1. Cho cơ sở dữ liệu Graph Neo4J với mô tả như sau:

* Tập các node person với các thuộc tính name (tên), born (năm sinh), SSN (căn cước công dân)
* Tập các node verhicle (phươn tiện giao thông) với các thuộc tính type (loại), brand (hiệu xe), verhicleID (số xe)
* Các mối quan hệ có thể có giữa các node
  + own: giữa 1 node person và 1 node verhicle, có thuộc tính from, to
  + drive: giữa 1 node person và node verhicle, có thuộc tính from, to
  + with: giữa 1 node person và 1 node person, có thuộc tính is

Viết các lệnh sau:

1. Tạo 1 node thuộc tập person có nodename = P123345 với thuộc tính name = Nguyen Tan, born:1964

CREATE (P123345:Person {name: 'Nguyen Tan', born: 1964, SSN: 'P123345'})

1. Tạo 1 node thuộc tập person có nodename = P123346 với thuộc tính name = Nguyen Tien, born:2000

CREATE (P123346:Person {name: 'Nguyen Tien', born: 2000, SSN: 'P123346'})

1. Tạo 1 node thuộc tập verhicle có nodename = V1234 với thuộc tính type = car, brand = toyota wigo 2023, verhicleID= 121324

CREATE (V1234:Vehicle {type: 'car', brand: 'toyota wigo 2023', vehicleID: '121324'})

1. Tạo mối quan hệ own giữa node P123345 và node V1234 với thuộc tính from = 2023, to = 2023

MATCH (P123345:Person {SSN: 'P123345'}), (V1234:Vehicle {vehicleID: '121324'})

CREATE (P123345)-[:OWN {from: 2023, to: 2023}]->(V1234)

1. Tạo mối quan hệ drive giữa node P123346 và node V1234 với thuộc tính from = 2023, to = 2023

MATCH (P123346:Person {SSN: 'P123346'}), (V1234:Vehicle {vehicleID: '121324'})

CREATE (P123346)-[:DRIVE {from: 2023, to: 2023}]->(V1234)

1. Tạo mối quan hệ with giữa node P123346 và node P123346 với thuộc tính is = father

MATCH (P123346:Person {SSN: 'P123346'}), (P123345:Person {SSN: 'P123345'})

CREATE (P123346)-[:WITH {is: 'father'}]->(P123345)

1. Tìm tên chủ xe, tên người lái xe toyota wigo 2023

MATCH (owner:Person)-[:OWN]->(v:Vehicle {brand: 'toyota wigo 2023'}), (driver:Person)-[:DRIVE]->(v)

RETURN owner.name AS OwnerName, driver.name AS DriverName

1. Tìm tổng số xe **mà** Nguyen Tan sở hữu

MATCH (p:Person {name: 'Nguyen Tan'})-[:OWN]->(v:Vehicle)

RETURN COUNT(v) AS TotalVehicles

1. Cho Mongodb có tên SM với 1 collection Persons với các document có thể có các thuộc tính ID, FirstName , Lname, Address (gồm street, city, state, zip), hobbies, income
2. Tìm tên những người có thu nhập trên 100

db.Persons.find(

{ income: { $gt: 100 } },

{ FirstName: 1, \_id: 0 }

)

1. Tìm người có họ là Nguyen và tên là Lan

db.Persons.findOne(

{ FirstName: "Lan", Lname: "Nguyen" }

)

1. Tìm thu nhập trung bình của tất cả người

db.Persons.aggregate([

{ $group: { \_id: null, averageIncome: { $avg: "$income" } } }

])

1. Tìm thu nhập trung bình của những người ở TPHCM

db.Persons.aggregate([

{ $match: { "Address.city": "TPHCM" } },

{ $group: { \_id: null, averageIncome: { $avg: "$income" } } }

])

1. Tìm thu nhập trung bình của từng thành phố.

db.Persons.aggregate([

{ $group: { \_id: "$Address.city", averageIncome: { $avg: "$income" } } }

])

1. Cho database Redis với dữ liệu có mẫu như sau

|  |  |
| --- | --- |
| Key (Mã tài sản: 2 số đầu là loại tài sản, các ký tự kế tiếp là số seri, 2 số cuối cùng là năm mua) | Values  (tên tài sản, đơn vị sử dụng, giá mua) |
| LTASUSFX50621 | Laptop ASUS TUF gamingF15 , Phòng đồ họa, 20.999.000 |
| … | … |

1. Chọn kiểu dữ liệu phù hợp
2. Thêm 1 key-values (mã tài sản và các thông tin của tài sản: tên, đơn vị sử dụng, giá mua) vào database

HSET LTASUSFX50722 "name" "Laptop ASUS ROG Strix G17" "unit" "Phòng máy chủ" "price" "35.999.000"

1. Tìm tài sản có 2 ký tự đầu là LT

SCAN 0 MATCH LT\* COUNT 100

1. Tìm tài sản có năm mua là 2019(19)

SCAN 0 MATCH \*19 COUNT 100