Question 1.   [6 MARKS]

Part (a)   [2 MARKS]

Consider this schema:

> A(<u>orange</u>, popcorn)
>
> B(sugar, <u>tea</u>)
>
> A[orange] $\subseteq$ B[sugar]
> → *subset*

Suppose relation A has 17 tuples. Circle the one statement below that is the strongest thing we can be certain of.

1. The number of tuples in relation B must be $>=$ 17.

2. The number of tuples in relation B must be $<=$ 17.

3. The number of tuples in relation B must be 17.

4. The number of tuples in relation B must be $>=$ 1.

5. The number of tuples in relation B must be $>=$ 0.

Explain your answer:

Because, $A \subseteq B$, ie A is a subset of B and for A to be a subset of B the number of elements should be equal on both sets.

Part (b)   [4 MARKS]

Consider this schema and dataset:

C(purple, quince, <u>red</u>)

D(<u>saffron</u>, teal)

C:

purple	quince	red
1	2	3
1	2	5
2	2	5

D:

saffron	teal
2	2
5	1
2	1

D[saffron] $\subseteq$ C[quince]

Does the dataset violate the schema? Circle True or False for each aspect of the schema below. 1 mark for each correct answer, -0.5 for each incorrect answer. The minimum grade for this question is 0.

It violates the key constraint for relation C.   (True)   False

It violates the key constraint for relation D.   (True)   False

It violates a foreign key constraint.   True   (False)

It violates another constraint. Specify it below: True   (False)

Question 2. [14 MARKS]

Here is part of the schema from assignment 1. Recall that *CN is an object's catalogue number*, and *who* is the Staff person who catalogued it

→ must be satisfied

Relations

- Object(<u>CN</u>, date, name, description, type, length, width, height, who)

- Donor(<u>DID</u>, surname, firstname, address, email)

- Donation(<u>NID</u>, date, DID)

- Contains(<u>NID, CN</u>)

- Staff(<u>SID</u>, surname, firstname, address, email, type, date)

Integrity constraints

- Object[who] $\subseteq$ Staff[SID]

- Contains[NID] $\subseteq$ Donation[NID]

- Contains[CN] $\subseteq$ Object[CN]

- Donation[DID] $\subseteq$ Donor[DID]

Answer the following questions in relational algebra, using only the basic operators $\Pi, \sigma, \bowtie, \times, \cap, \cup, -, \rho$.

Part (a) [7 MARKS]

Find donations that contain exactly two items. Report the donation's NID and the donor's email address.

Solution:

$Bad(OID) := (\Pi_{OID} Offering) - (\Pi_{OID} \sigma_{grade=100} Took)$

$TookBad(SID, OID) := \Pi_{SID,OID}(Took \bowtie Bad)$

$TwoPlus(SID) := \sigma_{TB1.SID=TB2.SID \wedge TB1.OID \neq TB2.OID}(\rho_{TB1} TookBad \times \rho_{TB2} TookBad)$

$\Pi_{SID}(Students) - TwoPlus$

Part (b)　[7 marks]

Find the SIDs of staff members who have catalogued at least one object out of each donation from the donor whose DID is 9876. Report simply the SIDs.

Question 3.   [11 marks]

Here is part of a schema you used for one of your Lecture Prep exercises.

Relations

- Employee(<u>eid</u>, name, salary, dept)

- Department(<u>did</u>, name, division)

- Sales(<u>eid</u>, <u>day</u>, amount)

Integrity constraints

- Employee[dept] ⊆ Department[did]

- Sales[eid] ⊆ Employee[eid]

Part (a)   [2 marks]

Suppose table Sales has this content:

```
eid |     day     | amount
-----+------------+--------
   4 | 2013-11-02 |       9
   4 | 2013-11-03 |      10
   4 | 2013-11-05 |      25
   4 | 2013-11-06 |     129
   5 | 2013-11-01 |      12
   5 | 2013-11-02 |       9
   6 | 2013-11-06 |     129
   6 | 2013-11-07 |      18
   7 | 2013-11-01 |      18
   7 | 2013-11-02 |       8
   8 | 2013-11-01 |      28
   8 | 2013-11-02 |     129
```

Below each of the following two queries, write its output.

```
(SELECT amount FROM Sales WHERE amount >= 25);
```

```
(SELECT amount FROM Sales WHERE amount >= 25)
union
(SELECT amount FROM Sales WHERE eid >= 6);
```

Solution:

```
-- First query:
csc343h-dianeh=> (SELECT amount FROM Sales WHERE amount >= 25);
 amount
--------
     25
    129
    129
     28
    129
(5 rows)


-- For the record, the second half of the union:
csc343h-dianeh=> (SELECT amount FROM Sales WHERE eid >= 6);
 amount
--------
    129
     18
     18
      8
     28
    129
(6 rows)


-- second query:
csc343h-dianeh=> (SELECT amount FROM Sales WHERE amount >= 25)
csc343h-dianeh-> union
csc343h-dianeh-> (SELECT amount FROM Sales WHERE eid >= 6);
 amount
--------
     25
      8
     28
     18
    129
(5 rows)
```

Part (b) [4 MARKS]

Complete each of the following queries so that they will find the highest value of amount in the Sales table.

```
SELECT *
FROM Sales s
WHERE amount  _____ (SELECT _____ FROM Sales);
```

```
SELECT *
FROM Sales s
WHERE NOT EXISTS (_____);
```

Solution:

```
csc343h-dianeh=> SELECT *
FROM Sales s
WHERE amount >=
(SELECT max(amount) FROM Sales);
 eid |     day      | amount
-----+------------+--------
   4 | 2013-11-06 |    129
   6 | 2013-11-06 |    129
   8 | 2013-11-02 |    129
(3 rows)

csc343h-dianeh=> SELECT *
FROM Sales s
WHERE NOT EXISTS
(SELECT * FROM Sales s2 WHERE
s2.amount > s.amount);
 eid |     day      | amount
-----+------------+--------
   4 | 2013-11-06 |    129
   6 | 2013-11-06 |    129
   8 | 2013-11-02 |    129
(3 rows)
```

Part (c) [5 MARKS]

Write a query in SQL to find the departments in which the minimum salary is at least 100. For each, report the department name and the sum of all salaries in the department.

Solution:

```
select dept, sum(salary)
from department join employee on dept = did
group by dept
having min(salary) >= 100;
```