Benjamin Katz
Neo4j

1. Retrieve all students, displaying their student id and name. Order the result set by ascending student id.

```
MATCH(s:Student)
RETURN s.stuId, s.name
ORDER BY s.stuID;
```

```
neo4j@neo4j> MATCH(s:Student)
             RETURN s.stuId, s.name
             ORDER BY s.stuID;
+--------------------------+
| s.stuId | s.name         |
+--------------------------+
| "S1001" | "Tom Smith"    |
| "S1002" | "Ann Chin"     |
| "S1005" | "Perry Lee"    |
| "S1010" | "Edward Burns" |
| "S1013" | "Owen McCarthy"|
| "S1015" | "Mary Jones"   |
| "S1020" | "Jane Rivera"  |
+--------------------------+

7 rows
```

2. Retrieve all rooms, displaying the room name. Order the result set by ascending room name.

```
MATCH(r:Room)
RETURN r.room
ORDER BY r.room;
```

```
neo4j@neo4j> MATCH(r:Room)
             RETURN r.room
             ORDER BY r.room;
+--------+
| r.room |
+--------+
| "H221" |
| "H225" |
| "M110" |
+--------+

3 rows
```

3. Retrieve all faculty, displaying their faculty id, name, and department. Order the result set by ascending name.

```
MATCH(f:Faculty)
RETURN f.facId, f.name, f.department
ORDER BY f.name;
```

```
ready to start consuming query after 30 ms, results
neo4j@neo4j> MATCH(f:Faculty)
             RETURN f.facId, f.name, f.department
             ORDER BY f.name;
+---------------------------------+
| f.facId | f.name   | f.department |
+---------------------------------+
| "F101"  | "Adams"  | "Art"        |
| "F110"  | "Byrne"  | "Math"       |
| "F115"  | "Smith"  | "History"    |
| "F221"  | "Smith"  | "CSC"        |
| "F105"  | "Tanaka" | "CSC"        |
+---------------------------------+

5 rows
```

4. Retrieve all class names, displaying the class number and class schedule. Order the result set by ascending class number.

```
MATCH(c:Class)
RETURN c.classNumber, c.schedule
ORDER BY c.classNumber;
```

```
neo4j@neo4j> MATCH(c:Class)
             RETURN c.classNumber, c.schedule
             ORDER BY c.classNumber;
+---------------------------+
| c.classNumber | c.schedule |
+---------------------------+
| "ART103A"     | "MWF9"     |
| "CSC201A"     | "uThF10"   |
| "CSC203A"     | "MThF12"   |
| "HST205A"     | "MWF11"    |
| "MTH101B"     | "MTuTh9"   |
| "MTH103C"     | "MWF11"    |
+---------------------------+

6 rows
```

5. How many students are Math majors?. The result-set attribute must be named numberMathMajors.

MATCH(s:Student)
WHERE s.major = 'Math'
RETURN count(s.stuId) AS numberMathMajors;

```
neo4j@neo4j> MATCH(s:Student)
             WHERE s.major = 'Math'
             RETURN count(s.stuId) AS numberMathMajors;
+------------------+
| numberMathMajors |
+------------------+
| 3                |
+------------------+
```

6. Which students are enrolled in classes taught by Adams?. The resultset should include all information about the student, the class, and

the person teaching the course, ordered by ascending student id.

```
MATCH(s)-[:ENROLLED_IN]-(c)-[:TAUGHT_BY]-(f)
WHERE f.name = 'Adams'
RETURN s, c, f
ORDER BY s.stuId;
```

```
neo4j@neo4j> MATCH(s)-[:ENROLLED_IN]-(c)-[:TAUGHT_BY]-(f)
             WHERE f.name = 'Adams'
             RETURN s, c, f
             ORDER BY s.stuId;
+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| s                                                                          | c                                                    | f                                                                                  |
+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| (:Student {name: "Tom Smith", stuId: "S1001", major: "History", credit: 90}) | (:Class {schedule: "MWF9", classNumber: "ART103A"}) | (:Faculty {name: "Adams", facId: "F101", rank: "Professor", department: "Art"}) |
| (:Student {name: "Ann Chin", stuId: "S1002", major: "Math", credit: 36})     | (:Class {schedule: "MWF9", classNumber: "ART103A"}) | (:Faculty {name: "Adams", facId: "F101", rank: "Professor", department: "Art"}) |
| (:Student {name: "Edward Burns", stuId: "S1010", major: "Art", credit: 63})  | (:Class {schedule: "MWF9", classNumber: "ART103A"}) | (:Faculty {name: "Adams", facId: "F101", rank: "Professor", department: "Art"}) |
+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+

3 rows
```

7. Which entities have any relationship (directed or undirected) to Class MTH101B?. Each tuple in the result set must be ordered as the unknown entity, followed by the Class entity.

```
MATCH(e)-[r]-(c:Class{classNumber:'MTH101B'})
RETURN e, c;
```

```
neo4j@neo4j> MATCH(e)-[r]-(c:Class{classNumber:'MTH101B'})
             RETURN e, c;
+-------------------------------------------------------------------------------------------------------------------------------------------+
| e                                                                          | c                                                      |
+-------------------------------------------------------------------------------------------------------------------------------------------+
| (:Faculty {name: "Byrne", facId: "F110", rank: "Assistant", department: "Math"}) | (:Class {schedule: "MTuTh9", classNumber: "MTH101B"}) |
| (:Room {room: "H225"})                                                     | (:Class {schedule: "MTuTh9", classNumber: "MTH101B"}) |
+-------------------------------------------------------------------------------------------------------------------------------------------+

2 rows
```

8. Return all students who earned an 'A' in a given course. Each tuple in the result set should include the student and the course in which they earned an 'A'. Order the result-set by ascending student id.

```
MATCH(s)-[:ENROLLED_IN{grade: 'A'}]-(c:Class)
RETURN s, c
```

ORDER BY s.stuId;

```
neo4j@neo4j> MATCH(s)-[:ENROLLED_IN{grade: 'A'}]-(c:Class)
             RETURN s, c
             ORDER BY s.stuId;
+-----------------------------------------------------------------------------------------------------------------------------------------------------+
| s                                                                                | c                                                                |
+-----------------------------------------------------------------------------------------------------------------------------------------------------+
| (:Student {name: "Tom Smith", stuId: "S1001", major: "History", credit: 90}) | (:Class {schedule: "MWF9", classNumber: "ART103A"}) |
+-----------------------------------------------------------------------------------------------------------------------------------------------------+

1 row
```

9. Return all entities (with all of their information) that have any relationship to a Room entity. The result-set should be a pair with the entity as the first part of the tuple, and the Room entity as the second. The result-set should be ordered by ascending room number.

MATCH(e)-[rel]-(r:Room)
RETURN e, r
ORDER BY r.room;

```
neo4j@neo4j> MATCH(e)-[rel]-(r:Room)
             RETURN e, r
             ORDER BY r.room;
+-----------------------------------------------------------------------------------+
| e                                                    | r                          |
+-----------------------------------------------------------------------------------+
| (:Class {schedule: "MWF11", classNumber: "HST205A"})  | (:Room {room: "H221"}) |
| (:Class {schedule: "MWF9", classNumber: "ART103A"})   | (:Room {room: "H221"}) |
| (:Class {schedule: "MWF11", classNumber: "MTH103C"})  | (:Room {room: "H225"}) |
| (:Class {schedule: "MTuTh9", classNumber: "MTH101B"}) | (:Room {room: "H225"}) |
| (:Class {schedule: "MThF12", classNumber: "CSC203A"}) | (:Room {room: "M110"}) |
| (:Class {schedule: "uThF10", classNumber: "CSC201A"}) | (:Room {room: "M110"}) |
+-----------------------------------------------------------------------------------+

6 rows
```