Basi di dati

Esercizi sui trigger
Assegnazione di borse di studio per attività di supporto alla didattica

Esercizio 1

- Sono date le relazioni seguenti (le chiavi primarie sono sottolineate, gli attributi opzionali hanno l'asterisco)

  STUDENTE(Matricola, NomeStudente, AnnoImmatricolazione, CorsoLaurea)

  CORSO(CodCorso, NomeCorso, NumeroCrediti)

  ESAMI_SOSTENUTI(CodCorso,Matricola, Data, Voto)

  GRADUATORIA_STUDENTI(Matricola, Punteggio)

  BORSE_STUDIO_ASSEGNATE(CodBorsa, Matricola, NumeroOre)

  DOMANDA_INSERIMENTO_GRADUATORIA(Matricola, DataDomanda)

  OFFERTA_BORSA_STUDIO(CodBorsa,CodCorso,NumeroOre)

  NOTIFICA_INFORMAZIONI(CodN, CodBorsa, Matricola*, Messaggio)

- Si scriva il trigger per la verifica di correttezza e eventuale correzione del seguente vincolo. Per ciascuna borsa di studio offerta, il numero di ore previsto deve essere pari ad almeno 15 ore. Se viene offerta una borsa di studio con un numero di ore inferiore a 15, il valore deve essere assegnato a 15.

```
CREATE OR REPLACE TRIGGER verifica_ore
BEFORE INSERT OR UPDATE of NumeroOre on OFFERTA_BORSA_STUDIO
FOR EACH ROW
When (NEW.NumeroOre < 15)
BEGIN
        :NEW.NumeroOre := 15;
END IF;
END;
```

- Si scriva il trigger che verifica il seguente vincolo: per ciascun corso le borse di studio offerte per il corso non possono superare complessivamente un monte ore pari a 300.

```
CREATE OR REPLACE TRIGGER verifica_ore_complessive
AFTER INSERT OR UPDATE OF CodCorso, NumeroOre ON OFFERTA_BORSA_STUDIO
DECLARE
N NUMBER;
BEGIN
SELECT COUNT(*) INTO N
FROM CORSO [oppure OFFERTA_BORSA_STUDIO]
WHERE CodCorso IN
        (SELECT CodCorso
        FROM OFFERTA_BORSA_STUDIO
        GROUP BY CodCorso
        HAVING SUM(NumOre) > 300);

IF (N <>0) THEN
   raise_application_error (XXXX, «corso con troppe ore assegnate»);
END IF;
END;
```

- *Inserimento di uno studente in graduatoria.* Lo studente presenta la domanda per l'assegnazione di borse di studio. La domanda viene accettata se lo studente non è già presente in graduatoria (tabella GRADUATORIA) ed ha acquisito almeno 120 crediti sugli esami superati. Se almeno uno dei requisiti non è soddisfatto, la domanda viene annullata. Altrimenti si deve aggiornare la graduatoria, assegnando allo studente un punteggio dato dal prodotto della media dei voti per gli esami superati (votazione maggiore o uguale a 18), per il numero di anni di iscrizione dello studente al corso di laurea (si assuma che l'anno corrente sia fornito dalla variabile SYSDATE).

Evento: INSERT ON DOMANDA_INSERIMENTO_GRADUATORIA

Condizione: inserita nel corpo del trigger (causa complessità)

Semantica: AFTER (Business Rule)

Granularità: ROW (elaboro una domanda alla volta)

Azioni:

1- Verificare la presenza dello studente in graduatoria

IF *Presente* THEN ERRORE

2- Verificare il numero di crediti acquisiti

IF *Troppo pochi* THEN ERRORE

3- Calcolo punteggio per inserimento

INSERT in graduatoria

```
CREATE OR REPLACE TRIGGER Nuova_domanda
AFTER INSERT ON DOMANDA_INSERIMENTO_GRADUATORIA
FOR EACH ROW
DECLARE
   N NUMBER;
   TotCrediti NUMBER;
   Media NUMBER;
   Anno NUMBER;
...
```

```
BEGIN
-- verifico se lo studente è già in graduatoria
        SELECT COUNT(*) INTO N
        FROM GRADUATORIA_STUDENTI
        WHERE Matricola = :NEW.Matricola;
        IF (N<>0) THEN
          -- studente già in graduatoria
          RAISE_APPLICATION_ERROR(…, 'Studente già in graduatoria')
        END IF;
…
```

```
…
-- verifico se lo studente ha acquisito almeno 120 crediti
        SELECT SUM(NumeroCrediti), AVG(Voto) INTO TotCrediti,
        Media
        FROM ESAMI_SOSTENUTI E, CORSO C
        WHERE E.CodCorso = C.CodCorso AND Voto >= 18
        AND Matricola = :NEW.Matricola;
        IF (TotCrediti IS NULL OR TotCrediti < 120) THEN
          -- numero di crediti acquisiti insufficiente oppure non sono stati
          -- superati alcuni esami
          RAISE_APPLICATION_ERROR(…, ‘Crediti insufficienti’)
        END IF;
…
```

```
…
-- leggo l’anno di immatricolazione dello studente
        SELECT AnnoImmatricolazione INTO Anno
        FROM STUDENTE
        WHERE Matricola = :NEW.Matricola;
-- inserisco lo studente in graduatoria
        INSERT INTO GRADUATORIA_STUDENTI(Matricola, Punteggio)
        VALUES (:NEW.Matricola, Media * (SYSDate-Anno));
END;
```

- *Assegnazione di una borsa di studio per un corso.* Quando viene offerta una borsa di studio per un corso, si seleziona dalla graduatoria lo studente a cui assegnare la borsa. Viene selezionato lo studente con punteggio più alto tra gli studenti che soddisfano i seguenti requisiti: lo studente ha superato l'esame per il corso per cui è offerta la borsa di studio, e lo studente complessivamente non svolge più di 150 ore sulle borse di studio assegnate. Si assuma che ci sia sempre al più un solo studente che soddisfi tutti i requisiti. Occorre notificare l'esito dell'operazione, sia che la borsa di studio sia stata assegnata, sia che non sia possibile assegnare la borsa (in questo caso, la matricola sarà NULL). L'attributo CodN è un contatore che viene incrementato ogni volta che viene inserita una nuova notifica. Se la borsa di studio è assegnata, si deve aggiornare la tabella BORSE_STUDIO_ASSEGNATE.

Evento: INSERT ON OFFERTA_BORSA_STUDIO

Condizione: inserita nel corpo del trigger (causa complessità)

Semantica: AFTER (Business Rule)

Granularità: ROW (elaboro una borsa alla volta)

Azioni:

1- Calcolo il valore del punteggio massimo tra gli studenti che soddisfano i requisiti (NULL se non esistono studenti che soddisfano)

2- Leggo il valore di CodN per eseguire la notifica

3- IF (Esiste uno studente eleggibile) THEN

Leggo la matricola dello studente, inserisco in BORSE_ASSEGNATE, notifico

ELSE

Notifico

```
CREATE OR REPLACE TRIGGER Offerta_borsa
AFTER INSERT ON OFFERTA_BORSA_STUDIO
FOR EACH ROW
DECLARE
   N NUMBER;
   Col NUMBER;
   Matr NUMBER;
…
```

```
BEGIN
-- calcolo il punteggio massimo (se esiste)
        SELECT MAX(Punteggio) INTO N
        FROM GRADUATORIA_STUDENTI
        WHERE Matricola IN
                (SELECT Matricola FROM ESAMI_SOSTENUTI
                 WHERE CodCorso = :NEW.CodCorso AND Voto >= 18)
        AND Matricola NOT IN
                (SELECT Matricola FROM BORSE_STUDIO_ASSEGNATE
                GROUP BY Matricola
                 HAVING SUM(NumeroOre) + :NEW.NumeroOre > 150);
...
```

```
...
-- leggo il valore massimo di CodN
        SELECT MAX(CodN) INTO Cod
      FROM NOTIFICA_INFORMAZIONI;
        IF (Cod IS NULL) THEN
       -- tabella vuota. Non ci sono notifiche
        Cod = 0;
        IF (N IS NOT NULL) THEN
        -- Esiste uno studente eleggibile: seleziono la sua matricola
                SELECT Matricola INTO Matr
                FROM GRADUATORIA_STUDENTI
                 WHERE CodCorso = :NEW.CodCorso AND Voto >= 18)
                AND Matricola NOT IN
                (SELECT Matricola FROM BORSE_STUDIO_ASSEGNATE
                GROUP BY Matricola
                 HAVING SUM(NumeroOre) + :NEW.NumeroOre > 150);
...
```

```
…
INSERT INTO BORSE_STUDIO_ASSEGNATE(…)
VALUES(:NEW.CodBorsa, Matr, :NEW.NumeroOre)
INSERT INTO NOTIFICA_INFORMAZIONI(CodN, CodBorsa, Matricola,
Messaggio)
VALUES(Cod+1,:NEW.CodBorsa, Matr, ‘Borsa Assegnata’);
ELSE
-- Non esiste alcun studente eleggibile
INSERT INTO NOTIFICA_INFORMAZIONI(CodN, CodBorsa, Matricola,
Messaggio)
VALUES (Cod+1, :NEW.CodBorsa, NULL, ‘Borsa non assegnata’);
END IF;
END;
```