

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ ГАГАРИНА Ю.А.»

Кафедра «Естественные и математические науки»

Специальность ПИНЖ — Программная инженерия

Проект базы данных к информационной системе  
«Личный кабинет образовательной организации.  
Подсистема личный кабинет студента»

Выполнил: студент X курса  
учебной группы XXXXXX  
очной формы обучения  
XXXXXXXXXX,

Проверил: преподаватель кафедры  
ЕМН XXXXXXXX

## СОДЕРЖАНИЕ

1 База данных	3
1.1 Анализ предметной области «Личный кабинет образовательной организации. Подсистема студенческого кабинета»	3
1.2 Модель сущность-связь	10
1.3 Дата-логическое (логическое) проектирование базы данных	10
1.4 Физическое моделирование базы данных	12
2 Функциональные возможности	26
2.1 Запросы на редактирование	26
2.1.1 Запрос на добавление записи в таблицу	26
2.1.2 Запрос на изменение данных таблицы	26
2.1.3 Запрос на удаление данных из таблицы	27
2.2 Запросы на выборку данных	27
2.2.1 Запрос на фильтрацию данных из таблицы	27
2.2.2 Запрос с использованием агрегатных функций	29
2.2.3 Запрос с объединением нескольких таблиц (JOIN)	29
2.2.4 Запрос на проверку уникальности данных	30
2.3 Триггеры	30
2.4 Процедуры	31

## **1 База данных**

### **1.1 Анализ предметной области «Личный кабинет образовательной организации. Подсистема студенческого кабинета»**

Областью автоматизации является учет достижений и формирования рейтинга студентов в рамках информационной системы «Личный кабинет образовательной организации».

В настоящее время в вузах процесс сбора информации о внеучебной деятельности студентов (научной, спортивной, культурной, общественной) (см. Приложениями Е, Ж, З) часто децентрализован. Студенты приносят грамоты и сертификаты в бумажном виде или отправляют сканы по электронной почте, что затрудняет подсчет рейтинговых баллов для назначения повышенных стипендий.

Разрабатываемая подсистема «Студенческий кабинет» предназначена для решения проблемы централизованного хранения и обработки результатов личностного/профессионального роста студента. Основной акцент в рамках данной работы сделан на модулях «Профиль» и «Портфолио».

#### **Нормативные документы**

1     Российская Федерация. Законы. Об образовании в Российской Федерации [Электронный ресурс] : федеральный закон от 29.12.2012 № 273-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_140174/](https://www.consultant.ru/document/cons_doc_LAW_140174/). – (Дата обращения: 10.10.2025). – Ст. 30, ч. 2.

Требования к содержанию, оформлению, ведению, хранению личных дел обучающихся по образовательным программам высшего образования федеральные нормативные акты не устанавливают, поэтому указанные требования определяются образовательной организацией самостоятельно, в том числе необходимость ведения личной карточки студента устанавливается самим вузом.

2      Российская Федерация. Законы. Об образовании в Российской Федерации [Электронный ресурс] : федеральный закон от 29.12.2012 № 273-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_140174/](https://www.consultant.ru/document/cons_doc_LAW_140174/). – (Дата обращения: 15.12.2025). – Ст. 28, ч. 3, п. 1, 11.

Следовательно единый порядок ведения личного дела студента законодательно не установлен. Следовательно, образовательная организация вправе самостоятельно разработать и утвердить такой порядок, в т.ч. включив в него порядок индивидуального учета результатов освоения образовательных программ и поощрений, а также хранения в архивах информации об этих результатах и поощрениях на бумажных и (или) электронных носителях

3      Российская Федерация. Законы. О персональных данных [Электронный ресурс] : федеральный закон от 27.07.2006 № 152-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_61801/](https://www.consultant.ru/document/cons_doc_LAW_61801/). – (Дата обращения: 15.12.2025).

Регулирует сбор, хранение, обработку и защиту личной информации студентов.

4      Министерство науки и высшего образования Российской Федерации. Об утверждении порядка приема на обучение по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры [Электронный ресурс] : приказ от 27.11.2024 № 821 // КонтурНорматив. – Режим доступа: <https://normativ.kontur.ru/document?moduleId=1&documentId=503133> (Дата обращения: 20.10.2025).

Устанавливает необходимость личного дела студента

5      Министерство обороны Российской Федерации. Об определении Порядка обучения граждан Российской Федерации в военных образовательных организациях высшего образования по программе военной подготовки сержантов, старшин запаса либо по программе военной

подготовки солдат, матросов запаса [Электронный ресурс] : приказ от 30.11.2018 № 693 (с изм. от 21.04.2023) // Гарант. – Режим доступа: <https://base.garant.ru/72131136/>. – (Дата обращения: 15.12.2025). – Прил.: Порядок обучения... Прил. 1: Личная карточка студента (рекомендуемый образец).

Рекомендуемый образец личной карточки, которую может взять и адаптировать под собственные нужды каждый институт.

6 Росархив. Об утверждении Перечня типовых управленческих архивных документов, образующихся в процессе деятельности государственных органов, органов местного самоуправления и организаций, с указанием сроков хранения [Электронный ресурс] : приказ от 20.12.2019 № 236 // КонсультантПлюс. – Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_345020/](https://www.consultant.ru/document/cons_doc_LAW_345020/). – (Дата обращения: 15.12.2025).

Определяет сроки хранения данных выпускников и отчисленных (например, 75 лет для личных дел). Влияет на политику архивирования и удаления записей из таблицы.

7 Рособрнадзор. Об утверждении Требований к структуре официального сайта образовательной организации в информационно-телекоммуникационной сети «Интернет» и формату представления на нем информации [Электронный ресурс] : приказ от 04.08.2023 № 1493 // КонсультантПлюс. – Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_463082/](https://www.consultant.ru/document/cons_doc_LAW_463082/) – (Дата обращения: 20.10.2025).

Определяет общие требования к публичной части сайта, но также включает общие требования к формату представления информации, распространяющиеся и на закрытые разделы.

8 ГОСТ Р 59382-2021. Информационные технологии. Методы и средства обеспечения безопасности. Основы управления идентичностью

[Электронный ресурс]. – Режим доступа: <https://base.garant.ru/403448112/> – (Дата обращения: 20.10.2025).

Регламентирует процессы создания, хранения, использования и удаления учетных записей студентов и преподавателей.

9 ГОСТ Р 52872-2019. Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Применение для инвалидов по зрению. Требования доступности [Электронный ресурс]. – Режим доступа: <https://base.garant.ru/73664694/> – (Дата обращения: 15.12.2025).

Обязательный стандарт для сайтов государственных и образовательных учреждений. Регулирует реализацию версии для слабовидящих, обеспечивая доступность контента ЛК.

10 ГОСТ Р ИСО 9241-210-2016. Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем [Электронный ресурс]. – Режим доступа: <https://internet-law.ru/gosts/gost/63110/>. – (Дата обращения: 15.12.2025).

В котором говорится, что у пользователя должен отсутствовать дискомфорта при использовании продукции и положительное отношение к ней.

11 Образец заявления на предоставление места в общежитии : приложение к Правилам внутреннего распорядка / Приволжский политехнический колледж ФГБОУ ВО СГТУ имени Гагарина Ю.А. – 2022. – URL: <https://ppk.sstu.ru/wp-content/uploads/2022/11/%D0%9E%D0%B1%D1%80%D0%B0%D0%B7%D1%86%D1%8B-%D0%B7%D0%B0%D1%8F%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9.pdf> (дата обращения: 10.11.2025).

В образце представлен образец заявления на предоставления места в общежитии

12 Положение о ведении журнала учебных занятий Энгельсского технологического института (филиала) СГТУ им. Гагарина Ю.А. : [утв.

директором ЭТИ (филиал) СГТУ А.В. Яковлевым в 2017 г.]. – Энгельс : ЭТИ (филиал) СГТУ им. Гагарина Ю.А., 2017. – URL: [http://techn.sstu.ru/Documentation/SPO/Doc/%D0%9F%D0%BE%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5\\_%D0%AD%D0%A2%D0%98\\_%D0%BE\\_%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B8-%D0%B6%D1%83%D1%80%D0%BD%D0%B0%D0%BB%D0%B0-%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D1%8B%D1%85-%D0%B7%D0%B0%D0%BD%D1%8F%D1%82%D0%B8%D0%B9.doc](http://techn.sstu.ru/Documentation/SPO/Doc/%D0%9F%D0%BE%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D0%AD%D0%A2%D0%98_%D0%BE_%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B8-%D0%B6%D1%83%D1%80%D0%BD%D0%B0%D0%BB%D0%B0-%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D1%8B%D1%85-%D0%B7%D0%B0%D0%BD%D1%8F%D1%82%D0%B8%D0%B9.doc) (дата обращения: 10.11.2025).

Регламентирует порядок заполнения, учета и хранения основного документа, фиксирующего учебный процесс

13 Приказ Федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Саратовский государственный технический университет имени Гагарина Ю.А.» от 29.05.2015 № 584-П «О формировании электронного портфолио обучающихся».

В качестве примера внутренней документации вуза, регулирующей структуру электронного портфолио, была рассмотрена служебная записка СГТУ имени Гагарина Ю.А. от начальника учебно-методического управления Лисицкой Н.М.

**Основные процессы учета достижений, автоматизируемые системой:**

1. **Идентификация студента:** Хранение персональных данных, информации о группе, специальности и форме обучения.

2. **Ведение портфолио:** Студент имеет возможность добавлять достижения в базу данных, указывая категорию деятельности (например, «Научная деятельность»), уровень мероприятия и прикрепляя подтверждающие документы.

3. **Расчет рейтинга:** Система автоматически сопоставляет достижение с критериями оценки и начисляет баллы. Рейтинг формируется в разрезе академических периодов (семестров).

Входные данные:

- 1) Личные данные студента [5]
- 2) Информация о достижениях [13]

Выводные данные

- 1) Персональная информация
- 2) Рейтинг и баллы
- 3) История достижений

Сформулируем возможные вопросы, ответы на которые можно будет получить с помощью запросов к базе данных:

1. Какие личные и академические данные имеет студент?
2. Какие достижения имеет студент?
3. Как обновить контакты студента?
4. Как удалить достижения?
5. Как обновить контакты?
6. Каков суммарный рейтинг студента по категориям?
7. Какие достижения имеет студент за определенный период/направление?

Анализ описания предметной области позволяет определить следующие независимые информационные объекты (сущности): СТУДЕНТ, КУРАТОР, СТУДЕНЧЕСКАЯ ГРУППА, ДОСТИЖЕНИЕ, КРИТЕРИЙ БАЛЛОВ ДОСТИЖЕНИЯ.

Анализ описания предметной области позволяет выделить следующий набор данных (атрибутов), которые должны храниться в проектируемой базе данных (в скобках указано краткое имя атрибута):

**Студент (Student) [5]**

1. Фамилия студента (surname)
2. Имя студента (first\_name)



3. Отчество студента (middle\_name)
4. Дата рождения (birth\_date)
5. Студенческая группа (student\_group\_id)
6. Номер зачетной книжки (gradebook\_number)
7. Номер телефона (phone)
8. Электронная почта (email)
9. Адрес проживания (address)
10. Пароль для входа (student\_password)
11. Фотография профиля (profile\_photo)

#### **Куратор (tutor)**

1. Фамилия (surname)
2. Имя (first\_name)
3. Отчество (middle\_name)

#### **Студенческая группа (Student\_group)**

1. Название группы (naming)
2. Специальность (specialty\_id)
3. Форма обучения (form\_study\_id)
4. Куратор (tutor\_id)
5. Номер курса (course)

#### **Достижение студента (Student\_Achievement) [13]**

1. Студент (student\_id)
2. Балл (criteria\_id)
3. Семестр (period\_id)
4. Количество достижений (quantity)
5. Название подтверждающего документа (document\_title)
6. Дата загрузки (created\_at)

**Критерий баллов достижения (Rating\_Criteria)** (см. Приложения Е, Ж, З)

1. Категория деятельности (category\_id)
2. Название раздела критериев (section\_naming)

3. Текст описания достижения (description\_text)
4. Уровень мероприятия (level\_type\_id)
5. Количество баллов (points)

## 1.2 Модель сущность-связь

Используя информационные объекты (сущности) спроектируем модель «сущность-связь», представленную на рисунке 1.

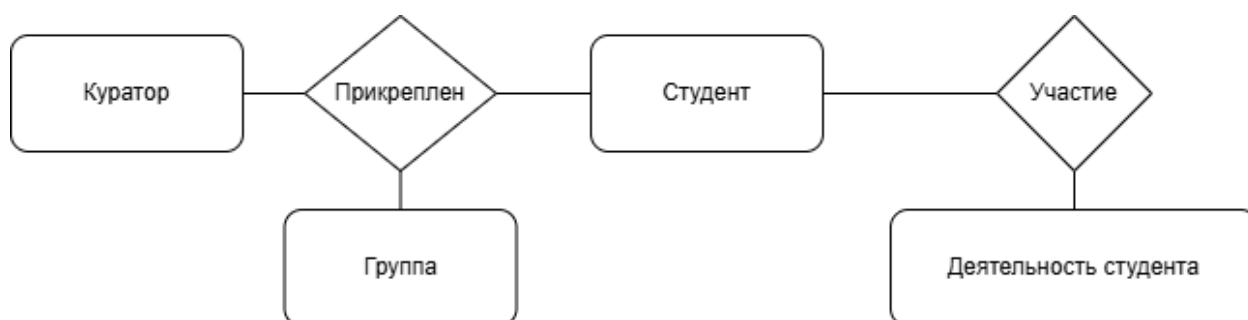


Рисунок 1 – Модель «Сущность-связь»

## 1.3 Дата-логическое (логическое) проектирование базы данных

Перейдем к созданию дата-логической модели базы данных. Для этого поставим в соответствие каждой сущности реляционную таблицу с соответствующими атрибутами и ключами.

Перед построением дата логической модели приведем нашу базу данных к третьей нормальной форме.

Нормализация к 1НФ не требуется, все атрибуты были атомизированы на этапе концептуального моделирования, согласно [5] (см. Приложения Е, Ж, З)

В процессе нормализации до 2НФ сформулированы идентификаторы к таблицам:

Выделены справочники: НАПРАВЛЕНИЯ, УРОВЕНЬ МЕРОПРИТИЯ, СПЕЦИАЛЬНОСТЬ, ФОРМА ОБУЧЕНИЯ, ПЕРИОДЫ ОБУЧЕНИЯ, КРИТЕРИИ

В процессе формирования третьей нормальной формы были сформированы связи: в таблице СТУДЕНТЧЕСКАЯ ГРУППА атрибуты «Специальность» и «Форма обучения» были заменены на внешние ключи, ведущие на справочники СПЕЦИАЛЬНОСТЬ и ФОРМА ОБУЧЕНИЯ соответственно. В таблице ДОСТИЖЕНИЯ атрибуты «Семестр» и «Балл» были заменены на внешние ключи, ведущие на справочники СПЕЦИАЛЬНОСТЬ и ФОРМА ОБУЧЕНИЯ соответственно.

Дата-логическая модель представленную на рисунке 2.

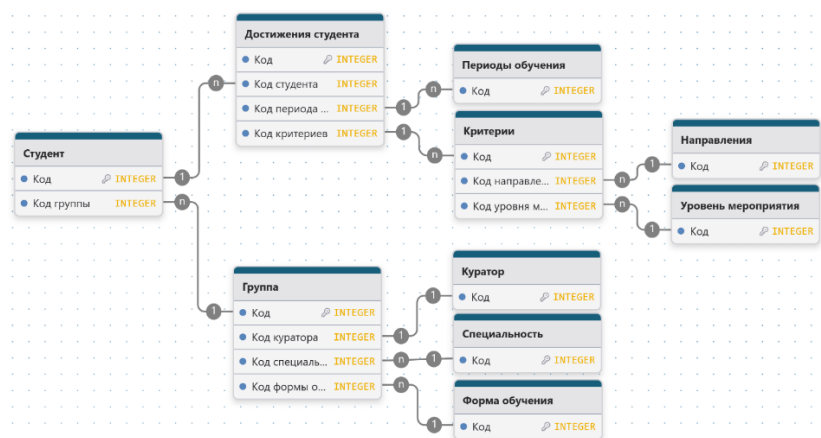


Рисунок 2 – Логическая структура базы данных

Рассмотрим описание логической модели базы данных в таблице 1.

Таблица 1 – Структура базы данных

Наименования таблицы	Описание таблицы
Студент	Представляет полную информацию о студенте
Группа	Включает в себя всю информации о группе студента
Куратор	Предоставляет информацию о кураторе
Специальность	Справочник, который хранит в себе специальности их сокращенное название и код
Форма обучения	Справочник, который хранит в себе список форм обучения
Достижения студента	Содержит информацию о достижения студента
Периоды обучения	Справочник, который хранит семестры и даты начала и окончания семестра
Критерии	Хранит критерии распределения баллов за достижение
Направление	Справочник, который хранит в себе направления достижений
Уровень мероприятия	Справочник, который хранит в себе уровни мероприятий

## 1.4 Физическое моделирование базы данных

**MySQL** – это открытая реляционная система управления базами данных (СУБД), представляющая собой надежное и производительное решение для хранения структурированных данных. Как часть стека технологий LAMP (Linux, Apache, MySQL, PHP/Python/Perl), MySQL широко используется в веб-разработке благодаря своей стабильности, высокой производительности и простоте интеграции с различными языками программирования.

**В контексте разрабатываемой системы "Личный кабинет студента"** MySQL была выбрана в качестве СУБД по следующим причинам:

- **Открытая лицензия (GPL)** позволяет использовать систему без дополнительных финансовых затрат
- **Полная поддержка SQL-стандарта** включает все необходимые функции: транзакции ACID, внешние ключи, хранимые процедуры, триггеры и представления (VIEW)
- **Высокая производительность** при работе с типичными операциями: частые SELECT-запросы для анализа данных, INSERT/UPDATE операций
- **Широкая экосистема инструментов** включает MySQL Workbench для визуального проектирования БД, драйверы для интеграции с Python (pymysql)
- **Надежность и отказоустойчивость** обеспечиваются механизмами репликации, резервного копирования и восстановления данных
- **Масштабируемость** позволяет системе расти вместе с увеличением количества пользователей (студентов) без значительных архитектурных изменений

Определим свойства полей для каждой таблицы. Тип данных в поле выбирается, исходя из характера данных, которые предполагается в нем хранить. Будем использовать типы данных, принятые в системе управления базами данных MySQL. Необходимо учесть, что поле, являющееся первичным

ключом таблицы, не может содержать значений, повторяющихся в двух или более строках таблицы.

1) Таблицы

А) Таблица «Студент». [5]

Типы данных для полей таблицы «Студент» представлены в таблице 2.

Таблица 2 – Физическая модель таблицы «Студент» [5]

Имя поля в таблице	Тип данных	Описание
id	INT PRIMARY KEY	Уникальный идентификатор студента
surname	VARCHAR(50)	Фамилия студента
first_name	VARCHAR(50)	Имя студента
middle_name	VARCHAR(50)	Отчество студента
birth_date	DATE	Дата рождения
student_group_id	INT	Внешний ключ на таблицу «Студенческая группа»
gradebook_name	VARCHAR(20)	Номер зачетной книжки
phone	VARCHAR(20)	Номер телефона
email	VARCHAR(100)	Адрес электронной почты
address	TEXT	Адрес фактического проживания
student_password	VARCHAR(255)	Пароль профиля студента
profile_photo	VARCHAR(255)	Адрес на фото студента
create_at	TIMESTAMP	Дата и время создания профиля
update_at	TIMESTAMP	Дата и время обновления профиля

Атрибут «id» (первичный ключ таблицы) однозначно определяет конкретного студента, а остальные атрибуты данного отношения индивидуальны относительно первичного ключа. Следовательно, таблица «Студент» содержит только одну функциональную зависимость от ключевого атрибута. Таким образом, таблица «Студент» удовлетворяет определению третьей нормальной формы, что исключает хранение избыточной информации и не требует дальнейшей декомпозиции.

Б) Таблица «Студенческая группа».

Типы данных для полей таблицы «Студенческая группа» представлены в таблице 3.

Таблица 3 – Физическая модель таблицы «Студенческая группа»

Имя поля в таблице	Тип данных	Описание
id	INT PRIMARY KEY	Уникальный идентификатор студенческой группы

naming	VARCHAR(20)	Название
specialty_id	INT	Внешний ключ на таблицу «Специальность»
form_study_id	INT	Внешний ключ на таблицу «Форма обучения»
tutor_id	INT	Внешний ключ на таблицу «Куратор»
course	INT	Номер курса
created_at	TIMESTAMP	Дата создания записи

Атрибут «id» (первичный ключ таблицы) однозначно определяет конкретную студенческую группу. Все остальные атрибуты находятся в полной функциональной зависимости от первичного ключа. Ссылочные атрибуты (specialty\_id, form\_study\_id, tutor\_id) обеспечивают связь с нормализованными справочниками, что соответствует принципам реляционной модели данных.

В таблице отсутствуют транзитивные зависимости — все неключевые атрибуты зависят только от первичного ключа, а не от других неключевых атрибутов. Связь со справочными таблицами реализована через внешние ключи, что обеспечивает ссылочную целостность без дублирования информации.

Таким образом, таблица «Студенческая группа» удовлетворяет определению третьей нормальной формы (3NF), что исключает хранение избыточной информации, аномалии вставки, обновления и удаления, и не требует дальнейшей декомпозиции.

В) Таблица «Специальность».

Типы данных для полей таблицы «Специальность» представлены в таблице 4.

Таблица 4 – Физическая модель таблицы «Специальность»

Имя поля в таблице	Тип данных	Описание
id	INT PRIMARY KEY	Уникальный идентификатор специальности
title	VARCHAR(50)	Полное название специальности
cut	VARCHAR(50)	Сокращенное название
cod_specialty	VARCHAR(20)	Код специальности

Атрибут «id» (первичный ключ таблицы) однозначно определяет конкретную специальность. Все остальные атрибуты находятся в полной функциональной зависимости от первичного ключа.

Таблица является справочником (reference table или lookup table) и содержит нормализованные данные о специальностях, предлагаемых образовательным учреждением. Уникальные ограничения на поля title и cut гарантируют отсутствие дублирования названий специальностей и их сокращений.

В таблице отсутствуют транзитивные зависимости — каждый неключевой атрибут напрямую зависит только от первичного ключа. Атрибут cod\_specialty содержит официальный код специальности согласно Федеральным государственным образовательным стандартам (ФГОС), что обеспечивает соответствие нормативным требованиям.

Таким образом, таблица «Специальность» удовлетворяет определению третьей нормальной формы (3NF), что исключает хранение избыточной информации и аномалии модификации данных. Таблица не требует дальнейшей декомпозиции и выполняет функцию нормализованного справочника в структуре базы данных.

#### Г) Таблица «Куратор»

Типы данных для полей таблицы «Куратор» представлены в таблице 5.

Таблица 5 – Физическая модель таблицы «Куратор»

Имя поля в таблице	Тип данных	Описание
id	INT	Уникальный идентификатор куратора
surname	VARCHAR(50)	Фамилия куратора
first_name	VARCHAR(50)	Имя куратора
middle_name	VARCHAR(50)	Отчество куратора (может отсутствовать)

Атрибут «id» (первичный ключ таблицы) однозначно идентифицирует каждого куратора. Все остальные атрибуты находятся в полной функциональной зависимости от первичного ключа.

Таблица является справочником и содержит нормализованные данные о преподавателях или административных сотрудниках, выполняющих функции

кураторов студенческих групп. Обязательные поля surname и first\_name гарантируют наличие базовой идентификационной информации, в то время как поле middle\_name является опциональным для учета полного ФИО.

В таблице отсутствуют транзитивные зависимости — каждый неключевой атрибут (фамилия, имя, отчество) напрямую зависит только от первичного ключа и не зависит от других не ключевых атрибутов. Таблица не содержит избыточных данных и обеспечивает уникальность записей о каждом кураторе.

Таким образом, таблица «Куратор» удовлетворяет определению третьей нормальной формы, что исключает аномалии вставки, обновления и удаления данных. Таблица не требует дальнейшей декомпозиции и служит нормализованным справочником в структуре базы данных.

#### Д) Таблица «Форма обучения»

Типы данных для полей таблицы «Форма обучения» представлены в таблице 6.

Таблица 6 – Физическая модель таблицы «Форма обучения»

Имя поля в таблице	Тип данных	Описание
id	INT PRIMARY KEY	Уникальный идентификатор формы обучения
title	VARCHAR(50)	Название формы обучения

Атрибут «id» (первичный ключ таблицы) однозначно идентифицирует каждую форму обучения. Атрибут title находится в полной функциональной зависимости от первичного ключа и содержит наименование формы обучения согласно классификации образовательных программ.

Таблица является нормализованным справочником и служит для стандартизации перечня форм обучения, предлагаемых образовательным учреждением. Уникальное ограничение на поле title гарантирует отсутствие дублирования названий форм обучения в системе.

В таблице присутствует единственная функциональная зависимость: id title. Отсутствуют транзитивные и частичные зависимости, так как таблица



содержит только первичный ключ и один неключевой атрибут. Минимальная структура таблицы обеспечивает максимальную нормализацию.

Таким образом, таблица «Форма обучения» удовлетворяет определению третьей нормальной формы.

#### Е) Таблица «Достижения студента» [13]

Типы данных для полей таблицы «Достижения студента» представлены в таблице 7.

Таблица 7 – Физическая модель таблицы «Достижения студента» [13]

Имя поля в таблице	Тип данных	Описание
id	INT PRIMARY KEY	Уникальный идентификатор достижений студента
student_id	INT	Внешний ключ на таблицу «Студент»
criteria_id	INT	Внешний ключ на таблицу «Критерии оценивания достижений»
period_id	INT	Внешний ключ на таблицу «Период»
quantity	INT	Количество повторений достижения
document_title	VARCHAR(255)	Название подтверждающего документа
document_pach	VARCHAR(255)	Путь к файлу документа
created_at	TIMESTAMP	Дата и время создания записи

Атрибут «id» (первичный ключ таблицы) однозначно определяет каждое достижение студента. Все остальные атрибуты находятся в полной функциональной зависимости от первичного ключа.

Таблица является основной операционной таблицей системы и содержит записи о конкретных достижениях студентов в различных видах деятельности. Внешние ключи student\_id, criteria\_id и period\_id обеспечивают связь с соответствующими справочниками, что соответствует принципам нормализованной реляционной базы данных.

В таблице отсутствуют транзитивные зависимости — каждый неключевой атрибут напрямую зависит только от первичного ключа. Атрибут quantity позволяет учитывать множественные участия в одном типе достижений, а поля document\_title и document\_path обеспечивают возможность прикрепления подтверждающих документов. Атрибут created\_at автоматически фиксирует дату добавления записи.

Таким образом, таблица «Достижения студента» удовлетворяет определению третьей нормальной формы, что исключает хранение избыточной информации и аномалии модификации данных. Таблица не требует дальнейшей декомпозиции и служит для оперативного учета достижений студентов в рамках системы рейтинговой оценки.

#### Ж) Таблица «Период»

Типы данных для полей таблицы «Период» представлены в таблице 8.

Таблица 8 – Физическая модель таблицы «Период»

Имя поля в таблице	Тип данных	Описание
id	INT PRIMARY KEY	Уникальный идентификатор периода
naming	INT	Название семестра
start_date	DATE	Начало семестра
end_date	DATE	Окончание семестра

Атрибут «id» (первичный ключ таблицы) однозначно определяет каждый академический период. Все остальные атрибуты находятся в полной функциональной зависимости от первичного ключа.

Таблица является справочником временных периодов и служит для учета достижений студентов в рамках конкретных учебных семестров. Атрибут naming содержит описательное название периода, а атрибуты start\_date и end\_date определяют временные границы семестра, что позволяет проводить анализ достижений в заданных временных интервалах.

В таблице отсутствуют транзитивные зависимости — каждый неключевой атрибут напрямую зависит только от первичного ключа. Отсутствуют также частичные зависимости, поскольку все атрибуты функционально полно зависят от первичного ключа.

Таким образом, таблица «Период» удовлетворяет определению третьей нормальной формы, что исключает хранение избыточной информации и аномалии модификации данных. Таблица не требует дальнейшей декомпозиции и служит для временной привязки достижений студентов в системе рейтинговой оценки.

Ж) Таблица «Критерии оценивания достижений» (см. Приложения Е, Ж, 3)

Типы данных для полей таблицы «Критерии оценивания достижений» представлены в таблице 9.

Таблица 9 – Физическая модель таблицы «Критерии оценивания достижений» (см. Приложения Е, Ж, 3)

Имя поля в таблице	Тип данных	Описание
id	INT PRIMARY KEY	Уникальный идентификатор критерий оценивания достижений
category_id	INT	Внешний ключ на таблицу «Направления достижений»
section_name	VARCHAR(255)	Название категории
description_text	VARCHAR(255)	Описание конкретное достижение
level_type_id	INT	Внешний ключ на таблицу «Уровень мероприятия»
points	INT	Количество баллов за достижение

Атрибут «id» (первичный ключ таблицы) однозначно определяет каждый критерий оценивания. Все остальные атрибуты находятся в полной функциональной зависимости от первичного ключа.

Таблица является нормализованным справочником правил оценивания и содержит детализированные критерии для начисления рейтинговых баллов за различные достижения студентов. Внешние ключи category\_id и level\_type\_id обеспечивают связь с соответствующими классификаторами, что позволяет структурировать критерии по направлениям деятельности и уровням мероприятий.

Атрибут section\_name определяет принадлежность критерия к конкретному разделу деятельности, description\_text содержит описание конкретного достижения, а points устанавливает количество баллов, начисляемое за выполнение данного критерия.

В таблице отсутствуют транзитивные зависимости — каждый неключевой атрибут напрямую зависит только от первичного ключа. Отсутствуют также частичные зависимости, поскольку все атрибуты функционально полно зависят от первичного ключа.

Таким образом, таблица «Критерии оценивания достижений» удовлетворяет определению третьей нормальной формы, что исключает хранение избыточной информации и аномалии модификации данных. Таблица не требует дальнейшей декомпозиции и служит основой для объективной оценки достижений студентов в рамках системы рейтинговой оценки.

### 3) Таблица «Направления достижений» (см. Приложения Е, Ж, З)

Типы данных для полей таблицы «Направления достижений» представлены в таблице 10.

Таблица 10 – Физическая модель таблицы «Направления достижений» (см. Приложения Е, Ж, З)

Имя поля в таблице	Тип данных	Описание
id	INT PRIMARY KEY	Уникальный идентификатор направления достижений
naming	INT	Название направления
cod	VARCHAR(255)	Системный код направления

Атрибут «id» (первичный ключ таблицы) однозначно определяет каждое направление достижений. Все остальные атрибуты находятся в полной функциональной зависимости от первичного ключа.

Таблица является справочником категорий деятельности и служит для классификации достижений студентов по основным направлениям. Атрибут naming содержит полное название направления деятельности, а атрибут cod — краткий системный код, используемый для программной идентификации направления в интерфейсе и бизнес-логике приложения.

В таблице отсутствуют транзитивные зависимости — каждый неключевой атрибут напрямую зависит только от первичного ключа. Отсутствуют также частичные зависимости, поскольку все атрибуты функционально полно зависят от первичного ключа.

Таким образом, таблица «Направления достижений» удовлетворяет определению третьей нормальной формы, что исключает хранение избыточной информации и аномалии модификации данных. Таблица не требует дальнейшей декомпозиции и служит для структурированного

разделения достижений студентов по видам деятельности в рамках системы рейтинговой оценки.

И) Таблица «Уровень мероприятия» (см. Приложения Е, Ж, З)

Типы данных для полей таблицы «Уровень мероприятия» представлены в таблице 10.

Таблица 10 – Физическая модель таблицы «Уровень мероприятия» (см. Приложения Е, Ж, З)

Имя поля в таблице	Тип данных	Описание
id	INT PRIMARY KEY	Уникальный идентификатор уровня
naming	INT	Название уровня мероприятия

Атрибут «id» (первичный ключ таблицы) однозначно определяет каждый уровень мероприятия. Атрибут title находится в полной функциональной зависимости от первичного ключа и содержит наименование уровня проведения мероприятия.

Таблица является нормализованным справочником и служит для классификации мероприятий по их масштабу и значимости. Примеры уровней мероприятий включают: университетский, городской, региональный, федеральный и международный. Использование отдельной справочной таблицы для уровней мероприятий позволяет стандартизировать классификацию и обеспечить консистентность данных в системе.

В таблице присутствует единственная функциональная зависимость: id title. Отсутствуют транзитивные и частичные зависимости, так как таблица содержит только первичный ключ и один неключевой атрибут. Минимальная структура таблицы обеспечивает максимальную нормализацию.

Таким образом, таблица «Уровень мероприятия» удовлетворяет определению третьей нормальной формы, что исключает хранение избыточной информации и аномалии модификации данных. Таблица не требует дальнейшей декомпозиции и служит для классификации мероприятий по уровню их проведения в рамках системы рейтинговой оценки.

## **2) Связи между таблицами**

Имена таблиц расположены в порядке указания направления связи. Например, Специальность 1 – Студенческая группа Много, где тип связи «один ко многим», обозначается как 1..М.

**А) Зависимость «Специальность – Студенческая группа».** Связь можно описать выражением:

- «Каждый идентификатор студенческой группы должен иметь идентификатор специальности и при том только один».
- «Каждый идентификатор специальности может быть связан с различными студенческими группами».

Таким образом, связь таблиц «Специальность» и «Студенческая группа» имеет мощность один – ко – многим и является обязательной.

**Б) Зависимость «Форма обучения – Студенческая группа».** Связь можно описать выражением:

- «Каждый идентификатор студенческой группы должен иметь идентификатор формы обучения и при том только один».
- «Каждый идентификатор формы обучения может быть связан с различными студенческими группами».

Таким образом, связь таблиц «Форма обучения» и «Студенческая группа» имеет мощность один – ко – многим и является обязательной.

**В) Зависимость «Куратор – Студенческая группа».** Связь можно описать выражением:

- «Каждый идентификатор студенческой группы должен иметь идентификатор куратора и при том только один».
- «Каждый идентификатор куратора может быть связан с одной студенческой группой».

Таким образом, связь таблиц «Куратор» и «Студенческая группа» имеет мощность один – к – одному и является обязательной.

**Г) Зависимость «Студенческая группа – Студент».** Связь можно описать выражением:

- «Каждый идентификатор студента должен иметь идентификатор группы и при том только один».

- «Каждый идентификатор студенческой группы может быть связан с различными студентами».

Таким образом, связь таблиц «Студенческая группа» и «Студент» имеет мощность один – ко – многим и является обязательной.

**Д) Зависимость «Направления достижений – Критерии оценивания достижений».** Связь можно описать выражением:

- «Каждый идентификатор критерия должен иметь идентификатор направления (категории) и при том только один».

- «Каждый идентификатор направления может быть связан с различными критериями оценивания».

Таким образом, связь таблиц «Направления достижений» и «Критерии оценивания достижений» имеет мощность один – ко – многим и является обязательной.

**Е) Зависимость «Уровень мероприятия – Критерии оценивания достижений».** Связь можно описать выражением:

- «Каждый идентификатор критерия должен иметь идентификатор уровня мероприятия и при том только один».

- «Каждый идентификатор уровня мероприятия может быть связан с различными критериями оценивания».

Таким образом, связь таблиц «Уровень мероприятия» и «Критерии оценивания достижений» имеет мощность один – ко – многим и является обязательной.

**Ж) Зависимость «Студент – Достижения студента».** Связь можно описать выражением:

- «Каждый идентификатор достижения должен иметь идентификатор студента и при том только один».

- «Каждый идентификатор студента может быть связан с различными записями о достижениях».

Таким образом, связь таблиц «Студент» и «Достижения студента» имеет мощность один – ко – многим и является обязательной.

**З) Зависимость «Критерии оценивания достижений – Достижения студента».** Связь можно описать выражением:

- «Каждый идентификатор достижения должен иметь идентификатор критерия оценивания и при том только один».
- «Каждый идентификатор критерия может быть связан с различными записями о достижениях студентов».

Таким образом, связь таблиц «Критерии оценивания достижений» и «Достижения студента» имеет мощность один – ко – многим и является обязательной.

**И) Зависимость «Период – Достижения студента».** Связь можно описать выражением:

- «Каждый идентификатор достижения должен иметь идентификатор периода и при том только один».
- «Каждый идентификатор периода может быть связан с различными записями о достижениях».

Таким образом, связь таблиц «Период» и «Достижения студента» имеет мощность один – ко – многим и является обязательной.

Физическое проектирование — создание схемы базы данных для конкретной СУБД с учетом ее специфики. Результатом физического проектирования логической схемы может быть скрипт на языке SQL.

Перейдем к физическому проектированию базы данных. Типовой пример реализации создания таблиц и отображения содержимого таблицы в базе данных представлен на рисунке 3





## 2 Функциональные возможности

Вопросы, обозначенные при анализе предметной области, преобразуются в функциональные элементы (функции базы данных), являющиеся запросами, процедурами, триггерами. Приведём основные типовые запросы, полный перечень реализован в коде приложения.

### 2.1 Запросы на редактирование

Запросы SQL к базе данных используют синтаксические особенности языка программирования python, что позволяет использовать динамические обращения к полям базы данных в зависимости от собираемых данных.

#### 2.1.1 Запрос на добавление записи в таблицу

Рассмотрим запрос на вставку данных в основную операционную таблицу системы. Данный запрос напрямую взаимодействует с разработанным триггером, который проверяет уникальность поля `document_title`. Запрос добавление новой записи о достижении в таблицу `Student_Achievement`. Запрос представлен на рисунке 5.

```
sql = """
INSERT INTO Student_Achievement
(student_id, criteria_id, period_id, quantity, document_title, created_at)
VALUES (%s, %s, %s, %s, %s, NOW())
"""
```

Рисунок 5 – Запрос на добавление достижения

#### 2.1.2 Запрос на изменение данных таблицы

Рассмотрим типовой пример запрос, используемый для изменения персональной контактной информации студента. Перед выполнением UPDATE приложение выполняет проверки на уникальность нового email и phone среди других записей в таблице `Student` для поддержания целостности данных. Запрос представлен на рисунке 6.

```
# Обновляем данные
update_sql = """
UPDATE Student
SET email = %s, phone = %s, address = %s
WHERE id = %s
"""
```

Рисунок 6 – Запрос на изменения контактной информации

По схожему принципу с предварительными проверками построены запросы обновления пароля и фотографии профиля

### 2.1.3 Запрос на удаление данных из таблицы

Рассмотрим типовой пример запроса удаления достижения. Запрос выполняет удаление записи из таблицы Student\_Achievement по ее уникальному идентификатору (id). Запрос представлен на рисунке 7.

```
delete_sql = """
DELETE
FROM Student_Achievement
WHERE id = %s"""
```

Рисунок 7 – Запрос на удаление достижения

## 2.2 Запросы на выборку данныхЗапрос на фильтрацию данных из таблицы

Рассмотрим пример сложного запроса на выборку данных, который имеет динамическую структуру и используется для построения детализированного портфолио студента. Данный запрос позволяет пользователю гибко настраивать вывод данных, фильтруя их по академическому периоду и категории деятельности. Формирование списка всех достижений студента, детализированных по критериям, с возможностью

динамической фильтрации по идентификатору периода (period\_id) и коду категории (category\_cod), используя таблицы Student\_Achievement, Rating\_Criteria, Activity\_Category, level\_type. Запрос представлен на рисунке 8.

```
# Базовый SQL запрос
sql = """
    SELECT sa.id, sa.quantity, sa.created_at, sa.document_title,
           rc.description_text, rc.points,
           ac.naming as category_name, ac.cod as category_cod,
           lt.title as level_title
    FROM Student_Achievement sa
    JOIN Rating_Criteria rc ON sa.criteria_id = rc.id
    JOIN Activity_Category ac ON rc.category_id = ac.id
    JOIN level_type lt ON rc.level_type_id = lt.id
    WHERE sa.student_id = %s
"""
params = [student_id]

# Динамически добавляем условия фильтрации
if period_id:
    sql += " AND sa.period_id = %s"
    params.append(period_id)

if category_cod:
    sql += " AND ac.cod = %s"
    params.append(category_cod)

sql += " ORDER BY sa.created_at DESC"
```

Рисунок 8 – Запрос на фильтрацию данных

В отличие от статических запросов, этот запрос собирается в коде приложения. Приложение сначала формирует базовую часть SELECT...FROM...WHERE student\_id = %s. Затем, если пользователь выбрал фильтр по периоду или категории, к строке запроса добавляются соответствующие условия AND sa.period\_id = %s или AND ac.cod = %s, а также добавляются соответствующие значения в список параметров. Это позволяет

избежать отправки в базу данных лишних данных и обеспечить высокую скорость отклика при фильтрации больших объемов информации.

### 2.2.2 Запрос с использованием агрегатных функций

Для формирования рейтинговой таблицы и сводки портфолио необходимо выполнять вычисления на стороне базы данных. Рассмотрим запрос, который вычисляет суммарное количество баллов студента по определенной категории деятельности, а также количество загруженных достижений. Запрос использует функции агрегации SUM() для подсчета баллов и COUNT() для подсчета количества записей, а также группирует данные с учетом текущего академического периода. Запрос представлен на рисунке 9.

```
points_sql = """
SELECT SUM(rc.points * sa.quantity) as category_total
FROM Student_Achievement sa
JOIN Rating_Criteria rc ON sa.criteria_id = rc.id
JOIN Academic_Period ap ON sa.period_id = ap.id
WHERE sa.student_id = %s AND rc.category_id = %s
AND ap.end_date >= CURDATE()
"""
```

Рисунок 9 – Запрос на формирование рейтинга

### 2.2.3 Запрос с объединением нескольких таблиц (JOIN)

Для отображения полных данных профиля студента и авторизации в системе используется сложный запрос на выборку, объединяющий данные из пяти таблиц: Student, Student\_group, Specialty, Form\_study и tutor. Это необходимо для вывода человеко-читаемых названий группы, специальности и ФИО куратора вместо их числовых идентификаторов. Запрос представлен на рисунке 10.

```

sql = """
SELECT s.*,
       DATE_FORMAT(s.birth_date, '%%d.%%m.%%Y') as birth_date_str,
       sg.naming as group_name, sg.course,
       sp.title as specialty_name, sp.cod_specialty,
       fs.title as form_study_name,
       t.surname as tutor_surname, t.first_name as tutor_first_name,
       t.middle_name as tutor_middle_name,
       YEAR(s.created_at) as created_year,
       YEAR(DATE_ADD(s.created_at, INTERVAL 4 YEAR)) as graduation_year
FROM Student s
JOIN Student_group sg ON s.student_group_id = sg.id
JOIN Specialty sp ON sg.specialty_id = sp.id
JOIN Form_study fs ON sg.form_study_id = fs.id
JOIN tutor t ON sg.tutor_id = t.id
WHERE s.id = %s
"""

```

Рисунок 10 – Запрос на формирование данных профиля

#### 2.2.4 Запрос на проверку уникальности данных

Перед регистрацией изменений в профиле система выполняет проверку на отсутствие дубликатов ключевых полей (email, телефон), чтобы исключить конфликты данных. Используется запрос на выборку с условием исключения текущего пользователя. Запрос представлен на рисунке 11.

```

check_sql = "SELECT id FROM Student WHERE email = %s AND id != %s"

```

Рисунок 11 – Запрос на формирование данных профиля

### 2.3 Триггеры

Данный триггер срабатывает перед добавлением новой записи в таблицу студенческих достижений (Student\_Achievement). Его задача — проверить, чтобы значение в поле quantity (количество достижений) не было отрицательным. Если при вставке указывается значение меньше нуля, триггер предотвращает операцию и возвращает пользовательскую ошибку с соответствующим сообщением. Это позволяет избежать некорректных

данных в системе учёта достижений. На рисунке 12 представлена блок-схема триггера, код триггера представлен в листинге 13.

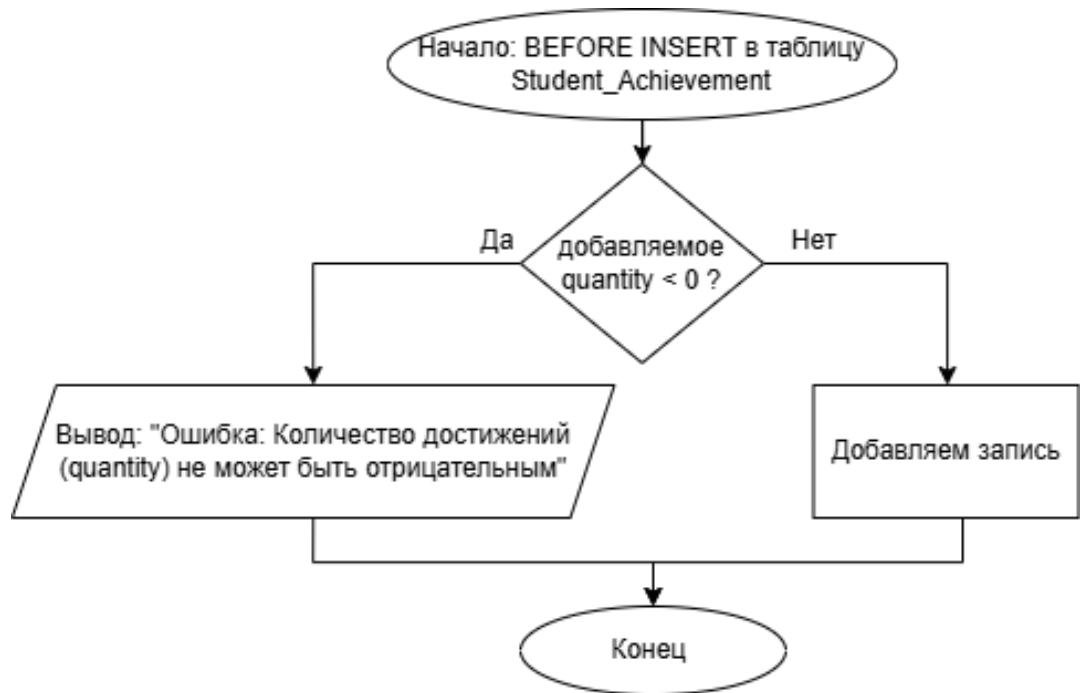


Рисунок 12 – Блок-схема триггера

```

DELIMITER //

-- Триггер для предотвращения вставки отрицательного числа
CREATE TRIGGER check_quantity_before_insert
BEFORE INSERT ON Student_Achievement
FOR EACH ROW
BEGIN
    IF NEW.quantity < 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Ошибка: Количество достижений (quantity) не может быть отрицательным.';
    END IF;
END//

DELIMITER ;
  
```

Рисунок 13 – Триггер проверки входного числа в поле quantity

## 2.4 Процедуры

Рассмотрим процедуру, которая принимает идентификатор студента и периода, определяет курс обучающегося и формирует сводную статистику по направлениям деятельности. Для каждой категории вычисляется сумма баллов

студента, его место в рейтинге среди однокурсников и общее количество участников. На рисунке 14 представлена блок-схема процедуры, код процедуры представлен в листинге 1.



Рисунок 14 – Блок-схема процедуры



Листинг 1 — код процедуры GetCourseRatingAnalysis

```
CREATE PROCEDURE GetCourseRatingAnalysis(  
    IN studentId INT,  
    IN periodId INT  
)  
BEGIN  
    DECLARE my_course INT;  
  
    -- 1. Узнаем курс студента  
    SELECT sg.course INTO my_course  
    FROM Student s  
    JOIN Student_group sg ON s.student_group_id = sg.id  
    WHERE s.id = studentId;  
  
    -- 2. Выводим статистику по каждой категории  
    SELECT  
        ac.naming AS category_name,  
        ac.cod AS category_cod,  
  
        -- Мои баллы в этой категории  
        (SELECT COALESCE(SUM(rc.points * sa.quantity), 0)  
         FROM Student_Achievement sa  
         JOIN Rating_Criteria rc ON sa.criteria_id = rc.id  
         WHERE sa.student_id = studentId  
         AND sa.period_id = periodId  
         AND rc.category_id = ac.id  
        ) as my_points,  
  
        -- Мое место на курсе (считаем, у скольких людей баллов БОЛЬШЕ, чем  
        у меня)
```

(SELECT COUNT(\*) + 1

FROM (

SELECT s\_sub.id, SUM(rc\_sub.points \* sa\_sub.quantity) as total\_p

FROM Student s\_sub

JOIN Student\_group sg\_sub ON s\_sub.student\_group\_id = sg\_sub.id

JOIN Student\_Achievement sa\_sub ON s\_sub.id = sa\_sub.student\_id

JOIN Rating\_Criteria rc\_sub ON sa\_sub.criteria\_id = rc\_sub.id

WHERE sg\_sub.course = my\_course -- Тот же курс

AND sa\_sub.period\_id = periodId

AND rc\_sub.category\_id = ac.id -- Та же категория

GROUP BY s\_sub.id

) as competitors

WHERE competitors.total\_p >

(SELECT COALESCE(SUM(rc2.points \* sa2.quantity), 0)

FROM Student\_Achievement sa2

JOIN Rating\_Criteria rc2 ON sa2.criteria\_id = rc2.id

WHERE sa2.student\_id = studentId

AND sa2.period\_id = periodId

AND rc2.category\_id = ac.id)

) as my\_rank,

-- Всего участников в этой категории на курсе (конкуренция)

(SELECT COUNT(DISTINCT sa\_sub.student\_id)

FROM Student\_Achievement sa\_sub

JOIN Student s\_sub ON sa\_sub.student\_id = s\_sub.id

JOIN Student\_group sg\_sub ON s\_sub.student\_group\_id = sg\_sub.id

JOIN Rating\_Criteria rc\_sub ON sa\_sub.criteria\_id = rc\_sub.id

WHERE sg\_sub.course = my\_course

AND sa\_sub.period\_id = periodId

AND rc\_sub.category\_id = ac.id

) as total\_participants

FROM Activity\_Category ac;  
END//

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1 Российская Федерация. Законы. Об образовании в Российской Федерации [Электронный ресурс] : федеральный закон от 29.12.2012 № 273-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_140174/](https://www.consultant.ru/document/cons_doc_LAW_140174/). – (Дата обращения: 10.10.2025). – Ст. 30, ч. 2.

2 Российская Федерация. Законы. Об образовании в Российской Федерации [Электронный ресурс] : федеральный закон от 29.12.2012 № 273-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_140174/](https://www.consultant.ru/document/cons_doc_LAW_140174/). – (Дата обращения: 15.12.2025). – Ст. 28, ч. 3, п. 1, 11.

3 Российская Федерация. Законы. О персональных данных [Электронный ресурс] : федеральный закон от 27.07.2006 № 152-ФЗ (ред. от 24.04.2024) // КонсультантПлюс. – Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_61801/](https://www.consultant.ru/document/cons_doc_LAW_61801/). – (Дата обращения: 15.12.2025).

4 Министерство науки и высшего образования Российской Федерации. Об утверждении порядка приема на обучение по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры [Электронный ресурс] : приказ от 27.11.2024 № 821 // КонтурНорматив. – Режим доступа: <https://normativ.kontur.ru/document?moduleId=1&documentId=503133> (Дата обращения: 20.10.2025).

5 Министерство обороны Российской Федерации. Об определении Порядка обучения граждан Российской Федерации в военных образовательных организациях высшего образования по программе военной подготовки сержантов, старшин запаса либо по программе военной подготовки солдат, матросов запаса [Электронный ресурс] : приказ от 30.11.2018 № 693 (с изм. от 21.04.2023) // Гарант. – Режим доступа:

<https://base.garant.ru/72131136/>. – (Дата обращения: 15.12.2025). – Прил.: Порядок обучения... Прил. 1: Личная карточка студента (рекомендуемый образец).

6 Росархив. Об утверждении Перечня типовых управленческих архивных документов, образующихся в процессе деятельности государственных органов, органов местного самоуправления и организаций, с указанием сроков хранения [Электронный ресурс] : приказ от 20.12.2019 № 236 // КонсультантПлюс. – Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_345020/](https://www.consultant.ru/document/cons_doc_LAW_345020/). – (Дата обращения: 15.12.2025).

7 Рособрнадзор. Об утверждении Требований к структуре официального сайта образовательной организации в информационно-телекоммуникационной сети «Интернет» и формату представления на нем информации [Электронный ресурс] : приказ от 04.08.2023 № 1493 // КонсультантПлюс. – Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_463082/](https://www.consultant.ru/document/cons_doc_LAW_463082/) – (Дата обращения: 20.10.2025).

8 ГОСТ Р 59382-2021. Информационные технологии. Методы и средства обеспечения безопасности. Основы управления идентичностью [Электронный ресурс]. – Режим доступа: <https://base.garant.ru/403448112/> – (Дата обращения: 20.10.2025).

9 ГОСТ Р 52872-2019. Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Применение для инвалидов по зрению. Требования доступности [Электронный ресурс]. – Режим доступа: <https://base.garant.ru/73664694/> – (Дата обращения: 15.12.2025).

10 ГОСТ Р ИСО 9241-210-2016. Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем [Электронный ресурс]. – Режим доступа: <https://internet-law.ru/gosts/gost/63110/>. – (Дата обращения: 15.12.2025).

11 Образец заявления на предоставление места в общежитии : приложение к Правилам внутреннего распорядка / Приволжский политехнический колледж ФГБОУ ВО СГТУ имени Гагарина Ю.А. – 2022. – URL: <https://ppk.sstu.ru/wp-content/uploads/2022/11/%D0%9E%D0%B1%D1%80%D0%B0%D0%B7%D1%86%D1%8B-%D0%B7%D0%B0%D1%8F%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9.pdf> (дата обращения: 10.11.2025).

12 Положение о ведении журнала учебных занятий Энгельсского технологического института (филиала) СГТУ им. Гагарина Ю.А. : [утв. директором ЭТИ (филиал) СГТУ А.В. Яковлевым в 2017 г.]. – Энгельс : ЭТИ (филиал) СГТУ им. Гагарина Ю.А., 2017. – URL: [http://techn.sstu.ru/Documentation/SPO/Doc/%D0%9F%D0%BE%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5\\_%D0%AD%D0%A2%D0%98\\_%D0%BE\\_%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B8-%D0%B6%D1%83%D1%80%D0%BD%D0%B0%D0%BB%D0%B0-%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D1%8B%D1%85-%D0%B7%D0%B0%D0%BD%D1%8F%D1%82%D0%B8%D0%B9.doc](http://techn.sstu.ru/Documentation/SPO/Doc/%D0%9F%D0%BE%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D0%AD%D0%A2%D0%98_%D0%BE_%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B8-%D0%B6%D1%83%D1%80%D0%BD%D0%B0%D0%BB%D0%B0-%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D1%8B%D1%85-%D0%B7%D0%B0%D0%BD%D1%8F%D1%82%D0%B8%D0%B9.doc) (дата обращения: 10.11.2025).

13 Приказ Федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Саратовский государственный технический университет имени Гагарина Ю.А.» от 29.05.2015 № 584-П «О формировании электронного портфолио обучающихся».