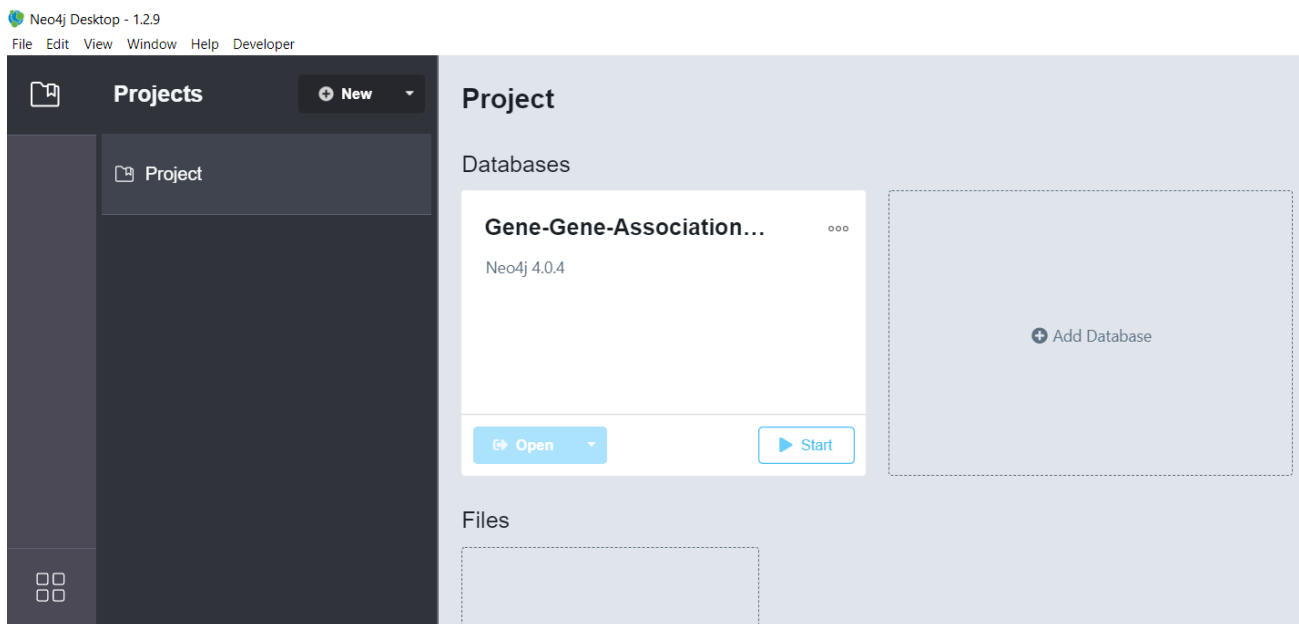


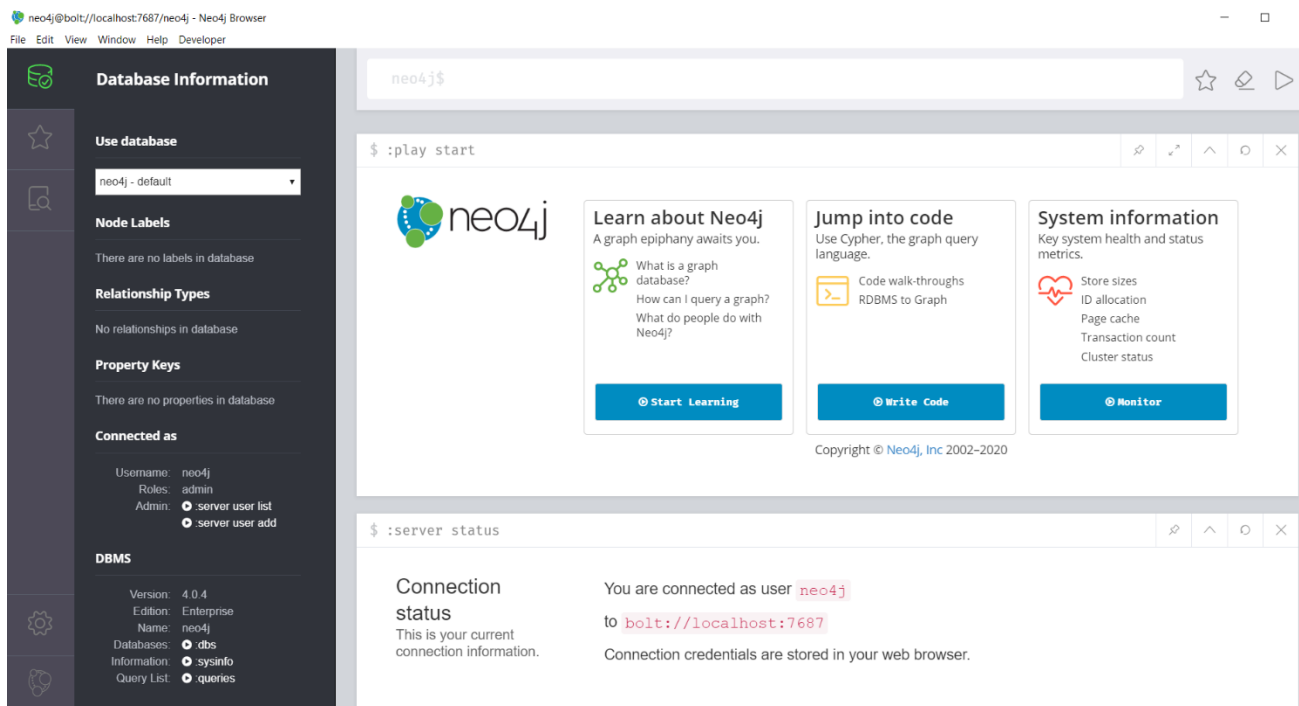
## Neo4j

### نصب Neo4j و import دیتاست

ابتدا از آدرس <https://neo4j.com/download/> نسخه Neo4j Desktop که free می باشد، دانلود و نصب گردید. (حجم فایل دانلودی حدود 530 MB است). پس از نصب ابتدا یک Project ساخته شده و در آن با کلیک بر روی گزینه Add Database، دیتابیس موردنظرمان را از نوع «Create a Local Graph» با نام Gene-Gene-Associations-50k می سازیم.



حال با انتخاب گزینه Start، دیتابیس را start کرده و سپس با زدن کلید open، دیتابیس روی پورت 7687 در browser بالا می آید که مشابه شکل زیر است. همانگونه که مشاهده می شود، هیچ Node و Relationship ای وجود ندارد. در قسمت بالای صفحه و بخش neo4j\$ امکان تایپ دستورات وجود دارد که با زدن کلید ctrl+enter اجرا خواهند شد.



ابتدا فایل `gene_gene_associations_50k.csv` به مسیر `import` در پوشه‌ای که دیتابیس ساخته شده که بصورت زیر بوده، کپی شد:

`F:\Programs\Neo4jDesktop\neo4jDatabases\database-81944467-6bc1-469a-a0dd-337376ae0f61\installation-4.0.4\import`

سپس دستور زیر در محیط `neo4j browser` نوشته و اجرا شد:

```
LOAD CSV WITH HEADERS FROM "file:///gene_gene_associations_50k.csv" AS row
```

```
merge(n:GeneNode{Name:row.OFFICIAL_SYMBOL_A})
```

```
merge(m:GeneNode{Name:row.OFFICIAL_SYMBOL_B})
```

```
merge(n)-[:TO{experimental_system:row.EXPERIMENTAL_SYSTEM}]->(m)
```

پس از اجرای دستورات بالا که چند دقیقه ای زمان می‌برد، تعداد 9656 نود و تعداد 46621 رابطه ایجاد می‌شود.

## تمرینها

باید بر اساس داده های import شده به سوالات زیر پاسخ داده شود :

1. What is the number of nodes returned?
2. What's the number of edges?
3. The number of loops in the graph is:
4. The query match (n)-[r]->(m) where m <> n return distinct n, m, count(r) gives us what?
5. The query match (n)-[r]->(m) where m <> n return distinct n, m, count(r) as myCount order by myCount desc limit 1 produces what?
6. The query match p=(n {Name:'BRCA1'})-[:AssociationType\*..2]->(m) return p produces what?
7. How many non-directed shortest paths are there between the node named 'BRCA1' and the node named 'NBR1'?
8. The top 2 nodes with the highest outdegree are:
9. Applying the example queries provided to you, create the degree histogram for the network. How many nodes in the graph have a degree of 3?

## تمرین اول

برای تعیین تعداد نودها؛ query زیر اجرا می شود :

```
neo4j$ match(n) return count(n)
```

	count(n)
1	9656

## تمرین دوم

برای تعیین تعداد یالها یا رابطه ها query زیر اجرا می شود :

```
neo4j$ match(n)-[r]→() return count(r)
```

	count(r)
1	46621

## تمرین سوم

برای تعیین تعداد loop ها، اسکریپت زیر نوشته شد:

```
neo4j$ match(n)-[r]→(n) return count(n)
```

	count(n)
1	1221

## تمرین چهارم

در این query، روابطی که به شکل loop نیستند، شناسایی شده و به ازای هر ترکیب غیر تکراری از دو نود مبدأ و مقصد، تعداد روابط بین آنها آورده شده است.

```
neo4j$ match (n)-[r]→(m) where m<n return distinct n,m,count(r)
```

"n"	"m"	"count(r)"
{"Name": "AKT1"}	{"Name": "MAP2K4"}	2
{"Name": "SPAG9"}	{"Name": "MAP2K4"}	1
{"Name": "ARRB2"}	{"Name": "MAP2K4"}	1
{"Name": "MAP3K8"}	{"Name": "MAP2K4"}	1
{"Name": "MAP3K4"}	{"Name": "MAP2K4"}	1
{"Name": "MAP4K2"}	{"Name": "MAP2K4"}	1
{"Name": "NPHS1"}	{"Name": "MAP2K4"}	1
{"Name": "MAP3K11"}	{"Name": "MAP2K4"}	1
{"Name": "MAP3K1"}	{"Name": "MAP2K4"}	1
{"Name": "MAPK8IP3"}	{"Name": "MAP2K4"}	1
{"Name": "FLNC"}	{"Name": "MAP2K4"}	1

### تمرین پنجم

در این query، بخش اول آن، مشابه تمرین قبلی است و شرایطی به انتهای آن اضافه شده است که بدین معناست که خروجی تمرین بالا بصورت نزولی بر اساس ستون myCount که به عنوان یک alias برای count(r) تعریف شده مرتب شود و سپس اولین سطر آن به عنوان خروجی برگردانده شود که نشان می دهد بیشترین تعداد رابطه از نود BRCA1 به نود RBBP8 به تعداد 7 رابطه می باشد که گراف متناظر خروجی هم، این را نشان می دهد. البته در گراف سایر روابط این دو نود هم نشان داده شده است.

```
neo4j$ match (n)-[r]→(m) where m<n return distinct n,m,count(r) as myCount order by myCount desc limit 1
```

```
neo4j$ match (n)-[r]→(m) where m<n return distinct n,m,count(r) as myCount order by m...
```

Graph

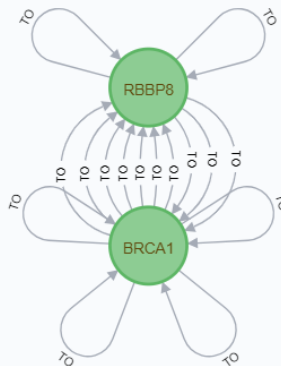
\*(2) GeneNode(2)

\*(16) TO(16)

Table

Text

Code



```
neo4j$ match (n)-[r]→(m) where m<n return distinct n,m,count(r) as myCount order by m...
```

Graph

"n"	"m"	"myCount"
{"Name": "BRCA1"}	{"Name": "RBBP8"}	7

Table

Text

## تمرین ششم

این query خروجی ندارد و بدین معناست که از نود BRCA1 به هر نود دیگری رابطه ای از نوع AssociationType تا دو همسایه بعد وجود ندارد.

```
neo4j$ match p=(n{Name:'BRCA1'})-[:AssociationType*..2]→(m) return p
```

Table

(no changes, no records)

Code

### تمرین هفتم

این query بصورت زیر نوشته شد و خروجی ندارد. می توان بعد از return نوشت `count(r)`. ولی در هر حال خروجی ندارد و یعنی اینکه بین این دو نود، رابطه ای برقرار نیست چه از BRCA1 به NBR1 و یا از NBR1 به BRCA1.

```
neo4j$ match (n{Name:'BRCA1'})-[r]-(m{Name:'NBR1'}) return n,m,r
```

Table

(no changes, no records)

Code

### تمرین هشتم

برای این تمرین، query بصورت زیر نوشته شد تا دو نود با بالاترین outdegree (تعداد رابطه های خروجی از نود) بدست آید:

```
neo4j$ match (n)-[r]→() return n.Name as Node, count(r) as Outdegree order by Outdegree desc limit 2
```

neo4j\$ match (n)-[r]→() return n.Name as Node, count(r) as Outdegree order by Outdegree...

Table

"Node"	"Outdegree"
"SNCA"	277
"BRCA1"	234

Text

Code

### تمرین نهم

برای این تمرین برای هر نود، degree (تعداد رابطه های ورودی و خروجی) آن محاسبه می شود و سپس نودهای دارای degree یکسان گروه بندی شده و بصورت صعودی بر اساس میزان degree نمایش داده می شوند. در شکل زیر، query و نتیجه آن آورده شده است. همانطور که پیداست، تعداد 821 نود دارای degree برابر 3 می باشند.

```
neo4j$ match (n)-[r]-() with n as nodes, count(distinct r) as degree return  
degree, count(nodes) order by degree asc
```

```
neo4j$ match (n)-[r]-() with n as nodes, count(distinct r) as degree return degree, coun...
```



Table



Text



Code

"degree"	"count(nodes)"
1	2610
2	1351
3	821
4	675
5	512
6	427
7	321
8	254
9	212
10	194