

Міністерство освіти і науки України  
Український державний університет науки і технологій

Факультет «Комп'ютерні технології та системи»

(назва факультету)

Кафедра «Електронних обчислювальних машин»

(повна назва кафедри)

Пояснювальна записка  
до кваліфікаційної роботи

перший (Бакалавр)

(ступінь вищої освіти)

*Роман Чайка*  
21.06.2022

на тему: Проектування бази даних навчально-методичних видань кафедри ЕОМ за методом нормальних форм та її обробка.

за освітньою програмою Комп'ютерна інженерія

зі спеціальності: 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

Виконав: студент групи: КС19120

*[Підпис студента]*

(підпис студента)

Роман ЧАЙКА

(Ім'я ПРІЗВИЩЕ)

Керівник:

*[Підпис керівника]*

(підпис)

Доцент, Вікторія ПАХОМОВА

(посада, Ім'я ПРІЗВИЩЕ)

Нормоконтролер:

*[Підпис нормоконтролера]*

(підпис)

Старший викладач, Володимир ДЗЮБА

(посада, Ім'я ПРІЗВИЩЕ)

Консультанти:

(назва розділу)

(підпис)

/

(посада, Ім'я ПРІЗВИЩЕ)

/

(назва розділу)

(підпис)

/

(посада, Ім'я ПРІЗВИЩЕ)

/

(назва розділу)

(підпис)

/

(посада, Ім'я ПРІЗВИЩЕ)

/

(назва розділу)

(підпис)

/

(посада, Ім'я ПРІЗВИЩЕ)

/

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

Студент

*[Підпис студента]*

(підпис)

Дніпро – 2022 рік

**Ministry of Education and Science of Ukraine**  
**Ukrainian State University of Science and Technologies**  
**Computer Technology and Systems**  
(faculty)

**Electronic Computers Machinery**  
(department)

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

on the topic: Designing a database of educational and methodical publications of the ECM department by the method of normal forms and its processing.  
according to educational curriculum Computer Engineering  
in the Speciality: 123 Computer Engineering  
Done by the student of the group: KC19120



Roma Chaika  
(name, surname)

Scientific Supervisor:



Docent, Victoria Pakhomova  
(position, name, surname)

Normative controller :



St. Lectures, Volodymyr Dziuba  
(position, name, surname)

Supervisors

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

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

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

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

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

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

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

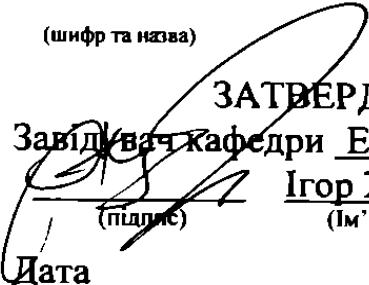
/ \_\_\_\_\_ /  
(position, name, surname)

Dnipro – 2022

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

Факультет: Комп'ютерних технологій і систем  
Кафедра: Електронних обчислювальних машин  
Рівень вищої освіти: перший (бакалаврський)  
Освітня програма: Комп'ютерна інженерія  
Спеціальність: 123 «Комп'ютерна інженерія»

(шифр та назва)

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

**З А В Д А Н Н Я**

на кваліфікаційну роботу перший (бакалаврський)  
(ступінь вищої освіти)

студенту Чайці Роману Руслановичу  
(Прізвище, Ім'я, По батькові)

1. Тема роботи: Проектування бази даних навчально-методичних видань  
кафедри ЕОМ за методом нормальних форм та її обробка

Керівник роботи: Пахомова Вікторія Миколаївна  
(Прізвище, Ім'я, По батькові, науковий ступінь, вчене звання)

затверджені наказом від "07" 12. 2021 р. № 68ст

2. Строк подання студентом роботи: 13.06.2022 р.

3. Вихідні дані до роботи: дані про видавництво навчально-методичних  
постанов, що розташовані в репозиторії УДУНТ; інформація про  
професорсько-викладацький склад кафедри ЕОМ.

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

4.1 Аналітична частина: огляд і аналіз предмета проектування.

4.2 Основна частина: проектування реляційної бази даних за методом  
нормальних форм; створення реляційної бази даних та її обробка з  
використанням «MySQL»; організація самостійної роботи з дисципліни «Бази  
даних» (використання DML; організація різних підходів до зв'язування  
таблиць; об'єднання запитів).

4.3 Охорона праці та захист навколишнього середовища:

4.4 Економічна частина:

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових  
креслень):

## 6. Консультанти розділів роботи:

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

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вступ	01.06.2022 – 06.06.2022.	5%
2	Огляд і аналіз предмету проекту	11.04.2022 – 02.05.2022	20%
3	Постановка задачі		10%
4	Проектування реляційної бази даних за методом нормальних форм	02.05.2022 – 08.05.2022	20%
5	Створення та обробка реляційної бази даних	08.05.2022 – 20.05.2022	20%
6	Організація самостійної роботи з дисципліни «Бази даних»	20.05.2022 – 01.06.2022	20%
7	Висновки та рекомендації	01.06.2022 – 13.06.2022.	5%
7	Подання кваліфікаційної роботи до кафедри	13.06.2022	
8	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	22.06.2022	

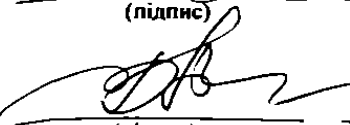
Студент

  
(підпис)

Роман Чайка

(Ім'я ПРІЗВИЩЕ)

Керівник роботи

  
(підпис)

Вікторія Пахомова

(Ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра: 50с., 40 рис., 3 додатки, 16 джерел.

Об'єкт розробки – база даних навчально-методичних видань кафедри «Електронних обчислювальних машин».

Мета кваліфікаційної роботи – проєктування бази даних навчально – методичних видань кафедри «Електронних обчислювальних машин» за методом нормальних форм та її обробка.

Методи дослідження – проєктування бази даних за методом нормальних форм, створення бази даних з використанням MySQL Workbench та зрівняння можливостей MySQL Workbench з SQL Microsoft Access.

Створено в MySQL Workbench спроектовану за методом нормальних форм, базу навчально-методичних видань кафедри «Електронних обчислювальних машин» на основі даних репозиторію УДУНТ та даних викладацького складу кафедри «Електронних обчислювальних машин».

Проведено зрівняння програм MySQL Workbench з Microsoft Access за наступними ознаками: загальні характеристики; зрівняння інтерфейсу; вартість. Виконано зрівняння можливостей MySQL Workbench з SQL Microsoft Access за напрямками досліджень: використання DML; організація різних підходів до зв'язування таблиць; об'єднання запитів.

Рекомендовано створену базу даних навчально-методичних видань до використання адміністрацією кафедри «Електронних обчислювальних машин».

Переведення лабораторних робіт з SQL Microsoft Access на MySQL Workbench не доцільно, але MySQL Workbench може бути рекомендований до використання здобувачів першого ступеня, спеціальності «Комп'ютерна інженерія» при виконанні самостійної роботи з дисципліни «Бази даних».

Ключові слова: БАЗА ДАНИХ, НАВЧАЛЬНО-МЕТОДИЧНІ ВИДАННЯ, НОРМАЛЬНІ ФОРМИ, MYSQL, WORKBENCH, DML, ЗВ'ЯЗУВАННЯ ТАБЛИЦЬ, ОБ'ЄДНАННЯ ЗАПИТІВ.

## ЗМІСТ

ВСТУП .....	6
1 ОГЛЯД І АНАЛІЗ ПРЕДМЕТУ ПРОЄКТУВАННЯ .....	8
1.1 Методи проєктування реляційної бази даних.....	8
1.2 Засоби програмної реалізації.....	12
1.3 Основні висновки .....	16
2 ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА ОБРОБКА БАЗ ДАНИХ .....	17
2.1 Постановка задачі .....	17
2.2 Проєктування реляційної бази даних за методом нормальних форм.....	17
2.3 Створення та обробка бази даних з використанням MySQL .....	21
2.4 Порівняння характеристик MySQL Workbench з SQL Access .....	27
2.5 Організація самостійних робіт з дисципліни «Бази даних» на основі використання MySQL Workbench .....	28
2.5.1 Використання DML .....	28
2.5.2 Організація різних підходів щодо зв'язування таблиць .....	32
2.5.3 Об'єднання запитів .....	38
2.6 Основні висновки.....	41
ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ.....	42
ПЕРЕЛІК ПОСИЛАНЬ.....	43
ДОДАТОК А.....	<b>Помилка! Закладку не визначено.</b>
Початкове відношення (фрагмент).....	<b>Помилка! Закладку не визначено.</b>
Акт на впровадження .....	<b>Помилка! Закладку не визначено.</b>
ДОДАТОК Б .....	<b>Помилка! Закладку не визначено.</b>
Інструкція до використання MySQL Workbench .....	<b>Помилка! Закладку не визначено.</b>
ДОДАТОК В.....	<b>Помилка! Закладку не визначено.</b>
Тези доповіді .....	<b>Помилка! Закладку не визначено.</b>



## ВСТУП

У сучасному світі бази даних становлять невід’ємну складову діяльності різних організацій. Постійне зростання обсягів інформації змушує обробляти великі обсяги даних. Зробити це власноруч фізично важко, майже неможливо, тому для їх зберігання та обробки доцільно використовувати базу даних, що підтверджує актуальність теми.

Мета кваліфікаційної роботи – проєктування бази даних навчально - методичних видань кафедри «Електронних обчислювальних машин» за методом нормальних форм та її обробка. Відповідно до мети, поставлені наступні завдання:

1. огляд й аналіз предметів проєктування;
2. виконання проєктування бази даних навчально – методичних видань за методом нормальних форм та її обробка;
3. створення та обробка бази даних в MySQL Workbench за допомогою MySQL;
4. порівняння можливостей MySQL Workbench з SQL Microsoft Access за наступними напрямками: використання DML; організація різних підходів до зв’язування таблиць; об’єднання запитів.

Найбільшого розповсюдження та використання на сьогоднішній день здобула реляційна база даних. При використанні цього типу інформація зберігається в реляційних таблицях, де кожен рядок має свій тип. Для створення, редагування та адміністрування баз даних використовують спеціальні програми, які дозволяють це робити, їх називають «Системи управління базами даних» або СУБД. До таких СУБД входять: Microsoft Access, Altibase, Azure, PostgreSQL, Oracle та MySQL.

В наш час майже всі сайти, інтернет-магазини та пошукові сервіси використовують реляційні бази даних, наприклад: Google, Yahoo, Rozetka, MOYO, Wikipedia та багато інших.

На теперішній час, досить багато вчених, студентів та звичайних розробників вивчають, розробляють та удосконалюють можливості створення баз даних,



один з них Роман Гелемб'юк, який у 2018 році опублікував статтю, у якій розповів про свій проєкт під назвою OurSQL. Який представляє з себе розширення MySQL, що дозволяє створити децентралізовану базу даних без вузлів із «особливими» правами.

У 2020 році Вікторією Іващенко, Галиною Горбенко та Сергієм Гіденко була розроблена термінографічна база даних для створення програмного забезпечення з можливістю вибору користувацького інтерфейсу англійською та 10-ма слов'янськими мовами за допомогою Cache методом «Сутність-зв'язок».

Дана кваліфікаційна робота складається з вступу, двох основних розділів та висновків.

У першому розділі розглянуто два методи проєктування реляційних баз даних, а саме метод нормальних форм та метод «Сутність-зв'язок». Також наведені правила щодо використання цих методів. Показані різні системи управління базами даних. В цілому, в першому розділі виконано огляд та аналіз предмета проєктування.

У другому розділі сформульовано постановку задачі та виконано проєктування бази даних навчально-методичних видань кафедри електронних обчислювальних машин за методом нормальних форм. Для проєктування та обробки використана програма MySQL Workbench. Проведено порівняння можливостей MySQL Workbench з SQL Microsoft Access за такими напрямками: використання DML; організація різних підходів до зв'язування таблиць; об'єднання запитів. Рекомендовано створену базу даних навчально - методичних видань до використання адміністрацією кафедри ЕОМ. Надано рекомендацію здобувачам першого (бакалаврського) ступеня спеціальності «Комп'ютерна інженерія», щодо виконання самостійної роботи з дисципліни «Бази даних».

Результати виконання кваліфікаційної роботи представлено у якості тез доповіді до Всеукраїнської науково-технічної конференції студентів і молодих учених, що відбулась у Українському державному університеті науки і технологій у 2022 році.

## **1 ОГЛЯД І АНАЛІЗ ПРЕДМЕТУ ПРОЄКТУВАННЯ**

### **1.1 Методи проєктування реляційної бази даних**

Реляційна база даних (РБД) – це база даних, в якій усі доступні користувачеві дані організовані в таблиці, а всі операції з даними зводяться до операцій над цими таблицями. Реляційна модель зосереджена на організації даних у двовимірних таблицях, де кожна таблиця є двовимірним масивом.

В проєктуванні РБД застосовуються два методи: метод з використанням нормалізації та метод «Сутність-зв'язок».

Спочатку розглянемо метод нормалізації, де весь процес проєктування виконується в реляційній моделі даних шляхом послідовного наближення до набору схем відносин, які будуть відповідати нормам. Початком є вираження предметної області у вигляді одного або кількох зв'язків і на кожному етапі проєктування обираються схеми відносин з найкращими властивостями.

Кожній нормальній формі відповідає свій набір обмежень і знаходження в деякій нормальній формі, якщо вона відповідає властивому їй набору обмежень. Прикладом набору обмежень є обмеження першої нормальної форми, в якій значення всіх атрибутів відносин є одною транзакцією. Оскільки вимога першої нормальної форми є базовою для класичної реляційної моделі даних, можна вважати, що вихідний набір відносин вже відповідає цій вимозі [1].

У реляційній базі даних виділяють наступні нормальні форми: перша нормальна форма (1НФ), друга нормальна форма (2НФ), третя нормальна форма (3НФ), нормальна форма Бойса-Кодда (БКНФ), четверта нормальна форма (4НФ), п'ята нормальна форма або нормальна форма проєкції-з'єднання (5НФ), доменно-ключова нормальна форма (ДКНФ), шоста нормальна форма (6НФ).

До основних властивостей нормальних форм відносять:

- кожна наступна нормальна форма в деякому сенсі краща, ніж попередня;
- при переході до наступної нормальної форми властивості попередніх властивостей зберігаються [1].

Перша нормальна форма. Відношення відповідає 1НФ, якщо всі його атрибути є простими, тобто повинні мати єдині значення. Вихідне відношення

проектується так, щоб воно було в 1НФ [2]. Для переходу відношень в наступну нормальну форму використовують метод «декомпозиції без втрат». Цей метод забезпечує перетворення, щоб запити до вихідного відношення і до відношень, які були отримані в результаті декомпозиції, дали однаковий результат.

Якщо присутня часткова залежність від ключа, то це призводить до наступного: у відношенні присутнє явне і неявне надлишкове дублювання; наслідком надлишкового дублювання є проблема його редагування. Частина надмірності усувається при переході відношення в 2НФ [3].

Друга нормальна форма. Відношення відповідає 2НФ, якщо воно відповідає 1НФ і кожний неключовий атрибут функціонально повно залежить від первинного ключа [2]. Для усунення часткової залежності і переходу відношення в 2НФ потрібно використати операцію проєкції та розкласти його на декілька відношень в такий спосіб:

- побудувати проєкцію без атрибутів, які перебувають у частковій функціональній залежності від первинного ключа;
- побудувати проєкції на частині складеного первинного ключа і атрибути, що залежать від цих частин [3, с.5].

Третя нормальна форма. Відношення відповідає 3НФ, якщо воно відповідає 2НФ і кожний неключовий атрибут нетранзитивно залежить від первинного ключа [2]. Взаємна незалежність атрибутів означає, що різна залежність між атрибутами відношення, у тому числі і транзитивні залежності між ними, відсутні. Зазвичай на практиці побудова 3НФ є достатньою, і приведенням до неї процес проєктування РБД закінчується. Якщо у відношенні є залежність атрибутів складеного ключа від неключових атрибутів, то є необхідним перейти до посиленої третьої нормальної форми.

Посилена третя нормальна форма або форма Бойса-Кодда (БКНФ). Відношення відповідає БКНФ, якщо воно перебуває в 3НФ й у ньому відсутні залежні ключі від неключових атрибутів. БКНФ є 3НФ, якщо відношення мають один потенційний ключ.

Четверта нормальна форма. Відношення відповідає 4НФ, якщо воно знаходиться в БКНФ і всі багатозначні залежності є функціональними від її потенційних ключів. У відношенні  $R(A, B, C)$  існує багатозадачна залежність  $R.A-R.B$  в тому і лише в тому випадку, якщо більшість значень  $B$ , відповідає парі значень  $A$  і  $C$ , залежить тільки від  $A$  і не залежить від  $C$ .

П'ята нормальна форма. Відношення відповідає 5НФ, якщо воно знаходиться в 4НФ і в ній відсутні тяжкі залежні сполучення між атрибутами. Якщо «Атрибут\_1» залежить від «Атрибута\_2», а «Атрибут\_2» залежить від «Атрибута\_3», а «Атрибут\_3» залежить від «Атрибута\_1», то всі три атрибути обов'язково входять в один кортеж [2].

На практиці знайти приклад реалізації для цієї вимоги, в чистому вигляді, досить складно. Будь-яка комбінація з'єднання двох відносин з трьох, неминуче призведе до вилучення невірної інформації. Деякі СУБД забезпечені спеціальними механізмами, які усувають вилучення недостовірної інформації. Проте слід дотримуватись загальної рекомендації: структуру баз даних необхідно будувати таким чином, щоб уникнути застосування 4НФ та 5НФ. П'ята нормальна форма орієнтована на роботу з залежними сполученнями. Указані залежні сполучення між трьома атрибутами зустрічаються дуже рідко. Залежні сполучення між чотирма, п'ятьма і більше атрибутами вказати практично неможливо [2].

Доменно-ключова нормальна форма. Змінна відношення відповідає ДКНФ тоді і лише тоді, коли кожне накладене на неї обмеження є логічним для обмежень доменів та обмежень ключів, які накладені на цю змінну відношення.

Обмеження домену – це обмеження, яке вимагає використовувати для певного атрибута значення лише з певного заданого домену. Обмеження за своєю сутністю є завданням переліку допустимих значень та оголошення про те, що зазначений атрибут має зазначений тип.

Обмеження ключа – це обмеження, яке стверджує, що певний атрибут або комбінація атрибутів є потенційним ключем.

Якщо змінна відповідає ДКНФ, то вона перебуває у 5НФ. Ане не всі змінні можливо привести до ДКНФ.

Шоста нормальна форма. Змінна відношення відповідає 6НФ тоді і лише тоді, коли вона задовольняє всі нетривіальні залежності з'єднання. З визначення зрозуміло, що змінна перебуває в 6НФ і тоді, коли вона може бути піддана подальшій декомпозиції без втрат. Кожна змінна відносин також перебуває у 5НФ [2].

Створення без даних за методом нормальних форм використовується майже у всіх сайтах. Наприклад візьмемо будь-яку онлайн бібліотеку, у якій знаходиться дуже багато книг, статей, журналів, і інформації про них дуже багато. Тому зазвичай при проєктуванні баз даних використовується математичний, більш точний метод нормальних форм.

Далі розглянемо другий метод проєктування, метод «Сутність-зв'язок», який часто називають ER-моделлю. Його відносять до семантичних моделей даних. Більшість сучасних підходів до проєктування баз даних ґрунтується на використанні моделі ER. У зв'язку з наочністю представлення концептуальних схем баз даних, моделі ER широко використовуються в системах CASE, що підтримують автоматизоване проєктування РБД. Основними поняттями моделі ER є сутність та атрибут.

Сутність – це реальний або представлений об'єкт, дані про який повинні зберігатися і бути доступними. У діаграмах ER-моделі сутність представляється у вигляді прямокутника, що містить в собі ім'я суті. Кожен екземпляр суті повинен відрізнитися від інших екземплярів тієї ж суті.

Зв'язок – це графічно зображена асоціація, яка встановлюється між двома сутностями. Зв'язок представляється у вигляді лінії, яка зв'язує дві сутності або веде від однієї суті до неї самої. Обов'язковий зв'язок зображується суцільною лінією, а необов'язковий – переривчастою [1].

Атрибут – це властивість певної сутності.

Як і в нормальних формах РБД, в ER-схемах є поняття нормальних форм, причому їх зміст не сильно відрізняється від змісту самих нормальних форм РБД.

Формулювання нормальних форм у ER-схемах роблять більш зрозумілішим сенс реляційних схем.

У першій нормальній формі ER-схеми усуваються атрибути або групи атрибутів, які повторюються, тобто здійснюється виявлення неявних сутностей, що «замасковані» під атрибути.

У другій нормальній формі ER-схеми усуваються атрибути, які залежать тільки від частин унікального ідентифікатора. Частина унікального ідентифікатора визначає окрему сутність.

У третій нормальній формі ER-схеми усуваються атрибути, які залежать від атрибутів, що не входять в унікальний ідентифікатор. Ці атрибути є основою окремої сутності [1].

## **1.2 Засоби програмної реалізації**

Система управління базами даних (СУБД) – це комплекс програмно-мовних засобів, які дозволяють створити та керувати базами даних. Іншими словами, СУБД – це набір програм, що дозволяють контролювати, адмініструвати та редагувати базу даних.

В наш час більшість сайтів не можуть працювати без бази даних, тому СУБД використовується майже всюди. Основні функції СУБД: підтримка мов баз даних; збереження історії змін, резервне копіювання та відновлення баз даних після збоїв; управління даними на зовнішній пам'яті; управління даними в оперативній пам'яті з використанням кеша [4].

Сьогодні найбільш популярною моделлю є реляційна база даних. Для керування реляційними базами даних використовується мова Structured Query Language (SQL – структурована мова запитів). Рейтинг найпопулярніших СУБД очолюють: Altibase, MySQL, Oracle, SQL Azure, PostgreSQL.

Altibase – це високопродуктивна реляційна СУБД корпоративного рівня з відкритим вихідним кодом. Єдина база даних, яка забезпечує високоінтенсивну обробку даних через пам'ять частин БД і великий обсяг сховища через частину БД на диску. Altibase використовує метод реляційної бази даних у пам'яті, яка у 10 разів швидша, ніж звичайні дискові бази даних. Має гібридну архітектуру, що

поєднує базу даних у пам'яті та базу даних на диску в одному механізмі. Також має надійний набір драйверів: ODBS, JDBC, NET і Entity Framework, OLEDB, вбудований SQL, CLI, Perl DBD, PHP. Присутня підтримка загальних протоколів зв'язку таких, як: TCP/IP (IPv4 і IPv6), доменний сокет UNIX, IPC. Вміщує в собі багатий набір інструментів та утиліт:

#### 1. Продуктивність і адміністрування:

- iSQL (утиліта SQL командного рядка);
- Orange Altibase (утиліта SQL для графічного інтерфейсу WareValley);
- менеджер реплікації (утиліта реплікації графічного інтерфейсу);
- altiComp (утиліта для порівняння бази даних командного рядка);
- altiProfile (утиліта профілювання бази даних командного рядка);
- altiMon (утиліта моніторингу бази даних командного рядка).

#### 2. Сумісність та міграція:

- центр міграції (утиліта міграції GUI);
- aExport (утиліта міграції командного рядка);
- iLoader (утиліта імпорту / експорту даних командного рядка);
- oraAdaptor (передача даних для Oracle Utility).

Altibase співпрацює з такими компаніями, як: Solaris, HP, IBM, Micrisoft, Linux [5].

PostgreSQL – це потужна система об'єктно-реляційних баз даних з відкритим вихідним кодом, яка використовує та розширює мову SQL у поєднанні з багатьма функціями, що безпечно зберігають і масштабують найскладніші робочі навантаження даних. PostgreSQL має міцну репутацію завдяки своїй перевірній архітектурі, надійності, цілісності даних та надійному набору функцій. Працює PostgreSQL на всіх основних операційних системах, також має ACID – сумісність з 2001 року і має потужні надбудови такі, як: популярний розширювач геопросторної бази даних PostGIS та інші. Система постачається з багатьма функціями, які допомагають розробникам створювати свої програми, адміністраторам захищати цілісність даних, а також допомагають керувати

даними незалежно від того, наскільки великий чи маленький набір даних. СУБД є безкоштовною, а також вона дуже розширювана. Наприклад, можна визначати власні типи даних, створювати власні функції та навіть писати код з різних мов програмування без перекомпіляції бази даних. Доведено, що PostgreSQL має високу масштабованість як за величезною кількістю даних, якими він може керувати, так і за кількістю одночасних користувачів, які він може в себе вмістити. У виробничих середовищах є активні кластери PostgreSQL, які керують багатьма терабайтами даних, і спеціалізовані системи, що керують петабайтами даних [6].

Oracle Database – це об’єктно-реляційна система управління базами даних, від компанії Oracle. Вона використовується для створення структури нової бази, її заповнення, редагування вмісту та відображення інформації. Однією з характеристик СУБД є незалежність фізичного зберігання даних від логічних структур даних. В Oracle Database схема бази даних – це набір логічних структур даних або об’єктів схеми. Також в Oracle присутнє PL/SQL. Воно інтегровано з Oracle Database, що дозволяє використовувати всі оператори функцій та типи даних Oracle Database SQL.

PL/SQL – це процедурне розширення Oracle SQL. Основною перевагою PL/SQL є можливість зберігати логіку програми в самій базі даних. Процедура або функція PL/SQL – це об’єкт схеми, що складається з набору операторів SQL та інших конструкцій PL/SQL, згрупованих разом, які зберігаються в базі даних і виконуються як одиниця для вирішення певної проблеми. Основна перевага програмування на стороні сервера полягає в тому, що вбудовані функції можуть бути розгорнуті в будь-якому місці. Oracle Database також може зберігати програмні блоки, написані на Java. Збережена процедура Java – це метод Java, опублікований у SQL і збережений у базі даних для загального використання [7].

Microsoft SQL Server – система управління реляційними базами даних, розроблена корпорацією Microsoft. З недавніх пір Microsoft створила нову систему баз даних, а саме SQL Azure.



База даних SQL Azure – це повністю керована платформа, як служба бази даних (PaaS), яка обробляє більшість функцій керування базою даних таких, як: оновлення, виправлення, резервне копіювання та моніторинг без участі користувача. База даних SQL Azure завжди працює на останній стабільній версії СУБД SQL Server і виправлена ОС із високою доступністю. Можливості PaaS, вбудовані в базу даних SQL Azure, дають змогу зосередитися на адмініструванні та оптимізації бази даних [8].

MySQL – реляційна СУБД з відкритим вихідним кодом, головними перевагами якої є її швидкість та гнучкість, що забезпечена підтримкою великої кількості різних типів таблиць. З самого початку система створювалась для обмеженого користування, але сьогодні вона сумісна з Windows, Linux, macOS та Ubuntu. Великі компанії (Google, Facebook, YouTube, Twitter та Yahoo!) використовують для зберігання даних MySQL. За популярністю MySQL поступається лише Oracle [9].

В MySQL базова структура системи клієнт-сервер дуже проста. Клієнт під'єднується до сервера через мережу. Клієнт робить запит з інтерфейсу користувача, і, якщо серверу зрозумілі інструкції, він видає інформацію [10]. У досвідчених користувачів та у тих, хто тільки почав вивчати все про базу даних, не виникне проблем з роботою в MySQL, тому що в мережі дуже багато інформації, яка допоможе в ній працювати. До основних переваг MySQL можна віднести такі:

- повністю безкоштовна СУБД;
- необмежений розрахований на велику кількість користувачів;
- безліч плагінів, які полегшують роботу з СУБД;
- підтримка різних типів таблиць (MyISAM, InnoDB, HEAP, MERGE);
- дозволяє додавати до 50 мільйонів рядків у таблиці.

Проте є й недоліки:

- обмежений функціонал (не реалізовані всі можливості SQL);
- можливі проблеми з надійністю зберігання та передачі даних через відкритий вихідний код.

Таким чином, MySQL – це безкоштовна, проста СУБД з відкритим кодом. Звісно, вона не позбавлена недоліків, але в більшості випадків являє собою оптимальне рішення під час роботи з даними.

### **1.3 Основні висновки**

1. Найбільше практичне застосування отримала реляційна база даних, в якій для проєктування використовується два методи: нормальних форм та «Сутність-зв'язок». Для проєктування обрано метод нормальних форм, тому що він математичний та являється більш точним.

2. Для реалізації існує багато програм, наприклад: Altibase, MySQL, Oracle, SQL Azure, PostgreSQL. Для створення бази даних навчально-методичних видань, обрано програму MySQL Workbench, так як, вона має всі необхідні інструменти, повністю безкоштовною та є простою в використанні.

## **2 ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА ОБРОБКА БАЗ ДАНИХ**

### **2.1 Постановка задачі**

Метою роботи є створення бази даних навчально-методичних видань кафедри ЕОМ за методом нормальних форм та її обробка.

Для вихідних даних взята інформація з репозиторію Українського державного університету науки і технологій (УДУНТ) із науковими розробками та даними про науково-педагогічний склад кафедри ЕОМ.

Головною задачею, яка виконується у процесі проєктування реляційної бази даних, є нормалізація її відношень. При проєктуванні бази даних наукових розробок кафедри застосовувався метод нормалізації, в якому весь процес виконується шляхом послідовного наближення до тих наборів відносин, що будуть відповідати нормам.

### **2.2 Проєктування реляційної бази даних за методом нормальних форм**

Спочатку складено вихідне відношення, в якому 13 атрибутів та 93 кортежів. Початкове відношення представлено в таблиці А.1.

Початкове відношення «DB\_EMP» (database of educational and methodical publications – база даних навчально-методичних видань). Воно знаходиться в першій нормальній формі, тому що всі його атрибути атомарні. Розглянемо атрибути бази даних:

- ПІБ – прізвище, ім'я, по батькові автора методичного видання;
- Науковий\_ступінь – науковий ступінь, присуджений автору;
- Посада – назва посади, яку займає автор;
- Електронна\_адреса – електронна адреса автора;
- Мобільний\_телефон – мобільний телефон автора;
- Спеціальність – номер та назва спеціальності;
- Предмет – назва предмету, для якого видано методичне видання;
- Назва\_видання – назва методичного видання;
- Вид\_видання – вид методичного видання;
- Видавництво – видавництво, яке видало видання;

- Рік\_видання – рік створення методичного видання;
- Кількість\_сторінок – загальна кількість сторінок в методичному виданні;
- Ступінь\_освіти – освітній ступінь, для якого розроблено методичне видання.

Первинний ключ відношення складений, складається з трьох атрибутів (ПІБ, Назва\_видання, Спеціальність).

На першому етапі проєктування знайдемо залежності між атрибутами. Залежність представлена на рисунку 2.1.

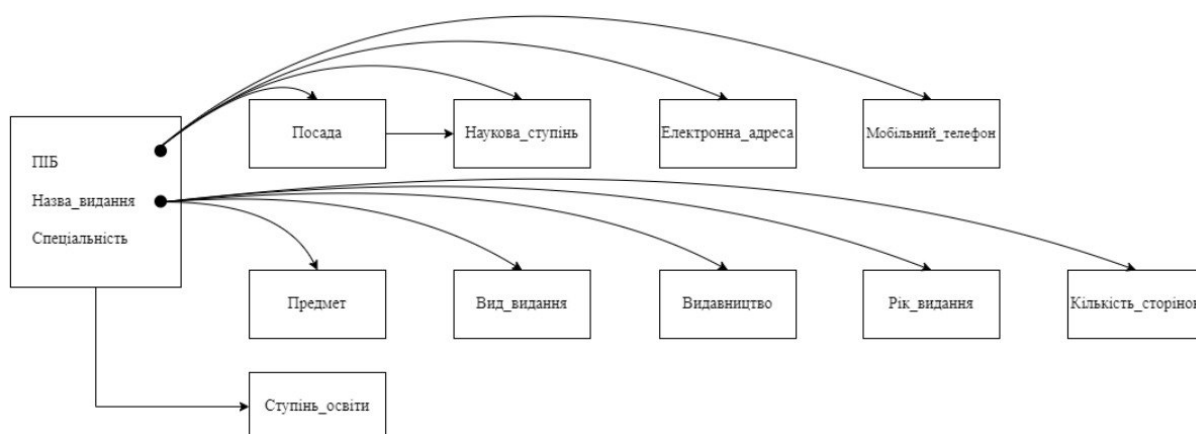


Рисунок 2.1 – Залежність між атрибутами відношення

Приведення відношення до другої нормальної форми.

Для усунення часткової залежності і переходу відношення в другу нормальну форму потрібно використати операцію проєкції, та розкласти його на декілька відношень:

- побудувати проєкцію без атрибутів, які перебувають у частковій функціональній залежності від первинного ключа;
- побудувати проєкції на частині складеного первинного ключа і атрибути, що залежать від цих частин.

В результаті отримано відношення R1, R2 та R3.

Відношення R1 представлено на рисунку 2.2.

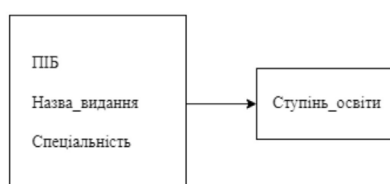


Рисунок 2.2 – Відношення R1

У відношенні R1 первинний ключ складений: (ПІБ, Назва\_видання, Спеціальність).

Проміжне відношення R1 представлено в таблиці А.2.

Відношення R2 представлено на рисунку 2.3.

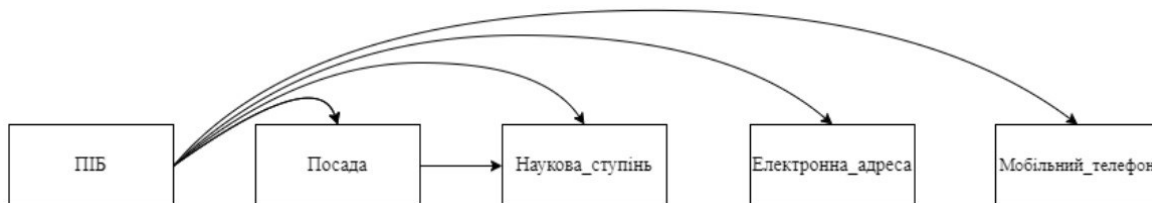


Рисунок 2.3 – Відношення R2

У відношенні R2 первинний ключ простий: ПІБ. Але у відношенні R2 спостерігається неявне дублювання даних.

Проміжне відношення R2 представлено в таблиці А.3.

Відношення R3 представлено на рисунку 2.4.

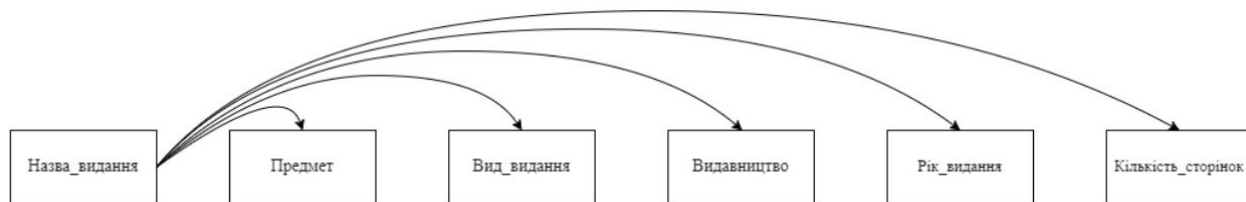


Рисунок 2.4 – Відношення R3

У відношенні R3 первинний ключ простий: Назва\_видання.

Проміжне відношення R3 представлено в таблиці А.4.

Відношення заходиться у третій нормальній формі, якщо воно знаходиться у другій нормальній формі і кожний його неключовий атрибут нетранзитивно залежить від первинного ключа. Тобто другий етап нормалізації – позбавлення транзитивних залежностей. Відношення R1 та R3, які отримані на попередньому етапі нормалізації, залишаються незмінними, тому що в них відсутні транзитивні залежності. А у відношенні R2 присутня транзитивна залежність: Посада; Науковий\_ступінь. Для ліквідації транзитивної залежності потрібно використати операцію проєкції на атрибути, тоді отримаємо відношення R4 та R5.

Відношення R4 представлено на рисунку 2.5.

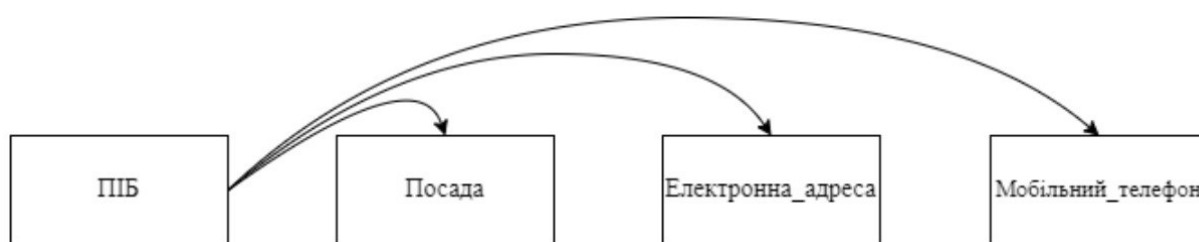


Рисунок 2.5 – Відношення R4

У відношенні R4 відсутні транзитивні залежності.

Проміжне відношення R4 представлено в таблиці А.5.

Відношення R5 представлено на рисунку 2.6.

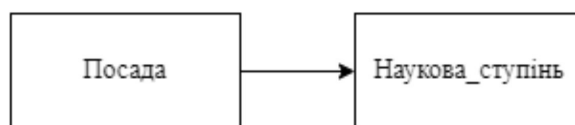


Рисунок 2.6 – Відношення R5

Кінцеве відношення R5 представлено в таблиці А.6.

Структура спроектованої бази даних за методом нормальних форм представлена на рисунку 2.7.



Рисунок 2.7 – Структура спроектованої бази даних за методом нормальних форм

Результатом проектування є база даних, яка складається з таблиць R1, R3, R4 та R5.

У відношенні R1 ключовий атрибут ПІБ не залежить від неключових атрибутів Назва\_видання, Спеціальність, .

У відношенні R3 ключовий атрибут ПІБ не залежить від атрибутів Науковий\_ступінь, Електронна\_адреса, Мобільний\_телефон.

У відношенні R4 ключовий атрибут Назва\_видання не залежить від атрибутів Предмет, Вид\_видання, Видавництво, Рік\_видання, Кількість\_сторінок.

У відношенні R5 ключовий атрибут Посада не залежить від атрибута Науковий\_ступінь.

Отримані відношення R1, R3, R4, R5 – задовольняють вимогам нормальної форми Бойса-Кодда, тому що вони знаходяться в третій нормальній формі, а їх первинні ключі не залежать від неключових атрибутів. Саме тому процес нормалізації можна вважати закінченим.

### **2.3 Створення та обробка бази даних з використанням MySQL**

MySQL Workbench – це візуальний інструмент проєктування баз даних, який поєднує проєктування, моделювання, створення та експлуатацію бази даних у єдиному середовищі для системи баз даних MySQL. Інструкція по використанню представлена у додатку Б.

Тепер, коли ознайомились з основними розділами програми, можна переходити до створення нової бази даних. Для створення бази даних існує, як мінімум, два способи. Розглянемо найбільш базовий.

Спочатку потрібно під'єднатися до системи MySQL Server, для цього на стартовому меню обираємо своє підключення до серверу та подвійним натиском на нього, під'єднуємось. Під'єднавшись, бачимо внутрішню структуру сервера, який представлений на рисунку 2.8.

Тепер перейдемо до створення самої бази даних. Для першого способу створення необхідно написати у вікні SQL редактора наступну команду, яка представлена на малюнку 2.9.

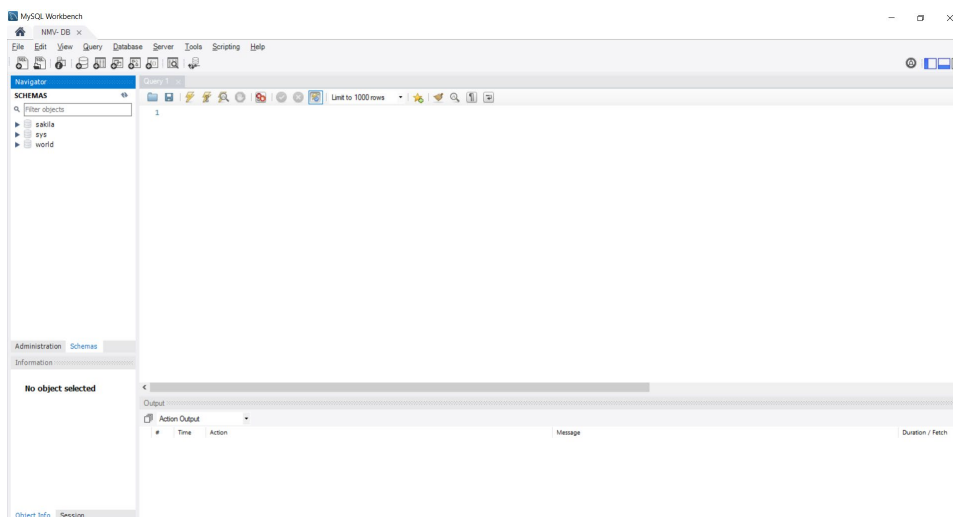


Рисунок 2.8 – Внутрішня структура SQL сервера

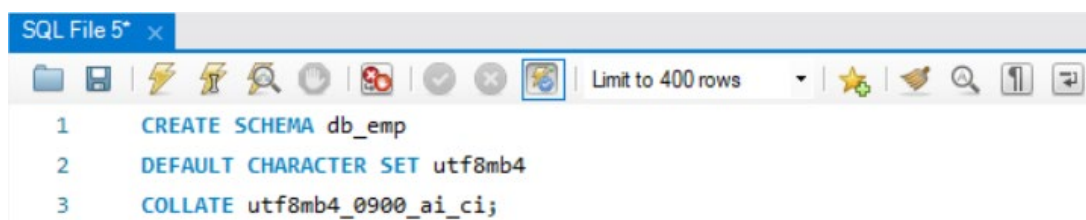


Рисунок 2.9 – Вікно SQL редактора

За замовчуванням бази даних MySQL підтримують стандартні юнікодівські символи, але в них певні українські символи відображаються некоректно. Тому в другому рядку команди вказали, який набір символів будемо використовувати за замовчуванням. А в третьому рядку задали правила пошуку тексту. Закінчення «сі» може означати нечутливість до регістру символів. Це налаштування вказує базі даних, що пошук по базі даних буде виконуватись на співпадіння і великими, і малими літерами. Коли команда написана, її можна виконати, для цього натискаємо на кнопку «виконання команди», що зображена у вигляді блискавки. Результат виконання зображений в нижній частині вікна, який представлений на рисунку 2.10. Тут показано, що команда виконалась успішно, - це можна зрозуміти по зеленій позначці.

Output					
Action Output					
#	Time	Action	Message	Duration / Fetch	
✓ 1	20:12:41	CREATE SCHEMA db_emp DEFAULT CHARA...	1 row(s) affected	0.015 sec	

Рисунок 2.10 – Результат виконання команди



І якщо тепер перейдемо в секцію SCHEMAS, яка зображена на рисунку 2.11 та натиснемо оновити, - побачимо нашу створену базу даних.

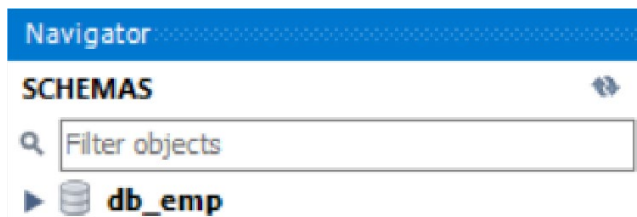


Рисунок 2.11 – Секція SCHEMAS.

Існування баз даних без таблиць не має сенсу, тому перейдемо до створення та наповнення інформацією таблиць.

В редакторі SQL напишемо команду для створення нової таблиці, вона представлена на рисунку 2.12.

Перший рядок команди означає, що створення таблиці буде відбуватись саме зі створеною базою даних під назвою «db\_emp».

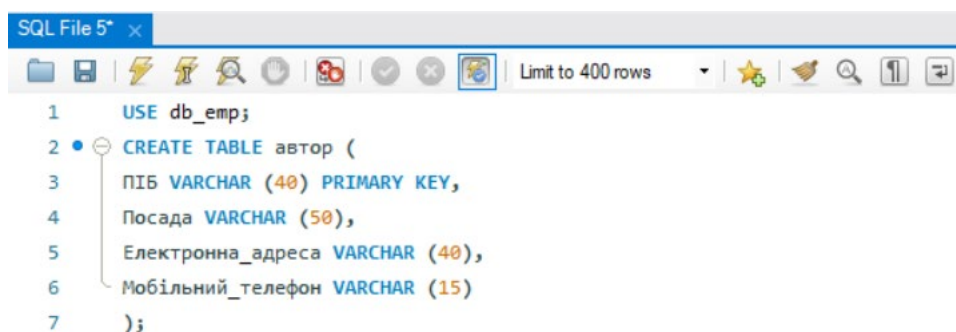


Рисунок 2.12 – Команда для створення нової таблиці

В наступному рядку команди створюємо таблицю під назвою «Автор». Назву таблиці пишемо маленькими літерами, тому що так потребує сама програма.

З третього по шостий рядок пишемо з чого буде складатися наша таблиця. Тут є три можливих атрибути: ім'я стовпця, тип даних та додаткові атрибути. Наприклад, в третьому рядку записано ім'я стовпця – ПІБ, вказали до якого типу даних відноситься ця інформація в стовпці – VARCHAR (текстовий тип даних) та зазначили кількість символів, яку дозволить записати база даних в цей стовпець, також назначили додатковий атрибут – PRIMARY KEY до стовпця ПІБ, який дає зрозуміти, що стовпець ПІБ є первинним ключем.

Тепер виділяємо всю команду та натискаємо на кнопку «виконати». Якщо подивитись на секції SCHEMAS на рядок Tables, який представлений на рисунку 2.13, та оновити дані, побачимо, що в списку з'явилась наша таблиця.

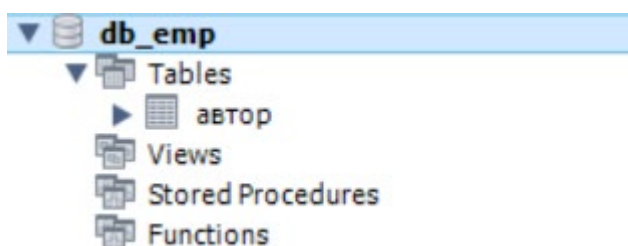


Рисунок 2.13 - Секції SCHEMAS з таблицею АВТОР

Після створення всіх потрібних нам таблиць секція SCHEMAS виглядає так, як показано на рисунку 2.14.

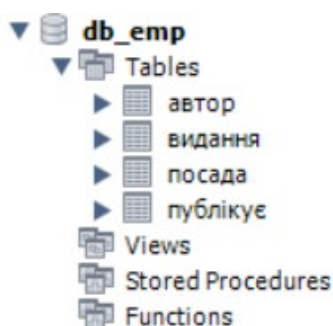


Рисунок 2.14 - Секція SCHEMAS з усіма таблицями

Тепер почнемо наповнювати саму таблицю всіма наявними даними. Для цього у вікні SQL редактора пишемо команду, яка представлена на рисунку 2.15.

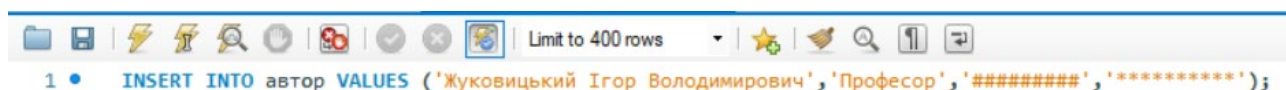


Рисунок 2.15 – Наповнення таблиці даними

Таким чином, виконується додавання в таблицю «автор» даних одного автора. Послідовність значень в дужках повинна співпадати з послідовністю записаних атрибутів.

Після виконання команди, зображеної на рисунку 2.18, дані, які ввели, додалися до таблиці «автор». Щоб переглянути їх наявність у таблиці, потрібно ввести команду виклику таблиці, яка зображена на рисунку 2.16.



Рисунок 2.16 – Команда виклику таблиці

Після виконання команди у нас з'явиться вікно з вибраною таблицею, зображеною на рисунку 2.17.

ПІБ	Посада	Електронна_адреса	Мобільний_телефон
Жуковицький Ігор Володимирович	Професор	#####	*****

Рисунок 2.17 – Таблиця АВТОР

Заповнюємо таблиці даними таким самим чином, як описано вище, і отримуємо результат, який зображений у таблиці 2.18.

ПІБ	Посада	Електронна_адреса	Мобільний_телефон
Жуковицький Ігор Володимирович	Професор	#####	*****
Косолапов Анатолій Аркадійович	Доцент	#####	*****
Доманська Галина Анатоліївна	Доцент	#####	*****
Егоров Олег Йосипович	Доцент	#####	*****
Остапеч Денис Олександрович	Доцент	#####	*****
Устенко Андрій Борисович	Доцент	#####	*****
Шаповалов Володимир Олександрович	Доцент	#####	*****
Беляев Микола Борисович	Старший викладач	#####	*****
Дзюба Володимир Володимирович	Старший викладач	#####	*****
Заєць Олексій Петрович	Старший викладач	#####	-
Івін Павло Вікторович	Старший викладач	#####	*****
Лобода Дмитро Геннадійович	Старший викладач	#####	-
Скабалланович Тетяна Іванівна	Старший викладач	#####	*****
Пахомова Вікторія Миколаївна	Доцент	#####	*****

Рисунок 2.18 – Заповнена таблиця АВТОР

Тепер перейдемо до побудови зв'язку між таблицями. Для цього необхідно ввести SQL команду, яка зображена на рисунку 2.19.

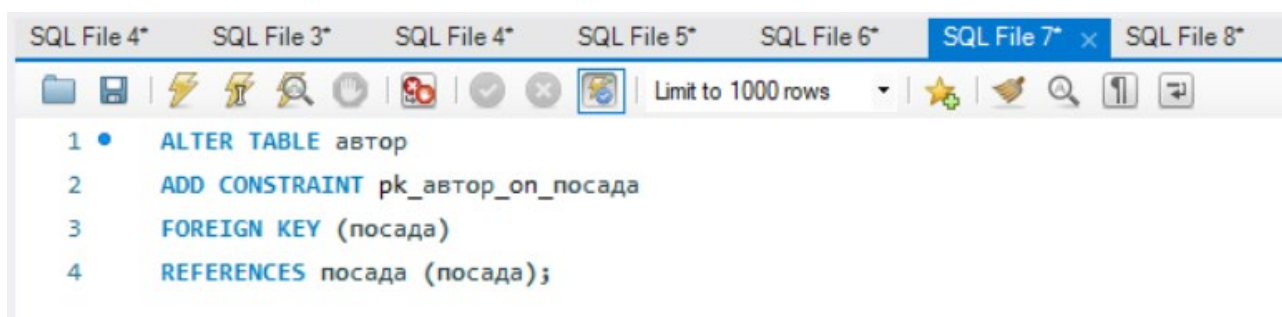


Рисунок 2.19 – Команда для створення зв'язку між таблицями

У першому рядку вказуємо, що будемо змінювати таблиці «автор». У другому рядку вказуємо, що додаємо обмеження (Constraint), тобто вказуємо, що до таблиці «автор», буде додано певне обмеження. Далі вказуємо назву обмеження. В третьому рядку показуємо, що це обмеження буде типу Foreign Key (зовнішній ключ) для стовпчика «Посада» в таблиці «автор». І тепер це

обмеження логічно прив'язує стовпець «Посада» в таблиці «автор» із стовпцем «Посада» в таблиці «Посада».

Щоб побачити чи з'явився зв'язок між двома таблицями, необхідно в секції SCHEMAS навести курсор на рядок таблиці, яку було з'єднано, та натиснути на значок гайкового ключа, як вказано на рисунку 2.20.

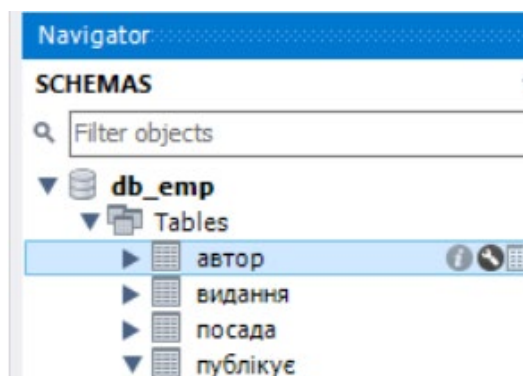


Рисунок 2.20 – Позначка гайкового ключа

Натиснувши на значок, з'явиться вікно таблиці, яку було обрано, далі необхідно перейти у розділ «Foreign Key», як показано на рисунку 2.21. І тепер побачимо зв'язок між таблицями, який щойно був доданий.

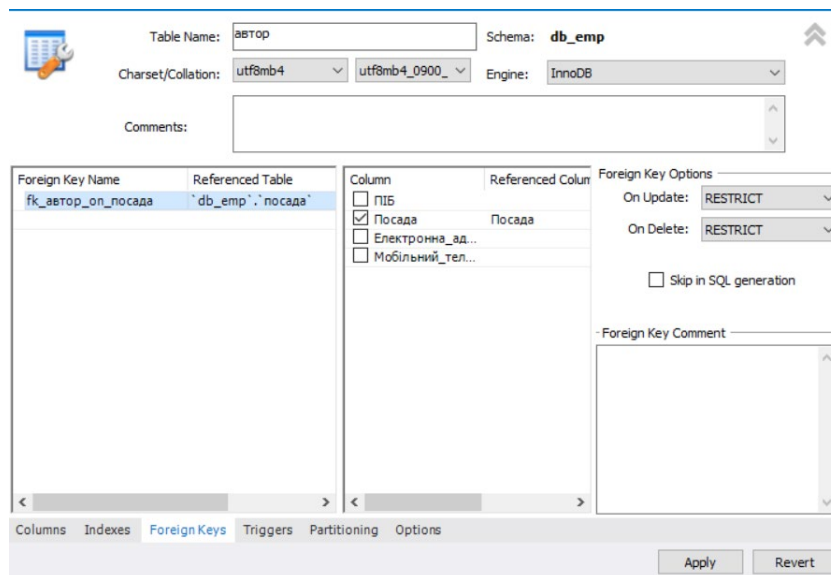


Рисунок 2.21 – Знаходження зовнішнього ключа

Таким самим способом будуюмо зв'язок між всіма таблицями.

Тепер наша база даних під назвою «DB\_EMP» готова.

## 2.4 Порівняння характеристик MySQL Workbench з SQL Access

Для зрівняння програм MySQL Workbench із Microsoft Access використаємо такі аспекти, як: основні характеристики, інтерфейс, вартість.

### 1. Основні характеристики.

MySQL – це потужна база даних корпоративного рівня, яка використовується багатьма сайтами та магазинами в світі. MySQL – це СУБД на основі SQL, яка розрахована на велику кількість користувачів, буде працюватиме надійно і продуктивно, та обмежена лише доступною оперативною пам'яттю, швидкістю диска і швидкістю обладнання, а не властивими програмними обмеженнями.

Microsoft Access – досить хороший для домашніх користувачів, але не може працювати з більш, ніж 2 гігабайтами даних і не повинен використовуватися на такому рівні. Microsoft Access – це чудова програма для вивчення баз даних. Microsoft Access – це інструмент, в основному, для роботи з базами даних, який ідеальний для доступу до різних корпоративних систем управління базами даних SQL, електронних таблиць Excel, CSV-файлів та інших джерел даних, також є можливість створення зручних форм, запитів, звітів і навіть цілих зовнішніх програм. У Microsoft Access вбудована СУБД SQL, яка, зазвичай, розрахована на одного користувача. Вона може використовуватися багатьма користувачами СУБД, але реальний максимум – шість користувачів, хоча при такій кількості, вона буде повільна та ненадійна.

### 2. Інтерфейс.

В обох системах управління базами даних інтерфейс дуже простий, зручний та інтуїтивно зрозумілий. MySQL більше підходить для користувачів, які вже мали досвід роботи в СУБД, а Microsoft Access краще підійде для початкового ознайомлення та подальшого вивчення предметної області, бо інтерфейс схожий з Microsoft Word та Excel .

### 3. Вартість.

MySQL є повністю безкоштовною СУБД для навчання та домашнього користування, але якщо користувач захоче вийти на більш високий рівень, він може придбати платну версію.

Microsoft Access є безкоштовною, але для її отримання необхідно придбати ліценцію Windows, тому вважати програму повністю безкоштовною частково невірно.

Крім того зрівняли можливості використання DML, організацію різних підходів щодо зв'язування таблиць та об'єднання запитів.

Порівняння MySQL Workbench з SQL Microsoft Access виконуємо на основі запитів підмов, які представлені в підпунктах 2.5.1, 2.5.2 та 5.2.3.

На підставі проведених порівнянь, виявлено, що різниця у синтаксисах є лише у деяких різновидах з'єднання таблиць.

У природньому з'єднанні в MySQL Workbench використовується оператор NATURAL JOIN, але в SQL Microsoft Access цей оператор не підтримується, а замість нього використовується INNER JOIN і ключове слово ON, після якого знаходиться умова відбору записів.

При умовному з'єднанні в MySQL Workbench використовується оператор JOIN ON, а у SQL Microsoft Access – INNER JOIN ON.

Також різниця запису синтаксису є і в зовнішньому з'єднанні. При лівому та правому з'єднанні в MySQL Workbench використовується оператор LEFT OUTER та RIGHT OUTER JOIN. В SQL Microsoft Access ключове слово OUTER, не допускається.

Судячи з невеликої кількості відмінностей та самою різницею операторів, повністю переробляти лабораторні роботи немає сенсу.

## **2.5 Організація самостійних робіт з дисципліни «Бази даних» на основі використання MySQL Workbench**

### **2.5.1 Використання DML**

DML – частина підмов SQL, яка дозволяє маніпулювати даними, тобто вносити реальні зміни в реляційні бази даних. За допомогою DML користувач може додавати у таблиці нові дані, оновлювати вже існуючі дані і видаляти їх з таблиць.

Для роботи з DML існує три команди: INSERT, UPDATE, DELETE.

Оператор INSERT використовують, щоб увести дані в таблицю.



Для списку VALUES існують наступні правила для значень у стовпцях:

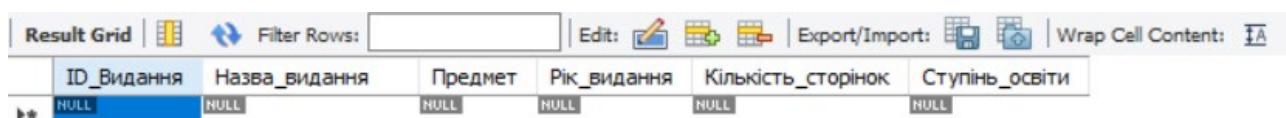
- для всіх стовпців таблиці повинні бути задані значення;
- символічні значення і значення дат беруться в лапки;
- для числових і порожніх значень, які задаються ключовим словом NULL, лапки не потрібні.

Завдання 1. Створити таблицю та додати до неї новий запис.

Приклад 1. Для початку створимо таблицю, наприклад, ВИДАННЯ, використовуючи таку команду:

```
CREATE TABLE видання (ID_Видання INT PRIMARY KEY,  
Назва_видання VARCHAR(250) NOT NULL,  
Предмет VARCHAR(40), Рік_видання INT,  
Кількість_сторінок INT, Ступінь_освіти VARCHAR(40) );
```

В результаті отримаємо пусту таблицю, яка зображена на рисунку 2.22.



ID_Видання	Назва_видання	Предмет	Рік_видання	Кількість_сторінок	Ступінь_освіти
NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 2.22 – Новостворена таблиця ВИДАННЯ

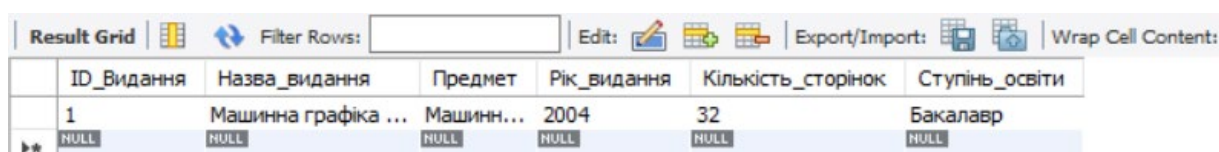
У таблицю ВИДАННЯ додаємо новий запис, у якому: ID\_Видання – 1, Назва видання – Машинна графіка : Метод. вказівки до лабораторних робіт, Предмет – Машинна графіка, Рік\_видання – 2004, Кількість\_сторінок – 32, Ступінь\_освіти – Бакалавр.

Порядок значень, які вводяться, повинні відповідати порядку стовпців у таблиці.

Для додавання нового запису в таблицю, використовуємо наступну команду:

```
INSERT INTO видання VALUES (1, 'Машинна графіка : Метод. вказівки до  
лабораторних робіт', 'Машинна графіка', 2004, 32, 'Бакалавр');
```

Результат виконаної команди зображено на рисунку 2.23.



ID_Видання	Назва_видання	Предмет	Рік_видання	Кількість_сторінок	Ступінь_освіти
1	Машинна графіка ...	Машинн...	2004	32	Бакалавр

Рисунок 2.23 – Результат виконаної команди

Також існує можливість вводити не всі дані, а тільки в обрані поля.

При такому варіанті слід дотримуватись наступного:

- список значень, які вводяться, повинні обов'язково відповідати списку обраних стовпців;
- список полів в операторі INSERT не обов'язково повинний відповідати списку стовпців у вибраній таблиці.

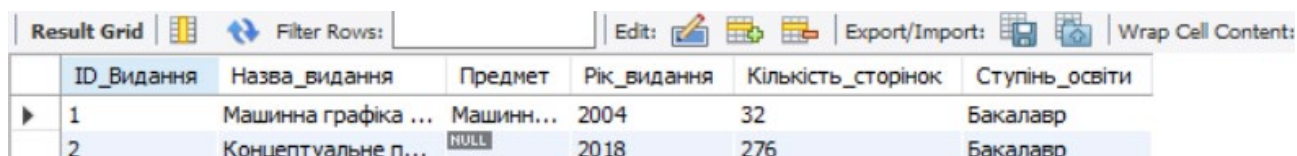
Приклад 2. В таблицю ВИДАННЯ, що зображена на рисунку 2.23, додаємо новий запис, у якому: ID\_Видання – 2, Назва видання – Концептуальне проєктування комп'ютерних систем реального часу, Рік\_видання – 2018, Кількість\_сторінок – 276, Ступінь\_освіти - Бакалавр.

Для додавання запису використовуємо таку команду:

INSERT INTO видання (ID\_Видання, Назва\_видання, Рік\_видання, Кількість\_сторінок, Ступінь\_освіти)

VALUES (2, Концептуальне проєктування комп'ютерних систем реального часу', ' Комп'ютерні системи ', 2018, 276, 'Бакалавр');

Результат виконаної команди зображено на рисунку 2.24.



ID_Видання	Назва_видання	Предмет	Рік_видання	Кількість_сторінок	Ступінь_освіти
1	Машинна графіка ...	Машинн...	2004	32	Бакалавр
2	Концептуальне п...	NULL	2018	276	Бакалавр

Рисунок 2.24– Команда запису

Як бачимо, записи створені лише для тих стовпців, які було обрано, а вийшло це завдяки тому, що стовпець Предмет не вимагає обов'язкової наявності даних у кожному записі, оскільки ці поля не визначені, як NOT NULL.

Дані одного чи декількох полів, які вже існують в таблиці, за потреби можна змінити за допомогою команди UPDATE.

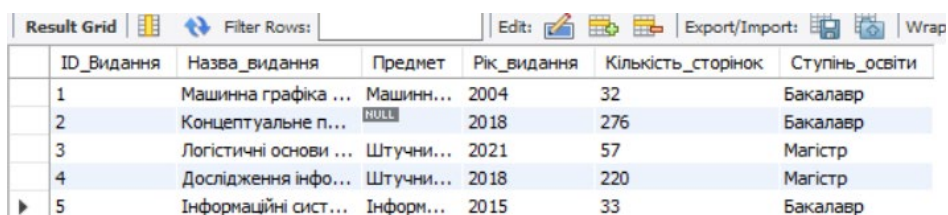
У найпростішій формі оператор UPDATE змінює один стовпець таблиці.

При зміні одного поля, також можна змінити одразу декілька записів або тільки один. Це визначається параметром, який заданий ключовим словом WHERE.

Завдання 2. Змінити запис в таблиці.



Приклад 1. У таблиці ВИДАННЯ, яка зображена на рисунку 2.25, оновити запис поля Предмет, установивши значення «Комп'ютерні системи», для якого значення в полі ID\_Видання дорівнює 2.



	ID_Видання	Назва_видання	Предмет	Рік_видання	Кількість_сторінок	Ступінь_освіти
1	1	Машинна графіка ...	Машинн...	2004	32	Бакалавр
2	2	Концептуальне п...	NULL	2018	276	Бакалавр
3	3	Логістичні основи ...	Штучни...	2021	57	Магістр
4	4	Дослідження інфо...	Штучни...	2018	220	Магістр
5	5	Інформаційні сист...	Інформ...	2015	33	Бакалавр

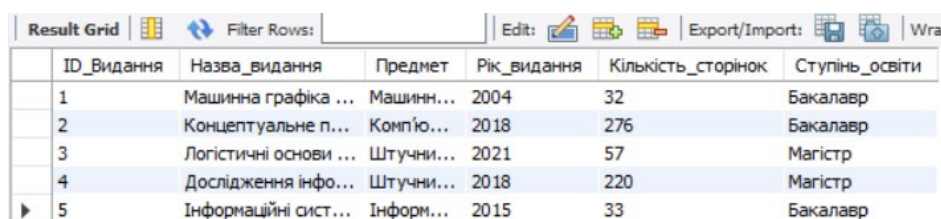
Рисунок 2.25 – Таблиця ВИДАННЯ

Відповідний оператор має такий вид:

UPDATE видання SET Предмет = 'Комп'ютерні системи'

WHERE ID\_Видання = 2;

У результаті роботи оператора був оновлений тільки один запис. Результат показаний на рисунку 2.26.



	ID_Видання	Назва_видання	Предмет	Рік_видання	Кількість_сторінок	Ступінь_освіти
1	1	Машинна графіка ...	Машинн...	2004	32	Бакалавр
2	2	Концептуальне п...	Комп'ю...	2018	276	Бакалавр
3	3	Логістичні основи ...	Штучни...	2021	57	Магістр
4	4	Дослідження інфо...	Штучни...	2018	220	Магістр
5	5	Інформаційні сист...	Інформ...	2015	33	Бакалавр

Рисунок 2.26 – Результат виконання оператора

Важливо зазначити, що при відсутності умови, яка задана ключовим словом WHERE, дані відповідного поля будуть оновлені у всіх записах, що використовується дуже рідко.

При відновленні декількох полів в одному чи кількох записах при написанні цього оператора необхідно знати:

- ключове слово SET використовується тільки один раз;
- при відновленні декількох полів, вони розділяються комами;
- завдання умови ключовим словом WHERE необов'язково;
- у реченні SET команди UPDATE можна використовувати скалярні вирази.

Приклад 2. У таблиці ВИДАННЯ, яка вказана на рисунку 2.26, виконати оновлення декількох полів: поле Рік\_видання, встановити значення 2019; поле Кількість\_сторінок, встановити значення 278, якщо значення в полі ID\_Видання дорівнює 2. Напишемо відповідний оператор:

UPDATE видання SET Рік\_видання = 2019,

Кількість\_сторінок = 278 WHERE ID\_Видання = 2;

В результаті виконання буде оновлений тільки один запис. Результат представлено на рисунку 2.27.

ID_Видання	Назва_видання	Предмет	Рік_видання	Кількість_сторінок	Ступінь_освіти
1	Машинна графіка ...	Машинн...	2004	32	Бакалавр
2	Концептуальне п...	Комп'ю...	2019	278	Бакалавр
3	Логістичні основи ...	Штучни...	2021	57	Магістр
4	Дослідження інфо...	Штучни...	2018	220	Магістр
5	Інформаційні сист...	Інформ...	2015	33	Бакалавр

Рисунок 2.27 – Результат виконання

Для видалення інформації з таблиць використовують команду DELETE. Команда DELETE призначена не для того, щоб видаляти значення окремих полів, а для видалення цілих записів.

Якщо ключове слово WHERE в операторі DELETE опущено, то будуть видалені всі записи з таблиці [11].

Завдання 1. Видалити дані з таблиці.

Приклад 1. З таблиці ВИДАННЯ, зображеної на рисунку 2.27, необхідно видалити записи про видання, для якого поле ID\_Видання дорівнює 1. Оператор для видалення:

DELETE FROM видання WHERE ID\_Видання = 5;

Результат виконаної команди зображений на рисунку 2.28.

ID_Видання	Назва_видання	Предмет	Рік_видання	Кількість_сторінок	Ступінь_освіти
1	Машинна графіка ...	Машинн...	2004	32	Бакалавр
2	Концептуальне п...	Комп'ю...	2019	278	Бакалавр
3	Логістичні основи ...	Штучни...	2021	57	Магістр
4	Дослідження інфо...	Штучни...	2018	220	Магістр
5	Інформаційні сист...	Інформ...	2015	33	Бакалавр

Рисунок 2.28 – Результат виконаної команди

## 2.5.2 Організація різних підходів щодо зв'язування таблиць

Зв'язок може поєднувати дві або більше таблиць для того, щоб забезпечити можливість витягувати дані з декількох таблиць. З цією метою використовується оператор JOIN.

З'єднання за рівністю (внутрішнє з'єднання). Найбільш поширений тип з'єднання, при якому таблиці з'єднуються за загальним стовпцем, що є ключовим у кожній таблиці [11].

У списку оператора SELECT разом з іменами кожного із стовпців зазначені імена відповідних таблиць. Це називається докладним визначенням стовпців у запиті. Аналогічний запит можна записати з оператором NATURAL JOIN.

Задача 1. Необхідно виконати з'єднання за рівністю двох таблиць.

Приклад 1. Створені дві таблиці АВТОР, представлена на рисунку 2.29, та ВИДАННЯ, яка представлена на рисунку 2.30.

ID_Автор	Автор	Посада	Електронна_адреса	Кількість_видань
1	Шаповалов Володимир Олександрович	Доцент	#####	5
2	Егоров Олег Йосипович	Доцент	#####	4
3	Устенко Андрей Борисович	Доцент	#####	4
4	Доманська Галина Анатоліївна	Доцент	#####	3
5	Жуковський Ігор Володимирович	Професор	#####	4
6	Косолапов Анатолій Аркадійович	Доцент	#####	6
7	Остапчук Денис Олександрович	Доцент	#####	3
8	Беляев Микола Борисович	Старший викладач	#####	0
9	Заць Олексій Петрович	Старший викладач	#####	0
10	Івн Павло Вікторович	Старший викладач	#####	2
11	Лобода Дмитро Геннадійович	Старший викладач	#####	0
12	Скабалланович Тетяна Іванівна	Доцент	#####	11
13	Пахоньова Вікторія Миколаївна	Доцент	#####	10

Рисунок 2.29 – Таблиця АВТОР

ID_Видання	Назва_видання	Предмет	Рік_видання	Кількість_сторінок	Ступінь_осв
1	Машинна графіка : Метод. вказівки до лабораторних робіт	Машинна графіка	2000	12	Бакалавр
2	Бази даних. Методичні вказівки до виконання лабораторних робіт	Бази даних	2007	30	Бакалавр
3	Методичне указання к лабораторній роботі по курсу "Вычислительная техника в инженерных и экономических..."	Основи програмування	2001	25	Бакалавр
4	Методичне указання к лабораторній роботі по курсу "Вычислительная техника в инженерных и экономических..."	Основи програмування	1999	12	Бакалавр
5	Обчислювальна техніка та програмування. Методичні вказівки до виконання індивідуального завдання «Обчисл...	Основи програмування	1991	18	Магістр
6	Методичне указання к лабораторній роботі по курсу "Вычислительная техника в инженерных и экономических..."	Логіка	2000	20	Бакалавр
7	Теорія інформації та кодування. Розділ: Інформативність повідомлень та їх кодування для зручності зберігання й ...	Теорія інформації	2019	56	Бакалавр
8	Методичні вказівки до лабораторних робіт з основ програмування на Turbo Pascalі. Ч.1	Програмування	1999	23	Бакалавр
9	Інформатика (ОТ та програмування) : Методичні вказівки до виконання лабораторних робіт з основ програм. нов...	Програмування	1998	17	Бакалавр
10	Комп'ютерна техніка та програмування : Методичні вказівки до лабораторних робіт з розділу "Операційна систе...	Комп'ютерна техніка та програмування	1980	45	Бакалавр
11	Обчислювальна техніка та програмування : Методичні вказівки з варіантами вихідних даних до контрольної робо...	Обчислювальна техніка	2005	32	Бакалавр
12	Інформаційні технології в управлінській, науковій та викладацькій діяльності : Методичні рекомендації до виконан...	Бази даних	2009	23	Магістр
13	Логістичні основи штучного інтелекту : Методичні вказівки до виконання лабораторних робіт	Штучний інтелект	2003	65	Магістр

Рисунок 2.30 – Таблиця ВИДАННЯ

Виконуємо з'єднання за допомогою такого синтаксису:

SELECT автор.Автор, видання.Назва\_видання

FROM автор, видання WHERE автор.ID\_Автор = видання.ID\_Видання;

Виконанням цієї команди, витягаємо стовпець Автор з таблиці АВТОР, та стовпець Назва\_видання з таблиці видання. Через те, що в кожній таблиці присутні ID, та вони взаємопов'язані, імена двох стовпців таблиць розглядаються як еквівалентні. Результат виконання зображений на рисунку 2.31.

Автор	Назва_видання
Дзюба Володимир Володимирович	Машинна графіка : Метод. вказівки до лабораторних робіт
Шаповалов Володимир Олександрович	Бази даних. Методичні вказівки до виконання лабораторних робіт
Егоров Олег Йосипович	Методические указания к лабораторным работам по курсу "Вычислительная техника в инженерных и экономическ...
Устенко Андрей Борисович	Методические указания к лабораторным работам по курсу "Основы автоматизации проектирования ЭВМ", Ч. 1"
Донанська Галина Анатоліївна	Обчислювальна техніка та програмування. Методичні вказівки до виконання індивідуального завдання «Обчисл...
Жуковський Ігор Володимирович	Методические указания к лабораторным работам по курсу "Вычислительная техника в инженерных и экономическ...
Косолапов Анатолій Аркадійович	Теорія інформації та кодування. Розділ: Інформативність повідомлень та їх кодування для зручності зберігання й ...
Остапеш Денис Олександрович	Методичні вказівки до лабораторних робіт з основ програмування на Турбо Паскалі. Ч. 1
Беляев Микола Борисович	Информатика (ОТ та програмування) : Методичні вказівки до виконання лабораторних робіт з основ програм. нов...
Заєць Олександр Петрович	Комп'ютерна техніка та програмування : Методичні вказівки до лабораторних робіт з розділу "Операційна систе...
Івн Павло Вікторович	Обчислювальна техніка та програмування : Методичні вказівки з варіантами вихідних даних до контрольної робо...
Лобода Дмитро Геннадійович	Інформаційні технології в управлінській, науковій та викладацькій діяльності : Методичні рекомендації до виконан...
Скабалланович Тетяна Іванівна	Логістичні основи штучного інтелекту : Методичні вказівки до виконання лабораторних робіт

Рисунок 2.31 – Результат виконання команди

Натуральне з'єднання схоже на з'єднання за рівністю, але при натуральному з'єднанні повторення еквівалентних стовпців виключається.

Також це з'єднання можна виконати за допомогою оператора NATURAL JOIN.

Задача 2. Виконати натуральне з'єднання двох таблиць.

Приклад 2. З'єднуємо таблицю АВТОР, яка зображена на рисунку 2.29, та ВИДАННЯ, представлена на рисунку 2.30.

Синтаксис оператора для натурального з'єднання:

SELECT Автор.\*, видання.Назва\_видання

FROM автор, видання WHERE автор.ID\_Автор = видання.ID\_Видання;

Виконанням цієї команди витягаємо всі дані з таблиці АВТОР та стовпець Назва\_видання з таблиці видання. Оскільки стовпці ID взаємопов'язані та містять однакову інформацію, витягається лише стовпець с таблиці АВТОР. Результат виконання зображений на рисунку 2.32.

ID_Автор	Автор	Посада	Електронна_адреса	Кількість_видань	Назва_видання
1	Дзюба Володимир Володимирович	Доцент	#####	1	Машинна графіка : Метод. вказівки до лабораторних робіт
2	Шаповалов Володимир Олександрович	Професор	#####	3	Бази даних. Методичні вказівки до виконання лабораторних робіт
3	Егоров Олег Йосипович	Доцент	#####	2	Методические указания к лабораторным работам по курсу "Вычислительная тех
4	Устенко Андрей Борисович	Доцент	#####	5	Методические указания к лабораторным работам по курсу "Основы автоматизац
5	Донанська Галина Анатоліївна	Доцент	#####	3	Обчислювальна техніка та програмування. Методичні вказівки до виконання інд
6	Жуковський Ігор Володимирович	Професор	#####	4	Методические указания к лабораторным работам по курсу "Вычислительная тех
7	Косолапов Анатолій Аркадійович	Доцент	#####	5	Теорія інформації та кодування. Розділ: Інформативність повідомлень та їх коду
8	Остапеш Денис Олександрович	Доцент	#####	2	Методичні вказівки до лабораторних робіт з основ програмування на Турбо Паск
9	Беляев Микола Борисович	Старший викладач	#####	2	Информатика (ОТ та програмування) : Методичні вказівки до виконання лаборат
10	Заєць Олександр Петрович	Старший викладач	#####	3	Комп'ютерна техніка та програмування : Методичні вказівки до лабораторних р
11	Івн Павло Вікторович	Старший викладач	#####	3	Обчислювальна техніка та програмування : Методичні вказівки з варіантами вих
12	Лобода Дмитро Геннадійович	Старший викладач	#####	2	Інформаційні технології в управлінській, науковій та викладацькій діяльності : Ме
13	Скабалланович Тетяна Іванівна	Доцент	#####	5	Логістичні основи штучного інтелекту : Методичні вказівки до виконання лабора

Рисунок 2.32 – Результат виконання команди

Умове з'єднання також схоже на з'єднання з умовою нерівності. Різниця між ними в тому, що в якості умови може виступати будь-який вираз, що записується після ключового слова ON, а не WHERE [12].

Задача 3. Виконати умове з'єднання двох таблиць.



Приклад 3. З'єднуємо таблицю АВТОР, яка зображена на рисунку 2.29, та ВИДАННЯ, представлена на рисунку 2.30.

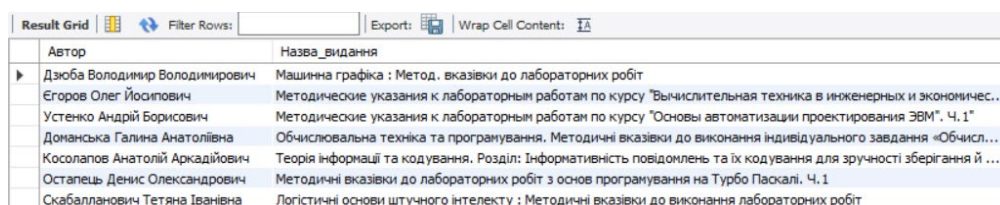
Синтаксис оператора для умовного з'єднання:

SELECT Автор.Автор, видання.Назва\_видання

FROM автор JOIN видання ON (автор.ID\_Автор = видання.ID\_Видання)

AND (Посада = 'Доцент');

Виконанням цієї команди витягуються зазначені дані з двох таблиць та додається умова, щоб дані бралися лише з рядків, де автор є доцентом. Результат виконання зображений на рисунку 2.33.



Автор	Назва_видання
Дзюба Володимир Володимирович	Машинна графіка : Метод. вказівки до лабораторних робіт
Егоров Олег Йосипович	Методические указания к лабораторным работам по курсу "Вычислительная техника в инженерных и экономических..."
Устенко Андрей Борисович	Методические указания к лабораторным работам по курсу "Основы автоматизации проектирования ЭВМ". Ч.1"
Донанська Галина Анатоліївна	Обчислювальна техніка та програмування. Методичні вказівки до виконання індивідуального завдання «Обчисл...
Косолапов Анатолій Аркадійович	Теорія інформації та кодування. Розділ: Інформативність повідомлень та їх кодування для зручності зберігання й ...
Остапеш Денис Олександрович	Методичні вказівки до лабораторних робіт з основ програмування на Турбо Паскалі, Ч.1
Скабалланович Тетяна Іванівна	Логістичні основи штучного інтелекту : Методичні вказівки до виконання лабораторних робіт

Рисунок 2.33 – Результат виконання роботи

З'єднання за нерівністю. При цьому з'єднані дві або більше таблиці поєднуються за умовою нерівності значення стовпця таблиці, значенню зі стовпця іншої таблиці [11].

Задача 4. Виконати з'єднання двох таблиць.

Приклад 4. З'єднуємо таблицю АВТОР, яка зображена на рисунку 2.29, та ВИДАННЯ, представлена на рисунку 2.30.

Синтаксис оператора для з'єднання за нерівністю:

SELECT Автор.Автор, видання.Назва\_видання FROM автор, видання

WHERE автор.ID\_Автор <> видання.ID\_Видання;

Ця команда витягає ім'я автора та назву видання для всіх авторів з невідомими ID. Тут будуть витягнуті 169 рядків, хоча в таблиці всього 13. З цих 169 - 13 записів з таблиці АВТОР, відповідають таблиці ВИДАННЯ. Через таке з'єднання з кожним записом з першої таблиці підставлятимуться всі записи з другої таблиці, за винятком того самого єдиного відповідного запису.

Через великий обсяг результат команди не наводиться.

Використання скорочень або псевдонімів для імен таблиць. Скорочення використовується для перейменування таблиці в рамках оператора SQL, який використовується на даний момент. При цьому ім'я таблиці в самій базі даних не змінюється. Найчастіше псевдоніми використовуються для скорочення загального об'єму набору символів, через що оператори SQL стають коротшими та більш зрозумілими.

Задача 5. Виконати з'єднання двох таблиць.

Приклад 5. З'єднуємо таблицю АВТОР, яка зображена на рисунку 2.29, та ВИДАННЯ, представлена на рисунку 2.30.

Синтаксис оператора для з'єднання з використанням скорочень або псевдонімів:

```
SELECT A.Автор, A.Посада, A.Кількість_видань, B.Назва_видання
FROM автор AS A, видання AS B
WHERE A.ID_Автор = B.ID_Видання AND A.Кількість_видань > 3;
```

Замість імені АВТОР використовується А, а замість ВИДАННЯ використовується В. Скорочення різних таблиць в одному операторі повинні відрізнятися. Результат виконання зображений на рисунку 2.34.

Автор	Посада	Кількість_видань	Назва_видання
Шаповалов Володимир Олександрович	Доцент	5	Методичні вказівки до лабораторних робіт з основ програмування на Турб...
Єгоров Олег Йосипович	Доцент	4	Архітектура комп'ютерів. Розділ: "Арифметико-логічні пристрої": Методи...
Устенко Андрій Борисович	Доцент	4	Методические указания к лабораторным работам по курсу "Основы автом...
Жуковицький Ігор Володимирович	Професор	4	Обчислювальна техніка та програмування : Метод. вказівки та варіанти з...
Косолапов Анатолій Аркадійович	Доцент	6	Системи штучного інтелекту : Методичні вказівки до практичних і лабора...
Пахомова Вікторія Миколаївна	Доцент	10	Дослідження інформаційно-телекомунікаційної системи залізничного транс...

Рисунок 2.34 – Результат виконання команди

Зовнішнє з'єднання використовується тоді, коли результат повинен вміщувати всі записи однієї з таблиць, навіть тоді, коли деякі її записи не мають відповідних записів в іншій таблиці.

ОПЕРАТОР – будь-який оператор порівняння: «=», «<», «>», «<=», «>=», «<>».

LEFT JOIN – невідповідні записи, які є в лівій таблиці, зберігаються в результатній таблиці, а дані, які є в правій, – видаляються.

RIGHT JOIN – невідповідні записи, які є в правій таблиці, зберігаються в результатній таблиці, а дані, які є в лівій, – видаляються.

Задача 6. Виконати з'єднання двох таблиць.

Приклад 6. З'єднуємо таблицю АВТОР, яка зображена на рисунку 2.29, та ВИДАННЯ, представлена на рисунку 2.30.

Синтаксис оператора для з'єднання з використанням LEFT JOIN:

SELECT Автор.Автор, видання.Назва\_видання

FROM автор LEFT JOIN видання ON автор.ID\_Автор = видання.ID\_Видання;

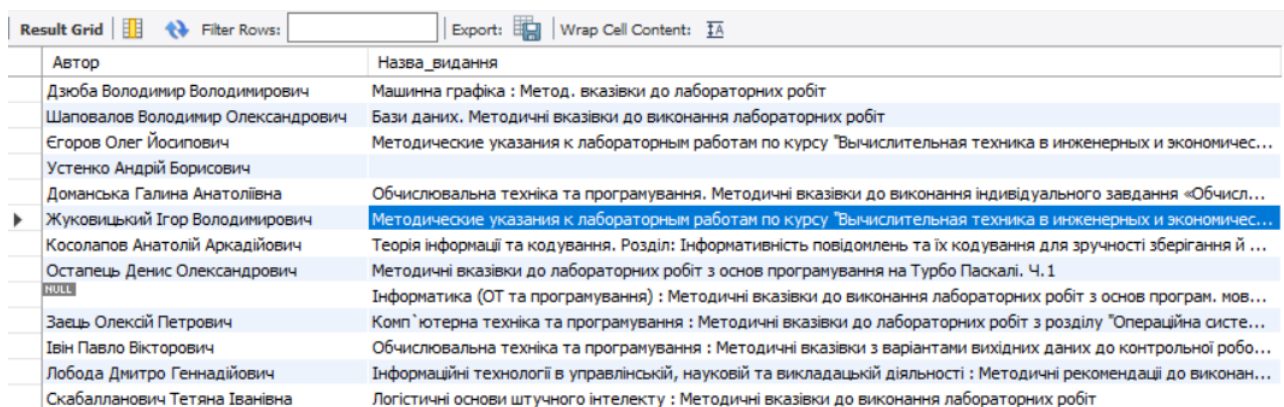
При лівому зовнішньому з'єднанні записи наявні в лівій таблиці, зберігаються в результативній таблиці, а наявні в правій – видаляються. Результат виконання зображений на рисунку 2.35

Синтаксис оператора для з'єднання з використанням RIGHT JOIN:

SELECT Автор.Автор, видання.Назва\_видання

FROM автор RIGHT JOIN видання ON автор.ID\_Автор = видання.ID\_Видання;

При правому зовнішньому з'єднанні записи наявні в правій таблиці, зберігаються в результативній таблиці, а наявні в лівій – видаляються. Результат виконання зображений на рисунку 2.35.



Автор	Назва_видання
Дзюба Володимир Володимирович	Машинна графіка : Метод. вказівки до лабораторних робіт
Шаповалов Володимир Олександрович	Бази даних. Методичні вказівки до виконання лабораторних робіт
Егоров Олег Йосипович	Методические указания к лабораторным работам по курсу "Вычислительная техника в инженерных и экономичес..."
Устенко Андрій Борисович	
Доманська Галина Анатоліївна	Обчислювальна техніка та програмування. Методичні вказівки до виконання індивідуального завдання «Обчисл...
Жуковичский Игорь Владимирович	Методические указания к лабораторным работам по курсу "Вычислительная техника в инженерных и экономичес..."
Косолапов Анатолій Аркадійович	Теорія інформації та кодування. Розділ: Інформативність повідомлень та їх кодування для зручності зберігання й ...
Остапеч Денис Олександрович	Методичні вказівки до лабораторних робіт з основ програмування на Турбо Паскалі. Ч.1
NULL	Інформатика (ОТ та програмування) : Методичні вказівки до виконання лабораторних робіт з основ програм. мов...
Заць Олексій Петрович	Комп'ютерна техніка та програмування : Методичні вказівки до лабораторних робіт з розділу "Операційна систе..."
Івн Павло Вікторович	Обчислювальна техніка та програмування : Методичні вказівки з варіантами вихідних даних до контрольної робо...
Лобода Дмитро Геннадійович	Інформаційні технології в управлінській, науковій та викладацькій діяльності : Методичні рекомендації до виконан...
Скабалланович Тетяна Іванівна	Логістичні основи штучного інтелекту : Методичні вказівки до виконання лабораторних робіт

Рисунок 2.35 – Результат виконання команд

Декартово з'єднання (кросс – з'єднання) – це результат відсутності з'єднання таблиць у запиті. Якщо в запиті вказуються дві або більше таблиць і не використовується їх з'єднання, то в результаті будуть присутні всі можливі комбінації рядків обраних таблиць.

Задача 7. Виконати з'єднання двох таблиць.

Приклад 6. З'єднуємо таблицю АВТОР, яка зображена на рисунку 2.29, та ВИДАННЯ, представлена на рисунку 2.30.

Синтаксис оператора для кросс-з'єднання:

SELECT автор.Автор, видання.Назва\_видання, видання.Рік\_видання FROM автор, видання;

Через те, що не було вказано, як з'єднати таблиці, сервер бази даних з'єднав кожен запис однієї таблиці з кожним записом іншої [11]. Результат виконання зображений на рисунку 2.36.

Автор	Назва_видання	Рік_видання
Скабалланович Тетяна Іванівна	Машинна графіка : Метод. вказівки до лабораторних робіт	2000
Лобода Дмитро Геннадійович	Машинна графіка : Метод. вказівки до лабораторних робіт	2000
Івн Павло Вікторович	Машинна графіка : Метод. вказівки до лабораторних робіт	2000
Заць Олексій Петрович	Машинна графіка : Метод. вказівки до лабораторних робіт	2000
Беляв Микола Борисович	Машинна графіка : Метод. вказівки до лабораторних робіт	2000

Рисунок 2.36 – Результат виконання команди

### 2.5.3 Об'єднання запитів

Декілька простих запитів SQL, які складаються з одного оператора SELECT, можна об'єднати в один за допомогою команд UNION і UNION ALL.

Команда UNION використовується для об'єднання результатів двох або більше операторів SELECT з виключенням рядків, які повторюються.

При використанні UNION потрібно знати: в кожному з операторів SELECT, що з'єднуються, повинно бути обрано однакове число стовпців, та стовпці повинні бути однакового типу і розставлені в тому ж порядку.

Приклад 1. Синтаксис виглядає так:

```
SELECT ID_Видання, Кількість_сторінок FROM видання
WHERE Рік_видання >=2000
UNION
SELECT ID_Видання, Кількість_сторінок FROM видання
WHERE Рік_видання <=2006;
```

Перша команда з таблиці ВИДАННЯ, що вказана на рисунку 2.30, виводить ID тих видань, в яких рік видання починається з 2000 року, а саме 1, 2, 3, 6, 7, 11, 12, 13. А другий запит виводить ID тих видань, які видані до 2006 року, а саме 4, 5, 8, 9, 10. Результат виконання зображений на рисунку 2.37.



Result Grid		Filter Rows:
	ID_Видання	Кількість_сторінок
▶	1	12
	2	30
	3	25
	6	20
	7	56
	11	32
	12	23
	13	65
	4	12
	5	18
	8	23
	9	17
	10	45

Рисунок 2.37 – Результат виконаної команди

Команда UNION ALL використовується для об'єднання результатів двох або більше операторів SELECT з рядками, які повторюються.

В результаті виконання цієї команди, ID\_видання, що виводяться в обох запитах з таблиці ВИДАННЯ, яка вказана на рисунку 2.30, будуть відображатися двічі, тому що UNION ALL допускає до виводу рядки, які повторюються. Результат виконання зображений на рисунку 2.38.

Result Grid		Filter Rows:
	ID_Видання	Кількість_сторінок
▶	1	12
	2	30
	3	25
	6	20
	7	56
	11	32
	12	23
	13	65
	1	12
	3	25
	4	12
	5	18
	6	20
	8	23
	9	17

Рисунок 2.38 – Результат виконаної команди

Використання ORDER BY. Ключове слово ORDER BY можна використовувати тільки для упорядкування результатів остаточного виводу обох команд. Тому допускається тільки одне вираження з ключовим словом ORDER BY, хоча сама команда може складатись з декількох операторів SELECT.

Приклад 3. Синтаксис виглядає так:

```
SELECT ID_Видання, Кількість_сторінок FROM видання
```

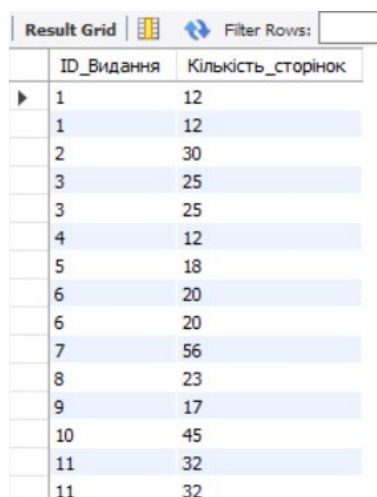
```
WHERE Рік_видання >=2000
```

```
UNION ALL
```

```
SELECT ID_Видання, Кількість_сторінок FROM видання
```

```
WHERE Рік_видання <=2006 ORDER BY 1;
```

В результаті цієї команди витягнуті дані будуть упорядковані за першим стовпцем. За допомогою сортування легко виявити повторні рядки. Результат виконання зображений на рисунку 2.39.



ID_Видання	Кількість_сторінок
1	12
1	12
2	30
3	25
3	25
4	12
5	18
6	20
6	20
7	56
8	23
9	17
10	45
11	32
11	32

Рисунок 2.39 – Результат виконаної команди

Використання GROUP BY. Ключове слово GROUP BY можна використати в будь-якій команді, розмістивши GROUP BY наприкінці. Крім того, разом з GROUP BY можна використовувати ключове слово HAVING.

Приклад 4. Синтаксис виглядає так:

```
SELECT Ступінь_освіти, COUNT(*) AS Результат FROM видання
```

```
WHERE Кількість_сторінок >20 GROUP BY Ступінь_освіти
```

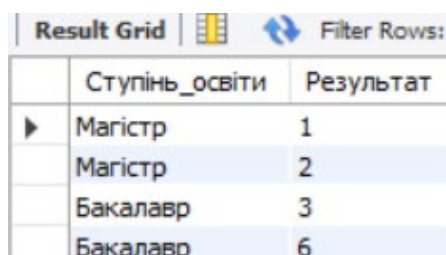
```
UNION ALL
```

```
SELECT Ступінь_освіти, COUNT(*) FROM видання
```

```
WHERE Кількість_сторінок <20 ORDER BY 2;
```

З таблиці ВИДАННЯ, яка вказана на рисунку 2.30, за першим запитом вибираються видання, кількість сторінок яких більше 20. Відібрані записи групуються за стовпцем Ступінь\_освіти. Другий запит виводить видання, кількість сторінок яких менше 20. Відібрані записи групуються за стовпцем Ступінь\_освіти.

Оскільки була використана команда UNION BY, то виводяться всі 4 рядки, включаючи повторювані [11]. Результат виконання зображений на рисунку 2.40.



	Ступінь_освіти	Результат
▶	Магістр	1
	Магістр	2
	Бакалавр	3
	Бакалавр	6

Рисунок 2.40 – Результат виконання команди

## 2.6 Основні висновки

1. Для проєктування бази даних навально-методичних видань на основі даних репозиторію та даних викладачів кафедри ЕОМ, взято відношення «DB\_EMP», що складається з 13 атрибутів та 93 кортежів.

2. Проєктування бази даних виконано за методом нормальних форм, в результаті чого отримано 4 відношення, які співпадають із отриманою структурою спроектованої бази даних за методом «Сутність – зв'язок» іншим здобувачем, а саме Довгою Ганною Романівною, що підтверджує вірність виконання проєктування.

3. Для створення та обробки «DB\_EMP» використовувалась програма MySQL Workbench за допомогою мови MySQL.

4. Проведено зрівняння програм MySQL Workbench та Microsoft Access за наступними ознаками: основні характеристики, інтерфейс та вартість. Виконано співставлення можливостей MySQL Workbench та SQL Microsoft Access за наступними дослідженнями: використання DML, організація різних підходів щодо зв'язування таблиць та об'єднання запитів.

5. Рекомендовано створену базу даних навчально-методичних видань кафедри ЕОМ до використання адміністрацією кафедри. Переведення лабораторних робіт з SQL Microsoft Access на MySQL Workbench не доцільно, але MySQL Workbench може бути рекомендовано до використання здобувачів першого ступеня спеціальності «Комп'ютерна інженерія» при виконанні самостійної роботи з дисципліни «Бази даних».

## ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

1. Найбільше практичне застосування отримала реляційна база даних, в якій для проєктування використовується два методи: нормальних форм та «Сутність-зв'язок». Для проєктування обрано метод нормальних форм, тому що він математичний та являється більш точним.

Для реалізації існує багато програм, наприклад: Altibase, MySQL, Oracle, SQL Azure, PostgreSQL. Для створення бази даних навчально-методичних видань, обрано програму MySQL Workbench, так як, вона має всі необхідні інструменти, повністю безкоштовна та є простою в використанні.

2. Для проєктування бази даних навчально-методичних видань на основі даних репозиторію та даних викладачів кафедри ЕОМ, взято відношення «DB\_EMP», що складається з 13 атрибутів та 93 кортежів. Виконано проєктування бази даних виконано за методом нормальних форм, в результаті чого отримано 4 відношення, які співпадають із отриманою структурою спроектованої бази даних за методом «Сутність – зв'язок» іншим здобувачем, а саме Довгою Ганною Романівною, що підтверджує вірність виконання проєктування.

3. Для створення та обробки «DB\_EMP» використовувалась програма MySQL Workbench за допомогою мови MySQL.

4. Проведено зрівняння програм MySQL Workbench та Microsoft Access за наступними ознаками: основні характеристики, інтерфейс та вартість. Виконано співставлення можливостей MySQL Workbench та SQL Microsoft Access за наступними дослідженнями: використання DML, організація різних підходів щодо зв'язування таблиць та об'єднання запитів.

5. Рекомендовано створену базу даних навчально-методичних видань кафедри ЕОМ до використання адміністрацією кафедри. Переведення лабораторних робіт з SQL Microsoft Access на MySQL Workbench не доцільно, але MySQL Workbench може бути рекомендовано до використання здобувачів першого ступеня спеціальності «Комп'ютерна інженерія» при виконанні самостійної роботи з дисципліни «Бази даних».

## ПЕРЕЛІК ПОСИЛАНЬ

1. Проектування реляційних баз даних [Електронний ресурс]. — Режим доступу: <https://ua-referat.com/>;
2. Нормалізація відношень. Шість нормальних форм [Електронний ресурс]. — Режим доступу: <https://habr.com/ru/post/>;
3. Проектування баз даних [Електронний ресурс]. — Режим доступу: <https://studfile.net/>;
4. Що таке СУБД? [Електронний ресурс]. — Режим доступу: <https://www.nic.ru/>;
5. Офіційний сайт Altibase [Електронний ресурс]. — Режим доступу: <https://www.altibase.com/en/>;
6. Офіційний сайт PostgreSQL [Електронний ресурс]. — Режим доступу: <https://www.postgresql.org>;
7. Офіційний сайт Oracle Database [Електронний ресурс]. — Режим доступу: <https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/introduction-to-oracle-database.html>;
8. Офіційний сайт SQL Azure [Електронний ресурс]. — Режим доступу: <https://docs.microsoft.com/ru-ru/azure/azure-sql/database/sql-database-paas-overview?view=azuresql>;
9. Що таке MySQL? [Електронний ресурс]. — Режим доступу: <https://freehost.com.ua/faq/wiki/chto-takoe-mysql/>;
10. Офіційний сайт MySQL [Електронний ресурс]. — Режим доступу: <https://www.mysql.com>.
11. Пахомова В. М. Методичні вказівки до лабораторних робіт з курсу «Бази даних» (розділ Access. SQL) для студентів спеціальності «Комп'ютерні системи та мережі». Дніпропетровськ: ДНУЗТ. 2002. 48 с.
12. Булатецька Л. В. Мова запитів SQL : текст лекцій нормативної навчальної дисципліни «Бази даних та розподілені інформаційно-аналітичні системи» / Булатецька Леся Віталіївна, Булатецький Віталій Вікторович. – Луцьк

: СНУ імені Лесі Українки, 2018. – 92 [Електронний ресурс]. — Режим доступу: [https://evnuir.vnu.edu.ua/bitstream/123456789/17722/3/SQL\\_2\\_ukr.pdf](https://evnuir.vnu.edu.ua/bitstream/123456789/17722/3/SQL_2_ukr.pdf)

13. Дистанційний курс з дисципліни «Бази даних»; укладач доц. Пахомова В.М. [Електронний ресурс]. — Режим доступу: <https://lider.diit.edu.ua/>

14. DOU Проектор: OurSQL — реплікація баз даних MySQL із використанням Blockchain [Електронний ресурс]. — Режим доступу: <https://dou.ua/>

15. Гайдаржи В.І., Савченко М.М. Засоби візуалізації класів і взаємозв'язків об'єктно-реляційної бази даних Cache. Системи збереження і масового розповсюдження даних. Національний технічний університет України, 2017. с. 56 – 63.

16. 13. Тези Довга Г.Р., Чайка Р.Р., науковий керівник Пахомова В.М. Молода академія 2022. Збірка тез доповідей. Всеукраїнська науково-технічна конференція студентів і молодих учених. Дніпро: УДУНТ, 2022. с. 14