Group 38
Haoqi Yu, Bjørn Magnus Hoddevik

Our approach

We began by uploading the CSV files somewhere public, we chose GitHub. Then we went through each file and created insert queries for each node, starting with students and programmes. If the file had extra information which belonged in a separate node or relationship, we added them as soon as possible. Meaning, we couldn't add the relationship between students and programmes until we had nodes for both already inserted into the database. From there we continued, sometimes making relationships and new nodes at the same time. The same formula worked for most cases except two. Our course offering node had its information split over two CSV files, which meant we had to use a match and set combo for the second file. And lastly, since not all registrations had a grade, we had to split it in two. First we create the relationship with status. Then we match those relationships and filter where grade is not null, and then set grade for that relationship.

For queries we worked iteratively, making each query more complex by adding more criteria. By this point we realized that we had forgotten to convert the types for the data from CSV files, so we had to go back and do that.

Decision and Assumptions

We made the decision to work at one file at a time, and at most one node and a relationship at a time. This made it more manageable and allowed us to check that each data load went as planned. We made all relationships one way. We did not make separate nodes for relations that have attributes, instead we kept them as relationships.

No assumptions were made

Queries

We have attached these queries in separate files:

- `Data.cypher` for data loading
- `Queries.cypher` for queries

Summary of results:

Question 1:

programme.programmeName	programme.director	department.departmentName
"P-01"	"19620522-0023"	"D1"
"P-11"	"19620424-0026"	"D2"
"P-12"	"19610623-0005"	"D2"
"P-13"	"19690408-0009"	"D2"
"P-14"	"19560812-0016"	"D2"
"P-21"	"19570615-0011"	"D3"
"P-31"	"19650303-0019"	"D4"
"P-32"	"19570826-0012"	"D4"
"P-33"	"19570828-0008"	"D4"
"P-34"	"19610918-0027"	"D4"
"P-41"	"19580218-0007"	"D5"
"P-42"	"19620831-0024"	"D5"
"P-51"	"19580515-0017"	"D6"
"P-52"	"19611219-0014"	"D6"
"P-53"	"19600905-0003"	"D6"
"P-54"	"19630126-0001"	"D6"
"P-61"	"19680712-0028"	"D7"
"P-71"	"19610620-0006"	"D8"
"P-72"	"19660630-0020"	"D8"
"P-73"	"19600814-0002"	"D8"
"P-74"	"19601021-0018"	"D8"

Question 2:

(student.studentName)
"TA 36"
"TA 138"
"TA 38"
"TA 74"
"TA 60"

Question 3:

teacher.teacherName
"Teacher 19"
"Teacher 19"
"TA 59"
"TA 59"
"Teacher 7"
"Teacher 7"
"TA 38"
"TA 38"
"Teacher 20"
"Teacher 20"
"TA 57"

Question 4:

student
(:Student {studentId: "19921201-0094",year: "2019",graduated: "False", studentName: "TA 94"})

Question 5:

programme.programmeName	totalOwnedCourses
"P-61"	45
"P-21"	33
"P-01"	32
"P-41"	21
"P-42"	20
"P-32"	14
"P-71"	14
"P-52"	13
"P-12"	12
"P-33"	12
"P-51"	11
"P-54"	11
"P-11"	10
"P-72"	10
"P-74"	9
"P-13"	8
"P-14"	8

"P-34"	6
"P-73"	6
"P-53"	3
"P-31"	2

Question 6:

Total senior teachers: 30
Total people: 440