LOAD CSV WITH HEADERS FROM 'file:///public\_use-industry-skills-needs.csv' AS row

WITH row

// Create Industry nodes

MERGE (industry:Industry {name: row.industry\_name})

// Create Year nodes

MERGE (year:Year {value: row.year})

// Create SkillGroup nodes

MERGE (skill:SkillGroup {name: row.skill\_group\_name, rank: toInteger(row.skill\_group\_rank)})

MERGE (skillCat:SkillCategory {name: row.skill\_group\_category)})

MERGE (isicSec:isicSec {name: row.isic\_section\_name, index:isic\_section\_index)})

// Create Relationships

MERGE (industry)-[:IN\_YEAR]->(year)

MERGE (industry)-[:REQUIRES\_SKILL]->(skill);

MERGE (skill)-[:FROM\_CATEGORY]->(skillCat)

MERGE (industry)-[:HAS\_isic]->(isicSec)

MATCH (n)-[r]->(m) RETURN n, r, m LIMIT 25;

MATCH p=()-[r:IN\_YEAR]->() RETURN p LIMIT 25

MATCH (industry:Industry)

RETURN industry

LIMIT 10;

MATCH (year:Year)

RETURN year

LIMIT 10;

MATCH ()-[r:IN\_YEAR]->()

RETURN r

LIMIT 10;

LOAD CSV WITH HEADERS FROM 'file:///public\_use-industry-skills-needs.csv' AS row

WITH row

WHERE row.year IS NOT NULL AND row.industry\_name IS NOT NULL

MATCH (industry:Industry {name: row.industry\_name})

MATCH (year:Year {value: row.year})

MERGE (industry)-[:IN\_YEAR]->(year);

MATCH (industry:Industry)-[:IN\_YEAR]->(year:Year)

RETURN industry, year

LIMIT 10;