

## ESERCIZI su TRIGGER, ASSERTZIONI e VISTE

Si consideri la seguente BD:

- **Imp**(matr, nome, stip, dip\*, superiore\*, livello)
- **Dip**(cod, nome, stip-tot, direttore\*, azienda\*)
- **Azienda**(cod, nome, città)

dove *superiore* e *direttore* sono chiavi secondarie definite sulla chiave primaria di Imp.

### TRIGGER

#### attributi calcolati

1. lo stipendio totale di un dipartimento deve essere aggiornato quando si inserisce un nuovo impiegato
2. lo stipendio totale di un dipartimento deve essere aggiornato quando un impiegato viene trasferito di dipartimento

#### integrità dei dati

3. il campo direttore di Dip deve essere messo a NULL quando la corrispondente tupla in Imp viene cancellata
4. Ogni Dip deve avere almeno un impiegato
5. Creare una regola che riduca del 10% lo stipendio di *tutti* gli impiegati di un Dip quando la media dei salari dei dipendenti di quel dip supera i 40.000€
6. La cancellazione di una azienda comporta la cancellazione dei suoi dipartimenti
7. Quando si cancella un dipendente, il campo *superiore* dei suoi subordinati va messo a NULL
8. Assegnare ad ogni nuovo impiegato lo stipendio minimo del suo dipartimento, a meno che non sia diversamente specificato

### ASSERTZIONI

1. Il direttore di un dipartimento deve essere un impiegato che afferisce allo stesso dipartimento
2. Un impiegato non può avere uno stipendio maggiore di quello del suo superiore
3. Ogni Dip deve avere almeno un impiegato
4. In un dipartimento non possono esserci più di 2 impiegati di livello I

### VISTE

1. Definire una vista per formulare la seguente interrogazione: per ogni azienda, numero medio di impiegati nei dipartimenti
2. Definire la vista ricorsiva Subordinato

## ATTRIBUTI CALCOLATI

1. lo stipendio totale di un dipartimento deve essere aggiornato quando si inserisce un nuovo impiegato- Esempio di operazione che scatena il trigger: Insert(252, aldo, 25.000, d1, 333)

```
CREATE TRIGGER Stip-Tot1
AFTER INSERT ON Impiegato           % evento
FOR EACH ROW
IF (NEW.dip IS NOT NULL)             % condizione
THEN UPDATE Dip                     % azione
    SET stip-tot = stip-tot+NEW.stip
    WHERE Dip.cod = NEW.dip)
```

2. lo stipendio totale di un dipartimento deve essere aggiornato quando un impiegato viene trasferito di dipartimento – Esempio di operazione che scatena il trigger:

```
Update Imp
Set dip = d2
Where matr=252
```

```
CREATE TRIGGER Stip-Tot2
AFTER UPDATE OF Dip ON Impiegato
FOR EACH ROW
    UPDATE Dip
    SET stip-tot = stip-tot - OLD.stip
    WHERE Dip.cod = OLD.dip;
    UPDATE Dip
    SET stip-tot = stip-tot + OLD.stip
    WHERE Dip.cod = NEW.dip; )
```

## INTEGRITA' DEI DATI

3. Il campo direttore di Dip deve essere messo a NULL quando la corrispondente tupla in Imp viene cancellata – Esempio di operazione che scatena il trigger: Delete Imp where matr=100

```
CREATE TRIGGER Integrità-Ref-Direttore
AFTER DELETE ON Impiegato
FOR EACH ROW
    UPDATE Dip
    SET direttore=NULL
    WHERE direttore=OLD.matr
```

4. Ogni dipartimento deve avere almeno un impiegato

```
CREATE TRIGGER Integrità-Dip
BEFORE Delete on IMP
FOR EACH ROW
IF (Select count(*) From Imp) == 1)
THEN RAISE ERROR
```

5. Creare una regola che riduca del 10% lo stipendio di *tutti* gli impiegati di un Dip quando la media dei salari dei dipendenti di quel dip supera i 40.000€

```
CREATE TRIGGER Rid-Salario
After Insert on IMP
For each row
IF (Select avg(stip) From Imp Where dip = NEW.dip > 40.000 )
THEN Update Imp
    Set stip = stip - 0,1*stip
    Where dip = NEW.dip
```

6. La cancellazione di una azienda comporta la cancellazione dei suoi dipartimenti

```
CREATE TRIGGER Integrità-Ref-Azienda
AFTER DELETE ON Azienda                                % evento
FOR EACH ROW
    DELETE Dip
    WHERE azienda=OLD.cod
```

Supponendo che la cancellazione di un dipartimento comporti la cancellazione dei suoi dipendenti attraverso il seguente trigger

```
CREATE TRIGGER Integrità-Ref-Dip
AFTER DELETE ON Dip                                    % evento
FOR EACH ROW
    DELETE Imp
    WHERE dip=OLD.cod
```

allora il trigger Integrità-Ref-Azienda innescia il trigger Integrità-Ref-Dip

7. Quando si cancella un dipendente, il campo *superiore* dei suoi subordinati va messo a NULL

```
CREATE TRIGGER Gestisci-Subordinati
AFTER DELETE ON Imp
FOR EACH ROW
    Update Imp
    Set superiore=NULL
    WHERE superiore = OLD.matr
```

8. assegnare ad ogni nuovo impiegato lo stipendio minimo del suo dipartimento, a meno che non sia diversamente specificato:

```
CREATE TRIGGER ImpStipMin
BEFORE INSERT ON Impiegato
FOR EACH ROW
WHEN (New.Stip = NULL)
    SET New.Stip = (SELECT MIN(Stip) FROM Impiegato WHERE dip = New.dip)
```

Oppure (soluzione meno efficiente)

```
CREATE TRIGGER ImpStipMin
AFTER INSERT ON Impiegato
FOR EACH ROW
WHEN (New.Stip = NULL)
    UPDATE Imp
    SET New.Stip = (SELECT MIN(Stip) FROM Impiegato WHERE dip = New.dip),
    Where matr=new.matr
```

## ASSERZIONI

1. il direttore di un dipartimento deve essere un impiegato che afferisce allo stesso dipartimento

```
CREATE ASSERTION Vincolo- Direttore
CHECK (NOT EXISTS
    (SELECT *
    FROM Dip D, Imp I
    WHERE D.dir = I.matr and D.cod <> I.dip )
```

2. un impiegato non può avere uno stipendio maggiore di quello del suo superiore.

```
CREATE ASSERTION Vincolo-Stipendi
CHECK (NOT EXISTS
    (SELECT *
    FROM Imp as I
    WHERE stip > (SELECT stip
    FROM Imp as S
    WHERE I.superiore=S.matr)
```

5. Ogni Dip deve avere almeno un impiegato

```
CREATE ASSERTION Vincolo-Esistenza
CHECK (NOT EXISTS
    (SELECT *
    FROM Dip
    WHERE cod NOT IN (SELECT dip
    FROM Imp)
```

6. In un dipartimento non possono esserci più di 2 impiegati di livello I

```
CREATE ASSERTION Vincolo-numMax-Livelli
CHECK (NOT EXISTS
    (SELECT dip
    FROM Imp, Dip
    WHERE I.dip=D.cod and livello = I
    GROUPBY dip
    HAVING count(*) >2
    )
```

7. Esiste almeno un impiegato

```
CREATE ASSERTION Vincolo-Esistenza  
CHECK (EXISTS (SELECT * FROM Imp))
```

## VISTE

1. Definire una vista per formulare la seguente interrogazione: per ogni azienda, numero medio di impiegati nei dipartimenti

```
CREATE VIEW Info-Dip(azienda, num-imp) AS  
  SELECT azienda, count(*)  
  FROM Imp, Dip  
  WHERE cod=dip  
  GROUPBY azienda
```

```
QUERY  
SELECT azienda, avg(num-imp)  
FROM Info-Dip  
GROUPBY azienda
```

2. Definire la vista ricorsiva Subordinato

```
CREATE VIEW Subordinato(sub, sup, livello) AS
```

```
  SELECT matr, superiore, 1  
  FROM Imp  
  UNION  
  SELECT X.matr, Y.sup, Y.livello+1  
  FROM Imp as X, Subordinato as Y  
  WHERE X.sup=Y.sub
```