

Постановка задачи

Исходная постановка задачи:

Самодеятельность со списком кружков (руководитель, список учеников кружка), списком учеников (список кружков ученика), списком руководителей (список кружков руководителя, список учеников руководителя). Состояния ученика:

- 1) желание – определение списка кружков;
- 2) поступление – включение в списки кружков;
- 3) уход – исключение из кружка

Формализация постановки задачи:

В задаче обозначаются 2 связанных сущности: студент и кружок. Студент может исполнять 2 роли по отношению к каждому из кружков (может не исполнять ни одной): руководитель кружка и записанный на кружок ученик.

На первом этапе необходимо спроектировать модель базы данных, которая будет содержать сущности студента и кружка и будет верно описывать существующую между ними связь.

На последующих этапах необходимо реализовать функциональную часть работы с базой данных:

- Студент может записаться на любые доступные ему кружки (которые он не возглавляет и в которых не состоит).
- Студент может покинуть любые из кружков, на которые он записан.
- Если студент возглавляет какой-либо кружок, то он не может входить в список учеников кружка.
- Реализована возможность удаления студента, доступная тогда и только тогда, когда студент не руководит ни одним кружком.
- При создании каждого кружка необходимо выбрать (или назначить) руководителя создаваемого кружка.
- Название каждого кружка должно быть уникальным.
- У любого существующего кружка можно изменить руководителя, выбрав нового из числа посещающих данный кружок студентов.
- При удалении кружка должны удаляться все записи, связанные с учениками, посещающими данный кружок.

В качестве среды разработки используется Visual Studio, СУБД – Microsoft SQL Server. Приложение реализовано в Microsoft Windows Forms .NET Framework.

Проектирование БД

В модели базы данных использованы 3 сущности: Студент (STUDENT), Кружок (CLUB), и вспомогательная сущность (STUDENT_CLUB), созданная для реализации связи многие ко многим между первыми двумя сущностями.

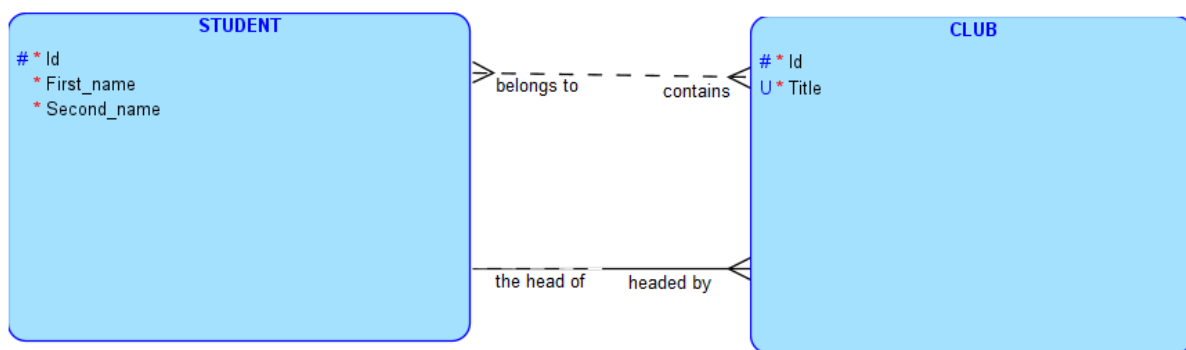
Сущность студента (STUDENT) содержит обязательные атрибуты: уникальный номер студента (UID сущности), имя и фамилию студента.

Сущность кружка (CLUB) содержит обязательные атрибуты: уникальный номер кружка (UID сущности) и уникальное название кружка (candidate UID).

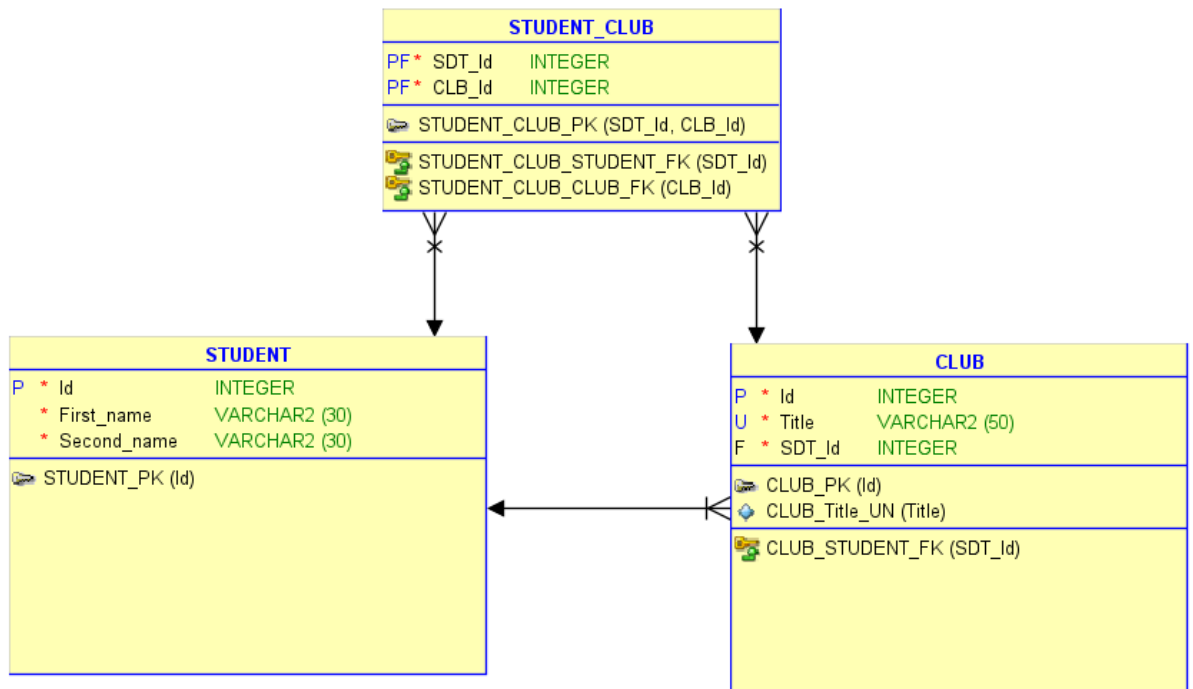
Между сущностями STUDENT и CLUB установлена связь один ко многим: один студент может руководить несколькими кружками (связь на стороне STUDENT необязательная), а каждый кружок управляется ровно одним студентом (связь на стороне CLUB обязательная).

Также между сущностями STUDENT и CLUB установлена связь многие-ко-многим: в одном кружке могут состоять несколько студентов (связь на стороне CLUB необязательная), а каждый студент может состоять в нескольких кружках (связь на стороне STUDENT необязательная). (При установлении данной связи в реляционной модели была автоматически сформирована новая сущность, переименованная в STUDENT_CLUB)

Концептуальная модель (ERD)



Реляционная модель (схема БД)



DDL операторы (SQL Server 2012)

```
CREATE TABLE CLUB
```

```
(
    Id INTEGER IDENTITY(1,1) NOT NULL ,
    Title VARCHAR (50) NOT NULL ,
    SDT_Id INTEGER NOT NULL
)
```

```
GO
```

```
ALTER TABLE CLUB ADD CONSTRAINT CLUB_PK PRIMARY KEY CLUSTERED (Id)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
```

```
GO
```

```
ALTER TABLE CLUB ADD CONSTRAINT CLUB_Title_UN UNIQUE NONCLUSTERED
(Title)
```

```
GO
```

```
CREATE TABLE STUDENT
```

```
(
    Id INTEGER IDENTITY(1,1) NOT NULL ,
    First_name VARCHAR (30) NOT NULL ,
    Second_name VARCHAR (30) NOT NULL
)
```

```
GO
```

```
ALTER TABLE STUDENT ADD CONSTRAINT STUDENT_PK PRIMARY KEY CLUSTERED
(Id)
```

```
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
```

```
GO
```

```
CREATE TABLE STUDENT_CLUB
(
    SDT_Id INTEGER NOT NULL ,
    CLB_Id INTEGER NOT NULL
)
GO
```

```
ALTER TABLE STUDENT_CLUB ADD CONSTRAINT STUDENT_CLUB_PK PRIMARY KEY
CLUSTERED (SDT_Id, CLB_Id)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO
```

```
ALTER TABLE CLUB
ADD CONSTRAINT CLUB_STUDENT_FK FOREIGN KEY
(
    SDT_Id
)
REFERENCES STUDENT
(
    Id
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO
```

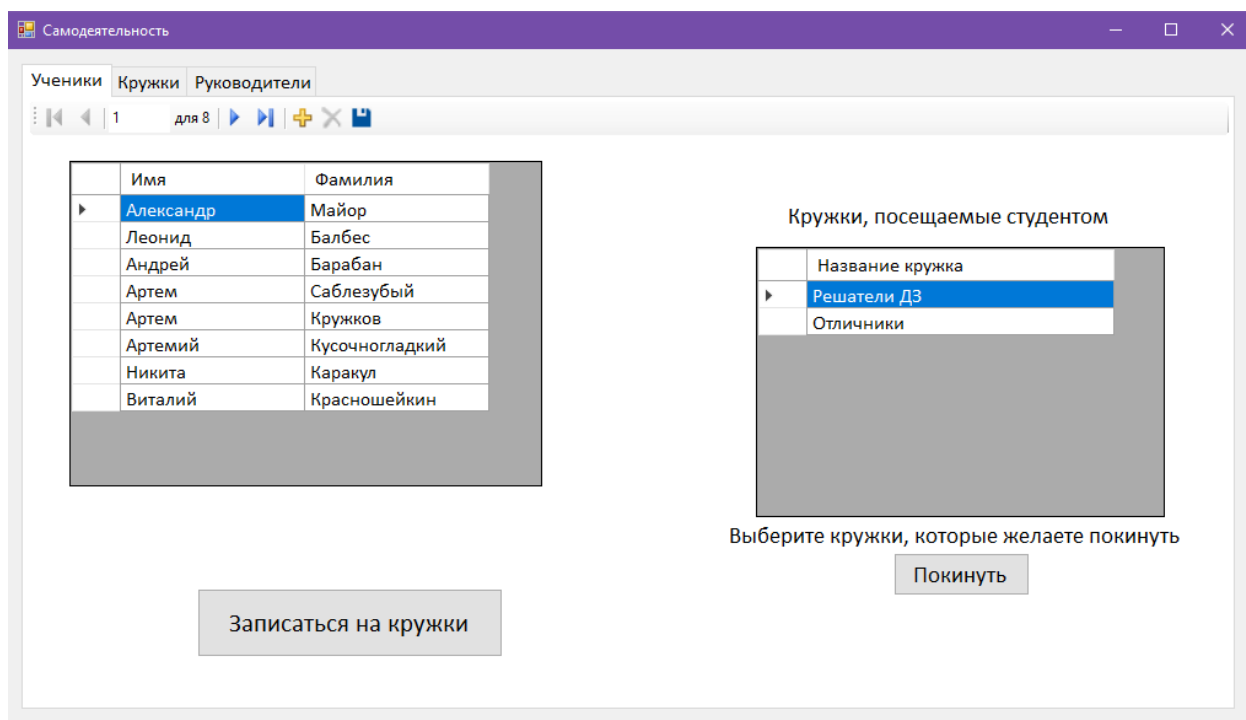
```
ALTER TABLE STUDENT_CLUB
ADD CONSTRAINT STUDENT_CLUB_CLUB_FK FOREIGN KEY
(
    CLB_Id
)
REFERENCES CLUB
(
    Id
)
ON DELETE CASCADE
ON UPDATE NO ACTION
GO
```

```
ALTER TABLE STUDENT_CLUB
ADD CONSTRAINT STUDENT_CLUB_STUDENT_FK FOREIGN KEY
(
    SDT_Id
)
REFERENCES STUDENT
(
    Id
)
ON DELETE CASCADE
```

Пользовательский интерфейс

Оконное приложение состоит из главного окна, которое представляет собой область с тремя вкладками – “Ученики”, “Кружки”, “Руководители”. Каждая вкладка реализует показ соответствующих списков из исходной постановки задачи с возможностью редактировать все основные списки (добавлять, удалять, редактировать записи) и с некоторым дополнительным функционалом. После ручного редактирования (вышеперечисленные операции) данных в основной таблице соответствующей вкладки следует нажать кнопку “Сохранить”.

Вкладка “Ученики”



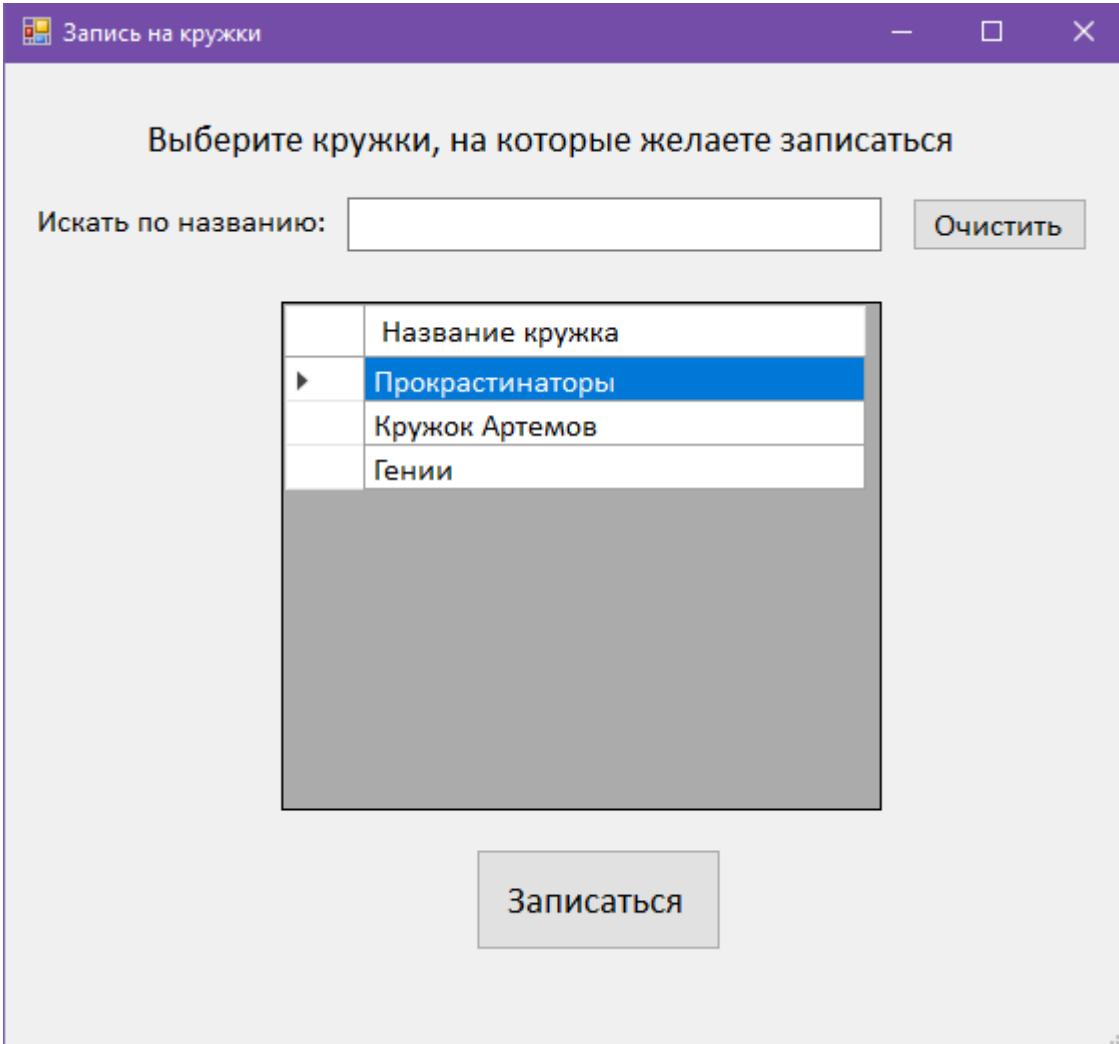
На первой вкладке основная таблица представляет список учеников, переключаясь между которыми автоматически заполняется таблица посещаемых студентом кружков и становятся активными/неактивными кнопки покидания кружков и записи на кружки.

Как было сказано ранее, основную таблицу можно редактировать, но необходимо придерживаться следующих правил (нарушить их система не даст):

- все ячейки имени и фамилии должны быть не пусты
- нельзя удалить ученика, который руководит хотя бы одним кружком (кнопка удаления станет неактивной при выборе такового)

Чтобы взаимодействовать с кнопкой “Покинуть”, пользователь должен быть записан на хотя бы 1 кружок и необходимо выбрать СТРОКИ кружков, которые хотим покинуть (в DataGridView для этого нужно нажать на любую строку самой левой пустой колонки таблицы и потянуть, удерживая ЛКМ / либо с нажатой клавишей Ctrl выбрать желаемые строки в этой колонке). Выбрав кружки, нажимаем “Покинуть”, после чего появится всплывающее окно с вопросом, уверены ли мы, что хотим покинуть выбранные кружки. Нажимаем “Да” и изменения сразу сохраняются в базу / нажимаем “Нет” и ничего не происходит.

Чтобы взаимодействовать с кнопкой “Записаться на кружки”, ученик должен находиться в базе данных (если выбранный ученик не сохранен, об этом подскажет метка с текстом красного цвета). Нажимаем “Записаться на кружки” и открывается форма выбора кружков для записи (в которых выбранный ученик не состоит и которыми он не руководит).



Запись на кружки

Выберите кружки, на которые желаете записаться

Искать по названию:

| | Название кружка |
|---|-----------------|
| ▶ | Прокрастинаторы |
| | Кружок Артемов |
| | Гении |

Здесь мы можем отфильтровать доступные выбранному ученику кружки по названию, после чего необходимо выбрать СТРОКИ кружков, на которые хотим записаться и нажимаем “Записаться”. Изменения сразу сохраняются в

базу, вспомогательное окно формы закроется. Если закроем данную форму нажатием на крестик, то ничего не произойдёт.

Вкладка “Кружки”

Самодетальность

Ученики | **Кружки** | Руководители

1 для 6

| Название |
|-----------------|
| Решатели ДЗ |
| Бегуны |
| Прокрастинаторы |
| Кружок Артемов |
| Гении |
| Отличники |

Руководитель: Леонид Балбес

Сменить руководителя

Список учеников кружка

| Имя | Фамилия |
|-----------|----------------|
| Александр | Майор |
| Артем | Саблезубый |
| Артемий | Кусочноглазкий |

На второй вкладке основная таблица представляет список кружков, переключаясь между которыми автоматически заполняется таблица учеников данного кружка и становится активной/неактивной кнопка смены руководителя текущего кружка.

Как было сказано ранее, основную таблицу можно редактировать, но необходимо придерживаться следующих правил (нарушить их система не даст):

- все ячейки названий кружков должны быть не пусты
- все названия кружков должны быть уникальны

При создании нового кружка (кнопка “Add” в навигаторе сверху) сразу же открывается форма добавления кружка. На ней нужно выбрать руководителя создаваемого кружка, (будут выведены все текущие ученики, сохранённые в базе), а также придумать уникальное название кружка.

Добавление кружка

Выберите руководителя

Искать по фамилии:

| | Имя | Фамилия |
|---|---------|--------------|
| ► | Никита | Каракул |
| | Миша | Махачкалахов |
| | Подлиза | Чабудка |
| | Юра | Косичка |
| | Толя | Стесняшка |

Введите название кружка

Клуб с таким названием уже существует!

Можно отфильтровать учеников по фамилии. Для выбора кандидата достаточно нажать на любую ячейку. Когда руководитель и название выбраны, нажимаем на кнопку “Создать”, после чего произойдёт одно из двух: если кружок с таким названием уже существует в базе, то появится предупреждающая метка с текстом красного цвета; иначе вспомогательное окно с формой закрывается, и можно сохранить добавленный кружок в базе. Если мы закрыли форму добавления по крестику, то создание нового кружка отменится.

Чтобы взаимодействовать с кнопкой “Сменить руководителя”, кружок должен находиться в базе данных (если выбранный кружок не сохранен, об этом подскажет метка с текстом красного цвета), а также данный кружок должен посещать хотя бы один ученик, так как новый руководитель может выбираться только среди учеников данного кружка. Нажимаем “Сменить руководителя” и открывается форма выбора нового руководителя кружка.

Смена руководителя

Выберите нового руководителя

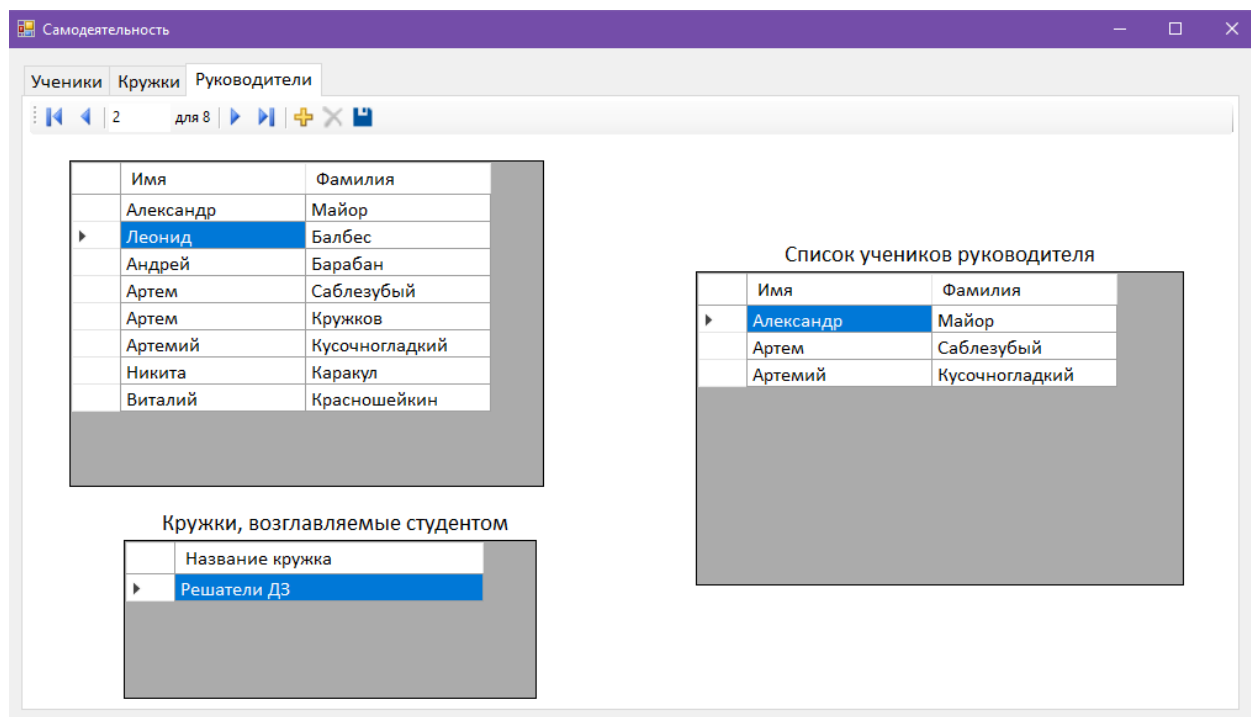
Искать по фамилии:

| | Имя | Фамилия |
|---|-----------|----------------|
| ▶ | Александр | Майор |
| | Артем | Саблезубый |
| | Артемий | Кусочногладкий |

Можно отфильтровать учеников по фамилии. При выборе кандидата (можно нажать на любую ячейку), жмём “*Выбрать*”, после чего вспомогательное окно с формой закрывается, изменения сразу сохраняются в базу, а также при этом сохраняются все несохраненные действия по редактированию основной таблицы второй вкладки (таблица кружков). Если мы закрыли форму смены руководителя по крестику, то ничего не произойдёт (действие по смене руководителя данного кружка отменится).

Вкладка “Руководители”

На третьей вкладке основная таблица представляет список учеников, переключаясь между которыми автоматически заполняется таблица руководимых студентом кружков и таблица со списком учеников данного студента. Основная таблица этой вкладки (руководители) тесно связана с основной таблицей первой вкладки, в частности текущий ученик общий для обеих вкладок, а также общие кнопки добавления, удаления и сохранения.



При редактировании таблицы “руководители” нужно придерживаться тех же правил, что и в таблице “ученики” (нарушить их система не даст):

- все ячейки имени и фамилии должны быть не пусты
- нельзя удалить ученика, который руководит хотя бы одним кружком (кнопка удаления станет неактивной при выборе такового)

Тестирование системы

1. Успешное добавление и удаление записей. Нажать на кнопку «+» в навигаторе сверху вкладки (в основной таблице любой вкладки) и попытаться заполнить её некорректными данными (превышена длина строки, незаполнение ячеек). Удалить добавленные записи (через кнопку удаления в навигаторе). Убедиться, что нельзя удалить учеников, возглавляющих хотя бы 1 кружок (это хорошо видно на вкладке «Руководители»).
2. Отображение таблиц. Переключаться между учениками / кружками / руководителями, наблюдая за изменением заполнения информационных таблиц. Посмотреть на отображение этих таблиц в сценариях добавления новых записей / удаления / сохранения / редактирования записей.
3. Доступность кнопок. Переключаться между учениками / кружками / руководителями, наблюдая за изменением доступности всех кнопок. Посмотреть на отображение этих кнопок в сценариях добавления новых записей / удаления / сохранения / редактирования записей.
4. Запись на кружки.

- Выбрать ученика «Александр МайорПейн» и нажать на кнопку «*Записаться на кружки*».
- В появившейся форме выбрать кружки «Гении» и «Коммивояжёрики» и нажать на кнопку «*Записаться*».

5. Покидание кружков.

- Выбрать ученика «Александр МайорПейн».
- Выбрать кружки «Гении» и «Коммивояжёрики» и нажать на кнопку «*Покинуть*».
- В появившемся всплывающем окне нажать «*Да*»

6. Создание кружка.

- Нажать на кнопку «+» на вкладке «Кружки».
- В появившейся форме выбрать руководителем ученика «Александр МайорПейн» и нажать на кнопку «*Выбрать*».
- Ввести название «Гении» и нажать Enter. Убедиться, что кружок с таким названием уже существует.
- Ввести другое название: «Кружок 1», нажать Enter и кнопку сохранения.

7. Смена руководителя кружка.

- Выбрать кружок «Кружок 1».
- Убедиться, что кнопка «*Сменить руководителя*» недоступна.
- Записать ученика «Леонид Балбес» на кружок «Кружок 1» и нажать на кнопку «*Сменить руководителя*».
- В появившейся форме выбрать руководителем ученика «Александр МайорПейн» и нажать на кнопку «*Выбрать*».

```

USE [MAYOROV_LAB]
GO
/***** Object:  StoredProcedure [dbo].[FetchAvailableClubs]    Script Date:
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: Процедура выводит список кружков (Id и Название),
-- доступных студенту с указанным Id для записи, то есть
-- в которых этот студент не состоит и которыми он не руководит,
-- а также содержащих в названии указанную подстроку (опциональный параметр)
-- =====
ALTER PROCEDURE [dbo].[FetchAvailableClubs]
    -- Add the parameters for the stored procedure here
    @studentId int,
    @title varchar(50) = ''
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT Id, Title
    FROM CLUB
    WHERE Title LIKE '%' + @title + '%'
    EXCEPT
    SELECT DISTINCT Id, Title
    FROM CLUB
    LEFT JOIN STUDENT_CLUB scb
    ON Id = scb.CLB_Id
    WHERE [scb].SDT_Id = @studentId OR [CLUB].SDT_Id = @studentId
    ORDER BY Id;
END

```

```
USE [MAYOROV_LAB]
GO
/***** Object:  StoredProcedure [dbo].[FetchManagedStudents]    Scr
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: Процедура выводит список учеников (Имя, Фамилия),
--              которые посещают кружки, возглавляемые студентом с указанным Id
-- =====
ALTER PROCEDURE [dbo].[FetchManagedStudents]
    -- Add the parameters for the stored procedure here
    @studentId int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT DISTINCT First_name, Second_name
    FROM STUDENT_CLUB scb
    JOIN STUDENT s ON scb.SDT_Id = s.Id
    JOIN CLUB c ON scb.CLB_Id = c.Id
    WHERE c.SDT_Id = @studentId;
END
```

```

USE [MAYOROV_LAB]
GO
/***** Object: StoredProcedure [dbo].[ListStudentsInClub]    Script Date:
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: Процедура выводит список учеников (Id, Имя, Фамилия),
--               которые могут стать новым руководителем кружка с указанным Id, то есть
--               всех учеников, посещающих данный кружок (кроме текущего руководителя)
--               а также содержащих в фамилии указанную подстроку (опциональный параметр)
-- =====
ALTER PROCEDURE [dbo].[ListStudentsInClub]
    -- Add the parameters for the stored procedure here
    @clubId int,
    @secondName varchar(30) = ''
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT Id, First_name, Second_name
    FROM STUDENT s
    INNER JOIN STUDENT_CLUB scb
    ON s.Id = scb.SDT_Id
    WHERE scb.CLB_Id = @clubId AND Second_name LIKE '%' + @secondName + '%';
END

```