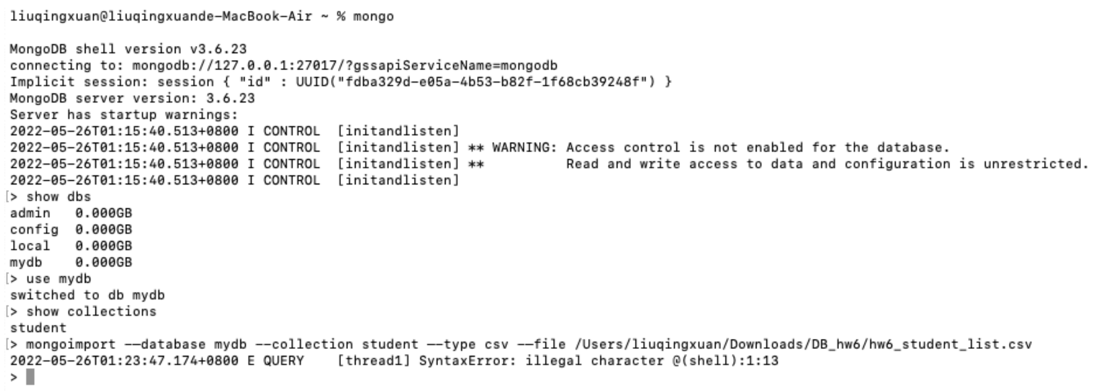
NTU Database Management System – from SQL to NoSQL – Homework 4

資料科學 **R10946013** 劉謦瑄

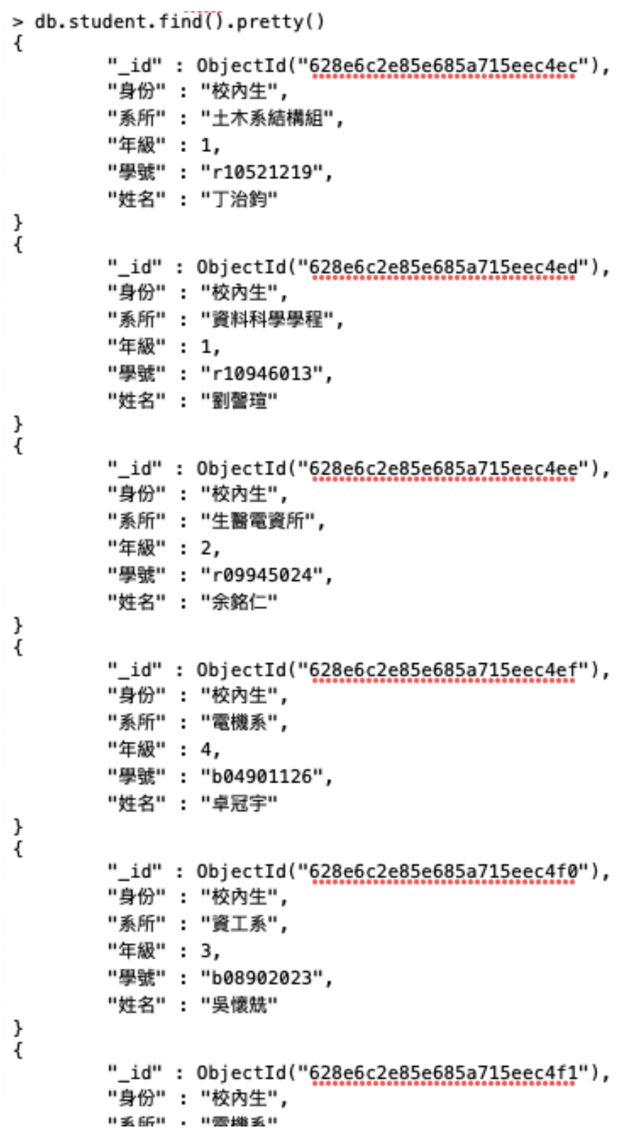
Part 1-1.

創建 mydb 後使用該資料庫，並創建 student 的 collection，再將 CSV import 到資料庫中顯示出結果。



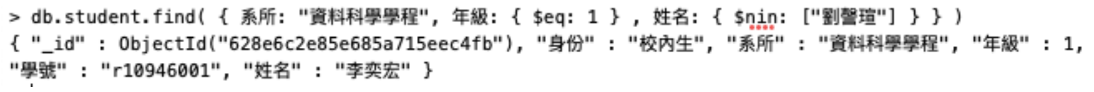
```
liuqingxuan@liuqingxuande-MacBook-Air ~ % mongo

MongoDB shell version v3.6.23
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("fdba329d-e05a-4b53-b82f-1f68cb39248f") }
MongoDB server version: 3.6.23
Server has startup warnings:
2022-05-26T01:15:40.513+0800 I CONTROL  [initandlisten]
2022-05-26T01:15:40.513+0800 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-05-26T01:15:40.513+0800 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2022-05-26T01:15:40.513+0800 I CONTROL  [initandlisten]
[> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
mydb    0.000GB
[> use mydb
switched to db mydb
[> show collections
student
[> mongoimport --database mydb --collection student --type csv --file /Users/liuqingxuan/Downloads/DB_hw6/hw6_student_list.csv
2022-05-26T01:23:47.174+0800 E QUERY    [thread1] SyntaxError: illegal character @(shell):1:13
> 
```

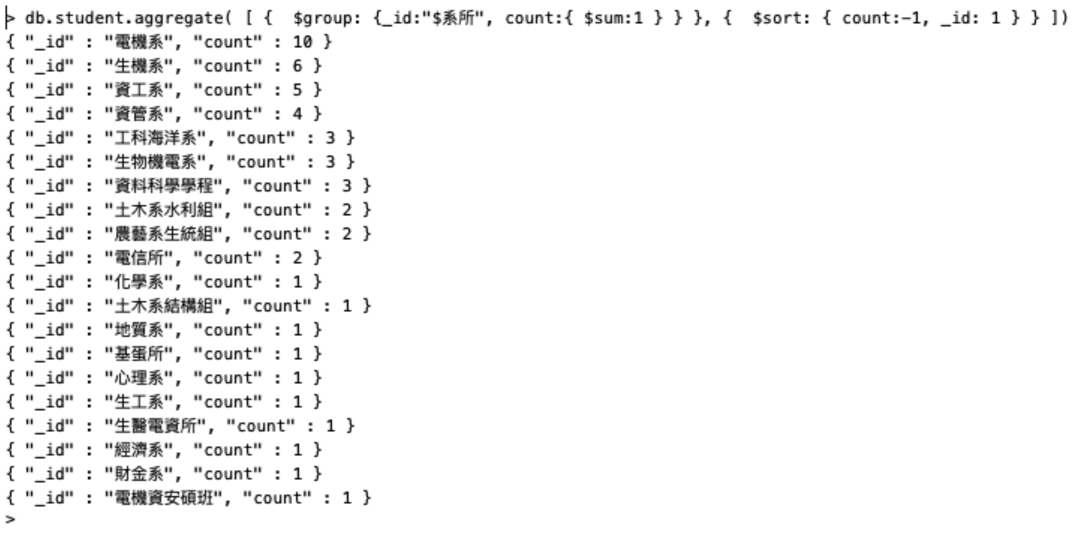


```
> db.student.find().pretty()
{
        "_id" : ObjectId("628e6c2e85e685a715eec4ec"),
        "身份" : "校內生",
        "系所" : "土木系結構組",
        "年級" : 1,
        "學號" : "r10521219",
        "姓名" : "丁治鈞"
}
{
        "_id" : ObjectId("628e6c2e85e685a715eec4ed"),
        "身份" : "校內生",
        "系所" : "資料科學學程",
        "年級" : 1,
        "學號" : "r10946013",
        "姓名" : "劉謦瑄"
}
{
        "_id" : ObjectId("628e6c2e85e685a715eec4ee"),
        "身份" : "校內生",
        "系所" : "生醫電資所",
        "年級" : 2,
        "學號" : "r09945024",
        "姓名" : "余銘仁"
}
{
        "_id" : ObjectId("628e6c2e85e685a715eec4ef"),
        "身份" : "校內生",
        "系所" : "電機系",
        "年級" : 4,
        "學號" : "b04901126",
        "姓名" : "卓冠宇"
}
{
        "_id" : ObjectId("628e6c2e85e685a715eec4f0"),
        "身份" : "校內生",
        "系所" : "資工系",
        "年級" : 3,
        "學號" : "b08902023",
        "姓名" : "吳懷昶"
}
{
        "_id" : ObjectId("628e6c2e85e685a715eec4f1"),
        "身份" : "校內生",
        "系所" : "電機系"
```

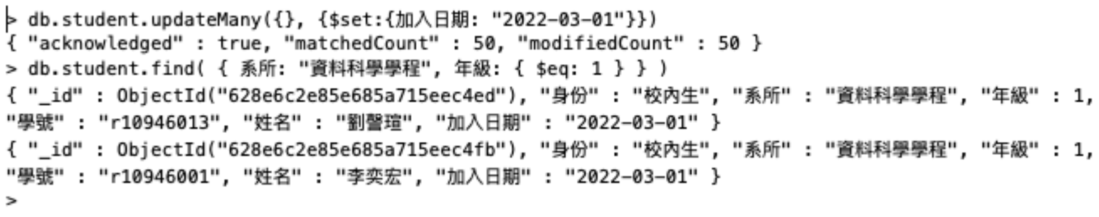
Part 1-2.


```
> db.student.find( { 系所: "資料科學學程", 年級: { $eq: 1 } , 姓名: { $nin: ["劉謦瑄"] } } )
{ "_id" : ObjectId("628e6c2e85e685a715eec4fb"), "身份" : "校內生", "系所" : "資料科學學程", "年級" : 1,
"學號" : "r10946001", "姓名" : "李奕宏" }
```

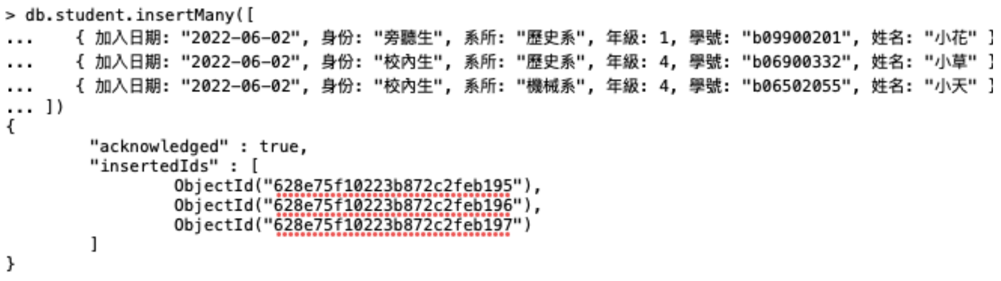
Part 1-3.

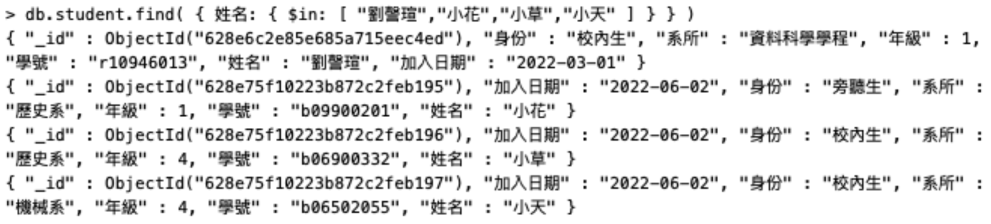

```
> db.student.aggregate( [ {  $group: {_id:"$系所", count:{ $sum:1 } } }, {  $sort: { count:-1, _id: 1 } } ])
{ "_id" : "電機系", "count" : 10 }
{ "_id" : "生機系", "count" : 6 }
{ "_id" : "資工系", "count" : 5 }
{ "_id" : "資管系", "count" : 4 }
{ "_id" : "工科海洋系", "count" : 3 }
{ "_id" : "生物機電系", "count" : 3 }
{ "_id" : "資料科學學程", "count" : 3 }
{ "_id" : "土木系水利組", "count" : 2 }
{ "_id" : "農藝系生統組", "count" : 2 }
{ "_id" : "電信所", "count" : 2 }
{ "_id" : "化學系", "count" : 1 }
{ "_id" : "土木系結構組", "count" : 1 }
{ "_id" : "地質系", "count" : 1 }
{ "_id" : "基蛋所", "count" : 1 }
{ "_id" : "心理系", "count" : 1 }
{ "_id" : "生工系", "count" : 1 }
{ "_id" : "生醫電資所", "count" : 1 }
{ "_id" : "經濟系", "count" : 1 }
{ "_id" : "財金系", "count" : 1 }
{ "_id" : "電機資安碩班", "count" : 1 }
>
```

Part 1-4.


```
> db.student.updateMany({}, {$set:{加入日期: "2022-03-01"}})
{ "acknowledged" : true, "matchedCount" : 50, "modifiedCount" : 50 }
> db.student.find( { 系所: "資料科學學程", 年級: { $eq: 1 } } )
{ "_id" : ObjectId("628e6c2e85e685a715eec4ed"), "身份" : "校內生", "系所" : "資料科學學程", "年級" : 1,
"學號" : "r10946013", "姓名" : "劉謦瑄", "加入日期" : "2022-03-01" }
{ "_id" : ObjectId("628e6c2e85e685a715eec4fb"), "身份" : "校內生", "系所" : "資料科學學程", "年級" : 1,
"學號" : "r10946001", "姓名" : "李奕宏", "加入日期" : "2022-03-01" }
>
```

Part 1-5.


```
> db.student.insertMany([
...     { 加入日期: "2022-06-02", 身份: "旁聽生", 系所: "歷史系", 年級: 1, 學號: "b09900201", 姓名: "小花" }
...     { 加入日期: "2022-06-02", 身份: "校內生", 系所: "歷史系", 年級: 4, 學號: "b06900332", 姓名: "小草" }
...     { 加入日期: "2022-06-02", 身份: "校內生", 系所: "機械系", 年級: 4, 學號: "b06502055", 姓名: "小天" }
... ])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("628e75f10223b872c2feb195"),
                ObjectId("628e75f10223b872c2feb196"),
                ObjectId("628e75f10223b872c2feb197")
        ]
}
```


```
> db.student.find( { 姓名: { $in: [ "劉謦瑄","小花","小草","小天" ] } } )
{ "_id" : ObjectId("628e6c2e85e685a715eec4ed"), "身份" : "校內生", "系所" : "資料科學學程", "年級" : 1,
"學號" : "r10946013", "姓名" : "劉謦瑄", "加入日期" : "2022-03-01" }
{ "_id" : ObjectId("628e75f10223b872c2feb195"), "加入日期" : "2022-06-02", "身份" : "旁聽生", "系所" :
"歷史系", "年級" : 1, "學號" : "b09900201", "姓名" : "小花" }
{ "_id" : ObjectId("628e75f10223b872c2feb196"), "加入日期" : "2022-06-02", "身份" : "校內生", "系所" :
"歷史系", "年級" : 4, "學號" : "b06900332", "姓名" : "小草" }
{ "_id" : ObjectId("628e75f10223b872c2feb197"), "加入日期" : "2022-06-02", "身份" : "校內生", "系所" :
"機械系", "年級" : 4, "學號" : "b06502055", "姓名" : "小天" }
```

Part 1-6.

Part 2-1.

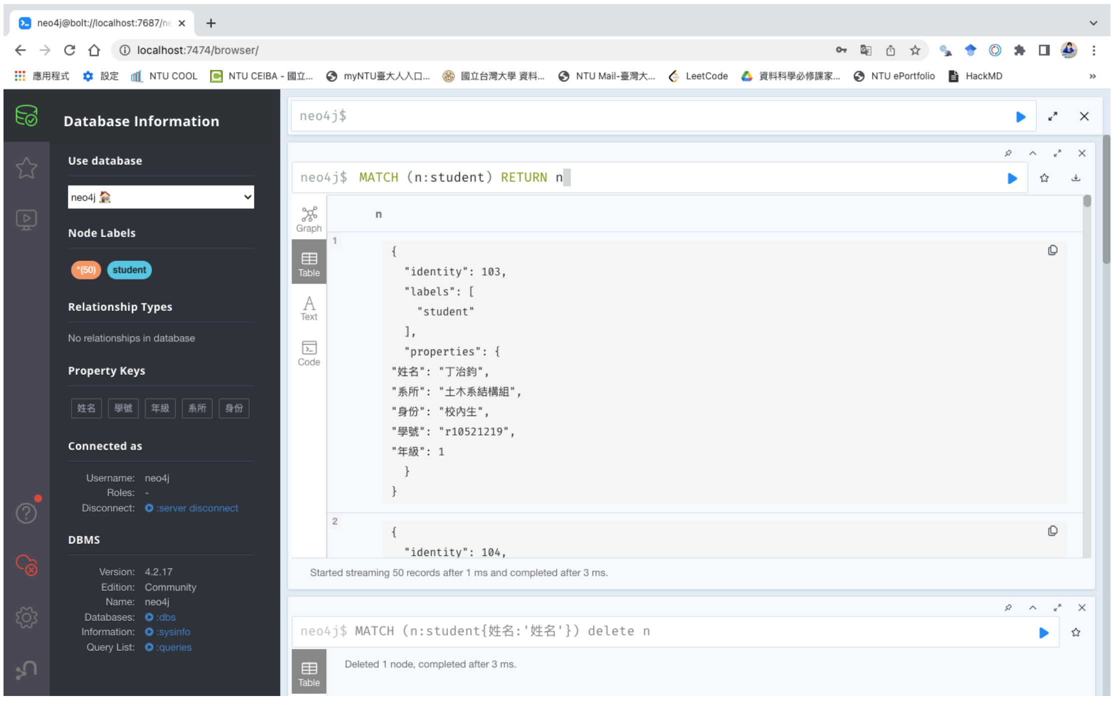
將 CSV 資料 LOAD 進資料庫後，建立一個 student 的 graph 資料表，並將表頭欄位刪除。

```
:use Neo4j

LOAD CSV FROM 'file:///hw6_student_list.csv' AS line
CREATE (:student {年級: toInteger(line[2]), 系所: line[1], 姓名: line[4], 學號: line[3], 身份: line[0]})

MATCH (n:student{姓名:'姓名'}) delete n

MATCH (n:student) RETURN n
```



Database Information

Use database: neo4j

Node Labels: *(50) student

Relationship Types: No relationships in database

Property Keys: 姓名 學號 年級 系所 身份

Connected as
Username: neo4j
Roles: -
Disconnect: :server disconnect

DBMS
Version: 4.2.17
Edition: Community
Name: neo4j
Databases: :dbs
Information: :sysinfo
Query List: :queries

```
neo4j$

neo4j$ MATCH (n:student) RETURN n

n
1
{
  "identity": 103,
  "labels": [
    "student"
  ],
  "properties": {
"姓名": "丁治鈞",
"系所": "土木系結構組",
"身份": "校內生",
"學號": "r10521219",
"年級": 1
  }
}
2
{
  "identity": 104,
```

Started streaming 50 records after 1 ms and completed after 3 ms.

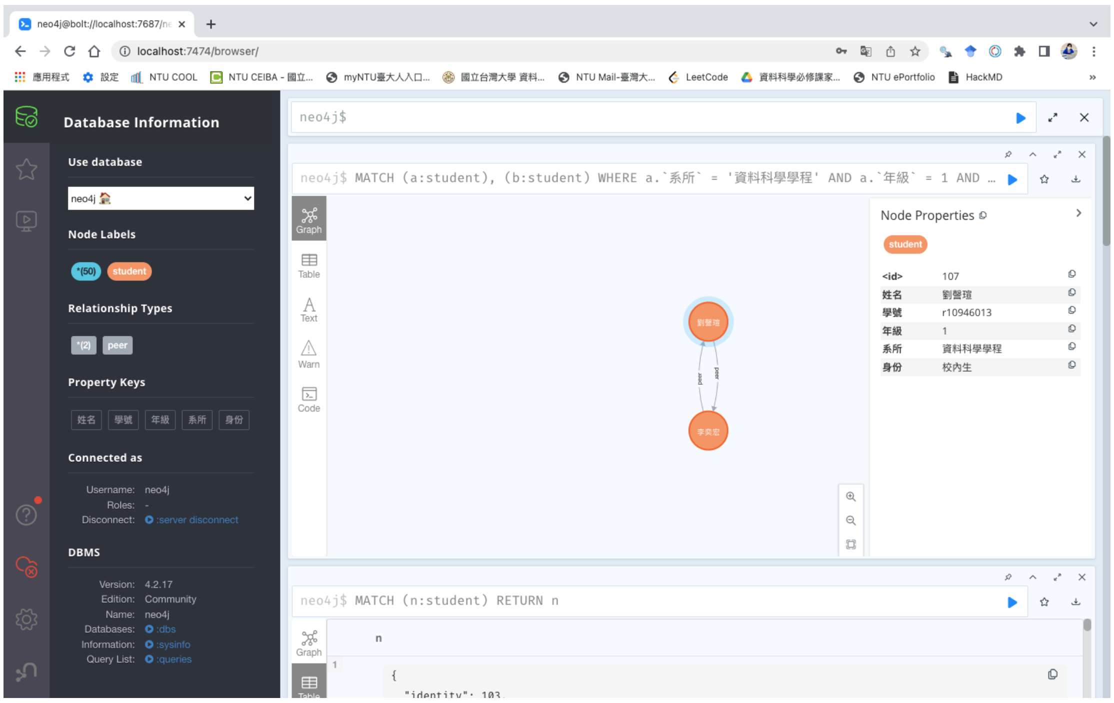
```
neo4j$ MATCH (n:student{姓名:'姓名'}) delete n
```

Deleted 1 node, completed after 3 ms.

Part 2-2.

透過指令找出年級系所相同的同學。

```
MATCH
  (a:student),
  (b:student)
WHERE a.`系所` = '資料科學學程' AND a.`年級` = 1 AND
      b.`系所` = '資料科學學程' AND b.`年級` = 1 AND
      a.`姓名` <> b.`姓名`
CREATE (a)-[r:peer]->(b)
RETURN b, type(r)
```



```
neo4j$ MATCH (a:student), (b:student) WHERE a.`系所` = '資料科學學程' AND a.`年級` = 1 AND …
```

Node Properties

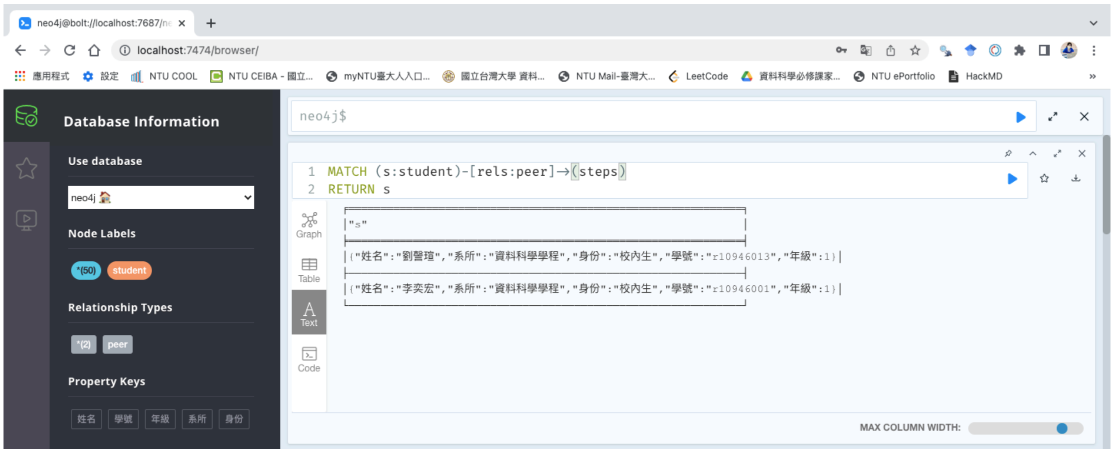
<id>	107
姓名	劉謦瑄
學號	r10946013
年級	1
系所	資料科學學程
身份	校內生

```
neo4j$ MATCH (n:student) RETURN n
```

```
{
  "identity": 103
```

Part 2-3.

```
MATCH (s:student)-[rels:peer]->(steps)
RETURN s
```

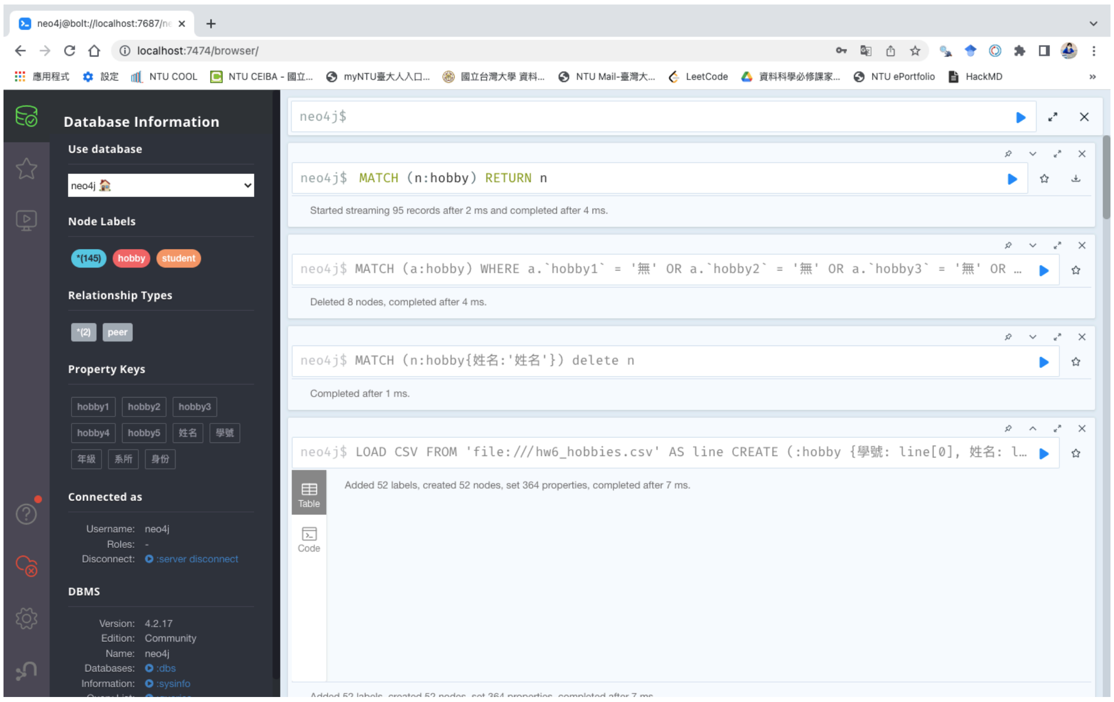


```
MATCH (s:student)-[rels:peer]→(steps)
RETURN s
```

```
╒══════════════════════════════════════════════════════════════╕
│"s"                                                           │
╞══════════════════════════════════════════════════════════════╡
│{"姓名":"劉謦瑄","系所":"資料科學學程","身份":"校內生","學號":"r10946013","年級":1}│
├──────────────────────────────────────────────────────────────┤
│{"姓名":"李奕宏","系所":"資料科學學程","身份":"校內生","學號":"r10946001","年級":1}│
└──────────────────────────────────────────────────────────────┘
```

Part 3-1.

- LOAD CSV (hobby.csv)



```
neo4j$ MATCH (n:hobby) RETURN n
```

Started streaming 95 records after 2 ms and completed after 4 ms.

```
neo4j$ MATCH (a:hobby) WHERE a.`hobby1` = '無' OR a.`hobby2` = '無' OR a.`hobby3` = '無' OR …
```

Deleted 8 nodes, completed after 4 ms.

```
neo4j$ MATCH (n:hobby{姓名:'姓名'}) delete n
```

Completed after 1 ms.

```
neo4j$ LOAD CSV FROM 'file:///hw6_hobbies.csv' AS line CREATE (:hobby {學號: line[0], 姓名: l…
```

Added 52 labels, created 52 nodes, set 364 properties, completed after 7 ms.

Database Information

Use database: neo4j

Node Labels: *(145) hobby student

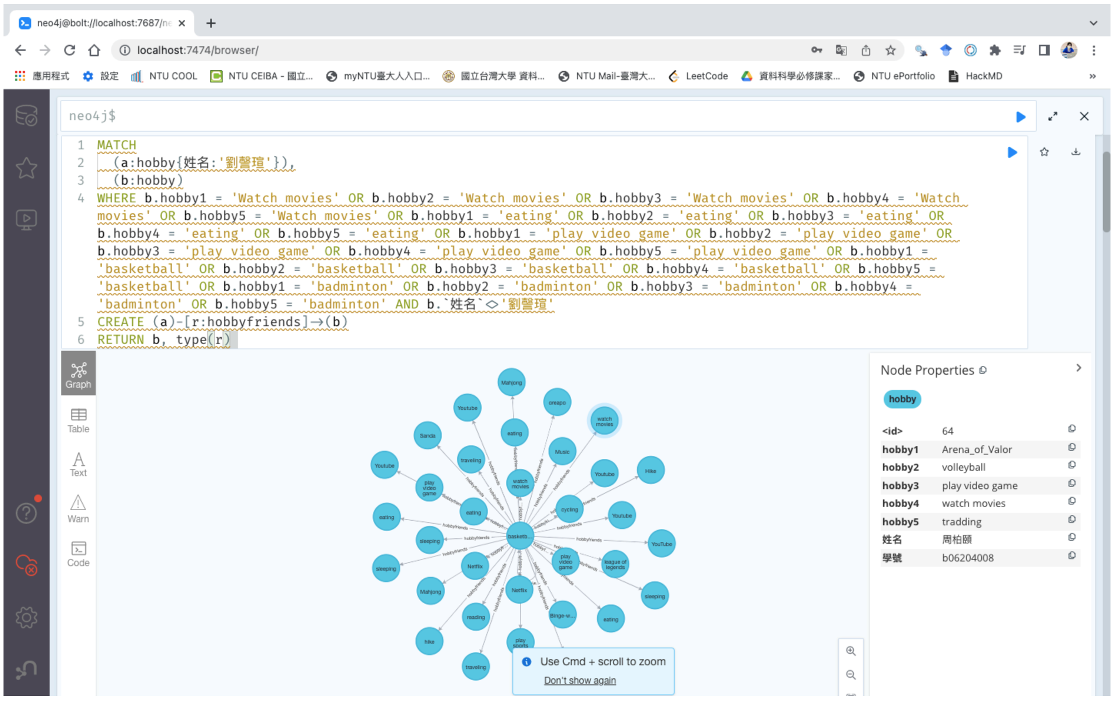
Relationship Types: *(2) peer

Property Keys: hobby1 hobby2 hobby3 hobby4 hobby5 姓名 學號 年級 系所 身份

Connected as
Username: neo4j
Roles: -
Disconnect: :server disconnect

DBMS
Version: 4.2.17
Edition: Community
Name: neo4j
Databases: :dbs
Information: :sysinfo

- Create "hobbyfriends" relationship



```
MATCH
  (a:hobby{姓名:'劉謦瑄'}),
  (b:hobby)
WHERE b.hobby1 = 'Watch movies' OR b.hobby2 = 'Watch movies' OR b.hobby3 = 'Watch movies' OR b.hobby4 = 'Watch movies' OR b.hobby5 = 'Watch movies' OR b.hobby1 = 'eating' OR b.hobby2 = 'eating' OR b.hobby3 = 'eating' OR b.hobby4 = 'eating' OR b.hobby5 = 'eating' OR b.hobby1 = 'play video game' OR b.hobby2 = 'play video game' OR b.hobby3 = 'play video game' OR b.hobby4 = 'play video game' OR b.hobby5 = 'play video game' OR b.hobby1 = 'basketball' OR b.hobby2 = 'basketball' OR b.hobby3 = 'basketball' OR b.hobby4 = 'basketball' OR b.hobby5 = 'basketball' OR b.hobby1 = 'badminton' OR b.hobby2 = 'badminton' OR b.hobby3 = 'badminton' OR b.hobby4 = 'badminton' OR b.hobby5 = 'badminton' AND b.`姓名`<>'劉謦瑄'
CREATE (a)-[r:hobbyfriends]->(b)
RETURN b, type(r)
```

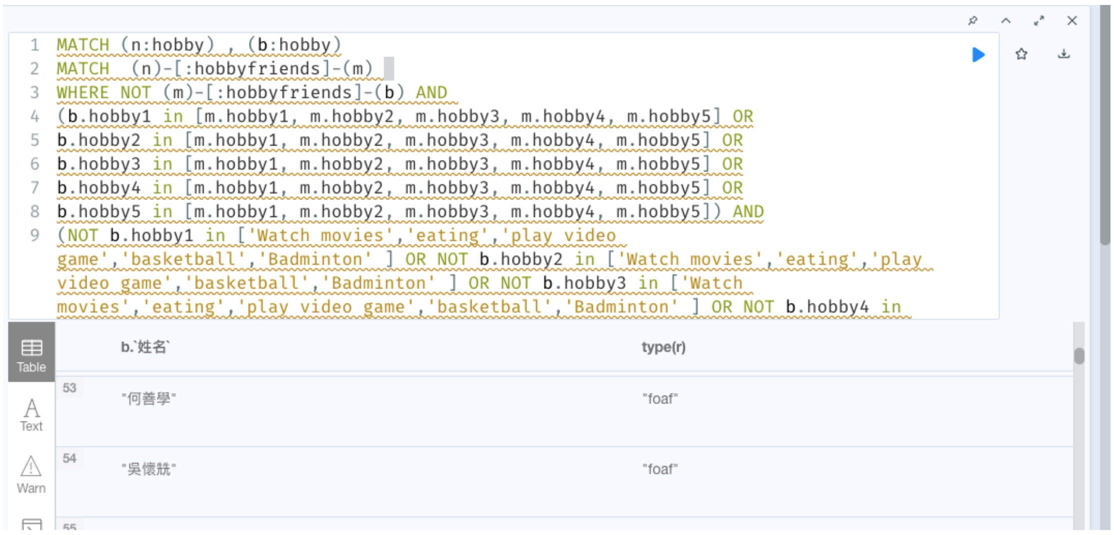
Node Properties

hobby

<id>	64
hobby1	Arena_of_Valor
hobby2	volleyball
hobby3	play video game
hobby4	watch movies
hobby5	tradding
姓名	周柏頤
學號	b06204008

Use Cmd + scroll to zoom
Don't show again

- Create “foaf” relationship

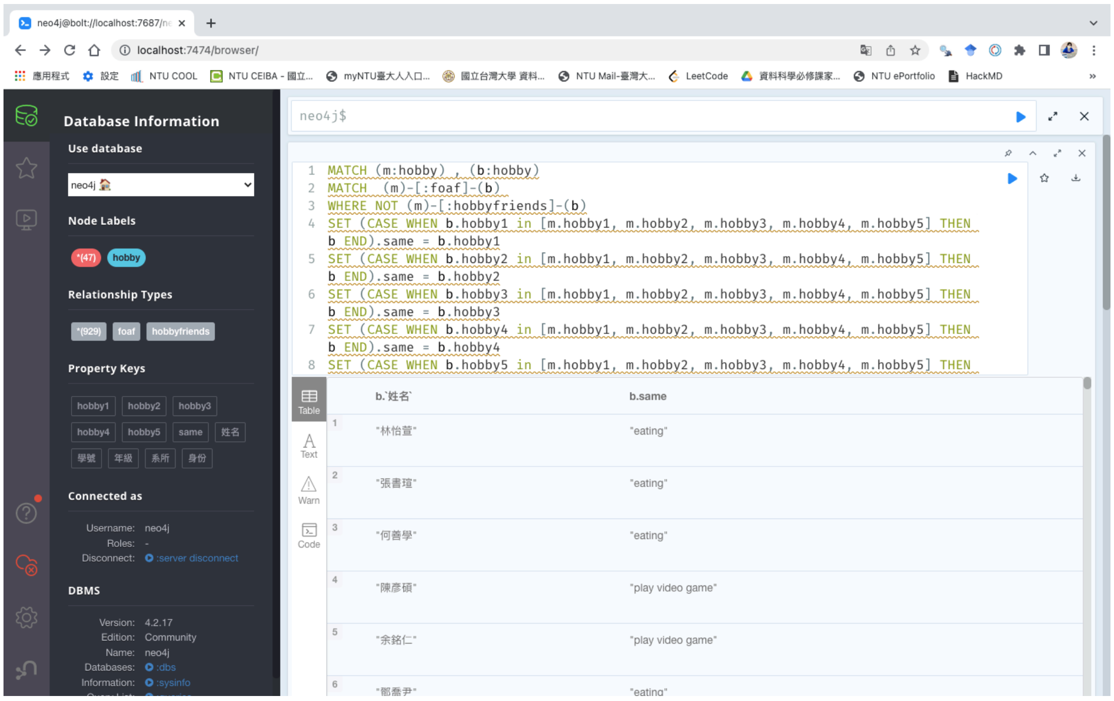


```
MATCH (n:hobby) , (b:hobby)
MATCH  (n)-[:hobbyfriends]-(m)
WHERE NOT (m)-[:hobbyfriends]-(b) AND
(b.hobby1 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] OR
b.hobby2 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] OR
b.hobby3 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] OR
b.hobby4 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] OR
b.hobby5 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5]) AND
(NOT b.hobby1 in ['Watch movies','eating','play video
game','basketball','Badminton' ] OR NOT b.hobby2 in ['Watch movies','eating','play
video game','basketball','Badminton' ] OR NOT b.hobby3 in ['Watch
movies','eating','play video game','basketball','Badminton' ] OR NOT b.hobby4 in
```

	b.`姓名`	type(r)
53	"何善學"	"foaf"
54	"吳懷兢"	"foaf"

Part 3-2.

透過指令將所有”foaf”關係的組合找出，並比對找出 hobby1 或 hobby2 或 hobby3 或 hobby4 或 hobby5 其中任一 hobby 與其相同者，將姓名與相同的 hobby 印出。



Connected as
Username: neo4j
Roles: -
Disconnect: :server disconnect

DBMS
Version: 4.2.17
Edition: Community
Name: neo4j
Databases: :dbs
Information: :sysinfo

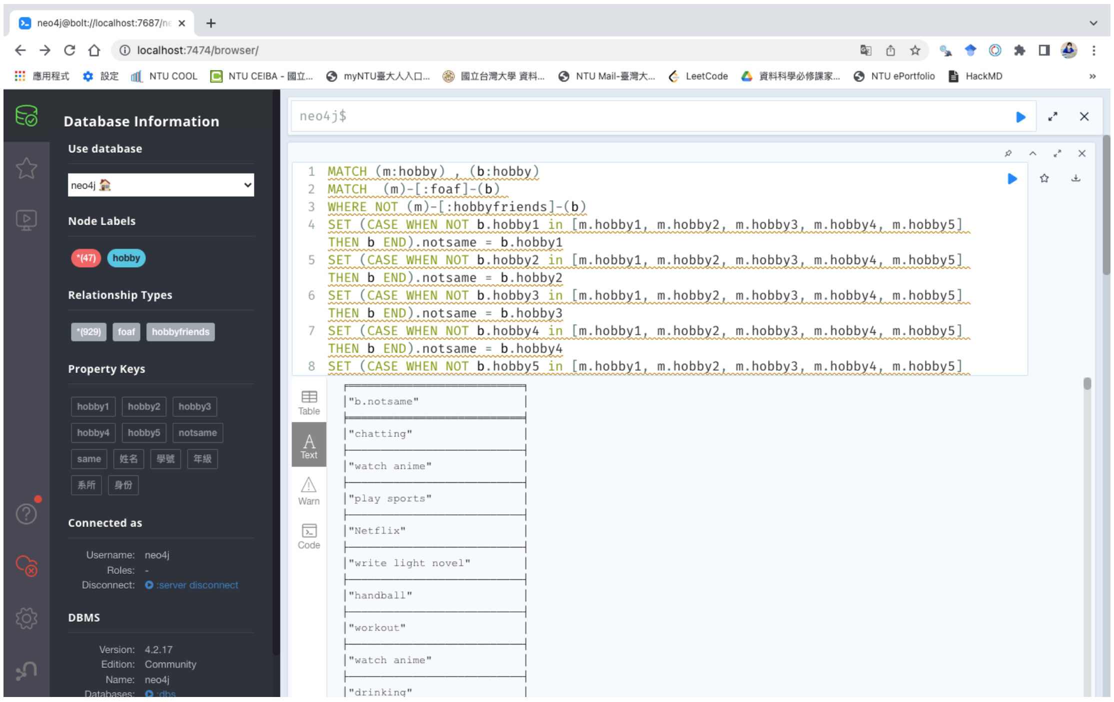
```
neo4j$
```

```
MATCH (m:hobby) , (b:hobby)
MATCH  (m)-[:foaf]-(b)
WHERE NOT (m)-[:hobbyfriends]-(b)
SET (CASE WHEN b.hobby1 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] THEN
b END).same = b.hobby1
SET (CASE WHEN b.hobby2 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] THEN
b END).same = b.hobby2
SET (CASE WHEN b.hobby3 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] THEN
b END).same = b.hobby3
SET (CASE WHEN b.hobby4 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] THEN
b END).same = b.hobby4
SET (CASE WHEN b.hobby5 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] THEN
```

	b.`姓名`	b.same
1	"林怡萱"	"eating"
2	"張書瑄"	"eating"
3	"何善學"	"eating"
4	"陳彥碩"	"play video game"
5	"余銘仁"	"play video game"
6	"鄒喬尹"	"eating"

Part 3-3.

透過指令將所有”foaf”關係的組合找出，並比對找出 hobby1 或 hobby2 或 hobby3 或 hobby4 或 hobby5 其中任一 hobby 與其相同者，將姓名與相同的 hobby 印出。



Username: neo4j
Roles: -
Disconnect: :server disconnect

Version: 4.2.17
Edition: Community
Name: neo4j
Databases: :dbs

```
neo4j$
```

```
MATCH (m:hobby) , (b:hobby)
MATCH  (m)-[:foaf]-(b)
WHERE NOT (m)-[:hobbyfriends]-(b)
SET (CASE WHEN NOT b.hobby1 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5]
THEN b END).notsame = b.hobby1
SET (CASE WHEN NOT b.hobby2 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5]
THEN b END).notsame = b.hobby2
SET (CASE WHEN NOT b.hobby3 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5]
THEN b END).notsame = b.hobby3
SET (CASE WHEN NOT b.hobby4 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5]
THEN b END).notsame = b.hobby4
SET (CASE WHEN NOT b.hobby5 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5]
```

```
╒══════════════════════╕
│"b.notsame"           │
╞══════════════════════╡
│"chatting"            │
├──────────────────────┤
│"watch anime"         │
├──────────────────────┤
│"play sports"         │
├──────────────────────┤
│"Netflix"             │
├──────────────────────┤
│"write light novel"   │
├──────────────────────┤
│"handball"            │
├──────────────────────┤
│"workout"             │
├──────────────────────┤
│"watch anime"         │
├──────────────────────┤
│"drinking"            │
```