

## AGGIORNAMENTI IN SQL

Gli esempi seguenti faranno riferimento o alla BD1

- **Fornitori(codf, nome, età, città) -- F**
- **Prodotti(codp, nome, colore) -- P**
- **Forniture(codf\*, codp\*, prezzo, quantità) – FP**

o alla BD2

- **Imp(matr, nome, dip\*, stip)**
- **Dip(cod, nome, direttore\*)**

**INSERT** - Con riferimento alla BD1, inserisci un certo numero di tuple di fornitore

```
INSERT INTO F
VALUES (<f1, aldo, 34, Roma>, ... <f9, maria, Genova>)
```

Una tabella può essere popolata attraverso una query, ad esempio

```
CREATE TABLE Forn_Giovani (codf, nome, età)
```

```
INSERT INTO Forn_Giovani
SELECT codf, nome, età
FROM F
WHERE età <40
```

Si noti che For\_Giovani è una relazione di base e non è una vista; ciò significa che le sue tuple sono fisicamente memorizzate nella BD.

Nell'inserimento di tuple può verificarsi un problema di vincolo referenziale non soddisfatto; ad esempio, quando si inserisce un impiegato nella relazione Imp della BD2, deve già esistere il dipartimento di appartenenza; ciò potrebbe però non essere possibile se l'impiegato è anche direttore di quel dipartimento

```
INSERT INTO Imp
VALUES (222, giada, d1,30.000)
```

```
INSERT INTO Dip
VALUES (d1, ricerca, 222)
```

La soluzione al problema è fornita dalle transazioni.

**DELETE** - La seguente istruzione produce la cancellazione dei fornitori di Milano

```
DELETE FROM F
WHERE Città = Milano
```

La clausola WHERE non differisce da quella di una interrogazione SQL, come dimostrano i seguenti esempi:

```
DELETE FROM Imp
WHERE dip = (SELECT dip
             FROM DIP
             WHERE nome=ricerca)
```

```
DELETE FROM F
WHERE codf IN (SELECT codf
              FROM FP
              WHERE codp IN (SELECT codp
                             FROM P
                             WHERE colore=rosso))
```

La cancellazione di una o più tuple può comportare, come nel caso dell'inserimento, problemi di violazione del vincolo d'integrità referenziale

```
DELETE FROM Imp
WHERE matr=222
```

Se 222 è la matricola dell'impiegato che è direttore del dipartimento d1, allora questa cancellazione lascerebbe il dipartimento d1 con un riferimento pendente. Il DBMS bloccherebbe quindi l'operazione. Il problema andrebbe preliminarmente risolto, ad esempio, cambiando il direttore del dipartimento d1 o mettendo il suo valore a NULL

**UPDATE** - Il seguente costrutto incrementa del 20% lo stipendio degli impiegati dei dipartimenti di Roma

```
UPDATE IMP
SET stip = stip*1,2
WHERE dip IN (SELECT dip
             FROM DIP
             WHERE città = roma)
```

Anche nel caso di UPDATE potrebbero esserci problemi di integrità referenziale:

```
UPDATE F
SET codf = f2
WHERE codf=f1
```

In questo esempio, il codice di f1 viene trasformato in f2 (assumiamo che f2 non esista nel DB). Quindi, tutte le tuple di Forniture che hanno f1 come valore della chiave secondaria codf vanno a violare il vincolo di integrità referenziale. Il DBMS è pertanto costretto a bloccare l'operazione. Anche in questo caso il problema andrebbe risolto opportunamente, in particolare, attraverso l'uso di una *transazione* per poter modificare contestualmente valori della chiave secondaria e della chiave primaria – infatti, non possiamo modificare prima la chiave primaria e poi la chiave secondaria (in Forniture), né fare il viceversa, perché in entrambi i casi avremmo una violazione del vincolo.