = @tnew

WHERE Scientific\_director\_Direction.direction\_id = @told

END

* ***Science\_council\_Direction по council\_id из Science\_council***

CREATE TRIGGER council

ON Science\_council

FOR UPDATE AS

BEGIN

DECLARE @told int

DECLARE @tnew int

SELECT @told = council\_id FROM deleted

SELECT @tnew = council\_id FROM inserted

UPDATE Science\_council\_Direction

SET Science\_council\_Direction .council\_id = @tnew

WHERE Science\_council\_Direction .council\_id = @told

END

* ***Science\_council\_Direction по direction\_id из Direction***

CREATE TRIGGER direction\_council

ON Direction

FOR UPDATE AS

BEGIN

DECLARE @told int

DECLARE @tnew int

SELECT @told = direction\_id FROM deleted

SELECT @tnew = direction\_id FROM inserted

UPDATE Science\_council\_Direction

SET Science\_council\_Direction .direction\_id = @tnew

WHERE Science\_council\_Direction .direction\_id = @told

END

* ***Graduate\_student по director\_id из Scientific\_director***

CREATE TRIGGER director\_student

ON Scientific\_director

FOR UPDATE AS

BEGIN

DECLARE @told int

DECLARE @tnew int

SELECT @told = director\_id FROM deleted

SELECT @tnew = director\_id FROM inserted

UPDATE Graduate\_student

SET Graduate\_student.director\_id = @tnew

WHERE Graduate\_student.director\_id = @told

END

* ***Graduate\_student по direction\_id из Direction***

CREATE TRIGGER direction\_student

ON Direction

FOR UPDATE AS

BEGIN

DECLARE @told int

DECLARE @tnew int

SELECT @told = direction\_id FROM deleted

SELECT @tnew = direction\_id FROM inserted

UPDATE Graduate\_student

SET Graduate\_student.direction\_id = @tnew

WHERE Graduate\_student.direction\_id = @told

END

* ***Publication по student\_id из Graduate\_Student***

CREATE TRIGGER student\_publication

ON Graduate\_student

FOR UPDATE AS

BEGIN

DECLARE @told int

DECLARE @tnew int

SELECT @told = student\_id FROM deleted

SELECT @tnew = student\_id FROM inserted

UPDATE Publication

SET Publication.student\_id = @tnew

WHERE Publication.student\_id = @told

END

* ***Defending по council\_id из Science\_council***

CREATE TRIGGER council\_defending

ON Science\_council

FOR UPDATE AS

BEGIN

DECLARE @told int

DECLARE @tnew int

SELECT @told = council\_id FROM deleted

SELECT @tnew = council\_id FROM inserted

UPDATE Publication

SET Defending.council\_id = @tnew

WHERE Defending.council\_id = @told

END

* ***Defending по student\_id из Graduate\_Student***

CREATE TRIGGER student\_defending

ON Graduate\_student

FOR UPDATE AS

BEGIN

DECLARE @told int

DECLARE @tnew int

SELECT @told = student\_id FROM deleted

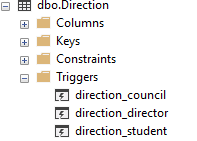
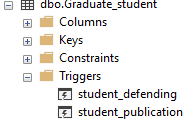
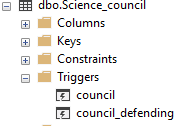
SELECT @tnew = student\_id FROM inserted

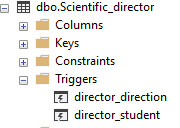
UPDATE Publication

SET Defending.student\_id = @tnew

WHERE Defending.student\_id = @told

END



## Хранимые процедуры

• учет сведений об аспирантах ВУЗа различных категорий (очные, заочные, соискатели, докторанты);

CREATE PROCEDURE students\_by\_category

@input varchar(20)

AS

BEGIN

select Graduate\_student.student\_id, Scientific\_director.director\_name,

Direction.direction\_name, Graduate\_student.student\_name, Graduate\_student.date\_of\_birth,

Graduate\_student.awards, Graduate\_student.diploms, Direction.category

from Graduate\_student

inner join Scientific\_director on Graduate\_student.director\_id = Scientific\_director.director\_id

inner join Direction on Graduate\_student.direction\_id = Direction.direction\_id

where Direction.category like '%' + @input +'%'

END

• формирование списка аспирантов по кафедрам и по научным руководителям;

CREATE PROCEDURE students\_by\_department

@input varchar(20)

AS

BEGIN

select Graduate\_student.student\_id, Scientific\_director.director\_name,

Direction.direction\_name, Graduate\_student.student\_name,

Graduate\_student.date\_of\_birth, Graduate\_student.awards,

Graduate\_student.diploms, Direction.direction\_department

from Graduate\_student

inner join Scientific\_director on Graduate\_student.director\_id = Scientific\_director.director\_id

inner join Direction on Graduate\_student.direction\_id = Direction.direction\_id

where Direction.direction\_department like '%'+ @input + '%'

order by Scientific\_director.director\_name

END

CREATE PROCEDURE students\_by\_director

@input varchar(20)

AS

BEGIN

select Graduate\_student.student\_id, Scientific\_director.director\_name,

Direction.direction\_name, Graduate\_student.student\_name,

Graduate\_student.date\_of\_birth, Graduate\_student.awards,

Graduate\_student.diploms, Direction.direction\_department

from Graduate\_student

inner join Scientific\_director on Graduate\_student.director\_id = Scientific\_director.director\_id

inner join Direction on Graduate\_student.direction\_id = Direction.direction\_id

where Scientific\_director.director\_name like '%' + @input + '%'

order by Direction.direction\_department

END

• учет научных публикаций аспирантов (у одной публикации только один автор);

CREATE PROCEDURE student\_publications

@input varchar(20)

AS

BEGIN

select Publication.publication\_id, Graduate\_student.student\_name, Publication.publication\_resource, Publication.publication\_date, Scientific\_director.director\_name

from Publication

inner join Graduate\_student on Publication.student\_id = Graduate\_student.student\_id

inner join Scientific\_director on Graduate\_student.director\_id = Scientific\_director.director\_id

where Graduate\_student.student\_name like '%' + @input + '%'

group by Publication.publication\_name, Publication.publication\_id, Graduate\_student.student\_name, Publication.publication\_resource, Publication.publication\_date, Scientific\_director.director\_name

END

• учет научных публикаций научного руководителя аспиранта;

CREATE PROCEDURE director\_publications

@input varchar(20)

AS

BEGIN

select Publication.publication\_id, Graduate\_student.student\_name, Publication.publication\_resource, Publication.publication\_date, Scientific\_director.director\_name

from Publication

inner join Graduate\_student on Publication.student\_id = Graduate\_student.student\_id

inner join Scientific\_director on Graduate\_student.director\_id = Scientific\_director.director\_id

where Scientific\_director.director\_name like '%' + @input + '%'

group by Publication.publication\_name, Publication.publication\_id, Graduate\_student.student\_name, Publication.publication\_resource, Publication.publication\_date, Scientific\_director.director\_name

END

• учет работы научных советов.

CREATE PROCEDURE council\_defendings\_by\_date

@input varchar(20)

AS

BEGIN

declare @d DATE = Convert(DATE,@input, 104)

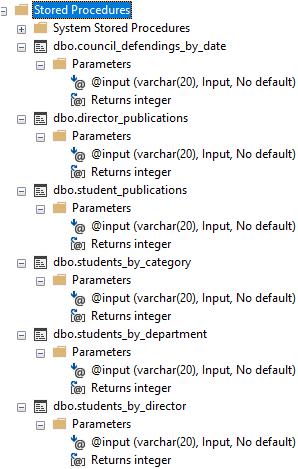
select Science\_council.council\_id, Science\_council.composition, Science\_council.number\_of\_successfull\_defendings, Science\_council.total\_number\_of\_defendings, Defending.defending\_date

from Science\_council

inner join Defending on Science\_council.council\_id = Defending.council\_id

where Defending.defending\_date = @d

END



• составление отчетов о состоявшихся защитах (по различным категориям, научным направлениям, кафедрам, научным руководителям и т.п.) – **Реализовано отдельно средствани языка С#**:

buf = textBox1.Text + report.qString3;

string search =report.qString1 +

$"select Defending.defending\_id, Graduate\_student.student\_name, Defending.defending\_date, Scientific\_director.director\_name, Science\_council.composition, Direction.direction\_department," +

$" Direction.direction\_name, Direction.category, Defending.council\_decision" +

$" from Defending" +

$" inner join Science\_council on Defending.council\_id = Science\_council.council\_id" +

$" inner join Graduate\_student on Defending.student\_id = Graduate\_student.student\_id" +

$" inner join Direction on Graduate\_student.direction\_id = Direction.direction\_id" +

$" inner join Scientific\_director on Graduate\_student.director\_id = Scientific\_director.director\_id " +

report.qString2 + buf;//\*/

SqlCommand com = new SqlCommand(search, database.GetConnection());

database.OpenConnection();

SqlDataReader read = com.ExecuteReader();

while (read.Read())

{

ReadSingleRow\_Report(dgv, read);

}

read.Close();

report.qString1 = "";

где report.qString1 и report.qString2 и report.qString3 – параметры, зависящие от выбора критерия поиска. Пример:

qString1 = $"declare @d DATE = Convert(DATE, '";

qString2 = $"where Defending.defending\_date =";

QueryWindow qw = new QueryWindow();

qw.QueryLabel.Text = "По дате защиты";

qw.report = this;

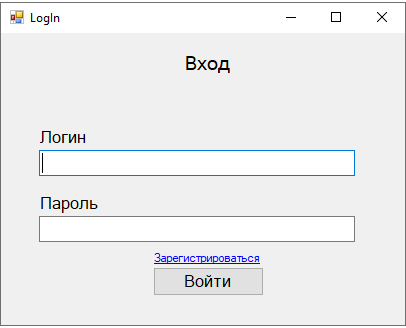
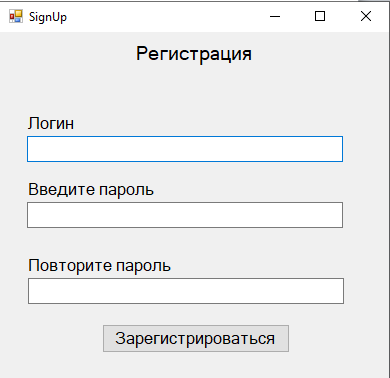
qw.Show();

Hide();

# Программная реализация

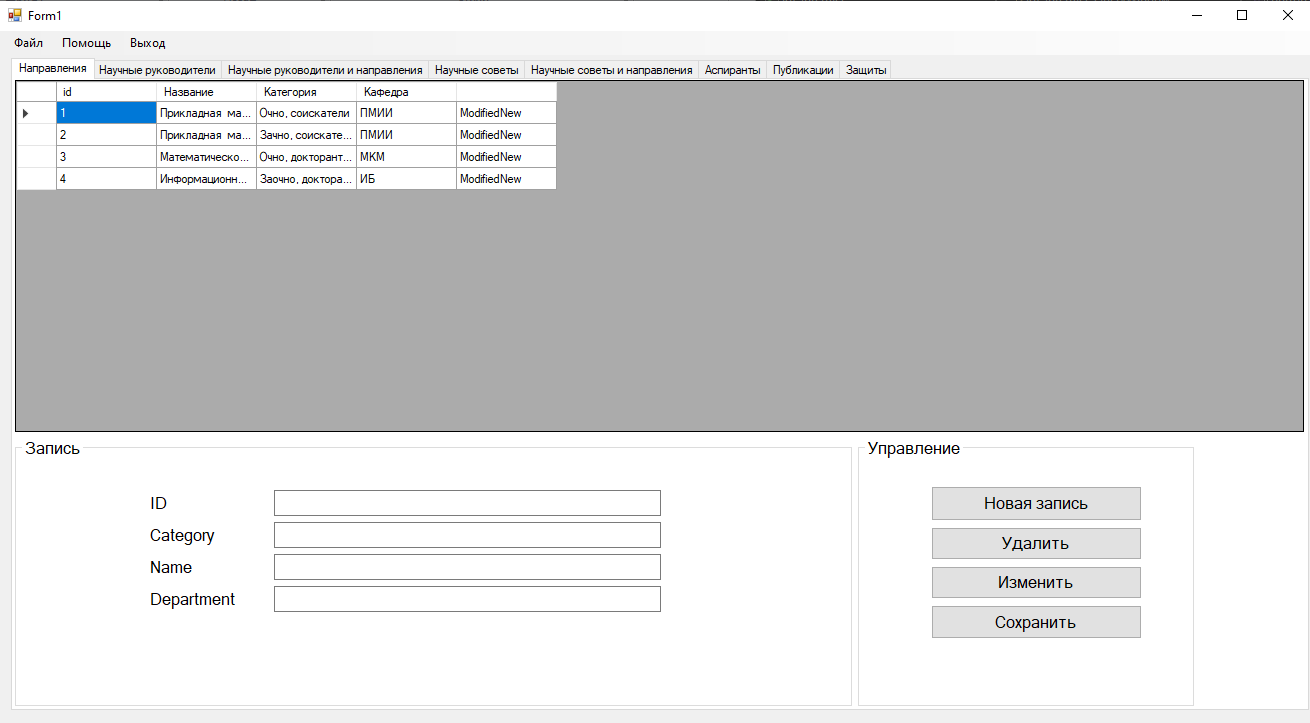
В качестве языка разработки мною выбран C#, в качестве среды разработки – Visual Studio 2022.Тип приложения – Windows Forms - это платформа пользовательского интерфейса для создания классических приложений Windows. Она обеспечивает один из самых эффективных способов создания классических приложений с помощью визуального конструктора в Visual Studio. Такие функции, как размещение визуальных элементов управления путем перетаскивания, упрощают создание классических приложений.

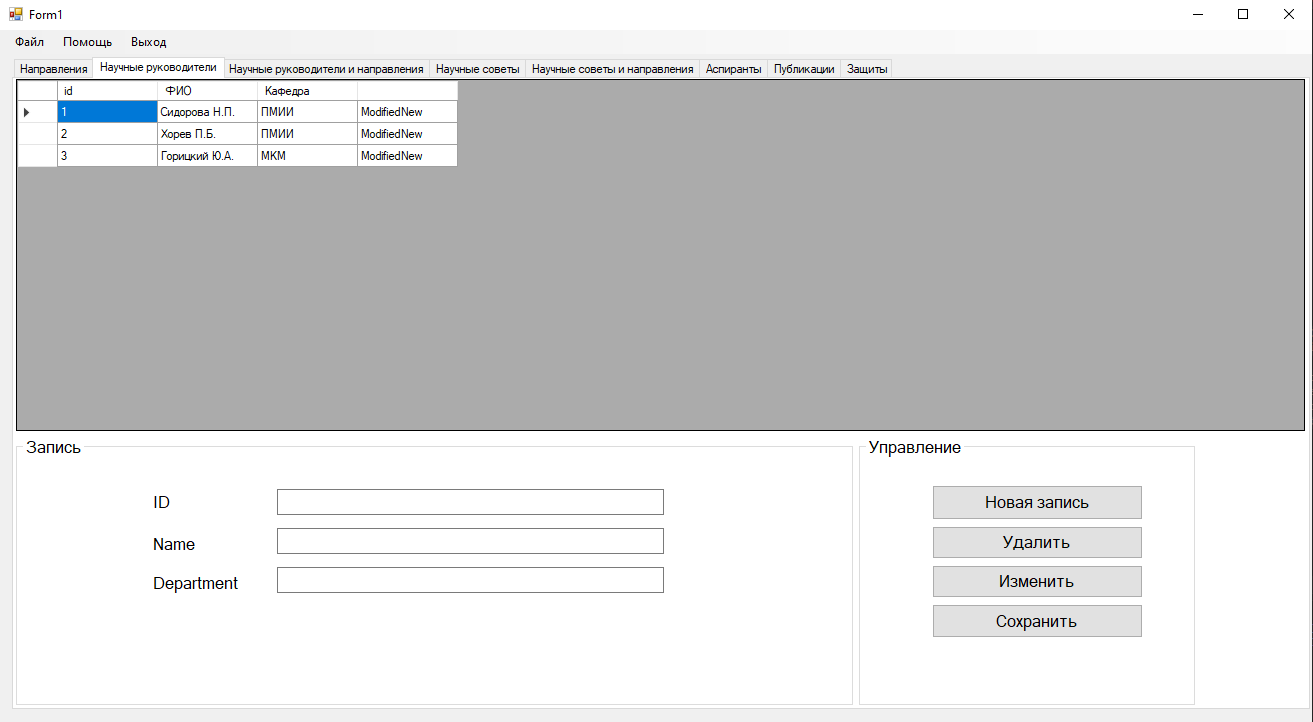
## Окна регистрации и авторизации

При нажатии кнопки «Зарегистрироваться», проверяется совпадение паролей в полях ввода, запись данных пользователя в отдельную таблицу БД и производится переход в окно входа. При нажатии кнопки «Войти» проверяется правильность введенных данных и производится переход в главное окно приложения.

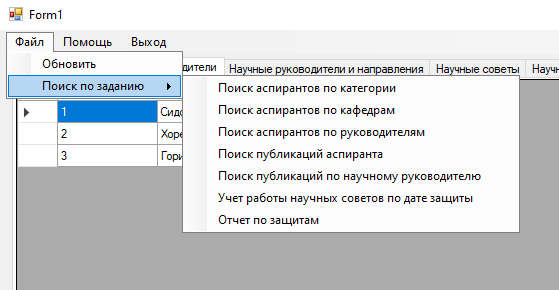
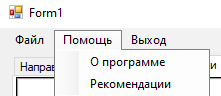
## Главное окно

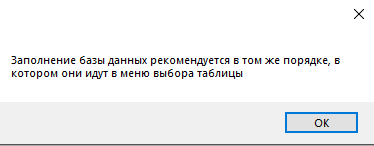
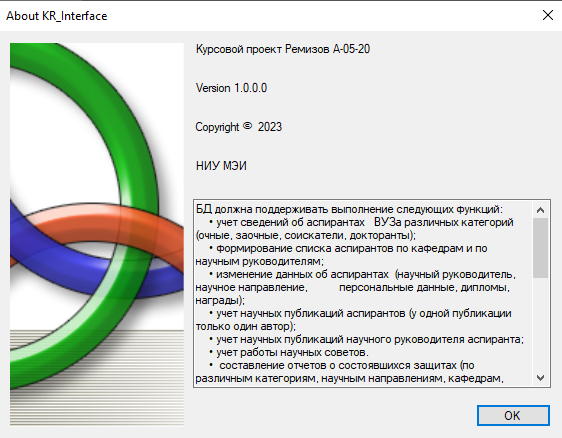




Выше показаны примеры отображения формы (в каждой вкладке соответствующая таблица)

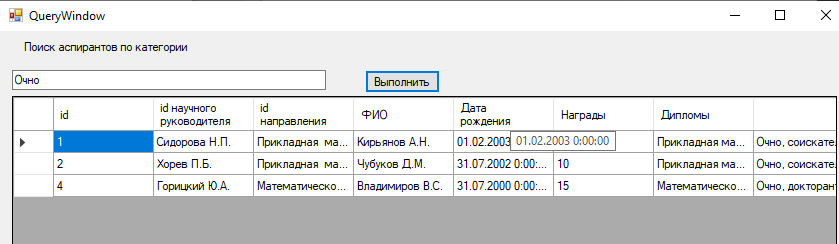
Интерфейс управления: кнопка «Обновить» обновляет все таблицы, кнопка «Поиск по заданию» производит переход в окно запроса; кнопка «О программе» производит переход в окно информации о задании (предметная область, автор и т.д.), кнопка «Рекомендации» вызывает окно с рекомендациями по заполнению БД; кнопка «Выход» закрывает главное окно приложения и открывает окно авторизации.

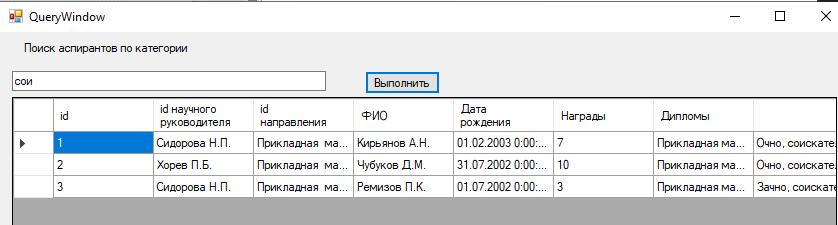
 

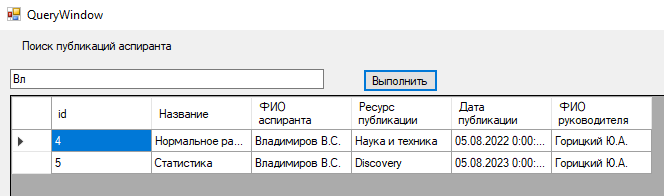
 

## Окно запросов

В зависимости от выбранного режима поиска окно запросов выполняет поиск по различным критериям.



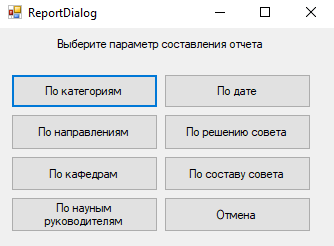


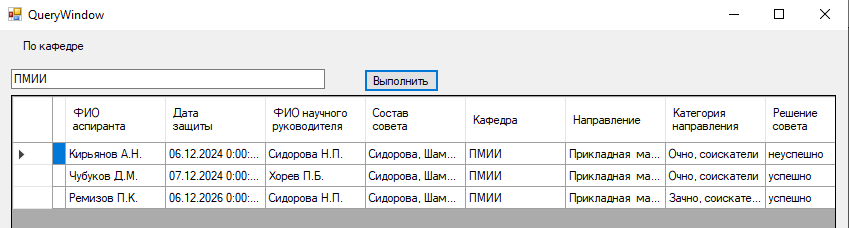


Примеры режимов поиска

## Окно отчетов

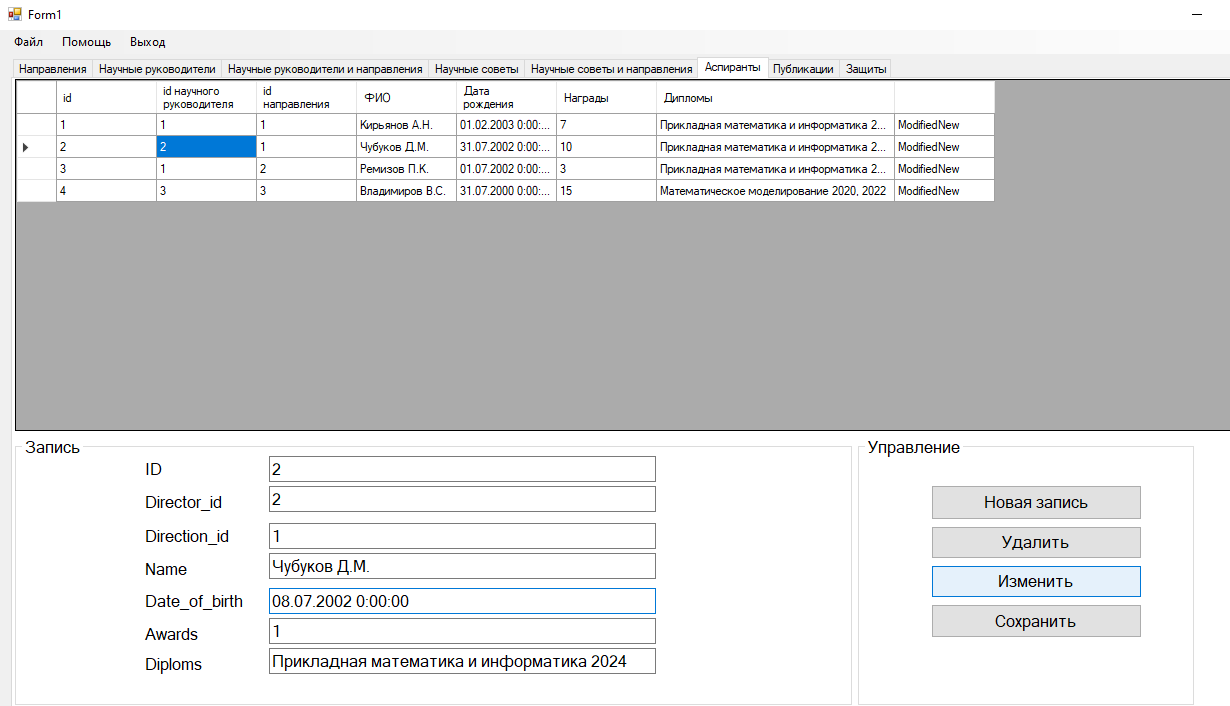
Режим отчета по защитам имеет собственное окно выбора критерия поиска



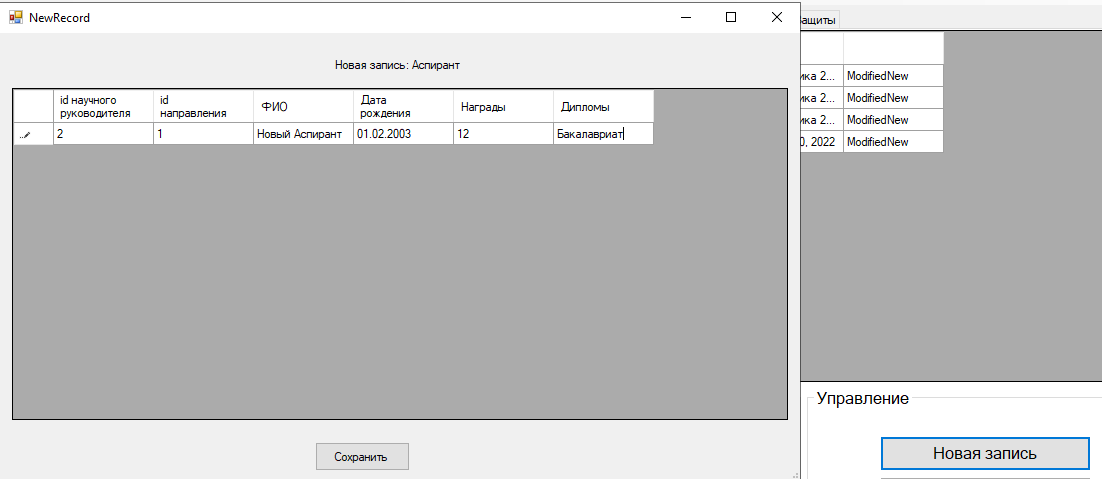


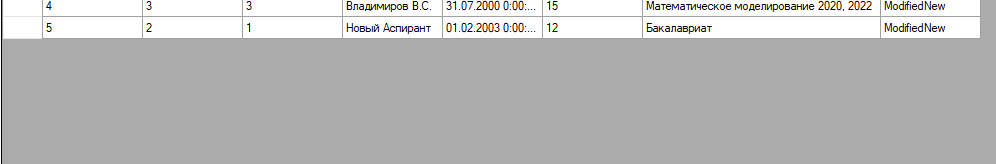
## Изменение данных аспирантов

Информация об аспирантах до изменения

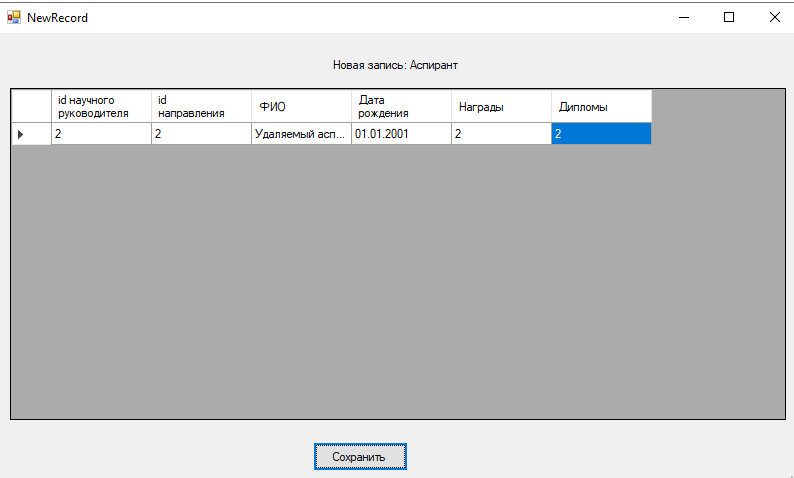


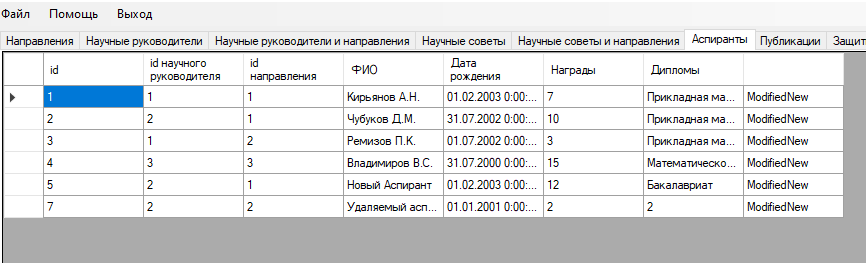
При нажатии кнопки «Новая запись» открывается окно ввода новой записи. Кнопка «Сохранить» сохраняет новую запись.



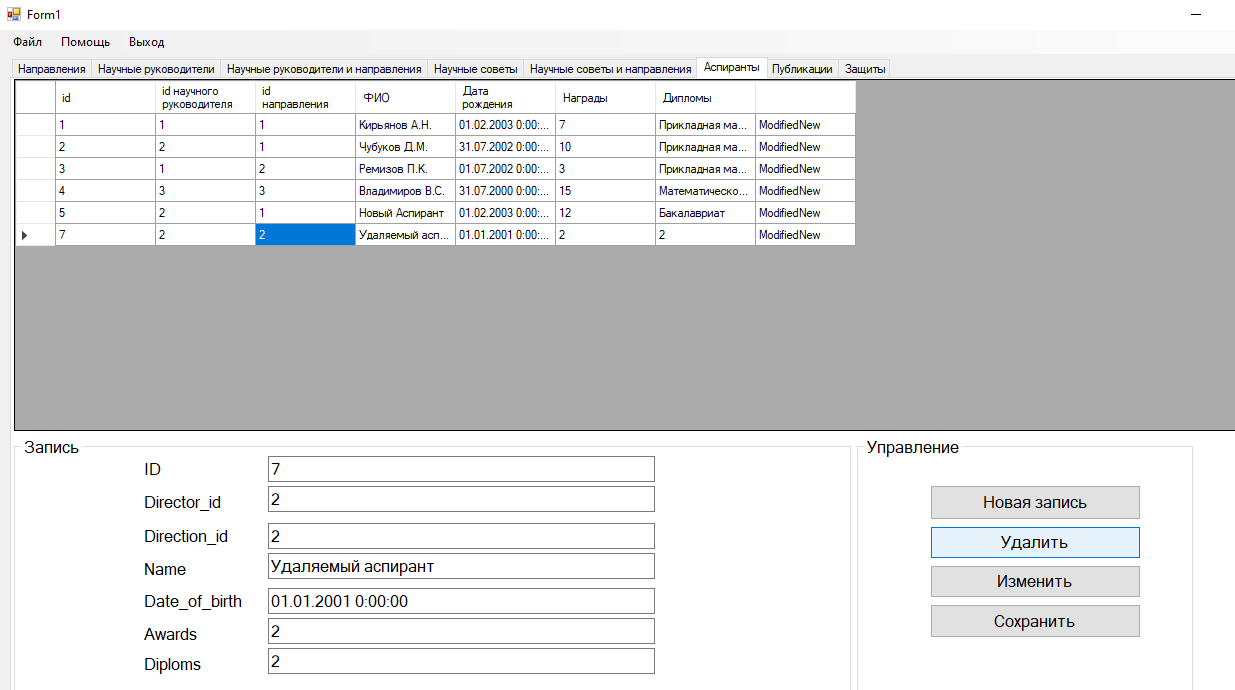


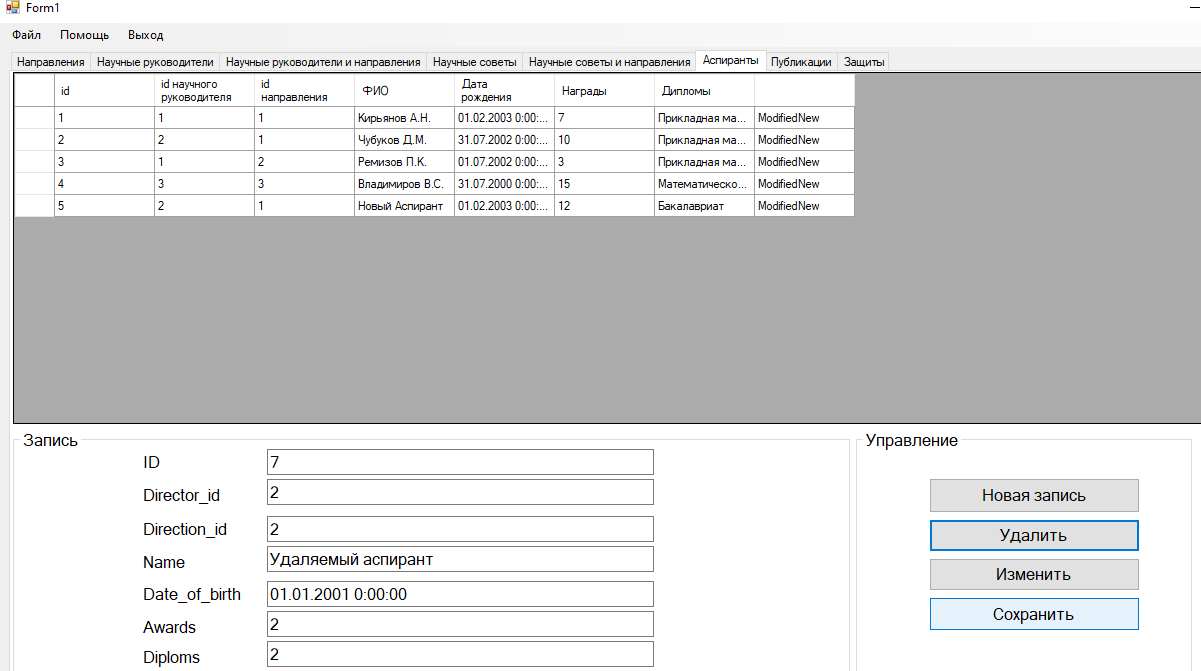
Создадим запись «Удаленный аспирант».



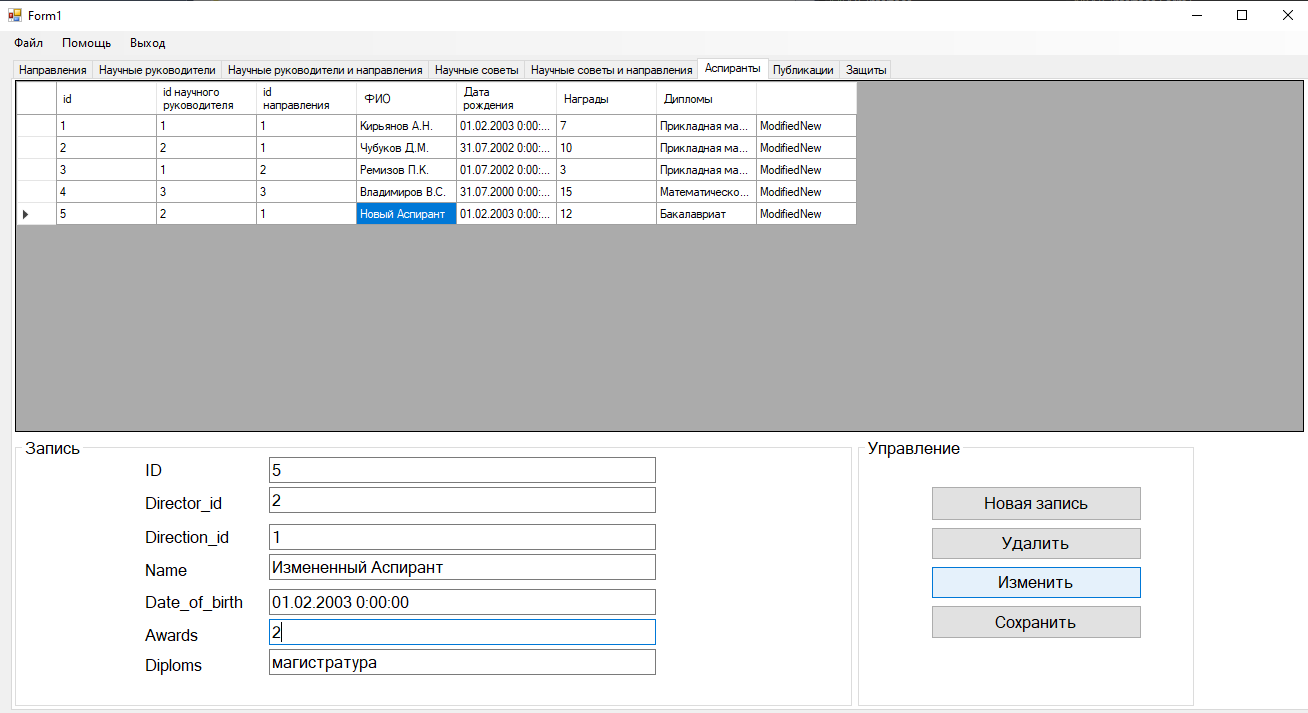


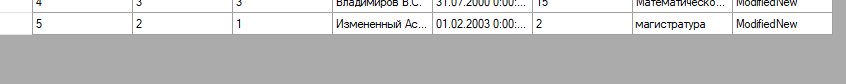
При выборе строки информация из таблицы передается в поля в нижней части окна. Нажатием кнопки «Удалить» удалим запись об аспиранте. Нажав кнопку «Обновить» увидим, что запись удалена.





Выберем ячейку с записью «Новый аспирант». В поля в нижней части окна введем новую информацию об аспиранте. Нажатием кнопки «Изменить» меняется информация в таблице. При нажатии кнопки «Сохранить» новая информация будет сохранена. При нажатии кнопки «Обновить» увидим, что новые данные были применены к записи





# Заключение

В ходе выполнения курсовой работы по разработке базы данных для отдела аспирантуры:

* был проведен анализ предметной области для БД и выделены основные информационные объекты;
* разработана ER – модель БД и описаны сущности;
* разработана реляционная модель БД, приведенная в 3-ю нормальную форму и включающая в себя 8 таблиц;
* разработана БД в среде Microsoft SQL Server, содержащая 8 таблиц, 9 тригеров, 6 хранимых процедур;
* реализованы функции БД с использованием хранимых процедур и класса SqlCommand;
* разработано приложение с БД на языке С# в IDE Visual Studio 2022.