

[Neo4j v5] HowTo convert a Neo4j *.dump file from BLOCK format to ALIGNED format

Neo4j introduces a new file format called BLOCK that they rolled out in April 2024. This new format is “enterprise only” which breaks portability of dump files. If you take a .dump file from AuraDB (which is an Enterprise version) and attempt to load it locally into our indr-neo4j docker container (which is “Community” version) it will fail with ...

```
Loading dump file...
```

```
Done: 218 files, 6.706MiB processed.
```

```
Failed to load database 'neo4j': Block format detected for database neo4j but unavailable in this edition. Load failed for databases: 'neo4j'
```

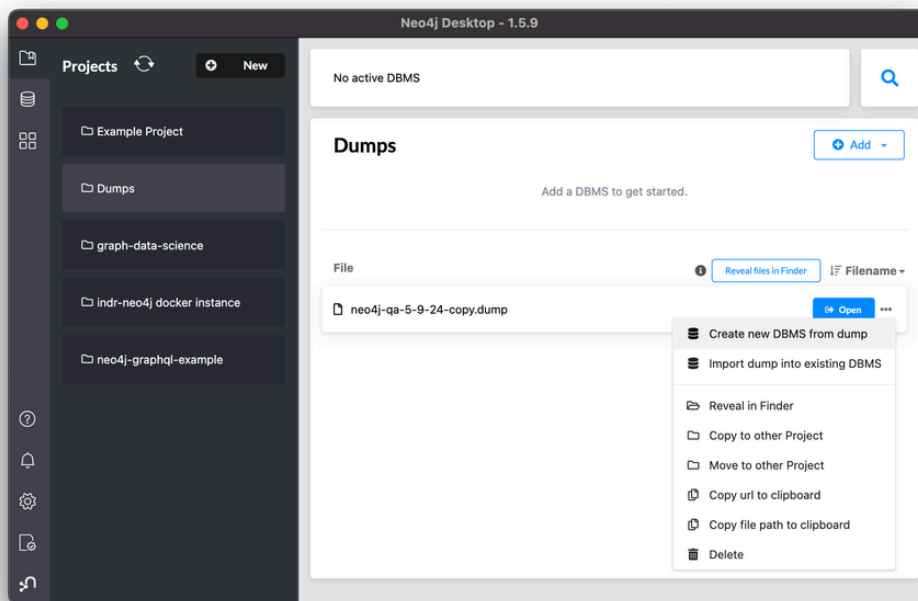
```
Load failed for databases: 'neo4j'
```

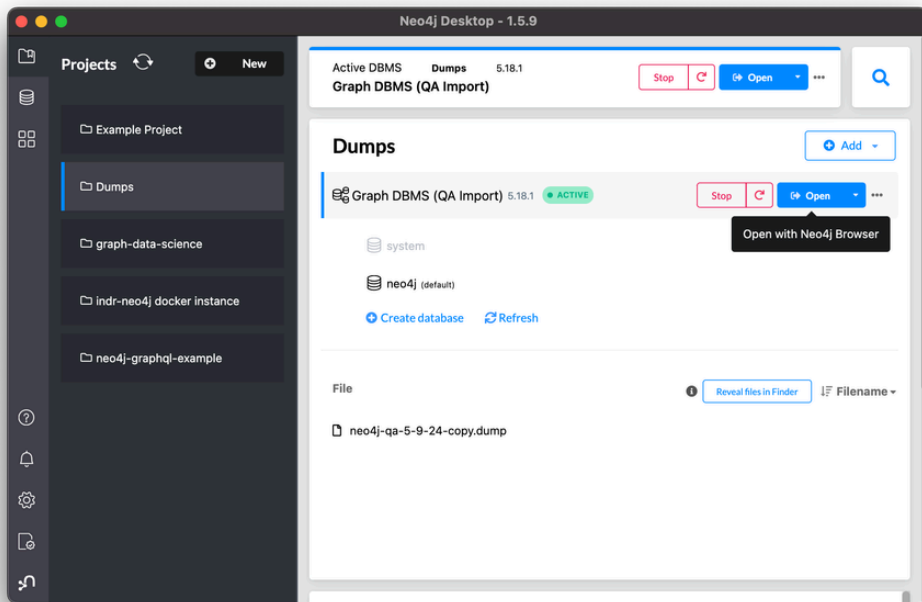
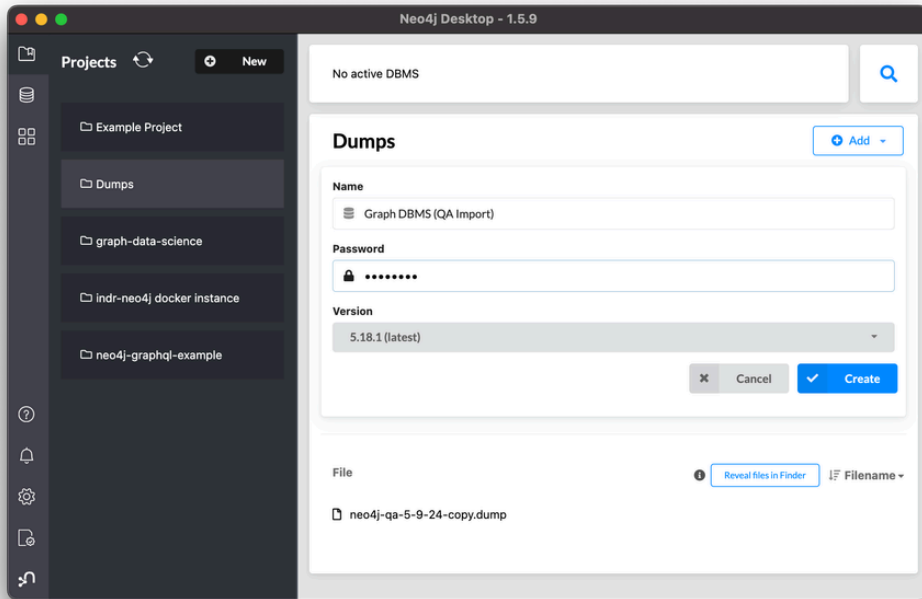
There is a cumbersome workaround which uses Neo4j Desktop (Desktop is actually a developer licensed version of Enterprise).

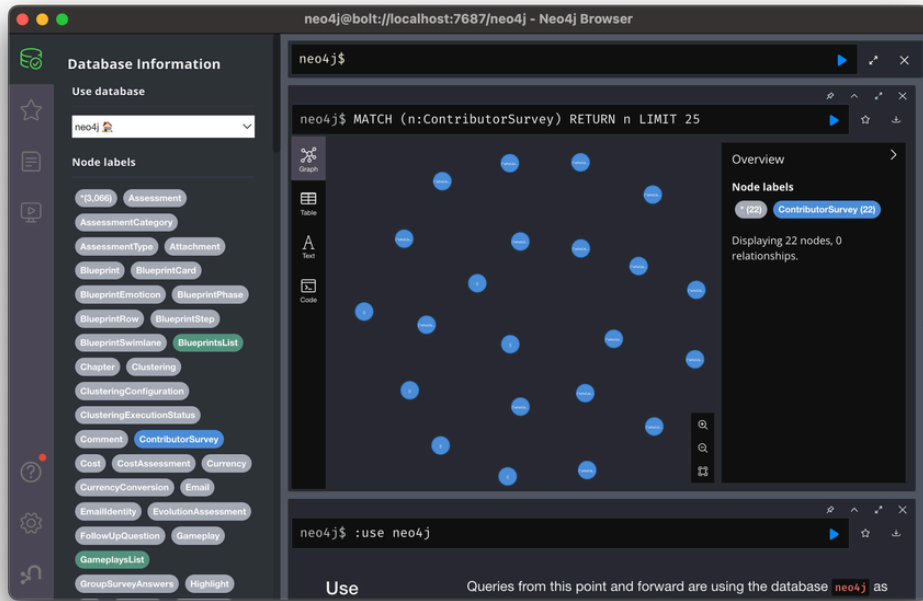
The steps below show how to import a BLOCK database into Neo4j Desktop and then copy it to a new database using the ALIGNED storage format. With an ALIGNED format we can import into our local Docker indr-neo4j instance

Steps

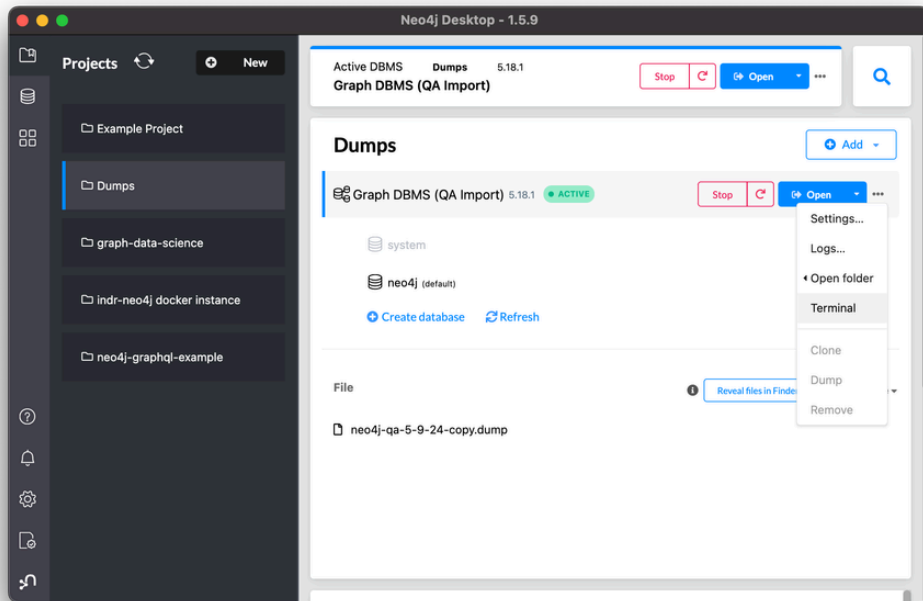
Add .dump file to Neo4j Desktop and **'Create new DBMS from dump'**







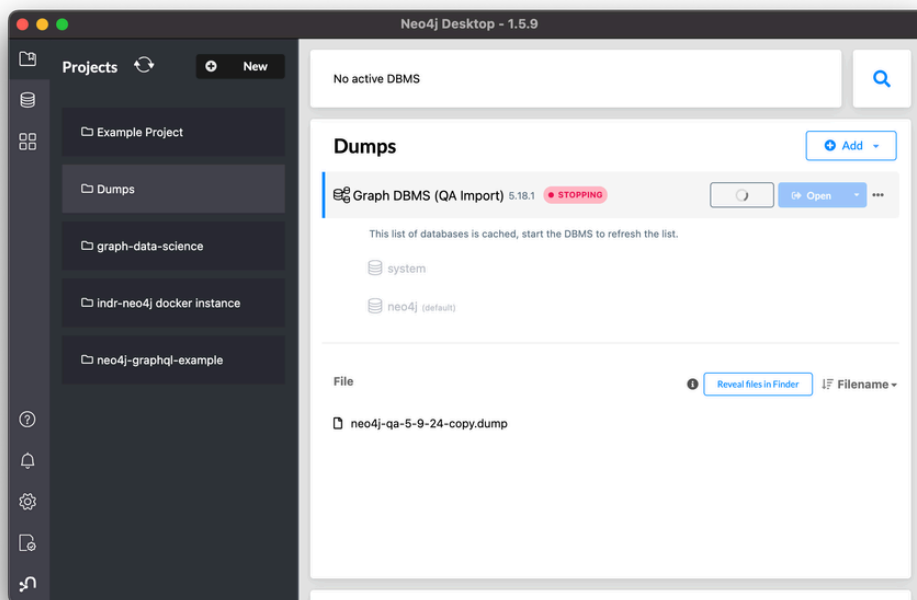
Open the Neo4j Desktop Terminal using menu



Will open a terminal window to a funky location where this graph db was created.

```
Neo4j Desktop Terminal - Graph DBMS (QA Import)
jblackwell@sugar21 dbms-50855278-68a7-4413-b881-44f2e5471108 % pwd
/Users/jblackwell/Library/Application Support/Neo4j Desktop/Application/relate-data/dbms/dbms-50855278-68a7-4413-b881-44f2e5471108
jblackwell@sugar21 dbms-50855278-68a7-4413-b881-44f2e5471108 % ls -al
total 288
drwxr-xr-x  22 jblackwell  staff    704 Apr 10 11:40 .
drwxr-xr-x   7 jblackwell  staff   224 Apr 10 11:39 ..
-rw-r--r--   1 jblackwell  staff 22099 Apr 10 11:39 LICENSE.txt
-rw-r--r--   1 jblackwell  staff 101215 Apr 10 11:39 LICENSES.txt
-rw-r--r--   1 jblackwell  staff   135 Apr 10 11:39 NOTICE.txt
-rw-r--r--   1 jblackwell  staff  1527 Apr 10 11:39 README.txt
-rw-r--r--   1 jblackwell  staff    94 Apr 10 11:39 UPGRADE.txt
drwxr-xr-x   6 jblackwell  staff   192 Apr 10 11:39 bin
drwxr-xr-x   4 jblackwell  staff   128 Apr 10 11:39 certificates
drwxr-xr-x   8 jblackwell  staff   256 Apr 10 11:39 conf
drwxr-xr-x   7 jblackwell  staff   224 Apr 10 11:40 data
drwxr-xr-x   2 jblackwell  staff    64 Apr 10 11:39 import
drwxr-xr-x   5 jblackwell  staff   160 Apr 10 11:39 labs
drwxr-xr-x  328 jblackwell  staff 10496 Apr 10 11:39 lib
drwxr-xr-x   2 jblackwell  staff    64 Apr 10 11:39 licenses
drwxr-xr-x   7 jblackwell  staff   224 Apr 10 11:40 logs
drwxr-xr-x  107 jblackwell  staff  3424 Apr 10 11:41 metrics
-rw-r--r--   1 jblackwell  staff    53 Apr 10 11:39 packaging_info
drwxr-xr-x   4 jblackwell  staff   128 Apr 10 11:39 plugins
drwxr-xr-x   8 jblackwell  staff   256 Apr 10 11:39 products
-rw-r--r--   1 jblackwell  staff  1262 Apr 10 11:40 relate.dbms.json
drwxr-xr-x   3 jblackwell  staff    96 Apr 10 11:40 run
jblackwell@sugar21 dbms-50855278-68a7-4413-b881-44f2e5471108 %
```

Need to stop database before we can copy



cd to "bin" and run the following command:

```
./neo4j-admin database copy --to-format=aligned neo4j neo4jCopy
```

careful with dashes in the database names.

from [Naming rules for databases - Operations Manual](#)

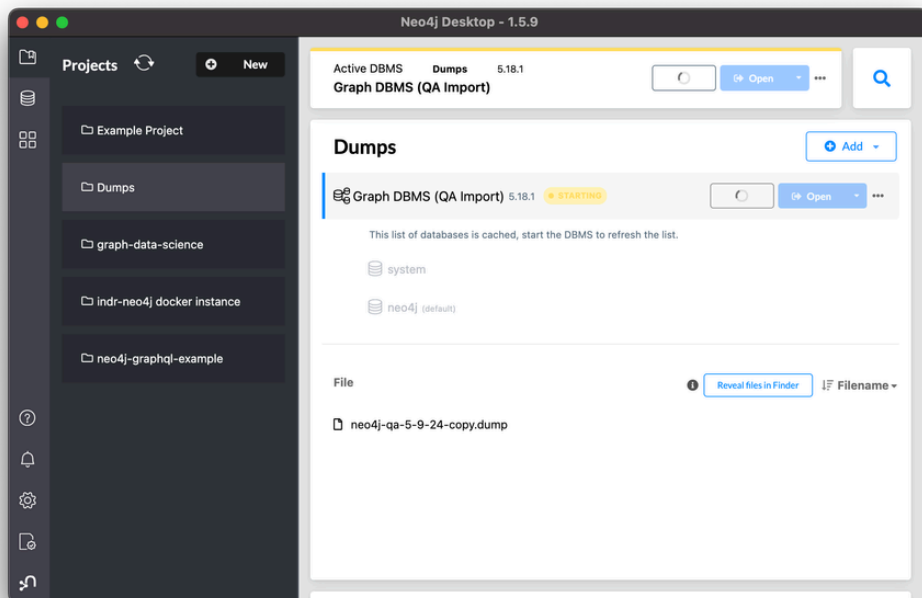
The - (dash) and . (dot) characters are not legal in Cypher variables. Names with a - in them must be enclosed within backticks. For example, `CREATE DATABASE `main-db`` is a valid database name

This will spit out a ton with hopefully no errors.

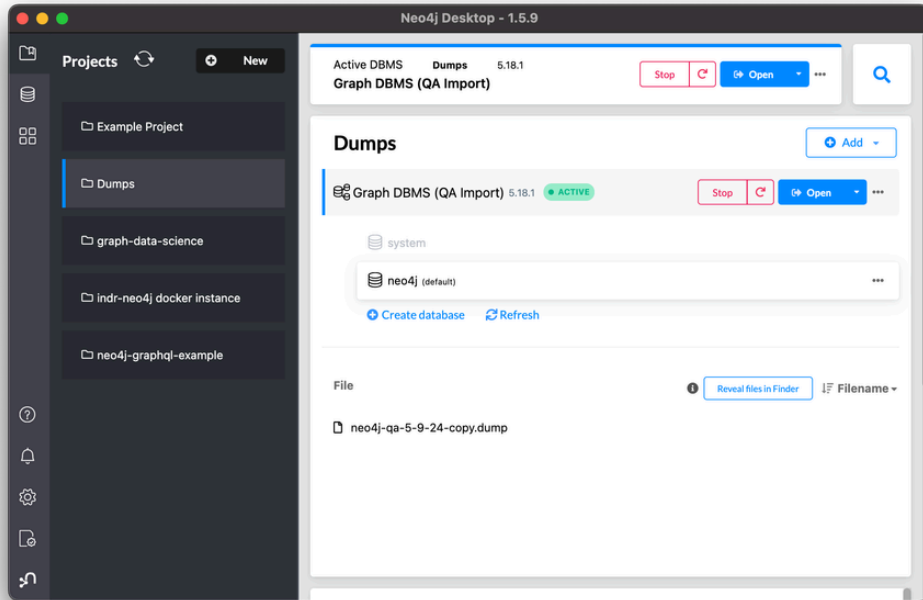


```
Neo4j Desktop Terminal - Graph DBMS (QA Import)
jblackwell@sugar21 bin % ls
completion      cypher-shell    neo4j           neo4j-admin
jblackwell@sugar21 bin % ./neo4j-admin database copy --to-format=aligned neo4j neo4jCOPY
```

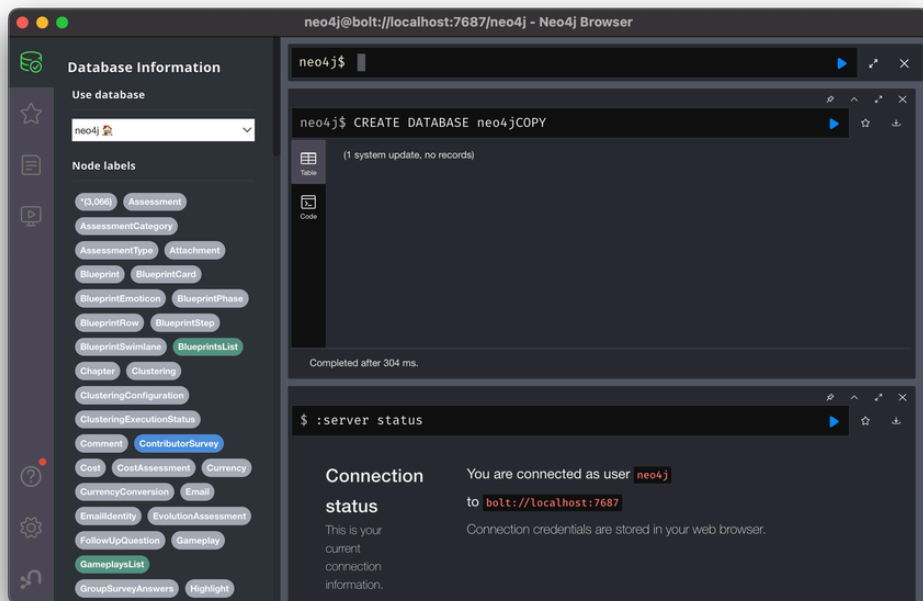
Start database



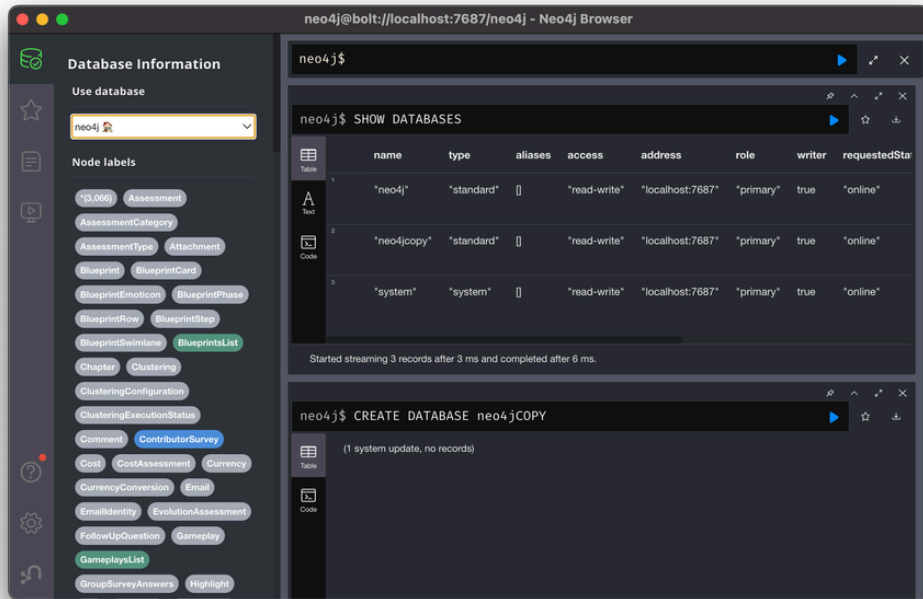
You will not see your newly copied database listed on restart. Don't worry, keep going...



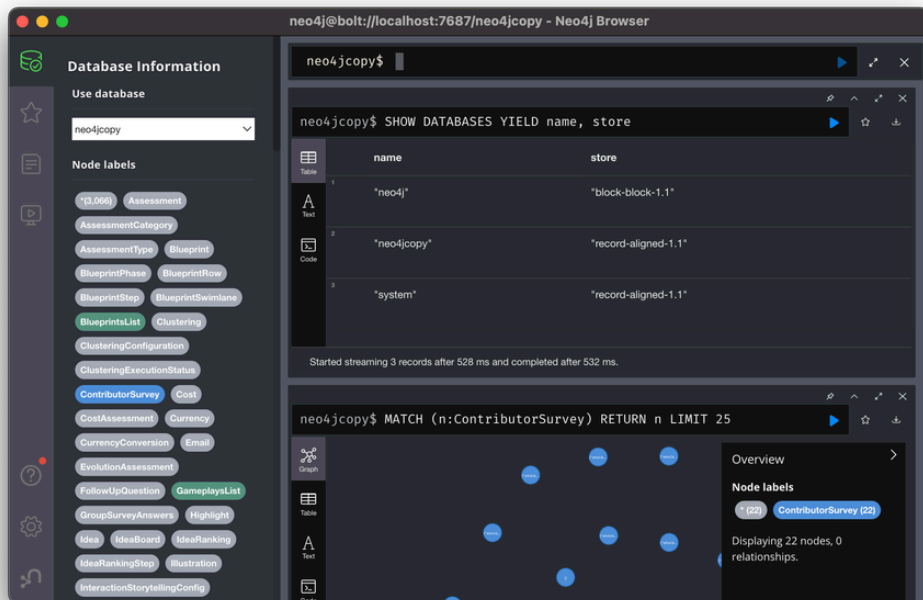
Weird part: Manually create a database with the same name as the one you copied to. In this case **neo4jCOPY**. It will pick up the database files on the file system automatically.



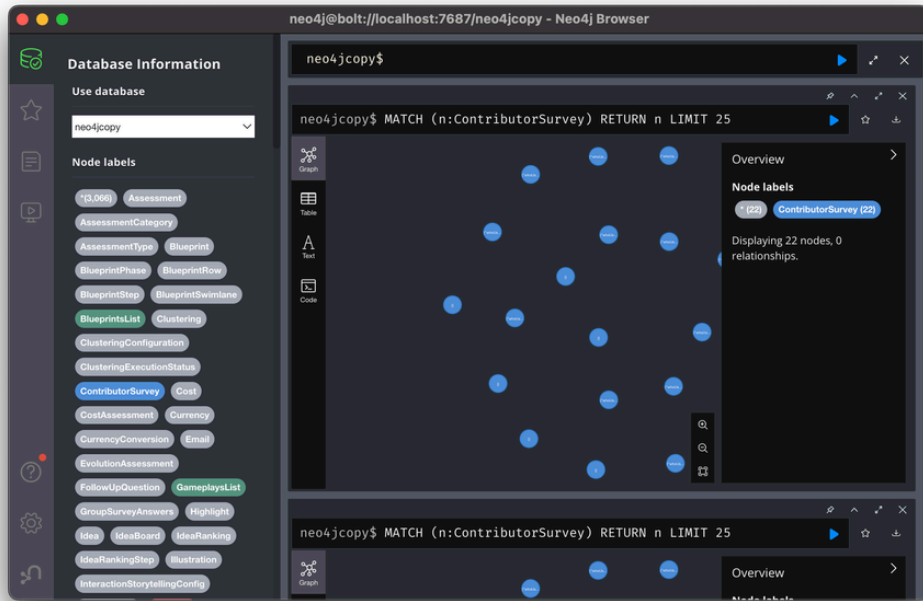
You should see it listed here now



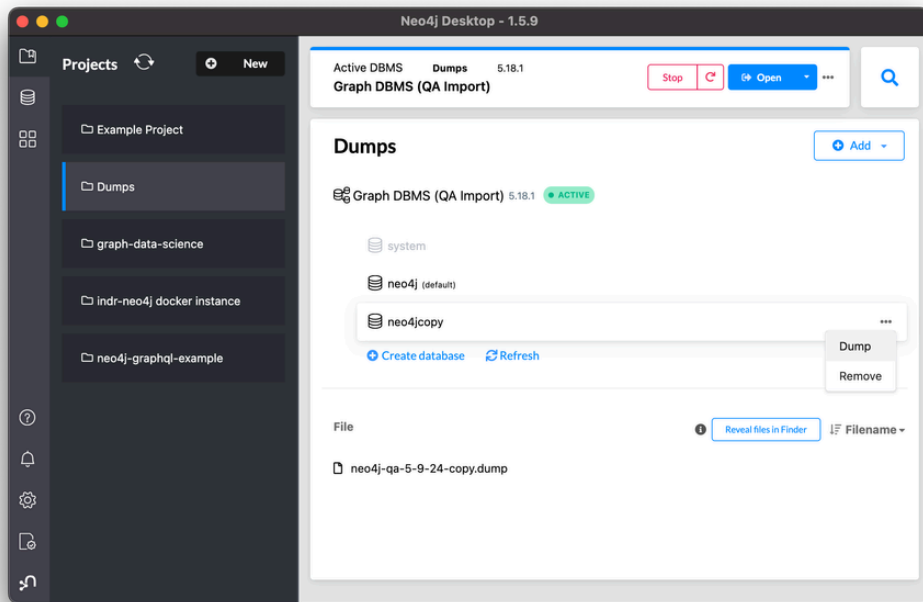
If you run `SHOW DATABASES YIELD name, store` you can see the storage format. Note your copied database says **"record-aligned-1.1"** instead of **"block-block-1.1"**



And if you select your new copied database you should see all your data.



The last step is to go back to Neo4j Desktop and select the elipses menu and “dump” the file.



You can now import this new .dump version into Docker indr-neo4j.


```
[root@ip-10-10-71-30 import]# ls
load.sh neo4j-05-08-24.dump neo4jcopy.dump neo4j-qa-04-28-24.dump neo4j-qa-5-9-24.dump neo4j-qa5-9-from-desktop.dump
[root@ip-10-10-71-30 import]# ./load.sh neo4jcopy.dump
Loading neo4jcopy.dump
Stopping Neo4j..... stopped.
Loading dump file...
Done: 70 files, 2.369MiB processed.
Directories in use:
home: /var/lib/neo4j
config: /etc/neo4j
logs: /var/log/neo4j
plugins: /var/lib/neo4j/plugins
import: /var/lib/neo4j/import
data: /var/lib/neo4j/data
certificates: /var/lib/neo4j/certificates
licenses: /var/lib/neo4j/licenses
run: /var/lib/neo4j/run
Starting Neo4j.
Started neo4j (pid:11714). It is available at https://0.0.0.0:7473
There may be a short delay until the server is ready.
[root@ip-10-10-71-30 import]#
```