

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних
технологій, обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Хильман Станіслав Сергійович
(прізвище, ім'я, по батькові здобувача освіти)

УДК 04.65:378.147

КВАЛІФІКАЦІЙНА РОБОТА

Інформаційна система обліку успішності студентів закладу вищої освіти
(тема роботи)

122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

Хильман Станіслав Сергійович
(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи
Николук Ольга Миколаївна
д.е.н., професор

Житомир – 2026

Висновок кафедри комп'ютерних технологій і моделювання систем
за результатами попереднього захисту: допущений

Протокол засідання кафедри комп'ютерних технологій і моделювання систем
№ _____ від «_____» _____ 20____ р.

Завідувач кафедри комп'ютерних технологій і моделювання систем
к. пед. н., доцент _____ КОВАЛЬЧУК М. О.
(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)
«_____» _____ 2026 р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти Хильман Станіслав Сергійович захистив (ла)
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК
лаборант кафедри _____ КОРОЛЬЧУК В. В.
(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)

ЗМІСТ

Анотація.....	3
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	5
ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИБІР ЗАСОБІВ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	8
1.1 Аналіз інформаційної інфраструктури та вимог до бази даних закладу вищої освіти	8
1.2. Обґрунтування вибору системи управління базами даних.....	9
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	13
2.1 Розробка логічної структури та створення фізичних таблиць бази даних інформаційної системи	13
2.2. Оптимізація функціонування бази даних та налаштування індексів	23
2.3 Реалізація користувацького інтерфейсу інформаційної системи	24
РОЗДІЛ 3. ОРГАНІЗАЦІЯ СИСТЕМИ ЗАХИСТУ ДАНИХ ТА ОБСЛУГОВУВАННЯ.....	28
3.1 Аналіз структури користувачів та розробка ролівої моделі	28
3.2 Реалізація системи розмежування прав доступу	29
3.3 Регламентне обслуговування та резервне копіювання даних	30
ВИСНОВКИ	34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	35
ДОДАТКИ	3
ДОДАТОК А	4
ДОДАТОК Б	8
ДОДАТОК В.....	9

Анотація

Хильман С. С. Інформаційна система обліку успішності студентів закладу вищої освіти. – Кваліфікаційна робота на правах рукопису. Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 122 – «Комп’ютерні науки». – Поліський національний університет, Житомир, 2026.

Кваліфікаційна робота присвячена процесам проектування, адміністрування та захисту бази даних інформаційної системи обліку успішності студентів вищого навчального закладу.

У першому розділі проаналізовано середовище функціонування та інформаційну інфраструктуру бази даних університету, а також обґрунтовано вибір Microsoft SQL Server як оптимальної системи управління базами даних. У другому розділі розроблено структуру бази даних, створено таблиці, представлення та налаштовано систему моніторингу за допомогою dbForge Monitor. Також реалізовано оптимізацію функціонування бази даних шляхом створення та реорганізації індексів і налаштовано регламентне обслуговування (резервне копіювання). У третьому розділі проаналізовано структуру користувачів, реалізовано систему розмежування прав доступу на основі ролей та впроваджено асиметричне шифрування конфіденційних даних студентів і викладачів за допомогою сертифікатів. Практична цінність роботи полягає у створенні надійної, оптимізованої та захищеної бази даних, яка може бути інтегрована в інформаційну систему навчального закладу.

Ключові слова: база даних, інформаційна система, моніторинг, облік успішності, оптимізація.

SUMMARY

Khylman S.S. Information system for student performance tracking in a higher education institution. – Qualification work as a manuscript. Qualification work for a bachelor's degree in specialty 122 – Computer Sciences. – Polissia National University, Zhytomyr, 2026.

The qualification work is devoted to the processes of designing, administering, and protecting the database of the information system for tracking student performance in a higher educational institution.

The first chapter analyzes the operating environment and information infrastructure of the university's database, and justifies the choice of Microsoft SQL Server as the optimal database management system. In the second chapter, the database structure is developed, tables and views are created, and a monitoring system is configured using dbForge Monitor. The functioning of the database is optimized by creating and reorganizing indexes, and routine maintenance (backup) is configured. The third chapter analyzes the structure of users, implements a role-based access control system, and introduces asymmetric encryption of confidential student and teacher data using certificates. The practical value of the work lies in the creation of a reliable, optimized, and secure database that can be integrated into the information system of an educational institution.

Key words: database, database management system, encryption, information system, monitoring.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

SSMS – SQL Server Management Studio

СУБД – Система Управління Базами Даних

T-SQL – Transact-SQL

БД – база даних

ВНЗ – вищий навчальний заклад (університет)

MSSQL – Microsoft SQL

OEM - Oracle Enterprise Manager

ВСТУП

Актуальність теми дослідження. Актуальність теми полягає в необхідності підвищення ефективності обліку успішності студентів у закладах вищої освіти. Використання інформаційної системи дає змогу автоматизувати зберігання, обробку та аналіз даних про результати навчання, зменшити кількість помилок і прискорити роботу користувачів. Розробка такої системи сприятиме покращенню організації освітнього процесу та підвищенню якості управління навчальною діяльністю.

Мета і завдання роботи. Метою кваліфікаційної роботи є оптимізація процесу обліку успішності студентів закладу вищої освіти шляхом розробки інформаційної системи обліку успішності студентів закладу вищої освіти. Для досягнення поставленої мети було вирішено такі завдання:

1. Проаналізувати предметну область та визначити функціональні вимоги до бази даних системи обліку успішності.
2. Спроекувати інформаційну систему обліку успішності студентів закладу вищої освіти.
3. Розробити підсистему безпеки: впровадити рольове розмежування доступу та асиметричне шифрування конфіденційних даних.
4. Розробити інформаційну систему обліку успішності студентів закладу вищої освіти

Результати дослідження опубліковано у таких працях:

– Хильман С. С. Організація системи захисту даних та автоматизації обслуговування університетської бази даних. Зб. наук. пр. за мат-ми XI Міжнар. студентської конф. «Наука сьогодення: від досліджень до стратегічних рішень» (26.06.2026, м. Умань).

– Хильман С. С. Проектування та оптимізація бази даних інформаційної системи обліку успішності. Зб. наук. пр. за мат-ми XI Всеукр. студентської конф. «Експериментальні та теоретичні дослідження в контексті сучасної науки» (10.07.2026, м. Львів).

Предмет та об'єкт дослідження. Об'єктом дослідження є процес застосування методів проєктування, розробки, тестування інформаційної системи обліку успішності студентів закладу вищої освіти. Предметом дослідження є процеси організації збереження даних, моніторингу та захисту інформації в базі даних успішності студентів.

Методи дослідження. У роботі використано методи системного аналізу, методи проєктування реляційних баз даних, методи інфологічного моделювання, а також методи криптографічного захисту інформації.

Практичне значення отриманих результатів. Отримані результати можуть бути безпосередньо впроваджені в роботу інформаційно-обчислювального центру або деканату закладу вищої освіти. Створена база даних має високий рівень продуктивності завдяки оптимізованим індексам та забезпечує надійний захист персональних даних відповідно до сучасних стандартів безпеки.

Структура та обсяг роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 41 сторінка основного тексту та 20 сторінок додатків. Робота містить 27 рисунків та 11 таблиць. Список використаних джерел налічує 10 найменувань.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИБІР ЗАСОБІВ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

1.1 Аналіз інформаційної інфраструктури та вимог до бази даних закладу вищої освіти

Університетські установи відзначаються унікальними особливостями та великим обсягом інформації, яку слід зберігати, обробляти та забезпечувати доступність для різних учасників освітнього процесу. Під час адміністрування баз даних університету, важливо не лише враховувати технічні аспекти функціонування системи, але й ретельно аналізувати середовище, в якому вона діє, а також інформаційну інфраструктуру, яка визначає умови її роботи та взаємодії з іншими системами.

Першочерговим завданням є вивчення структури та організаційної моделі університетського середовища. Це включає розуміння внутрішньої ієрархії, функціональних підрозділів та їхніх взаємовідносин.

База даних повинна включати інформацію про студентів, викладачів, групу, дисципліни. Для коректної роботи в БД має бути встановлено певні обмеження, або умови, наприклад: при додаванні студента потрібно прив'язувати до нього спеціальність, групу і т. д. Для зручності користування в базі даних має бути можливість швидко переглядати інформацію.

Аналіз технічного середовища передбачає дослідження апаратних та програмних засобів, які використовуються для забезпечення роботи баз даних. Це включає серверне обладнання, мережеві з'єднання та застосування конкретних технологій, зокрема MS SQL Server.

На основі проведеного аналізу предметної області було визначено основних користувачів інформаційної системи та їх функціональні можливості. Основними акторами є студент, викладач та адміністратор (рис. 1.1). Студент має можливість переглядати результати навчання та розклад занять. Викладач здійснює авторизацію, перегляд інформації про студентів, внесення та редагування оцінок. Адміністратор відповідає за ведення довідників, керування користувачами, резервне копіювання та налаштування прав доступу.

Побудована діаграма прецедентів відображає основні функції інформаційної системи та взаємодію користувачів із нею.

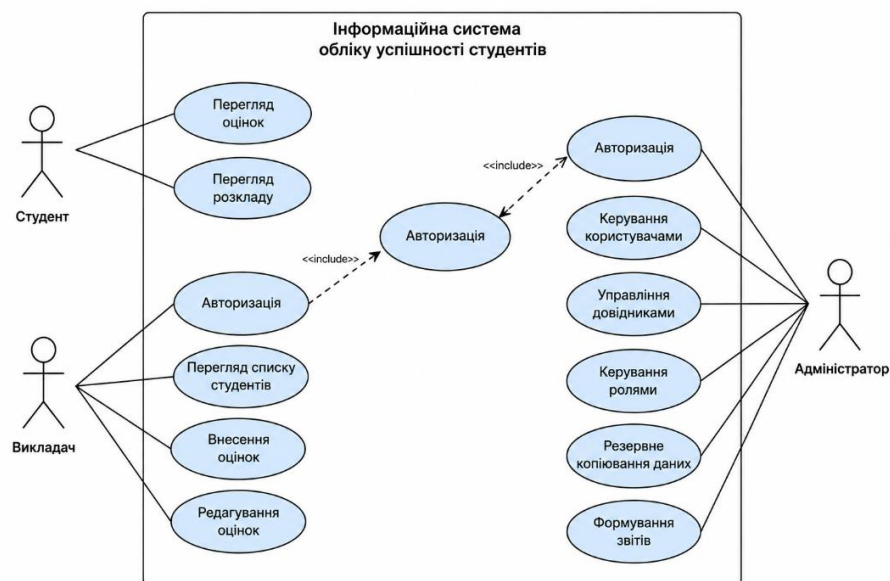


Рисунок 2.1 – Діаграма прецедентів

1.2. Обґрунтування вибору системи управління базами даних

Вибір підходящого засобу для організації збереження даних є стратегічно важливим завданням, оскільки від цього залежить ефективність роботи системи управління базами даних в університетському середовищі.

Перед вибором конкретного продукту, в даному розділі буде проведено аналіз таких продуктів, як Oracle, MS SQL Server та MySQL Server.

Порівнюватись СУБД будуть за такими ознаками:

- функціональність;
- масштабованість;
- вартість ліцензії;
- можливість інтеграції з іншими інформаційними системами університету.

Oracle Database

Oracle Database - це реляційна система управління базами даних, розроблена компанією Oracle Corporation. Ця система широко використовується

великими корпораціями та установами для зберігання, управління та обробки великих обсягів даних.

Oracle використовує мову запитів SQL (Structured Query Language) для взаємодії з базою даних.

Ціна Oracle Database може бути значною, особливо для підприємств або установ, які використовують повний набір його функцій. Ліцензійна модель включає різні пакети, залежно від потреб користувача.

Oracle Database є дуже функціональним середовищем. СУБД включає в себе багатий набір функцій, таких як оптимізований механізм кешування, робота з XML-документами, інтеграція з Java, продвинуті можливості аналітики, робота з геопросторовою інформацією та багато іншого.

СУБД є високомасштабованим рішенням, здатним впоратися з великими обсягами даних та високим навантаженням. Також Oracle відомий своєю високою рівнем безпеки та надійності. Він має інтегровані засоби захисту даних, механізми резервного копіювання та відновлення, а також засоби моніторингу та виявлення помилок.

Oracle Database добре інтегрується з іншими продуктами, а також має засоби для інтеграції з продуктами сторонніх вендорів. Це дозволяє створювати комплексні інформаційні системи, в яких база даних взаємодіє з іншими компонентами інфраструктури.

Microsoft SQL Server

Microsoft SQL Server - це система управління базами даних, розроблена корпорацією Microsoft. Вона використовується для зберігання та управління даними в різноманітних інформаційних системах.

MS SQL Server використовує мову запитів Transact-SQL (T-SQL), яка є розширенням стандартного SQL з додатковими функціями.

Ціна на MS SQL Server залежить від обраної редакції (Standard, Enterprise, Express) та виду ліцензійної моделі (перманентна, абонементна). Перманентні ліцензії можуть вимагати великих інвестицій.

СУБД включає в себе багатий функціонал, включаючи роботу з колекціями даних, аналітичні функції, підтримку інтегрованого зберігання та обробки XML-документів, а також засоби бізнес-аналітики.

MS SQL Server забезпечує хорошу маштабованість, включаючи можливість використання великої кількості фізичних та віртуальних ресурсів для обробки завдань з різних областей.

MS SQL Server визначається високою рівнем безпеки та надійністю. Засоби шифрування, моніторингу та аудиту, а також системи резервного копіювання забезпечують захист даних від втрат та несанкціонованого доступу.

MS SQL Server легко інтегрується з іншими продуктами Microsoft, такими як Microsoft Excel, Power BI, SharePoint, а також може взаємодіяти з різними іншими інформаційними системами через стандартні протоколи та інтерфейси.

MySQL Server

MySQL Server - це відкрите програмне забезпечення, реляційна система управління базами даних (RDBMS), яка широко використовується у різних галузях для зберігання та обробки даних.

MySQL використовує стандартну мову запитів SQL для взаємодії з базою даних. Він є безкоштовним програмним забезпеченням відкритого коду, що робить його вартісним рішенням для тих, хто шукає економічне вирішення для зберігання даних.

MySQL пропонує базовий, але відмінний функціонал для роботи з реляційними даними. Він включає в себе підтримку транзакцій, індексацію, тригери, хранилища процедур та інші стандартні можливості СУБД.

СУБД має хорошу маштабованість та може бути використаний як для невеликих проектів, так і для більших систем. MySQL забезпечує базові засоби забезпечення безпеки, такі як можливості автентифікації, авторизації та шифрування, а також має системи резервного копіювання для забезпечення надійності даних.

Порівняння всіх СУБД наведено в додатку А.

Висновки з першого розділу

У першому розділі кваліфікаційної роботи було проведено аналіз середовища функціонування та інформаційної інфраструктури баз даних в університетських установах. Розглянуто та порівняно основні аспекти адміністрування баз даних, зокрема архітектурні особливості, вимоги до продуктивності та безпеки даних.

Даний розділ допоміг вибрати засоби для збереження даних визначається не лише технічними характеристиками, але й конкретними вимогами та завданнями університетського середовища. Три системи мають певні переваги та обмеження щодо продуктивності та безпеки, і вибір має бути здійснений з урахуванням конкретних потреб організації.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Розробка логічної структури та створення фізичних таблиць бази даних інформаційної системи

Після аналізу предметної області було визначено структуру інформаційної системи обліку успішності студентів. Основою її функціонування є база даних, яка забезпечує зберігання інформації про студентів, викладачів, дисципліни, навчальні групи, результати навчання та розклад занять. Перед створенням таблиць було визначено взаємозв'язки між основними об'єктами системи, що забезпечує цілісність даних та підтримує реалізацію основних функцій інформаційної системи.

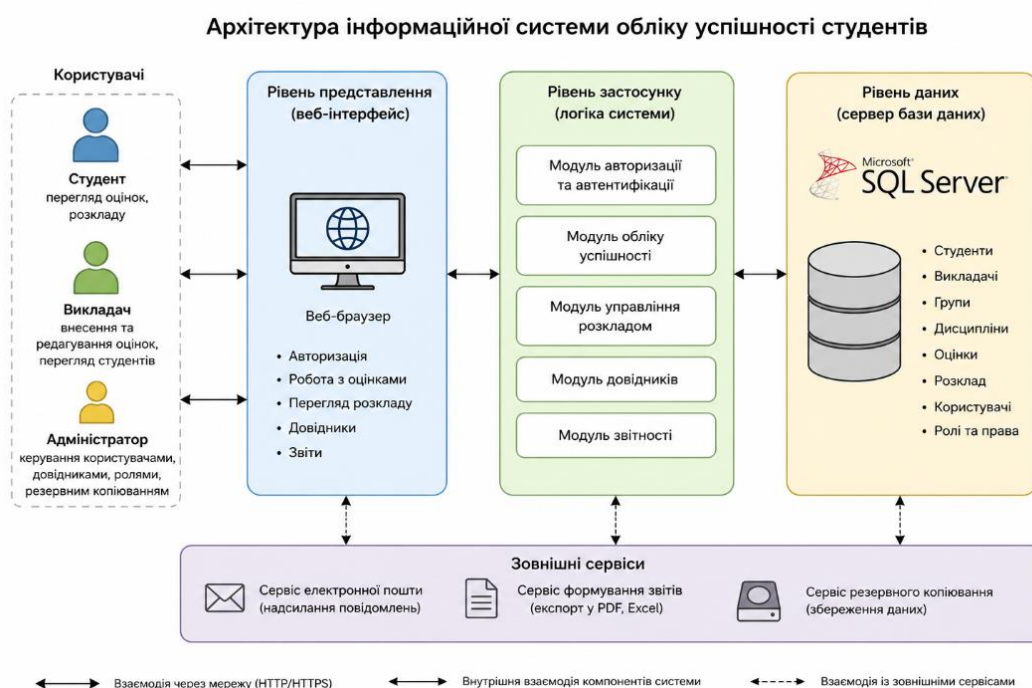


Рисунок 2.1 – Архітектура інформаційної системи

Для створення бази даних використовую запит:

CREATE DATABASE [База даних університету]

До бази даних університету буде включено такі таблиці:

1. Факультет
2. Кафедра
3. Спеціальність
4. Група

5. Студенти
6. Викладачі
7. Дисципліни
8. Рейтинг
9. Користувачі
10. Розклад

Таблиця «Факультет» призначена для збереження даних про факультети, які знаходяться в університеті. Структура таблиці наведена нижче:

Таблиця 2.1

Структура таблиці «Факультет»

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Унікальний ідентифікатор факультету
Назва факультету	Varchar(255)	-	-	Назва факультету
Рік заснування	Date	-	-	Рік заснування факультету
Електронна пошта	Nvarchar	-	-	Унікальна електронна пошта факультету

Лістинг запиту на створення таблиці:

```
CREATE TABLE Факультет (
    Id INT PRIMARY KEY IDENTITY(1,1),
    [Назва факультету] VARCHAR(255) NOT NULL,
    [Рік заснування] DATE DEFAULT GETDATE() CHECK ([Рік заснування] <=
GETDATE()),
    [Електронна пошта] NVARCHAR(35) UNIQUE
);
```

Таблиця «Кафедра» призначена для збереження даних про кафедри, які відносяться до певних факультетів. Структура таблиці наведена нижче:

Таблиця 2.2

Структура таблиці «Кафедра»

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Унікальний ідентифікатор кафедри
Id факультету	Int	-	+	Ідентифікатор факультету
Назва кафедри	Varchar(255)	-	-	Назва кафедри
Керівник	Varchar(60)	-	-	ПІБ керівника кафедри

Таблиця «Спеціальність» призначена для збереження даних про спеціальності, на яких вчаться студенти. Структура таблиці наведена нижче:

Таблиця 2.3

Структура таблиці «Спеціальність»

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Унікальний ідентифікатор спеціальності
Id кафедри	Int	-	+	Ідентифікатор кафедри

Продовження таблиці 2.3

Назва спеціальності	Varchar(255)	-	-	Назва спеціальності
Код спеціальності	Varchar(15)	-	-	Код, що ідентифікує спеціальність

Лістинг запиту на створення таблиці:

```
CREATE TABLE Спеціальність (
    Id INT PRIMARY KEY IDENTITY(1,1),
    [Id кафедри] INT,
    [Назва спеціальності] VARCHAR(255) NOT NULL,
    [Код спеціальності] VARCHAR(15),
    FOREIGN KEY ([Id кафедри]) REFERENCES Кафедра(Id)
);
```

Таблиця «Група» призначена для збереження даних про групи, до яких відносяться студенти. Структура таблиці наведена нижче:

Таблиця 2.4

Структура таблиці «Група»

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Унікальний ідентифікатор групи
Id спеціальності	Int	-	+	Ідентифікатор спеціальності
Назва групи	Varchar(50)	-	-	Назва групи
Рік формування	Date	-	-	Рік формування групи

Таблиця «Студенти» призначена для збереження даних про студентів, які навчаються в університеті. Структура таблиці наведена у табл. 2.5.

Таблиця 2.5

Структура таблиці «Студенти»

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Унікальний ідентифікатор студента
Ім'я	Varchar(50)	-	-	Ім'я студента
Прізвище	Varchar(50)	-	-	Прізвище студента
По-батькові	Varchar(50)	-	-	По-батькові студента
Ступінь	Varchar(30)	-	-	Рівень ступіня (бакалавр, магістр)
Тип навчання	Varchar(30)	-	-	Тип навчання (бюджет, контракт)
Статус	Varchar(30)	-	-	Статус студента (активний, відрахований, відпустка)
Дата народження	Date	-	-	Дата народження студента
Адреса проживання	Varchar(255)	-	-	Адреса, за якою проживає студент
Номер телефона	Varchar(20)	-	-	Номер телефона студента
Id групи	Int	-	+	Ідентифікатор групи

Таблиця «Викладачі» призначена для збереження даних про викладачів, які працюють в університеті. Структура таблиці наведена нижче:

Таблиця 2.6

Структура таблиці «Викладачі»

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Унікальний ідентифікатор викладача
Ім'я	Varchar(50)	-	-	Ім'я викладача
Прізвище	Varchar(50)	-	-	Прізвище викладача
По-батькові	Varchar(50)	-	-	По-батькові викладача
Дата народження	Date	-	-	Дата народження викладача
Науковий ступінь	Varchar(50)	-	-	Науковий ступінь викладача
Стаж викладання	Varchar(50)	-	-	Стаж викладання (у роках)
Номер телефона	Varchar(20)	-	-	Номер телефона викладача
Дата прийому	Date	-	-	Дата прийому на роботу в університет

Таблиця «Дисципліни» призначена для збереження даних про дисципліни, які вивчають студенти. Структура таблиці наведена у табл. 2.7.

Таблиця 2.7

Структура таблиці «Дисципліни»

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Унікальний ідентифікатор дисципліни
Назва	Varchar(255)	-	-	Назва дисципліни
Id викладача	Int	-	+	Ідентифікатор викладача
Тип	Varchar(50)	-	-	Тип дисципліни (загальна, спеціалізована)
Кількість кредитів	Int	-	-	Кількість кредитів, пов'язаних з дисципліною
Кількість годин на семестр	Int	-	-	Кількість годин виведених на дисципліну в університеті

Таблиця «Рейтинг» призначена для збереження даних про успішність студентів, які навчаються в університеті. Структура таблиці наведена у табл. 2.8.

Таблиця 2.8

Структура таблиці «Рейтинг»

Назва	Тип даних	ПК	ЗК	Опис поля
Id студента	Int	+	+	Ідентифікатор студента
Id дисципліни	Int	+	+	Ідентифікатор дисципліни
Тип контролю	Varchar(50)	-	-	Тип контролю (залік, екзамен)
Оцінка	Int	-	-	Оцінка студента в межах від 0 до 100

Таблиця «Розклад» призначена для зберігання даних про навчальний процес. Структура таблиці наведена у табл. 2.9.

Таблиця 2.9

Структура таблиці «Розклад»

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Унікальний ідентифікатор
Id дисципліни	Int	-	+	Ідентифікатор дисципліни
Id групи	Int	-	+	Ідентифікатор групи
Id викладача	Int	-	+	Ідентифікатор викладача
День тижня	Varchar(20)	-	-	День тижня
Номер тижня	Varchar(20)	-	-	Номер тижня (парний, непарний)
Тип	Varchar(20)	-	-	Тип заняття (лекція, лабораторна, практика)
Початок заняття	Time	-	-	Час початку заняття
Кінець заняття	Time	-	-	Час закінчення заняття
Тривалість заняття	Int	-	-	Тривалість заняття
Аудиторія	Varchar(20)	-	-	Номер аудиторії, в якій проводиться заняття

Таблиця «Користувачі» призначена для зберігання логінів та паролів всіх користувачів, які користуються базою даних або іншими ресурсами університету. Структура таблиці наведена нижче:

Таблиця 2.10

Структура таблиці «Користувачі»

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор користувача
ПІБ	Varchar(100)	-	-	ПІБ користувача
Вид користувача	Varchar(50)	-	-	Вид користувача (студент, викладач, адміністратор, технік)
Електронна пошта	Varchar(255)	-	-	Створена електронна пошта користувача
Пароль електронної пошти	Varchar(255)	-	-	Пароль для електронної пошти
Логін	Varchar(50)	-	-	Логін на освітньому порталі університету
Пароль на порталі	Varchar(255)	-	-	Пароль на освітньому порталі університету
Дата створення	Date	-	-	Дата створення облікового запису користувача

Щоб переглянути список всіх створених таблиць в базі даних потрібно використовувати запит:

```
SELECT @@Servername AS ServerName ,
TABLE_CATALOG ,
TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE'
ORDER BY TABLE_NAME;
```

В результаті виконання даного запиту ми бачимо ім'я сервера, ім'я бази даних та назву самої таблиці:

	ServerName	TABLE_CATALOG	TABLE_NAME
1	DESKTOP-R1Q87E0	База даних університету	sysdiagrams
2	DESKTOP-R1Q87E0	База даних університету	Викладачі
3	DESKTOP-R1Q87E0	База даних університету	Група
4	DESKTOP-R1Q87E0	База даних університету	Дисципліни
5	DESKTOP-R1Q87E0	База даних університету	Кафедра
6	DESKTOP-R1Q87E0	База даних університету	Користувачі
7	DESKTOP-R1Q87E0	База даних університету	Рейтинг
8	DESKTOP-R1Q87E0	База даних університету	Розклад
9	DESKTOP-R1Q87E0	База даних університету	Спеціальність
10	DESKTOP-R1Q87E0	База даних університету	Студенти
11	DESKTOP-R1Q87E0	База даних університету	Факультет

Рисунок 2.1 – Результат виведення всіх створених таблиць в БД

Одним із основних процесів інформаційної системи є внесення оцінок викладачем. Діаграма послідовності демонструє порядок взаємодії між користувачем, інформаційною системою та базою даних під час виконання цієї операції (рис. 2.2). Після успішної авторизації викладач відкриває журнал успішності, вводить оцінку, після чого система передає відповідний SQL-запит до бази даних та повідомляє користувача про успішне збереження інформації.

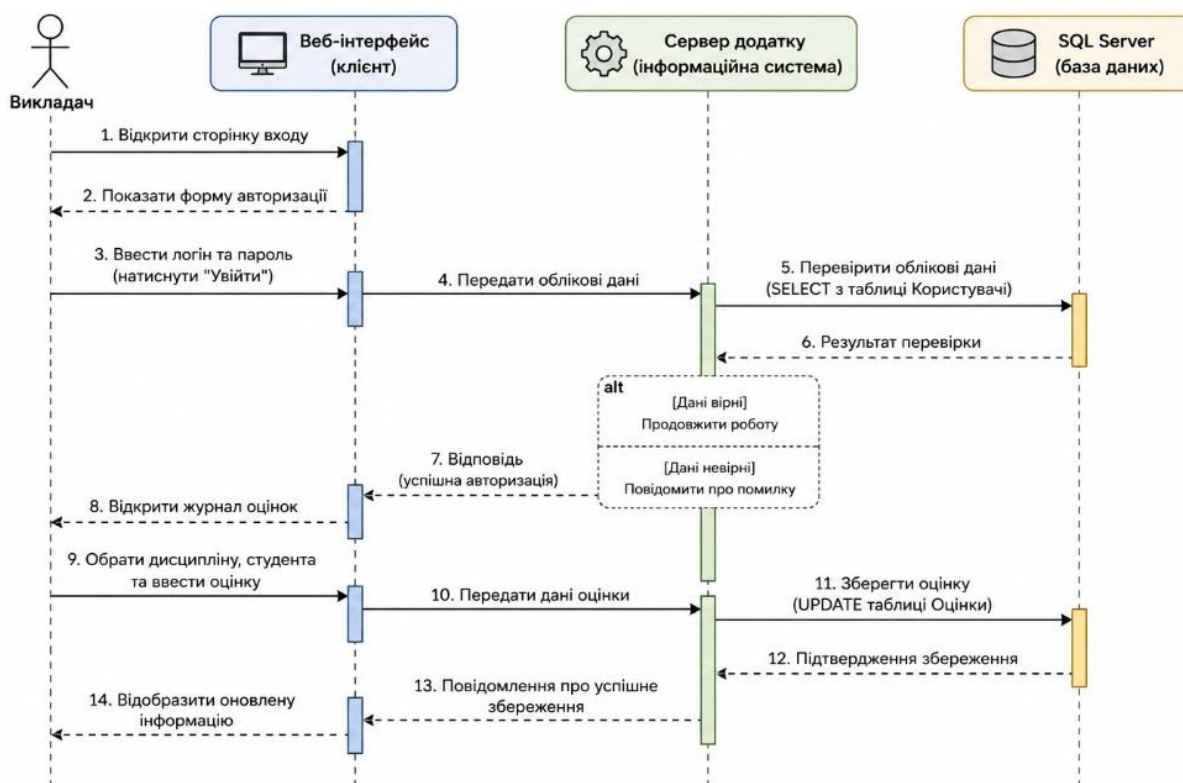


Рисунок 2.2 – Діаграма послідовності внесення оцінки викладачем
 Переглянемо створену діаграму бази даних:

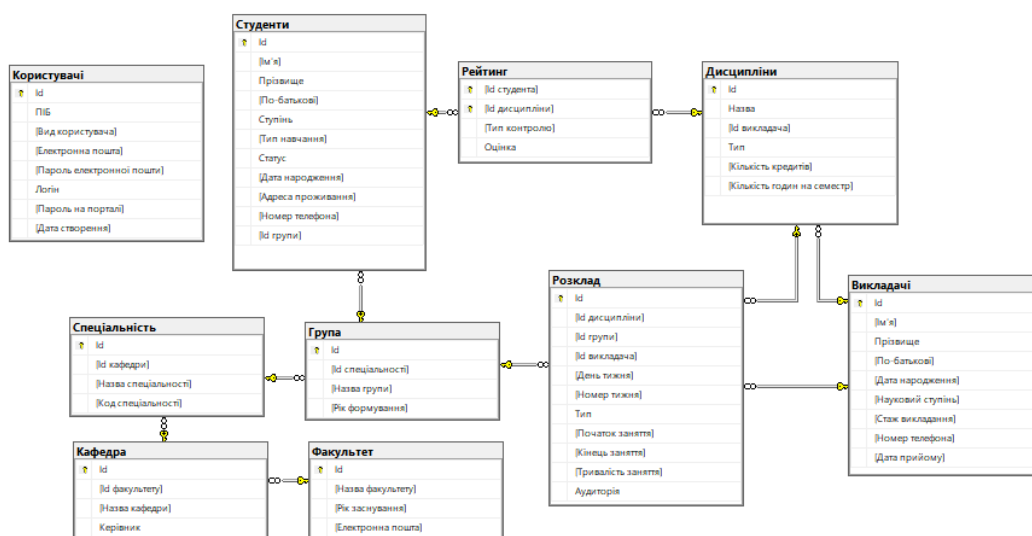


Рисунок 2.3 – Діаграма бази даних

Наступним кроком буде заповнення таблиць. Для заповнення таблиці «Групи» використовуємо запит:

```
INSERT INTO Група ([Id спеціальності], [Назва групи], [Рік формування])
VALUES
```

```
(1, 'ПМ-1', '2022-09-01'),
(2, 'КН-21-1', '2021-09-01'),
(2, 'КН-22-1', '2022-09-01'),
(3, 'ІПЗ-21-1', '2021-09-01'),
```

```
(3, 'ІПЗ-22-1', '2022-09-01'),
(3, 'ІПЗ-22-2', '2022-09-01'),
(4, 'СП-21-1', '2021-09-01'),
(5, 'ІСТ-21-1', '2021-09-01'),
(5, 'ІСТ-22-1', '2022-09-01'),
(6, 'М-1', '2022-09-01'),
(7, 'СТ-1', '2022-09-01'),
(8, 'КБ-21-1', '2021-09-01'),
(8, 'КБ-22-1', '2022-09-01'),
(9, 'КІ-21-1', '2021-09-01'),
(9, 'КІ-22-1', '2022-09-01');
```

Для заповнення таблиці «Спеціальність» використовуємо запит:

```
INSERT INTO Спеціальність ([Id кафедри], [Назва спеціальності], [Код спеціальності])
VALUES
```

```
(1, 'Прикладна математика', '113'),
(2, 'Комп'ютерні науки', '122'),
(2, 'Інженерія програмного забезпечення', '121'),
(3, 'Системне програмування', '120'),
(3, 'Інформаційні системи та технології', '126'),
(4, 'Математика', '111'),
(4, 'Статистика', '112'),
(5, 'Кібербезпека', '125'),
(5, 'Ком'ютерна інженерія', '123');
```

Після заповнення таблиць потрібно перевірити чи дані були коректно додані до таблиці та об'єднані. Для перевірки було створено вибірки на виведення даних з таблиці.

Вибірка інформації про студентів включає в себе також інформацію про групу та спеціальність студента. Результат виконання даної вибірки показаний на рисунку 2.4.

	ID Студента	Ім'я	Прізвище	По-батькові	Група	Спеціальність
1	1	Іван	Петров	Олександрович	ПМ-1	Прикладна математика
2	2	Олена	Сидоренко	Ігорівна	ПМ-1	Прикладна математика
3	3	Максим	Іванов	Васильович	ПМ-1	Прикладна математика
4	4	Анна	Коваленко	Андріївна	ПМ-1	Прикладна математика
5	5	Віталій	Мельник	Олегович	ПМ-1	Прикладна математика
6	6	Людмила	Павленко	Сергіївна	ПМ-1	Прикладна математика
7	7	Владислав	Григоренко	Ігорович	ПМ-1	Прикладна математика
8	8	Тетяна	Ковальчук	Миколаївна	ПМ-1	Прикладна математика
9	9	Ігор	Семенов	Олександрович	КН-21-1	Комп'ютерні науки
10	10	Олексій	Іванов	Віталійович	КН-21-1	Комп'ютерні науки
11	11	Катерина	Петренко	Анатоліївна	КН-21-1	Комп'ютерні науки
12	12	Денис	Захарчук	Андрійович	КН-21-1	Комп'ютерні науки
13	13	Наталія	Сидорова	Володимирівна	КН-21-1	Комп'ютерні науки
14	14	Артем	Гриценко	Анатолійович	КН-21-1	Комп'ютерні науки
15	15	Ірина	Павлюк	Володимирівна	КН-21-1	Комп'ютерні науки
16	16	Віктор	Ігнат'єв	Петрович	КН-21-1	Комп'ютерні науки
17	17	Ольга	Коваленко	Михайлівна	КН-22-1	Комп'ютерні науки

Рисунок 2.4 – Результат виконання вибірки

Вибірка розкладу включає в себе інформацію дисципліну, викладача та групу. Результат виконання даної вибірки показаний на рисунку 2.5:

	ID Заняття	Дисципліна	Викладач	Група	День тижня	Номер тижня	Тип	Початок заняття	Кінець заняття	Тривалість заняття	Аудиторія
1	1	Основи програмування	Іван Петренко	ПМ-1	Понеділок	Парний	Лекція	08:30:00.0000000	10:00:00.0000000	90	101
2	2	Диференціальні рівняння	Олександр Сидоренко	ПМ-1	Понеділок	Парний	Лекція	10:15:00.0000000	11:45:00.0000000	90	102
3	3	Основи програмування	Марія Іваненко	ПМ-1	Вівторок	Парний	Практика	08:30:00.0000000	10:00:00.0000000	90	201
4	4	Математичне моделювання	Тетяна Ковальчук	ПМ-1	Вівторок	Парний	Практика	10:15:00.0000000	11:45:00.0000000	90	202
5	5	Математичний аналіз	Олександр Сидоренко	ПМ-1	Середа	Парний	Лекція	08:30:00.0000000	10:00:00.0000000	90	103
6	6	Алгоритми та структури даних	Андрій Захарчук	ПМ-1	Середа	Парний	Лекція	10:15:00.0000000	11:45:00.0000000	90	104
7	7	Теорія ймовірностей	Марія Іваненко	ПМ-1	Четвер	Парний	Практика	08:30:00.0000000	10:00:00.0000000	90	203
8	8	Операційні системи	Оксана Литвиненко	ПМ-1	П'ятниця	Парний	Практика	10:15:00.0000000	11:45:00.0000000	90	204
9	9	Основи програмування	Іван Петренко	ПМ-1	Понеділок	Непарний	Лекція	08:30:00.0000000	10:00:00.0000000	90	101
10	10	Диференціальні рівняння	Олександр Сидоренко	ПМ-1	Понеділок	Непарний	Лекція	10:15:00.0000000	11:45:00.0000000	90	102
11	11	Основи програмування	Марія Іваненко	ПМ-1	Вівторок	Непарний	Практика	08:30:00.0000000	10:00:00.0000000	90	201
12	12	Математичне моделювання	Тетяна Ковальчук	ПМ-1	Вівторок	Непарний	Практика	10:15:00.0000000	11:45:00.0000000	90	202
13	13	Математичний аналіз	Олександр Сидоренко	ПМ-1	Середа	Непарний	Лекція	08:30:00.0000000	10:00:00.0000000	90	103
14	14	Алгоритми та структури даних	Андрій Захарчук	ПМ-1	Середа	Непарний	Лекція	10:15:00.0000000	11:45:00.0000000	90	104

Рисунок 2.5 – Результат виконання вибірки

Наступним кроком буде створення декількох представлень.

Перше представлення використовується для виведення кількості студентів у кожній групі. Результат виконання представлення наведений на рисунку 2.6.

	Назва групи	Кількість студентів
1	ІПЗ-21-1	16
2	ІПЗ-22-1	16
3	ІПЗ-22-2	16
4	ІСТ-21-1	16
5	ІСТ-22-1	16
6	КБ-21-1	24
7	КБ-22-1	16
8	КІ-21-1	16
9	КІ-22-1	16
10	КН-21-1	16
11	КН-22-1	16
12	М-1	24
13	ПМ-1	16
14	СП-21-1	16
15	СТ-1	24

Рисунок 2.6 – Результат представлення «Кількість студентів у групах»

Друге представлення використовується для виведення студентів та їх оцінок. Результат виконання представлення наведений на рисунку 2.7.

	ПІБ Студента	Дисципліна	Оцінка
1	Іван Петров	Основи програмування	Зараховано
2	Іван Петров	Математичний аналіз	92
3	Іван Петров	Теорія ймовірностей	Зараховано
4	Іван Петров	Проектування програмного забезпечення	85
5	Олена Сидоренко	Основи програмування	Зараховано
6	Олена Сидоренко	Математичний аналіз	78
7	Олена Сидоренко	Теорія ймовірностей	Зараховано
8	Олена Сидоренко	Проектування програмного забезпечення	88
9	Максим Іванов	Основи програмування	Зараховано
10	Максим Іванов	Математичний аналіз	90
11	Максим Іванов	Теорія ймовірностей	Зараховано
12	Максим Іванов	Проектування програмного забезпечення	76
13	Анна Коваленко	Основи програмування	Зараховано
14	Анна Коваленко	Математичний аналіз	85
15	Анна Коваленко	Теорія ймовірностей	Зараховано
16	Анна Коваленко	Проектування програмного забезпечення	78
17	Віталій Мельник	Основи програмування	Зараховано
18	Віталій Мельник	Математичний аналіз	87
19	Віталій Мельник	Теорія ймовірностей	Зараховано
20	Віталій Мельник	Проектування програмного забезпечення	94

Рисунок 2.7 – Результат виконання представлення «Студенти та оцінки»

Для перегляду всіх створених представлень можна використовувати даний запит:

```

SELECT @@Servername AS ServerName ,
DB_NAME() AS DBName ,
Name AS ViewName ,
create_date
FROM sys.Views
ORDER BY Name

```

В результаті виконання даного запиту ми можемо побачити створені запити. Результат виведений у вигляді таблиці показано на рисунку 2.8.

	ServerName	DBName	ViewName	create_date
1	DESKTOP-R1Q87E0	База даних університету	Кількість студентів у групах	2023-12-25 16:02:42.107
2	DESKTOP-R1Q87E0	База даних університету	Студенти та оцінки	2023-12-25 16:02:51.147

Рисунок 2.8 – Виведення створених представлень в БД

2.2. Оптимізація функціонування бази даних та налаштування індексів

Оптимізація бази даних - це процес налаштування та покращення продуктивності бази даних для забезпечення ефективного та швидкого доступу до даних. Цей процес включає в себе ряд кроків і стратегій, спрямованих на поліпшення використання ресурсів, зменшення часу виконання запитів та підвищення загальної ефективності системи.

Для оптимізації бази даних можна вдосконалити SQL-запити. Потрібно уникати великих вибірок даних, використовувати індексацію та оператори JOIN, WHERE.

Індекси дозволяють базі даних прискорити процес пошуку конкретних записів в таблиці, зменшуючи кількість рядків, які потрібно переглядати. Більшість індексів використовують структуру двійкового дерева (рис. 2.9) для ефективного організованого зберігання та пошуку даних.

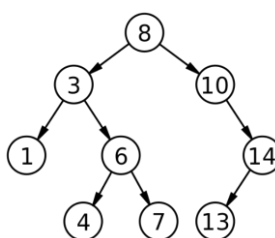


Рисунок 2.9 – Двійкове дерево

Індекс може бути унікальним або складним. Унікальний індекс не дозволяє повторювані значення для індексованих полів. Складний індекс

створюється для кількох стовпців, що дозволяє оптимізувати запити, які фільтрують або сортують за цими стовпцями.

В базі даних ВНЗ найчастіше будуть використовуватись таблиці «Студенти», «Розклад» та «Рейтинг». В цих таблицях зберігається найбільше даних. Для цих таблиць потрібно створити індекси. Результат виконання запитів показаний на рисунках 2.10, 2.11.

	index_type_desc	page_count	record_count	avg_page_space_used_in_percent	avg_fragment_size_in_pages
1	CLUSTERED INDEX	5	264	84,5861131702496	2,5
2	CLUSTERED INDEX	1	5	0,778354336545589	1
3	NONCLUSTERED INDEX	1	264	45,6387447491969	1

Рисунок 2.10 – Список індексів в таблиці «Студенти»

	IndexName	IndexType	ColumnName
1	PK__Рейтинг__3FF6AA70C5C55830	CLUSTERED	Id студента
2	PK__Рейтинг__3FF6AA70C5C55830	CLUSTERED	Id дисципліни
3	IX_Rating_Type	NONCLUSTERED	Тип контролю

Рисунок 2.11 – Список індексів в таблиці «Рейтинг»

З часом дані в індексі перетворюються на фрагментовані.

Фрагментація може призвести до погіршення продуктивності запитів, тому її важливо мінімізувати або управляти.

Для управління та того, щоб пришвидшити роботу системи можна перебудувати та реорганізувати індекси. Результат реорганізації показана на рисунку 2.12.

	index_type_desc	avg_fragment_size_in_pages
1	CLUSTERED INDEX	1,5
2	CLUSTERED INDEX	1
3	NONCLUSTERED INDEX	2
4	NONCLUSTERED INDEX	1

Рисунок 2.12 – Значення фрагментації індексів в таблиці Рейтинг

2.3 Реалізація користувацького інтерфейсу інформаційної системи

Після розробки бази даних було реалізовано користувацький інтерфейс інформаційної системи обліку успішності студентів. Інтерфейс забезпечує взаємодію користувачів із базою даних, надає доступ до основних функцій системи та дозволяє виконувати операції без використання SQL-запитів.

Реалізовані програмні модулі забезпечують перегляд інформації про студентів, ведення журналу оцінок, аналіз успішності та формування звітності.

Після авторизації користувач переходить до головної панелі інформаційної системи (рис. 2.13). На головній сторінці відображаються основні статистичні показники, зокрема кількість студентів, середній бал, кількість навчальних дисциплін та студентів, які потребують додаткової уваги. Крім того, користувач має можливість переглядати останні внесені оцінки та узагальнену інформацію щодо успішності студентів за дисциплінами.

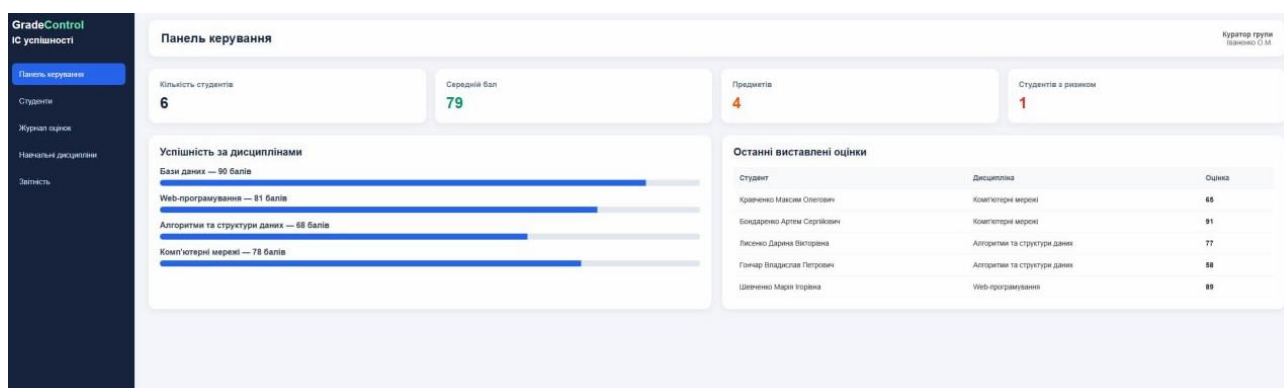


Рисунок 2.13 – Головна панель інформаційної системи

Наступним модулем інформаційної системи є модуль управління студентами (рис. 2.14). Він призначений для перегляду інформації про студентів, їх належність до академічних груп, курсу навчання та контактних даних. Також у таблиці відображається середній бал кожного студента та його поточний статус успішності. Для швидкого пошуку необхідного запису реалізовано функцію пошуку.

The 'Студенти' window displays the following table:

ПІВ	Група	Курс	Email	Середній бал	Статус
Бондаренко Артем Сергійович	КН-31	3	bondarenko@university.edu.ua	89	Висока успішність
Мельник Софія Андріївна	КН-31	3	melnyk@university.edu.ua	94	Висока успішність
Кравченко Максим Олександрович	КН-32	3	kvaschenko@university.edu.ua	89	Середня успішність
Шевченко Марія Тарасівна	КН-32	3	shevchenko@university.edu.ua	89	Висока успішність
Гончар Владислав Петрович	К5-21	2	honsch@university.edu.ua	88	Потрібна увага
Лісенок Дарина Вікторівна	К5-21	2	lysenko@university.edu.ua	77	Середня успішність

Рисунок 2.14 – Вікно перегляду інформації про студентів

Для ведення результатів навчання реалізовано журнал оцінок (рис. 2.15). За допомогою даного модуля викладач може обрати студента та навчальну дисципліну, ввести оцінку, а також переглянути вже внесені результати. Передбачено можливість фільтрації інформації за академічними групами та дисциплінами, що значно спрощує роботу користувача.

Дата	Студент	Група	Дисципліна	Тип контролю	Оцінка
11.03.2025	Бондаренко Артем Сергійович	КН-31	Базис дані	Модульний контроль	85
11.03.2025	Мельник Софія Андріївна	КН-31	Базис дані	Модульний контроль	94
14.03.2025	Кравченко Максим Євгенович	КН-32	Web-програмування	Практична робота	72
14.03.2025	Шатченко Марія Ігорівна	КН-32	Web-програмування	Практична робота	89
18.03.2025	Гончар Владислав Петрович	КБ-21	Алгоритми та структури даних	Лабораторна робота	58
18.03.2025	Лисенко Дарина Вікторівна	КБ-21	Алгоритми та структури даних	Лабораторна робота	77
02.04.2025	Бондаренко Артем Сергійович	КН-31	Комп'ютерні мережі	Тестування	91
02.04.2025	Кравченко Максим Євгенович	КН-32	Комп'ютерні мережі	Тестування	65

Рисунок 2.15 – Журнал обліку успішності студентів

Для аналізу результатів навчання в інформаційній системі реалізовано модуль звітності (рис. 2.16). Він дозволяє автоматично формувати узагальнену інформацію щодо успішності академічних груп, визначати середній бал студентів та формувати рекомендації щодо покращення навчального процесу. Отримані результати можуть бути використані адміністрацією університету та кураторами академічних груп під час аналізу навчальної діяльності студентів.

Назва дисципліни	Викладач	Кредити ЄКТС	Семестр	Форма контролю
Базис дані	Петренко В.В.	5	5	Екзамени
Web-програмування	Іванченко О.М.	4	5	Зачеки
Алгоритми та структури даних	Коваленко Н.П.	6	4	Екзамени
Комп'ютерні мережі	Сидоренко А.А.	5	4	Екзамени

Рисунок 2.12 – Модуль формування звітності

Висновки з другого розділу

У другому розділі було здійснено проектування та реалізацію бази даних для інформаційної системи обліку успішності студентів. Розроблено логічну структуру інформаційної структури та модель архітектури. На основі

розробленої моделі створено фізичні таблиці за допомогою засобів T-SQL та реалізовано зв'язки між ними для забезпечення цілісності даних. Побудовано діаграми послідовності та прецедентів. Створено кластеризовані та некластеризовані індекси для найбільш навантажених таблиць, а також реалізовано алгоритми їх періодичної перебудови та реорганізації для усунення фрагментації. Розроблено інтерфейс системи.

РОЗДІЛ 3. ОРГАНІЗАЦІЯ СИСТЕМИ ЗАХИСТУ ДАНИХ ТА ОБСЛУГОВУВАННЯ

3.1 Аналіз структури користувачів та розробка рольової моделі

В рамках аналізу структури користувачів важливо визначити ролі та обов'язки кожного типу користувача системи. Це може включати адміністраторів баз даних, викладачів, студентів та інших учасників системи. Детальне розглядання функціональних обов'язків допомагає визначити необхідні рівні доступу та можливості для кожного користувача. Аналіз структури груп та їх відносин дозволяє краще розуміти логіку призначення прав та обмежень.

Кожному користувачеві має бути призначений конкретний рівень доступу залежно від його функцій та відповідальностей. Визначення прав доступу до таблиць, перегляду чи редагування певних полів, а також виконання певних операцій є ключовим етапом аналізу.

На перетині рядків і стовпців зазначено дії, які може виконувати користувач даного типу ролі: 1 – перегляд даних, 2 – редагування, 3 – видалення, 4 – повний доступ

Таблиця 3.1

Матриця доступу для користувачів в базі даних

	Адміністратор	Викладач	Студент	Технік
Факультет	4	0	0	1,2
Кафедра	4	0	0	1,2
Спеціальність	4	0	0	1,2
Група	4	1	1	1,2
Студенти	4	1	0	4
Викладачі	4	1	1	4
Рейтинг	4	1,2	1	1,2
Користувачі	4	0	0	4
Розклад	4	1,2	1	4

3.2 Реалізація системи розмежування прав доступу

Для реалізації потрібно створити ролі. Потім потрібно згрупувати користувачів у відповідності визначених дозволів. Для перевірки чи створилися ролі можна використувати запит:
EXEC sp_helprole

	RoleName	RoleId	IsAppRole
1	public	0	0
2	Administrator	9	0
3	Teacher	10	0
4	Student	11	0
5	Technician	12	0
6	db_owner	16384	0
7	db_accessadmin	16385	0
8	db_securityadmin	16386	0
9	db_ddladmin	16387	0
10	db_backupoperator	16389	0

Рисунок 3.1 – Результат виведення ролей

Далі створимо користувачів.

Лістинг запиту на створення користувачів:

```
CREATE LOGIN AdminUser WITH PASSWORD = '12345';
CREATE USER AdminUser FOR LOGIN AdminUser;

CREATE LOGIN TeacherUser WITH PASSWORD = '12345';
CREATE USER TeacherUser FOR LOGIN TeacherUser;

CREATE LOGIN StudentUser WITH PASSWORD = '12345';
CREATE USER StudentUser FOR LOGIN StudentUser;

CREATE LOGIN TechnicianUser WITH PASSWORD = '12345';
CREATE USER TechnicianUser FOR LOGIN TechnicianUser;
```

Наступним кроком буде перевірка, на те чи створилися користувачі:

EXEC sp_helpuser

	UserName	RoleName	LoginName	DefDBName	DefSchemaName	UserID	SID
1	AdminUser	db_owner	AdminUser	master	dbo	5	0x7F44C4630953DE45ADCAB8242F353462
2	dbo	db_owner	owner	master	dbo	1	0x09037C743CF5DA45AA0760D1C8403B8C
3	guest	public	NULL	NULL	guest	2	0x00
4	INFORMATION_SCHEMA	public	NULL	NULL	NULL	3	NULL
5	StudentUser	db_datareader	StudentUser	master	dbo	7	0x494C336B0A31894D8F0E91CBC6F9598C
6	sys	public	NULL	NULL	NULL	4	NULL
7	TeacherUser	db_datareader	TeacherUser	master	dbo	6	0xEAB52E4C0E6477419488447D88ACDDA8
8	TechnicianUser	public	TechnicianUser	master	dbo	8	0x45A2819F19F2184EA8B7E1127B32821D

Рисунок 3.2 – Результат виведення інформації про користувачів

Щоб перевірити чи правильно користувачам налаштувались дозволи і заборони, можна на прикладі одного користувача, наприклад викладача, перевірити перегляд, редагування таблиць і т. д. Перевірка показана нижче:

```
EXECUTE AS USER = 'TeacherUser'
SELECT * FROM [Рейтинг]
```

```
SELECT * FROM [Користувачі]
DELETE FROM [Студенти] WHERE [Id групи] = 7
```

	Id студента	Id дисципліни	Тип контролю	Оцінка
1	1	2	Залік	Зараховано
2	1	3	Екзамен	92
3	1	4	Залік	Зараховано
4	1	11	Екзамен	85
5	2	2	Залік	Зараховано
6	2	3	Екзамен	78
7	2	4	Залік	Зараховано
8	2	11	Екзамен	88
9	3	2	Залік	Зараховано

Рисунок 3.3 – Успішне виконання запиту

Повідомлення 229, рівень 14, стан 5, рядок 106

Заборонено дозвіл "SELECT" на об'єкт "Користувачі" бази даних "База даних університету", схеми "dbo".

Повідомлення 229, рівень 14, стан 5, рядок 107

Заборонено дозвіл "DELETE" на об'єкт "Студенти" бази даних "База даних університету", схеми "dbo".

Рисунок 3.4 – Доступ на виконання скрипта заборонено

Для всіх ролей були надані відповідні права та створені користувачі.

3.3 Регламентне обслуговування та резервне копіювання даних

Регламентне обслуговування серверів баз даних є важливою частиною діяльності адміністраторів баз даних. Цей процес включає в себе ряд дій та процедур, спрямованих на забезпечення стабільності, ефективності та безпеки роботи серверів баз даних.

Регулярне створення резервних копій баз даних є важливим етапом обслуговування. Це забезпечує можливість відновлення даних у випадку випадкового втрати чи пошкодження. Регулярне оновлення системи управління базами даних та встановлення патчів потрібне для забезпечення безпеки та виправлення виявлених помилок. Результати створення резервних копій показані на рисунках 3.5-3.6.

> Цей ПК > Локальний диск (C:) > sql > backups	
Ім'я	Дата зміни
backup_diff.bak	26.12.2023 14:53
full_backup.bak	26.12.2023 14:53

Рисунок 3.5 – Створення повної та диференціальної резервної копії

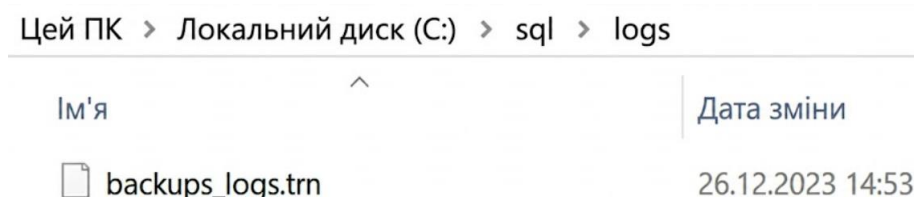


Рисунок 3.6 – Створення журналів транзакцій

Для бази даних можна створити план обслуговування. План обслуговування баз даних — це стратегія та набір дій, спрямованих на забезпечення нормальної функціональності та високої ефективності бази даних. Основною метою плану обслуговування є збереження стабільності та продуктивності системи баз даних протягом тривалого часу.

План обслуговування визначає, як керувати ресурсами, визначає часові вікна для виконання важливих завдань, щоб мінімізувати вплив на продуктивність під час робочого часу.

План обслуговування дозволяє відстежувати та аналізувати використання ресурсів бази даних. Створення плану обслуговування показано на рисунку 3.7.

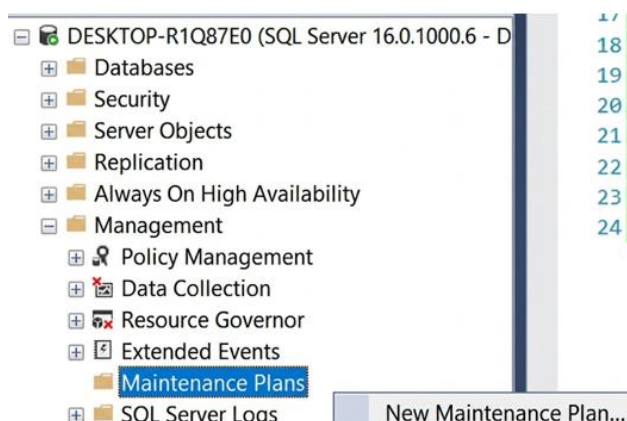


Рисунок 3.7 – Створення плану обслуговування

План обслуговування буде включати в себе такі задачі:

- перевірка бази даних на цілісність;
- перебудова індекса;
- реорганізація індекса;
- оновлення статистики;
- резервне копіювання бази даних;
- очищення журналу.

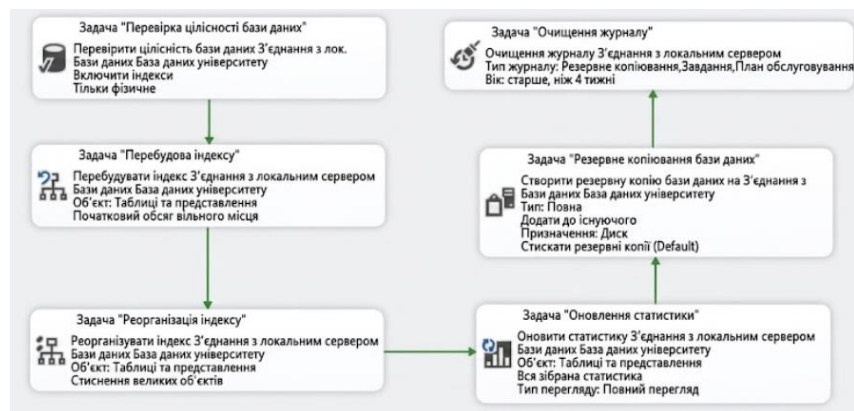


Рисунок 3.8 – Схема плану обслуговування

Налаштування розкладу виконання плану обслуговування показано на рисунку 3.9. План буде виконуватись в вівторок, четвер та суботу о 02:00 без дати закінчення.

Рисунок 3.9 – Розклад плану обслуговування

Перед запуском плану обслуговування потрібно спочатку запуснути службу Агент SQL Server. Після запуску служби, можна запускати план обслуговування (рис 3.10).

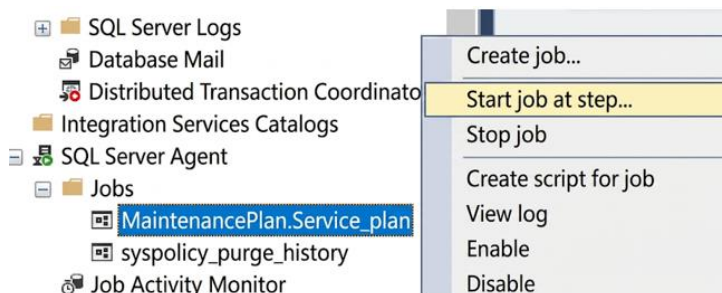


Рисунок 3.10 – Запуск плану обслуговування



Рисунок 3.11 – Успішний запуск плану обслуговування

Висновки з третього розділу

У третьому розділі було вирішено комплекс завдань із забезпечення безпеки, надійності та безперебійності роботи розробленої бази даних. Проведено аналіз структури користувачів та розроблено матрицю доступу, на основі якої реалізовано систему рольового розмежування привілеїв. Кожна категорія користувачів отримала строго визначений набір прав лише до тих об'єктів бази даних, які необхідні для виконання їхніх обов'язків.

Для забезпечення стійкості системи до збоїв було налаштовано регламентне обслуговування за допомогою плану обслуговування, який включає автоматичну перевірку цілісності, оновлення статистики, а також створення повних, диференціальних копій та бекапів журналів транзакцій за розкладом.

ВИСНОВКИ

У кваліфікаційній роботі було успішно вирішено актуальне завдання з проектування, оптимізації та організації захисту бази даних для інформаційної системи обліку успішності студентів закладу вищої освіти. На початковому етапі дослідження було проаналізовано предметну область та інформаційну інфраструктуру сучасного університету. За результатами порівняльного аналізу систем управління базами даних було обґрунтовано вибір Microsoft SQL Server як найбільш оптимальної платформи.

У ході виконання практичної частини було створено необхідні таблиці з налаштуванням зв'язків між ними для забезпечення цілісності даних, а також розроблено відповідні представлення для зручного виведення зведеної інформації. Для оптимізації роботи з великими масивами даних та прискорення пошуку було впроваджено систему кластерних і некластерних індексів. Крім того, реалізовано алгоритми їх регулярної перебудови та реорганізації, що дозволило усунути фрагментацію та суттєво підвищити загальну швидкість виконання запитів.

Особливу увагу в дослідженні приділено питанням безпеки та розмежування прав доступу до даних. На основі аналізу структури користувачів було реалізовано багаторівневу рольову модель доступу. Кожна категорія користувачів отримала строго визначений набір привілеїв, що унеможливило несанкціоновану зміну оцінок чи інших критичних даних.

Для гарантування безперебійної роботи та надійності розробленої системи було автоматизовано процеси її регламентного обслуговування. Зокрема, налаштовано плани обслуговування серверів, які забезпечують регулярне створення резервних копій, перевірку цілісності бази та оновлення статистики без необхідності постійного ручного втручання. Таким чином, мета кваліфікаційної роботи досягнута повною мірою, а створена надійна, оптимізована та захищена база даних повністю готова до практичного впровадження та інтеграції в комплексну інформаційну систему навчального закладу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Автоматизація резервного копіювання в MS SQL Server [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/264927/>.
2. Автоматизована інформаційна система обробки результатів навчання студентів з урахуванням компетентнісного підходу. Інформаційні технології і засоби навчання. 2021. URL: <https://journal.iitta.gov.ua/index.php/itlt/article/view/3584/1859.pdf> (дата звернення: 12.04.2026).
3. Агент SQL Server [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/sql/ssms/agent/sql-server-agent?view=sql-server-ver15#benefits-of-sql-server-agent>
4. Бродський Ю. Б., Молодецька К. В., Николюк О. М. Системний аналіз в економіці : навч. посіб. Житомир : ЖНАЕУ, 2014. 175 с.
5. Відновлення бази даних SQL [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/ruru/sql/relational-databases/backup-restore/database-to-a-point-intime-full-recovery-model?view=sql-server-ver15>.
6. Діденко О. В., Купрієнко Д. А. Електронний журнал обліку успішності слухачів (курсантів, студентів) як засіб раціоналізації навчально-виховного процесу. Інформаційні технології і засоби навчання. 2015. Т. 47, № 3. URL: https://www.researchgate.net/publication/331467354_ELEKTRONNIJ_ZURNAL_O_BLIKU_USPISNOSTI_SLUHACIV_KURSANTIV_STUDENTIV_AK_ZASIB_RA_CIONALIZACII_NAVCALNO-VIHOVNOGO_PROCESU (дата звернення: 11.04.2026).
7. Документація по MS SQL Server [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/sql/sql-server/tutorials-for-sql-server2016?view=sql-server-ver15>.
8. Задачі SQL [Електронний ресурс] – Режим доступу до ресурсу: <http://sql-tutorial.ru/sqlbook/ru>

9. Катренко, А. В. Системний аналіз об'єктів та процесів комп'ютеризації : навч. посіб. Львів : Новий Світ-2000, 2003. 416 с.
10. Ковалюк Т. В. Основи проєктування баз даних. Київ : ВД «Києво-Могилянська академія», 2020. 360 с.
11. Литвин В.В., Пасічник В.В., Шаховська Н.Б. Проектування інформаційних систем. Навчальний посібник (затв. МОН України). Львів : Магнолія 2006, 2024. 380 с.
12. Основи обслуговування БД [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/company/softlab/blog/266033/>.
13. Проектування інформаційних систем: посібник / В. С. Пономаренко, О. І. Пушкар, І. В. Журавльова [та ін.] ; за ред. В. С. Пономаренка. Київ : Академія, 2002. 486 с.
14. Створення плану обслуговування в MS SQL Server [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/sql/relationaldatabases/maintenance-plans/create-a-maintenance-plan?view=sql-server-ver15>.
15. Створення плану обслуговування в MS SQL Server [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/sql/relationaldatabases/maintenance-plans/create-a-maintenance-plan?view=sql-server-ver15>
16. Титова Н. М. Електронний журнал обліку успішності студентів як ефективний інструмент моніторингу якості вищої освіти. Інженерна та освітня діяльність. 2017. URL: <https://enpuir.udu.edu.ua/entities/publication/88ea3f2e-8be4-4cf7-8391-5238bdaf68b6> (дата звернення: 12.04.2026).
17. Усенко А. А., Вакалюк Т. А. Автоматизована інформаційна система обліку успішності. Наука, освіта, технології: матеріали конференції. Житомир, 2020. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2020/05/193.pdf> (дата звернення: 12.04.2026).

18. Create IDEF0 Diagrams : Microsoft Support. URL: <https://support.microsoft.com/en-us/office/create-idef0-diagrams-ea7a9289-96e0-4df8-bb26-a62ea86417fc> (дата звернення: 25.06.2026).

19. Databases (SQL Server). Microsoft Learn. URL: <https://learn.microsoft.com/en-us/sql/relational-databases/databases/databases?view=sql-server-ver17> (дата звернення: 25.02.2026).

20. Guide to UML Diagramming and Database Modeling. Microsoft. URL: <https://www.microsoft.com/en-us/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling> (дата звернення: 05.03.2026).

21. IDEF0 Function Modeling Method : вебсайт. URL: https://www.idef.com/idefo-function_modeling_method/ (дата звернення: 25.06.2026).

22. ISO/IEC 19505-1:2012. Information technology — Object Management Group Unified Modeling Language (OMG UML) – Part 1: Infrastructure. Geneva : ISO, 2012. URL: <https://www.omg.org/spec/UML/ISO/19505-1/PDF> (дата звернення: 18.03.2026).

23. Object Management Group. OMG Unified Modeling Language (OMG UML), Version 2.5.1. URL: <https://www.omg.org/spec/UML/2.5.1> (дата звернення: 25.06.2026).

24. Object Management Group. Unified Modeling Language (UML) Resource Hub. URL: <https://www.omg.org/uml/resource-hub.htm> (дата звернення: 23.03.2026).

25. SQL Server technical documentation. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver17> (дата звернення: 25.02.2026).

26. SQL. Основні відомості [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/SQL>.

27. SQL-довідник [Електронний ресурс] – Режим доступу до ресурсу: <http://www.sql.ru/forum>

28. UML Practical Guide – All you need to know about UML modeling. Available at : <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-practical-guide/> (дата звернення: 05.03.2026).

29. UML Practical Guide - All you need to know about UML modeling. UML Practical Guide. Visual Paradigm. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-practical-guide/> (дата звернення: 30.04.2026).

30. UML Tutorial. Available at : <https://www.tutorialspoint.com/uml/index.htm> (дата звернення: 05.03.2026).

ДОДАТКИ

ДОДАТОК А

Таблиця 1.1

Основне порівняння СУБД

Характеристика	Oracle Database	MS SQL Server	MySQL Server
Адміністративне керування	Високий рівень контролю	Зручний інтерфейс	Легке керування
Графічні інструменти	OEM	SMSS	MySQL Workbench
Простота обслуговування	Висока складність	Середня складність	Низька складність
Механізм даних	Таблицевий, об'єктно-орієнтовний	Таблицевий, об'єктно-орієнтовний	Таблицевий
Робота с декількома ЦП	+	+	+
Функції з'єднання	+	+	+
Обробка даних мультимедіа	+	+	Базова підтримка
Одночасний доступ декількох користувачів	Відмінна	Відмінна	Задовільна
Підключення к Web	+	+	+
Повнотекстовий пошук	+	+	+
Сумісність	Відмінна	Відмінна	Задовільна
Інтеграція з іншими СУБД	+	+	+
Єдина реєстрація	-	+	-
Робота під керівництвом ОС	Кросс-платформеність	Windows	Кросс-платформеність

Продовження таблиці 1.1

Можливості програмування	PL/SQL	T-SQL	SQL
Процедури, та тригери	+	+	+
Вбудована мова програмування	Java, PL/SQL	.NET, T-SQL	-
Побудова БД	OEM, командний рядок	SSMS, командний рядок	MySQL Workbench, командний рядок
Мова SQL	SQL, PL/SQL	T-SQL	SQL
Можливість віддаленого доступу	+	+	+
Організація сховищ даних і підготовка звітів	Розширені можливості, OLAP	Розширені можливості, OLAP	Базова функціональність
Дистанційне адміністрування	+	+	+

Таблиця 1.2

Порівняння за вимогами ТЗ

Характеристика	Oracle Database	MS SQL Server	MySQL Server
Локалізація інтерфейсу користувача, можливість побудови і сортування полів БД, що містять символи кирилиці	+	+	+
Підтримка структури відносних даних	+	+	+
Підтримка технології клієнт/сервер	+	+	+

Продовження таблиці 1.2

Підтримка багатопроцесорної архітектури	+	+	+
Підтримка кластерної архітектури	+	+	+/-
Наявність засобів для створення індексів і кластерів для підвищення ефективності використання даних	+	+	+
Відновлення баз даних із використанням журналу транзакцій	+	+	+
Механізм блокування транзакцій під час запису або на рівні сторінок	+	+	+
Підтримка ANSI SQL	+	+	+
Підтримка стандарту SQL-3	+	+	+/-
Підтримка ODBC	+	+	+
Контроль цілісності БД	+	+	+
Підтримка утиліт резервування БД	+	+	+
Імпорт/експорт таблиць БД	+	+	+
Сумісність з ОС модулів користувача та сервера	+	+	+
Підтримка визначених мережових протоколів	+	+	+
Наявність графічного інтерфейсу для адміністраторів БД	+	+	+
Контроль доступу до даних. Аутентифікація засобами СУБД	+	+	+
Централізоване керування користувачами	+	+	+/-
Наявність оптимізатора запитів для оптимізації планів виконання	+	+	+
Підтримка великих двійкових об'єктів (BLOB)	+	+	+
Підтримка OLAP технологій	+	+	-
Підтримка протоколів VI SAN (Virtual Interface System Area Network)	+	-	-

Таблиця 1.3

Експертна оцінка багатористувацьких СУБД

СУБД	Продуктивність	Конкурентний доступ	Кількість користувачів	Великі БД	Готовність
Oracle Database	7	9	8	8	8
MS SQL Server	6	7	5	5	6
MySQL Server	5	6	5	4	5

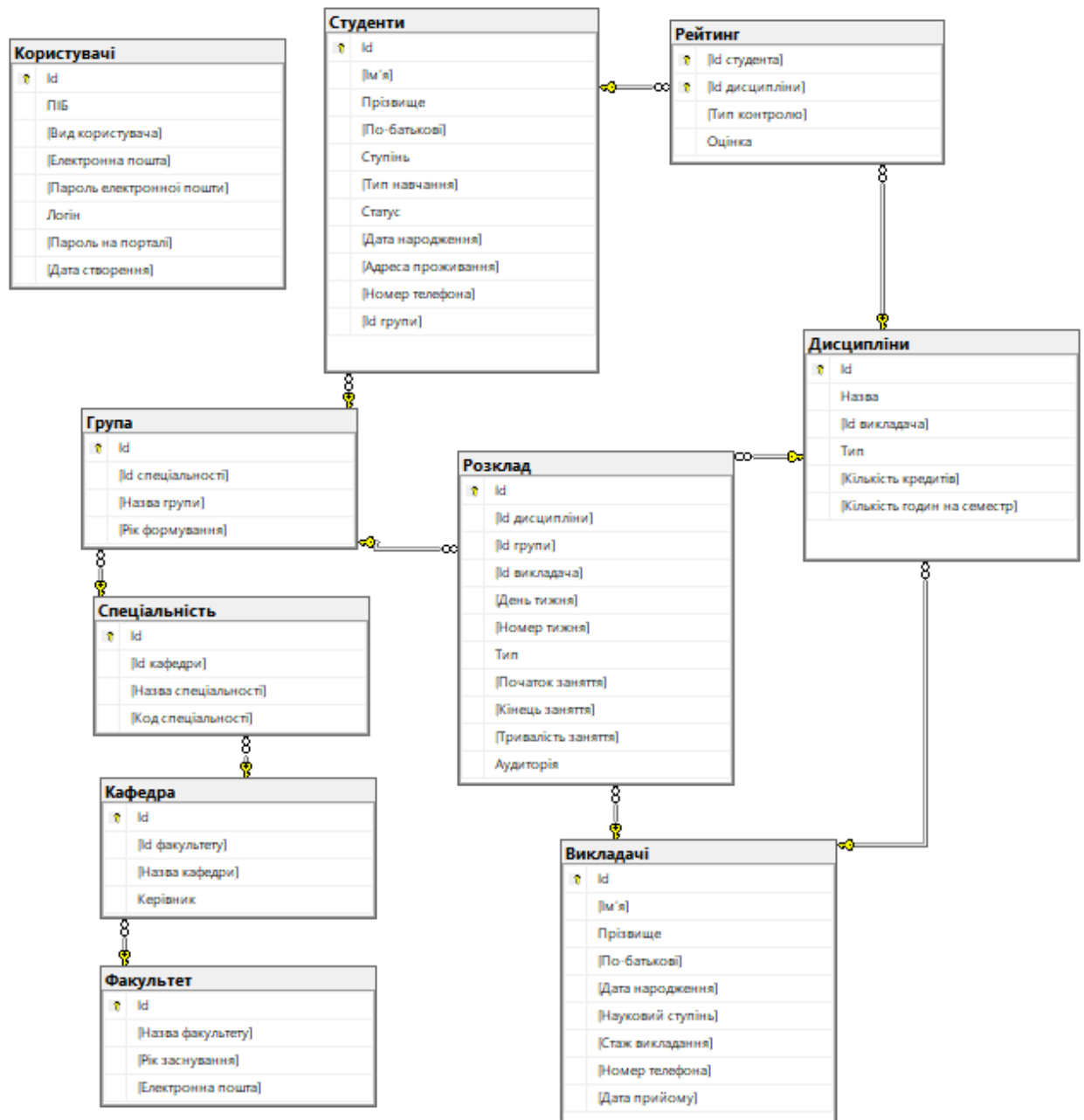


Рисунок А.1 - Діаграма бази даних

Лістинг запитів

```
CREATE DATABASE [База даних університету]

USE [База даних університету]

-- Таблиця "Факультет"
CREATE TABLE Факультет (
    Id INT PRIMARY KEY IDENTITY(1,1),
    [Назва факультету] VARCHAR(255) NOT NULL,
    [Рік заснування] DATE DEFAULT GETDATE() CHECK ([Рік заснування] <=
GETDATE()),
    [Електронна пошта] NVARCHAR(35) UNIQUE
);

-- Таблиця "Кафедра"
CREATE TABLE Кафедра (
    Id INT PRIMARY KEY IDENTITY(1,1),
    [Id факультету] INT,
    [Назва кафедри] VARCHAR(255) NOT NULL,
    [Керівник] VARCHAR(60),
    FOREIGN KEY ([Id факультету]) REFERENCES Факультет(Id)
);

-- Таблиця "Спеціальність"
CREATE TABLE Спеціальність (
    Id INT PRIMARY KEY IDENTITY(1,1),
    [Id кафедри] INT,
    [Назва спеціальності] VARCHAR(255) NOT NULL,
    [Код спеціальності] VARCHAR(15),
    FOREIGN KEY ([Id кафедри]) REFERENCES Кафедра(Id)
);

-- Таблиця "Група"
CREATE TABLE Група (
    Id INT PRIMARY KEY IDENTITY(1,1),
    [Id спеціальності] INT,
    [Назва групи] VARCHAR(50) NOT NULL,
    [Рік формування] DATE DEFAULT GETDATE() CHECK ([Рік формування] <=
GETDATE()),
    FOREIGN KEY ([Id спеціальності]) REFERENCES Спеціальність(Id)
);

-- Таблиця "Студенти"
CREATE TABLE Студенти (
    Id INT PRIMARY KEY IDENTITY(1,1),
    [Ім'я] VARCHAR(50) NOT NULL,
    Прізвище VARCHAR(50) NOT NULL,
    [По-батькові] VARCHAR(50) NOT NULL,
    [Ступінь] VARCHAR(30),
```

```

        [Тип навчання] VARCHAR(30),
        [Статус] VARCHAR(30),
        [Дата народження] DATE,
        [Адреса проживання] VARCHAR(255),
        [Номер телефона] VARCHAR(20),
        [Id групи] INT,
        FOREIGN KEY ([Id групи]) REFERENCES Група(Id)
    );

SELECT * FROM [Студенти]

-- Таблиця "Викладачі"
CREATE TABLE Викладачі (
    Id INT PRIMARY KEY IDENTITY(1,1),
    [Ім'я] VARCHAR(50) NOT NULL,
    Прізвище VARCHAR(50) NOT NULL,
    [По-батькові] VARCHAR(50) NOT NULL,
    [Дата народження] DATE,
    [Науковий ступінь] VARCHAR(50),
    [Стаж викладання] VARCHAR(50),
    [Номер телефона] VARCHAR(20),
    [Дата прийому] DATE
);

-- Таблиця "Дисципліни"
CREATE TABLE Дисципліни (
    Id INT PRIMARY KEY IDENTITY(1,1),
    Назва VARCHAR(255) NOT NULL,
    [Id викладача] INT,
    [Тип] VARCHAR(50),
    [Кількість кредитів] INT CHECK ([Кількість кредитів] >= 0),
    [Кількість годин на семестр] INT CHECK ([Кількість годин на семестр]
    >= 0),
    FOREIGN KEY ([Id викладача]) REFERENCES Викладачі(Id)
);

-- Таблиця "Рейтинг"
CREATE TABLE Рейтинг (
    [Id студента] INT,
    [Id дисципліни] INT,
    [Тип контролю] VARCHAR(50) CHECK ([Тип контролю] IN ('Залік',
    'Екзамен')),
    Оцінка VARCHAR(20),
    PRIMARY KEY ([Id студента], [Id дисципліни]),
    FOREIGN KEY ([Id студента]) REFERENCES Студенти(Id),
    FOREIGN KEY ([Id дисципліни]) REFERENCES Дисципліни(Id)
);

-- Таблиця "Користувачі"
CREATE TABLE Користувачі (
    Id INT PRIMARY KEY IDENTITY(1,1),

```

```

        ПІБ VARCHAR(100) NOT NULL,
        [Вид користувача] VARCHAR(50) CHECK ([Вид користувача] IN ('Студент',
'Викладач', 'Адміністратор', 'Технік')),
        [Електронна пошта] VARCHAR(255) UNIQUE,
        [Пароль електронної пошти] VARCHAR(255),
        Логін VARCHAR(50) UNIQUE,
        [Пароль на порталі] VARCHAR(255),
        [Дата створення] DATE DEFAULT GETDATE() CHECK ([Дата створення] <=
GETDATE())
    );

```

-- Таблиця "Розклад"

```

CREATE TABLE Розклад (
    Id INT PRIMARY KEY IDENTITY(1,1),
    [Id дисципліни] INT,
    [Id групи] INT,
    [Id викладача] INT,
    [День тижня] VARCHAR(20) CHECK ([День тижня] IN ('Понеділок',
'Вівторок', 'Середа', 'Четвер', 'П`ятниця')),
    [Номер тижня] VARCHAR(20) CHECK ([Номер тижня] IN ('Парний',
'Непарний')),
    [Тип] VARCHAR(20) CHECK ([Тип] IN ('Лекція', 'Лабораторна робота',
'Практика')),
    [Початок заняття] TIME,
    [Кінець заняття] TIME,
    [Тривалість заняття] INT CHECK ([Тривалість заняття] > 0), -- у
хвилинах
    Аудиторія VARCHAR(20),
    FOREIGN KEY ([Id дисципліни]) REFERENCES Дисципліни(Id),
    FOREIGN KEY ([Id групи]) REFERENCES Група(Id),
    FOREIGN KEY ([Id викладача]) REFERENCES Викладачі(Id)
);

```

--===== Вибірки
=====

```

SELECT @@Servername AS ServerName ,
TABLE_CATALOG ,
TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE'
ORDER BY TABLE_NAME;

```

-- Вибірка інформації про студентів разом з інформацією про їх групу та спеціальність

```

SELECT
    Студенти.Id AS 'ID Студента',
    Студенти.[Ім'я],
    Студенти.Прізвище,

```

```

        Студенти.[По-батькові],
        Група.[Назва групи] AS 'Група',
        Спеціальність.[Назва спеціальності] AS 'Спеціальність'
FROM Студенти
JOIN Група ON Студенти.[Id групи] = Група.Id
JOIN Спеціальність ON Група.[Id спеціальності] = Спеціальність.Id;

-- Вибірка розкладу з включенням інформації про дисципліну, викладача та
групу
SELECT
    Розклад.Id AS 'ID Заняття',
    Дисципліни.Назва AS 'Дисципліна',
    Викладачі.[Ім'я] + ' ' + Викладачі.Прізвище AS 'Викладач',
    Група.[Назва групи] AS 'Група',
    Розклад.[День тижня],
    Розклад.[Номер тижня],
    Розклад.[Тип],
    Розклад.[Початок заняття],
    Розклад.[Кінець заняття],
    Розклад.[Тривалість заняття],
    Розклад.Аудиторія
FROM Розклад
JOIN Дисципліни ON Розклад.[Id дисципліни] = Дисципліни.Id
JOIN Викладачі ON Розклад.[Id викладача] = Викладачі.Id
JOIN Група ON Розклад.[Id групи] = Група.Id;

--
=====Представлення=====
=====--

-- Представлення для виведення кількості студентів у кожній групі
GO
CREATE VIEW [Кількість студентів у групах] AS
SELECT
    Група.[Назва групи],
    COUNT(Студенти.Id) AS 'Кількість студентів'
FROM Група
LEFT JOIN Студенти ON Група.Id = Студенти.[Id групи]
GROUP BY Група.[Назва групи];
GO

SELECT *
FROM [Кількість студентів у групах]
GO

-- Представлення для виведення студентів та їх оцінок

CREATE VIEW [Студенти та оцінки] AS
SELECT
    Студенти.[Ім'я] + ' ' + Студенти.Прізвище AS 'ПІБ Студента',
    Дисципліни.Назва AS 'Дисципліна',
    Рейтинг.Оцінка

```

```

FROM
    Студенти
JOIN Рейтинг ON Студенти.Id = Рейтинг.[Id студента]
JOIN Дисципліни ON Рейтинг.[Id дисципліни] = Дисципліни.Id;
GO

SELECT *
FROM [Студенти та оцінки]
GO

-- Перегляд створених представлень
SELECT @@Servername AS ServerName ,
DB_NAME() AS DBName ,
Name AS ViewName ,
create_date
FROM sys.Views
ORDER BY Name

USE [База даних університету]

-- До таблиці "Користувачі"
INSERT INTO Користувачі (ПІБ, [Вид користувача], [Електронна пошта],
[Пароль електронної пошти], Логін, [Пароль на порталі], [Дата створення])
VALUES
    ('Адміністратор', 'Адміністратор', 'admin@example.com', 'admin_pass',
'admin_login', GETDATE()),
    ('Технік 1', 'Технік', 'tech1@example.com', 'tech1_pass',
'tech1_login', GETDATE()),
    ('Технік 2', 'Технік', 'tech2@example.com', 'tech2_pass',
'tech2_login', GETDATE()),
    ('Технік 3', 'Технік', 'tech3@example.com', 'tech3_pass',
'tech3_login', GETDATE()),
    ('Технік 4', 'Технік', 'tech4@example.com', 'tech4_pass',
'tech4_login', GETDATE()),
    ('Викладач 1', 'Викладач', 'teacher1@example.com', 'teacher1_pass',
'teacher1_login', GETDATE()),
    ('Викладач 2', 'Викладач', 'teacher2@example.com', 'teacher2_pass',
'teacher2_login', GETDATE()),
    ('Викладач 3', 'Викладач', 'teacher3@example.com', 'teacher3_pass',
'teacher3_login', GETDATE()),
    ('Викладач 4', 'Викладач', 'teacher4@example.com', 'teacher4_pass',
'teacher4_login', GETDATE()),
    ('Викладач 5', 'Викладач', 'teacher5@example.com', 'teacher5_pass',
'teacher5_login', GETDATE());

-- До таблиці "Факультет"
INSERT INTO Факультет ([Назва факультету], [Рік заснування], [Електронна
пошта])
VALUES
    ('Факультет прикладної математики', '1991-05-27',
'appmath@tt.edu.ua'),

```

```
( 'Факультет математики та інформатики', '1995-03-10',  
'math@tt.edu.ua');
```

```
-- До таблиці "Кафедра"
```

```
INSERT INTO Кафедра ([Id факультету], [Назва кафедри], [Керівник])  
VALUES
```

```
(1, 'Кафедра прикладної математики', 'Олександр Володимирович  
Литвин'),  
(1, 'Кафедра програмного забезпечення комп'ютерних систем', 'Марія  
Ігорівна Сидоренко'),  
(1, 'Кафедра системного програмування і спеціалізованих комп'ютерних  
систем', 'Сергій Олексійович Петров'),  
(2, 'Кафедра математичного аналізу та теорії ймовірностей', 'Юлія  
Володимирівна Захарчук'),  
(2, 'Кафедра інформаційних технологій', 'Олег Андрійович Кравець');
```

```
-- До таблиці "Спеціальність"
```

```
INSERT INTO Спеціальність ([Id кафедри], [Назва спеціальності], [Код  
спеціальності])
```

```
VALUES
```

```
(1, 'Прикладна математика', '113'),  
(2, 'Комп'ютерні науки', '122'),  
(2, 'Інженерія програмного забезпечення', '121'),  
(3, 'Системне програмування', '120'),  
(3, 'Інформаційні системи та технології', '126'),  
(4, 'Математика', '111'),  
(4, 'Статистика', '112'),  
(5, 'Кібербезпека', '125'),  
(5, 'Ком'ютерна інженерія', '123');
```

```
-- До таблиці "Група"
```

```
INSERT INTO Група ([Id спеціальності], [Назва групи], [Рік формування])  
VALUES
```

```
(1, 'ПМ-1', '2022-09-01'),  
(2, 'КН-21-1', '2021-09-01'),  
(2, 'КН-22-1', '2022-09-01'),  
(3, 'ІПЗ-21-1', '2021-09-01'),  
(3, 'ІПЗ-22-1', '2022-09-01'),  
(3, 'ІПЗ-22-2', '2022-09-01'),  
(4, 'СП-21-1', '2021-09-01'),  
(5, 'ІСТ-21-1', '2021-09-01'),  
(5, 'ІСТ-22-1', '2022-09-01'),  
(6, 'М-1', '2022-09-01'),  
(7, 'СТ-1', '2022-09-01'),  
(8, 'КБ-21-1', '2021-09-01'),  
(8, 'КБ-22-1', '2022-09-01'),  
(9, 'КІ-21-1', '2021-09-01'),  
(9, 'КІ-22-1', '2022-09-01');
```

```
-- До таблиці "Студенти"
```

```

INSERT INTO Студенти ([Ім'я], Прізвище, [По-батькові], [Ступінь], [Тип
навчання], [Статус], [Дата народження], [Адреса проживання], [Номер
телефона], [Id групи])
VALUES
    -- ПМ-1
    ('Іван', 'Петров', 'Олександрович', 'Бакалавр', 'Бюджет', 'Активний',
'2000-03-15', 'м. Харків, вул. Шевченка, 12, кв. 34', '+380666626744',
1),
    ('Олена', 'Сидоренко', 'Ігорівна', 'Бакалавр', 'Бюджет', 'Активний',
'2001-05-20', 'м. Рівне, вул. Леніна, 5, кв. 21', '+380923253117', 1),
    ('Максим', 'Іванов', 'Васильович', 'Бакалавр', 'Контракт',
'Активний', '2000-11-10', 'м. Чернівці, пр. Гагаріна, 30, кв. 56',
'+380971217025', 1),
    ('Анна', 'Коваленко', 'Андріївна', 'Бакалавр', 'Бюджет', 'Активний',
'2002-01-25', 'м. Харків, вул. Київська, 17, кв. 8', '+380980916176', 1),
    ('Віталій', 'Мельник', 'Олегович', 'Бакалавр', 'Бюджет', 'Активний',
'2000-06-05', 'м. Чернівці, пр. Перемоги, 42, кв. 13', '+380979877437',
1),
    ('Людмила', 'Павленко', 'Сергіївна', 'Бакалавр', 'Контракт',
'Активний', '2001-09-18', 'м. Харків, вул. Шевченка, 8, кв. 45',
'+380965529936', 1),
    ('Владислав', 'Григоренко', 'Ігорович', 'Бакалавр', 'Бюджет',
'Активний', '2000-04-03', 'м. Чернівці, пр. Героїв Дніпра, 25, кв. 19',
'+380968530053', 1),
    ('Тетяна', 'Ковальчук', 'Миколаївна', 'Бакалавр', 'Бюджет',
'Активний', '2001-07-22', 'м. Харків, вул. Лермонтова, 14, кв. 27',
'+380996836003', 1);

-- До таблиці "Дисципліни"
INSERT INTO Дисципліни (Назва, [Id викладача], [Тип], [Кількість
кредитів], [Кількість годин на семестр])
VALUES
    ('Основи програмування', 1, 'Загальна', 5, 60), --1
    ('Математичний аналіз', 2, 'Загальна', 3, 72), --2
    ('Теорія ймовірностей', 3, 'Загальна', 2, 72), --3
    ('Диференціальні рівняння', 2, 'Загальна', 5, 72), --4
    ('Математичне моделювання', 3, 'Спеціалізована', 4, 84), --5
    ('Алгоритми та структури даних', 4, 'Загальна', 6, 72), --6
    ('Операційні системи', 5, 'Спеціалізована', 4, 84), --7
    ('Інформаційна безпека', 6, 'Спеціалізована', 3, 84), --8
    ('Інженерія вимог', 7, 'Спеціалізована', 4, 68), --9
    ('Проектування програмного забезпечення', 7, 'Спеціалізована', 5,
90), --10
    ('Тестування програмного забезпечення', 8, 'Загальна', 5, 80), --11
    ('Управління конфігурацією', 5, 'Спеціалізована', 4, 90), --12
    ('Архітектура комп'ютерів', 5, 'Загальна', 5, 72), --13
    ('Оптимізація коду', 4, 'Спеціалізована', 4, 68), --14
    ('Системне програмування на мові асемблера', 4, 'Спеціалізована',
4, 72), --15
    ('Розробка операційних систем', 5, 'Спеціалізована', 6, 94), --16
    ('Бази даних', 9, 'Загальна', 4, 72), --17

```

```

        ('Веб-розробка', 10, 'Загальна', 5, 72), --18
        ('Інформаційні технології в бізнесі', 8, 'Спеціалізована', 5, 78),
--19
        ('Інформаційний пошук і обробка даних', 6, 'Спеціалізована', 5,
68), --20
        ('Алгебра та геометрія', 2, 'Загальна', 4, 68), --21
        ('Математична логіка', 3, 'Загальна', 3, 64), --22
        ('Комплексний аналіз', 3, 'Спеціалізована', 5, 72), --23
        ('Математична статистика', 11, 'Спеціалізована', 5, 74), --24
        ('Статистичний аналіз даних', 11, 'Спеціалізована', 5, 72), --25
        ('Криптографія', 12, 'Спеціалізована', 6, 76), --26
        ('Мережева безпека', 13, 'Спеціалізована', 5, 72), --27
        ('Етика та правові аспекти в кібербезпеці', 14, 'Спеціалізована',
5, 62), --28
        ('Аналіз захисної стійкості', 6, 'Спеціалізована', 5, 82), --29
        ('Схемотехніка та логічне проектування', 5, 'Спеціалізована', 6,
78), --30
        ('Системи високої доступності', 5, 'Спеціалізована', 6, 90); --31

```

-- До таблиці "Викладачі"

```

INSERT INTO Викладачі ([Ім'я], Прізвище, [По-батькові], [Дата
народження], [Науковий ступінь], [Стаж викладання], [Номер телефону],
[Дата прийому])
VALUES
    ('Іван', 'Петренко', 'Олександрович', '1980-05-15', 'Доктор наук',
10, '+380983430922', '2021-09-01'),
    ('Марія', 'Іваненко', 'Василівна', '1975-12-02', 'Кандидат наук', 15,
'+380961658978', '2022-03-15'),
    ('Олександр', 'Сидоренко', 'Ігорович', '1985-08-20', 'Доктор наук',
8, '+380634611955', '2021-02-10'),
    ('Тетяна', 'Ковальчук', 'Миколаївна', '1978-04-30', 'Кандидат наук',
12, '+380951416249', '2022-07-20'),
    ('Андрій', 'Захарчук', 'Анатолійович', '1982-11-08', 'Доктор наук',
9, '+380983306633', '2022-01-05'),
    ('Оксана', 'Литвиненко', 'Сергіївна', '1970-06-25', 'Професор', 20,
'+380503575028', '2022-09-01'),
    ('Віталій', 'Кравець', 'Вікторович', '1988-03-12', 'Кандидат наук',
7, '+380963472387', '2022-04-15'),
    ('Лариса', 'Мельник', 'Олегівна', '1983-09-18', 'Доктор наук', 11,
'+380687110415', '2021-11-30'),
    ('Євген', 'Семененко', 'Петрович', '1973-07-05', 'Професор', 18,
'+380638584581', '2022-06-10'),
    ('Наталія', 'Григоренко', 'Ігорівна', '1986-01-22', 'Доктор наук', 8,
'+380686636208', '2022-08-02'),
    ('Сергій', 'Павленко', 'Олексійович', '1984-10-14', 'Кандидат наук',
10, '+380670235234', '2022-03-20'),
    ('Юлія', 'Ігнат'єва', 'Володимирівна', '1977-02-28', 'Професор', 16,
'+380984486761', '2021-12-15'),
    ('Михайло', 'Коваль', 'Андрійович', '1981-06-08', 'Доктор наук', 12,
'+380632440135', '2022-07-01'),

```

```

        ('Надія', 'Білоус', 'Григорівна', '1976-12-17', 'Професор', 14,
'+380678472462', '2022-05-25'),
        ('Павло', 'Сергієнко', 'Олександрович', '1987-04-03', 'Кандидат
наук', 9, '+380968502907', '2021-09-05');

```

-- До таблиці "Розклад"

```

INSERT INTO Розклад ([Id дисципліни], [Id групи], [Id викладача], [День
тижня], [Номер тижня], [Тип], [Початок заняття], [Кінець заняття],
[Тривалість заняття], Аудиторія)
VALUES

```

```

        -- ПМ-1
        (2, 1, 1, 'Понеділок', 'Парний', 'Лекція', '08:30', '10:00', 90,
'101'),
        (5, 1, 3, 'Понеділок', 'Парний', 'Лекція', '10:15', '11:45', 90,
'102'),
        (2, 1, 2, 'Вівторок', 'Парний', 'Практика', '08:30', '10:00', 90,
'201'),
        (6, 1, 4, 'Вівторок', 'Парний', 'Практика', '10:15', '11:45', 90,
'202'),
        (3, 1, 3, 'Середа', 'Парний', 'Лекція', '08:30', '10:00', 90, '103'),
        (7, 1, 5, 'Середа', 'Парний', 'Лекція', '10:15', '11:45', 90, '104'),
        (4, 1, 2, 'Четвер', 'Парний', 'Практика', '08:30', '10:00', 90,
'203'),
        (8, 1, 6, 'П`ятниця', 'Парний', 'Практика', '10:15', '11:45', 90,
'204'),
        (2, 1, 1, 'Понеділок', 'Непарний', 'Лекція', '08:30', '10:00', 90,
'101'),
        (5, 1, 3, 'Понеділок', 'Непарний', 'Лекція', '10:15', '11:45', 90,
'102'),
        (2, 1, 2, 'Вівторок', 'Непарний', 'Практика', '08:30', '10:00', 90,
'201'),
        (6, 1, 4, 'Вівторок', 'Непарний', 'Практика', '10:15', '11:45', 90,
'202'),
        (3, 1, 3, 'Середа', 'Непарний', 'Лекція', '08:30', '10:00', 90,
'103'),
        (7, 1, 5, 'Середа', 'Непарний', 'Лекція', '10:15', '11:45', 90,
'104'),
        (4, 1, 2, 'Четвер', 'Непарний', 'Практика', '08:30', '10:00', 90,
'203'),
        (8, 1, 6, 'П`ятниця', 'Непарний', 'Практика', '10:15', '11:45', 90,
'204');

```

-- До таблиці "Рейтинг"

```

INSERT INTO Рейтинг ( [Id студента], [Id дисципліни], [Тип контролю],
Оцінка)
VALUES

```

```

        -- ПМ-1
        (1, 11, 'Екзамен', '85'),
        (1, 2, 'Залік', 'Зараховано'),
        (1, 3, 'Екзамен', '92'),
        (1, 4, 'Залік', 'Зараховано'),

```

```

(2, 11, 'Екзамен', '88'),
(2, 2, 'Залік', 'Зараховано'),
(2, 3, 'Екзамен', '78'),
(2, 4, 'Залік', 'Зараховано'),

(3, 11, 'Екзамен', '76'),
(3, 2, 'Залік', 'Зараховано'),
(3, 3, 'Екзамен', '90'),
(3, 4, 'Залік', 'Зараховано'),

(4, 11, 'Екзамен', '78'),
(4, 2, 'Залік', 'Зараховано'),
(4, 3, 'Екзамен', '85'),
(4, 4, 'Залік', 'Зараховано'),

(5, 11, 'Екзамен', '94'),
(5, 2, 'Залік', 'Зараховано'),
(5, 3, 'Екзамен', '87'),
(5, 4, 'Залік', 'Зараховано'),

(6, 2, 'Залік', 'Зараховано'),
(6, 3, 'Екзамен', '79'),
(6, 4, 'Залік', 'Зараховано'),
(6, 11, 'Екзамен', '91'),

(7, 11, 'Екзамен', '88'),
(7, 2, 'Залік', 'Зараховано'),
(7, 3, 'Екзамен', '82'),
(7, 4, 'Залік', 'Зараховано'),

(8, 11, 'Залік', 'Зараховано'),
(8, 2, 'Екзамен', '93'),
(8, 3, 'Залік', 'Зараховано'),
(8, 4, 'Екзамен', '75');

```

USE [База даних університету];

-- Перегляд індексів в таблицях

```

EXEC sp_helpindex 'Факультет';
EXEC sp_helpindex 'Кафедра';
EXEC sp_helpindex 'Спеціальність';
EXEC sp_helpindex 'Група';
EXEC sp_helpindex 'Студенти';
EXEC sp_helpindex 'Викладачі';
EXEC sp_helpindex 'Дисципліни';
EXEC sp_helpindex 'Рейтинг';
EXEC sp_helpindex 'Користувачі';
EXEC sp_helpindex 'Розклад';

```

```

SELECT index_type_desc, page_count, record_count,
avg_page_space_used_in_percent, avg_fragment_size_in_pages
FROM sys.dm_db_index_physical_stats
(DB_ID(N'База даних університету'), OBJECT_ID(N'Рейтинг'), NULL, NULL,
'Detailed')

```

```

SELECT
    i.name AS IndexName,
    i.type_desc AS IndexType,
    c.name AS ColumnName
FROM sys.indexes AS i
INNER JOIN sys.index_columns AS ic ON i.object_id = ic.object_id AND
i.index_id = ic.index_id
INNER JOIN sys.columns AS c ON ic.object_id = c.object_id AND
ic.column_id = c.column_id
WHERE i.object_id = OBJECT_ID('Рейтинг');

```

-- Індекси для таблиць

```

CREATE INDEX IX_Rating_Type ON Рейтинг([Тип контролю]);
CREATE INDEX IX_Students_GroupId ON Студенти([Id групи]);
CREATE INDEX IX_Schedule_GroupId ON Розклад([Id групи]);
CREATE NONCLUSTERED INDEX IX_Department_FacultyId ON Кафедра([Id
факультету]);
CREATE NONCLUSTERED INDEX IX_Specialty_DepartmentId ON Спеціальність([Id
кафедри]);
CREATE NONCLUSTERED INDEX IX_Group_SpecialtyId ON Група([Id
спеціальності]);
CREATE NONCLUSTERED INDEX IX_Student_GroupId ON Студенти([Id групи]);
CREATE NONCLUSTERED INDEX IX_Discipline_TeacherId ON Дисципліни([Id
викладача]);
CREATE NONCLUSTERED INDEX IX_Schedule_DisciplineId ON Розклад([Id
дисципліни]);
CREATE NONCLUSTERED INDEX IX_Schedule_TeacherId ON Розклад([Id
викладача]);

```

-- Перебудова та реорганізація індексів для всіх таблиць

-- Для кластерних індексів

```

DECLARE @TableName NVARCHAR(255);
DECLARE @IndexName NVARCHAR(255);
DECLARE TableCursor CURSOR FOR
    SELECT t.name AS TableName, ix.name AS IndexName
    FROM sys.tables t
    INNER JOIN sys.indexes ix ON t.object_id = ix.object_id
    WHERE ix.type_desc = 'CLUSTERED';
OPEN TableCursor;
FETCH NEXT FROM TableCursor INTO @TableName, @IndexName;
WHILE @@FETCH_STATUS = 0
BEGIN
    DECLARE @RebuildSQL NVARCHAR(MAX);
    SET @RebuildSQL = 'ALTER INDEX ' + @IndexName + ' ON ' + @TableName +
    ' REBUILD WITH (FILLFACTOR = 80, ONLINE = ON)';

```

```

EXEC sp_executesql @RebuildSQL;
FETCH NEXT FROM TableCursor INTO @TableName, @IndexName;
END
CLOSE TableCursor;
DEALLOCATE TableCursor;

-- Для некластерних індексів
DECLARE @TableNameNonClustered NVARCHAR(255);
DECLARE @IndexNameNonClustered NVARCHAR(255);
DECLARE TableCursorNonClustered CURSOR FOR
    SELECT t.name AS TableName, ix.name AS IndexName
    FROM sys.tables t
    INNER JOIN sys.indexes ix ON t.object_id = ix.object_id
    WHERE ix.type_desc = 'NONCLUSTERED';
OPEN TableCursorNonClustered;
FETCH NEXT FROM TableCursorNonClustered INTO @TableNameNonClustered,
@IndexNameNonClustered;
WHILE @@FETCH_STATUS = 0
BEGIN
    DECLARE @ReorganizeSQL NVARCHAR(MAX);
    SET @ReorganizeSQL = 'ALTER INDEX ' + @IndexNameNonClustered + ' ON '
+ @TableNameNonClustered + ' REORGANIZE WITH (LOB_COMPACTION = ON);';
    EXEC sp_executesql @ReorganizeSQL;
    FETCH NEXT FROM TableCursorNonClustered INTO @TableNameNonClustered,
@IndexNameNonClustered;
END
CLOSE TableCursorNonClustered;
DEALLOCATE TableCursorNonClustered;

-- Створення повної резервної копії
USE [База даних університету];
BACKUP DATABASE [База даних університету]
TO DISK = 'C:\sql\backups\full_backup.bak'
WITH INIT, FORMAT, COMPRESSION,
NAME = 'Full Backup',
DESCRIPTION = 'Full Backup';

-- Резервування журналів транзакцій
BACKUP LOG [База даних університету]
TO DISK = 'C:\sql\logs\backups_logs.trn'
WITH INIT, FORMAT, COMPRESSION;

-- Створення диференціальної копії
DECLARE @path NVARCHAR(MAX);
SET @path = 'C:\sql\backups\backup_diff.bak';

BACKUP DATABASE [База даних університету]
TO DISK = @path
WITH DIFFERENTIAL, NOFORMAT, NOINIT,

```

```
NAME = 'Differential Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10;  
USE [База даних університету]
```

```
GO
```

```
CREATE ROLE Administrator;  
CREATE ROLE Teacher;  
CREATE ROLE Student;  
CREATE ROLE Technician;
```

```
-- Створення користувача Адміністратор
```

```
CREATE LOGIN AdminUser WITH PASSWORD = '12345';  
CREATE USER AdminUser FOR LOGIN AdminUser;
```

```
-- Додаємо користувача до ролі адміністратора бази даних
```

```
EXEC sp_addrolemember 'db_owner', 'AdminUser';
```

```
-- Для кожної таблиці надаємо права доступу відповідно до вказаних прав  
(1 – перегляд, 2 – редагування, 3 – видалення, 4 – повний доступ)
```

```
GRANT SELECT ON [dbo].[Факультет] TO AdminUser;  
GRANT UPDATE, INSERT, DELETE ON [dbo].[Факультет] TO AdminUser;  
GRANT SELECT ON [dbo].[Кафедра] TO AdminUser;  
GRANT UPDATE, INSERT, DELETE ON [dbo].[Кафедра] TO AdminUser;  
GRANT SELECT ON [dbo].[Спеціальність] TO AdminUser;  
GRANT UPDATE, INSERT, DELETE ON [dbo].[Спеціальність] TO AdminUser;  
GRANT SELECT ON [dbo].[Група] TO AdminUser;  
GRANT UPDATE, INSERT, DELETE ON [dbo].[Група] TO AdminUser;  
GRANT SELECT ON [dbo].[Студенти] TO AdminUser;  
GRANT UPDATE, INSERT, DELETE ON [dbo].[Студенти] TO AdminUser;  
GRANT SELECT ON [dbo].[Викладачі] TO AdminUser;  
GRANT UPDATE, INSERT, DELETE ON [dbo].[Викладачі] TO AdminUser;  
GRANT SELECT ON [dbo].[Дисципліни] TO AdminUser;  
GRANT UPDATE, INSERT, DELETE ON [dbo].[Дисципліни] TO AdminUser;  
GRANT SELECT ON [dbo].[Рейтинг] TO AdminUser;  
GRANT UPDATE, INSERT, DELETE ON [dbo].[Рейтинг] TO AdminUser;  
GRANT SELECT ON [dbo].[Користувачі] TO AdminUser;  
GRANT UPDATE, INSERT, DELETE ON [dbo].[Користувачі] TO AdminUser;  
GRANT SELECT ON [dbo].[Позклад] TO AdminUser;  
GRANT UPDATE, INSERT, DELETE ON [dbo].[Позклад] TO AdminUser;
```

```
-- Створення користувача Викладач
```

```
CREATE LOGIN TeacherUser WITH PASSWORD = '12345';  
CREATE USER TeacherUser FOR LOGIN TeacherUser;
```

```
-- Додаємо користувача до ролі вчителя бази даних
```

```
EXEC sp_addrolemember 'db_datareader', 'TeacherUser';
```

```
-- Для кожної таблиці надаємо права доступу відповідно до вказаних прав  
(0 – без доступу, 1 – перегляд, 2 – редагування, 3 – видалення, 4 –  
повний доступ)
```

```
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Факультет] TO TeacherUser;
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Кафедра] TO TeacherUser;
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Спеціальність] TO
TeacherUser;
GRANT SELECT ON [dbo].[Група] TO TeacherUser;
GRANT SELECT, INSERT ON [dbo].[Студенти] TO TeacherUser;
GRANT SELECT, INSERT ON [dbo].[Викладачі] TO TeacherUser;
GRANT SELECT, UPDATE ON [dbo].[Дисципліни] TO TeacherUser;
GRANT SELECT, UPDATE ON [dbo].[Рейтинг] TO TeacherUser;
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Користувачі] TO
TeacherUser;
GRANT SELECT, UPDATE ON [dbo].[Розклад] TO TeacherUser;
```

-- Створення користувача Студент

```
CREATE LOGIN StudentUser WITH PASSWORD = '12345';
CREATE USER StudentUser FOR LOGIN StudentUser;
```

-- Додаємо користувача до ролі студента бази даних

```
EXEC sp_addrolemember 'db_datareader', 'StudentUser';
```

```
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Факультет] TO StudentUser;
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Кафедра] TO StudentUser;
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Спеціальність] TO
StudentUser;
GRANT SELECT ON [dbo].[Група] TO StudentUser;
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Студенти] TO StudentUser;
GRANT SELECT ON [dbo].[Викладачі] TO StudentUser;
GRANT SELECT ON [dbo].[Дисципліни] TO StudentUser;
GRANT SELECT ON [dbo].[Рейтинг] TO StudentUser;
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Користувачі] TO
StudentUser;
GRANT SELECT ON [dbo].[Розклад] TO StudentUser;
```

-- Створення користувача Технік

```
CREATE LOGIN TechnicianUser WITH PASSWORD = '12345';
CREATE USER TechnicianUser FOR LOGIN TechnicianUser;
```

```
GRANT SELECT, UPDATE ON [dbo].[Факультет] TO TechnicianUser;
GRANT SELECT, UPDATE ON [dbo].[Кафедра] TO TechnicianUser;
GRANT SELECT, UPDATE ON [dbo].[Спеціальність] TO TechnicianUser;
GRANT SELECT, UPDATE ON [dbo].[Група] TO TechnicianUser;
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Студенти] TO
TechnicianUser;
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Викладачі] TO
TechnicianUser;
GRANT SELECT, UPDATE ON [dbo].[Дисципліни] TO TechnicianUser;
GRANT SELECT, UPDATE ON [dbo].[Рейтинг] TO TechnicianUser;
DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Користувачі] TO
TechnicianUser;
```

```

DENY SELECT, UPDATE, INSERT, DELETE ON [dbo].[Розклад] TO TechnicianUser;

---
=====
=====
EXEC sp_helprole

EXEC sp_helpuser

SELECT name,type_desc FROM sys.sql_logins

GO
EXECUTE AS USER = 'TeacherUser'
SELECT * FROM [Рейтинг]

SELECT * FROM [Користувачі]
DELETE FROM [Студенти] WHERE [Id групи] = 7

-----ШИФРУВАННЯ АСИМЕТРИЧНИМ КЛЮЧЕМ-----
--

CREATE ASYMMETRIC KEY Asym_Key
WITH ALGORITHM = RSA_2048
ENCRYPTION BY PASSWORD = '12345';

-- Створення копії таблиці-----
-----
CREATE TABLE [Студент_зашифровано] (
    [Id] INT PRIMARY KEY,
    [Ім'я] NVARCHAR(MAX),
    [Прізвище] NVARCHAR(MAX),
    [По-батькові] NVARCHAR(MAX),
    [Ступінь] NVARCHAR(MAX),
    [Тип навчання] NVARCHAR(MAX),
    [Статус] NVARCHAR(MAX),
    [Дата народження] NVARCHAR(MAX),
    [Адреса проживання] NVARCHAR(MAX),
    [Номер телефона] NVARCHAR(MAX),
    [Id групи] NVARCHAR(MAX)
);

-- Вставка зашифрованих даних-----
-----
INSERT INTO [Студент_зашифровано] (Id, [Ім'я], [Прізвище], [По-батькові],
[Ступінь], [Тип навчання], [Статус], [Дата народження], [Адреса
проживання], [Номер телефона], [Id групи] )
SELECT
    Id,
    CONVERT(NVARCHAR(MAX), EncryptByAsymKey(AsymKey_ID('Asym_Key'),
CAST([Ім'я] AS NVARCHAR(MAX))))) ,

```

```

        CONVERT(NVARCHAR(MAX), EncryptByAsymKey(AsymKey_ID('Asym_Key'),
CAST([Прізвище] AS NVARCHAR(MAX))))) ,
        CONVERT(NVARCHAR(MAX), EncryptByAsymKey(AsymKey_ID('Asym_Key'),
CAST([По-батькові] AS NVARCHAR(MAX))))) ,
        [Ступінь],
        CONVERT(NVARCHAR(MAX), EncryptByAsymKey(AsymKey_ID('Asym_Key'),
CAST([Тип навчання] AS NVARCHAR(MAX))))) ,
        [Статус],
        CONVERT(NVARCHAR(MAX), EncryptByAsymKey(AsymKey_ID('Asym_Key'),
CAST(CONVERT(VARCHAR, [Дата народження]) AS NVARCHAR(MAX))))) ,
        CONVERT(NVARCHAR(MAX), EncryptByAsymKey(AsymKey_ID('Asym_Key'),
CAST([Адреса проживання] AS NVARCHAR(MAX))))) ,
        CONVERT(NVARCHAR(MAX), EncryptByAsymKey(AsymKey_ID('Asym_Key'),
CAST(CONVERT(VARCHAR, [Номер телефона]) AS NVARCHAR(MAX))))) ,
        [Id групи]
FROM [Студенти];

SELECT * FROM [Студент_зашифровано]

```

```

-- Вибірка розшифрованих даних-----
-----
SELECT
    Id,
    CONVERT(VARCHAR(30), CONVERT(NVARCHAR(MAX),
DecryptByAsymKey(AsymKey_ID('Asym_Key'), [Ім'я], N'12345')))) AS [Ім'я],
    CONVERT(VARCHAR(30), CONVERT(NVARCHAR(MAX),
DecryptByAsymKey(AsymKey_ID('Asym_Key'), [Прізвище], N'12345')))) AS
[Прізвище],
    CONVERT(VARCHAR(30), CONVERT(NVARCHAR(MAX),
DecryptByAsymKey(AsymKey_ID('Asym_Key'), [По-батькові], N'12345')))) AS
[По-батькові],
    [Ступінь],
    CONVERT(VARCHAR(30), CONVERT(NVARCHAR(MAX),
DecryptByAsymKey(AsymKey_ID('Asym_Key'), [Тип навчання], N'12345')))) AS
[Тип навчання],
    [Статус],
    CONVERT(DATE, CONVERT(NVARCHAR(MAX),
DecryptByAsymKey(AsymKey_ID('Asym_Key'), [Дата народження], N'12345'))))
AS [Дата народження],
    CONVERT(VARCHAR(30), CONVERT(NVARCHAR(MAX),
DecryptByAsymKey(AsymKey_ID('Asym_Key'), [Номер телефона], N'12345')))) AS
[Номер телефона],
    [Id групи]
FROM [Студент_зашифровано]

DROP TABLE [Студент_зашифровано]

```

-----ШИФРУВАННЯ ЗА ДОПОМОГОЮ СЕРТИФІКАТУ-----

```
CREATE CERTIFICATE Cert_Key
ENCRYPTION BY PASSWORD = 'p1234'
WITH SUBJECT = 'Cert_Key_Encry';
```

-- Створення копії таблиці із зашифрованими даними за допомогою
сертифіката-----

```
CREATE TABLE [Викладачі_зашифровано] (
    Id INT PRIMARY KEY,
    [Ім'я] NVARCHAR(Max),
    [Прізвище] NVARCHAR(Max),
    [По-батькові] NVARCHAR(Max),
    [Дата народження] NVARCHAR(Max),
    [Науковий ступінь] NVARCHAR(Max),
    [Стаж викладання] NVARCHAR(Max),
    [Номер телефона] NVARCHAR(Max),
    [Дата прийому] NVARCHAR(Max)
);
```

-- Вставка зашифрованих даних в нову таблицю з використанням сертифіката-----

```
INSERT INTO [Викладачі_зашифровано] (Id, [Ім'я], [Прізвище], [По-
батькові], [Дата народження], [Науковий ступінь], [Стаж викладання],
[Номер телефона], [Дата прийому])
SELECT
    Id,
    EncryptByCert(Cert_ID('Cert_Key'), CAST([Ім'я] AS NVARCHAR(Max))),
    EncryptByCert(Cert_ID('Cert_Key'), CAST([Прізвище] AS
NVARCHAR(Max))),
    EncryptByCert(Cert_ID('Cert_Key'), CAST([По-батькові] AS
NVARCHAR(Max))),
    EncryptByCert(Cert_ID('Cert_Key'), CAST([Дата народження] AS
NVARCHAR(Max))),
    [Науковий ступінь],
    [Стаж викладання],
    EncryptByCert(Cert_ID('Cert_Key'), CAST([Номер телефона] AS
NVARCHAR(Max))),
    EncryptByCert(Cert_ID('Cert_Key'), CAST([Дата прийому] AS
NVARCHAR(Max)))
FROM [Викладачі];
```

```
Select * from [Викладачі_зашифровано]
```

-- Вибірка даних з зашифрованої таблиці з використанням сертифіката-----

```
SELECT
```

```

    Id,
    CONVERT(NVARCHAR(50), DecryptByCert(Cert_ID('Cert_Key'), [Ім'я],
N'p1234')) AS [Ім'я],
    CONVERT(NVARCHAR(50), DecryptByCert(Cert_ID('Cert_Key'), [Прізвище],
N'p1234')) AS [Прізвище],
    CONVERT(NVARCHAR(50), DecryptByCert(Cert_ID('Cert_Key'), [По-
батькові], N'p1234')) AS [По-батькові],
    CONVERT(DATE, CONVERT(NVARCHAR(MAX),
DecryptByCert(Cert_ID('Cert_Key'), [Дата народження], N'p1234')))) AS
[Дата народження],
    [Науковий ступінь],
    [Стаж викладання],
    CONVERT(DATE, CONVERT(NVARCHAR(MAX),
DecryptByCert(Cert_ID('Cert_Key'), [Дата прийому], N'p1234')))) AS [Дата
прийому]
FROM [Викладачі_зашифровано];

```