//Create Post Label

USING PERIODIC COMMIT

LOAD CSV WITH HEADERS

FROM "file:///Posts1.csv" AS line

CREATE(p:Post{Id:toInteger(line.Id),PostTypeId:toInteger(line.PostTypeId),

AcceptedAnswerId:toInteger(line.AcceptedAnswerId),CreationTime:toInteger(line.CreationTime),

OwnerUserId:toInteger(line.OwnerUserId),LastEditorUserId:toInteger(line.LastEditorUserId),

LastEditTime:toInteger(line.LastEditTime),Title:line.Title,Tags:line.Tags,

ParentId:toInteger(line.ParentId)})

//Set Post property

MATCH (p:Post)

SET p.Tags=split(p.Tags,',')

//Create User Label

USING PERIODIC COMMIT

LOAD CSV WITH HEADERS

FROM "file:///Users.csv" AS row

CREATE(u:User{id:toInteger(row.Id),creationdate:row.CreationDate,

displayname:row.DisplayName,upvotes:toInteger(row.UpVotes),

downvotes:toInteger(row.DownVotes)})

//Create Vote Label

USING PERIODIC COMMIT

LOAD CSV WITH HEADERS

FROM "file:///Votes1.csv" AS line

CREATE(v:Vote{Id:toInteger(line.Id),PostId:toInteger(line.PostId),

VoteTypeId:toInteger(line.VoteTypeId),CreationTime:toInteger(line.CreationTime),

CreationDate:line.CreationDate,UserId:toInteger(line.UserId),

BountyAmount:toInteger(line.BountyAmount)})

//Create OWNED relationship

MATCH (p:Post),(u:User)

WHERE p.OwnerUserId = u.id

CREATE (p)<-[:OWNED]-(u)

//Create EDITED relationship

MATCH (p:Post),(u:User)

WHERE p.LastEditorUserId = u.id

CREATE (p)<-[:EDITED]-(u)

//Create ANSWERED relationship

MATCH (p1:Post),(p2:Post)

WHERE p2.ParentId=p1.Id

CREATE (p2)-[a:ANSWERED]->(p1)

//Create ACCEPTED

MATCH (p1:Post),(p2:Post)

WHERE p1.AcceptedAnswerId=p2.Id

CREATE (p1)-[a:ACCEPTED{time:p2.CreationTime-p1.CreationTime}]->(p2)

//Create NOT\_ACCEPTED

MATCH (p1:Post)-[:ANSWERED]->(p2:Post)

WHERE NOT EXISTS(p2.AcceptedAnswerId) OR p1.Id<>p2.AcceptedAnswerId

CREATE (p1)<-[:NOT\_ACCEPTED]-(p2)

//UPVOTED

MATCH (p:Post),(v:Vote)

WHERE v.PostId=p.Id AND v.VoteTypeId=2

CREATE (v)-[:UPVOTED{CreationTime:v.CreationTime}]->(p)

//DECIDED

MATCH (p:Post),(v:Vote)

WHERE v.PostId=p.Id AND v.VoteTypeId=1

CREATE (v)<-[:DECIDED]-(p)

//Set Post property

MATCH (p:Post)-[d:DECIDED]-(v:Vote)

SET p.DecisionDay=v.CreationTime

//Create INVOLVED\_IN relationship

MATCH (p1:Post)<-[:ANSWERED]-(p2:Post)

OPTIONAL MATCH (u1:User)-[:EDITED]-(p1)

OPTIONAL MATCH (u2:User)-[:OWNED]-(p1)

OPTIONAL MATCH (u3:User)-[:EDITED]-(p2)

OPTIONAL MATCH (u4:User)-[:OWNED]-(p2)

WHERE EXISTS(u1.id) OR EXISTS(u2.id) OR EXISTS(u3.id) OR EXISTS(u4.id)

WITH p1,(collect(u1)+collect(u2)+collect(u3)+collect(u4)) as users

UNWIND users as user

WITH p1,user

MERGE (p1)<-[:INVOLVED\_IN]-(user)