Student Number: 

Last Name: 

First Name: 

Section normally attended (Circle one): L0101/L2001 (1pm) L0201/L2003 (3pm) L5101/L2501 (6pm)

Do not turn this page until you have received the signal to start. In the meantime, please fill out the identification section above, and read the instructions below.

This test consists of 4 questions on 6 pages (including this one). When you receive the signal to start, please make sure that your copy of the test is complete.

If you use any space for rough work or need to scratch out an answer, circle the part that you want us to mark.

You may write in pencil, however, work written in pencil will not be considered for remarking.

Good Luck!
Question 1.  [12 marks]

Twitter is a social media platform where users post messages called “tweets”. Users can “follow” other users, and can “like” a tweet. Consider this relational schema for Twitter data. Keys are underlined.

As this question considers relational algebra, assume all relations are sets containing no nulls.

User(userID, name, email)
A Twitter user.

Tweet(tweetID, userID, content, date)
The user with userID made a tweet containing content on date.

Follows(a, b)
User a follows user b on Twitter, which means that a has subscribed to b’s tweets.

Likes(who, tweetID, userID)
User who liked tweet tweetID, userID.

Part (a)  [2 marks]
Does the scheme enforce this constraint: a user cannot like his or her own tweet? Circle one:

Yes  No.

If yes, explain; if not write a new constraint to enforce it. You cannot change any of the existing constraints, rather write a new constraint to enforce it:

Part (b)  [2 marks]
Does the schema enforce this constraint: every userID in Tweet must have a name and email (recorded in User)? Circle one:

Yes  No.

If yes, explain; if not write a new constraint to enforce it:

Part (c)  [2 marks]
Suppose relation Tweet has 100 tuples and \( \Pi_{\text{tweetID}}(\text{Tweet}) \) has 10 tuples. How many tuples could Users have? Circle all that apply:

0  1  100  10,000  100,000
Part (d) [4 Marks]
Which of the following pairs of queries are equivalent? Circle each pair that returns the same results on all database instances.

1. $\Pi_{\text{tweetID}}(\text{Likes} \bowtie T\text{weet}) = \Pi_{\text{tweetID}}(T\text{weet})$
2. $\text{User} = \text{User} \times \text{User}$
3. $\Pi_{\text{userID}}(\sigma_{\text{tweetID}=1}(T\text{weet}) = \Pi_{\text{userID}}(\sigma_{\text{tweetID}=1}(\text{Likes}))$
4. $\Pi_{\text{name}}(\text{User} \bowtie T\text{weet} \bowtie \text{Likes}) = \Pi_{\text{name}}(\text{User} \bowtie T\text{weet} \bowtie \text{Likes} \bowtie \text{Follows})$

Part (e) [2 Marks]
Which of the following queries can be expressed using the same form of relational algebra that we used in class and on Assignment 1, that is assignment, and the operators $\Pi$, $\sigma$, $\bowtie$, condition, $\times$, $\cap$, $\cup$, $\rho$? Circle all that apply.

1. Find all users who have never tweeted.
2. Find dates on which every tweet that was posted was liked by at least two users.
3. Find the email of the most popular user (the user whose tweets are liked by the most people).
4. Find the first date on which a tweet was made.
5. Find the users with the fewest followers.
Question 2. [8 marks]

Here is the schema from Assignment 1. A few attributes and relations have been omitted for simplicity.

Relations

Product(DIN, manufacturer, name, form, schedule)
A tuple in this relation represents a drug product.

Price(DIN, price)
The price of a drug product.

Prescription(RxID, date, patient, drug, doctor)
A prescription for drug was written on date for patient by doctor. Attribute patient is the patient’s OHIP number.

Filled(RxID, date, pharmacist)
Prescription RxID was filled by pharmacist on date.
Attribute pharmacist is the pharmacist’s OCP number.

Integrity constraints

Price[DIN] ⊆ Product[DIN]

Prescription[drug] ⊆ Product[DIN]

Filled[RxID] ⊆ Prescription[RxID]

Π schedule Product ⊆ 
{“prescription”, “narcotic”, “OTC”}

For every drug product that has been prescribed, report its DIN, the date on which the first prescription for it was written, and the date on which the last prescription for it was written.

Use only the basic operators Π, σ, ×, ∞, ρ, and assignment.
Continue your answer here if more space is needed.
Question 3. [6 marks]

Suppose we have implemented the Twitter schema from Question 1 in SQL, and the tables currently contain the following:

<table>
<thead>
<tr>
<th>User</th>
<th>Follows</th>
</tr>
</thead>
<tbody>
<tr>
<td>userid</td>
<td>name</td>
</tr>
<tr>
<td>--------+--------------------------+--------------</td>
<td></td>
</tr>
<tr>
<td>adele</td>
<td>Adele Laurie Blue Adkins</td>
</tr>
<tr>
<td>drizzy</td>
<td>Drake</td>
</tr>
<tr>
<td>potus</td>
<td>Barack Obama</td>
</tr>
<tr>
<td>rjm</td>
<td>Renee Miller</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>potus</td>
<td>drizzy</td>
</tr>
<tr>
<td>drizzy</td>
<td>rjm</td>
</tr>
<tr>
<td>drizzy</td>
<td>adele</td>
</tr>
<tr>
<td>adele</td>
<td>drizzy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tweet</th>
<th>Likes</th>
</tr>
</thead>
<tbody>
<tr>
<td>tweetid</td>
<td>userid</td>
</tr>
<tr>
<td>---------+--------+----------+------------</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>adele</td>
</tr>
<tr>
<td>61</td>
<td>adele</td>
</tr>
<tr>
<td>33</td>
<td>potus</td>
</tr>
<tr>
<td>28</td>
<td>rjm</td>
</tr>
</tbody>
</table>

| who | tweetid | userid |
|-----|---------+--------|
| drizzy | 15 | adele |
| drizzy | 61 | adele |
| potus | 33 | potus |
| potus | 61 | adele |

Show the output of each of the following queries.

1. \[
\text{SELECT who} \\
\text{FROM Likes natural full join Tweet} \\
\text{WHERE tweetID < 30;}
\]

2. \[
\text{SELECT t.userID, count(*)}, \text{max(date)} \\
\text{FROM Tweet t, Likes l} \\
\text{where t.userID = l.who} \\
\text{GROUP BY t.userID;}
\]

3. \[
\text{SELECT count(*)} \\
\text{FROM User, Likes;}
\]

4. \[
\text{SELECT DISTINCT date} \\
\text{FROM User NATURAL JOIN Tweet;}
\]
SELECT T.TweetID, max(date) 
FROM Tweet T, Likes L 
WHERE T.userid = L.userid 
GROUP BY T.TweetID 
HAVING count(*) < 1; 

(SELECT userID 
FROM Likes) 
EXCEPT ALL 
(SELECT userID 
FROM Tweet);

Question 4. [4 MARKS]

Write a query to find the name of users who have only liked tweets posted on the same date that they have made a tweet. Ensure that your query would work on any instance of the database, not simply the one above.
[Use the space below for rough work. This page will not be marked unless you clearly indicate the part of your work that you want us to mark.]
# 1: _____/12
# 2: _____/ 8
# 3: _____/ 6
# 4: _____/ 4

TOTAL: _____/30