

1. Question ID 596: Suppose we start with the following relation-schema declaration:

```
CREATE TABLE Emps (  
(1)      id INT,  
(2)      ssNo INT,  
(3)      name CHAR(20),  
(4)      managerID INT  
);
```

The numbers (1) through (4) are not part of the declaration, but will be used to refer to certain lines. Our intent is that both `id` and `ssNo` are keys by themselves, and that the value of `managerID` must be one of the values that appears in the `id` attribute of the same relation `Emps`. 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** the goals stated; it must only result in legal SQL.

The correct answers (i.e., incorrect SQL) violate one of two rules:

Question

Explanation:

1. You cannot have two primary keys (although you can have many "uniques").
2. A foreign key can only reference a key, primary or unique.

2. Question ID 597: Here are declarations of two relations R and S:

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

`R(a,b)` currently contains the four tuples (0,4), (1,5), (2,4), and (3,5). `S(c,d)` currently contains the four tuples (2,10), (3,11), (4,12), and (5,13). As a result, certain insertions and deletions on R and S are illegal. You should develop simple tests for illegal operations of these four kinds. Then, show your understanding by indicating which of the following modifications will **not** violate any constraint.

Question Explanation: The fact that `c` is a key of S says we cannot insert into S any tuple with first component 2, 3, 4, or 5.

The fact that `a` is a key of R says we cannot insert into R any tuple

with first component 0, 1, 2, or 3.

The fact that b is a foreign key referencing $S.c$ puts two constraints on us. First, we cannot insert into R any tuple whose second component is not one of 2, 3, 4, or 5. Second, we cannot delete from S any tuple whose first component is a second component of R , that is, 4 or 5.

3. Question ID 598: Here are declarations of two relations R and S :

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

$R(a,b)$ currently contains the four tuples (0,4), (1,5), (2,4), and (3,5). $S(c,d)$ currently contains the four tuples (2,10), (3,11), (4,12), and (5,13). As a result, certain insertions and deletions on S are illegal, as are certain updates or insertions on R . You should develop simple tests for illegal operations of these four kinds. Then, show your understanding by indicating which of the following modifications will **not** be rejected because of a constraint violation.

Question Explanation: The fact that c is a key of S says we cannot insert into S any tuple with first component 2, 3, 4, or 5.

The fact that a is a key of R says we cannot insert into R any tuple with first component 0, 1, 2, or 3.

The CHECK constraint says we cannot insert into R a tuple whose second component is not one of 2, 3, 4, or 5. Neither can we update a tuple to have a value of $R.b$ that is none of 2, 3, 4, or 5.

4. Question ID 599: Relation R has schema:

```
CREATE TABLE R (  
    a INT PRIMARY KEY,  
    b INT DEFAULT 0,  
    c INT NOT NULL  
);
```

R is currently empty. Develop a test that determines whether an insertion into R is currently legal. Then apply your test to determine which of the following INSERT statements is allowable.

Question Explanation: The DEFAULT condition has no affect on the legality of inserting a tuple. The NOT NULL condition does require that we provide a value of c in every inserted tuple. Another requirement is that we provide a value for the primary-key attribute a ; that is, "PRIMARY KEY" implies "NOT NULL."

One additional potential error involves the use of NULL as a value in an inserted tuple. The SQL standard does not allow this option - -- you need to omit a value for that component in the VALUES clause. However, some commercial DBMS's allow this option.

5. Question ID 712: Relation SPP(supID, partID, projID, qty) is a relation whose tuples mean that supplier supID supplies part partID to project projID in quantity qty. Suppose that each of the four attributes are integers. Write a CREATE TABLE statement for relation SPP. Do not forget to declare that supID, partID, and projID together form the key. and that these are each foreign keys referencing the relations Suppliers, Parts, and Projects, respectively. Then, identify from the list below the element that would appear as part of the definition.

Question Explanation: Here is a typical definition:

```
CREATE TABLE SPP (  
    supID    INTEGER,  
    partID   INTEGER,  
    projID   INTEGER,  
    qty      INTEGER,  
    PRIMARY KEY (supID, partID, projID),  
    FOREIGN KEY supID REFERENCES Suppliers,  
    FOREIGN KEY partID REFERENCES Parts,  
    FOREIGN KEY projID REFERENCES Projects,  
)
```