

**Міністерство освіти і науки України**  
**Український державний університет науки і технологій**

Комп'ютерних технологій і систем  
(назва факультету)

Електронних обчислювальних машин  
(повна назва кафедри)

**Пояснювальна записка**  
до кваліфікаційної роботи  
перший(бакалаврський)  
(ступінь вищої освіти)

*до захисту*  
*2.06.2022*

на тему: Створення бази даних здобувачів спеціальності «Кібербезпека» та її захист

за освітньою програмою Кібербезпека

зі спеціальності: 125 Кібербезпека

(шифр і назва спеціальності)

Виконав: студент групи КБ 19120

Керівник: \_\_\_\_\_  
(підпис студента)

/ Неллі Падафа /  
(Ім'я ПРІЗВИЩЕ)

Керівник: \_\_\_\_\_  
(підпис)

/ доцент, Вікторія Пахомова /  
(посада, Ім'я ПРІЗВИЩЕ)

Нормоконтролер: \_\_\_\_\_  
(підпис)

/ старший викл., Володимир Дзюба /  
(посада, Ім'я ПРІЗВИЩЕ)

Консультанти:

_____	_____	/
(назва розділу)	(підпис)	(посада, Ім'я ПРІЗВИЩЕ)
_____	_____	/
(назва розділу)	(підпис)	(посада, Ім'я ПРІЗВИЩЕ)
_____	_____	/
(назва розділу)	(підпис)	(посада, Ім'я ПРІЗВИЩЕ)
_____	_____	/
(назва розділу)	(підпис)	(посада, Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент

\_\_\_\_\_ *СВ*  
(підпис)

Дніпро – 2022 рік

**Ministry of Education and Science of Ukraine**  
**Ukrainian State University of Science and Technologies**

Computer technology and systems  
(faculty)

Electronic computers  
(department)

**Explanatory Note**  
**to Bachelor's Thesis**  
**first (bachelor's)**  
(higher education degree)

on the topic: Creating a database of students majoring in "Cybersecurity" and its protection

according to educational curriculum Cybersecurity

in the Speciality: 125 Cybersecurity

(speciality and its code)

Done by the student of the group: CS 19120 / Nelly Padafa /  
(name, surname)

Scientific Supervisor:  / docent, Viktoria Pakhomova /  
(position, name, surname)

Normative controller :  / senior lecturer, Vladimir Dziuba /  
(position, name, surname)

Supervisors

\_\_\_\_\_  
(Chapter title heading)

/ \_\_\_\_\_ /  
(position, name, surname)

\_\_\_\_\_  
(Chapter title heading)

/ \_\_\_\_\_ /  
(position, name, surname)

\_\_\_\_\_  
(Chapter title heading)

/ \_\_\_\_\_ /  
(position, name, surname)

\_\_\_\_\_  
(Chapter title heading)

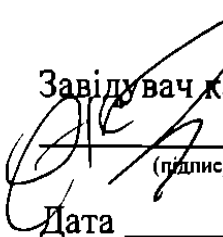
/ \_\_\_\_\_ /  
(position, name, surname)

Dnipro – 2022

**Міністерство освіти і науки України**  
**Український державний університет науки і технологій**

Факультет: Комп'ютерних технологій і систем  
Кафедра: Електронних обчислювальних машин  
Рівень вищої освіти: Перший (бакалаврський)  
Освітня програма: Кібербезпека  
Спеціальність: 125 Кібербезпека

(шифр та назва)

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри **ЕОМ**  
 **Ігор Жуковицький**  
(підпис) (Ім'я ПРІЗВИЩЕ)  
Дата \_\_\_\_\_

**ЗАВДАННЯ**

на кваліфікаційну роботу перший (бакалаврський)

(ступінь вищої освіти)

студенту Падафа Неллі Євгенівні

(Прізвище, Ім'я По батькові)

Тема  
роботи:

Створення бази даних здобувачів спеціальності «Кібербезпека» та її захист

Керівник роботи: Пахомова Вікторія Миколаївна, к.т.н., доцент кафедри ЕОМ

(Прізвище, Ім'я, По батькові, науковий ступінь, вчене звання)

затверджені наказом від "07" 12 2021 р. № 67ст

2. Строк подання студентом 13.06.2022 р.  
роботи:

3. Вихідні дані до роботи: дані про здобувачів першого курсу спеціальності «Кібербезпека»; навчальний план першого курсу спеціальності «Кібербезпека».

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):

4.1 Аналітична частина: огляд і аналіз предмета проектування.

4.2 Основна частина: постановка задачі; проектування бази даних за методом нормальних форм; створення та обробка баз даних за допомогою HeidiSQL та MySQL; організація захисту бази даних (рольове управління доступом, процедурний контроль цілісності даних, відключення віддаленого доступу); підготовка самостійної роботи з дисципліни «Бази даних».

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Завдання видав: (підпис консультанта, дата)	Завдання прийняв: (підпис студента, дата)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вступ	03.06.2022-05.06.2022	5%
2	Огляд і аналіз предмета проектування	25.04.2022-16.05.2022	20%
3	Постановка задачі	25.04.2022-02.05.2022	10%
4	Проектування бази даних за методом нормальних форм	02.05.2022-13.05.2022	20%
5	Створення та обробка бази даних за допомогою HeidiSQL та MySQL	13.05.2022-27.05.2022	20%
6	Організація захисту бази даних та використання в навчальному процесі	16.05.2022-31.05.2022	20%
7	Висновки та рекомендації	03.06.2022-05.06.2022	5%
8	Подання кваліфікаційної роботи до кафедри	13.06.2022	
9	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	23.06.2022	

Студент

Керівник роботи

(підпис)

(підпис)

Неллі Падафа

(Ім'я ПРІЗВИЩЕ)

Вікторія Пахомова

(Ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра: 45 с., 35 рис., 3 додатки, 15 джерел.

Об'єкт розробки – реляційна база даних здобувачів першого курсу спеціальності «Кібербезпека».

Мета кваліфікаційної роботи – створення захищеної бази даних здобувачів спеціальності «Кібербезпека».

Методи дослідження – проектування бази даних за методом нормальних форм та створення з використанням HeidiSQL та MySQL.

Створена за допомогою HeidiSQL та MySQL базу даних здобувачів першого курсу спеціальності «Кібербезпека». Виконано організацію захисту бази даних наступними засобами: рольове управління доступом; процедурний контроль цілісності даних; відключення віддаленого доступу. Рекомендовано створену та захищену базу даних до використання куратору наступних груп: КБ2111(915); КБ2112(916); КБ21130(920Б). Використання HeidiSQL та MySQL може бути рекомендовано здобувачам першого (бакалаврського) ступеня спеціальності «Кібербезпека» при виконанні самостійної роботи з дисципліни «Бази даних».

Ключові слова: БАЗА ДАНИХ, НОРМАЛЬНІ ФОРМИ, ЗАХИСТ, MySQL, HeidiSQL.

## ЗМІСТ

ВСТУП .....	6
1 ОГЛЯД І АНАЛІЗ ПРЕДМЕТА ПРОЕКТУВАННЯ.....	8
1.1 Методи проектування реляційної бази даних.....	8
1.2 Засоби програмної реалізації .....	10
1.3 Основні висновки.....	13
2 ПРОЕКТУВАННЯ, СТВОРЕННЯ ТА ОБРОБКА ЗАХИЩЕНОЇ БАЗИ ДАНИХ.....	14
2.1 Постановка задачі .....	14
2.2 Проектування бази даних за методом нормальних форм.....	14
2.3 Створення та обробка бази даних за допомогою HeidiSQL та MySQL .....	22
2.4 Організація захисту бази даних.....	33
2.4.1 Рольове управління доступом.....	33
2.4.2 Відключення віддаленого доступу .....	34
2.4.3 Зміна пароля адміністратора.....	35
2.4.4 Видалення значень за замовченням .....	37
2.5 Підготовка самостійної роботи з дисципліни «Бази даних».....	37
2.6 Основні висновки.....	42
ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ.....	43
ПЕРЕЛІК ПОСИЛАНЬ .....	44
ДОДАТОК А ПОЧАТКОВЕ ВІДНОШЕННЯ (ФРАГМЕНТ) .....	<b>Помилка!</b>
<b>Закладку не визначено.</b>	
ДОДАТОК Б АКТ НА ВПРОВАДЖЕННЯ..	<b>Помилка! Закладку не визначено.</b>
ДОДАТОК В ТЕЗИ ДО КОНФЕРЕНЦІЇ .....	<b>Помилка! Закладку не визначено.</b>

## ВСТУП

Сьогодні у всіх діяльностях пошук вирішення проблем раціонального використання актуальних методів та засобів обробки інформації має одне з важливих значень для функціонування. Розвиваються технічні та програмні засоби, що дозволяють реалізувати нові технології. Відомо також, що система управління файлами має багато проблем, які в свою чергу долає система управління реляційною базою даних. Оскільки потреба в інформації зростає, сутність системи баз даних стала неминучою. Саме в результаті цього з'явилася реляційна база даних, щоб вирішити ці недоліки файлової базової системи та інших систем, що використовуються для зберігання та отримання інформації. Що досить важливо, такі бази даних треба надійно захищати. Під цим поняттям маються на увазі заходи, створені задля запобігання її втрати, розкрадання чи зміни, що підтверджує актуальність теми.

Мета кваліфікаційної роботи – створення захищеної бази даних здобувачів спеціальності «Кібербезпека». Відповідно до мети поставлені наступні завдання:

1. Виконати огляд та аналіз предмета проектування.
2. Провести проектування бази даних здобувачів спеціальності «Кібербезпека» за методом нормальних форм.
3. Створити базу даних з використання засобів HeidiSQL та MySQL.
4. Виконати організацію захисту бази даних.

На сьогоднішній день існує багато моделей представлення бази даних. До них відносяться: ієрархічна, мережна, реляційна, пост реляційна, багатовимірна та об'єктно-орієнтована модель даних. Найпопулярніша сучасна модель даних – це реляційна. Ця модель використовує деякі математичні поняття, такі як теорія множин та логіка предикатів. Перша програма реляційних баз даних Інтернету була створена в 1995 році. В епоху реляційної бази даних введено

багато інших моделей, таких як об'єктно-орієнтована модель, об'єктно-реляційна модель тощо. Для проектування реляційної бази даних існує два методи: графічний та математичний. Прикладом реляційної бази даних є MySQL, Oracle і база даних Microsoft SQL Server. Для організації баз даних є багато програм, до яких відносяться: Microsoft Access, Oracle, MySQL, HeidiSQL, SQL Server та ін. Вони є системами керування базами даних.

Дана кваліфікаційна робота складається із вступу, двох розділів та висновків.

У першому розділі розглянуті можливі методи проектування реляційних баз даних, а також проведено огляд можливих додатків в яких створюються бази даних, таких як: СУБД Oracle, Microsoft Access, СУБД MS SQL Server, СУБД Borland Interbase, СУБД MySQL, HeidiSQL та ін.

У другому розділі виконано проектування бази даних здобувачів першого курсу спеціальності «Кібербезпека» за методом нормальних форм. Спроектвана база даних створена за допомогою HeidiSQL та MySQL. Проведена організація її захисту, розглянуто наступні засоби: рольове управління доступом, відключення віддаленого доступу, зміна паролю адміністратора та видалення значень за замовченням. Крім того, подана рекомендація до використання самостійної роботи з дисципліни «Бази даних» куратору груп КБ2111 (915); КБ2112 (916); КБ21130 (920Б) Пахомовій В.Н, у якій розглянуті питання яка містить у собі пояснення щодо встановлення таких методів захисту, як: парольний захист, рольове управління доступом, видалення значень за замовченням. Студентам першого курсу спеціальності «Кібербезпека» запропоновано виконати завдання з організації захисту бази даних у HeidiSQL та оформити відповідний звіт з результатами.

Результати кваліфікаційної роботи подавались у вигляді тез доповіді до Всеукраїнської науково-технічної конференції студентів і молодих учених, що відбулась у Українському Державному Університеті Науки та Технологій (УДУНТ) у 2022 році.



## **1 ОГЛЯД І АНАЛІЗ ПРЕДМЕТА ПРОЕКТУВАННЯ**

### **1.1 Методи проектування реляційної бази даних**

На сьогоднішній день існує декілька моделей представлення бази даних. Насамперед, треба визначити, що собою представляє база даних. База даних — це організована сукупність структурованої інформації або даних, які зазвичай зберігаються в електронній формі в комп'ютерній системі. [1]

Логічна структура, яка зберігається в базі даних, називається моделлю подання даних. В зв'язку з цим, до основних моделей, можна віднести: ієрархічну, мережну, реляційну, пост реляційну, багатовимірну та об'єктно-орієнтовану модель даних.

На практиці набула визнання реляційна модель даних, так як має чіткий математичний апарат. Математичним апаратом називають відношення. Також ця модель даних дуже проста, зрозуміла та зручна для реалізації на ЕОМ, що також стало основною причиною її популярності. Прикладом реляційної бази даних є MySQL, Oracle і база даних Microsoft SQL Server.

Прикладом використання реляційної бази даних є американська компанія Amazon. Це найбільший інтернет-сервіс продажу реальних товарів, які користуються масовим попитом. База даних цього сервісу базується на Amazon AWS, яка популярна за рахунок широких функціональних можливостей, максимальної безпеки, надійності та швидкого впровадження інновацій. [2]

Реляційна модель даних вперше була запропонована співробітником фірми IBM Едгаром Коддом. Ця модель даних представляє собою сукупність пов'язаних між собою відношень, які можуть змінюватися з часом. До основних елементів реляційної моделі даних можна віднести: сутність,

відношення, кортеж, атрибут, схема відношення, тіло відношення, домен, значення атрибуту, тип даних, первинний ключ. [3]

Під відношенням мають на увазі безліч елементів, які називаються кортежами. Кожен рядок таблиці представляє собою однакову структуру та складається з полів. Будь-яка таблиця реляційної бази даних складається з рядків (їх називають кортежами) і стовпців (так звані атрибути відношення). [4]

Існує також декілька методів проектування бази даних. До них відносяться: метод «Сутність – Зв'язок» та метод нормальних форм.

Метод «Сутність – Зв'язок» називають також методом «ER-діаграм». Цей метод заснований на використанні діаграм, званих відповідно діаграмами ER-екземплярів та діаграмами ER-Типу. Основними поняттями методу «Сутність–Зв'язок» є такі: сутність, атрибут сутності, ключ сутності, зв'язок між сутностями, ступінь зв'язку, клас приналежності екземплярів сутності, діаграми ER-примірників, діаграми ER-типу. [5]

Процес проектування бази даних з використанням методу нормальних форм є ітераційним. Він полягає у послідовному переведенні відношень з першої нормальної форми в нормальні форми високого ступеню за визначеними правилами. Кожний етап нормальної форми обмежує визначений тип функціональних залежностей, ліквідує аномалії та зберігає властивості попередніх нормальних форм.

На сьогодні є сформована послідовність нормальних форм. Це:

1. Перша нормальна форма (1НФ).
2. Друга нормальна форма (2НФ).
3. Третя нормальна форма (3НФ).
4. Посилена третя нормальна форма або нормальна форма Бойса-Кодда (БКНФ).
5. Четверта нормальна форма (4НФ).
6. П'ята нормальна форма (5НФ).

Для кожної нормальної форми є свої правила.

Для першої нормальної форми (1НФ) характерно те, що відношення знаходиться у 1НФ тільки тоді, коли всі його атрибути є простими (тобто мають одне значення).

Відношення знаходиться у другій нормальній формі (2НФ), якщо воно знаходиться у 1НФ та кожний не ключовий його атрибут функціонально повно залежить від первинного (складеного) ключа.

Для третьої нормальної форми є два визначення. Перше визначення представляє собою, що відношення знаходиться у 3НФ, якщо воно знаходиться у 2НФ і кожний його не ключовий атрибут не транзитивно залежить від первинного ключа. Друге визначення звучить так: відношення перебуває у 3НФ у тому і лише тому випадку, якщо всі не ключові атрибути відношення взаємно незалежні і залежать від первинного ключа.

Далі йде посилена третя форма або нормально форма Бойса-Кодда (БКНФ). Відношення знаходиться в БКНФ, якщо воно знаходиться в 3НФ і немає залежності ключів (атрибутів складового ключа) від не ключових атрибутів. Зазвичай на цьому етапі проектування закінчується. [6]

Як приклад бази даних, яка спроектована за методом нормальних форм, можна навести електронну бібліотеку навчального закладу УДУНТ. [7]

## **1.2 Засоби програмної реалізації**

Розвиток інформаційно-комунікаційних технологій дозволяє перевести великі обсяги інформації в електронний вигляд. З'явилися системи управління базами даних (СУБД), які дозволили спрощувати роботу у всіх сферах діяльності: освіту, медицину, фінансові інститути, виробничі підприємства, криміналістику тощо. СУБД забезпечують правильність, повноту та несуперечність даних, а також зручний доступ до них.

До таких програмних засобів реалізації баз даних можна віднести: Microsoft Access, Oracle, MS SQL Server, Borland Interbase, MySQL, HeidiSQL.

### **1.2.1 Microsoft Access**

Прикладна програма пакета Microsoft Office відноситься до реляційних, тобто побудованих на підставі таблиць, СУБД. Ця програма містить широкий спектр функцій, включаючи пов'язані запити, зв'язок із зовнішніми таблицями та базами даних. Завдяки вбудованій мові VBA, у Microsoft Access можна писати програми, що працюють із базами даних на мові SQL. У Access при заповненні осередку дані зберігаються автоматично, що дуже зручно і безпечно з боку цілісності даних. [8]

### 1.2.2 СУБД Oracle

База даних Oracle — це система управління реляційною базою даних. Його також називають OracleDB, або просто Oracle. Його виробляє та продає корпорація Oracle. Це один з найпопулярніших механізмів реляційних баз даних на ринку ІТ для зберігання, упорядкування та отримання даних.

База даних Oracle була першою БД, розробленою для корпоративних мережеских обчислень і сховищ даних. Корпоративні мережні обчислення надають найбільш гнучкий та економічно ефективний спосіб керування інформацією та додатками. Він використовує SQL-запити як мову для взаємодії з базою даних. [9]

### 1.2.3 СУБД MS SQL Server

Microsoft SQL Server — це система керування реляційною базою даних (RDBMS), яка підтримує широкий спектр додатків для обробки транзакцій, бізнес-аналітики та аналітики в корпоративних ІТ-середовищах. Microsoft SQL Server побудовано на основі SQL, стандартизованої мови програмування, яку адміністратори баз даних фахівці використовують для керування базами даних і запитів даних, які вони містять. [10]

### 1.2.4 СУБД Borland Interbase

InterBase — це повнофункціональна реляційна база даних, яку можна вбудовувати в розроблювані програмні продукти. InterBase відрізняється потужністю, низькою вартістю, високою продуктивністю, не вимагає

адміністрування, просто встановлюється, розгортається та масштабується, а також пропонує вбудовані можливості захисту даних. Крім того, редакція InterBase Server постачається з потужними функціями аварійного відновлення, які забезпечують збереження даних навіть у разі збою сервера. [11]

### 1.2.5 СУБД MySQL

MySQL — це база даних з відкритим вихідним кодом, яка сприяє ефективному управлінню базами даних, підключаючи їх до програмного забезпечення. Це стабільне, надійне та потужне рішення з розширеними функціями, зокрема:

#### — Безпека даних

MySQL відомий у всьому світі як найбезпечніша та надійна система керування базами даних, яка використовується в популярних веб-додатках, включаючи WordPress, Drupal, Joomla, Facebook і Twitter.

#### — Масштабування за вимогою

MySQL пропонує неперевершену масштабованість, щоб полегшити керування глибоко вбудованими програмами, використовуючи меншу площу, навіть у великих сховищах, які накопичують терабайти даних.

#### — Висока ефективність

MySQL має чітку структуру системи зберігання даних, яка полегшує системним адміністраторам конфігурацію сервера баз даних MySQL для бездоганної роботи. [12]

### 1.2.6 HeidiSQL

Розробники описують HeidiSQL як клієнт графічного інтерфейсу для MariaDB, MySQL, Microsoft SQL Server і PostgreSQL. HeidiSQL — це корисний і надійний інструмент, розроблений для веб-розробників, які використовують популярний сервер MariaDB або MySQL, бази даних Microsoft SQL або PostgreSQL. Він дає змогу переглядати та редагувати дані, створювати та редагувати таблиці, представлення даних, процедури, тригери та заплановані

події. Крім того, можна експортувати структуру та дані у файл SQL, буфер обміну або на інші сервери. [13]

### **1.3 Основні висновки**

У першому розділі кваліфікаційної роботи розглянуто питання огляду та аналізу предмета проектування.

1. Огляд джерел продемонстрував, що для проектування реляційної бази даних існує декілька методів: метод нормальних форм та метод «Сутність-Зв'язок». Для виконання кваліфікаційної роботи обрано метод нормальних форм у зв'язку з тим, що: він дуже зручний, простий, забезпечує захист даних, робить базу даних більш гнучкою, простою та зрозумілою, тим самим усуваючи надмірність та неузгоджені залежності.

2. Огляд джерел та їх аналіз продемонстрував існування наступних засобів реалізації баз даних, а саме: Microsoft Access, Oracle, MS SQL Server, Borland Interbase, MySQL, HeidiSQL та ін.

Для реалізації спроектованої реляційної бази даних здобувачів спеціальності «Кібербезпека» обрані засоби HeidiSQL та MySQL. Вони обрані у зв'язку з тим, що MySQL – це стабільне, надійне та потужне рішення з розширеними функціями безпеки даних, масштабуванням за вимогою та високої ефективності. Для спрощеної роботи з MySQL йде графічний інтерфейс HeidiSQL. Ця програма є дуже простою в освоєнні та роботі з базами даних, що дозволяє максимально ефективно створити потрібну базу даних та надійно її захистити.

## **2 ПРОЕКТУВАННЯ, СТВОРЕННЯ ТА ОБРОБКА ЗАХИЩЕНОЇ БАЗИ ДАНИХ**

### **2.1 Постановка задачі**

Для створення бази даних здобувачів спеціальності «Кібербезпека», вхідними даними є: робочий навчальний план першого курсу та дані про студентів наступних груп: КБ2111 (915); КБ2112 (916); КБ21130 (920Б). Робочий навчальний план першого курсу наведений у Додатку А. Дані про студентів представлені їх куратором Пахомовою В.М.

Основною задачею, що обмірковується в процесі проектування реляційної бази даних, є задача нормалізації її відношень. При проектуванні бази даних здобувачів спеціальності «Кібербезпека» використаний метод нормальних форм, який є чітким математичним засобом.

### **2.2 Проектування бази даних за методом нормальних форм**

Складене початкове відношення «Здобувачі\_кафедри\_ЕОМ» має таблиці здобувачів спеціальності «Кібербезпека», яке подано в Додатку Б, знаходиться в першій нормальній формі, тому що всі атрибути відношення атомарні.

Деякі пояснення щодо атрибутів відношення:

- 1) ПІБ\_студента – прізвище, ім'я, по-батькові здобувача.
- 2) Дата\_народження – день народження здобувача.
- 3) Телефон\_студента – мобільний номер здобувача.

- 4) Місто\_народження\_студента – місто, де народився здобувач.
- 5) Адреса\_проживання\_студента – адреса нинішнього проживання здобувача.
- 6) Група – група, в якій вчиться здобувач.
- 7) Email\_студента – електронна адреса здобувача.
- 8) Рейтинг\_при\_вступі – рейтинг, який має здобувач при зданих іспитах.
- 9) Форма\_навчання – форма навчання здобувача (держзамовлення/контракт).
- 10) ПІБ\_матері – прізвище, ім'я, по-батькові матері здобувача.
- 11) Рік\_народження\_матері – рік народження матері здобувача.
- 12) Телефон\_матері – мобільний номер матері здобувача.
- 13) Адреса\_матері – адреса нинішнього проживання матері здобувача.
- 14) ПІБ\_батька – прізвище, ім'я, по-батькові батька здобувача.
- 15) Рік\_народження\_батька – рік народження батька здобувача.
- 16) Телефон\_батька – мобільний номер батька здобувача.
- 17) Адреса\_батька – адреса нинішнього проживання батька здобувача.
- 18) Предмет – назва предмету.
- 19) ПІБ\_викладача – прізвище, ім'я, по-батькові викладача, який веде предмет.
- 20) Форма\_складання\_іспиту – форма того, як буде проходити іспит.
- 21) Год/кр – години та кредити, які виділені на предмет.
- 22) Дата\_складання\_іспиту – дата, коли буде проходити іспит.
- 23) Бали за іспит – скільки отримав здобувач за складання іспиту.

Ступінь – це кількість атрибутів відношення та ця характеристика є статичною.

Потужність – це кількість кортежів, ця характеристика є динамічною.

У даному початковому відношенні кількість атрибутів дорівнює 23, а кількість кортежів 33.



Первинний ключ відношення складений, складається з трьох атрибутів (ПІБ\_студента, Предмет, Дата\_складання\_іспиту). Відношення знаходиться у 1НФ, так як всі атрибути атомарні, але відношення містить надлишкове дублювання даних, що є причиною аномалій редагування. Явна надлишковість полягає в тому, що у відношенні «Здобувачі\_кафедри\_ЕОМ» рядки з даними про студентів, повторюються декілька разів. Тому, щойно один із студентів змінить прізвище, то цей факт повинен бути відображений в усіх кортежах. В іншому випадку буде мати місце протиріччя у даних, що є прикладом аномалії редагування, обумовленої явною надлишковістю даних у відношенні.

Його залежність наведена на рисунку 2.1 та результуюче відношення наведено у Додатку Б.



Рисунок 2.1 – Залежність між атрибутами відношення

Далі йде приведення відношення «Здобувачі\_кафедри\_ЕОМ» до 2НФ.

Відношення знаходиться в другій нормальній формі (2НФ), якщо воно вже знаходиться в 1НФ і кожен не ключовий атрибут відношення функціонально повно залежить від первинного (складеного) ключа. [14]

Для ліквідації часткової функціональної залежності та переведення відношення до 2НФ необхідно, використовуючи операції проєкції, розкласти його на декілька відношень:

- 1) Побудувати проєкцію без атрибутів, що знаходяться в частковій функціональній залежності від первинного ключа;
- 2) Побудувати проєкцію на складові складеного первинного ключа та атрибути, що залежать від первинного ключа. [14]

У результаті отримано відношення R1, яке наведено на рисунку 2.2. У відношенні R1 первинний ключ складений: ПІБ\_студента, Предмет, Дата\_складання\_іспиту.

Отримане відношення R1 наведено на рисунку 2.3. Результуюче відношення наведено у Додатку Б.

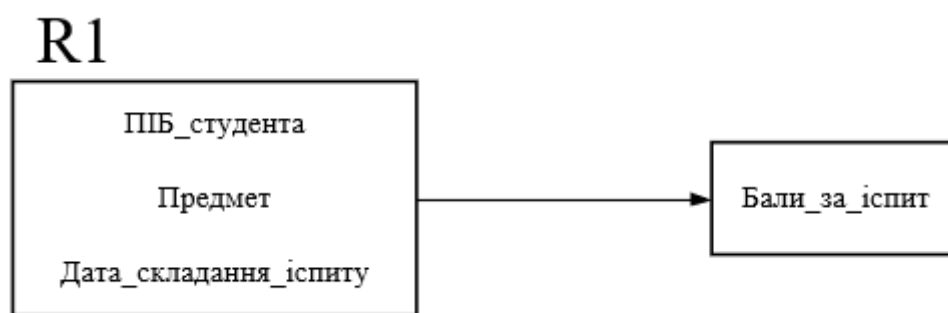


Рисунок 2.2 – Графічна інтерпретація відношення R1

Слідом ми отримали відношення R2, яке наведено на рисунку 2.3. У відношенні R2 первинний ключ простий: ПІБ\_студента.

Результуюче відношення наведено у Додатку Б.

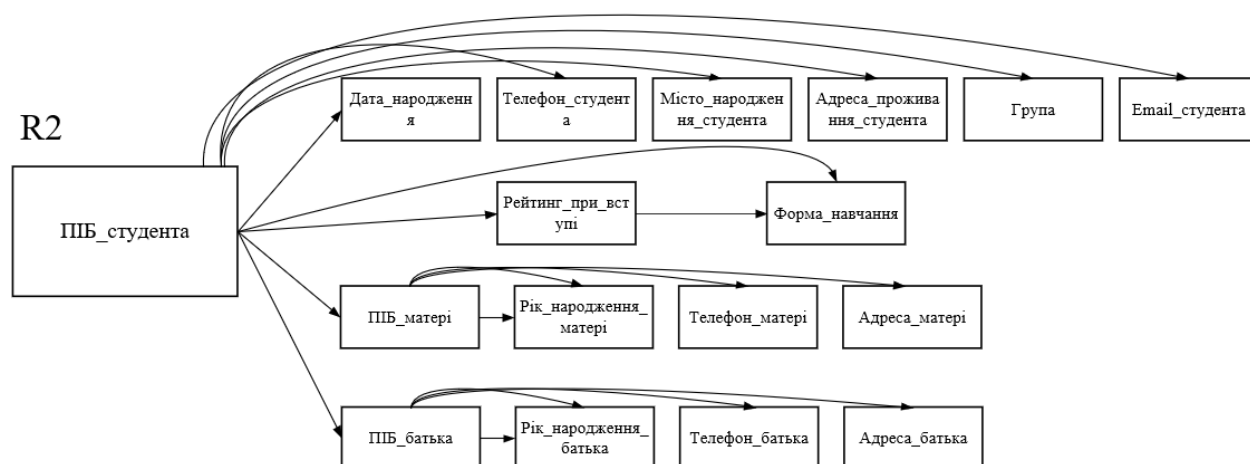


Рисунок 2.3 – Залежність між атрибутами відношення R2

Також отримано відношення R3, яке наведено на рисунку 2.4. Таблиця наведена у Додатку Б. У відношенні R3 первинний ключ простий: Предмет.

Таблиця відношення наведена у Додатку Б.

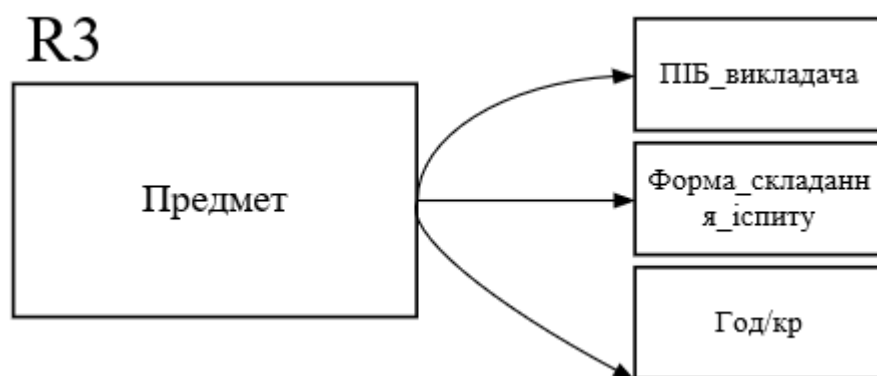


Рисунок 2.4 – Графічна інтерпретація відношення R3

Відношення знаходиться у третій нормальній формі (3НФ), якщо воно вже знаходиться у 2НФ і кожний його не ключовий атрибут нетранзитивно залежить від первинного ключа. [14]

Тобто мета цього етапу нормалізації – позбавлення від транзитивних залежностей.

Відношення R1, що отримане на попередньому етапі нормалізації, залишається без змін, тому що транзитивні залежності у ньому відсутні. У відношенні R2 присутні наступні транзитивні залежності: Рейтинг\_при\_вступі та Форма\_навчання, ПІБ\_матері та Рік\_народження\_матері, ПІБ\_батька та Рік\_народження\_батька.

Транзитивні залежності породжують надлишкове дублювання даних у відношенні. Ліквідуємо їх, для цього використовуємо операцію проєкції на атрибути, отримаємо відношення R4, R5, R6, R7.

Відношення R4 наведено на рисунку 2.5 та таблиця відношення наведена у Додатку Б.

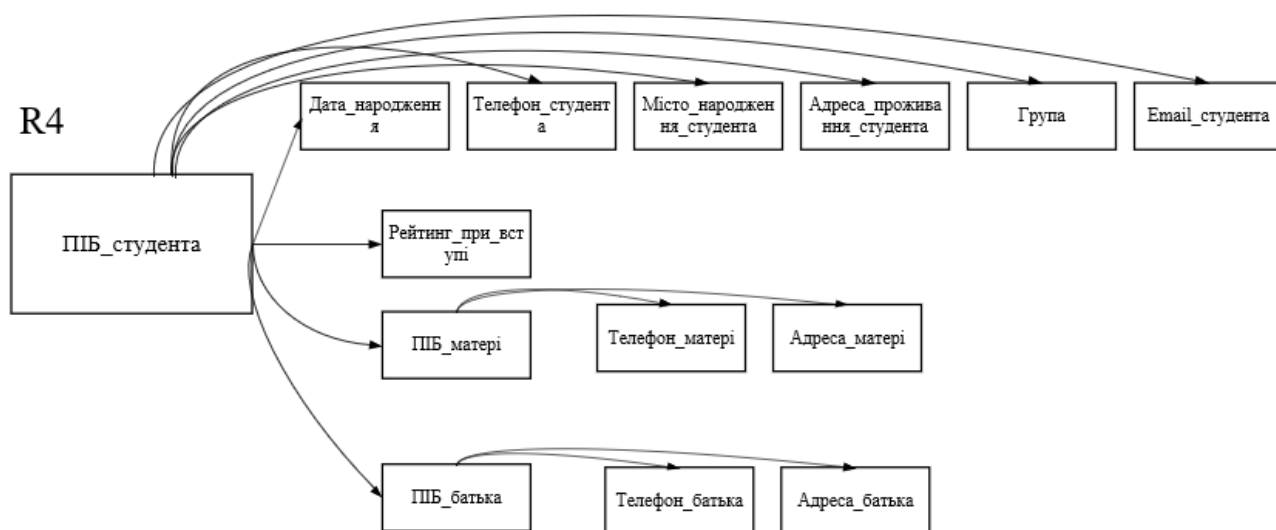


Рисунок 2.5 – Залежність між атрибутами відношення R4

Відношення R5 наведено на рисунку 2.6 та таблиця відношення наведена у Додатку Б.

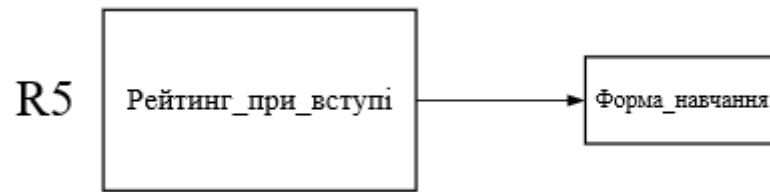


Рисунок 2.6 – Графічна інтерпретація відношення R5

Відношення R6 наведено на рисунку 2.7 та таблиця відношення наведена у Додатку Б.

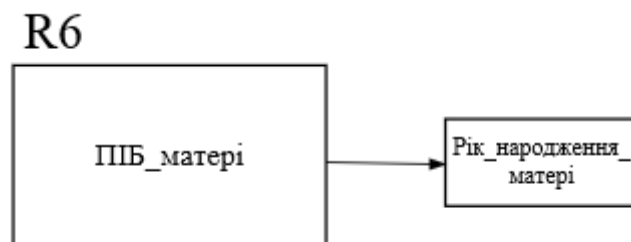


Рисунок 2.7 – Графічна інтерпретація відношення R6

Відношення R7 наведено на рисунку 2.8 та таблиця відношення наведена у Додатку Б.

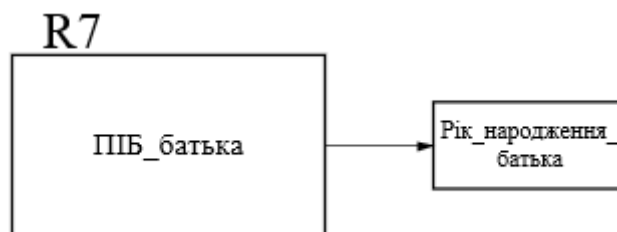


Рисунок 2.7 – Графічна інтерпретація відношення R7

У відношеннях R4, R5, R6, R7 відсутні транзитивні залежності.

Структура спроектованої бази даних за методом нормальних форм представлена на рисунку 2.9.

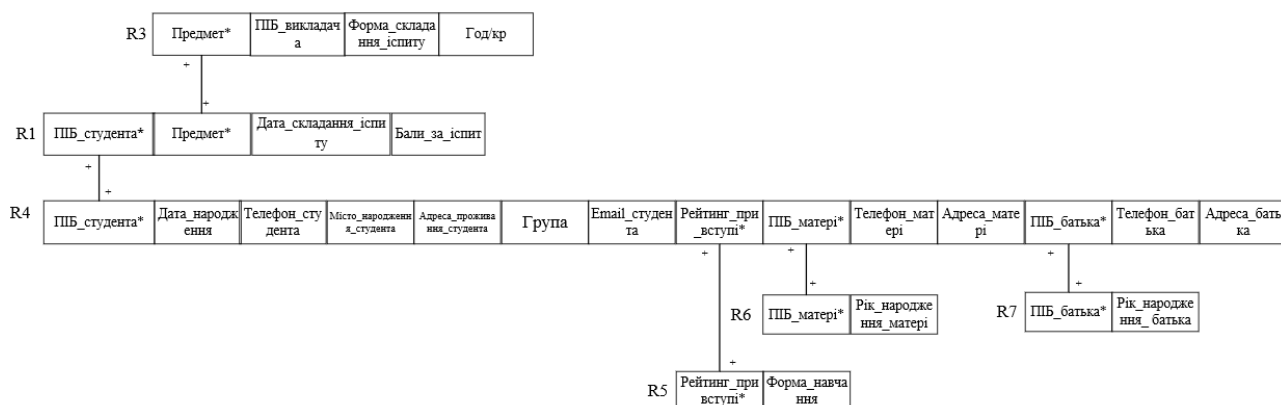


Рисунок 2.9 – Структура спроектованої бази даних за методом нормальних форм

У даній роботі результатом проектування є база даних, що складається з таблиць R1, R3, R4, R5, R6 та R7.

У відношенні R1 ключовий атрибут ПІБ\_студента не залежить від не ключових атрибутів: Предмет, Дата\_складання\_іспиту, Бали\_за\_іспит.

У відношенні R3 ключ Предмет не залежить від атрибутів: ПІБ\_викладача, Форма\_складання\_іспиту, Год/кр.

У відношенні R4 ключ ПІБ\_студента не залежить від атрибутів: Дата\_народження, Телефон\_студента, Місто\_народження\_студента, Адреса\_проживання\_студента, Група, Email\_студента, Рейтинг\_при\_вступі, ПІБ\_матері, Телефон\_матері, Адреса\_матері, ПІБ\_батька, Телефон\_батька, Адреса\_батька.

У відношенні R5 ключ Рейтинг\_при\_вступі не залежить від атрибута Форма\_навчання.

У відношенні R6 ключ ПІБ\_матері не залежить від атрибута Рік\_народження\_матері.

У відношенні R7 ключ ПІБ\_батька не залежить від атрибута Рік\_народження\_батька.

Усі отримані відношення: R1, R3, R4, R5, R6 та R7 – задовольняють вимогам БКНФ – знаходяться в третій нормальній формі, а їх первинні ключі не

залежать від не ключових атрибутів, тому процес нормалізації можна вважати завершеним. [14]

### 2.3 Створення та обробка бази даних за допомогою HeidiSQL та MySQL

Створення бази даних здобувачів спеціальності «Кібербезпека» проходило у програмі HeidiSQL при підтримці мови SQL. Для початку створення бази даних необхідно у менеджері сеансів HeidiSQL створити новий сеанс, що представлений на рисунку 2.10. Це буде нашим сервером, на якому буде зберігатися наша база даних. У подальшому саме в цьому вікні можна буде її надійно захистити.

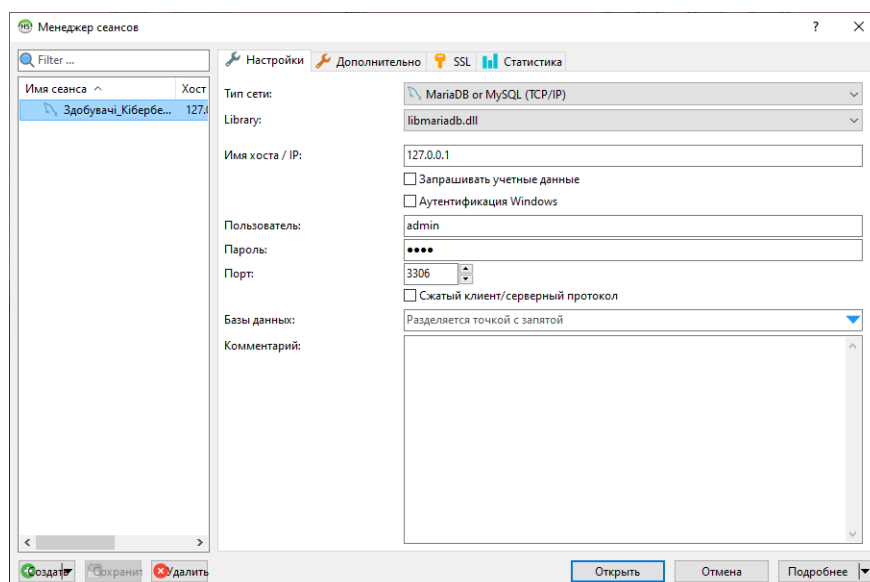


Рисунок 2.10 – Менджер сеансів

Далі створено 6 таблиць: СТУДЕНТ, РЕЙТИНГ, ПРЕДМЕТ, МАТИ, БАТЬКО та БАЛИ\_ЗА\_СКЛАДАННЯ\_ІСПИТУ. Це показано на рисунку 2.11 та фрагмент запиту наведено на рисунку 2.12.

здобувачі	192,0 KiB
бали_за_складання_іспиту	48,0 KiB
батько	16,0 KiB
мати	16,0 KiB
предмет	16,0 KiB
рейтинг	16,0 KiB
студент	64,0 KiB

Рисунок 2.11 – Створені таблиці





```
CREATE TABLE `студенти` ( `ПІБ_студента` INT NULL, `Дата_народження` INT NULL, `Телефон_студента` INT NULL, `Місто_народження` INT NULL,
SELECT `DEFAULT_COLLATION_NAME` FROM `information_schema`.`SCHEMATA` WHERE `SCHEMA_NAME`='здобувачі';
SHOW TABLE STATUS FROM `здобувачі`;
SHOW FUNCTION STATUS WHERE `Db`='здобувачі';
SHOW PROCEDURE STATUS WHERE `Db`='здобувачі';
SHOW TRIGGERS FROM `здобувачі`;
SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM `information_schema`.`EVENTS` WHERE `EVENT_SCHEMA`='здобувачі';
SELECT * FROM `information_schema`.`COLUMNS` WHERE TABLE_SCHEMA='здобувачі' AND TABLE_NAME='студенти' ORDER BY ORDINAL_POSITION;
```

Рисунок 2.12 – Фрагмент запиту створення однієї з таблиці

Далі потрібно визначитися з типом даних кожного атрибута, що є дуже важливою частиною створення бази даних. Зробити це можна через контекстне меню відображених атрибутів. Це можна побачити на рисунку 2.13.

#	Имя	Тип данных	Длина/Значение
1	ПІБ_студента	VARCHAR	50
2	Дата_народження	DATE	
3	Телефон_студента	VARCHAR	50
4	Місто_народження	VARCHAR	50
5	Адреса_проживання	VARCHAR	50
6	Група	VARCHAR	50
7	Email_студента	VARCHAR	50
8	ПІБ_матері	VARCHAR	50
9	Телефон_матері	VARCHAR	50
10	Адреса_матері		
11	ПІБ_батька		
12	Телефон_батька		
13	Адреса_батька		

Помощь Отмена

Фильтр Регулярное выражение

бывачі';

Флаги

Текст

VARCHAR

CHAR

TINYTEXT

TEXT

MEDIUMTEXT

Рисунок 2.13 – Вибір типу даних для кортежів

Майже для кожного атрибута визначений тип даних VARCHAR, а для наступних атрибутів: Дата\_народження\_студента, Рік\_народження\_батька, Рік\_народження\_матері обрані типи даних – YEAR та DATE. Це дозволяє коректно відобразити дату народження у базі даних.

Далі треба заповнити наші таблиці даними для подальшої роботи з ними. Кортежі можна додати, натиснувши кнопку «Вставте строку в таблицю» або

через кнопку Insert на клавіатурі. Зовнішній вигляд кнопки наведено на рисунку 2.14.

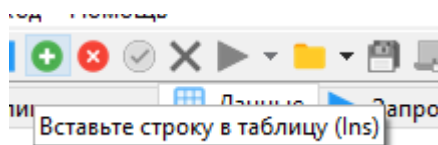


Рисунок 2.14 – Кнопка створювання кортежів

Заповнюємо нашу таблицю. Наприклад, додамо нашого першого студента та заповнимо його ПІБ, Дата\_народження, Телефон\_студента, Місто\_народження. Фрагмент запиту заповнювання наведений на рисунку 2.15.

```
INSERT INTO `здобувачі`.`студенти` (`ПІБ_студента`, `Дата_народження`,
`Телефон_студента`, `Місто_народження`) VALUES ('Говоруха Єгор Євгенійович',
'2001-11-18', ' ', 'м.Дніпро');
SELECT `ПІБ_студента`, `Дата_народження`, `Телефон_студента`,
`Місто_народження`, `Адреса_проживання`, `Група`, `Email_студента`,
`ПІБ_матері`, `Телефон_матері`, `Адреса_матері`, `ПІБ_батька`, `Телефон_батька`,
`Адреса_батька` FROM `здобувачі`.`студенти` WHERE `ПІБ_студента`='Говоруха Єгор
Євгенійович' AND `Дата_народження`='2001-11-18' AND
`Телефон_студента`=' ' AND `Місто_народження`='м.Дніпро'
```

Рисунок 2.15 – Фрагмент запиту заповнювання таблиці

І за таким прикладом заповнюємо кожну з наших таблиць. Наприкінці, це виглядає як на рисунку 2.16 та на рисунку 2.17.

ПІБ_студента	Дата_народження	Телефон_студента	Місто_народження_студента	Група	Email_студента	Адреса_проживання_студента
Говоруха Єгор Євгенійович	2001-11-18	*****	м.Дніпро	920-5	*****	*****
Гугучія Меггі Папунівна	2001-12-01	*****	м.Дніпро	920-5	*****	*****
Серов Олександр Олександрович	2002-03-07	*****	м.Дніпро	920-5	*****	*****
Шиліна Лілія Сергіївна	2002-03-07	*****	м.Новомосковськ	920-5	*****	*****
Гончаренко Артем Константинович	2004-08-03	*****	м.Дніпро	916-5	*****	*****
Гринь Мирослав Володимирович	2004-01-06	*****	м.Покровськ	916-5	*****	*****
Заруцький Петро Андрійович	2004-02-29	*****	м.Дніпро	916-5	*****	*****
Кабринська Анна Андріївна	2004-01-29	*****	м.Дніпро	916-5	*****	*****
Корнюк Вероніка Максимівна	2004-07-14	*****	м.Дніпро	916-5	*****	*****
Мельников Владислав Сергійович	2004-08-08	*****	м.Донецьк	916-5	*****	*****
Метьюлін Ігор Сергійович	2004-03-04	*****	м.Лисичанськ	916-5	*****	*****
Ніколаєнко Марк Дмитрович	2003-09-19	*****	м.Павлоград	916-5	*****	*****
Панін Данило Васильович	2004-01-19	*****	м.Дніпро	916-5	*****	*****
Репін Євгеній Сергійович	2004-01-16	*****	м.Дніпро	916-5	*****	*****

Рисунок 2.16 – Заповнена таблиця «Студент»

Предмет	ПІБ_викладача	Форма_склад_іспиту	Год/кр
Вища математика	Бабиц.Ю.П.	екзамен	165/5.5
Основи програмування	Івін П.В.	екзамен	135/4.5
Фізика технікум-вуз	Беляев М.Б.	екзамен	120/4
Історія та культура України	Лященко І.С.	екзамен	150/5
Іноземна мова	Мірошніченко І Г	залік	90/3
Фізичне виховання	Лутаева Н.П.	залік	60/2
Арифметичні та логічні основи ЕОМ	Івін П.В.	залік	120/4
Вступ до спеціальності	Остапек	залік	90/3
Фізика спец. розділи	Штапенко Е.П	екзамен	120/4
Комп'ютерна електроніка	Беляев М.Б.	екзамен	90/3
Основи програмування технікум-вуз	Івін П.В.	екзамен	120/4
Історія та культура України технікум-вуз	Лященко І.С.	залік	90/3
Іноземна мова технікум-вуз	Мірошніченко І.Г.	залік	60/2

Рисунок 2.17 – Заповнена таблиця «Предмет»

Після закінчення заповнення усіх таблиць, нам потрібно їх зв'язати, щоб кожна таблиця могла контактувати між собою та синхронізувати дані для подальшої роботи з базою даних.

Для того, щоб таблиці зв'язувались між собою, потрібно додати унікальні ідентифікатори для: ПІБ\_студента, Предмет, ПІБ\_матері, ПІБ\_батька, Рейтинг\_при\_вступі, так як мова програмування не дозволяє створювати ключі, які мають текстовий тип даних. Тому прийняте рішення додавання змінної, яка має тип даних INT та вираз за замовченням AUTO\_INCREMENT (тобто автоматичне присвоювання базою даних унікального значення для кожного унікального кортежу).

Таким чином, до кожного атрибуту ПІБ\_студента присвоєний унікальний ключ ID\_студента. Це наглядно показано на рисунку 2.18.

ID_студента	ПІБ_студента
1	Говоруха Єгор Євгенійович
2	Гугучія Меггі Папунівна
3	Серов Олексій Олексійович
4	Шиліна Лілія Сергіївна
5	Гончаренко Артем Константинович

Рисунок 2.18 – Доданий унікальний ключ

До кожного атрибута ПІБ\_матері був доданий ключ ID\_матері, до ПІБ\_батька – ID\_батька, до Предмет – ID\_предмета та до кожного Рейтинг\_при\_вступі – ID\_рейтинг. То, як це виглядає на практиці, показано на рисунках 2.19, 2.20, 2.21, 2.22.

ID_мати	ПІБ_матері
1	Демченко Галина Олегівна
2	Велика Світлана Федорівна
3	Серова Надія Василівна
4	Шиліна Тетяна Володимирівна
5	Гончаренко Світлана Олексіївна
6	Гринь Алла Леонидівна
7	Заруцька Надія Володимирівна
8	Кабринська Світлана Дмитрівна
9	Дружкевич Юлія Константинівна
10	Мельникова Марина Володимирівна

Рисунок 2. 19 – Доданий унікальний ключ «ID\_мати»

ID_батька	ПІБ_батька
1	Говоруха Євгеній Михайлов
2	помер
3	Серов Олексій Іванович
4	Шилін Сергій Миколайович
5	Гончаренко Константин Борисович
6	Гринь Володимир Миколайович
7	Заруцький Андрій Антонович
8	Якубовський Андрій Васильович
9	Дружкевич Максим Олегович
10	Мельников Сергій Миколайович

Рисунок 2. 20– Доданий унікальний ключ «ID\_батька»

ID_предмета	Предмет
1	Вища математика
2	Основи програмування
3	Фізика технікум-вуз
4	Історія та культура України
5	Іноземна мова
6	Фізичне виховання
7	Арифметичні та логічні основи ЕОМ
8	Вступ до спеціальності
9	Фізика спец. розділи
10	Комп'ютерна електроніка
11	Основи програмування технікум-вуз
12	Історія та культура України технікум-вуз
13	Іноземна мова технікум-вуз

Рисунок 2.21 – Доданий унікальний ключ «ID\_предмета»

ID_рейтинг	Рейтинг_при_вступі
1	138,50
2	147,25
3	116,00
4	142,50
5	126,137
6	179,963
7	138,873
8	152,745
9	159,273

Рисунок 2.22 – Доданий унікальний ключ «ID\_рейтинг»

Після додавання унікальних ключів, можна зв'язувати таблиці зовнішніми ключами.

Для таблиці «Бали\_за\_складання\_іспиту» створені два зовнішніх ключа. Це ключі «Предмет» та «Студент». Ключ «Предмет» бере дані з таблиці з назвами предметів, а ключ «Студент» бере дані з таблиці дані про ПІБ\_студента. Це можна зробити за допомогою вкладки «Зовнішні ключі», як зображено на рисунку 2.23.

Основной Опции Индексы (2) Внешние ключи (2) Проверить ограничения (0) Партиции Код CREATE Код ALTER									
Добавить	Имя ключа	Столбцы	Справочная таблица	Внешние столбцы	Пр				
Удалить	Предмет	ID_предмета	здобувачі.предмет	ID_предмета	Н				
Очистить	Студент	ID_студента	здобувачі.студент	ID_студента	Н				

Столбцы: + Добавить x Удалить ▲ Вверх ▼ Вниз									
#	Имя	Тип данных	Длина/Значения	Беззнаковое	Разрешить N...	Zerofill	По умолчанию	Комм	
1	ID_студента	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
2	ID_предмета	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
3	Дата_складання_ісп...	DATE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
4	Бали_за_іспит	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		

Рисунок 2.23 – Зовнішні ключі для таблиці «Предмет»

Таким чином, таблиця «Бали\_за\_складання\_іспиту» буде брати дані з таблиць «Предмет» та «Студент».

В таблиці це буде відображатися, як унікальний код для кожного студента та його предмета, під яким буде ПІБ або Предмет. Приклад наведений на рисунку 2.24.

ID_студента	ID_предмета	Дата_складання_іспиту	Бали_за_іспит
1: Говоруха Єгор	3	2022-01-05	60
1: Говоруха Єгор	1	2022-01-10	68
5: Гончаренко А	11	2022-01-12	75
6: Гринь Миросл	10	2022-01-15	0
2: Гугучія Меггі	3	2022-01-05	0
22: Довгаль Ілон	1	2022-01-10	73
7: Заруцький Пе	11	2022-01-12	69
23: Іванов Даніл	10	2022-01-14	0
8: Кабринська А	3	2022-01-05	67
24: Кожурін Стан	1	2022-01-10	75
9: Корнюк Верон	11	2022-01-12	82
25: Малюк Андрі	10	2022-01-14	0
10: Мельников В	3	2022-01-05	75
11: Метьолкін Іг	1	2022-01-10	84
26: Назаренко Се	11	2022-01-12	75
12: Ніколаєнко М	10	2022-01-14	0
27: Новосьолов	3	2022-01-05	75
4	11	2022-01-12	75
4	10	2022-01-14	0
5	4	2022-01-12	83
5	9	2022-01-14	69
5	1	2022-01-10	80
5	2	2022-01-05	86
6	4	2022-01-12	88
6	9	2022-01-14	66

Рисунок 2.24– Відображення внутрішніх даних ключа

В таблиці «Студент» буде три зовнішніх ключа: Батько, Мати та Рейтинг. Кожен з них підтягує інформацію до основної таблиці про студента. Це можна побачити на рисунку 2.25.

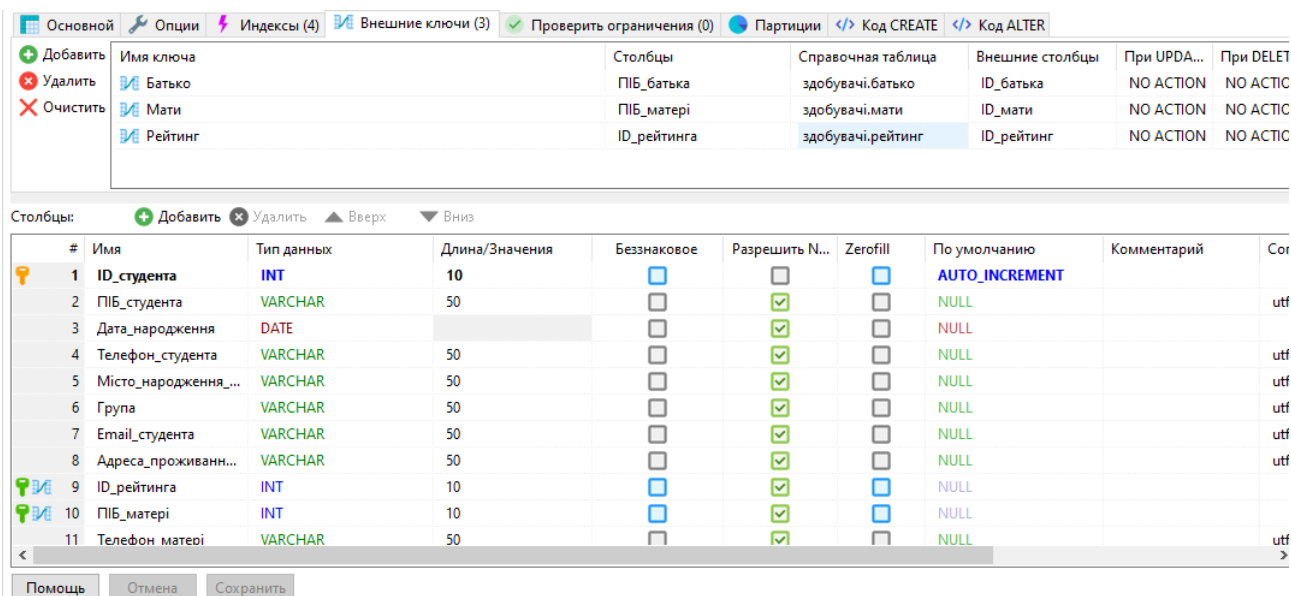


Рисунок 2.25 – Зовнішні ключі в таблиці «Студент»

Таким чином, для студента, який носить унікальний код «3», база даних покаже його рейтинг. Це видно на рисунку 2.26.

ID_рейтинга	ПИБ_матері
1	1
2	2
3: 116,00	3
3: 116,00	4
12: 116,025	5
13: 119,340	6
5: 126,137	7
31: 126,582	8
25: 132,855	9
15: 133,569	10
1: 138,50	11
7: 138,873	12
27: 138,873	13
14: 140,556	14
23: 141,117	15
19: 141,598	16
4: 142,50	17
33: 143,769	18
18: 145,299	19
	20
	21
	22

Рисунок 2.26 – Відображення внутрішніх даних ключа

Також це видно на прикладі атрибутів ПИБ\_матері та ПИБ\_батька. Це є на рисунку 2.27 та 2.28.

ID_рейтинга	ПІБ_матері
1	1
2	2
3	3: Серова Наді
4	25: Малюк Ірин
5	10: Мельников
6	11: Метьюлкіна
7	26: Назаренко
8	32: Нікіфорова
9	12: Ніколаєнко
10	29: Попик Олен
11	14: Репіна Мар
12	3: Серова Наді
13	30: Солопова І
14	20: Старюк Дан
15	16: Танцура Ол
16	31: Терентьєва
17	19: Чайка Ірина
18	4: Шиліна Тетя
19	33: Шипілова С

Рисунок 2.27– Відображення внутрішніх даних ключа ПІБ\_матері

ПІБ_батька	Телефон_батька
2	*****
3: Серов Олекс	*****
28: Пірогов Анд	*****
13: Погребецьк	*****
2: помер	*****
29: Попик Андр	*****
14: Репін Сергій	*****
3: Серов Олекс	*****
15: Сліпець Анд	*****
30: Солопов Ан	*****
16: Танцура Ігор	*****
31: Терентьев О	*****

Рисунок 2.28– Відображення внутрішніх даних ключа ПІБ\_батька

Таким чином, усі таблиці пов'язані друг з другом та мають відповідну інформацію.

Щоб дізнатися конкретну інформацію, наприклад, про тих, хто народився у м.Дніпро та склав іспити на п'ять, котрого немає в основній таблиці про студентів, треба написати запит та виконати його. Вигляд запиту є на рисунку 2.29.



1	SELECT ПІБ_студента, Предмет, Бали_за_іспит, Місто_народження_студента
2	FROM студент JOIN Бали_за_складання_іспиту
3	ON студент.`ID_студента` = Бали_за_складання_іспиту.`ID_студента`
4	JOIN Предмет ON Бали_за_складання_іспиту.`ID_предмета`=Предмет.`ID_предмета`
5	WHERE Місто_народження_студента = 'м.Дніпро' AND Бали_за_іспит > 90

Результат #1 (10r × 4c)			
ПІБ_студента	Предмет	Бали_за_іспит	Місто_народження_студента
Корнюк Вероніка Максимівна	Вища математика	92	м. Дніпро
Корнюк Вероніка Максимівна	Основи програмування	94	м. Дніпро
Танцура Артем Ігорович	Вища математика	92	м. Дніпро
Андреев Назар Михайлович	Вища математика	93	м. Дніпро
Андреев Назар Михайлович	Основи програмування	92	м. Дніпро
Малюк Андрій Дмитрович	Вища математика	92	м. Дніпро
Попик Маргарита Андріївна	Історія та культура України	96	м. Дніпро
Попик Маргарита Андріївна	Фізика спец. розділи	93	м. Дніпро
Попик Маргарита Андріївна	Вища математика	95	м. Дніпро
Попик Маргарита Андріївна	Основи програмування	92	м. Дніпро

Рисунок 2.29– Запит для відображення тих, хто народився у м.Дніпро та склав іспити на п'ять

Якщо треба побачити, які студенти складають ті або чи інші предмети, та що це за предмет, ми також напишемо запит, який видно на рисунку 2.30.

Хост: 127.0.0.1

База даних: здобувачі

Таблиця: предмет

Данные

Запрос\*

1

SELECT \* FROM предмет JOIN студент WHERE предмет.`ID\_предмета` = студент.`ID\_студента`

Результат #1 (13r × 20c)

ID_предмета	Предмет	ПІБ_викладача	Форма_склад_іспиту	Год/кр	ID_студента	ПІБ_студента
1	Вища математика	Бабич.Ю.П.	екзамен	165/5.5	1	Говоруха Єгор Євгенійович
2	Основи програмування	Івін П.В.	екзамен	135/4.5	2	Гугучія Меггі Папунівна
3	Фізика технікум-вуз	Беляев М.Б.	екзамен	120/4	3	Серов Олексій Олексійович
4	Історія та культура України	Лященко І.С.	екзамен	150/5	4	Шиліна Лілія Сергіївна
5	Іноземна мова	Мірошниченко І.Г.	залік	90/3	5	Гончаренко Артем Константинович
6	Фізичне виховання	Лутаєва Н.П.	залік	60/2	6	Гринь Мирослав Володимирович
7	Арифметичні та логічні основи ЕОМ	Івін П.В.	залік	120/4	7	Заруцький Петро Андрійович
8	Вступ до спеціальності	Остапеч	залік	90/3	8	Кабринська Анна Андріївна
9	Фізика спец. розділи	Штапенко Е.П.	екзамен	120/4	9	Корнюк Вероніка Максимівна
10	Комп'ютерна електроніка	Беляев М.Б.	екзамен	90/3	10	Мельников Владислав Сергійович
11	Основи програмування технікум-вуз	Івін П.В.	екзамен	120/4	11	Метьолькін Ігор Сергійович
12	Історія та культура України технікум-вуз	Лященко І.С.	залік	90/3	12	Ніколаєнко Марк Дмитрович
13	Іноземна мова технікум-вуз	Мірошниченко І.Г.	залік	60/2	13	Панін Данило Васильович

Рисунок 2. 30– Запит для відображення предмету для студента

Таким чином, ми отримали повноцінну базу даних, яка повністю функціонує та зберігає у собі потрібні дані.

## 2.4 Організація захисту бази даних

### 2.4.1 Рольове управління доступом

До захисту інформації можна віднести комплекс заходів, які спрямовані на забезпечення найважливіших аспектів інформаційної безпеки. Система називається безпечною, якщо вона, використовуючи відповідні апаратні і програмні засоби, управляє доступом до інформації так, що тільки належним чином авторизовані особи або ж діють від їхнього імені процеси отримують право читати, писати, створювати і видаляти інформацію.

У цій кваліфікаційній роботі вжиті такі організації захисту бази даних, як: рольове управління доступом, відключення віддаленого доступу, зміна пароля адміністратора, видалення значень за замовченням users/db (видалення всіх облікових записів, крім головного локального).

У програмі HeidiSQL існує меню «Управление пользователями и привилегиями», яке дозволяє додати ролі для користуванням сервером бази даних. Його можна побачити на рисунку 2.31.

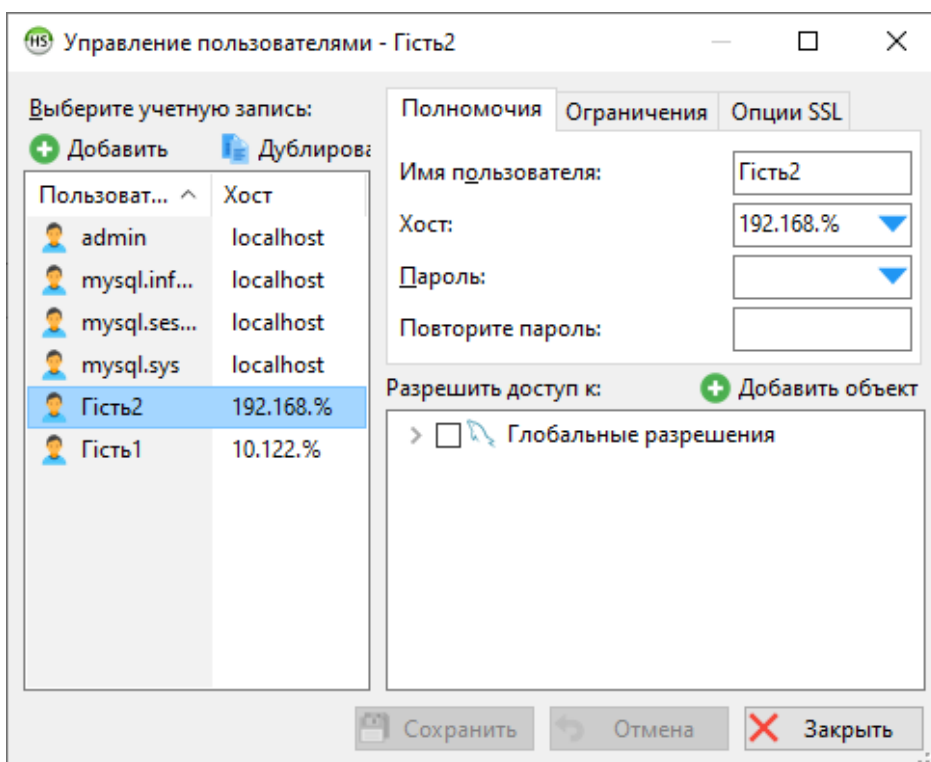


Рисунок 2.31 – Меню управління користувачами

Як ми бачимо, є обліковий запис адміністратора з існуючим паролем, який знає тільки він. Йому дозволен доступ до глобальних дозволів. В нього є необмежена кількість запитів до бази даних, її оновлень, підключень у годину.

Та є два облікових записи гостей, яким дозволено підключатися до бази даних лише з певних хостів. Кожному обліковому запису можна обмежити доступ на перегляд та редагування бази даних. Наприклад, на рисунках 2.32 та 2.33 можна побачити, що обліковий запис «Гість1» може тільки просматрювати таблицю «Студент» без можливості її редагування. Також можна встановити можливість редагування та створення нових таблиць.

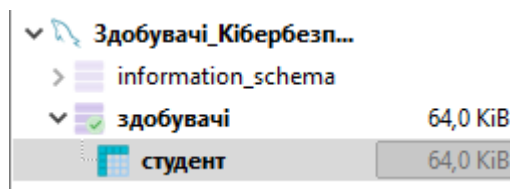


Рисунок 2.32 – Обмеження облікового запису «Гість»

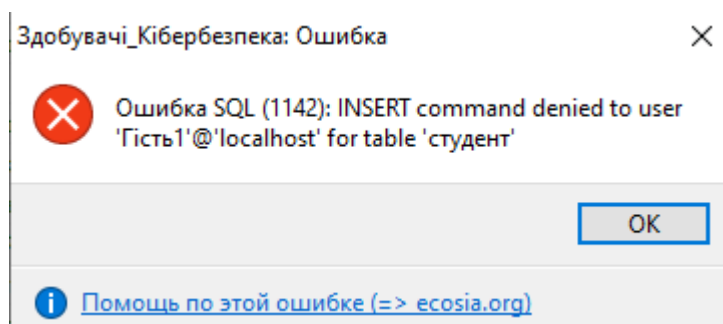


Рисунок 2.33 – Неможливість редагування бази даних

## 2.4.2 Відключення віддаленого доступу

Оскільки, згідно з початковими припущеннями захисту, база даних буде використовуватися тільки локально встановленими програмами, тому ми можемо вільно відключити прослуховування стандартного порту. Це обмежить можливість нападу базу даних MySQL підключеннями з інших хостів. Для того, щоб відключити віддалений доступ, потрібно лише з облікового запису

адміністратора виставити «Доступ только с сервера». Це можна побачити на рисунку 2.34.

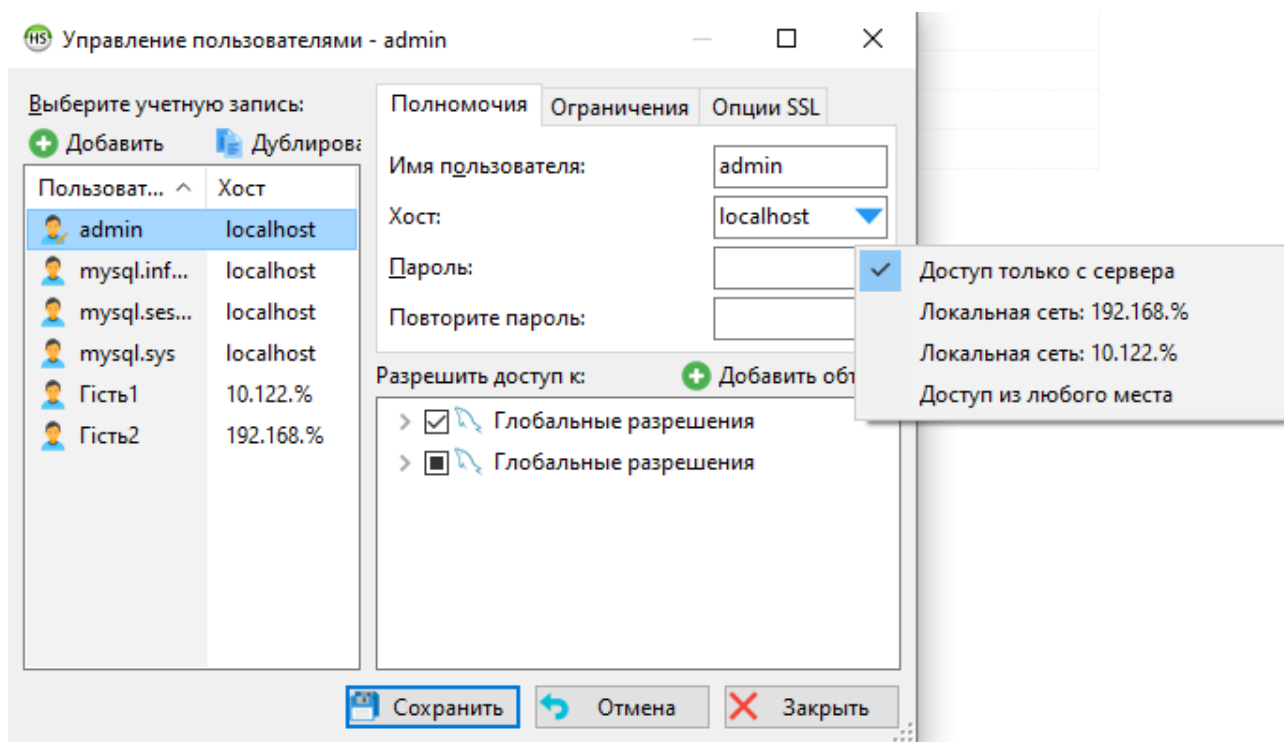


Рисунок 2.34 – Відключення віддаленого доступу

### 2.4.3 Зміна пароля адміністратора

Одним з найбільш важливих кроків у захисті бази даних є зміна пароля адміністратора бази даних, який є за промовчанням порожнім. Це особливо важливо, якщо на сервері працюють кілька користувачів. Щоб зробити це, ми повинні запустити HeidiSQL та у меню «Управление пользователями и привилегиями» обрати обліковий запис адміністратора. Можна використати сгенерований пароль, як приклад наведений на рисунку 2.35.

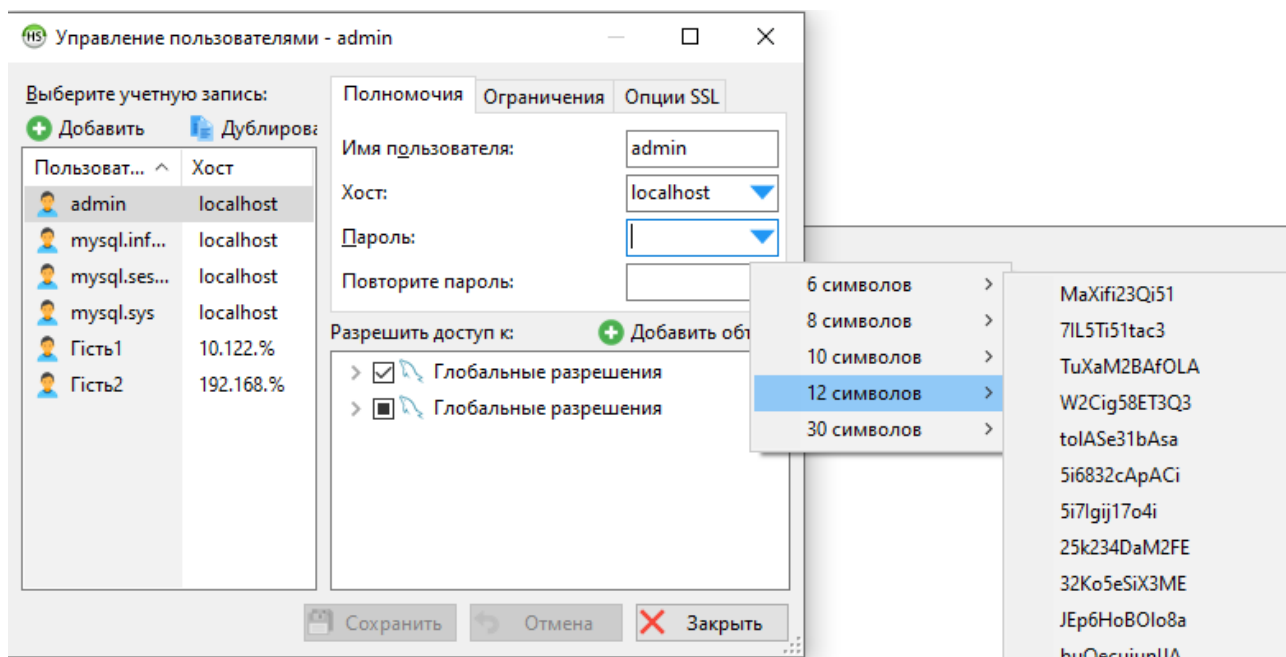


Рисунок 2.35 – Сгенерований пароль

Надійним рішенням буде придумати пароль самостійно та написати його до вкладки «Пароль», після чого підтвердити свій вибір. Після того, як ми знову зайдемо в програму, можемо побачити, що парольний захист працює. Це можна спостерігати на рисунку 2.36.

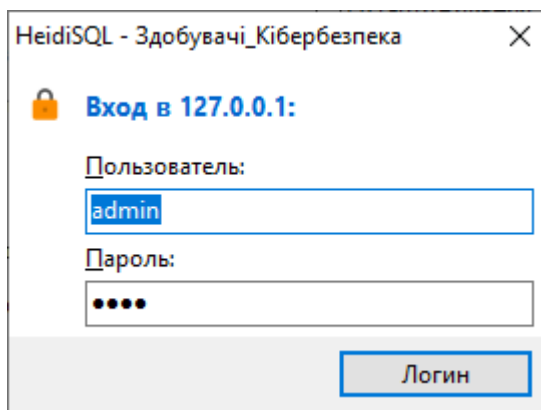


Рисунок 2.36 – Парольний захист

Для додаткового захисту можна встановити прапорець «Аутентифікація Windows». Ця можливість дає додатковий захист, тому що програма буде зчитувати ім'я хоста користувача комп'ютером.

#### 2.4.4 Видалення значень за замовченням

Щоб надійно захистити базу даних, потрібно видалити всі облікові записи, крім головного локального облікового запису (адміністратора). Це запобігає у нашій базі даних встановлення анонімних підключень, а також, віддалених підключень. Видалення облікових записів треба робити через запит DROP USER. Запит та результат наведені на рисунку 2.37.

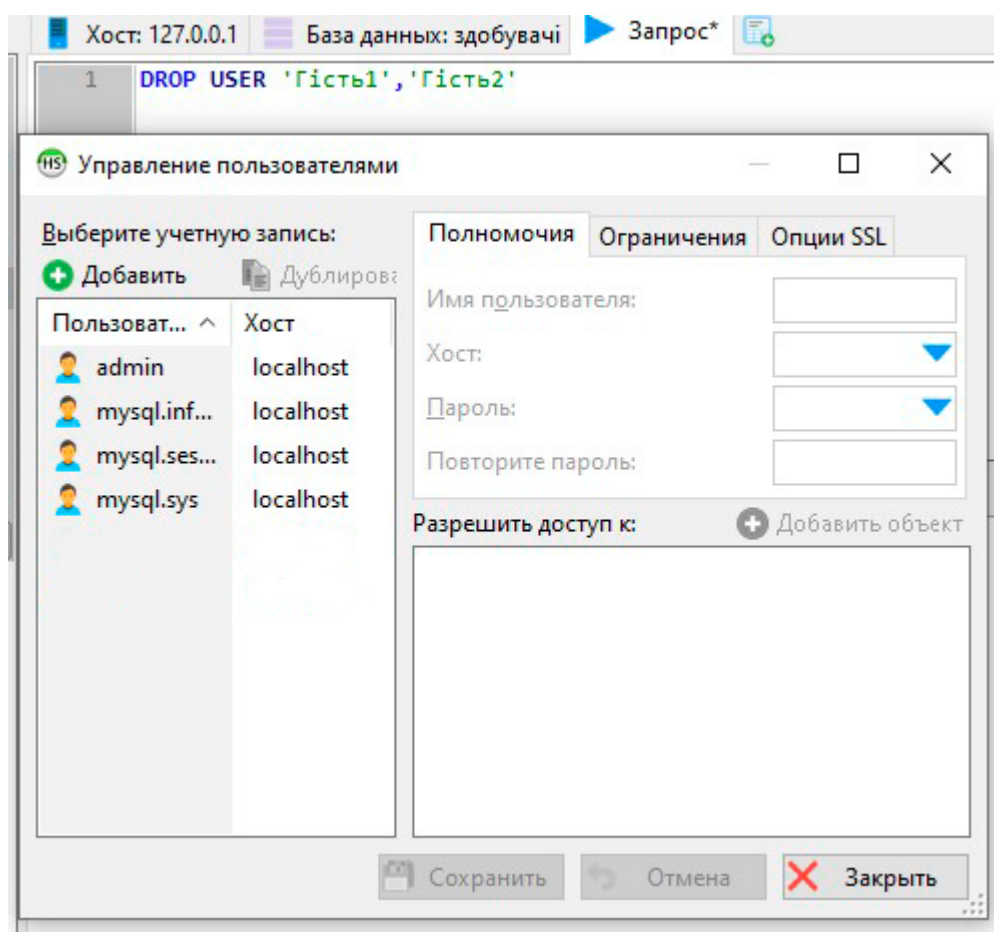


Рисунок 2.37 – Видалення усіх записів, окрім головного локального

#### 2.5 Підготовка самостійної роботи з дисципліни «Бази даних»

Пропонується самостійна робота з дисципліни «Бази даних» для здобувачів третього курсу спеціальності «Кібербезпека».

На сучасному етапі використано Microsoft Access, але можна рекомендувати HeidiSQL. Microsoft Access пропонує здійснювати ефективне управління базами даних. Це таблиці, файли текстового типу та дані інших популярних форматів.

Система управління базами даних (СУБД) значно спрощує процес. Завдяки цій програмі можна зберігати великі обсяги інформації, а також обробляти їх. Причому все це робиться у вигляді, який зручний користувачам. Він має високий рівень безпеки та надійності.

Теж саме гарантує програма HeidiSQL, але у порівнянні з Microsoft Access, вона більш складніша у освоєванні та має складний інтерфейс, з котрим буде важко розібратися в перші етапи створення бази даних.

Обидві програми можна рекомендувати до самостійної роботи. Вони рівнозначні у опрацюванні. Тому прийняте рішення підготувати самостійну роботу з захисту бази даних в HeidiSQL.

### **Самостійна робота з організації захисту бази даних в HeidiSQL**

**Тема:** Організація захисту бази даних в HeidiSQL.

**Мета:** 1. Ознайомитися з можливими способами захисту бази даних.

2. Придбати практичні навички в організації захисту таблиць БД.

## **1. Теоретичні відомості**

### **1.1 Способи захисту даних в HeidiSQL.**

В HeidiSQL реалізовані наступні способи захисту даних: парольний захист, рольове управління, видалення значень за замовченням users/db.

**1.2 Парольний захист** є дуже важливою частиною у захисті бази даних. В деяких випадках важливо змінити за замовченням ім'я облікового запису адміністратора (яке за замовченням завжди root), на будь-яке складніше значення. Така заміна ускладнить виконання «лобових» та «словникових» атак на пароль адміністратора. В цьому випадку, вторгнений повинен буде припустити не тільки пароль, а й насамперед ім'я облікового запису адміністратора. Парольний захист може використовуватися додатково для захисту на рівні користувача. У цьому випадку встановлювати парольний захист може користувач, що володіє правами адміністратора БД.

**Процедура встановлення парольного захисту виглядає так:**

1. Відкрити базу даних у HeidiSQL.

2. У меню обрати пункт «Управление пользователями и привилегиями».
3. Коли вікно «Управление пользователями и привилегиями» відкрилося, в полі з назвою «Имя пользователя» встановлюємо ім'я облікового запису.
4. У полі з назвою «Пароль» пишемо пароль. Пароль має містити в собі цифри та символи нижнього та верхнього регістрів.
5. У полі «Повторите пароль» повторюємо створений пароль.
6. Зберігти зміни.

При наступному запуску бази даних програма спитає ваше ім'я облікового запису та пароль.

**1.3 Рольове управління** дозволяє встановити обмеження та привілеї для кожного облікового запису. Таке управління дає додатковий захист, так як так виключаються можливості руйнування вмісту бази даних.

Процедура встановлення рольового управління включає:

1. Відкрити базу даних у HeidiSQL.
2. У меню обрати пункт «Управление пользователями и привилегиями».
3. Обрати обліковий запис гостя та у віконці «Разрешить доступ к: » натиснути на кнопку «Добавить объект».
4. У вікні вибору, обрати базу даних та/або таблицю, яку треба обрати, щоб тільки її бачив цей обліковий запис. Підтверджуємо вибір.
5. У меню «Управление пользователями и привилегиями» обрати SELECT та SHOW VIEW (рисунок 2.35). Це означає, що користувач зможе тільки просматрювати базу даних, без можливості її редагування.
6. Зберігти зміни.



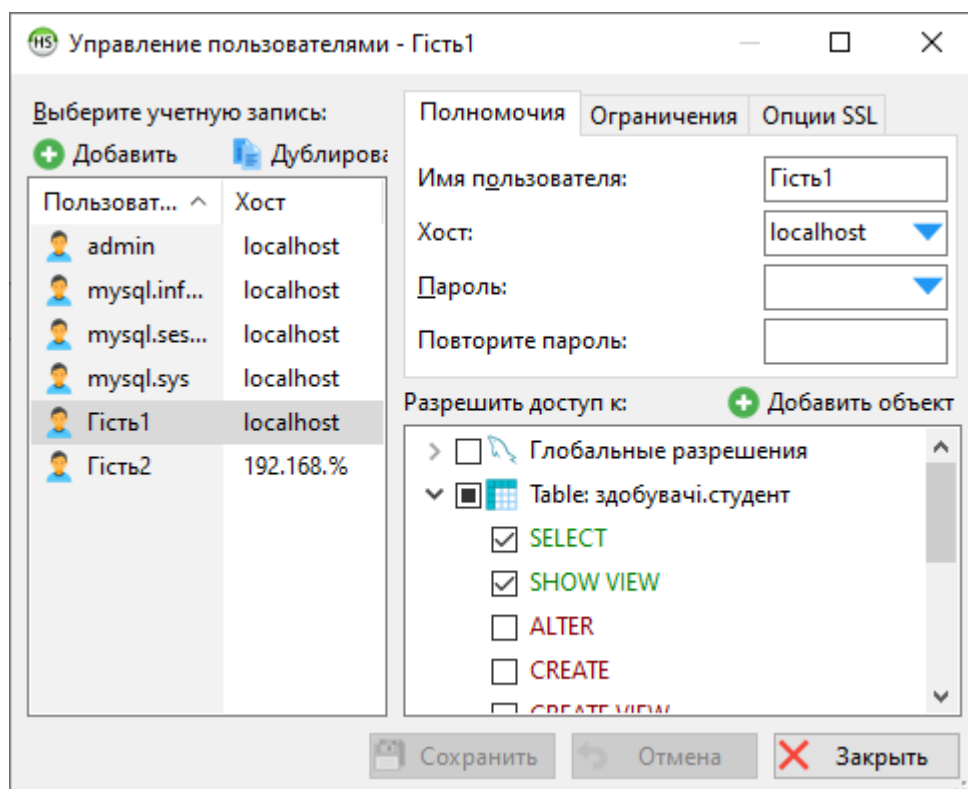


Рисунок 2.35 – Редагування можливостей облікового запису

**1.4 Видалення значень за замовченням users/db** має на увазі видалення усіх облікових записів, окрім головного локального. Це дає можливість адміністратору керувати обліковими записами, задля захисту від анонімних підключень.

Видалення значень за замовченням включає:

1. Відкрити базу даних у HeidiSQL.
2. Зайти у обліковий запис адміністратора.
3. Зайти у вкладку «Запрос».
4. Написати запит, який виглядає так: **DROP USER** 'ім'я облікового запису'.
5. Виповнити запит SQL кнопкою у контекстному меню.
6. Зайти у меню «Управление пользователями и привилегиями» та перевірити відсутність облікового запису.

## **2. Послідовність виконання роботи**

2.1 Встановити парольний захист таблиць БД.

2.2. Встановити обмеження редагування таблиці для першої облікової записи гостя: можливість перегляду без редагування будь-якої таблиці.

2.3. Встановити обмеження редагування таблиці для другої облікової записи гостя: можливість перегляду бази даних та створення атрибутів у будь-якій таблиці.

2.4. Видалити усі облікові записи, окрім адміністратора.

2.5. Оформити звіт до самостійної роботи, що містить: тема й мета самостійної роботи; теоретичні відомості; послідовність виконання самостійної роботи з поданням лістингів відповідних діалогових вікон; висновки по використанню кожного зі способів захисту.

2.6. Представити звіт до самостійної роботи та захистити її.

### **КОНТРОЛЬНІ ЗАПИТАННЯ**

1. Для чого треба захищати базу даних? Приклади.
2. Які бувають методи захисту бази даних у HeidiSQL?
3. Який метод захисту найефективніший на вашу думку? Пояснити чому.
4. Порядок установки парольного захисту у HeidiSQL.
5. Що можна дозволити та заборонити користувачу?

### **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Microsoft SQL Server [Електронний ресурс]. – Режим доступу: <https://www.techtarget.com/searchdatamanagement/definition/SQL-Server>;

2. Що таке MySQL? [Електронний ресурс]. – Режим доступу: <https://uk.education-wiki.com/5876083-what-is-mysql>;

3. Дистанційний курс з дисципліни «Бази даних», укладач доцент Пахомова В.М. [Електронний ресурс] – <https://lider.diit.edu.ua/course/view.php?id=341>.

## 2.6 Основні висновки

У другому розділі кваліфікаційної роботи розглянуті питання проектування, створення та обробка захищеної бази даних.

1. Результатом проектування стала база даних за методом НФ. У результаті проектування, отримані відношення: R1, R3, R4, R5, R6 та R7, які задовольняють вимогам БКНФ – знаходяться в третій нормальній формі, а їх первинні ключі не залежать від не ключових атрибутів. Тому спроектована база даних готова до створення програмно.

2. У результаті створення бази даних у програмі HeidiSQL, отримано базу даних, яка містить такі таблиці: Батько, Мати, Студент, Предмет, Бали\_за\_іспит та Рейтинг. Усі вони зв'язані між собою зовнішніми ключами, використовуючи унікальні ідентифікатори.

3. Після програмного створення бази даних, розглянуті організації захисту бази даних. До них віднесли: рольове управління доступом, відключення віддаленого доступу, зміна паролю адміністратора та видалення значень за замовченням. Усі ці методи використані по відношенню до існуючої бази даних, що показує найвищий рівень її захисту.

4. Була підготовлена та розроблена самостійна робота з дисципліни «Бази даних», яка містить у собі пояснення та завдання щодо встановлення таких методів захисту, як: парольний захист, рольове управління доступом, видалення значень за замовченням.

## ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

У даній кваліфікаційній роботі була розроблена та захищена реляційна база даних студентів спеціальності «Кібербезпека».

1. Виконаний огляд та аналіз предмета проектування. Серед існуючих додатків обрано HeidiSQL та MySQL. Вони обрані у зв'язку з тим, що MySQL – це стабільне, надійне та потужне рішення з розширеними функціями безпеки даних, масштабуванням за вимогою та високої ефективності. Для спрощеної роботи з MySQL йде графічний інтерфейс HeidiSQL.

2. Спроектовано базу даних здобувачів спеціальності «Кібербезпека» за методом нормальних форм. На основі робочого навчального плану першого курсу та даних про студентів наступних груп: КБ2111 (915); КБ2112 (916); КБ21130 (920Б). Дані про студентів представлені їх куратором Пахомовою В.М.

3. Створено у програмі HeidiSQL та MySQL базу здобувачів спеціальності «Кібербезпека», яка містить шість наступних таблиць: СТУДЕНТ, РЕЙТИНГ, ПРЕДМЕТ, МАТИ, БАТЬКО та БАЛИ\_ЗА\_СКЛАДАННЯ\_ІСПИТУ.

4. Виконано захист створеної бази даних з використанням наступних механізмів: рольове управління доступом, відключення віддаленого доступу, зміна паролю адміністратора та видалення значень за замовченням.

5. Підготовлена, розроблена та рекомендована самостійна робота з дисципліни «Бази даних» здобувачам першого (бакалаврського) курсу спеціальності «Кібербезпека», яка містить у собі пояснення та завдання щодо встановлення таких методів захисту, як: парольний захист, рольове управління доступом, видалення значень за замовченням.

## ПЕРЕЛІК ПОСИЛАНЬ

- 1 What is a Database? Definition, Meaning, Types with Example [Електронний ресурс]. – Режим доступу: <https://www.guru99.com/introduction-to-database-sql.html>;
- 2 Что такое реляционная база данных? [Електронний ресурс]. – Режим доступу: <https://aws.amazon.com/ru/relational-database/>;
- 3 Реляційна модель даних. Основна термінологія. Види ключів. Цілісність реляційних даних. Офіційний сайт Центральноукраїнського державного педагогічного університету. [Електронний ресурс]. – Режим доступу: <https://wiki.cuspu.edu.ua/index.php/>;
- 4 Бантюков С.Є., Бутенко В.М., Пчолін В.Г. Конспект лекцій з дисципліни “Інформаційні системи на залізничному транспорті”. – Харків: УкрДАЗТ, 2008. – Ч. 1. – 64 с.
- 5 Основні поняття моделі «сутність-зв'язок» [Електронний ресурс]. – Режим доступу: <https://tubushow.ru/uk/devices/model-sushchnost-svyaz-biblioteka-yuza-model-sushchnost-atribut-svyaz-er/>;
- 6 Сидоренко В.В., Константинова Л.В., Смірнов С.А. ТЗ8 Організація баз даних: Навчальний посібник. – Кропивницький: ЦНТУ, 2018. – 274 с.
- 7 Науково-технічна бібліотека Дніпровського національного університету залізничного транспорту імені академіка В. Лазаряна [Електронний ресурс]. – Режим доступу: [http://ecat.diit.edu.ua/cgi-bin/irbis64r\\_12/cgiirbis\\_64.exe?C21COM=F&I21DBN=DB2&P21DBN=DB2](http://ecat.diit.edu.ua/cgi-bin/irbis64r_12/cgiirbis_64.exe?C21COM=F&I21DBN=DB2&P21DBN=DB2);
- 8 What Is Microsoft Access and What Do You Use It For? [Електронний ресурс]. – Режим доступу: <https://www.dummies.com/article/technology/software/microsoft-products/access/what-is-microsoft-access-and-what-do-you-use-it-for-258024/>;
- 9 What is Oracle? [Електронний ресурс]. – Режим доступу: <https://www.javatpoint.com/what-is-oracle>;

10 Microsoft SQL Server [Електронний ресурс]. – Режим доступу: <https://www.techtarget.com/searchdatamanagement/definition/SQL-Server>;

11 InterBase FAQ [Електронний ресурс]. – Режим доступу: <https://www.embarcadero.com/ru/products/interbase/faq>;

12 Що таке MySQL? [Електронний ресурс]. – Режим доступу: <https://uk.education-wiki.com/5876083-what-is-mysql>;

13 HeidiSQL vs MySQL. Офіційний сайт Stackshare. [Електронний ресурс]. – Режим доступу: <https://stackshare.io/stackups/heidisql-vs-mysql>;

14 Дистанційний курс з дисципліни «Бази даних», укладач доцент Пахомова В.М. [Електронний ресурс] – <https://lider.diit.edu.ua/course/view.php?id=341>;

15 Падафа Н.Є., керівник Пахомова В.Н., Створення бази даних здобувачів спеціальності «Кібербезпека» та її захист, Молода академія 2022. Збірка тез доповідей. Всеукраїнська науково-технічна конференція студентів і молодих учених, Дніпро: УДУНТ, 2022, с. 145.