1. Pretty Graph with all node and relationship types (but not too busy):

| MATCH (c:COURSE)-[]-(s:SKILL)-[]-(l:LISTING)-[]-(lo:LOCATION)  WHERE  s.skill\_name CONTAINS "python"  OR s.skill\_name CONTAINS "data"  OR s.skill\_name CONTAINS "java"  WITH c,s,l,lo  MATCH (co:COMPANY)-[]-(l:LISTING)-[]-(o:OCCUPATION)-[]-(f:CAREER\_OUTLOOK)  RETURN c,s,l,o,lo,co,f LIMIT 300 |
| --- |

1. What skills are needed for a six figure job?

| MATCH (s:SKILL) <- [n:NEEDS] - (l:LISTING) - [b:BELONGS\_TO] -> (o:OCCUPATION)  WHERE toInteger(o.occupation\_salary) >= 100000  RETURN o, s, l  LIMIT 1000 |
| --- |

1. What are the ten most popular skills needed for a six figure job?

| MATCH (s:SKILL) <- [n:NEEDS] - (l:LISTING) - [b:BELONGS\_TO] -> (o:OCCUPATION) - [f:HAS\_FUTURE] -> (c:CAREER\_OUTLOOK)  WHERE toInteger(o.occupation\_salary) >= 100000  AND c.career\_outlook = "Bright"  RETURN o.occupation\_title, s.skill\_name, COUNT(\*)  ORDER BY COUNT(\*) DESC  LIMIT 10 |
| --- |

1. How many of these skills do each coursera course teach?

| CALL  {  MATCH (s:SKILL) <- [n:NEEDS] - (l:LISTING) - [b:BELONGS\_TO] -> (o:OCCUPATION) - [f:HAS\_FUTURE] -> (c:CAREER\_OUTLOOK)  WHERE toInteger(o.occupation\_salary) >= 100000  AND c.career\_outlook = "Bright"  RETURN s.skill\_name AS top\_skills, count(\*)  ORDER BY COUNT(\*) DESC  LIMIT 10  }  WITH COLLECT(top\_skills) AS top\_skills\_array  MATCH (course:COURSE)-[teach:TEACHES]->(skill:SKILL)  WHERE skill.skill\_name in top\_skills\_array  RETURN course.course\_name, course.course\_url, count(skill)  ORDER BY count(skill) DESC |
| --- |

1. Given a favorite job (such as “linux”), find all job\_listings that need said favorite skill and count how many skills each job needs.

| CALL {  MATCH (favjobs:LISTING)-[:NEEDS]-(favskill:SKILL)  WHERE "python" in favskill.`aliases[]`  RETURN favjobs as j  }  WITH j  MATCH (j)-[:NEEDS]-(s:SKILL)  RETURN j.listing\_title as job\_listing\_title,  count(s.skill\_name) as number\_of\_skills\_needed,  collect(s.skill\_name) as skills\_needed  LIMIT 100 |
| --- |

Bonus code for a visual:

| CALL { MATCH (favjobs:LISTING)-[:NEEDS]-(favskill:SKILL)  WHERE "python" in favskill.`aliases[]`  RETURN favjobs as j  }  WITH j  MATCH (j)-[n:NEEDS]-(s:SKILL)  RETURN j,n,s  LIMIT 100 |
| --- |

1. Given a person’s skillset (such as “python”, “aws”, and “sql”), what job\_listings need my whole skillset?”

| CALL { MATCH (j:LISTING)-[:NEEDS]->(s:SKILL)  RETURN collect(s.skill\_name) AS basic\_skills, j  }  WITH basic\_skills, j  MATCH (j:LISTING)-[:NEEDS]->(s:SKILL)  WHERE all(x in ["python", "aws", "sql"] where x in basic\_skills)  RETURN j.listing\_title, collect(s.skill\_name)  LIMIT 100 |
| --- |

Bonus code for visual:

| CALL { MATCH (j:LISTING)-[:NEEDS]->(s:SKILL)  RETURN collect(s.skill\_name) AS basic\_skills, j  }  WITH basic\_skills, j  MATCH (j:LISTING)-[n:NEEDS]->(s:SKILL)  WHERE all(x in ["python", "aws", "sql"] where x in basic\_skills)  RETURN j,n,s  LIMIT 100 |
| --- |

1. What job listings are available for a Software Developer occupation in San Diego?

| MATCH (loc:LOCATION)-[]-(l:LISTING)-[]-(o:OCCUPATION)  WHERE o.occupation\_title CONTAINS "Software Developer"  AND loc.location\_name CONTAINS "San Diego"  RETURN l LIMIT 100 |
| --- |

1. What companies have the most listings for a Software Engineer in San Diego?

| MATCH (c:COMPANY)-[p:POSTED]->(l:LISTING)  WITH c,p,l  MATCH (o:OCCUPATION)<-[bel:BELONGS\_TO]-(l:LISTING)-[located:LOCATED\_IN]->(loc:LOCATION)  WHERE o.occupation\_title CONTAINS "Software Developer"  AND loc.location\_name CONTAINS "San Diego"  RETURN c.company\_name, count(l) |
| --- |