**Neo4j Code**

**Students:**

**First:**

CREATE CONSTRAINT UniqueStudent ON (s:Students) ASSERT s.students\_id IS UNIQUE;

To say what is PK

**AND**

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

WITH row.students\_id AS students\_id,

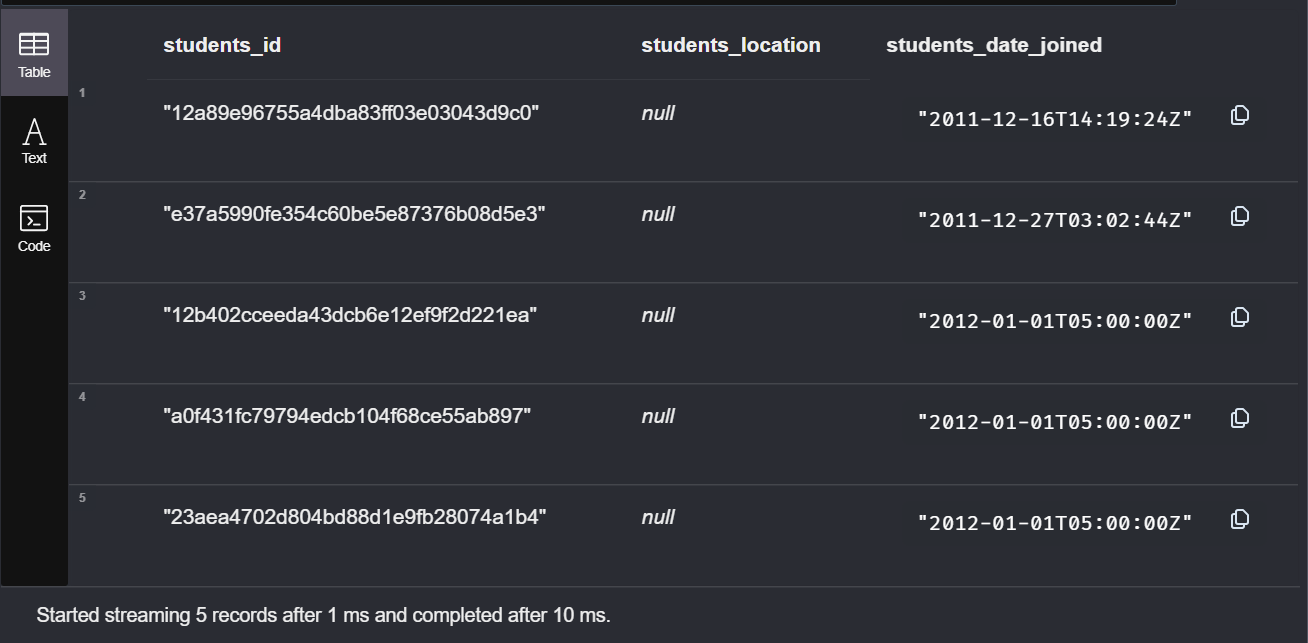
row.students\_location AS students\_location,

datetime(replace(replace(row.students\_date\_joined, " UTC", ""), " ", "T")) AS students\_date\_joined

RETURN students\_id, students\_location, students\_date\_joined

LIMIT 5;

To check if data loads correctly



**Then load:**

:auto USING PERIODIC COMMIT

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

WITH row.students\_id AS students\_id,

row.students\_location AS students\_location,

datetime(replace(replace(row.students\_date\_joined, " UTC", ""), " ", "T")) AS students\_date\_joined

MERGE (s:Students {students\_id: students\_id})

  SET s.students\_location = students\_location, s.students\_date\_joined = students\_date\_joined

RETURN count(s);

Added 30971 labels, created 30971 nodes, set 92854 properties, started streaming 1 records after 115 ms and completed after 2869 ms.

**Professionals:**

First:

CREATE CONSTRAINT UniqueProfessional ON (p:Professional) ASSERT p.professionals\_id IS UNIQUE;

AND

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

WITH row.professionals\_id AS professionals\_id,

row.professionals\_location AS professionals\_location,

row.professionals\_industry AS professionals\_industry,

row.professionals\_headline AS professionals\_headline,

datetime(replace(replace(row.professionals\_date\_joined, " UTC", ""), " ", "T")) AS professionals\_date\_joined

RETURN professionals\_id, professionals\_location, professionals\_industry, professionals\_headline, professionals\_date\_joined

LIMIT 5;

A screenshot of a computer

Description automatically generated with medium confidence

LOAD;

:auto USING PERIODIC COMMIT

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

WITH row.professionals\_id AS professionals\_id,

row.professionals\_location AS professionals\_location,

row.professionals\_industry AS professionals\_industry,

row.professionals\_headline AS professionals\_headline,

datetime(replace(replace(row.professionals\_date\_joined, " UTC", ""), " ", "T")) AS professionals\_date\_joined

MERGE (p:Professional {professionals\_id: professionals\_id})

  SET p.professionals\_location = professionals\_location, p.professionals\_industry = professionals\_industry,

p.professionals\_headline = professionals\_headline,

p.professionals\_date\_joined = professionals\_date\_joined

RETURN count(p);

Added 28152 labels, created 28152 nodes, set 140733 properties, started streaming 1 records after 95 ms and completed after 1741 ms.

Questions:

FIRST:

CREATE CONSTRAINT UniqueQuestion ON (q:Questions) ASSERT q.questions\_id IS UNIQUE;

AND

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

WITH row.questions\_id AS questions\_id,

row.questions\_author\_id AS questions\_author\_id,

datetime(replace(replace(row.questions\_date\_added, " UTC", ""), " ", "T")) AS questions\_date\_added,

row.questions\_title AS questions\_title,

row.questions\_body AS questions\_body

RETURN questions\_id, questions\_author\_id, questions\_date\_added, questions\_title, questions\_body

LIMIT 5;

Graphical user interface, text

Description automatically generated

LOAD:

:auto USING PERIODIC COMMIT

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

WITH row.questions\_id AS questions\_id,

row.questions\_author\_id AS questions\_author\_id,

datetime(replace(replace(row.questions\_date\_added, " UTC", ""), " ", "T")) AS questions\_date\_added,

row.questions\_title AS questions\_title,

row.questions\_body AS questions\_body

MERGE (q:Questions {questions\_id: questions\_id})

  SET q.questions\_author\_id = questions\_author\_id, q.questions\_date\_added = questions\_date\_added,

q.questions\_title = questions\_title,

q.questions\_body = questions\_body

RETURN count(q);

Added 23931 labels, created 23931 nodes, set 119655 properties, started streaming 1 records after 91 ms and completed after 1521 ms.

Relation ship Student -> posts -> Questions:

MATCH (s:Students {students\_id:s.students\_id})

MATCH (q:Questions {questions\_author\_id:s.students\_id})

MERGE (s)-[:posts] ->(q)

Created 23801 relationships, completed after 441 ms.

A screenshot of a computer

Description automatically generated with low confidence

ASWERS:

FISRT:

CREATE CONSTRAINT UniqueAnswer ON (a:Answers) ASSERT a.answers\_id IS UNIQUE;

AND

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

WITH row.answers\_id AS answers\_id,

row.answers\_author\_id AS answers\_author\_id,

row.answers\_question\_id AS answers\_question\_id,

datetime(replace(replace(row.answers\_date\_added, " UTC", ""), " ", "T")) AS answers\_date\_added,

row.answers\_body AS answers\_body

RETURN answers\_id, answers\_author\_id, answers\_question\_id, answers\_date\_added, answers\_body

LIMIT 5;

Graphical user interface, text

Description automatically generated

LOAD:

:auto USING PERIODIC COMMIT

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

WITH row.answers\_id AS answers\_id,

row.answers\_author\_id AS answers\_author\_id,

row.answers\_question\_id AS answers\_question\_id,

datetime(replace(replace(row.answers\_date\_added, " UTC", ""), " ", "T")) AS answers\_date\_added,

row.answers\_body AS answers\_body

MERGE (a:Answers {answers\_id: answers\_id})

  SET a.answers\_author\_id = answers\_author\_id, a.answers\_question\_id = answers\_question\_id,

a.answers\_date\_added = answers\_date\_added,

a.answers\_body = answers\_body

RETURN count(a);

Added 51123 labels, created 51123 nodes, set 255615 properties, started streaming 1 records after 47 ms and completed after 1746 ms.

Relation: Professionals -> posts -> Answers, Answers -> to -> Questions, Students -> posts -> Answers

MATCH (p:Professional {professionals\_id:p.professionals\_id})

MATCH (a:Answers {answers\_author\_id:p.professionals\_id})

MERGE (p)-[:posts] ->(a)

AND

MATCH (s:Students {students\_id:s.students\_id})

MATCH (a:Answers {answers\_author\_id:s.students\_id})

MERGE (s)-[:posts] ->(a)

Only 46 records

AND

MATCH (q:Questions {questions\_id:q.questions\_id})

MATCH (a:Answers {answers\_question\_id:q.questions\_id})

MERGE (a)-[:to] ->(q)

Diagram

Description automatically generated

COMMENTS:

First:

CREATE CONSTRAINT UniqueComment ON (c:Comments) ASSERT c.comments\_id IS UNIQUE;

AND

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

WITH row.comments\_id AS comments\_id,

row.comments\_author\_id AS comments\_author\_id,

row.comments\_parent\_content\_id AS comments\_parent\_content\_id,

datetime(replace(replace(row.comments\_date\_added, " UTC", ""), " ", "T")) AS comments\_date\_added,

row.comments\_body AS comments\_body

RETURN comments\_id, comments\_author\_id, comments\_parent\_content\_id, comments\_date\_added, comments\_body

LIMIT 5;

LOAD:

:auto USING PERIODIC COMMIT

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

WITH row.comments\_id AS comments\_id,

row.comments\_author\_id AS comments\_author\_id,

row.comments\_parent\_content\_id AS comments\_parent\_content\_id,

datetime(replace(replace(row.comments\_date\_added, " UTC", ""), " ", "T")) AS comments\_date\_added,

row.comments\_body AS comments\_body

MERGE (c:Comments {comments\_id: comments\_id})

  SET c.comments\_author\_id = comments\_author\_id, c.comments\_parent\_content\_id = comments\_parent\_content\_id,

c.comments\_date\_added = comments\_date\_added,

c.comments\_body = comments\_body

RETURN count(c);

Added 14966 labels, created 14966 nodes, set 74830 properties, started streaming 1 records after 40 ms and completed after 474 ms.

RELATION: Studends and Proffesionals -> posts -> comments -> on -> answers and questions:

MATCH (s:Students {students\_id:s.students\_id})

MATCH (c:Comments {comments\_author\_id:s.students\_id})

MERGE (s)-[:posts] ->(c)

AND

MATCH (p:Professional{professionals\_id:p.professionals\_id})

MATCH (c:Comments {comments\_author\_id:p.professionals\_id})

MERGE (p)-[:posts] ->(c)

AND

MATCH (a:Answers{answers\_id:a.answers\_id})

MATCH (c:Comments {comments\_parent\_content\_id:a.answers\_id})

MERGE (c)-[:on] ->(a)

AND

MATCH (q:Questions {questions\_id:q.questions\_id})

MATCH (c:Comments {comments\_parent\_content\_id:q.questions\_id})

MERGE (c)-[:on] ->(q)

A picture containing chart

Description automatically generated

(Hard to find comments on questions cuz they are rare, 16% of relations :on)

EMAILS:

First:

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

WITH toInteger(row.emails\_id) AS emails\_id,

row.emails\_recipient\_id AS emails\_recipient\_id,

datetime(replace(replace(row.emails\_date\_sent, " UTC", ""), " ", "T")) AS emails\_date\_sent,

row.emails\_frequency\_level AS emails\_frequency\_level

RETURN emails\_id, emails\_recipient\_id, emails\_date\_sent, emails\_frequency\_level

LIMIT 5;

AND

CREATE CONSTRAINT UniqueEmail ON (e:Emails) ASSERT e.emails\_id IS UNIQUE;

LOAD:

:auto USING PERIODIC COMMIT

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

WITH toInteger(row.emails\_id) AS emails\_id,

row.emails\_recipient\_id AS emails\_recipient\_id,

datetime(replace(replace(row.emails\_date\_sent, " UTC", ""), " ", "T")) AS emails\_date\_sent,

row.emails\_frequency\_level AS emails\_frequency\_level

MERGE (e:Emails {emails\_id: emails\_id})

  SET e.emails\_recipient\_id = emails\_recipient\_id, e.emails\_date\_sent = emails\_date\_sent,

e.emails\_frequency\_level = emails\_frequency\_level

RETURN count(e);

Had to add WITH row LIMIT 500000

Because of:

java.lang.OutOfMemoryError: Java heap space

The database has encountered a critical error, and needs to be restarted. Please see database logs for more details.

Added 500000 labels, created 500000 nodes, set 2000000 properties, started streaming 1 records after 209 ms and completed after 23600 ms.

MATCHES:

RELATIONSHIP: questions – matches – emails

:auto USING PERIODIC COMMIT

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

WITH toInteger(row.matches\_email\_id) AS matches\_email\_id,

row.matches\_question\_id AS matches\_question\_id

MATCH (q:Questions {questions\_id:matches\_question\_id})

MATCH (e:Emails {emails\_id:matches\_email\_id})

MERGE (q)-[rel:matches]-(e)

RETURN count(rel);

Without LIMIT takes very long time, added WITH row LIMIT 4000000.

Created 890891 relationships, started streaming 1 records after 51 ms and completed after 49628 ms.

A picture containing graphical user interface

Description automatically generated

RELATIONSHIP: Emails -> send\_to -> students AND Emails -> send\_to -> Profesionals

MATCH (s:Students {students\_id:s.students\_id})

MATCH (e:Emails {emails\_recipient\_id:s.students\_id})

MERGE (e)-[:send\_to] ->(s)

NO MATCHES

MATCH (p:Professional {professionals\_id:p.professionals\_id})

MATCH (e:Emails {emails\_recipient\_id:p.professionals\_id})

MERGE (e)-[:send\_to] ->(p)

500000 MATCHES, all emails that were loaded are for proffesionals

SCHOOL\_MEMBERSCHIP:

FIRST:

LOAD CSV WITH HEADERS FROM 'file:///school\_memberships.csv' AS row

WITH toInteger(row.school\_memberships\_school\_id) AS school\_memberships\_school\_id,

row.school\_memberships\_user\_id AS school\_memberships\_user\_id

RETURN school\_memberships\_school\_id, school\_memberships\_user\_id

LIMIT 5;

LOAD:

:auto USING PERIODIC COMMIT

LOAD CSV WITH HEADERS FROM 'file:///school\_memberships.csv' AS row

WITH toInteger(row.school\_memberships\_school\_id) AS school\_memberships\_school\_id,

row.school\_memberships\_user\_id AS school\_memberships\_user\_id

MERGE (sch:School {school\_memberships\_school\_id: school\_memberships\_school\_id})

  SET sch.school\_memberships\_user\_id = school\_memberships\_user\_id

RETURN count(sch);

Added 2706 labels, created 2706 nodes, set 8344 properties, started streaming 1 records after 39 ms and completed after 2009 ms.

RELATIONSHIP: student -> has -> school\_membership

MATCH (s:Students {students\_id:s.students\_id})

MATCH (sch:School {school\_memberships\_user\_id:s.students\_id})

MERGE (s)-[:has] ->(sch)

RELATIONSHIP: Professional -> has -> school\_membership

MATCH (p:Professional {professionals\_id:p.professionals\_id})

MATCH (sch:School {school\_memberships\_user\_id:p.professionals\_id})

MERGE (p)-[:has] ->(sch)

Created 1984 relationships, completed after 44 ms.

A picture containing background pattern

Description automatically generated

GROUPS:

First:

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

WITH row.groups\_id AS groups\_id,

row.groups\_group\_type AS groups\_group\_type

RETURN groups\_id, groups\_group\_type

LIMIT 5;

AND

CREATE CONSTRAINT UniqueGroup ON (g:Groups) ASSERT g.groups\_id IS UNIQUE;

LOAD:

:auto USING PERIODIC COMMIT

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

WITH row.groups\_id AS groups\_id,

row.groups\_group\_type AS groups\_group\_type

MERGE (g:Groups {groups\_id: groups\_id})

  SET g.groups\_group\_type = groups\_group\_type

RETURN count(g);

Added 49 labels, created 49 nodes, set 98 properties, started streaming 1 records after 26 ms and completed after 52 ms.

GROUP\_MEMBERSHIP:

First:

LOAD CSV WITH HEADERS FROM 'file:///group\_memberships.csv' AS row

WITH row.group\_memberships\_group\_id AS group\_memberships\_group\_id,

row.group\_memberships\_user\_id AS group\_memberships\_user\_id

RETURN group\_memberships\_group\_id, group\_memberships\_user\_id

LIMIT 5;

Relationship: Student -> has\_membership -> Groups And Proffesionals -> has\_membership -> Groups

:auto USING PERIODIC COMMIT

LOAD CSV WITH HEADERS FROM 'file:///group\_memberships.csv' AS row

WITH row.group\_memberships\_group\_id AS group\_memberships\_group\_id,

row.group\_memberships\_user\_id AS group\_memberships\_user\_id

MATCH (s:Students {students\_id:group\_memberships\_user\_id})

MATCH (g:Groups {groups\_id:group\_memberships\_group\_id})

MERGE (s)-[rel:has\_membership]->(g)

RETURN count(rel);

Created 310 relationships, started streaming 1 records after 33 ms and completed after 70 ms.

:auto USING PERIODIC COMMIT

LOAD CSV WITH HEADERS FROM 'file:///group\_memberships.csv' AS row

WITH row.group\_memberships\_group\_id AS group\_memberships\_group\_id,

row.group\_memberships\_user\_id AS group\_memberships\_user\_id

MATCH (p:Professional {professionals\_id:group\_memberships\_user\_id})

MATCH (g:Groups {groups\_id:group\_memberships\_group\_id})

MERGE (p)-[rel:has\_membership]->(g)

RETURN count(rel);

Created 727 relationships, started streaming 1 records after 35 ms and completed after 70 ms.

A screenshot of a map

Description automatically generated with medium confidence

TAGS:

FIRST:

CREATE CONSTRAINT UniqueTag ON (t:Tags) ASSERT t.tags\_tag\_id IS UNIQUE;

AND

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

WITH row.tags\_tag\_id AS tags\_tag\_id,

row.tags\_tag\_name AS tags\_tag\_name

RETURN tags\_tag\_id, tags\_tag\_name

LIMIT 5;

LOAD:

:auto USING PERIODIC COMMIT

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

WITH row.tags\_tag\_id AS tags\_tag\_id,

row.tags\_tag\_name AS tags\_tag\_name

MERGE (t:Tags {tags\_tag\_id: tags\_tag\_id})

  SET t.tags\_tag\_name = tags\_tag\_name

RETURN count(t);

Added 16269 labels, created 16269 nodes, set 32537 properties, started streaming 1 records after 27 ms and completed after 489 ms.

TAG\_QUESTIONS:

RELATIONSHIP: Questions -> has\_tag -> Tags

:auto USING PERIODIC COMMIT

LOAD CSV WITH HEADERS FROM 'file:///tag\_questions.csv' AS row

WITH toInteger(row.tag\_questions\_tag\_id) AS tag\_questions\_tag\_id,

row.tag\_questions\_question\_id AS tag\_questions\_question\_id

MATCH (q:Questions {questions\_id:tag\_questions\_question\_id})

MATCH (t:Tags {tags\_tag\_id:tag\_questions\_tag\_id})

MERGE (q)-[rel:has\_tag]->(t)

RETURN count(rel);

Created 76553 relationships, started streaming 1 records after 1 ms and completed after 1889 ms.

TAG\_USERS:

RELATIONSHIP: Students -> has\_tag -> Tags AND Professionals -> has\_tag -> Tags

:auto USING PERIODIC COMMIT

LOAD CSV WITH HEADERS FROM 'file:///tag\_users.csv' AS row

WITH toInteger(row.tag\_users\_tag\_id) AS tag\_users\_tag\_id,

row.tag\_users\_user\_id AS tag\_users\_user\_id

MATCH (s:Students {students\_id:tag\_users\_user\_id})

MATCH (t:Tags {tags\_tag\_id:tag\_users\_tag\_id})

MERGE (s)-[rel:has\_tag]->(t)

RETURN count(rel);

Created 19078 relationships, started streaming 1 records after 46 ms and completed after 1401 ms.

:auto USING PERIODIC COMMIT

LOAD CSV WITH HEADERS FROM 'file:///tag\_users.csv' AS row

WITH toInteger(row.tag\_users\_tag\_id) AS tag\_users\_tag\_id,

row.tag\_users\_user\_id AS tag\_users\_user\_id

MATCH (p:Professional {professionals\_id:tag\_users\_user\_id})

MATCH (t:Tags {tags\_tag\_id:tag\_users\_tag\_id})

MERGE (p)-[rel:has\_tag]->(t)

RETURN count(rel);

Created 116829 relationships, started streaming 1 records after 38 ms and completed after 2296 ms.

NO students has the same tag as professional

MATCH (s:Students) - [r:has\_tag] -> (t:Tags) <- [r1:has\_tag] - (p:Professional) WHERE r=r1 RETURN s,r,t,p LIMIT 20;

A picture containing background pattern

Description automatically generated A picture containing chart

Description automatically generated

Question\_score:

First:

LOAD CSV WITH HEADERS FROM 'file:///question\_scores.csv' AS row

WITH row.id AS id,

toInteger(row.score) AS score

RETURN id, score

LIMIT 5;

LOAD:

:auto USING PERIODIC COMMIT

LOAD CSV WITH HEADERS FROM 'file:///question\_scores.csv' AS row

WITH row.id AS id,

toInteger(row.score) AS score

MERGE (qs:Qscore {id: id})

  SET qs.score = score

RETURN count(qs);

Added 23928 labels, created 23928 nodes, set 47856 properties, started streaming 1 records after 26 ms and completed after 172737 ms.

RELATIONSHIP: Questions -> has\_score -> Qscore

MATCH (q:Questions {questions\_id:q.questions\_id})

MATCH (qs:Qscore {id:q.questions\_id})

MERGE (q)-[:has\_score] ->(qs)

Created 23928 relationships, completed after 384 ms.

Answer\_score:

LOAD:

:auto USING PERIODIC COMMIT

LOAD CSV WITH HEADERS FROM 'file:///answer\_scores.csv' AS row

WITH row.id AS id,

toInteger(row.score) AS score

MERGE (ass:Ascore {id: id})

  SET ass.score = score

RETURN count(ass);

Added 51138 labels, created 51138 nodes, set 102276 properties, started streaming 1 records after 29 ms and completed after 774941 ms.

RELATIONSHIP: Answers -> has\_score -> Qscore

MATCH (a:Answers {answers\_id:a.answers\_id})

MATCH (ass:Ascore {id:a.answers\_id})

MERGE (a)-[:has\_score] ->(ass)

Created 51107 relationships, completed after 440 ms.

END SCORES:

Graphical user interface, application

Description automatically generated