Create Nodes

```
CREATE (c1:Course {courseNr:1}) – [:TAKESPLACEIN] –> (r1:Room {roomName:"Pascal"});
CREATE (c1:Course {courseNr:1}) – [:TAKESPLACEIN] –> (r3:Room {roomName:"Alpha"});
CREATE (c1:Course {courseNr:1}) – [:TAKESPLACEIN] –> (r4:Room {roomName:"Beta"});
CREATE (c2:Course {courseNr:2}) – [:TAKESPLACEIN] –> (r2:Room {roomName:"Seminar
C"});
CREATE (s1:Student {studentID:1}) – [:ENROLLEDIN] –> (c1:Course {courseNr:1});
CREATE (s2:Student {studentID:2}) – [:ENROLLEDIN] –> (c1:Course {courseNr:1});
CREATE (s3:Student {studentID:3}) – [:ENROLLEDIN] –> (c2:Course {courseNr:2});
CREATE (s4:Student {studentID:4}) – [:ENROLLEDIN] –> (c1:Course {courseNr:1});
CREATE (s1:Student{studentID:1}) – [:WORKSON {hours: "1"}] –> (p1:Project {projectNr:"34"});
CREATE (s1:Student{studentID:1}) – [:WORKSON {hours: "2"}] –> (p2:Project {projectNr:"24"});
CREATE (s2:Student{studentID:2}) – [:WORKSON {hours: "3"}] –> (p1:Project {projectNr:"34"});
CREATE (s2:Student{studentID:2}) – [:WORKSON {hours: "4"}] –> (p2:Project {projectNr:"24"});
CREATE (s2:Student{studentID:2}) – [:WORKSON {hours: "1"}] –> (p3:Project {projectNr:"13"});
CREATE (s2:Student{studentID:2}) – [:WORKSON {hours: "1"}] –> (p4:Project {projectNr:"26"});
CREATE (s3:Student{studentID:3}) – [:WORKSON {hours: "1"}] –> (p1:Project {projectNr:"34"});
CREATE (s3:Student{studentID:3}) – [:WORKSON {hours: "2"}] –> (p2:Project {projectNr:"24"});
CREATE (s3:Student{studentID:3}) – [:WORKSON {hours: "3"}] –> (p4:Project {projectNr:"26"});
```

Create relations

```
MATCH (c1:Course {courseNr:"1"}), (r1:Room {roomName:"Pascal"}) MERGE
(c1)–[:TAKESPLACEIN]–> (r1);
MATCH (c1:Course {courseNr:"1"}),(r3:Room {roomName:"Alpha"}) MERGE (c1) –
[:TAKESPLACEIN] –>(r3) ;
MATCH (c1:Course {courseNr:"1"}),(r4:Room {roomName:"Beta"}) MERGE (c1) –
[:TAKESPLACEIN] –> (r4);
MATCH (c2:Course {courseNr:"2"}),(r2:Room {roomName:"Seminar C"}) MERGE (c2) –
[:TAKESPLACEIN] –> (r2);
MATCH (s1:Student{studentID:"1"}),(p1:Project {projectNr:"34"}) MERGE (s1) – [:WORKSON
{hours: "1"}] –> (p1);
MATCH (s1:Student{studentID:"1"}),(p2:Project {projectNr:"24"}) MERGE (s1) – [:WORKSON
{hours: "2"}] –> (p2);
MATCH (s2:Student{studentID:"2"}),(p1:Project {projectNr:"34"}) MERGE (s2) – [:WORKSON
{hours: "3"}] –> (p1);
MATCH (s2:Student{studentID:"2"}),(p2:Project {projectNr:"24"}) MERGE (s2) – [:WORKSON
{hours: "4"}] –> (p2);
MATCH (s2:Student{studentID:"2"}),(p3:Project {projectNr:"13"}) MERGE (s2) – [:WORKSON
{hours: "1"}] –> (p3);
```

```
MATCH (s2:Student{studentID:"2"}),(p4:Project {projectNr:"26"}) MERGE (s2) – [:WORKSON
{hours: "1"}] –> (p4);
MATCH (s3:Student{studentID:"3"}),(p1:Project {projectNr:"34"}) MERGE (s3) – [:WORKSON
{hours: "1"}] –> (p1);
MATCH (s3:Student{studentID:"3"}),(p2:Project {projectNr:"24"}) MERGE (s3) – [:WORKSON
{hours: "2"}] –> (p2);
MATCH (s3:Student{studentID:"3"}),(p4:Project {projectNr:"26"}) MERGE (s3) – [:WORKSON
{hours: "3"}] –> (p4);

MATCH (s1:Student {studentID:"1"}),(c1:Course {courseNr:"1"}) MERGE (s1)– [:ENROLLEDIN]
–> (c1);
MATCH (s2:Student {studentID:"2"}),(c1:Course {courseNr:"1"}) MERGE (s2) – [:ENROLLEDIN]
–> (c1);
MATCH (s3:Student {studentID:"3"}),(c2:Course {courseNr:"2"}) MERGE (s3) – [:ENROLLEDIN]
–> (c2);
MATCH (s4:Student {studentID:"4"}),(c1:Course {courseNr:"1"}) MERGE (s4) – [:ENROLLEDIN]
–> (c1);
```

Queries

Specify the following queries in Cypher and execute them in Neo4j.

In which rooms does course with course number "1" take place in? Retrieve the course name and the names of the rooms in which the course takes place.

```
MATCH (c:Course {courseNr:"1"}) -[r:TAKESPLACEIN]-> (r1:Room)
RETURN c,r1
```

How many hours and in which projects does student with student number "1" works on? Retrieve the first name of the student, the project the student works on and the corresponding number of hours worked on the project.

```
 MATCH (s:Student {studentID:"1"}) -[w:WORKSON] -> (p:Project)
RETURN s.firstName, w.hours, p.projectName
```

Which students and how many hours do they work on the project with project number "24"? Retrieve the project name, the last name of the student and the corresponding number of hours worked on the project.

```
MATCH (s:Student) -[w:WORKSON] -> (p:Project{projectNr:"24"})
RETURN s.firstName, w.hours, p.projectName
```

Which students work in which projects and how many hours? Retrieve the last name of the students, the name of the projects they work on, and the corresponding number of hours. Order the results by the last name of the students. Limit the results to four.

```
 MATCH (s:Student) -[w:WORKSON] -> (p:Project)
RETURN s.lastName, w.hours, p.projectName
ORDER BY s.lastName
LIMIT 4
```

Which students work on more than two projects and on how many projects exactly? Retrieve the last name of the students and the corresponding number of projects. Order the results by the number of projects.

```
MATCH (s:Student) -[w:WORKSON] -> (p:Project)

WITH count(p) as pnumber, s.lastName as lastname, s.firstName as firstname

WHERE pnumber >= 3

RETURN lastname,pnumber

ORDER By pnumber
```

Which students have the same last name and work on the same projects? Retrieve the first name of the students and the name of projects they share.

```
MATCH (s:Student) -[w:WORKSON] -> (p:Project) <-[:WORKSON]- (s1: Student)

WHERE s.lastName = s1.lastName

RETURN s.firstName,s1.firstName, p.projectName
```