

Лабораторна робота № 3. Графові СУБД: Neo4j

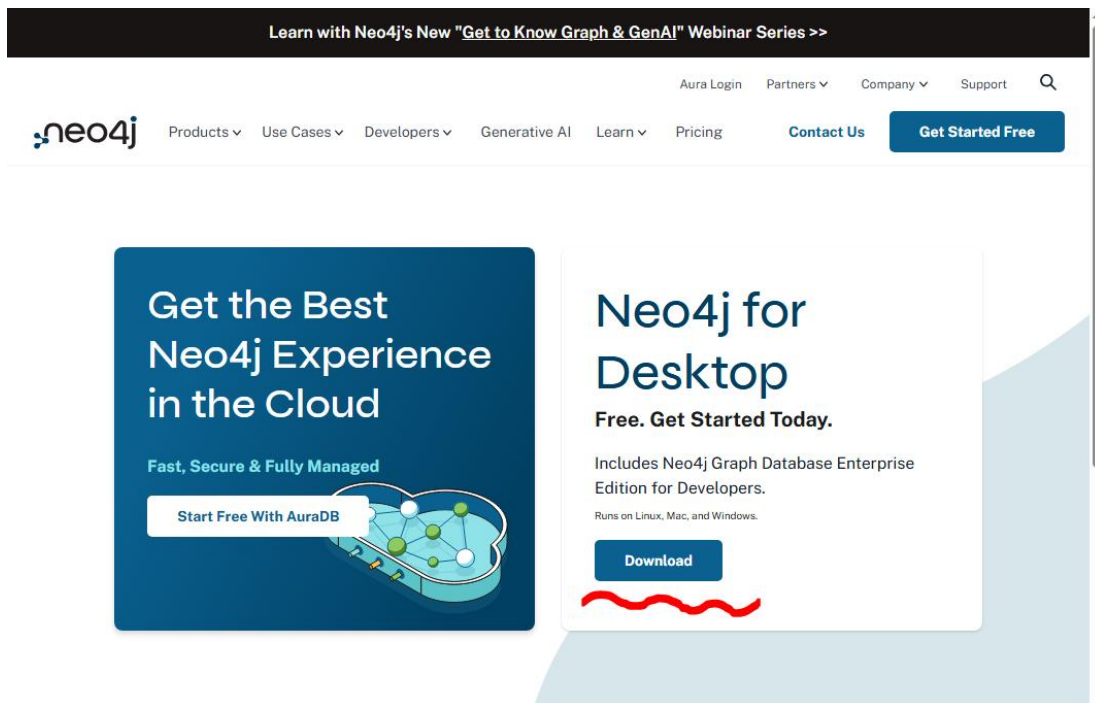
МЕТА РОБОТИ: отримати навички роботи з графовими базами даних.

ПОСТАНОВКА ЗАДАЧІ

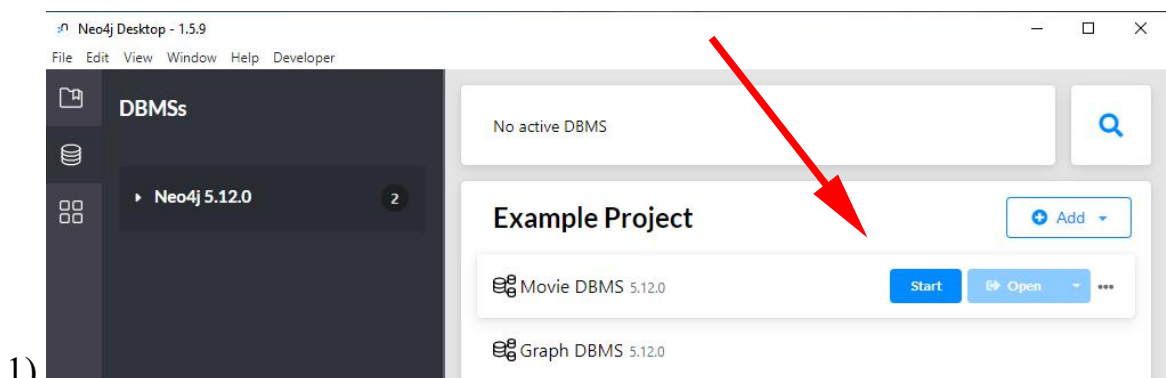
Реалізувати запити на пошук, додавання та видалення даних з навчальної бази даних **movies**, що поставляється разом з графовою СУБД Neo4j.

ПОРЯДОК ВИКОНАННЯ

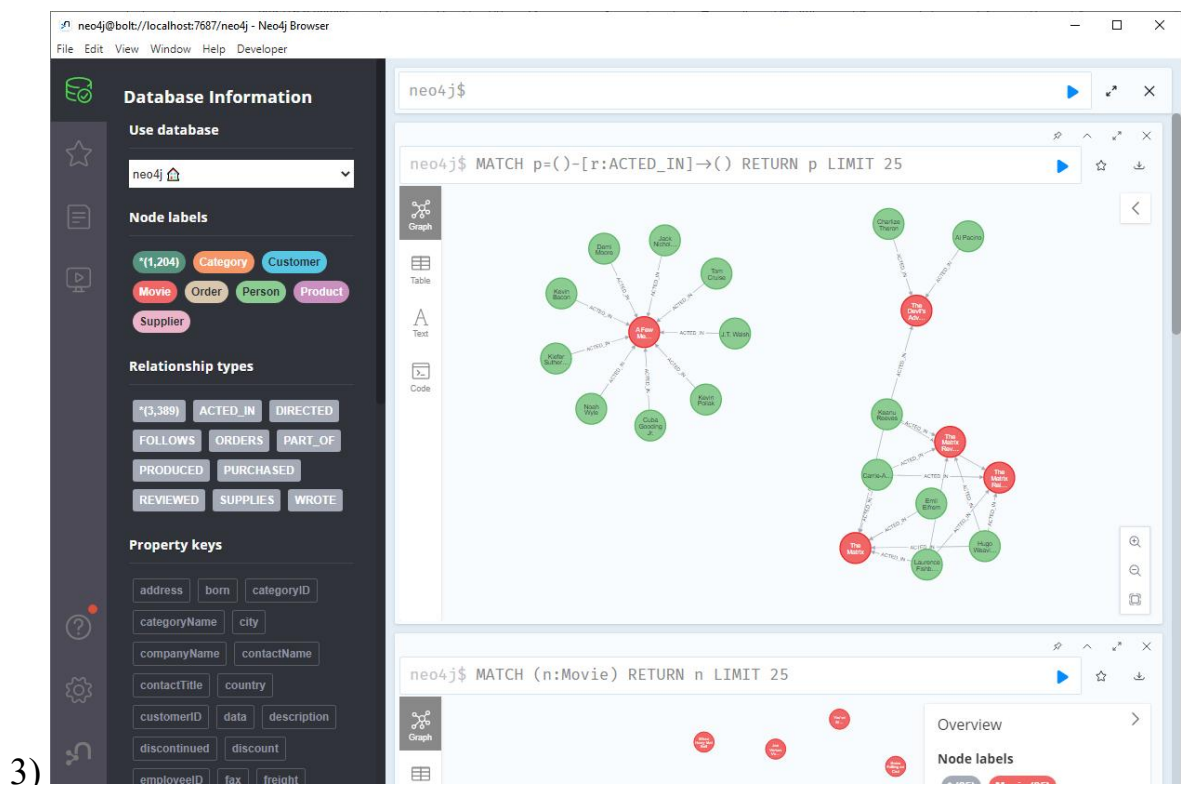
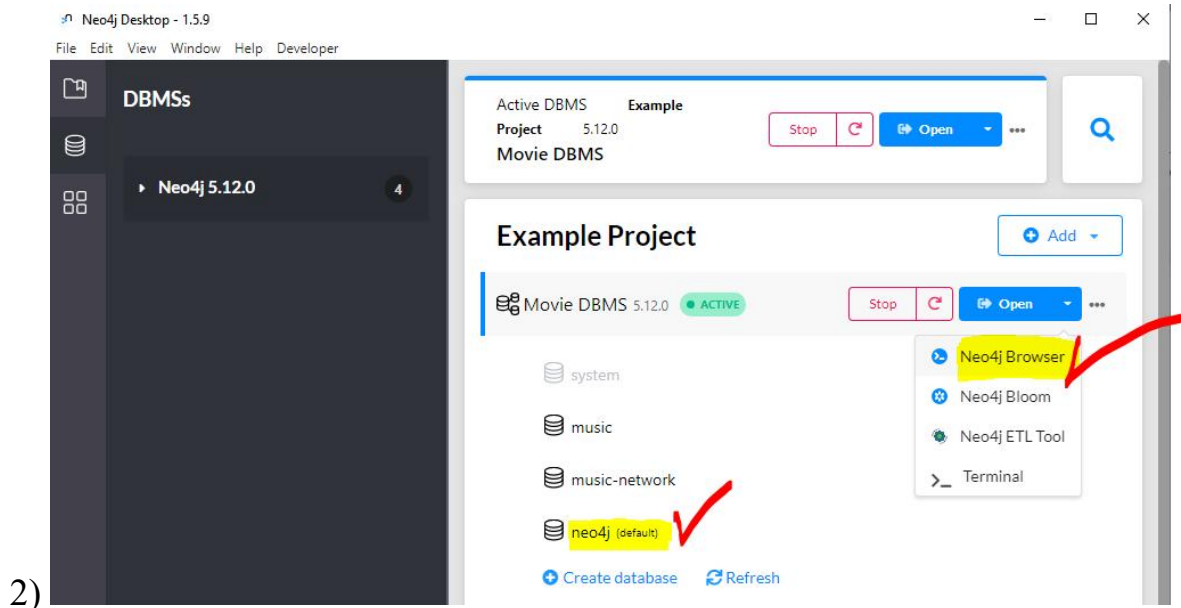
1. Встановити СУБД Neo4j з офіційного сайту <https://neo4j.com/download/> (для завантаження дистрибутиву вимагається реєстрація на сайті).



2. Використовуючи графічний інтерфейс Neo4j Browser (Neo4j Bloom) отримати навички написання запитів в графовій СУБД Neo4j на прикладі навчальної бази даних movies, виконавши розташовані нижче приклади запитів.



1)



ПРИКЛАДИ ЗАПИТІВ:

Запити на пошук даних:

1. Показати всіх акторів

```
MATCH (a:Person)-[:ACTED_IN]->(:Movie)
RETURN DISTINCT a.name AS Actor
LIMIT 10
```

Знаходимо всі унікальні імена людей, які знімалися у фільмах.

2. Знайти всі фільми, в яких грав "Tom Hanks"

```
MATCH (p:Person {name: "Tom Hanks"})-[:ACTED_IN]->(m:Movie)
RETURN m.title AS Movie, m.released AS Year
```

Пошук фільмів за актором.

3. Хто знімався в одному фільмі з "Tom Hanks"

```
MATCH (p:Person {name: "Tom Hanks"})-[:ACTED_IN]->(m:Movie)<-[:ACTED_IN]-(coActor)
WHERE coActor.name <> "Tom Hanks"
RETURN DISTINCT coActor.name AS CoActor
```

Знаходимо співакторів, які грали з Tom Hanks.

4. Скільки фільмів зняв кожен режисер

```
MATCH (d:Person)-[:DIRECTED]->(m:Movie)
RETURN d.name AS Director, count(m) AS MoviesDirected
ORDER BY MoviesDirected DESC
LIMIT 10
```

Агрегація: підрахунок фільмів на кожного режисера.

5. Показати фільми, в яких актор також був режисером

```
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)<-[:DIRECTED]-(p)
RETURN p.name AS ActorDirector, m.title AS Movie
```

Виводимо імена тих, хто одночасно був актором і режисером одного фільму.

6. Побудувати граф зв'язків навколо "Keanu Reeves"

```
MATCH (p:Person {name: "Keanu Reeves"})--(related)
RETURN p, related
```

Простий граф: вузол "Keanu Reeves" та всі пов'язані з ним.

7. Побудувати граф: актори й фільми, в яких вони грали

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
```

RETURN a, m
LIMIT 50

Простий граф для візуалізації відношень ACTED_IN.

8. Показати деталі фільму "The Matrix"

MATCH (m:Movie {title: "The Matrix"})
RETURN m.title AS Title, m.tagline AS Tagline, m.released AS Year

Дивимося опис фільму.

9. Показати всіх людей, пов'язаних з фільмом "The Matrix"

MATCH (p:Person)-->(m:Movie {title: "The Matrix"})
RETURN DISTINCT p.name AS Person

Усі, хто мав відношення до фільму (актор, режисер, тощо).

10. Побудувати повний граф фільму "The Matrix"

MATCH (p:Person)-->(m:Movie {title: "The Matrix"})
RETURN p, m

Візуалізуємо весь підграф.

Запити на додавання:

Примітка! "Example Movie" та "John Test" замінити на конкретні значення

1. Додати новий фільм "Example Movie"

CREATE (m:Movie {title: "Example Movie", released: 2025, tagline: "This is a test movie"})

2. Додати актора "John Test" та зв'язок ACTED_IN з фільмом

MATCH (m:Movie {title: "Example Movie"})
CREATE (p:Person {name: "John Test"})
CREATE (p)-[:ACTED_IN]->(m)

Важливо: переконайтесь, що фільм існує перед створенням зв'язку.

Запити на видалення:

1. Видалити зв'язки та акторів, пов'язаних із "Example Movie"

```
MATCH (p:Person)-[r:ACTED_IN]->(m:Movie {title: "Example Movie"})  
DETACH DELETE p
```

*Видаляє акторів та їхні зв'язки. **Обережно:** видаляються всі вузли Person, які грали **тільки** в цьому фільмі (не перевіряється участь у інших фільмах).*

2. Видалити сам фільм

```
MATCH (m:Movie {title: "Example Movie"})  
DETACH DELETE m
```

DETACH DELETE — автоматично видаляє вузол разом з усіма його зв'язками.

Примітка ! Якщо ви намагаєтесь видалити вузол без DETACH, і в нього є зв'язки — буде помилка. Завжди перевіряйте, чи не зачіпаєте інші об'єкти в базі!

3. Оформити звіт, в якому навести скріни, що демонструють результати виконання команд та відповіді на контрольні запитання.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. До якого типу NoSQL СУБД відноситься Neo4j? Які ще СУБД з цього класу є популярними на поточний момент? Аргументуйте свою відповідь, прикріпивши посилання на відповідні рейтинги.
2. В якому вигляді графові СУБД зберігають дані?
3. Яка мова запитів використовується в СУБД Neo4j?
4. Які типові варіанти застосування графових СУБД?
5. За рахунок чого в задачах на пошук оптимального шляху графові СУБД
6. З чим пов'язана досить висока швидкість виконання операцій пошуку найкоротшого шляху графових СУБД у порівнянні з реляційними?
7. Які основні недоліки можна виділити для графових СУБД?
8. Чи підтримує Neo4j механізм транзакцій?
9. Чи підтримує Neo4j механізм реплікацій та шардингу?