// Create User nodes and Friendship relationships

LOAD CSV WITH HEADERS FROM 'file:///combined.csv' AS row

MERGE (u1:User {user\_id: toInteger(row.user\_id)})

ON CREATE SET u1.name = row.name, u1.age = toInteger(row.age), u1.city = row.city

MERGE (u2:User {user\_id: toInteger(row.friend\_id)})

ON CREATE SET u2.name = "", u2.age = 0, u2.city = ""

MERGE (u1)-[:FRIEND]->(u2);

// Create Post nodes and Connect User nodes to their respective Post nodes

WITH DISTINCT row WHERE row.content IS NOT NULL

MERGE (p:Post {post\_id: toInteger(row.post\_id)})

ON CREATE SET p.content = row.content

MERGE (u:User {user\_id: toInteger(row.user\_id)})

ON CREATE SET u.name = "", u.age = 0, u.city = ""

MERGE (u)-[:POSTED]->(p);

// Create Comment nodes and Connect User nodes, Post nodes, and Comment nodes

WITH DISTINCT row WHERE row.comment\_id IS NOT NULL AND row.content IS NOT NULL

MERGE (c:Comment {comment\_id: toInteger(row.comment\_id)})

ON CREATE SET c.content = row.content

MERGE (u:User {user\_id: toInteger(row.user\_id)})

ON CREATE SET u.name = "", u.age = 0, u.city = ""

MERGE (p:Post {post\_id: toInteger(row.post\_id)})

ON CREATE SET p.content = ""

MERGE (u)-[:COMMENTED]->(c)

MERGE (c)-[:ON]->(p);

// Create Like relationships

WITH DISTINCT row WHERE row.like\_id IS NOT NULL

MERGE (u:User {user\_id: toInteger(row.user\_id)})

ON CREATE SET u.name = "", u.age = 0, u.city = ""

MERGE (p:Post {post\_id: toInteger(row.post\_id)})

ON CREATE SET p.content = ""

MERGE (u)-[:LIKED]->(p);