1. Question ID 559: Suppose relation R(a,b,c) has the following tuples: (1,1,3), (1,2,3), (2,1,4), (2,3,5), (2,4,1), (3,2,4), and (3,3,6). Define the view *V* by:

```
CREATE VIEW V AS
    SELECT a+b AS d, c FROM R;
```

What is the result of the query:

```
SELECT d, SUM(c) FROM V
GROUP BY d
HAVING COUNT(*) <> 1;
```

Question Explanation: First, for the current value of R, V(d,c) consists of the following tuples: {(2,3), (3,3), (3,4), (5,5), (6,1), (5,4), (6,6)}. In the query itself, we first create groups for d = 2, 3, 5, and 6. However, the group for d = 2 has exactly one member, and so does not survive the HAVING clause. In the group for d = 3, we sum c = 3 and c = 4, so the tuple from this group is (3,7). Similarly, the tuple from the group d = 5 is (5,9) and the tuple for the group d = 6 is (6,7).

2. Question ID 600: Here is a SQL standard trigger on relation R(a,b):

```
CREATE TRIGGER T
AFTER INSERT ON R
REFERENCING NEW ROW AS Newtuple
FOR EACH ROW
WHEN(Newtuple.a * Newtuple.b > 10)
INSERT INTO R VALUES(Newtuple.a - 1, Newtuple.b + 1);
```

When we insert a tuple into R, the trigger may cause another tuple to be inserted, which may cause yet another tuple to be inserted, and so on, until finally a tuple is inserted that does not cause the trigger to fire. Your problem is to examine the behavior of this trigger and determine under what circumstances exactly three tuples are inserted. Demonstrate your understanding by identifying, from the list below, which of the following tuples, if inserted into an initially empty relation R, results, after all instances of the trigger are allowed to execute, in exactly three tuples being present in R(a,b)?

Question Explanation: Notice that the trigger keeps inserting tuples, but lowers the first component by 1 and raises the second component by 1, until the product of the components is 10 or less. Thus, if we insert (x,y) first, then in order to insert exactly three tuples, we need for x*y and (x-1)*(y+1) to be greater than 10, but for (x-2)*(y+2) to be 10 or less.