**Міністерство освіти і науки України**

**Національний університет «Одеська політехніка»**

**Методичні вказівки до виконання лабораторних робіт з дисципліни**

**"Організація баз даних"**

**Одеса 2023**

Міністерство освіти і науки України

Національний університет «Одеська політехніка»

Методичні вказівки до виконання лабораторних робіт з дисципліни

"Організація баз даних"

Затверджено

на засіданні кафедри комп'ютерних

інтелектуальних систем та мереж

Протокол № 7 від 23.03.2023

Одеса 2023

Методичні вказівки до виконання лабораторних робіт з дисципліни "Організація баз даних" для здобувачів спеціальності 123 – "Комп’ютерна інженерія". /Укл.: О.В. Головачова, П.М. Тішин, А.А. Нагорна – Одеса: Національний університет «Одеська політехніка», 2023. – 28 с.

Укладачі:

Головачова О.В., ст. викладач

Тішин П.М., к.ф-м.н, доцент

Нагорна А.А., асистент

**ЗМІСТ**

1 Загальні положення 4

2 Оформлення лабораторних робіт 5

3 Предметні області лабораторних робіт 8

4 Лабораторна робота №1 11

5 Лабораторна робота №2 13

6 Лабораторна робота №3 15

7 Лабораторна робота №4 18

8 Лабораторна робота №5 21

10 Лабораторна робота №6 23

11 Лабораторна робота №7 25

Перелік джерел посилань 27

Додаток А Титульна сторінка 28

**1 Загальні положення**

Метою лабораторних робіт є поглиблення та закріплення знань, отриманих здобувачами, першого рівня вищої освіти, у процесі вивчення дисципліни «Організація баз даних». Оволодіння необхідними знаннями щодо створення сучасних систем обробки інформації. В результаті виконання лабораторних робіт здобувач має показати вміння самостійно сформулювати, вирішити завдання та набутти практичних навичок у реалізації його у середовищі MySQL з використанням мови структурованих запитів SQL. Формулювати власні, самостійні, обґрунтовані, аргументовані висновки.

Кожна, виконана, лабораторна робота оформлюється як протокол на аркушах паперу розміром А4. Титульна сторінка протоколу наведена у додатку А.

Структурними елементами протоколу є:

- титульна сторінка;

- цілі та завдання до лабораторної роботи;

- постановка задачі. Перелік завдань, які необхідно виконати в роботі. Завдання може включати опис предметної області, а також умови та обмеження, якими необхідно керуватися при додаванні, видаленні, модифікації або вибірці інформації з бази даних;

- формулювання програмного коду або моделі бази даних відповідно до завдання;

- результати виконання;

- висновок.

**2 Оформлення лабораторних робіт**

На, кожну, лабораторну роботу оформлюється протокол роботи.

Текст протоколу пишеться державною мовою в друкованому вигляді на аркушах формату А4, шрифтом Times New Roman, чорного кольору, прямого накреслення, 14 пунктів, міжрядковий інтервал 1,5; між абзацний інтервал – нуль; абзацний відступ –1,25см. (відповідає 5 символам); поля: зліва – 2,5 см, справа – 1 см, зверху – 2 см, знизу – 2 см; номер сторінки – в правому верхньому куті аркуша. Нумерація сторінок починається зі «Змісту» – 3 стор.

Необхідно дотримуватися рівномірної насиченості, контрастності та чіткості зображень. Усі лінії, літери, цифри та знаки мають бути чіткі та нерозпливчасті по всій роботі.

Необхідно дотримуватись єдиної термінології та позначень. Необхідно вживати наукову чи науково–технічну термінологію, запроваджену національними стандартами на терміни та визначення понять.

Якщо останнім елементом на сторінці є рисунок, таблиця, схема, то обов’язково, необхідно додати до двох рядків тексту (кожна сторінка має завершуватися текстом).

Забороняється застосовувати напівжирний шрифт в тексті.

Оформлення рисунків:

- рисунки (схеми, графіки, креслення) мають бути якісними та чіткими;

- рисунки необхідно розташовувати в роботі після тексту, у якому вони згадуються вперше або на наступній сторінці. На всі рисунки мають бути посилання в тексті із зазначенням його номера (наприклад: ER-діаграма наведена на рисунку 1);

- при необхідності рисунки доповнюють пояснювальними даними, які розміщуються у межах рисунку;

- рисунки треба розміщувати по центру аркуша;

- кожен рисунок повинен мати назву. Назва рисунка має відображати його зміст, бути стислою та конкретною;

- рисунки позначають словом Рисунок та послідовно нумерують;

- номер рисунку повинен складатися з порядкового номера рисунка. Номер рисунку та йог назву розміщують послідовно під рисунком, після пояснювальних даних, по центру аркуша. (Наприклад: Рисунок 1 – ER-діаграма (перший рисунок лабораторної роботи);

- якщо рисунки створені не автором роботи, подаючи їх у роботі, треба дотримуватися вимог чинного законодавства України про авторське право;

- рисунки пустими рядками від тексту не виділяються.

Оформлення таблиць:

- таблиці необхідно розташовувати в роботі після тексту, у якому вони згадуються вперше або на наступній сторінці. На всі таблиці мають бути посилання в тексті із зазначенням її номера (наприклад: вихідні дані наведено у таблиці 1);

- кожна таблиця повинна мати номер та назву, які розміщують над самою таблицею ліворуч, починаючі з абзацного відступу;

- заголовки граф повинні починатися з великих літер, підзаголовки – з маленьких, якщо вони складають одне речення із заголовком, а з великих, якщо вони є самостійними. Графу з порядковими номерами рядків до таблиці включати непотрібно;

- якщо дані в будь-якому рядку таблиці не подають, то в ньому треба ставити прочерк;

- таблиці нумерують послідовно у межах лабораторної роботи ( наприклад: Таблиця 1 - Вихідні дані);

- при переносі частини таблиці на інший аркуш слово “Таблиця”, її номер та назву вказують один раз ліворуч над першою частиною таблиці, на інших аркушах, над іншими частинами таблиці пишуть “Продовження таблиці” або “Кінець таблиці”, без повторення її назви і вказують номер таблиці (наприклад: Продовження таблиці 1) .

- якщо таблиці створені не автором роботи, подаючи їх у роботі, треба дотримуватися вимог чинного законодавства України про авторське право;

- таблиці пустими рядками від тексту не виділяються.

Оформлення перелічень:

- перед переліченням ставлять двокрапку (крім пояснювальних перелічень на рисунках);

- якщо перелічення одного рівня підпорядкованості, на які у тексті роботи немає посилань, то перед кожним із перелічень ставлять знак – (тире);

- якщо у тексті роботи є посилання на перелічення, підпорядкованість позначають малими літерами української абетки, далі — арабськими цифрами, далі — через знаки – (тире);

- після цифри або літери певної позиції перелічення треба ставити круглу дужку;

- текст кожної позиції перелічення треба починати з малої літери з абзацного відступу відносно попереднього рівня підпорядкованості.

Оформлення кодів програм:

- шрифтом CourierNew 11 пунктів з одинарним міжрядковим інтервалом;

- обов’язково, після має бути посилання на результат виконання запиту.

Оформлення результатів виконання запитів виконується як оформлення рисунків.

Оформлення діаграм (схем даних):

- написи мають бути читабельними;

- блоки мають бути якісними та чіткими;

- блоки повинні бути в одному масштабі;

- лінії з’єднань мають бути однакової товщини,

- розташовувати блоки таким чином, щоб лінії з’єднань не мали багато перетинів та не закривали ці лінії;

- розташовувати блоки так, щоб лінії з’єднань мали кут вигину 90о .

**3 Предметні області лабораторних робіт**

Перелік предметних областей:

1) Бібліотека;

2) Університет;

3) Оптова база;

4) Виробництво;

5) Мережа магазинів;

6) Авторемонтна майстерня;

7) Деканат;

8) Договірна діяльність організацій;

9) Поліклініка;

10) Телефонна станція;

11) Спортивна організація;

12) Сільськогосподарські роботи;

13) Міський транспорт;

14) Географія;

15) Будинкоуправління;

16) Аеропорт;

17) Комп'ютерна техніка;

18) Відомості про студентів;

19) Іподром;

20) Червона книга;

21) Магазин радіодеталей;

22) Лісне господарство;

23) Автотранспортне підприємство;

24) Планетарій;

25) Відеопрокат;

26) Рецепти;

27) Крюїнгова компанія;

28) Ріелтерська компанія;

29) Агенція з працевлаштування;

30) Страхова компанія;

31) Фотоцентр;

32) Ательє з пошиття одягу;

33) Відділ кадрів;

34) Кадрова агенція;

35) Будівельна організація;

36) Мережа ресторанів;

37) Охорона агенція;

38) Меблевий магазин;

39) Компанія з надання телекомунікаційних послуг;

40) Юридична агенція;

41) Нотаріальна агенція;

42) Авіакомпанія;

43) Хлібозавод;

44) Туристична агенція;

45) Студія звукозапису;

46) Аптека;

47) Готель;

48) Банк;

49) Школа;

50) Дитячий садок;

51) Видавництво;

52) Порт;

53) Ж/Д вокзал;

54) Автовокзал; - Ткаченко

55) Архів;

56) Кіностудія;

57) Радіостанція;

58) Кабельне телебачення; - Риженко

59) Ломбард;

60) Обмінний пункт;

61) Аукціонний дім;- Дядюк

62) Комісійний магазин; Кисельов

63) Поліцейський відділок; - Солом’яний

64) Пожежна частина;- Онощенко

65) Вантажні перевезення; - Верба

66) Служба таксі; - Подоленчук

67) Відділ збуту; - Горенко

68) Морські вантажні перевезення;- Чумак

69) Освітні курси; - Фіщук

70) Станція технічного обслуговування - Яровий;

71) Ринок\_\_\_; Артем Рибак

72) Ресторан;- кармілін

73) Event-агенція;-

74) Музей- Халілов;

75) Служба доставки - Мельник;

76) Інтернет провайдер Демченко.

**4 Лабораторна робота №1**

Тема: Проєктування бази даних

Мета: отримати навички проєктування концептуальної та логічної моделей бази даних із використанням сучасних технологій.

Етапи побудови інформаційної моделі:

1 - визначення сутностей;

2 - визначення залежностей між сутностями;

3 - задання первинних та альтернативних ключів;

4 - визначення атрибутів сутностей;

5 - приведення моделі до необхідного рівня нормальної форми;

6 - перехід до фізичного опису моделі: призначення відповідностей назва сутності – назва таблиці, атрибут сутності – атрибут таблиці;

7 - завдання тригерів, процедур та обмежень;

8 - генерація бази даних.

Основними елементами концептуальної моделі є - сутності, зв'язки між ними та їх властивості (атрибути).

Для побудови логічної моделі потрібно:

- сформувати таблиці для кожної Сутності та кожних Зв'язків;

- для кожної таблиці задати ключі (первинний, зовнішній);

- визначити типи зв'язків.

Зауваження:

- предметна область - частина реального світу;

- сутність – будь-який відмінний об'єкт, інформація про який зберігається в базі даних;

- екземпляр сутності – конкретний представник сутності;

- атрибут – названа характеристика сутності (властивість деякої сутності -поле таблиці);

- ключ – мінімальний набір атрибутів за значеннями яких можна однозначно знайти екземпляр сутності;

- зв'язок – асоціація, яка встановлена між сутностями;

- домен атрибута - безліч допустимих значень, які може приймати атрибут;

- кортеж - кінцева безліч взаємопов'язаних допустимих значень атрибутів, які разом описують деяку сутність (рядок таблиці);

- відношення - кінцева безліч кортежів (таблиця);

- схема відношення - кінцева безліч атрибутів, що визначають деяку сутність. Іншими словами, це структура таблиці, що складається з конкретного набору полів;

- проекція - відношення, отримане з заданого шляхом видалення та (або) перестановки деяких атрибутів;

- при створенні логічної моделі необхідно враховувати обмеження, що накладаються системою керування базою даних на проєктовану базу даних, а також вимоги предметної області до цілісності даних;

- логічна модель не залежить від системи керування базою даних.

Завдання:

1. З переліку предметних областей обрати тему для лабораторних робіт.

2. Описати концептуальний рівень проектування бази даних.

3. Описати логічний рівень проектування бази даних.

Контрольні питання:

1. Які основні етапи проектування баз даних?

2. Що передбачає концептуальне проектування бази даних?

3. Чим відрізняється концептуальна модель бази даних від інфологічної?

4. Архітектура бази даних? Фізична та логічна незалежність?

5. Операції над відношенями. Реляційна алгебра

**5 Лабораторна робота №2**

Тема: Розробка фізичної моделі бази даних.

Мета: отримання навичок розробки сховищ даних з використанням реляційних баз даних, проєктування фізичної моделей бази даних.

Зауваження:

- основна мета проектування реляційної бази даних полягає у групуванні атрибутів у відношенні таким чином, щоб мінімізувати надмірність даних;

- процес нормалізації є формальним методом, що дозволяє визначати відношення на основі їх первинних чи потенційних ключів та функціональних залежностей, що існують між їх атрибутами;

- відношення знаходиться в 1НФ, якщо всі його атрибути є простими, всі використовувані домени містять тільки скалярні значення;

- відношення знаходиться в 2НФ, якщо воно знаходиться в 1НФ і кожен не ключовий атрибут неприводимо залежить від первинного ключа;

- відношення знаходиться в 3НФ, коли воно знаходиться у 2НФ і кожен не ключовий атрибут нетранзитивно залежить від первинного ключа;

- первинний ключ - це атрибут або набір атрибутів, за сукупним значенням яких можна виділити єдиний кортеж відношення;

- базові типи даних: числові типи даних, дані типу дата та час, рядкові типи даних, просторові типи даних;

- декомпозиція - метод поділу цілого на частини;

- зовнішній ключ є обмеженням, мета якого обмежити дані , які вводяться таким чином, щоб зовнішній та первинний ключі відповідали принципам цілісності посилання;

- 

Завдання:

1. Описати приведення таблиць до 3 нормальної форми.

2. Описати фізичний рівень проектування БД.

3. Для таблиць визначити первинні та зовнішні ключі.

4. Створити загальну схему- даних бази даних з вказанням типів зв’язків.

Контрольні питання:

1. Які є основні типи даних?

2. Які можна накладати обмеження на атрибути при створені таблиць?

3. Для чого використовується модель "сутність-зв'язок"?

4. Яка мета нормалізації?

5. Що називають аномалією та які вини бувають?

**6 Лабораторна робота №3**

Тема: Ознайомлення з середовищем MySQL та мовою SQL.

Мета: Вивчити набір команд мови SQL, пов'язаний із створенням бази даних, створенням, модифікацією структури таблиць та їх видаленням.

Зауваження:

1. Запустити MySQL 8.0 Command Line Client.

Ввести пароль, який був встановлений при установці MySQL для користувача root. І після вдалого підключення, можна відправляти серверу команди через консольний клієнт.

2 Переглянути, які бази даних є на сервері:

show databases;

3. Створити базу даних:

create database *ім'я \_БД*;

Видалити базу даних:

drop database *ім'я \_БД;*

Видалити таблицю з бази даних:

DROP TABLE *ім'я \_таблиці;*

4. Під’єднатися до бази даних.

Щоб звернутися до певної бази даних, спочатку треба встановити потрібну базу даних як поточну. Для цього потрібно виконати команду:

use *ім'я \_БД*;

5. Переглянути список таблиць:

show tables;

**6. Переглянути структуру таблиці:**

**describe *ім'я \_таблиці*;**

**7. Скопіювати таблицю в іншу базу даних:**

**CREATE TABLE DB1.NewTable AS SELECT \* FROM DB2.OldTable**

 **де DB1.NewTable – куди копіювати, ім'я базы данных.** **ім'я таблиці,**

 **DB2.OldTable – звідки копіювати, ім'я базы данных.** **ім'я таблиці.**

**8. Створити копію табліці в поточній базі даних:**

**CREATE TABLE *нове\_****ім'я****\_таблиці* LIKE *старе\_****ім'я****\_таблиці*;**

**INSERT** *нове\_ім'я\_таблиці* **SELECT \* FROM** *старе\_ім'я\_таблиці***;**

**9. Перейменування таблиці:**

**RENAME TABLE** *старе\_ім'я\_таблиці***TO** *нове\_ім'я\_таблиці;*

10. Створення таблиць:

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] *ім'я \_таблиці*

(*ім'я\_стовпця* тип\_даних [NOT NULL | NULL]

 [DEFAULT {літерал | (вираз)} ]

 [AUTO\_INCREMENT]

 [UNIQUE [KEY]] [[PRIMARY] KEY]

 [COMMENT 'текст']

 …..

 [CONSTRAINT [символ]] CHECK (вираз) [[NOT] ENFORCED]

);

11. Заповнення таблиць даними:

INSERT [INTO] *ім'я \_таблиці* [(*стовпець1, стовпець2,…, стовпецьN*)]

VALUES (*значення1, значення2, ..., значенняN*)

12. Видалення стовпця з таблиці:

alter table *ім'я\_таблиці* DROP COLUMN *ім'я\_стовпця*;

13. Додати стовпець до таблиці:

ALTER TABLE *ім'я\_таблиці* ADD COLUMN *ім'я\_стовпця* Тип даних[( )];

14. Змінити назву стовпця:

ALTER TABLE *ім'я\_таблиці* CHANGE *старе*\_*ім'я\_стовпця*

 *нове*\_*ім'я\_стовпця* Тип даних[( )];

15. Створити стовпець з унікальним ім'ям, що б уникнути дублікатів у назвах:

ALTER TABLE [*ім'я\_таблиці*] ADD UNIQUE ([*ім'я\_стовпця*]);

16. Змінити розмір стовпця:

ALTER TABLE [*ім'я\_таблиці*] MODIFY [*ім'я\_стовпця*] Тип даних( );

Завдання:

1. Запустити MySQL 8.0 Command Line Client.

2. Переглянути, які бази даних є на сервері.

3. П’єднатися до будь-якої базі даних з існуючих.

4. Переглянути список таблиць.

5. Переглянути структуру будь-якої таблиці.

6. Створити свою базу даних, за темою предметної області, з набором таблиць які були зпроєктувані у фізичної моделей у попередній лабораторній роботі.

7. Забезпечити цілісність даних.

8. Заповнити кожну таблицю даними (не менше 10 записів).

Контрольні питання:

1. Що таке база даних?

2. Що таке схема бази даних?

3. Навіщо використовують CASE інструменти?

4. Для чого використовується мова SQL?

5. Наведіть обґрунтування вибраних типів даних для атрибутів схеми.

**7 Лабораторна робота №4**

Тема: Використання запитів на вибірку.

Мета: Виконання запитів та складнопідрядних запитів на мові SQL.

Зауваження:

1. Загальний синтаксис оператору SELECT:

SELECT

[ALL | DISTINCT | DISTINCTROW]

[FROM назва\_таблиці

[WHERE умови]

[GROUP BY {назва\_полів | вираз | становище}, ...]

[HAVING умови]

[ORDER BY {назва\_полів | вираз | становище}

[ASC | DESC], ...]

;

2. Оператор Select використовується:

- для вибірки даних із бази даних;

- для отримання нових рядків у складі оператора Insert;

- для оновлення інформації у складі оператора Update.

3. Після оператора WHERE вказується умова: OR( ||), AND( &&), XOR, IS, NOT, BETWEEN, IN, LIKE, !=, <>, >, >=, <, <=, &, |, <<, >>, +, -, \*, /, ~, и %.

4. Для запитів із пошуком за шаблоном, які базуються на пошуку під рядків у полях, використовується ключове слово LIKE з:

- % - замінює послідовність символів;

- \_ замінює будь-який одиночний символ.

Включення у вираз ключового слова NOT породжує умову із зворотним змістом.

5. Агрегатні функції:

- AVG - обчислює середнє значення;

- SUM - обчислює суму значень;

- MIN - обчислює найменше значення;

- MAX - обчислює найбільше значення;

- COUNT - обчислює кількість рядків у запиті.

6. Результати виконання оператора ALL:

- x > ALL (1, 2) еквівалентно x > 2;

- x < ALL (1, 2) еквівалентно x < 1;

- x = ALL (1, 2) еквівалентно (x = 1) AND (x = 2);

- x <> ALL (1, 2) еквівалентно x NOT IN (1, 2).

7. Результати виконання оператора ANY (а також SOME):

- x > ANY (1, 2) еквівалентно x > 1;

- x < ANY (1, 2) еквівалентно x < 2;

- x = ANY (1, 2) еквівалентно x NOT (1, 2);

- x <> ANY (1, 2) еквівалентно (x <> 1) OR (x <> 2).

7. Пріоритет операцій:

!

- (унарний мінус), ~ (унарна інверсія )

^

\*, /, DIV, %, MOD

-, +

<<, >>

&

|

= (порівняння), <=>, >=, >, <=, <, <>, !=, IS, LIKE, REGEXP, IN

BETWEEN, CASE, WHEN, THEN, ELSE

NOT

AND, &&

XOR

OR, ||

= (розподіл), :=

Завдання:

1. Використати запити на вибірку з явним та неявним з'єднанням таблиць,

2. Задати псевдоніми для таблиць,

3. Об'єднати дві вибірки використовуючи оператор UNION,

4. При формуванні запитів, використовувати агрегатні функції,

5. Вибирати не всі дані з БД, а тільки ті, які відповідають певній умові (обов’язкове використання операторів LIKE, IN, BETWEEN, REGEXP.

6. Передбачити сортування та групування даних,

7. Виключити з набору значення повторюваних даних,

8. Застосувати навички вживання оператору UPDATE,

9. Застосувати навички вживання оператору DELETE по деякій умові.

10 Створити, як найменше, один корелюючий підзап.

Загальна кількість запитів повинна бути не менше 20.

Контрольні питання:

1. Що таке корельований запит? Чим відрізняється корельований запит від некорельованого?

2. Як зберегти результати запиту у таблиці?

3. Які існують обмеження на формування корельованого запиту?

4. Що таке зовнішнє з'єднання?

5. Які існують засоби групування SQL? Як вони використовуються?

6. Якими засобами SQL реалізуються такі операції реляційної алгебри: обмеження, декартів добуток, проекція, перетин, об'єднання, різниця, з'єднання?

**8 Лабораторна робота №5**

Тема: Створення та використання зберігаючих процедур в MySQL.

Мета: Навчитися основ роботи зі зберігаючими процедурами в MySQL, а також ознайомитися з їх властивостями, що впливають на безпеку даних у таблицях.

Зауваження:

- інструкція CALL викликає збережену процедуру, яка була визначена раніше за допомогою CREATE PROCEDURE;

- збережені процедури, які не приймають аргументів, можуть бути викликані без дужок;

- CREATE PROCEDURE мають мати привілеї CREATE ROUTINE;

- за замовчуванням MySQL автоматично надає привілеї ALTER ROUTINE та EXECUTE процедурі;

- зазначення параметра як IN, OUT або INOUT дійсне лише для процедур;

- той самий ідентифікатор можна використовувати для параметра процедури, локальної змінної та стовпця таблиці. Крім того, те саме ім’я локальної змінної можна використовувати у вкладених блоках;

- будь-які збережені процедури, які взаємодіють із таблицями MySQL, повинні бути повторно створені шляхом виконання операторів CREATE PROCEDURE на кожному сервері MySQL;

- локальні змінні знаходяться в області видимості лише під час виконання збереженої процедури;

- для збереженої процедури треба використовувати SQL SECURITY INVOKER у визначенні об’єкта, щоб його могли використовувати лише користувачі з дозволами, відповідними для операцій, які виконує об’єкт;

- на збережену процедуру не може посилатися користувач, який не має EXECUTE на це прав;

- SQL SECURITY DEFINER за замовчуванням, коли виконується процедура з характеристикою, MySQL Server не встановлює жодних активних ролей для облікового запису MySQL, зазначеного в DEFINER, лише ролі за замовчуванням;

- якщо викликати процедуру з іншої збереженої процедури або функції, також можна передати параметр або локальну змінну як IN або INOUT параметр;

- перед викликом процедури треба ініціалізувати змінну, яка буде передана як INOUT параметр;

- параметр IN передає значення в процедуру;

- параметр OUT передає значення з процедури.

Завдання:

1. Під'єднатися до бази даних, створеної та заповненої на попередніх лабораторних роботах.

2. Створити, як найменше, 3 процедури (CREATE PROCEDURE).

3. При створенні враховувати характеристики зберігаючої процедури: Language, Deterministic, SQL Security, Comment.

4. Викликати створені процедури (CALL), використовуючи різні параметри.

5. Змінити процедуру, оголосивши у ній кілька змінних (DECLARE. Організувати обробку даних у процедурі, використовуючи конструкції IF, CASE, WHILE).

Контрольні питання:

1. Для чого використовуються зберігаючи процедури?

2. Для чого використовуються InnoDB таблиці?

3. Який контроль доступу до збережених процедур?

4. Де зберігаються збережені процедури?

**9 Лабораторна робота №6**

Тема: Створення та використання тригерів у MySQL.

Мета: Навчитися створювати тригери у середовищі MySQL та за допомогою їх забезпечувати виконання обмежень цілісності даних.

Зауваження:

- тригери в різних схемах можуть мати однакові назви;

- тригери можуть бути прив'язані не до таблиці, а до уявлення (VIEW). У цьому випадку з їх допомогою реалізується механізм " оновлюючого уявлення";

- на стовпці таблиці, до якої прив'язаний тригер, можна посилатися за допомогою псевдонімів OLD і NEW (OLD.назва\_стовпця);

- після видалення таблиці, MySQL автоматично видаляє прив'язані до неї тригери;

- для таблиці можна визначити кілька тригерів з однаковою дією тригера та часом дії. За замовчуванням тригери з однаковою дією та часом активуються в порядку їх створення;

- тригер не може використовувати CALL оператор для виклику збережених процедур, які повертають дані клієнту або використовують динамічний SQL;

- тригер не може використовувати оператори, які явно чи неявно починають або завершують транзакцію (START TRANSACTION, COMMIT або ROLLBACK;

- для нетранзакційних таблиць відкат неможливо виконати;

- використовуючи конструкцію BEGIN ... END можна визначити тригер, який виконує кілька операторів;

- збережені підпрограми можуть забезпечити покращену продуктивність, оскільки потрібно передавати менше інформації між сервером і клієнтом;

- збережені функції не можна використовувати рекурсивно;

- збережена функція або тригер не може змінити таблицю, яка вже використовується (для читання або запису) оператором, який викликав функцію або тригер.

Завдання:

1. Під'єднатися до бази даних, створеної та заповненої на попередніх лабораторних роботах.

2. Для будь-яких таблиць бази даних створити тригер, який активізується операцією введення та після додавання запису виведе на екран повідомлення «Запис додано!».

3. Перевірити роботу тригера, виконавши операцію додавання запису у таблиці.

4. Для будь-яких таблиць бази даних створити тригер, який при видаленні запису з однієї таблиці спочатку видаляє всі пов'язані з нею записи з інших, а потім видаляє запис.

5. Виконати тестування роботи тригера.

6. Для будь-якої з таблиць БД створити тригер, що демонструє відкат операції (ROLLBACK).

7. Створити, як найменше, один корелюючий запит за допомогою тригера (наприклад: зміна загальної кількості).

Контрольні питання:

1. Яка роль тригерів при контролі цілісності даних?

2. Які відбуваються дії, якщо всередині тіла тригера відбудеться помилка?

3. Що таке обмеження цілісності?

4. У чому важливість задання обмежень цілісності?

**10 Лабораторна робота №7**

Тема: Основи безпеки баз даних засобами mysql.

Мета: Отримання навичок із проектування та реалізації розмежування прав доступу до бази даних командами мови SQL. Ознайомитись з утилітами, що входять до складу системи управління базами даних MySQL та отримати навички роботи з ними.

Зауваження:

- при введенні пароля треба використовувати функцію password(), інакше пароль не працюватиме;

- з метою безпеки всі паролі шифруються mysql, тому в полі Password, у відповідній таблиці, буде відображено набір, деяких, символів;

- не треба призначати привілеї користувачам у таблиці user, тому що в цьому випадку вони будуть глобальними та поширюються на всі бази даних. Треба надавати привілеї кожному користувачеві до конкретної бази даних;

- при заданні імені хоста для входу через мережу рекомендується явно вказувати повне ім'я хоста, а не '%'. Можна також вказувати IP-адреси комп’ютерів та маски підмереж для більшої безпеки.

Завдання:

1. Повторити синтаксис команд мови SQL, які пов’язані зі створенням ролей користувачів бази даних та призначенням їм привілеїв:

- CREATE ROLE;

- ALTER ROLE;

- DROP ROLE;

- GRANT;

- REVOKE.

2. Використовуючи базу даних, створену та заповнену на попередніх лабораторних роботах створити трьох користувачів та задати для них паролі.

3. Створити три ролі.

4. Наділити першу роль усіма привілеями (додавання, зміна, видалення).

5. Наділити другу роль привілеями на перегляд та зміну частини таблиць.

6. Призначити третій ролі другу роль але скасувати одну з привілеїв.

7. Відповідно, для кожного користувача призначити ролі.

8. Змінити другу роль та видалити другу.

На кожному етапі увійти до бази даних під відповідним користувачем та перевірити доступність/недоступність привілеїв.

Контрольні питання:

1. Які основні функції адміністратора бази даних?

2. Для чого потрібна перевірка повноважень?

3. Як організовано захист інформації у базах даних?

4. Які є утиліти для адміністрування сервера та які функції вони виконують?

5. Що таке привілеї та яке їх призначення?

**Перелік джерел посилань**

1. Francesco Corea An Introduction to Data Everything You Need to Know About AI, Big Data and Data Science - Springer Nature Switzerland AG 2019

2. Raghu Ramakrishnan, Johannes Gehrke, Jeff Derstadt, Scott Selikoff, and Lin Zhu - DATABASE MANAGEMENT SYSTEMS SOLUTIONS MANUAL THIRD EDITION - Ithaca, NY, USA, 2004 [Електронний ресурс]. URL: https://www.pdfdrive.com/database-management-systems-solutions-manual-third-edition-e1087973.html

3. https://dev.mysql.com/doc/refman/8.0/en/ - Reference Manual

4. https://dev.mysql.com/doc/workbench/en/ - Workbench Manual

5. https://dev.mysql.com/doc/ - MySQL Documentation

6. https://dev.mysql.com/downloads/shell/ - MySQL Shell 8.0.32

Міністерство освіти і науки України

Національний університет «Одеська політехніка»

Інститут комп’ютерних систем

Кафедра комп'ютерних інтелектуальних систем та мереж

Лабораторна робота №\_\_\_

з дисципліни

«Організація баз даних»

предметна область

«Міська енергетична компанія»

Виконав здобувач

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Групи\_\_\_\_\_\_\_\_\_\_\_

Перевірив викладач

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Одеса 2023