```
LOAD CSV WITH HEADERS FROM 'file:/songs.csv' AS row
CREATE (:songs{songID: row.songID, title: row.title, year: row.year})

LOAD CSV WITH HEADERS FROM 'file:/users.csv' AS row
CREATE (:users {userID: row.userID, name: row.name})

LOAD CSV WITH HEADERS FROM 'file:/userRatedSong.csv' AS row
MATCH (s:songs {songID: row.songID})
MATCH (u:users {userID: row.userID})
MERGE (u)-[r:RATED]->(s)
ON CREATE SET r.userRating = toFloat(row.userRating)

MATCH(u1:users)-[x:RATED]->(s:songs)<-[y:RATED]-(u2:users)
WITH ABS(SUM(REDUCE(x.userRating * y.userRating))) AS Aggregate,
    ABS(REDUCE(xDot = 0, i IN COLLECT(x.userRating) | xDot + toInteger(i))) AS xLength,
    ABS(REDUCE(yDot = 0, j IN COLLECT(y.userRating) | yDot + toInteger(j))) AS yLength,
    u1, u2
MERGE (u1)-[s:SIMILARITY]-(u2)
SET s.value = ((ABS(Aggregate))/((ABS(xLength))+(ABS(yLength))-(ABS(Aggregate)))+2)

MATCH(u1:users {name: 'Steve'})-[s:SIMILARITY]-(u2:users)
WITH u2,s.value AS sim
ORDER BY sim DESC
RETURN u2.name AS Neighbor, sim AS Similarity

MATCH(u1:users)-[x:RATED]->(s:songs)<-[y:RATED]-(u2:users)
WITH u1, u2
MERGE (u1)-[:IS_LIKE]->(u2)
```