CSE 462: Test 1 (02/10/10)

You are given the following relational schema (keys are underlined):

`BANK(Name,Address)`
`OWES(Debtor,Creditor,Amount)`

where `Debtor` and `Creditor` are bank names (they are foreign keys referencing `BANK(Name)`). A bank can be a debtor and a creditor at the same time but not of the same bank. `Amount` has to be greater than zero and represents the debt of the `Debtor` to the `Creditor`.

Problem 1 (40 pts)

Write the following queries in relational algebra:

- **Query 1.1:** *List all banks located in Buffalo that owe money to some bank located in Buffalo.*

Write the following queries in SQL, possibly using views:

- **Query 1.2:** *List all banks that do not owe anything to other banks.*
- **Query 1.3:** *For every bank B, return its net debt defined as the difference between*
  - *the sum of all debts that B owes to other banks, and*
  - *the sum of all debts that other banks owe to B.*

The net debt can be negative, positive, or zero.
Solution.
Query 1.1:

$$\pi_{N_1}((\sigma_{A_1='Buffalo'}(BANK(N_1,A_1)) \bowtie OWES(A_1,A_2,M)) \bowtie (\sigma_{A_2='Buffalo'}(BANK(N_2,A_2)))).$$

Query 1.2:

```
SELECT b1.Name FROM BANK b1
WHERE NOT EXISTS
  (SELECT * FROM OWES
   WHERE OWES.Debtor=b1.Name)
```

Query 1.3:

```
CREATE VIEW DEBT(Name,Amount) AS
(SELECT Debtor, SUM(Amount) FROM OWES
 GROUP BY Debtor)
 UNION
(SELECT b1.Name, 0 FROM BANK b1
 WHERE NOT EXISTS
 (SELECT * FROM OWES
  WHERE OWES.Debtor=b1.Name));
```

```
CREATE VIEW CREDIT(Name,Amount) AS
(SELECT Creditor, SUM(Amount) FROM OWES
 GROUP BY Creditor)
 UNION
(SELECT b1.Name,0 FROM BANK b1
 WHERE NOT EXISTS
 (SELECT * FROM OWES
  WHERE OWES.Creditor=b1.Name);

SELECT CREDIT.Name, DEBT.Amount-CREDIT.Amount
FROM CREDIT, DEBT
WHERE CREDIT.Name=DEBT.Name
```

Problem 2 (20 pts)

Consider the following queries:

Query 2.1:

```
SELECT Address FROM BANK
WHERE Name <> 'Citi'
```

Query 2.2:

```
SELECT b1.Address FROM BANK b1
WHERE NOT EXISTS
  (SELECT * FROM Bank b2
   WHERE b2.Name='Citi'
   AND b2.Address=b1.Address)
```

To do:

1. Explain what each query is doing.
2. Is there a database instance for which **Query 2.1** and **Query 2.2** return different results? Justify your answer.

Solution.
Query 2.1: Return the addresses of banks different than 'Citi'.
Query 2.2: Return the addresses of banks whose address is different from the address of 'Citi'.
For the following instance:

Name	Address
Citi	New York
HSBC	New York

Query 2.1 returns 'New York,' while **Query 2.2** returns the empty set.