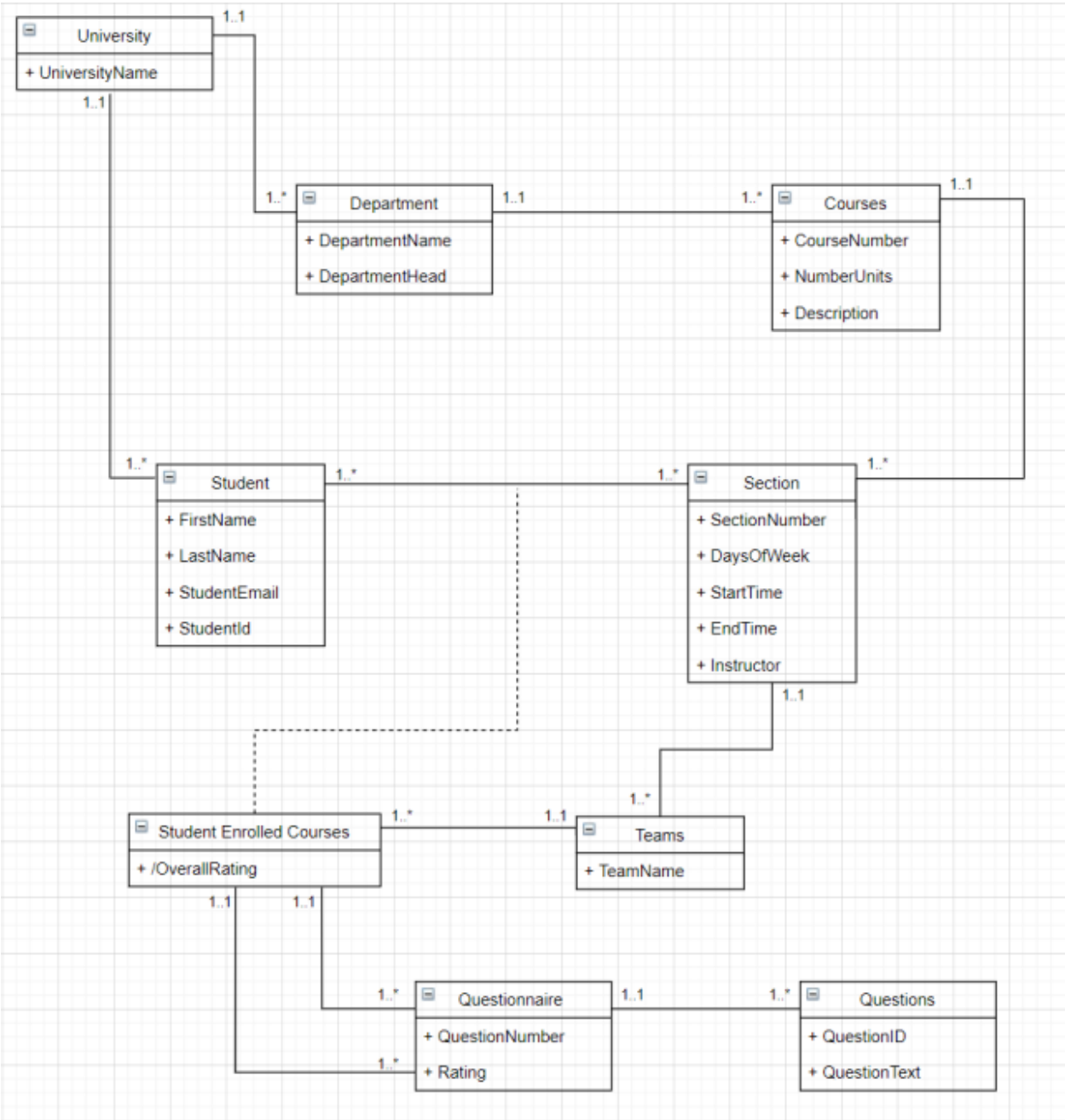
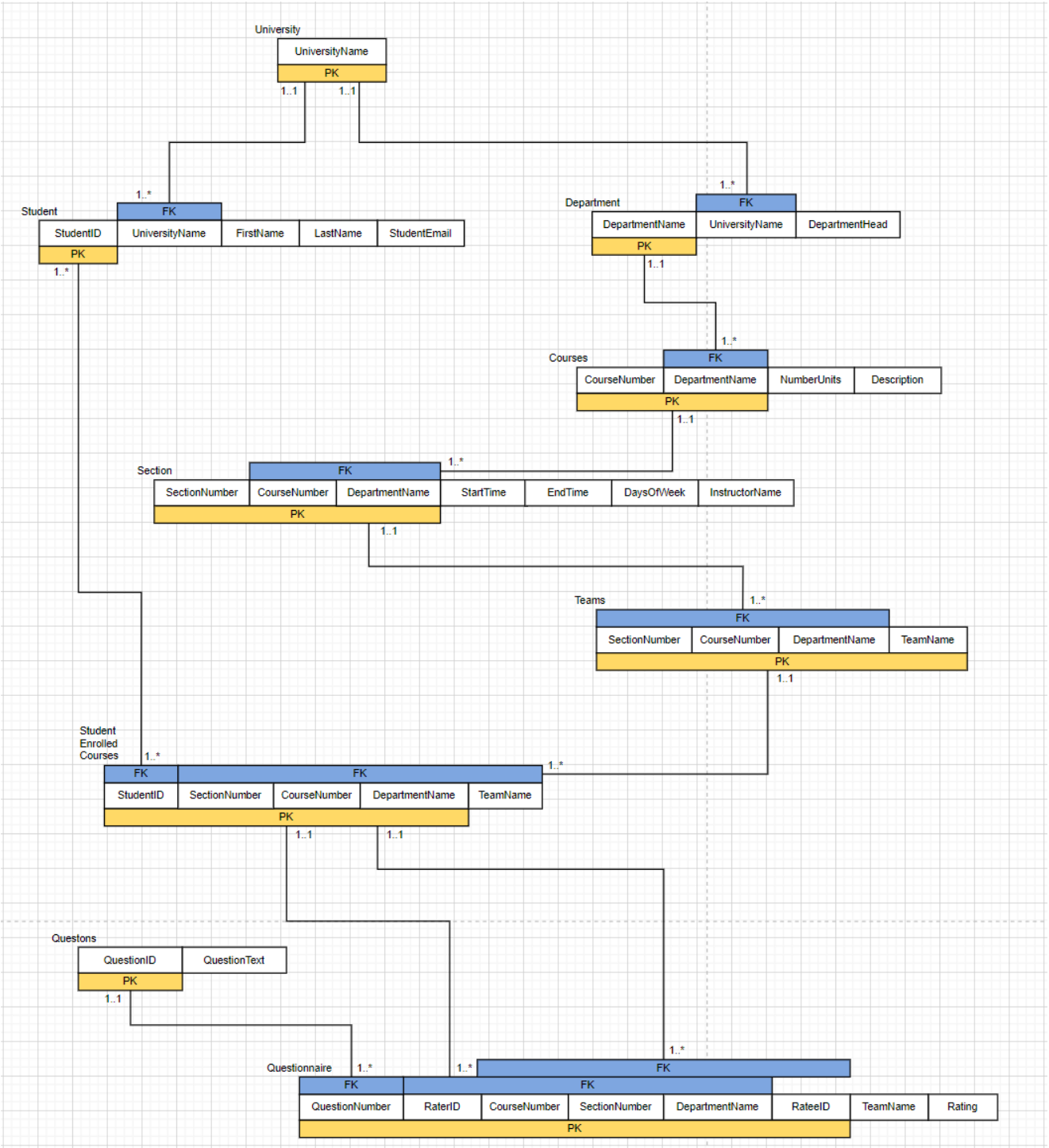
UML AND RELATIONAL SCHEME

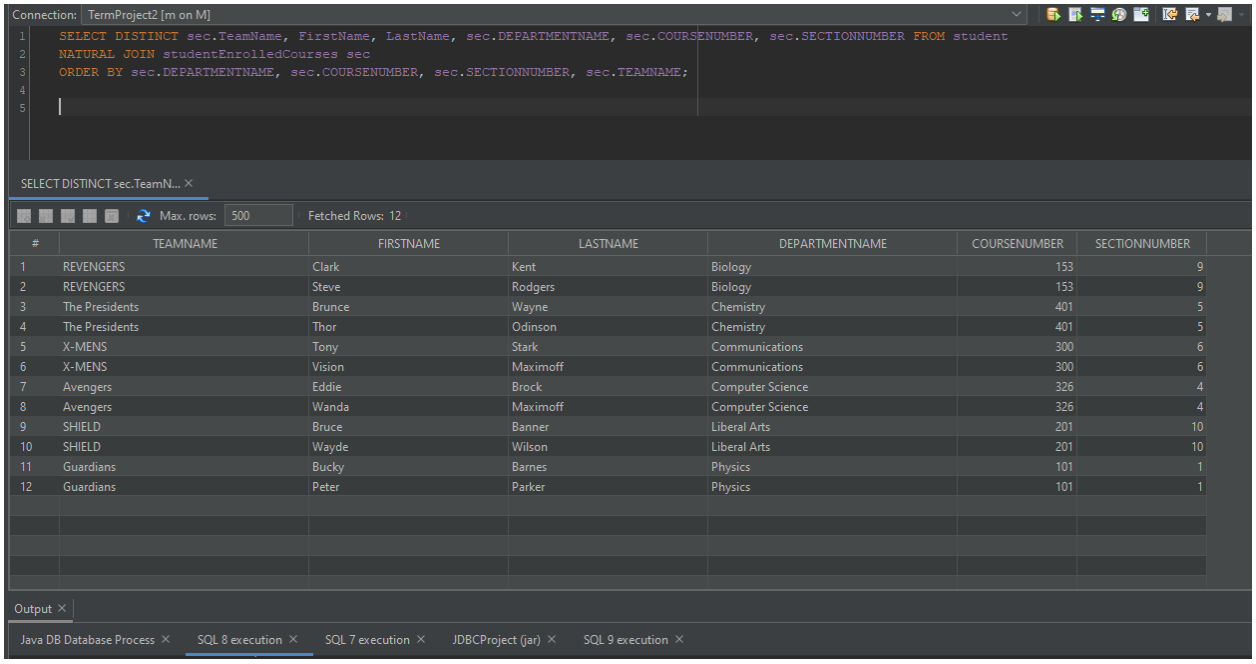




QUERIES AND RESULTS

1. For each section within each course, list the teams and the students who are in each team. List one row of output per team. Order them by department, course number, section number, then team name.

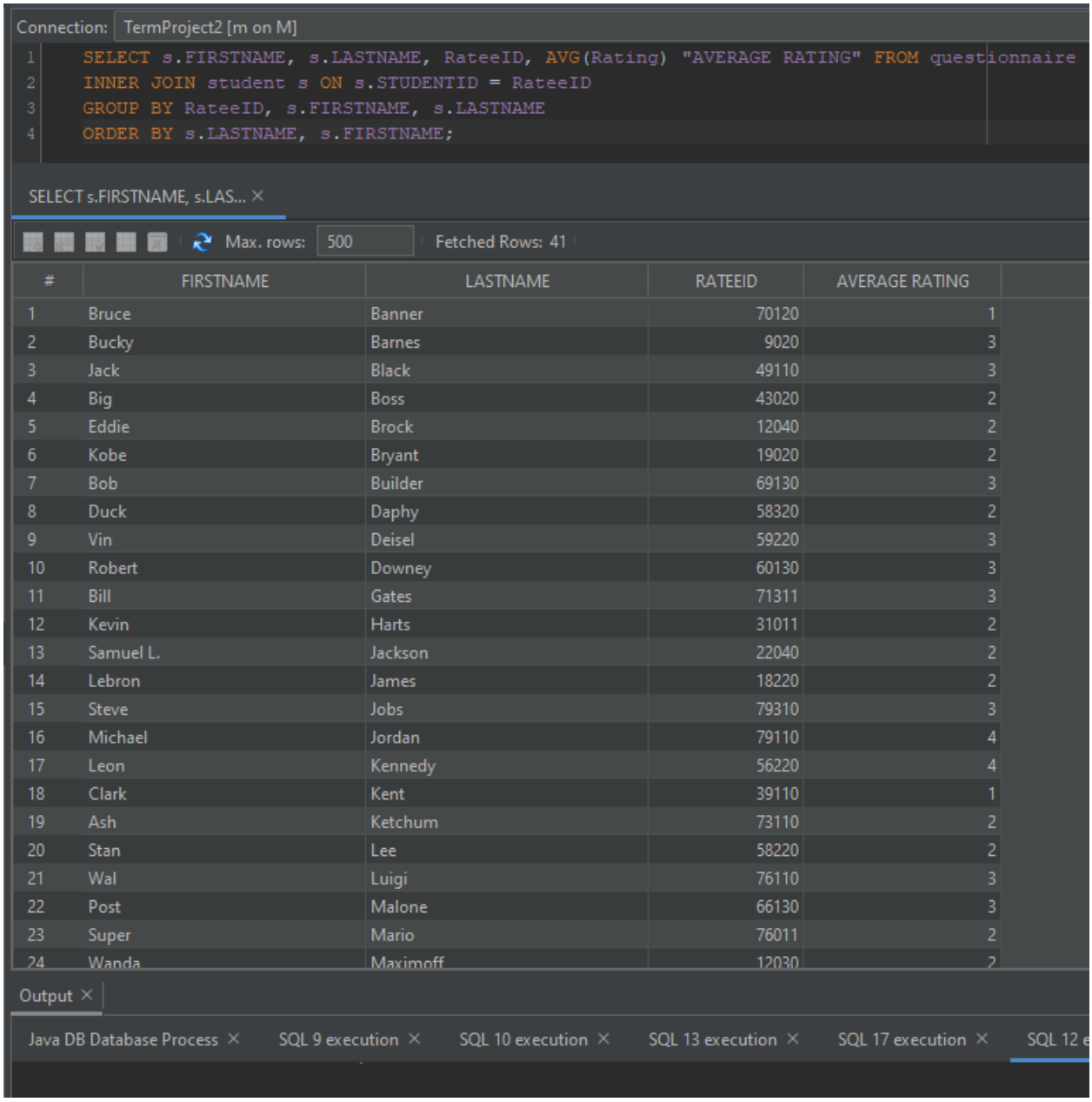
SELECT DISTINCT sec.TeamName, FirstName, LastName, sec.DEPARTMENTNAME, sec.COURSENUMBER, sec.SECTIONNUMBER FROM student
NATURAL JOIN studentEnrolledCourses sec
ORDER BY sec.DEPARTMENTNAME, sec.COURSENUMBER, sec.SECTIONNUMBER, sec.TEAMNAME;



#	TEAMNAME	FIRSTNAME	LASTNAME	DEPARTMENTNAME	COURSENUMBER	SECTIONNUMBER
1	REVENGERS	Clark	Kent	Biology	153	9
2	REVENGERS	Steve	Rodgers	Biology	153	9
3	The Presidents	Brunce	Wayne	Chemistry	401	5
4	The Presidents	Thor	Odinson	Chemistry	401	5
5	X-MENS	Tony	Stark	Communications	300	6
6	X-MENS	Vision	Maximoff	Communications	300	6
7	Avengers	Eddie	Brock	Computer Science	326	4
8	Avengers	Wanda	Maximoff	Computer Science	326	4
9	SHIELD	Bruce	Banner	Liberal Arts	201	10
10	SHIELD	Wayde	Wilson	Liberal Arts	201	10
11	Guardians	Bucky	Barnes	Physics	101	1
12	Guardians	Peter	Parker	Physics	101	1

2. For each student in each team, list the average of the scores that that student received **from their peers**. Students pretty much always give themselves a 5 across the board, so do not consider that score. Produce one row of output per student per course section.

SELECT s.FIRSTNAME, s.LASTNAME, RateeID, AVG(Rating) "AVERAGE RATING"
FROM questionnaire
INNER JOIN student s ON s.STUDENTID = RateeID
GROUP BY RateeID, s.FIRSTNAME, s.LASTNAME
ORDER BY s.LASTNAME, s.FIRSTNAME;



3. Produce a report that shows all the teams whose number of students is > 2 higher or lower than the average number of students in the teams within that particular section

   SELECT TeamName, COUNT(StudentID) FROM StudentEnrolledCourses
   GROUP BY TeamName
   HAVING COUNT(StudentID) > 2;

```
6   SELECT TeamName, COUNT(StudentID) FROM StudentEnrolledCourses
7   GROUP BY TeamName
8   HAVING COUNT(StudentID) > 2;
```

SELECT TeamName, COUNT(St... ×

Max. rows: 500 Fetched Rows: 11

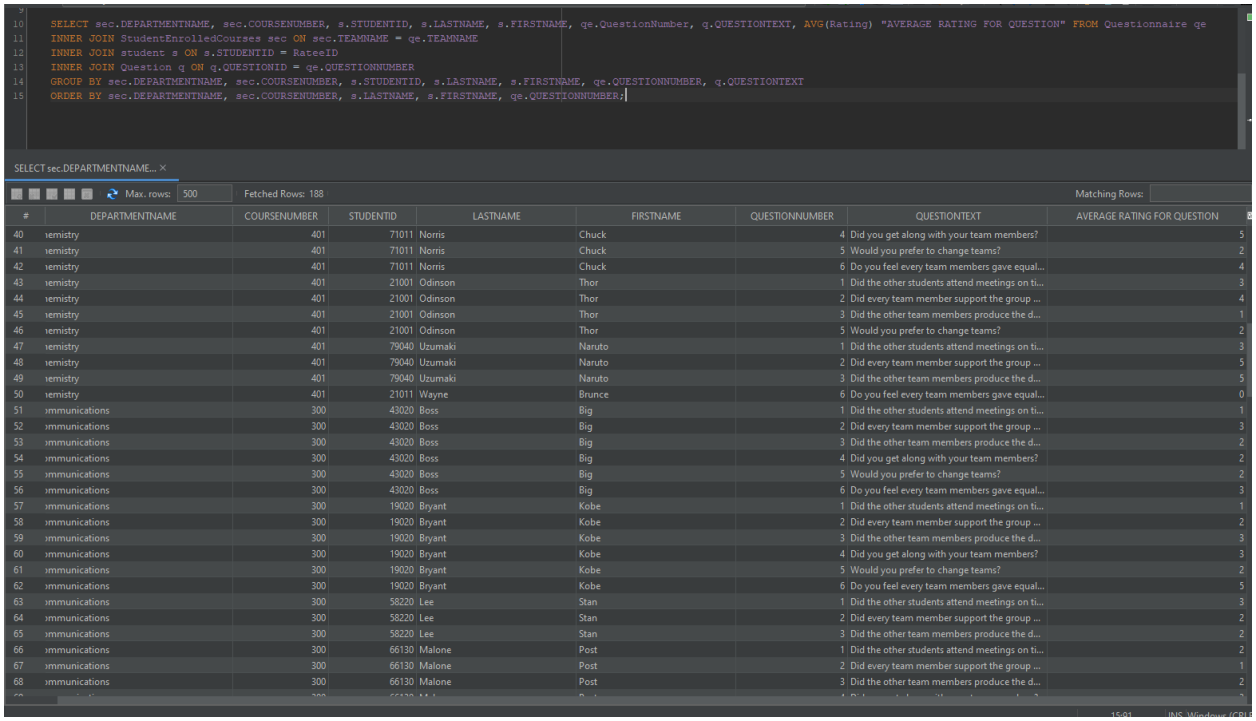
#	TEAMNAME	2
1	Avengers	3
2	Clippers	3
3	Guardians	3
4	Heat	3
5	Knicks	3
6	Lakers	3
7	REVENGERS	3
8	Raptors	3
9	SHIELD	3
10	The Presidents	3
11	X-MENS	3

Output ×

4. Report the average value that the student received from their peers for each rating dimension within a given course section. One row for each rating dimension for each student within a given course section. Order by department, course number, student last name, student first name, and dimension text

   SELECT sec.DEPARTMENTNAME, sec.COURSENUMBER, s.LASTNAME, s.FIRSTNAME, qe.QuestionNumber, q.QUESTIONTEXT, AVG(Rating) "AVERAGE RATING FOR QUESTION" FROM Questionnaire qe
   INNER JOIN StudentEnrolledCourses sec ON sec.TEAMNAME = qe.TEAMNAME
   INNER JOIN student s ON s.STUDENTID = RateeID

INNER JOIN Question q ON q.QUESTIONID = qe.QUESTIONNUMBER
GROUP BY sec.DEPARTMENTNAME, sec.COURSENUMBER, s.LASTNAME, s.FIRSTNAME, qe.QUESTIONNUMBER, q.QUESTIONTEXT
ORDER BY sec.DEPARTMENTNAME, sec.COURSENUMBER, s.LASTNAME, s.FIRSTNAME, qe.QUESTIONNUMBER;



```
SELECT sec.DEPARTMENTNAME, sec.COURSENUMBER, s.STUDENTID, s.LASTNAME, s.FIRSTNAME, qe.QuestionNumber, q.QUESTIONTEXT, AVG(Rating) "AVERAGE RATING FOR QUESTION" FROM Questionnaire qe
INNER JOIN StudentEnrolledCourses sec ON sec.TEAMNAME = qe.TEAMNAME
INNER JOIN student s ON s.STUDENTID = RateeID
INNER JOIN Question q ON q.QUESTIONID = qe.QUESTIONNUMBER
GROUP BY sec.DEPARTMENTNAME, sec.COURSENUMBER, s.STUDENTID, s.LASTNAME, s.FIRSTNAME, qe.QUESTIONNUMBER, q.QUESTIONTEXT
ORDER BY sec.DEPARTMENTNAME, sec.COURSENUMBER, s.LASTNAME, s.FIRSTNAME, qe.QUESTIONNUMBER;
```

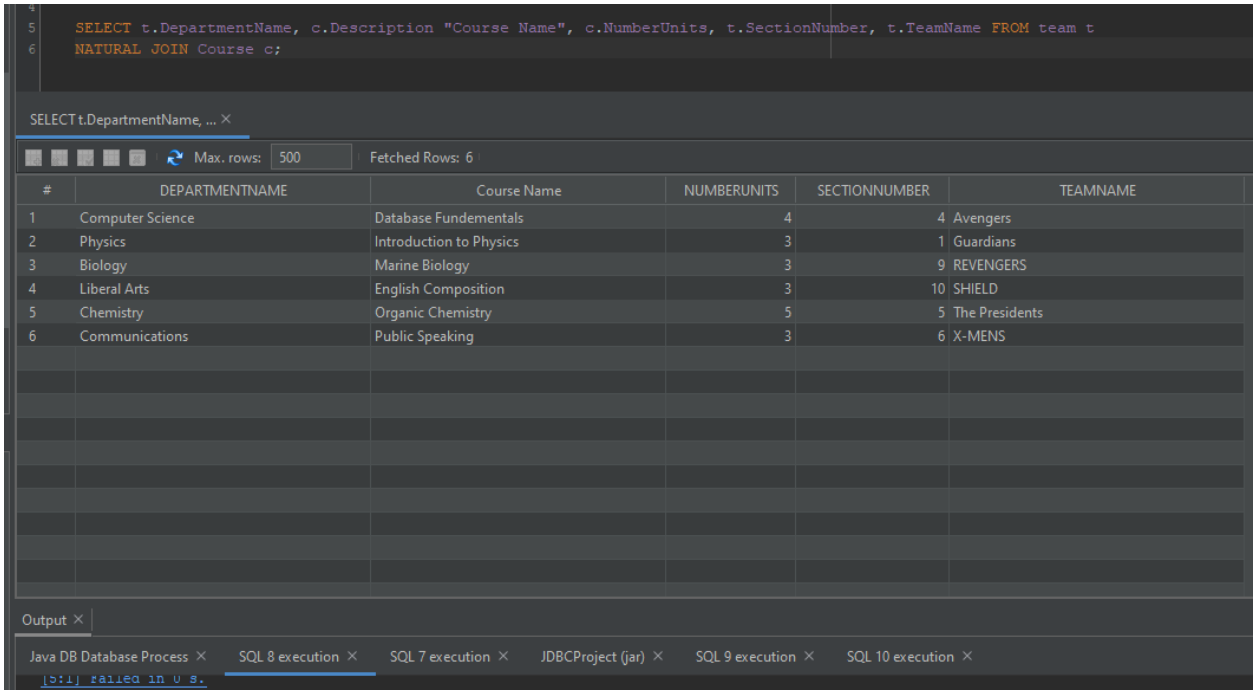
SELECT sec.DEPARTMENTNAME... ×

Max. rows: 500 Fetched Rows: 188 Matching Rows:

#	DEPARTMENTNAME	COURSENUMBER	STUDENTID	LASTNAME	FIRSTNAME	QUESTIONNUMBER	QUESTIONTEXT	AVERAGE RATING FOR QUESTION
40	emistry	401	71011	Norris	Chuck	4	Did you get along with your team members?	5
41	emistry	401	71011	Norris	Chuck	5	Would you prefer to change teams?	2
42	emistry	401	71011	Norris	Chuck	6	Do you feel every team members gave equal...	4
43	emistry	401	21001	Odinson	Thor	1	Did the other students attend meetings on ti...	3
44	emistry	401	21001	Odinson	Thor	2	Did every team member support the group ...	4
45	emistry	401	21001	Odinson	Thor	3	Did the other team members produce the d...	1
46	emistry	401	21001	Odinson	Thor	5	Would you prefer to change teams?	2
47	emistry	401	79040	Uzumaki	Naruto	1	Did the other students attend meetings on ti...	3
48	emistry	401	79040	Uzumaki	Naruto	2	Did every team member support the group ...	5
49	emistry	401	79040	Uzumaki	Naruto	3	Did the other team members produce the d...	5
50	emistry	401	21011	Wayne	Brunce	6	Do you feel every team members gave equal...	0
51	mmunications	300	43020	Boss	Big	1	Did the other students attend meetings on ti...	1
52	mmunications	300	43020	Boss	Big	2	Did every team member support the group ...	3
53	mmunications	300	43020	Boss	Big	3	Did the other team members produce the d...	2
54	mmunications	300	43020	Boss	Big	4	Did you get along with your team members?	2
55	mmunications	300	43020	Boss	Big	5	Would you prefer to change teams?	2
56	mmunications	300	43020	Boss	Big	6	Do you feel every team members gave equal...	3
57	mmunications	300	19020	Bryant	Kobe	1	Did the other students attend meetings on ti...	1
58	mmunications	300	19020	Bryant	Kobe	2	Did every team member support the group ...	2
59	mmunications	300	19020	Bryant	Kobe	3	Did the other team members produce the d...	3
60	mmunications	300	19020	Bryant	Kobe	4	Did you get along with your team members?	3
61	mmunications	300	19020	Bryant	Kobe	5	Would you prefer to change teams?	2
62	mmunications	300	19020	Bryant	Kobe	6	Do you feel every team members gave equal...	5
63	mmunications	300	58220	Lee	Stan	1	Did the other students attend meetings on ti...	3
64	mmunications	300	58220	Lee	Stan	2	Did every team member support the group ...	2
65	mmunications	300	58220	Lee	Stan	3	Did the other team members produce the d...	2
66	mmunications	300	66130	Malone	Post	1	Did the other students attend meetings on ti...	2
67	mmunications	300	66130	Malone	Post	2	Did every team member support the group ...	1
68	mmunications	300	66130	Malone	Post	3	Did the other team members produce the d...	2

5. For each team, list the department name, the course name, the course units, the section number, the team name,

   SELECT t.DepartmentName, c.Description "Course Name", c.NumberUnits, t.SectionNumber, t.TeamName FROM team t
   NATURAL JOIN Course c;



```
SELECT t.DepartmentName, c.Description "Course Name", c.NumberUnits, t.SectionNumber, t.TeamName FROM team t
NATURAL JOIN Course c;
```

SELECT t.DepartmentName, ... ×

Max. rows: 500 Fetched Rows: 6

#	DEPARTMENTNAME	Course Name	NUMBERUNITS	SECTIONNUMBER	TEAMNAME
1	Computer Science	Database Fundementals	4	4	Avengers
2	Physics	Introduction to Physics	3	1	Guardians
3	Biology	Marine Biology	3	9	REVENGERS
4	Liberal Arts	English Composition	3	10	SHIELD
5	Chemistry	Organic Chemistry	5	5	The Presidents
6	Communications	Public Speaking	3	6	X-MENS

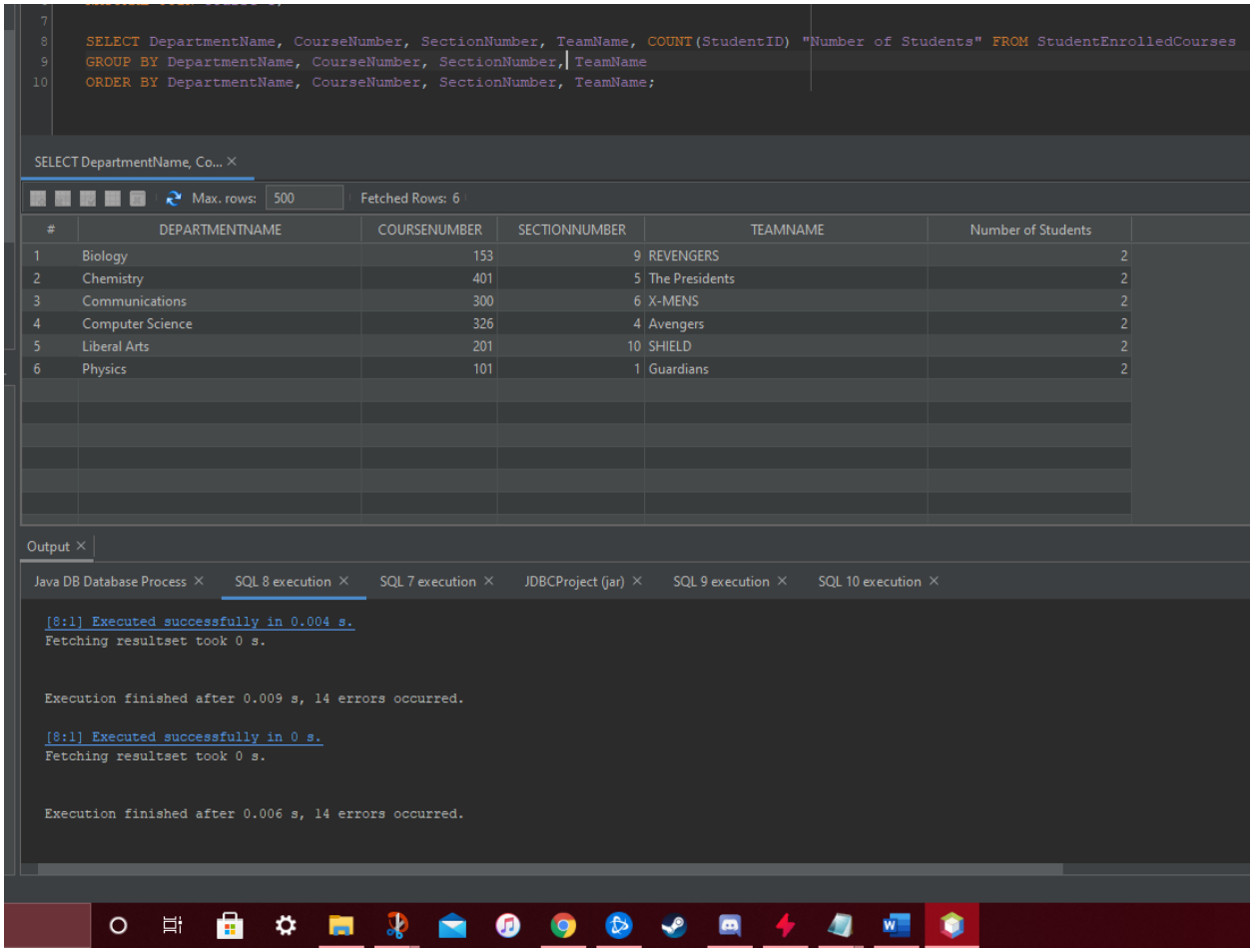
Output ×

Java DB Database Process × SQL 8 execution × SQL 7 execution × JDBCProject (jar) × SQL 9 execution × SQL 10 execution ×

[5:1] failed in 0 s.

6. For each team, report the department name, the course number, the section number, the team name, and the number of students in the team. Order by department name, course number, section number and team name.

   SELECT DepartmentName, CourseNumber, SectionNumber, TeamName, COUNT(StudentID) "Number of Students" FROM StudentEnrolledCourses GROUP BY DepartmentName, CourseNumber, SectionNumber, TeamName ORDER BY DepartmentName, CourseNumber, SectionNumber, TeamName;


SELECT DepartmentName, CourseNumber, SectionNumber, TeamName, COUNT(StudentID) "Number of Students" FROM StudentEnrolledCourses
GROUP BY DepartmentName, CourseNumber, SectionNumber, TeamName
ORDER BY DepartmentName, CourseNumber, SectionNumber, TeamName;
SELECT DepartmentName, Co...
Max. rows: 500
Fetched Rows: 6
DEPARTMENTNAME COURSENUMBER SECTIONNUMBER TEAMNAME Number of Students
1 Biology 153 9 REVENGERS 2
2 Chemistry 401 5 The Presidents 2
3 Communications 300 6 X-MENS 2
4 Computer Science 326 4 Avengers 2
5 Liberal Arts 201 10 SHIELD 2
6 Physics 101 1 Guardians 2
Output
Java DB Database Process
SQL 8 execution
SQL 7 execution
JDBCProject (jar)
SQL 9 execution
SQL 10 execution
[8:1] Executed successfully in 0.004 s.
Fetching resultset took 0 s.
Execution finished after 0.009 s, 14 errors occurred.
[8:1] Executed successfully in 0 s.
Fetching resultset took 0 s.
Execution finished after 0.006 s, 14 errors occurred.