

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления  
Кафедра интеллектуальных информационных технологий

ОТЧЁТ  
по лабораторной работе №3  
по дисциплине

## **ПРОЕКТИРОВАНИЕ БАЗ ЗНАНИЙ**

Предметная область: теория множеств

Студент гр. 121701  
Руководитель

Р. В. Липский  
Н. Г. Липницкая

Минск 2023

# СОДЕРЖАНИЕ

1	Постановка задачи . . . . .	2
2	Запросы к базе данных . . . . .	3
2.1	Получить все узлы и отношения в базе данных . . . . .	3
2.2	Получить все существующие виды множеств . . . . .	4
2.3	Получить все разбиения множества . . . . .	5
2.4	Получить имена всех типов множеств . . . . .	6
2.5	Посчитать количество подвидов множеств . . . . .	7
2.6	Получить описания всех видов множеств в базе данных . . . . .	8
2.7	Получить все виды множеств без описания . . . . .	9
2.8	Получить все декомпозиции множеств, имеющих собственные декомпозиции . . . . .	10
2.9	Получить все узлы, не связанные отношением . . . . .	11
2.10	Посчитать все IS отношения в базе . . . . .	12
	Список использованных источников . . . . .	12

# 1 ПОСТАНОВКА ЗАДАЧИ

**Тема:** Изучение средств создания базы данных и выполнения запросов к ней с использованием графовой СУБД neo4j.

**Цель:** Получить навыки создания базы данных и выполнения запросов к ней с помощью средств графовой СУБД neo4j.

**Задача:**

- Сформировать базу данных по выбранной предметной области.
- Составить список из 10 запросов к базе данных.
- С помощью шаблонов запросов получить выборку для каждого запроса из п.2).
- В отчёте отразить в графической или текстовой форме содержимое базы данных, шаблоны запросов и полученные выборки с комментариями.

## 2 ЗАПРОСЫ К БАЗЕ ДАННЫХ

### 2.1 Получить все узлы и отношения в базе данных

Текст запроса:

```
match (n)
optional match (n)-[r]->()
return n, r
```

Получим все узлы n, если для узла n существует отношение r, получим и его.

Результат:

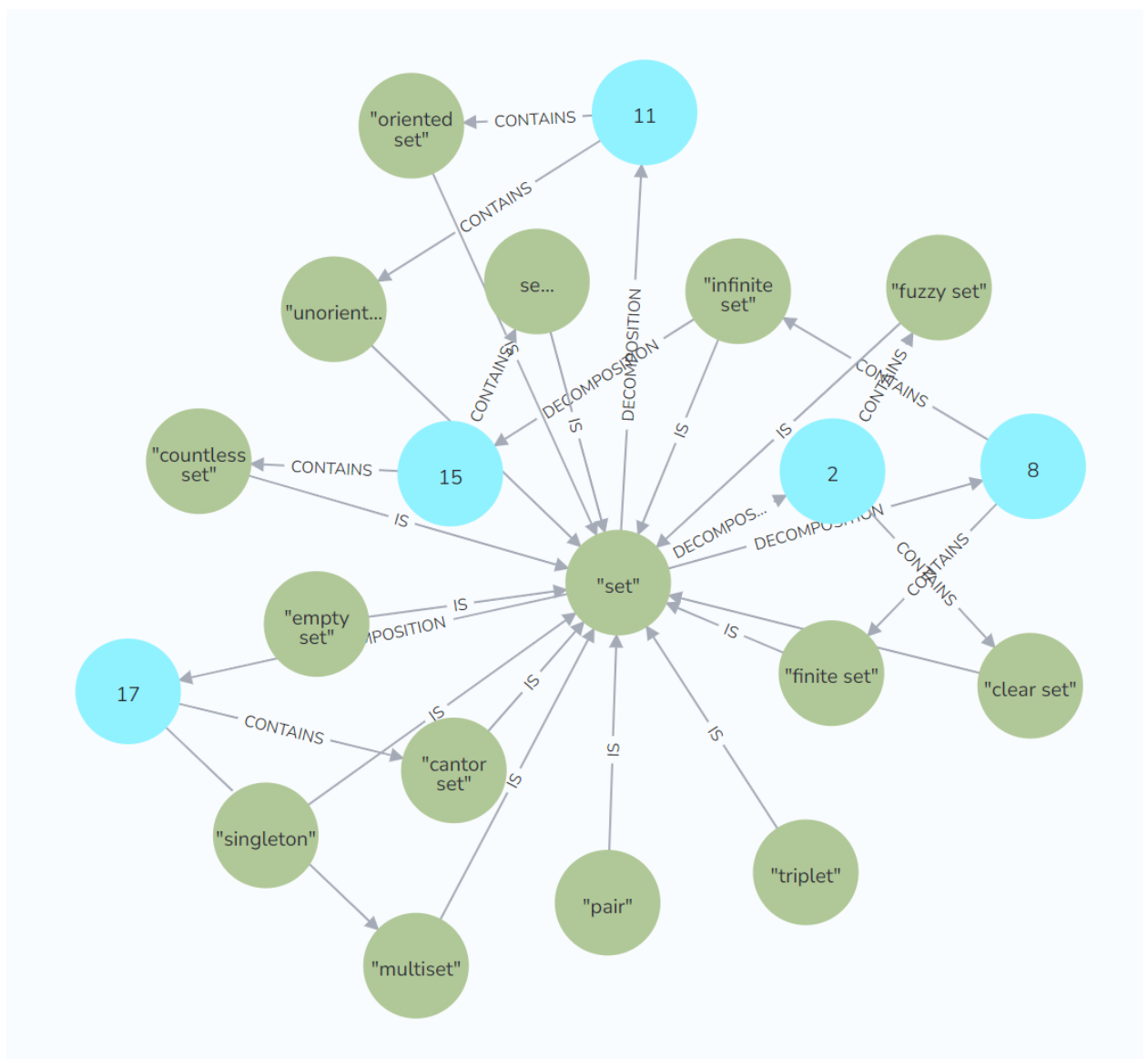


Рисунок 2.1 – Получить все узлы и отношения в базе данных

## 2.2 Получить все существующие виды множеств

Текст запроса:

```
match (n: Set {en_name: 'set'})<-[is:IS]-(r)
return n, is, r
```

Получим все узлы  $r$ , которые связаны отношением  $IS$  с узлом  $n$  с английским именем "set".

Результат:

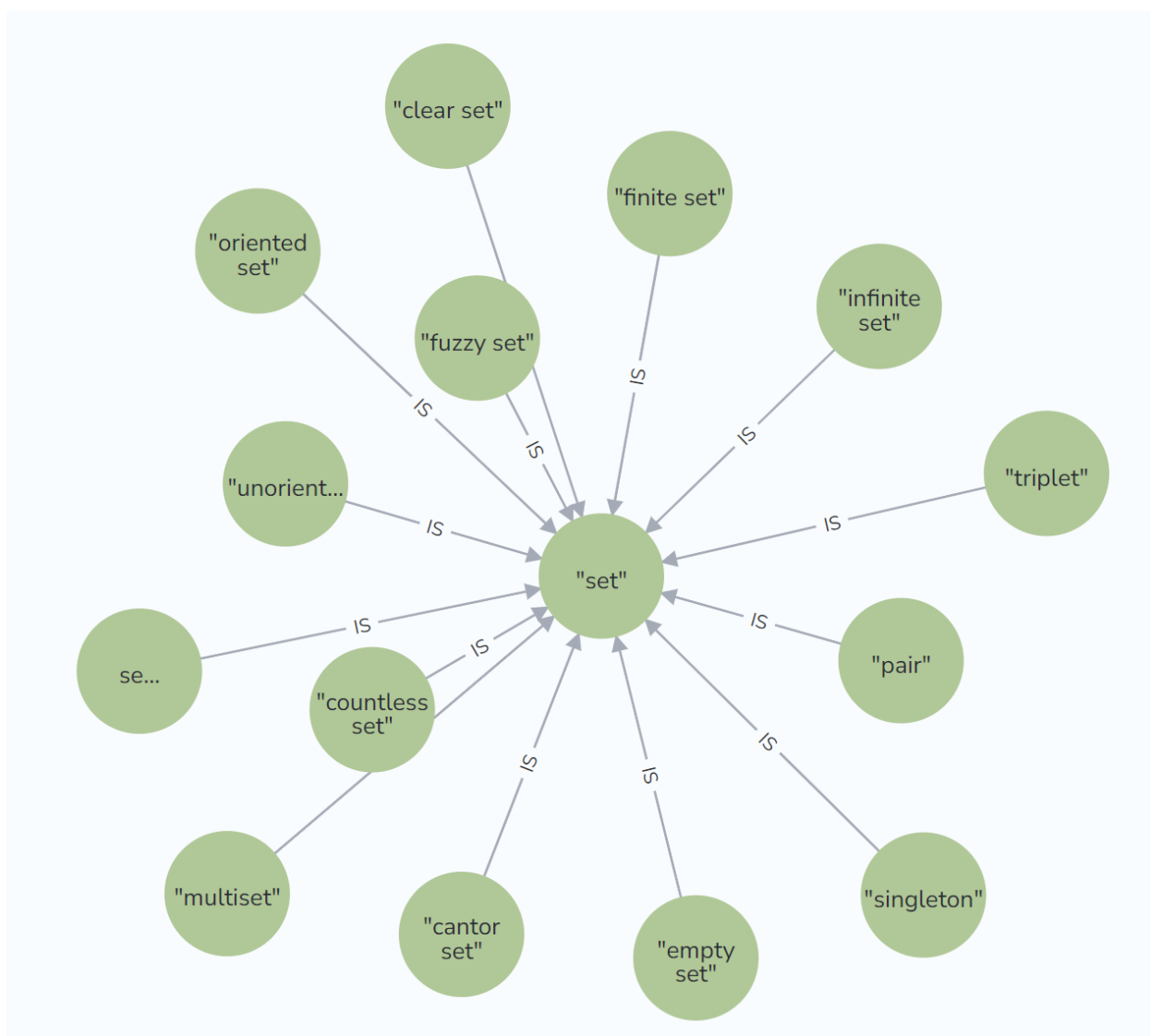


Рисунок 2.2 – Получить все существующие виды множеств

## 2.3 Получить все разбиения множества

Текст запроса:

```
match (n: Set {en_name: 'set'})-[d:DECOMPOSITION]
      ->(dec:Decomposition)-[c:CONTAINS]->(set)
return n, d, dec, c, set
```

Получить все узлы set, которые связаны отношением CONTAINS с узлами dec, которые связаны отношением DECOMPOSITION с узлом n с английским именем "set".

Результат:

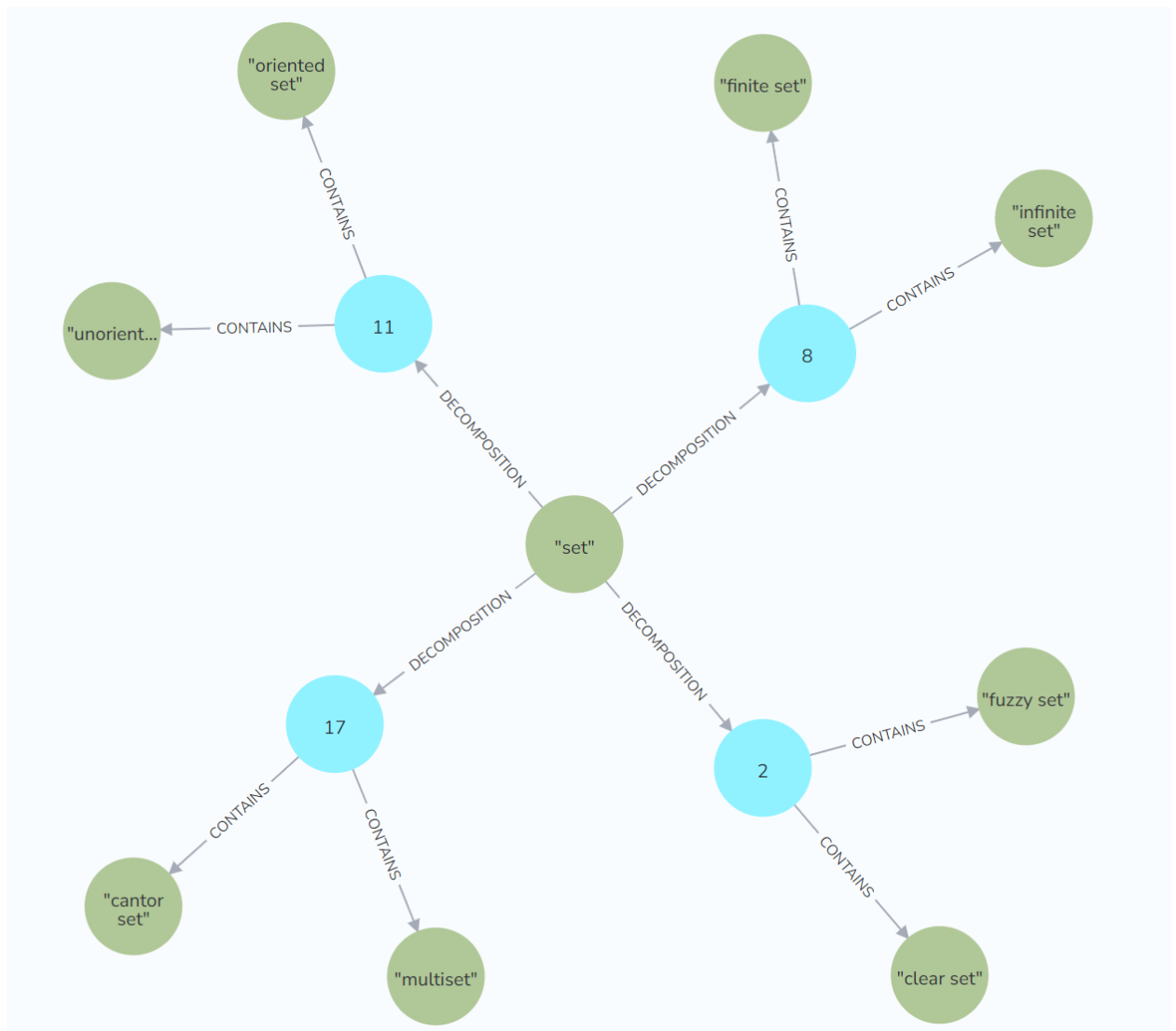


Рисунок 2.3 – Получить все разбиения множества

## 2.4 Получить имена всех типов множеств

Текст запроса:

```
MATCH (set: Set {en_name: 'set'})<-[:IS]-(subset)
RETURN subset.en_name
```

Получить атрибут en\_name для всех узлов subset, связанных отношением IS с узлом с английским именем «set».

Результат:

subset.en\_name

"empty set"

"singleton"

"pair"

"triplet"

"infinite set"

"finite set"

"clear set"

"fuzzy set"

"oriented set"

"unoriented set"

"countable set"

"countless set"

"multiset"

"cantor set"

Рисунок 2.4 – Получить имена всех типов множеств

## 2.5 Посчитать количество подвидов множеств

Текст запроса:

```
MATCH (set:Set {en_name: 'set'})<-[:IS]-(subset:Set)
RETURN count(subset) AS subsetCount
```

Посчитать количество узлов subset, связанных отношением IS с узлом с английским именем «set».

Результат:

subsetCount
14

Рисунок 2.5 – Посчитать количество подвидов множеств



## 2.6 Получить описания всех видов множеств в базе данных

Текст запроса:

```
MATCH (set: Set {en_name: 'set'})<-[:IS]-(subset)
RETURN subset.en_name, subset.description
```

Получить атрибуты en\_name и description для узлов subset, связанных отношением IS с узлом, имеющим английское имя «set».

Результат:

subset.en_na...	subset.description
"empty set"	"пустое множество – это множество, которому не принадлежит ни один элемент."
"singleton"	"синглетон – это множество, состоящее из одного элемента."
"pair"	"пара – это множество, состоящее из двух элементов."
"triplet"	"тройка – это множество, состоящее из трех элементов."
"infinite set"	"бесконечное множество – это множество, в котором для любого натурального числа n найдётся конечное подмножество из элементов."
"finite set"	"конечное множество – это множество, количество элементов которого конечно, то есть, существует неотрицательное целое число k, равное количеству элементов."
"clear set"	"четкое множество – это множество, принадлежность элементов которому достоверно."
"fuzzy set"	"нечеткое множество – это множество, которое представляет собой совокупность элементов произвольной природы, относительно которых нельзя точно сказать, принадлежат ли они множеству."
"oriented set"	"ориентированное множество – это множество, представляющее собой упорядоченный набор элементов, т.е. такое множество, порядок элементов в котором имеет значение."
"unoriented set"	null
"countable set"	"счетное множество – это бесконечное множество, для которого существует взаимно-однозначное соответствие с натуральным рядом чисел."

Рисунок 2.6 – Получить описания всех видов множеств в базе данных

## 2.7 Получить все виды множеств без описания

Текст запроса:

```
MATCH (set: Set {en_name: 'set'})<-[:IS]-(subset)
WHERE subset.description is null
RETURN subset.en_name
```

Получить атрибут `en_name` для узлов `subset`, связанных отношением `IS` с узлом, имеющим английское имя «set», при этом атрибут `description` отсутствует.

Результат:

---

subset.en_name
----------------

"unoriented set"
------------------

---

Рисунок 2.7 – Получить все виды множеств без описания

## 2.8 Получить все декомпозиции множеств, имеющих собственные декомпозиции

```
MATCH (set: Set {en_name: 'set'})
  -[decomposition_rel:DECOMPOSITION]->(decomposition)
  -[contains1:CONTAINS]->(decomposed1)
  -[decomposition_rel_2:DECOMPOSITION]->(decomposition2)
  -[contains2:CONTAINS]->(decomposed2)

return set,
       decomposition_rel,
       decomposition,
       contains1,
       decomposed1,
       decomposition_rel_2,
       decomposition2,
       contains2,
       decomposed2
```

Результат:

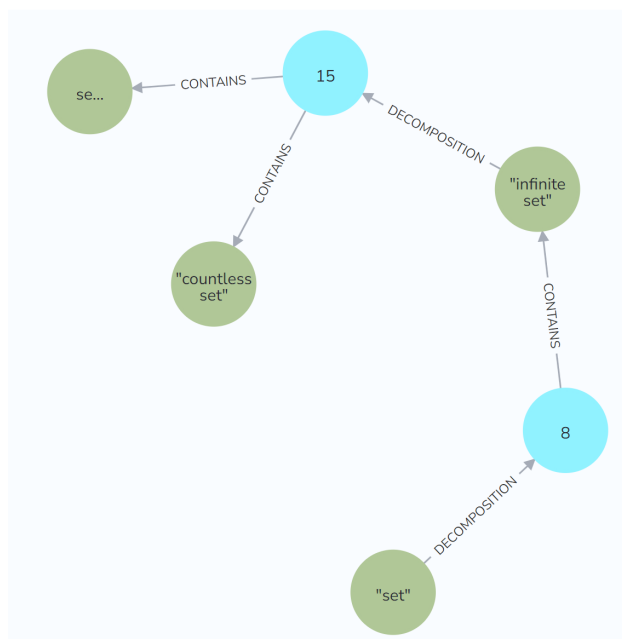


Рисунок 2.8 – Получить все декомпозиции множеств, имеющих собственные декомпозиции

## 2.9 Получить все узлы, не связанные отношением

Текст запроса:

```
MATCH (n)
WHERE NOT EXISTS((n)-[:IS]->(:Set {en_name: 'set'}))
RETURN n
```

Получить все узлы n, не связанные отношением IS с узлом с английским именем «set».

Результат:

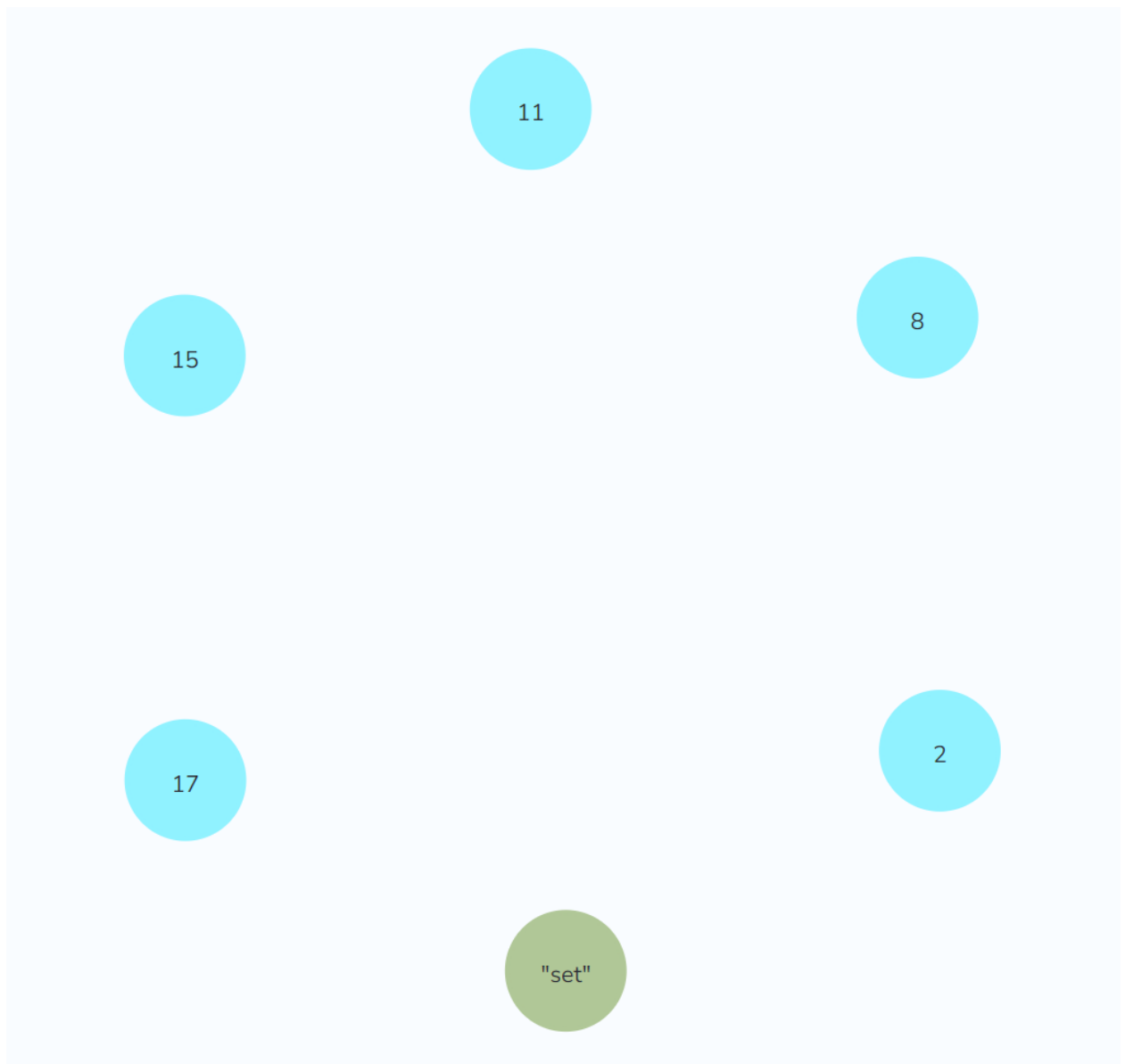


Рисунок 2.9 – Получить все узлы, не связанные отношением

## 2.10 Посчитать все IS отношения в базе

Текст запроса:

```
MATCH ()-[:IS]->()  
RETURN count(*) AS relationshipCount
```

Посчитать все отношения IS.

Результат:

relationshipCount

---

14

---

Рисунок 2.10 – Посчитать все IS отношения в базе