МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**по дисциплине: «Безопасность систем баз данных»**

|  |  |
| --- | --- |
| Выполнил:  Студент гр. «АБ-320», «АВТФ»  *Сычук Алексей Александрович*  «30» сентября 2025г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Проверил:  Ассистент кафедры ЗИ  *Питько Яна Андреевна*  «\_\_\_» \_\_\_\_\_\_ 2025г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Новосибирск 2025

**Цель работы:** спроектировать доменную схему, классифицировать данные и настроить доступ с разделением обязанностей.

**Тема:** «Защита БД образовательных учреждений – хранение оценок, личных дел студентов и преподавателей».

**Задание 1.** ER-модель и классификация.

**Задание 2.** DDL и схемы.

**Задание 3.** Роли и привилегии.

**Задание 4.** Логирование подключений пользователей.

**Задание 1**

В первую очередь были определены сущности, необходимые для реализации базы данных по выбранной теме:

* educational\_institutions – хранит данные учебных заведений, содержит ссылку на ректора;
* faculties – содержит информацию о факультетах с привязкой к заведению и декану;
* departments – хранит данные кафедр с заведующими, организует преподавателей по подразделениям;
* study\_groups – определяет учебные группы с годом поступления, служит основой для организации учебного процесса;
* students – центральная таблица личных данных студентов (ФИО, контакты, статус);
* teachers – хранит личные данные преподавателей с учеными степенями и званиями, обеспечивает управление преподавательским составом;
* teacher\_departments – связывает преподавателей с кафедрами, разрешает множественную принадлежность с указанием основной должности;
* subjects – справочник учебных дисциплин;
* academic\_plans – определяет учебные планы, связывая группы с дисциплинами по семестрам с указанием часов и формы контроля;
* final\_grade\_types – справочник систем итогового оценивания (5-балльная, зачет/незачет), обеспечивает гибкость в оценках;
* final\_grades – хранит итоговые оценки студентов с привязкой к преподавателям, семестрам и типам оценок – ядро системы учета успеваемости;
* interim\_grades – фиксирует промежуточные оценки, позволяет отслеживать текущую успеваемость;
* student\_documents – хранит данные документов студентов (паспорта, аттестаты) с сериями, номерами и датами выдачи;
* class\_schedule – управляет расписанием занятий по неделям и дням с указанием аудиторий и корпусов;
* login\_log – хранит данные о входе пользователей в систему.

На основе определенных выше сущностей была составлена ER-диаграмма домена, где были уточнены атрибуты каждой отдельной сущности (Рисунок 1).

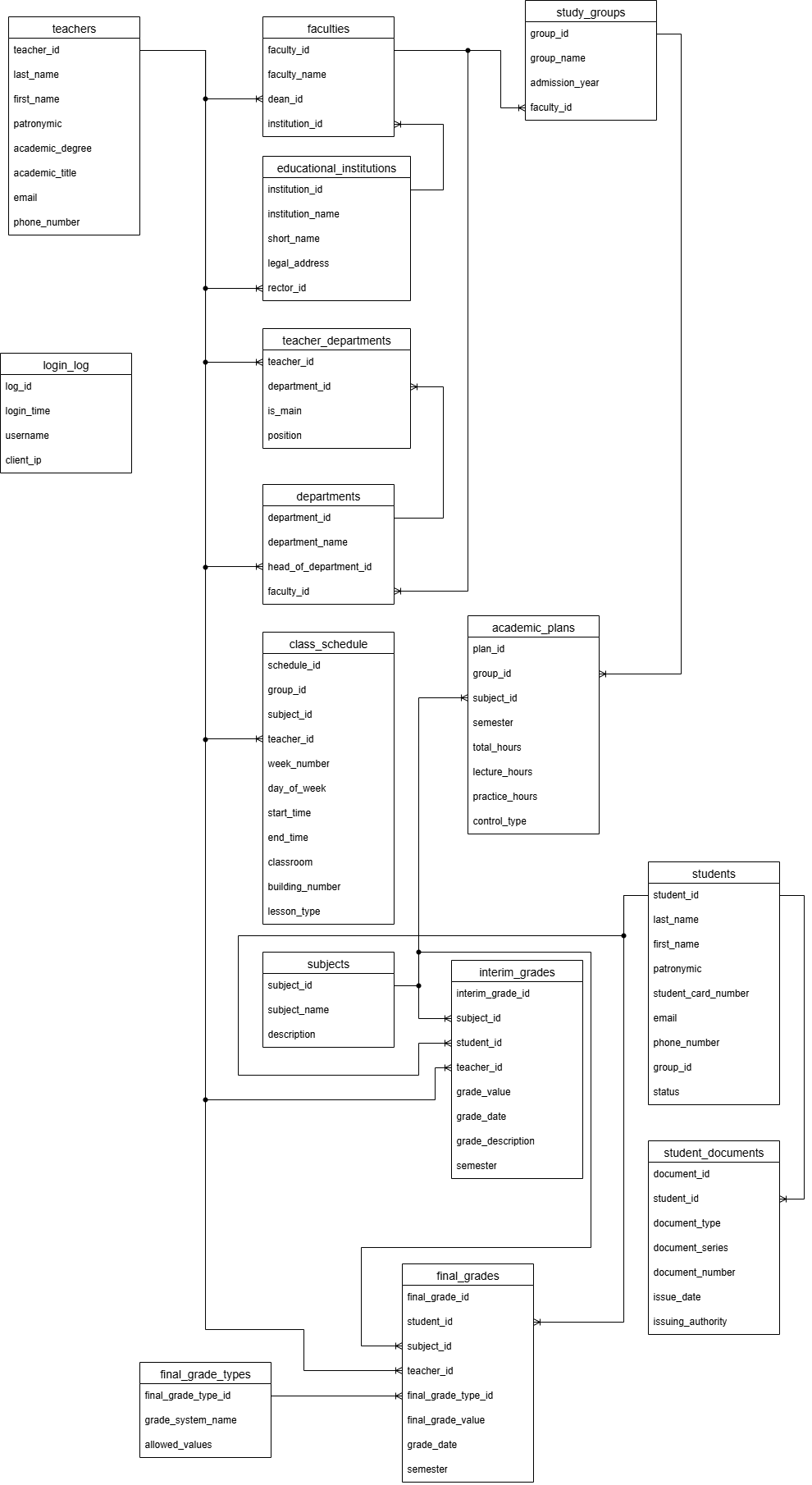


Рисунок 1 – ER-диаграмма домена

Далее будет произведена классификация данных на следующих уровнях:

1. Public – публичные данные;
2. Internal – внутренняя информация учреждения, которая не подлежит свободному распространению;
3. Confidential – данные доступные только ограниченному кругу лиц;
4. Restricted – конфиденциальные данные (в условиях темы – PII).

Классифицированные данные представлены в таблицах 1–15 в формате матриц вида «колонка → класс».

Таблица 1 – Классификация данных таблицы educational\_institutions

|  |  |
| --- | --- |
| Колонка | Класс |
| institution\_id | Internal |
| institution\_name | Public |
| short\_name | Public |
| legal\_address | Public |
| rector\_id | Internal |

Таблица 2 – Классификация данных таблицы faculties

|  |  |
| --- | --- |
| Колонка | Класс |
| faculty\_id | Internal |
| faculty\_name | Public |
| dean\_id | Internal |
| institution\_id | Internal |

Таблица 3 – Классификация данных таблицы departments

|  |  |
| --- | --- |
| Колонка | Класс |
| department\_id | Internal |
| department\_name | Public |
| head\_of\_department\_id | Internal |
| faculty\_id | Internal |

Таблица 4 – Классификация данных таблицы study\_groups

|  |  |
| --- | --- |
| Колонка | Класс |
| group\_id | Internal |
| group\_name | Public |
| admission\_year | Internal |
| faculty\_id | Internal |

Таблица 5 – Классификация данных таблицы students

|  |  |
| --- | --- |
| Колонка | Класс |
| student\_id | Internal |
| last\_name | Restricted |
| first\_name | Restricted |
| patronymic | Restricted |
| student\_card\_number | Confidential |
| email | Restricted |
| phone\_number | Restricted |
| group\_id | Internal |
| status | Internal |

Таблица 6 – Классификация данных таблицы teachers

|  |  |
| --- | --- |
| Колонка | Класс |
| teacher\_id | Internal |
| last\_name | Public |
| first\_name | Public |
| patronymic | Public |
| academic\_degree | Public |
| academic\_title | Public |
| email | Restricted |
| phone\_number | Restricted |

Таблица 7 – Классификация данных таблицы teacher\_departments

|  |  |
| --- | --- |
| Колонка | Класс |
| teacher\_id | Internal |
| department\_id | Internal |
| is\_main | Internal |
| position | Public |

Таблица 8 – Классификация данных таблицы subjects

|  |  |
| --- | --- |
| Колонка | Класс |
| subject\_id | Internal |
| subject\_name | Public |
| description | Public |

Таблица 9 – Классификация данных таблицы academic\_plans

|  |  |
| --- | --- |
| Колонка | Класс |
| plan\_id | Internal |
| group\_id | Internal |
| subject\_id | Internal |
| semester | Public |
| total\_hours | Public |
| lecture\_hours | Public |
| practice\_hours | Public |
| control\_type | Public |

Таблица 10 – Классификация данных таблицы final\_grade\_types

|  |  |
| --- | --- |
| Колонка | Класс |
| final\_grade\_type\_id | Internal |
| grade\_system\_name | Internal |
| allowed\_values | Internal |

Таблица 11 – Классификация данных таблицы final\_grades

|  |  |
| --- | --- |
| Колонка | Класс |
| final\_grade\_id | Internal |
| student\_id | Internal |
| subject\_id | Internal |
| teacher\_id | Internal |
| final\_grade\_type\_id | Internal |
| final\_grade\_value | Confidential |
| grade\_date | Confidential |
| semester | Confidential |

Таблица 12 – Классификация данных таблицы interim\_grades

|  |  |
| --- | --- |
| Колонка | Класс |
| interim\_grade\_id | Internal |
| student\_id | Internal |
| subject\_id | Internal |
| teacher\_id | Internal |
| grade\_value | Confidential |
| grade\_date | Confidential |
| grade\_description | Confidential |
| semester | Confidential |

Таблица 13 – Классификация данных таблицы student\_documents

|  |  |
| --- | --- |
| Колонка | Класс |
| document\_id | Internal |
| student\_id | Internal |
| document\_type | Restricted |
| document\_series | Restricted |
| document\_number | Restricted |
| issue\_date | Restricted |
| issuing\_authority | Restricted |

Таблица 14 – Классификация данных таблицы class\_schedule

|  |  |
| --- | --- |
| Колонка | Класс |
| schedule\_id | Internal |
| group\_id | Internal |
| subject\_id | Internal |
| teacher\_id | Internal |
| week\_number | Public |
| day\_of\_week | Public |
| start\_time | Public |
| end\_time | Public |
| classroom | Public |
| building\_number | Public |
| lesson\_type | Public |

Таблица 15 – Классификация данных таблицы login\_log

|  |  |
| --- | --- |
| Колонка | Класс |
| log\_id | Internal |
| login\_time | Confidential |
| username | Confidential |
| client\_ip | Restricted |

**Задание 2**

Все операции по созданию базы данных представлены по ссылке в Приложении А в файле «init.sql».

В базе данных «education\_db» были созданы следующие схемы:

1. app – основные бизнес-данные приложения, включает следующие таблицы:
   * students;
   * teachers;
   * final\_grades;
   * interim\_grades;
   * academic\_plans;
   * class\_schedule;
   * student\_documents;
   * teacher\_departments;
2. ref – справочники и классификаторы, включает следующие таблицы:
   * educational\_institutions;
   * faculties;
   * departments;
   * subjects;
   * final\_grade\_types;
   * study\_groups;
3. audit – данные аудита, включает следующие таблицы:
   * login\_log;
4. stg – временное хранение данных, однако на данный момент не используется.

В организации первичных ключей используется стандартный подход с SERIAL, за исключением таблицы teacher\_departments, где применен составной первичный ключ для связи многие-ко-многим.

Особое внимание уделено ограничениям целостности. UNIQUE-ограничения защищают критически важные данные: номера студенческих билетов, комбинации в учебных планах и уникальность групп с учетом года поступления. Check-ограничения обеспечивают валидацию семестров и учебных недель.

Набор индексов включает только необходимые: базовые для внешних ключей, а также составные индексы для рядовых ситуаций (построения расписания и академических отчетов по семестрам).

Поля осознанно сделаны nullable там, где это соответствует бизнес-логике – отчества для иностранных студентов, ученые звания и серии документов.

**Задание 3**

С операциями по созданию ролей и назначения им привилегий можно ознакомиться по ссылке, представленной в Приложении А в файле «init.sql».

Доступ схемы PUBLIC ко всем созданным объектам был запрещен, все привилегии назначались явно через механизм default privileges для обеспечения безопасности по умолчанию.

Созданы три основные бизнес-роли: app\_reader с правами только на чтение данных в схемах приложения и справочников, app\_writer с дополнительными правами на запись и использование последовательностей в бизнес-схемах, и app\_owner с полным доступом к схемам приложения включая права на создание объектов и управление триггерами.

Для аудита создана роль auditor с исключительным доступом на чтение к схеме audit. Реализовано требование о запрете записи напрямую в аудиторские таблицы – INSERT возможен только через триггеры.

Дополнительно созданы административные роли разделения обязанностей: ddl\_admin для управления структурой базы данных с правами на создание объектов во всех схемах, dml\_admin для операций с данными и security\_admin для управления безопасностью.

Роли security\_admin предоставлены расширенные привилегии включая права аудитора, что позволяет осуществлять мониторинг безопасности и проводить расследования инцидентов с доступом к полной истории действий в системе. Также security\_admin имеет права на управление ролями, просмотр системной информации и административные функции для обслуживания базы данных.

Исходный код сценариев проверки доступа представлен по ссылке в Приложении А в файле «test.sh».

Написанные тесты работают по принципу присваивания тестовому пользователю проверяемой роли и дальнейшая попытка выполнения различных команд.

Всего есть два возможных варианта выполнения теста: либо операция проходит успешно, либо пользователю отказывается в доступе. Если ожидаемый результат совпадает с действительным, то на экран выводится сообщение об успехе.

В конце тестирования происходит очистка тестовых данных для того, чтобы не оставлять в базе лишних записей.

Демонстрация работы сценариев проверки представлена на рисунке 2.

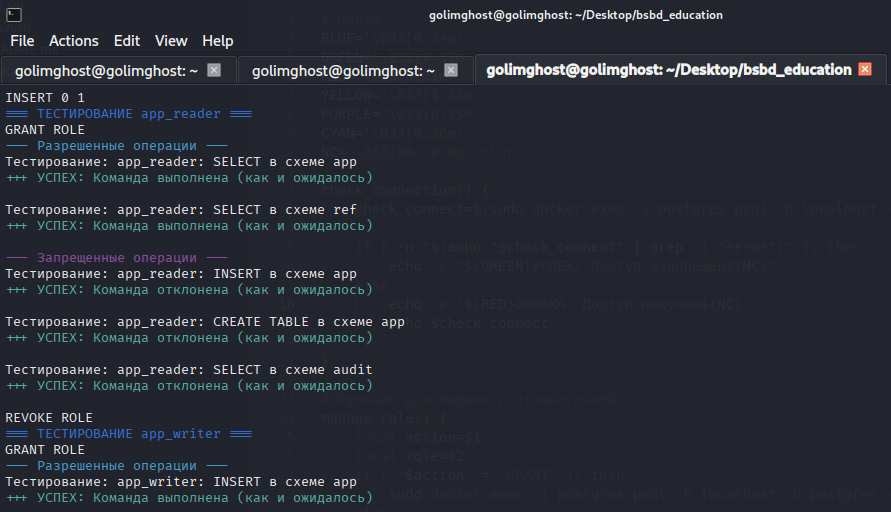


Рисунок 2 – Демонстрация сценариев проверки

**Задание 4**

Исходный код триггера представлен по ссылке в Приложении А в файле «init.sql».

В случае срабатывания триггера данные о подключившемся пользователе помещаются в таблицу audit.login\_log. Тестирование работы триггера также производится с использованием файла «test.sh». Результат работы теста, а также записи в таблице представлены на рисунках 3 и 4.

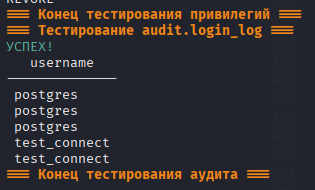


Рисунок 3 – Тестирование триггера

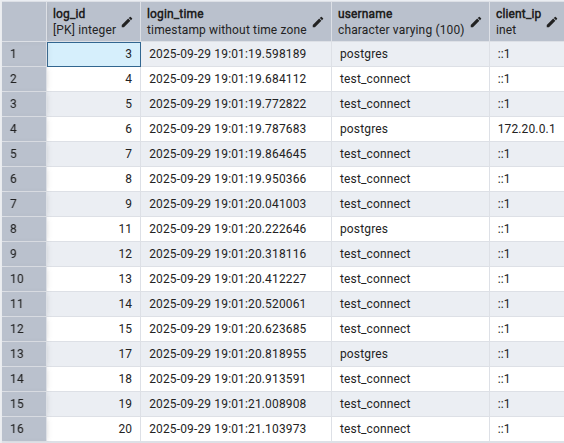


Рисунок 4 – Результат записи подключений в таблицу

**Вывод**

В ходе выполнения лабораторной работы были закреплены навыки проектирования баз данных по теме «Защита БД образовательных учреждений – хранение оценок, личных дел студентов и преподавателей».

Разработанная многоуровневая архитектура базы данных, основанная на принципе разделения схем, позволяет изолировать бизнес-логику, справочные данные и аудиторский след, что создает фундамент для управления доступом.

Классификация данных по уровням конфиденциальности является критически важным этапом, позволяющим формализовать политики разграничения доступа.

Создание административных ролей позволяет реализовать модель разделения обязанностей, что является ключевым механизмом предотвращения злоупотреблений. Механизм принудительного аудита через триггеры подключения обеспечивает постоянность регистрации действий пользователей, создавая основу для последующего анализа инцидентов безопасности.

Таким образом, можно сделать вывод о том, что совмещение организационных мер с техническими средствами защиты позволяет создать сбалансированную систему защиты информации, которая обеспечивает как конфиденциальность персональных данных, так и функциональность базы данных.

**Приложение А**

Ссылка‍ ‌на GitHub-репозиторий: https://github.com/golimghostpy/bsbd\_education.