Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДЕНО:
Директор
А. В. Замятин

Оценочные материалы по дисциплине

Постреляционные базы данных

по направлению подготовки

09.03.03 Прикладная информатика

Направленность (профиль) подготовки:
Искусственный интеллект и большие данные

Форма обучения
Очная

Квалификация
Бакалавр

Год приема
2024

СОГЛАСОВАНО:
Руководитель ОП
С.П. Сущенко

Председатель УМК
С.П. Сущенко

Томск – 2024

1. Компетенции и индикаторы их достижения, проверяемые данными оценочными материалами

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-10 Способен решать задачи в профессиональной деятельности на основе информационной и библиографической культуры, цифровых технологий и систем искусственного интеллекта.

ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-10.1 Выбирает, применяет и адаптирует методы исследования для решения задач профессиональной деятельности с использованием систем искусственного интеллекта

ИОПК-2.1 Обладает необходимыми знаниями в области информационных технологий и программных средств, в том числе понимает принципы их работы

ИОПК-2.2 Применяет знания, полученные в области информационных технологий и программных средств, при решении задач профессиональной деятельности

ИОПК-2.3 Использует современные информационные технологии, в том числе отечественного производства на всех этапах разработки программных систем

2. Оценочные материалы текущего контроля и критерии оценивания

Элементы текущего контроля:

- тесты;
- защита лабораторных работ.

2.1.Примеры тестов

Пример теста (ИОПК-10.1)

1) Причинами появления нереляционных баз данных являются:
a) Резкое возрастание объемов информации
b) Широкое использование неструктурированных данных
c) Требования к унификации хранимой информации
d) Увеличение количества серверов с реляционными базами данных

2) ACID свойства (атомарность, согласованность, изолированность, долговечность) характеризуют реляционные БД, а какое название носят принципы для нереляционных БД?
a) CASE
b) BASE
c) ACROS
d) ACID

3) Достоинством нереляционных БД является:
a) Горизонтальное масштабирование
b) Вертикальное масштабирование
c) Диагональное масштабирование
d) Не масштабируются

4) Отсутствие схемы данных в NoSQL базах в отличие от реляционных структур данных не регламентирована — в отдельной строке или документе можно добавить произвольное поле без предварительного декларативного изменения структуры/

a) Верно
b) Неверно

5) Установка драйвера к БД в среде Python осуществляется командой:
   a) Pyp
   b) Pip
   c) Set driver
   d) Import

Пример теста (ИОПК-2.1)

1) Что представляет из себя MongoDB?
   a) это реляционная база данных
   b) это язык программирования для данных
   c) это система для управления базами данных
   d) это консольная программа, позволяющая получать данные

2) Правильная команда для создания базы данных в MongoDB
   a) create db_name
   b) using db_name
   c) make db_name
   d) use db_name

3) Выберите верную команду для создания коллекции в MongoDB:
   a) createCollection("some")
   b) db.createCollection(some)
   c) db.collection("some")
   d) db.createCollection("some")
   e) db.create("some")

4) Что необходимо запустить для старта MongoDB ?
   a) Программу Mongo Compass
   b) Mongoose
   c) Сервис MongoDB
   d) Mongo Atlass

5) Верное удаление коллекции в MongoDB:
   a) db.drop(some)
   b) db.some.delete()
   c) db.some.drop()
   d) some.drop()

Пример теста (ИОПК-2.2)

1) Для агрегирования данных в запросах на Cypher используется утверждение?
   a) GROUP BY
   b) ORDER BY
   c) RETURN()
   d) Нет специального утверждения

2) Для сортировки данных в запросе на Cypher используется:
   a) ORDER BY
   b) Такое же утверждение что и в SQL
   c) В утверждение RETURN указывается направление с помощью “->

d) Сортировка в запросах на Cypher невозможна

3) Индексы в Neo4j обновляются:
   a) Раз в 1 час
   b) По запросу пользователя
   c) При каждом изменении данных узлов
   d) Никогда

4) Для создания и удаления индекса в Neo4j используются команды:
   a) CREATE и DROP
   b) INSERT и REMOVE
   c) SET и DELETE
   d) CREATE и REMOVE

5) Когда команда импорта данных LOAD CSV WITH HEADERS FROM "file:///.....csv" в Neo4j загрузит данные неправильно?
   a) Если импортируемый файл лежит не в папке импорта
   b) Если в файле отсутствуют заголовки
   c) Всегда загружает правильно
   d) Если файл имеет другое расширение

Пример теста (ИОПК-2.3)

1) К какому классу относится БД Redis?
   a) Графовые базы данных
   b) Документно-ориентированные базы данных
   c) Колоночные базы данных
   d) Реляционные базы данных
   e) Хранилище «ключ-значение»

2) Для каких задач чаще всего используется REDIS?
   a) Кэширование
   b) Хранение больших массивов данных
   c) Организация чатов и обмена сообщениями
   d) Организация очередей
   e) Хранение сессий пользователей
   f) Аналитика в режиме реального времени
   g) Долгосрочное хранилище информации

3) В каких случаях БД Redis может использоваться для работы с медиа контентом?
   a) Хранение медиа контента
   b) Потоковая передача в режиме реального времени
   c) Хранение метаданных пользователей медиа сервисов
   d) Сохранение медиа файлов пользователей

4) Основными характеристиками БД Redis являются:
   a) Хранилище в памяти
   b) Высокая производительность
   c) Большой объем хранения данных
   d) Высокая степень сохранности данных

5) Срок существования ключей в Redis можно назначить как:
   a) Абсолютные значения времени в формате Unix

b) Оставшееся время существования в минутах
c) Оставшееся время существования в секундах
d) Нельзя задать срок существования ключей

Ключи: ИОПК-10.1 - 1) a,b 2) b 3) a 4) a) 5)b
ИОПК-2.1 – 1) c 2) d 3)d 4) c 5)c
ИОПК-2.2 – 1)d 2)a,b 3) c 4) a 5)b
ИОПК-2.3 - 1) e 2)a,c,d,e,f 3) b,c 4)a,b 5) a,c
Критерии оценивания: тест считается пройденным, если обучающий ответил правильно как минимум на половину вопросов.

Пример заданий для лабораторных работ (ИОПК-10.1, ИОПК-2.1, ИОПК-2.2, ИОПК-2.3)

Лабораторная работа. Работа с базой данных neo4j
Ход работы
Раздел 1 Создание БД Neo4j
1.1 Скачали с сайта https://neo4j.com/download/ и установили дистрибутив. Создали проект, базу данных (установили ей имя и пароль) и активировали ее.

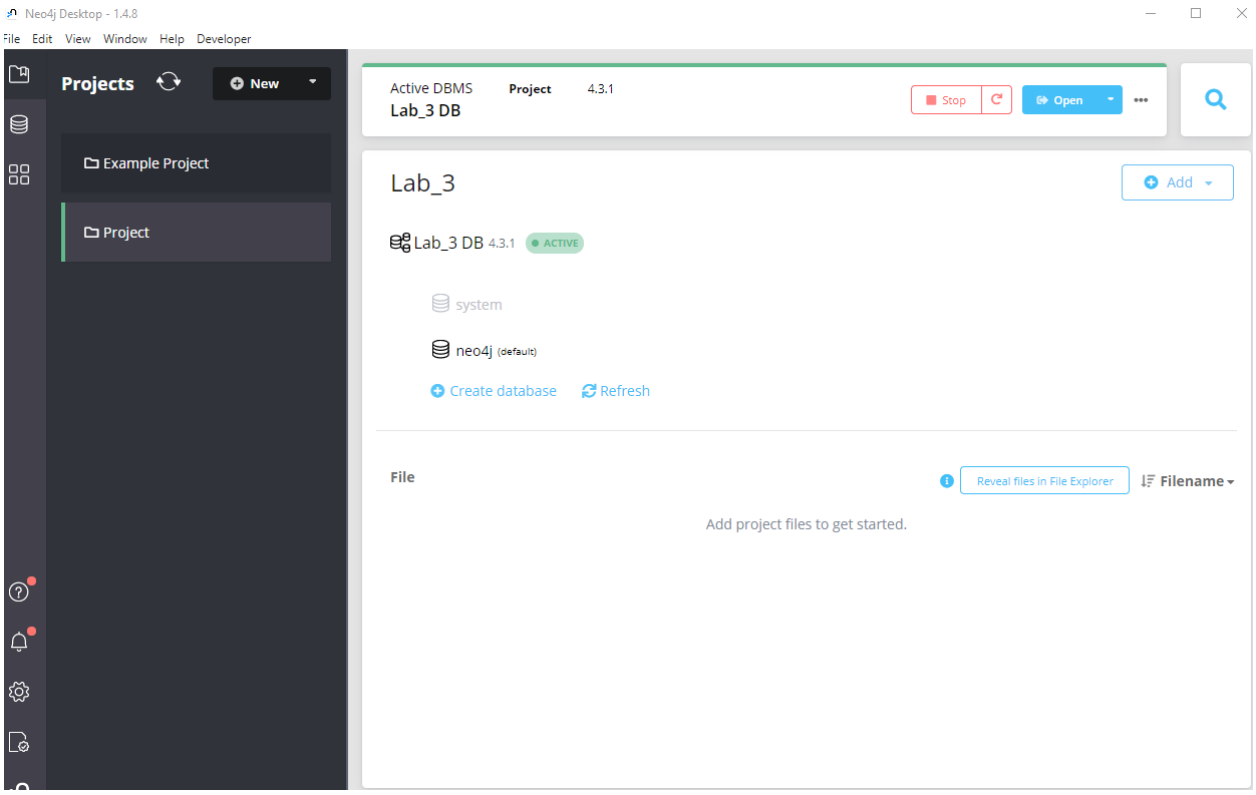

Рис. 1 БД с именем Lab_3 BD создана и активирована

1.2 С помощью СУБД Neo4j ввели команды для управления базой данных. В нашем случае база данных – это персонажи телесериала «Аббатство Даунтон». Использовали команды двух типов: для создания узлов графов (члены семьи аристократов, их личные слуги и животные) и для создания связей между ними (см. Таблица 1).

Таблица 1. Команды для создания нодов и связей

№	Команда
	Ноды
1	create(:noble{FN:'Robert',LN: 'Crawley'})
2	create(:noble{FN:'Cora',LN:'Crawley'})
3	create(:noble{FN:'MaryMary',LN:'Crawley'})

4	create(:noble{FN:'Edith',LN:'Crawley'})
5	create(:noble{FN:'Sybil',LN:'Crawley'})
6	create(:servant{FN:'John',LN: 'Bates'})
7	create(:servant{FN:'Anna',LN:'Bates'})
8	create(:servant{FN:'Sarah',LN:'O'Brain'})
9	create(:servant{FN:'Lawinia',LN:'Swire'})
10	create(:servant{FN:'Daisy',LN: 'Mason'})
11	create(:Pet{name:'Isida',specie:'dog'})
12	create(:Pet{name:'Ferdinand',specie:'horse'})
	Связи
1	match(n:noble{FN:'Robert'}),(m:noble{FN:'Cora'}) create (n)-[:married]->(m) return *
2	match(n:noble{FN:'Cora'}),(m:noble{FN:'Robert'}) create (n)-[:married]->(m) return *
3	match(n:servant{FN:'John'}),(m:servant{FN:'Anna'}) create (n)-[:married]->(m) return *
4	match(n:servant{FN:'Anna'}),(m:servant{FN:'John'}) create (n)-[:married]->(m) return *
5	match(n:noble{FN:'Robert'}),(m:noble{FN:'Mary'}) create (n)-[:father]->(m) return *
6	match(n:noble{FN:'Robert'}),(m:noble{FN:'Edith'}) create (n)-[:father]->(m) return *
7	match(n:noble{FN:'Robert'}),(m:noble{FN:'Sybil'}) create (n)-[:father]->(m) return *
8	match(n:noble{FN:'Mary'}),(m:noble{FN:'Robert'}) create (n)-[:daughter]->(m) return *
9	match(n:noble{FN:'Edith'}),(m:noble{FN:'Robert'}) create (n)-[:daughter]->(m) return *
10	match(n:noble{FN:'Sybil'}),(m:noble{FN:'Robert'}) create (n)-[:daughter]->(m) return *
11	match(n:noble{FN:'Cora'}),(m:noble{FN:'Mary'}) create (n)-[:mother]->(m) return *
12	match(n:noble{FN:'Cora'}),(m:noble{FN:'Edith'}) create (n)-[: mother]->(m) return *
13	match(n:noble{FN:'Cora'}),(m:noble{FN:'Sybil'}) create (n)-[: mother]->(m) return *
14	match(n:noble{FN:'Mary'}),(m:noble{FN:'Cora'}) create (n)-[:daughter]->(m) return *
15	match(n:noble{FN:'Edith'}),(m:noble{FN: 'Cora'}) create (n)-[:daughter]->(m) return *
16	match(n:noble{FN:'Sybil'}),(m:noble{FN:'Cora'}) create (n)-[:daughter]->(m) return *
17	match(n:noble{FN:'Mary'}),(m:Pet{name:'Ferdinand'}) create (n)-[:owns]->(m) return *
18	match(n:noble{FN:'Robert'}),(m:Pet{name:'Isida'}) create (n)-[:owns]->(m) return *
19	match(n:noble{FN:'Robert'}),(m:servant{FN:'John'}) create (n)-[:master]->(m) return *
20	match(n:servant{FN:'John'}),(m:noble{FN:'Robert'}) create (n)-[:valet]->(m) return *
21	match(n:noble{FN:'Cora'}),(m:servant{FN:'Sarah'}) create (n)-[:master]->(m) return *
22	match(n:servant{FN:'Sarah'}),(m:noble{FN:'Cora'}) create (n)-[:chambermaid]->(m) return *
23	match(n:noble{FN:'Mary'}),(m:servant{FN:'Anna'}) create (n)-[:master]->(m) return *
24	match(n:servant{FN:'Anna'}),(m:noble{FN:'Mary'}) create (n)-[:chambermaid]->(m) return *
25	match(n:noble{FN:'Edith'}),(m:servant{FN:'Lawinia'}) create (n)-[:master]->(m) return *
26	match(n:servant{FN:'Lawinia'}),(m:noble{FN:'Edith'}) create (n)-[:chambermaid]->(m) return *
27	match(n:noble{FN:'Sybil'}),(m:servant{FN:'Daisy'}) create (n)-[:master]->(m) return *
28	match(n:servant{FN:'Daisy'}),(m:noble{FN:'Sybil'}) create (n)-[:chambermaid]->(m) return *

Сначала сгенерировали 3 вида сущностей: nobel – члены семьи аристократов, servant – их слуги, pet – их животные.
Потом построили последовательно связи по одной:

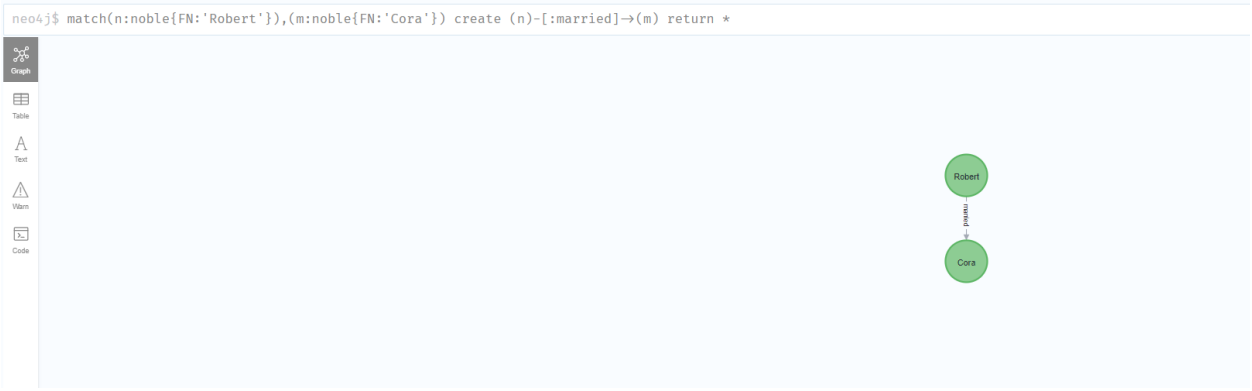

Рис. 2 Построение связи

Далее свели связи в одну схему:

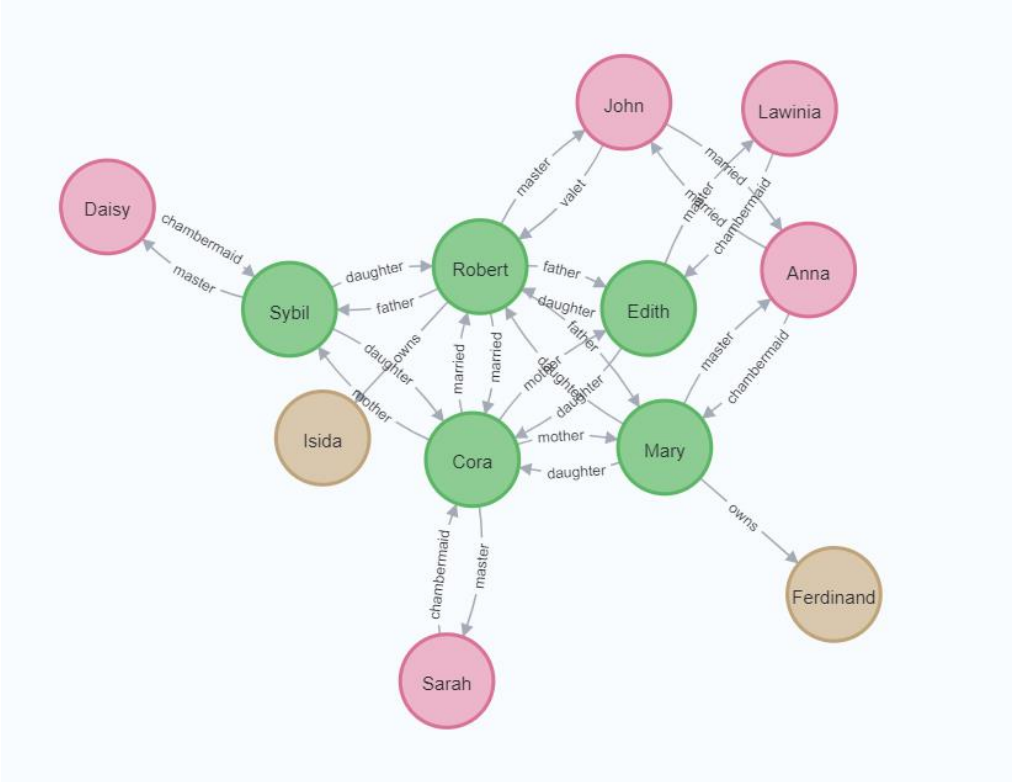

Рис. 3 Узлы со связями телесериала «Аббатство Даунтон»

Раздел 2 Импорт данных в Neo4j Desktop

2.1 С сайта Kaggle был скачан архив с датасетами и скопирован в папку импорта:

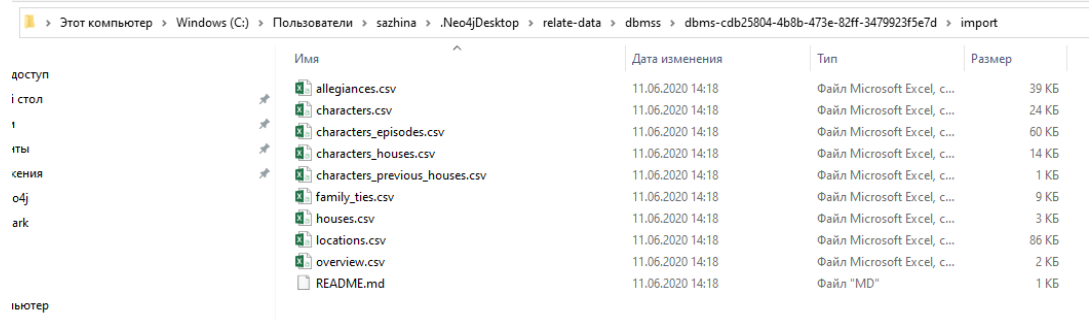

Рис. 4 Путь сохранения данных

2.2 При помощи команды LOAD CSV загрузили данные, установили связи между созданными узлами (выполнили несколько запросов для проверки правильности импорта).

```
LOAD CSV WITH HEADERS FROM "file:///overview.csv" AS EP CREATE (ep:Episode{epid
:EP.episodeId, season:EP.season, episode:EP.episode, title:EP.title})

LOAD CSV WITH HEADERS FROM "file:///characters.csv" AS CH CREATE (ch:Character{li
nk:CH.link, Name:CH.character })

LOAD CSV WITH HEADERS FROM "file:///characters_episodes.csv" AS Part CREATE
(Pt:Participation{eId:Part.episodeId, char_link:Part.character})

MATCH (e:Episode),(ch:Character),(p:Participation) where e.epId=p.eId and
p.char_link=ch.link create (ch)-[:take_part]->(e)
LOAD CSV WITH HEADERS FROM "file:///houses.csv" AS HS CREATE (hs:House{name:H
S.name, link:HS.link})

LOAD CSV WITH HEADERS FROM "file:///characters_houses.csv" AS Rel CREATE (r:Relat
ion{char_link:Rel.character, house_link:Rel.house})

MATCH (ch:Character),(h:House),(r:Relation) where h.link=r.house_link and
r.char_link=ch.link create (ch)-[:lives]->(h)
LOAD CSV WITH HEADERS FROM "file:///characters_episodes.csv" AS Part MATCH
(e:Episode),(ch:Character) where e.epId=Part.episodeId and Part.character=ch.link create (ch)-
[:take_part]->(e)
MATCH (e:Episode),(ch:Character) where ch.Name="Robb Stark" return *
MATCH (h:House),(ch:Character) where h.Name='House Mormont' return *
```

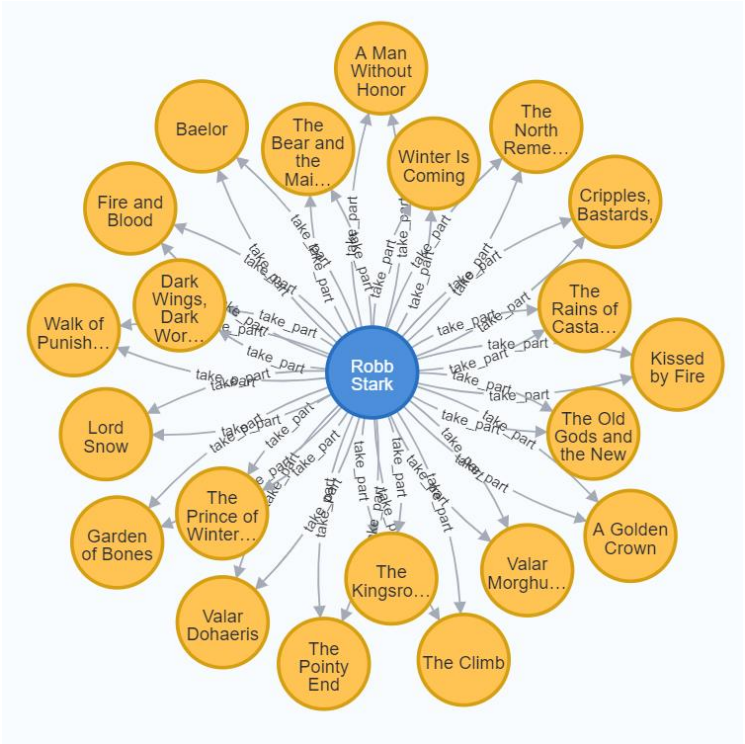

Рис. 5 Эпизоды, в которых фигурирует Robb Stark

```
neo4j$ MATCH (hs:House),(ch:Character) where ch.Name='Robb Stark' return *
```

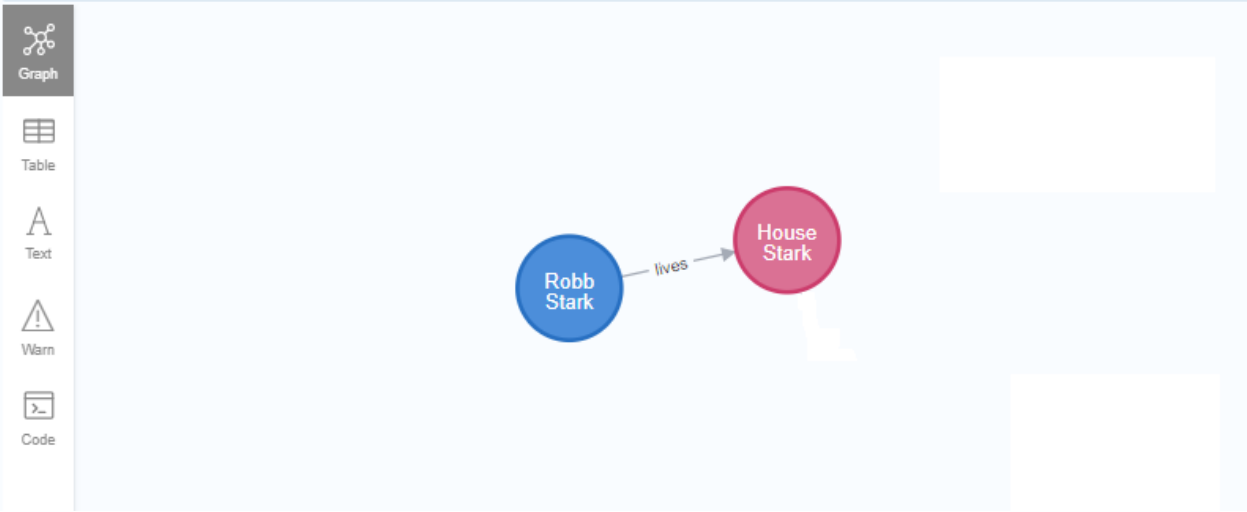

Рис. 6 Дома, которым принадлежит Robb Stark

Рис. 7 Персонажи, принадлежащие дому Lannister

Раздел 3 Подключение к Neo4j через Python

3.1 Для подключения через Python установили необходимый драйвер и библиотека py2neo.

```
In [7]: pip install py2neo
```

```
In [8]: pip install neo4j
```

```
In [1]: from py2neo import Graph
        from neo4j import GraphDatabase
        from neo4j import __version__ as neo4j_version
        print(neo4j_version)
```

```
4.3.7
```

Рис. 8 Импорты

3.2 Выполнили ряд запросов, добавили членов семьи Кроули (из Раздела 1) и установили связи между ними (связь между отцом Робертом Кроули и его дочерью Мэри Кроули). Также выполнили запрос, выводящий список всех персонажей.

```
In [8]: #create driver
uri = "bolt://localhost:7687"
driver = GraphDatabase.driver(uri, auth=("neo4j", "Qwerty"))

#srart session
session = driver.session()

#Add Robert Crawley
session.run("create(:noble{FN:'Robert',LN:'Crawley'})")

#Add relation Robert Crawley-married->Cora Robert Crawley
session.run("match(n:noble{FN:'Robert'}),(m:noble{FN:'Cora'}) create (n)-[:married]->(m) return *;")

#Add Mary Crawley
session.run("create(:noble{FN:'Mary',LN:'Crawley'})")

#Add relations Mary Crawley-daughter->Robert Crawley, Robert Crawley-father->Mary Crawley
session.run("match(n:noble{FN:'Mary'}),(m:noble{FN:'Robert'}) create (n)-[:daughter]->(m) return *;")
session.run("match(n:noble{FN:'Robert'}),(m:noble{FN:'Mary'}) create (n)-[:father]->(m) return *;")

#Get list of persons
result = session.run("MATCH (a:noble) RETURN a.FN AS name")
names = [record["name"] for record in result]
print(names)

#Close session and close driver
session.close()
driver.close()

['Robert', 'Mary']
```

Рис. 9 Код запросов на языке Python

3.3 Проверили правильность запросов в Neo4j Desktop.

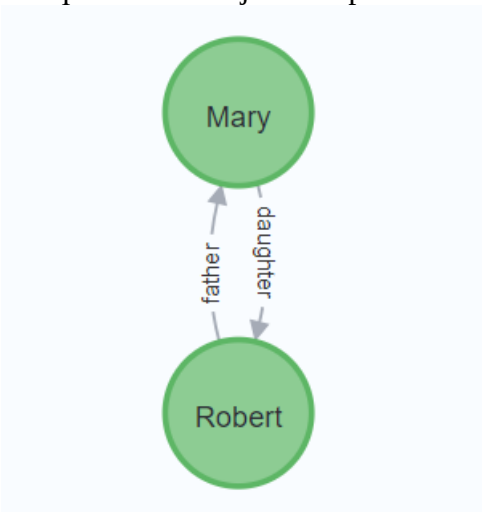

Рис. 10 Добавлены два персонажа и связи между ними

Вывод

В ходе данной лабораторной работы было настроено программное обеспечение для работы с Neo4j, а также самостоятельно импортированы dataset’ы и построены простые запросы к данным.

Оценочные материалы итогового контроля (промежуточной аттестации) и критерии оценивания

Схема определения итоговой балльно-рейтинговой оценки

	Виды учебной деятельности	**Виды оценивания (балльные оценки)**				
1	Изучение дисциплины (форма контроля - **зачет, диф. зачет**)	Мероприятия текущего контроля в семестре			=	Итоговая рейтинговая оценка
		Максимум 100 баллов				
2	Изучение дисциплин (форма контроля - **экзамен**)	Мероприятия текущего контроля в семестре	+	Мероприятия промежуточной аттестации (экзамен)	=	Итоговая рейтинговая оценка
		Максимум 80 баллов		Максимум 20 баллов		

Тематический план курса

№	**Модуль, тема**	**Вид занятия (ЛК/ЛБ)**	**Максимальное количество баллов**	
			Зачет, дифзачет	**Экзамен**
1	**Тема 1. Проектирование хранилищ данных**			
	Тема 1.1 Определение понятия «хранилище данных». Сравнение систем OLTP и хранилищ данных. Проблемы разработки и сопровождения хранилищ данных. Архитектура хранилища данных. Информационные потоки в хранилище данных. Инструменты и технологии хранилищ данных. Магазины данных..	ЛК		
	Тема 1.2 Проектирование базы данных для хранилища данных. Моделирование размерностей. Методика проектирования базы данных для хранилища данных. Критерии оценки размерностей хранилища данных. Проектирование хранилища данных с использованием современных средств	ЛК		
	Лабораторная работа № 1.	ЛБ	20	15
2	**Тема 2. OLAP-технология.**			
	Тема 2.1. Оперативная аналитическая обработка данных (OLAP). Приложения OLAP. Преимущества OLAP. Представление многомерных данных..	ЛК		
	Тема 2.2. Инструменты OLAP. Категории инструментов OLAP. Расширения языка SQL для поддержки OLAP.	ЛК		
	Лабораторная работа № 2.	ЛБ	20	15
3	**Тема 3. Объектно-ориентированная и объектно-реляционная модели данных.**			

№	Модуль, тема	Вид занятия (ЛК/ЛБ)	Максимальное количество баллов	
			Зачет, дифзачет	Экзамен
	Тема 3.1 Объектно-ориентированная и объектно-реляционная модели данных. Основные понятия. Сравнение этих моделей с реляционной моделью данных и с объектно-ориентированной парадигмой программирования.	ЛК		
	Тема 3.2 Реализация объектно-реляционной модели данных.	ЛК		
	Лабораторная работа № 3.	ЛБ	20	15
4	**Тема 4. NOSQL-модели данных.**			
	Тема 4.1 Обзор и классификация NOSQL-моделей и инструментов долговременного хранения данных.	ЛК		
	Тема 4.2 Key-Value Stores, Wide Column Stores, Document Stores, Graph DBMS, RDF Stores, Native XML DBMS, Content Stores, Search Engines. Особенности этих моделей.	ЛК		
	Лабораторная работа № 4.	ЛБ	15	15
	Лабораторная работа № 5.	ЛБ	15	10
	Лабораторная работа № 6.	ЛБ	10	10
	ЭКЗАМЕН (ФИНАЛЬНЫЙ ПРОЕКТ)			20
	ИТОГО		**100**	**100**

Критерии оценки отчетов по лабораторным работам и финальному проекту (экзамену)

№	Критерий	Максимальный балл
		Экзамен
1.	На проверку преподавателю представлено:	
	1) файл-отчет с исходным кодом	1,5
	2) отчет в формате doc или pdf	2
	3) все массивы данных, на основе которых была проведена работа (можно в архиве). или ссылка на открытый источник данных	1
2.	Отчет содержит подробные комментариями к коду и полученным результатам.	2
4.	Отчет содержит общий вывод по работе.	3
5.	Общий вывод по работе отражает не только факты (выполнено, построено и т.п.), но и аргументированные выводы.	3
7.	Все выводы построены грамотно, аргументированно.	3
8.	В программном коде в отчете нет ошибок.	0,5
Выбор: устная защита или оценка по отчету		

№	Критерий	Максимальный балл
		Экзамен
9.1	При защите студент ответил на все дополнительные вопросы	2
9.2	В целом, отчет выполнен аккуратно, порядок этапов выполнения работы логичен.	2
	ВСЕГО:	**20**

3. Оценочные материалы для проверки остаточных знаний (сформированности компетенций)

3.1. Тест

1) Причинами появления нереляционных баз данных являются:
a) Резкое возрастание объемов информации
b) Широкое использование неструктурированных данных
c) Требования к унификации хранимой информации
d) Увеличение количества серверов с реляционными базами данных

2) ACID свойства (атомарность, согласованность, изолированность, долговечность) характеризуют реляционные БД, а какое название носят принципы для нереляционных БД?
a) CASE
b) BASE
c) ACROS
d) ACID

3) Достоинством нереляционных БД является:
a) Горизонтальное масштабирование
b) Вертикальное масштабирование
c) Диагональное масштабирование
d) Не масштабируются

4) Отсутствие схемы данных в NoSQL базах в отличие от реляционных структур данных не регламентирована — в отдельной строке или документе можно добавить произвольное поле без предварительного декларативного изменения структуры
a) Верно
b) Неверно

5) Установка драйвера к БД в среде Python осуществляется командой:
a) Pyp
b) Pip
c) Set driver
d) Import

6) К какому классу относится БД Redis?
a) Графовые базы данных
b) Документно-ориентированные базы данных
c) Колоночные базы данных
d) Реляционные базы данных
e) Хранилище «ключ-значение»

7) Для каких задач чаще всего используется REDIS?

a) Кэширование
b) Хранение больших массивов данных
c) Организация чатов и обмена сообщениями
d) Организация очередей
e) Хранение сессий пользователей
f) Аналитика в режиме реального времени
g) Долгосрочное хранилище информации

8) В каких случаях БД Redis может использоваться для работы с медиа контентом?

a) Хранение медиа контента
b) Потоковая передача в режиме реального времени
c) Хранение метаданных пользователей медиа сервисов
d) Сохранение медиа файлов пользователей

9) Основными характеристиками БД Redis являются:

a) Хранилище в памяти
b) Высокая производительность
c) Большой объем хранения данных
d) Высокая степень сохранности данных

10) Срок существования ключей в Redis можно назначить как:

a) Абсолютные значения времени в формате Unix
b) Оставшееся время существования в минутах
c) Оставшееся время существования в секундах
d) Нельзя задать срок существования ключей

Ключи: 1) a,b 2) b 3) a 4) a) 5)b 6) e 7)a,c,d,e,f 8) b,c 9)a,b 10) a,c

Информация о разработчиках

Мокина Елена Евгеньевна, старший преподаватель кафедры теоретических основ информатики ТГУ