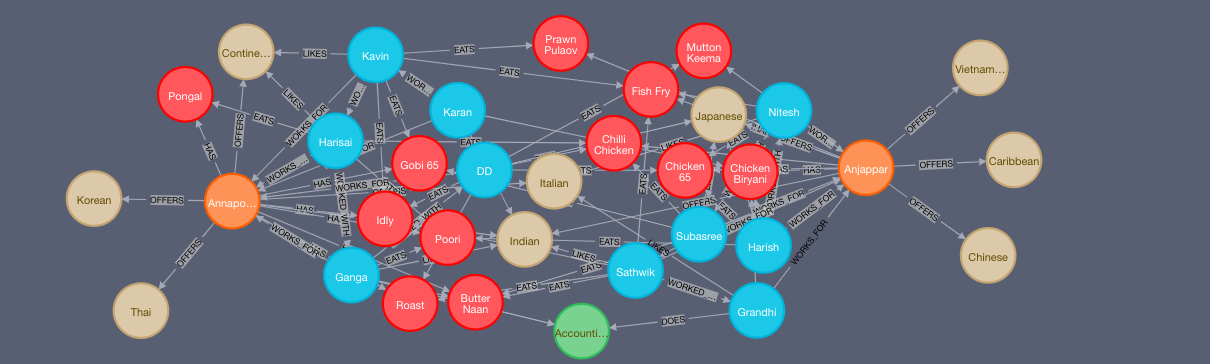
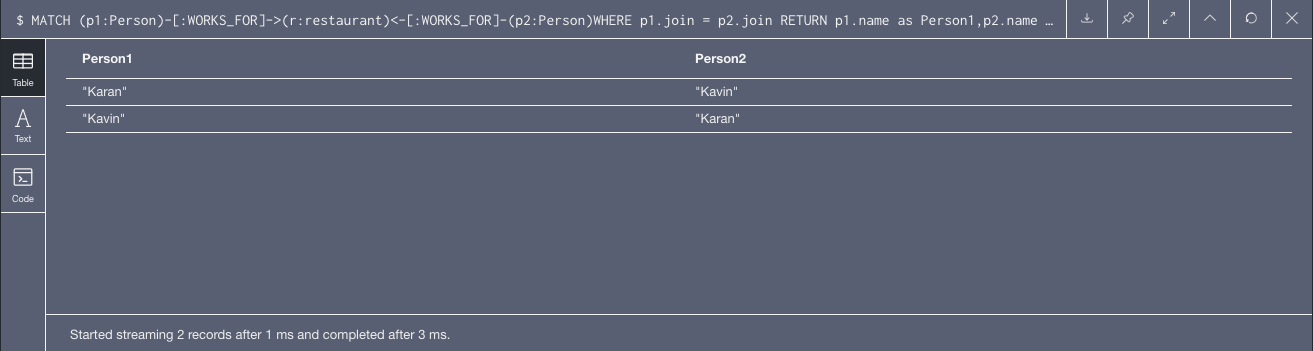
Bon Appétit!



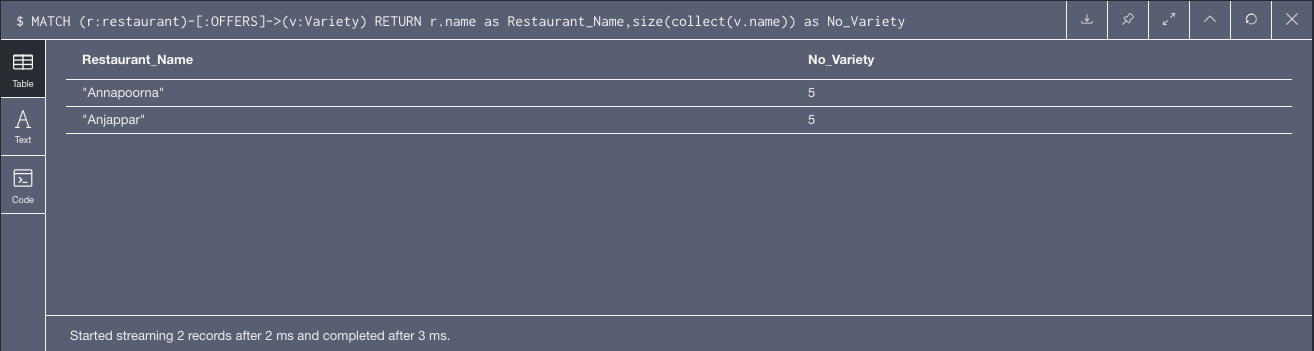
There were two restaurants - a vegetarian and a non vegetarian. Each had their own set of staffs and separate menus. Each restaurant offered five different varieties of cuisine. The customers were free to choose and dine at their preferred restaurant between the two.

**1. Display the name of persons who work in the same restaurant and their year of joining is the same.**

MATCH (p1:Person)-[:WORKS\_FOR]->(r:restaurant)<-[:WORKS\_FOR]-(p2:Person)WHERE p1.join = p2.join RETURN p1.name as Person1,p2.name as Person2

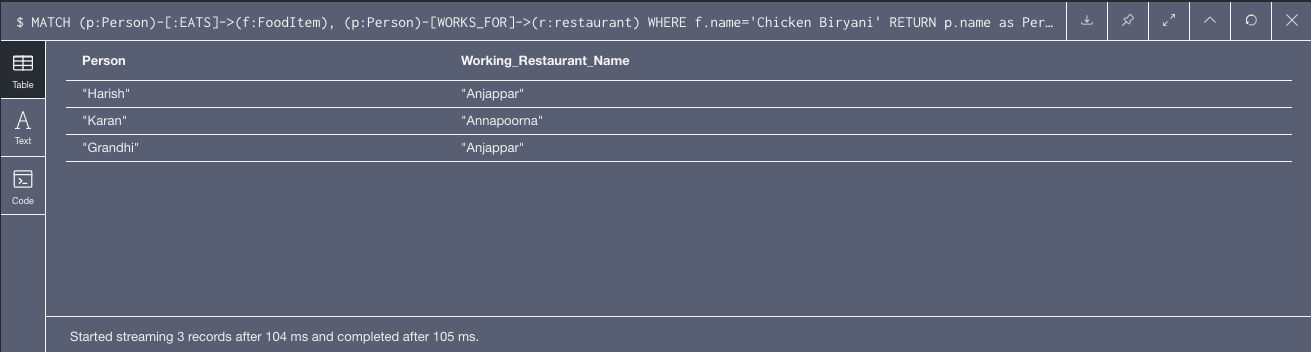
**2. Display the restaurants along with the number of varieties they offer.**

MATCH (r:restaurant)-[:OFFERS]->(v:Variety) RETURN r.name as Restaurant\_Name,size(collect(v.name)) as No\_Variety



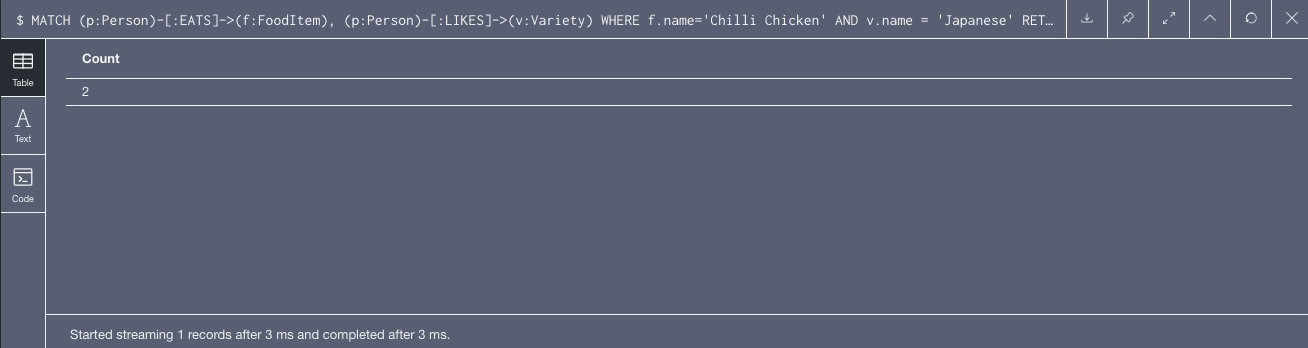
**3. Display the name of persons who eat chicken biryani and the name of the restaurant they work in.**

MATCH (p:Person)-[:EATS]->(f:FoodItem), (p:Person)-[:WORKS\_FOR]->(r:restaurant) WHERE f.name='Chicken Biryani' RETURN p.name as Person, r.name as Working\_Restaurant\_Name



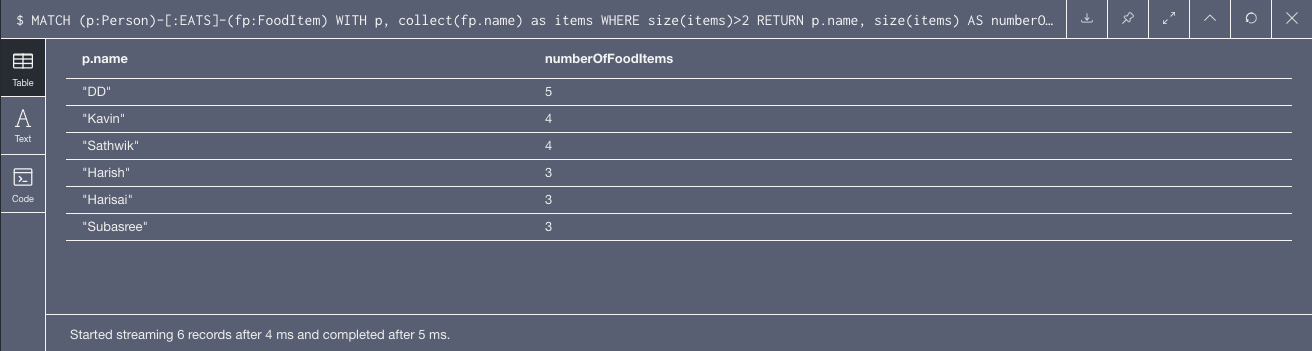
**4. Display the number of people who like Japanese cuisine but eat chilli chicken.**

MATCH (p:Person)-[:EATS]->(f:FoodItem), (p:Person)-[:LIKES]->(v:Variety) WHERE f.name='Chilli Chicken' AND v.name = 'Japanese' RETURN count(\*) as Count

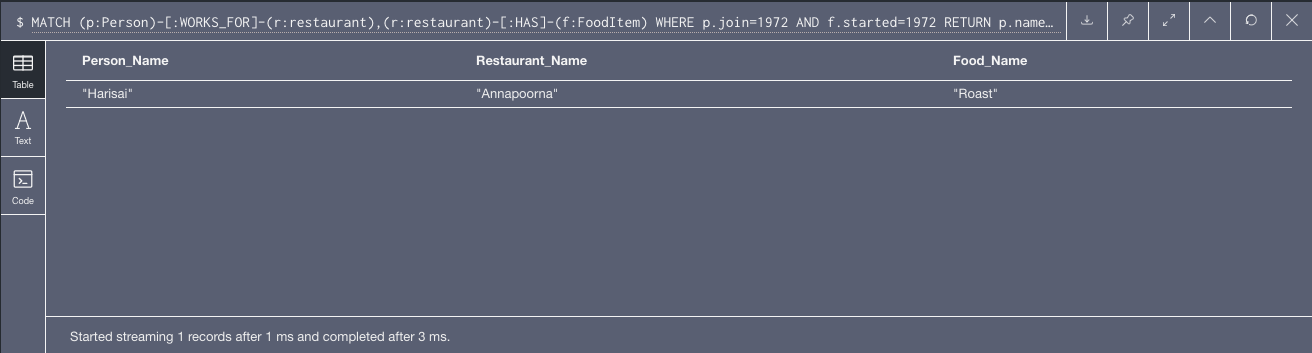


**5. Display the name of people who eat more than two food items, along with their count in the descending order.**

MATCH (p:Person)-[:EATS]-(fp:FoodItem) WITH p, collect(fp.name) as items WHERE size(items)>2 RETURN p.name, size(items) AS numberOfFoodItems ORDER BY numberOfFoodItems DESC

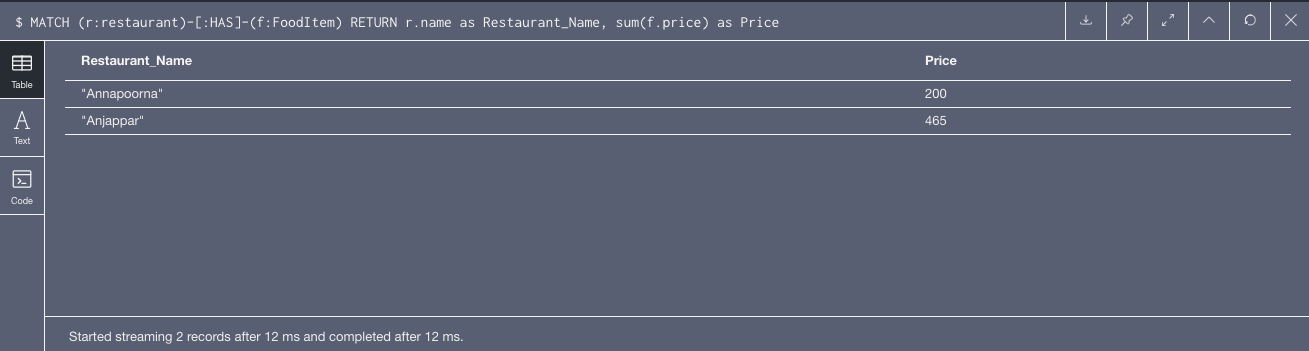


**6. Display the details of people/food items which started in 1972.**

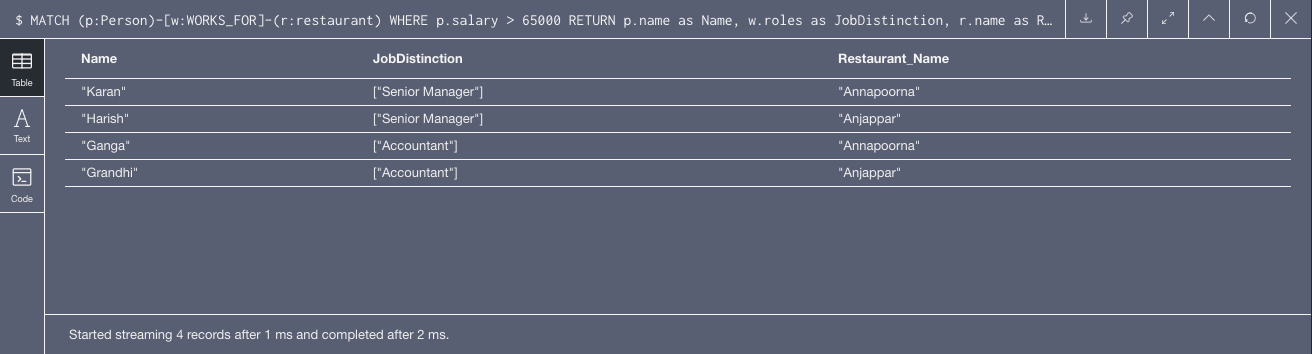
MATCH (p:Person)-[:WORKS\_FOR]-(r:restaurant),(r:restaurant)-[:HAS]-(f:FoodItem) WHERE p.join=1972 AND f.started=1972 RETURN p.name as Person\_Name ,r.name as Restaurant\_Name, f.name as Food\_Name

**7. Display the cumulative price of all the food Items in both restaurants.**

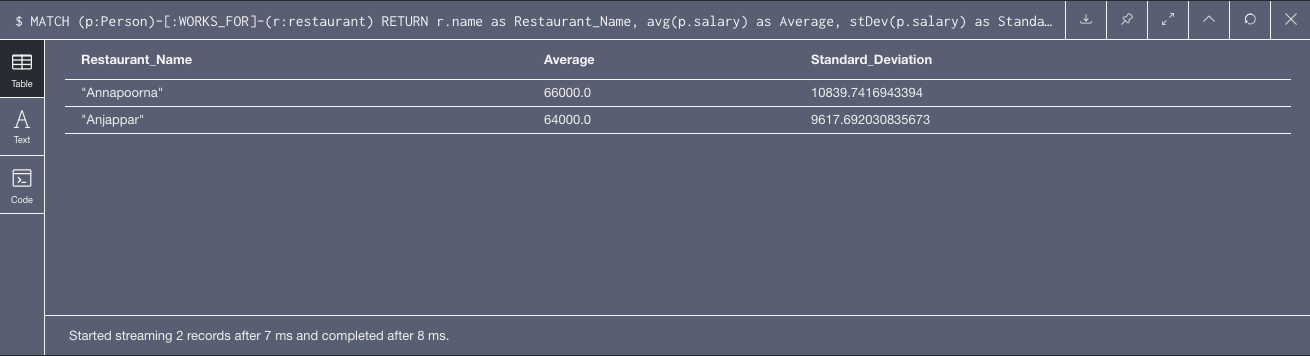
MATCH (r:restaurant)-[:HAS]-(f:FoodItem) RETURN r.name as Restaurant\_Name, sum(f.price) as Price



**8. Display the name of people whose salary is more than 65000 along with their distinction.**

MATCH (p:Person)-[w:WORKS\_FOR]-(r:restaurant) WHERE p.salary > 65000 RETURN p.name as Name, w.roles as JobDistinction, r.name as Restaurant\_Name

**9. Display the mean and standard deviation of salary of employees in both the restaurants.**

MATCH (p:Person)-[:WORKS\_FOR]-(r:restaurant) RETURN r.name as Restaurant\_Name, avg(p.salary) as Average, stDev(p.salary) as Standard\_Deviation

**10. Display the food items and number of food items eaten by the employees in the descending order.**

MATCH p=(subject)-[:EATS]->(f:FoodItem)-[:EATS\*0..2]-(f:FoodItem)

<-[:EATS]-(person:Person)-[:LIKES]->(v:Variety)

WHERE person<>subject

WITH person, v, f, max(length(p)) as pathLength

ORDER BY v.name RETURN person.name AS Name, count(f) AS NoOfItems, collect(f.name) AS FoodItems ORDER BY NoOfItems DESC

