

## КЛАСТЕРИЗАЦИЯ В ГРАФОВЫХ СИСТЕМАХ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

### CLUSTERING IN GRAPHIC DATABASE MANAGEMENT SYSTEMS

*O. Ogorodnikova  
A. Dubrovin*

*Summary.* The purpose of the article is to study database management systems. Graphic databases are considered. Data Mining technology must be used in graph databases. In particular, the clustering method. The possibilities of graph DBMS Neo4j for the implementation of clustering methods are investigated.

*Keywords:* database management system, data mining, clustering method.

**Огородникова Ольга Викторовна**

Старший инженер группы автоматизации,  
ФКУ «Следственный изолятор № 2» УФСИН России  
по г. Санкт-Петербургу и Ленинградской области;  
адъюнкт, Воронежский институт ФСИН России  
*Olga-ogogordnikova@yandex.ru*

**Дубровин Анатолий Станиславович**

Д.т.н., Воронежский институт ФСИН России  
*asd\_kiziltash@mail.ru*

*Аннотация.* Цель статьи заключается в исследовании систем управления базами данных. Рассмотрены графовые базы данных как наиболее актуальные для работы с большими объемами данных. Технология Data Mining необходимо использовать в графовых базах данных. В частности, рассматривается метод кластеризации. Исследованы возможности графовой СУБД Neo4j для реализации методов кластеризации.

*Ключевые слова:* система управления базами данных, графовая база данных, интеллектуальный анализ данных, метод кластеризации.

### Введение

**В** последнее время реляционные системы управления базами данных начинают вытеснять NoSQL СУБД. Замена устоявшихся и понятных СУБД может быть осуществлена в связи со значительной практической пользой. Модели данных могут быть переведены на графовые, когда необходимо увеличить производительность, когда необходимо создавать сложные и гибкие запросы. Рассмотрим системы управления графовыми базами данных (далее — графовые базы данных). С помощью графовых баз данных стало возможно моделировать базы данных по-новому, используя мощные инструменты. Графовые базы данных дают возможность представлять большие и сложные зависимости, что невозможно реализовать на SQL языке запросов, а также осуществляют хранение взаимосвязей и навигацию в них. Графовые базы данных являются универсальными: изменение требований и функциональных возможностей не приведет к проблемам. Базы данных могут меняться и модифицироваться. Так графы пришли на смену таблицам (совокупности таблиц) в реляционных моделях. Графовая база данных хранит данные в виде интеллектуальной схемы, которая позволяет легко найти и построить любые отношения в виде графа с узлами и ребрами. Все это дает возможность хранить и работать со связанными данными. В данном случае связь между данными является более ценной, чем сами данные. Дан-

ные хранятся эффективно путем записи узлов и отношений, близких друг к другу. В конечном итоге предъявляется меньше технических требований к оборудованию, но при этом выполнение запросов происходит более быстро.

### Data Mining в графовых базах данных

Информация, которая содержится в СУБД, нередко содержит ошибки и неточности. Как правило, вследствие неточного ввода данных растет число ненужных записей, бессмысленных и ложных результатов, что приводит в процессе поиска и анализа информации к значительным проблемам. Справиться с этим упущением поможет интеллектуальный анализ данных. Data Mining — технология, которая обнаруживает знания в данных. Происходит «раскопка» данных, «промывка» данных.

Рассмотрим возможности технологии обнаружения знаний. Кластерный анализ является наиболее сложной задачей Data Mining. Поиск алгоритма кластеризации является актуальным, решает задачи оптимизации в графовых базах данных. Основной задачей является разбиение множества объектов сходной структуры на заранее неизвестные группы (кластеры), которые характеризуются похожими свойствами.

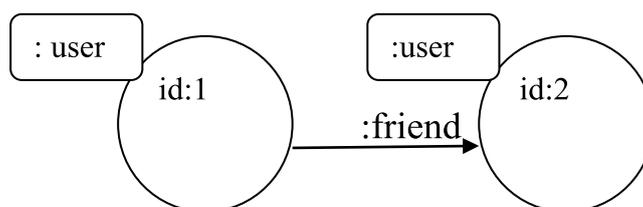


Рис.1

### Средства кластеризации графов в графовой СУБД Neo4j

В СУБД Neo4j имеется встроенная библиотека `Graph algorithms` для работы с графами. Для кластеризации графов данная библиотека реализуется с помощью таких алгоритмов как `Louvain` — он основан на оптимизации модулярности. Узлы объединяются в графы так, чтобы увеличить модулярность. `Louvain` самый быстрый алгоритм для работы с большими данными. Алгоритм `Label Propagation` кластеризует граф, используя только его структуру. Каждая вершина помещается только в тот кластер, которому принадлежит большая часть его соседей. Если таких кластеров несколько, то выбирается один случайным образом. Алгоритм `Triangle Counting` — треугольник представляет собой набор из трех узлов, в котором каждый узел имеет связи со всеми другими узлами. На основе полученных данных определяется коэффициент кластеризации. СУБД Neo4j предоставляет инструменты для работы с данными, которые можно использовать для методов кластеризации графов.

В разработанной системе для работы с Neo4j была использована библиотека `neo4j.v1` для языка Python. Данные в Neo4j хранятся в виде рисунка 1

Для тестирования алгоритмов кластеризации графов было сгенерировано случайный граф. Приведем пример запроса для выделения сообществ методом `Louvain`:

```
CALL algo.louvain('User','Friend',
{write: true, writeProperty:'community'})
```

`YIELD nodes, communityCount, iterations, loadMillis, computeMillis, writeMillis;`

После выполнения запроса в каждой вершине графа по теме: `user` появится свойство `community`, который будет содержать номер кластера, к которому метод `Louvain` отнесет соответствующую вершину.

Запрос к базе данных для выполнения алгоритма кластеризации `Label Propagation`. После выполнения данного запроса в каждой вершине графа по теме: `User` появится свойство `partition`, которая будет содержать номер кластера, к которому метод `Label Propagation` отнесет соответствующую вершину. Запрос к базе данных для выполнения алгоритма кластеризации `Triangle Counting`: После выполнения данного запроса к каждой вершине графа по теме: явится свойство `triangle`, который будет содержать номер треугольника, в котором находится вершина

### Вывод

Исследованы возможности графовой СУБД Neo4j для реализации методов кластеризации. Для выделения кластеров она предлагает несколько реализованных в ее библиотеке `Graph algorithms` алгоритм, а именно `Louvain`, `Label Propagation` и `Triangle Counting`. Другие алгоритмы кластеризации графов надо реализовывать самостоятельно, но Neo4j предоставляет много удобных инструментов для работы с данными, которые можно использовать для реализации методов кластеризации графов меньшими усилиями, чем без использования Neo4j.

### ЛИТЕРАТУРА:

1. Гуральник Р. И. Некоторые задачи на графовых базах данных Труды ИСП РАН, том 28, вып. 4, 2016, стр. 193–216. DOI: 10.15514/ISPRAS-2016–28(4)-12
2. Jeevan Joishi, Ashlish Sureka, "Graph or Relational Databases: A Speed Comparison for Process Mining Algorithm", CoRRabs/1701.00072 (2017)
3. Laurel Orr and Jennifer Ortiz, "Clustering with the DBLP Bibliography to Measure External Impact of a Computer Science Research Area", доступен по ссылке <http://homes.cs.washington.edu/~jortiz16/images/MLProjectPaper.pdf>
4. Neo4J Graph Database. URL: <http://www.neo4j.org>