

Міністерство освіти і науки України
Національний університет водного господарства та
природокористування
Навчально-науковий інститут
кібернетики, інформаційних технологій та інженерії
Кафедра комп'ютерних наук та прикладної математики

04-01-85М

МЕТОДИЧНІ ВКАЗІВКИ ТА ЗАВДАННЯ

**до виконання лабораторних робіт
з навчальної дисципліни
«Бази даних»**

для здобувачів вищої освіти першого (бакалаврського) рівня
за освітньо-професійною програмою «Інтернет речей»
спеціальності 121 «Інженерія програмного забезпечення» та за
освітньо-професійною програмою «Комп'ютерні науки»
спеціальності 122 «Комп'ютерні науки»
денної та заочної форм навчання

Рекомендовано
навчально-методичною
радою з якості ННІКІТІ
Протокол № 4
від 24 лютого 2025р.

Рівне – 2025

Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Бази даних» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Інтернет речей» спеціальності 121 «Інженерія програмного забезпечення» та за освітньо-професійною програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» денної та заочної форм навчання. [Електронне видання] / Бачишина Л. Д., Харів Н. О. – Рівне : НУВГП, 2025. – 23 с.

Укладачі: Бачишина Л. Д., к.е.н., доцентка кафедри комп'ютерних наук та прикладної математики;
Харів Н. О., старша викладачка кафедри комп'ютерних наук та прикладної математики.

Відповідальний за випуск: Турбал Ю. В. – д.т.н., професор, завідувач кафедри комп'ютерних наук та прикладної математики.

Керівник групи забезпечення спеціальності 122 «Комп'ютерні науки» Каштан С. С.

Керівник групи забезпечення спеціальності 121 «Інженерія програмного забезпечення» Жуковський В. В.

Попередня версія МВ:04-01-13

ЗМІСТ

Лабораторна робота № 1. Тема: Інсталяція <i>MySQL</i>	4
Лабораторна робота № 2. Проектування та моделювання бази даних.....	5
Лабораторна робота № 3. Створення структури таблиць.....	8
Лабораторна робота № 4. Редагування структури таблиць. Внесення даних.....	11
Лабораторна робота № 5-6. Прості запити. Групові операції. Використання агрегатних функцій.	12
Лабораторна робота № 7. Багатотабличні запити.	14
Лабораторна робота № 8. Багатотабличні запити. Вкладені запити.	16
Лабораторна робота № 9. Представлення.	17
Лабораторна робота № 10. Збережені процедури.	18
Лабораторна робота № 11. Збережені функції і тригери.	20
Лабораторна робота № 12. Безпека бази даних. Користувачі, ролі, права.	21
Інформаційні ресурси.....	23

Методичні вказівки призначені для закріплення теоретичних знань з дисципліни «Бази даних», отримання практичних навичок роботи в реляційній СКБД *MySQL* і, зокрема, графічній оболонці *Workbench*. Особлива увага приділяється використанню структурованої мови запитів *SQL* (*Structured Query Language*), яка надає засоби створення і обробки даних в реляційних базах даних і є основною базовою мовою в різних СКБД.

Лабораторна робота № 1

Тема: Інсталяція *MySQL*.

Теоретичні відомості:

1) *MySQL* 8.0 є клієнт-серверною системою керування реляційними базами даних (СКБД). Для ознайомлення з *MySQL* достатньо в пошуковій системі Google задати *mysql* і перейти на офіційний сайт <https://www.mysql.com/>. Інсталяційний пакет можна завантажити з сайту *DOWNLOADS* ⇒ *MySQL Community (GPL) Downloads* ⇒ *MySQL Installer for Windows* або за посиланням <https://dev.mysql.com/downloads/installer/> (*mysql-installer-community-8.0.40.0.msi*). Для встановлення рекомендується вибирати інсталяцію **Full**, що включає *MySQL Server*, графічну оболонку *MySQL Workbench*, консольну оболонку *MySQL Shell*, *MySQL Router*, документацію і приклади баз даних. Для роботи з *MySQL* у різних системах програмування рекомендується при потребі встановити додатково конектори. Під час інсталяції необхідно задати для користувача-адміністратора **root** пароль.

2) Відкрити *MySQL Workbench*.

3) На вкладці **Models**  бічної панелі головного вікна ознайомитись із схемою даних *sakila*.

4) Встановити сеанс роботи із сервером на вкладці **MySQL**

Connections



5) Перевірити роботу сервера на панелі **Navigator** ⇒

вкладка **Administration** ⇒ **Server Status**



6) Переглянути таблиці бази даних *sakila* на панелі **Navigator** ⇒ вкладка **Schemas**.

7) Завершити роботу.

Контрольні запитання:

- 1) Які компоненти входять в інсталяційний пакет *MySQL*?
- 2) Для чого вони призначені?
- 3) Який користувач є адміністратором СКБД *MySQL*?
- 4) Для чого використовується *MySQL Workbench*?
- 5) Які вкладки містить бічна панель *MySQL Workbench*?
- 6) Як розпочати сеанс роботи з *MySQL* сервером?
- 7) Як перевірити роботу *MySQL* сервера?

Лабораторна робота № 2


Тема: Проектування та моделювання бази даних

Завдання до виконання:

- 1) Запустити *MySQL Workbench* і вибрати вкладку **Models**




бічної панелі головного вікна.

2) За допомогою кнопки із знаком +  створити нову модель бази даних.

3) За замовчуванням створюється модель `mydb`. Двічі клацнути мишкою по імені `mydb`. У вікні властивостей перейменувати модель в `univer`. Задати кодування **Charset/Collation:** `cp1251/cp1251_ukrainian_ci`.

- 4) На вкладці редактора моделей **MySQL Model** за

допомогою піктограми **Add Diagram**  створити нову діаграму **EER** (*Extended Entity-Relationship Model* – розширена модель сутність-зв'язок).

5) На вкладці редактора діаграм **EER Diagram** побудувати таблиці `student` і `uspish` для бази даних `univer` за

допомогою інструмента **Place a New Table** .

6) Двічі клацнувши мишкою таблицю, відкрити її в редакторі таблиць.

- 7) Задати поля відповідно до вимог:

Таблиця student

snom – номер залікової книжки студента є первинним ключем, тип даних INT; задати значення **NN** (*Not Null*) і **PK** (*Primary Key*). Для полів sfam (прізвище), sima (ім'я) задати тип даних VARCHAR(15). Поле group – група, в якій навчається студент. Для нього задати значенням за замовчуванням назву своєї групи в одинарних лапках. Поле stip (стипендія) має тип даних DECIMAL з 2 знаками після коми. Для поля foto (фото студента) вибрати тип даних BLOB (LONG BLOB). Це поле, що містить дані великого об'єму (*Binary Large Object*) такі як графіка, текст, цифровий звук у двійковому вигляді. Для поля form (форма навчання) задати тип даних ENUM(), що допускає ввід значень «платна», «бюджет» або NULL.

Таблиця usplash

nom є первинним ключем, задати тип даних INT і опцію **Auto Incremental** для функції лічильника. Для поля data (дата здачі іспиту) тип даних DATE; pnom – номер предмета. Поле pnom буде використовуватись для зв'язку з таблицею predmet. Поле ocinka (оцінка за іспит) є цілим. Для таблиці usplash зовнішній ключ створювати не потрібно. Він автоматично додається при побудові зв'язку з таблицею student.

8) За допомогою інструмента **Place a New 1:n One-To-**



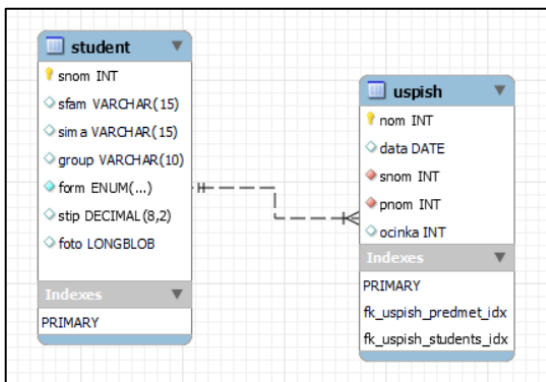
Many Non-Identifying Relationship

задати зв'язок від таблиці usplash до student. Для цього необхідно клацнути на таблиці із зв'язками «багато», потім на таблиці, що має первинний ключ. Зв'язок буде неідентифікований.

При створенні первинних і зовнішніх ключів у таблицях автоматично задаються індекси, які прискорюють доступ і пошук інформації при виконанні запитів.

9) У редакторі таблиць для таблиці usplash на вкладці **Foreign Keys** задати умови цілісності, що задовольняють умовам каскадування. Назву створеного ключа рекомендується змінити на snom.

10) В результаті буде створена EER-діаграма, подібна до даної



11) За допомогою команди **Model** ⇒ **Object Notation** і **Model** ⇒ **Relationship Notation** переглянути модель бази даних у різних нотаціях.

12) За допомогою команди **Database** ⇒ **Forward Engineer** за даною моделлю побудувати фізичну базу даних. Попередній перегляд **SQL** скрипта перед виконанням команди **Execute** зберегти у файлі **univer.sql**.

13) За допомогою команди **File** ⇒ **Save Model As** зберегти модель бази даних у файлі **univer.mwb**.

14) Повернутися до головного вікна програми,



натиснувши



15) Вибрати **MySQL Connections**. Відкрити сеанс роботи із сервером.

16) На панелі **Navigator** ⇒ вкладка **Schemas** переглянути об'єкти бази даних.

17) Завершити роботу.

Контрольні запитання:

- 1) Що таке логічна модель бази даних?
- 2) Як створити логічну модель бази даних?
- 3) Що являє собою EER-діаграма?

- 4) Як додати таблиці у EER-діаграму?
- 5) Які є типи даних?
- 6) Як встановити зв'язки між таблицями?
- 7) Які є типи зв'язків?
- 8) Що являє собою неідентифікований зв'язок?
- 9) У яких нотаціях можна описати EER-діаграми?
- 10) Як побудувати фізичну базу даних на основі логічної моделі?

Лабораторна робота № 3


Тема: Створення структури таблиць.

Завдання до виконання:

- 1) Запустити *MySQL Workbench* і встановити підключення

до сервера на вкладці **MySQL Connections**  бічної панелі головного вікна.

2) На панелі **Navigator** ⇒ вкладка **Schemas** активувати базу даних *univer*, клацнувши її два рази мишкою. Назва виділиться жирним окресленням і буде використовуватись за замовчуванням.

- 3) За допомогою інструмента **Create a new table**  створити нову таблицю *vyklad* відповідно до вимог.

Таблиця *vyklad*

vnom – табельний номер викладача, є цілим, не допускає невизначених значень, задається як первинний ключ таблиці; *vfam* – прізвище; *vima* – ім'я; *kaf* – місце роботи; *posada* – посада викладача; *oklad* – оклад викладача. Для окладу викладача вибрати тип даних **DECIMAL**, задати 2 знаки після коми.

- 4) Відкрити файл **univer.sql**, створений на попередній лабораторній роботі. Ознайомитись із командами створення таблиць.

5) На вкладці **Query1** (або кнопка **Create a new SQL tab ...**



панелі інструментів) за допомогою команди CREATE TABLE створити таблицю `predmet` відповідно до вимог:

Таблиця `predmet`

`pnom` – номер предмета в навчальному плані, є цілим і є первинним ключем; `pname` – назва предмета; `god` – кількість годин з даного предмета; `semestr` – семестр, в якому викладається предмет; `vnom` – табельний номер викладача. Для таблиці `predmet` задати зв'язок із таблицею `vyklad` за полем `vnom`. Задати умови цілісності: при видаленні – SET NULL, при зміні – CASCADE.

Спрощений синтаксис команди:

```
CREATE TABLE ім'я_таблиці
(ім'я_поля тип_даних [(розмір_поля [, точність])]
[обмеження],
ім'я_поля тип_даних [(розмір_поля [, точність])]
[обмеження], ...,
[PRIMARY KEY (ім'я_поля)],
[UNIQUE (ім'я_поля)],
[CONSTRAINT ім'я_ключа]
[FOREIGN KEY (ім'я_поля) REFERENCES
ім'я_таблиці1
(ім'я_поля_первинного_ключа_таблиці1)
[ON DELETE {NO ACTION|CASCADE|SET NULL}]
[ON UPDATE {NO ACTION|CASCADE|SET NULL}]]
);
```

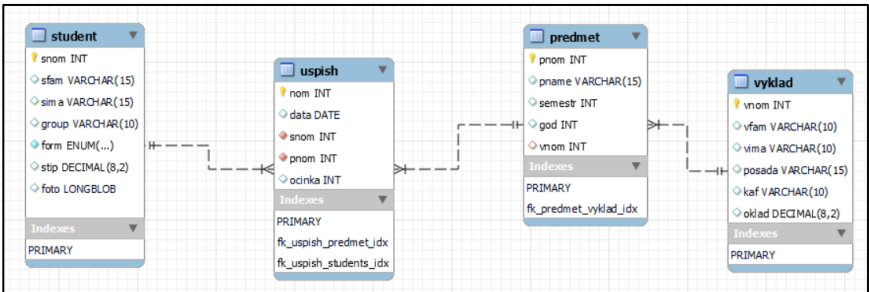
6) На панелі **Navigator** для таблиці `uspish` вибрати

редактор таблиці (2 кнопка) 

На вкладці **Foreign keys** редактора задати зв'язок із таблицею `predmet` за полем `pnom`. Задати умови цілісності: при видаленні – NO ACTION, при зміні – NO ACTION.

7) За допомогою команди **Database** ⇒ **Reverse Engineer** за базою даних побудувати логічну модель.

8) За допомогою команди **File** ⇒ **Save Model As** зберегти модель бази даних у файлі **univer.mwb**.



9) Завершити роботу.

Контрольні запитання:

- 1) Як створити таблицю в *MySQL Workbench*?
- 2) Як створити таблицю з використанням команди мови *sql*?
- 3) Що таке первинний ключ?
- 4) Які є типи обмежень для таблиць?
- 5) Як задати зв'язок між таблицями за допомогою команди мови *sql*?
- 6) Які правила підтримки цілісності бази даних можуть задаватись для зовнішнього ключа при видаленні записів?
- 7) Які правила можуть задаватись для зовнішнього ключа при зміні записів?
- 8) Які побудувати логічну модель на основі створеної бази даних?

Лабораторна робота № 4

Тема: Редагування структури таблиць. Внесення даних.

Завдання до виконання:

- 1) Запустити *MySQL Workbench* і встановити підключення

до сервера на вкладці **MySQL Connections**  бічної панелі головного вікна.

2) На панелі **Navigator** ⇒ вкладка **Schemas** активувати базу даних *univer*, клацнувши два рази мишкою.

3) Додати у таблицю *predmet* поле *pole* типу **FLOAT**.

4) Змінити назву поля *pole* на *pole1*.

5) Змінити тип даних на **VARCHAR(15)**.

6) Розташувати дане поле третім по порядку.

7) Видалити поле *pole1*.

8) Додати обмеження для поля *ocinka* таблиці *uspish*, що допускає ввід цілих значень від 1 до 5 включно або **NULL**.

9) Ввести дані у таблиці *vyklad*, *predmet* (не менше 6 записів) за допомогою команди **INSERT**.

```
INSERT INTO ім'я_таблиці [ (ім'я_поля  
[ , ім'я_поля ... ] ) ]  
VALUES (значення [ , значення ... ] );
```

10) Видалити запис про викладача, прізвище якого Бачишина.

11) Модифікувати у таблиці *predmet* запис про предмет бази даних, задавши семестр 5 і кількість годин 180.

12) Створити розрахункове поле *oplata*, в якому вираховується податок 13% від зарплати.

13) Видалити поле *oplata*.

14) Відкрити таблицю *student*, натиснувши 3-ю кнопку



. Ввести 7 значень у дану таблицю. Закрити таблицю.



15) Через **Form Editor** або за допомогою команди **Load Value From File** з контекстного меню клітинки таблиці, що має містити фото, завантажити фотографії для 1-2 студентів.

16) Відкрити таблицю `uspish`, ввести 15 значень у дану таблицю. Для цього можна використати **Form Editor**.

17) Завершити роботу.

Контрольні запитання:

- 1) Яка команда використовується для модифікації структури таблиці?
- 2) Які дії можна виконати за допомогою команди `ALTER TABLE . . .`?
- 3) Як ввести дані у таблицю?
- 4) Як змінити дані, які задовольняють певній умові?
- 5) Якою командою можна видалити дані з таблиці?
- 6) Як додати розрахункове поле в таблицю?
- 7) Чому не рекомендується створювати розрахункові поля у таблицях?

Лабораторна робота № 5-6

Тема: Прості запити. Групові операції. Використання агрегатних функцій.

Завдання до виконання:

- 1) Запустити *MySQL Workbench* і встановити підключення

до сервера на вкладці **MySQL Connections**



бічної панелі головного вікна.

2) На панелі **Navigator** ⇒ вкладка **Schemas** активувати базу даних `univer`, клацнувши два рази мишкою.

3) На вкладці **Query1** виконати наступні запити до бази даних `univer`, використовуючи команду `SELECT`. Результати роботи оформити у вигляді звіту.

```
SELECT [DISTINCT] перелік_полів
FROM перелік_таблиць
```

[WHERE умова]

[GROUP BY ім'я_поля [HAVING умова_для_групи]]

[ORDER BY перелік_полів];

1. Вивести на екран прізвища, імена та розмір стипендії всіх студентів. Дані відсортувати в алфавітному порядку по прізвищах.
2. Вивести розрахунок стипендії після підвищення її у два рази для кожного студента. Задати назву для нового стовпця `nstip`.
3. Вивести на екран прізвища та імена студентів, стипендія яких дорівнює 75.
4. Вивести на екран номери залікових книжок студентів, які не мають трійок по певному предмету.
5. Вивести інформацію про студентів, які мають 4 і 5 по певному предмету.
6. За допомогою оператора `IN` вивести прізвища студентів, імена яких Анатолій або Володимир.
7. Вивести інформацію про студентів, прізвища яких починаються з літер із діапазону «А» – «М».
8. Вивести номери залікових книжок студентів, які склали іспити від 1.06.2005 до 15.06.2005.
9. Вивести номери залікових книжок студентів, які склали іспити в червні.
10. Вивести номери залікових книжок студентів, які не принесли фото.
11. Вивести номери залікових книжок студентів, які не здали іспити (отримали незадовільні оцінки). Дані не повинні повторюватись.
12. Вивести список викладачів, прізвища яких починаються на літеру «Б».
13. Вивести список дисциплін, в назві яких є слово «бази».
14. Підрахувати кількість записів у таблиці `student`.
15. Підрахувати суму стипендій, яка виплачується на групу
16. Підрахувати кількість студентів, які отримують стипендію.
17. Вивести середній бал успішності по таблиці `uspish`.
18. Вивести мінімальну і максимальну оцінки для кожного студента по результатам сесії.

19. Вивести мінімальну позитивну оцінку для кожного студента.
 20. Підрахувати кількість студентів, які склали іспит з кожного предмета (на будь-яку оцінку).
 21. Підрахувати кількість студентів, які склали іспит з кожного предмета позитивно.
 22. Вивести середній бал кожного студента.
 23. Вивести список студентів (snom), середній бал яких >4,3.
 24. Підрахувати кількість студентів кожної форми навчання.
- 4) Завершити роботу.

Контрольні запитання:

- 1) Яка команда використовується для формування запитів?
- 2) Як задати сортування записів в запитах?
- 3) Як вибрати унікальні записи з таблиці?
- 4) Як задаються умови відбору для запитів?
- 5) Для чого використовується опція GROUP BY?
- 6) Які агрегатні функції ви знаєте?
- 7) Яка функція використовується для роботи з даними типу DATE?
- 8) Чим функція COUNT (*) відрізняється від COUNT (ім'я_поля) ?

Лабораторна робота № 7

Тема: Багатотабличні запити.

Завдання до виконання:

- 1) Запустити *MySQL Workbench* і встановити підключення

до сервера на вкладці **MySQL Connections**



бічної панелі головного вікна.

- 2) На панелі **Navigator** ⇒ вкладка **Schemas** активувати базу даних *univer*, клацнувши два рази мишкою.

3) На вкладці **Query1** виконати наступні запити до бази даних `univer`, використовуючи команду `SELECT`. Результати роботи оформити у вигляді звіту:

1. Вивести прізвища та імена студентів і викладачів, відсортувати їх в алфавітному порядку по прізвищу.
 2. До запиту 1 додати вивід стовпця `posada`, який включає значення «студент» і «викладач» відповідно для кожного запису.
 3. Вивести прізвища викладачів і назви предметів, які вони викладають.
 4. Вивести прізвища викладачів, назви предметів, які вони викладають, і кількість годин з кожної дисципліни, якщо кількість годин >100 .
 5. За допомогою команди `INSERT` доповнити таблицю `student` записами про двох нових студентів. Вивести прізвища усіх студентів і номери предметів, з яких вони склали іспити (лівостороннє об'єднання).
 6. Записати попередній запит з використанням правостороннього об'єднання.
 7. Вивести прізвища студентів і назви предметів, іспити з яких вони повинні скласти, тобто розклад іспитів (операція декартового добутку).
 8. Вивести прізвища студентів і назви предметів, іспити з яких вони склали (задати зв'язок між таблицями).
- 4) Завершити роботу.

Контрольні запитання:

- 1) Яке призначення оператора `UNION`?
- 2) Яким чином задається зв'язок між двома таблицями в запитах?
- 3) Які є різновидності оператора `JOIN`?
- 4) Які є види зв'язків між таблицями?

Лабораторна робота № 8

Тема: Багатотабличні запити. Вкладені запити.

Завдання до виконання:

- 1) Запустити *MySQL Workbench* і встановити підключення

до сервера на вкладці **MySQL Connections**  бічної панелі головного вікна.

- 2) На панелі **Navigator** ⇒ вкладка **Schemas** активувати базу даних *univer*, клацнувши два рази мишкою.

- 3) На вкладці **Query1** виконати наступні вкладені запити до бази даних *univer*, використовуючи команду `SELECT`. Результати роботи оформити у вигляді звіту:

1. Вивести інформацію про предмети, які викладає викладач Бачишина.
2. Вивести назви дисциплін і прізвища викладачів, які викладають їх і є доцентами.
3. Вивести прізвища студентів, які отримують максимальну стипендію (розмір максимальної стипендії наперед невідомий).
4. Вивести назви тих дисциплін, для вивчення яких відведена однакова кількість годин.
5. Використовуючи оператори `EXISTS`, `ANY`, `ALL`, `SOME`, сформувані запити, що реалізують наступні завдання:
 - 5.1. Вивести дані про студентів при умові, що існує викладач Бачишина у таблиці *vyklad*.
 - 5.2. Вивести дані про студентів, що мають незадовільні оцінки.
 - 5.3. Вивести прізвища тих студентів, що мають лише одну 2.
 - 5.4. Вивести дані про студентів, що здали всі іспити.
 - 5.5. Вивести прізвища тих викладачів, що читають більше, ніж один предмет.
 - 5.6. Визначити викладачів, які читають хоча б один курс лекцій.

- 5.7. Вивести прізвища всіх студентів, що здавали іспит у вказаний день.
 - 5.8. Вивести прізвища всіх студентів, які здавали іспити (без повторень).
 - 5.9. Вивести дані про тих студентів, чії оцінки вище від оцінок студента з номером 200107
- 4) Завершити роботу.

Контрольні запитання:

- 1) Що таке вкладені запити?
- 2) Які оператори використовуються в вкладених запитах?
- 3) Які підзапити є корельованими?
- 4) Яке значення є результатом виконання оператора EXISTS?
- 5) Чи можна використовувати оператори ANY, ALL з оператором =?

Лабораторна робота № 9


Тема: Представлення.

Завдання до виконання:

- 1) Запустити *MySQL Workbench* і встановити підключення

до сервера на вкладці **MySQL Connections**  бічної панелі головного вікна.

2) На панелі **Navigator** ⇒ вкладка **Schemas** активувати базу даних *univer*, клацнувши два рази мишкою по назві.

3) За допомогою кнопки **Create a new view ...**  створити представлення для виводу інформації з таблиці *predmet*, що містить номер предмета, назву, кількість годин. Задати нові назви для представлення і виводу полів. Зберегти представлення. Переглянути результати виконання.

4) Додати умову для виведення предметів з кількістю годин, більшою від 100, у дане представлення.

5) За допомогою команди `INSERT INTO` представлення ввести інформацію про предмет з кількістю 60 годин. Перевірити наявність нового предмета у таблиці `predmet`.

6) Змінити дане представлення, додавши умову `WITH CHECK OPTION`.

7) За допомогою команди `INSERT INTO` представлення ввести інформацію про предмет з кількістю 90 годин. Перевірити наявність нового предмета у таблиці `predmet`. Який висновок?

8) За допомогою команди `INSERT INTO` представлення ввести інформацію про предмет з кількістю 150 годин. Перевірити наявність нового предмета у таблиці `predmet`. Який висновок?

9) Створити немодифіковане представлення на власний розсуд.

10) Завершити роботу.

Контрольні запитання:

- 1) Що таке представлення?
- 2) Чим представлення відрізняються від запитів?
- 3) Чи може бути створене представлення на основі іншого представлення?
- 4) Чи зберігаються дані, отримані в представленнях, у базі даних?
- 5) Що являє собою модифікація представлень?
- 6) Які представлення є модифікованими?
- 7) Які умови накладаються на модифіковані представлення?
- 8) Як заборонити модифікацію представлень?

Лабораторна робота № 10

Тема: Збережені процедури.

Завдання до виконання:

- 1) Запустити *MySQL Workbench* і встановити підключення

до сервера на вкладці **MySQL Connections**



бічної панелі головного вікна.

2) На панелі **Navigator** ⇒ вкладка **Schemas** активувати базу даних `univer`, клацнувши два рази мишкою по назві.

3) За допомогою кнопки **Create a new stored procedure ...**



створити нову процедуру.

4) Створити процедуру, яка визначає мінімальну стипендію групи. Вивести результат у вихідний параметр **min_stip**.

5) Створити процедуру, що визначає прізвища студентів, які отримують максимальну стипендію в групі.

6) Створити процедуру, яка індексує стипендію на 0,25. (Коефіцієнт індексації може змінюватись). Задати коефіцієнт індексації як вхідний параметр.

7) Створити процедуру, що виводить прізвища студентів, які отримують вказану стипендію. Задати розмір стипендії в програмі як змінну.

8) Створити процедуру, яка обчислює сумарну кількість годин, прочитану кожним викладачем.

9) Створити процедуру, яка знищує з таблиці `student` запис по прізвищу Іванов.

10) Створити 2-3 процедури для бази даних `univer`.

11) Завершити роботу.

Контрольні запитання:

- 1) Що таке збережені процедури?
- 2) Якою командою створюються збережені процедури?
- 3) Які команди і оператори може включати тіло процедури?
- 4) Як описати локальний параметр?
- 5) Яким чином виконуються збережені процедури?

Лабораторна робота № 11

Тема: Збережені функції і тригери.


Завдання до виконання:

1) Запустити *MySQL Workbench* і встановити підключення

до сервера на вкладці **MySQL Connections**  бічної панелі головного вікна.

2) На панелі **Navigator** ⇒ вкладка **Schemas** активувати базу даних *univer*, клацнувши два рази мишкою по назві.

3) Виконати наступні завдання на тему «Функції» (Для створення функції можна скористатись кнопкою **Create a new**

function ...  панелі інструментів):

1. Створити функцію, яка переводить оцінку студента із 100-бальної (чи 5-бальної) шкали оцінювання в національну а) для іспитів («відмінно», «добре», «задовільно», «незадовільно»); б) для заліків («зараховано», «не зараховано»).
2. Створити функцію *hello*, що виводить привітання. Використати її для привітання а) усієї групи, б) кожного студента.
3. Створити самостійно 2-3 функції для бази даних *univer*. Перевірити їх виконання.

4) Виконати наступні завдання на тему «Тригери». (Для створення тригера необхідно відкрити структуру таблиці,

натиснувши 3-ю кнопку , і вибрати вкладку **Triggers**.

1. Створити тригер, який перетворює назви предметів таблиці *predmet* у великі літери при модифікації чи додаванні нових записів у дану таблицю.
2. Створити тригер, який при додаванні нового запису у таблицю *student* нараховує стипендію 1800 грн. студентам державної форми навчання.

3. Створити тригер, який при введенні недопустимої форми навчання при додаванні записів у таблицю `student`, видає повідомлення: «Введіть правильно форму навчання: платна або бюджет».
 4. Створити самостійно 2-3 тригера до будь-якої таблиці бази даних `univer`.
 5. Перевірити виконання тригерів.
- 5) Завершити роботу.

Контрольні запитання:

- 1) Що являє собою збережена функція?
- 2) Чим збережена функція відрізняється від процедури?
- 3) Що таке тригер?
- 4) Які є види тригерів?
- 5) Чи може тригер використовуватись як окрема програма?
- 6) До яких дій по відношенню до таблиць приводить виконання тригерів?
- 7) Які оператори використовуються в тригерах?
- 8) Який зміст мають змінні `NEW` та `OLD` в тілі тригера?
- 9) Для чого використовуються виключення?

Лабораторна робота № 12

Тема: Безпека бази даних. Користувачі, ролі, права.

Завдання до виконання:

- 1) Запустити *MySQL Workbench* і встановити підключення

до сервера на вкладці **MySQL Connections**



бічної панелі головного вікна.

2) На панелі **Navigator** ⇒ вкладка **Schemas** активувати базу даних `univer`, клацнувши два рази мишкою по назві.

3) Перейти на вкладку адміністрування за допомогою команди **Server** ⇒ **Users and Privileges**.

4) Створити новий акаунт, натиснувши кнопку **Add Account**.

5) Задати інформацію про користувача `st1` з паролем `pass1` на вкладці **Login**.


б) Вибрати вкладку **Administrative Roles**. Ознайомитись із можливими ролями і відповідними привілеями для користувачів баз даних. Задати права доступу на перегляд і вибірку даних **SELECT**. Автоматично створюється роль **Custom**, яку можна задавати і для інших користувачів. На вкладці **Schema Privileges** натиснути кнопку **Add Entry...**, вибрати перемикач **Selected schema**, додати базу даних **univer**, задати для користувача права доступу **SELECT**.

7) Встановити нове підключення **test1** до сервера на

вкладці **MySQL Connections**  бічної панелі головного вікна для користувача **st1**.

8) Виконати запит до будь-якої таблиці бази даних **univer**.

9) Додати запис до будь-якої таблиці бази даних **univer**. Проаналізувати результати.

10) Встановити підключення до сервера для користувача **root** на вкладці **MySQL Connections** .

11) У командному режимі на вкладці **Query** або **SQL File** створити користувача **st2** з паролем **pass2**. Надати йому права доступу на зміну до таблиці **predmet**.

12) Провести сеанс роботи з базою даних **univer** для користувача **st2**. Зробити висновок. Чи можна змінювати інші таблиці?

13) Перегляньте, які права мають користувачі **st1** і **st2** за допомогою команди **SHOW GRANTS**.

14) Відмінити права доступу для користувача **st2**.

15) Самостійно створити користувачів і ролі, надавши їм певні права. Переконайтесь у правильній роботі користувачів по відношенню до різних об'єктів бази даних.

16) Завершити роботу.

Контрольні запитання:

- 1) Який користувач має повний доступ до бази даних?
- 2) Як зареєструвати користувача у програмі *Workbench*?
- 3) Яка команда використовується для надання прав

користувачам?

- 4) Чи може користувач призначати свої права доступу до бази даних іншим користувачам?
- 5) Що таке ролі?
- 6) Для чого створюються ролі?
- 7) Який механізм використання ролі?
- 8) Які є зареєстровані ролі у програмі *Workbench*?
- 9) Чим роль відрізняється від реєстрації користувача?
- 10) Як відмінити права, надані користувачу?
- 11) Як знищити роль?

Інформаційні ресурси

- 1) Документація MySQL. URL: <https://dev.mysql.com/doc/>.
- 2) Посібник Workbench. URL: <https://dev.mysql.com/doc/workbench/en/wb-intro.html>.
- 3) Команди мови SQL. URL: <https://dev.mysql.com/doc/refman/8.4/en/sql-statements.html>.
- 4) Попелюха. Бази даних. URL: https://www.youtube.com/playlist?list=PL9mn2EBC_SSwi3XiORhAi75Hk1TNRidN1