USING PERIODIC COMMIT

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

CREATE (exercise:Exercise {id: row.ID, name: row.Name})

USING PERIODIC COMMIT

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

CREATE (muscle:Muscle {id: row.ID, name: row.Name, group: row.Group})

USING PERIODIC COMMIT

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

CREATE (equipment:Equipment {id: row.ID, name: row.Name})

USING PERIODIC COMMIT

LOAD CSV WITH HEADERS FROM 'file:///exercise\_types.csv' AS row

CREATE (exercise\_type:Exercise\_Type {id: row.ID, name: row.Name})

USING PERIODIC COMMIT

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

CREATE (workout:Workout {id: row.ID, date: row.Date, exercises: row.Exercises\_By\_ID})

USING PERIODIC COMMIT

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

CREATE (user:User {id: row.ID, first\_name: row.First\_Name, last\_name: row.Last\_Name, age: row.Age, weight\_in\_pounds: row.Weight\_In\_Pounds, height\_in\_inches: row.Height\_In\_Inches, bmi: row.BMI})

CREATE INDEX ON :Exercise(name)

CREATE INDEX ON :Muscle(name)

CREATE INDEX ON :Equipment(name)

CREATE INDEX ON :Exercise\_Type(id)

CREATE INDEX ON :Workout(id)

CREATE INDEX ON :User(id)

USING PERIODIC COMMIT

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

MATCH (e:Exercise {id: row.ID})

MATCH (m:Muscle {id: row.Muscle\_ID})

MERGE (e)-[:USES]->(m)

USING PERIODIC COMMIT

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

MATCH (e:Exercise {id: row.ID})

MATCH (t:Exercise\_Type {id: row.Exercise\_Type\_ID})

MERGE (e)-[:IS\_TYPE\_OF]->(t)

USING PERIODIC COMMIT

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

MATCH (e:Equipment {id: row.ID})

MATCH (t:Exercise\_Type {id: row.Type\_ID})

MERGE (e)-[:TRAINS]->(t)

USING PERIODIC COMMIT

LOAD CSV WITH HEADERS FROM 'file:///workout\_relations.csv' AS row

MATCH (u:User {id: row.User\_ID})

MATCH (w:Workout {id: row.Workout\_ID})

MERGE (u)-[:PARTICIPATES\_IN]->(w)

USING PERIODIC COMMIT

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

MATCH (w:Workout {id: row.ID})

UNWIND split(w.exercises, ',') AS exercise

MATCH (e:Exercise {id: exercise})

MERGE (w)-[:HAS]->(e)