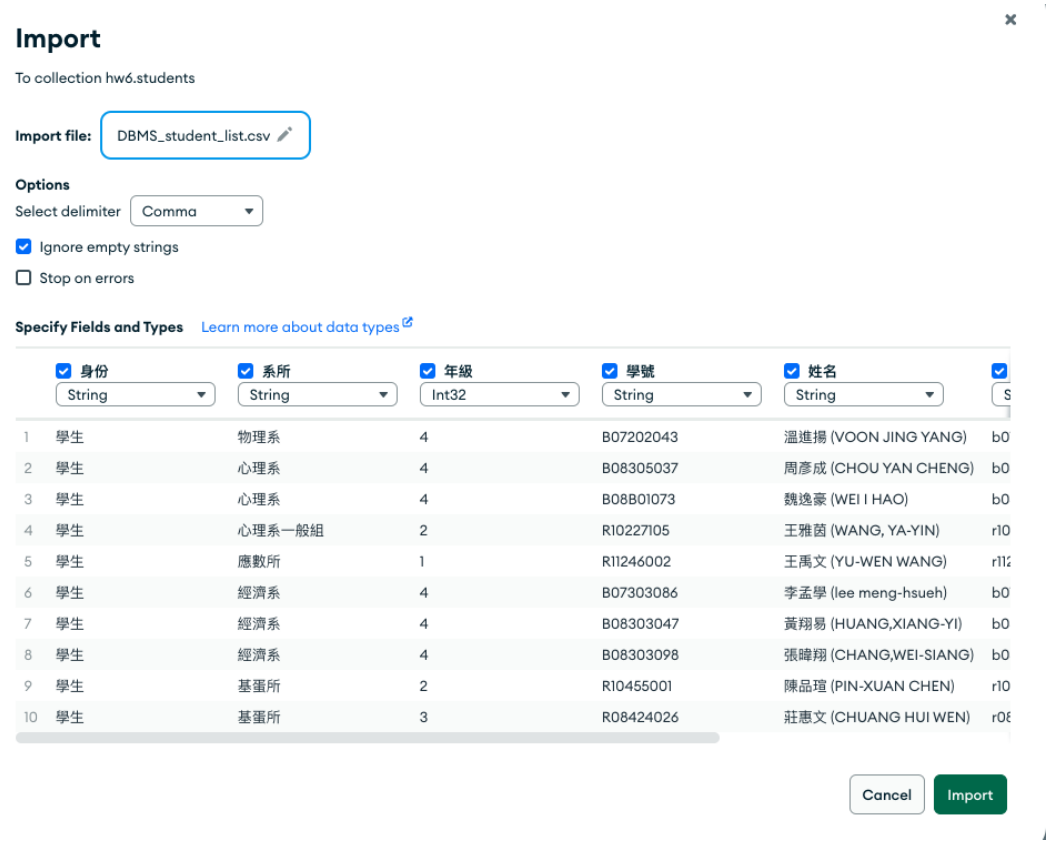
HW6 MongoDB Part1

Q1: Load the EE5178 student CSV file into collection “students” in database “hw6” in MongoDB, and write a MongoDB query to return the information (the document) about yourself.

因為我使用的是 MongoDB Compass，可直接 add data → import JSON or CSV file

**我在 hw6_mongo.txt 中寫的是用 mongoimport 讀取 CSV 的方式



Return myself

```
> use hw6
< switched to db hw6
> db.students.find({'學號': 'R10455001'})
< {
    _id: ObjectId("647ec6f4fc49808d381e3770"),
    '身份': '學生',
    '系所': '基蛋所',
    '年級': 2,
    '學號': 'R10455001',
    '姓名': '陳品瑄 (PIN-XUAN CHEN)',
    '信箱': 'r10455001@ntu.edu.tw',
    '班別': '資料庫系統-從SQL到NoSQL (EE5178)'
  }
```

Q2: Write a MongoDB query to return the information about you and the students in your group.

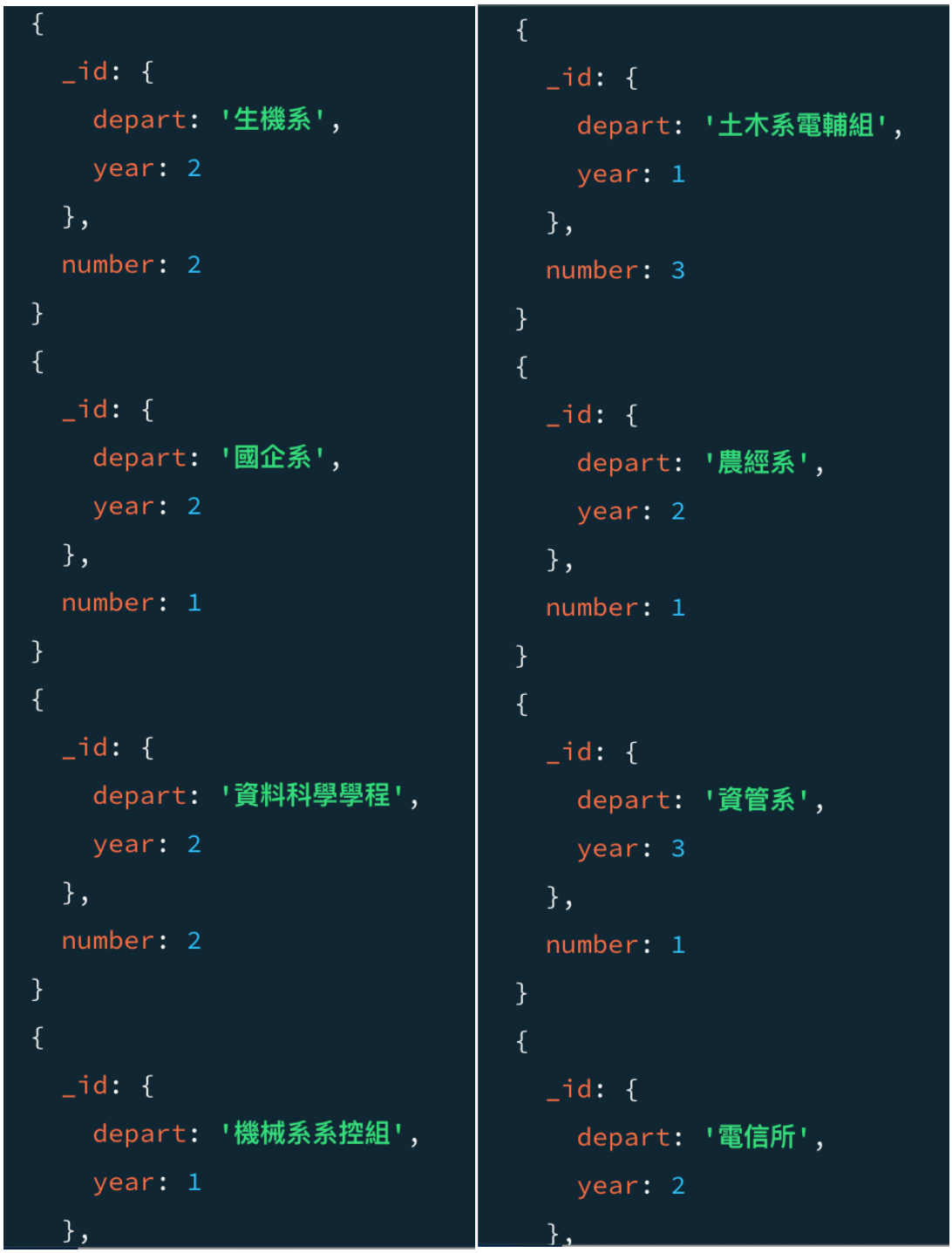
用學號找出我的組員

```
> db.students.find({'學號': {$in: [ 'R10455001', 'R08424026', 'R11631028', 'F08921A05', 'D06921008']}})
< {
    _id: ObjectId("647ec6f4fc49808d381e3770"),
    '身份': '學生',
    '系所': '基蛋所',
    '年級': 2,
    '學號': 'R10455001',
    '姓名': '陳品瑄 (PIN-XUAN CHEN)',
    '信箱': 'r10455001@ntu.edu.tw',
    '班別': '資料庫系統-從SQL到NoSQL (EE5178)'
  }
  {
    _id: ObjectId("647ec6f4fc49808d381e3771"),
    '身份': '學生',
    '系所': '基蛋所',
    '年級': 3,
    '學號': 'R08424026',
    '姓名': '莊惠文 (CHUANG HUI WEN)',
    '信箱': 'r08424026@ntu.edu.tw',
    '班別': '資料庫系統-從SQL到NoSQL (EE5178)'
  }
  {
    _id: ObjectId("647ec6f4fc49808d381e377c"),
    '身份': '學生',
    '系所': '生機系',
    '年級': 1,
    '學號': 'R11631028',
    '姓名': '劉易霖 (LIU, YIH-LIN)',
    '信箱': 'r11631028@ntu.edu.tw',
    '班別': '資料庫系統-從SQL到NoSQL (EE5178)'
  }
```

```
  {
    _id: ObjectId("647ec6f4fc49808d381e37a0"),
    '身份': '學生',
    '系所': '電機系',
    '年級': 3,
    '學號': 'F08921A05',
    '姓名': '陳憶賢 (YI-HSIEN CHEN)',
    '信箱': 'f08921a05@ntu.edu.tw',
    '班別': '資料庫系統-從SQL到NoSQL (EE5178)'
  }
  {
    _id: ObjectId("647ec6f4fc49808d381e37a1"),
    '身份': '學生',
    '系所': '電機系',
    '年級': 6,
    '學號': 'D06921008',
    '姓名': '黃尹姿 (HUANG YIN TZU)',
    '信箱': 'd06921008@ntu.edu.tw',
    '班別': '資料庫系統-從SQL到NoSQL (EE5178)'
  }
```

Q3: For each “系級" in the “students” collection, find out the number of students in it. Write a MongoDB query to return this information
用 aggregate 計算出不同系所和年級的人數，並以 number 表示人數印出
（只截取部分結果）

```
> db.students.aggregate([
    {
      $group: {
        _id: {"depart":"$系所",
              "year":"$年級"
             },
        number: {
          $sum: 1, },
      },
    },
  ])
< {
    _id: {
      depart: '工科海洋系',
      year: 3
    },
    number: 1
  }
  {
    _id: {
      depart: '應數所',
      year: 1
    },
    number: 1
  }
```



```
{
  _id: {
    depart: '生機系',
    year: 2
  },
  number: 2
}
{
  _id: {
    depart: '國企系',
    year: 2
  },
  number: 1
}
{
  _id: {
    depart: '資料科學學程',
    year: 2
  },
  number: 2
}
{
  _id: {
    depart: '機械系系控組',
    year: 1
  },
```

```
{
  _id: {
    depart: '土木系電輔組',
    year: 1
  },
  number: 3
}
{
  _id: {
    depart: '農經系',
    year: 2
  },
  number: 1
}
{
  _id: {
    depart: '資管系',
    year: 3
  },
  number: 1
}
{
  _id: {
    depart: '電信所',
    year: 2
  },
```

Q4: For the documents for each student, add a new field “join_date", and set the “2023-03-01”. Then return the information about you and your group members
使用 ISODate 插入加入日期，並印出我的組員確認

```
> db.students.updateMany({}, {$set: {join_date: ISODate("2023-03-01")}})
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 98,
    modifiedCount: 98,
    upsertedCount: 0
  }
```

```
> db.students.find({'學號': {$in: [ 'R10455001', 'R08424026', 'R11631028', 'F08921A05', 'D06921008']}})
< {
    _id: ObjectId("6489759a73b218ca7187eb02"),
    '身份': '學生',
    '系所': '基蛋所',
    '年級': 2,
    '學號': 'R10455001',
    '姓名': '陳品瑄 (PIN-XUAN CHEN)',
    '信箱': 'r10455001@ntu.edu.tw',
    '班別': '資料庫系統-從SQL到NoSQL (EE5178)',
    join_date: 2023-03-01T00:00:00.000Z
  }
  {
    _id: ObjectId("6489759a73b218ca7187eb03"),
    '身份': '學生',
    '系所': '基蛋所',
    '年級': 3,
    '學號': 'R08424026',
    '姓名': '莊惠文 (CHUANG HUI WEN)',
    '信箱': 'r08424026@ntu.edu.tw',
    '班別': '資料庫系統-從SQL到NoSQL (EE5178)',
    join_date: 2023-03-01T00:00:00.000Z
  }
```

```
{
  _id: ObjectId("6489759a73b218ca7187eb0e"),
  '身份': '學生',
  '系所': '生機系',
  '年級': 1,
  '學號': 'R11631028',
  '姓名': '劉易霖 (LIU, YIH-LIN)',
  '信箱': 'r11631028@ntu.edu.tw',
  '班別': '資料庫系統-從SQL到NoSQL (EE5178)',
  join_date: 2023-03-01T00:00:00.000Z
}
{
  _id: ObjectId("6489759a73b218ca7187eb32"),
  '身份': '學生',
  '系所': '電機系',
  '年級': 3,
  '學號': 'F08921A05',
  '姓名': '陳憶賢 (YI-HSIEN CHEN)',
  '信箱': 'f08921a05@ntu.edu.tw',
  '班別': '資料庫系統-從SQL到NoSQL (EE5178)',
  join_date: 2023-03-01T00:00:00.000Z
}
{
  _id: ObjectId("6489759a73b218ca7187eb33"),
  '身份': '學生',
  '系所': '電機系',
  '年級': 6,
  '學號': 'D06921008',
  '姓名': '黃尹姿 (HUANG YIN TZU)',
  '信箱': 'd06921008@ntu.edu.tw',
  '班別': '資料庫系統-從SQL到NoSQL (EE5178)',
  join_date: 2023-03-01T00:00:00.000Z
}
```


Q5: Add new students into your “students” collection, using the “new_student_list.csv” file provided. Write a query to return yourself and these new students

直接使用 InsertMany 插入三位新同學

```
> db.students.insertMany([
          {身份: '旁聽生', 系所: '電機所', 年級: 2, 學號: 'R10123456',姓名: '小紅', join_date: ISODate('2023-06-02')},
          {身份: '學生', 系所: '物理系', 年級: 3, 學號: 'B09987653',姓名: '小黃', join_date: ISODate('2023-06-02')},
          {身份: '觀察者', 系所: '電信所', 年級: 1, 學號: 'R11123001',姓名: '小綠', join_date: ISODate('2023-06-02')}
          ])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64897656b1b9c0d3a70fafec"),
    '1': ObjectId("64897656b1b9c0d3a70fafed"),
    '2': ObjectId("64897656b1b9c0d3a70fafee")
  }
}
```

```
> db.students.find({'學號': {$in: [ 'R10455001', 'R10123456','B09987653','R11123001']}})
< {
    _id: ObjectId("6489759a73b218ca7187eb02"),
    '身份': '學生',
    '系所': '基蛋所',
    '年級': 2,
    '學號': 'R10455001',
    '姓名': '陳品瑄 (PIN-XUAN CHEN)',
    '信箱': 'r10455001@ntu.edu.tw',
    '班別': '資料庫系統-從SQL到NoSQL (EE5178)',
    join_date: 2023-03-01T00:00:00.000Z
  }
  {
    _id: ObjectId("64897656b1b9c0d3a70fafec"),
    '身份': '旁聽生',
    '系所': '電機所',
    '年級': 2,
    '學號': 'R10123456',
    '姓名': '小紅',
    join_date: 2023-06-02T00:00:00.000Z
  }
  {
    _id: ObjectId("64897656b1b9c0d3a70fafed"),
    '身份': '學生',
    '系所': '物理系',
    '年級': 3,
    '學號': 'B09987653',
    '姓名': '小黃',
    join_date: 2023-06-02T00:00:00.000Z
  }
```

```
  {
    _id: ObjectId("64897656b1b9c0d3a70fafee"),
    '身份': '觀察者',
    '系所': '電信所',
    '年級': 1,
    '學號': 'R11123001',
    '姓名': '小綠',
    join_date: 2023-06-02T00:00:00.000Z
  }
```

Q6: Design a increment aggregation pipeline to calculate the number of student for each 系 級", and store your result in a “tally” document in your “students” collection. Run your query with date as to “2023-03-31” first. Print out the “tally” document. Then run your query again with date set to “2023-06-10”, and print out the “tally” document.

首先設定時間：2023-03-31

```
> var date = new ISODate("2023-03-31")
> db.students.aggregate([
    { $match: { join_date: { $lte: date } } },
    {
      $group: {
        _id: { "depart": "$系所",
               "year": "$年級"
             },
        number: {
          $sum: 1, },
      },
    },
    {
      $project: {
        _id: 0,
        depart: "$_id.depart",
        year: "$_id.year",
        number: 1
      },
    },
    { $out: "tally" }
  ])
```

印出來的結果：

```
> db.tally.find();
< {
    _id: ObjectId("648976ae1af1ac3ec90492a6"),
    number: 2,
    depart: '資工系',
    year: 1
  }
  {
    _id: ObjectId("648976ae1af1ac3ec90492a7"),
    number: 4,
    depart: '資管系',
    year: 1
  }
  {
    _id: ObjectId("648976ae1af1ac3ec90492a8"),
    number: 2,
    depart: '電機系',
    year: 2
  }
  {
    _id: ObjectId("648976ae1af1ac3ec90492a9"),
    number: 3,
    depart: '電信所',
    year: 1
  }
  {
    _id: ObjectId("648976ae1af1ac3ec90492aa"),
    number: 2,
    depart: '心理系',
    year: 4
  }
```

接著設定時間：2023-06-10

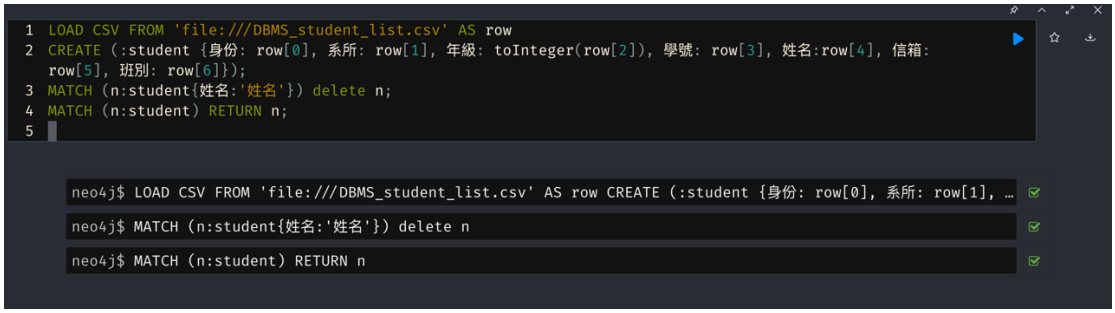
```
> var date = new ISODate("2023-06-10")
> db.students.aggregate([
    { $match: { join_date: { $lte: date } } },
    {
      $group: {
        _id: { "depart": "$系所",
               "year": "$年級"
             },
        number: {
          $sum: 1, },
      },
    },
    {
      $project: {
        _id: 0,
        depart: "$_id.depart",
        year: "$_id.year",
        number: 1
      },
    },
    { $out: "tally" }
  ])
<
> db.tally.find()
```

當把時間設定成 2023-06-10 之後，可以看到 tally document 有把新插入的三位學生算進去，電信所一年級人數由三人（參考上頁 2023-03-10 print console）變四人

```
{
  _id: ObjectId("6489f7f41af1ac3ec972a11a"),
  number: 4,
  depart: '電信所',
  year: 1
}
```

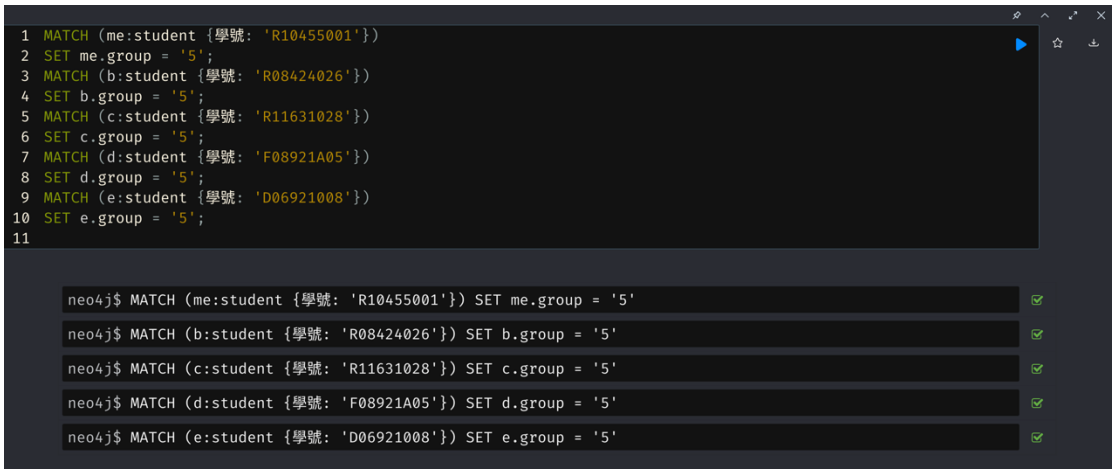
HW6 Neo4j Part2

Q1: Load the EE5178 student CSV into Neo4J graph database and create one node for each student.



Q2: Modify the database so that you and the students in your project group are recorded as in the same group

以學號搜尋並加入 group property



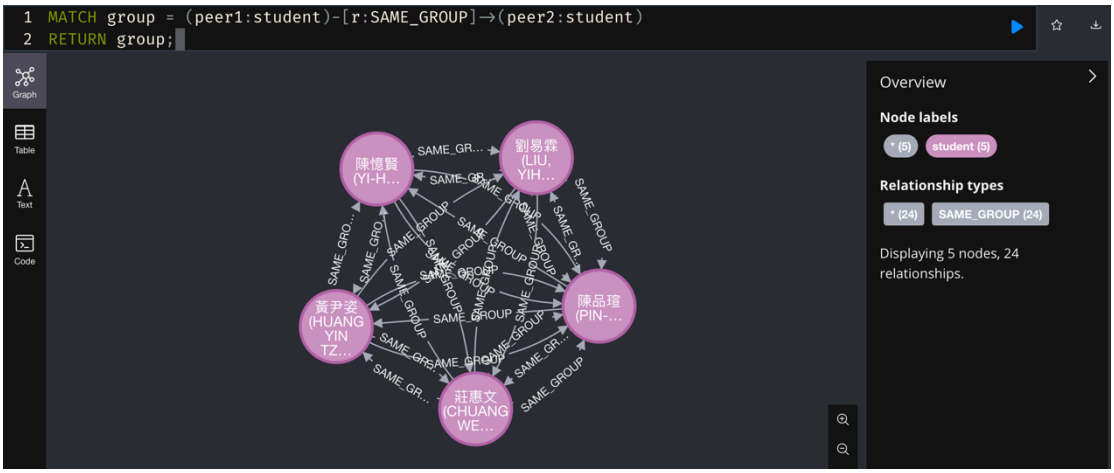
Q3: Write a Cypher query to return you, and students in your group in a list

使用 {group: '5'} match people and create relationship

```
MATCH (peer1:student),(peer2:student)
WHERE peer1.group = '5' AND peer2.group = '5' AND peer1.姓名 <> peer2.姓名
CREATE (peer1)-[r:SAME_GROUP]→(peer2);
```

Created 20 relationships, completed after 42 ms.

```
MATCH group = (peer1:student)-[r:SAME_GROUP]→(peer2:student)
RETURN group;
```



group
(:student {姓名: "莊惠文 (CHUANG HUI WEN)",身份: "學生",班別: "資料庫系統-從SQL到NoSQL (EE5178)",信箱: "r08424026@ntu.edu.tw",系所: "基蛋所",學號: "R08424026",年級: 3,group: "5"})-[:SAME_GROUP]->(:student {姓名: "陳品瑄 (PIN-XUAN CHEN)",身份: "學生",班別: "資料庫系統-從SQL到NoSQL (EE5178)",信箱: "r10455001@ntu.edu.tw",系所: "基蛋所",學號: "R10455001",年級: 2,group: "5"})
(:student {姓名: "劉易霖 (LIU, YIH-LIN)",身份: "學生",班別: "資料庫系統-從SQL到NoSQL (EE5178)",信箱: "r11631028@ntu.edu.tw",系所: "生機系",學號: "R11631028",年級: 1,group: "5"})-[:SAME_GROUP]->(:student {姓名: "陳品瑄 (PIN-XUAN CHEN)",身份: "學生",班別: "資料庫系統-從SQL到NoSQL (EE5178)",信箱: "r10455001@ntu.edu.tw",系所: "基蛋所",學號: "R10455001",年級: 2,group: "5"})
(:student {姓名: "陳憶賢 (YI-HSIEN CHEN)",身份: "學生",班別: "資料庫系統-從SQL到NoSQL (EE5178)",信箱: "f08921a05@ntu.edu.tw",系所: "電機系",學號: "F08921A05",年級: 3,group: "5"})-[:SAME_GROUP]->(:student {姓名: "陳品瑄 (PIN-XUAN CHEN)",身份: "學生",班別: "資料庫系統-從SQL到NoSQL (EE5178)",信箱: "r10455001@ntu.edu.tw",系所: "基蛋所",學號: "R10455001",年級: 2,group: "5"})
(:student {姓名: "黃尹姿 (HUANG YIN TZU)",身份: "學生",班別: "資料庫系統-從SQL到NoSQL (EE5178)",信箱: "d06921008@ntu.edu.tw",系所: "電機系",學號: "D06921008",年級: 6,group: "5"})-[:SAME_GROUP]->(:student {姓名: "陳品瑄 (PIN-XUAN CHEN)",身份: "學生",班別: "資料庫系統-從SQL到NoSQL (EE5178)",信箱: "r10455001@ntu.edu.tw",系所: "基蛋所",學號: "R10455001",年級: 2,group: "5"})
(:student {姓名: "陳品瑄 (PIN-XUAN CHEN)",身份: "學生",班別: "資料庫系統-從SQL到NoSQL (EE5178)",信箱: "r10455001@ntu.edu.tw",系所: "基蛋所",學號: "R10455001",年級: 2,group: "5"})-[:SAME_GROUP]->(:student {姓名: "莊惠文 (CHUANG HUI WEN)",身份: "學生",班別: "資料庫系統-從SQL到NoSQL (EE5178)",信箱: "r08424026@ntu.edu.tw",系所: "基蛋所",學號: "R08424026",年級: 3,group: "5"})

HW6 Neo4j Part3

Q1: Load the student hobbies CSV into the database and create the necessary nodes, relationships, and/or properties so that hobby information is recorded into the database

```
LOAD CSV FROM 'file:///hw6_hobbies.csv' AS row
CREATE (:hobby {學號: row[0], 姓名: row[1], hobby1: row[2], hobby2: row[3], hobby3: row[4], hobby4: row[5],
hobby5: row[6]});
MATCH (n:hobby{姓名:'姓名'}) delete n;
```

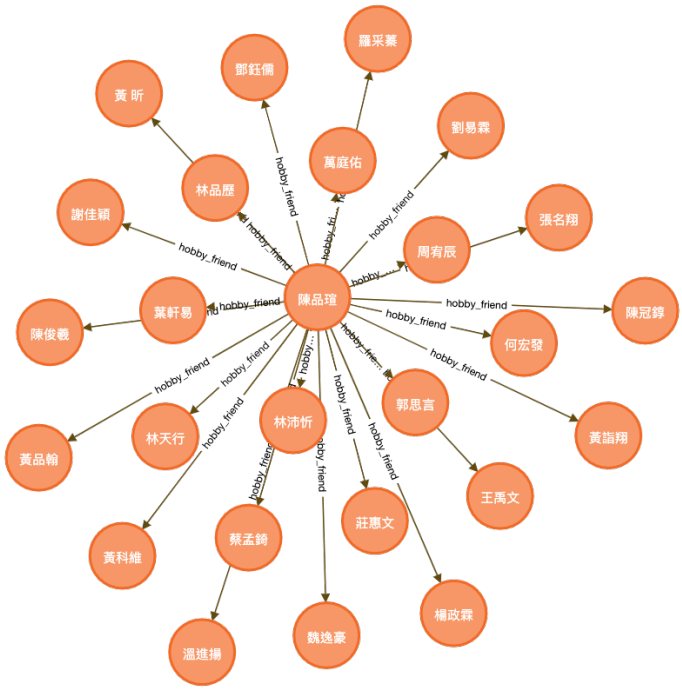
```
neo4j$ LOAD CSV FROM 'file:///hw6_hobbies.csv' AS row CREATE (:hobby {學號: row[0], 姓名: row[1], hobby1: row[2],…
neo4j$ MATCH (n:hobby{姓名:'姓名'}) delete n
```

Q2: Write a cypher query to print out your “hobby friends” and their associate new hobbies

他們的興趣至少要有一個跟我一樣，用 OR 篩選

```
MATCH (me:hobby {學號:'R10455001'})
MATCH (a:hobby)
WHERE (a.hobby1 in [me.hobby1, me.hobby2, me.hobby3, me.hobby4, me.hobby5] OR
      a.hobby2 in [me.hobby1, me.hobby2, me.hobby3, me.hobby4, me.hobby5] OR
      a.hobby3 in [me.hobby1, me.hobby2, me.hobby3, me.hobby4, me.hobby5] OR
      a.hobby4 in [me.hobby1, me.hobby2, me.hobby3, me.hobby4, me.hobby5] OR
      a.hobby5 in [me.hobby1, me.hobby2, me.hobby3, me.hobby4, me.hobby5]) AND
      a <> me
CREATE (me)-[rel:hobby_friend]→(a);

Created 53 relationships, completed after 26 ms.
```



```
MATCH (me:hobby {學號: 'R10455001'})-[rel:hobby_friend]→(a:hobby)
RETURN a.姓名 AS hobbyFriend, a.hobby1, a.hobby2, a.hobby3,
a.hobby4, a.hobby5;
```

Table | Text | Code

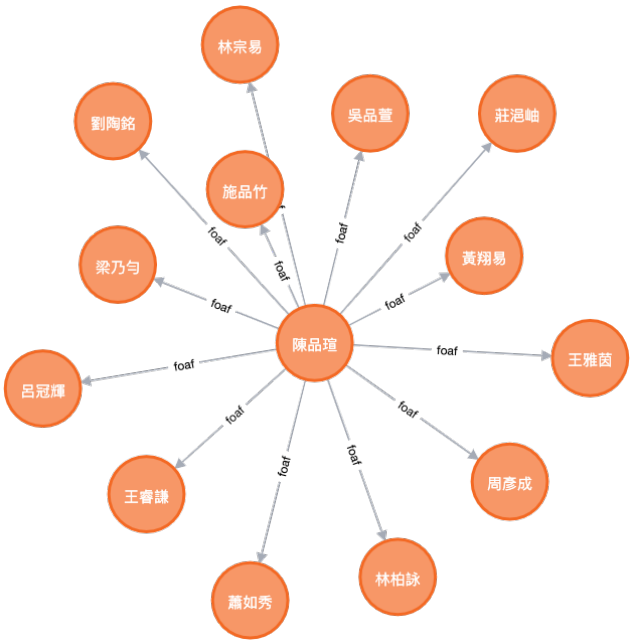
	hobbyFriend	a.hobby1	a.hobby2	a.hobby3	a.hobby4	a.hobby5
1	"林沛忻"	"movie"	"cat"	"travelling"	"volleyball"	"game"
2	"楊政霖"	"snooze"	"doze"	"rest"	"nap"	"sleeping"

Q3: find other students who have at least one common hobby with your hobby friends. Write a cypher query to print out your “foaf” and their associate new hobbies

Foaf 必須滿足三個條件：不是我的 hobby friend（也就是不在 hobby_friend relationship 中），至少有一個興趣跟我的 hobby friend 一樣 (OR 篩選), 以及他們的全部興趣都不能跟我一樣 (AND NOT 篩選)

```
MATCH (a:hobby)
MATCH (m:hobby)
MATCH (me: hobby {學號:'R10455001'})
MATCH (me)-[rel:hobby_friend]-(m)
WHERE NOT (me)-[rel:hobby_friend]-(a) AND
(a.hobby1 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] OR
 a.hobby2 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] OR
 a.hobby3 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] OR
 a.hobby4 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] OR
 a.hobby5 in [m.hobby1, m.hobby2, m.hobby3, m.hobby4, m.hobby5] ) AND
(NOT a.hobby1 in [me.hobby1, me.hobby2, me.hobby3, me.hobby4, me.hobby5] AND NOT
 a.hobby2 in [me.hobby1, me.hobby2, me.hobby3, me.hobby4, me.hobby5] AND NOT
```

Table | Created 13 relationships, completed after 91 ms.



```
MATCH (me)-[r:foaf]→(a)
RETURN a.姓名, a.hobby1, a.hobby2, a.hobby3, a.hobby4, a.hobby5;
```

	a.姓名	a.hobby1	a.hobby2	a.hobby3	a.hobby4	a.hobby5
1	"周彥成"	"basketball"	"movie"	"coffee"	"psychology"	"novel"
2	"王雅茵"	"anime"	"shopping"	"dessert"	"bl"	"coffee"
3	"黃翔易"	"comic"	"music"	"movie"	"art"	"anime"
4	"莊浥岫"	"billiard"	"camping"	"music"	"netflix"	"lego"
5	"吳品萱"	"gel-nail"	"music"	"k-pop"	"makeup"	"hiking"
6	"林宗易"	"piano"	"orchestra"	"cat"	"film"	"music"

Q4: the hobby friends of your group members can also become your friends. Let's call them "foaf2". Write a cypher query to print your "foaf2"

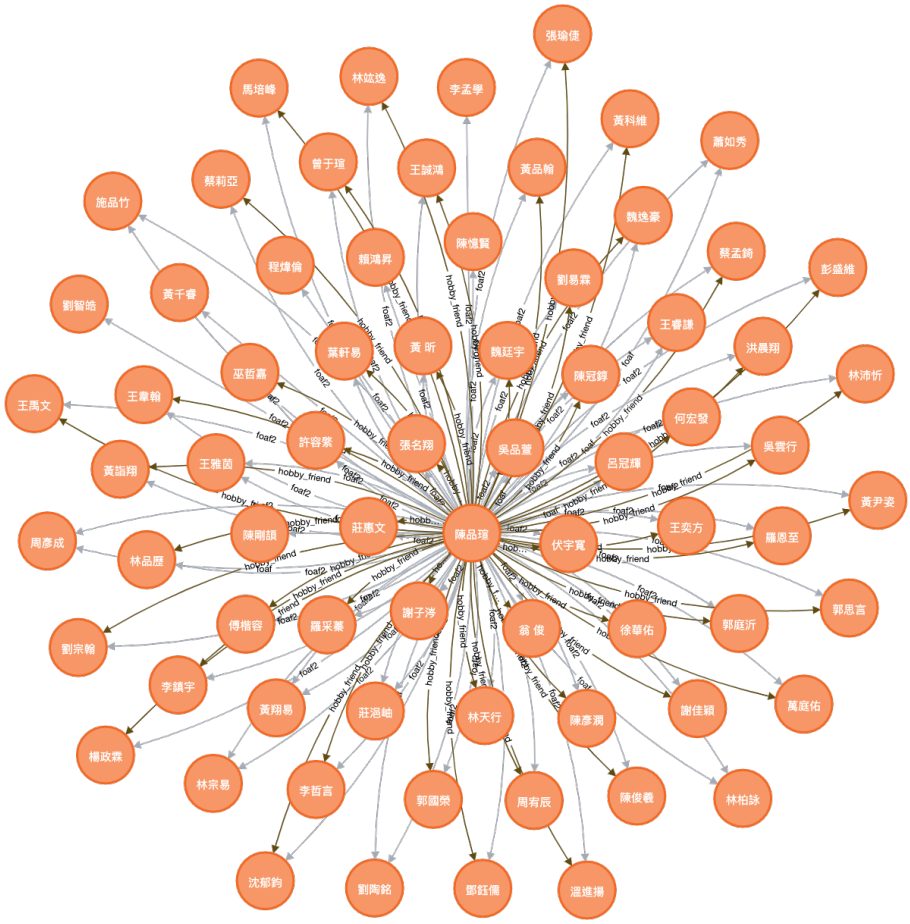
先加入 group property

```
MATCH (me:hobby {學號: 'R10455001'})
SET me.group = '5';
MATCH (b:hobby {學號: 'R08424026'})
SET b.group = '5';
MATCH (c:hobby {學號: 'R11631028'})
SET c.group = '5';
MATCH (d:hobby {學號: 'F08921A05'})
SET d.group = '5';
MATCH (e:hobby {學號: 'D06921008'})
SET e.group = '5';
```

找出和我的 group member 至少有一個興趣相同的 foaf2

```
MATCH (me:hobby {學號:'R10455001'})
MATCH (g:hobby {group:'5'})
MATCH (a:hobby)
WHERE (a.hobby1 in [g.hobby1, g.hobby2, g.hobby3, g.hobby4, g.hobby5] OR
      a.hobby2 in [g.hobby1, g.hobby2, g.hobby3, g.hobby4, g.hobby5] OR
      a.hobby3 in [g.hobby1, g.hobby2, g.hobby3, g.hobby4, g.hobby5] OR
      a.hobby4 in [g.hobby1, g.hobby2, g.hobby3, g.hobby4, g.hobby5] OR
      a.hobby5 in [g.hobby1, g.hobby2, g.hobby3, g.hobby4, g.hobby5]) AND
      a <> me AND g <> me
MERGE (me)-[relation:foaf2]→(a);
```

Created 72 relationships, completed after 42 ms.



Q5: Write a Cypher query to list all your “foaf2” excluding “foaf”