Each multiple-choice quiz problem is based on a "root question," from which the system generates different correct and incorrect choices each time you take the quiz. Thus, you can test yourself on the same material multiple times. We strongly urge you to continue testing on each topic until you complete the quiz with a perfect score at least once. Simply click the "Reset" button at the bottom of the page for a new variant of the quiz.

After submitting your selections, the system will score your quiz, and for incorrect answers will provide an "explanation" (sometimes for correct ones too). These explanations should help you get the right answer the next time around. To prevent rapid-fire guessing, the system enforces a minimum of 10 minutes between each submission of solutions.

---

## Multiple Choice

7/8 points (graded)

[Q1] Consider the following SQL table declaration:

```
CREATE TABLE R (a INT, b INT, c INT, CHECK( [fill-in] ));
```

Currently R contains the tuples (1,4,14), (2,3,15), and (3,3,16). Which of the following tuple-based CHECK constraints will cause the following insertion to be rejected?

```
INSERT INTO R VALUES (4,4,9);
```

Note: When a tuple-based check is invoked for an insert and includes a subquery over the same table, the subquery is evaluated on the table *including* the inserted tuple.

- ○ c > ALL (SELECT a + b FROM R)
- [✓ c >= (SELECT SUM(b) FROM R)]
- ○ c <= ALL (SELECT b + c FROM R)

○ b > (SELECT AVG(a) FROM R)

[Q2] Consider the following trigger over a table R(a,b), specified using the trigger language of the SQL standard:

```
CREATE TRIGGER Rins
AFTER INSERT ON R
REFERENCING NEW ROW AS new
FOR EACH ROW
INSERT INTO R(a,b)
  (SELECT DISTINCT R.a, new.b
   FROM R
   WHERE R.b = new.a)
  EXCEPT
  (SELECT DISTINCT a,b FROM R)
```

Suppose table R is empty initially. We issue three commands to insert tuples into R: first we insert (1,2), then we insert (2,3), then we insert (3,4). After some of these inserts, trigger **Rins** may insert further tuples into R, which may activate the trigger recursively. After all the inserts are done, which of these tuples is NOT in table R?

[✓ (4,3)]

○ (2,4)

○ (1,3)

○ (1,4)

[Q3] Consider the following SQL table declaration:

```
CREATE TABLE Emps (id INT, ssNo INT, name CHAR(20), managerID INT);
```

We would like to extend the table declaration to enforce that each of **id** and **ssNo** is a key (by itself), and each value of **managerID**must be one of the values that appears in the **id** attribute of the same table. Which of the following is *not* a legal addition of SQL standard key and/or foreign-key constraints? Note: The addition does not have to achieve all of the stated goals; it only must result in legal SQL.

○ Add "UNIQUE" after each of the first two INT's.

○ Add "PRIMARY KEY" after the first INT, and add "UNIQUE" after the second INT.

[✓ ● Add "UNIQUE" after each of the first two INT's, and add "REFERENCES Emps(id,ssNo)" before the closing parenthesis.]

○ Add "PRIMARY KEY" after the first INT, and add ", FOREIGN KEY (managerID) REFERENCES Emps(id)" before the closing parenthesis.

**Answer-Selection Feedback**

A foreign key must reference a declared key. In this case, both **id** and **ssNo** are declared as keys by themselves, but (id,ssNo) is not declared to be a key (even though it must be one).

[Q4] Here are SQL declarations for two tables S and T:

```
CREATE TABLE S(c INT PRIMARY KEY, d INT);
CREATE TABLE T(a INT PRIMARY KEY, b INT REFERENCES S(c));
```

Suppose S(c,d) contains four tuples: (2,10), (3,11), (4,12), (5,13). Suppose T(a,b) contains four tuples: (0,4), (1,5), (2,4), (3,5). As a result of the constraints in the table declarations, certain insertions, deletions, and/or updates on S and T are disallowed. Which of the following modifications will *not* violate any constraint?

○ Inserting (0,1) into S

○ Inserting (5,12) into S

○ Inserting (2,5) into T

● Inserting (5,6) into T

**Answer-Selection Feedback**

It violates the foreign-key constraint, since 6 is not a current value of S.c.

[Q5] Here are SQL declarations for two tables S and T:

```
    CREATE TABLE S(c INT PRIMARY KEY, d INT);
    CREATE TABLE T(a INT PRIMARY KEY, b INT, CHECK(b IN (SELECT c FROM S)));
```

Suppose S(c,d) contains the four tuples: (2,10), (3,11), (4,12), (5,13). Suppose T(a,b) contains the four tuples: (0,4), (1,5), (2,4), (3,5). As a result of the constraints in the table declarations, certain insertions, deletions, and/or updates on S and T are disallowed. Which of the following modifications will *not* violate any constraint?

○ Inserting (5,0) into T

● Inserting (7,1) into S

○ Updating (2,4) in T to be (2,8)

○ Inserting (3,3) into T

[Q6] Consider the following trigger over a table R(a,b), specified using the trigger language of the SQL standard:

```
CREATE TRIGGER Rins
AFTER INSERT ON R
REFERENCING NEW ROW AS new
FOR EACH ROW
WHEN (new.a * new.b > 10)
INSERT INTO R VALUES (new.a - 1, new.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. Suppose we begin with table R empty. Consider the following possible tuples inserted into R. After trigger execution completes, which of the insertions results in R containing exactly 3 tuples?

- ○ (5,3)
- ○ (2,9)
- ○ (5,4)
- [✓ (2,50)]

[Q7] Here are SQL declarations for three tables R, S, and T:

```
CREATE TABLE R(e INT PRIMARY KEY, f INT);
CREATE TABLE S(c INT PRIMARY KEY, d INT REFERENCES R(e) ON DELETE CASCADE);
CREATE TABLE T(a INT PRIMARY KEY, b INT REFERENCES S(c) ON DELETE CASCADE);
```

Suppose R(e,f) contains tuples (1,0), (2,4), (3,5), (4,3), and (5,7). Suppose S(c,d) contains tuples (1,5), (2,2), (3,3), (4,5), and (5,4). Suppose T(a,b) contains tuples (0,2), (1,2), (2,3), (3,4), and (4,4). As a result of the referential integrity actions in the table declarations, certain deletions may cause additional deletions to be performed automatically. Which of the following deletions, after all integrity actions, leaves table T empty?

- [✓ delete from R where f>3]

- delete from R where e+f<=8
- delete from R where e=5 or f=5
- delete from R where e*f>=10

---

[Q8] Here are SQL declarations for three tables R, S, and T:

```
CREATE TABLE R(e INT PRIMARY KEY, f INT);
CREATE TABLE S(c INT PRIMARY KEY REFERENCES R(e) ON UPDATE CASCADE, d INT);
CREATE TABLE T(a INT PRIMARY KEY, b INT REFERENCES S(c) ON UPDATE CASCADE);
```

Suppose R(e,f) contains tuples (1,1), (3,4), (5,6), and (7,2). Suppose S(c,d) contains tuples (1,7), (3,2), (5,1) and (7, 5). Suppose T(a,b) contains tuples (1,1), (2,5), (3,5), and (4,3). As a result of the referential integrity actions in the table declarations, certain updates may cause additional updates to be performed automatically. Which of the following updates, after all integrity actions, leaves table T in a state such the sum of its **b** values is greater than 11 but less than 18?

- [✓ update R set e=e+3 where e<3]
- update R set e=e-2 where f<8
- update R set e=e-3 where f>5
- update R set e=e+3 where e>=1

Submit