Cypher List Expressions, Union, and Subqueries

1. Make a list of five Zip Codes. Use the list to determine what diagnoses are most common in those Zip Codes.

```
WITH ["02351","01364", "02124","02420","02703"] AS zipCodes

MATCH (p:Patient)-[:HAS_ZIPCODE]->(z:ZipCode)<-[:HAS_ZIPCODE]-(p2:Patient)-
[:HAS_ENCOUNTER]->(:Encounter)-[:HAS_DIAGNOSIS]->(d:Diagnosis)

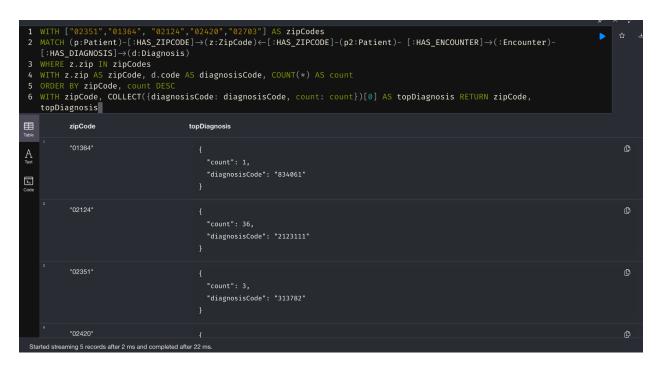
WHERE z.zip IN zipCodes

WITH z.zip AS zipCode, d.code AS diagnosisCode, COUNT(*) AS count

ORDER BY zipCode, count DESC

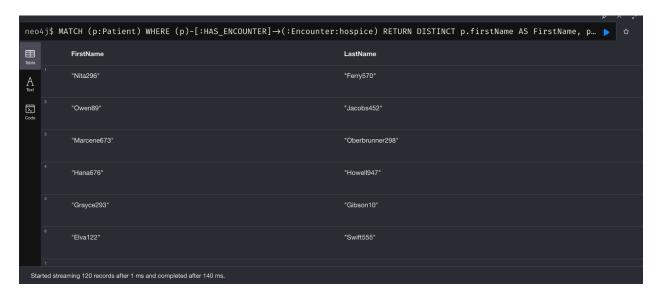
WITH zipCode, COLLECT({diagnosisCode: diagnosisCode, count: count})[0] AS topDiagnosis

RETURN zipCode,topDiagnosis
```



2. Which patients had hospice encounters?

MATCH (p:Patient)
WHERE (p)-[:HAS_ENCOUNTER]->(:Encounter:hospice)
RETURN DISTINCT p.firstName AS FirstName, p.lastName AS LastName



3. Make a list of the patient IDs as a list. Use the list in a CALL{} to obtain their SNOMED codes and providers in which they had their hospice encounter.

```
MATCH (p:Patient)

WITH COLLECT(p.id) AS patient_ids

UNWIND patient_ids AS patient_id

CALL {

WITH patient_id

MATCH (p:Patient)-[:HAS_ENCOUNTER]->(e:Encounter:hospice)-[:HAS_PROVIDER]->(pr:Provider)

MATCH (e:Encounter)-[:OF_TYPE]->(s:SNOMED_CT)

WHERE p.id = patient_id

RETURN collect(s.code) AS sno_code, collect(pr.name) AS pro }

RETURN patient_id, sno_code, pro
```

