

Санкт-Петербургский колледж информационных технологий

ОТЧЕТ

по работе "Создание базы данных в Neo4j"

по МДК07.01 "Управление и автоматизация баз данных"

Специальность 09.02.07

"Информационные системы и программирование"

Специализация:

"Администратор баз данных"

Выполнил:

студент 4-го курса группы 404 Колинко Роман

Преподаватель:

Фомин Александр Валерьевич

Санкт-Петербург, 2024 год

ВВЕДЕНИЕ

Целью работы является создание базы данных в Neo4j.

Работа выполнена в ОС Linux в дистрибутиве Ubuntu 20.04.6 LTS.

ВЫПОЛНЕНИЕ

Была установлена система управления базами данных Neo4j и настроена на возможность подключения к ней с разных хостов (рисунок 1-2)



Рисунок 1 – Подключение к базе данных Neo4j на локальном хосту

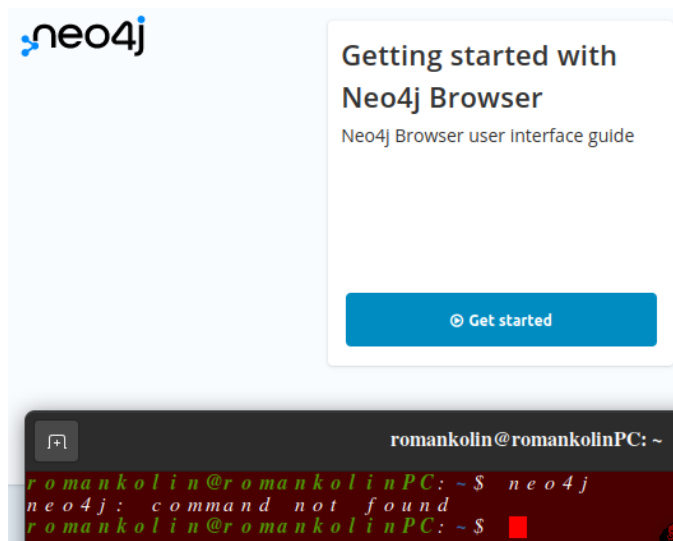


Рисунок 2 – Подключение к базе данных Neo4j на удаленном хосту

Были разработаны граф базы данных пациентов (рисунок 3-5). Также было установлено, что при реализации реляционной базы данных потребовалось бы создать либо одну таблицу – Patient (id smallint primary key, status varchar(10) check in('Cured', 'Infected', 'Died'), infectday smallint, curday smallint null, deathday

smallint null), либо две таблицы – Patient (id smallint primary key, status varchar(10) check in('Cured', 'Infected', 'Died')), Illness (id smallint primary key, pat smallint, infectday smallint, curday smallint null, deathday smallint null, foreign key patient references patient(id)) со связью один ко многим, что надежнее создания одной таблицы, так как пациент может заболеть более одного раза.

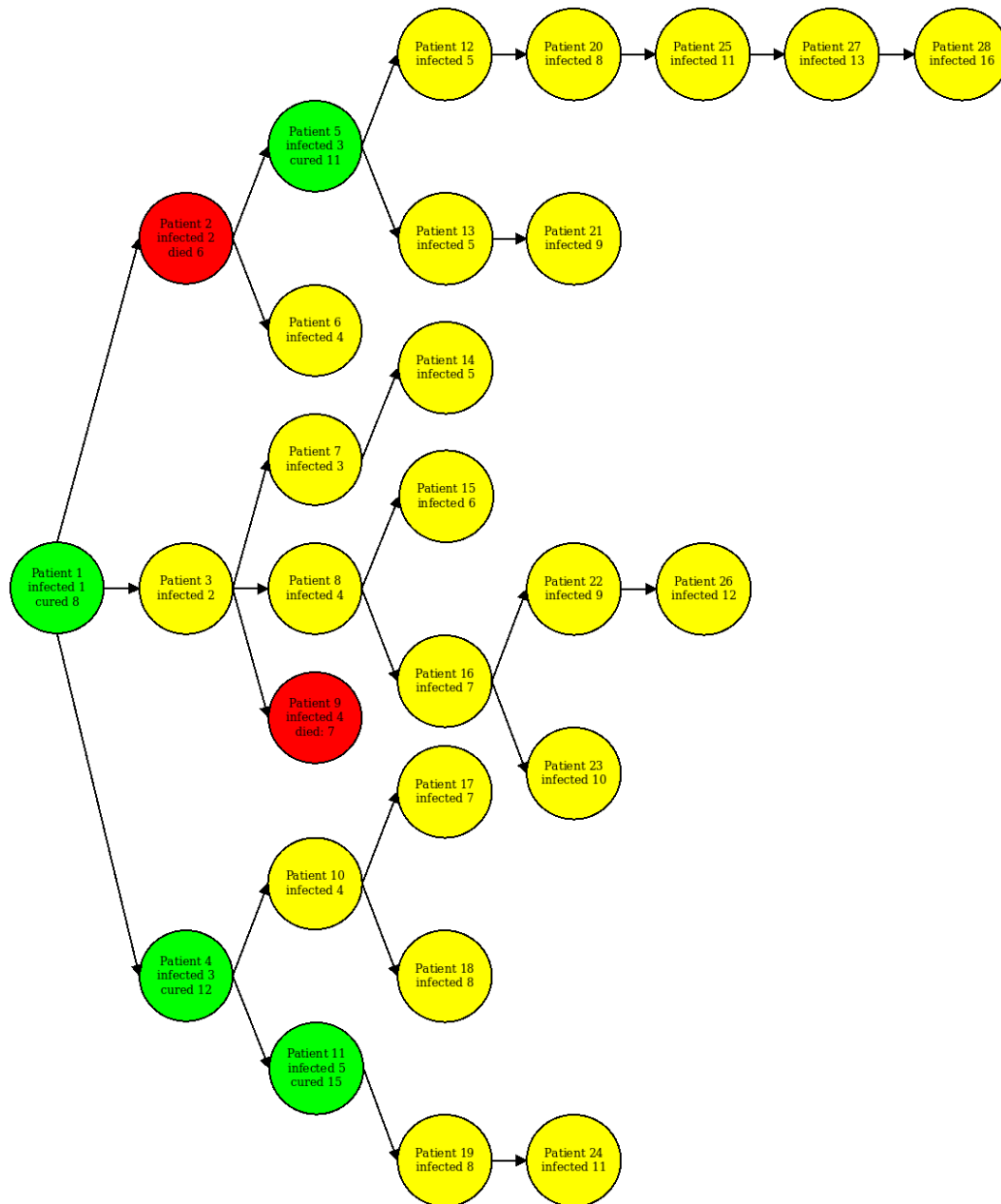


Рисунок 3 – Граф базы данных

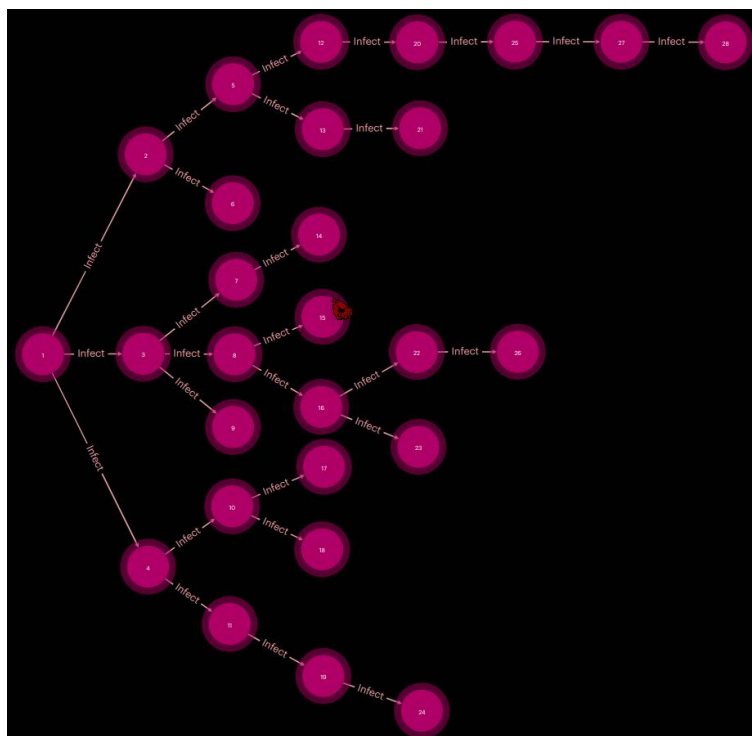


Рисунок 4 – Граф базы данных в RedisInsight

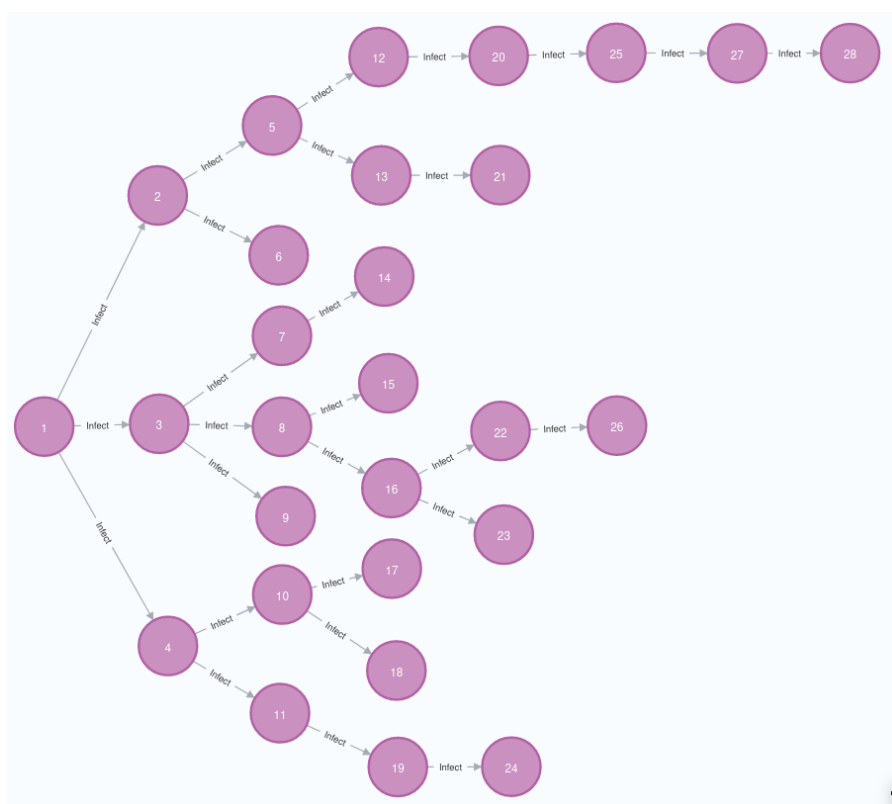


Рисунок 5 – Граф базы данных в Neo4j

Была создана база данных в Redis и Neo4j путем написания скрипта на языке Cypher (рисунок 6-7).

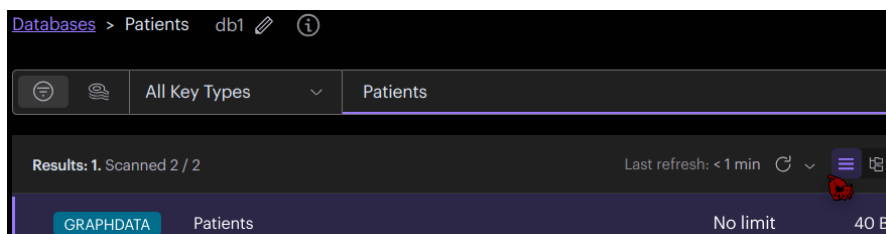


Рисунок 6 – База данных в Redis

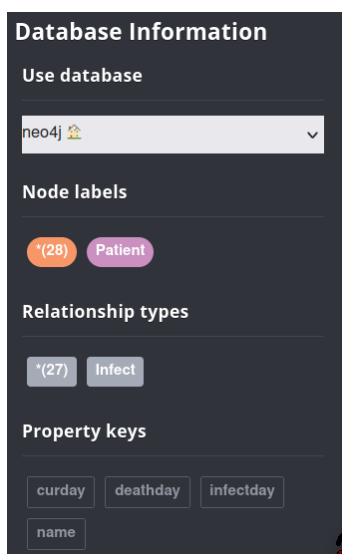


Рисунок 7 – База данных в Neo4j

Далее были написаны запросы к базам данных в Redis и Neo4j на выборку заразившихся в первые три дня пациентов, излечившихся пациентов, умерших пациентов, зараженных пациентов, количества пациентов по дням, зараженных напрямую пятым пациентом пациентов, количества пациентов зараженных восьмым пациентом, заразивших двадцать восьмого пациента через первого пациентов, репродуктивное число пациента, среднее репродуктивное число и на изменение статуса седьмого пациента, удаление двадцать шестого пациента; после чего сделан дамп базы данных в Redis (рисунок 8-20).

neo4j\$ MATCH(infected:Patient) WHERE infected.infectday ≤ 3 RETURN infected.name AS Name

	Name
1	"1"
2	"2"
3	"3"
4	"4"
5	"5"
6	"7"

Started streaming 6 records after 9 ms and completed after 11 ms.

Patients [db1] - Workbench

Window View Help

1 Command(s) - 1 success, 0 error(s) Feb 15, 08:17:35 7,084 msec

```
[db1] > GRAPH.QUERY Patients "MATCH(infected:Patient) WHERE infected.infectday <= 3 RETURN infected.name AS Name"
1) 1) "Name"
2) 1) 1) "1"
   2) 1) "2"
   3) 1) "3"
   4) 1) "4"
   5) 1) "5"
   6) 1) "7"
3) 1) "Cached execution: 0"
   2) "Query internal execution time: 4.101568 milliseconds"
```

Рисунок 8 – Заразившиеся в первые три дня пациенты

neo4j\$ MATCH(cured:Patient) WHERE cured.curday IS NOT null RETURN cured.name AS Name

	Name
1	"1"
2	"4"
3	"5"
4	"11"

Started streaming 4 records after 6 ms and completed after 8 ms.

Patients [db1] - Workbench

Window View Help

1 Command(s) - 1 success, 0 error(s) Feb 15, 08:19:50 1,959 msec

```
[db1] > GRAPH.QUERY Patients "MATCH(cured:Patient) WHERE cured.curday IS NOT null RETURN cured.name AS Name"
1) 1) "Name"
2) 1) 1) "1"
   2) 1) "4"
   3) 1) "5"
   4) 1) "11"
3) 1) "Cached execution: 0"
   2) "Query internal execution time: 0.436427 milliseconds"
```

Рисунок 9 – Излечившиеся пациенты

neo4j\$ MATCH(died:Patient) WHERE died.deathday IS NOT null RETURN died.name AS Name

	Name
1	"2"
2	"9"

Patients [db1] - Workbench

Window View Help

1 Command(s) - 1 success, 0 error(s) Feb 15, 08:20:47 1,711 msec

```
[db1] > GRAPH.QUERY Patients "MATCH(died:Patient) WHERE died.deathday IS NOT null RETURN died.name AS Name"
1) 1) "Name"
2) 1) 1) "2"
   2) 1) "9"
3) 1) "Cached execution: 0"
   2) "Query internal execution time: 0.688200 milliseconds"
```

Рисунок 10 – Умершие пациенты

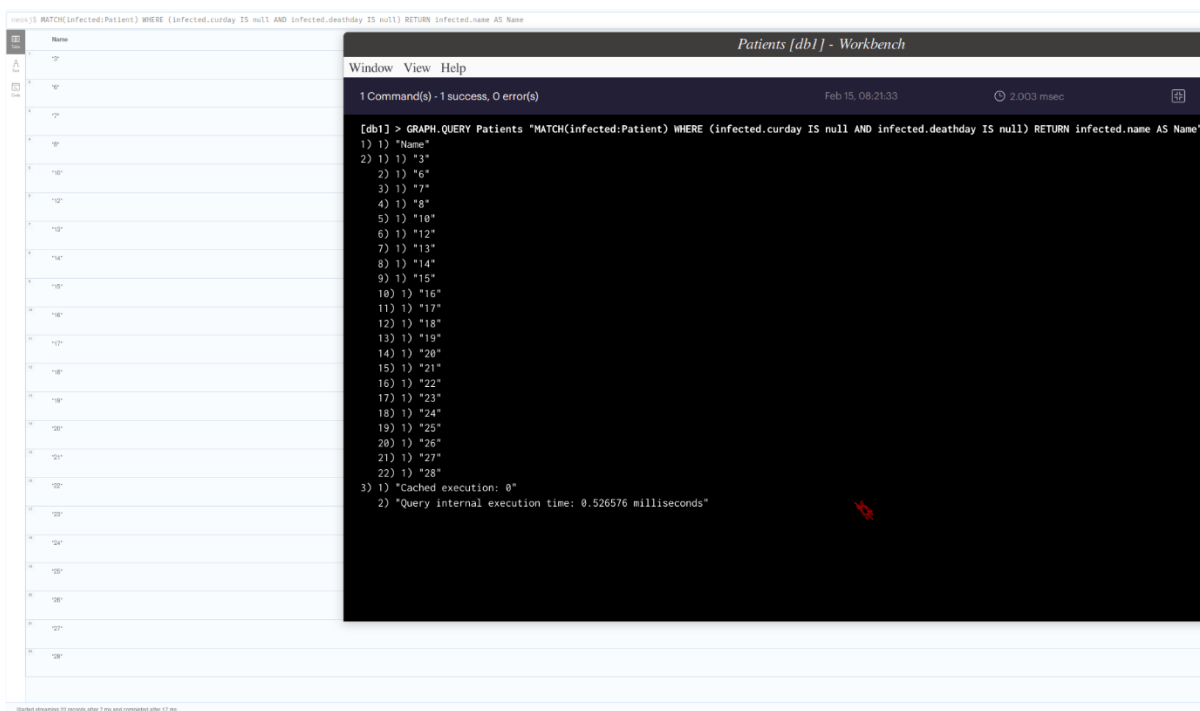


Рисунок 11 – Зараженные пациенты

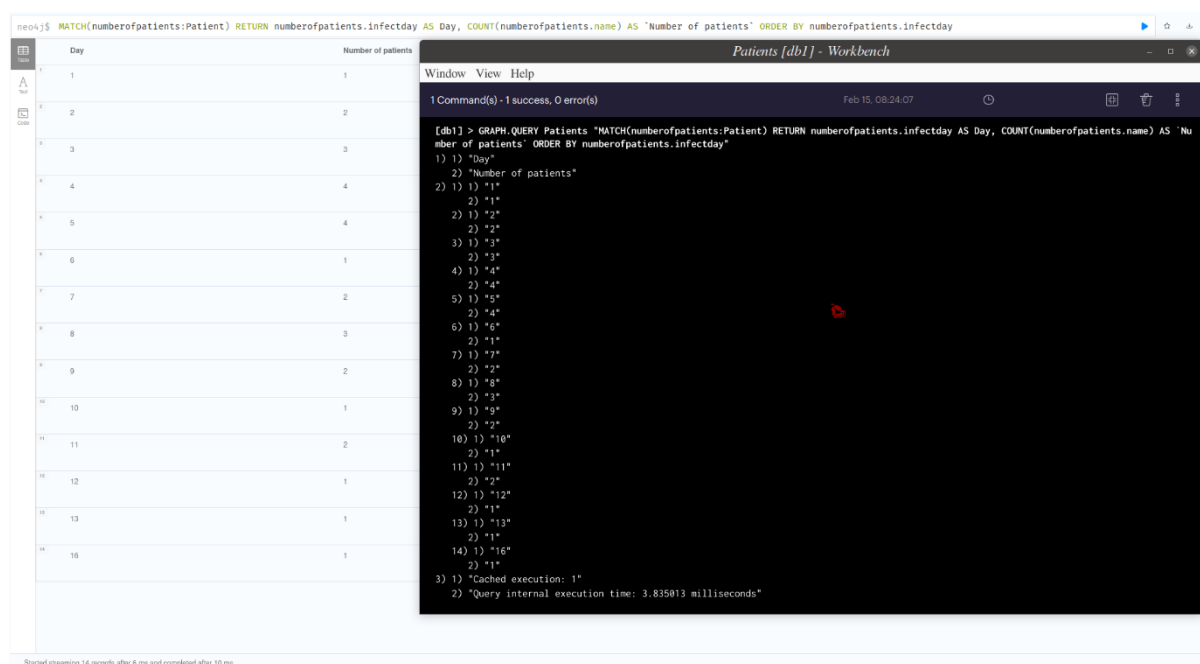


Рисунок 12 – Количество пациентов по дням

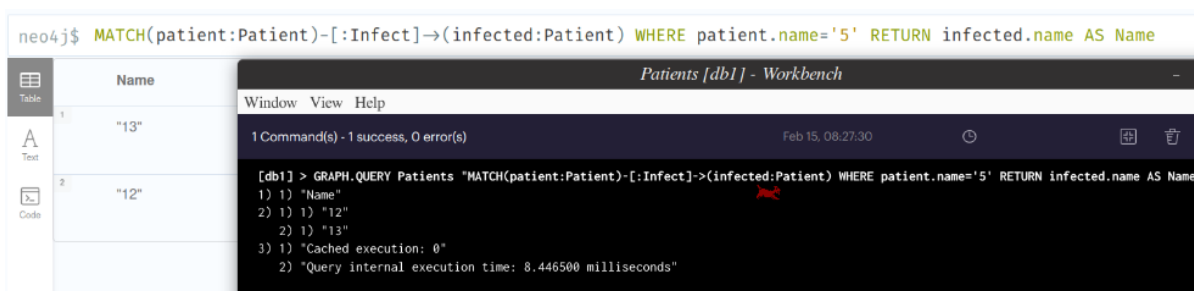


Рисунок 13 – Зараженные напрямую пятым пациентом пациенты



Рисунок 14 – Количество зараженных пациентов восьмым пациентом

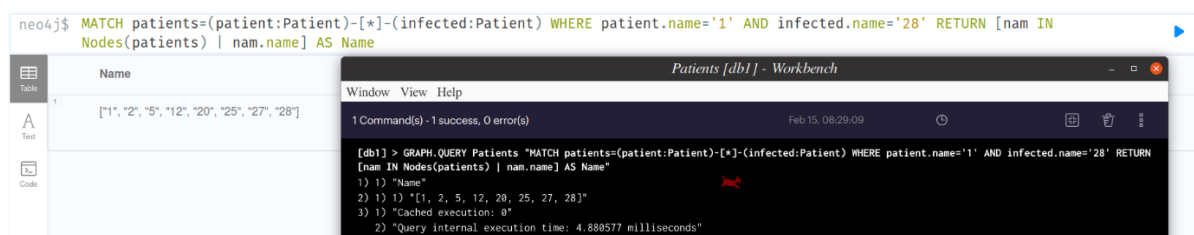


Рисунок 15 – Заразившие двадцать восьмого пациента через первого пациенты

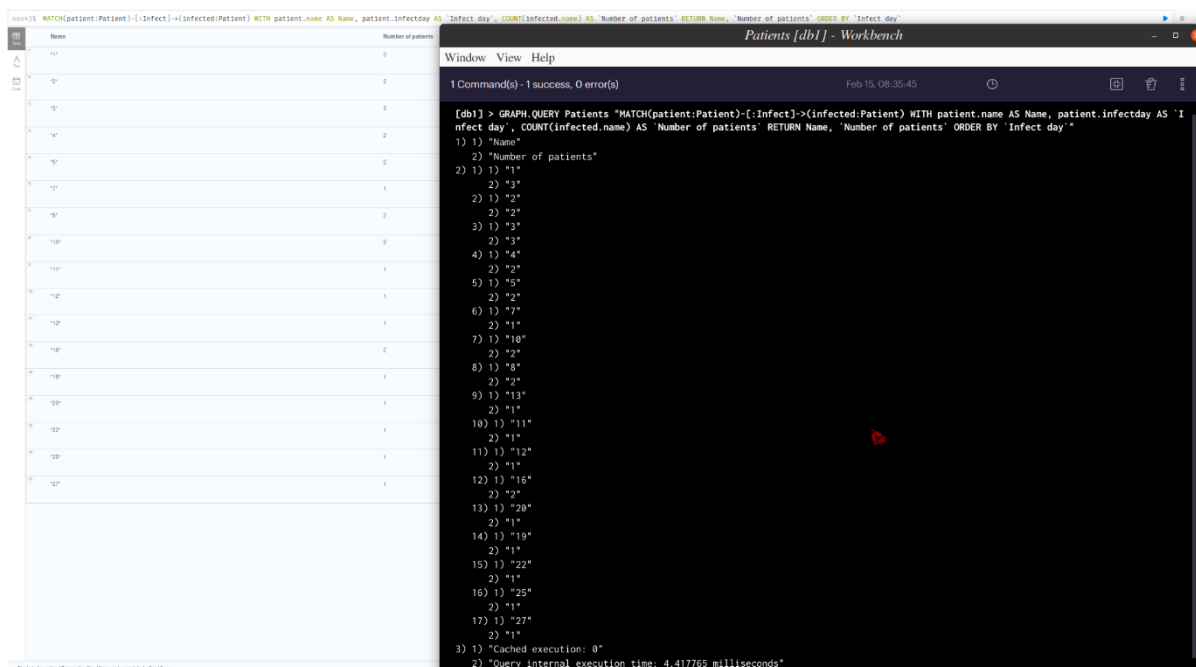


Рисунок 16 – Репродуктивное число пациента

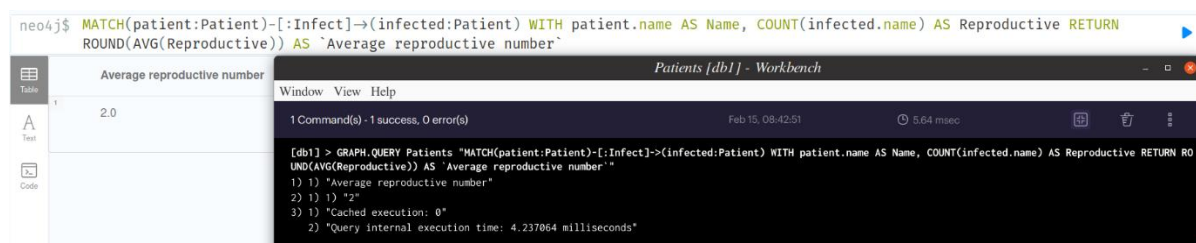


Рисунок 17 – Среднее репродуктивное число

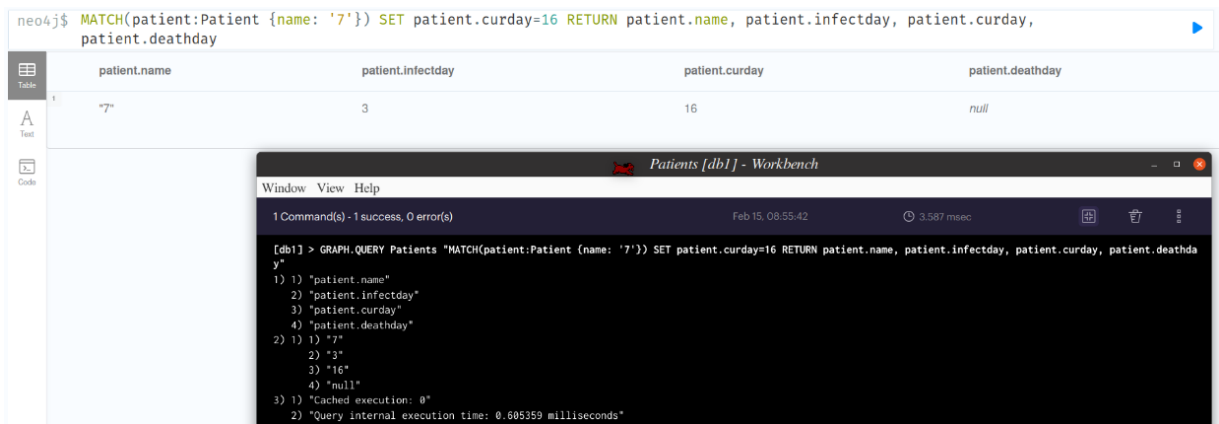


Рисунок 18 – Изменение статуса седьмого пациента

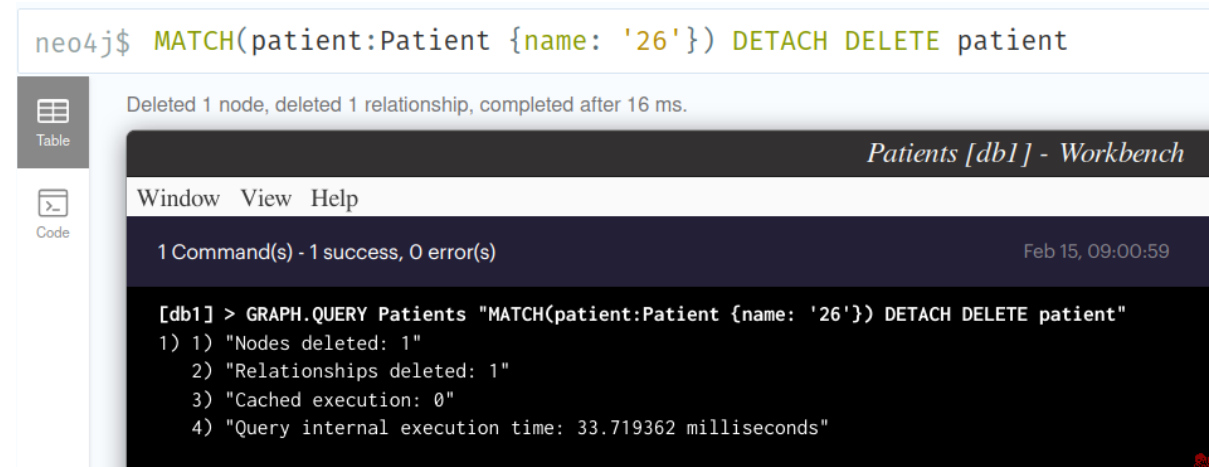


Рисунок 19 – Удаление двадцать шестого пациента



Рисунок 20 – Дамп базы данных в Redis

Также был просмотрен план выполнения запроса на выборку зараженных пациентов десятым пациентом, создан индекс на имя пациента и просмотрен план выполнения запроса на выборку зараженных пациентов десятым пациентом с учетом индекса (рисунок 21-22).

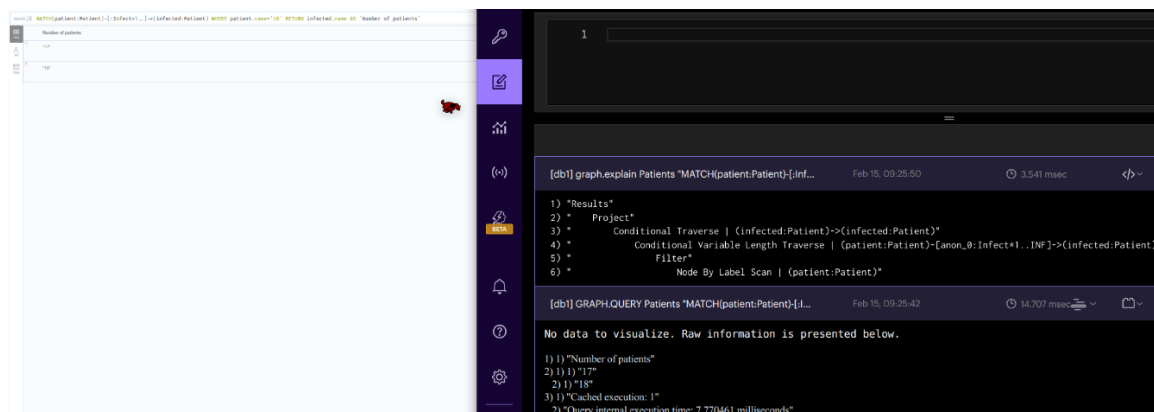


Рисунок 21 – Запрос и его план на выборку зараженных пациентов десятым пациентом

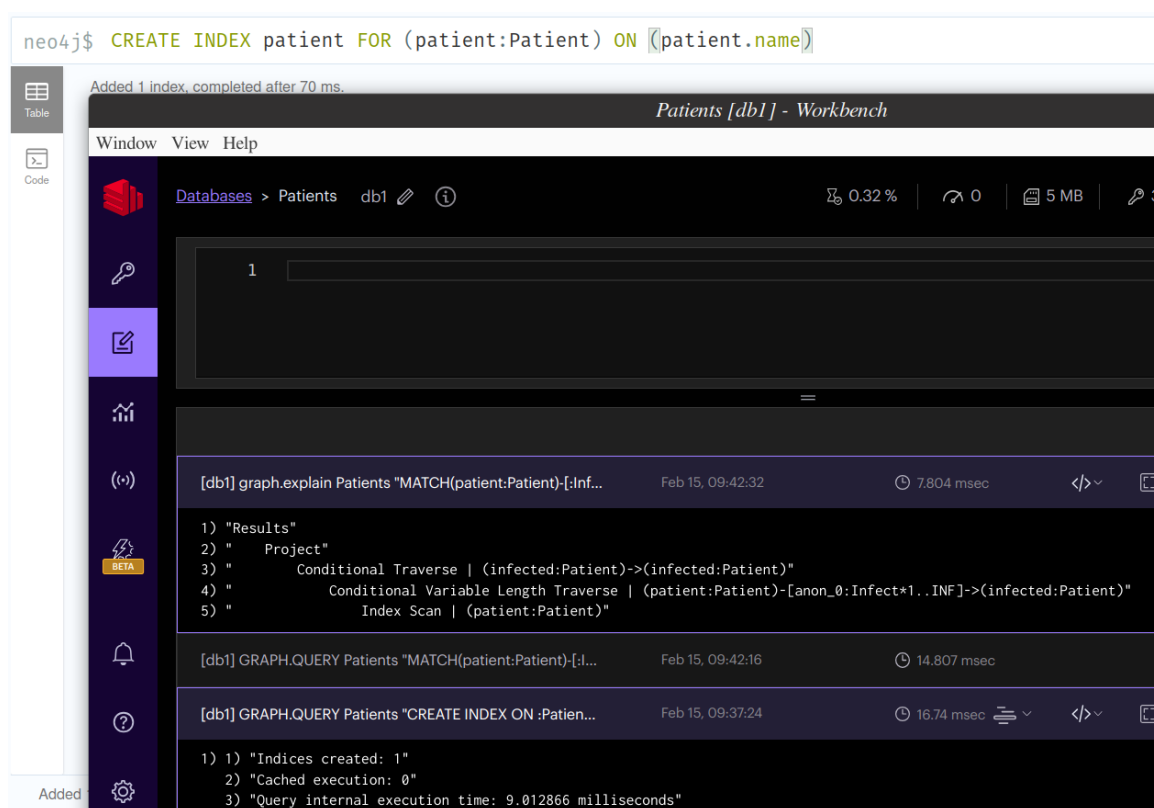


Рисунок 22 – Индекс и план запроса на выборку зараженных пациентов десятым пациентом