

Assignment 3 Solutions:

Task 1:

```
set serveroutput on;
CREATE OR REPLACE TRIGGER trig1
  BEFORE INSERT ON SIGHTINGS
  REFERENCING NEW AS newRow
  FOR EACH ROW
  DECLARE
    cnt NUMBER;
  BEGIN
    SELECT count(LOCATION) INTO cnt FROM FEATURES WHERE LOCATION =
:newRow.LOCATION;
    IF cnt = 0 THEN
      INSERT INTO FEATURES VALUES(:newRow.LOCATION,
'UNKNOWN',NULL,NULL,NULL,NULL);
      DBMS_OUTPUT.PUT_LINE('Warning: Insert into the SIGHTINGS table
references location ' || :newRow.LOCATION || ' that is not found in the
database.');
```

.

```
    END IF;
  END trig1;
run;
```

Task 2:

```
set serveroutput on;
CREATE OR REPLACE TRIGGER trig2
  BEFORE INSERT ON SIGHTINGS
  REFERENCING NEW AS newRow
  FOR EACH ROW
  DECLARE
    cnt NUMBER;
    cname FLOWERS.COMNAME%TYPE;
    CURSOR flower_cur IS
    SELECT * FROM FLOWERS;
    flower_row flower_cur%ROWTYPE;

  BEGIN
    SELECT count(COMNAME) INTO cnt FROM FLOWERS WHERE COMNAME =
:newRow.NAME;
    IF cnt = 0 THEN
      FOR flower_row IN flower_cur LOOP
        IF (:newRow.NAME =flower_row.GENUS || ' ' ||
flower_row.SPECIES) THEN
          cname:=flower_row.COMNAME;
          EXIT;
        END IF;
      END LOOP;
      IF cname IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Warning: Your insert into the
SIGHTINGS table seemed to use the Latin name ' || :newRow.NAME || ' for the
flower ' || cname || '. I used the common name instead.');
```

.

```
        :newRow.Name:=cname;
      ELSE
        RAISE_APPLICATION_ERROR(-20999,'Error: Your insert into the
SIGHTINGS table uses an unknown flower name');
```

.

```
      END IF;
```

```

        END IF;
    END trig2;
.
run;

```

Task 3:

```

set serveroutput on;
CREATE OR REPLACE TRIGGER trig3
    BEFORE INSERT ON SIGHTINGS
    REFERENCING NEW AS newRow
    FOR EACH ROW
    DECLARE
        cnt NUMBER;
        matchednum NUMBER;
        i NUMBER;
        newloc SIGHTINGS.LOCATION%TYPE;
        ch1 CHAR;
        ch2 CHAR;
        CURSOR loc_cur IS
        SELECT LOCATION FROM FEATURES;
        loc_row loc_cur%ROWTYPE;
        loc FEATURES.LOCATION%TYPE;
        lname FEATURES.LOCATION%TYPE;
    BEGIN
        newloc:= :newRow.LOCATION;
        SELECT count(LOCATION) INTO cnt FROM FEATURES WHERE LOCATION = newloc;
        IF cnt = 0 THEN
            matchednum:=0;
            FOR loc_row IN loc_cur LOOP
                loc:=loc_row.LOCATION;
                IF LENGTH(loc)=LENGTH(newloc) THEN
                    i:=1;
                    WHILE i<=LENGTH(loc) AND
SUBSTR(loc,i,1)=SUBSTR(newloc,i,1) LOOP
                        i:=i+1;
                    END LOOP;
                    ch1:=SUBSTR(loc,i,1);
                    ch2:=SUBSTR(newloc,i,1);
                    IF (ch1='a' OR ch1='e' OR ch1='o' OR ch1='i' OR
ch1='u') AND (ch2='a' OR ch2='e' OR ch2='o' OR ch2='i' OR ch2='u') AND
SUBSTR(loc,i+1,LENGTH(loc)-i)=SUBSTR(newloc,i+1,LENGTH(loc)-i) THEN
                        matchednum:=matchednum+1;
                        lname:=loc;
                    ELSE
                        i:=i+1;
                        WHILE i<=LENGTH(loc) AND
SUBSTR(loc,i,1)=SUBSTR(newloc,i,1) LOOP
                            i:=i+1;
                        END LOOP;
                        IF i<=LENGTH(loc) AND ch2=SUBSTR(loc,i,1) AND
ch1=SUBSTR(newloc,i,1) AND SUBSTR(loc,i+1,LENGTH(loc)-
i)=SUBSTR(newloc,i+1,LENGTH(loc)-i) THEN
                            matchednum:=matchednum+1;
                            lname:=loc;
                        END IF;
                    END IF;
                END IF;
            END IF;
        END IF;
    END IF;

```

```

                ELSIF LENGTH(loc)-1=LENGTH(newloc) THEN
                    i:=1;
                    WHILE i<=LENGTH(loc) AND
SUBSTR(loc,i,1)=SUBSTR(newloc,i,1) LOOP
                        i:=i+1;
                    END LOOP;
                    IF SUBSTR(loc,i+1,LENGTH(loc)-
i)=SUBSTR(newloc,i,LENGTH(loc)-i) THEN
                        matchednum:=matchednum+1;
                        lname:=loc;
                    END IF;
                END IF;
            END LOOP;
            IF matchednum=1 THEN
                DBMS_OUTPUT.PUT_LINE('Warning: Your insert into the
SIGHTINGS table seemed to use an incorrect name ' || newloc ||' for the
location ' || lname || '. I used the correct name instead.');
```

:newRow.LOCATION:=lname;

```

            ELSE
                RAISE_APPLICATION_ERROR(-20999,'Error: Your insert into the
SIGHTINGS table uses an unknown location');
            END IF;
        END IF;
    END trig3;
.
run;
```

Task 4:

```

set serveroutput on;
/* The running time is up to 2 minutes for each execution */
/* X "dominates" Y */
CREATE TABLE XYTABLE (X VARCHAR2(50),Y VARCHAR2(50));
/*person dominates Y */
CREATE TABLE YTABLE (Y VARCHAR2(50));
CREATE VIEW FlowerSeen AS SELECT DISTINCT PERSON, NAME FROM SIGHTINGS;
CREATE VIEW ShareCount AS SELECT s1.PERSON AS X,s2.PERSON AS Y, COUNT(*) AS
SHARECNT FROM FlowerSeen s1, FlowerSeen s2 WHERE s1.NAME=s2.NAME GROUP BY
s1.PERSON , s2.PERSON;
CREATE VIEW TotalCount AS SELECT PERSON AS Y, COUNT(NAME) AS TOTALCNT FROM
FlowerSeen GROUP BY PERSON;
CREATE OR REPLACE PACKAGE Domination AS
    PROCEDURE GETTHEM (person VARCHAR2,p NUMBER);
END Domination;
.
run;
CREATE OR REPLACE PACKAGE BODY Domination AS
    PROCEDURE GETTHEM (person VARCHAR2,p NUMBER) IS
        cnt NUMBER;
        CURSOR y_cur IS
            SELECT * FROM YTABLE;
        y_row y_cur%ROWTYPE;
    BEGIN
        DELETE FROM XYTABLE;
        DELETE FROM YTABLE;
```

```

        INSERT INTO XYTABLE SELECT s3.X,s3.Y FROM ShareCount s3,
TotalCount s4 WHERE s3.Y=s4.Y AND s3.SHARECNT/s4.TOTALCNT>=p;
        INSERT INTO YTABLE VALUES(person);
        cnt:=1;
        WHILE cnt>0 LOOP
            SELECT COUNT(*) INTO cnt FROM XYTABLE WHERE X IN (SELECT *
FROM YTABLE) AND NOT (Y IN (SELECT * FROM YTABLE));
            INSERT INTO YTABLE SELECT DISTINCT Y FROM XYTABLE WHERE X
IN (SELECT * FROM YTABLE) AND NOT (Y IN (SELECT * FROM YTABLE));
            END LOOP;
            FOR y_row IN y_cur LOOP
                DBMS_OUTPUT.PUT_LINE(y_row.Y);
            END LOOP;
        END GETTHEM;
END Domination;
.
run;

```