

# Esercizio (1)

- Sono date le relazioni seguenti (le chiavi primarie sono sottolineate):  
SERRA(CodSerra, Locazione, NumSensori)  
SENSORE(CodSensore, GrandezzaMisurata, CodSerra, OffsetCritico)  
LOG-EVENTI(CodE, TimeStamp, TipoEvento, CodSensore, Valore)  
MISURA(CodSensore, TimeStamp, Valore)  
SINTESI-GIORNO(Data, CodSensore, MediaValore)  
NOTIFICA(CodN, CodSerra, Locazione, Messaggio)
- Si scriva il trigger che gestisce l'arrivo di un nuovo evento di tipo misura (*TipoEvento='M'*). Per questo tipo di eventi, occorre inserire una nuova misura nella tabella MISURA. Inoltre, occorre verificare se si è verificata una situazione critica nella serra dove si trova il sensore che ha effettuato la misura.
- Una situazione è critica *se più di metà dei sensori presenti nella serra considerata hanno un valore eccessivo per l'ultima misura effettuata in ordine di tempo*. Il valore di una misura è eccessivo se è maggiore della misura media del giorno per quel sensore di una quantità superiore a OffsetCritico. La misura media del giorno è memorizzata nella tabella SINTESI-GIORNO per ogni sensore. Questa tabella è mantenuta automaticamente aggiornata da un trigger non considerato nella presente applicazione. Per estrarre la data dall'attributo TimeStamp, si utilizzi una generica funzione DATE(TimeStamp).
- Se si verifica una situazione critica, occorre inserire una richiesta di notifica nella tabella NOTIFICA. Il codice identificativo univoco CodN è un contatore che deve essere incrementato ogni volta che è inserita una nuova notifica.

```
CREATE OR REPLACE TRIGGER NUOVA MISURA  
AFTER INSERT ON LOG-EVENTI  
FOR EACH ROW  
WHEN (NEW.TipoEvento = 'M')
```

```
DECLARE  
NC, CS, NS NUMBER; LOCALIZIONE VARCHAR(20);  
CN NUMBER;  
BEGIN
```

```
INSERT INTO MISURA (Cod Sensore, Timestamp, Valore)  
VALUES (:NEW.CodSensore, :NEW.Timestamp, :NEW.Valore);
```

-- Leggo le info relative alle server in cui si trova  
-- il sensore

```
SELECT S.CodServo, SE.NumSensori, SE.Locazione  
INTO  
CS, NS, Locazione
```

```
FROM SENSORE S, SERRA SE
```

```
WHERE S.CodServo = SE.CodServo AND  
S.CodSensore = :NEW.CodSensore;
```

- Contare il numero di sensori presenti nelle
- serre CS che hanno rilevato un valore
- eccessivo per l'ultima misura

SELECT COUNT(\*) INTO NC

FROM MISURA M, SINTESI-GIORNO SG,  
SENSORE SE

WHERE M.CodSensore = SE.CodSensore AND  
SG.CodSensore = M.CodSensore AND  
SG.Data = DATE(M.Timestamp) AND  
DATE(M.Timestamp) = DATE(:NEW.Timestamp) AND

Cond.  
Join

Valore > MediaValori + OffsetCritico AND

SE.CodSensore = CS AND

M.Timestamp = (SELECT MAX(Timestamp)

FROM MISURA M1  
WHERE M1.CodSensore = M.CodSensore)

Cond. di  
criticita  
Cond.  
Seme

Ultima  
misura  
x ogni CodSensore

```
IF (NC > NS/2) THEN
```

```
  SELECT MAX(CODN) INTO CN  
  FROM NOTIFICA;
```

```
IF (CN IS NULL) THEN
```

```
  CN := 1;
```

```
ELSE
```

```
  CN := CN + 1;
```

```
END IF;
```

```
INSERT INTO NOTIFICA (
```

```
  VALUES (CN, CS, LOCAZIONE, "situazione critica");
```

```
END IF;
```

```
END;
```