1. What is the average student age? 20.6

MATCH (s:Student)

with s.age as ageString

return avg(toInteger(ageString))

1. Count the number of rooms

MATCH (n:Room) RETURN count(\*)

1. List the course names

MATCH (n:Course) RETURN n.courseName

1. List the project names

MATCH (n:Project) RETURN n.projectName

1. Count the number of rooms databases takes place in

MATCH (course:Course {courseName:"Databases"}) – [:TAKESPLACEIN] -> (room)

RETURN count(\*)

1. Count the number of rooms programming takes place in

match (course:Course {courseName:'Programming'})-[:TAKESPLACEIN]->(room)

return count(room)

1. Create a relationship to show that programming will take place in the Beta room

MATCH (c:Course),(r:Room)

WHERE c.courseName = 'Programming'AND r.roomName = 'Beta'

CREATE p = (c)-[t:TAKESPLACEIN{ name:"Programming- Beta" }]-> (r)

RETURN p

1. Count the number of rooms programming takes place in

MATCH (c:Course {courseName:'Programming'})-[:TAKESPLACEIN]->(r)

return count(\*)

// the cypher below is more appropriate but ‘r:Room’ doesn’t seem to be necessary. ‘r’ on it’s own seems to work

MATCH (c:Course {courseName:'Programming'})-[:TAKESPLACEIN]->(r:Room)

return count(\*)

1. List the rooms that programming takes place in

MATCH (c:Course {courseName:'Programming'})-[:TAKESPLACEIN]->(r:Room)

return r.roomName

1. List the courses that take place in the Beta room

MATCH (c: Course)-[:TAKESPLACEIN]->(r:Room)

where r.roomName = 'Beta'

return c.courseName

1. Count the number of courses that take place in the Beta room

MATCH (c: Course)-[:TAKESPLACEIN]->(r:Room)

where r.roomName = 'Beta'

return count(c)