

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
(повна назва інституту/факультету)

КАФЕДРА ІНФОРМАТИКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ
(повна назва кафедри)

КУРСОВА РОБОТА

з дисципліни «Бази даних»
назва дисципліни

на тему: База даних для підтримки екзаменаційної та залікової сесії

Студентки 2 курсу ІТ-01 групи
спеціальності 121 “Інженерія програмного
забезпечення”

Дмитрієвої І. І.
(прізвище та ініціали)

Керівник

ст. викладач канд. техн. наук Марченко О. І.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка ECTS _____

Члени комісії _____
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

ЗМІСТ

ВСТУП	5
1. Опис предметного середовища.....	8
2. Опис вхідних даних	10
3. Постановка завдання.....	10
4.1. Концептуальна модель бази даних.....	12
4.2. Опис сутностей та їх атрибутів	13
4.3. Діаграма "сутність-зв'язок"	16
4.4. Модель користувачів	18
5. Логічна модель бази даних.....	18
6. Реалізація бази даних.....	22
ВИСНОВКИ.....	42
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	43

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Інформатики та обчислювальної техніки
(повна назва)

Кафедра Інформатики та програмної інженерії
(повна назва)

Дисципліна Бази даних

Курс 2 Група IT-01 Семестр 1

**З А В Д А Н Н Я
НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Дмитрієва Ірина Ігорівна
(прізвище, ім'я, по батькові)

1. Тема роботи: База даних для підтримки екзаменаційної та залікової сесії
керівник роботи: ст. викладач Марченко Олена Іванівна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
2. Строк подання студентом роботи 24.12.2021
3. Вихідні дані до роботи: методичні вказівки до виконання курсової роботи з дисципліни “Бази даних” першого бакалаврського рівня вищої освіти; варіант з темою курсової роботи, що визначається номером списку у групі; задачі, положення, завдання, тематика та вимоги до виконання курсової роботи.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вивчення літератури, аналіз предметного середовища, побудова ER-діаграми, побудова реляційної схеми, створення бази даних, створення користувачів бази даних, імпорт даних з використанням засобів СУБД, створення запитів до розробленої БД, оптимізація роботи запитів (за необхідності)
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
-
6. Дата видачі завдання: 31.10.2021

№ з/П	Назва етапів виконання курсового проекту	Строк виконання етапів проекту	Примітка
1	Вивчення літератури	05.12.2021	
2	Аналіз предметного середовища	05.12.2021	
3	Побудова ER-діаграми	10.12.2021	
4	Побудова реляційної схеми	10.12.2021	
5	Створення бази даних	15.12.2021	
6	Створення користувачів бази даних	17.12.2021	
7	Імпорт даних з використанням засобів СУБД	20.12.2021	
8	Створення запитів до розробленої БД	20.12.2021	
9	Оптимізація роботи запитів (за необхідності)	20.12.2021	

Студент

_____ Дмитрієва Ірина Ігорівна
(підпис) (прізвище та ініціали)

Керівник роботи

_____ Марченко О. І.
(підпис) (прізвище та ініціали)

ВСТУП

В сучасних ринкових умовах, коли швидкість життя невинно набирає оберти, відбувається розвиток в науково-технологічному напрямі, з'являється велика кількість інформації. Це зумовлює пошук нових підходів використання та обробки даних, систему автоматизації для забезпечення прийняття ефективних управлінських рішень, конкурентоспроможного ведення бізнесу.

Нині кожен суб'єкт господарювання прагне до підвищення конкурентоспроможності. Сьогодення вимагає від керівника швидких змін та використання новітніх технологій для збереження, збільшення інтегрованого розвитку для стабільного майбутнього.

Діджиталізація – це той чинник, який надає значні переваги для розвитку в цьому напрямку. Економічне зростання та трансформація відбуваються завдяки розширенню та більш зручному використанню цифрових технологій. Сфера використання інформаційних систем нині настільки широка, що немає такої області, де їх застосування було б недоцільним.

Отже, швидкі зміни для розвитку та конкурентоспроможності будь-якої установи, підприємства, організації в напрямку використання цифрових перетворень є невідворотними.

Серед іншого, застосування сучасних інформаційних технологій актуальне і для закладів вищої освіти з метою підвищення ефективності робіт з планування, обліку, оцінювання якості навчального процесу, статистичного аналізу та опрацювання зібраних даних. Використання інформаційних технологій надає можливість швидко і якісно здійснювати ці процеси.

Основним елементом будь якої сучасної інформаційної системи є інформація, що зберігається в базах даних.

Метою курсової роботи є проектування бази даних для підтримки екзаменаційної та залікової сесії (далі – БД “Сесія”).

Впровадження такої бази даних забезпечить автоматизацію процесів ведення обліку як індивідуальної студентської успішності, так і в розрізі

навчальних груп. Також система може використовуватися викладачами для обліку успішності студентів з предметів, що ними викладаються.

Основними завданнями курсової роботи є: [1]

- аналіз предметного середовища, виявлення необхідного набору сутностей, визначення атрибутів та зв'язків між об'єктами;
- розроблення ER-моделі та її опис в одній з нотацій;
- розроблення моделі користувачів бази даних з описом їх прав;
- побудова реляційної схеми з ER-моделі;
- створення бази даних;
- створення користувачів бази даних;
- імпорт даних з використанням засобів СУБД в створену базу даних;
- розробка оптимізація роботи SQL-запитів для забезпечення потреб користувачів.

В ході виконання курсової роботи був проведений аналіз можливостей сучасних систем керування баз даних (далі – СКБД) та для вирішення поставленого завдання обрана MySQL як швидка, надійна та універсальна реляційна СКБД, за деякими оцінками найпопулярніша у світі. Хоча їй не вистачає великих можливостей PostgreSQL, вона відмінно підходить для широкого спектру додатків.

Основними причинами вибору є наступні характеристики СКБД MySQL [2], [3]:

- 1) відкритий вихідний код, безкоштовність - СУБД поширюється її власником (корпорацією Oracle) під "General Public License" (GNU), якою "забезпечуються" усе open-source програмне забезпечення (далі – ПЗ);
- 2) довга історія: MySQL доступна з 1995 року;
- 3) підтримка MySQL розвиненою відданою спільнотою ентузіастів, що завжди допоможе вирішити проблеми;
- 4) висока швидкість обробки даних, стабільність, надійність;

5) високий рівень кросплатформенності даних та коду, створених за допомогою MySQL;

6) MySQL входить у стандартний стек LAMP – набір веб-додатків Linux, Apache HTTP Server, MySQL та PHP з відкритим вихідним кодом, а також до складу іншого сучасного ПЗ – Drupal, Joomla и WordPress тощо;

7) серед користувачів MySQL DB: Facebook, Google, Flickr, GitHub, NASA, Netflix, Spotify, Tesla, Twitter, Uber, ВМФ США, WeChat, Вікіпедія, YouTube, Zappos, Zendesk.

8) висока оцінка користувачів: рейтинг MySQL становить 4,4 зірки (з п'яти) за підсумками 1526 відгуків на G2 Crowd (<https://www.g2.com/products/mysql/reviews/mysql-review-1775714>) [4].

1. Опис предметного середовища

Під час розробки БД “Сесія” був проведений аналіз інформаційних потоків, об’єктів та процесів, що виникають під час виконання робіт з планування навчання та оцінювання студентів, обліку результатів здачі екзаменів та заліків, статистичного аналізу накопичених результатів.

За результатами проведеної роботи визначені характеристики предметного середовища та бізнес-правила, які є суттєвими при проектуванні бази даних [5]:

- в університеті існує декілька факультетів;
- факультет об’єднує декілька кафедр;
- на кафедрах навчаються групи студентів за певними спеціальностями;
- до складу групи входять студенти;
- на кожній кафедрі працює декілька викладачів;
- на кожній спеціальності викладається ряд дисциплін;
- з кожної дисципліни своєї спеціальності студенти складають екзамен або залік.

В основі БД “Сесія” лежать контрольні заходи, які є необхідним елементом зворотного зв’язку у процесі навчання для визначення відповідності рівня набутих студентами знань, умінь та навичок вимогам нормативних документів щодо вищої освіти і забезпечення своєчасного коригування навчального процесу.

Відповідно до вимог Положення про поточний, календарний та семестровий контроль результатів навчання в КПІ ім. Ігоря Сікорського, затвердженого наказом ректора КПІ ім. Ігоря Сікорського № 7-137 від 05.08.2020, в освітньому процесі Університету використовуються такі основні види контрольних заходів: вхідний, ректорський, поточний, календарний та підсумковий (семестровий контроль та атестація) контроль [6].

Залік та екзамен, які є предметом досліджень, є видами семестрового контролю.

Семестровий контроль проводиться для встановлення рівня досягнення здобувачами програмних результатів навчання з навчальної дисципліни (освітнього компонента), як правило, за семестр. Семестровий контроль проводиться відповідно до навчального плану у вигляді заліку або екзамену в терміни, встановлені графіком навчального процесу.

Екзамен є видом семестрового контролю для оцінювання рівня засвоєння здобувачем теоретичного та практичного матеріалу з навчальної дисципліни (освітнього компонента), що проводиться в період екзаменаційної сесії за затвердженим розкладом.

Залік є видом семестрового контролю, який планується за відсутності екзамену і передбачає можливість однозначно визначити рівень засвоєння здобувачем теоретичного та практичного матеріалу з навчальної дисципліни (освітнього компонента) за результатами поточного контролю навчання здобувача упродовж семестру.

Окремим видом заліку є захист курсового проекту/роботи або захист звіту практики.

Оцінювання результатів семестрового контролю здійснюється згідно з рейтинговою системою оцінювання результатів навчання здобувачів (далі - РСО) з певної навчальної дисципліни (освітнього компонента), яка містить критерії оцінювання, що формуються з урахуванням вимог Положення про систему оцінювання результатів навчання в КПІ ім. Ігоря Сікорського.

Семестровий контроль проводить екзаменатор. За наявності поважних причин у екзаменатора (стан здоров'я, відрядження тощо), завідувач кафедри може призначати екзаменатором іншого НПП для проведення запланованого заходу семестрового контролю.

Залік проводиться в період останніх двох тижнів теоретичного навчання у семестрі, як правило, на останньому за розкладом занятті з відповідної навчальної дисципліни (освітнього компонента).

Екзаменатор вносить результати семестрового контролю (оцінки) до відомості семестрового контролю та залікової книжки здобувача.

У відомості семестрового контролю навпроти прізвища конкретного здобувача екзаменатор проставляє оцінки за 100-бальною та університетською шкалою або робить відповідні записи (табл.1):

Таблиця 1

100-бальна шкала оцінок	Університетська шкала оцінок
95...100	Відмінно
85...94	Дуже добре
75...84	Добре
65...74	Задовільно
60...64	Достатньо
Менше 60	Незадовільно
Невиконання умов допуску до семестрового контролю	Не допущений
Порушення принципів академічної доброчесності або морально-етичних норм поведінки	Усунений

2. Опис вхідних даних

Виходячи з аналізу предметного середовища були визначені вхідні для введення в базу даних:

- інформація про студентів;
- інформація про предмети навчання та види контролю по них;
- інформація про екзаменаторів;
- інформація про підрозділи: факультети, кафедри, групи;
- інформація про результати оцінювання студентів.

3. Постановка завдання

3.1 Необхідно розробити БД, яка реалізує наступні вимоги до функціональності, надійності БД та захисту даних:

1) збереження протягом усього часу навчання студента персональної інформації про кожного студента, успішність по кожному предмету та розподіл студентів за групами;

2) виведення у зручній формі даних за наступними запитам користувача:

- пошук заданого студента по прізвищу (з уточненням за необхідності по іншим атрибутам) або номеру залікової книжки;

- вибірка всіх даних про результати оцінювання заданого студента за поточний навчальний рік та за весь час навчання;

- вибірка даних по студентах, що мають академічну заборгованість;

- середній бал з кожного предмета;

- розрахунок кількості студентів за групами;

- середня оцінка з предметів та груп;

3) автоматизація обробки інформації при наступних бізнес-операціях:

- прийом нового студента;

- корекція даних про студента та його успішність;

- формування особистої відомості успішності;

4) виведення на друк документів:

- відомість середньої успішності факультету;

- список студентів за групами;

- відомість успішності студентів з груп та предметів.

5) адекватність відображення предметної галузі університету, зокрема ефективності навчального процесу (повнота, цілісність, несуперечність, актуальність даних);

6) реалізація вимог із захисту інформації, цілісності та доступності даних:

- можливість використання бази даних різними категоріями користувачів;

- забезпечення взаємної незалежності програм та даних;

- забезпечення надійності БД;

- захист даних від випадкового та навмисного руйнування;

- можливість швидкого та повного відновлення даних у разі збоїв у системі;

7) дружність інтерфейсу.

4. Проектування бази даних

4.1. Концептуальна модель бази даних

З концептуального проектування починається створення концептуальної схеми БД, в основі якої лежить концептуальна (інфологічна) модель.

Призначення концептуальної моделі полягає у наданні найбільш загальних способів збору та представлення інформації, що зберігається у базі даних.

Найбільш поширеною концептуальною моделлю є модель "сутність – зв'язок" (Entity Relationship model, ER-модель), яка складається із сутностей (entities), зв'язків (relationships), атрибутів (attributes). [7]

Моделювання даних відображає логічну структуру даних так само, як блок-схеми алгоритмів відображають логічну структуру програми.

ER-моделювання являє собою низхідний підхід до проектування БД, який починається з визначення найбільш важливих даних – сутностей, і зв'язків між даними, які повинні бути представлені в моделі. Потім в модель заноситься інформація про властивості сутностей і зв'язків – атрибути, а також всі обмеження, які відносяться до сутностей, зв'язків і атрибутів. ER-модель дає графічне представлення логічних об'єктів і їх відношень в структурі БД.

Сутність має унікальне ім'я у межах системи (БД) та дозволяє моделювати клас однотипних об'єктів. Оскільки сутність відповідає деякому класу однотипних об'єктів, то передбачається, що в системі існує багато екземплярів даної сутності.

Об'єкт, якому відповідає сутність, має набір атрибутів, які характеризують його властивості. При цьому набір атрибутів повинен бути таким, щоби можна було розрізняти конкретні екземпляри сутності. Значення кожного атрибута вибирають з відповідної множини значень, яка включає всі потенційні значення, які можуть бути присвоєні атрибуту. Ця множина значень називається доменом.

Набір атрибутів, що однозначно ідентифікує конкретний екземпляр сутності, називають ключовим.

Між сутностями встановлюються зв'язки, які вказують яким чином сутності співвідносяться або взаємодіють між собою. Розрізняють такі зв'язки:

- між двома сутностями (бінарний зв'язок);
- між трьома сутностями (тернарний зв'язок);
- між N сутностями (N-арний зв'язок);
- між однією сутністю (рекурсивний зв'язок).

Найбільш поширеними є бінарні зв'язки. Зв'язок показує яким чином екземпляри сутностей зв'язані між собою. Бінарні зв'язки бувають:

- 1:1 (один до одного);
- 1:M (один до багатьох);
- N:M (багато до багатьох).

Виходячи з наведеного, в ході проектування бази даних “Сесія” були визначені сутності, зв'язки між ними, атрибути та побудована ER-модель в нотації П.Чена.

4.2. Опис сутностей та їх атрибутів

У відповідності з описом предметної області були визначені сутності, які необхідно зберігати в базі даних, а також їхні основні характеристики (атрибути) (табл. 2 – табл. 11).

Сутність “Студент”

Таблиця 2

Назва атрибута	Тип даних	Приклад представлення даних екземпляром сутності	Ключ
Код	Ціле	7	+
Прізвище	Строка	Григоренко	
Ім'я	Строка	Сергій	
По батькові	Строка	Петрович	
№ залікової книжки	Строка	ВН23423	

Сутність “Екзаменатор”

Таблиця 3

Назва атрибута	Тип даних	Приклад представлення даних екземпляром сутності	Ключ
Код	Ціле	1	+
Прізвище	Строка	Тарасюк	
Ім'я	Строка	Роман	
По батькові	Строка	Володимирович	
Посада	Строка	Старший викладач	
Вчене звання	Строка	Доцент	
Науковий ступінь	Строка	Канд. тех. наук	

Сутність Факультет”

Таблиця 4

Назва атрибута	Тип даних	Приклад представлення даних екземпляром сутності	Ключ
Код	Ціле	1	+
Повна назва	Строка	Факультет інформатики та обчислювальної техніки	
Скорочена назва	Строка	ФІОТ	

Сутність “Кафедра”

Таблиця 5

Назва атрибута	Тип даних	Приклад представлення даних екземпляром сутності	Ключ
Код	Ціле	1	+
Повна назва	Строка	Кафедра інформатики та програмної інженерії	
Скорочена назва	Строка	ІІІ	

Сутність “Група”

Таблиця 6

Назва атрибута	Тип даних	Приклад представлення даних екземпляром сутності	Ключ
----------------	-----------	--------------------------------------------------	------

Код	Ціле	1	+
Назва	Строка	IT-01	
Рік вступу	Ціле	2020	

Сутність “Дисципліна”

Таблиця 7

Назва атрибута	Тип даних	Приклад представлення даних екземпляром сутності	Ключ
Код	Ціле	1	+
Назва	Строка	Бази даних	
Кількість годин	Ціле	50	

Сутність “Реєстр відомостей семестрового контролю”

Таблиця 8

Назва атрибута	Тип даних	Приклад представлення даних екземпляром сутності	Ключ
Код	Ціле	1	+
Дата контролю	Дата	28.12.2021	
Вид контролю	Строка	Залік	

Сутність “Відомість семестрового контролю”

Таблиця 9

Назва атрибута	Тип даних	Приклад представлення даних екземпляром сутності	Ключ
Код	Ціле	1	+
Бал	Ціле	34	
Накопичений бал за семестр	Ціле	45	

Сутність “Додаткові бали”

Таблиця 10

Назва атрибута	Тип даних	Приклад представлення даних екземпляром сутності	Ключ
----------------	-----------	--------------------------------------------------	------

Код	Ціле	1	+
Заслуга	Строка	Участь у спартакіаді	
Бал	Ціле	10	

Сутність “Спеціальності”

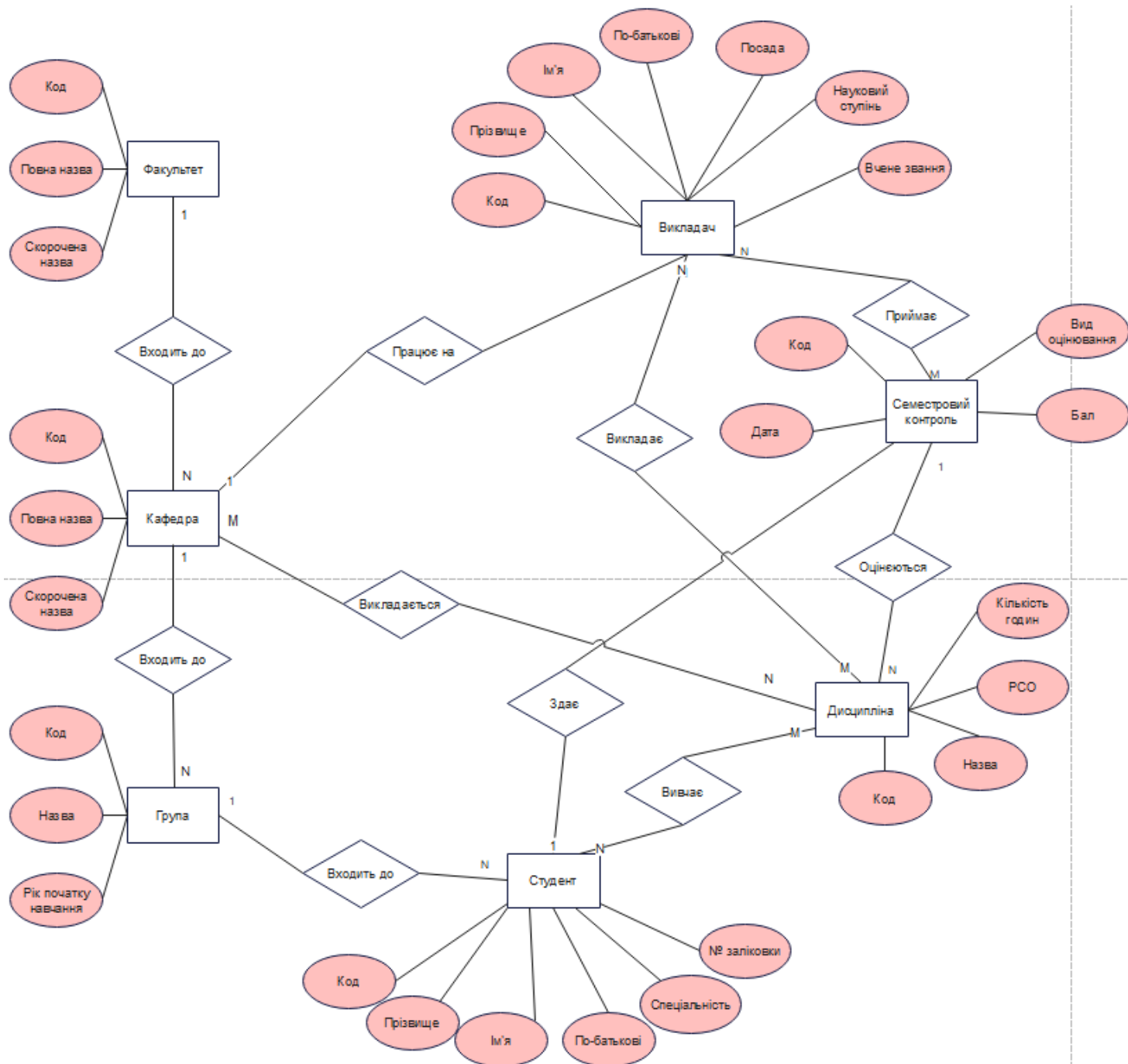
Таблиця 11

Назва атрибута	Тип даних	Приклад представлення даних екземпляром сутності	Ключ
Код	Ціле	123	+
Назва	Строка	Комп’ютерна інженерія	

4.3. Діаграма "сутність-зв'язок"

На основі задач, які були поставлені, та на основі аналізу предметної області, побудована ER-діаграма в нотації А.Чена, на якій відображені всі сутності, їх атрибути та зв'язки між сутностями. (рис. 1).

Рисунок 1



4.4. Модель користувачів

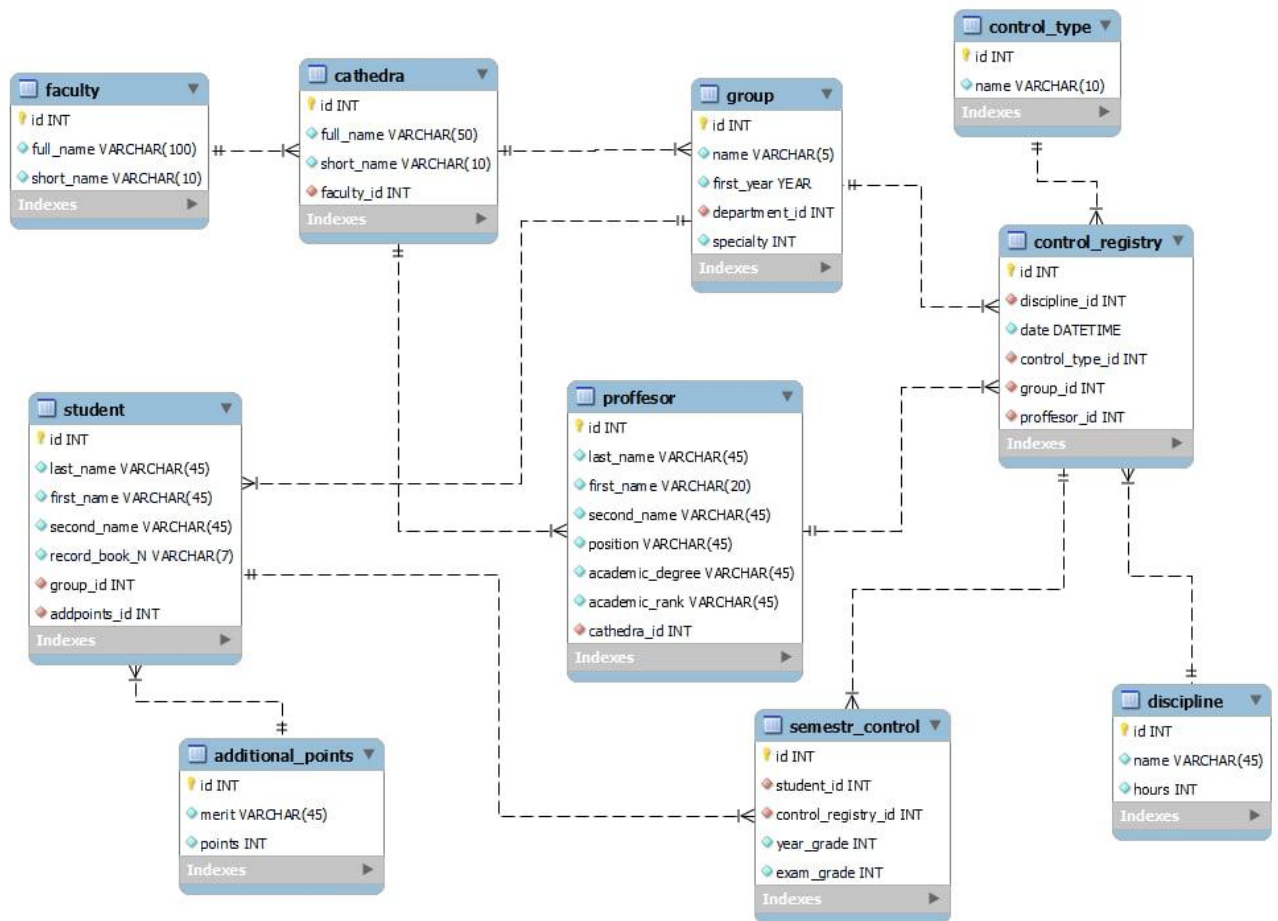
Виконання робіт з базою даних, та забезпечення її функціонування передбачається такими групами користувачів:

- адміністратор(и) бази даних – здійснює розгортання, налаштування програмного забезпечення та структури бази даних, її резервне копіювання, відновлення (за необхідності), заведення (блокування) облікових записів користувачів, надання їх прав доступу відповідно до затверджених політик безпеки (таблиць розмежування доступу);
- викладачі (екзаменатори) – заводять дані за результатами проведених заходів з оцінювання студентів (приймання екзаменів, заліків, захисту курсових робіт тощо), отримують дані успішності студентів з предметів, що ними викладаються;
- співробітники кафедр та факультетів, які відповідають за планування та контроль навчального процесу, оцінювання студентів, опрацювання статистичних звітів – заводять установчі дані по студентах, викладачах, дисциплінах, отримують відомості з результатами контроль навчання.

5. Логічна модель бази даних

5.1. На етапі створення логічної моделі була проаналізована розроблена ER-модель на предмет наявності зв'язків, які є несумісними з реляційною моделлю (двосторонніх зв'язків M:N, складних зв'язків, багатозначних атрибутів, рекурсивних зв'язків, зв'язків з атрибутами) та застосовані перетворення з метою усунення таких зв'язків [1], [7].

Отримана логічна структура БД “Сесія” наведена на рис. 2.



5.2. Нормалізація таблиць логічної моделі даних

Створений на під етапі 5.1. набір відношень логічної моделі БД “Сесія” був перевірений на коректність об’єднання атрибутів у кожному відношенні.

Перевірка виконувалася шляхом застосування до кожного відношення процедури послідовної нормалізації – поетапного процесу заміни сукупності відношень іншою сукупністю (схемою), в якій відношення мають просту і регулярну структуру.

До бази даних “Сесія” були послідовно застосовані процедури приведення в першу, другу та третю нормальні форми (1НФ, 2НФ, 3НФ).

Перша нормальна форма вимагає, щоб кожне поле таблиці бази даних було неділимим і не містило повторень.

Друга нормальна форма вимагає, щоб всі атрибути таблиці залежали від первинного ключа.

Третя нормальна форма вимагає, щоб в таблиці не малося транзитивних залежностей між атрибутами таблиці.

Результатом нормалізації стала остаточна версія логічної моделі БД “Сесія”, що не має протиріч і має мінімальну збитковість.

Результуюча логічна схема бази даних наведена в таблицях 12 – 21.

Таблиця 12 - additional_points

Первинний ключ	Атрибути
id	merit
	points

Таблиця 13 - control_type

Первинний ключ	Атрибути
id	name

Таблиця 14 - group

Первинний ключ	Атрибути
id	name
	first_year

Таблиця 15 - faculty

Первинний ключ	Атрибути
id	full_name
	short_name

Таблиця 16 - group

Первинний ключ	Атрибути
id	year_grade
	exam_grade

Таблиця 17 - cathedra

Первинний ключ	Атрибути
id	full_name
	short_name

Таблиця 18 - specialty

Первинний ключ	Атрибути
id	name
	N

Таблиця 19 - student

Первинний ключ	Атрибути
id	last_name
	first_name
	second_name
	record_book_N

Таблиця 20 - professor

Первинний ключ	Атрибути
id	last_name
	first_name
	second_name
	position
	academic_rank
	academic_degree

Таблиця 21 - discipline

Первинний ключ	Атрибути
id	name

	hours
--	-------

Таблиця 22 - control_registy

Первинний ключ	Атрибути
id	date

6. Реалізація бази даних

Для реалізації бази даних “Сесія” була обрана вільна система керування реляційними базами даних MySQL.

Обґрунтування вибору СКБД для вирішення поставлених завдань наведено у вступі.

Розроблена логічна схема БД “Сесія” була реалізована у середовищі MySQL Workbench, який є інструментом для візуального проектування баз даних, що інтегрує проектування, моделювання, створення й експлуатацію БД в єдине безкоштовне оточення для системи баз даних MySQL.

З метою забезпечення цілісності даних були застосовані методи:

1) забезпечення структурної цілісності (біли виконані такі умови: наявність тільки однорідних структур даних типу "реляційне відношення", відсутність дублікатів кортежів, обов'язкова наявність у кожному відношенні первинного ключа);

2) обмеження реальних значень даних (обмеження довжини строк, що вводяться, обмеження року початку навчання: в таблиці «Група» ≥ 2017 тощо);

3) забезпечення посилкової цілісності (зміни в таблицях виконуються синхронно, а зміст двох пов'язаних таблиць відповідати таким правилам: кожному запису основної таблиці відповідає нуль або більше записів підлеглої таблиці; в підлеглій таблиці немає записів, які не мають батьківських записів в основній таблиці; кожний запис підлеглої таблиці має тільки один батьківський запис основної таблиці.

Умови цілісності даних визначають, які дані можуть бути записані в БД у результаті додавання або оновлення даних. При маніпулюванні даними в

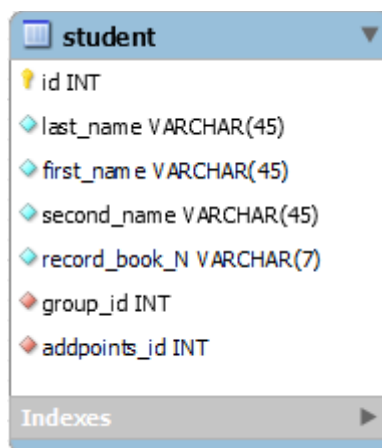
таблицях виконується контроль дій за правилом “CASCADE” відповідно до табл. 22.

Таблиця 22

Операція	Пояснення
Вилучення (DELETE)	При вилученні рядка з батьківської таблиці в підлеглий таблиці всі рядки-нащадки автоматично вилучаються
Оновлення (UPDATE)	При зміні первинного ключа в рядку батьківської таблиці в підлеглий таблиці відповідні значення зовнішнього ключа також автоматично змінюються у всіх рядках-нащадках для того, щоби відповідати новому значенню первинного ключа

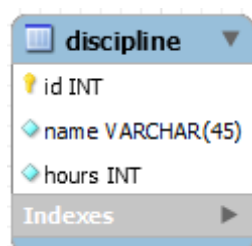
Результат реалізації БД “Сесія” наведений в таблицях 23-32 та в скриптах.

Таблиця 23



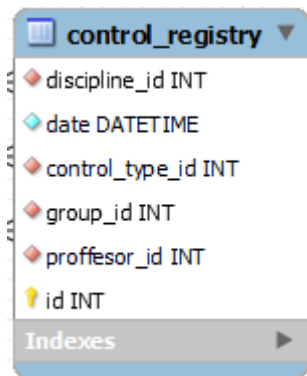
Сутність «Студент» має свої вище описані атрибути ПІБ студента, номер залікової книжки, а також зовнішній ключ номеру групи та зовнішній ключ додаткових балів, на які студент може розраховувати.

Таблиця 24



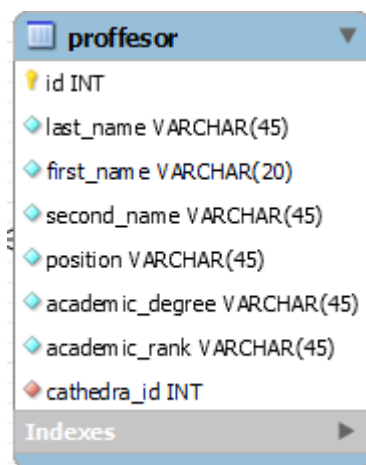
Сутність «Дисципліна» має атрибути імені предмету та кількості годин викладання.

Таблиця 25



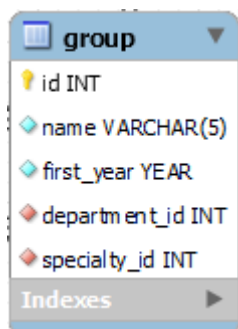
Сутність «Реєстр відомостей семестрового контролю» має власний атрибут дати проведення контролю, а також зовнішні ключі дисципліни, типу контролю, номеру групи, викладача.

Таблиця 26



Сутність «Екзаменатор», окрім власних атрибутів ПІБ, посади, наукового ступеню та вченого звання має також зовнішній ключ кафедри викладача.

Таблиця 27



Сутність «Група» має свої атрибути першого року навчання, назви групи, а також зовнішні ключі кафедри та спеціальності.

Таблиця 28

faculty	
id	INT
full_name	VARCHAR(100)
short_name	VARCHAR(10)
Indexes	

Сутність «Факультет» має лише власні атрибути повного імені та короткої аббревіатури.

Таблиця 29

specialty	
id	INT
name	VARCHAR(50)
N	INT
Indexes	

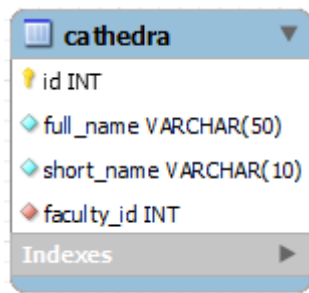
Сутність «Спеціальність» має лише власні атрибути повного імені та номеру спеціальності.

Таблиця 30

semestr_control	
id	INT
student_id	INT
control_registry_id	INT
year_grade	INT
exam_grade	INT
Indexes	

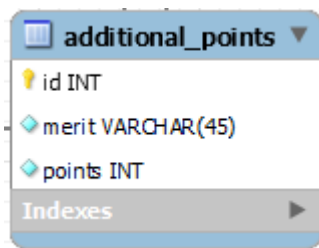
Сутність «Відомість семестрового контролю» має атрибути балів, виставлених за контроль та набутих за семестр, а також зовнішні ключі студента й відомості у реєстрі.

Таблиця 31



Сутність «Кафедра» має атрибути повного імені й аббревіатури, а також зовнішній ключ факультету.

Таблиця 32



Сутність «Додаткові бали» описує заслугу студента та кількість балів у винагороду.

Дані таблиці 23-32 з усіма атрибутами та зовнішніми ключами створювалися за скриптом 1:

Скрипт 1

```
CREATE DATABASE IF NOT EXISTS coursework;
USE coursework;
```

```
-----
-- ----- coursework.faculty-----
-----
```

```
CREATE TABLE IF NOT EXISTS coursework.faculty (
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    full_name VARCHAR(100) NOT NULL UNIQUE,
    short_name VARCHAR(10) NOT NULL UNIQUE);
```

```

-----
-- ----- coursework.cathedra-----
-- -----

```

```

CREATE TABLE IF NOT EXISTS coursework.cathedra (
  id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
  full_name VARCHAR(50) NOT NULL UNIQUE,
  short_name VARCHAR(10) NOT NULL UNIQUE,
  faculty_id INT NOT NULL,
  CONSTRAINT fk_cathedra_faculty FOREIGN KEY (faculty_id)
    REFERENCES coursework.faculty (id)
    ON DELETE CASCADE
    ON UPDATE CASCADE);

```

```

-----
-- ----- coursework.group-----
-- -----

```

```

CREATE TABLE IF NOT EXISTS coursework.group (
  id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
  name VARCHAR(5) NOT NULL,
  first_year YEAR NOT NULL CHECK (first_year >= 2017),
  department_id INT NOT NULL,
  specialty_id INT NOT NULL,
  CONSTRAINT fk_group_department FOREIGN KEY (department_id)
    REFERENCES coursework.cathedra (id)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT fk_group_specialty FOREIGN KEY (specialty_id)
    REFERENCES coursework.specialty (id)
    ON DELETE CASCADE
    ON UPDATE CASCADE);

```

```

-----
-- ----- coursework.proffesor-----

```

```

-----
CREATE TABLE IF NOT EXISTS coursework.proffesor (
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    last_name VARCHAR(45) NOT NULL,
    first_name VARCHAR(20) NOT NULL,
    second_name VARCHAR(45) NOT NULL,
    position VARCHAR(45) NOT NULL,
    academic_degree VARCHAR(45) NOT NULL,
    academic_rank VARCHAR(45) NOT NULL,
    department_id INT NOT NULL,
    CONSTRAINT fk_proffesor_cathedra FOREIGN KEY (cathedra_id)
        REFERENCES coursework.cathedra (id)
        ON DELETE CASCADE
        ON UPDATE CASCADE);

```

```

-----
-- ----- coursework.discipline-----
-----

```

```

CREATE TABLE IF NOT EXISTS coursework.discipline (
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    name VARCHAR(45) NOT NULL UNIQUE,
    hours INT NOT NULL CHECK (hours >= 30 AND hours <=150));

```

```

-----
-- ----- coursework.student-----
-----

```

```

CREATE TABLE IF NOT EXISTS coursework.student (
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    last_name VARCHAR(45) NOT NULL,
    first_name VARCHAR(45) NOT NULL,
    second_name VARCHAR(45) NOT NULL,
    record_book_N VARCHAR(7) NOT NULL UNIQUE,

```

```

group_id INT NOT NULL,
addpoints_id INT NOT NULL,
CONSTRAINT fk_student_group FOREIGN KEY (group_id)
    REFERENCES coursework.group (id)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
CONSTRAINT fk_student_addpoints_id
    FOREIGN KEY (addpoints_id)
    REFERENCES additional_points (id)
    ON DELETE CASCADE
    ON UPDATE CASCADE);

```

```

-- -----
-- ----- coursework.control_type-----
-- -----

```

```

CREATE TABLE IF NOT EXISTS coursework.control_type (
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL UNIQUE);

```

```

-- -----
-- ----- coursework.specialty-----
-- -----

```

```

CREATE TABLE IF NOT EXISTS coursework.specialty (
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    `name` VARCHAR(50) NOT NULL UNIQUE,
    N INT NOT NULL UNIQUE);

```

```

-- -----
-- ----- coursework.control_registry-----
-- -----

```

```

CREATE TABLE IF NOT EXISTS coursework.control_registry (
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,

```

```

discipline_id INT NOT NULL,
`date` DATETIME NOT NULL,
control_type_id INT NOT NULL,
group_id INT NOT NULL,
proffesor_id INT NOT NULL,
CONSTRAINT      fk_control_registry_control_type      FOREIGN      KEY
(control_type_id)
REFERENCES coursework.control_type (id)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT      fk_control_registry_discipline      FOREIGN      KEY
(discipline_id)
REFERENCES coursework.discipline (id)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT fk_control_registry_group FOREIGN KEY (group_id)
REFERENCES coursework.group (id)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT fk_control_registry_proffesor
FOREIGN KEY (proffesor_id)
REFERENCES coursework.proffesor (id)
ON DELETE CASCADE
ON UPDATE CASCADE);

```

```

-----
-- ----- coursework.semestr_control-----
-- -----

```

```

CREATE TABLE IF NOT EXISTS coursework.semestr_control (
  id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
  student_id INT NOT NULL,
  control_registry_id INT NOT NULL,
  year_grade INT NOT NULL DEFAULT 0,
  exam_grade INT NOT NULL DEFAULT 0,

```

```

CONSTRAINT fk_semestr_control_student FOREIGN KEY (student_id)
REFERENCES coursework.student (id)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT fk_semestrcontrol_controlregistry FOREIGN KEY
(control_registry_id)
REFERENCES coursework.control_registry (id)
ON DELETE CASCADE
ON UPDATE CASCADE);

```

```

-- -----
-- ----- coursework.additional_points-----
-- -----

```

```

CREATE TABLE IF NOT EXISTS coursework.additional_points (
id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
merit VARCHAR(45) NOT NULL,
points INT NOT NULL CHECK (points >= 0));

```

Дані заповнювалися як методом INSERT INTO, так і засобами СУБД, імпортуванням даних через файл.

Скрипт 2

```

-- -----
-- ----- coursework.additional_points-----
-- -----

```

```

INSERT INTO coursework.additional_points (id, merit, points)
VALUES ("Неактивний студент", 0),
("Міжнародні олімпіади", 40),
("Всеукраїнські олімпіади", 20),
("Громадське життя університету", 15),
("Спортивні змагання", 10);

```

```

SELECT * FROM additional_points;

```

	id	merit	points
▶	1	Неактивний студент	0
	2	Міжнародні олімпіади	40
	3	Всеукраїнські олімпіади	20
	4	Громадське життя університету	15
	5	Спортивні змагання	10

 -- coursework.control_type-----

```
INSERT INTO coursework.control_type (`name`)
VALUES ("Курсова"), ("Екзамен"), ("Залік");
SELECT * FROM control_type ORDER BY id;
```

	id	name
▶	1	Курсова
	2	Екзамен
	3	Залік

 -- coursework.specialty-----

```
INSERT INTO coursework.specialty (name, N)
VALUES ("Фізика та астрономія", 104),
("Інженерія програмного забезпечення", 121),
("Комп'ютерні науки", 122),
("Комп'ютерна інженерія", 123),
("Системний аналіз", 124),
("Кібербезпека", 125),
("Інформаційні системи та технології", 126),
("Прикладна механіка", 131);
```

```
SELECT * FROM specialty;
```

	id	name	N
▶	1	Фізика та астрономія	104
	2	Інженерія програмного забезпечення	121
	3	Комп'ютерні науки	122
	4	Комп'ютерна інженерія	123
	5	Системний аналіз	124
	6	Кібербезпека	125
	7	Інформаційні системи та технології	126
	8	Прикладна механіка	131


```

-----
-- ----- coursework.faculty-----
-----

```

```

INSERT INTO coursework.faculty (full_name, short_name)
VALUES ("Фізико-математичний факультет", "ФМФ"),
("Інститут прикладного системного аналізу", "ІПСА"),
("Факультет інформатики та обчислювальної техніки", "ФІОТ"),
("Фізико-технічний інститут", "ФТІ"),
("Інститут спеціального зв'язку та захисту інформації", "ІСЗЗІ"),
("Механіко-машинобудівний інститут", "ММІ");

```

```

SELECT * FROM faculty;

```

	id	full_name	short_name
►	1	Фізико-математичний факультет	ФМФ
	2	Інститут прикладного системного аналізу	ІПСА
	3	Факультет інформатики та обчислювальної ...	ФІОТ
	4	Фізико-технічний інститут	ФТІ
	5	Інститут спеціального зв'язку та захисту інф...	ІСЗЗІ
	6	Механіко-машинобудівний інститут	ММІ

```

-----
-- ----- coursework.cathedra-----
-----

```

```

INSERT INTO coursework.cathedra (full_name, short_name, faculty_id)
VALUES ("Кафедра інформатики та програмної інженерії", "ІПІ", 3),
("Кафедра інформаційних систем та технологій", "ІСТ", 3),
("Кафедра обчислювальної техніки", "ОТ", 3),
("Кафедра математичних методів системного аналізу", "ММСА", 2),
("Кафедра системного проектування", "СП", 2),
("Кафедра загальної фізики", "КЗФ", 1),
("Спеціальна кафедра №1", "СП №1", 5),
("Кафедра конструювання машин", "КМ", 6),
("Кафедра прикладної фізики", "ПФ", 4),
("Кафедра інформаційної безпеки", "ІБ", 4);

```

```

SELECT * FROM cathedra;

```

	id	full_name	short_name	faculty_id
▶	1	Кафедра інформатики та програмної інженерії	ІПІ	3
	2	Кафедра інформаційних систем та технологій	ІСТ	3
	3	Кафедра обчислювальної техніки	ОТ	3
	4	Кафедра математичних методів системного ...	ММСА	2
	5	Кафедра системного проектування	СП	2
	6	Кафедра загальної фізики	КЗФ	1
	7	Спеціальна кафедра №1	СП №1	5
	8	Кафедра конструювання машин	КМ	6
	9	Кафедра прикладної фізики	ПФ	4
	10	Кафедра інформаційної безпеки	ІБ	4

```

-----
-- ----- coursework.group-----
-----

```

```

INSERT INTO coursework.group (name, first_year, department_id,
specialty_id)

```

```

VALUES ("IT-01", 2020, 1, 2),
("IT-91", 2019, 1, 2),
("IT-03", 2020, 1, 2),
("IC-84", 2018, 1, 4),
("IP-01", 2020, 2, 2),
("IP-95", 2019, 3, 2),
("IA-02", 2020, 1, 7),
("KP-73", 2017, 5, 5),
("KP-15", 2021, 3, 3),
("CP-11", 2021, 6, 1),
("IB-92", 2019, 10, 6);

```

```

SELECT * FROM coursework.group;

```

	id	name	first_year	department_id	specialty_id
	3	IT-03	2020	1	2
	4	IC-84	2018	1	4
	5	IP-01	2020	2	2
	6	IP-95	2019	3	2
	7	IA-02	2020	1	7
	8	KP-73	2017	5	5
	9	KP-15	2021	3	3
	10	CP-11	2021	6	1
	11	IB-92	2019	10	6

```

-----

```

```
-- ----- coursework.proffesor-----
-- -----
```

```
LOAD DATA INFILE 'C:/Temp/proffesors.csv'
INTO TABLE coursework.proffesor
CHARACTER SET utf8mb4
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;
```

```
SELECT * FROM proffesor;
```

	id	last_name	first_name	second_name	position	academic_degree	academic_rank	cathedra_id
►	1	Тимко	Ірина	Павлівна	доцент	к.т.н	доцент	1
	2	Гаврильченко	Олена	Андріївна	старший викладач	-	-	2
	3	Брусов	Анатолій	Іванович	завідувач кафедри	д.т.н	професор	3
	4	Смовж	Валерій	Станіславович	професор	д.т.н	професор	2
	5	Шпонько	Геннадій	Дмитрович	старший викладач	к.т.н	-	1

Для забезпечення багатокористувацького режиму було створено дві ролі: «Викладач» та «Менеджер».

Ролі «Викладач» наданий доступ до перегляду інформації по всіх таблицях, але обмежений доступ до вставки. Викладач може заповнювати даними лише таблицю семестрового контролю.

Скрипт 3

```
CREATE USER 'teacher'@'localhost' IDENTIFIED BY '111';
GRANT SELECT ON *.* TO 'teacher'@'localhost';
GRANT INSERT, UPDATE ON coursework.semestr_control TO
'teacher'@'localhost';
SELECT * FROM coursework.group;
```

	id	name	first_year	department_id	specialty_id
►	1	ІТ-01	2020	1	2
	2	ІТ-91	2019	1	2
	3	ІТ-03	2020	1	2
	4	ІС-84	2018	1	4
	5	ІП-01	2020	2	2
	6	ІП-95	2019	3	2
	7	ІА-02	2020	1	7
	8	КР-73	2017	5	5
	9	КП-15	2021	3	3
	10	СР-11	2021	6	1
	11	ІБ-92	2019	10	6

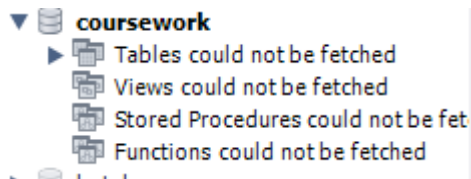
```
INSERT INTO coursework.group (name, first_year, department_id,
specialty_id) VALUES ('АП-11', '2019', '9', '2');
```

```
Operation failed: There was an error while applying the SQL script to the database.
Executing:
INSERT INTO `coursework`.`group` (`name`, `first_year`, `department_id`, `specialty_id`) VALUES
('АП-11', 2019, '9', '2');

ERROR 1142: 1142: INSERT command denied to user 'teacher'@'localhost' for table 'group'
SQL Statement:
INSERT INTO `coursework`.`group` (`name`, `first_year`, `department_id`, `specialty_id`) VALUES
('АП-11', 2019, '9', '2')
```

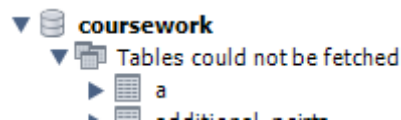
«Менеджер» може тільки створювати таблиці, проте не може переглядати чи маніпулювати даними у них.

```
CREATE USER 'manager'@'localhost' IDENTIFIED BY '222';
GRANT CREATE ON *.* TO 'manager'@'localhost';
```



```
USE coursework;
```

```
CREATE TABLE IF NOT EXISTS a(
    id INT,
    d INT);
```



Розробка БД потребує виведення у зручній формі даних за наступними запитам користувача:

- пошук заданого студента по прізвищу (з уточненням за необхідності по іншим атрибутам);

```
SELECT s.last_name AS "Прізвище", s.first_name AS "Ім'я", g.name AS
"Група", sp.name AS "Спеціальність"
FROM student AS s
LEFT JOIN coursework.group AS g ON
```

```
s.group_id = g.id
LEFT JOIN specialty as sp ON
g.specialty_id = sp.id WHERE s.last_name = "Білий";
```

	Прізвище	Ім'я	Група	Спеціальність
▶	Білий	Сергій	ІП-01	Інженерія програмного забезпечення

- пошук заданого студента по номеру залікової книжки;

```
SELECT s.last_name AS "Прізвище", s.first_name AS "Ім'я",
s.record_book_N AS "№ заліковки", sp.N AS "Спеціальність"
FROM student AS s
LEFT JOIN coursework.group AS g ON
s.group_id = g.id
LEFT JOIN specialty as sp ON
g.specialty_id = sp.id WHERE s.record_book_N = "КЗ-0101";
```

	Прізвище	Ім'я	№ заліковки	Спеціальність
▶	Кофер	Кирило	КЗ-0101	121

- вибірка всіх даних про результати оцінювання заданого студента за поточний навчальний семестр;

```
SELECT st.last_name AS "Прізвище", st.first_name AS "Ім'я", d.name
AS "Дисципліна", ct.name AS "Тип контролю", sc.year_grade + sc.exam_grade
+ ap.points AS "Балл"
FROM student AS st
LEFT JOIN additional_points as ap ON
st.addpoints_id = ap.id
RIGHT JOIN semestr_control AS sc ON
st.id = sc.student_id
LEFT JOIN control_registry AS cr ON
sc.control_registry_id = cr.id
LEFT JOIN discipline AS d ON
cr.discipline_id = d.id
LEFT JOIN control_type AS ct ON
cr.control_type_id = ct.id WHERE last_name = "Сірко";
```

	Прізвище	Ім'я	Дисципліна	Тип контролю	Балл
▶	Сірко	Макар	Алгоритми	Залік	70
	Сірко	Макар	Англійська	Залік	74

- вибірка даних по студентах, що мають академічну заборгованість: лише список студентів;

```
SELECT DISTINCT st.last_name AS "Прізвище", st.first_name AS "Ім'я"
FROM student AS st
LEFT JOIN additional_points as ap ON
st.addpoints_id = ap.id
RIGHT JOIN semestr_control AS sc ON
st.id = sc.student_id
WHERE sc.year_grade + sc.exam_grade + ap.points < 60;
```

	Прізвище	Ім'я
▶	Марк	Кирило
	Гринь	Олег
	Бойко	Кирило
	Варно	Сергій
	Кіт	Олег

- вибірка даних по студентах, що мають академічну заборгованість: відсортований список студентів разом з їхніми балами та предметами;

```
SELECT st.last_name AS "Прізвище", st.first_name AS "Ім'я", d.name
AS "Дисципліна", sc.year_grade + sc.exam_grade + ap.points AS "Балл"
FROM student AS st
LEFT JOIN additional_points as ap ON
st.addpoints_id = ap.id
RIGHT JOIN semestr_control AS sc ON
st.id = sc.student_id
LEFT JOIN control_registry AS cr ON
sc.control_registry_id = cr.id
LEFT JOIN discipline AS d ON
cr.discipline_id = d.id WHERE sc.year_grade + sc.exam_grade +
ap.points < 60 ORDER BY last_name;
```

	Прізвище	Ім'я	Дисципліна	Балл
►	Бойко	Кирило	Бази даних	44
	Бойко	Кирило	Алгоритми	59
	Бойко	Кирило	Англійська	0
	Варно	Сергій	Бази даних	0
	Варно	Сергій	Алгоритми	33
	Гринь	Олег	Бази даних	35
	Гринь	Олег	Алгоритми	22
	Кіт	Олег	Бази даних	34
	Марк	Кирило	Бази даних	51

Була розроблена процедура, аби вираховувати курс студента та виводити повну інформацію про нього:

Скрипт

```

now() - TIMESTAMP

DELIMITER //

CREATE PROCEDURE `p2` ()
LANGUAGE SQL
DETERMINISTIC
SQL SECURITY DEFINER
COMMENT 'A procedure'
BEGIN
    DECLARE today TIMESTAMP DEFAULT CURRENT_DATE;
    SELECT st.first_name AS "Ім'я", st.last_name AS "Прізвище",
gr.name AS "Група", gr.first_year AS "Рік вступу", sp.name AS
"Спеціальність", YEAR(today)-gr.first_year+1 as "Курс"
    FROM student as st JOIN coursework.group as gr
    ON st.group_id = gr.id JOIN specialty as sp
    ON gr.specialty_id = sp.id;
END //
```

```
CALL p2();
```

	Ім'я	Прізвище	Група	Рік вступу	Спеціальність	Курс
►	Олег	Мак	ІТ-01	2020	Інженерія програмного забезпечення	2
	Макар	Сірко	ІТ-01	2020	Інженерія програмного забезпечення	2
	Ольга	Гаврюшка	ІТ-01	2020	Інженерія програмного забезпечення	2
	Кирило	Кофер	ІТ-01	2020	Інженерія програмного забезпечення	2
	Сергій	Варно	ІТ-01	2020	Інженерія програмного забезпечення	2
	Павло	Параска	ІТ-91	2019	Інженерія програмного забезпечення	3
	Сергій	Ткач	ІТ-91	2019	Інженерія програмного забезпечення	3
	Олег	Кіт	ІТ-91	2019	Інженерія програмного забезпечення	3
	Тетяна	Сергієнко	ІТ-03	2020	Інженерія програмного забезпечення	2
	Павло	Добрик	ІТ-03	2020	Інженерія програмного забезпечення	2
	Олег	Михнюк	ІС-84	2018	Комп'ютерна інженерія	4
	Кирило	Сирчук	ІС-84	2018	Комп'ютерна інженерія	4
	Павло	Корло	ІП-01	2020	Інженерія програмного забезпечення	2
	Петро	Косар	ІП-01	2020	Інженерія програмного забезпечення	2
	Сергій	Білий	ІП-01	2020	Інженерія програмного забезпечення	2
	Олесь	Нікулін	ІП-95	2019	Інженерія програмного забезпечення	3
	Сергій	Корф	ІА-02	2020	Інформаційні системи та технології	2
	Кирило	Марк	КР-73	2017	Системний аналіз	5
	Олег	Гринь	СР-11	2021	Фізика та астрономія	1
	Кирило	Бойко	ІБ-92	2019	Кібербезпека	3

Була створена процедура, що визначає «букву» студента з того чи іншого предмету.

```
DELIMITER //
CREATE PROCEDURE res_exam(i INT)
    COMMENT 'Повертає результати здавання екзамена групою по студентах'
    begin
        DROP VIEW IF EXISTS report_exam;
        CREATE VIEW report_exam AS
        SELECT      st.first_name,      st.last_name,      st.record_book_N,
s_ctrl1.exam_grade, gr.name, ds.name, c_ref.id, c_reg.date,
        CASE
            WHEN      s_ctrl1.exam_grade+s_ctrl1.year_grade      <      60      THEN
"Незадовільно"
            WHEN      s_ctrl1.exam_grade+s_ctrl1.year_grade      >=      60      AND
s_ctrl1.exam_grade+s_ctrl1.year_grade < 65 THEN "Достатньо"
```



```

        WHEN      s_ctrl.exam_grade+s_ctrl.year_grade      >=      65      AND
s_ctrl.exam_grade+s_ctrl.year_grade < 75 THEN "Задовільно"
        WHEN      s_ctrl.exam_grade+s_ctrl.year_grade      >=      75      AND
s_ctrl.exam_grade+s_ctrl.year_grade < 85 THEN "Добре"
        WHEN      s_ctrl.exam_grade+s_ctrl.year_grade      >=      85      AND
s_ctrl.exam_grade+s_ctrl.year_grade < 65 THEN "Дуже добре"
        WHEN s_ctrl.exam_grade+s_ctrl.year_grade >= 95 THEN "Відмінно"
    END AS "Універ.оцін"

    FROM semestr_control as s_ctrl, control_registry as c_reg, group
as gr, student as st, discipline as ds

    WHERE          s_ctrl.control_registry_id=c_reg.id          AND
c_reg.discipline_id=ds.id AND c_reg.group_id = gr.id AND c_reg.id = i;

    END //

```

Був створений тригер, що додає дані до таблиці log при вставці та оновленні таблиці студента.

```

CREATE TABLE `log` (
  `id` INT( 11 ) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `msg` VARCHAR( 255 ) NOT NULL,
  `time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `row_id` INT( 11 ) NOT NULL
)

CREATE TRIGGER `update_test`
AFTER INSERT, UPDATE ON student
FOR EACH ROW BEGIN
    INSERT INTO log Set msg = 'insert', row_id = NEW.id;
END

```

ВИСНОВКИ

Метою курсової роботи було проектування бази даних для підтримки екзаменаційної та залікової сесії.

Під час виконання курсової роботи були проведені всі необхідні дослідження, що стосуються розробки стратегії автоматизації та оптимізації робіт з планування, обліку, оцінювання якості навчального процесу, статистичного аналізу та опрацювання зібраних даних у закладах вищої освіти.

В ході роботи був виконаний повний цикл розробки бази даних, а саме:

проведений аналіз предметного середовища та сформульовані бізнес-правила як основа завдання обмежень при проектуванні та реалізації бази даних;

визначені необхідний набір сутностей, атрибутів для кожної сутності та зв'язки між об'єктами;

ER-модель описана в нотації Чена;

розроблена модель користувачів бази даних з описом їх прав;

побудована реляційна схема з ER-моделі

створена база даних у форматі обраної системи керування базою даних MySQL;

відповідно до розробленої багатокористувацької моделі створені користувачі бази даних;

з використанням засобів СУБД імпортовані дані в створену базу даних;

розроблені SQL-запити для визначених на етапі аналізу предметного середовища потреб користувачів;

Завдання, поставлені на курсову роботу, виконані, мета досягнута.

Створена база даних є актуальним і корисним продуктом та може бути застосована для підтримки екзаменаційної та залікової сесії.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Методичні вказівки до виконання курсової роботи з дисципліни “Бази даних” першого (бакалаврського) рівня вищої освіти [Електронний ресурс]: метод. вказівки для студ. освітньої програми “Інженерія програмного забезпечення інформаційних систем” спеціальності 121 “Інженерія програмного забезпечення” / КПІ ім. Ігоря Сікорського; уклад.: К.І. Ліщук, О.І. Марченко. – Електронні текстові дані (1 файл: 1.8 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 15 с.
2. The 5 best reasons to choose mysql – and its 5 biggest challenges [Електронний ресурс] – 2021. – Режим доступу до ресурсу: URL <https://dataconomy.com/2017/04/5-reasons-challenges-mysql/>
3. Top 10 Databases to Use in 2021 [Електронний ресурс] – 2021. – Режим доступу до ресурсу: URL <https://towardsdatascience.com/top-10-databases-to-use-in-2021-d7e6a85402ba>
4. G2 Crowd. MySQL Reviews & Product Details [Електронний ресурс] – 2021. – Режим доступу до ресурсу: URL <https://www.g2.com/products/mysql/reviews/mysql-review-1775714>
5. Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського” [Електронний ресурс] – 2021. – Режим доступу до ресурсу: URL https://kpi.ua/kpi_faculty
6. Положення про поточний, календарний та семестровий контроль результатів навчання в КПІ ім. Ігоря Сікорського, затверджене наказом ректора КПІ ім. Ігоря Сікорського № 7-137 від 05.08.2020. ” [Електронний ресурс] – 2021. – Режим доступу до ресурсу: URL <https://kpi.ua/grading>
7. Г.А.Гайна. Основи проектування баз даних: Навчальний посібник. – К.: КНУБА, 2005. – 204 с.
8. Текст лекцій до розділу “Проектування баз даних” дисципліни “Бази даних” для студ. спец. „Автоматизоване управління технологічними процесами” / Уклад.: Л.Д. Ярощук. – К.НТУУ «КПІ», 2012. - 117 с.
9. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання. Національний стандарт України – Київ: ДП «УкрНДНЦ», 2016. – 26 с.
10. ДСТУ ГОСТ 7.1:2006. Бібліографічний запис, бібліографічний опис. Загальні вимоги та правила складання : метод. рекомендації з впровадження / уклали: Галевич О. К., Штогрин І. М.