

Міністерство освіти і науки України
Український державний університет науки і технологій

Комп'ютерні технології та системи
(назва факультету)

Електронні обчислювальні машини
(повна назва кафедри)

Пояснювальна записка
до кваліфікаційної роботи

перший (бакалаврський)
(ступінь вищої освіти)

на тему: Проектування бази даних навчально-методичних видань кафедри ЕОМ
за методом «Сутність-зв'язок» та її обробка
за освітньою програмою Комп'ютерна інженерія
зі спеціальності: 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

Виконала: студентка групи: **КС19120**

Керівник:

Нормоконтролер:

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)

Дніпро – 2022 рік

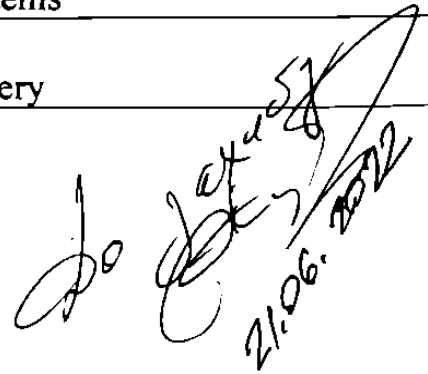
Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies
Computer Technology and Systems

(faculty)

Electronic Computers Machinery

(department)

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


21.06.2022

on the topic: Designing a database of educational and methodical publications of the ECM department by the method of «Essence Relation» and its processing

according to educational curriculum Computer Engineering

in the Speciality: 123 Computer Engineering

(speciality and its code)

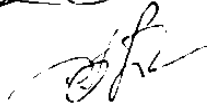
Done by the student of the group: KC19120



Hanna Dovha

(name, surname)

Scientific Supervisor:



Docent, Victoria Pakhomova

(position, name, surname)

Normative controller :



St. Lectures, Volodymyr Dzuba

(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: Електронні обчислювальні машини
Рівень вищої освіти: Перший (бакалаврський)
Освітня програма: Комп'ютерна інженерія
Спеціальність: 123 «Комп'ютерна інженерія»

(шифр та пазла)

ЗАТВЕРДЖУЮ

Завідувач кафедри ЕОМ

Ігор **ЖУКОВИЦЬКИЙ**
(Ім'я ПРІЗВИЩЕ)

Дата _____

З А В Д А Н Н Я

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

студентці Довгій Ганні Романівні

(Прізвище, Ім'я, По батькові)

1. Тема роботи: Проектування бази даних навчально-методичних видань
кафедри ЕОМ за методом «Сутність-зв'язок» та її обробка

Керівник роботи: Пахомова Вікторія Миколаївна

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

затверджені наказом від

"07" 12. 2021 р. № 68 ст

2. Строк подання студентом роботи: 13.06.2022 р.

3. Вихідні дані до роботи: дані про видавництво навчально-методичних видань
що розташовані в репозиторію 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 (БД).

4.2 Основна частина: проектування реляційної бази даних за методом
«Сутність-зв'язок»; створення та обробка реляційної бази даних з
використанням «MySQL»; рекомендації щодо організації індивідуального
завдання з дисципліни «Бази даних» (використання DDL; можливості DQL;
створення вкладених 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)	Закінчено: (підпис, дата)

КАЛЕНДАРНИЙ ПЛАН

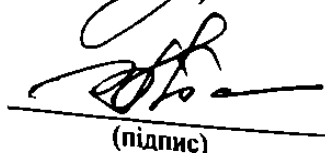
№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Пр.
1	Вступ	01.06.2022 – 06.06.2022	5%
2	Огляд і аналіз предмета проектування	11.04.2022 – 02.05.2022	20%
3	Постановка задачі		10%
4	Проектування реляційної бази даних за методом «сутність- зв'язок»	02.05.2022 – 08.05.2022	20%
5	Створення та обробка реляційної бази даних	08.05.2022 – 20.05.2022	20%
6	Рекомендації щодо організації індивідуального завдання здобувачів з дисципліни «Бази даних» на основі використання MySQL Workbench	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)

Ганна ДОВГА
(Ім'я ПРІЗВИЩЕ)

Керівник роботи


(підпис)

Вікторія ПАХОМОВ
(Ім'я ПРІЗВИЩЕ)

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 предмета проектування	11.04.2022 – 02.05.2022	20%
3	Постановка задачі		10%
4	Проектування реляційної бази даних за методом «сутність- зв'язок»	02.05.2022 – 08.05.2022	20%
5	Створення та обробка реляційної бази даних	08.05.2022 – 20.05.2022	20%
6	Рекомендації щодо організації індивідуального завдання здобувачів з дисципліни «Бази даних» на основі використання MySQL Workbench	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)

Ганна Довга

(Ім'я ПРІЗВИЩЕ)

Керівник роботи

(підпис)

Вікторія Пахомова

(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра:

74с., 33 рис., 2 табл., 2 додатки, 19 джерел.

Об'єкт розробки – база даних навчально-методичних видань кафедри електронних обчислювальних машин.

Мета роботи – проектування бази даних навчально-методичних видань кафедри електронних обчислювальних машин за методом «Сутність-зв'язок» та її обробка.

Методи дослідження – порівняння характеристик додатків MySQL Workbench з Microsoft Access, співставлення можливостей MySQL Workbench з можливостями SQL Microsoft Access за наступними напрямками: використання DDL, використання DQL, а також створення вкладених запитів.

Створено в MySQL Workbench спроектовану за методом «Сутність-зв'язок» базу даних навчально-методичних видань кафедри ЕОМ на основі даних репозиторію УДУНТ та даних викладацького складу кафедри. Проведено зрівняння програм MySQL Workbench з Microsoft Access за наступними ознаками: загальні характеристики, інтерфейс та вартість. Виконано зрівняння можливостей MySQL Workbench та SQL Microsoft Access за напрямками дослідження: використання DDL та DQL, створення вкладених запитів.

Рекомендовано створену базу даних навчально-методичних видань до використання на сайті кафедри ЕОМ. Переведення лабораторних робіт з SQL Microsoft Access на MySQL Workbench недоцільно, але MySQL Workbench може бути рекомендований до використання здобувачами першого (бакалаврського) ступеня спеціальності «Комп'ютерна інженерія» при виконанні індивідуального завдання з дисципліни «Бази даних».

Ключові слова: БАЗА ДАНИХ, СУТНІСТЬ-ЗВ'ЯЗОК, WORKBENCH, MYSQL, НАВЧАЛЬНО-МЕТОДИЧНІ ВИДАННЯ, САЙТ КАФЕДРИ

ЗМІСТ

ВСТУП	6
1 ОГЛЯД І АНАЛІЗ ПРЕДМЕТА ПРОЄКТУВАННЯ	8
1.1 Методи проєктування реляційної бази даних.....	8
1.2 Засоби програмної реалізації.....	13
1.3 Основні висновки	17
2 ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА ОБРОБКА РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ.....	18
2.1 Постановка задачі	18
2.2 Проєктування реляційної бази даних за методом «Сутність- зв'язок».....	18
2.3 Створення та обробка реляційної бази даних в MySQL	22
2.4 Співставлення характеристик MySQL Workbench з SQL Microsoft Access.....	27
2.5 Рекомендації щодо організації індивідуального завдання здобувачів з дисципліни «Бази даних» на основі використання MySQL Workbench.....	29
2.5.1 Використання DDL	29
2.5.2 Можливості DQL	33
2.5.3 Створення вкладених запитів	37
2.6 Основні висновки.....	40
ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ.....	41
ПЕРЕЛІК ПОСИЛАНЬ	42
ДОДАТОК А.....	Помилка! Закладку не визначено.
Початкове, проміжні та кінцеве відношення (фрагмент).....	Помилка! Закладку не визначено.
ДОДАТОК Б	Помилка! Закладку не визначено.
Використання бази даних навчально-методичних видань на сайті кафедри ЕОМ.....	Помилка! Закладку не визначено.
ДОДАТОК В	Помилка! Закладку не визначено.
Тези доповіді	Помилка! Закладку не визначено.

ВСТУП

Найбільш сучасним способом збору та пошуку інформації в комп'ютерних системах є бази даних, які дозволяють зберігати великі обсяги інформації й обробляти запити з високою швидкістю пошуку та оновлення. В даний час робота з базами даних використовується майже у всіх сферах нашого життя, наприклад, у навчальних закладах, банках, магазинах, заводах, державних установах тощо, і є необхідною практично на кожному етапі роботи з інформацією, що підтверджує актуальність теми.

Метою кваліфікаційної роботи є проєктування бази даних навчально-методичних видань кафедри «Електронних обчислювальних машин» за методом «Сутність-зв'язок» та її обробка. Відповідно до мети поставлені наступні завдання:

1. виконати огляд і аналіз предмета проєктування;
2. виконати проєктування бази даних навчально-методичних видань кафедри електронних обчислювальних машин за методом «Сутність-зв'язок»;
3. створити та виконати обробку бази даних за допомогою додатку MySQL Workbench;
4. співставити можливості MySQL Workbench з можливостями SQL Microsoft Access за наступними напрямками: використання DDL, використання DQL, а також створення вкладених запитів.

Найпоширенішим типом створення баз даних є реляційний тип. Бази даних, створені за цим типом, дозволяють зберігати інформацію в реляційних таблицях з певними стовпцями певного типу. Для створення та редагування таких баз даних використовують системи управління базами даних, наприклад: MySQL, Microsoft Access, Oracle, Altibase, SQL Azure, PostgreSQL та інші. Використання реляційних баз даних запропоновано доктором Коддом із компанії IBM.

На сьогоднішній день за допомогою реляційних баз даних створено багато інтернет-сайтів, наприклад: Comfy, Цитрус, Фокстрот, Алло, Eva та інші.

У 2019 році Гарбера І.В. у статті «Лінгвістична база даних “концепт людина у фразеології східностепових українських говірок”: структура та функції» розглянула питання щодо окреслення досвіду використання лінгвістичної бази даних у фразеології у додатку Microsoft Access методом нормалізації відношень.

У 2020 році Бідолак Д.І., Кузьович та В.С. Лакида П.І. у Науковому віснику НЛТУ України опублікували свою просторову базу даних урболандшафтів на прикладі зелених насаджень міста Бережани. Спроектували базу даних за допомогою методу «Сутність-зв'язок» у Microsoft Access.

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

У першому розділі виконано огляд і аналіз предмета проектування, розглянуто методи проектування бази даних. Усього їх два – метод нормальних форм та метод «Сутність-зв'язок», розглянуто основні правила щодо побудови бази даних цими методами. Також наведені приклади використання різних систем управління базами даних для побудови бази даних.

У другому розділі сформульовано постановку задачі та проведено проектування бази даних навчально-методичних видань кафедри електронних обчислювальних машин за методом «Сутність-зв'язок». Створено і оброблено базу даних у додатку MySQL Workbench за допомогою MySQL. Проведено порівняння характеристик MySQL Workbench з Microsoft Access, а також співставлення можливостей MySQL Workbench з можливостями SQL Microsoft Access за наступними напрямками: використання операторів DDL та DQL, створення вкладених запитів. Надано рекомендацію щодо організації індивідуального завдання здобувачів першого ступеня (бакалавр) з дисципліни «Бази даних». Рекомендовано створену базу даних до використання на сайті кафедри ЕОМ.

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 ПРЕДМЕТА ПРОЄКТУВАННЯ

1.1 Методи проєктування реляційної бази даних

В першу чергу розглянемо, що собою представляє реляційна база даних. Реляційна база – це база даних, яка заснована на реляційній моделі даних і призначена для збереження інформації у вигляді взаємозв'язаних таблиць. Пошук здійснюється в одній таблиці на основі зазначених ключових полів іншої таблиці [1].

Існує два методи побудови бази даних: метод нормальних форм та метод «Сутність-зв'язок». Для того, щоб простіше побачити взаємозв'язок сутностей, які зберігаються в базі даних використовують різні методи схем моделей даних. Одним із найбільш розповсюджених є метод ER-діаграм. Історично розробку ER-діаграм приписують Пітеру Чену, який почав використовувати її для проєктування баз даних у 70-х роках минулого століття. ER-діаграма – це подання бази даних у вигляді наглядних графічних діаграм. Вона розшифровується як Essence Relation Diagram, тобто діаграма «Сутність-зв'язок», що призначена для наглядного представлення відношень між об'єктами. Для великих баз даних побудова ER-діаграм дозволяє уникнути помилок проєктування, які важко виправити, особливо якщо проєкт тестується або вже використовується. Помилки при створенні структури бази даних можуть призвести до повної переробки коду проєкту, тому витрачені ресурси будуть використані неефективно. ER-діаграми складаються з сутностей, зв'язків та атрибутів, проаналізуймо та пояснімо ці поняття.

Сутності – це об'єкти такі, як люди, події чи поняття.

Тип сутності — набір однорідних особистостей, які діють, як єдине ціле.

У методі «Сутність-зв'язок» є два типи сутностей: слабкий тип (цей тип сутності покладається на сильну сутність), сильний тип (самостійний тип сутності, який ні від кого не залежить). Кожен тип має певний набір атрибутів. Атрибути призначені для опису певної сутності.

Розрізняють такі види атрибутів:

- прості атрибути (це такі атрибути, які можуть бути частиною складених атрибутів і складаються з одного компонента);
- складені атрибути (це такі атрибути, які складаються з декількох простих атрибутів);
- однозначні атрибути (це такі атрибути, які мають лише одне конкретне значення для певної сутності);
- багатозначні атрибути (ці атрибути можуть мати декілька значень);
- довільні атрибути (це такі атрибути, значення яких формується на основі значень інших атрибутів) [2].

На ER-діаграмі елементи даних розташовані в прямокутниках, атрибути даних – в еліпсах, а відношення між елементами – в ромбах. Якщо атрибут є первинним ключем, то його назву підкреслюють. Відношення між об'єктами, зазвичай, включає в себе принаймні одне дієслово. Потрібно пам'ятати, що сутності і відношення завжди пишуться великими літерами, як подано на рисунку 1.1.

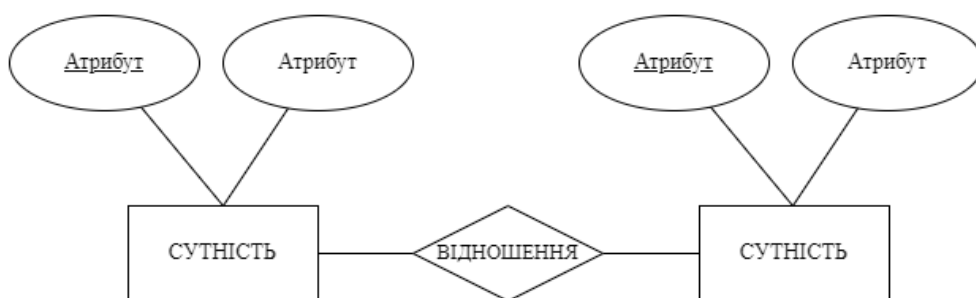


Рисунок 1.1 – ER-діаграма

Залежність між сутностями визначає зв'язок між ними.

Зв'язки поділяються на чотири типи:

- один-до-одного (1:1) – одна сутність із набору сутностей X може бути пов'язана з однією сутністю набору сутностей Y і навпаки;
- один-до-багатьох (1:Б) – одна сутність із набору сутностей X може бути пов'язана з декількома сутностями набору сутностей Y, але сутність із набору сутностей Y може бути пов'язана принаймні з однією сутністю;

– багато-до-одного (Б:1) – більше однієї сутності з набору сутностей X можуть бути пов'язані з однією сутністю набору сутностей Y, однак сутність з набору сутностей Y може або не може бути пов'язана з більш, ніж однією сутністю з набору сутностей X;

– багато-до-багатьох (Б: Б) – одна сутність з набору сутностей X може бути пов'язана з більш, ніж однією сутністю з набору сутностей Y і навпаки [3].

В зв'язках сутностей вводиться таке поняття, як клас приналежності. Він може бути обов'язковим (О), якщо кожен екземпляр сутності повинен брати участь у зв'язку, та необов'язковим (Н), якщо не всі екземпляри сутності повинні брати участь у зв'язку. Зв'язок може бути обов'язковим з одного боку та необов'язковим з іншого. Також для коректної побудови відношень за методом «Сутність-зв'язок» потрібно визначити правила формування відношень. Усього їх шість:

– Перше. Якщо застосовується ступінь зв'язку «один-до-одного» і обов'язкові обидва класи сутностей, формується одне відношення, первинним ключем якого є один із ключів сутностей.

– Друге. Якщо застосовується ступінь зв'язку «один-до-одного» і обов'язковий лише один клас приналежності сутності, формується два відношення, по одному для кожної сутності. Ключ сутності з необов'язковим класом додається у якості неключового атрибуту у відношення, сутність якого має обов'язковий клас.

– Третє. Якщо застосовується ступінь зв'язку «один-до-одного» і необов'язкові обидва класи приналежності екземплярів сутностей, в такому випадку формується три відношення, по одному на кожну сутність, первинними ключами яких є зазначені ключі відповідної сутності. Третє відношення є зв'язним і його ключами є ключі сутностей, що зв'язуються.

– Четверте. Якщо застосовується ступінь зв'язку «один-до-багатьох» чи «багато-до-одного» і сутність зі степенем зв'язку «багато» має обов'язковий клас приналежності екземплярів, в такому випадку формується два відношення, по

одному на кожну сутність, ключ сутності зі степенем зв'язку «один» додається у якості зовнішнього ключа у відношення зі степенем зв'язку «багато».

– П'яте. Якщо застосовується ступінь зв'язку «один-до-багатьох» чи «багато-до-одного» і сутність зі степенем зв'язку «багато» має необов'язковий клас приналежності екземплярів, у такому випадку формується три відношення, по одному на кожну сутність, первинними ключами яких є зазначені ключі відповідної сутності. Третє відношення є зв'язним і його ключами є ключі сутностей, що зв'язуються.

– Шосте. Якщо застосовується ступінь зв'язку «багато-до-багатьох», у такому випадку завжди формується три відношення, по одному на кожну сутність, первинними ключами яких є зазначені ключі відповідної сутності. Третє відношення є зв'язним і його ключами є ключі сутностей, що зв'язуються.

У загальному випадку між двома сутностями може бути задано довільну кількість зв'язків [4].

Бази даних, побудовані методом ER-діаграм, мають безліч переваг. Вони є легкими у розумінні, бо графічне зображення ефективно використовується для візуального представлення макета та дозволяє зрозуміти відносини між різними сутностями [1]. Внаслідок застосування методу «Сутність-зв'язок» полегшується спілкування між замовниками, розробниками і користувачами незалежно від їх професіоналізму в предметній області. Тому саме цей метод обраний для проєктування реляційної бази даних.

ER-діаграми можуть застосовуватись в різних галузях таких, як опис різних відносин і операцій всередині організації. Яскравим прикладом використання діаграм ER-типу є проєкт побудови інтернет-магазину, що представлений на рисунку 1.3. На сьогоднішній день найпопулярнішими є такі інтернет-магазини, як: Rozetka.com.ua, Allo.ua, Comfy.ua, Цитрус.ua, Epicentrk.ua, Foxtrot.com.ua, Kasta.ua, Eva.ua, Eldorado.ua, Moyo.ua та інші. На рисунку 1.2 показані найбільш популярні інтернет-магазини за 2021-2022 роки.

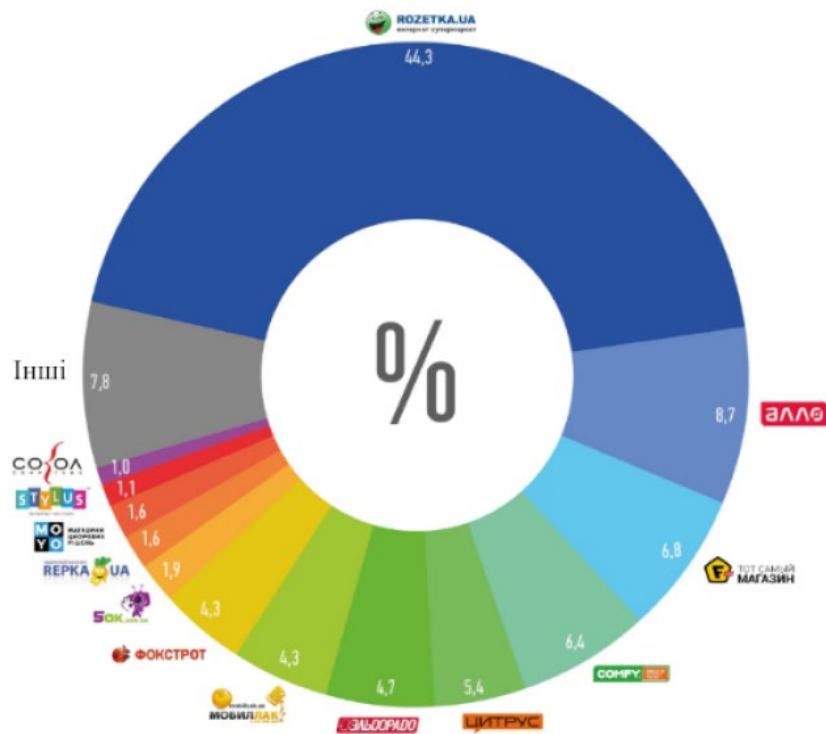


Рисунок 1.2 – Популярні інтернет-магазини

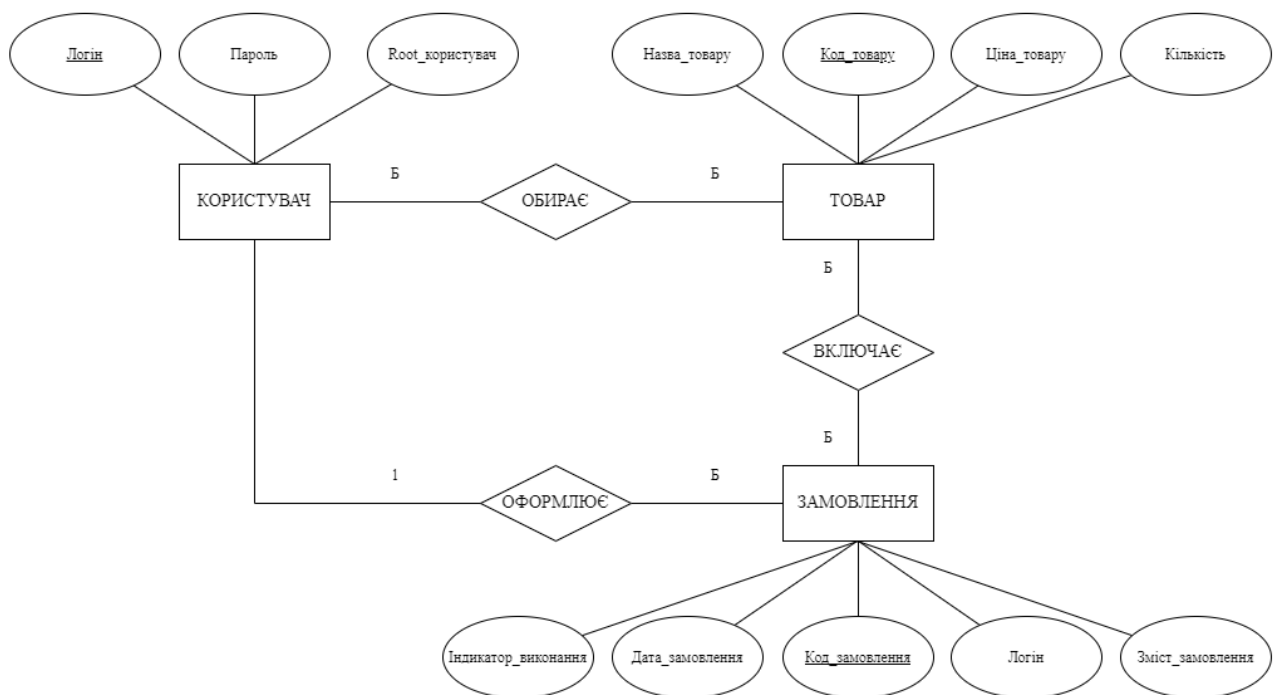


Рисунок 1.3 – Приклад ER-діаграми інтернет магазину

Після закінчення побудови ER-діаграми дані в реляційній моделі потрібно нормалізувати. Нормалізація – це другий метод проєктування реляційної бази даних, який полягає у тому, щоб згрупувати атрибути даних в таблиці, позбувшись дублювання, а таблиці згрупувати в базу даних.

Для того, щоб виконати нормалізацію моделі бази даних, необхідно представити її у вигляді сутності з набором певних атрибутів, які будуть характеризувати і описувати елемент предметної області. Процес проектування бази даних з використанням метода нормальних форм є ітераційним і полягає в послідовному переведенні відношень з першої нормальної форми в нормальні форми більш високого ступеня за визначеними правилами.

Існують такі види нормалізації:

- перша нормальна форма (кожен атрибут має бути атомарним, тобто мати лише одне значення в кожному кортежі відношення);
- друга нормальна форма (відношення повинно знаходитись в першій нормальній формі і кожен неключовий атрибут відношення функціонально повно має залежати від первинного (складеного) ключа);
- третя нормальна форма (відношення вже повинно знаходитись у другій нормальній формі і кожний неключовий атрибут нетранзитивно має залежати від первинного ключа. Атрибут N залежить від атрибуту M транзитивно, якщо для атрибутів N, M, K виконується:

$$M \rightarrow K \text{ і } K \rightarrow N, \quad (1.1)$$

а обернена залежність відсутня);

- посилена третя нормальна форма або нормальна форма Бойса-Кодда (відношення вже повинно знаходитись у третій нормальній формі, і у відношенні мають бути відсутні будь-які залежності первинного ключа, а також ключових від неключових атрибутів) [5].

1.2 Засоби програмної реалізації

Система управління базами даних – це сукупність мовних і програмних засобів, призначення яких полягає у створенні, введенні і використанні інформації бази даних багатьма користувачами. На сьогоднішній день найбільшою популярністю користуються СУБД, спроектовані на основі реляційної моделі даних. Вони мають назву реляційні СУБД. За ступенем універсальності розрізняють два класи СУБД – системи загального призначення і спеціалізовані системи. Перші, системи загального призначення, мають вигляд

комерційного виробу. Вони реалізуються як програмний продукт і слугують для виконання сукупності функцій, пов'язаних із створенням та експлуатацією бази даних. Такі системи управління володіють засобами налаштування на роботу з конкретною базою даних. Використання систем загального призначення дозволяє суттєво знизити час розробки та зекономити трудові ресурси. Спеціалізовані СУБД створюються у рідких випадках при неможливості чи недоцільності використання СУБД загального призначення. На сьогоднішній день ринок програмного забезпечення має велику кількість багатосторонніх систем управління базою даних загального призначення практично для всіх моделей ЕОМ і для різних операційних систем. Сучасні системи управління базою даних володіють засобами забезпечення цілісності даних і надійної безпеки, що надають користувачеві можливість гарантувати безпеку даних [6].

Розглянемо основні характеристики деяких СУБД – лідерів на ринку програм. До складу групи таких програмних продуктів увійшли: MySQL, Altibase, Oracle, SQL Azure, PostgreSQL. Опишемо їх детальніше.

1. MySQL працює на платформах Windows, Linux та Mac з такими мовами, як: SQL та C, C++, Java, Perl, PHP, Python та Tcl. MySQL – це високопродуктивна реляційна система управління базами даних, організована у вигляді взаємопов'язаних таблиць. Ця система управління базами даних з відкритим кодом має різні версії – Enterprise Edition, Standard Edition та Classic Edition. Компанія виробник пропонує різні функції для кожного з них [7]. З самого початку СУБД була дуже схожою на mSQL, бо MySQL виникла, як спроба застосувати mSQL до власних розробок компанії. Звідки походить назва «MySQL» — невідомо, є лише два припущення від розробників. Перше – всі напрацювання компанії починалися з префіксу «Му», а друге – на честь доньки Майкла Монті Віденіуса, одного з розробників системи, який назвав дитину ім'ям Му. MySQL розробила шведська компанія у 1984 році. 2008-го систему управління купила американська компанія Sun Microsystems. А 2010-го її викупив виробничий гігант Oracle. Логотипом програми є дельфін на ім'я «Sakila». Ім'я обрано з великого списку запропонованих користувачами.

Після того, як компанія Sun Microsystems стала частиною компанії Oracle Corporation, MySQL перейшла у власність Oracle. На сьогоднішній день MySQL — компактний багатопотоковий сервер баз даних, який характеризується високою швидкістю, стійкістю і простотою використання.

Серед можливостей і переваг програми слід виділити такі, як:

- простота у використанні;
- підтримка необмеженої кількості користувачів, що одночасно працюють з базою даних;
- кількість рядків у таблицях може досягати 50 мільйонів;
- висока швидкість виконання команд;
- наявність простої та ефективної системи безпеки.

За рахунок своєї простоти у використанні, продуктивності та надійної системи безпеки такі застосунки, як Google, YouTube, Twitter, Facebook та Yahoo, використовують MySQL для зберігання своїх даних [8].

2. Altibase працює на платформі Linux і підтримує такі мови, як C, C ++, PHP та усі мови, що використовуються ODBC або JDBC. Altibase – це система управління базами даних у пам'яті з швидкою та простою у використанні архітектурою. Це особливо корисно для випадків, що вимагають високої пропускної здатності та дуже низької затримки. Altibase пропонує високу продуктивність, відмовостійкість та простоту управління транзакціями, особливо в галузі зв'язку, онлайн-банкінгу, торгівлі цінними паперами, додатків реального часу та вбудованих систем [9]. Компанія Altibase першою у світі розробила та випустила на ринок гібридну СУБД, що дозволяє багатьом клієнтам використовувати оперативну пам'ять та диски, як носії інформації. У 2018 році вихідний код Altibase став відкритим майже для усіх країн, окрім Китаю [10].

3. Oracle працює на платформах Windows та Linux. Підтримує такі мови, як: C, C ++, Java, COBOL, PL / SQL та Visual Basic. Компанія спеціалізується на випуску систем управління базами даних, проміжного програмного забезпечення та бізнес-додатків. Найвідоміший продукт компанії – Oracle Database, який

випускається з моменту заснування Oracle, може використовуватися малими, середніми чи великими організаціями [11].

4. SQL Azure працює на платформах Windows та Linux. Підтримує такі мови, як: C ++, Python, Ruby, Java, PHP, Visual Basic, Delphi, Go та R. SQL-сервер може використовуватись у будь-яких корпораціях, від малих до великих організацій. Система управління базами даних SQL Azure – це повністю керований механізм, який автоматизує більшість функцій управління базою даних таких, як: оновлення, здійснення виправлень, резервне копіювання та моніторинг. Microsoft обробляє весь код SQL, виправлення та оновлення операційної системи, тому користувачеві не потрібно самостійно керувати базовою інфраструктурою [12]. Microsoft Azure SQL засновані на Microsoft SQL Server, але надають лише підмножину типів даних. У цій системі управління базами даних підтримуються основні типи: точні числа, рядки символів (включаючи Unicode), дата та час, просторові, двійкові та інші типи даних [13].

5. PostgreSQL – це високопродуктивна система об'єктно-реляційних баз даних з відкритим вихідним кодом, яка використовує та розширює мову SQL у поєднанні з багатьма функціями для безпечного зберігання та масштабування найскладніших робочих навантажень даних. PostgreSQL має більш, ніж 30 років активного розвитку на базовій платформі. Компанія заробила міцну репутацію завдяки перевірній архітектурі, надійності, цілісності даних, солідному набору функцій, розширюваності та прихильності спільноти відкритого вихідного коду, що стоїть за програмним забезпеченням, для послідовного надання продуктивних та інноваційних рішень. PostgreSQL працює у всіх основних операційних системах, сумісних з ACID і має потужні надбудови такі, як популярний розширювач геопросторових баз даних PostGIS [14].

1.3 Основні висновки

1. Для побудови реляційної бази даних використовують два методи – метод «Сутність-зв'язок» та метод нормальних форм. Для подальшого використання у кваліфікаційній роботі обрано метод «Сутність-зв'язок» – графічний метод, що візуалізує майбутню реляційну базу даних та дозволяє уникнути помилок у проєктуванні, які потім важко виправити.

2. На сьогоднішній день існують такі додатки для побудови бази даних: MySQL Workbench, Altibase, Oracle, SQL Azure, PostgreSQL, Microsoft Access та інші. Для розробки проєкту обрано додаток MySQL Workbench. Він має зручний, зрозумілий користувачеві інтерфейс, дозволяє швидко обробляти команди, має ефективну систему безпеки та не потребує купівлі ліцензії, тобто є безкоштовним.

2 ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА ОБРОБКА РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ

2.1 Постановка задачі

У даній кваліфікаційній роботі необхідно створити реляційну базу даних навчально-методичних видань кафедри електронних обчислювальних машин за методом «Сутність-зв'язок». Вихідними даними для побудови бази даних є інформація про наукові методичні видання, що знаходяться у репозиторії, та педагогічний склад кафедри електронних обчислювальних машин Українського державного університету науки і технологій.

2.2 Проєктування реляційної бази даних за методом «Сутність-зв'язок»

База даних навчально-методичних видань має назву Database educational methodical publication (DB_EMP). Атрибути, які входять до бази даних, представлені в таблиці 2.1.

Таблиця 2.1 – Атрибути DB_EMP

Назва	Позначення
ПІБ	прізвище, ім'я, по батькові автора, викладача кафедри електронних обчислювальних машин
Науковий ступінь	науковий ступінь автора
Посада	назва посади, яку займає автор
Електронна адреса	електронна адреса, яку використовує автор для роботи
Мобільний телефон	номер мобільного телефону, який використовує автор
Спеціальність	номер та назва спеціальності
Предмет	назва предмету методичних видань
Назва видання	назва методичного видання
Ступінь освіти	освітній ступень, для якого розроблено видання
Вид видання	вид методичного видання
Видавництво	видавництво, що видало методичне видання
Рік видання	рік публікації видання
Кількість сторінок	кількість сторінок методичного видання

Спочатку складемо вихідне відношення, в якому 13 атрибутів та 93 кортежів. Початкове відношення представлено в таблиці А.1. У відношенні первинний ключ складений – ПІБ, Назва_видання та Спеціальність. На першому етапі проєктування потрібно виявити сутності і їх зв'язки, ключі сутностей підкреслити.

Виділимо такі сутності: АВТОР (ПІБ); ПОСАДА (Посада); МЕТОДИЧНЕ_ВИДАННЯ (Назва_видання); ВИДАВНИЦТВО (Видавництво, Вид_видання).

Виділимо такі зв'язки між сутностями:

- АВТОР ЗАЙМАЄ ПОСАДУ;
- АВТОР ПУБЛІКУЄ МЕТОДИЧНЕ_ВИДАННЯ;
- ВИДАННЯ МАЄ ВИГЛЯД.

Після закінчення першого етапу можна перейти до створення діаграми.

На рисунку 2.1 представлена діаграма ER-типу для відношення «DB_EMP».

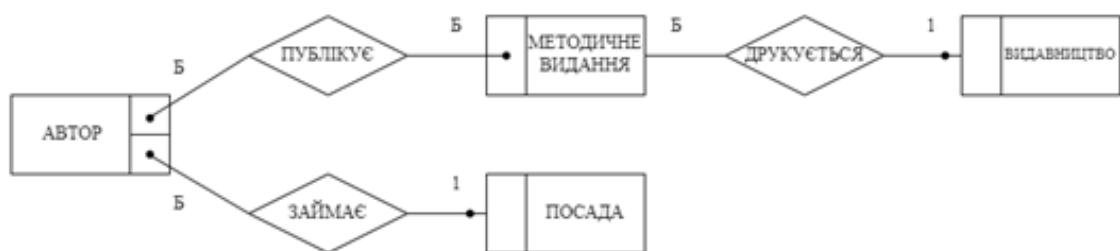


Рисунок 2.1 - діаграма ER-типу для відношення «DB_EMP»

Розглянемо зв'язки між сутностями та виявимо їх ступінь.

1. АВТОР ЗАЙМАЄ ПОСАДУ.

Ступень зв'язку ЗАЙМАЄ «багато-до-одного», тому що кожен автор займає певну посаду і одну й ту ж саму посаду можуть займати декілька авторів. Сутність АВТОР має обов'язковий клас приналежності, бо кожен автор займає посаду. Сутність ПОСАДА має необов'язковий клас приналежності, так як існують посади, які не займає жоден автор.

2. АВТОР ПУБЛІКУЄ МЕТОДИЧНЕ_ВИДАННЯ.

Ступінь зв'язку ПУБЛІКУЄ «багато-до-багатьох», так як автор може публікувати декілька методичних видань і одне методичне видання може бути

опубліковано декількома авторами. В даному випадку обидві сутності мають обов'язковий клас приналежності, бо кожен автор має методичне видання і існують видання опубліковані декількома авторами.

3. МЕТОДИЧНЕ_ВИДАННЯ ДРУКУЄТЬСЯ ВИДАВНИЦТВОМ

Ступень зв'язку ДРУКУЄТЬСЯ «багато-до-одного», тому що одне методичне видання може бути видано одним видавництвом і одне й те саме видавництво може видавати декілька методичних видань. Сутність МЕТОДИЧНЕ_ВИДАННЯ має обов'язковий клас приналежності, бо кожне видання видано видавництвом. Сутність ВИДАВНИЦТВО має необов'язковий клас приналежності, так як існують такі видавництва, які не видали жодного методичного видання.

Коли роботу з попередніми двома етапами закінчено, переходимо до третього – формування попередніх відношень за правилами №1-6.

Зв'язок АВТОР ЗАЙМАЄ ПОСАДУ задовольняє умовам правила №4, відповідно до якого отримуємо два відношення:

- АВТОР (ПІБ, Посада, ...);
- ПОСАДА (Посада, ...),

як зображено на рисунку 2.2.

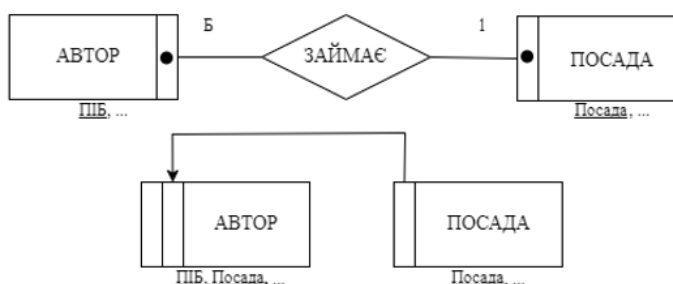


Рисунок 2.2 – Графічне зображення правила №4

Зв'язок АВТОР ПУБЛІКУЄ МЕТОДИЧНЕ_ВИДАННЯ задовольняє умовам правила №6, відповідно до якого отримаємо три відношення:

- АВТОР (ПІБ, ...);
- МЕТОДИЧНЕ_ВИДАННЯ (Назва_видання, ...);
- ПУБЛІКУЄ (ПІБ, Назва_видання, ...),

як показано на рисунку 2.3.

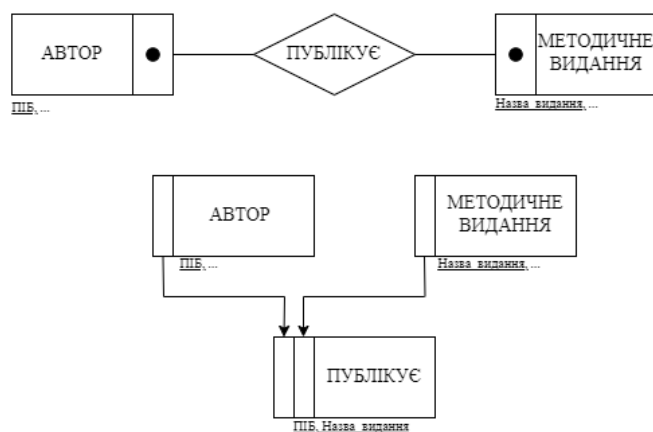


Рисунок 2.3 – Графічне зображення правила №6

Зв'язок МЕТОДИЧНЕ_ВИДАННЯ МАЄ ВИГЛЯД задовольняє умовам правила №4, відповідно до якого отримаємо два відношення:

- МЕТОДИЧНЕ_ВИДАННЯ (Назва видання, Вид видання, ...);
- ВИДАВНИЦТВО (Видавництво, Вид видання),

як зображено на рисунку 2.4.

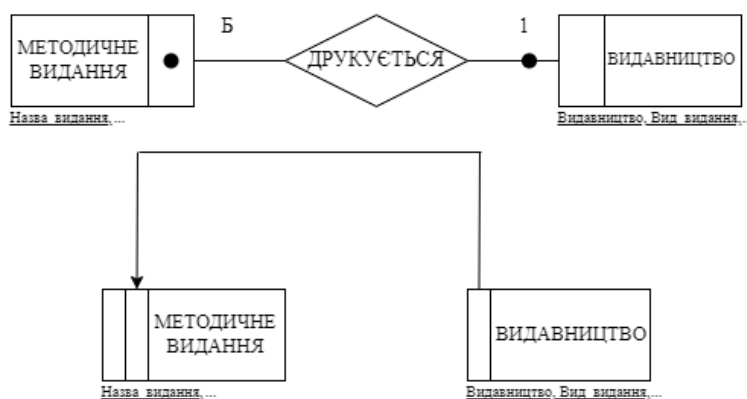


Рисунок 2.4 – Графічне зображення правила №4

Тепер необхідно додати неключові атрибути до відношень з тією умовою, щоб вони відповідали вимогам нормальної форми Бойса-Кодда, – це четвертий етап. Отримані в результаті четвертого етапу відношення мають вигляд:

- АВТОР (ПІБ, Посада, Електронна_адреса, Мобільний телефон);
- ПОСАДА (Посада, Науковий_ступінь);
- МЕТОДИЧНІ_ВИДАННЯ (Назва_видання, Предмет, Вид_видання, Форма_представлення, Рік_видання, Кількість_сторінок);
- ПУБЛІКУЄ (ПІБ, Назва_видання, Спеціальність, Ступінь_освіти).
- ВИДАВНИЦТВО (Видавництво, Вид_видання).

Проміжне відношення АВТОР представлено в таблиці А.2.

Проміжне відношення ПОСАДА представлено в таблиці А.3.

Проміжне відношення ПУБЛІКУЄ представлено в таблиці А.4.

Кінцеве відношення МЕТОДИЧНІ_ВИДАННЯ представлено в таблиці А5.

Дані відношення відповідають вимогам БКНФ і в результаті отримано такі самі відношення, як і в проєктуванні методом нормальних форм іншим здобувачем, Чайкою Романом Руслановичем.

Структура бази даних за методом «Сутність-зв'язок» показана на рисунку 2.5.

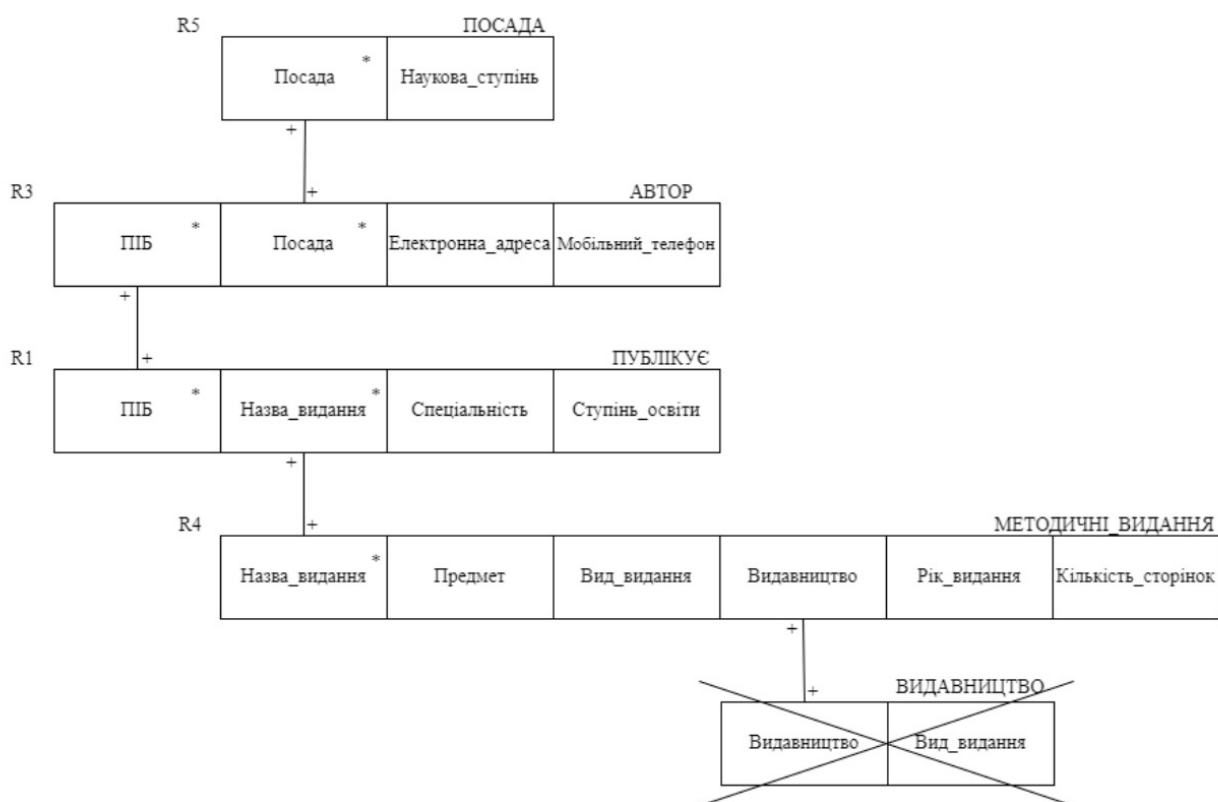


Рисунок 2.5 – Структура проєкту за методом «Сутність-зв'язок»

2.3 Створення та обробка реляційної бази даних в MySQL

MySQL Workbench – універсальний інструмент для роботи з базами даних та управління ними. З його допомогою можна створювати, редагувати таблиці та інші об'єкти, керувати доступом користувачів та повноцінно керувати базою даних. Початковий екран програми представлений на рисунку 2.6.

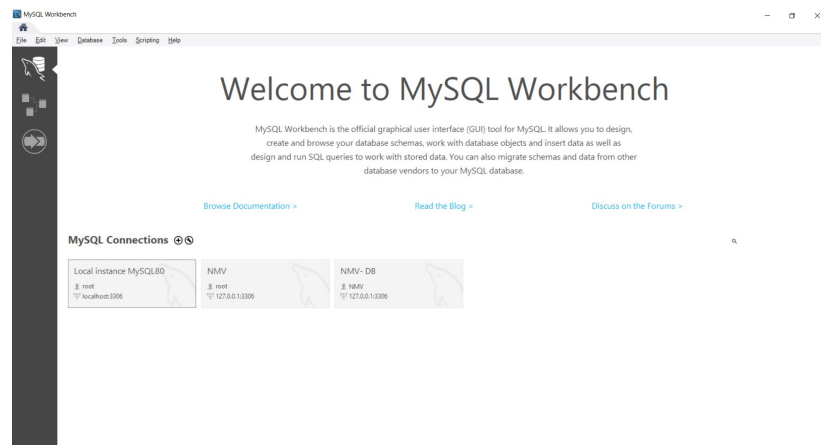


Рисунок 2.6 – Початковий екран програми MySQL Workbench

Ліворуч можна помітити три групи вкладок, позначені як дельфін, схема та стрілка. На першій вкладці можна адмініструвати чи налаштовувати сервер для подальшої роботи. На другій вкладці можна моделювати базу даних, тобто створювати діаграми. На третій вкладці можна перевести базу даних з інших СУБД у MySQL. На рисунку 2.7 показані усі три групи вкладок.

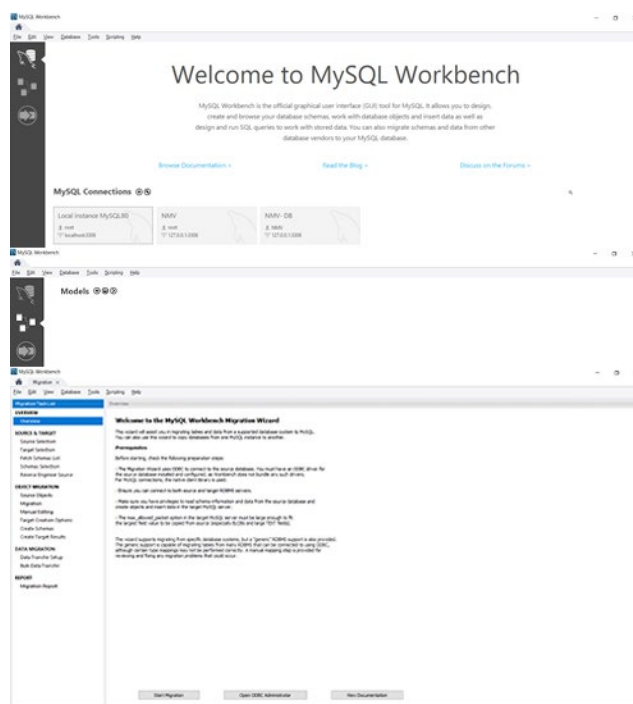


Рисунок 2.7 – Екрани груп вкладок MySQL Workbench

Для того, щоб почати створення бази даних, необхідно під'єднатися до серверу MySQL. На першій вкладці обираємо підключення і натискаємо на нього двічі. Після цього з'явиться екран з внутрішньою структурою серверу. Він показаний на рисунку 2.8.

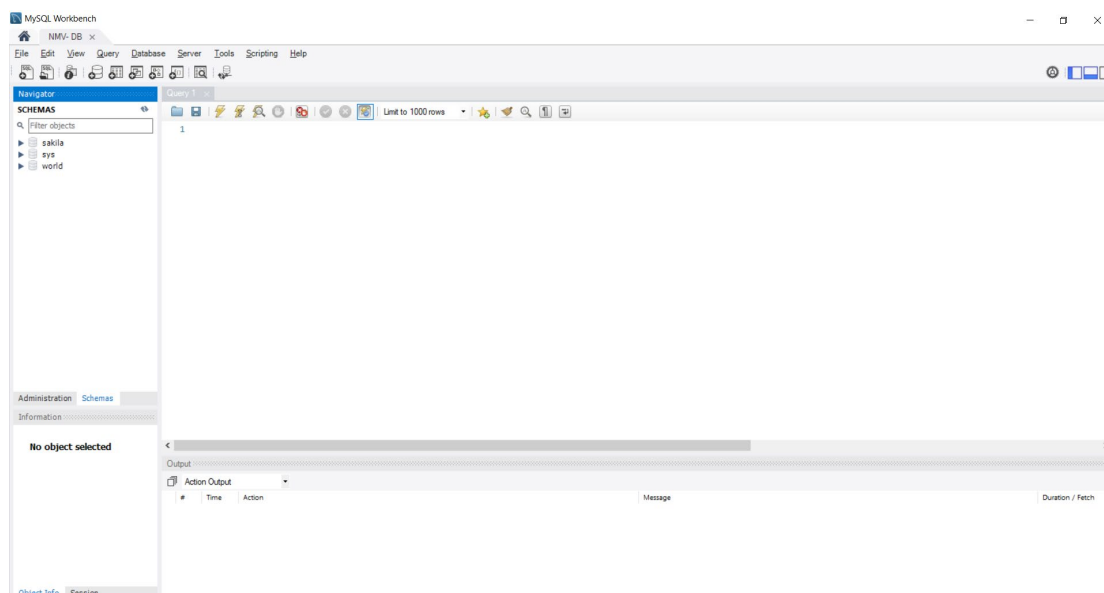


Рисунок 2.8 – Внутрішня структура серверу

Після успішного під'єднання переходимо до створення бази даних.

Для цього у вікні редактора пишемо команди, як показано на рисунку 2.9.

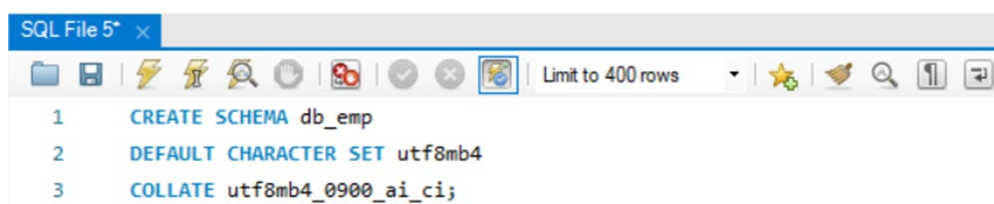


Рисунок 2.9 – Створення бази даних у вікні редактора

Потрібно зазначити, що назва бази даних обов'язково пишеться англійською мовою маленькими літерами. У другому рядку обрано набір символів за замовчуванням. У даному прикладі – це українська мова. У третьому рядку вказаний пошук у базі даних, який є нечутливим до великих чи маленьких літер.

Після написання команд необхідно приступити до їх виконання. Це можна зробити, натиснувши на кнопку у вигляді блискавки на панелі інструментів. Результат виконання показаний у нижній частині вікна з зеленою позначкою. Він означає, що команда коректна, як зображено на рисунку 2.10.

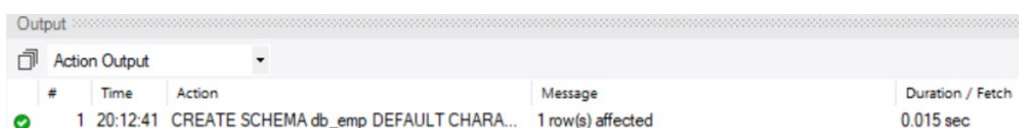


Рисунок 2.10 – Результат виконання команди

Якщо оновити секцію «SCHEMAS», то можна побачити створену базу даних. Це показано на рисунку 2.11.



Рисунок 2.11 – створена база даних у секції «SCHEMAS»

Тепер необхідно заповнити базу даних інформацією. Для цього створимо таблиці. У секції «SCHEMAS» поруч з назвою бази даних натиснемо на позначку розгортання списку та оберемо «Tables». У вікні редактору напишемо команди для створення таблиці, як показано на рисунку 2.12.

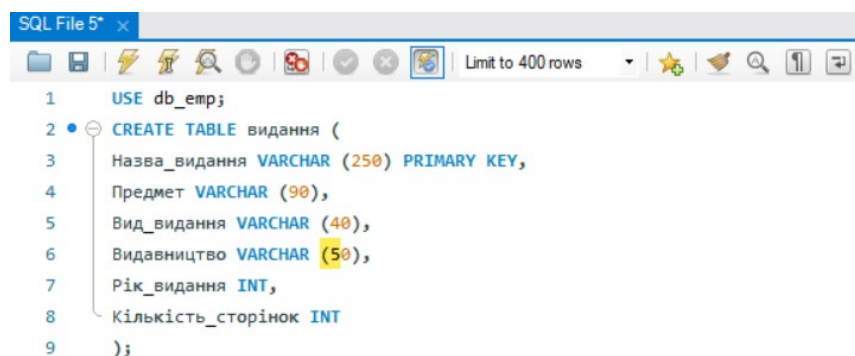


Рисунок 2.12 – Створення таблиці «Видання»

В першому рядку зазначаємо назву бази даних, в якій створюємо таблицю. У другому рядку створюємо таблицю з назвою «Видання». Важливо зазначити, що назву таблиць завжди пишемо маленькими літерами. З третього по дев'ятий рядок заповнюємо таблицю атрибутами і вказуємо їх тип. В даному прикладі назва видання є первинним ключем. Оновлюємо секцію «SCHEMAS» і бачимо створену таблицю, як на рисунку 2.13.



Рисунок 2.13 – Створена таблиця

Таким самим чином створюємо інші таблиці для бази даних. Секція «SCHEMAS» після закінчення створення таблиць виглядає так, як показано на рисунку 2.14.

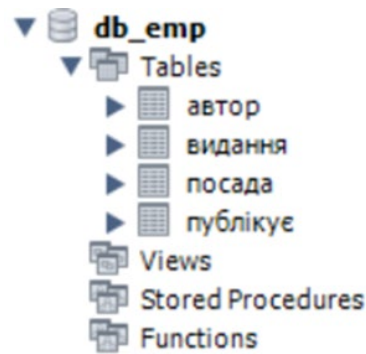


Рисунок 2.14 – Створені таблиці

Тепер заповнимо таблиці необхідною інформацією.

Приклад заповнення показаний на рисунку 2.15.

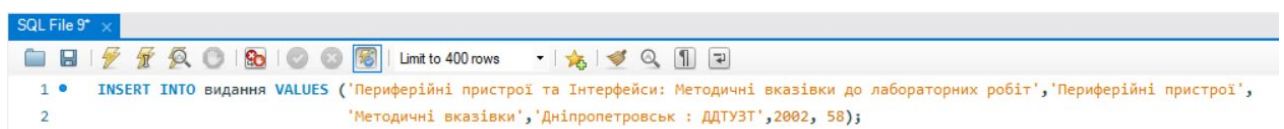


Рисунок 2.15 – Заповнення таблиці даними

Важливо, щоб дані у дужках співпадали з раніше зазначеними атрибутами [див. рис. 2.12]. Подивитися заповнену таблицю можна за допомогою команди, яка показана на рисунку 2.16.

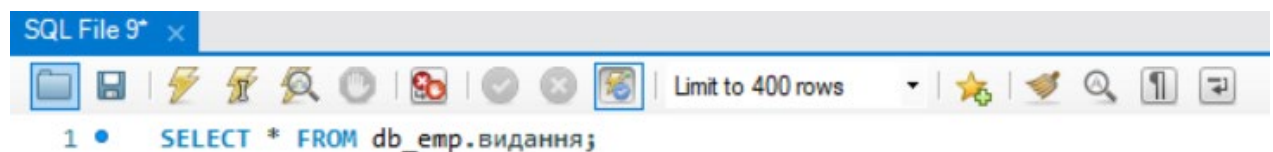


Рисунок 2.16 – команда для перегляду даних у таблиці

Приклад заповненої таблиці показано на рисунку 2.17.

Назва_видання	Спеціальність	Ступінь_освіти	Вид_видання	Видавництво	Рік_видання	Кількість_сторінок
Периферійні пристрої та Інтерфейси: Методи...	(123) Комп'ютерна інженерія	Бакалавр	Методичні вказівки	Дніпропетровськ : ДДУЗТ	2002	58
Машинна графіка : Метод. вказівки до лабор...	(123) Комп'ютерна інженерія	Бакалавр	Методичні вказівки	Дніпропетровськ : ДДУЗТ	2004	32
Основи прикладної криптографії : Методичні...	(123) Комп'ютерна інженерія	Бакалавр	Методичні вказівки	Дніпропетровськ : ДДУЗТ	2008	30
Основи прикладної криптографії : Методичні...	(125) Кібербезпека	Бакалавр	Методичні вказівки	Дніпропетровськ : ДДУЗТ	2008	30
Концептуальне проектування комп'ютерних...	(123) Комп'ютерна інженерія	Бакалавр	Монографія	Дніпропетровськ : ДДУЗТ	2018	276
Методичнеосие указання к лабораторным ра...	(123) Комп'ютерна інженерія	Бакалавр	Методичні вказівки	Дніпропетровськ : ДДУЗТ	1992	26
Системи штучного інтелекту : Методичні вк...	(123) Комп'ютерна інженерія	Магістр	Методичні вказівки	Дніпропетровськ : ДДУЗТ	2012	46
Системи штучного інтелекту : Методичні вк...	(125) Кібербезпека	Магістр	Методичні вказівки	Дніпропетровськ : ДДУЗТ	2012	46
Аналітичні моделі масового обслуговування ...	(123) Комп'ютерна інженерія	Бакалавр	Навчальний посібник	Дніпропетровськ : "Like Print" ФОП Гечка Т. О.	2015	186
Аналітичні моделі масового обслуговування ...	(125) Кібербезпека	Бакалавр	Навчальний посібник	Дніпропетровськ : "Like Print" ФОП Гечка Т. О.	2015	186
Логістичні основи штучного інтелекту : Метод...	(123) Комп'ютерна інженерія	Магістр	Методичні вказівки	Дніпро : Вид-во "Нова ідеологія"	2021	57

Рисунок 2.17 – Заповнена таблиця

Побудуємо зв'язки між таблицями. Для цього необхідно ввести SQL команду, яка зображена на рисунку 2.18.

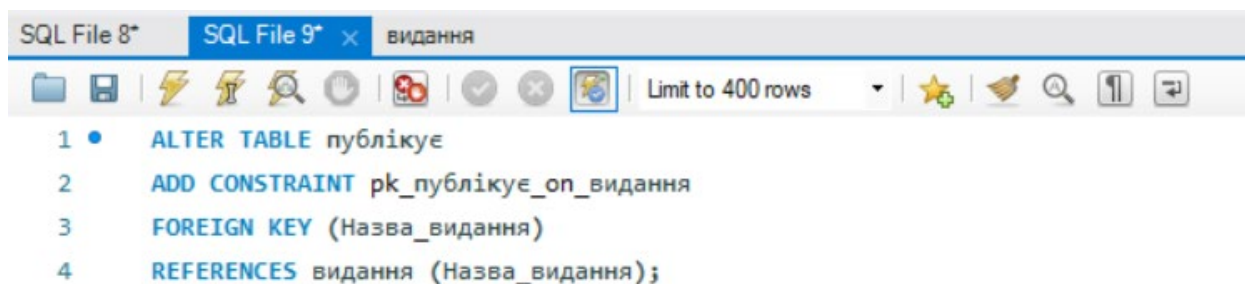


Рисунок 2.18 – Команда для створення зв'язку між таблицями

В першому рядку вказуємо, що будемо змінювати таблицю «публікує». В другому рядку ми вказуємо, що додаємо обмеження (Constraint), тобто зазначаємо, що до таблиці «публікує», додано певне обмеження, пишемо його назву. В третьому рядку показуємо тип обмеження – Foreign Key (зовнішній ключ) для стовпчика «Назва_видання» в таблиці «публікує». І тепер це обмеження логічно прив'язує стовпець «Назва_видання» в таблиці «публікує» до стовпця «Назва_видання» в таблиці «видання». Таким самим способом будемо зв'язок між всіма таблицями. Тепер база даних під назвою «DB_EMP» готова.

2.4 Співставлення характеристик MySQL Workbench з SQL Microsoft Access

Для порівняння характеристик додатків MySQL Workbench із Microsoft Access використаємо такі аспекти, як: основні характеристики, інтерфейс, вартість.

1. Основні характеристики.

MySQL Workbench – це потужна база даних корпоративного рівня, яка використовується багатьма корпораціями та сервісами в світі. MySQL Workbench – це СУБД SQL, яка розрахована на велику кількість користувачів, працюватиме надійно і з розумною продуктивністю при тій кількості користувачів, що обмежена доступною оперативною пам'яттю, швидкістю диска і швидкістю обладнання, а не властивими програмними обмеженнями.

Microsoft Access – досить хороший для домашніх користувачів і для одночасного користування десятима особами, але не може працювати з більш, ніж 2 гігабайтами даних і не повинен використовуватися на такому рівні. Microsoft Access – це відмінна програма для вивчення баз даних, має зручний

спосіб розробки додатків. 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 надає можливість створення зручних форм, запитів, звітів і навіть цілих зовнішніх програм. У Microsoft Access також вбудована СУБД SQL, що, 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 Workbench більше підходить для користувачів, які вже мали досвід роботи в СУБД, а Microsoft Access краще підійде для початку вивчення предметної області, бо інтерфейс схожий з Microsoft Word та Excel .

3. Вартість.

MySQL Workbench 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 ліцензію Windows, тому вважати програму повністю безкоштовною частково невірно.

Співставлення можливостей MySQL Workbench з SQL Microsoft Access за наступними напрямками: використання DDL та DQL, створення вкладених запитів.

Порівняння MySQL Workbench із SQL Microsoft Access виконуємо на основі запитів підмов, які представлені в підпунктах 2.5.1, 2.5.2 та 5.2.3. При порівнянні синтаксисів у двох програмах, знайдено лише одну різницю запису синтаксису з ключовим словом ORDER BY. MySQL Workbench при сортуванні даних з використанням лімітів використовує ключове слово LIMIT. У SQL Microsoft

Access ключове слово LIMIT не підтримується, замість нього після ключових слів ALL і DISTINCT після SELECT, використовується ключове слово TOP, яке має таку ж функцію, як і LIMIT. У таблиці 2.1 представлено синтаксис запиту з ключовим словом ORDER BY.

Таблиця 2.1 – Синтаксис порівняння

MySQL Workbench	SQL Microsoft Access
SELECT [* предикат_добору поле1, поле2] FROM таблиця1 [, таблиця2] WHERE умова добору ORDER BY поле1 ціле_значення [ASC DESC] LIMIT ... ;	SELECT [* предикат_добору поле1, поле2] FROM таблиця1 [, таблиця2] WHERE умова добору ORDER BY поле1 ціле_значення [ASC DESC] ;

Судячи з того, що різниці майже немає, переробляти лабораторні роботи для здобувачів курсу «Бази даних» спеціальності «Комп'ютерна інженерія» немає ніякого сенсу.

2.5 Рекомендації щодо організації індивідуального завдання здобувачів з дисципліни «Бази даних» на основі використання MySQL Workbench

2.5.1 Використання DDL

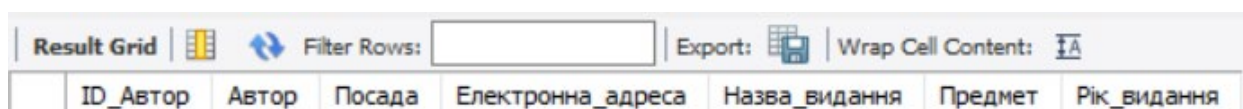
Оператори DDL (Data Definition Language) – мова опису даних, яка є частиною мови SQL, і дозволяє користувачу створювати, модифікувати та редагувати структуру бази даних. [15] У мові SQL існує шість основних команд DDL, які розглянемо нижче у вигляді прикладів.

Приклад 1. Використання CREATE TABLE для створення таблиці.

Створити таблицю АВТОР_ПУБЛІКУЄ_ВИДАННЯ з використанням команди CREATE TABLE. Запишемо відповідну конструкцію мови SQL:

```
CREATE TABLE автор_публікує_видання (
  ID_Автор INT UNSIGNED NOT NULL,
  Автор TINYTEXT NOT NULL,
```


Посада VARCHAR (20) NOT NULL,
 Електронна_адреса VARCHAR (30),
 Назва_видання VARCHAR (250) NOT NULL,
 Предмет VARCHAR (15) NOT NULL,
 Рік_видання INT UNSIGNED);
 Результат показаний на рисунку 2.19.



ID_Автор	Автор	Посада	Електронна_адреса	Назва_видання	Предмет	Рік_видання
----------	-------	--------	-------------------	---------------	---------	-------------

Рисунок 2.19 – Використання CREATE TABLE

Приклад 2. Створення первинного ключа.

У таблиці АВТОР_ПУБЛІКУЄ_МЕТОДИЧНІ_ВИДАННЯ зазначити поле ID_Автор первинним ключем. Запишемо відповідну конструкцію мови SQL:

```
CREATE TABLE автор_публікує_методичне_видання (
  ID_Автор INT UNSIGNED PRIMARY KEY NOT NULL,
  Автор TINYTEXT NOT NULL,
  Посада VARCHAR (40) NOT NULL,
  Електронна_адреса VARCHAR (40),
  Назва_видання VARCHAR (250) NOT NULL,
  Предмет VARCHAR (15) NOT NULL,
  Рік_видання INT UNSIGNED);
```

Приклад 3. Вимога унікальності.

У таблиці АВТОР_ПУБЛІКУЄ_ВИДАННЯ зазначити поле ID_Автор унікальним. В такому випадку поле не буде використовуватись, як ключ. Між ключовим та унікальним полем немає великої різниці, окрім того, що ключ використовується для упорядкування даних таблиці і для зв'язування таблиць.

Запишемо відповідну конструкцію мови SQL:

```
CREATE TABLE автор_публікує_видання (
  ID_Автор INT UNSIGNED UNIQUE NOT NULL,
  Автор TINYTEXT NOT NULL,
  Посада VARCHAR (40) NOT NULL,
```


Електронна_адреса VARCHAR (40),
 Назва_видання VARCHAR (250) NOT NULL,
 Предмет VARCHAR (15) NOT NULL,
 Рік_видання INT UNSIGNED);

Індекс – показник на дані в таблиці. Індеси бувають прості (створюються за даними одного стовпця таблиці) та унікальні (не дозволяють ввести в таблицю дублікати даних).

Зовнішній ключ – це стовпець у дочірній таблиці, що посиляється на ключ у батьківській таблиці.

Приклад 4. Створити таблиці АВТОР і ПУБЛІКУЄ_ВИДАННЯ. При цьому передбачити:

- визначення ключа в таблиці АВТОР;
- створення унікального індексу у таблиці АВТОР;
- призначення зовнішнього ключа у таблиці ПУБЛІКУЄ_ВИДАННЯ.

Запишемо відповідну конструкцію мови SQL:

```
CREATE TABLE автор (
  ID_Автор INT UNSIGNED NOT NULL,
  Автор TINYTEXT NOT NULL,
  Посада VARCHAR (40) NOT NULL,
  Електронна_адреса VARCHAR (40),
  PRIMARY KEY (ID_Автор));
CREATE UNIQUE INDEX ID_Автор_INDEX
  ON автор (ID_Автор);
CREATE TABLE публікує_видання (
  ID_Автор INT UNSIGNED NOT NULL,
  Назва_видання VARCHAR (250) NOT NULL,
  Предмет VARCHAR (15) NOT NULL,
  Рік_видання INT UNSIGNED,
  CONSTRAINT FK_ID_Автор FOREIGN KEY (ID_Автор)
  REFERENCES автор (ID_Автор));
```

У цьому прикладі ID_Автор призначений зовнішнім ключем таблиці ПУБЛІКУЄ_ВИДАННЯ. Цей зовнішній ключ посилається на стовпець ID_Автор таблиці АВТОР. Наявність зовнішнього ключа гарантує, що для кожного значення ID_Автор у таблиці ПУБЛІКУЄ_ВИДАННЯ знайдеться відповідне значення ID_Автор у таблиці АВТОР. Такий зв'язок називають батьківськодочірнім відношенням. Після створення таблиці її можна модифікувати.

Додавання (ADD). За допомогою цієї команди можна додавати поля та обмеження. При додаванні поля у вже існуючу таблицю з наявними в ній даними новому полю не можна призначити атрибут NOT NULL, так як NOT NULL означає, що поле повинне містити значення для кожного рядка в таблиці. Якщо поле, що додається, має атрибут NOT NULL, то виходить протиріччя, оскільки в таблиці немає значень для нового поля. Однак є можливість додати поле, що вимагає обов'язкового введення даних:

- 1) додайте поле, задавши йому атрибут NULL, тобто в полі необов'язково повинні бути присутні дані;
- 2) уведіть дані в кожен рядок нового стовпця таблиці;
- 3) якщо стовпець містить значення в кожному з рядків таблиці, змініть атрибут поля на NOT NULL.

Приклад 5. Додати у таблицю АВТОР поле Стаж.

Запишемо відповідну конструкцію мови SQL:

```
ALTER TABLE автор
```

```
ADD Стаж INT UNSIGNED;
```

Результат показаний на рисунку 2.20.



	ID_Автор	Автор	Посада	Електронна_адреса	стаж
*	NULL	NULL	NULL	NULL	NULL

Рисунок 2.20 – Додання поля Стаж

За допомогою команди DROP можна видаляти поля та обмеження.

Приклад 6. Із таблиці ПУБЛІКУЄ_ВИДАННЯ необхідно видалити існуючий зовнішній ключ, а потім знову його додати.

Синтаксис видалення зовнішнього ключа:

```
ALTER TABLE публікує_видання
```

```
DROP constraint fk_ID_Автор;
```

Синтаксис додавання зовнішнього ключа:

```
ALTER TABLE публікує_видання
```

```
ADD CONSTRAINT fk_ID_Автор
```

```
FOREIGN KEY (ID_Автор)
```

```
REFERENCES автор (ID_Автор);
```

Приклад 7. Із таблиці АВТОР необхідно видалити існуючий унікальний індекс.

Синтаксис наступний:

```
DROP INDEX ID_Автор_INDEX ON автор;
```

Приклад 8. Необхідно видалити таблицю ПУБЛІКУЄ_ВИДАННЯ.

Синтаксис наступний:

```
DROP TABLE публікує_видання;
```

2.5.2 Можливості DQL

Мова запитів даних (DQL) є частиною основної групи підмов SQL. Оператори DQL використовують для виконання запитів до даних. Призначення команд DQL – отримати відношення схеми із надісланого йому запиту.

Ключове слово SELECT використовується разом із ключовим словом

FROM для організації витягу даних із бази даних.

На рисунку 2.21 показана таблиця АВТОР, на основі якої розглянемо оператор SELECT.

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:					
	ID_Автор	Автор	Посада	Електронна_адреса	Стаж
▶	1	Дзюба Володимир Володимирович	Старший викладач	z.dzu@gmail.com	15
	2	Шаповалов Володимир Олександрович	Професор	shapovalov@gmail.c...	40
	3	Егоров Олег Йосипович	Доцент	egorof@ukr.net	20
	4	Устенко Андрій Борисович	Доцент	ustenko@gmail.com	30
	5	Доманська Галина Анатоліївна	Доцент	galina5@gmail.com	22

Рисунок 2.21 – таблиця АВТОР

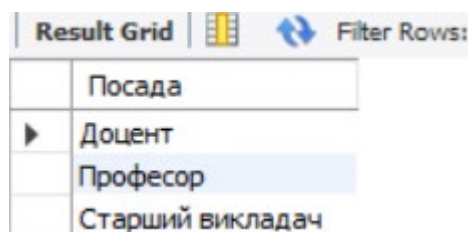
Приклад 1. Необхідно визначити посаду авторів на кафедрі електронних обчислювальних машин. Запишемо відповідну конструкцію мови SQL:

```
SELECT DISTINCT
```

```
Посада
```

```
FROM автор;
```

Результат запити показаний на рисунку 2.22.



The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. It contains a table with one column and four rows of job titles in Ukrainian. The third row, 'Професор', is highlighted in blue.

	Посада
▶	Доцент
	Професор
	Старший викладач

Рисунок 2.22 – Посади авторів

Ключове слово FROM – обов'язковий параметр у даному операторі. Він вказує на ім'я таблиці, з якої будуть обрані дані. Вираження FROM може включати як одну, так і кілька таблиць. Ключове слово WHERE задає умову, на основі якої відбираються дані. Вираження WHERE може містити декілька умов. Якщо умов декілька, то вони зв'язуються операціями кон'юнкції і диз'юнкції. Ці операції задаються за допомогою ключових слів AND і OR. Ключове слово AND дозволяє зв'язати логічним множенням дві умови у вираженні ключового слова WHERE. Щоб оператор SELECT був виконаний, обидві, зв'язані ключовим словом AND, умови повинні повернути true. Ключове слово OR дозволяє зв'язати логічним додаванням умови у вираженні ключового слова WHERE. Щоб оператор SELECT був виконаний, хоча б одна зі зв'язаних ключовим словом OR умов повинна повернути true.

Приклад 2. Виберемо з таблиці АВТОР (див. рис. 2.21) авторів, стаж яких від 15 до 30 років. Запишемо відповідну конструкцію мови SQL:

```
SELECT * FROM автор
```

```
WHERE Стаж BETWEEN 15 AND 30;
```

Результат виконання показаний на рисунку 2.23.

Result Grid Filter Rows: Edit: Export/Import:					
	ID_Автор	Автор	Посада	Електронна_адреса	Стаж
►	1	Дзюба Володимир Володимирович	Доцент	z.dzu@gmail.com	15
	3	Єгоров Олег Йосипович	Доцент	egorof@ukr.net	20
	4	Устенко Андрій Борисович	Доцент	ustenko@gmail.com	30
	5	Доманська Галина Анатоліївна	Доцент	galina5@gmail.com	22

Рисунок 2.23 – Автори, стаж яких від 15 до 30.

Ключове слово ORDER BY визначає сортування виводу даних.

Приклад 3. Виберемо з таблиці АВТОР Стаж і відсортуємо записи за алфавітом ініціалів авторів. Запишемо відповідну конструкцію мови SQL:

```
SELECT ALL *
FROM автор
WHERE Стаж = 20
ORDER BY Автор ASC;
```

Отриманий запит наведений на рис. 2.24.

Result Grid Filter Rows: Edit: Export/Import: Wrap					
	ID_Автор	Автор	Посада	Електронна_адреса	Стаж
►	11	Беляев Микола Борисович	Старший викладач	belyaevnb51@gmail.com	20
	3	Єгоров Олег Йосипович	Доцент	egorof@ukr.net	20
	13	Заець Олексій Петрович	Старший викладач	zaets.evm.diit@gmail.com	20
	14	Івін Павло Вікторович	Старший викладач	pavelvi1980@gmail.com	20
	7	Косолапов Анатолій Аркадійович	Доцент	kosolapov1920@i.ua	20
	8	Остапещь Денис Олександрович	Доцент	da.evm.diit@gmail.com	20
	16	Скабалланович Тетяна Іванівна	Старший викладач	sti19447@gmail.com	20

Рисунок 2.24 – Результат запиту прикладу 4

Приклад 4. Виберемо з таблиці АВТОР трьох авторів зі стажем 30 за посадою доцент і відсортуємо записи за ініціалами автора. Запишемо відповідну конструкцію мови SQL:

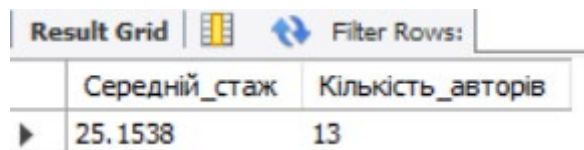
```
SELECT ALL *
FROM автор
WHERE Стаж = 30 AND Посада = 'Доцент'
ORDER BY Автор ASC
LIMIT 3;
```

Ключове слово GROUP BY використовується для того, щоб поєднувати повторювані значення в групи. Ключове слово GROUP BY повинне впливати за вираженням WHERE і передувати ключовому слову ORDER BY.

Приклад 5. На основі таблиці АВТОР необхідно визначити середній стаж авторів та загальну кількість авторів. Запишемо відповідну конструкцію мови SQL:

```
SELECT ALL
AVG(Стаж) AS 'Середній_стаж',
COUNT(Автор) AS 'Кількість_авторів'
FROM автор;
```

Результат показаний на рисунку 2.25.



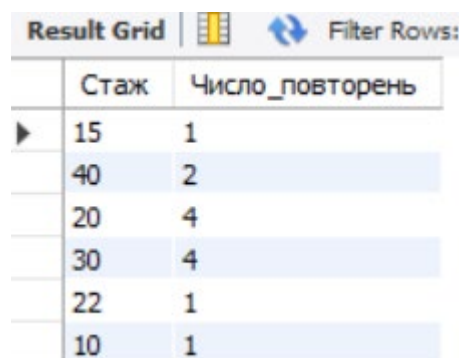
	Середній_стаж	Кількість_авторів
▶	25.1538	13

Рисунок 2.25 – Результат запити прикладу 5

Приклад 6. На основі таблиці АВТОР необхідно побудувати запит, за яким виводяться всі стажі авторів і число повторюваних значень. Запишемо відповідну конструкцію мови SQL:

```
SELECT ALL
Стаж, COUNT(Стаж) AS 'Число_повторень'
FROM Автор
GROUP BY Стаж;
```

Результат показаний на рисунку 2.26.



	Стаж	Число_повторень
▶	15	1
	40	2
	20	4
	30	4
	22	1
	10	1

Рисунок 2.26 – Результат запити прикладу 6

Ключове слово HAVING використовується в операторі SELECT разом із ключовим словом GROUP BY, щоб указати, які з груп повинні бути представлені у виводі. Ключове слово HAVING в операторі SELECT повинне впливати за

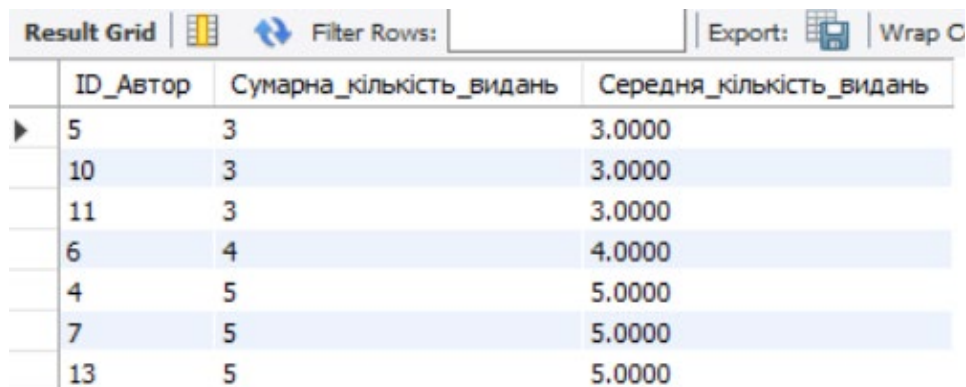
вираженням ключового слова GROUP BY і передувати ключовому слову ORDER BY, якщо останнє використовується.

Приклад 7. На основі таблиці АВТОР визначити сумарне і середнє значення видань за всіма ID_Автор, крім другого. Дані повинні бути згруповані за значеннями стовпця ID_Автор, але відображені тільки ті групи, для яких середнє значення кількості видань більша за 2. Результат відсортувати за значенням сумарної кількості видань.

Запишемо відповідну конструкцію SQL-мови:

```
SELECT ID_Автор,
SUM(Кількість_видань) as Сумарна_кількість_видань,
AVG(Кількість_видань) as Середня_кількість_видань
FROM автор
WHERE ID_Автор <> 2
GROUP BY ID_Автор
HAVING AVG (Кількість_видань) > 2
ORDER BY 2;
```

Результат показаний на рисунку 2.27.



	ID_Автор	Сумарна_кількість_видань	Середня_кількість_видань
▶	5	3	3.0000
	10	3	3.0000
	11	3	3.0000
	6	4	4.0000
	4	5	5.0000
	7	5	5.0000
	13	5	5.0000

Рисунок 2.27 – Результат запиту прикладу 7

2.5.3 Створення вкладених запитів

Приклад 1. Оператор повертає ID_Автор, Стаж, Кількість_видань, Назва_видання і Кількість_сторінок для усіх авторів, у яких кількість_сторінок перевищує кількість_сторінок у виданні під номером 7.

Запишемо відповідну конструкцію SQL-мови:


```

SELECT  A.ID_Автор,  A.Стаж,  A.Кількість_видань,  B.Назва_видання,
B.Кількість_сторінок
FROM Автор AS A, Видання AS B
WHERE A.ID_Автор = B.ID_Видання
AND B.Кількість_сторінок > (select Кількість_сторінок
FROM Видання
WHERE Видання.ID_Видання = '7'
LIMIT 1);

```

Результат запити представлений на рисунку 2.27.

Result Grid					Filter Rows:	Export:	Wrap Cell Content:
ID_Автор	Стаж	Кількість_видань	Назва_видання	Кількість_сторінок			
13	30	5	Логістичні основи штучного інтелекту : Методичні вказівки до виконання лабораторних робіт	65			

Рисунок 2.27 – Результат запити прикладу 7

Приклад 2. Існує таблиця АВТОР_ПУБЛІКУЄ_ВИДАННЯ. Потрібно ввести дані в поля ID_Автор, Автор, Посада, Електронна_адреса, Стаж, Кількість_видань, ID_Видання, Назва_видання, Предмет, Рік_видання та Кількість_сторінок із створених раніше таблиць АВТОР і ВИДАННЯ для тих видань, в яких кількість_сторінок перевищує кількість_сторінок ID_Видання 3.

Запишемо відповідну конструкцію SQL-мови:

```

INSERT INTO автор_публікує_видання
SELECT  A.ID_Автор,  A.Автор, A.Посада,  A.Електронна_адреса, A.Стаж,
A.Кількість_видань,
B.ID_Видання,      B.Назва_видання,      B.Предмет,      B.Рік_видання,
B.Кількість_сторінок
FROM автор AS A, Видання AS B
WHERE A.ID_Автор = B.ID_Видання
AND B.Кількість_сторінок > (select Кількість_сторінок
FROM Видання
WHERE Видання.ID_Видання = '3'
LIMIT 1);

```

Результат запити представлений на рисунку 2.28.

ID_Автор	Автор	Посада	Електронна_адреса	Стаж	Кількість_видань	ID_Видання	Назва_видання	Предмет	Рік_видання	Кількість_сторінок
2	Шаповалов Володи...	Профе...	shapovalov@gmail.c...	40	3	2	Бази даних. М...	Бази да...	2007	30
7	Косолапов Анатолі...	Доцент	kosolapov1920@i.ua	20	5	7	Теорія інформа...	Теорія і...	2019	56
10	Заєць Олексій Петр...	Старш...	zaets.evm.dit@gmail...	20	3	10	Комп'ютерна ...	Комп'ю...	1980	45
11	Івїн Павло Вікторович	Старш...	pavelvi1980@gmail....	30	3	11	Обчислювальн...	Обчисл...	2005	32
13	Скабалланович Тет...	Доцент	sti19447@gmail.com	30	5	13	Логістичні осн...	Шткчни...	2003	65

Рисунок 2.28 – Результат запиту прикладу 2

Приклад 3. Обновити дані стовпця Стаж таблиці АВТОР_ПУБЛІКУЄ_ВИДАННЯ, установивши значення 25 для тих авторів із таблиці АВТОР, для котрих Вік_видання датується 1980 роком. Запишемо відповідний оператор UPDATE із підзапитом:

UPDATE автор_публікує_видання

SET Стаж = 25

WHERE ID_Автор IN (select ID_Видання

FROM видання

WHERE Рік_видання = 1980);

Результат запиту представлений на рисунку 2.29.

Result Grid											Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:	
ID_Автор	Автор	Посада	Електронна_адреса	Стаж	Кількість_	ID_Видання	Назва_видання	Предмет	Рік_видання	Кількість_сторінок								
2	Шаповалов Володи...	Професор	shapovalov@gmail.com	40	3	2	Бази даних. Методичні вказі...	Бази даних	2007	30								
7	Косолапов Анатолі...	Доцент	kosolapov1920@i.ua	20	5	7	Теорія інформації та кодува...	Теорія інформації	2019	56								
10	Заєць Олексій Петр...	Старший викладач	zaets.evm.dit@gmail.com	25	3	10	Комп'ютерна техніка та пр...	Комп'ютерна техн...	1980	45								
11	Івїн Павло Вікторович	Старший викладач	pavelvi1980@gmail.com	30	3	11	Обчислювальна техніка та ...	Обчислювальна те...	2005	32								
13	Скабалланович Тет...	Доцент	sti19447@gmail.com	30	5	13	Логістичні основи штучного ...	Шткчний інтелект	2003	65								

Рисунок 2.29 – Результат запиту прикладу 3

Приклад 4. Із таблиці АВТОР_ПУБЛІКУЄ_ВИДАННЯ видалити дані про авторів, у котрих Стаж 30 і Рік_видання 2003. Запишемо відповідну конструкцію SQL-операторів:

DELETE FROM автор_публікує_видання

WHERE ID_Автор=(select * FROM (SELECT ID_Автор FROM автор_публікує_видання

WHERE Стаж=30 AND Рік_видання=2003

LIMIT 1) AS автор_публікує_видання);

Результат запиту представлений на рисунку 2.30 [16].

Result Grid											Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:	
	ID_Автор	Автор	Посада	Електронна_адреса	Стаж	Кількість	ID_Видання	Назва_видання	Предмет	Рік_видання	Кількість_сторінок							
2		Шаповалов Володи...	Професор	shapovalov@gmail.com	40	3	2	Бази даних. Методичні вказі...	Бази даних	2007	30							
7		Косолапов Анатолі...	Доцент	kosolapov1920@i.ua	20	5	7	Теорія інформації та кодув...	Теорія інформації	2019	56							
10		Заєць Олексій Петр...	Старший викладач	zaets.evm.dit@gmail.com	25	3	10	Комп'ютерна техніка та пр...	Комп'ютерна техн...	1980	45							
11		Івн Павло Вікторович	Старший викладач	pavelvi1980@gmail.com	30	3	11	Обчислювальна техніка та ...	Обчислювальна те...	2005	32							

Рисунок 2.30 – Результат запиту прикладу 4

2.6 Основні висновки

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

2. Для проєктування бази даних навчально-методичних видань у якості початкового відношення взяті дані репозиторію УДУНТ, яка виконана методом «Сутність – зв'язок», в результаті чого отримано 4 відношення, які співпадають з отриманою структурою бази даних за методом нормальних форм, виконаної іншим здобувачем, Чайкою Романом Руслановичем, що підтверджує правильність проєктування.

3. Для створення та обробки бази даних методичних видань, використовувався додаток MySQL Workbench, мова – MySQL.

4. Виконано порівняння додатків MySQL Workbench з Microsoft Access за такими аспектами, як: основні характеристики, інтерфейс та вартість. Виконано співставлення можливостей MySQL Workbench з SQL Microsoft Access за наступними напрямками: використання DDL та DQL, створення вкладених запитів.

5. Пропонується використання MySQL Workbench для індивідуального завдання здобувачів першого (бакалаврського) ступеню для спеціальності «Комп'ютерна інженерія», переробляти лабораторні роботи з дисципліни «Бази даних» немає ніякого сенсу. Також рекомендується створену базу даних навчально-методичних видань «DB_EMP» до використання на сайті кафедри ЕОМ.

ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

1. У кваліфікаційній роботі виконано огляд і аналіз предмета проєктування бази даних навчально-методичних видань кафедри електронних обчислювальних машин. У роботі розглянуто метод «Сутність-зв'язок» – графічний метод, що візуалізує майбутню базу даних та дозволяє уникнути недоліків, які потім важко виправити, особливо якщо проєкт вже експлуатується. Існують різні додатки для побудови бази даних, наприклад: MySQL Workbench, Altibase, Oracle, SQL Azure, PostgreSQL, Microsoft Access та інші. Для виконання кваліфікаційної роботи обрано MySQL Workbench, який має зручний інтерфейс, дозволяє швидко обробляти команди, має ефективну систему безпеки та не потребує купівлі ліцензії.

2. Виконано проєктування бази даних навчально-методичних видань за методом «Сутність-зв'язок», на основі даних взятих з репозиторію та даних викладачів кафедри електронно обчислювальних машин. Створено початкове відношення «DB_EMP», що складається із 13 атрибутів та 93 кортежів. В результаті проєктування отримано 4 відношення, які збігаються зі структурою бази даних, отриманою методом нормальних форм іншим здобувачем, Чайкою Романом Руслановичем, що підтверджує правильність проєкту.

3. Створено базу даних та виконано її обробку за допомогою додатку MySQL Workbench.

4. Виконано порівняння додатків MySQL Workbench з Microsoft Access за такими аспектами, як: основні характеристики, інтерфейс та вартість. Виконано співставлення можливостей MySQL Workbench з SQL Microsoft Access за наступними напрямками: використання DDL та DQL, створення вкладених запитів.

5. Переробляти лабораторні роботи з дисципліни «Бази даних» немає ніякого сенсу, тому пропонується використання MySQL Workbench для індивідуального завдання здобувачів першого (бакалаврського) ступеню для спеціальності «Комп'ютерна інженерія». Також рекомендується створену базу даних

навчально-методичних видань «DB_EMP» до використання на сайті кафедри ЕОМ.

ПЕРЕЛІК ПОСИЛАНЬ

1. Основні поняття реляційних БД: нормалізація, зв'язок та ключі [Електронний ресурс]. – Режим доступу: <https://bondarenko.dn.ua/osnovni-ponyattya-relyatsijnih-bd-normalizatsiya-zv-yazok-ta-klyuchi/>
2. Поняття ER-моделі. Поняття сутності (entity). Атрибути. Види атрибутів [Електронний ресурс]. – Режим доступу: <https://www.bestprog.net/uk/2019/01/24/the-concept-of-er-model-the-concept-of-essence-and-communication-attributes-attribute-types-ua/>
3. Модель «Сутність - зв'язок» [Електронний ресурс]. – Режим доступу: https://stud.com.ua/154733/informatika/model_sutnist_zvyazok
4. Дистанційний курс з дисципліни «Бази даних», укладач доц. Пахомова В.М. [Електронний ресурс]. – Режим доступу: <https://lider.diit.edu.ua/>
5. 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
6. Програмні засоби роботи з базами даних. Система управління базами даних [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/7716502/page:52/>
7. Что такое MySQL? [Електронний ресурс]. – Режим доступу: <https://freehost.com.ua/faq/wiki/chto-takoe-mysql/>
8. Вікіпедія. MySQL [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/MySQL>
9. Введение в Altibase [Електронний ресурс]. – Режим доступу: <https://russianblogs.com/article/6320380196/>
10. Офіційний сайт «Altibase» [Електронний ресурс]. – Режим доступу: <https://altibase.com/en/index.php>

11. Вікіпедія. Oracle [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Oracle>
12. Что такое база данных SQL Azure? [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/ru-ru/azure/azure-sql/database/sql-database-paas-overview?view=azuresql>
13. Вікіпедія. SQL Azure [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/SQL_Azure
14. Офіційний сайт «PostgreSQL» [Електронний ресурс]. – Режим доступу: <https://www.postgresql.org>
15. Ярцев В.П. Організація баз даних та знань: навчальний посібник.-К. ДУТ 2018.-214с.
16. Пахомова В. М. Методичні вказівки до індивідуальних робіт з дисципліни «Бази даних» /Дніпропетр. нац. ун-т залізнич. трансп. ім. акад. В. Лазаряна. [Електронний ресурс]. – Режим доступу: https://lider.diit.edu.ua/pluginfile.php/16939/mod_resource/content/3/Метод_БД_Защита.pdf
17. Гарбера І. В. Лінгвістична база даних "Концепт ЛЮДИНА у фразеології східноєвропейських українських говірок": структура та функції. Журнал:Лінгвістичні студії. Вип. 37, 2019. с. 123-130.
18. Бідолак Д.І., Кузьович та В.С. Лакида П.І. Просторова база даних урболандшафтів на прикладі зелених насаджень міста Бережани. Науковий вісник НЛТУ України. 2020. с.51-56.
19. Довга Г. Р., Чайка Р. Р., науковий керівник Пахомова В.М. Проектування бази даних навчально-методичних видань кафедри електронних обчислювальних машин. Молода академія 2022. Збірка тез доповідей. Всеукраїнська науково-технічна конференція студентів і молодих учених. Дніпро: УДУНТ, 2022. с. 146