Государственное бюджетное профессиональное образовательное учреждение города Москвы

«Московский государственный колледж электромеханики и

информационных технологий»

(ГБПОУ МГКЭИТ)

Отчет по практической работе № 1

Работа с официальной документацией различных СУБД

Выполнил студент гр.: 3 ИП11-19

Кочержина С. А.

**ВВЕДЕНИЕ**

Цель – получить актуальные знания по разным СУБД.

Актуальность – важность выбора СУБД при разработке БД.

Задачи:

1) анализ назначения СУБД,

2) анализ основных возможностей СУБД,

3) анализ типов данных в СУБД,

4) анализ языка запроса в СУБД.

Предмет исследования – исследование СУБД.

Объект использования – исследование Redis, Neo4j, PostgreSQL.

**ОСНОВНАЯ ЧАСТЬ**

**1 REDIS**

**1.1 Назначение**

Redis – это хранилище структур данных в памяти с открытым исходным кодом (под лицензией BSD), используемое в качестве базы данных, кеша и брокера сообщений.

**1.2 Основные возможности**

Redis позволяет хранить не только строки, но и массивы (которые могут использоваться в качестве очередей или стеков), словари, множества без повторов, большие массивы бит, а также множества, отсортированные по некой величине.

**1.3 Типы данных**

Redis предоставляет такие структуры данных, как [строки](https://redis.io/topics/data-types-intro#strings), [хэши](https://redis.io/topics/data-types-intro#hashes), [списки](https://redis.io/topics/data-types-intro#lists), [наборы](https://redis.io/topics/data-types-intro#sets), [отсортированные наборы](https://redis.io/topics/data-types-intro#sorted-sets) с запросами диапазона, [точечные рисунки](https://redis.io/topics/data-types-intro#bitmaps), [гиперлоги](https://redis.io/topics/data-types-intro#hyperloglogs), [геопространственные индексы](https://redis.io/commands/geoadd) и [потоки](https://redis.io/topics/streams-intro).

**1.4 Язык запроса**

Кэширование данных.

Запись данных в хранилище.

**public** **function** **setValueToCache**(string $key, $value)

{

**$this**->getRedis()->rawCommand('SET', $key, $value);

}

Получение данных из хранилища

**public** **function** **getValueFromCache**(string $key)

{

**return** **$this**->getRedis()->rawCommand('GET', $key);

}

Очереди

Отправление задачи в очередь

**public** **function** **pushToQueue**(string $queueName, $payload)

{

**$this**->getRedis()->rawCommand('RPUSH', $queueName,erialize($payload));

}

Также можно писать в консоли

Создание, выборка, модификация, удаление и базовая информация об объектах.

redis 127.0.0.1:6379> set test:1:string "my binary safe string"

OK

redis 127.0.0.1:6379> get test:1:string

"my binary safe string"

redis 127.0.0.1:6379> getset test:1:string "other value"

"my binary safe string"

redis 127.0.0.1:6379> type test:1:string

string

redis 127.0.0.1:6379> set test:1:vlaue "487"

OK

redis 127.0.0.1:6379> rename test:1:vlaue test:1:value

OK

redis 127.0.0.1:6379> exists test:1:vlaue

(integer) 0

redis 127.0.0.1:6379> exists test:1:value

(integer) 1

redis 127.0.0.1:6379> keys test:1:\*

1) "test:1:string"

2) "test:1:value"

redis 127.0.0.1:6379> del test:1:value

(integer) 1

redis 127.0.0.1:6379> keys test:1:\*

1) "test:1:string"

**2 Neo4j**

**2.1 Назначение**

Neo4j предоставляет декларативный язык запросов для визуального представления графика с использованием синтаксиса ascii-art.

**2.2 Основные возможности**

Neo4j, работающий на базе собственной графовой базы данных, хранит данные и управляет ими в их более естественном, подключенном состоянии, поддерживая отношения между данными, которые доставляют молниеносные запросы, более глубокий контекст для аналитики и легко изменяемую модель данных.

**2.3 Типы данных**

Язык запроса графа является Cypher.

Cypher обеспечивает первоклассную поддержку для ряда типов данных.

Они делятся на несколько категорий, которые будут подробно описаны в следующих подразделах: типы недвижимости, структурные типы, составные типы.

Типы недвижимости включают:

1. Number, абстрактный тип, который имеет подтипы Integer и Float,
2. String,
3. Логический,
4. Пространственный тип Point,
5. Временные типы: Date, Time, LocalTime, DateTime, LocalDateTime и Duration.

К структурным типам относятся:

Узлы, в состав которых входят:

1. Id,
2. Этикетка,
3. Карта (собственности).

Отношения, включающие:

1. Id,
2. Type,
3. Карта (собственности),
4. Идентификатор начального и конечного узлов.

Пути.

Составные типы включают:

1. Списки представляют собой разнородные упорядоченные наборы значений, каждое из которых имеет какое-либо свойство, структурный или составной тип,
2. Карты представляют собой разнородные неупорядоченные коллекции пар (ключ, значение), где: ключ - это строка, значение имеет любое свойство, структурный или составной тип.

**2.4 Язык запроса**

Cypher заимствует свою структуру из SQL - запросы строятся с использованием различных предложений. Предложения объединены в цепочку и передают друг другу промежуточные наборы результатов. Язык запросов состоит из нескольких отдельных предложений.

**3 PostgreSQL**

**3.1 Назначение**

PostgreSQL предоставляет командный интерфейс для работы с системным каталогом, с помощью которого можно не только получать информацию об объектах системы, но и создавать новые.

**3.2 Основные возможности заключаются в:**

1. Поддержке БД неограниченного размера,
2. Мощные и надежные транзакции и репликации,
3. Расширяемая система встроенных языков программирования и поддержки загрузки С-совместные модулей,
4. Наследование,
5. Легкая расширяемость.

**3.3 Типы данных**

PostgreSQL предоставляет пользователям богатый ассортимент встроенных типов данных. Кроме того, пользователи могут создавать свои типы в PostgreSQL, используя команду [CREATE TYPE](https://postgrespro.ru/docs/postgresql/9.4/sql-createtype). Основными являются:

1. Числовые типы,
2. Денежные типы,
3. Символьные типы,
4. Двоичные типы данных,
5. Типы даты/времени,
6. Логические типы,
7. Типы перечислений,
8. Геометрические типы,
9. Типы, описывающие сетевые адреса,
10. Битовые строки,
11. Типы, предназначенные для текстового поиска,
12. Тип UUID,
13. Тип XML,
14. Типы JSON,
15. Массивы,
16. Составные типы,
17. Диапазонные типы,
18. Идентификаторы объектов,
19. Псевдонимы.

**3.4 Язык запроса**

В основе лежит синтаксис языка SQL, дальше создается структура для хранения данных, после идет сам запрос.

**ЗАКЛЮЧЕНИЕ**

В процессе выполнения практической работы были получены актуальные знания по разным СУБД.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1) <https://redis.io/>

2) <https://habr.com/ru/company/manychat/blog/507136/>

3) <https://habr.com/ru/post/204354/>

4) <https://redis.io/commands#generic>

5) <https://neo4j.com/>

6) <https://neo4j.com/docs/developer-manual/current/cypher/syntax/values/>

7)<https://coderlessons.com/tutorials/bazy-dannykh/uznaite-neo4j/neo4j-kratkoe-rukovodstvo>

8) <https://neo4j.com/docs/developer-manual/current/cypher/#cypher-intro>

9) <https://www.postgresql.org/>

10) <https://habr.com/ru/company/pgdayrussia/blog/331460/>

11) <https://postgrespro.ru/docs/postgresql/9.4/datatype>

12) <https://postgrespro.ru/docs/postgresql/9.4/sql>