

---

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа «Программная инженерия»

## **КУРСОВАЯ РАБОТА**

**Разработка многопользовательской автоматизированной системы  
управления организацией**  
по дисциплине «Базы данных»

Выполнил  
студент  
гр.3530904/00105  
Руководитель

С.А.Пелиниченко

И.А.Гасанова

«3» декабря 2022 г.

Санкт-Петербург  
2022

## СОДЕРЖАНИЕ

### Оглавление

Задание.....	3
Анализ предметной области .....	4
Описание таблиц и связей, схема .....	11
Запросы.....	17
Описание программы .....	26
Работа приложения .....	27
Заключение .....	40
Список литературы .....	41
Приложение.....	41

# Задание

**ЗАДАНИЕ**  
**НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА (КУРСОВОЙ РАБОТЫ)**  
студенту группы 3530904/00105 Пелиниченко Степан Александрович  

---

(номер группы)

---

(фамилия, имя, отчество)

**1. Тема проекта (работы)** *Разработка многопользовательской автоматизированной системы управления организацией.*

Объект автоматизации – кафедра университета . *Задание № 13*

**2. Срок сдачи студентом законченного проекта (работы)** 25.12.2022

**3. Исходные данные к проекту (работе)**

*Описание предметной области*

**4. Содержание пояснительной записки** (перечень подлежащих разработке вопросов: введение, основная часть (раскрывается структура основной части), заключение, список использованных источников, приложения).  
*Введение. Анализ предметной области. Проектирование схемы данных.*

*Реализация базы данных в среде SQL Server. Разработка представлений и хранимых процедур. Разработка клиентского приложения. Тестирование.*

*Заключение. Список использованных источников.*

Примерный объем пояснительной записки 15-20 страниц машинописного текста

**5. Перечень графического материала ( с указанием обязательных чертежей и плакатов)** не предоставляется

**6. Консультанты**

**7. Дата получения задания:** «11» сентября 2022 г.

Руководитель

---

(подпись)

Гасанова И.А.

---

(инициалы, фамилия)

Задание принял к исполнению

---

(подпись)

Пелиниченко С.А.

---

(инициалы, фамилия)

11.09.2022

---

(дата)

## Анализ предметной области

Группы пользователей разрабатываемой информационной системы (ИС)

№ пп	Наименование пользователя
1	Администратор
2	Лектор
3	Студент

Функции групп пользователей.

№ пп	Выполняемая функция	Входные данные	Выходные данные	Функции, которые должны быть реализованы в ИС
<i>Администратор</i>				
1	Регистрация лекторов	first_name; last_name; father_name; email; mobile_phone; passport; dob; salary; login; password; role;	Таблица lectors; Таблица logins;	Добавить запись с новым лектором в таблицу lectors, logins.
2	Регистрация студентов	first_name; last_name; father_name; email; passport; dob; education_form; group_id; login; password; role;	Таблица students; Таблица logins;	Добавить запись с новым студентом в таблицу students, logins.
3	Добавление предмета	subject_name; subject_description;	Таблица subjects;	Добавить запись с новым предметом в таблицу subjects.

4	Формирование группы	group_number; count_person; date_begin;	Таблица groups;	Добавить запись с новой группой в таблицу groups
5	Добавление лектору группу с предметом	lector_id; group_id; subject_id;	Таблица lectors_groups_subjects;	Добавить запись в таблицу lectors_groups_subjects;
6	Изменение предмета	subject_id; subject_name; subject_description;	Таблица subjects;	Изменить соответствующую запись в таблице subjects
7	Изменение личных данных студента	student_id; first_name; last_name; father_name; email; passport; dob; education_form; group_id; login; password;	Таблица students; Таблица logins;	Изменить соответствующую запись в таблице students, logins.
8	Добавление оценки студенту	student_id; passed_date; semester; assessment; lector_group_subject_id;	Таблица assessments;	Добавить запись в таблицу assessments;
9	Изменение информации о конкретной оценке у конкретного студента.	assessment_id; student_id; passed_date; semester; assessment; lector_group_subject_id;	Таблица assessments;	Изменить соответствующую запись в таблице assessments;
10	Просмотр всех студентов		Таблица students;	Вывести все данные из таблицы students;
11	Просмотр всех лекторов		Таблица lectors;	Вывести все данные из таблицы lectors;
12	Просмотр всех предметов		Таблица subjects;	Вывести все данные из таблицы subjects;

1 3	Просмотр всех студентов с их группами.		Таблица groups; Таблица students;	Вывести из таблицы students: first_name, last_name, father_name, education_form, а также из таблицы groups: group_number;
1 4	Просмотр всех оценок		Таблица assessments;	Вывести все данные из таблицы assessments;
1 5	Просмотр всех оценок у конкретного студента	student_id;	Таблица assessments; Таблица lectors; Таблица students;	Вывести поля из таблицы students: first_name, last_name, father_name. Из таблицы subjects: subject_name. Из таблицы assessments: assessment, date_passed, semester
1 6	Просмотр всех данных аутентификации		Таблица logins;	Вывести все записи из таблицы logins
1 7	Добавление новой записи в расписание	day_week; date_par; lector_group_subject_id; format_par; cabinet; housing;	Таблица timetables;	Добавить новую запись в таблицу
1 8	Просмотр всех лекторов и их предметов		Таблица lectors; Таблица subjects	Вывести из таблицы lectors: first_name, last_name, father_name, email, а также из таблицы subjects: subject_name
<b>Лекторы</b>				

1	Просмотр личной информации	lector_id;	Таблица lectors;	Вывести соответствующую запись из таблицы lectors.
2	Изменение личной информации	lector_id; first_name; last_name; father_name; email; mobile_phone; passport; dob; salary; login; password;	Таблица lectors;	Изменить соответствующую запись в таблице lectors;
3	Просмотр расписания на неделю по группе	group_id;	Таблица timetables;	Вывести соответствующие записи из таблицы timetables, а также из lectors: first_name, last_name, father_name, из groups: group_number
4	Просмотр всех студентов		Таблица students; Таблица logins;	Вывести все данные из таблицы students, кроме login_id, из таблицы groups: group_number
5	Просмотр всех лекторов		Таблица lectors; Таблица subjects;	Вывести все данные из таблицы lectors, кроме login_id, из subjects: subject_name
6	Просмотр всех лекторов и их предметов		Таблица subjects; Таблица lectors;	Вывести из таблицы lectors: first_name, last_name, father_name, email, из subjects:subject_name
7	Просмотр всех групп		Таблица groups;	Вывести все данные из таблицы groups
8	Просмотр студентов	group_id;	Таблица students;	Вывести все соответствующие записи из таблицы

	конкретной группы			students, кроме login_id;
9	Изменение информации об конкретной оценке у студента	assessment_id: student_id; date_passed; assessment; lector_group_subject_id;	Таблица assessments;	Изменить соответствующую запись в таблице assessments
10	Редактирование расписания	timetable_id: day_week; date_par; lector_group_subject_id; format_par; cabinet; housing;	Таблица timetables;	Изменить соответствующую запись в таблице timetables
11	Просмотр оценок конкретного студента	student_id;	Таблица assessmnets;	Вывести все соответствующие записи из таблицы assessmnets;
12	Просмотр личной информации студента	student_id;	Таблица students;	Вывести соответствующую запись из таблицы students;
13	Добавить описание к предмету	subject_id: subject_description;	Таблица subjects;	Изменить соответствующую запись в таблице subjects;
<b>Студенты</b>				
1	Просмотр личных данных	student_id;	Таблица students; Таблица logins;	Внести соответствующую запись из таблицы students, logins
2	Просмотр всех предметов		Таблица subjects;	Вывести все данные из таблицы subjects
3	Просмотр всех лекторов		Таблица lectors;	Вывести данные из таблицы lectors, кроме login_id
4	Просмотр всех групп		Таблица groups;	Вывести все данные из таблицы groups
5	Просмотр всех лекторов и их предметов		Таблица subjects; Таблица lectors;	Вывести данные из таблицы lectors: first_name, last_name,



				father_name, email, а также из таблицы subjects: subject_name
6	Просмотр расписания на неделю	group_id;	Таблица timetables; Таблица lectors; Таблица groups; Таблица subjects;	Вывести все соответствующие записи таблицы timetables, а также из таблицы lectors: first_name, last_name, father_name, а также из таблицы groups вывести group_number, и из subjects:subject_name
7	Посмотреть какие предметы есть у группы	group_id	Таблица subjects; Таблица groups;	Вывести из таблицы groups: group_number, а также из таблицы subjects_subject_name
8	Изменение личных данных	student_id; first_name; last_name; father_name; email; passport; dob; education_form; group_id; login; password;		
9	Просмотр своих одноклассников	group_id;	Таблица students;	Вывести все соответствующие записи из таблицы students, кроме login_id
10	Просмотр своих оценок	student_id;	Таблица assessmnets; Таблица lectors; Таблица students; Таблица subjects;	Вывести поля из таблицы students: first_name, last_name, father_name. Из таблицы subjects: subject_name, из таблицы assessmnets:

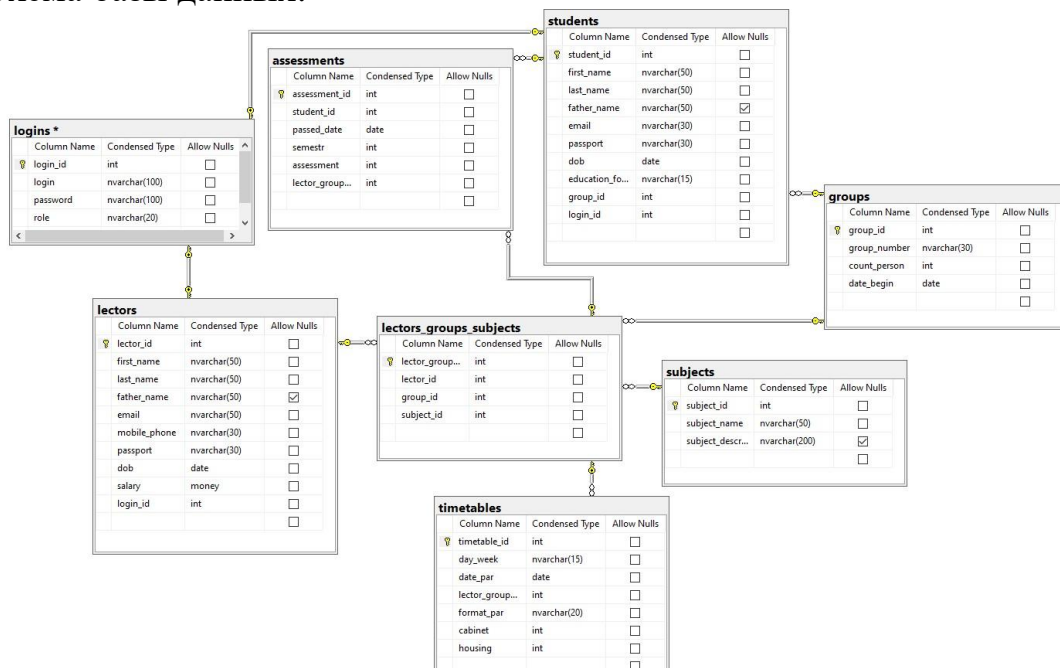
				assessment, date_passed, semestr. Из таблицы lectors: first_name, last_name, father_name
--	--	--	--	--

### Хранимые данные

№ пп	Хранимые данные	Пользователи, которым разрешен доступ	Ограничения по типу и значению
1	logins;	Администратор, лектор, студент.	login UNIQUE
2	subjects;	Администратор, лектор, студент (просмотр)	subject_name UNIQUE
3	students;	Администратор, лектор (просмотр), студент	email UNIQUE; passport UNIQUE; login_id UNIQUE;
4	lectors;	Администратор, лектор, студент (просмотр)	email UNIQUE; mobile_phone UNIQUE; passport UNIQUE; login_id UNIQUE;
5	assessments;	Администратор, лектор, студент (просмотр)	Assessmnets INT NOT NULL CHECK(assessment >= 2 AND assessment <= 5)
6	groups;	Администратор, лектор, студент (просмотр)	group_number UNIQUE
7	timetables;	Администратор, лектор, студент (просмотр)	Cabinet INT NOT NULL, Housing INT NOT NULL, CHECK(cabinet >=1 AND cabinet <= 100), CHECK(housing 1 >= 1 AND housing <=10)
8	lectors_group_subjects	Администратор, лектор (просмотр), студент (просмотр)	

# Описание таблиц и связей, схема

Схема базы данных:



## Перечень таблиц:

- logins – содержит данные аутентификации.
- lectors – содержит информацию о лекторах.
- students – содержит информацию о студентах.
- subjects – содержит информацию о предметах.
- lectors\_groups\_subjects – соединяет группы с лекторами и предметами. У одной группы может быть несколько лекторов. У одного лектора может быть несколько групп. А также один лектор может вести несколько предметов.
- timetables – содержит информацию о расписании.
- assessments – содержит информацию об оценках студента.
- groups – содержит информацию о группах.

## Связи между таблицами:

- lectors – logins: login\_id – login\_id (Один к одному).
- students – logins: login\_id – login\_id (Один к одному).
- groups – students: group\_id – group\_id (Один ко многим).
- students – assessments: student\_id – student\_id (Один ко многим).
- lectors – lectors\_groups\_subjects: lector\_id – lector\_id (Один ко многим).
- subjects – lectors\_groups\_subjects: subject\_id – subject\_id (Один ко многим).
- lectors\_groups\_subjects – timetables: lector\_group\_subject\_id – lector\_group\_subjects\_id (Одним ко многим).

- groups – lectors\_groups\_subjects: group\_id – group\_id(Один ко многим).
- lectors\_groups\_subjects – assessments:  
lector\_group\_subject\_id -lector\_group\_subject\_id

### Таблицы:

#### Assessments

	Column Name	Data Type	Allow Nulls
🔑	assessment_id	int	<input type="checkbox"/>
	student_id	int	<input type="checkbox"/>
	passed_date	date	<input type="checkbox"/>
	semestr	int	<input type="checkbox"/>
	assessment	int	<input type="checkbox"/>
	lector_group_subject_id	int	<input type="checkbox"/>
			<input type="checkbox"/>

- **assessment\_id** – NOT NULL, PRIMARY KEY, тип INT. Идентификатороценки.
- **student\_id** – NOT NULL, тип INT. Идентификатор студента, которыйполучил оценку. По этому полю таблица связана с таблицей students.
- **passed\_date** – NOT NULL, тип DATE. Дата получения оценки.
- **semestr** – NOT NULL, тип INT. (**semestr >= 1 AND semester <= 8**).Семестр в котором была получена оценка.
- **assessments** – NOT NULL, тип INT. (**assessments >= 2 AND assessments<= 5**). Оценка.
- **lector\_group\_subject\_id** – NOT NULL, тип INT. Идентификатор записив таблице lectors\_groups\_subjects. Будем знать по какому предмету получена оценка, т.к лектор может вести разные предметы. По этому полю таблица связана с таблицей **lectors\_groups\_subjects**.


#### Groups

	Column Name	Data Type	Allow Nulls
🔑	group_id	int	<input type="checkbox"/>
	group_number	nvarchar(30)	<input type="checkbox"/>
	count_person	int	<input type="checkbox"/>
	date_begin	date	<input type="checkbox"/>
			<input type="checkbox"/>

- **group\_id** – NOT NULL, PRIMARY KEY, тип INT. Идентификаторгруппы. По этому полю таблица связана с таблицей **students**.

- **group\_number** – NOT NULL, тип NVARCHAR(30). **UNIQUE**. Номер группы.
- **count\_person** – NOT NULL, тип INT. Количество человек в группе.
- **date\_begin** – NOT NULL, тип Date. Дата начала обучения.  
Студенты этой группы все вместе начали учиться в одно время.

## Lectors

Column Name	Data Type	Allow Nulls
 <b>lector_id</b>	int	<input type="checkbox"/>
first_name	nvarchar(50)	<input type="checkbox"/>
last_name	nvarchar(50)	<input type="checkbox"/>
father_name	nvarchar(50)	<input checked="" type="checkbox"/>
email	nvarchar(50)	<input type="checkbox"/>
mobile_phone	nvarchar(30)	<input type="checkbox"/>
passport	nvarchar(30)	<input type="checkbox"/>
dob	date	<input type="checkbox"/>
salary	money	<input type="checkbox"/>
login_id	int	<input type="checkbox"/>
		<input type="checkbox"/>

- **lector\_id** – NOT NULL, PRIMARY KEY, тип INT. Идентификатор лектора. По этому полю таблица связана с таблицей **lectors\_groups\_subjects**.
- **first\_name** – NOT NULL, тип NVARCHAR(50). Имя лектора.
- **last\_name** – NOT NULL, тип NVARCHAR(50). Фамилия лектора.
- **father\_name** - NULL, тип NVARCHAR(50). Отчество лектора.
- **email** - NOT NULL, тип NVARCHAR(50), **UNIQUE**. Электронная почта лектора.
- **mobile\_phone** – NOT NULL, тип NVARCHAR(30), **UNIQUE**. Мобильный телефон лектора.
- **passport** – NOT NULL, тип NVARCHAR(30), **UNIQUE**. Паспортные данные лектора.
- **dob** – NOT NULL, тип DATE. Дата рождения лектора.
- **salary** – NOT NULL, ТИП MONEY. Зарплата лектора.
- **login\_id** – NOT NULL, тип INT. **UNIQUE**. По этому полю таблица связана с таблицей **logins**. Данные аутентификации лектора.

## Lectors\_groups\_subjects

	Column Name	Data Type	Allow Nulls
PK	lector_group_subject_id	int	<input type="checkbox"/>
	lector_id	int	<input type="checkbox"/>
	group_id	int	<input type="checkbox"/>
	subject_id	int	<input type="checkbox"/>
			<input type="checkbox"/>

- **lector\_group\_subject\_id** – NOT NULL, PRIMARY KEY, тип INT. Идентификатор лектора, группы, предмета. У одной группы может быть несколько лекторов. Лектор может читать лекции нескольким группам. Один лектор может преподавать несколько предметов. По этому полю таблица связана с таблицей **timetable**, **assessments**.
- **lector\_id** – NOT NULL, тип INT. Идентификатор лектора. По этому полю таблица связана с таблицей **lectors**.
- **group\_id** – NOT NULL, тип INT. Идентификатор группы. По этому полю таблица связана с таблицей **groups**.
- **subject\_id** – NOT NULL, Тип INT. Идентификатор предмета. По этому полю таблица связана с таблицей **subjects**.

## logins

	Column Name	Data Type	Allow Nulls
PK	login_id	int	<input type="checkbox"/>
	login	nvarchar(100)	<input type="checkbox"/>
	password	nvarchar(100)	<input type="checkbox"/>
	role	nvarchar(20)	<input type="checkbox"/>
			<input type="checkbox"/>

- **login\_id** – NOT NULL, PRIMARY KEY, тип INT. Идентификатор данных для аутентификации. По этому полю таблица с таблицей **students**, **lectors**.
- **login** – NOT NULL, NVARCHAR(100), **UNIQUE**. Логин.
- **password** – NOT NULL, NVARCHAR(100). Пароль.
- **Role** – NOT NULL, NVARCHAR(20). Роль.

## students

	Column Name	Data Type	Allow Nulls
🔑	student_id	int	<input type="checkbox"/>
	first_name	nvarchar(50)	<input type="checkbox"/>
	last_name	nvarchar(50)	<input type="checkbox"/>
	father_name	nvarchar(50)	<input checked="" type="checkbox"/>
	email	nvarchar(30)	<input type="checkbox"/>
	passport	nvarchar(30)	<input type="checkbox"/>
	dob	date	<input type="checkbox"/>
	education_form	nvarchar(15)	<input type="checkbox"/>
	group_id	int	<input type="checkbox"/>
	login_id	int	<input type="checkbox"/>
			<input type="checkbox"/>


- **student\_id** – NOT NULL, PRIMARY KEY, тип INT. Идентификатор студента. По этому полю таблица связана с таблицей **assessments**.
- **first\_name** – NOT NULL, тип NVARCHAR(50). Имя студента.
- **last\_name** – NOT NULL, тип NVARCHAR(50). Фамилия студента.
- **father\_name** – NULL, тип NVARCHAR(50). Отчество студента.
- **email** – NOT NULL, тип NVARCHAR(30), **UNIQUE**. Электронная почта студента.
- **passport** – NOT NULL, тип NVARCHAR(30), **UNIQUE**. Паспортные данные студента.
- **dob** – NOT NULL, тип DATE. Дата рождения студента.
- **education\_form** – NOT NULL, тип NVARCHAR(15). Форма обучения студента.
- **group\_id** – NOT NULL, тип INT. Идентификатор группы студента. По этому полю таблица связана с таблицей **groups**.
- **login\_id** – NOT NULL, тип INT **UNIQUE**. Данные аутентификации студента. По этому полю таблица связана с таблицей **logins**.

## subjects

	Column Name	Data Type	Allow Nulls
🔑	subject_id	int	<input type="checkbox"/>
	subject_name	nvarchar(50)	<input type="checkbox"/>
	subject_description	nvarchar(200)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- **subject\_id** – NOT NULL, PRIMARY KEY, тип INT. По этому полю таблица связана с таблицей **lectors\_groups\_subjects\_id**.
- **subject\_name** – NOT NULL, тип NVARCHAR(50) **UNIQUE**. Название предмета.

## Timetables

	Column Name	Data Type	Allow Nulls
	timetable_id	int	<input type="checkbox"/>
	day_week	nvarchar(15)	<input type="checkbox"/>
	date_par	date	<input type="checkbox"/>
	lector_group_subject_id	int	<input type="checkbox"/>
	format_par	nvarchar(20)	<input type="checkbox"/>
	cabinet	int	<input type="checkbox"/>
	housing	int	<input type="checkbox"/>
			<input type="checkbox"/>

- **timetables\_id** – NOT NULL, PRIMARY KEY, тип INT. Идентификатор расписания.
- **day\_week** – NOT NULL, тип NVARCHAR(15). День недели.
- **date\_par** – NOT NULL, тип DATE. Дата.
- **lector\_group\_subject\_id** – NOT NULL, тип INT. Конкретный лектор ведет у конкретной группы конкретный предмет. По этому полю таблица связана с таблицей **lectors\_groups\_subjects**.
- **format\_par** – NOT NULL, тип NVARCHAR(20). Формат проведения пары.
- **cabinet** – NOT NULL, тип INT. Кабинет.
- **housing** – NOT NULL, тип INT. Корпус.



## Запросы

```
/*Добавить новый предмет в таблицу subjects.*/
INSERT INTO subjects(subject_name)
VALUES
(''),
('')
/*Добавить новую группу таблицы groups.*/
INSERT INTO groups(group_number, count_person, date_begin)
VALUES
('', 2, '2020-09-01')
/*Добавить новую оценку студенту по предмету в таблицу assessments*/
INSERT INTO assessments(student_id, passed_date, semestr, assessment,
lector_group_subject_id)
VALUES
(27, '2020-10-02', 1, 5, 1011)
/*Добавить новую запись в таблицу timetables.*/
INSERT INTO timetables(day_week, date_par, lector_group_subject_id, format_par, cabinet,
housing)
VALUES
('Суббота', '2022-11-17', 1008, 'Очно', 1, 1)
/*Добавить лектору новую группу с предметом lectors_groups_subjects.*/
INSERT INTO lectors_groups_subjects(lector_id, group_id, subject_id)
VALUES
(1,1,1)
/*Вывести всех студентов и их группу. Поля для вывода (имя студента, фамилия студента,
отчество студента, номер группы студента).*/
SELECT students.first_name AS 'Имя', students.last_name AS 'Фамилия',
students.father_name AS 'Отчество', groups.group_number AS 'Номер группы' FROM students
JOIN groups
ON students.group_id = groups.group_id
ORDER BY group_number
```

	Имя	Фамилия	Отчество	Номер группы
1	Алексей	Петров	Олегович	353/01
2	Олег	Сидоров	Александрович	353/01
3	Антон	Павлов	Дмитриевич	353/01
4	Иван	Петров	Дмитриевич	353/02

```
/*Вывести всех лекторов и их предметы. Поля для вывода (имя лектора, фамилия лектора,
отчество лектора, название предмета)*/
SELECT DISTINCT lectors.first_name AS 'Имя', lectors.last_name AS 'Фамилия',
lectors.father_name AS 'Отчество', subjects.subject_name AS 'Название предмета' FROM
lectors JOIN lectors_groups_subjects
ON lectors.lector_id = lectors_groups_subjects.lector_id JOIN subjects
ON lectors_groups_subjects.subject_id = subjects.subject_id
```

	Имя	Фамилия	Отчество	Название предмета
1	Алексей	Иванов	Сергеевич	С#
2	Алексей	Иванов	Сергеевич	Kotlin
3	Алексей	Иванов	Сергеевич	Вычислительная математика
4	Алексей	Иванов	Сергеевич	Вышмат
5	Василий	Пустов	Константинович	С++
6	Виталий	Смирнов	Олегович	С++
7	Виталий	Смирнов	Олегович	Архитектура ЭВМ
8	Константин	Марков	Евгеньевич	Операционные системы

```
/*Вывести всех лекторов и номера групп, у которых преподают. Поля для вывода (имя
лектора, фамилия лектора, отчество лектора, номер группы).*/
SELECT DISTINCT lectors.first_name AS 'Имя', lectors.last_name AS 'Фамилия',
lectors.father_name AS 'Отчество', groups.group_number AS 'Номер группы' FROM lectors
JOIN lectors_groups_subjects
ON lectors.lector_id = lectors_groups_subjects.lector_id JOIN groups
```

```
ON lectors_groups_subjects.group_id = groups.group_id JOIN subjects ON
lectors_groups_subjects.subject_id = subjects.subject_id
ORDER BY group_number
```

	Имя	Фамилия	Отчество	Номер группы
1	Алексей	Иванов	Сергеевич	353/01
2	Виталий	Смирнов	Олегович	353/01
3	Константин	Марков	Евгеньевич	353/01
4	Алексей	Иванов	Сергеевич	353/02
5	Василий	Пустов	Константинович	353/02
6	Виталий	Смирнов	Олегович	353/02

```
/*Вывести всех студентов, у которых нет оценок. Поля для вывода (имя студента, фамилия студента, отчество студента, форма обучения, email, номер группы).*/
```

```
SELECT students.first_name AS 'Имя', students.last_name AS 'Фамилия',
students.father_name AS 'Отчество', students.education_form AS 'Форма обучения',
students.email AS 'Email', groups.group_number AS 'Номер группы' FROM students JOIN
groups
```

```
ON students.group_id = groups.group_id
```

```
WHERE NOT EXISTS(SELECT * FROM assessments WHERE assessments.student_id =
students.student_id)
```

	Имя	Фамилия	Отчество	Форма обучения	Email	Номер группы
1	Олег	Сидоров	Александрович	Контракт	olegSid@mail.ru	353/01

```
/*Вывести всех студентов, которые получили конкретную оценку . Поля для вывода (имя студента, фамилия студента, отчество студента, номер группы, название предмета).*/
```

```
SELECT DISTINCT students.first_name AS 'Имя', students.last_name AS 'Фамилия',
students.father_name AS 'Отчество', groups.group_number AS 'Номер группы',
subjects.subject_name AS 'Название предмета' FROM students JOIN groups
```

```
ON students.group_id = groups.group_id JOIN assessments
```

```
ON students.student_id = assessments.student_id JOIN lectors_groups_subjects
```

```
ON lectors_groups_subjects.lector_group_subject_id = assessments.lector_group_subject_id
JOIN subjects
```

```
ON lectors_groups_subjects.subject_id = subjects.subject_id
```

```
WHERE assessments.assessment = 4
```

	Имя	Фамилия	Отчество	Номер группы	Название предмета
1	Алексей	Петров	Олегович	353/01	C#
2	Алексей	Петров	Олегович	353/01	Kotlin
3	Иван	Петров	Дмитриевич	353/02	Вычислительная математика
4	Иван	Петров	Дмитриевич	353/02	Вышмат

```
/*Вывести все предметы, которые есть у группы. Поля для вывода (название предмета, описание предмета, номер группы).*/
```

```
SELECT DISTINCT subjects.subject_name AS 'Название предмета', subject_description AS
'Описание предмета', groups.group_number AS 'Номер группы' FROM subjects JOIN
lectors_groups_subjects
```

```
ON subjects.subject_id = lectors_groups_subjects.subject_id JOIN groups
```

```
ON lectors_groups_subjects.group_id = groups.group_id
```

```
ORDER BY groups.group_number
```

	Название предмета	Описание предмета	Номер группы
1	C#	Нужен Visual Studio	353/01
2	Kotlin		353/01
3	Архитектура ЭВМ		353/01
4	Вычислительная математика		353/01
5	Операционные системы		353/01
6	C++		353/02
7	Вычислительная математика		353/02
8	Вышмат		353/02

```
/*Вывести всех лекторов.*/
```

```
SELECT * FROM lectors
```

	lector_id	first_name	last_name	father_name	email	mobile_phone	passport	dob	salary	login_id
1	1115	Владимир	Калашников	Васильевич	alexKalash@mail.ru	+791821363	9847 837432	1974-02-16	45000,50	1143
2	1116	Виталий	Смирнов	Олегович	vitalSmr@mail.ru	+791237983	1235 343098	1974-06-08	45843,00	1144
3	1117	Василий	Пустов	Константинович	vasPust@mail.ru	+791230910	3281 982123	1965-09-09	65329,00	1145
4	1118	Константин	Марков	Евгеньевич	kostyamar@mail.ru	+791987910	1239 009321	1983-04-01	45454,00	1146
5	1119	Владимир	Кашин	Петрович	vladimirKash@mail.ru	+791987632	9826 234723	1985-04-03	45000,00	1147
6	1120	Иван	Сергеев	Петрович	ivanPetr@mail.ru	+79182373	3244 324783	1974-05-11	54333,00	1153
7	2120	Алексей	Иванов	Сергеевич	alexeyIvanov@mail.ru	+912832631	2343 343422	1975-11-30	32432,00	2151
8	2121	Сергей	Котов	Евгеньевич	sergKotov@mail.ru	+71982732	3422 273467	1964-02-05	44376,00	2152
9	3120	Екатерина	Алексеева	Александровна	ekatr@mail.ru	+71238947	2343 232432	1991-12-02	-50000,00	3151
10	3121	Екатерина	Алексеева	Александровна	ekat1r@mail.ru	+712381947	2243 232432	1991-12-02	-50000,00	3152
11	3122	Алексей	Иванов	Алексеевич	ale1x@mail.ru	+79381264	3423 324243	2022-12-02	5555,00	3153
12	3123	Олег	Олегович	Олегов	ole1g@mail.ru	+70981222	3242 23424	2022-12-02	-1000,00	3154

/\*Вывести все предметы.\*/

```
SELECT * FROM subjects
```

	subject_id	subject_name	subject_description
1	1018	С#	Нужен Visual Studio
2	1020	Базы данных	Нужен MS SQL Server
3	1021	Вычислительная математика	
4	1023	Вышмат	
5	1026	Архитектура ЭВМ	
6	1029	Операционные системы	
7	1039	Kotlin	
8	1040	C++	

/\*Вывести все группы\*/

```
SELECT * FROM groups
```

	group_id	group_number	count_person	date_begin
1	1014	353/01	30	2020-09-01
2	1015	353/02	15	2020-09-01

/\*Вывести данные студента по id.\*/

```
SELECT * FROM students
WHERE students.student_id = 1028
```

	student_id	first_name	last_name	father_name	email	passport	dob	education_fom	group_id	login_id
1	1028	Алексей	Петров	Олегович	alexeyOleg@mail.ru	4354 327723	1983-11-12	Контракт	1014	1151

/\*Вывести расписания по id.\*/

```
SELECT * FROM timetables
WHERE timetables.timetable_id=4
```

	timetable_id	day_week	date_par	lector_group_subject_id	format_par	cabinet	housing
1	4	среда	2022-11-30	2097	Очно	11	3

/\*Вывести все оценки.\*/

```
SELECT * FROM assessments
```

	assessment_id	student_id	passed_date	semestr	assessment	lector_group_subject_id
1	3092	2028	2022-12-01	4	3	2091
2	3093	1028	2022-12-05	3	3	2092
3	3094	2028	2022-12-05	3	3	2092
4	3095	1028	2022-11-25	3	3	2091
5	3096	1028	2022-11-04	2	4	2091
6	3097	1028	2022-10-09	1	4	2095
7	3099	1028	2022-11-23	2	4	2091
8	3100	1028	2022-12-02	2	3	2095
9	3101	1028	2022-09-01	1	2	2091
10	3102	1028	2022-12-02	2	2	2091
11	3103	3028	2022-09-01	1	4	2093
12	3104	3028	2022-09-01	1	4	2094
13	3105	3028	2022-09-01	1	3	2101
14	3106	3028	2022-09-01	1	2	2101
15	3107	3028	2022-12-04	1	5	2093

/\*Вывод всех студентов конкретной группы по id группы.\*/

```
SELECT * FROM students
WHERE students.group_id = 1014
```

	student_id	first_name	last_name	father_name	email	passport	dob	education_form	group_id	login_id
1	1028	Алексей	Петров	Олегович	alexeyOleg@mail.ru	4354 327723	1983-11-12	Контракт	1014	1151
2	1029	Олег	Сидоров	Александрович	olegSid@mail.ru	4534 230978	1975-04-01	Контракт	1014	1152
3	2028	Антон	Павлов	Дмитриевич	antonPav@mail.ru	3476 347201	2002-03-06	Контракт	1014	2153

/\*Вывести оценки по id студента.\*/

```
SELECT * FROM assessments
WHERE assessments.student_id = 1028;
```

	assessment_id	student_id	passed_date	semestr	assessment	lector_group_subject_id
1	3093	1028	2022-12-05	3	3	2092
2	3095	1028	2022-11-25	3	3	2091
3	3096	1028	2022-11-04	2	4	2091
4	3097	1028	2022-10-09	1	4	2095
5	3099	1028	2022-11-23	2	4	2091
6	3100	1028	2022-12-02	2	3	2095
7	3101	1028	2022-09-01	1	2	2091
8	3102	1028	2022-12-02	2	2	2091

/\*Изменить название или описание предмета по id.\*/

GO

```
CREATE PROCEDURE update_subject @subject_name AS NVARCHAR(50), @subject_description AS
NVARCHAR(200), @oldSubject_name AS NVARCHAR(50)
```

AS

BEGIN

```
UPDATE subjects
```

```
SET
```

```
subject_name = @subject_name,
```

```
subject_description = @subject_description
```

```
WHERE subject_id = (SELECT subject_id FROM subjects WHERE subject_name =
@oldSubject_name)
```

END

/\*Изменение данных студента\*/

GO

```
CREATE PROCEDURE update_student @first_name AS NVARCHAR(50), @last_name AS NVARCHAR(50),
@father_name AS NVARCHAR(50), @email AS NVARCHAR(50), @passport AS NVARCHAR(50), @dob AS
DATE, @education_form AS NVARCHAR(50), @group_number AS NVARCHAR(30), @login AS
NVARCHAR(100), @password AS VARCHAR(100), @helpMail AS NVARCHAR(50)
```

AS

BEGIN

```
UPDATE logins
```

```
SET
```

```

login = @login,
password = @password,
role = 'student'
WHERE login_id = (SELECT login_id FROM students WHERE email=@helpMail)
UPDATE students
SET
first_name = @first_name,
last_name = @last_name,
father_name = @father_name,
email = @email,
passport = @passport,
dob = @dob,
education_form = @education_form,
group_id = (SELECT group_id FROM groups WHERE group_number = @group_number)
WHERE student_id = (SELECT student_id FROM students WHERE email = @helpMail)
END
/*Обновление данных лектора*/
GO
CREATE PROCEDURE update_lector @first_name AS NVARCHAR(50), @last_name AS NVARCHAR(50),
@father_name AS NVARCHAR(50), @email NVARCHAR(50), @mobile_phone AS NVARCHAR(50),
@passport AS NVARCHAR(50), @dob AS DATE, @salary AS MONEY, @login AS NVARCHAR(100),
@password AS NVARCHAR(50)
AS
BEGIN
UPDATE logins
SET
login = @login,
password = @password
WHERE login_id=(SELECT login_id FROM lectors WHERE lectors.email=@helpMail)
UPDATE lectors
SET
first_name = @first_name,
last_name = @last_name,
father_name = @father_name,
email = @email,
mobile_phone = @mobile_phone,
passport = @passport,
dob = @dob,
salary = @salary
WHERE lector_id = (SELECT lector_id FROM lectors WHERE email = @helpMail)
END
/*Изменение информации об оценке по id.*/
UPDATE assessments
SET
student_id = 15,
passed_date = '2019-06-09',
semestr = 2,
assessment = 4,
lector_group_subject_id = 30
WHERE student_id = 1028
/*Изменить пару в расписании по id.*/
UPDATE timetables
SET
day_week = 'понедельник',
date_par='28-11-2022',
lector_group_subject_id = 2003,
format_par = 'Очно',
cabinet = 2,
housing = 1
WHERE timetable_id = 5;
/*Удалить строку по id из таблицы subjects;*/
DELETE FROM subjects
WHERE subject_id = 1016
/*Удалить запись в расписании*/

```

```

DELETE FROM timetables
WHERE timetable_id=2
/*Добавить студента*/
GO
CREATE PROCEDURE insert_student @first_name AS NVARCHAR(50), @last_name AS NVARCHAR(50),
@father_name AS NVARCHAR(50), @email AS NVARCHAR(50), @passport AS NVARCHAR(50), @dob AS
DATE, @education_form AS NVARCHAR(50), @group_number AS NVARCHAR(30), @login AS
NVARCHAR(100), @password AS NVARCHAR(100), @role AS NVARCHAR(30)
AS
BEGIN
INSERT INTO logins(login, password, role)
VALUES (@login, @password, @role)
INSERT INTO students(first_name, last_name, father_name, email, passport, dob,
education_form, group_id, login_id)
VALUES (@first_name, @last_name, @father_name, @email, @passport, @dob, @education_form,
(SELECT groups.group_id FROM groups WHERE group_number=@group_number), (SELECT
MAX(logins.login_id) FROM logins))
END
/*Регистрация лектора*/
GO
CREATE PROCEDURE insert_lector @first_name AS NVARCHAR(50), @last_name AS NVARCHAR(50),
@father_name AS NVARCHAR(50), @email AS NVARCHAR(50), @mobile_phone AS NVARCHAR(300) ,
@passport AS NVARCHAR(30), @dob AS DATE, @salary AS MONEY, @login AS NVARCHAR(100),
@password AS NVARCHAR(100), @role AS NVARCHAR(100)
AS
BEGIN
INSERT INTO logins(login, password, role)
VALUES (@login, @password, @role)
INSERT INTO lectors(first_name, last_name, father_name, email, mobile_phone, passport,
dob, salary, login_id)
VALUES (@first_name, @last_name, @father_name, @email, @mobile_phone, @passport, @dob,
@salary, (SELECT MAX(logins.login_id) FROM logins))
END
/*Вывести оценки конкретного студента по конкретному предмету. Поля
для вывода (имя студента, фамилия студента, отчество студента,
название предмета, оценка, дата получения оценки).*/
GO
CREATE PROCEDURE show_student_assessments_specific_subject @first_name AS NVARCHAR(50),
@last_name AS NVARCHAR(50), @father_name AS NVARCHAR(50), @subject_name AS NVARCHAR(50)
AS
BEGIN
SELECT DISTINCT students.first_name AS 'Имя', students.last_name AS 'Фамилия',
students.father_name AS 'Отчество', subjects.subject_name AS 'Название предмета',
assessments.assessment AS 'Оценка', assessments.passed_date AS 'Дата получения оценки'
FROM students INNER JOIN assessments
ON students.student_id = assessments.student_id INNER JOIN lectors_groups_subjects
ON assessments.lector_group_subject_id = lectors_groups_subjects.lector_group_subject_id
INNER JOIN lectors
ON lectors_groups_subjects.lector_id = lectors.lector_id INNER JOIN subjects
ON lectors_groups_subjects.subject_id = subjects.subject_id
WHERE students.first_name = @first_name AND students.last_name = @last_name AND
students.father_name = @father_name AND subjects.subject_name = @subject_name
END
EXEC show_student_assessments_specific_subject 'Алексей', 'Петров', 'Олегович', 'С#'

```

	Имя	Фамилия	Отчество	Название предмета	Оценка	Дата получения оценки
1	Алексей	Петров	Олегович	С#	2	2022-09-01
2	Алексей	Петров	Олегович	С#	2	2022-12-02
3	Алексей	Петров	Олегович	С#	3	2022-11-25
4	Алексей	Петров	Олегович	С#	4	2022-11-04
5	Алексей	Петров	Олегович	С#	4	2022-11-23

/\*Вывести все оценки у конкретного студента. Поля для вывода (имя студента, фамилия студента, отчество студента, оценка, дата получения оценки, название предмета, имя лектора, фамилия лектора, отчество



```

лектора).*/
GO
CREATE PROCEDURE show_all_assessments_specific_students @email AS NVARCHAR(30) AS
BEGIN
SELECT students.first_name AS 'Имя студента', students.last_name AS 'Фамилия студента',
students.father_name AS 'Отчество студента', assessments.assessment AS 'Оценка',
assessments.passed_date AS 'Дата получения оценки', subjects.subject_name AS 'Название
предмета', lectors.first_name AS 'Имя лектора', lectors.last_name AS 'Фамилия лектора',
lectors.father_name AS 'Отчество лектора' FROM students JOIN groups
ON students.group_id = groups.group_id JOIN assessments
ON students.student_id = assessments.student_id JOIN lectors_groups_subjects
ON assessments.lector_group_subject_id = lectors_groups_subjects.lector_group_subject_id
JOIN lectors
ON lectors_groups_subjects.lector_id = lectors.lector_id JOIN subjects
ON lectors_groups_subjects.subject_id = subjects.subject_id
WHERE students.email = @email
END
EXEC show_all_assessments_specific_students 'alexeyOleg@mail.ru'

```

	Имя студента	Фамилия студента	Отчество студента	Оценка	Дата получения оценки	Название предмета	Имя лектора	Фамилия лектора	Отчество лектора
1	Алексей	Петров	Олегович	3	2022-12-05	Вычислительная математика	Алексей	Иванов	Сергеевич
2	Алексей	Петров	Олегович	3	2022-11-25	С#	Алексей	Иванов	Сергеевич
3	Алексей	Петров	Олегович	4	2022-11-04	С#	Алексей	Иванов	Сергеевич
4	Алексей	Петров	Олегович	4	2022-10-09	Kotlin	Алексей	Иванов	Сергеевич
5	Алексей	Петров	Олегович	4	2022-11-23	С#	Алексей	Иванов	Сергеевич
6	Алексей	Петров	Олегович	3	2022-12-02	Kotlin	Алексей	Иванов	Сергеевич
7	Алексей	Петров	Олегович	2	2022-09-01	С#	Алексей	Иванов	Сергеевич
8	Алексей	Петров	Олегович	2	2022-12-02	С#	Алексей	Иванов	Сергеевич

```

/*Вывести расписание по группе на конкретную неделю*/
GO
CREATE PROCEDURE timetable_on_weekv2 @group_number AS NVARCHAR(30), @date AS DATE
AS
BEGIN
(SELECT DATENAME(WEEKDAY, date_par) AS 'День недели', timetables.date_par AS 'Дата пары',
lectors.first_name AS 'Имя', lectors.last_name AS 'Фамилия', lectors.father_name AS
'Отчество', subjects.subject_name AS 'Предмет', timetables.format_par AS 'Формат',
timetables.cabinet AS 'Кабинет', timetables.housing AS 'Корпус' FROM timetables JOIN
lectors_groups_subjects ON
timetables.lector_group_subject_id = lectors_groups_subjects.lector_group_subject_id JOIN
lectors
ON lectors_groups_subjects.lector_id = lectors.lector_id JOIN groups
ON lectors_groups_subjects.group_id = groups.group_id JOIN subjects
ON lectors_groups_subjects.subject_id = subjects.subject_id
WHERE group_number = @group_number AND day_week = 'Понедельник' AND DATENAME(WEEK, @date)
= DATENAME(WEEK, date_par)
UNION
SELECT DATENAME(WEEKDAY, date_par) AS 'День недели', timetables.date_par AS 'Дата пары',
lectors.first_name AS 'Имя', lectors.last_name AS 'Фамилия', lectors.father_name AS
'Отчество', subjects.subject_name AS 'Предмет', timetables.format_par AS 'Формат',
timetables.cabinet AS 'Кабинет', timetables.housing AS 'Корпус' FROM timetables JOIN
lectors_groups_subjects
ON timetables.lector_group_subject_id = lectors_groups_subjects.lector_group_subject_id
JOIN lectors
ON lectors_groups_subjects.lector_id = lectors.lector_id JOIN groups
ON lectors_groups_subjects.group_id = groups.group_id JOIN subjects
ON lectors_groups_subjects.subject_id = subjects.subject_id
WHERE group_number = @group_number AND day_week = 'Вторник' AND DATENAME(WEEK, @date) =
DATENAME(WEEK, date_par)
UNION
SELECT DATENAME(WEEKDAY, date_par) AS 'День недели', timetables.date_par AS 'Дата пары',
lectors.first_name AS 'Имя', lectors.last_name AS 'Фамилия', lectors.father_name AS
'Отчество', subjects.subject_name AS 'Предмет', timetables.format_par AS 'Формат',
timetables.cabinet AS 'Кабинет', timetables.housing AS 'Корпус' FROM timetables JOIN
lectors_groups_subjects
ON timetables.lector_group_subject_id = lectors_groups_subjects.lector_group_subject_id
JOIN lectors
ON lectors_groups_subjects.lector_id = lectors.lector_id JOIN groups

```

```

ON lectors_groups_subjects.group_id = groups.group_id JOIN subjects
ON lectors_groups_subjects.subject_id = subjects.subject_id
WHERE group_number = @group_number AND day_week = 'Среда' AND DATEPART(WEEK, @date) =
DATEPART(WEEK, date_par)
UNION
SELECT DATENAME(WEEKDAY, date_par) AS 'День недели', timetables.date_par AS 'Дата пары',
lectors.first_name AS 'Имя', lectors.last_name AS 'Фамилия', lectors.father_name AS
'Отчество', subjects.subject_name AS 'Предмет', timetables.format_par AS 'Формат',
timetables.cabinet AS 'Кабинет', timetables.housing AS 'Корпус' FROM timetables JOIN
lectors_groups_subjects
ON timetables.lector_group_subject_id = lectors_groups_subjects.lector_group_subject_id
JOIN lectors
ON lectors_groups_subjects.lector_id = lectors.lector_id JOIN groups
ON lectors_groups_subjects.group_id = groups.group_id JOIN subjects
ON lectors_groups_subjects.subject_id = subjects.subject_id
WHERE group_number = @group_number AND day_week = 'Четверг' AND DATEPART(WEEK, @date) =
DATEPART(WEEK, date_par)
UNION
SELECT DATENAME(WEEKDAY, date_par) AS 'День недели', timetables.date_par AS 'Дата пары',
lectors.first_name AS 'Имя', lectors.last_name AS 'Фамилия', lectors.father_name AS
'Отчество', subjects.subject_name AS 'Предмет', timetables.format_par AS 'Формат',
timetables.cabinet AS 'Кабинет', timetables.housing AS 'Корпус' FROM timetables JOIN
lectors_groups_subjects
ON timetables.lector_group_subject_id = lectors_groups_subjects.lector_group_subject_id
JOIN lectors
ON lectors_groups_subjects.lector_id = lectors.lector_id JOIN groups
ON lectors_groups_subjects.group_id = groups.group_id JOIN subjects
ON lectors_groups_subjects.subject_id = subjects.subject_id
WHERE group_number = @group_number AND day_week = 'Пятница' AND DATEPART(WEEK, @date) =
DATEPART(WEEK, date_par)
UNION
SELECT DATENAME(WEEKDAY, date_par) AS 'День недели', timetables.date_par AS 'Дата пары',
lectors.first_name AS 'Имя', lectors.last_name AS 'Фамилия', lectors.father_name AS
'Отчество', subjects.subject_name AS 'Предмет', timetables.format_par AS 'Формат',
timetables.cabinet AS 'Кабинет', timetables.housing AS 'Корпус' FROM timetables JOIN
lectors_groups_subjects
ON timetables.lector_group_subject_id = lectors_groups_subjects.lector_group_subject_id
JOIN lectors
ON lectors_groups_subjects.lector_id = lectors.lector_id JOIN groups
ON lectors_groups_subjects.group_id = groups.group_id JOIN subjects
ON lectors_groups_subjects.subject_id = subjects.subject_id
WHERE group_number = @group_number AND day_week = 'Суббота' AND DATEPART(WEEK, @date) =
DATEPART(WEEK, date_par))
ORDER BY date_par
END
EXEC timetable_on_weekv2 '353/01', '01-12-2022'

```

	День недели	Дата пары	Имя	Фамилия	Отчество	Предмет	Формат	Кабинет	Корпус
1	понедельник	2022-11-28	Алексей	Иванов	Сергеевич	С#	Очно	2	2
2	вторник	2022-11-29	Алексей	Иванов	Сергеевич	Вычислительная математика	Очно	4	2
3	среда	2022-11-30	Алексей	Иванов	Сергеевич	Kotlin	Очно	23	1
4	среда	2022-11-30	Константин	Марков	Евгеньевич	Операционные системы	Очно	11	3
5	четверг	2022-12-01	Алексей	Иванов	Сергеевич	Kotlin	Очно	11	2
6	четверг	2022-12-01	Виталий	Смирнов	Олегович	Архитектура ЭВМ	Очно	33	2
7	пятница	2022-12-02	Алексей	Иванов	Сергеевич	С#	Очно	44	4
8	пятница	2022-12-02	Алексей	Иванов	Сергеевич	Вычислительная математика	Очно	1	2

/\*Вывести всех студентов (имя, фамилия, отчество, номер группы) и их данные аутентификации.\*/

```

CREATE VIEW all_students_logins AS
SELECT students.first_name AS 'Имя', students.last_name AS 'Фамилия',
students.father_name AS 'Отчество', groups.group_number AS 'Номер группы', logins.login
AS 'Логин', logins.password AS 'Пароль', logins.role AS 'Роль'
FROM students INNER JOIN logins
ON students.login_id = logins.login_id INNER JOIN groups
ON students.group_id = groups.group_id

```



```
SELECT * FROM all_students_logins
```

	Имя	Фамилия	Отчество	Номер группы	Логин	Пароль	Роль
1	Алексей	Петров	Олегович	353/01	alex1	IQTafyv/ZpY=	student
2	Олег	Сидоров	Александрович	353/01	olegSid1	IQTafyv/ZpY=	student
3	Антон	Павлов	Дмитриевич	353/01	anton	IQTafyv/ZpY=	student
4	Иван	Петров	Дмитриевич	353/02	ivannn	M884HSgG16Q=	student

```
/*Вывести всех лекторов (имя, фамилия, отчество) и их данные аутентификации.*/
```

```
CREATE VIEW all_lectors_logins AS
```

```
SELECT lectors.first_name AS 'Имя', lectors.last_name AS 'Фамилия', lectors.father_name AS 'Отчество', logins.login AS 'Логин', logins.password AS 'Пароль', logins.role AS 'Роль'
```

```
FROM lectors INNER JOIN logins
```

```
ON lectors.login_id = logins.login_id
```

```
SELECT * FROM all_lectors_logins
```

	Имя	Фамилия	Отчество	Логин	Пароль	Роль
1	Владимир	Калашников	Васильевич	alex	IQTafyv/ZpY=	lector
2	Виталий	Смирнов	Олегович	vitalSmr	6ZGmBXFzpZd87yR/mDaZA==	lector
3	Василий	Пустов	Константинович	vasPus	xM056cqhcw=	lector
4	Константин	Марков	Евгеньевич	kostyamark	mlx0OcdEtY4=	lector
5	Владимир	Кашин	Петрович	vladimirKash	mlx0OcdEtY4=	lector
6	Иван	Сергеев	Петрович	ivanPetr	WRsozn/SlojKyX1ZknN5hg==	lector
7	Алексей	Иванов	Сергеевич	alexeylv	IQTafyv/ZpY=	lector
8	Сергей	Котов	Евгеньевич	sergeyK	xM056cqhcw=	lector
9	Екатерина	Алексеева	Александровна	ekat	Htt0edJJle4=	lector
10	Екатерина	Алексеева	Александровна	ekat1	Htt0edJJle4=	lector

```
/*все данные аутентификации*/
```

```
CREATE VIEW all_logins AS
```

```
SELECT logins.login AS 'Логин', logins.password AS 'Пароль', logins.role AS 'Роль' FROM logins
```

```
WHERE role IN('student','lector')
```

```
SELECT * FROM all_logins
```

2	alex	IQTafyv/ZpY=	lector
3	vitalSmr	6ZGmBXFzpZd87yR/mDaZA==	lector
4	vasPus	xM056cqhcw=	lector
5	kostyamark	mlx0OcdEtY4=	lector
6	vladimirKash	mlx0OcdEtY4=	lector
7	alex1	IQTafyv/ZpY=	student
8	olegSid1	IQTafyv/ZpY=	student
9	ivanPetr	WRsozn/SlojKyX1ZknN5hg==	lector
10	alexeylv	IQTafyv/ZpY=	lector
11	sergeyK	xM056cqhcw=	lector
12	anton	IQTafyv/ZpY=	student
13	ekat	Htt0edJJle4=	lector

## Описание программы

Необходимо было реализовать систему для автоматизации работы кафедры университета. В системе должно быть 3 типа пользователей. Администратор, лектор, студент. Система должна хранить информацию о каждом лекторе, студенте. Хранит личную информацию каждого, оценки каждого студента, предметы лектора и расписание кафедры. Администратор может регистрировать лекторов и студентов, также может редактировать личную информацию каждого студента и лектора. Добавлять и редактировать оценки и расписание, просматривать различную информацию.

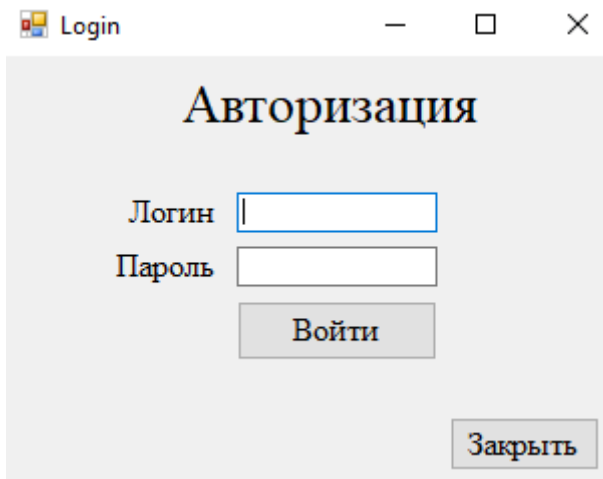
Лектор может добавлять, изменять оценки и расписание. Может менять информацию о себе, а также просматривать разную информацию, в том числе личную информацию о себе.

Студент может просматривать свои оценки, свои предметы и своих одногруппников, информацию о себе, а также имеет возможность ее редактировать. Может посмотреть свой средний балл по всем предметам.

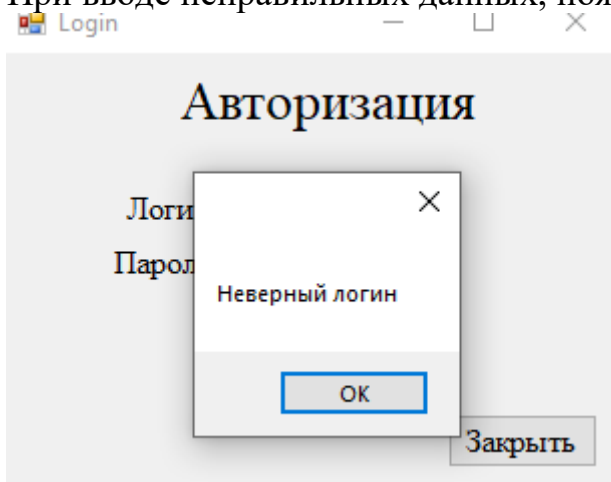
Для реализации программы я выбрал язык программирования C# и использовал СУБД MS SQL Server. В C# очень удобно взаимодействовать с формами, настроить каждую форму под себя. Также удобно работать с выпадающими списками. Среду разработки я выбрал Microsoft Visual Studio 2022 Community Version.

## Работа приложения

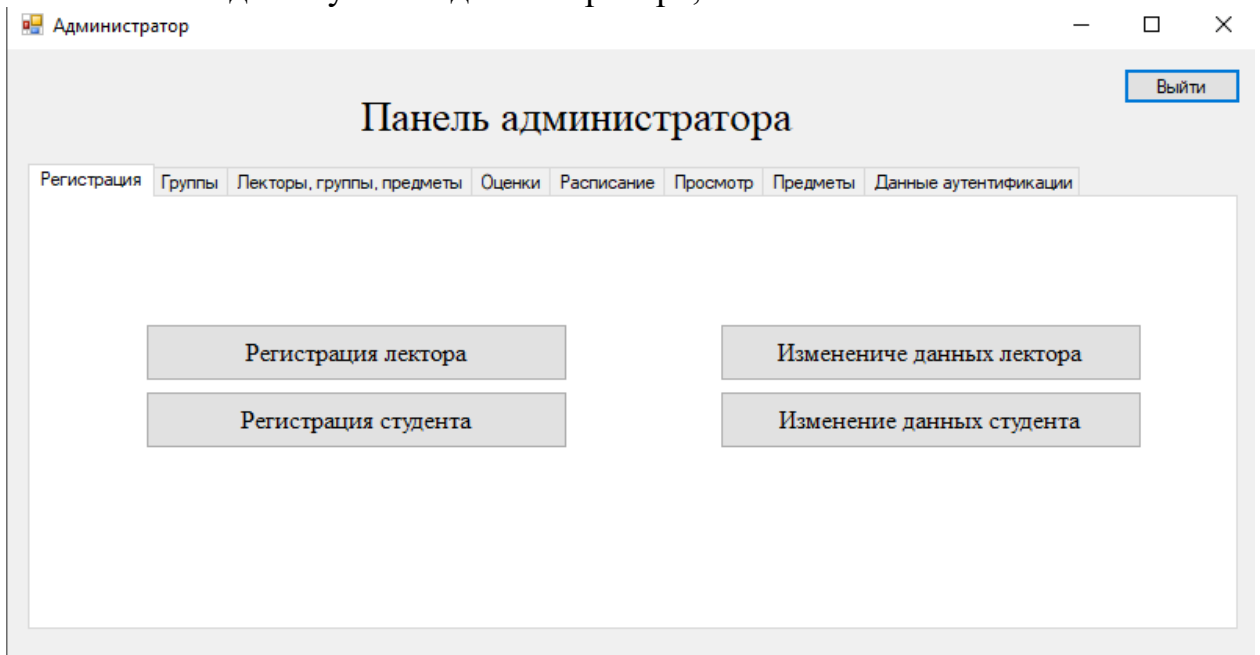
При запуске программы появляется окно авторизации:



При вводе неправильных данных, появится окно ошибки:



Если зайти под аккаунтом администратора, то появится его окно:



Формирование группы администратором:

Формирование группы

Номер группы

Количество человек

Дата начала обучения

Также у администратор имеется вкладка с оценками:

Администратор

Панель администратора

Регистрация Группы Лекторы, группы, предметы **Оценки** Расписание Просмотр Предметы Данные аутентификации

Добавление оценки администратором:

Добавить оценку

Почта	<input type="text" value="alexeyOleg@mail.ru"/>	Студент	Имя	<input type="text" value="Алексей"/>
Дата получения	<input type="text" value="03.12.2022"/>	Фамилия	<input type="text" value="Петров"/>	
Семестр	<input type="text"/>	Отчество	<input type="text" value="Олегович"/>	
Оценка	<input type="text"/>	Лектор	Имя	<input type="text" value="Алексей"/>
Лектор	<input type="text" value="alexeyIvanov@mail.ru"/>	Фамилия	<input type="text" value="Иванов"/>	
Предмет	<input type="text" value="C#"/>	Отчество	<input type="text" value="Сергеевич"/>	

Просмотр студентов по конкретной оценке:

Показ студентов по конкретной оценке

	Имя	Фамилия	Отчество	Номер группы	Название предмета	Оценка
▶	Алексей	Петров	Олегович	353/01	C#	<input type="text" value="3"/>
	Алексей	Петров	Олегович	353/01	Kotlin	
	Алексей	Петров	Олегович	353/01	Вычислительна...	
	Антон	Павлов	Дмитриевич	353/01	C#	
	Антон	Павлов	Дмитриевич	353/01	Вычислительна...	
	Иван	Петров	Дмитриевич	353/02	C++	

Администратор также имеет возможность редактирования оценки:

Обновить оценку

Почта студента alexeyOleg@mail.ru

Почта лектора alexeyivanov@mail.ru

Предмет С#

Дата 01.09.2022

Семестр 1

Оценка 2

Найти

Почта студента alexeyOleg@mail.ru

Дата 03.12.2022

Семестр

Оценка

Лектор alexeyivanov@mail.ru

Предмет С#

Изменить

Показать

Закрыть

Добавления расписания:

Добавить запись в расписание

Дата 03.12.2022

Группа 353/01

Лектор alexeyivanov@mail.ru

Предмет С#

Формат пары Очно

Кабинет 2

Корпус 2

Добавить

Закрыть

Редактирование расписания:

Редактирование расписания

Группа	353/01
Лектор	alexeyivanov@mail.ru
Предмет	C#
Дата	21.11.2022
Формат пары	Очно
Кабинет	23
Корпус	2
Найти	

Группа	353/01
Лектор	alexeyivanov@mail.ru
Предмет	C#
Дата	03.12.2022
Формат пары	
Кабинет	
Корпус	
Изменить	

Закреть

### Добавление предмета:

Регистрация Группы Лекторы, группы, предметы

	Имя	Фамилия	С
▶	Алексей	Иванов	С
	Алексей	Иванов	С
	Алексей	Иванов	С
	Алексей	Иванов	С
	Алексей	Иванов	С
	Виталий	Смирнов	О
	Константин	Марков	Е
	Виталий	Смирнов	О
	Василий	Пустов	Константинович

Добавить новое...

Название

Описание

Добавить

Закреть

Данные аутентификации

Добавить предмет

Добавить лектору группу и предмет

Посмотреть группы и их лекторов с предметами

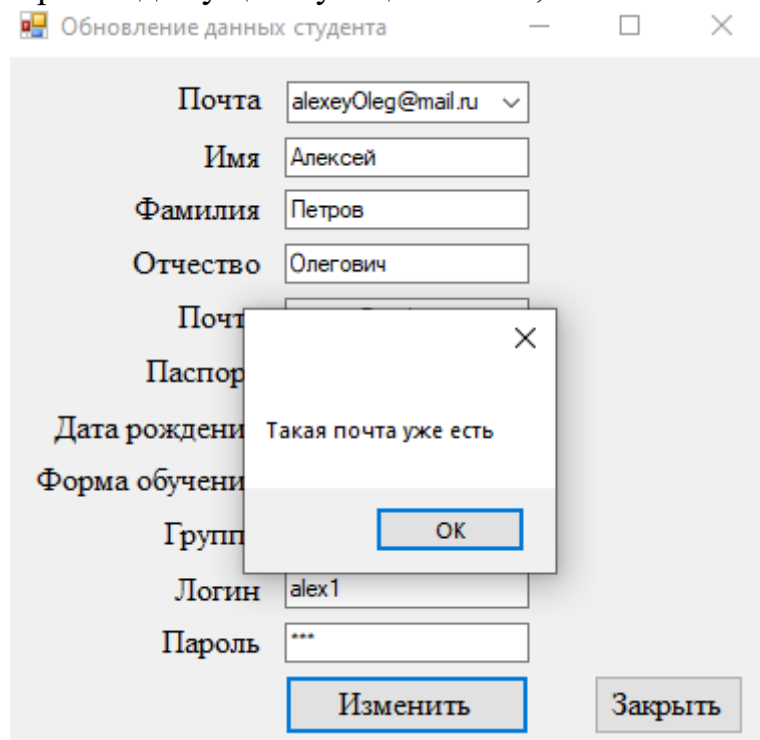
### Добавление лектору новой группы и предмета.

Добавить лектору новую группу с предметом

Почта	ale1x@mail.ru	Лектор	Имя	Алексей
Группа	353/01	Фамилия	Иванов	
Предмет	C#	Отчество	Алексеевич	
Добавить				
Закреть				

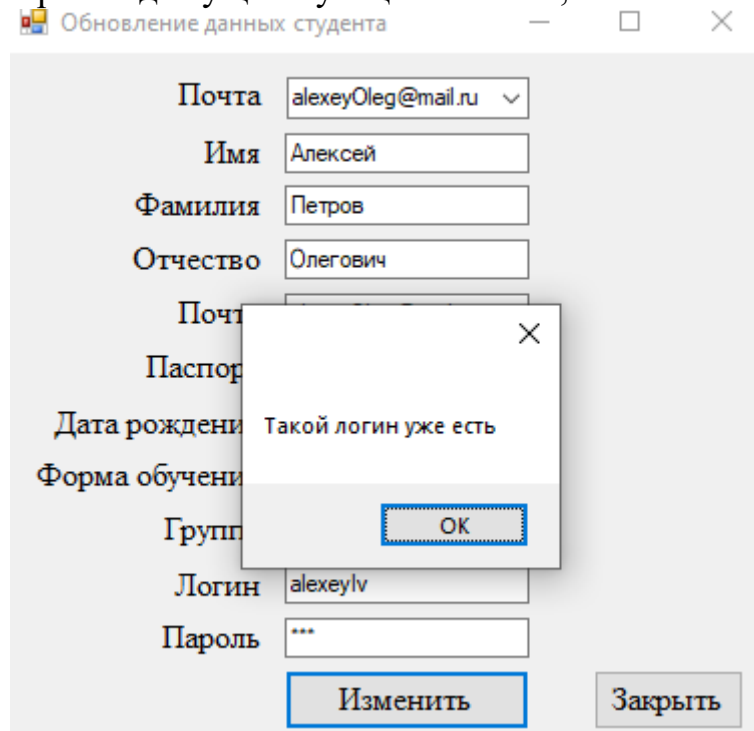
Также имеются проверки во многих местах на корректный ввод. Проверка реализованы на уровне программы, сначала проверяются данные, а потом уже вызывается процедура.

При вводе существующей почты, появится окно ошибки:



The screenshot shows a window titled "Обновление данных студента" (Update student data). It contains several input fields: "Почта" (Email) with "alexeyOleg@mail.ru", "Имя" (Name) with "Алексей", "Фамилия" (Surname) with "Петров", "Отчество" (Patronymic) with "Олегович", "Паспорт" (Passport), "Дата рождения" (Date of birth), "Форма обучения" (Form of study), "Группа" (Group), "Логин" (Login) with "alex1", and "Пароль" (Password) with "\*\*\*". There are "Изменить" (Change) and "Заккрыть" (Close) buttons at the bottom. A modal error dialog is displayed in the center with the text "Такая почта уже есть" (This email already exists) and an "ОК" button.

При вводе существующего логина, появится окно ошибки:

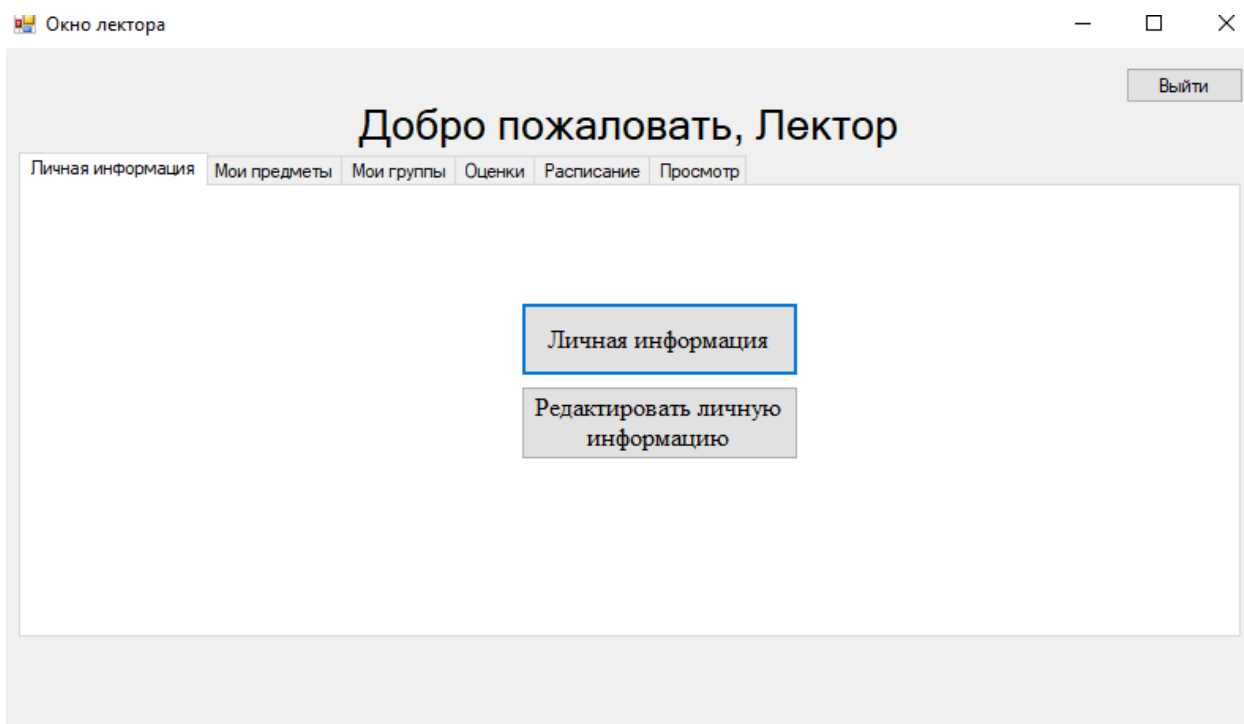


The screenshot shows the same "Обновление данных студента" window. The "Логин" (Login) field now contains "alexeylv". The modal error dialog displays the text "Такой логин уже есть" (This login already exists) with an "ОК" button.

Если зайти под аккаунтом лектора, то для каждого лектора будет своя информация, которая соответствует ему.

Окно лектора:





Личная информация лектора:

The screenshot shows a window titled 'Данные лектора' (Lecturer Data). The form contains the following fields and values:

Имя	Алексей
Фамилия	Иванов
Отчество	Сергеевич
Почта	alexeyivanov@mail.
Телефон	+912832631
Паспорт	2343 343422
Дата рождения	30.11.1975
Зарплата	32432
Логин	alexeylv
Пароль	***

At the bottom right of the form, there is a 'Закрыть' (Close) button.

Редактирование личной информации:

Обновление данных лект...

Имя	Алексей
Фамилия	Иванов
Отчество	Сергеевич
Почта	alexeyivanov@mail.
Телефон	+912832631
Паспорт	2343 343422
Дата рождения	30.11.1975
Зарплата	32432
Логин	alexeylv
Пароль	***

Изменить

Заккрыть

У лектора есть кнопка мои предметы, там можно увидеть предметы, которые он ведет:

Предметы лектора

	Имя	Фамилия	Отчество	Название предмета
▶	Алексей	Иванов	Сергеевич	C#
	Алексей	Иванов	Сергеевич	Kotlin
	Алексей	Иванов	Сергеевич	Вычислительна...
	Алексей	Иванов	Сергеевич	Вышмат

Заккрыть

У лектора есть возможность посмотреть свои группы, у которых он ведет пары.

	Имя	Фамилия	Отчество	Номер группы	Предмет
▶	Алексей	Иванов	Сергеевич	353/01	C#
	Алексей	Иванов	Сергеевич	353/01	Kotlin
	Алексей	Иванов	Сергеевич	353/01	Вычислительна...
	Алексей	Иванов	Сергеевич	353/02	Вычислительна...
	Алексей	Иванов	Сергеевич	353/02	Вышмат

Все группы

Группы, у которых веду пары

Студенты конкретной группы

У лектора есть возможность добавления оценки:

Добавление оценки лектором
—
□
×

Почта

alexeyOleg@mail.ru

Дата получения

03.12.2022

Семестр

Оценка

Лектор

alexeyIvanov@mail.ru

Предмет

C#

Добавить оценку

Студент

Имя

Алексей

Фамилия

Петров

Отчество

Олегович

Лектор

Имя

Алексей

Фамилия

Иванов

Отчество

Сергеевич

Заккрыть

Также есть возможность смотреть расписание по конкретной группе на конкретную неделю.

ShowTimetable

День недели	Дата пары	Имя	Фамилия	Отчество	Предмет	Формат	Кабинет	Корпус
▶ понедельник	28.11.2022	Алексей	Иванов	Сергеевич	C#	Очно	2	2
вторник	29.11.2022	Алексей	Иванов	Сергеевич	Вычислительна...	Очно	4	2
среда	30.11.2022	Алексей	Иванов	Сергеевич	Kotlin	Очно	23	1
среда	30.11.2022	Константин	Марков	Евгеньевич	Операционные ...	Очно	11	3
четверг	01.12.2022	Алексей	Иванов	Сергеевич	Kotlin	Очно	11	2
четверг	01.12.2022	Виталий	Смирнов	Олегович	Архитектура ЭВМ	Очно	33	2
пятница	02.12.2022	Алексей	Иванов	Сергеевич	C#	Очно	44	4
пятница	02.12.2022	Алексей	Иванов	Сергеевич	Вычислительна...	Очно	1	2

Номер группы  
353/01

Дата  
03.12.2022

Показать

Заккрыть

Окно студенты выглядит следующим образом:

Окно студента

Добро пожаловать, студент!

Выйти

Моя личная информация | Мои оценки | Мои лекторы | Мои предметы | Мои одногруппники | Расписание

Моя личная информация

Редактирование личной информации

Окно просмотра личной информации:

Информация студента

Имя	<input type="text" value="Алексей"/>
Фамилия	<input type="text" value="Петров"/>
Отчество	<input type="text" value="Олегович"/>
Почта	<input type="text" value="alexeyOleg@mail.ru"/>
Паспорт	<input type="text" value="4354 327723"/>
Дата рождения	<input type="text" value="12.11.1983"/>
Форма обучения	<input type="text" value="Контракт"/>
Группа	<input type="text" value="353/01"/>
Логин	<input type="text" value="alex1"/>
Пароль	<input type="password" value="***"/>
<input type="button" value="Заккрыть"/>	

Окно редактирования личной информации:

UpdateStudentByStudent

Имя	<input type="text" value="Алексей"/>
Фамилия	<input type="text" value="Петров"/>
Отчество	<input type="text" value="Олегович"/>
Почта	<input type="text" value="alexeyOleg@mail.ru"/>
Паспорт	<input type="text" value="4354 327723"/>
Дата рождения	<input type="text" value="12.11.1983"/>
Форма обучения	<input type="text" value="Контракт"/>
Группа	<input type="text" value="353/01"/>
Логин	<input type="text" value="alex1"/>
Пароль	<input type="password" value="***"/>
<input type="button" value="Изменить"/>	
<input type="button" value="Заккрыть"/>	

Студент имеет возможность просмотреть свои оценки по предметам.

Показ оценок конкретного студента

	Имя студента	Фамилия студента	Отчество студента	Оценка	Дата получения оценки	Название предмета	Имя лектора	Фамилия лектора	Отчество лектора
▶	Алексей	Петров	Олегович	3	05.12.2022	Вычислительна...	Алексей	Иванов	Сергеевич
	Алексей	Петров	Олегович	3	25.11.2022	C#	Алексей	Иванов	Сергеевич
	Алексей	Петров	Олегович	4	04.11.2022	C#	Алексей	Иванов	Сергеевич
	Алексей	Петров	Олегович	4	09.10.2022	Kotlin	Алексей	Иванов	Сергеевич
	Алексей	Петров	Олегович	4	23.11.2022	C#	Алексей	Иванов	Сергеевич
	Алексей	Петров	Олегович	3	02.12.2022	Kotlin	Алексей	Иванов	Сергеевич
	Алексей	Петров	Олегович	2	01.09.2022	C#	Алексей	Иванов	Сергеевич
	Алексей	Петров	Олегович	2	02.12.2022	C#	Алексей	Иванов	Сергеевич

Заккрыть

Есть функция просмотра среднего балла:

3,1 Мой средний бал

Посмотреть своих лекторов:

Мои лекторы

	Имя	Фамилия	Отчество	Предмет	Описание предмета
▶	Алексей	Иванов	Сергеевич	C#	Нужен Visual Stu...
	Алексей	Иванов	Сергеевич	Kotlin	
	Алексей	Иванов	Сергеевич	Вычислительна...	
	Виталий	Смирнов	Олегович	Архитектура ЭВМ	
	Константин	Марков	Евгеньевич	Операционные ...	

Заккрыть

Посмотреть свои предметы:

	Предмет	Описание предмета
►	C#	Нужен Visual Stu...
	Kotlin	
	Архитектура ЭВМ	
	Вычислительна...	
	Операционные ...	

Показать

Может посмотреть своих одноклассников:

	Имя	Фамилия	Отчество	Почта
►	Алексей	Петров	Олегович	alexeyOleg@mail...
	Олег	Сидоров	Александрович	olegSid@mail.ru
	Антон	Павлов	Дмитриевич	antonPav@mail.ru

Заккрыть

## Заключение

В ходе курсовой работы была реализована система для работы кафедры университета. Для её реализации была использована СУБД MS SQL Server. Использовался язык программирования C# на платформе Visual Studio 2022 Community Version. Были трудности по проектированию пользовательского интерфейса. Нужно было сделать так, чтобы программой мог пользоваться любой человек. Для этого нужно было удобно расположить все кнопки с функциями.



# Список литературы

1. <https://learn.microsoft.com/en-us/dotnet/csharp/>

## Приложение

```
USE [UniversityDepartment]
GO
/***** Object: Table [dbo].[assessments]    Script Date: 03.12.2022 22:53:57 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[assessments](
    [assessment_id] [int] IDENTITY(1,1) NOT NULL,
    [student_id] [int] NOT NULL,
    [passed_date] [date] NOT NULL,
    [semestr] [int] NOT NULL,
    [assessment] [int] NOT NULL,
    [lector_group_subject_id] [int] NOT NULL,
    CONSTRAINT [PK__assessme__00B98C2600096C3E] PRIMARY KEY CLUSTERED
(
    [assessment_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[groups]    Script Date: 03.12.2022 22:53:57 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[groups](
    [group_id] [int] IDENTITY(1,1) NOT NULL,
    [group_number] [nvarchar](30) NOT NULL,
    [count_person] [int] NOT NULL,
    [date_begin] [date] NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [group_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[lectors]    Script Date: 03.12.2022 22:53:57 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[lectors](
    [lector_id] [int] IDENTITY(1,1) NOT NULL,
    [first_name] [nvarchar](50) NOT NULL,
    [last_name] [nvarchar](50) NOT NULL,
    [father_name] [nvarchar](50) NULL,
    [email] [nvarchar](50) NOT NULL,
    [mobile_phone] [nvarchar](30) NOT NULL,
    [passport] [nvarchar](30) NOT NULL,
    [dob] [date] NOT NULL,
    [salary] [money] NOT NULL,
    [login_id] [int] NOT NULL,
    PRIMARY KEY CLUSTERED
(
```

```

        [lector_id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
    [PRIMARY],
    UNIQUE NONCLUSTERED
    (
        [mobile_phone] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
    [PRIMARY],
    UNIQUE NONCLUSTERED
    (
        [passport] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
    [PRIMARY],
    UNIQUE NONCLUSTERED
    (
        [email] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
    [PRIMARY],
    UNIQUE NONCLUSTERED
    (
        [login_id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
    [PRIMARY]
    ) ON [PRIMARY]
GO
/***** Object: Table [dbo].[lectors_groups_subjects]    Script Date: 03.12.2022
22:53:57 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[lectors_groups_subjects](
    [lector_group_subject_id] [int] IDENTITY(1,1) NOT NULL,
    [lector_id] [int] NOT NULL,
    [group_id] [int] NOT NULL,
    [subject_id] [int] NOT NULL,
    CONSTRAINT [PK__lectors__9AF358957935A97E] PRIMARY KEY CLUSTERED
    (
        [lector_group_subject_id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
    [PRIMARY]
    ) ON [PRIMARY]
GO
/***** Object: Table [dbo].[logins]    Script Date: 03.12.2022 22:53:57 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[logins](
    [login_id] [int] IDENTITY(1,1) NOT NULL,
    [login] [nvarchar](100) NOT NULL,
    [password] [nvarchar](100) NOT NULL,
    [role] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK__logins__C2C971DB0E6D39CD] PRIMARY KEY CLUSTERED
    (
        [login_id] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY],
    CONSTRAINT [UQ__logins__7838F272C1BE29C3] UNIQUE NONCLUSTERED
(
    [login] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[students]    Script Date: 03.12.2022 22:53:57 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[students](
    [student_id] [int] IDENTITY(1,1) NOT NULL,
    [first_name] [nvarchar](50) NOT NULL,
    [last_name] [nvarchar](50) NOT NULL,
    [father_name] [nvarchar](50) NULL,
    [email] [nvarchar](50) NOT NULL,
    [passport] [nvarchar](30) NOT NULL,
    [dob] [date] NOT NULL,
    [education_form] [nvarchar](15) NOT NULL,
    [group_id] [int] NOT NULL,
    [login_id] [int] NOT NULL,
    CONSTRAINT [PK__students__2A33069A6D53A9E2] PRIMARY KEY CLUSTERED
(
    [student_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY],
    CONSTRAINT [UQ__students__5E2A085707760F04] UNIQUE NONCLUSTERED
(
    [passport] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY],
    CONSTRAINT [UQ__students__AB6E61642057089A] UNIQUE NONCLUSTERED
(
    [email] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY],
    CONSTRAINT [UQ__students__C2C971DAD34D3E4C] UNIQUE NONCLUSTERED
(
    [login_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[subjects]    Script Date: 03.12.2022 22:53:57 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[subjects](
    [subject_id] [int] IDENTITY(1,1) NOT NULL,
    [subject_name] [nvarchar](50) NOT NULL,
    [subject_description] [nvarchar](200) NULL,
    PRIMARY KEY CLUSTERED
(

```

```

        [subject_id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
    [PRIMARY]
    ) ON [PRIMARY]
GO
/***** Object: Table [dbo].[timetables]    Script Date: 03.12.2022 22:53:57 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[timetables](
    [timetable_id] [int] IDENTITY(1,1) NOT NULL,
    [day_week] [nvarchar](15) NOT NULL,
    [date_par] [date] NOT NULL,
    [lector_group_subject_id] [int] NOT NULL,
    [format_par] [nvarchar](20) NOT NULL,
    [cabinet] [int] NOT NULL,
    [housing] [int] NOT NULL,
    CONSTRAINT [PK__timetabl__26DC9588211C4C01] PRIMARY KEY CLUSTERED
(
    [timetable_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
    [PRIMARY]
    ) ON [PRIMARY]
GO
ALTER TABLE [dbo].[assessments] WITH CHECK ADD CONSTRAINT
    [fk_assessments_lectors_groups_subjects] FOREIGN KEY([lector_group_subject_id])
    REFERENCES [dbo].[lectors_groups_subjects] ([lector_group_subject_id])
    ON DELETE CASCADE
GO
ALTER TABLE [dbo].[assessments] CHECK CONSTRAINT [fk_assessments_lectors_groups_subjects]
GO
ALTER TABLE [dbo].[assessments] WITH CHECK ADD CONSTRAINT [fk_assessments_students]
    FOREIGN KEY([student_id])
    REFERENCES [dbo].[students] ([student_id])
GO
ALTER TABLE [dbo].[lectors_groups_subjects] CHECK CONSTRAINT
    [fk_lectors_groups_subjects_groups]
GO
ALTER TABLE [dbo].[lectors_groups_subjects] CHECK CONSTRAINT
    [fk_lectors_groups_subjects_lectors]
GO
ALTER TABLE [dbo].[lectors_groups_subjects] WITH CHECK ADD CONSTRAINT
    [fk_lectors_groups_subjects_subjects] FOREIGN KEY([subject_id])
    REFERENCES [dbo].[subjects] ([subject_id])
    ON DELETE CASCADE
GO
ALTER TABLE [dbo].[lectors_groups_subjects] CHECK CONSTRAINT
    [fk_lectors_groups_subjects_subjects]
GO
ALTER TABLE [dbo].[students] WITH CHECK ADD CONSTRAINT [fk_students_groups] FOREIGN
    KEY([group_id])
    REFERENCES [dbo].[groups] ([group_id])
GO
ALTER TABLE [dbo].[students] CHECK CONSTRAINT [fk_students_groups]
GO
ALTER TABLE [dbo].[students] WITH CHECK ADD CONSTRAINT [FK_students_logins] FOREIGN
    KEY([login_id])
    REFERENCES [dbo].[logins] ([login_id])
GO
ALTER TABLE [dbo].[students] CHECK CONSTRAINT [FK_students_logins]
GO

```

```

ALTER TABLE [dbo].[timetables] WITH CHECK ADD CONSTRAINT [fk_timetables_lectors_groups]
FOREIGN KEY([lector_group_subject_id])
REFERENCES [dbo].[lectors_groups_subjects] ([lector_group_subject_id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[timetables] CHECK CONSTRAINT [fk_timetables_lectors_groups]
GO
ALTER TABLE [dbo].[assessments] WITH CHECK ADD CONSTRAINT
[CK__assessmen__asses__3D5E1FD2] CHECK (([assessment]>=(2) AND [assessment]<=(5)))
GO
ALTER TABLE [dbo].[assessments] CHECK CONSTRAINT [CK__assessmen__asses__3D5E1FD2]
GO
ALTER TABLE [dbo].[assessments] WITH CHECK ADD CONSTRAINT
[CK__assessmen__semes__3E52440B] CHECK (([semestr]>=(1) AND [semestr]<=(8)))
GO
ALTER TABLE [dbo].[assessments] CHECK CONSTRAINT [CK__assessmen__semes__3E52440B]
GO
ALTER TABLE [dbo].[timetables] WITH CHECK ADD CONSTRAINT
[CK__timetable__cabin__4316F928] CHECK (([cabinet]>=(1) AND [cabinet]<=(100)))
GO
ALTER TABLE [dbo].[timetables] CHECK CONSTRAINT [CK__timetable__cabin__4316F928]
GO
ALTER TABLE [dbo].[timetables] WITH CHECK ADD CONSTRAINT
[CK__timetable__housi__440B1D61] CHECK (([housing]>=(1) AND [housing]<=(10)))
GO
ALTER TABLE [dbo].[timetables] CHECK CONSTRAINT [CK__timetable__housi__440B1D61]
GO

```