CSCB20 midterm exam

Sood Shruti

TOTAL POINTS

32.75 / 47.25

QUESTION 1

5 pts

1.1 1 / 1

✓ **- 0 pts Correct**

- 0.25 pts Incorrect HTML reasoning

- 0.25 pts Incorrect JavaScript reasoning

- 1 pts No valid difference stated

- 1 pts Left question blank

1.2 2 / 2

✓ **- 0 pts Correct**

- 0.75 pts Partial information. Mostly correct, and hence, partial credit awarded.

- 2 pts Completely missing sufficient information

- 2 pts Left question blank

1.3 1 / 2

- 0 pts Correct

- 0.75 pts Incorrect definition of Flask

✓ **- 1 pts Invalid advantage**

- 2 pts Left question blank

QUESTION 2

18 pts

2.1 2 / 2

✓ **- 0 pts Correct**

- 1 pts Example is wrong or missing

- 1 pts Doesn't answer the question explicitly

- 2 pts Incorrect or question not attempted

2.2 3 / 3

✓ **- 0 pts Correct**

- 1 pts Missing one super key

- 2 pts Missing two super keys

- 3 pts Missing 3 super keys

2.3 1 / 2

- 0 pts Correct

✓ **- 1 pts Multiple Candidate keys listed, when there should only be one**

- 2 pts Incorrect answer or not answered

2.4 2 / 2

✓ **- 0 pts Correct**

- 1 pts Multiple candidate keys when there should only be one

- 2 pts Incorrect answer or no answer

- 1 pts Wrong format.

2.5 3 / 3

✓ **- 0 pts Correct**

- 1 pts A few incorrect superkeys

- 3 pts Incorrect answer or no answer

- 1 pts Missing one superkey

- 2 pts Missing 2+ superkeys

- 2 pts Many incorrect superkeys

2.6 1.5 / 3

- 0 pts Correct

- 1 pts Doesn't specify the relationship PNO belongs to

- 0.5 pts Doesn't specify where PNO is referencing to

✓ **- 1.5 pts Doesn't give one foreign key**

- 2 pts Doesn't specify any foreign keys

- 3 pts Incorrect answer or no answer

- 1 pts Doesn't specify the relationship BNO belongs to

- 0.5 pts Doesn't specify where BNO is referencing to

- **1 pts** Extra foreign key specified
- **0.5 pts** Incorrectly specified the relationship of BNO
- **0.5 pts** Incorrectly specified the relationship of PNO
- **0.5 pts** Incorrect reference for BNO
- **0.5 pts** Incorrect reference for PNO

2.7 3 / 3

✓ **- 0 pts Correct**
- **1 pts** Missing books schema
- **1 pts** Improper books schema
- **1 pts** Improper publisher schema
- **1 pts** Missing publisher schema
- **1 pts** Improper publishes schema
- **1 pts** Missing publishes schema
- **3 pts** Incorrect answer or answer missing

QUESTION 3

8 pts

3.1 2 / 2

✓ **- 0 pts Correct (true)**
- **2 pts** Incorrect (answer should be true)
- **1 pts** Correct, but wrong explanation
- **1 pts** No explanation given, but answer correct

3.2 2 / 2

✓ **- 0 pts Correct (false)**
- **2 pts** Incorrect (answer should be false)
- **1 pts** .Correct, but wrong explanation

3.3 2 / 4

- **0 pts** Correct (2,5)
- **4 pts** Both options wrong

✓ **- 2 pts One option correct with correct explanation**
- **3 pts** One option correct with wrong explanation
- **1 pts** Additional option given which is not correct
- **4 pts** No explanation given

QUESTION 4

16 pts

4.1 1 / 3

- **0 pts** Correct
- **0.25 pts** Has correct idea but has some minor syntax errors
- **1 pts** Has correct idea but is missing some minor part of the WHERE condition
- **1.5 pts** Has correct idea but is missing some major part of the WHERE condition (i.e. missing f.name = 'Ana' or s.level = 'Junior')

✓ **- 2 pts Has significant syntax or logical errors**
- **2 pts** Solution includes some irrelevant queries
- **3 pts** Completely incorrect

4.2 3 / 3

✓ **- 0 pts Correct**
- **0.25 pts** Has correct idea but has minor syntax or logical errors
- **1.5 pts** Has correct idea but significant syntax or logical errors
- **1.5 pts** HAVING and/or GROUP BY are used incorrectly
- **2 pts** Does not use COUNT
- **2 pts** Does not use GROUP BY and/or HAVING
- **2 pts** Has irrelevant queries
- **2 pts** Does not SELECT correct column (i.e. does not select the names of all classes that meet criteria)
- **3 pts** Completely wrong
- **3 pts** Answer left blank

4.3 1 / 1

✓ **- 0 pts Correct**
- **1 pts** Incorrect
- **1 pts** Solution is incomprehensible
- **1 pts** Student guessed correct option but has no response for part (ii)
- **1 pts** Answer was left blank

4.4 2 / 2

✓ **- 0 pts Explanation is sufficient**
- **1 pts** Explanation has right idea but is based upon incorrect claims
- **1 pts** Explanation has right idea but is not

sufficiently detailed

- **2 pts** Student selected YES for part (i)
- **2 pts** Explanation is completely wrong or lacks any detail
- **2 pts** Explanation is incomprehensible
- **2 pts** No explanation given

4.5 0 / 3

- **0 pts** Correct
- **0.25 pts** Has correct idea but has minor syntax or logical errors
- **1 pts** FROM condition has irrelevant tables
- **1.5 pts** Has correct idea but has major syntax or logical errors
- **2 pts** Does not SELECT correct columns
- **1 pts** Uses GROUP BY and/or HAVING
- **2 pts** Has irrelevant queries
- **3 pts** Answer left blank

✓ **- 3 pts Completely wrong**

4.6 0 / 4

- **0 pts** Correct
- **0.25 pts** Has right idea but minor syntax or logical errors
- **1 pts** FROM statement has irrelevant or incorrect tables
- **1.5 pts** Has right idea but some irrelevant queries
- **1.5 pts** Has right idea but major syntax or logical errors
- **2 pts** Does not SELECT correct columns
- **2 pts** Does not use NOT IN
- **2 pts** No subquery after NOT IN
- **2 pts** Has irrelevant queries that indicate lack of understanding

✓ **- 4 pts Completely wrong**

- **4 pts** Answer left blank

QUESTION 5

5 Bonus 0.25 / 0.25

✓ **- 0 pts Correct**

- **0.25 pts** No name written

CSCB20
UTSC
10th February 2019
Duration 90 mins
Midterm Exam

Enter your First Name: Shruti

Enter your Last Name: Sood

Enter your Student#: 1005593715

Question	Points	Score
1	5	
2	18	
3	8	
4	16	
Total:	47	

For each of the question, explain your answer in detail and clearly. If you make any assumptions, these must be stated explicitly. If you need extra paper, use the last blank page on your exam.

Use the following notation for 'joins' in relational algebra when writing your answer:

$$A \times B \quad (1)$$

$$A \,_{\theta}\times\, B \quad (2)$$

$$A \; naturalJoin \; B \quad (3)$$

- 1) refers to cross join between A and B
- 2) refers to theta join between A and B where θ is some boolean condition specified by you.
- 3) refers to natural join between A and B
- THERE WILL BE NO QUESTIONS TAKEN DURING THIS EXAM. IF YOU HAVE A DOUBT, STATE YOUR ASSUMPTION AND CONTINUE WRITING.
- Do not take off the stapler pins from your exam. We will award you +.25 bonus points if you do not remove the stapler pins (including the appendix pages) and write your first name and last name on every single page of the exam.

Shruti Sood
1005593715

1. (a) What is one major difference between HTML and JavaScript? (1)

HTML = Hyper text markup language, it is used to give structure to a web page. It is the skeleton of a web page. It has the format <HTML>
<HEAD> <TITLE> </TITLE> </HEAD>
<BODY> </BODY>
</HTML>.

Javascript = It is used to bring a web page to life. It declares the functionalities of various components of web page. It is included in the HTML code as <script.....>

(b) What is an IP address? What is it used for? (2)

IP address is the internet protocol address.
Every computer has its own IP address, hence it is used to identify computers or distinguish between them by data packages or any information coming in.

(c) What is Flask and what is its one major advantage? (2)

Flask is a virtual environment which we use to 'host' web pages we created on the internet.

Total for Question 1: 5

2. Consider the following relational database that represents, **books**, **publishers**, and **publishes** that contains information about books published by publishers.

books

BNO	Title	Author	Date	Edition
231	The Soul of a New Machine	Tracy Kidder	1981	1
77	Programming Pearls	Jon Bentley	2000	2
23	Programming Pearls	Jon Bentley	1981	1
2	Tess of the d'Urbevilles	Thomas Hardy	1850	1

publishers

PNO	Publisher	City	Web Site
1	Back Bay Books	Boston	backbay.com
2	Addison Wesley	New York	addisonwesley.com
3	Modern Library	London	randomhouse.com
4	Penguin	New York	penguin.com

publishes

PNO	BNO	Pages	Copyright
1	231	293	1981
2	77	235	2001
2	23	200	1980
3	2	565	2001
4	2	540	1990

Figure 1: Relation of 'books', 'publishers' and 'publishes'

- If a book 'b' is published by publisher 'p' then 'b' is in the books relation, 'p' is in the publishers relation, and a row is added to the publishes relation.
- The date in the books relation represents the year when the book was originally issued.
- The date in the publishes table represents the copyright date when that particular publisher issued it.
- BNO numbers are unique book identifiers (similar to an ISBN number)
- PNO (publisher numbers) are unique publisher identifiers.

(a) Does this database allow for a book to have more than one publisher? Explain why or why not using an example. (2)

Yes, this database allows for a book to have more than one publisher. Eg: the last two rows of publishes, for the same book with BNO=2, the PNO = 3 & 4 in different rows. Hence Book with BNO=2 has 2 publishers 3 and 4 respectively.

(b) List three super keys of the **books** relation. (3)

1. BNO, Edition
2. BNO, Title, Edition
3. BNO, Title, Edition, date

(c) List all candidate keys of the **books** relation. (2)

1. BNO, Edition

Shruti Sood
1005593715

(d) List all of the candidate keys of the **publishes** relation. (2)

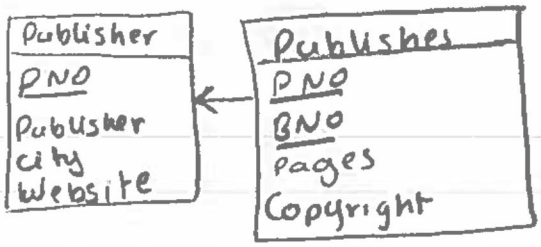
1. PNO, BNO

(e) List all of the superkeys of the **publishes** relation. (3)

1. PNO , BNO
2. PNO, BNO, Pages
3. PNO, BNO, Copyright
4. PNO, BNO, Pages, Copyright

(f) List all of the foreign keys in the database. Be explicit by naming the referencing and referenced relations and the appropriate attributes. (3)

PNO is a foreign key in publishes from publishers



(g) What is the relational schema for this database? Use the proper notation for specifying a schema as we looked at in lecture 1 of this term. (3)

books (BNO, Title, Author, Date, Edition)

publishers(PNO, Publisher, City , Website)

publishes (PNO, BNO , Paages, Copyright)

Total for Question 2: 18

[redacted] Sem
1005593715

3. (a) Consider the following database schema:

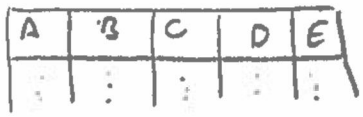
R(A, B)
S(A, B, C)
T(B, D, E)

For the following questions, clearly circle either **True or False.** Make sure to provide an explanation that supports your answer.

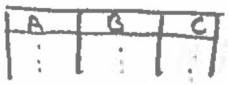
i. The following two relational algebra queries are equivalent: (2)

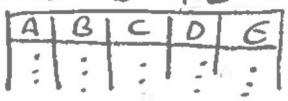
$$Q_1 = \sigma_{A=1,B>2}((R \bowtie S) \bowtie T)$$
$$Q_2 = (\sigma_{A=1,B>2}(R \bowtie S)) \bowtie T$$

True

Q_1: will give you result 

where A is common in R & S, and B is common in R, S & T, and A=1 and B>2.

Q_2: will first give you result 

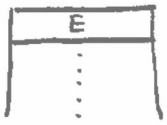
where A=1 and B>2, and A & B have some values in tables R and S.
and then adds cols D & E from T where A=1 and B>2.
ending up with 

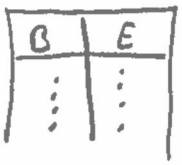
which will be the same table as produced by query 1.

ii. The following two relational algebra queries are equivalent: (2)

$$Q_3 = \Pi_E(\sigma_{D=1}(T \bowtie S))$$
$$Q_4 = \Pi_B(S) \bowtie \Pi_{B,E}(\sigma_{D=1}(T))$$

False

Because Q_3 will give an output looking like : 

and Q_4 will give an output looking like : 

which obviously leads to the conclusion that the Relational algebra queries are not equal since they have different outputs.

Shruti Sood
1005593715

(b) Consider the following database schema: (4)

```
Likes (drinker, beer)
Frequents (drinker, bar)
Serves (bar, beer)
```

Which of the following[1] relational algebra queries answer the following question: **which drinkers frequent only bars that serve only beers they like?**. Make sure to explain your answer.

1. $\Pi_{drinker}(Likes \bowtie Frequents \bowtie Serves)$ ✗
2. $\Pi_{drinker}(Frequents) - \Pi_{drinker}(((\Pi_{drinker}(Frequents) \times \Pi_{beer}(Serves)) - Likes) \bowtie Frequents \bowtie Serves$ ✓
3. $\Pi_{drinker}(Frequents) - \Pi_{drinker}(Frequents - \Pi_{drinker,bar}(Likes \bowtie Serves))$ ✓
4. $\Pi_{drinker}(Frequents) - \Pi_{drinker}(Likes \bowtie Frequents \bowtie Serves)$ ✗
5. $\Pi_{drinker}(Frequents) - \Pi_{drinker}(\Pi_{drinker,beer}(Frequents \bowtie Serves) - Likes)$ ✗

Total for Question 3: 8

Relational Algebra queries 2 and 3 answer the question which drinker frequents only bars that serve only beers they like

Lets have a sample database:

Likes

drinker	Beer
Joe	red
Joe	blue
Tom	red
Tom	black
Charlie	blue

Frequents

drinker	bar
Joe	1
Charlie	1
Tom	2

Serves

bar	beer
1	red
1	blue
2	red
2	black

The required output to satisfy condition is

drinker
Joe

1. gives output

drinker
Joe
Charlie
Tom

, Hence it does not satisfy query

2. gives output

drinker
Joe

, Hence it satisfies query

3. gives output

drinker
Joe

, Hence it satisfies query

※ Question Sol^n continued on the last page

[1] There could be multiple solutions or no solution at all

※ Note answer continued on last page due to lack of space. After Pg 11.

[redacted] Soda
[redacted] 0055 93715

4. Consider the following database schema:

Student(*snum:* integer, *sname:* string, *major:* string, *level:* string, *age:* integer)
Class(*name:* string, *meets_at:* string, *room:* string, *fid:* integer)
Enrolled(*snum:* integer, *cname:* string)
Faculty(*fid:* integer, *fname:* string, *deptid:* integer)

Figure 2: Relation of 'Student', 'Class', 'Enrolled' and 'Faculty'

(a) Find the names of all Junior's (i.e. level in Student is Junior) who are enrolled in a class taught by Ana i.e. (fname in Faculty is Ana). We have completed portion of the SQL query and asking you to complete the 'Where' clause: (3)

```
SELECT DISTINCT S.Sname
FROM Student S, Class C, Enrolled E, Faculty F
WHERE S.level='junior' AND C.fid IN (SELECT F1.fid
                                     FROM Faculty F1
                                     WHERE F1.fname='Ana');
```

(b) Find the names of all classes that either meet in room R128 (i.e. room of Class is R128) or have five or more students enrolled. We have completed portion of the SQL query and asking you to complete everything inside the IN portion. You must use 'Group_By' and 'Having' in your solution. (3)

```
SELECT C.name
FROM Class C
WHERE C.room = 'R128'
OR C.name IN ( SELECT E.cname
               FROM Enrolled E
               GROUP BY E.cname
               HAVING (COUNT (E.cname)) >= 5
);
```

* For this question, I am assuming cname in Enrolled is a class name, similar to name in Class

Shruti Sood
1005593715

(c) Here is one proposed solution to the query **Find the names of all students who are enrolled in two classes that meet at the same time.**

```
SELECT DISTINCT S.sname
FROM Student S
WHERE S.snum IN (SELECT E1.snum
FROM Enrolled E1, Enrolled E2, Class C1, Class C2
WHERE E1.snum = E2.snum AND E1.cname <> E2.cname
AND E1.cname = C1.name
AND E2.cname = C2.name AND C1.meets_at = C2.meets_at)
```

i. Is there a mistake in the proposed solution? Yes or No. (1)

No, there is no mistake in the proposed solution.

★ (Assuming all students have different names for different snums, as we are using DISTINCT sname)

ii. If there is a mistake with the above solution, explain clearly where the mistake is and how you would correct it. If there is no mistake with the solution, just state that it is correct and explain in few words why do you think it is correct. (2)

No, there is no mistake.

The from table creats a big table which is the result of a cross product between two enrolled and h Class tables.

Then the conditions check, if the two classes meet at same time for the same student number in enrolled and different course names (cname) in enrolled and E1.cname = C1.name & E2.cname = C2.name makes sure if the classes are right.

So, the inside/inner SELECT statement gives us the snum of students according to the query and the outter SELECT statements give us the Sname corresponding to the inner query, hence giving us sname of the query asked, hence giving us the solution.

Mruthi Sooq
1005593715

(d) Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five. We have completed portion of the SQL query and asking you to complete everything inside the parenthesis. You must use the count function and theta join. (3)

```
SELECT DISTINCT F.fname
FROM Faculty F
WHERE 5 > ( SELECT  COUNT (snum)
        FROM (Faculty F INNER JOIN Class C INNER JOIN Enrolled E
              ON F.fid = C.fid AND C.name = E.cname)
        GROUP BY FID
);  (NOTE: INNERJOIN is the same thing as a theta join)
```

(e) Write a SQLite query that finds the names of students not enrolled in any class. You must use 'NOT IN' in your solution. (4)

```
SELECT DISTINCT S.sname
FROM (Student S NATURAL JOIN Enrolled E)
WHERE S.snum NOT IN E.snum;
```

Total for Question 4: 16

Shruti Sood
1005593715

Appendix

Consider the following relational table `OrderDetails`.

Table 1: Table: OrderDetails

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10249	14	9

Here are the following SQL queries that involve `sum`, `count` and `avg`. These may be useful for the exam.

- ```
  SELECT SUM(Quantity)
  FROM OrderDetails;
  ```
  The output of the above SQL query is a table with one column called `Sum(Quantity)` with a single row containing 31 i.e. the sum function sums up every single item in that column.

- ```
  SELECT AVG(Quantity)
  FROM OrderDetails;
  ```
  The output of the above SQL query is a table with one column called `AVG(Quantity)` with a single row containing 10.33 i.e. the AVG function returns the average of all the items in that column.

- ```
  SELECT COUNT(Quantity)
  FROM OrderDetails;
  ```
  The output of the above SQL query is a table that has one column called `Count(Price)` and one row with the value 3 i.e. the count function just counts all the tuples that were present from `OrderDetails`.

Note: You can always add a `WHERE` clause to the above examples if you like to perform 'selection' and select only certain tuples that meet some condition as specified in the `WHERE` clause. Now, let us look at some other example involving 'Group By' and 'Having'.

- ```
  SELECT COUNT(OrderID), OrderID
  FROM OrderDetails
  GROUP BY OrderID
  HAVING COUNT(OrderID) > 1
  ```
  The best way to understand this is first start with 'Group By'. What 'Group By' does is that it looks at the column 'OrderID' and figures out how many unique items are there in that column. We know from the table above that there are 2 unique items i.e. 10248, 10249. Ok, now let us look at the `SELECT` statement. The output of this entire SQL query will have two columns i.e. `Count(OrderID)` and `OrderID` where the column `OrderID` will contain those unique values fetched from 'Group By'. Now in the first column `CountOrderDetailID`, you are simply going to count how many tuples in the entire table have the `OrderID` as 10248 and 10249. Finally, we only want those values that have `COUNT(OrderID)` of > 1. Here is the final output produced:

  ```
  COUNT(OrderID), OrderID
  2                 10248
  ```

- Let us look at another similar example:

  ```
  SELECT COUNT(OrderID), OrderID
  FROM OrderDetails
  GROUP BY OrderID
  HAVING COUNT(OrderID) > 0
  ```

Here is the final output produced:

```
COUNT(OrderID), OrderID
2                10248
1                10249
```

IN operator

The IN operator is a shorthand for multiple OR conditions.

- ```
  SELECT * FROM Customers
  WHERE Country IN (`Germany', `France', `UK');
  ```

  The above SQL statement selects all customers that are located in "Germany", "France" and "UK":

- ```
  SELECT * FROM Customers
  WHERE Country NOT IN (`Germany', `France', `UK');
  ```

  The above SQL statement selects all customers that are NOT located in "Germany", "France" or "UK":

- ```
  SELECT * FROM Customers
  WHERE Country IN (SELECT Country FROM Suppliers);
  ```

  The above SQL statement selects all customers that are from the same countries as the suppliers:

NOT IN operator

Very similar to the IN operator, however, the NOT negates it. The following statement returns a list of tracks whose genre id is not in a list of (1,2,3).

- ```
  SELECT
   trackid,
   name,
   genreid
  FROM
   tracks
  WHERE
   genreid NOT IN (1,2,3);
  ```

  You can also replace the hard coded values of (1,2,3) with some other SELECT statement that returns back a relation of one column.

NOT EXISTS

The SQLite EXISTS condition can also be combined with the NOT operator. For example,

- ```
      SELECT *
  FROM departments
  WHERE NOT EXISTS (SELECT *
                    FROM employees
                    WHERE departments.department_id = employees.department_id);
  ```

This SQLite EXISTS example will return all records from the departments table where there are no records in the employees table for the given department_id.
  ```

Shruti Sood
10055 93715

Q.3 part b answer continued

4. gives output

drinker
NULL

, Hence, it does not satisfy query

5. gives output

drinker
Charlie

, Hence it does not satisfy query.

Hence only RA queries 2 & 3 satisfy the query. Hence that is our solution.