**Task 1**

**Trigger:**

CREATE OR REPLACE TRIGGER task1\_insert\_sighting

BEFORE INSERT ON SIGHTINGS

FOR EACH ROW

DECLARE

counter number;

BEGIN

SELECT COUNT(location)

INTO counter

FROM features

WHERE location = :NEW.location;

IF counter = 0 THEN

INSERT INTO features VALUES (:NEW.location, 'UNKNOWN', NULL, NULL, NULL, NULL);

dbms\_output.put\_line('Warning: Insert into the SIGHTINGS table references location ' ||''''|| :NEW.location ||''''|| ' that is not found in the database');

END IF;

END;

**Output:**

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

NAME PERSON

------------------------------ ------------------------------

LOCATION SIGHTED

------------------------------ ---------

Douglas dustymaiden Person B

Double Mountain 28-NOV-05

Douglas dustymaiden Person A

Shirley Peak 18-AUG-06

Douglas dustymaiden Person B

Grouse Meadow 28-NOV-06

NAME PERSON

------------------------------ ------------------------------

LOCATION SIGHTED

------------------------------ ---------

Douglas dustymaiden Person C

Grouse Meadow 16-AUG-06

Douglas dustymaiden Person A

Piute 17-FEB-07

LOCATION CLASS LATITUDE

------------------------------ ------------------------------ ----------

LONGITUDE MAP ELEV

---------- ------------------------------ ----------

Piute UNKNOWN

**DBMS Output:**

Warning: Insert into the SIGHTINGS table references location 'Piute' that is not found in the database

**Task 2**

**Trigger:**

CREATE OR REPLACE TRIGGER task2\_insert\_sighting

BEFORE INSERT ON sightings

FOR EACH ROW

DECLARE

comname\_cnt number;

genus\_cnt number;

species\_cnt number;

newGen flowers.genus%TYPE;

newSpec flowers.species%TYPE;

newCName flowers.comname%TYPE;

BEGIN

SELECT COUNT(comname)

INTO comname\_cnt

FROM flowers

WHERE comname = :NEW.name;

IF comname\_cnt = 0 THEN

newGen := SUBSTR(:NEW.name, 1, INSTR(:NEW.name, ' ')-1);

newSpec := SUBSTR(:NEW.name, INSTR(:NEW.name, ' ')+1);

SELECT COUNT(genus)

INTO genus\_cnt

FROM flowers

WHERE genus = newGen;

SELECT COUNT(species)

INTO species\_cnt

FROM flowers

WHERE species = newSpec;

IF genus\_cnt > 0 AND species\_cnt > 0 THEN

BEGIN

SELECT comname

INTO newCName

FROM flowers

WHERE genus = newGen AND

species = newSpec;

END;

dbms\_output.put\_line('Your insert into the SIGHTINGS table seemed to use the Latin name ' ||''''|| :NEW.name ||''''|| ' for the flower ' ||''''|| newCName ||''''|| '. I used the common name instead.');

:NEW.name := newCName;

END IF;

END IF;

END;

**Output:**

Error starting at line : 1 in command -

INSERT

INTO SIGHTINGS

VALUES ('Sky pilot', 'Person X', 'Grouse Meadow', TO\_DATE('18-Aug-06', 'DD-MON-YY'))

Error report -

SQL Error: ORA-02291: integrity constraint (SYSTEM.FK1\_SIGHTINGS) violated - parent key not found

02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"

\*Cause: A foreign key value has no matching primary key value.

\*Action: Delete the foreign key or add a matching primary key.

Error starting at line : 5 in command -

INSERT

INTO SIGHTINGS

VALUES ('Hoar buckwheat', 'Person X', 'Grouse Meadow', TO\_DATE('18-Aug-06', 'DD-MON-YY'))

Error report -

SQL Error: ORA-02291: integrity constraint (SYSTEM.FK1\_SIGHTINGS) violated - parent key not found

02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"

\*Cause: A foreign key value has no matching primary key value.

\*Action: Delete the foreign key or add a matching primary key.

1 row inserted.

1 row inserted.

1 row inserted.

NAME PERSON

------------------------------ ------------------------------

LOCATION SIGHTED

------------------------------ ---------

Death camas Person X

Grouse Meadow 18-AUG-06

Mud sedge Person Y

Grouse Meadow 18-AUG-06

Draperia Person Z

Grouse Meadow 18-AUG-06

**DBMS Output:**

Your insert into the SIGHTINGS table seemed to use the Latin name 'Zigadenus venenosus' for the flower 'Death camas'. I used the common name instead.

Your insert into the SIGHTINGS table seemed to use the Latin name 'Carex limosa' for the flower 'Mud sedge'. I used the common name instead.

**Task 3  
Trigger:**

CREATE OR REPLACE TRIGGER task1\_insert\_sighting

BEFORE INSERT ON SIGHTINGS

FOR EACH ROW

DECLARE

counter number;

SpCh\_loc features.location%TYPE;

BEGIN

SELECT location

INTO SpCh\_loc

FROM features

GROUP BY location

HAVING UTL\_MATCH.EDIT\_DISTANCE(location, :NEW.location) =

(SELECT MIN(UTL\_MATCH.EDIT\_DISTANCE(location, :NEW.location)) AS minDist

FROM features) AND

UTL\_MATCH.EDIT\_DISTANCE(location, :NEW.location) <= 2;

SELECT COUNT(location)

INTO counter

FROM features

WHERE location = SpCh\_loc;

IF counter = 0 THEN

INSERT INTO features VALUES (SpCh\_loc, 'UNKNOWN', NULL, NULL, NULL, NULL);

dbms\_output.put\_line('Warning: Insert into the SIGHTINGS table references location ' ||''''|| :NEW.location ||''''|| ' that is not found in the database');

ELSE

:NEW.location := SpCh\_loc;

END IF;

END;

**Output:**

1 row inserted.

1 row inserted.

Error starting at line : 9 in command -

INSERT

INTO SIGHTINGS

VALUES ('Red mountain heather', 'Joe', 'Borwn Paek', TO\_DATE('18-Aug-06', 'DD-MON-YY'))

Error report -

SQL Error: ORA-01403: no data found

ORA-06512: at "SYSTEM.TASK1\_INSERT\_SIGHTING", line 5

ORA-04088: error during execution of trigger 'SYSTEM.TASK1\_INSERT\_SIGHTING'

01403. 00000 - "no data found"

\*Cause: No data was found from the objects.

\*Action: There was no data from the objects which may be due to end of fetch.

1 row inserted.

1 row inserted.

Error starting at line : 21 in command -

INSERT

INTO SIGHTINGS

VALUES ('Oak violet', 'Joe', 'Scodi Mountians', TO\_DATE('18-Aug-06', 'DD-MON-YY'))

Error report -

SQL Error: ORA-01403: no data found

ORA-06512: at "SYSTEM.TASK1\_INSERT\_SIGHTING", line 5

ORA-04088: error during execution of trigger 'SYSTEM.TASK1\_INSERT\_SIGHTING'

01403. 00000 - "no data found"

\*Cause: No data was found from the objects.

\*Action: There was no data from the objects which may be due to end of fetch.

1 row inserted.

1 row inserted.

NAME PERSON

------------------------------ ------------------------------

LOCATION SIGHTED

------------------------------ ---------

Leopard lily Joe

Frog Meadows Campground 18-AUG-06

Alpine sheep sorrel Joe

Lone Star Mine 18-AUG-06

Globe gilia Joe

The George Lodge 18-AUG-06

NAME PERSON

------------------------------ ------------------------------

LOCATION SIGHTED

------------------------------ ---------

Ithuriels spear Joe

San Emigdio Mountains 18-AUG-06

Diamond clarkia Joe

Camp Alto Campground 18-AUG-06

Broad-seeded rock-cress Joe

Brush Mountain 18-AUG-06

6 rows selected

**DBMS Output:**

**Task 4**

**Package:**

CREATE OR REPLACE PACKAGE Domination AS -- spec

PROCEDURE GetThem (input\_person VARCHAR2, percentage NUMBER);

END Domination;

/

CREATE OR REPLACE PACKAGE BODY Domination AS -- body

PROCEDURE GetThem (input\_person VARCHAR2, percentage NUMBER) AS

CURSOR people\_cur IS

SELECT DISTINCT person

FROM sightings;

people\_t people\_cur%ROWTYPE;

TYPE people\_ntt IS TABLE OF people\_t%TYPE;

l\_people people\_ntt;

matches NUMBER;

individ NUMBER;

BEGIN

OPEN people\_cur;

FETCH people\_cur BULK COLLECT INTO l\_people;

CLOSE people\_cur;

dbms\_output.put\_line('These people are ' || percentage\*100 || '% dominated by ' || input\_person || ':');

FOR i IN 1..l\_people.COUNT LOOP

--dbms\_output.put\_line('Current person: ' || l\_people(i).person);

-- Count flowers in common for input\_person and other people

SELECT COUNT(name)

INTO matches

FROM

(SELECT DISTINCT sightings.name, sightings.person

FROM sightings

INNER JOIN

(SELECT name

FROM sightings

WHERE person = input\_person) curr\_person\_sightings

ON sightings.name = curr\_person\_sightings.name)

WHERE person = l\_people(i).person;

--dbms\_output.put\_line('Matches: ' || matches);

SELECT COUNT(name)

INTO individ

FROM

(SELECT DISTINCT name, person

FROM sightings)

WHERE person = l\_people(i).person;

--dbms\_output.put\_line('Individual: ' || individ);

IF matches/individ >= percentage THEN

dbms\_output.put\_line(l\_people(i).person);

END IF;

END LOOP;

END;

END Domination;

/

**Output:**

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

**DBMS Output:**

These people are 30% dominated by Brad:

Brad

Tim

Pete

These people are 95% dominated by Brad:

Brad

These people are 95% dominated by Donna:

Donna

These people are 95% dominated by Sandra:

Sandra

Brad

These people are 95% dominated by Jennifer:

Michael

Robert

Joe

Helen

John

Brad

Jennifer

Donna

James

Tim

Pete

These people are 99% dominated by Michael:

Michael

Brad

Tim

Pete

These people are 50% dominated by Sandra:

Sandra

John

Brad

James

Tim

Pete

These people are 50% dominated by Jennifer:

Michael

Sandra

Robert

Joe

Maria

Helen

John

Brad

Jennifer

Donna

James

Tim

Pete