**Частина 1. Робота зі структурами даних у Cassandra**

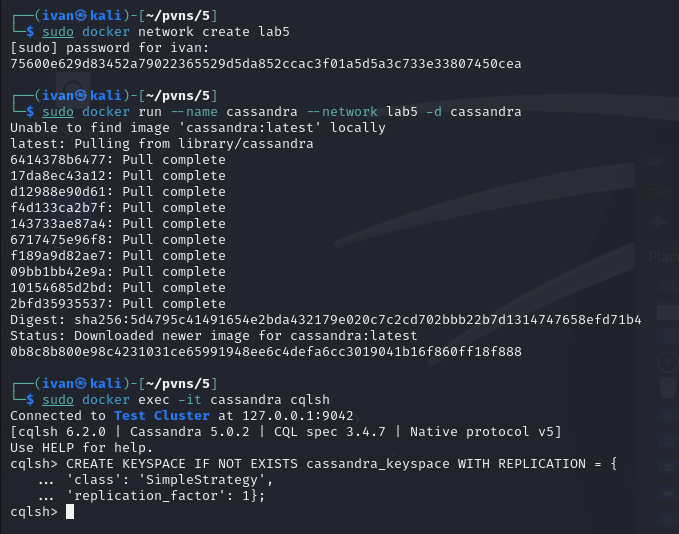
**Завдання:**

Ознайомтеся з особливістю моделювання даних у Cassandra:

* <https://cassandra.apache.org/doc/latest/cassandra/data_modeling/index.html>
* <https://www.instaclustr.com/blog/cassandra-data-modeling/>

Створіть keyspace з найпростішої стратегією реплікації

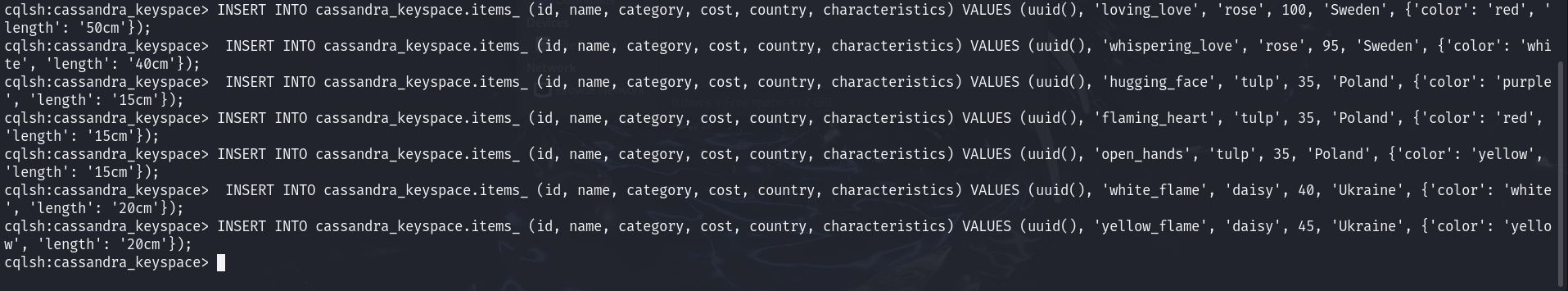
<https://docs.datastax.com/en/cql/cassandra-5.0/develop/keyspace-create.html>



В цьому keyspace необхідно буде створити дві таблиці: *items* та *orders*



У таблиці *items* містить різноманітні товари (тобто у яких різний набір властивостей).



Для набору властивостей товару виберіть базові характеристики однакові для всіх товарів (назва, категорія, ціна, виробник, ...), а для властивостей які відрізняються використовуйте тип *map* (з індексом для можливості пошуку по її вмісту)

<https://docs.datastax.com/en/cql/cassandra-5.0/develop/collections-create.html#map-collections>

<https://cassandra.apache.org/doc/4.1/cassandra/cql/types.html#maps>

<https://cassandra.apache.org/doc/4.1/cassandra/cql/indexes.html#indexes-on-map-keys>

**Необхідно, щоб пошук швидко працював для *категорії* товарів. Ця вимога має бути врахована при створенні ключа для таблиці.**

**!!! У запитах заборонено використовувати *ALLOW FILTERING* !!!**

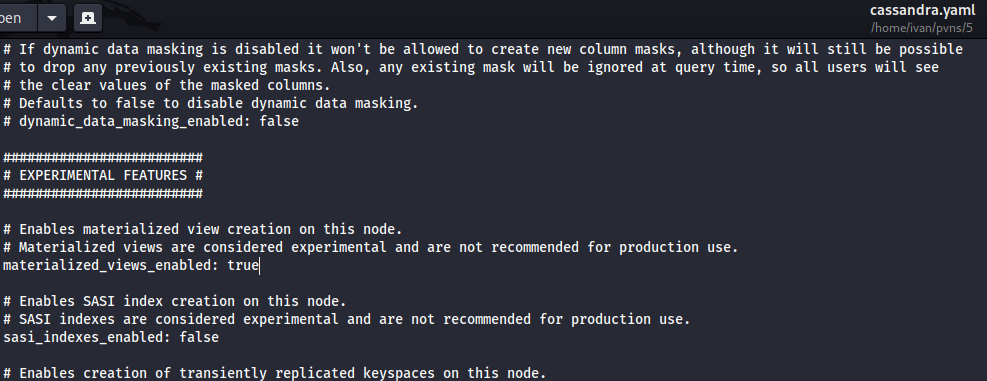
1. Напишіть запит, який показує структуру створеної таблиці (команда *DESCRIBE*)

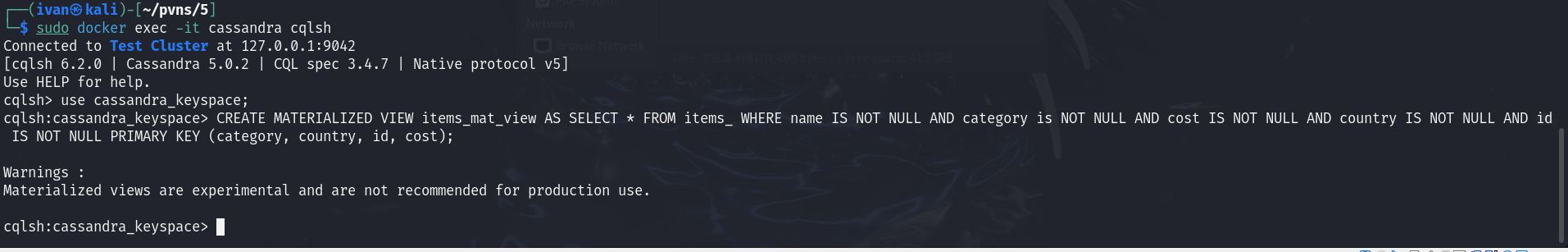


1. Напишіть запит, який виводить усі товари в певній категорії відсортовані за ціною

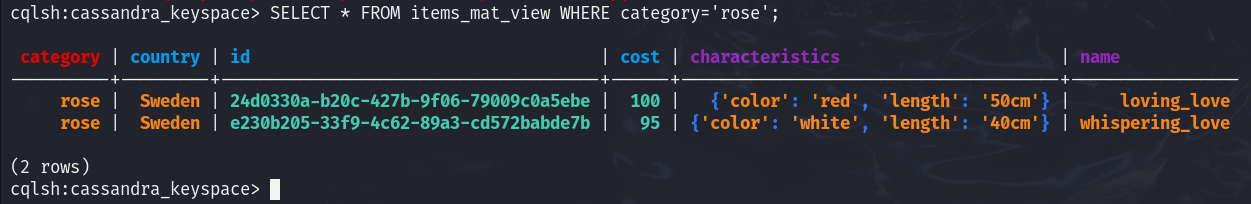


1. Напишіть запити, які вибирають товари за різними критеріями в межах певної категорії (тут де треба замість індексу використайте Matirialized view):

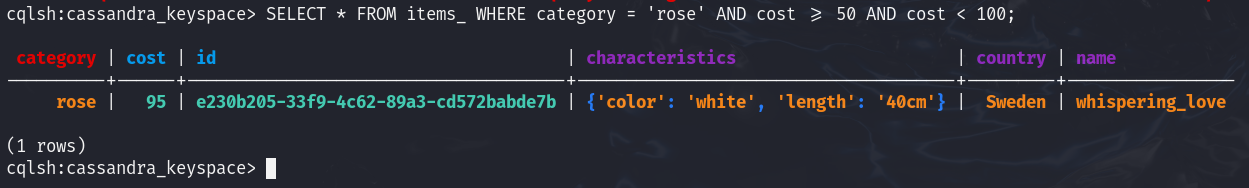




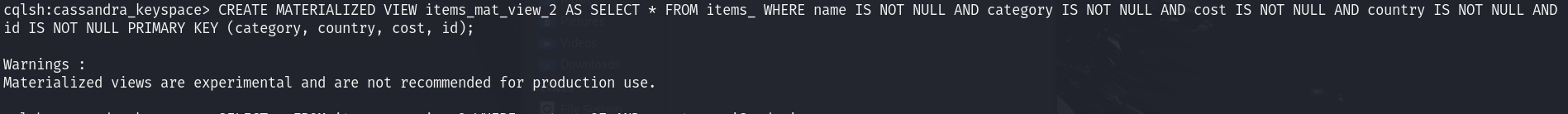
* + назва,

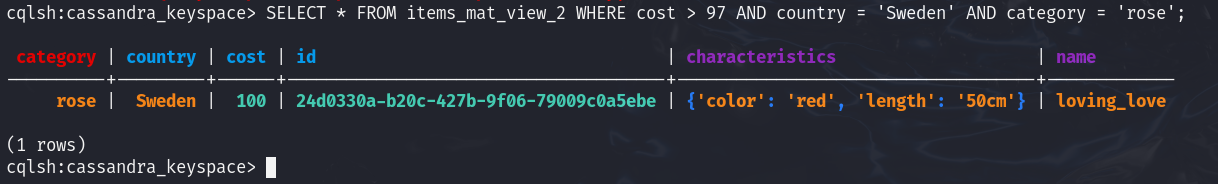


* + ціна (в проміжку),

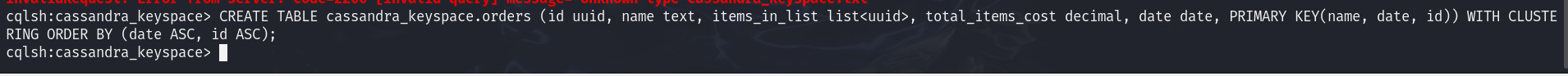


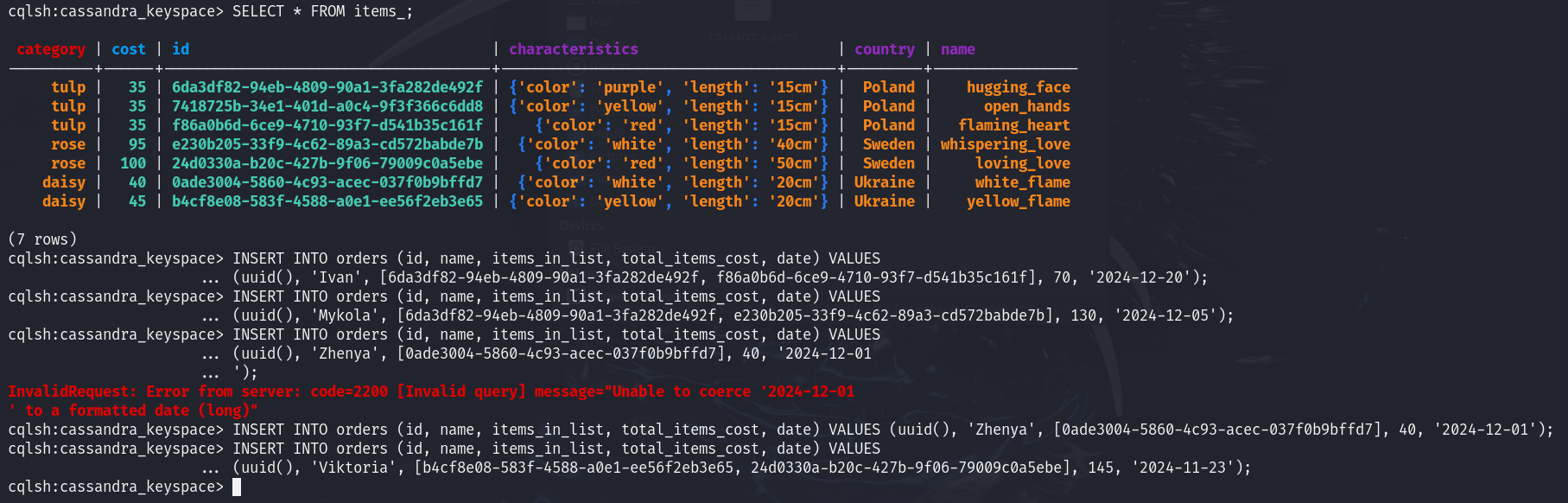
* + ціна та виробник





Створіть таблицю *orders* в якій міститься ім'я замовника і інформація про замовлення: перелік id-товарів у замовленні, вартість замовлення, дата замовлення, ....





**Для кожного замовника повинна бути можливість швидко шукати його замовлення і виконувати по них запити. Ця вимога має бути врахована при створенні ключа для таблиці.**

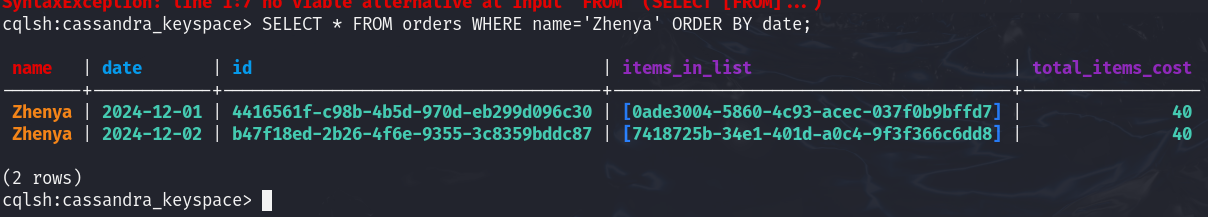
1. Напишіть запит, який показує структуру створеної таблиці (команда *DESCRIBE*)



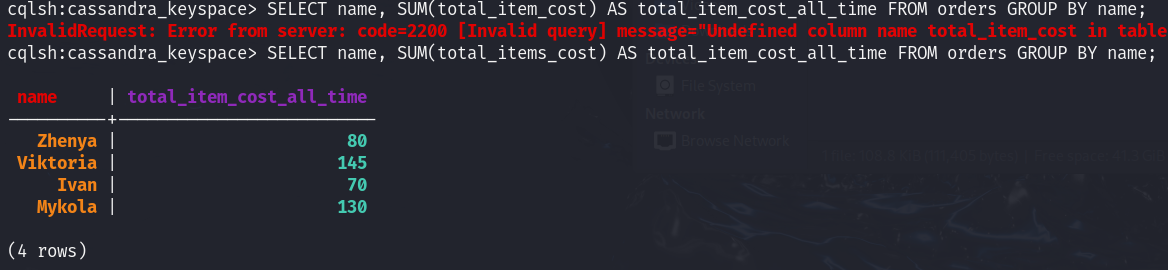
1. Для замовника виведіть всі його замовлення відсортовані за часом коли вони були зроблені

Спочатку додам ще один запис, бо в мене унікальні клієнти були:

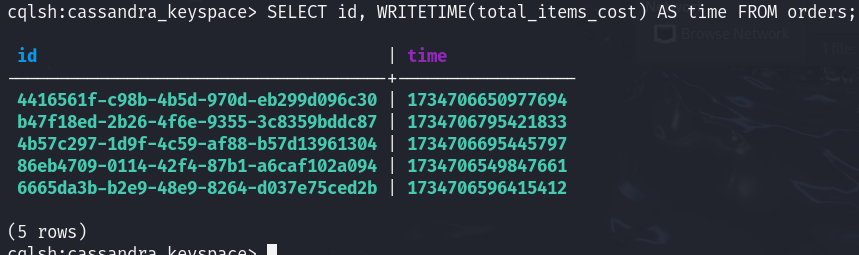




1. Для кожного замовників визначте суму на яку були зроблені усі його замовлення



1. Для кожного замовлення виведіть час коли його ціна були занесена в базу (SELECT WRITETIME)



**Частина 2. Налаштування реплікації у Cassandra**

**Завдання**

1. Сконфігурувати кластер з 3-х нод:

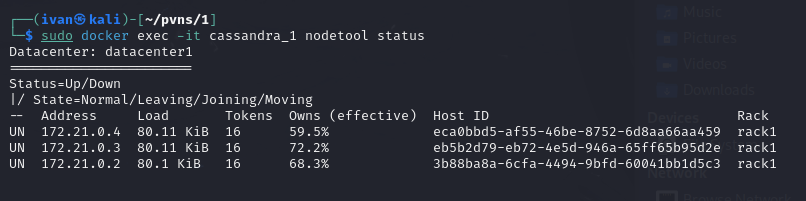
* <https://hub.docker.com/_/cassandra>
* <https://medium.com/@kayvan.sol2/deploying-apache-cassandra-cluster-3-nodes-with-docker-compose-3634ef8345e8>



1. Перевірити правильність конфігурації за допомогою

*nodetool status*

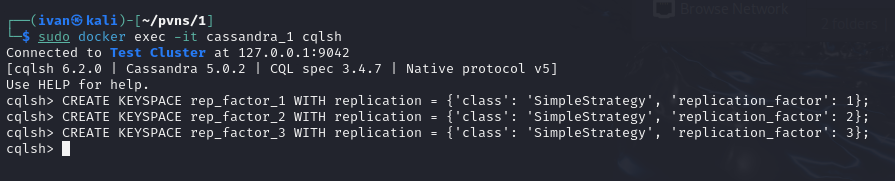
<https://docs.datastax.com/en/dse/6.9/managing/tools/nodetool/status.html>



1. Викоритовуючи  *cqlsh*, створити три *Keyspace* з replication factor 1, 2, 3 з SimpleStrategy

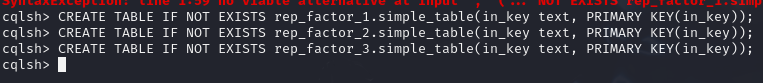
<https://www.tutorialspoint.com/cassandra/cassandra_create_keyspace.htm>

<https://docs.datastax.com/en/cql/cassandra-5.0/develop/keyspace-create.html>



1. В кожному з кейспейсів створити прості таблиці

<https://www.tutorialspoint.com/cassandra/cassandra_create_table.htm>

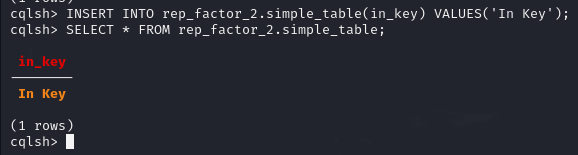


1. Спробуйте писати і читати в ці таблиці підключаюсь на різні ноди.

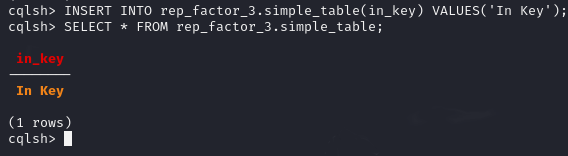
Пишу та зчитую з ноди cassandra\_1 в/на rep\_factor\_1:



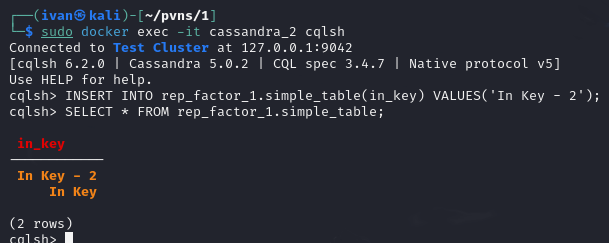
Пишу та зчитую з ноди cassandra\_1 в/на rep\_factor\_2:



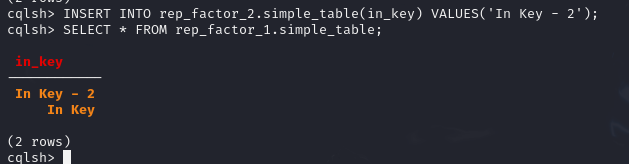
Пишу та зчитую з ноди cassandra\_1 в/на rep\_factor\_3:



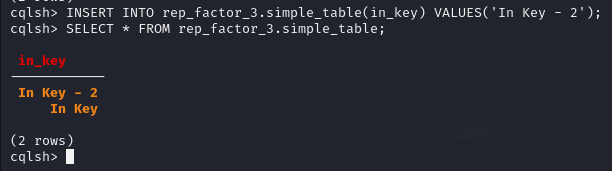
Пишу та зчитую з ноди cassandra\_2 в/на rep\_factor\_1:



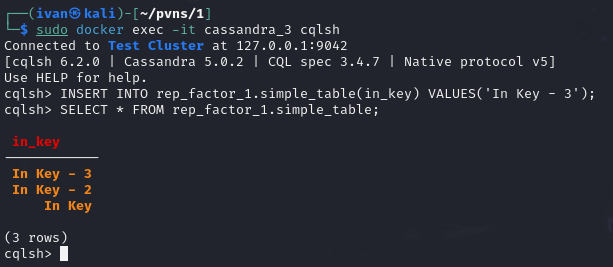
Пишу та зчитую з ноди cassandra\_2 в/на rep\_factor\_2:



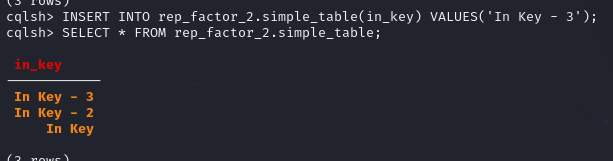
Пишу та зчитую з ноди cassandra\_2 в/на rep\_factor\_3:



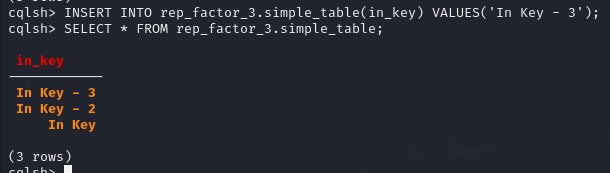
Пишу та зчитую з ноди cassandra\_3 в/на rep\_factor\_1:



Пишу та зчитую з ноди cassandra\_3 в/на rep\_factor\_2:



Пишу та зчитую з ноди cassandra\_3 в/на rep\_factor\_3:



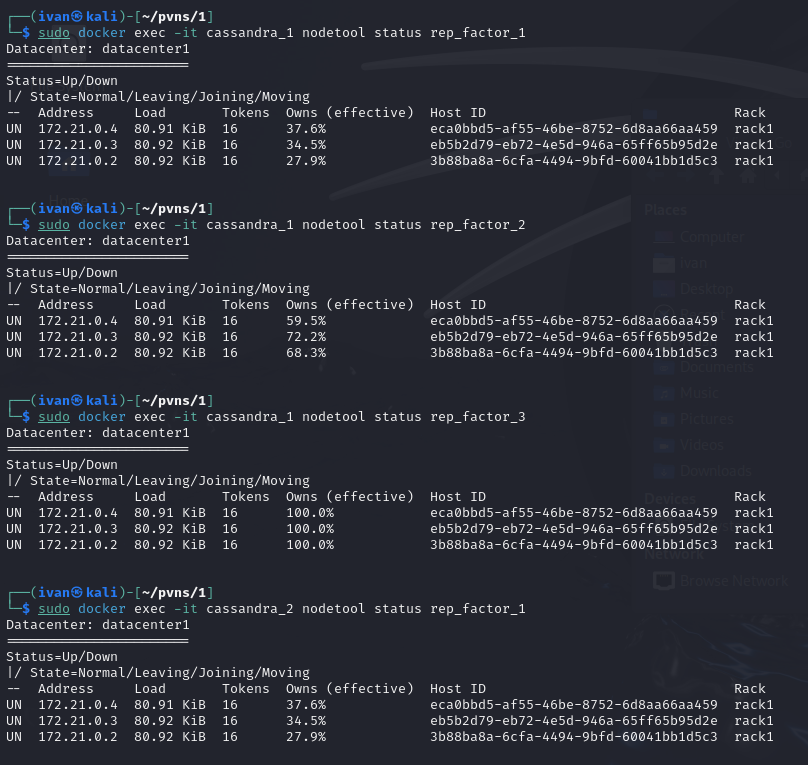
1. Вставте дані в створені таблиці і подивіться на їх розподіл по вузлах кластера для кожного з кейспесов (команда *nodetool status*)

<https://docs.datastax.com/en/cql/3.1/cql/cql_reference/insert_r.html>

<https://docs.datastax.com/en/cql/3.1/cql/cql_reference/select_r.html>

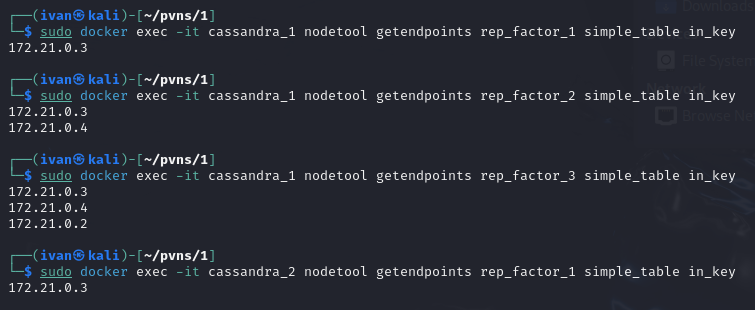
<https://www.tutorialspoint.com/cassandra/cassandra_create_data.htm>

<https://www.tutorialspoint.com/cassandra/cassandra_read_data.htm>

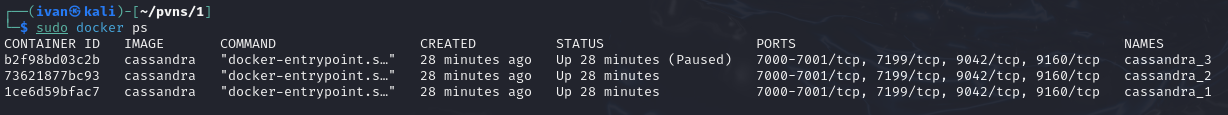


1. Для якогось запису з кожного з кейспейсу виведіть ноди на яких зберігаються дані

<https://docs.datastax.com/en/dse/5.1/dse-admin/datastax_enterprise/tools/nodetool/toolsGetEndPoints.html>



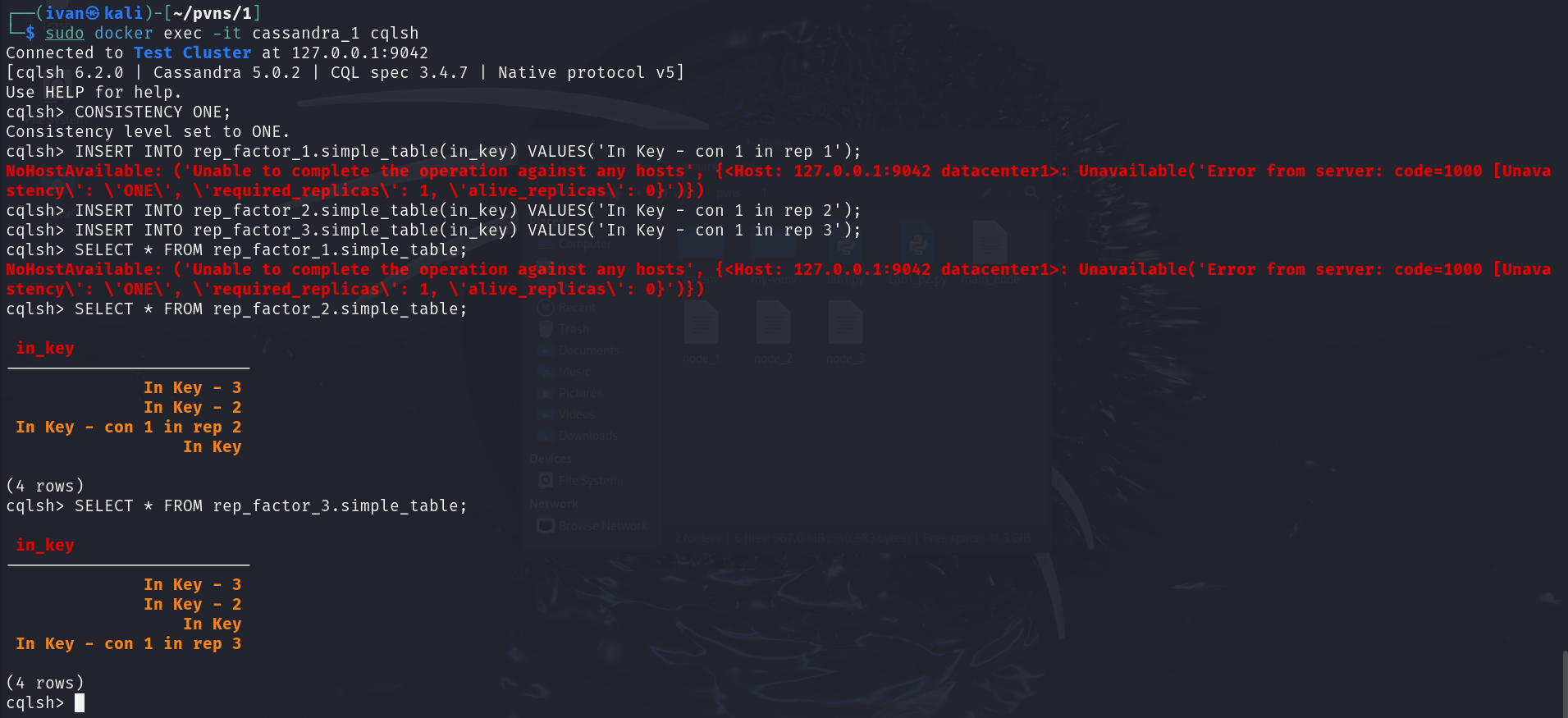
1. Відключити одну з нод. Для кожного з кейспейсів перевірити з якими рівнями *consistency* можемо читати та писати



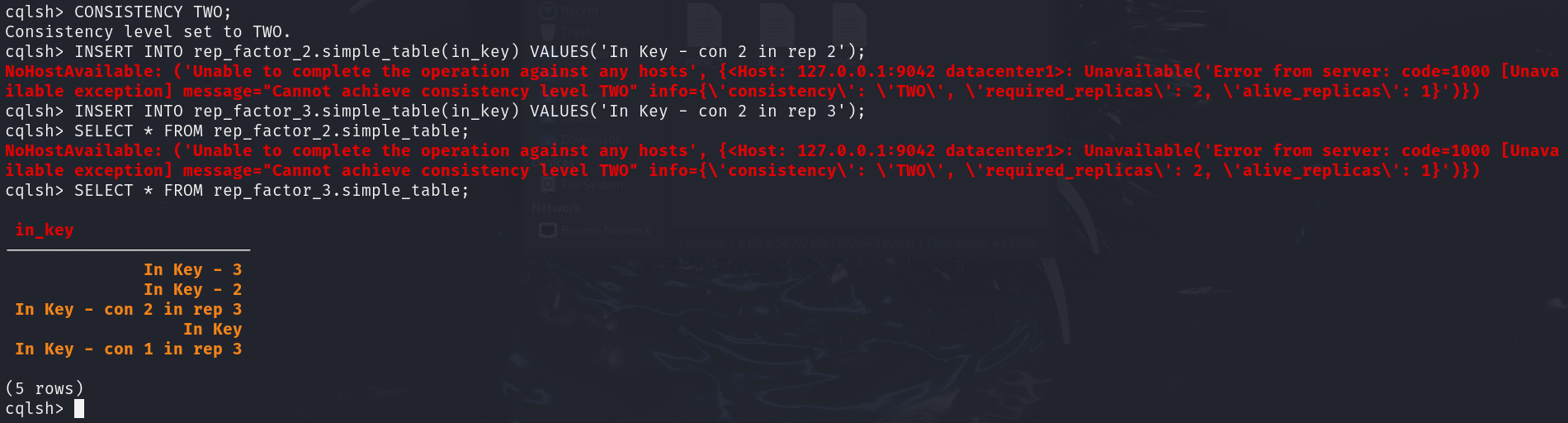
* для *Keyspace* з replication factor 1 - **CONSISTENCY ONE**
* для *Keyspace* з replication factor 2 - **CONSISTENCY ONE/TWO**
* для *Keyspace* з replication factor 3 - **CONSISTENCY ONE/TWO/THREE**

<https://docs.datastax.com/en/cql/3.1/cql/cql_reference/consistency_r.html>

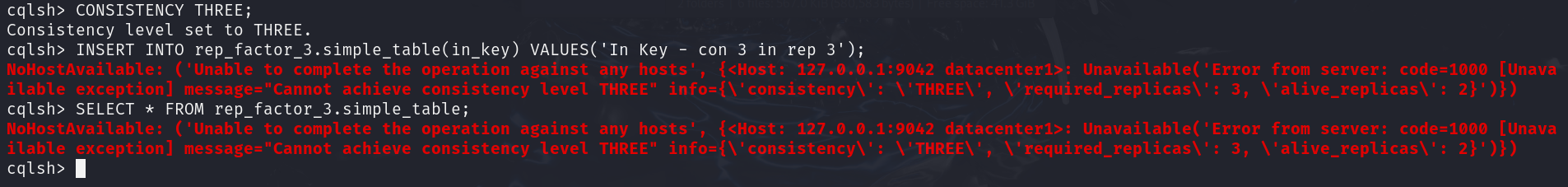
CONSISTENCY ONE для keyspace з усіма replication\_factor 1, 2 та 3:



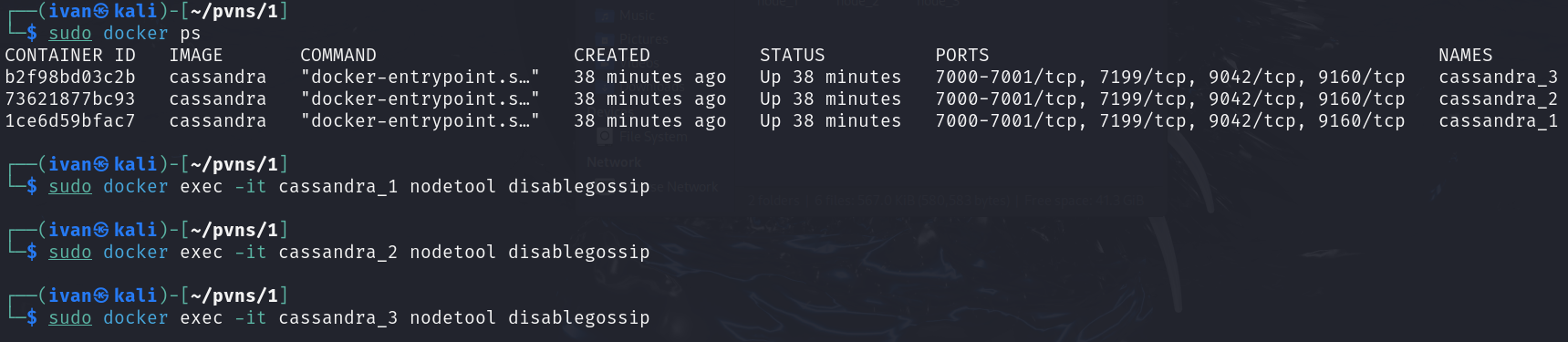
CONSISTENCY TWO для keyspace з replication factor 2 та 3:



 CONSISTENCY THREE для keyspace з replication factor 3:



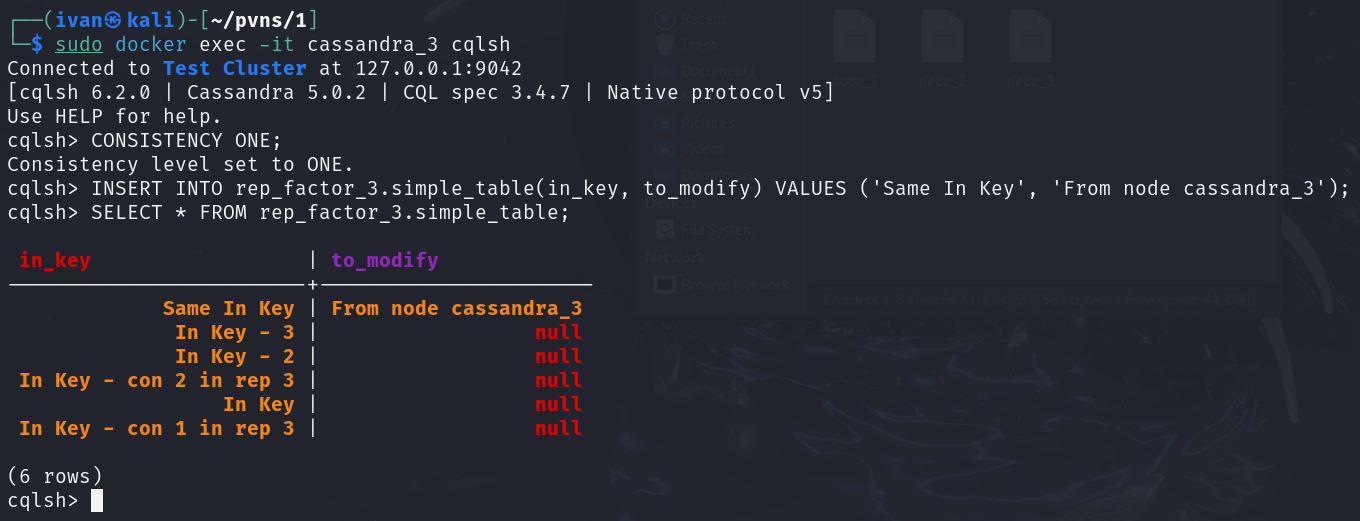
1. Зробить так щоб три ноди працювали, але не бачили одна одну по мережі (заблокуйте чи відключити зв'язок між ними)

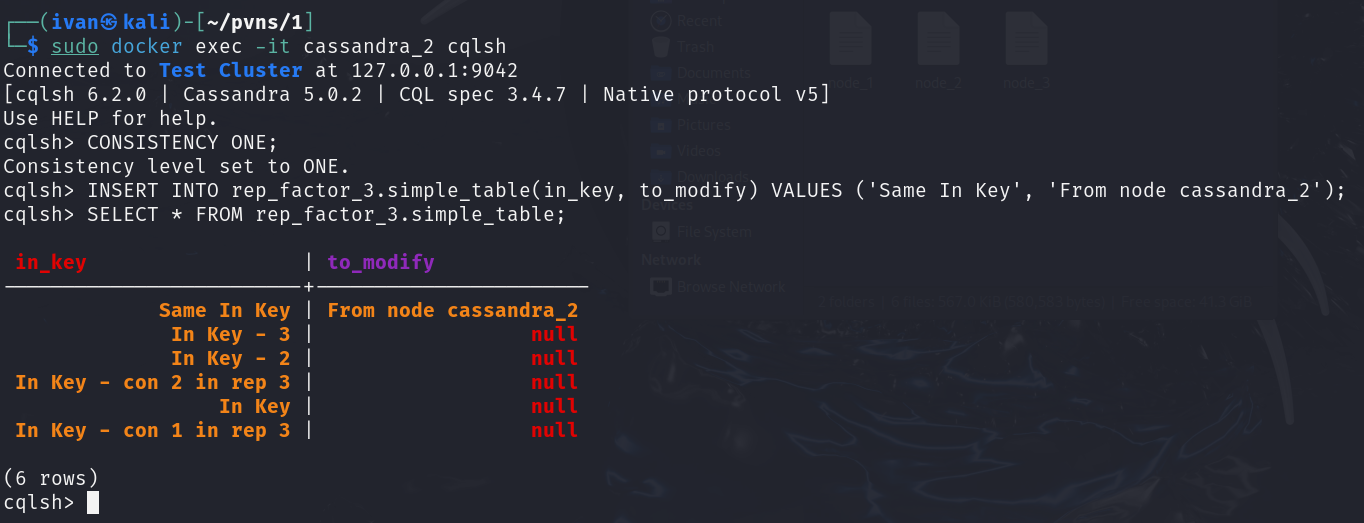


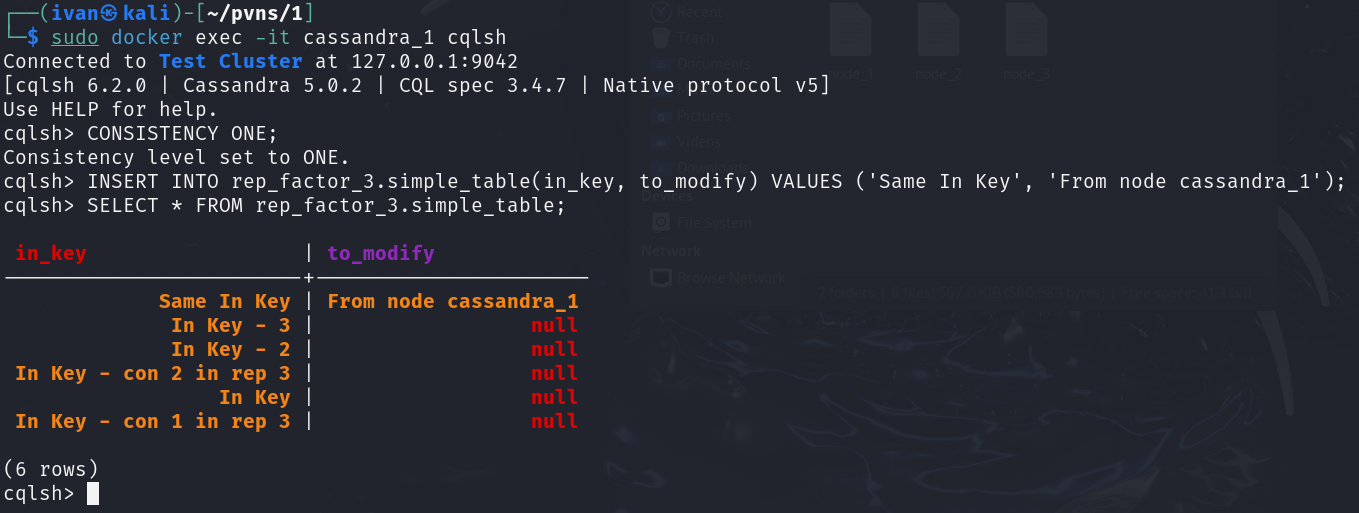
1. Для кейспейсу з *replication factor* 3 задайте рівень consistency рівним 1. Виконайте по черзі запис значення з однаковим primary key, але різними іншими значенням окремо на кожну з нод (тобто створіть конфлікт)

Модифікував прості таблиці на кожному replication factor:

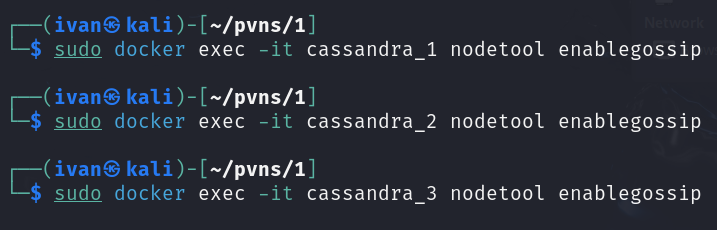


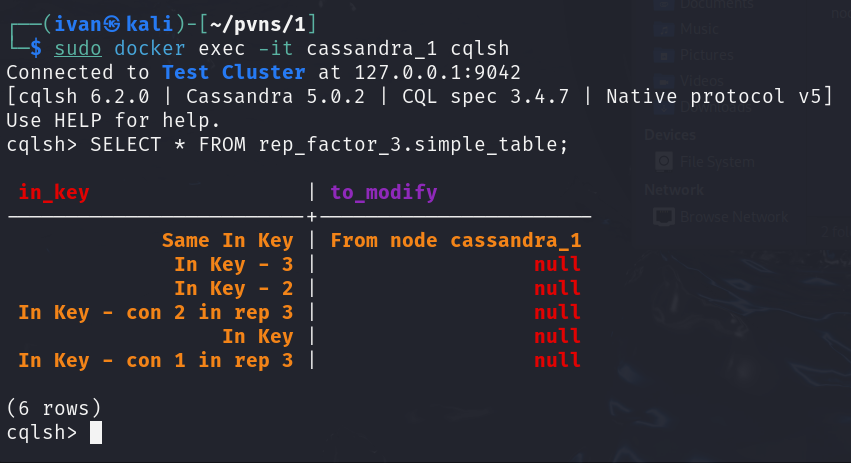






1. Відновіть зв'язок між нодами, і перевірте що вони знову об'єдналися у кластер. Визначте яким чином була вирішений конфлікт даних та яке значення було прийнято кластером та за яким принципом





Кластер обрав значення «From node cassandra\_1» для за принципом «last-write-wins», оскільки я спочатку записував значення на ноду cassandra\_3, потім на ноду cassandra\_2 і лише потім на ноду cassandra\_1.