Tên: Mai Chiến Nô

Mssv: 21106521

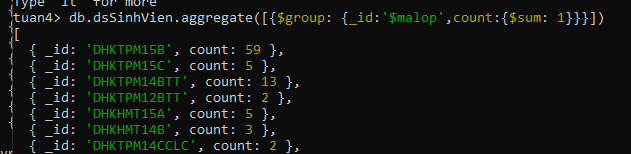
**TUẦN 4-5-6-7: AGGREGATION OPERATIONS**

**Bài 1. Collection Sinhvien**

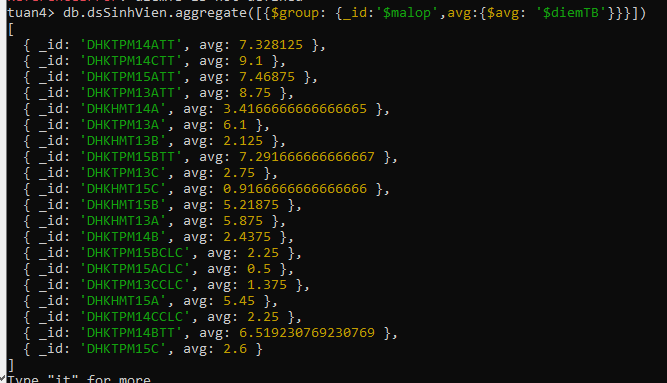
**1. Đếm số sinh sinh của mỗi lớp, thông tin gồm mã lớp và số sinh viên.**



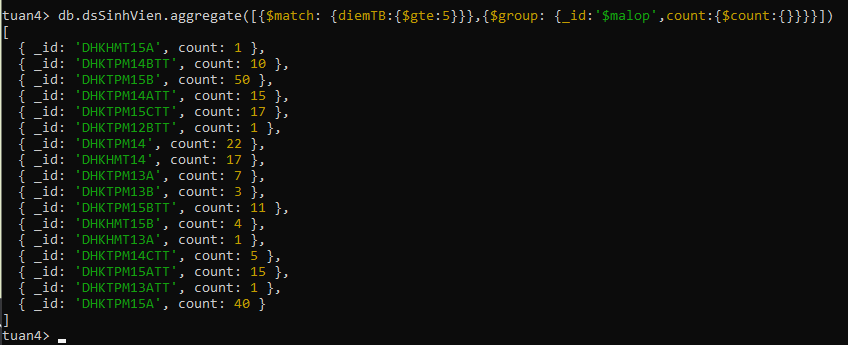
**Hoặc cách 2:**

****

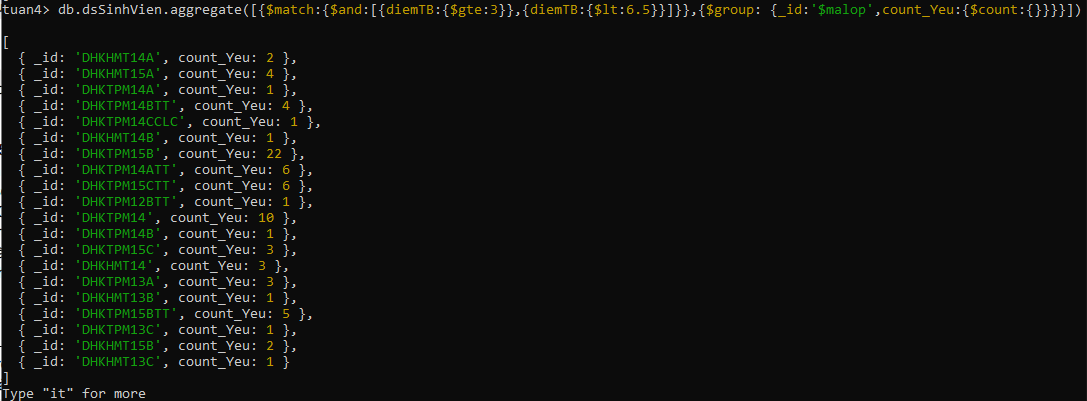
**2. Tính điểm trung bình của từng lớp, thông tin gồm mã lớp và điểm trung bình.**

****

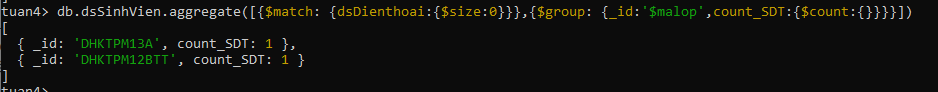
**3. Đếm số sinh viên có điểm trung bình từ 5 trở lên của từng lớp, thông tin gồm mã lớp, số sinh viên.**

****

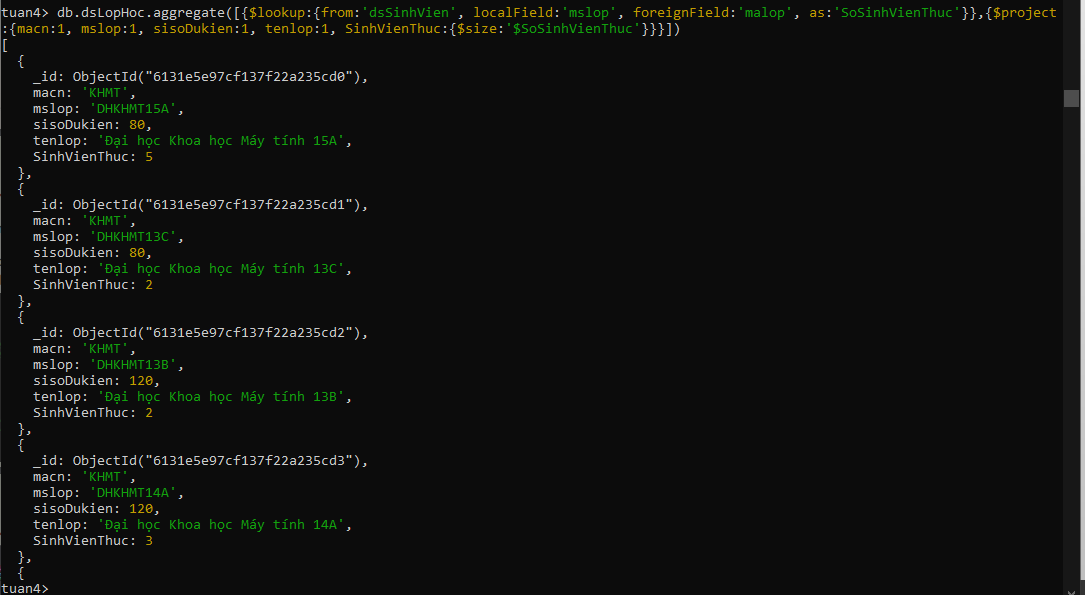
**4. Đếm số sinh viên có điểm trung bình từ 3.0 đến dưới 6.5 của từng lớp, thông tin gồm mã lớp, số sinh viên.**

****

**5. Đếm số sinh viên không có số điện thoại của từng lớp, thông tin gồm mã lớp, số sinh viên không điện thoại.**

****

**6. Đếm số sinh viên thực tế theo từng lớp, thông tin bao gồm thông tin của lớp học và tổng số sinh viên thực. (gồm 2 collection)**

****

**7. Tìm lớp học có tổng số sinh viên thực tế cao nhất. Thông tin bao gồm thông tin của lớp học và tổng số sinh viên. [ { \_id: 'DHKTPM15B', soluong: 59 } ]**

****

**8. Đếm số sinh viên có điểm trung bình từ 9.0 trở lên theo từng lớp, sắp xếp theo tên sinh viên. Thông tin bao gồm thông tin của lớp học và tổng số sinh viên.**

**Text

Description automatically generated**

**9. Xuất danh sách sinh viên có điểm trung bình từ 9.0 trở lên theo từng lớp, sắp xếp theo tên sinh viên ra collection riêng biệt. Thông tin bao gồm thông tin của sinh viên.**

**Text

Description automatically generated**

**10. Đếm số lớp theo chuyên ngành. Thông tin gồm mã chuyên ngành, số lớp**

**Graphical user interface, text

Description automatically generated**

**11. Tính tổng số sinh viên theo từng chuyên ngành. Thông tin bao gồm thông tin của chuyên ngành và tổng số sinh viên.**

Text

Description automatically generated

**Bài 2. Collection customers**

1. Xuất danh sách khách hàng với các thông tin address, city, state.

Text

Description automatically generated

2. Xuất các khách hàng có city ở “Woburn” và state là “MA”.

Text

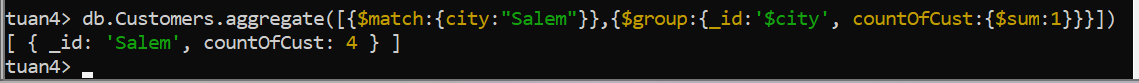
Description automatically generated

3. Xuất thông tin khách hàng bao gồm individual, first\_name, last\_name của officer.

Text

Description automatically generated

4. Nhóm danh sách khách hàng theo city và cho biết số lượng khách hàng tương ứng với điều kiện là các khách hàng có city ở “Salem”.



5. Xuất danh sách 5 khách hàng đầu tiên được sắp xếp theo thứ tự tăng dần của postal\_code và giảm dần của fed\_id.

Text

Description automatically generated

6. Lọc các khách hàng có state là “MA”. Sau đó, thực hiện nhóm các khách hàng vừa tìm được theo city và đếm số lượng khách hàng tương ứng với danh sách city vừa nhóm. Tiếp theo thực hiện xuất 03 dữ liệu cuối cùng được sắp xếp kết quả giảm dần theo postal\_code.

db.Customers.aggregate([{$match:{state:"MA"}},{$group:{\_id:'$city', countOfCust:{$sum:1}}},{$sort:{postal\_code:-1}},{$skip:3},{$limit:3}])

Text

Description automatically generated

7. Xuất danh sách khách hàng có tồn tại field business.

Text

Description automatically generated

8. Xuất danh sách khách hàng có tồn tại field business với các thông tin fed\_id, cust\_type\_cd, state, khuyenMai. Trong đó field khuyenMai được tính như sau: nếu field state = “MA” thì sẽ được hưởng khuyến mãi (khuyenMai: true), ngược lại sẽ không được hưởng khuyến mãi (khuyenMai: false). KM:{ $cond: { if: {state:”MA”}, then: “giam 10%”, else: “khong km” } }

db.Customers.aggregate([{$project:{fed\_id:1, cust\_type\_cd:1, state:1, KM: {$cond: {if:{$eq:['$state','MA']}, then: "true", else: "false"}}}}])

Text

Description automatically generated

9. Nếu field khuyenMai là true (câu 8) thì sẽ hiển thị là “Được giảm 10%”, ngược lại sẽ hiển thị “Không được hưởng khuyến mãi”.

db.Customers.aggregate([{$project:{fed\_id:1, cust\_type\_cd:1, state:1, KM: {$cond: {if:{$eq:['$state','MA']}, then: "true", else: "false"}}}}, {$project:{fed\_id:1, cust\_type\_cd:1, state:1, KM: {$cond: {if:{$eq:['$KM','true']}, then: "Duoc giam 10%", else: "Khong duoc huong khuyen mai"}}}}])

Text

Description automatically generated

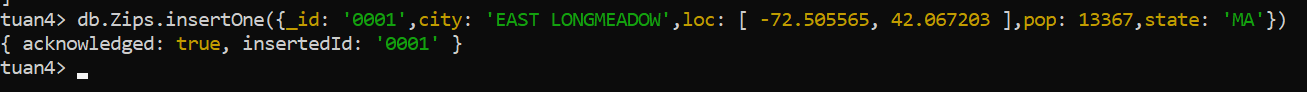
**Bài 3. Collection zips**

1. Hiển thị n documents từ document thứ k. (n, k tùy ý)

Text

Description automatically generated

2. Chèn thêm 1 document mới. (tùy ý)

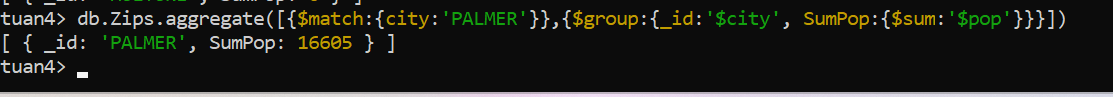


3. Cập nhật thông tin của một document khi biết id bất kỳ.

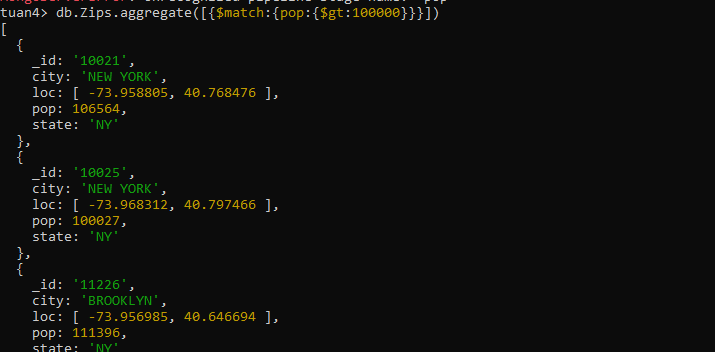
Text

Description automatically generated

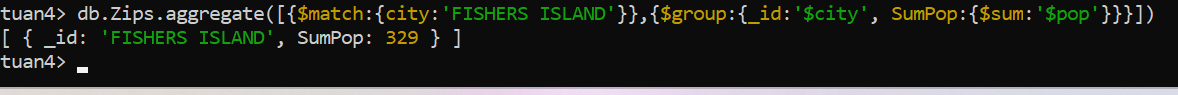
4. Tìm dân số của thành phố (city) PALMER.



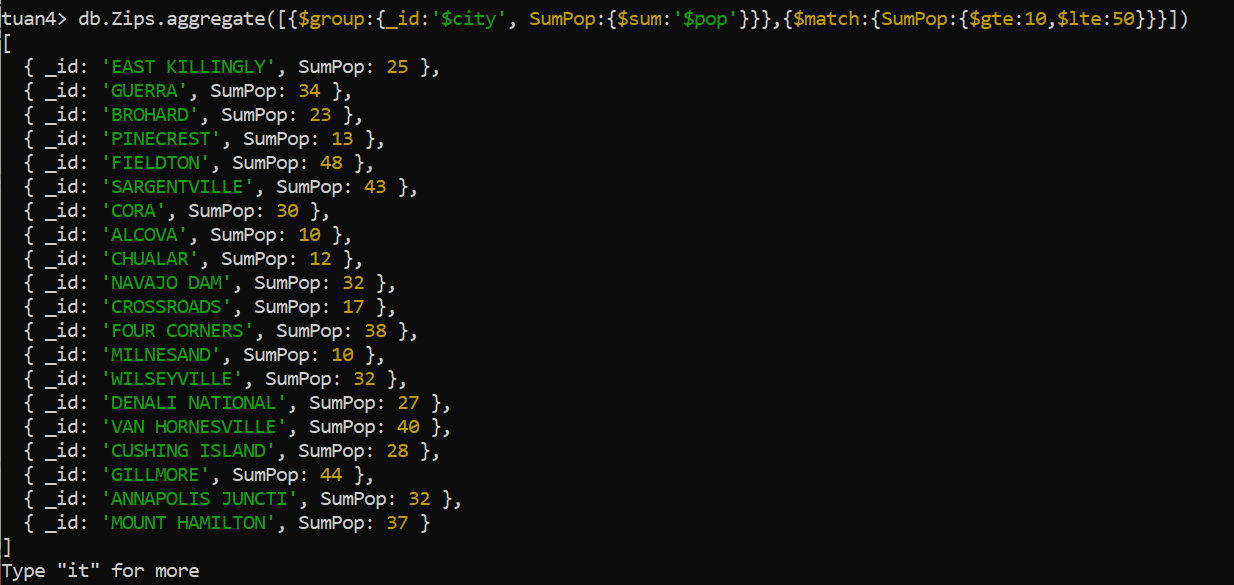
5. Tìm các document có dân số (pop) >100000



6. Tìm dân số của thành phố (city) FISHERS ISLAND



7. Tìm các thành phố có dân số từ 10 – 50



8. Tìm tất cả các thành phố của bang MA có dân số trên 500

Text

Description automatically generated

9. Tìm tất cả các bang (không trùng)

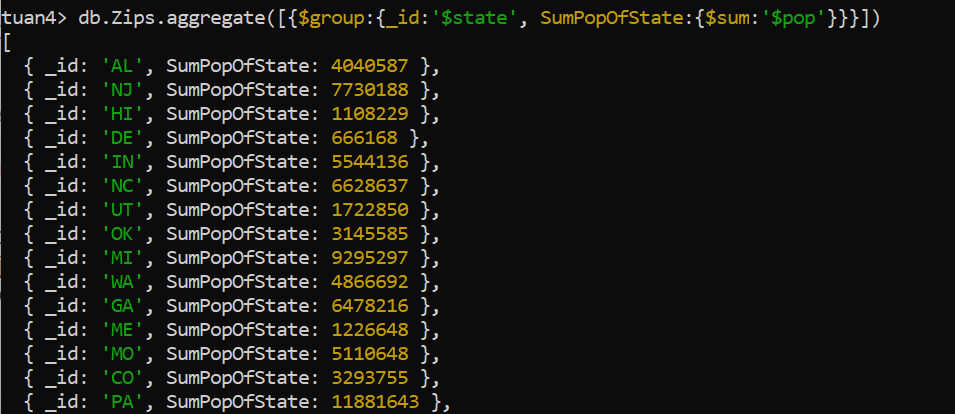
Text

Description automatically generated

10.Tìm tất cả các bang mà có chứa ít nhất 1 thành phố có dân số trên 100000



11.Tính tổng số dân (pop) theo từng bang (state).

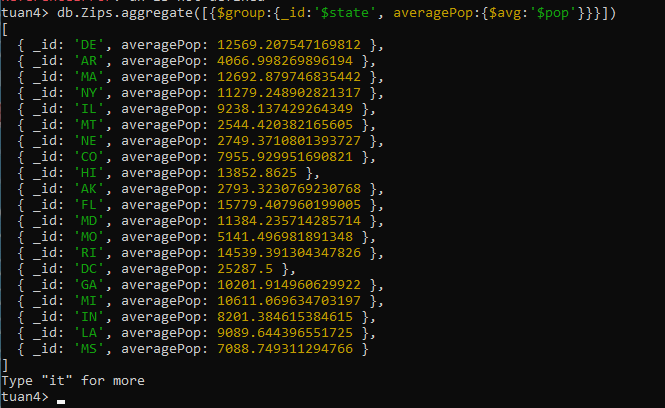


12.Tìm tất cả các bang có tổng dân số trên 10.000.000

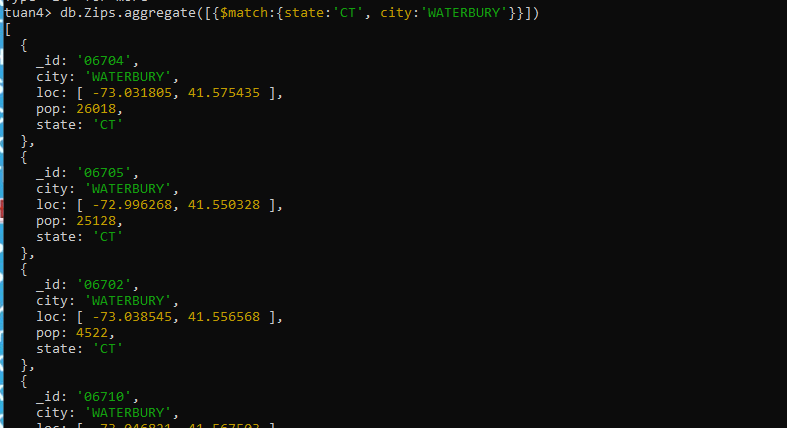
Text

Description automatically generated

13.Tính dân số trung bình (các thành phố) theo từng bang (state).



14.Tìm những document của bang 'CT' và thành phố 'WATERBURY'



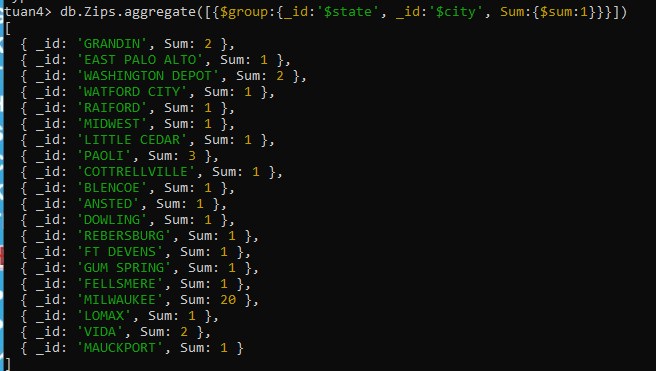
15.Bang WA có bao nhiêu city (nếu trùng chỉ tính 1 lần)



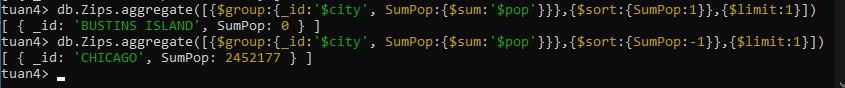
16.Tính số city của mỗi bang (nếu trùng chỉ tính 1 lần), kết quả giảm dần theo số city

17.Group theo state va city, sum:1; Group state, sum

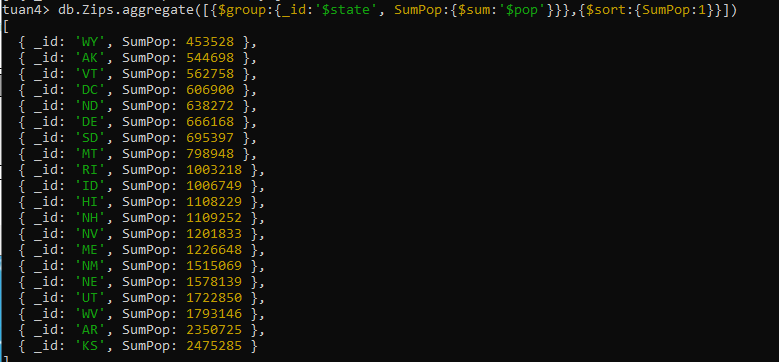
(Thành phố cùng tên ở nhiều bang xác định)



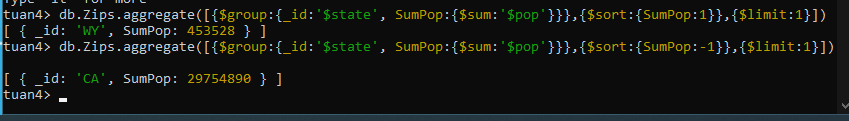
18.Tìm ra các thành phố có dân số (pop) lớn (nhỏ) nhất.



19.Sort theo pop, $groupΛ)



20.Tìm bang có dân số (pop) lớn (nhỏ) nhất.



21.Xuất những document có dân số dưới dân số trung bình của mỗi city