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
```

```
1 WITH ["02351","01364", "02124","02420","02703"] AS zipCodes
2 MATCH (p:Patient)-[:HAS_ZIPCODE]→(z:ZipCode)←[:HAS_ZIPCODE]-(p2:Patient)- [:HAS_ENCOUNTER]→(:Encounter)-
  [:HAS_DIAGNOSIS]→(d:Diagnosis)
3 WHERE z.zip IN zipCodes
4 WITH z.zip AS zipCode, d.code AS diagnosisCode, COUNT(*) AS count
5 ORDER BY zipCode, count DESC
6 WITH zipCode, COLLECT({diagnosisCode: diagnosisCode, count: count})[0] AS topDiagnosis RETURN zipCode,
  topDiagnosis
```

Table | Text | Code

	zipCode	topDiagnosis
1	"01364"	{ "count": 1, "diagnosisCode": "834061" }
2	"02124"	{ "count": 36, "diagnosisCode": "2123111" }
3	"02351"	{ "count": 3, "diagnosisCode": "313782" }
4	"02420"	{

Started streaming 5 records after 2 ms and completed after 22 ms.

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
```

neo4j$ MATCH (p:Patient) WHERE (p)-[:HAS_ENCOUNTER]→(:Encounter:hospice) RETURN DISTINCT p.firstName AS FirstName, p…

Table | Text | Code

	FirstName	LastName
1	"Nita296"	"Ferry570"
2	"Owen89"	"Jacobs452"
3	"Marcene673"	"Oberbrunner298"
4	"Hana676"	"Howell947"
5	"Grayce293"	"Gibson10"
6	"Elva122"	"Swift555"
7		

Started streaming 120 records after 1 ms and completed after 140 ms.

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
```

```
neo4j$ MATCH (p:Patient) WITH COLLECT(p.id) AS patient_ids UNWIND patient_ids AS patient_id CALL { WITH patient_id MA…
```

Table | Text | Code

	patient_id	sno_code	pro
1	"000cf124-5321-ec70-ab86-ba7053eca8fb"	[]	[]
2	"0016bbd4-b85d-6e4c-29e2-a513715f9cc9"	[]	[]
3	"005708ca-7515-5cd8-8407-7af1e2973347"	["305336008"]	[]
4	"00c4a540-852e-002b-5f4f-fd0c24ffa09e"	[]	[]
5	"00e2b4c9-a258-01fa-2e2c-d168fd960ccb"	[]	[]
6	"013be793-896b-e757-8d35-d7c4d3884f76"	[]	[]
7			

Started streaming 1174 records after 5 ms and completed after 10 ms, displaying first 1000 rows.