

Basi di dati

Esercitazione #6 – Trigger - Bozza di soluzioni

ESERCIZIO 1

Base di dati prima dell'esecuzione del trigger

Tabella IMP

| EMPNO | DEPTNO | ENAME | JOB | SAL |
|-------|--------|-------|----------|------|
| 7000 | 10 | SMITH | CLERK | 850 |
| 7010 | 10 | SCOTT | ANALYST | 1000 |
| 7020 | 10 | BLAKE | SALESMAN | 1400 |
| 7030 | 10 | SMITH | MANAGER | 2500 |
| 7040 | 20 | SMITH | CLERK | 800 |
| 7050 | 20 | SCOTT | ANALYST | 1600 |
| 7060 | 30 | ADAMS | CLERK | 000 |
| 7070 | 30 | JAMES | CLERK | 1000 |
| 7080 | 40 | ALLEN | CLERK | 850 |

Tabella DIP

| DEPTNO | DNAME | LOC | MINSAL | MAXSAL |
|--------|------------|----------|--------|--------|
| 10 | ACCOUNTING | NEW YORK | 100 | 2500 |
| 20 | RESEARCH | DALLAS | 150 | 3000 |
| 30 | SALES | CHICAGO | 120 | 2500 |
| 40 | OPERATIONS | BOSTON | 200 | 2100 |

Trigger per la gestione dell'aggiornamento del campo DNAME in DIP

```
CREATE OR REPLACE TRIGGER UP_SAL
AFTER UPDATE OF DNAME ON DIP
FOR EACH ROW
WHEN (OLD.DNAME='ACCOUNTING' AND NEW.DNAME='SALES')
BEGIN
--- aggiorna il salario degli impiegati del dipartimento cambiato
UPDATE IMP
SET SAL=SAL+100
WHERE DEPTNO = :OLD.DEPTNO;
END;
/
```

Istruzione di aggiornamento

```
UPDATE DIP SET DNAME = 'SALES' WHERE DNAME='ACCOUNTING';
```

Base di dati dopo l'esecuzione del trigger

Tabella IMP

| DEPTNO | DNAME | LOC | MINISAL | MAXSAL |
|--------|------------|----------|---------|--------|
| 10 | SALES | NEW YORK | 100 | 2500 |
| 20 | RESEARCH | DALAS | 150 | 3000 |
| 30 | SALES | CHICAGO | 120 | 2500 |
| 40 | OPERATIONS | BOSTON | 200 | 2100 |

Tabella DIP

| EMPNO | DEPTNO | ENAME | JOB | SAL |
|-------|--------|-------|----------|------|
| 7000 | 10 | SMITH | CLERK | 950 |
| 7010 | 10 | SCOTT | ANALYST | 1700 |
| 7020 | 10 | BLAKE | SALESMAN | 1500 |
| 7030 | 10 | SMITH | MANAGER | 2600 |
| 7040 | 20 | SMITH | CLERK | 800 |
| 7050 | 20 | SCOTT | ANALYST | 1600 |
| 7060 | 30 | ADAMS | CLERK | 900 |
| 7070 | 30 | JAMES | CLERK | 1000 |
| 7080 | 40 | ALLEN | CLERK | 850 |

Commenti

Il trigger UP_SAL implementa la funzionalità richiesta ed è attivato in seguito ad un'operazione di *update* sull'attributo DNAME della tabella DIP (trigger di tipo *after*). Il trigger ha inoltre granularità di tupla in quanto occorre poter accedere all'attributo DNAME del record aggiornato (e di quello originale) per verificare che il tipo di aggiornamento sia quello richiesto ('ACCOUNTING' -> 'SALES').

ESERCIZIO 2

Base di dati prima dell'esecuzione del trigger

Tabella IMP

| EMPNO | ENAME | JOB | SAL |
|-------|---------|------------|------|
| 1 | VERDI | SEGRETARIA | 800 |
| 2 | ROSSI | BANCHIERE | 900 |
| 3 | BIANCHI | BANCHIERE | 1100 |

Tabella SUMMARY

| JOB | NUM |
|------------|-----|
| SEGRETARIA | 1 |
| BANCHIERE | 2 |

Trigger per la gestione dell'inserimento di un record in IMP

```
CREATE OR REPLACE TRIGGER INS_IMP
```

```
AFTER INSERT ON IMP
```

```
FOR EACH ROW
```

```
DECLARE
```

```
N NUMBER;
```

```
M NUMBER;
```

```
BEGIN
```

```
--- controlla se ci sono impiegati
```

```
--- che fanno lo stesso lavoro
```

```
SELECT COUNT(*) INTO N
```

```
FROM SUMMARY
```

```
WHERE JOB=:NEW.JOB;
```

```
IF(N=0) THEN
```

```
--- è il primo impiegato che
```

```
--- fa quel lavoro
```

```
INSERT INTO SUMMARY (JOB, NUM)
```

```
VALUES (:NEW.JOB, 1);
```

```
ELSE
```

```
--- esiste almeno un impiegato che
```

```
--- fa già quel lavoro
```

```
SELECT NUM INTO M
```

```
FROM SUMMARY
```

```
WHERE JOB=:NEW.JOB;
```

```
UPDATE SUMMARY
```

```
SET NUM=M+1
```

```
WHERE JOB=:NEW.JOB;
```

```
END IF;
```

```
END;
```

```
/
```

Istruzione di aggiornamento

```
INSERT INTO IMP(EMPNO, ENAME, JOB, SAL) VALUES(4, 'NERI', 'CORRIERE', 750);
```

Tabella IMP

| EMPNO | ENAME | JOB | SAL |
|-------|---------|------------|------|
| 1 | VERDI | SEGRETARIA | 800 |
| 2 | ROSSI | BANCHIERE | 900 |
| 3 | BIANCHI | BANCHIERE | 1100 |
| 4 | NERI | CORRIERE | 750 |

Tabella SUMMARY

| JOB | NUM |
|------------|-----|
| SEGRETARIA | 1 |
| BANCHIERE | 2 |
| CORRIERE | 1 |

Trigger per la gestione dell'aggiornamento del campo JOB in IMP

```
CREATE OR REPLACE TRIGGER UPD_IMP
AFTER UPDATE OF JOB ON IMP
FOR EACH ROW
DECLARE
N NUMBER;
M NUMBER;
BEGIN
--- conta quanti impiegati hanno il nuovo lavoro
SELECT COUNT(*) INTO N
FROM SUMMARY
WHERE JOB=:NEW.JOB;
---incrementa il numero di impiegati per il nuovo lavoro
IF (N=0) THEN
---l'impiegato inserito è il primo impiegato per il nuovo lavoro
INSERT INTO SUMMARY(JOB, NUM)
VALUES(:NEW.JOB, 1);
ELSE
---c'è almeno un altro impiegato per il nuovo lavoro
UPDATE SUMMARY
SET NUM = NUM +1
WHERE JOB = :NEW.JOB;
END IF;
SELECT NUM INTO M
FROM SUMMARY
WHERE JOB = :OLD.JOB;
IF (M=1) THEN
---c'era un solo impiegato per il vecchio lavoro. Cancella il record dalla tabella SUMMARY
DELETE FROM SUMMARY
WHERE JOB = :OLD.JOB;
ELSE
---decrementa NUM nel corrispondente record nella tabella SUMMARY
UPDATE SUMMARY
SET NUM = NUM -1
WHERE JOB = :OLD.JOB;
END IF;
END;
/
```

Istruzione di aggiornamento

```
UPDATE IMP SET JOB='CORRIERE' WHERE EMPNO=2;
```

Base di dati dopo l'esecuzione del trigger

Tabella IMP

| EMPNO | ENAME | JOB | SAL |
|-------|---------|------------|------|
| 1 | VERDI | SEGRETARIA | 800 |
| 2 | ROSSI | CORRIERE | 900 |
| 3 | BIANCHI | BANCHIERE | 1100 |
| 4 | NERI | CORRIERE | 750 |

Tabella SUMMARY

| JOB | HUM |
|------------|-----|
| SEGRETARIA | 1 |
| BANCHIERE | 1 |
| CORRIERE | 2 |

Commenti

Entrambi i trigger sono a granularità di tupla in quanto occorre avere visibilità sul campo JOB della tupla inserita o aggiornata in IMP. In questo modo è possibile verificare se, dopo la modifica su IMP, esiste almeno una tupla (e se si quante) con lo stesso valore di JOB nella tabella SOMMARIO. NOTA: Questa verifica non può essere effettuata eseguendo direttamente un conteggio delle tuple in IMP, in quanto in un trigger a granularità di tupla non è possibile accedere (in lettura/scrittura) alla tabella mutante.