Reminders:
1. Q1 - no partial credit for this question
2. Q2 - 5 points for pointing out the correct operators with the type, 5 points for the explanation and 5 points for examples
3. Partial credit may be awarded for Q3, Q4, and Q5 if the answers are incorrect but correct explanations are provided

Q1. RA: Warming up [12 points]
Consider the following two tables, T1 and T2:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>b</td>
<td>8</td>
</tr>
<tr>
<td>25</td>
<td>a</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>b</td>
<td>6</td>
</tr>
<tr>
<td>25</td>
<td>c</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>b</td>
<td>5</td>
</tr>
</tbody>
</table>

Show the results of the following relational algebra queries (2 points each):

Q1.1. \( \pi_Q (T1) \cap \pi_B (T2) \)
Ans:

\[
\begin{array}{c}
\emptyset \\
\end{array}
\]

Q1.2. \( \sigma_{R>5 \lor Q=a} (T1) \)
Ans:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>b</td>
<td>8</td>
</tr>
<tr>
<td>25</td>
<td>a</td>
<td>6</td>
</tr>
</tbody>
</table>

Q1.3. \( T1 \bowtie_{T1.A=T2.A} T2 \)
Ans:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
<td>10</td>
<td>b</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
<td>10</td>
<td>b</td>
<td>5</td>
</tr>
<tr>
<td>25</td>
<td>a</td>
<td>6</td>
<td>25</td>
<td>c</td>
<td>3</td>
</tr>
</tbody>
</table>
Q1.4.  \( T_1 \bowtie_{T_1.Q=T_2.B} T_2 \)

Ans:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>b</td>
<td>8</td>
<td>10</td>
<td>b</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>b</td>
<td>8</td>
<td>10</td>
<td>b</td>
<td>5</td>
</tr>
</tbody>
</table>

Q1.5.  \( T_1 \bowtie T_2 \) (assume the natural join happens for columns P and A)

Ans:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
<td>b</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
<td>b</td>
<td>5</td>
</tr>
<tr>
<td>25</td>
<td>a</td>
<td>6</td>
<td>c</td>
<td>3</td>
</tr>
</tbody>
</table>

Q1.6. \( T_1 \bowtie_{T_1.P=T_2.A \land T_1.R=T_2.C} T_2 \)

Ans:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
<td>10</td>
<td>b</td>
<td>5</td>
</tr>
</tbody>
</table>

Q2. RA: Monotone Operators [15 points]

An operator on relations is said to be **monotone** if whenever we add a tuple to one of its arguments, the result contains all the tuples that it contained before adding the tuple, plus perhaps more tuples. Which of the 5 fundamental RA operators we saw in class (see Lecture 2) are monotone? For each either explain why it is monotone, or give a small example showing it is not.

ANS:

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>A</td>
</tr>
<tr>
<td>Q</td>
<td>B</td>
</tr>
<tr>
<td>R</td>
<td>C</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>25</td>
</tr>
<tr>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

Monotone operators change the original output when there are any changes made to the input tables.

1. Selection – Monotone
   
   In case of selection, there will be no removal of rows if we add an extra row to the input. It is possible that the new row(s) may not be selected, but the original rows will be present.

   Ex: \( \sigma_{R>5 \lor Q=a}(T1) \) is
If we add an extra row say 

The output will remain the same as above.

2. Projection – Monotone
Suppose we do a projection on table T1 as above on the column P. The result set will contain all the values in the column P. Suppose we added the new row as in (1), the value of P in that row will become part of the output as well. There will be no changes to any of the original rows.

3. Union – Monotone

If we do T1 U T2, the result set would be:

If we added an extra row say 

The result:

has all 3 rows, none of the rows in the original result set will disappear.

4. Cross Product – Monotone
Suppose we have 2 tables:
The cartesian product of the above 2 tables is:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>b</td>
<td>8</td>
</tr>
</tbody>
</table>

If I were to add an extra row to either one of these tables, two more rows would be added to the output, but there would be no removal of rows that were present in the original output.

5. Difference - Non-Monotone

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
<td>25</td>
<td>a</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>b</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(T1) (T2)

If we perform T1 – T2, we get the following result,

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>b</td>
<td>8</td>
</tr>
</tbody>
</table>

If following row was added to T2,

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>b</td>
<td>8</td>
</tr>
</tbody>
</table>

the result set would be:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
</tr>
</tbody>
</table>

One of the rows that was in the original output, wouldn’t be presenting making this a non-monotone operator.

Q3. RA: Products and Manufacturers [22 points]

Consider the following relational database that stores information about computer hardware products (keys are underlined, field types are omitted):

Product (maker, model, type)
PC (model, speed, ram, hd, price)
Laptop (model, speed, ram, hd, screen, price)
Printer (model, color, type, price)

The Product relation gives the manufacturer, model number and type (PC/laptop/printer) of various products. The PC relation gives for each model number that is a PC the speed (of the processor, in gigahertz), the amount of RAM (in GB), the size of hard disk (in GB), and the price. The Laptop relation is similar, with screen size
included. The Printer relation records whether it is a color printer or not (Yes/No), and the type (laser/ink-jet) and the price.

**Write the following queries in relational algebra:**

Q3.1. (2 points) Find the model numbers of all color laser printers.
   Ans: \( \pi_{\text{model}}(\sigma_{\text{type}='\text{laser}' \land \text{color}='\text{yes'}}(\text{Printer})) \)

Q3.2. (3 points) Find the model number and price of all products (of any type) made by manufacturer ‘HP’.
   Ans:
   \[
   \pi_{\text{model}, \text{price}}(\rho_{\text{T1}}(\sigma_{\text{maker}='\text{HP'}}(\text{Product})) \bowtie_{\text{T1.model} = \text{T2.model}} \rho_{\text{T2}}(\pi_{\text{model, price}}(\text{Printer}) \cup \pi_{\text{model, price}}(\text{Laptop}) \cup \pi_{\text{model, price}}(\text{PC})))
   \]

Q3.3. (5 points) Find those manufacturers of at least two different computers (PCs or laptops) with speeds of at least 2.80.
   *Hint:* You may need a ‘self’ Cartesian product and a join.
   Ans:
   \[
   \pi_{\text{T1.maker}}(\sigma_{\text{T1.maker} \geq \text{T2.maker}}(\rho_{\text{T1}}(\pi_{\text{model, p.price}}(\text{Product}) \bowtie_{\text{p.model} = \text{FC.model}} \rho_{\text{FC}}((\pi_{\text{model}}(\sigma_{\text{speed} \geq 2.8}(\text{PC}))) \cup \pi_{\text{model}}(\sigma_{\text{speed} \geq 2.8}(\text{Laptop})))))) \times \rho_{\text{T2}}(\text{Product}))
   \]

Q3.4. (6 points) Find the manufacturers who sell exactly three different models of PC.
   *Hint:* Similar in spirit to Q3.3, but you may have to use the set difference as well.
   Ans:
   \[
   \pi_{\text{T.maker}}(\sigma_{\text{itemcount} = 3}(\gamma_{\text{T.maker.count(maker)}}(\text{itemcount}(\rho_{\text{T}}(\text{Product} \bowtie_{\text{Product.model} = \text{PC.model}} \text{PC})))))
   \]

Q3.5. (6 points) Find the manufacturers who sell PCs with at least the set of all the RAM sizes seen in laptops.
   Ans:
   \[
   \pi_{\text{Product.maker}}((\pi_{\text{PC, ram}}(\text{Product.maker}(\text{PC} \bowtie_{\text{PC.model} = \text{Product.model}} \text{Product})) \div \pi_{\text{ram}}(\text{Laptop})))
   \]

Q4. SQL: The School DB [21 points]
The schema of the database is provided below (keys are underlined, field types are omitted):

- student(sid, surname, sex, age, year, gpa)
- dept(dname, numphds)
- prof(fname, dname)
- course(cno, cname, dname)
In this assignment, you will only deal with querying part of SQL. You are NOT allowed to tamper with (change the contents of) the database, i.e., CREATE, INSERT, DELETE, ALTER, UPDATE etc.

Write SQL queries that answer the questions below (one query per question). The query answers must not contain duplicates, but you should use the SQL keyword distinct only when necessary. For this question, creation of temporary tables is NOT allowed, i.e., for each question you have to write exactly one SQL statement (possible using nested SQL).

Q4.1. (2 points) Find the names and gpas of the students who are enrolled in 312.
   Ans: select s.sname, s.gpa from student as s inner join enroll as e on s.sid = e.sid where e.cno=312;

Q4.2. (2 points) Find the name of the oldest student.
   Ans: select sname from student where age = (select max(age) from student);

Q4.3. (3 points) Find the names and majors of students who are taking one of the Artificial Intelligence courses (courses containing the name Artificial Intelligence).
   Ans: select s.sname,m.dname from student s inner join major m on s.sid = m.sid inner join enroll e on s.sid=e.sid inner join course as c on c.cno = e.cno where cname like '%Artificial Intelligence%';

Q4.4. (4 points) Find the names of students who are enrolled in a course from both the "Computer Sciences" and "Chemical Engineering" departments.
   Ans: select sname from student where sid = (select distinct(e.sid) from (enroll as e join course as c on e.cno = c.cno) inner join (enroll as e1 join course as c1 on e1.cno = c1.cno) on e.sid = e1.sid where (c.dname = 'Computer Science' and c1.dname = 'Chemical Engineering') or (c.dname = 'Chemical Engineering' and c1.dname = 'Computer Science'));

Q4.5. (5 points) How many students have more than one major? (Hint: requires a nested query)
   Ans: select count(*) from (select sid from major group by sid having count(*) > 1) as groupedmajor;

Q4.6. (5 points) Find the name(s) of the oldest first year student (year = 1) (Hint: requires a nested query)
   Ans: select sname from student where age = (select max(age) from student);

Q5: Movie Ratings [30 points]
This question is on a movie ratings database. Download and install SQLite3 from http://www.sqlite.org
Warm-up
Follow the documentation and load the sample database at:
http://courses.cs.vt.edu/~cs4604/Fall18/homeworks/hw1/cs4604-hw1.db

It has a table recommendation which is like the table of the Netflix competition: people rate movies, with ratings from 1-5 (1 for ‘I hate it!’ to 5, for ‘I love it!’). As a sanity check that you have the correct database, running the following command at a Unix/Linux/Cygwin prompt-your-machine% sqlite3 cs4604-hw1.db 'select count(*) from recommendation'
should return
14

We want to write SQL queries to do the following:
• Query1: Return all movies with a rating of 5 from at least one reviewer.
• Query2: Return all the reviewers who rated ‘Fight Club’.

Large CSV file
A bigger raw comma separated value (csv) file is given here:
http://courses.cs.vt.edu/~cs4604/Fall18/homeworks/hw1/movie_ratings.csv

It is a subset from the Netflix-competition dataset (If you are curious, the official Netflix USD 1 Million prize dataset is at http://www.netflixprize.com/). The dataset is anonymized, hence customers will be represented by a random, integer id. We want to write queries to do the following:
• Query3: Return the count of reviews where the rating is 5.
• Query4: Return the count of reviewers who gave a rating of 1 to ‘Gone with the wind’.

Life without SQL
Finally, in your favorite language (Python/Perl/Ruby/Java/C++ etc.) write procedural code to do both queries above (Query3 and Query4) on the csv data file directly. Notice: the end-of-line convention is the DOS one (CR LF).

Deliverables
Q5.1. (2 points) The SQL query for Query1.
Ans: select distinct movie from recommendation where rating >=5;

Q5.2. (2 points) The SQL query for Query2.
Ans: select customer from recommendation where movie = 'Fight Club';

Q5.3. (2 points) The output of running Query1 in SQLite on the sample database.
Ans:
```
sqlite> select distinct movie from recommendation where rating >=5;
Gone with the wind
Fight Club
```

Q5.4. (2 points) The output of running Query2 in SQLite on the sample database.
Ans:
```
sqlite> select customer from recommendation where movie = 'Fight Club';
John Doe
Rob Smith
Jane Matthew
Kate Brown
Bob Miller
```

Q5.5. (4 points) The SQL query for Query3.
Ans: select count(*) from recommendation where rating = 5;

Q5.6. (4 points) The SQL query for Query4.
Ans: select count(*) from recommendation where movie = 'Gone with the wind' and rating = 1;

Q5.7. (3 points) The output of running Query3 on the csv file after loading it in SQLite.
Ans:
```
sqlite> select count(*) from recommendation where rating = 5;
2191
```

Q5.8. (3 points) The output of running Query4 on the csv file after loading it in SQLite.
Ans:
```
sqlite> select count(*) from recommendation where movie = 'Gone with the wind' and rating = 1;
28
```

Q5.9. (4 points) Hard copy of your python/perl/etc code for doing Query3 on the raw csv file directly.
Ans:
```python
import csv

count = 0
with open('movie_ratings.csv') as ratings:
    movieRating = csv.reader(ratings, delimiter=',')
    for row in movieRating:
        if int(row[2]) >= 5:
            count = count + 1
print(count)
```
Q5.10. (4 points) Hard copy of your python/perl/etc code for doing Query4 on the raw csv file directly.

Ans:

```python
import csv

count = 0
with open('movie_ratings.csv') as ratings:
    movieRating = csv.reader(ratings, delimiter=',')
    for row in movieRating:
        if int(row[2]) == 1 and row[1] == 'Gone with the wind':
            count = count + 1

print(count)
```

Hints

For loading the csv file,

- Again, the end-of-line convention follows the DOS format (CR LF).
- Use the .import and .mode csv commands of sqlite3 or check the link: [https://www.sqlite.org/cli.html](https://www.sqlite.org/cli.html)
• A cheat sheet for sqlite3 commands

• Again as a sanity check, the command

  your-machine% wc -l movie_ratings.csv

  should return

  10000 movie_ratings.csv