

Міністерство освіти і науки України

Національний університет водного господарства та
природокористування

Навчально-науковий інститут енергетики, автоматики та водного гос-
подарства

Кафедра гідротехнічного будівництва та гідравліки

01-04-79М

Методичні вказівки

до практичних занять та самостійного вивчення
навчальної дисципліни

«Системи управління базами даних у водній інженерії»
для здобувачів вищої освіти першого (бакалаврського) рівня
за освітньо-професійною програмою «Гідротехнічне будівництво,
водна інженерія та водні технології» спеціальності 194
«Гідротехнічне будівництво, водна інженерія та водні технології»
всіх форм навчання

Рекомендовано науково-методичною
радою з якості ННІЕАВГ
Протокол № 2 від 29.10.2024 р.

Рівне – 2024

Методичні вказівки до практичних занять та самостійного вивчення навчальної дисципліни «**Системи управління базами даних у водній інженерії**» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Гідротехнічне будівництво, водна інженерія та водні технології» спеціальності 194 «Гідротехнічне будівництво, водна інженерія та водні технології» всіх форм навчання. [Електронне видання] / Новачок О. М. – Рівне : НУВГП, 2024. – 59 с.

Укладач: **Новачок О. М.**, кандидат сільськогосподарських наук, доцент кафедри гідротехнічного будівництва та гідравліки.

Відповідальний за випуск: **Волк Л. Р.**, кандидат технічних наук, доцент, в.о. завідувача кафедри гідротехнічного будівництва та гідравліки.

Гарант освітньо-професійної програми «Гідротехнічне будівництво, водна інженерія та водні технології» **Хлапук М. М.**, д.т.н., професор кафедри гідротехнічного будівництва та гідравліки.

© О. М. Новачок, 2024
© НУВГП, 2024

Зміст

| | |
|--|----|
| 1. Загальні вказівки | 3 |
| 1.1. Загальні рекомендації до вивчення дисципліни | 3 |
| 1.2. Мета та завдання навчальної дисципліни | 4 |
| 1.3. Рекомендована література | 4 |
| 2. Методичні рекомендації до практичних занять та самостійної роботи з навчальної дисципліни..... | 5 |
| 2.1. Теоретична частина..... | 5 |
| Тема 1: Вступ до систем управління базами даних (СУБД)..... | 5 |
| Тестові питання..... | 7 |
| Тема 2: Основи роботи з MariaDB | 7 |
| Тестові питання..... | 11 |
| Тема 3: Управління даними в водній інженерії | 11 |
| Тестові питання..... | 16 |
| Тема 4: Використання мови програмування R для аналізу даних..... | 16 |
| Тестові питання..... | 20 |
| Тема 5: Проектування та реалізація рішень для водної інженерії..... | 20 |
| Тестові питання..... | 24 |
| 2.2. Практична частина | 24 |
| Практичне заняття 1: Встановлення та налаштування операційної системи Linux (Ubuntu) та MariaDB..... | 24 |
| Практичне заняття 2: Основи SQL в MariaDB | 27 |
| Практичне заняття 3: Використання індексів та оптимізація запитів.... | 30 |
| Практичне заняття 4: Структура даних у водній інженерії | 34 |
| Практичне заняття 5: Підключення до бази даних з мови програмування R | 37 |
| Практичне заняття 6: Аналіз даних з використанням R..... | 41 |
| Практичне заняття 7: Створення звітів та візуалізація даних..... | 45 |
| Практичне заняття 8: Реплікація та резервне копіювання даних | 48 |
| Практичне заняття 9: Захист та безпека баз даних | 52 |
| Практичне заняття 10: Реалізація комплексного проекту..... | 56 |

1. Загальні вказівки

1.1. Загальні рекомендації до вивчення дисципліни

Освітній компонент Д41.5 «Системи управління базами даних у водній інженерії» є складовою частиною вибіркових компонентів освітньої програми для підготовки студентів за освітньою програмою «Гідротехнічне будівництво, водна інженерія та водні технології» спеціальності 194 «Гідротехнічне будівництво, водна інженерія та водні технології».

1.2. Мета та завдання навчальної дисципліни

Метою навчальної дисципліни «Системи управління базами даних у водній інженерії» є підготовка студентів до розробки та підтримки баз даних, що використовуються для управління водними ресурсами, моніторингу якості води, прогнозування повеней та інших аспектів водної інженерії.

Цей курс надасть студентам знання та навички, необхідні для проектування, розробки та управління базами даних у сфері водної інженерії. Основна увага приділяється роботі з операційною системою Linux (Ubuntu), СУБД MariaDB та мовою програмування R.

Після вивчення дисципліни студенти зможуть:

- встановлювати та налаштовувати операційну систему Linux (Ubuntu) та СУБД MariaDB;
- створювати, налаштовувати та керувати базами даних в MariaDB;
- виконувати запити до баз даних з використанням SQL;
- використовувати мову програмування R для аналізу даних з баз даних;
- розробляти рішення для задач водної інженерії, використовуючи СУБД та мову R..

1.3. Рекомендована література

Основна

1. Мікула М. П. Організація баз даних та знань : навчальний посібник для студентів спеціальності «Комп'ютерні науки» / Мікула М. П., Коцюк Ю. А., Мікула О. М. Острог : Вид-во НаУОА, 2021. 194 с.
2. David Clinton, Christopher Negus. Ubuntu Linux Bible. John Wiley & Sons, 2021. 755 p.
3. Ben Forta. MySQL Crash Course. Addison-Wesley Professional, 2024. 304 p.
4. Robert Kabacoff. Modern Data Visualization with R. CRC Pressr, 2024. 272 p.
5. Margot Tollefson. Visualizing Data in R 4: Graphics Using the base, graphics, stats, and ggplot2 Packages. Apress, 2021. 503 p.

Допоміжна

6. Практичне використання СУБД MySQL: навч. посіб. / Гавриленко В. В., Струневич Л. М. Київ : НТУ, 2017. 143 с.
7. Балик Н. Р., Мандзюк В.І. Бази даних MySQL : навчальний посібник. Тернопіль : Навчальна книга – Богдан, 2010. 160 с.
8. Пасічник В. В., Резниченко В. А. Організація баз даних та знань. К. : Видавнича група BHV, 2006. 384 с.

Інформаційні ресурси

9. Сайт розробників вільної системи управління базами даних MariaDB. URL: <https://mariadb.com/> :

10. Сайт розробників дистрибутиву Linux Ubuntu. URL: <https://ubuntu.com/>
11. Сайт розробників мови програмування R, програмного середовища для статистичних обчислень, аналізу та представлення даних в графічному вигляді. URL: <https://cran.r-project.org/>

2.Методичні рекомендації до практичних занять та самостійної роботи з навчальної дисципліни

2.1.Теоретична частина

Тема 1: Вступ до систем управління базами даних (СУБД)

1. Вступ

Сучасні інформаційні системи неможливі без надійного та ефективного управління даними. Для обробки великих обсягів даних у водній інженерії, як і в інших галузях, використовуються системи управління базами даних (СУБД). Вони допомагають автоматизувати процеси зберігання, обробки, доступу та аналізу даних, що має важливе значення для таких задач, як моделювання водних ресурсів, моніторинг стану екосистем, управління гідротехнічними спорудами та інше.

2. Поняття бази даних (БД) та СУБД

База даних (БД) — це організований набір даних, що зберігається на комп'ютері та використовується для зручного доступу, обробки та управління. Прикладом можуть бути дані про водні об'єкти, їх параметри, інформація про гідротехнічні споруди та природні ресурси.

Система управління базами даних (СУБД) — це програмне забезпечення, яке забезпечує управління базами даних і доступ до них. Основні функції СУБД:

- **Зберігання та маніпуляція даними.**
- **Забезпечення доступу до даних для декількох користувачів.**
- **Безпека та захист даних.**
- **Резервне копіювання та відновлення даних.**

3. Компоненти СУБД

Основні компоненти СУБД включають:

1. **Мова запитів:** Найчастіше використовується мова SQL (Structured Query Language) для виконання операцій над базою даних, таких як вибірка, вставка, оновлення та видалення даних.
2. **Диспетчер збереження даних:** Забезпечує управління фізичним зберіганням даних на диску.

3. **Диспетчер запитів:** Аналізує та виконує запити користувачів.
4. **Контроль доступу та безпеки:** Захищає базу даних від несанкціонованого доступу.

4. Типи баз даних

Розрізняють кілька типів баз даних залежно від способу організації даних:

1. **Реляційні бази даних (РБД):** Дані організовані у вигляді таблиць, які можуть бути пов'язані між собою. Це найбільш поширений тип БД, де використовуються SQL-запити для роботи з даними.

Приклад: база даних, що зберігає дані про гідрологічні вимірювання, де кожна таблиця представляє певний об'єкт (річка, водосховище, рівень води тощо).

2. **Об'єктно-орієнтовані бази даних:** Дані зберігаються у вигляді об'єктів, як у програмуванні. Цей тип баз даних підходить для складних структур даних, таких як моделі 3D-об'єктів.
3. **NoSQL бази даних:** Використовуються для зберігання великих масивів неструктурованих або слабо структурованих даних. Наприклад, підходять для зберігання даних з IoT пристроїв у системах моніторингу води.

5. Архітектура СУБД

СУБД можуть мати різні архітектурні рішення:

1. **Централізовані СУБД:** Всі дані та СУБД знаходяться на одному сервері. Це підходить для малих систем або локальних застосунків.
2. **Розподілені СУБД:** Дані зберігаються на кількох серверах, що розташовані в різних місцях. Це корисно для великих систем, які повинні працювати на значних географічних відстанях, що є важливим для глобальних проєктів з управління водними ресурсами.

6. Використання СУБД у водній інженерії

У водній інженерії СУБД допомагають вирішувати завдання збору та аналізу великих обсягів даних, таких як:

- Моніторинг якості води.
- Управління водосховищами та річковими системами.
- Моделювання водних потоків.
- Зберігання та аналіз інформації про гідротехнічні споруди.

Приклад використання: База даних для зберігання інформації про рівень води в річках, які вимірюються за допомогою датчиків. Дані можуть бути оброблені для прогнозування повеней або посух.

7. Переваги та недоліки СУБД

Переваги:

- Ефективне управління великими обсягами даних.
- Можливість роботи кількох користувачів одночасно.
- Підвищена безпека та цілісність даних.
- Резервне копіювання і відновлення.

Недоліки:

- Висока вартість впровадження та обслуговування.
- Необхідність кваліфікованого персоналу для управління СУБД.
- Складність налаштування та масштабування для специфічних потреб.

8. Висновки

СУБД є невід'ємною частиною сучасних інформаційних систем, включаючи ті, що використовуються у водній інженерії. Вони дозволяють ефективно управляти даними, забезпечують безпеку та надійність інформації, а також сприяють автоматизації важливих процесів у галузі. Майбутнє СУБД вбачає тісну інтеграцію з новими технологіями, такими як штучний інтелект та великі дані.

Тестові питання

1. Що таке база даних і як вона використовується у водній інженерії?
2. Які функції виконує система управління базами даних?
3. Перерахуйте основні типи баз даних і їхні особливості.
4. Як реляційні бази даних відрізняються від NoSQL баз даних?
5. Що таке централізовані та розподілені СУБД? Наведіть приклади їх використання.
6. Які переваги та недоліки використання СУБД у водній інженерії?
7. Як СУБД допомагає в управлінні водними ресурсами?
8. Чому важливо забезпечувати безпеку даних у СУБД?

Тема 2: Основи роботи з MariaDB

1. Вступ

MariaDB — це потужна, популярна і відкрита реляційна система управління базами даних (СУБД), яка була створена як відгалуження від MySQL. Вона підтримує використання SQL для виконання операцій над даними і використовується в різних галузях, включаючи водну інженерію, для зберігання та аналізу великих обсягів даних. У порівнянні з MySQL, MariaDB забезпечує вищу продуктивність, безпеку та додаткові функції, що робить її оптимальним вибором для складних проєктів з управління водними ресурсами.

2. Основні поняття MariaDB

СУБД **MariaDB** використовує реляційну модель даних, де дані організовані у вигляді таблиць. Кожна таблиця складається зі стовпців (атрибутів) та рядків (записів).

Основні елементи MariaDB:

- **Таблиця** — структура для зберігання даних, організованих у вигляді рядків та стовпців.
- **Запис (рядок)** — один елемент даних у таблиці, що містить значення для кожного стовпця.
- **Стовпець** — атрибут, що описує властивості або характеристики об'єкта (наприклад, ім'я, дата, показник рівня води).
- **Первинний ключ** — унікальний ідентифікатор запису в таблиці.
- **Зовнішній ключ** — атрибут, що використовується для створення зв'язків між таблицями.

3. Встановлення MariaDB

Для початку роботи з MariaDB необхідно встановити її на сервері або локальному комп'ютері. Більшість сучасних дистрибутивів Linux мають MariaDB у своїх репозиторіях, тож встановлення здійснюється через команду:

```
bashКопіювати код
```

```
sudo apt-get install mariadb-server
```

Після встановлення потрібно запустити сервер MariaDB та налаштувати початкові параметри безпеки:

```
bashКопіювати код
```

```
sudo systemctl start mariadb
```

```
sudo mysql_secure_installation
```

4. Підключення до MariaDB

Після встановлення, щоб підключитися до MariaDB через термінал, потрібно виконати команду:

```
bashКопіювати код
```

```
mysql -u root -p
```

Де:

- `-u root` — вказуємо користувача "root" (адміністратора).
- `-p` — система попросить ввести пароль адміністратора.

5. Основні операції в MariaDB

5.1. Створення бази даних

Для створення нової бази даних використовується команда:

```
sqlКопіювати код
```



```
CREATE DATABASE water_data;
```

Ця команда створить базу даних, яку можна використовувати для зберігання даних, наприклад, гідрологічних вимірювань.

5.2. Використання бази даних

Щоб почати працювати з конкретною базою даних, необхідно вибрати її командою:

```
sqlКопіювати код  
USE water_data;
```

5.3. Створення таблиці

Для створення таблиці, що зберігатиме, наприклад, дані про рівень води, використовується наступна команда:

```
sqlКопіювати код  
CREATE TABLE water_levels (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    river_name VARCHAR(50),  
    measurement_date DATE,  
    water_level FLOAT  
);
```

Таблиця `water_levels` містить такі поля:

- **id** — унікальний ідентифікатор для кожного запису.
- **river_name** — назва річки.
- **measurement_date** — дата вимірювання.
- **water_level** — рівень води.

5.4. Вставка даних

Для додавання даних у таблицю використовується команда `INSERT`:

```
sqlКопіювати код  
INSERT INTO water_levels (river_name, measurement_date,  
water_level)  
VALUES ('Dnipro', '2024-10-05', 4.5);
```

Ця команда додає новий запис з інформацією про рівень води в річці Дніпро.

5.5. Вибірка даних

Для вибірки даних з таблиці використовується команда `SELECT`. Наприклад, для отримання всіх записів:

```
sqlКопіювати код  
SELECT * FROM water_levels;
```

Для фільтрації результатів використовується умова `WHERE`:

```
sqlКопіювати код
```

```
SELECT * FROM water_levels WHERE river_name = 'Dnipro';
```

Ця команда повертає записи лише для річки Дніпро.

5.6. Оновлення даних

Щоб оновити існуючий запис, використовується команда UPDATE:

```
sqlКопіювати код
UPDATE water_levels
SET water_level = 5.0
WHERE river_name = 'Dnipro' AND measurement_date =
'2024-10-05';
```

5.7. Видалення даних

Для видалення записів з таблиці використовується команда DELETE:

```
sqlКопіювати код
DELETE FROM water_levels
WHERE river_name = 'Dnipro' AND measurement_date =
'2024-10-05';
```

6. Резервне копіювання та відновлення

Однією з важливих операцій для безпеки даних є резервне копіювання бази даних. У MariaDB це можна зробити за допомогою утиліти **mysqldump**:

```
bashКопіювати код
mysqldump -u root -p water_data > water_data_backup.sql
```

Для відновлення даних із резервної копії використовується команда:

```
bashКопіювати код
mysql -u root -p water_data < water_data_backup.sql
```

7. Безпека та контроль доступу

MariaDB забезпечує потужні інструменти для управління доступом до даних.

Для створення нового користувача з обмеженими правами використовують наступні команди:

```
sqlКопіювати код
CREATE USER 'user1'@'localhost' IDENTIFIED BY
'password';
GRANT SELECT ON water_data.* TO 'user1'@'localhost';
```

Ці команди створюють користувача **user1** з правами доступу тільки на вибірку даних з бази **water_data**.

8. Оптимізація та продуктивність

Для великих баз даних у водній інженерії важливо оптимізувати роботу запитів. Використання індексів на ключових полях таблиці дозволяє прискорити

пошук даних. Наприклад, індексація стовпця **measurement_date**:

```
sqlКопіювати код
CREATE INDEX idx_date ON water_levels
(measurement_date);
```

9. Висновок

MariaDB є надійною системою для управління базами даних у водній інженерії, дозволяючи ефективно зберігати, обробляти та аналізувати дані. Завдяки відкритій ліцензії та широким можливостям оптимізації, вона підходить для вирішення різноманітних задач, від моніторингу водних ресурсів до моделювання екосистем.

Тестові питання

1. Що таке MariaDB і які її основні функції?
2. Як встановити MariaDB на операційну систему Linux?
3. Яка команда використовується для підключення до MariaDB через термінал?
4. Як створити нову базу даних у MariaDB? Наведіть приклад.
5. Які типи даних використовуються при створенні таблиць у MariaDB?
6. Яка команда використовується для вибірки всіх даних з таблиці?
7. Як створити індекс у таблиці і для чого він використовується?
8. Як виконати резервне копіювання бази даних у MariaDB?
9. Які команди використовуються для створення нового користувача з обмеженими правами доступу?
10. Як оптимізувати запити до бази даних у MariaDB?

Тема 3: Управління даними в водній інженерії

1. Вступ

Управління даними є одним з ключових компонентів сучасної водної інженерії. Водні системи, як правило, генерують великі обсяги даних з різних джерел: датчики рівня води, моніторинг якості води, гідрологічні дослідження, інформація про гідротехнічні споруди тощо. Ефективне зберігання, аналіз і використання цих даних допомагає приймати правильні управлінські рішення, прогнозувати надзвичайні ситуації (повені, посухи), планувати використання водних ресурсів і здійснювати контроль за їх якістю.

2. Джерела даних у водній інженерії

Основними джерелами даних у водній інженерії є:

1. Датчики та IoT-пристрої:

- Вимірювання рівня води в річках і водосховищах.
- Датчики для моніторингу якості води (рН, температура, кількість

забруднюючих речовин).

- Метеорологічні дані (опади, температура повітря).

2. Гідрологічні дослідження:

- Збір даних про водний баланс, характеристики водних об'єктів (потoki, обсяги).
- Оцінка стану річок, водосховищ, підземних вод і водоносних горизонтів.

3. Гідротехнічні споруди:

- Інформація про роботу гребель, шлюзів, водоскидних споруд.
- Дані про стан і обслуговування інфраструктури.

4. Соціально-економічні дані:

- Використання води у сільському господарстві, промисловості та побуті.
- Дані про водопостачання та водовідведення.

3. Роль баз даних в управлінні водними ресурсами

Бази даних є основним інструментом для організації, зберігання та аналізу даних у водній інженерії. Вони дозволяють інтегрувати дані з різних джерел та забезпечують наступні можливості:

- **Зберігання великих обсягів даних:** Інформація, зібрана з датчиків або отримана в результаті гідрологічних досліджень, зберігається у структурованій формі.
- **Цілісність та безпека даних:** Завдяки системам управління базами даних (СУБД), інформація захищена від втрат і несанкціонованого доступу.
- **Масштабованість:** Сучасні бази даних дозволяють працювати з великими масивами інформації і можуть бути легко масштабовані для підтримки додаткових джерел даних або нових користувачів.
- **Аналіз і моделювання:** СУБД можуть бути інтегровані з аналітичними інструментами для моделювання водних потоків, прогнозування повеней та інших природних явищ.

4. Типи даних у водній інженерії

У водній інженерії використовуються різні типи даних, що можуть бути збережені в базах даних:

1. Числові дані:

- Рівень води, температура води, кількість опадів, швидкість потоку тощо.

2. Геопросторові дані:

- Координати річок, водосховищ, зон затоплення та водозбірних басейнів.

3. Часові ряди:

- Дані, які змінюються з часом, наприклад, рівень води протягом року або доби.

4. Текстові дані:

- Опис стану водних об'єктів, дані про якість води.

5. Зображення та відео:

- Аерофотознімки, супутникові знімки, зображення з камер спостереження.

5. Процеси управління даними

5.1. Збір даних

Збір даних — це перший етап управління даними. Він може відбуватися автоматично через датчики або вручну, під час польових досліджень. Важливо забезпечити стандартизацію даних, щоб вони могли бути легко інтегровані в базу даних.

5.2. Зберігання даних

Дані, що зібрані з різних джерел, повинні зберігатися у надійних базах даних, таких як MariaDB, PostgreSQL або спеціалізовані геопросторові бази, наприклад, PostGIS. СУБД забезпечують організацію даних, їхнє швидке зберігання та пошук.

5.3. Обробка та аналіз даних

Для прийняття рішень в водній інженерії важливо мати інструменти для аналізу даних. Це може бути статистичний аналіз, моделювання гідрологічних процесів або візуалізація даних у вигляді графіків та карт.

5.4. Візуалізація даних

Візуалізація допомагає перетворити числові дані в зрозумілі для інженерів, керівників та громадськості форми. Це можуть бути графіки змін рівня води, карти ризику затоплення або інші візуальні інструменти.

5.5. Прийняття рішень

На основі оброблених даних приймаються рішення щодо управління водними ресурсами. Наприклад, можна оптимізувати водоскиди з водосховищ

для запобігання повеням або регулювати водопостачання під час посух.

6. Виклики управління даними у водній інженерії

6.1. Великий обсяг даних

Управління водними ресурсами вимагає обробки великих масивів даних, що можуть надходити в реальному часі з безлічі джерел. Це створює потребу в потужних обчислювальних ресурсах та оптимізованих базах даних.

6.2. Якість даних

Дані можуть бути неповними, містити помилки або суперечності. Необхідно здійснювати очищення даних перед їх використанням для аналізу та прийняття рішень.

6.3. Інтеграція даних

Інформація може надходити з різних систем і в різних форматах (текст, цифри, геопросторові дані). Інтеграція даних з різних джерел є важливим завданням для побудови цілісної картини стану водних ресурсів.

7. Інструменти для управління даними у водній інженерії

Для управління даними використовуються різноманітні інструменти та технології:

1. СУБД (системи управління базами даних):

- MariaDB, PostgreSQL, Oracle — для зберігання та управління великими масивами даних.

2. Геоінформаційні системи (ГІС):

- ArcGIS, QGIS — для зберігання та обробки геопросторових даних.

3. Інструменти аналітики та візуалізації:

- Python, R, Excel, Power BI — для аналізу даних та побудови графіків.

4. Обчислювальні платформи:

- Cloud-платформи (Amazon Web Services, Google Cloud) дозволяють зберігати великі обсяги даних і забезпечують масштабовані ресурси для обробки і аналізу даних у водній інженерії.

8. Приклади використання управління даними в водній інженерії

8.1. Прогнозування повеней

Дані про рівень води, кількість опадів та стан річкових басейнів можуть бути інтегровані в моделі для прогнозування можливих повеней. За допомогою баз даних і гідрологічних моделей можна передбачити зміни рівня води і своєчасно вжити заходів для запобігання катастрофам.

8.2. Управління водосховищами

Сучасні системи управління водосховищами використовують дані в режимі реального часу для оптимізації роботи шлюзів і водоскидних споруд. Наприклад, у випадку високого рівня води можна автоматично регулювати обсяги скиду води для запобігання затопленням.

8.3. Моніторинг якості води

Дані, отримані від датчиків якості води (рН, температура, концентрація забруднюючих речовин), дозволяють виявляти потенційні проблеми з забрудненням і своєчасно реагувати на них. Аналітика на основі таких даних може допомогти визначити джерела забруднень і розробити плани їх усунення.

8.4. Підземні води та їх моніторинг

Дані про рівень підземних вод з різних свердловин можуть бути використані для оцінки їхнього стану і прогнозування змін у водоносних горизонтах. Це важливо для забезпечення стабільного водопостачання, особливо у посушливих регіонах.

9. Перспективи управління даними в водній інженерії

З розвитком технологій, управління даними в водній інженерії стає ще більш ефективним і точним. Інтеграція штучного інтелекту та машинного навчання відкриває нові можливості для прогнозування природних явищ та автоматизації процесів управління водними ресурсами.

9.1. Штучний інтелект та машинне навчання

Системи штучного інтелекту можуть аналізувати історичні дані і передбачати природні явища, такі як повені, з більшою точністю. Це дозволяє створювати більш ефективні системи управління ризиками.

9.2. Хмарні технології

Хмарні платформи стають все більш популярними для зберігання і обробки

великих обсягів даних. Вони дозволяють отримувати доступ до даних з будь-якої точки світу і забезпечують високу продуктивність для обробки в реальному часі.

9.3. Інтернет речей (IoT)

З розвитком технологій IoT управління даними у водній інженерії стане ще більш автоматизованим. Датчики зможуть збирати дані в реальному часі, а системи штучного інтелекту — автоматично приймати рішення на основі цих даних.

10. Висновок

Ефективне управління даними є критичним компонентом у сучасній водній інженерії. Воно дозволяє не тільки зберігати й аналізувати великі обсяги даних, але й приймати обґрунтовані рішення для сталого управління водними ресурсами. Використання сучасних інструментів, таких як бази даних, геоінформаційні системи та технології штучного інтелекту, допомагає вдосконалити процеси моніторингу, аналізу та управління водними системами, зменшувати ризики та сприяти збереженню навколишнього середовища.

Тестові питання

1. Які основні джерела даних використовуються у водній інженерії?
2. Які типи даних можуть бути збережені у базах даних для водної інженерії?
3. Як бази даних сприяють прийняттю рішень в управлінні водними ресурсами?
4. Які етапи управління даними в водній інженерії?
5. Чому важлива стандартизація даних при їх зборі?
6. Які технології використовуються для обробки та аналізу великих обсягів даних?
7. Яка роль візуалізації даних у водній інженерії?
8. Які основні виклики стоять перед управлінням даними в водній інженерії?
9. Як Інтернет речей (IoT) впливає на управління водними ресурсами?
10. Які перспективи використання штучного інтелекту для аналізу водних даних?

Тема 4: Використання мови програмування R для аналізу даних

1. Вступ

Мова програмування R — один із найпотужніших інструментів для аналізу даних, особливо в наукових дослідженнях, статистиці та інженерних дисциплінах. У водній інженерії R може бути використана для обробки великих обсягів даних, аналізу тимчасових рядів, створення моделей і візуалізації даних.

Важливою перевагою R є його велика кількість спеціалізованих пакетів, які полегшують роботу з даними, а також інтеграція з базами даних і геоінформаційними системами (ГІС).

2. Основи мови R

2.1. Встановлення та середовища розробки

Мову R можна завантажити з офіційного сайту R Project. Для зручної роботи часто використовують інтегроване середовище розробки RStudio, яке надає зручний інтерфейс для написання коду, роботи з графіками та керування пакетами.

2.2. Основні поняття

- **Об'єкти та типи даних:** У R всі дані зберігаються у вигляді об'єктів. Основними типами даних є вектори, матриці, списки та фрейми даних.
- **Оператори:** Включають арифметичні операції, логічні оператори та операції з індексацією.
- **Функції:** В R існує безліч вбудованих функцій для роботи з даними, а також можливість створювати власні функції.

2.3. Основні команди для роботи з даними

- **Зчитування даних:** `read.csv()`, `read.table()`, `readxl::read_excel()`.
- **Маніпуляція даними:** `dplyr`, `tidyr` — популярні пакети для обробки та структурування даних.
- **Статистичний аналіз:** Основні функції для обчислення середніх значень, стандартних відхилень, регресій та кореляцій.

3. Аналіз даних у водній інженерії з використанням R

R активно застосовується для аналізу різноманітних типів даних у водній інженерії, таких як тимчасові ряди (рівень води, опади), гідрологічні моделі, оцінка якості води тощо. Нижче наведені основні інструменти, які часто використовуються в цій сфері.

3.1. Обробка тимчасових рядів

Для водної інженерії важливо аналізувати зміни в часі, наприклад рівень води у водосховищі або кількість опадів. У R існують спеціальні бібліотеки для обробки таких даних.

- **Пакет `xts` та `zoo`:** Використовуються для роботи з часовими

рядами. Наприклад, можна створити об'єкт тимчасового ряду і виконати аналіз тенденцій у змінах рівня води.

```
rКопіювати код
library(xts)
data <- xts(c(10, 20, 15, 30, 25),
order.by=as.Date(c('2024-01-01', '2024-02-01',
'2024-03-01', '2024-04-01', '2024-05-01')))
plot(data, main="Зміна рівня води за часом")
```

3.2. Статистичний аналіз даних

R надає широкий вибір статистичних інструментів для аналізу даних:

- **Описова статистика:** Обчислення середнього, медіани, стандартного відхилення для даних рівня води або забруднень у воді.

```
rКопіювати код
summary(data) # Статистична зведена таблиця
sd(data) # Стандартне відхилення
```

- **Кореляція та регресія:** Ці методи можуть бути корисними для дослідження залежностей між різними параметрами водної системи, наприклад, залежність між кількістю опадів і рівнем води.

```
rКопіювати код
cor(data$rain, data$water_level) # Кореляція
model <- lm(water_level ~ rain, data = dataset)
# Лінійна регресія
summary(model)
```

3.3. Візуалізація даних

Візуалізація — важливий етап аналізу даних. У R є потужні інструменти для створення графіків та карт.

- **Базові графіки:** Можна створювати графіки змін рівня води, температури або інших параметрів.

```
rКопіювати код
plot(data, type="l", col="blue", main="Зміни рівня води",
xlab="Дата", ylab="Рівень води")
```

- **Пакет ggplot2:** Для створення більш складних та інформативних графіків. Наприклад, можна візуалізувати тенденції в змінах якості води або залежності між різними параметрами.

```
rКопіювати код
library(ggplot2)
ggplot(dataset, aes(x=rain, y=water_level)) +
```

```
geom_point() +
geom_smooth(method="lm") +
labs(title="Залежність між кількістю опадів і
рівнем води", x="Кількість опадів", y="Рівень
води")
```

3.4. Геоінформаційний аналіз

R також підтримує роботу з геопросторовими даними, що є важливим для водної інженерії. Існує кілька пакетів для обробки географічних даних:

- **sf**: Пакет для роботи з просторовими даними, який дозволяє працювати з координатами річок, водосховищ, зон затоплення тощо.

```
rКопіювати код
library(sf)
water_data <- st_read("shapefiles/rivers.shp")
plot(water_data)
```

4. Переваги використання R для аналізу даних у водній інженерії

1. **Відкрита платформа:** R є безкоштовною та активно підтримуваною спільнотою інженерів і науковців, що забезпечує наявність великої кількості пакетів для аналізу даних.
2. **Підтримка великих обсягів даних:** R може працювати з великими масивами даних, що є особливо важливим для інженерних задач.
3. **Інтеграція з базами даних:** R легко інтегрується з різними базами даних, що дозволяє автоматизувати зчитування та аналіз даних у реальному часі.
4. **Потужні можливості для візуалізації:** Можливість створювати графіки, карти та інші візуальні репрезентації даних робить R незамінним інструментом для інженерів.
5. **Гнучкість:** R підтримує широкий спектр статистичних і аналітичних методів, що дозволяє адаптувати аналіз під конкретні завдання водної інженерії.

5. Приклади використання R у водній інженерії

1. **Моделювання рівня води:** Використання тимчасових рядів для прогнозування змін рівня води у водосховищах або річках.
2. **Аналіз якості води:** Візуалізація та аналіз параметрів якості води (рівень забруднень, хімічний склад) для оцінки екологічного стану водних ресурсів.
3. **Прогнозування повеней:** Використання даних про кількість опадів і

рівень води для прогнозування ризиків затоплення.

4. **Оцінка водних ресурсів:** Статистичний аналіз даних про споживання води, метеорологічні умови та гідрологічні характеристики.

6. Висновок

R — це потужний інструмент для аналізу та моделювання даних у водній інженерії. Його можливості у статистичному аналізі, візуалізації та роботі з великими обсягами даних дозволяють вирішувати широке коло завдань, пов'язаних з управлінням водними ресурсами та моніторингом їх якості. Інженери та дослідники, які освоюють R, отримують ефективний засіб для покращення точності аналізу та прийняття рішень.

Тестові питання

1. Які основні типи даних використовуються у R для обробки та аналізу?
2. Як у R можна зчитати дані з файлу CSV?
3. Які пакети використовуються для обробки тимчасових рядів у R?
4. Що таке `ggplot2` і як його використовують у візуалізації даних?
5. Які функції можна використовувати для описової статистики в R?
6. Як у R виконується регресійний аналіз?
7. Яка роль пакету `dplyr` у маніпуляції даними?
8. Як інтегрувати дані про водні ресурси з геоінформаційними системами у R?
9. Які переваги використання R у водній інженерії порівняно з іншими мовами програмування?
10. Як можна використовувати R для прогнозування природних явищ, таких як повені?

Тема 5: Проектування та реалізація рішень для водної інженерії

1. Вступ

Водна інженерія відіграє важливу роль у забезпеченні стійкого управління водними ресурсами, захисту від повеней, очищення води, гідроенергетики та багато іншого. Успішна реалізація таких проектів вимагає ефективного використання інформаційних систем, баз даних і технологій для обробки та аналізу великих обсягів даних. Сучасні рішення для водної інженерії включають інтеграцію баз даних, ГІС, інтернет речей (IoT), хмарних платформ і алгоритмів штучного інтелекту. Вони допомагають не тільки моніторити стан водних ресурсів, але й прогнозувати природні явища, приймати рішення на основі даних і автоматизувати процеси управління.

2. Вимоги до проектування рішень у водній інженерії

2.1. Функціональні вимоги

Функціональні вимоги визначають основні завдання, які повинна виконувати система. Для водної інженерії це можуть бути:

- **Моніторинг рівня води:** Збір та аналіз даних про рівень води у водосховищах, річках, підземних водах тощо.
- **Моніторинг якості води:** Оцінка хімічних і фізичних характеристик води, таких як рівень рН, температура, концентрація забруднюючих речовин.
- **Прогнозування природних явищ:** Моделі прогнозування повеней, посух або інших екстремальних подій.
- **Управління водними ресурсами:** Оптимізація використання води для іригації, водопостачання та гідроенергетики.

2.2. Нефункціональні вимоги

Ці вимоги описують характеристики системи, такі як продуктивність, масштабованість, надійність та безпека:

- **Масштабованість:** Система повинна підтримувати зростання обсягів даних у міру збільшення кількості джерел даних (датчиків, користувачів тощо).
- **Надійність:** Важливо, щоб система працювала безперебійно, адже будь-які помилки можуть призвести до значних фінансових або екологічних збитків.
- **Безпека:** Оскільки водні ресурси є критичними для інфраструктури, захист даних від несанкціонованого доступу є пріоритетом.

3. Архітектура рішень для водної інженерії

3.1. Системи управління базами даних (СУБД)

Основою будь-якого рішення для водної інженерії є база даних. СУБД використовуються для зберігання та управління різноманітними типами даних:

- **Геопросторові дані:** Включають карти водних ресурсів, річок, водосховищ та інші об'єкти.
- **Темпоральні дані:** Дані, що змінюються з часом, наприклад рівень води, кількість опадів або показники якості води.
- **Метеорологічні дані:** Інформація про погодні умови, яка є важливою для прогнозування змін у водних системах.

Серед популярних рішень для баз даних у водній інженерії — PostgreSQL з підтримкою геопросторового розширення PostGIS або MariaDB.

3.2. Геоінформаційні системи (ГІС)

ГІС використовуються для візуалізації та аналізу географічних даних. Водні інженери можуть використовувати ГІС для створення карт водосховищ, річок, зон затоплення або для визначення зон ризику при повенях:

- **ArcGIS** або **QGIS** — це популярні програмні продукти для інтеграції з базами даних і виконання просторових аналізів.
- **Інтеграція з СУБД:** ГІС-системи можуть підключатися до баз даних і використовувати дані для візуалізації та аналізу.

3.3. Хмарні платформи

Хмарні технології дозволяють масштабувати системи управління водними ресурсами та забезпечувати доступ до даних у будь-який час і з будь-якого місця:

- **Microsoft Azure, Amazon Web Services (AWS)** та **Google Cloud** надають потужні інструменти для зберігання та обробки даних у хмарі.
- **Інтеграція IoT:** Хмарні платформи дозволяють збирати дані з IoT-пристроїв у реальному часі, що важливо для систем моніторингу.

3.4. Інтернет речей (IoT)

IoT технології допомагають автоматизувати збір даних у водній інженерії. Наприклад, датчики, встановлені у водосховищах або річках, можуть автоматично передавати дані про рівень води, якість води та інші параметри:

- **Датчики якості води:** Можуть відстежувати рівень забруднень, температуру, кисневий баланс та інші показники.
- **Метеорологічні датчики:** Допомагають вимірювати кількість опадів, швидкість вітру, температуру повітря та інші фактори, які можуть впливати на стан водних ресурсів.

3.5. Штучний інтелект і машинне навчання

Алгоритми штучного інтелекту (ШІ) та машинного навчання стають невід'ємною частиною рішень для водної інженерії. Вони допомагають прогнозувати природні явища та оптимізувати управління водними ресурсами:

- **Прогнозування повеней:** Моделі машинного навчання можуть аналізувати історичні дані про рівень води і опади для прогнозування можливих повеней.
- **Оптимізація використання водних ресурсів:** ШІ може допомогти автоматизувати процеси управління водними ресурсами для зменшення споживання води або покращення її якості.

4. Проектування баз даних для водної інженерії

4.1. Моделювання даних

Моделювання даних включає створення концептуальної та логічної моделей, які відображають структуру даних і взаємозв'язки між ними. Для водної інженерії важливо враховувати:

- **Ієрархічну структуру:** Наприклад, річкові басейни складаються з приток, кожен із яких має власні показники рівня води та якості.
- **Відношення між об'єктами:** Наприклад, водосховища можуть бути пов'язані з річками, а датчики з об'єктами моніторингу.

4.2. Створення фізичної моделі бази даних

Фізична модель визначає конкретну структуру бази даних та її реалізацію на сервері. Для водної інженерії це включає:

- **Таблиці:** Окремі таблиці для географічних об'єктів (річки, водосховища), даних датчиків (рівень води, якість води), метеорологічних даних тощо.
- **Зв'язки:** Використання зовнішніх ключів для визначення відношень між об'єктами (наприклад, річка – водосховище).
- **Індекси:** Для оптимізації запитів, наприклад для швидкого пошуку даних про рівень води за датами.

5. Реалізація рішень для водної інженерії

5.1. Збір та інтеграція даних

На етапі реалізації важливо забезпечити надійний збір і інтеграцію даних:

- **Збір даних із датчиків:** Встановлення датчиків IoT для моніторингу рівня води, якості води або погодних умов.
- **Інтеграція з ГІС:** Дані з датчиків можуть бути інтегровані з геоінформаційними системами для візуалізації результатів.
- **Зовнішні джерела:** Дані метеорологічних агентств або супутникові дані можуть бути використані для додаткового аналізу.

5.2. Обробка та аналіз даних

Реалізація аналітичних інструментів дозволяє проводити глибокий аналіз даних:

- **Обробка тимчасових рядів:** Аналіз змін рівня води за певний період.
- **Просторовий аналіз:** Оцінка впливу географічного розташування на якість або кількість водних ресурсів.
- **Прогнозування:** Використання моделей машинного навчання для

прогнозування екстремальних подій, таких як повені.

6. Висновки

Проектування та реалізація рішень для водної інженерії є комплексним процесом, що включає використання різних технологій: баз даних, геоінформаційних систем, хмарних платформ, IoT і алгоритмів штучного інтелекту. Такі рішення допомагають забезпечити ефективне управління водними ресурсами, що є важливим для екологічної стійкості та безпеки суспільства. Інтеграція сучасних інструментів дозволяє інженерам не тільки моніторити стан водних ресурсів, але й приймати оптимальні рішення на основі аналізу даних.

Тестові питання

1. Які функціональні вимоги до рішень для водної інженерії?
2. Що таке нефункціональні вимоги і як вони впливають на проектування рішень?
3. Які системи управління базами даних найчастіше використовуються у водній інженерії?
4. Що таке ГІС, і як її використовують у водній інженерії?
5. Які переваги використання хмарних платформ для рішень водної інженерії?
6. Як технології IoT можуть бути застосовані для моніторингу водних ресурсів?
7. Як штучний інтелект допомагає у прогнозуванні природних явищ?
8. Які основні етапи проектування баз даних для водної інженерії?
9. Яка роль просторового аналізу в управлінні водними ресурсами?
10. Які дані необхідні для прогнозування повеней за допомогою машинного навчання?

2.2. Практична частина

Практичне заняття 1: Встановлення та налаштування операційної системи Linux (Ubuntu) та MariaDB

Мета заняття:

Ознайомитися з основними етапами встановлення операційної системи Linux (Ubuntu) та СУБД MariaDB, а також налаштувати середовище для подальшої роботи з базами даних у водній інженерії.

Теоретична основа:

Linux — це вільно поширювана операційна система з відкритим вихідним кодом, яка широко використовується для серверних і наукових рішень завдяки своїй стабільності та гнучкості. Однією з найпопулярніших версій є **Ubuntu** — дружня до користувача дистрибуція Linux, яка підходить для розробки,

досліджень та інженерних задач.

MariaDB — це потужна реляційна система управління базами даних, яка є форком MySQL. Вона забезпечує високу продуктивність та зручність управління базами даних, що робить її ідеальною для використання у проєктах, пов'язаних з аналізом даних у водній інженерії.

Необхідні інструменти та матеріали:

- Комп'ютер з доступом до інтернету
- Дистрибутив операційної системи Ubuntu (сайт для завантаження: <https://ubuntu.com/download>)
- Інструкції для встановлення MariaDB (<https://mariadb.org/download/>)

Завдання заняття:

1. Встановлення операційної системи Ubuntu.

- Ознайомитися з процесом встановлення Ubuntu на комп'ютері або віртуальній машині.
- Налаштувати базові параметри системи (користувачі, мова, мережеві налаштування).
- Ознайомитися з основними командами Linux (термінал, навігація файловою системою, пакетний менеджер).

2. Оновлення системи і встановлення пакетів.

- Оновити список пакетів і встановити оновлення:

```
bashКопіювати код
sudo apt update
sudo apt upgrade
```
- Встановити базові пакети для подальшої роботи:

```
bashКопіювати код
sudo apt install build-essential curl wget
```

3. Встановлення MariaDB.

- Виконати встановлення MariaDB за допомогою пакетного менеджера APT:

```
bashКопіювати код
sudo apt install mariadb-server
```
- Перевірити статус сервера MariaDB:

```
bashКопіювати код
sudo systemctl status mariadb
```

4. Налаштування MariaDB.

- Виконати початкове налаштування MariaDB:

```
bashКопіювати код
sudo mysql_secure_installation
```
- Створити адміністративного користувача та встановити пароль для root:
 - Використати вбудовані параметри для налаштування безпеки (видалити тестові бази, вимкнуті анонімних користувачів тощо).

5. Підключення до MariaDB через термінал.

- Запустити клієнт MySQL і підключитися до сервера:

```
bashКопіювати код
sudo mysql -u root -p
```
- Створити нову базу даних для майбутнього використання:

```
sqlКопіювати код
CREATE DATABASE water_engineering_db;
```

6. Тестування встановлення.

- Створити таблицю для зберігання даних у MariaDB:

```
sqlКопіювати код
USE water_engineering_db;
CREATE TABLE example_table (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    value FLOAT
);
```
- Додати кілька записів у таблицю:

```
sqlКопіювати код
INSERT INTO example_table (name, value) VALUES
('Water Level', 23.5), ('Flow Rate', 7.2);
```
- Перевірити, чи дані були успішно додані:

```
sqlКопіювати код
SELECT * FROM example_table;
```

7. Завдання для самостійної роботи:

- Ознайомитися з системними журналами та перевірити, чи були помилки під час роботи MariaDB.
- Налаштувати автоматичний запуск MariaDB при завантаженні системи:

```
bashКопіювати код
```

```
sudo systemctl enable mariadb
```

Підсумки:

Після виконання завдань студенти повинні розуміти основи роботи з операційною системою Linux (Ubuntu) і базові принципи налаштування та використання СУБД MariaDB. Ці навички є важливими для подальшого ефективного управління даними в інженерних проєктах, пов'язаних з водними ресурсами.

Запитання для самоперевірки:

1. Які команди використовуються для оновлення системи в Ubuntu?
2. Як встановити MariaDB на Ubuntu?
3. Як налаштувати сервер MariaDB після встановлення?
4. Які кроки необхідно виконати для створення нової бази даних і таблиці?
5. Як підключитися до MariaDB через термінал?

Додаткові матеріали:

- Офіційна документація Ubuntu
- Офіційна документація MariaDB

Ця практична робота є основою для подальшої роботи з базами даних у водній інженерії, а також навчить студентів ефективно налаштовувати середовище для управління великими обсягами даних.

Практичне заняття 2: Основи SQL в MariaDB

Мета заняття:

Ознайомитися з базовими командами SQL для роботи з базами даних у MariaDB та навчитися виконувати основні операції, такі як створення баз даних, таблиць, внесення та вибірка даних. Це практичне заняття спрямоване на розвиток навичок, необхідних для роботи з даними у водній інженерії.

Теоретична основа:

SQL (Structured Query Language) — це мова запитів для управління реляційними базами даних. SQL дозволяє створювати, змінювати та видаляти структури баз даних, а також виконувати різноманітні операції над даними: додавання, зміна, вибірка, сортування тощо. **MariaDB** — реляційна система управління базами даних, яка використовує SQL як основний інструмент для маніпуляції даними.

Необхідні інструменти:

- Встановлена СУБД MariaDB
- Доступ до терміналу або інтерфейсу для виконання SQL-запитів (CLI або GUI)

Завдання заняття:

1. Підключення до сервера MariaDB.

- Запустіть термінал та підключіться до MariaDB:

```
bashКопіювати код
sudo mysql -u root -p
```
- Введіть пароль адміністратора для доступу до системи.

2. Створення нової бази даних.

- Створіть базу даних для зберігання даних, пов'язаних із водними ресурсами:

```
sqlКопіювати код
CREATE DATABASE water_engineering_db;
```
- Перевірте наявність нової бази даних:

```
sqlКопіювати код
SHOW DATABASES;
```

3. Створення таблиць.

- Підключіться до щойно створеної бази даних:

```
sqlКопіювати код
USE water_engineering_db;
```
- Створіть таблицю для зберігання даних про рівень води в річках:

```
sqlКопіювати код
CREATE TABLE river_levels (
  id INT AUTO_INCREMENT PRIMARY KEY,
  river_name VARCHAR(255),
  measurement_date DATE,
  water_level FLOAT
);
```
- Перевірте, чи таблиця була створена:

```
sqlКопіювати код
SHOW TABLES;
```

4. Внесення даних у таблицю.

- Додайте дані про рівень води в різних річках:

```
sqlКопіювати код
INSERT INTO river_levels (river_name,
measurement_date, water_level)
VALUES ('Dnipro', '2024-10-01', 5.4),
      ('Desna', '2024-10-01', 3.7),
      ('Pivdennyi Buh', '2024-10-01', 6.2);
```

5. Вибірка даних.

- Виведіть всі записи з таблиці:

```
sqlКопіювати код
SELECT * FROM river_levels;
```

- Виберіть лише записи для річки "Dnipro":

```
sqlКопіювати код
SELECT * FROM river_levels WHERE river_name =
'Dnipro';
```

6. Оновлення даних.

- Оновіть рівень води для річки "Desna":

```
sqlКопіювати код
UPDATE river_levels SET water_level = 4.0
WHERE river_name = 'Desna';
```

- Перевірте, чи зміни були успішно внесені:

```
sqlКопіювати код
SELECT * FROM river_levels WHERE river_name =
'Desna';
```

7. Видалення даних.

- Видаліть запис для річки "Pivdennyi Buh":

```
sqlКопіювати код
DELETE FROM river_levels WHERE river_name =
'Pivdennyi Buh';
```

8. Додаткові SQL-запити:

- Сортування даних за рівнем води:

```
sqlКопіювати код
SELECT * FROM river_levels ORDER BY
water_level DESC;
```

- Обчислення середнього рівня води:

```
sqlКопіювати код
SELECT AVG(water_level) AS avg_water_level
FROM river_levels;
```

9. Самостійна робота:

- Створіть нову таблицю для зберігання даних про опади у різних регіонах:

```
sqlКопіювати код
CREATE TABLE rainfall_data (
  id INT AUTO_INCREMENT PRIMARY KEY,
  region VARCHAR(255),
  rainfall_amount FLOAT,
  measurement_date DATE
);
```

- Внесіть кілька записів і виведіть їх.

Підсумки:

Після виконання цього заняття студенти повинні володіти базовими навичками роботи з SQL у MariaDB, включаючи створення баз даних, таблиць, внесення, вибірку, оновлення та видалення даних. Ці навички важливі для управління та аналізу даних, необхідних у водній інженерії.

Запитання для самоперевірки:

1. Яка команда використовується для створення бази даних у MariaDB?
2. Як додати нові записи у таблицю?
3. Як виконати вибірку даних з таблиці за певною умовою?
4. Яка команда використовується для оновлення записів у таблиці?
5. Як видалити запис з таблиці?

Додаткові матеріали:

- MariaDB Documentation
- SQL Tutorial

Це заняття закладає основу для подальшого розуміння та використання SQL-запитів у системах управління базами даних для водної інженерії.

Практичне заняття 3: Використання індексів та оптимізація запитів

Мета заняття:

Ознайомитися з принципами створення та використання індексів у базах даних MariaDB, а також навчитися оптимізувати SQL-запити для підвищення ефективності роботи з великими обсягами даних у водній інженерії.

Теоретична основа:

Індекси в базах даних — це спеціальні структури, які дозволяють швидше

здійснювати пошук і вибірку даних. Індокси можна порівняти зі змістом книги: замість перегляду всієї бази даних, система використовує індекс для швидкого знаходження потрібних рядків.

Оптимізація запитів — це процес удосконалення SQL-запитів і структур бази даних для зменшення часу виконання запитів і підвищення загальної продуктивності. Для цього використовуються різні техніки, такі як правильний вибір індексів, оптимізація запитів та нормалізація даних.

Необхідні інструменти:

- Встановлена СУБД MariaDB
- Доступ до терміналу або GUI для виконання SQL-запитів

Завдання заняття:

1. Ознайомлення з наявною структурою бази даних.

- Підключіться до MariaDB:
`bash`Копіювати код
`sudo mysql -u root -p`
- Перегляньте структуру бази даних, яка містить дані про водні ресурси:

```
sqlКопіювати код
USE water_engineering_db;
SHOW TABLES;
```

2. Створення індексів.

- Уявімо, що ви часто виконуєте запити на вибірку рівня води в річках за певною датою. Щоб прискорити пошук, необхідно створити індекс для стовпця `measurement_date` в таблиці `river_levels`:

```
sqlКопіювати код
CREATE INDEX idx_measurement_date ON
river_levels (measurement_date);
```

- Перевірте наявність індексів у таблиці:

```
sqlКопіювати код
SHOW INDEX FROM river_levels;
```

3. Аналіз виконання запитів за допомогою EXPLAIN.

- Використайте команду EXPLAIN, щоб проаналізувати план виконання запиту та перевірити, як система використовує індекс для вибірки даних:

```
sqlКопіювати код
```

```
EXPLAIN SELECT * FROM river_levels WHERE
measurement_date = '2024-10-01';
```

- Оцініть, як індекс впливає на швидкість виконання запити.

4. Оптимізація складних запитів.

- Створіть запит, який виконує агрегацію даних, наприклад, обчислює середній рівень води за певний період:

```
sqlКопіювати код
SELECT river_name, AVG(water_level) AS
avg_level
FROM river_levels
WHERE measurement_date BETWEEN '2024-01-01'
AND '2024-12-31'
GROUP BY river_name;
```

- Проаналізуйте цей запит за допомогою EXPLAIN і переконайтеся, що він використовує індекси ефективно:

```
sqlКопіювати код
EXPLAIN SELECT river_name, AVG(water_level) AS
avg_level
FROM river_levels
WHERE measurement_date BETWEEN '2024-01-01'
AND '2024-12-31'
GROUP BY river_name;
```

5. Видалення індексів.

- Якщо індекс більше не потрібен, його можна видалити:

```
sqlКопіювати код
DROP INDEX idx_measurement_date ON
river_levels;
```

- Перевірте, чи був індекс успішно видалений:

```
sqlКопіювати код
SHOW INDEX FROM river_levels;
```

6. Оптимізація таблиць.

- Використовуйте команду OPTIMIZE TABLE, щоб покращити продуктивність таблиць після виконання великих операцій із вставлення або видалення даних:

```
sqlКопіювати код
OPTIMIZE TABLE river_levels;
```

7. Практичне завдання з оптимізації запитів.

- Студенти повинні самостійно створити складний SQL-запит,

який виконує операції вибірки даних про опади в регіонах за певний період. Створіть індекси для стовпців, які найчастіше використовуються в умовах запити.

- Використайте команду EXPLAIN, щоб перевірити, як індекси впливають на продуктивність запити.

Приклади складних запитів для аналізу:

1. Запит з об'єднанням таблиць (JOIN).

- Уявімо, що у вас є ще одна таблиця з даними про опади в тих самих регіонах. Ви хочете об'єднати дані про рівень води та опади:

```
sqlКопіювати код
SELECT rl.river_name, rl.measurement_date,
rl.water_level, rd.rainfall_amount
FROM river_levels rl
JOIN rainfall_data rd ON rl.measurement_date =
rd.measurement_date;
```

- Перевірте, чи використовується індекс під час об'єднання таблиць:

```
sqlКопіювати код
EXPLAIN SELECT rl.river_name,
rl.measurement_date, rl.water_level,
rd.rainfall_amount
FROM river_levels rl
JOIN rainfall_data rd ON rl.measurement_date =
rd.measurement_date;
```

Підсумки:

Після виконання цього практичного заняття студенти повинні розуміти, як використовувати індекси для покращення продуктивності SQL-запитів у MariaDB, а також як аналізувати та оптимізувати запити для зменшення часу їх виконання.

Запитання для самоперевірки:

1. Що таке індекси і яка їхня роль у базах даних?
2. Як створити індекс у таблиці?
3. Яка команда використовується для аналізу виконання SQL-запитів?
4. Як індекси впливають на швидкість виконання запитів?
5. Як можна видалити індекс з таблиці?

Додаткові матеріали:

- MariaDB Documentation: Indexing
- SQL Optimization Techniques

Це заняття допоможе студентам розвинути навички оптимізації запитів у MariaDB, що є необхідним для ефективного управління великими обсягами даних у водній інженерії.

Практичне заняття 4: Структура даних у водній інженерії

Мета заняття:

Ознайомитися з типами та структурою даних, що використовуються у водній інженерії, та навчитися правильно організовувати їх для зберігання та аналізу в базах даних. Студенти розглянуть, як створювати структури баз даних для відображення реальних процесів у водній інженерії та як ці структури сприяють ефективному управлінню даними.

Теоретична основа:

У водній інженерії використовуються різноманітні дані, включаючи вимірювання рівня води, об'єми опадів, дані про ґрунт, якість води та кліматичні показники. Ефективне управління цими даними вимагає створення правильної структури бази даних, яка включає таблиці, зв'язки між ними та відповідні типи даних для кожного параметра.

Основні компоненти структури даних:

1. **Таблиці** — основні об'єкти, що зберігають дані.
2. **Атрибути** (поля) — кожна таблиця складається з атрибутів, що описують певні характеристики даних.
3. **Типи даних** — кожен атрибут має певний тип даних (числовий, текстовий, дата і час тощо).
4. **Зв'язки між таблицями** — визначають взаємозв'язки між різними об'єктами бази даних (реляційні зв'язки).

Необхідні інструменти:

- Встановлена СУБД MariaDB
- Доступ до терміналу або інструменту для роботи з базами даних

Завдання заняття:

1. **Огляд наявних даних у водній інженерії.**
 - Дані для водної інженерії можуть включати наступні параметри:
 - Рівень води в річках та водоймах
 - Обсяги опадів у різних регіонах

- Дані про якість води (рівень забруднень)
- Кліматичні показники (температура, вологість)
- Обговоріть зі студентами, які типи даних можна зберігати для кожної категорії. Наприклад:
 - Рівень води — числовий тип (FLOAT або DECIMAL)
 - Дата вимірювання — тип даних DATE
 - Назва річки — текстовий тип (VARCHAR)

2. Створення структури таблиць.

- Створіть таблицю для зберігання даних про рівень води в річках:

sqlКопіювати код

```
CREATE TABLE river_levels (
    id INT AUTO_INCREMENT PRIMARY KEY,
    river_name VARCHAR(100),
    measurement_date DATE,
    water_level DECIMAL(5,2),
    region VARCHAR(100)
);
```

- Створіть таблицю для зберігання даних про обсяги опадів:

sqlКопіювати код

```
CREATE TABLE rainfall_data (
    id INT AUTO_INCREMENT PRIMARY KEY,
    region VARCHAR(100),
    measurement_date DATE,
    rainfall_amount DECIMAL(5,2)
);
```

3. Визначення типів даних для кожного поля.

- Розгляньте, чому важливо правильно вибирати типи даних. Наприклад:
 - Числові поля для зберігання рівня води або опадів обираються з типом DECIMAL, щоб забезпечити точність.
 - Для зберігання дат вимірювань використовується тип DATE, що дозволяє ефективно працювати з часовими рядами.

4. Встановлення зв'язків між таблицями.

- Обговоріть необхідність створення зв'язків між таблицями для зберігання комплексних даних. Наприклад, якщо дані про рівень води та опади зберігаються окремо, можна створити зв'язок між ними за регіоном і датою.
- Додайте зовнішній ключ, що зв'яже дані про рівень води з регіоном:

sqlКопіювати код

```
ALTER TABLE river_levels
ADD CONSTRAINT fk_region
FOREIGN KEY (region) REFERENCES
rainfall_data(region);
```

5. Введення даних у таблиці.

- Введіть дані про рівень води у таблицю `river_levels`:

```
sqlКопіювати код
INSERT INTO river_levels (river_name,
measurement_date, water_level, region)
VALUES ('Dnipro', '2024-10-01', 3.45, 'Kyiv');
```

- Введіть дані про обсяги опадів у таблицю `rainfall_data`:

```
sqlКопіювати код
INSERT INTO rainfall_data (region,
measurement_date, rainfall_amount)
VALUES ('Kyiv', '2024-10-01', 12.5);
```

6. Запити на вибірку даних.

- Виконайте запит на вибірку даних про рівень води в певному регіоні:

```
sqlКопіювати код
SELECT river_name, water_level FROM
river_levels WHERE region = 'Kyiv';
```

- Виконайте запит на вибірку даних про опади за певною датою:

```
sqlКопіювати код
SELECT rainfall_amount FROM rainfall_data
WHERE measurement_date = '2024-10-01';
```

7. Аналіз і візуалізація даних.

- Обговоріть можливість використання даних для прийняття рішень. Наприклад, можна оцінити, як рівень води у річках змінюється в залежності від кількості опадів.
- Студенти можуть побудувати простий SQL-запит для аналізу залежностей:

```
sqlКопіювати код
SELECT rl.river_name, rl.water_level,
rd.rainfall_amount
FROM river_levels rl
JOIN rainfall_data rd ON rl.measurement_date =
rd.measurement_date
WHERE rl.region = 'Kyiv';
```

Підсумки:

Після виконання цього практичного заняття студенти повинні вміти створювати структури баз даних для зберігання різноманітних даних водної інженерії, розуміти значення правильного вибору типів даних, а також знати, як організувати взаємозв'язки між таблицями.

Запитання для самоперевірки:

1. Які основні типи даних використовуються для зберігання даних про рівень води та опади?
2. Як створювати таблиці для зберігання даних у MariaDB?
3. Яка роль зв'язків між таблицями?
4. Які типи зв'язків між таблицями ви знаєте? Наведіть приклади.
5. Як можна аналізувати взаємозв'язки між рівнем води та обсягами опадів?

Додаткові матеріали:

- Database Design and Data Types
- Relational Database Design

Це заняття дозволяє студентам отримати практичні навички з проектування та організації структур баз даних, необхідних для ефективного управління інформацією в водній інженерії.

Практичне заняття 5: Підключення до бази даних з мови програмування R

Мета заняття:

Ознайомити студентів з підключенням до бази даних MariaDB з використанням мови програмування R, а також з методами взаємодії з базою даних для отримання, вставки та аналізу даних. Студенти навчатимуться використовувати спеціальні пакети в R для роботи з базами даних, що є важливою навичкою для обробки даних у водній інженерії.

Теоретична основа:

R широко використовується для аналізу та візуалізації даних у багатьох галузях, включаючи водну інженерію. Здатність безпосередньо підключатися до бази даних дозволяє автоматизувати процеси аналізу даних і працювати з великими наборами даних ефективно.

Основні пакети, що використовуються в R для роботи з базами даних:

1. **DBI** — стандартний пакет для роботи з базами даних у R.
2. **RMySQL** або **RMariaDB** — пакети для роботи з MySQL або MariaDB.

3. **dplyr** — пакет для маніпуляції даними, що може працювати напряму з базами даних.

Необхідні інструменти:

- Встановлена операційна система Linux (Ubuntu) або Windows з R.
- Встановлена СУБД MariaDB.
- Встановлена мова програмування R і відповідні пакети (DBI, RMariaDB, dplyr).

Завдання заняття:

1. Підготовка робочого середовища:

- Перевірте, чи MariaDB працює на вашому комп'ютері або сервері. Якщо ні, запустіть сервер MariaDB:

```
bashКопіювати код
sudo systemctl start mariadb
```
- Переконайтеся, що у вас є створена база даних для водної інженерії (наприклад, `water_engineering_db`), а також користувач з правами доступу до неї.

2. Встановлення необхідних пакетів у R:

- Відкрийте RStudio або консоль R і встановіть необхідні пакети для роботи з базою даних:

```
RКопіювати код
install.packages("DBI")
install.packages("RMariaDB")
install.packages("dplyr")
```

3. Підключення до бази даних:

- Створіть новий R-скрипт, де ви будете підключатися до MariaDB за допомогою пакету RMariaDB.
- Для підключення до бази даних використовуйте наступний код:

```
RКопіювати код
library(DBI)
library(RMariaDB)
```

```
# Параметри підключення
con <- dbConnect(RMariaDB::MariaDB(),
                 dbname =
"water_engineering_db",
                 host = "localhost",
                 user = "your_username",
```

```
password = "your_password")
```

```
# Перевірка підключення  
dbListTables(con) # Показує наявні таблиці в  
базі даних
```

4. Виконання SQL-запитів:

- Виконайте кілька базових SQL-запитів, щоб отримати дані з таблиць бази даних.
- Наприклад, отримайте всі дані з таблиці `river_levels`:

```
RКопіювати код  
# Виконання запиту  
query <- "SELECT * FROM river_levels"  
river_data <- dbGetQuery(con, query)
```

```
# Виведення результатів  
print(river_data)
```

5. Вставка нових даних у таблиці:

- Студенти навчаться додавати нові дані до таблиці з R. Наприклад, можна додати дані про нове вимірювання рівня води:

```
RКопіювати код  
# Вставка нових даних  
dbExecute(con, "INSERT INTO river_levels  
(river_name, measurement_date, water_level,  
region)  
VALUES ('Dnipro', '2024-10-  
05', 3.67, 'Kyiv')")
```

6. Використання пакету `dplyr` для маніпуляції даними:

- Пакет `dplyr` дозволяє працювати з SQL-запитами в більш інтуїтивному стилі. Він також може використовуватися для маніпуляції даними з баз даних.
- Підключіть таблицю бази даних як об'єкт у R:

```
RКопіювати код  
library(dplyr)
```

```
# Підключення до таблиці  
river_tbl <- tbl(con, "river_levels")
```

```
# Виконання запиту з фільтрацією та вибіркою  
query <- river_tbl %>%  
  filter(region == "Kyiv") %>%  
  select(river_name, water_level)
```

```
# Виведення результатів
query %>% collect()
```

7. Вивантаження результатів у R для аналізу:

- Отримані дані можна використовувати для подальшого аналізу і візуалізації в R. Наприклад, можна побудувати графік зміни рівня води у річці:

```
RКопіювати код
library(ggplot2)
```

```
# Створення графіка
ggplot(river_data, aes(x = measurement_date, y
= water_level, color = river_name)) +
  geom_line() +
  labs(title = "Зміна рівня води в річках",
        x = "Дата",
        y = "Рівень води (м)")
```

8. Закриття підключення до бази даних:

- Після завершення роботи з базою даних важливо закрити підключення:

```
RКопіювати код
dbDisconnect(con)
```

Підсумки:

Після виконання цього практичного заняття студенти повинні:

- Вміти підключатися до бази даних MariaDB за допомогою мови R.
- Виконувати SQL-запити на вибірку та вставку даних з R.
- Використовувати пакет `dplyr` для роботи з базою даних.
- Застосовувати отримані дані для аналізу і візуалізації в R.

Запитання для самоперевірки:

1. Який пакет використовується для підключення до MariaDB з R?
2. Як виконати SQL-запит на вибірку даних у R?
3. Як можна використовувати пакет `dplyr` для маніпуляції даними з бази даних?
4. Як можна побудувати графік на основі даних, отриманих з бази даних?
5. Чому важливо закривати підключення до бази даних після завершення роботи?

Додаткові матеріали:

- DBI Package Documentation
- RMariaDB Package Documentation

Це заняття дозволяє студентам засвоїти основи інтеграції баз даних із R, що є важливою частиною аналітичних процесів у водній інженерії.

Практичне заняття 6: Аналіз даних з використанням R

Мета заняття:

Ознайомити студентів з основними методами аналізу даних у R для водної інженерії. Студенти навчатимуться проводити базові аналітичні операції, такі як обробка, візуалізація та статистичний аналіз даних. Це заняття дозволить набути навичок, необхідних для прийняття рішень на основі даних.

Теоретична основа:

R — це потужна мова програмування для статистичного аналізу та візуалізації даних. У водній інженерії аналіз даних використовується для моделювання, моніторингу водних ресурсів, прогнозування природних явищ, таких як повені, та оптимізації управлінських рішень.

Основні аспекти аналізу даних в R:

1. **Завантаження даних** (CSV, бази даних, API).
2. **Очищення та підготовка даних** (робота з пропущеними значеннями, категоризація даних).
3. **Аналіз та маніпуляція даними** (статистичні розрахунки, групування, зведення).
4. **Візуалізація** (створення графіків і діаграм).
5. **Статистичне моделювання** (регресійний аналіз, тимчасові ряди).

Необхідні інструменти:

- Встановлена мова програмування R.
- Встановлені пакети для аналізу даних (`dplyr`, `ggplot2`, `readr`, `lubridate`, `tidyverse`).
- Дані для аналізу (можуть бути з водної бази даних або з файлу CSV, наприклад, дані про рівень води в річках або опади).

Завдання заняття:

1. **Завантаження та імпорт даних:**
 - Студенти завантажуть необхідні дані, наприклад, дані про рівень води в річках або кількість опадів, з файлу CSV або з бази даних.
 - Використайте пакет `readr` для імпорту CSV-файлу:

```

RКопіювати код
library(readr)

# Завантаження даних з файлу
water_data <-
read_csv("data/river_levels.csv")

# Перегляд перших кількох рядків
head(water_data)

```

2. Очищення даних:

- Обробка пропущених значень і форматування колонок.
- Використовуйте пакет `dplyr` для очищення даних і підготовки до аналізу:

```

RКопіювати код
library(dplyr)

# Очищення даних: видалення рядків з пропущеними значеннями
clean_data <- water_data %>%
  filter(!is.na(water_level))

# Перегляд очищених даних
summary(clean_data)

```

3. Аналіз даних:

- Використання функцій для обчислення базової статистики: середнє, медіана, стандартне відхилення.
- Приклад аналізу даних про рівень води:

```

RКопіювати код
# Розрахунок основних статистичних показників
summary_stats <- clean_data %>%
  summarise(mean_level =
mean(water_level),
            median_level =
median(water_level),
            sd_level =
sd(water_level))

print(summary_stats)

```

4. Групування даних:

- Студенти навчатися групувати дані за різними ознаками (наприклад, за річками або регіонами).

- Використайте функцію `group_by` для зведення даних:


```
RКопіювати код
# Групування за річками та розрахунок середнього рівня води
grouped_data <- clean_data %>%
  group_by(river_name) %>%
  summarise(mean_water_level =
mean(water_level))

print(grouped_data)
```

5. Візуалізація даних:

- Створення графіків з використанням пакету `ggplot2`.
- Приклад побудови графіка зміни рівня води у певній річці:


```
RКопіювати код
library(ggplot2)

# Побудова лінійного графіка рівня води з часом
ggplot(clean_data, aes(x = measurement_date, y = water_level, color = river_name)) +
  geom_line() +
  labs(title = "Зміна рівня води в річках",
        x = "Дата",
        y = "Рівень води (м)") +
  theme_minimal()
```

6. Статистичне моделювання:

- Основи регресійного аналізу для моделювання залежності між змінними.
- Виконайте регресійний аналіз для прогнозування рівня води на основі інших факторів:


```
RКопіювати код
# Проста лінійна регресія
model <- lm(water_level ~ rainfall +
temperature, data = clean_data)

# Результати моделі
summary(model)
```

7. Аналіз часових рядів:

- Аналіз і моделювання часових рядів для прогнозування майбутніх показників.
- Створіть графік та виконайте аналіз часових рядів для рівня

ВОДИ:

```
RКопіювати код
library(lubridate)

# Перетворення дати на формат часу
clean_data$measurement_date <-
ymd(clean_data$measurement_date)

# Візуалізація часових рядів
ggplot(clean_data, aes(x = measurement_date, y
= water_level)) +
  geom_line() +
  labs(title = "Часові ряди рівня води", x =
"Дата", y = "Рівень води (м)")
```

8. Вивантаження результатів аналізу:

- Результати аналізу можуть бути експортовані в нові файли CSV або збережені в базі даних для подальшого використання.
- Збережіть результати у CSV-файл:
RКопіювати код
Збереження очищених та агрегованих даних у файл
write_csv(grouped_data,
"output/average_water_levels.csv")

Підсумки:

Після виконання цього практичного заняття студенти повинні:

- Вміти завантажувати та очищати дані у R.
- Застосовувати базові аналітичні методи, такі як обчислення статистичних показників і групування даних.
- Створювати візуалізації для аналізу даних.
- Виконувати простий регресійний аналіз та аналіз тимчасових рядів.

Запитання для самоперевірки:

1. Як завантажити дані з CSV у R?
2. Які функції використовуються для очищення та групування даних у R?
3. Як побудувати лінійний графік зміни рівня води з часом?
4. Які основні методи використовуються для аналізу тимчасових рядів?
5. Як виконати регресійний аналіз у R?

Додаткові матеріали:

- Tidyverse Documentation
- ggplot2 Documentation
- R for Data Science Book

Це заняття дозволяє студентам отримати практичні навички роботи з даними у R, які необхідні для аналізу та прийняття рішень у водній інженерії.

Практичне заняття 7: Створення звітів та візуалізація даних

Мета заняття:

Навчити студентів створювати звіти на основі даних із використанням засобів візуалізації для аналізу та інтерпретації інформації. Студенти навчаться використовувати графіки та діаграми для наочного подання результатів досліджень у водній інженерії, а також генерувати звіти з використанням R Markdown.

Теоретична основа:

Візуалізація даних є критичним елементом у водній інженерії, оскільки вона дозволяє краще зрозуміти великі обсяги інформації, виявляти тенденції, закономірності та допомагати у прийнятті рішень. Крім того, звіти є важливою частиною робочого процесу, оскільки вони систематизують результати аналізу і надають їх у зручному для користувачів вигляді.

Інструменти для створення звітів і візуалізації:

- **R Markdown** — система для створення інтерактивних звітів у форматах PDF, HTML або Word.
- **ggplot2** — один з найпотужніших інструментів для візуалізації даних у R.

Необхідні інструменти:

- Мова програмування R.
- Пакети `ggplot2`, `rmarkdown`, `knitr` для створення графіків та звітів.
- Дані з бази даних або попередніх практичних занять для аналізу.

Завдання заняття:

1. **Ознайомлення зі створенням звітів у R Markdown:**
 - Студенти навчаться створювати документ у R Markdown, де можна одночасно писати текст, вставляти код R і генерувати графіки.
 - Приклад створення R Markdown документа:

```

RКопіювати код---
title: "Звіт з аналізу водних даних"
author: "Студент"
date: "`r Sys.Date()`"
output: html_document
---
```

```

# Вступ
Цей звіт включає аналіз даних рівня води в річ-
ках з візуалізацією.
```

```

```{r setup, include=FALSE}
library(ggplot2)
library(readr)
Копіювати код
```

## 2. Візуалізація даних за допомогою ggplot2:

- Створення кількох видів графіків для візуалізації даних, таких як лінійні графіки, гістограми, діаграми розсіювання та діаграми коробок.
- Приклад лінійного графіка для відображення змін рівня води з часом:

```

RКопіювати код
Лінійний графік рівня води з часом
ggplot(water_data, aes(x = measurement_date, y
= water_level)) +
 geom_line(color = "blue") +
 labs(title = "Зміна рівня води", x = "Дата",
y = "Рівень води (м)") +
 theme_minimal()
```

## 3. Створення кількох графіків для порівняння:

- Порівняння рівнів води в різних річках або регіонах.
- Використання кольорів або фільтрів для полегшення інтерпретації даних:

```

RКопіювати код
ggplot(water_data, aes(x = measurement_date, y
= water_level, color = river_name)) +
 geom_line() +
 labs(title = "Порівняння рівнів води в річ-
ках", x = "Дата", y = "Рівень води (м)") +
 theme_minimal()
```

## 4. Підготовка інтерактивних звітів:

- Використання knitr для інтеграції R-коду і виведення результатів (графіків, таблиць) безпосередньо у звіт.
- Приклад інтеграції результатів аналізу у звіт:  
RКопіювати код  

```
Статистичний аналіз рівня води
summary(water_data$water_level)
```

## 5. Додавання коментарів і опису результатів:

- Опис кожного кроку аналізу та інтерпретація графіків і статистичних показників. Це допоможе зробити звіт зрозумілим для інших фахівців, які не працюють з вихідними даними.

RКопіювати код  
# Опис

У цьому розділі представлено графік зміни рівня води протягом останніх 5 років. Як видно з графіка, у певні місяці відбувалися значні коливання рівня води.

## 6. Створення додаткових елементів візуалізації:

- Використання коробкових діаграм для відображення розподілу рівнів води між різними річками.
- Приклад створення коробкової діаграми:

RКопіювати код  
# Коробкова діаграма для порівняння рівня води в річках  

```
ggplot(water_data, aes(x = river_name, y = water_level)) +
 geom_boxplot() +
 labs(title = "Порівняння рівнів води в річках", x = "Річка", y = "Рівень води (м)") +
 theme_minimal()
```

## 7. Експорт звітів:

- Студенти навчаться зберігати звіти у форматах PDF, HTML або Word.
- Використання RStudio для генерації фінального звіту.

## 8. Візуалізація на картах:

- Студенти дізнаються, як відображати дані на картах за допомогою пакету leaflet або gmap.
- Приклад інтеграції географічної візуалізації:

RКопіювати код  
library(leaflet)

```
leaflet() %>%
 addTiles() %>%
 addMarkers(lng = 30.52, lat = 50.45, popup =
"Київ") %>%
 addMarkers(lng = 30.26, lat = 50.62, popup =
"Річка Дніпро")
```

### Підсумки:

Після виконання цього практичного заняття студенти повинні:

- Вміти створювати звіти з використанням R Markdown.
- Використовувати пакет `ggplot2` для візуалізації даних.
- Інтерпретувати та описувати результати аналізу у вигляді текстових коментарів.
- Створювати звіти у форматах PDF, HTML або Word і поширювати результати для інших спеціалістів.

### Запитання для самоперевірки:

1. Як створити документ у R Markdown?
2. Які типи графіків можна створювати за допомогою `ggplot2`?
3. Як інтегрувати R-код у звіти, щоб автоматично генерувати результати?
4. Як зберегти звіт у форматі PDF або HTML?
5. Яка роль візуалізації на картах у водній інженерії?

### Додаткові матеріали:

- R Markdown Documentation
- ggplot2 Documentation
- leaflet for R

Це заняття дозволяє студентам отримати важливі навички створення звітів та візуалізації даних, що є необхідними для аналізу великих обсягів інформації у водній інженерії.

## Практичне заняття 8: Реплікація та резервне копіювання даних

### Мета заняття:

Ознайомити студентів із процесами реплікації та резервного копіювання даних у системах управління базами даних (СУБД). Студенти навчаться створювати надійні механізми збереження та відновлення даних, що є особливо важливим у водній інженерії для забезпечення доступності даних, навіть у випадку відмов системи або втрати даних.



## Теоретична основа:

Водна інженерія потребує ефективного управління даними, адже системи моніторингу, аналізу водних ресурсів та екологічного контролю постійно генерують значні обсяги даних. Зберігання цих даних повинно бути надійним, а резервні копії і механізми реплікації забезпечують безперервну доступність інформації.

- **Реплікація** — це процес копіювання і зберігання даних на декількох серверах або вузлах для забезпечення їх доступності та захисту від втрат. Існують кілька видів реплікації: майстер-слейв, майстер-майстер та інші.
- **Резервне копіювання** — це створення резервної копії бази даних, яку можна відновити в разі пошкодження чи втрати основної бази.

## Інструменти:

- СУБД MariaDB для налаштування реплікації та резервного копіювання.
- Термінал Linux для виконання команд реплікації та резервного копіювання.

## Завдання заняття:

1. **Ознайомлення з основами реплікації:**
  - Вступ до поняття реплікації даних і її видів (майстер-слейв, майстер-майстер).
  - Пояснення, як реплікація допомагає у водній інженерії для забезпечення постійного доступу до даних, особливо в розподілених системах.
2. **Налаштування майстер-слейв реплікації в MariaDB:**
  - Студенти налаштують реплікацію між двома серверами MariaDB.
  - Підготовка серверів:
    - Майстер-сервер — головний сервер, що містить дані.
    - Слейв-сервер — сервер, який отримуватиме копію даних з майстра.
  - Кроки:
    1. **Налаштування майстер-сервера:**
      - Відкриття конфігураційного файлу MariaDB `/etc/mysql/my.cnf` і додавання параметрів для реплікації:

```
bashКопіювати код
[mysqld]
log-bin
server-id=1
```

## 2. Налаштування слейв-сервера:

- Відкриття конфігураційного файлу на слейв-сервері і додавання параметрів:

```
bashКопіювати код
[mysqld]
server-id=2
replicate-do-db=water_db
```

## 3. Запуск процесу реплікації:

- Створення резервної копії бази даних на майстер-сервері та імпорт її на слейв-сервер:

```
bashКопіювати код
mysqldump -u root -p water_db >
water_db_backup.sql
mysql -u root -p -h SLAVE_IP water_db <
water_db_backup.sql
```

- Налаштування слейв-сервера на отримання змін:

```
bashКопіювати код
CHANGE MASTER TO
MASTER_HOST='MASTER_IP',
MASTER_USER='replication_user',
MASTER_PASSWORD='password',
MASTER_LOG_FILE='mysql-bin.000001',
MASTER_LOG_POS=107;
START SLAVE;
```

## 3. Моніторинг і управління реплікацією:

- Студенти вивчать команди для моніторингу реплікації:

```
bashКопіювати код
SHOW SLAVE STATUS\G;
```

- Вивчення можливих помилок реплікації та способів їх усунення.

## 4. Ознайомлення з резервним копіюванням даних:

- Важливість регулярного резервного копіювання даних для запобігання їх втратам.
- Основні види резервного копіювання: повне, диференційне та інкрементальне.
- Інструменти для резервного копіювання в MariaDB:

- **mysqldump** — стандартний інструмент для створення резервної копії бази даних.

## 5. Практичне створення резервної копії бази даних:

- Виконання повного резервного копіювання бази даних:

```
bashКопіювати код
mysqldump -u root -p water_db >
water_db_backup.sql
```

- Створення інкрементального резервного копіювання для великих баз даних:

```
bashКопіювати код
mysqldump -u root -p --single-transaction --
quick --lock-tables=false water_db >
water_db_incremental_backup.sql
```

## 6. Автоматизація процесу резервного копіювання:

- Створення скрипту для автоматичного резервного копіювання даних за допомогою CRON:

```
bashКопіювати код
0 2 * * * /usr/bin/mysqldump -u root -p
password water_db > /path_to_backup/backup.sql
```

## 7. Відновлення бази даних із резервної копії:

- Студенти навчаться відновлювати базу даних після її пошкодження або втрати.
- Відновлення бази даних із повної резервної копії:

```
bashКопіювати код
mysql -u root -p water_db <
water_db_backup.sql
```

## 8. Резервне копіювання великих баз даних:

- Вивчення методів зберігання та стиснення великих обсягів даних:

```
bashКопіювати код
mysqldump -u root -p water_db | gzip >
water_db_backup.sql.gz
```

## Підсумки:

Після виконання цього практичного заняття студенти повинні:

- Вміти налаштувати майстер-слейв реплікацію в MariaDB.
- Виконувати резервне копіювання даних і відновлення бази даних.
- Розуміти важливість реплікації і резервного копіювання для

забезпечення доступності даних у водній інженерії.

### Запитання для самоперевірки:

1. Які типи реплікації підтримуються в MariaDB?
2. Які основні кроки налаштування майстер-слейв реплікації?
3. Як виконати повне резервне копіювання бази даних за допомогою `mysqldump`?
4. Як можна автоматизувати процес резервного копіювання?
5. Чим відрізняється інкрементальне резервне копіювання від повного?

### Додаткові матеріали:

- MariaDB Replication Guide
- Backup and Restore in MariaDB

Це заняття допоможе студентам зрозуміти важливість резервного копіювання та реплікації даних для забезпечення безперервної роботи систем у водній інженерії та уникнення втрат даних.

## Практичне заняття 9: Захист та безпека баз даних

### Мета заняття:

Ознайомити студентів із основними принципами захисту та безпеки баз даних у водній інженерії. Студенти навчатимуться застосовувати методи забезпечення конфіденційності, цілісності та доступності даних, використовуючи СУБД MariaDB. Безпека даних є критично важливою для систем моніторингу водних ресурсів, де будь-які порушення можуть призвести до катастрофічних наслідків для екології та інфраструктури.

### Теоретична основа:

У водній інженерії безпека баз даних є необхідною для захисту конфіденційної інформації, такої як дані про водні ресурси, які можуть бути важливими для державного управління або приватних підприємств. Важливо уникнути несанкціонованого доступу, захистити дані від зовнішніх та внутрішніх загроз, а також забезпечити можливість відновлення в разі аварійних ситуацій.

Основні аспекти безпеки баз даних включають:

1. **Аутентифікація та авторизація** — контроль доступу до бази даних.
2. **Шифрування** — захист даних при зберіганні та передачі.
3. **Резервне копіювання та аварійне відновлення** — забезпечення доступності даних.
4. **Аудит та моніторинг активності** — спостереження за діями користувачів.

## Інструменти:

- **MariaDB** — для налаштування користувачів, прав доступу, шифрування даних та моніторингу активності.
- **Ubuntu Linux** — для конфігурації безпечного доступу до системи та баз даних.
- **Файрволи та інструменти для шифрування (SSL/TLS).**

## Завдання заняття:

### 1. Аутентифікація користувачів у MariaDB:

- Ознайомлення з поняттями аутентифікації та авторизації.
- Студенти навчатимуться створювати користувачів з різними рівнями доступу до бази даних.

**Приклад** створення нового користувача з обмеженими правами:

```
sqlКопіювати код
CREATE USER 'water_user'@'localhost' IDENTIFIED
BY 'strong_password';
GRANT SELECT, INSERT ON water_db.* TO
'water_user'@'localhost';
FLUSH PRIVILEGES;
```

### 2. Налаштування ролей і прав доступу:

- Створення ролей для користувачів із різними рівнями доступу до бази даних (адміністратори, аналітики, користувачі з обмеженими правами).
- Налаштування правил для окремих таблиць, щоб обмежити доступ до чутливої інформації.

**Приклад** надання ролей:

```
sqlКопіювати код
CREATE ROLE 'db_admin';
GRANT ALL PRIVILEGES ON water_db.* TO 'db_admin';
```

### 3. Моніторинг дій користувачів та аудит:

- Важливість відстеження дій користувачів для виявлення потенційних загроз.
- Включення журналу аудиту (audit logging) у MariaDB для збереження інформації про виконані запити.

**Приклад** команди для моніторингу підключень:

```
sqlКопіювати код
SHOW PROCESSLIST;
```

Студенти вивчатимуть також як налаштувати логування дій користувачів і зберігати ці дані для аудиту.

### 4. Захист паролів і шифрування даних:

- Важливість використання сильних паролів і їх шифрування.
- Вивчення методів шифрування даних у базах даних MariaDB.

**Приклад** шифрування поля в таблиці:

```
sqlКопіювати код
CREATE TABLE secure_data (
 id INT PRIMARY KEY,
 sensitive_data VARBINARY(255) ENCRYPTED
);
```

## 5. Шифрування даних при передачі (SSL/TLS):

- Студенти навчаться налаштувати захищене з'єднання для MariaDB, щоб дані передавались у зашифрованому вигляді між клієнтом та сервером.

**Приклад** налаштування SSL/TLS для MariaDB:

- Налаштування конфігурації MariaDB для використання SSL:

```
bashКопіювати код
[mysqld]
ssl-ca=/path_to/ca-cert.pem
ssl-cert=/path_to/server-cert.pem
ssl-key=/path_to/server-key.pem
```

- Включення SSL для конкретного користувача:

```
sqlКопіювати код
GRANT USAGE ON *.* TO 'water_user'@'%' REQUIRE
SSL;
```

## 6. Налаштування брандмауера та обмеження доступу:

- Важливість захисту баз даних на рівні мережі.
- Студенти налаштують простий брандмауер (Firewall) в Ubuntu для обмеження доступу до MariaDB з певних IP-адрес.

**Приклад** налаштування правил брандмауера:

```
bashКопіювати код
sudo ufw allow from 192.168.1.0/24 to any port
3306
```

## 7. Захист від SQL-ін'єкцій:

- Пояснення загрози SQL-ін'єкцій і як правильно писати запити, щоб уникнути цієї загрози.
- Використання підготовлених запитів (prepared statements) для захисту.

**Приклад** підготовленого запиту в MariaDB:

```
sqlКопіювати код
PREPARE stmt FROM 'SELECT * FROM water_data WHERE
id = ?';
```

```
SET @id = 1;
EXECUTE stmt USING @id;
```

## 8. Відновлення паролів і управління політиками паролів:

- Опис сценаріїв відновлення доступу у випадку втрати паролю адміністратора бази даних.
- Налаштування політики для регулярного зміни паролів і перевірки їх складності.

**Приклад** зміни політики паролів у MariaDB:

```
sqlКопіювати код
SET GLOBAL validate_password_policy = MEDIUM;
```

## 9. Бекап та аварійне відновлення як частина безпеки:

- Резервне копіювання є частиною стратегії безпеки, оскільки дані можуть бути втрачені через атаки або апаратні збої.
- Вивчення процедур відновлення бази даних після збоїв.

## Підсумки:

Після завершення цього заняття студенти повинні:

- Розуміти основні загрози безпеці баз даних і як захистити їх від атак.
- Вміти налаштувати безпечний доступ до бази даних.
- Налаштувати користувачів із різними правами доступу та моніторинг їх дій.
- Вміти налаштувати шифрування даних у базі даних та при їх передачі.
- Застосовувати методи захисту від SQL-ін'єкцій.

## Запитання для самоперевірки:

1. Що таке SQL-ін'єкція і як від неї захиститися?
2. Як можна шифрувати дані в MariaDB?
3. Яка команда дозволяє моніторити активність користувачів у базі даних?
4. Чому важливо використовувати SSL/TLS для передачі даних?
5. Як налаштувати брандмауер для захисту бази даних?

## Додаткові матеріали:

- MariaDB Security Features
- Encrypting Data in MariaDB
- How to Protect Against SQL Injection Attacks

Це практичне заняття дасть студентам фундаментальні знання та навички для забезпечення безпеки баз даних, що є критично важливим для управління водними ресурсами та іншими системами водної інженерії.

## Практичне заняття 10: Реалізація комплексного проекту

### Мета заняття:

Застосувати знання та навички, набуті протягом курсу, для створення комплексного проекту, що вирішує конкретну задачу управління базами даних у водній інженерії. Студенти будуть виконувати повний цикл розробки рішення: від проектування бази даних до її реалізації та інтеграції з зовнішніми системами, включаючи аналіз даних і створення звітів.

### Опис проекту:

Студенти повинні розробити та реалізувати базу даних для управління інформацією про водні ресурси регіону. Проект охоплює всі етапи створення бази даних: від аналізу вимог до оптимізації запитів та забезпечення безпеки.

Прикладом завдання може бути система моніторингу водних ресурсів, яка збирає дані про рівень води, якість води та стан водних об'єктів (річки, озера, водосховища). Дані можуть використовуватись для прийняття рішень у галузі управління водними ресурсами.

### Основні завдання:

#### 1. Аналіз вимог:

- Визначення, які дані необхідно зберігати в системі (рівень води, температура, якість води тощо).
- Визначення користувачів системи (адміністратори, аналітики, фахівці з водних ресурсів) та їх прав доступу.

#### 2. Проектування структури бази даних:

- Створення ER-діаграми (діаграми сутність-зв'язок), яка відображає взаємозв'язки між сутностями (наприклад, водні об'єкти, виміри, сенсори).
- Визначення основних таблиць та полів: таблиця об'єктів моніторингу (річки, озера), таблиця вимірів (рівень води, температура, рН).

#### 3. Створення бази даних у MariaDB:

- Реалізація структури бази даних у MariaDB з використанням команд SQL для створення таблиць, індексів та відношень між ними.

#### Приклад створення таблиці:

```
sqlКопіювати код
CREATE TABLE water_objects (
 id INT PRIMARY KEY AUTO_INCREMENT,
```



```
name VARCHAR(255),
location VARCHAR(255)
);
```

#### **Приклад створення таблиці для зберігання вимірів:**

sqlКопіювати код

```
CREATE TABLE measurements (
 id INT PRIMARY KEY AUTO_INCREMENT,
 water_object_id INT,
 measurement_date DATE,
 water_level FLOAT,
 water_temperature FLOAT,
 ph_level FLOAT,
 FOREIGN KEY (water_object_id) REFERENCES
water_objects(id)
);
```

#### **4. Заповнення бази даних тестовими даними:**

- Наповнення бази даних тестовими записами, що симулюють реальні вимірювання для різних водних об'єктів.

##### **Приклад вставки даних:**

sqlКопіювати код

```
INSERT INTO water_objects (name, location) VALUES
('Dnipro River', 'Kyiv');
INSERT INTO measurements (water_object_id,
measurement_date, water_level, water_temperature,
ph_level)
VALUES (1, '2024-09-25', 3.5, 18.2, 7.2);
```

#### **5. Створення індексів та оптимізація запитів:**

- Студенти оптимізують запити до бази даних шляхом створення індексів для поліпшення швидкості пошуку та обробки даних.

##### **Приклад створення індексу:**

sqlКопіювати код

```
CREATE INDEX idx_measurement_date ON
measurements(measurement_date);
```

#### **6. Інтеграція з мовою R для аналізу даних:**

- Підключення до бази даних з мови R для вибірки даних та їх аналізу. Студенти виконують аналіз даних, зібраних в базі, для отримання статистики та візуалізації.

##### **Приклад підключення до бази даних у R:**

rКопіювати код

```
library(DBI)
con <- dbConnect(RMariaDB::MariaDB(), user =
'user', password = 'password', dbname =
```

```
'water_db', host = 'localhost')
data <- dbGetQuery(con, "SELECT * FROM
measurements WHERE water_object_id = 1")
```

#### 7. Аналіз даних та візуалізація результатів:

- Виконання аналізу даних (наприклад, статистичний аналіз рівня води) і створення графіків для візуалізації результатів.

##### **Приклад аналізу та візуалізації:**

```
rКопіювати код
library(ggplot2)
ggplot(data, aes(x = measurement_date, y =
water_level)) + geom_line() + labs(title = "Water
Level Over Time")
```

#### 8. Створення звітів:

- Генерація звітів, що відображають результати аналізу, використовуючи можливості R та SQL. Ці звіти можуть бути представлені у вигляді таблиць або графіків для демонстрації динаміки рівня води, температури тощо.

#### 9. Безпека та резервне копіювання даних:

- Налаштування резервного копіювання бази даних для забезпечення безпеки даних.
- Забезпечення ролей і прав доступу для різних користувачів, які працюватимуть з базою даних.

##### **Приклад резервного копіювання:**

```
bashКопіювати код
mysqldump -u user -p water_db >
water_db_backup.sql
```

#### 10. Презентація проекту:

- Кожен студент або група студентів повинні презентувати результати своєї роботи: структуру бази даних, результати аналізу та звіти, а також висновки, що впливають з оброблених даних.

### **Очікувані результати:**

Після завершення проекту студенти зможуть:

- Проектувати та реалізовувати бази даних для вирішення реальних задач у водній інженерії.
- Застосовувати методи аналізу даних для отримання корисної інформації з бази даних.
- Створювати звіти та візуалізації для представлення результатів аналізу.
- Забезпечувати безпеку даних та налаштовувати резервне копіювання.

- Оптимізувати запити та роботу з великою кількістю даних.

#### **Додаткові матеріали:**

- MariaDB Documentation
- R Documentation
- ERD Tool

Це практичне заняття дозволить студентам об'єднати всі здобуті знання та навички в реальному комплексному проекті, який імітує реальну задачу управління водними ресурсами.