Tutorial 9 (Solution)
SQL

Classroom Exercise

Question 1

(i)

```
CREATE ASSERTION NoRedAndGreenParts CHECK
(NOT EXISTS
   (
       (SELECT       C.sid
       FROM  Catalog C, Parts P
       WHERE         C.pid = P.pid AND P.color = “red”)
       INTERSECTS
       (SELECT       C.sid
       FROM  Catalog C, Parts P
       WHERE         C.pid = P.pid AND P.color = “green”)
   )
);
```

(ii)

```
CREATE ASSERTION NoLowerPriceThanSid1 CHECK
(NOT EXISTS
       ( SELECT      *
         FROM        Catalog C1, Catalog C2
         WHERE       C1pid = C2.pid AND
                     C2.sid = 1 AND
                     C1.price < C2.price
       )
);
```

Question 2

The following two triggers are needed. Note the difference between the two.

```
CREATE TRIGGER fd_enforcer_update
BEFORE UPDATE on R
REFERENCING OLD ROW AS old
REFERENCING NEW ROW AS new
FOR EACH ROW
DECLARE counter INT
BEGIN
  SELECT COUNT(*) INTO counter
  FROM R
  WHERE R.A = NEW.A AND R.B = NEW.B AND R.C <> NEW.C AND
       NOT (R.A = OLD.A AND R.B = OLD.B AND R.C =   OLD.C
       AND R.D = OLD.D AND R.E = OLD.E);
  IF (counter > 0)
       THEN raise_exception('AB->C on R was violated');
END;
```

```
CREATE TRIGGER fd_enforcer_insert
BEFORE INSERT on R
Reference new row as new
FOR EACH ROW
DECLARE counter INT
BEGIN
   SELECT COUNT(*) INTO counter
   FROM R
   WHERE R.A = NEW.A AND R.B = NEW.B AND R.C <> NEW.C;
IF (counter > 0 )
   THEN raise_exception('AB->C on R was violated');
END;
```

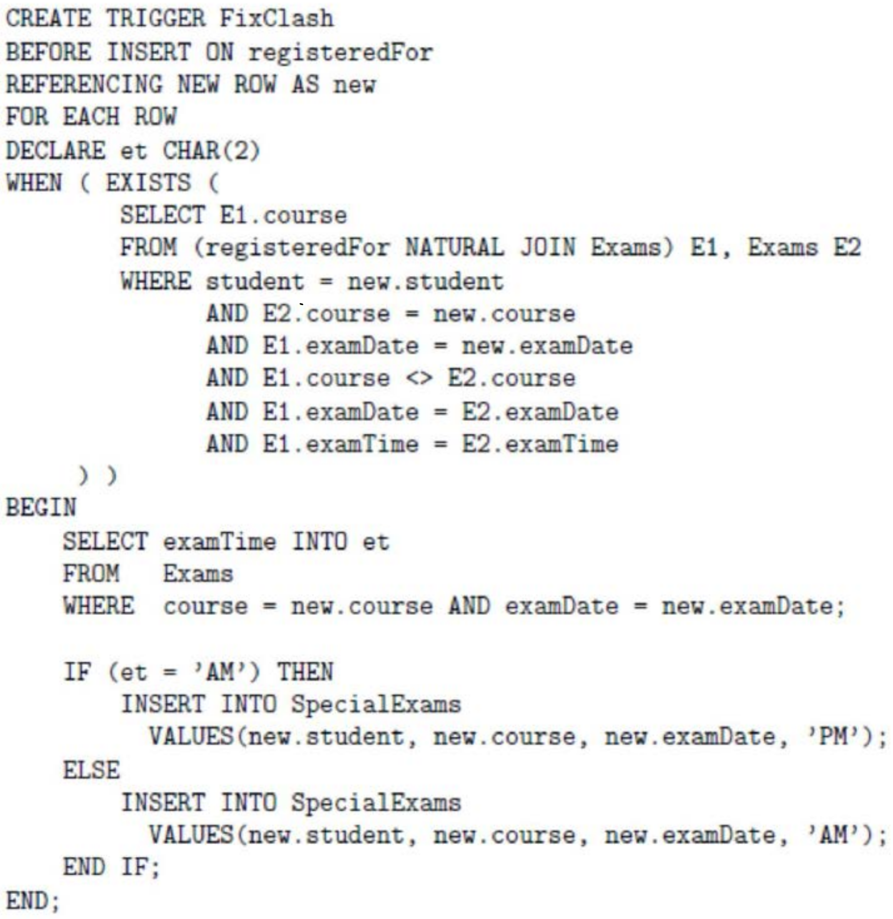
Question 3

```
CREATE TRIGGER SetPriceOfRemainingSeats
AFTER INSERT ON Booking
REFERENCING NEW ROW AS new
FOR EACH ROW
WHEN
        ( ( ( SELECT COUNT( BREF)
              FROM    Booking
              WHERE   flightNo = new.flightNo
```

```
                AND day = new.day
                AND month = new.month
                AND year = new.year ) -
                ( SELECT numSeats
                  FROM    Flight
                  WHERE   flightNo = new.flightNo
                  AND day = new.day
                  AND month = new.month
                  AND year = new.year ) ) > -20
               )

BEGIN
     UPDATE Flight
     SET     price = 4000
     WHERE   flightNo = new.flightNo
     AND day = new.day
     AND month = new.month
     AND year = new.year;
END;
```

Question 4



```
CREATE TRIGGER FixClash
BEFORE INSERT ON registeredFor
REFERENCING NEW ROW AS new
FOR EACH ROW
DECLARE et CHAR(2)
WHEN ( EXISTS (
        SELECT E1.course
        FROM (registeredFor NATURAL JOIN Exams) E1, Exams E2
        WHERE student = new.student
              AND E2.course = new.course
              AND E1.examDate = new.examDate
              AND E1.course <> E2.course
              AND E1.examDate = E2.examDate
              AND E1.examTime = E2.examTime
     ) )
BEGIN
    SELECT examTime INTO et
    FROM   Exams
    WHERE  course = new.course AND examDate = new.examDate;

    IF (et = 'AM') THEN
        INSERT INTO SpecialExams
          VALUES(new.student, new.course, new.examDate, 'PM');
    ELSE
        INSERT INTO SpecialExams
          VALUES(new.student, new.course, new.examDate, 'AM');
    END IF;
END;
```

Question 5

To enforce the foreign key constraint, both the deletion on course and the update/insertion on registration must be monitored. Consequently, the following two triggers are needed:

One is to implement the DELETE CASCASE:

```
CREASTE TRIGGER reference_constraint_on_course
AFTER DELETE ON COURSE
FOR EACH ROW
BEGIN
DELETE registration WHERE course_id = OLD.course_id;
END;
```

One is to make sure any registered course is indeed recorded in the course table.

```
CREASTE TRIGGER foregin_key_on_registration
BEFORE INSERT OR UPDATE ON registration
FOR EACH ROW
DECLARE counter INT
BEGIN
SELECT COUNT(*) INTO counter
FROM course
WHERE course_id = NEW.course_id;
IF (counter < 1 )
THEN raise_exception('the foreign key constraint violated
END;
```