



# ГРАФОВІ БАЗИ ДАНИХ

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Другий (магістерський)</i>
Галузь знань	<i>Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>1-й курс, 2-й семестр</i>
Обсяг дисципліни	<i>5 кредитів / 150 год. (очна форма: лекцій 36 год., лаб. 18 год., СРС 96 год.; заочна форма: лекцій 10 год., лаб. 6 год., СРС 134 год.)</i>
Семестровий контроль/ контрольні заходи	<i>Залік, модульна контрольна робота</i>
Розклад занять	<i><a href="http://schedule.kpi.ua/">http://schedule.kpi.ua/</a></i>
Мова викладання	<i>Українська/Англійська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: д. е. н., професор Сігайов Андрій Олександрович Практичні / Семінарські: Лабораторні: Сігайов А. О.</i>
Розміщення курсу	

### Програма навчальної дисципліни

#### 1 Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

##### **Чому майбутньому фахівцю варто вчити саме цю дисципліну?**

Графові дані закривають розрив між тим, як люди та комп'ютери бачать світ. В той час, як комп'ютери покладаються на статичні рядки та стовпчики даних, люди розмірковують про життя у термінах взаємостосунків. Цей практичний курс демонструє яким чином графові дані поєднують ці два підходи. Працюючи з концепціями теорії графів, схем баз даних, розподілених систем і аналізу даних, ми дістаємося унікальної точки перетину, відомої як графове мислення.

Цей курс показує інженерам даних, фахівцям науки про дані та аналітикам даних як розв'язувати складні задачі за допомогою графових баз даних. Ми дослідимо шаблони побудови за допомогою графової технології разом з прикладами, які демонструють, яким чином команди можуть думати про свої застосунки у термінах графових даних.

**Мета дисципліни.** Ознайомити студентів з графовою моделлю даних та сучасними графовими базами даних.

**Предмет дисципліни.** Огляд мови графових баз даних, включаючи:

- побудову прикладу архітектури застосунку за допомогою реляційної та графової технологій;

- використання графової технології для розробки застосунку Customer 360, найпопулярнішого зараз паттерна графових даних;
- занурення у ієрархічні дані та засвоєння нової парадигми, що впливає з роботи із графовими даними;
- знаходження маршрутів у графових даних та з'ясування, чого довіра до різних траєкторій мотивує та інформує ваші переваги;
- використання колаборативної фільтрації для розробки Netflix-подібної рекомендаційної системи.

#### **Очікувані результати навчання.**

#### **Фахові компетентності.**

ФК 13. Здатність реалізовувати застосунки з використанням концепцій інженерії даних та знань.

#### **Програмні результати навчання.**

ПРН 17. Збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела.

ПРН 21. Розробляти застосунки з використанням концепцій інженерії даних та знань.

### **2 Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)**

Дисципліна вивчається у другому семестрі. Пререквізитами є курси бакалаврського рівня "Комп'ютерна дискретна математика" та "Бази даних". Постреквізитів у даного курсу на магістерському рівні немає.

### **3 Зміст навчальної дисципліни**

1. Графове мислення.
2. Еволюція від реляційного до графового мислення.
3. Починаємо: простий застосунок Customer 360.
4. Дослідження околу на етапі розробки.
5. Дослідження околу на етапі експлуатації.
6. Використання дерев на етапі розробки.
7. Використання дерев на етапі експлуатації.
8. Знаходження маршрутів на етапі розробки.
9. Знаходження маршрутів на етапі експлуатації.
10. Рекомендаційні системи на етапі розробки.
11. Просте розв'язання сутностей у графах.
12. Рекомендаційні системи на етапі експлуатації.
13. Підсумки.

### **4 Навчальні матеріали та ресурси**

#### **Базова література:**

Gosnell, D., Broecheler, M. *The Practitioner's Guide to Graph Data: Applying Graph Thinking and Graph Technologies to Solve Complex Problems*: Sebastopol, CA: O'Reilly Media, 2020. 420 с. URL: <http://libgen.rs/book/index.php?md5=2F852C74D4139268D520CB3E4B1662D3>

### **Додаткова література:**

1. Bechberger, D., Perryman, J. *Graph Databases in Action: Shelter Island, NY: Manning Publications, 2020. 336 с. URL: <http://libgen.rs/book/index.php?md5=7F8919C423C42D10A3BBEAABF5026F3B>*
2. Needham, M., Hodler, A. E. *Graph Algorithms: Practical Examples in Apache Spark and Neo4j: Sebastopol, CA: O'Reilly Media, 2019. 256 с. URL: <http://libgen.rs/book/index.php?md5=cc42a7b9970c7f4930f014ababe5e03a>*
3. Lee, V., Nguyen, P. K., Thomas, A. *Graph-Powered Analytics and Machine Learning with TigerGraph: Driving Business Outcomes with Connected Data: Sebastopol, CA: O'Reilly Media, 2023. 314 с. URL: <http://libgen.rs/book/index.php?md5=375C33F4657BAD2EBC7F76A2E2BC842D>*

## **Навчальний контент**

### **5 Методика опанування навчальної дисципліни (освітнього компонента)**

1. *Графове мислення.*
  - 1.1. *Чому зараз це актуально? Історичний контекст технологій БД.*
    - 1.1.1. *1960–1980: ієрархічна модель.*
    - 1.1.2. *1980–2000: реляційна модель.*
    - 1.1.3. *2000–2020: NoSQL.*
    - 1.1.4. *2020–?: графова модель.*
  - 1.2. *Що таке графове мислення?*
    - 1.2.1. *Складні задачі та складні системи.*
    - 1.2.2. *Складні задачі у бізнесі.*
  - 1.3. *Ухвалення технологічних рішень для розв'язання складних задач.*
    - 1.3.1. *Отже, у вас є графові дані. Що далі?*
    - 1.3.2. *Побачити велику картину.*
  - 1.4. *Початок подорожі до графового мислення.*
2. *Еволюція від реляційного до графового мислення.*
  - 2.1. *Стислий огляд: переклад реляційних концепцій у графову термінологію.*
  - 2.2. *Реляційна модель проти графової: яка різниця?*
  - 2.3. *Реляційне моделювання даних.*
    - 2.3.1. *Сутності та атрибути.*
    - 2.3.2. *Побудова діаграми “сутність-зв'язок”.*
  - 2.4. *Концепції графових даних.*
    - 2.4.1. *Фундаментальні елементи графа.*
    - 2.4.2. *Суміжність.*
    - 2.4.3. *Окіл.*

- 2.4.4. Відстань.
- 2.4.5. Ступінь.
- 2.5. Мова опису графів.
  - 2.5.1. Мітки вершин і ребер.
  - 2.5.2. Властивості.
  - 2.5.3. Напрямок ребра.
  - 2.5.4. Петлі.
  - 2.5.5. Мультиплікативна розмірність графа.
  - 2.5.6. Повний приклад графової моделі.
- 2.6. Реляційна модель проти графової: рішення, що треба ухвалити.
  - 2.6.1. Моделювання даних.
  - 2.6.2. Розуміння графових даних.
  - 2.6.3. Поєднання дизайну баз даних із застосовною метою.
- 2.7. Підсумки.
- 3. Починаємо: простий застосунок Customer 360.
  - 3.1. Стислий огляд: реляційний підхід проти графового.
  - 3.2. Основний юз-кейс для графових даних: C360.
    - 3.2.1. Чому для бізнесу важливий C360?
  - 3.3. Реалізація застосунку C360 у реляційній системі.
    - 3.3.1. Моделі даних.
    - 3.3.2. Реляційна реалізація.
    - 3.3.3. Приклади запитів до C360.
  - 3.4. Реалізація застосунку C360 у графовій системі.
    - 3.4.1. Моделі даних.
    - 3.4.2. Графова реалізація.
    - 3.4.3. Приклади запитів до C360.
  - 3.5. Реляційна модель проти графової: як обирати?
    - 3.5.1. Реляційна модель проти графової: моделювання даних.
    - 3.5.2. Реляційна модель проти графової: представлення стосунків.
    - 3.5.3. Реляційна модель проти графової: мови запитів.
    - 3.5.4. Реляційна модель проти графової: головні особливості.
  - 3.6. Підсумки.
    - 3.6.1. Чому не реляційна модель?
    - 3.6.2. Як зробити технологічний вибір для вашого застосунку C360.
- 4. Дослідження околу на етапі розробки.
  - 4.1. Стислий огляд: побудова реалістичнішого застосунку Customer 360.
  - 4.2. Моделювання графових даних 101.

- 4.2.1. *Це має бути вершина чи ребро?*
- 4.2.2. *Загубилися? Будемо рухатися за напрямком.*
- 4.2.3. *Граф без імені: поширені помилки в іменуванні.*
- 4.2.4. *Наша повністю розроблена графова модель.*
- 4.2.5. *Перед початком побудови.*
- 4.2.6. *Наші думки про важливість даних, запитів і кінцевого користувача.*
- 4.3. *Деталі реалізації для дослідження околів під час розробки.*
  - 4.3.1. *Більше даних для нашого розширеного прикладу.*
- 4.4. *Базова навігація у Gremlin.*
- 4.5. *Передовий Gremlin: оформлення результатів нашого запиту.*
  - 4.5.1. *Оформлення результатів запиту за допомогою кроків `project()`, `fold()` і `unfold()`.*
  - 4.5.2. *Усунення даних з результатів за допомогою патерну `where(neq())`.*
  - 4.5.3. *Планування стійких результатів за допомогою кроку `coalesce()`.*
- 4.6. *Рух від розробки до експлуатації.*
- 5. *Дослідження околу на етапі експлуатації.*
  - 5.1. *Стислий огляд: розподілені графові дані в Apache Cassandra.*
  - 5.2. *Робота з графовими даними в Apache Cassandra.*
    - 5.2.1. *Найважливіша для розуміння графового моделювання тема: первинні ключі.*
    - 5.2.2. *Ключі розбиття та локальність даних у розподіленому середовищі.*
    - 5.2.3. *Ребра, частина I: ребра у списках суміжності.*
    - 5.2.4. *Ребра, частина II: кластеризація стовпчиків.*
    - 5.2.5. *Ребра, частина III: матеріалізовані представлення обходу графа.*
  - 5.3. *Моделювання графових даних 201.*
    - 5.3.1. *Знаходження індексів за допомогою інтелектуальної індексної рекомендаційної системи.*
  - 5.4. *Деталії реалізації у фазі експлуатації.*
    - 5.4.1. *Матеріалізовані представлення та додавання часу до ребер.*
    - 5.4.2. *Наша кінцева схема для production застосунку C360.*
    - 5.4.3. *Оновлення наших запитів Gremlin для використання часу на ребрах.*
  - 5.5. *Переходимо до складніших задач на розподілених графах.*
    - 5.5.1. *Наші перші 10 порад для пререходу від фази розробки до експлуатації.*
- 6. *Використання дерев на етапі розробки.*
  - 6.1. *Стислий огляд: дерева, ієрархічні дані та цикли.*
  - 6.2. *Ієрархії та вкладені дані: три приклади.*
    - 6.2.1. *Ієрархічні дані у списку складових.*
    - 6.2.2. *Ієрархічні дані у системах контролю версій.*
    - 6.2.3. *Ієрархічні дані у самоорганізованих мережах.*

- 6.2.4. *Навіщо застосовувати графову технологію для ієрархічних даних?*
- 6.3. *Як не заблукати у термінологічному лісі.*
  - 6.3.1. *Дерева, корені, листя.*
  - 6.3.2. *Глибина вкладеності у маршрутах, ланцюгах і циклах.*
- 6.4. *Розуміння ієрархій з нашими сенсорними даними.*
  - 6.4.1. *Розуміння даних.*
  - 6.4.2. *Концептуальна модель у нотації GSL.*
  - 6.4.3. *Реалізація схеми.*
  - 6.4.4. *Перед початком побудови наших запитів.*
- 6.5. *Запити від листя до коренів у розробці.*
  - 6.5.1. *Куди цей сенсор надсилає інформацію?*
  - 6.5.2. *Який маршрут від цього сенсора до будь-якої вежі?*
  - 6.5.3. *Від “знизу до гори” до “згори додолу”.*
- 6.6. *Запити від коренів до листя у розробці.*
  - 6.6.1. *Налаштувальний запит: яка вежа має найбільше зв’язків з сенсорами, щоб ми могли дослідити її у нашому прикладі?*
  - 6.6.2. *Які сенсори з’єднані напряму з Джорджтауном?*
  - 6.6.3. *Знаходимо сенсори, з’єднані з Джорджтауном.*
  - 6.6.4. *Обмеження глибини рекурсії.*
- 6.7. *Подорож назад у часі.*
- 7. *Використання дерев на етапі експлуатації.*
  - 7.1. *Стислий огляд: степінь вершини, глибина та час на ребрах.*
  - 7.2. *Розуміння часу у сенсорних даних.*
    - 7.2.1. *Прикінцеві думки про часові ряди у графах.*
  - 7.3. *Розуміння степіні вершини у нашому прикладі.*
    - 7.3.1. *Що таке степінь вершини?*
    - 7.3.2. *Як приборкати степінь вершини.*
  - 7.4. *Схема експлуатації для наших сенсорних даних.*
  - 7.5. *Запити від листя до коренів на етапі експлуатації.*
    - 7.5.1. *Куди цей сенсор надіслав інформацію, та у який момент часу?*
    - 7.5.2. *Від цього сенсора знайти всі дерева, що ведуть до вежі, за часом.*
    - 7.5.3. *Від цього сенсора знайти дійсне дерево.*
    - 7.5.4. *Передовий Gremlin: розуміння патерну where().by().*
  - 7.6. *Запити від коренів до листя на етапі експлуатації.*
    - 7.6.1. *Які сенсори напряму з’єднані з Джорджтауном за часом?*
    - 7.6.2. *Які дійсні маршрути ми можемо знайти від Джорджтауна до всіх сенсорів?*
  - 7.7. *Застосування наших запитів за сценаріїв відмови вежі.*

- 7.7.1. *Кінцеві результати нашої складної проблеми.*
- 7.8. *Побачити ліс за деревами.*
- 8. *Знаходження маршрутів на етапі розробки.*
  - 8.1. *Стислий огляд: квантифікація довіри у мережах.*
  - 8.2. *Думки про довіру: три приклади.*
    - 8.2.1. *Наскільки ви довіряєте цьому відкритому запрошенню?*
    - 8.2.2. *Наскільки придатна до захисту версія звинувачення?*
    - 8.2.3. *Як компанії моделюють доставку відправлень?*
  - 8.3. *Фундаментальні концепції маршрутів.*
    - 8.3.1. *Найкоротші маршрути.*
    - 8.3.2. *Пошук в глибину та в ширину.*
    - 8.3.3. *Навчитися бачити застосовні задачі як задачі пошуку маршрутів.*
  - 8.4. *Знаходження маршрутів у мережах довіри.*
    - 8.4.1. *Початкові дані.*
    - 8.4.2. *Термінологічний вступ до Bitcoin.*
    - 8.4.3. *Створення схеми розвитку даних.*
    - 8.4.4. *Завантаження даних.*
    - 8.4.5. *Дослідження спільнот довіри.*
  - 8.5. *Обход графів з нашої мережі довіри Bitcoin.*
    - 8.5.1. *Які адреси знаходяться у першому околі?*
    - 8.5.2. *Які адреси знаходяться у другому околі?*
    - 8.5.3. *Які адреси знаходяться у другому околі, але не у першому?*
    - 8.5.4. *Стратегії обчислення з мовою запитів Gremlin.*
    - 8.5.5. *Обрання випадкової адреси для використання у нашому прикладі.*
  - 8.6. *Запити для знаходження найкоротшого маршруту.*
    - 8.6.1. *Знаходження маршрутів фіксованої довжини.*
    - 8.6.2. *Знаходження маршрутів довільної довжини.*
    - 8.6.3. *Доповнення наших маршрутів рейтингами довіри.*
    - 8.6.4. *Чи довіряєте ви цій особі?*
- 9. *Знаходження маршрутів на етапі експлуатації.*
  - 9.1. *Стислий огляд: ваги, відстані та спрощення графів.*
  - 9.2. *Зважені маршрути та алгоритми пошуку.*
    - 9.2.1. *Постановка задачі про пошук найкоротшого зваженого маршруту.*
    - 9.2.2. *Оптимізація пошуку найкоротшого зваженого маршруту.*
  - 9.3. *Нормалізація вагів ребер для задач пошуку найкоротшого маршруту.*
    - 9.3.1. *Нормалізація вагів ребер.*
    - 9.3.2. *Оновлення графу.*

- 9.3.3. *Дослідження нормалізованих реберних вагів.*
- 9.3.4. *Деякі думки перед початком конструювання запитів щодо найкоротших зважених маршрутів.*
- 9.4. *Запити про найкоротші зважені маршрути.*
  - 9.4.1. *Побудова запиту про найкоротший зважений маршрут на етапі експлуатації.*
- 9.5. *Зважені маршрути та довіра на етапі експлуатації.*
- 10. *Рекомендаційні системи на етапі розробки.*
  - 10.1. *Стислий огляд: колаборативна фільтрація для рекомендацій щодо кінострічок.*
  - 10.2. *Приклади рекомендаційних систем.*
    - 10.2.1. *Як ми отримуємо рекомендації щодо охорони здоров'я.*
    - 10.2.2. *Як ми отримуємо рекомендації у соціальних медіа.*
    - 10.2.3. *Як ми використовуємо тісно пов'язані дані для рекомендацій у електронній комерції.*
  - 10.3. *Вступ до колаборативної фільтрації.*
    - 10.3.1. *Розуміння задачі та домену.*
    - 10.3.2. *Колаборативна фільтрація графових даних.*
    - 10.3.3. *Рекомендації шляхом поелементної колаборативної фільтрації графових даних.*
    - 10.3.4. *Три різних моделі для ранжування рекомендацій.*
  - 10.4. *Дані щодо кінострічок: схема, завантаження та перегляд запиту.*
    - 10.4.1. *Модель даних для кінорекомендацій.*
    - 10.4.2. *Код схеми для кінорекомендацій.*
    - 10.4.3. *Завантаження даних кінострічок.*
    - 10.4.4. *Запити щодо околу у даних кінострічок.*
    - 10.4.5. *Деревоподібні запити у даних кінострічок.*
    - 10.4.6. *Запити про маршрути у даних кінострічок.*
  - 10.5. *Поелементна колаборативна фільтрація за допомогою Gremlin.*
    - 10.5.1. *Модель 1: Підрахунок кількості маршрутів у множині рекомендацій.*
    - 10.5.2. *Модель 2: NPS-натхнення.*
    - 10.5.3. *Модель 3: нормалізований NPS/*
    - 10.5.4. *Обираємо власну пригоду: кінострічки та графові задачі.*
- 11. *Просте розв'язання сутностей у графах.*
  - 11.1. *Стислий огляд: злиття декількох баз даних у один граф.*
  - 11.2. *Визначення відмінної складної задачі: розв'язання сутностей.*
    - 11.2.1. *Бачення складної задачі.*
  - 11.3. *Аналіз двох дйтасетів про кінострічки.*
    - 11.3.1. *MovieLens Dataset.*



11.3.2. *Kaggle Dataset.*

11.3.3. *Розробка схеми.*

11.4. *Узгодження та злиття даних про кінострічки.*

11.4.1. *Наш процес узгодження.*

11.5. *Резолюція помилок першого роду.*

11.5.1. *Помилки першого роду, знайдені в MovieLens Dataset.*

11.5.2. *Додаткові помилки, виявлені в процесі розв'язання сутностей.*

11.5.3. *Прикінцевий аналіз процесу злиття.*

11.5.4. *Роль графової структури у злитті даних про кінострічки.*

12. *Рекомендаційні системи у production.*

12.1. *Стислий огляд: розуміння ребер найкоротшого шляху, попереднє бчислення та спрощення графів.*

12.2. *Ребра найкоротшого шляху для рекомендацій у реальному часі.*

12.2.1. *Де наш процес розробки не масштабується.*

12.2.2. *Як ми лагодимо проблеми масштабування: ребра найкоротшого маршруту.*

12.2.3. *Що ми розробили для відправки у експлуатацію.*

12.2.4. *Спрощення графів: різні шляхи для попереднє бчислення ребер найкоротшого маршруту.*

12.2.5. *Міркування щодо оновлення наших рекомендацій.*

12.3. *Розрахунок ребер найкоротшого шляху для наших даних про кінострічки.*

12.3.1. *Розбиття складних задач на попередньо обчислені ребра найкоротшого маршруту.*

12.3.2. *Слон у посудній лавці: пакетні обчислення.*

12.4. *Експлуатаційна схема та завантаження даних для кінорекомендацій.*

12.4.1. *Експлуатаційна схема для кінорекомендацій.*

12.4.2. *Завантаження даних для кінорекомендацій на етапі експлуатації.*

12.5. *Рекомендаційні черги з ребрами найкоротшого маршруту.*

12.5.1. *Підтвердження того, що ребра завантажені правильно.*

12.5.2. *Рекомендації для користувача на етапі експлуатації.*

12.5.3. *Оцінка часу відгуку на етапі експлуатації шляхом підрахунку варіантів розбиття ребер.*

12.5.4. *Прикінцеві думки про продуктивність роботи запитів на розподілених графах.*

13. *Підсумки.*

13.1. *Напрямки подальшого розвитку спеціаліста.*

13.1.1. *Графові алгоритми*

13.1.2. *Розподілені графи.*

13.1.3. *Теорія графів.*

13.1.4. Теорія мереж.

13.2. Залишайтеся на зв'язку.

## 6 Самостійна робота студента/аспіранта

Студент витратить 3-4 година на тиждень на самостійну роботу з матеріалом курсу.

### Політика та контроль

## 7 Політика навчальної дисципліни (освітнього компонента)

Студенти отримують бали за правильне та вчасне виконання лабораторних робіт. Загальний рейтинг (кількість балів) складається з: 1) лабораторних робіт (у формі практичних завдань з програмування) 60%, 2) заліку 40%.

Наразі в курсі наявні три лабораторні роботи, кожне оцінюється до 20 балів. Студент повинен здати правильно виконану лабораторну роботу протягом двох тижнів з дня видачі завдання для отримання повної кількості балів, в іншому випадку застосовуються штрафні бали не більше 40% від загальної кількості за лабораторну роботу.

## 8 Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль: МКР, лабораторні роботи

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Семестровий контроль: залік

Умови допуску до семестрового контролю: зарахування усіх лабораторних робіт, семестровий рейтинг не менше 40 балів.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

## 9 Додаткова інформація з дисципліни (освітнього компонента)

- перелік питань, які виносяться на семестровий контроль (див. додаток до силабусу).

### Робочу програму навчальної дисципліни (силабус):

**Складено** Професор кафедри інженерії програмного забезпечення в енергетиці, д.е.н., професор А. О. Сігайов

**Ухвалено** кафедрою інженерії програмного забезпечення в енергетиці (протокол № 28 від 15 травня 2023 р.)

**Погоджено** Методичною комісією Навчально-наукового інституту атомної і теплової енергетики КПІ ім. Ігоря Сікорського (протокол № 9 від 26 травня 2023 р.)