CS 410 Medium-stakes Assignment 3: Writing SQL Queries

Nick Alexander

March 30, 2015

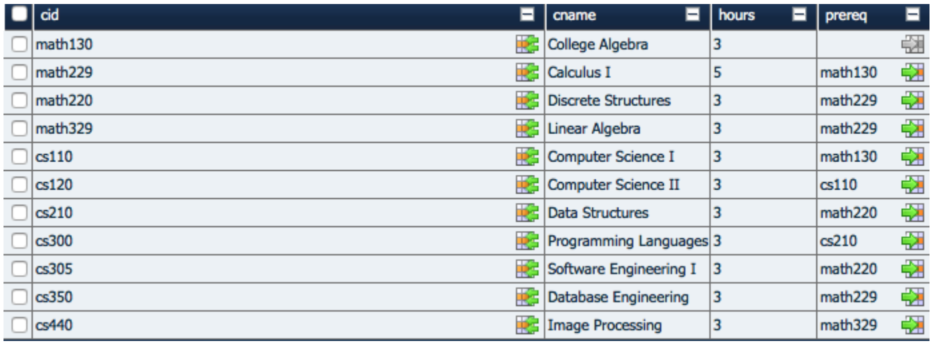
Contents

1 SQL Queries

In addition to the SQL queries, also show the results retrieved from the PostgreSQL database. You may want to use \begin{verbatim} and \end{verbatim} environment and include the SQL query and retrieved results between \begin{verbatim} and \end{verbatim}.

1. Retrieve all information about courses.

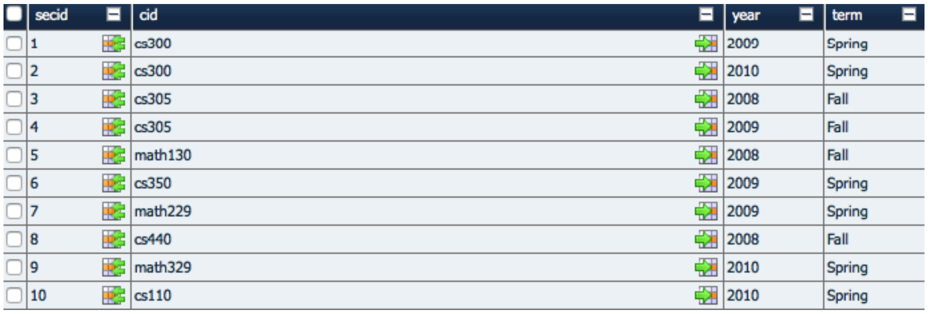
```
SELECT *
FROM COURSE
```



cid	cname	hours	prereq
math130	College Algebra	3	
math229	Calculus I	5	math130
math220	Discrete Structures	3	math229
math329	Linear Algebra	3	math229
cs110	Computer Science I	3	math130
cs120	Computer Science II	3	cs110
cs210	Data Structures	3	math220
cs300	Programming Languages	3	cs210
cs305	Software Engineering I	3	math220
cs350	Database Engineering	3	math229
cs440	Image Processing	3	math329

2. Retrieve section id, course id, year, and term information for all sections.

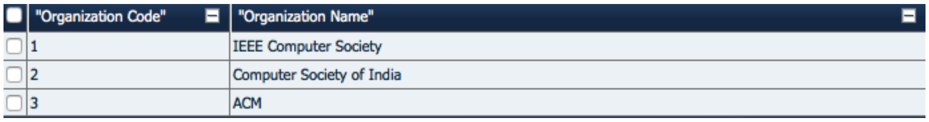
```
SELECT secid, cid, year, term
FROM SECTION
```



secid	cid	year	term
1	cs300	2009	Spring
2	cs300	2010	Spring
3	cs305	2008	Fall
4	cs305	2009	Fall
5	math130	2008	Fall
6	cs350	2009	Spring
7	math229	2009	Spring
8	cs440	2008	Fall
9	math329	2010	Spring
10	cs110	2010	Spring

3. Retrieve organization id and organization name for all organizations. Rename organization id as "Organization Code" and organization name as "Organization Title" in the result set.

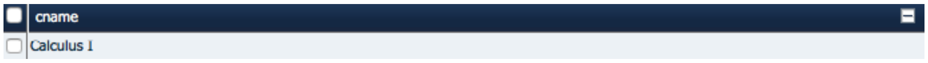
```
SELECT oid AS "Organization Code", oname AS "Organization Name"
FROM ORG
```



"Organization Code"	"Organization Name"
1	IEEE Computer Society
2	Computer Society of India
3	ACM

4. List course name of courses that are either 2.0 credit hours or 5.0 credit hours.

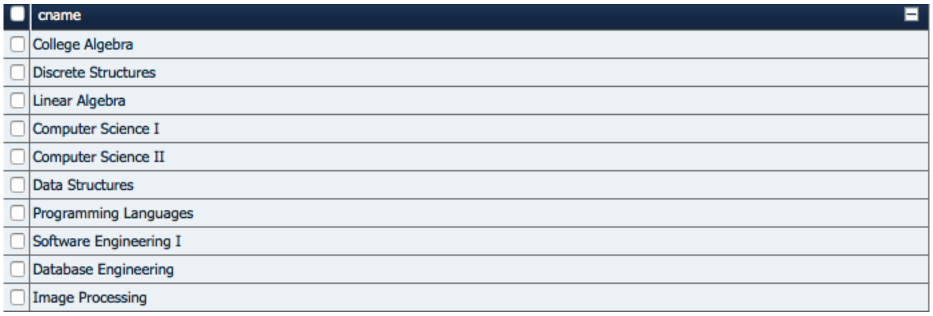
```
SELECT cname
FROM COURSE
WHERE hours=2.0 OR hours=5.0
```



cname
Calculus I

5. List course name of courses that are not 5.0 credit hours.

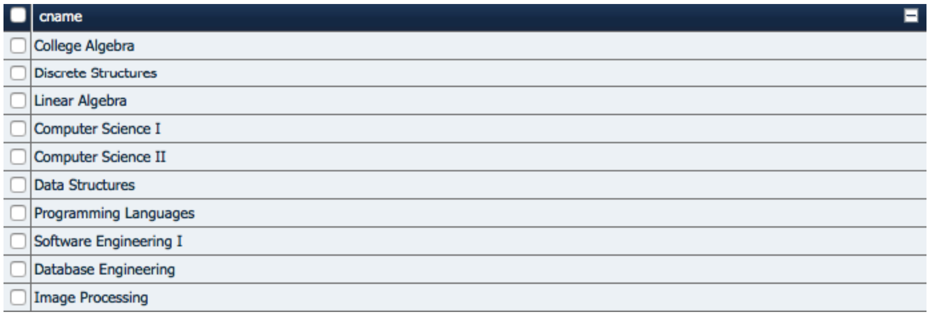
```
SELECT cname
FROM COURSE
WHERE NOT hours=5.0
```



cname
College Algebra
Discrete Structures
Linear Algebra
Computer Science I
Computer Science II
Data Structures
Programming Languages
Software Engineering I
Database Engineering
Image Processing

6. List course name of courses whose credit hours are greater than or equal to 2.0 and less than or equal to 4.0.

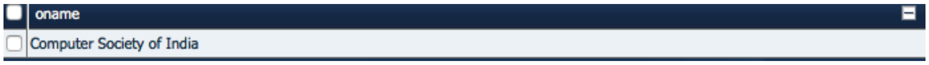
```
SELECT cname
FROM COURSE
WHERE hours>=2.0 AND hours<=4.0
```



cname
College Algebra
Discrete Structures
Linear Algebra
Computer Science I
Computer Science II
Data Structures
Programming Languages
Software Engineering I
Database Engineering
Image Processing

7. List organization name whose annual fee is less than $30.0.

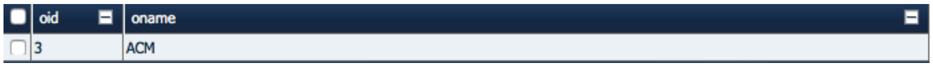
```
SELECT oname
FROM ORG
WHERE fee<30.0
```



□	oname
□	Computer Society of India

8. List organization id and organization name of organizations that charge more than $50.0 for annual membership fees.

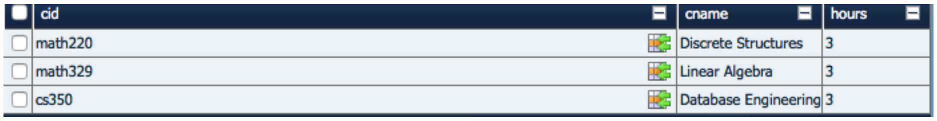
```
SELECT oid, oname
FROM ORG
WHERE fee>50.0
```



□	oid	oname
□	3	ACM

9. List course id, course name, and credit hours for courses for which math229 is prerequisite.

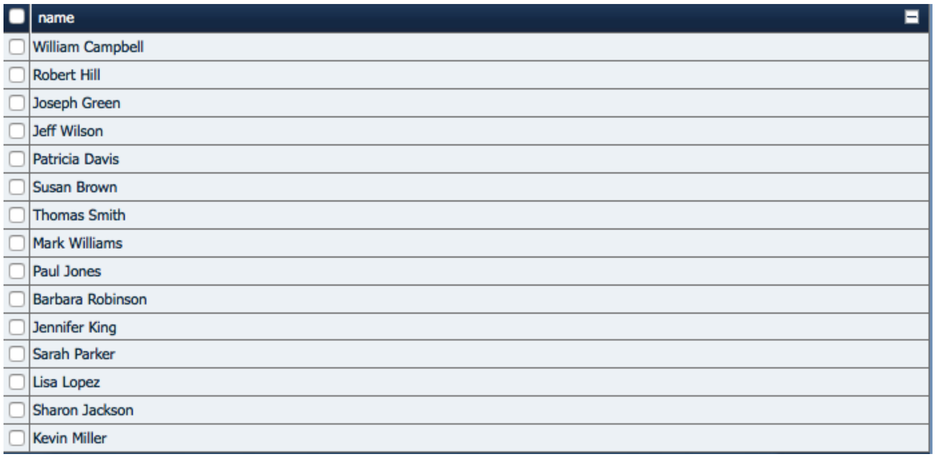
```
SELECT cid, cname, hours
FROM COURSE
WHERE prereq=’math229’
```



□	cid	cname	hours
□	math220	Discrete Structures	3
□	math329	Linear Algebra	3
□	cs350	Database Engineering	3

10. List student id and student name for all students. Concatenate fist name and last name with a space in between them and show this name under column titled as Name in the result set.

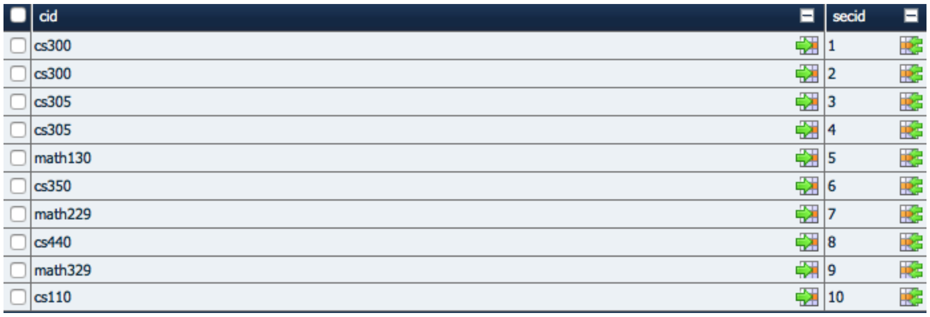
```
SELECT fname || ’ ’ || lname AS Name
FROM STUDENT
```



□	name
□	William Campbell
□	Robert Hill
□	Joseph Green
□	Jeff Wilson
□	Patricia Davis
□	Susan Brown
□	Thomas Smith
□	Mark Williams
□	Paul Jones
□	Barbara Robinson
□	Jennifer King
□	Sarah Parker
□	Lisa Lopez
□	Sharon Jackson
□	Kevin Miller

11. List course ids of courses that were offered at least once.

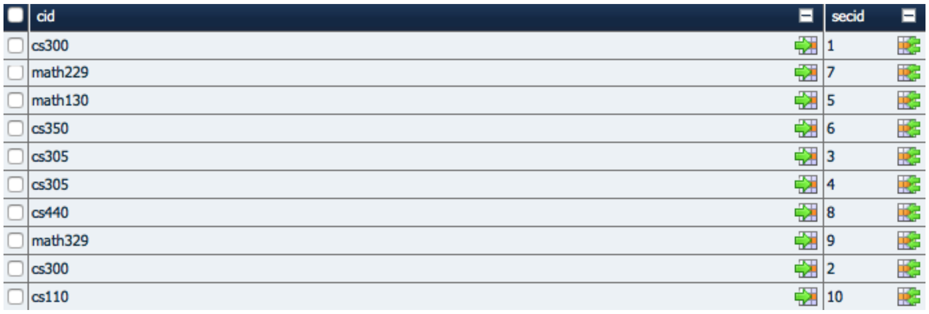
```
SELECT cid
FROM SECTION
```



☐	cid	secid
☐	cs300	1
☐	cs300	2
☐	cs305	3
☐	cs305	4
☐	math130	5
☐	cs350	6
☐	math229	7
☐	cs440	8
☐	math329	9
☐	cs110	10

12. List distinct courses ids of courses that were offered at least once.

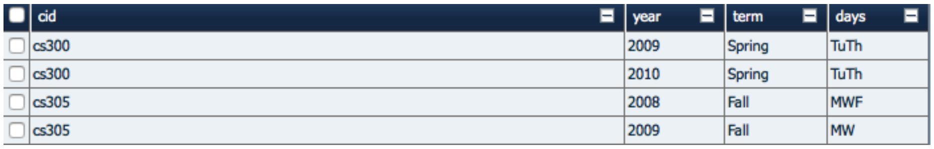
```
SELECT DISTINCT cid
FROM SECTION
```



☐	cid	secid
☐	cs300	1
☐	math229	7
☐	math130	5
☐	cs350	6
☐	cs305	3
☐	cs305	4
☐	cs440	8
☐	math329	9
☐	cs300	2
☐	cs110	10

13. List course id, year, term, and days for courses that were offered at 11.00 A.M.

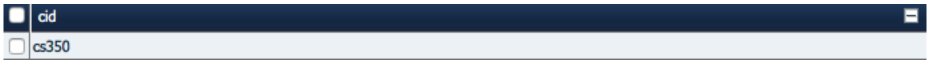
```
SELECT cid, year, term, days
FROM SECTION
WHERE EXTRACT(HOUR FROM stime)=11
```



☐	cid	year	term	days
☐	cs300	2009	Spring	TuTh
☐	cs300	2010	Spring	TuTh
☐	cs305	2008	Fall	MWF
☐	cs305	2009	Fall	MW

14. List course id for courses whose class meeting time ends at 4.45 P.M.

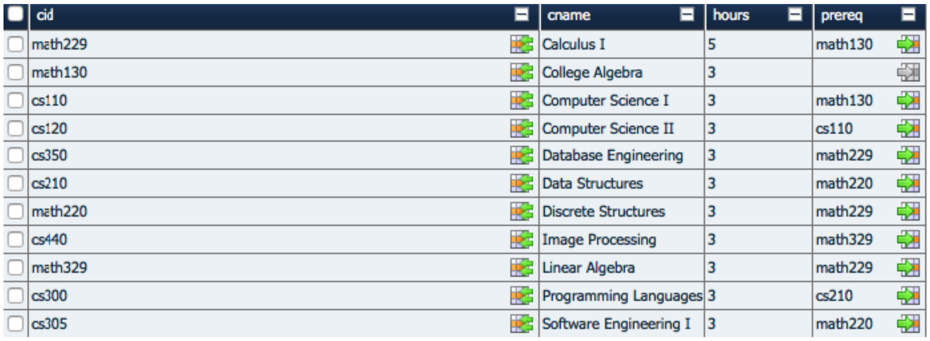
```
SELECT cid
FROM SECTION
WHERE EXTRACT(HOUR FROM etime)=16 AND EXTRACT(MINUTE FROM etime)=45
```



□	cid
□	cs350

15. List all information about courses sorted in decreasing order on course name.

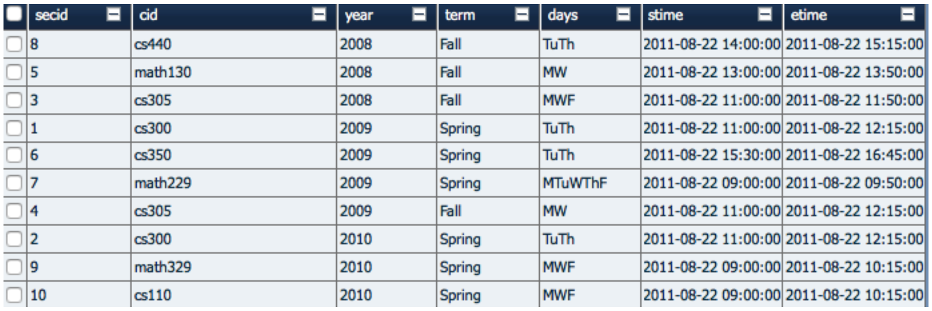
```
SELECT *
FROM COURSE
ORDER BY cname ASC;
```



□	cid	cname	hours	prereq
□	math229	Calculus I	5	math130
□	math130	College Algebra	3	
□	cs110	Computer Science I	3	math130
□	cs120	Computer Science II	3	cs110
□	cs350	Database Engineering	3	math229
□	cs210	Data Structures	3	math220
□	math220	Discrete Structures	3	math229
□	cs440	Image Processing	3	math329
□	math329	Linear Algebra	3	math229
□	cs300	Programming Languages	3	cs210
□	cs305	Software Engineering I	3	math220

16. List all information about section, first sorted in increasing order on year and then in decreasing order on term.

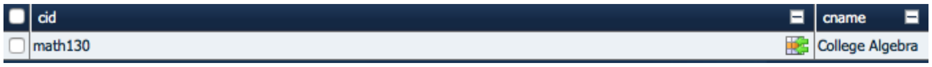
```
SELECT *
FROM (
    SELECT *
    FROM SECTION
    ORDER BY term DESC
) AS TMP
ORDER BY year ASC;
```



□	secid	cid	year	term	days	stime	etime
□	8	cs440	2008	Fall	TuTh	2011-08-22 14:00:00	2011-08-22 15:15:00
□	5	math130	2008	Fall	MW	2011-08-22 13:00:00	2011-08-22 13:50:00
□	3	cs305	2008	Fall	MWF	2011-08-22 11:00:00	2011-08-22 11:50:00
□	1	cs300	2009	Spring	TuTh	2011-08-22 11:00:00	2011-08-22 12:15:00
□	6	cs350	2009	Spring	TuTh	2011-08-22 15:30:00	2011-08-22 16:45:00
□	7	math229	2009	Spring	MTuWThF	2011-08-22 09:00:00	2011-08-22 09:50:00
□	4	cs305	2009	Fall	MW	2011-08-22 11:00:00	2011-08-22 12:15:00
□	2	cs300	2010	Spring	TuTh	2011-08-22 11:00:00	2011-08-22 12:15:00
□	9	math329	2010	Spring	MWF	2011-08-22 09:00:00	2011-08-22 10:15:00
□	10	cs110	2010	Spring	MWF	2011-08-22 09:00:00	2011-08-22 10:15:00

17. List course id and course name for courses that don't have prerequisites.

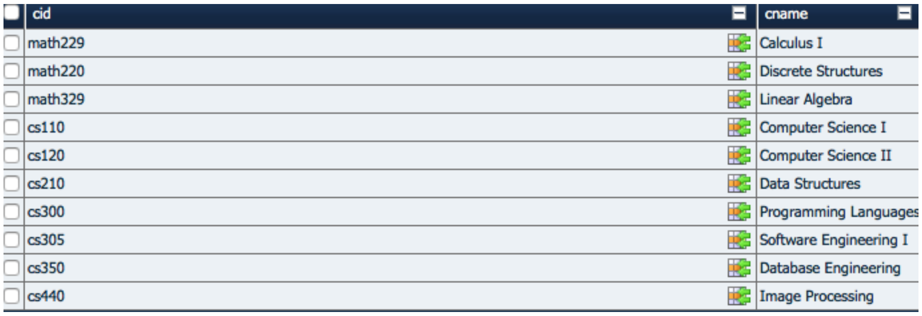
```
SELECT cid, cname
FROM COURSE
WHERE prereq IS NULL
```



cid	cname
math130	College Algebra

18. List course id and course name for courses that have prerequisites.

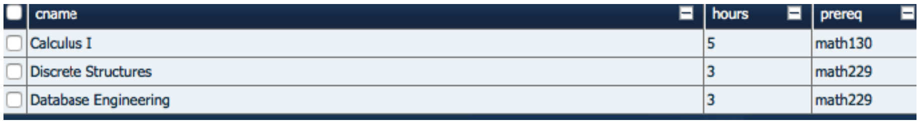
```
SELECT cid, cname
FROM COURSE
WHERE prereq IS NOT NULL
```



cid	cname
math229	Calculus I
math220	Discrete Structures
math329	Linear Algebra
cs110	Computer Science I
cs120	Computer Science II
cs210	Data Structures
cs300	Programming Languages
cs305	Software Engineering I
cs350	Database Engineering
cs440	Image Processing

19. List name, credit hours, and prerequisite for courses which are either five credit hours, OR course name begins with the letter ’D’ and has math229 as prerequisite:

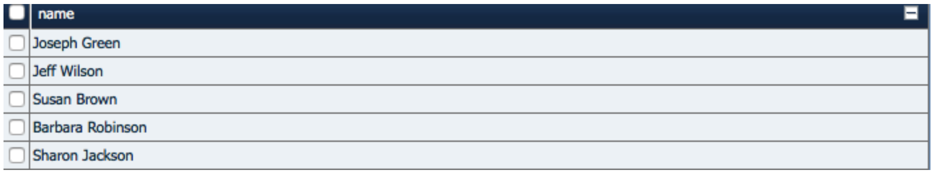
```
SELECT cname, hours, prereq
FROM COURSE
WHERE hours=5 OR (SUBSTRING(cname,1, 1)=’D’ AND prereq=’math229’)
```



cname	hours	prereq
Calculus I	5	math130
Discrete Structures	3	math229
Database Engineering	3	math229

20. Retrieve the names of students whose last name ends with the letter ‘n’.

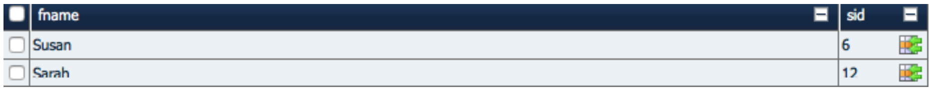
```
SELECT (fname || ’ ’ || lname) AS Name
FROM STUDENT
WHERE SUBSTRING(lname, LENGTH(lname), 1)=’n’
```



□	name
□	Joseph Green
□	Jeff Wilson
□	Susan Brown
□	Barbara Robinson
□	Sharon Jackson

21. Retrieve first name of students whose first name begins with the letter ’S’, followed by any two characters, followed by the letter ‘a’, and followed by any number of characters.

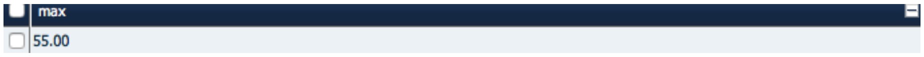
```
SELECT fname
FROM STUDENT
WHERE fname LIKE ’S__a%’
```



□	fname	sid
□	Susan	6
□	Sarah	12

22. What is the highest membership fee charged by any organization?

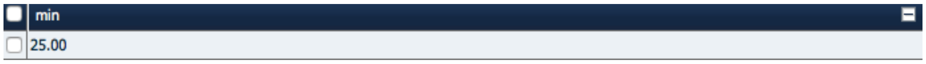
```
SELECT MAX(fee)
FROM ORG
```



□	max
□	55.00

23. What is the lowest membership fee charged by any organization?

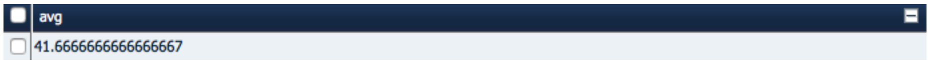
```
SELECT MIN(fee)
FROM ORG
```



□	min
□	25.00

24. What is the average membership fee across all organizations?

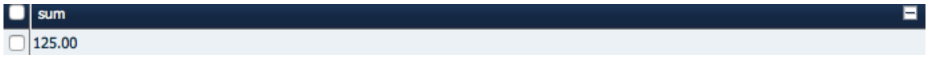
```
SELECT AVG(fee)
FROM ORG
```



□	avg
□	41.6666666666666667

25. Let us say that you wanted to become member of all organization. How much money do you need for membership fee?

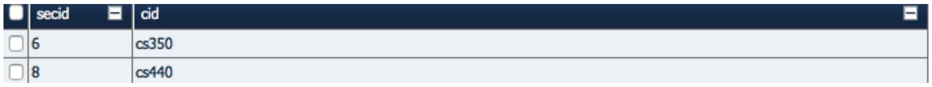
```
SELECT SUM(fee)
FROM ORG
```



sum
125.00

26. List section id and course id for courses that begin after 1.00 PM.

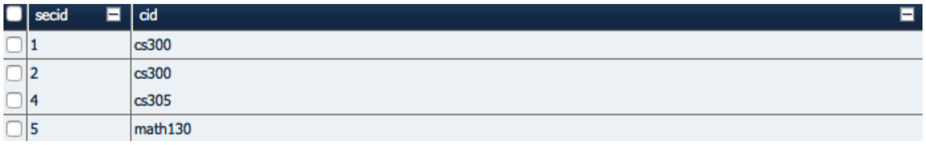
```
SELECT secid, cid
FROM SECTION
WHERE EXTRACT(HOUR FROM stime)>13
```



secid	cid
6	cs350
8	cs440

27. List section id and course id of courses that end some time between 12:00 noon and 1.00 PM.

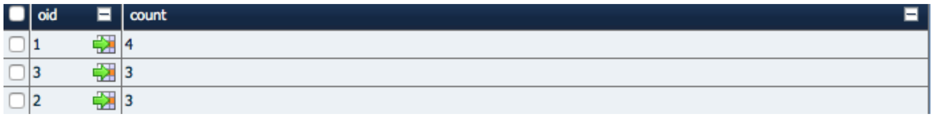
```
SELECT secid, cid
FROM SECTION
WHERE EXTRACT(HOUR FROM etime)>=12 AND EXTRACT(HOUR FROM etime)<=13
```



secid	cid
1	cs300
2	cs300
4	cs305
5	math130

28. For each organization, list organization id and its membership count (i.e., the number of people who are members of the organization).

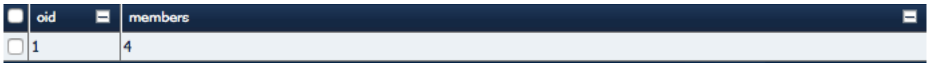
```
SELECT oid, COUNT(sid)
FROM MEMBERSHIP
GROUP BY oid
```



□	oid	count
□	1	4
□	3	3
□	2	3

29. For those organizations with a membership count of more than three, list organization id and membership count.

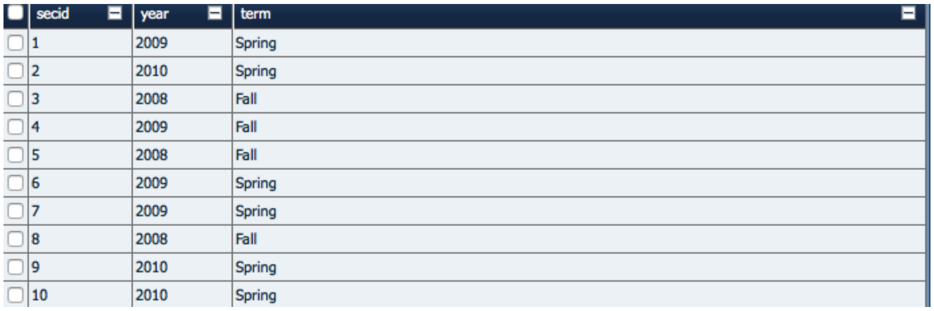
```
SELECT oid, members
FROM (
     SELECT oid, COUNT(sid)AS members
     FROM MEMBERSHIP
     GROUP BY oid
) AS TMP
WHERE members>3
```



□	oid	members
□	1	4

30. For each offering of the course named “Programming Languages,” list section id, year, and term.

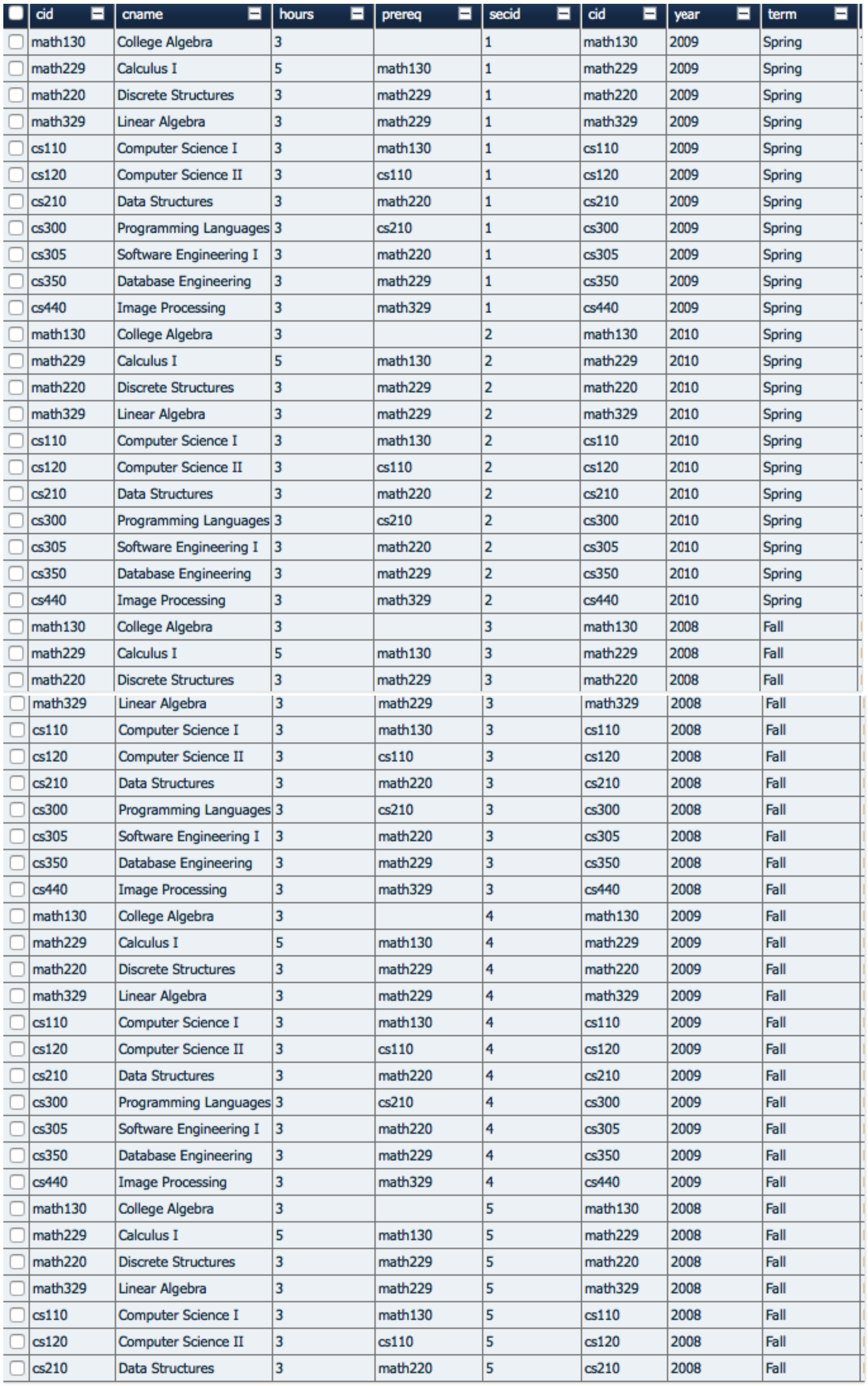
```
SELECT secid, year, term
FROM SECTION, COURSE
WHERE COURSE.cname=’Programming Languages’
```



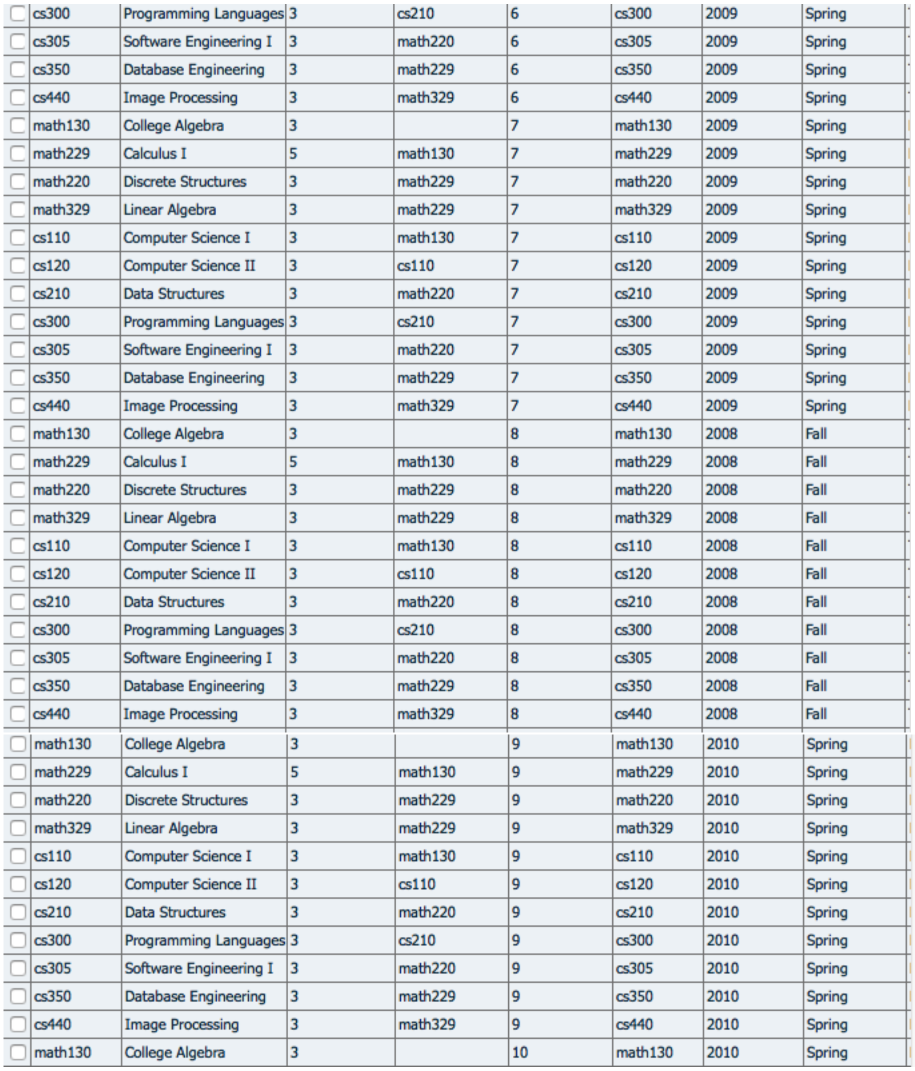
□	secid	year	term
□	1	2009	Spring
□	2	2010	Spring
□	3	2008	Fall
□	4	2009	Fall
□	5	2008	Fall
□	6	2009	Spring
□	7	2009	Spring
□	8	2008	Fall
□	9	2010	Spring
□	10	2010	Spring

31. List all possible combinations of course and section tables. Examine the result set. Are all the rows in the result set meaningful?

```
SELECT *
FROM COURSE CROSS JOIN SECTION
```



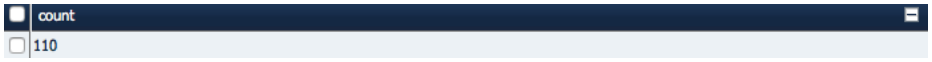
☐	cid	cname	hours	prereq	secid	cid	year	term
☐	math130	College Algebra	3		1	math130	2009	Spring
☐	math229	Calculus I	5	math130	1	math229	2009	Spring
☐	math220	Discrete Structures	3	math229	1	math220	2009	Spring
☐	math329	Linear Algebra	3	math229	1	math329	2009	Spring
☐	cs110	Computer Science I	3	math130	1	cs110	2009	Spring
☐	cs120	Computer Science II	3	cs110	1	cs120	2009	Spring
☐	cs210	Data Structures	3	math220	1	cs210	2009	Spring
☐	cs300	Programming Languages	3	cs210	1	cs300	2009	Spring
☐	cs305	Software Engineering I	3	math220	1	cs305	2009	Spring
☐	cs350	Database Engineering	3	math229	1	cs350	2009	Spring
☐	cs440	Image Processing	3	math329	1	cs440	2009	Spring
☐	math130	College Algebra	3		2	math130	2010	Spring
☐	math229	Calculus I	5	math130	2	math229	2010	Spring
☐	math220	Discrete Structures	3	math229	2	math220	2010	Spring
☐	math329	Linear Algebra	3	math229	2	math329	2010	Spring
☐	cs110	Computer Science I	3	math130	2	cs110	2010	Spring
☐	cs120	Computer Science II	3	cs110	2	cs120	2010	Spring
☐	cs210	Data Structures	3	math220	2	cs210	2010	Spring
☐	cs300	Programming Languages	3	cs210	2	cs300	2010	Spring
☐	cs305	Software Engineering I	3	math220	2	cs305	2010	Spring
☐	cs350	Database Engineering	3	math229	2	cs350	2010	Spring
☐	cs440	Image Processing	3	math329	2	cs440	2010	Spring
☐	math130	College Algebra	3		3	math130	2008	Fall
☐	math229	Calculus I	5	math130	3	math229	2008	Fall
☐	math220	Discrete Structures	3	math229	3	math220	2008	Fall
☐	math329	Linear Algebra	3	math229	3	math329	2008	Fall
☐	cs110	Computer Science I	3	math130	3	cs110	2008	Fall
☐	cs120	Computer Science II	3	cs110	3	cs120	2008	Fall
☐	cs210	Data Structures	3	math220	3	cs210	2008	Fall
☐	cs300	Programming Languages	3	cs210	3	cs300	2008	Fall
☐	cs305	Software Engineering I	3	math220	3	cs305	2008	Fall
☐	cs350	Database Engineering	3	math229	3	cs350	2008	Fall
☐	cs440	Image Processing	3	math329	3	cs440	2008	Fall
☐	math130	College Algebra	3		4	math130	2009	Fall
☐	math229	Calculus I	5	math130	4	math229	2009	Fall
☐	math220	Discrete Structures	3	math229	4	math220	2009	Fall
☐	math329	Linear Algebra	3	math229	4	math329	2009	Fall
☐	cs110	Computer Science I	3	math130	4	cs110	2009	Fall
☐	cs120	Computer Science II	3	cs110	4	cs120	2009	Fall
☐	cs210	Data Structures	3	math220	4	cs210	2009	Fall
☐	cs300	Programming Languages	3	cs210	4	cs300	2009	Fall
☐	cs305	Software Engineering I	3	math220	4	cs305	2009	Fall
☐	cs350	Database Engineering	3	math229	4	cs350	2009	Fall
☐	cs440	Image Processing	3	math329	4	cs440	2009	Fall
☐	math130	College Algebra	3		5	math130	2008	Fall
☐	math229	Calculus I	5	math130	5	math229	2008	Fall
☐	math220	Discrete Structures	3	math229	5	math220	2008	Fall
☐	math329	Linear Algebra	3	math229	5	math329	2008	Fall
☐	cs110	Computer Science I	3	math130	5	cs110	2008	Fall
☐	cs120	Computer Science II	3	cs110	5	cs120	2008	Fall
☐	cs210	Data Structures	3	math220	5	cs210	2008	Fall



□	cs300	Programming Languages	3	cs210	6	cs300	2009	Spring
□	cs305	Software Engineering I	3	math220	6	cs305	2009	Spring
□	cs350	Database Engineering	3	math229	6	cs350	2009	Spring
□	cs440	Image Processing	3	math329	6	cs440	2009	Spring
□	math130	College Algebra	3		7	math130	2009	Spring
□	math229	Calculus I	5	math130	7	math229	2009	Spring
□	math220	Discrete Structures	3	math229	7	math220	2009	Spring
□	math329	Linear Algebra	3	math229	7	math329	2009	Spring
□	cs110	Computer Science I	3	math130	7	cs110	2009	Spring
□	cs120	Computer Science II	3	cs110	7	cs120	2009	Spring
□	cs210	Data Structures	3	math220	7	cs210	2009	Spring
□	cs300	Programming Languages	3	cs210	7	cs300	2009	Spring
□	cs305	Software Engineering I	3	math220	7	cs305	2009	Spring
□	cs350	Database Engineering	3	math229	7	cs350	2009	Spring
□	cs440	Image Processing	3	math329	7	cs440	2009	Spring
□	math130	College Algebra	3		8	math130	2008	Fall
□	math229	Calculus I	5	math130	8	math229	2008	Fall
□	math220	Discrete Structures	3	math229	8	math220	2008	Fall
□	math329	Linear Algebra	3	math229	8	math329	2008	Fall
□	cs110	Computer Science I	3	math130	8	cs110	2008	Fall
□	cs120	Computer Science II	3	cs110	8	cs120	2008	Fall
□	cs210	Data Structures	3	math220	8	cs210	2008	Fall
□	cs300	Programming Languages	3	cs210	8	cs300	2008	Fall
□	cs305	Software Engineering I	3	math220	8	cs305	2008	Fall
□	cs350	Database Engineering	3	math229	8	cs350	2008	Fall
□	cs440	Image Processing	3	math329	8	cs440	2008	Fall
□	math130	College Algebra	3		9	math130	2010	Spring
□	math229	Calculus I	5	math130	9	math229	2010	Spring
□	math220	Discrete Structures	3	math229	9	math220	2010	Spring
□	math329	Linear Algebra	3	math229	9	math329	2010	Spring
□	cs110	Computer Science I	3	math130	9	cs110	2010	Spring
□	cs120	Computer Science II	3	cs110	9	cs120	2010	Spring
□	cs210	Data Structures	3	math220	9	cs210	2010	Spring
□	cs300	Programming Languages	3	cs210	9	cs300	2010	Spring
□	cs305	Software Engineering I	3	math220	9	cs305	2010	Spring
□	cs350	Database Engineering	3	math229	9	cs350	2010	Spring
□	cs440	Image Processing	3	math329	9	cs440	2010	Spring
□	math130	College Algebra	3		10	math130	2010	Spring

32. How many rows are there in all possible combinations of course and section tables.

```
SELECT COUNT(*)
FROM COURSE CROSS JOIN SECTION
```



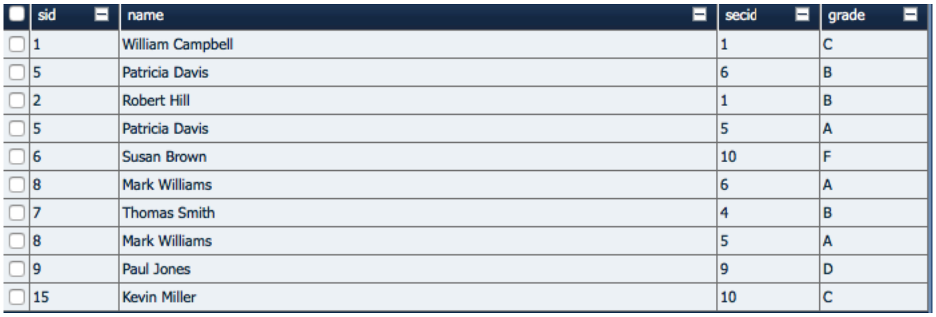
□	count
□	110

33. For students who are members of organizations, list student id and name (first and last), and the names of the organizations.

```
Your answer goes here.
```

34. List student id and name (first and last) for all students, and if they have enrolled in any sections, list section id and grade for each section.

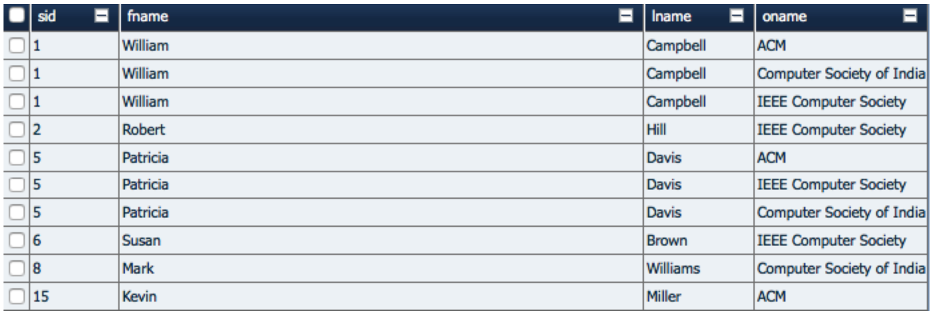
```
SELECT STUDENT.sid, fname || ’ ’ || lname AS name, secid, grade
FROM STUDENT, ENROLLMENT
WHERE STUDENT.sid=ENROLLMENT.sid
```



□	sid	name	secid	grade
□	1	William Campbell	1	C
□	5	Patricia Davis	6	B
□	2	Robert Hill	1	B
□	5	Patricia Davis	5	A
□	6	Susan Brown	10	F
□	8	Mark Williams	6	A
□	7	Thomas Smith	4	B
□	8	Mark Williams	5	A
□	9	Paul Jones	9	D
□	15	Kevin Miller	10	C

35. List student id and name (first and last) of all students, as well as the names of organizations if they happen to be members.

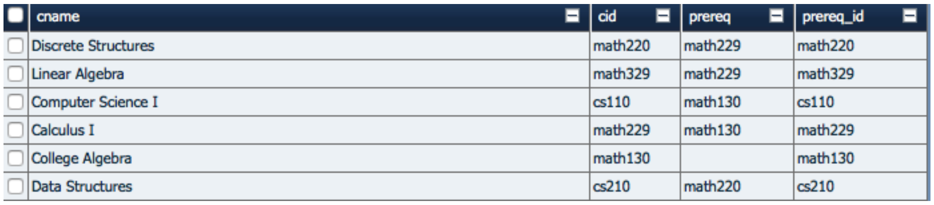
```
SELECT STUDENT.sid, fname, lname, oname
FROM STUDENT, MEMBERSHIP, ORG
WHERE STUDENT.sid=MEMBERSHIP.sid AND MEMBERSHIP.oid=ORG.oid
ORDER BY STUDENT.sid
```



□	sid	fname	lname	oname
□	1	William	Campbell	ACM
□	1	William	Campbell	Computer Society of India
□	1	William	Campbell	IEEE Computer Society
□	2	Robert	Hill	IEEE Computer Society
□	5	Patricia	Davis	ACM
□	5	Patricia	Davis	IEEE Computer Society
□	5	Patricia	Davis	Computer Society of India
□	6	Susan	Brown	IEEE Computer Society
□	8	Mark	Williams	Computer Society of India
□	15	Kevin	Miller	ACM

36. For courses that have prerequisites, list course id, course name, prerequisite course id, and prerequisite course name.

```
SELECT DISTINCT cname
FROM course c, (SELECT prereq
      FROM course) as temp
WHERE c.cid = temp.prereq
```



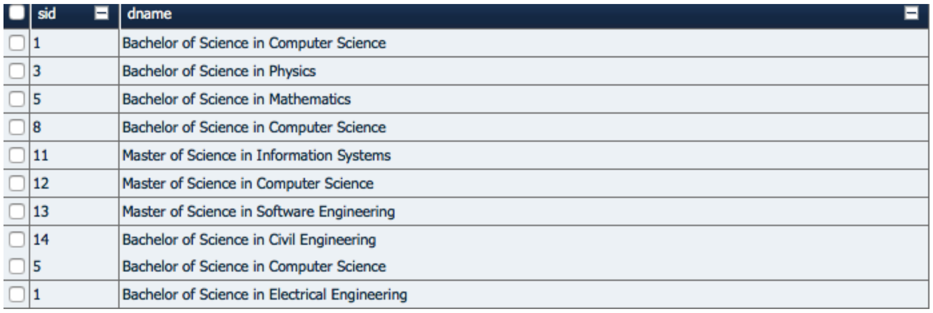
□	cname	cid	prereq	prereq_id
□	Discrete Structures	math220	math229	math220
□	Linear Algebra	math329	math229	math329
□	Computer Science I	cs110	math130	cs110
□	Calculus I	math229	math130	math229
□	College Algebra	math130		math130
□	Data Structures	cs210	math220	cs210

37. List course id, course name, prerequisite course name (if any) for all courses.

```
Your answer goes here.
```

38. For students pursuing degree programs, list student id and the name of the degree program.

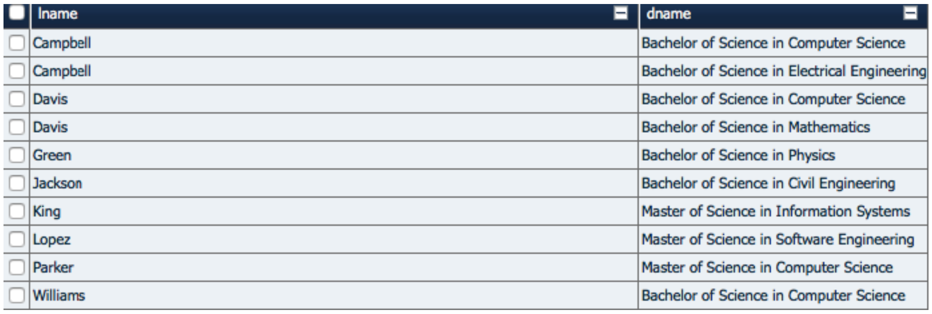
```
SELECT STUDENT.sid, DEGREE.dname
FROM STUDENT, MAJOR, DEGREE
WHERE STUDENT.sid=MAJOR.sid AND MAJOR.dcode=DEGREE.dcode
```



□	sid	dname
□	1	Bachelor of Science in Computer Science
□	3	Bachelor of Science in Physics
□	5	Bachelor of Science in Mathematics
□	8	Bachelor of Science in Computer Science
□	11	Master of Science in Information Systems
□	12	Master of Science in Computer Science
□	13	Master of Science in Software Engineering
□	14	Bachelor of Science in Civil Engineering
□	5	Bachelor of Science in Computer Science
□	1	Bachelor of Science in Electrical Engineering

39. For each student, list last name and the name(s) of degree programs the student is majoring in, sorted by student last name.

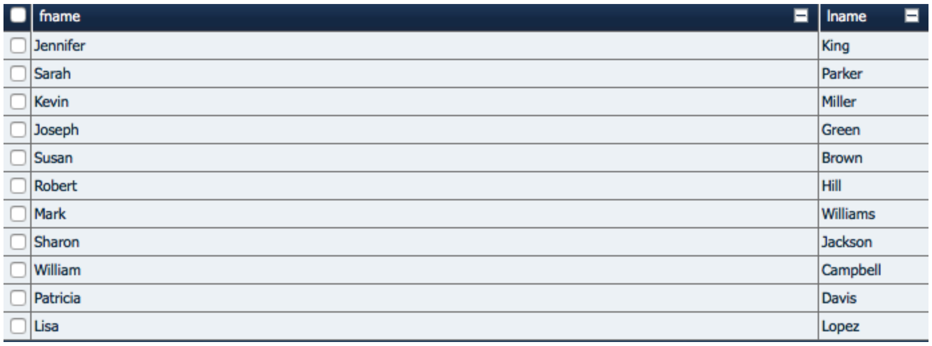
```
SELECT STUDENT.lname, DEGREE.dname
FROM STUDENT, DEGREE, MAJOR
WHERE STUDENT.sid=MAJOR.sid AND MAJOR.dcode=DEGREE.dcode
ORDER BY STUDENT.lname
```



□	lname	dname
□	Campbell	Bachelor of Science in Computer Science
□	Campbell	Bachelor of Science in Electrical Engineering
□	Davis	Bachelor of Science in Computer Science
□	Davis	Bachelor of Science in Mathematics
□	Green	Bachelor of Science in Physics
□	Jackson	Bachelor of Science in Civil Engineering
□	King	Master of Science in Information Systems
□	Lopez	Master of Science in Software Engineering
□	Parker	Master of Science in Computer Science
□	Williams	Bachelor of Science in Computer Science

40. List last names of students who are either enrolled in at least one degree program or members of some organization.

```
SELECT DISTINCT STUDENT.fname, STUDENT.lname
FROM STUDENT, MAJOR, MEMBERSHIP
WHERE STUDENT.sid=MAJOR.sid OR STUDENT.sid=MEMBERSHIP.sid
```



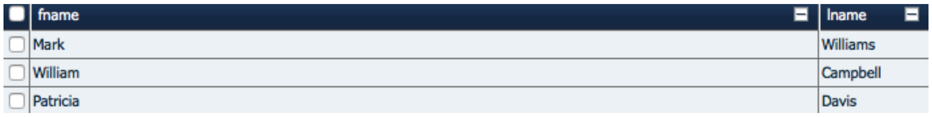
□	fname	lname
□	Jennifer	King
□	Sarah	Parker
□	Kevin	Miller
□	Joseph	Green
□	Susan	Brown
□	Robert	Hill
□	Mark	Williams
□	Sharon	Jackson
□	William	Campbell
□	Patricia	Davis
□	Lisa	Lopez

41. List last names of students who are either enrolled in at least one degree program *or* members of some organization. Don't remove duplicate last names.

```
Your answer goes here.
```

42. List last names of students who are enrolled in at least one degree program *and* are also members of some organization.

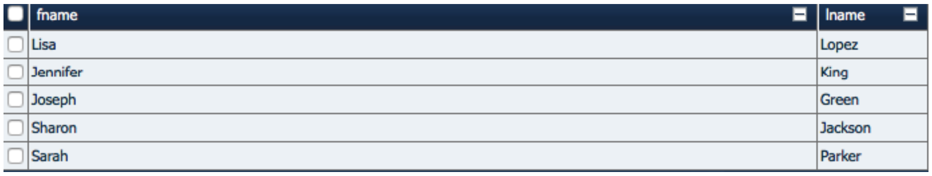
```
SELECT fname, lname
FROM (
    SELECT sid
    FROM MAJOR
    INTERSECT
    SELECT sid
    FROM MEMBERSHIP
) AS TMP, STUDENT
WHERE STUDENT.sid=TMP.sid
```



□	fname	lname
□	Mark	Williams
□	William	Campbell
□	Patricia	Davis

43. List last names of students who are enrolled in at least one degree program but are not members of any organization.

```
SELECT fname, lname
FROM (
    SELECT sid
    FROM MAJOR
    EXCEPT
    SELECT sid
    FROM MEMBERSHIP
) AS TMP, STUDENT
WHERE STUDENT.sid=TMP.sid
```



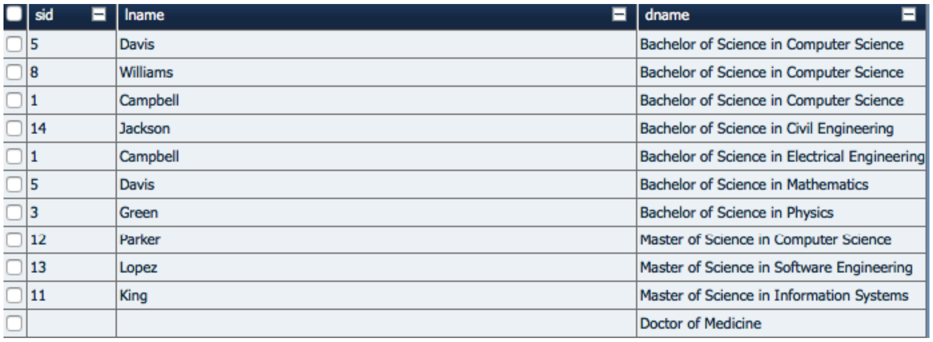
□	fname	lname
□	Lisa	Lopez
□	Jennifer	King
□	Joseph	Green
□	Sharon	Jackson
□	Sarah	Parker

44. List student id and last name of all students, as well as the name(s) of degree programs that they are pursuing (if any). The result set should also show the names of degree programs even if no student is pursuing such degree programs.

```
SELECT sid, lname, dname
FROM (

    SELECT STUDENT.sid, STUDENT.lname, MAJOR.dcode
    FROM STUDENT, MAJOR
    WHERE STUDENT.sid=MAJOR.sid
) AS TMP
RIGHT OUTER JOIN DEGREE
```

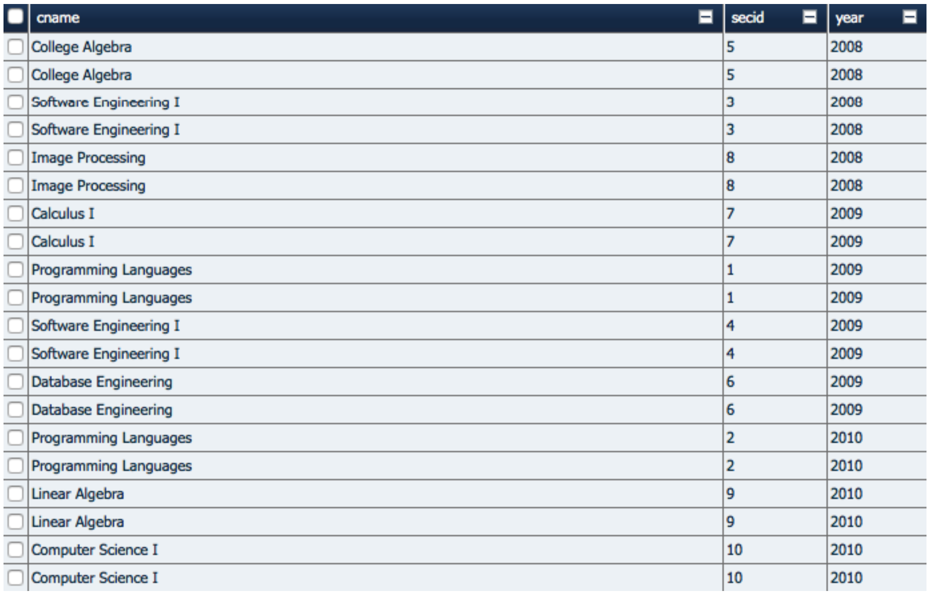
```
ON TMP.dcode=DEGREE.dcode
```



□	sid	lname	dname
□	5	Davis	Bachelor of Science in Computer Science
□	8	Williams	Bachelor of Science in Computer Science
□	1	Campbell	Bachelor of Science in Computer Science
□	14	Jackson	Bachelor of Science in Civil Engineering
□	1	Campbell	Bachelor of Science in Electrical Engineering
□	5	Davis	Bachelor of Science in Mathematics
□	3	Green	Bachelor of Science in Physics
□	12	Parker	Master of Science in Computer Science
□	13	Lopez	Master of Science in Software Engineering
□	11	King	Master of Science in Information Systems
□			Doctor of Medicine

45. For courses that have been offered one or more times, list course name, section id, and year.

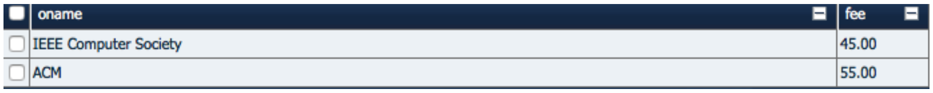
```
SELECT COURSE.cname, SECTION.secid, SECTION.year
FROM (
    SELECT COUNT(*) AS offerings, cid
    FROM SECTION
    GROUP BY cid
) AS TMP, SECTION, COURSE
WHERE TMP.offerings>1 AND COURSE.cid=SECTION.cid
ORDER BY SECTION.year
```



□	cname	secid	year
□	College Algebra	5	2008
□	College Algebra	5	2008
□	Software Engineering I	3	2008
□	Software Engineering I	3	2008
□	Image Processing	8	2008
□	Image Processing	8	2008
□	Calculus I	7	2009
□	Calculus I	7	2009
□	Programming Languages	1	2009
□	Programming Languages	1	2009
□	Software Engineering I	4	2009
□	Software Engineering I	4	2009
□	Database Engineering	6	2009
□	Database Engineering	6	2009
□	Programming Languages	2	2010
□	Programming Languages	2	2010
□	Linear Algebra	9	2010
□	Linear Algebra	9	2010
□	Computer Science I	10	2010
□	Computer Science I	10	2010

46. List name and fee of organizations whose membership fee is greater than the average membership fee across all organizations.

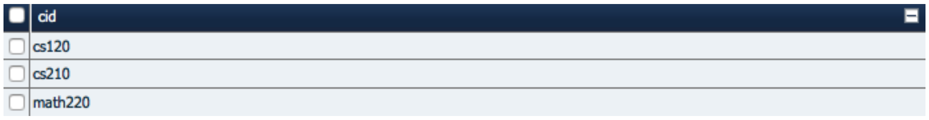
```
SELECT ORG.oname, ORG.fee
FROM (
     SELECT AVG(fee) AS avg_fee
     FROM ORG
) as TMP, ORG
WHERE ORG.FEE>TMP.avg_fee
```



□	oname	fee
□	IEEE Computer Society	45.00
□	ACM	55.00

47. List course id and course name for courses that have never been offered.

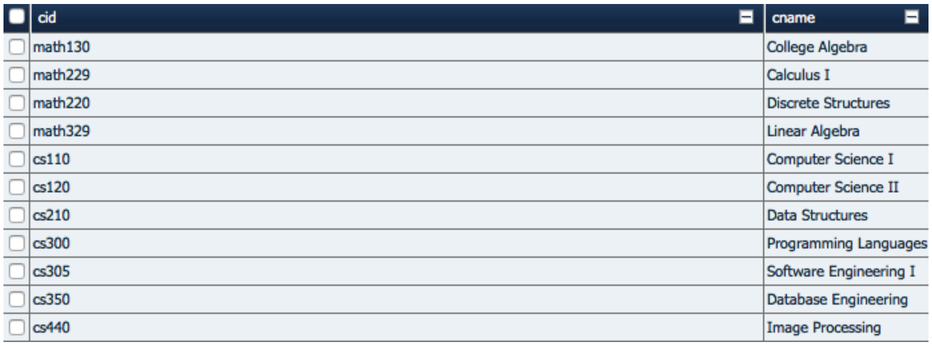
```
SELECT cid
FROM COURSE
EXCEPT
SELECT cid
FROM SECTION
```



□	cid
□	cs120
□	cs210
□	math220

48. List course id and course name for courses that have been offered at least once.

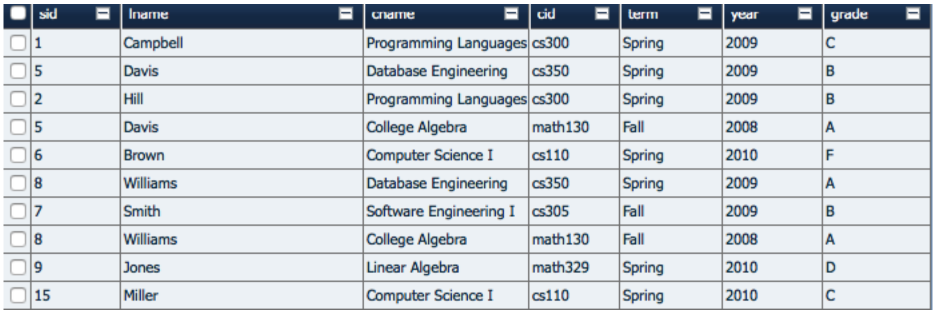
```
SELECT cid, cname
FROM COURSE
WHERE EXISTS (
    SELECT cid
    FROM SECTION
)
```



□	cid	cname
□	math130	College Algebra
□	math229	Calculus I
□	math220	Discrete Structures
□	math329	Linear Algebra
□	cs110	Computer Science I
□	cs120	Computer Science II
□	cs210	Data Structures
□	cs300	Programming Languages
□	cs305	Software Engineering I
□	cs350	Database Engineering
□	cs440	Image Processing

49. For students who have taken courses (i.e., enrolled in sections), list student id and last name, as well as course id, course name, year, term, and grade for each course. Use natural join.

```
SELECT STUDENT.sid, STUDENT.lname, COURSE.cname, SECTION.cid, SECTION.term, SECTION.ye
FROM STUDENT, ENROLLMENT, COURSE, SECTION
WHERE STUDENT.sid=ENROLLMENT.sid AND ENROLLMENT.secid=SECTION.secid AND COURSE.cid=SEC
```



□	sid	lname	cname	cid	term	year	grade
□	1	Campbell	Programming Languages	cs300	Spring	2009	C
□	5	Davis	Database Engineering	cs350	Spring	2009	B
□	2	Hill	Programming Languages	cs300	Spring	2009	B
□	5	Davis	College Algebra	math130	Fall	2008	A
□	6	Brown	Computer Science I	cs110	Spring	2010	F
□	8	Williams	Database Engineering	cs350	Spring	2009	A
□	7	Smith	Software Engineering I	cs305	Fall	2009	B
□	8	Williams	College Algebra	math130	Fall	2008	A
□	9	Jones	Linear Algebra	math329	Spring	2010	D
□	15	Miller	Computer Science I	cs110	Spring	2010	C

50. For students who have taken courses (i.e., enrolled in sections), list student id and last name, as well as course id, course name, year, term, and grade for each course. The result set should also include a new column at the very end titled "Remarks." If the student has earned an 'A', Remarks column should display Superior. Likewise, Good for a 'B', Average for 'C', Poor for 'D', Incomplete for 'I', Fail for 'F', and Withdrawn for 'W.'

```
Your answer goes here.
```

2 Preparing Written Report

Use the LaTeX template to generate your solution in PDF format.

3 Rubric

Use the following rubric to evaluate your response to this assignment.

Perf Level *Trait*	*Outstanding*	*Good*	*Fair*	*Poor*
Problem Analysis	Precise and concise documentation to support the claim that the problem is correctly and thoroughly analyzed prior to writing the query is provided.	Sufficient documentation, though verbose and non-coherent, to support the claim that the problem is correctly analyzed prior to writing the query is provided.	Some documentation to support the claim that the problem is superficially analyzed prior to writing the query is provided.	Documentation to support the claim that the problem is analyzed prior to writing the query is not provided.
Query Execution	The query compiles and produces correct results. Also, the query runs efficiently.	Though inefficient, the query compiles and produces correct results.	The query compiles but produces incorrect results.	The query does not compile.
Correctness Arguments	The correctness of the query is argued using rigorous statements.	The correctness of the query is argued using informal logical statements.	The correctness of the query is argued using illogical statements.	There are no statement about the query correctness.
Completeness	Answers to all questions in the assignment are provided. Queries compile and run. Problem analysis documentation and correctness arguments are provided.	Answers to less than 75% of the questions in the assignment are provided. Queries compile and run. Problem analysis documentation and correctness arguments are provided.	Answers to less than 50% of the questions in the assignment are provided. Queries compile and run. Problem analysis documentation and correctness arguments are provided.	Answers to less than 25% of the questions in the assignment are provided. Queries compile and run. Problem analysis documentation and correctness arguments are provided.

4 Self-assessment

Use the following table and the rubric of section 3 to score your solution. Circle the appropriate number in each row. For example, to circle 20, use the LaTeX markup code `\circled{20}`, which produces ⑳.

Perf Level	*Outstanding*	*Good*	*Fair*	*Poor*
Trait				
Problem Analysis	[✓ 10]	8	6	4
Query Execution	[✓ 30]	25	20	15
Correctness Arguments	[✓ 10]	8	6	4
Completeness	[✓ 50]	40	30	20