Triggers

The following relations are given (primary keys are underlined, optional attributes are denoted with *):

GREENHOUSE(GHCode, Location, Sensor#)
SENSOR(SCode, Measurement, GHCode, CriticalOffset)
EVENT-LOG(ECod, TimeStamp, EventType, SCode, Value)
MEASURE(SCode, TimeStamp, Value)
SENSOR-DAILY-SUMMARY(Date, SCode, AVGValue)
NOTIFICATION(NCode, GHCode, Location, Message)

Esercise #1
Write the trigger managing new events of measure type (EventType='M'). When a new event of measure type occurs, the new measure has to be inserted in the MEASURE table. Furthermore, it is necessary to check if a critical condition has occurred in the greenhouse where the corresponding sensor is located.
A condition is critical when more than half of the sensors in the considered greenhouse have sensed an exceptional value for the last sensed measurement. A measurement is exceptional if the difference between the last sensed measurement and the daily average for the sensor is greater than CriticalOffset. The daily average for each sensor is stored in the SENSOR-DAILY-SUMMARY table. This table is automatically updated by a trigger not considered in this application. To extract the date from the TimeStamp attribute, a generic function DATE(TimeStamp) can be exploited.
When a critical condition occurs, a notification request is inserted in the NOTIFICATION table. NCode is a counter, which should be incremented each time a new notification is inserted in the NOTIFICATION table.

Esercise #2
Write a trigger to verify the correctness of a new event of measure type (EventType='M') inserted in the EVENT_LOG table. For the sensors which measure the temperature (Measurement='Temperature'), the temperature measure cannot be lower than -50. If the value is lower assign -50 to the Value attribute in the EVENT-LOG table.

Esercise #3
Write a trigger that implements the constraint that a greenhouse cannot have more than 200 sensors with critical offset larger than 50.

Exercise #1 - Draft solution

```
CREATE OR REPLACE TRIGGER NewMeasure
AFTER INSERT ON EVENT_LOG
FOR EACH ROW
WHEN (NEW.EventType = 'M')
DECLARE
        GH NUMBER;
        LOC VARCHAR(10);
        TotSensors NUMBER;
        NOK NUMBER;
        X NUMBER;
BEGIN

---insert the new measure
        INSERT INTO MEASURE(SCode, TimeStamp, Value)
        VALUES (:NEW.SCode, :NEW.TimeStamp, :NEW.Value);

---read the information about the greenhouse where the sensor is located
        SELECT SE.GHCode, Location, Sensor# INTO GH, LOC, TotSensors
        FROM SENSOR SE, GREENHOUSE G
        WHERE SE.SCode = :NEW.SCode AND SE.GHCode = G.GHCode;

--- count the number of sensors in the considered greenhouse than have sensed an exceptional value
--- for the last sensed measurement
        SELECT COUNT(*) INTO NOK
        FROM SENSOR SE, MEASURE M, SENSOR-DAILY-SUMMARY S
        WHERE SE.GHCode = GH
        AND M.SCode = SE.SCode AND DATE(M.TimeStamp) = DATE(:NEW.TimeStamp)
        AND M.TimeStamp = (SELECT MAX(M1.TimeStamp)
                           FROM MEASURE M1
                           WHERE M1.SCode = M.SCode)
        AND S.SCode = SE.SCode AND S.DATE = DATE (:NEW.TimeStamp)
        AND M.Value > S.AVGValue + SE.CriticalOffset;

--- check if the condition in the considered greenhouse is critical
        IF (NOK > TotSensors/2) THEN
                SELECT MAX(NCode) INTO X
                FROM NOTIFICATION;

                IF (X IS NULL) THEN
                        X := 1;
                ELSE
                        X := X+1;
                END IF;

                INSERT INTO NOTIFICATION(NCode, GHCode, Location, Message)
                VALUES (X, GH, LOC, 'Critical situation');
        END IF; END;
```

Exercise #2 - Draft solution

Event: INSERT ON EVENT_LOG
Condition:

- EventType = 'M' AND Value < -50 (specified in the "WHEN" clause)
- the condition that the sensor measures the temperature has to be checked in the trigger body

Action: correct the measure by assigning Value := -50

```
CREATE OR REPLACE TRIGGER CorrectValue
BEFORE INSERT ON EVENT_LOG
FOR EACH ROW
WHEN (NEW.EventType = 'M' AND NEW.Value < -50)
DECLARE
        M       VARCHAR(10);
BEGIN

        SELECT Measurement INTO M
        FROM SENSOR
        WHERE SCode = :NEW.SCode;

        IF (M = 'Temperature') THEN
                :NEW.Value := -50;
        END IF;
END;
```

Exercise #3- Draft solution

Constraint: Greenhouses that do not satisfy the constraint

```
SELECT GHCode
FROM SENSOR
WHERE CriticalOffset > 50
GROUP BY GHCode
HAVING COUNT(*) > 200
```

Event: events that can violate the constraint

```
INSERT ON SENSOR
UPDATE OF GHCode ON SENSOR
UPDATE OF CriticalOffset ON SENSOR
```

Condition: No

Action: rollback

A statement trigger is used.

```
CREATE OR REPLACE TRIGGER CheckGreenhouse
AFTER INSERT OR UPDATE OF GHCode, CriticalOffset ON SENSOR
DECLARE
        X       NUMBER;
BEGIN

---count the number of critical greenhouses
        SELECT COUNT(*) INTO X
        FROM GREENHOUSE
        WHERE GHCode IN
                (SELECT GHCode
                FROM SENSOR
                WHERE CriticalOffset > 50
                GROUP BY GHCode
                HAVING COUNT(*) > 200);

        IF (X < > 0) THEN
                raise_application_error (-20500, 'Constraint violated');
        END IF;
END;
```

SENSOR

SCODE	GH CODE	CRITICAL OFFSET
S1	G1	10
S2	G1	30
S3	G2	40

MEASURE

Scode	TIMESTAMP	VALUE	
S1	1/11/11 8:00	3	
S1	2/11/11 10:00	5	
S1	2/11/11 11:00	5	||
S2	2/11/11 10:00	4	||
S3	2/11/11 8:00	10	

GH = G1

2/11/11

SUMMARY

DATE	SCODE	AVG VALUE	
1/11/11	S1	3	
2/11/11	S1	5	||
2/11/11	S2	4	||

CONDITION:

VALUE - AVG VALUE > CRITICAL OFFSET