Appunti Basi di Dati

Luca Seggiani

9 Maggio 2024

1 Manipolazione dei dati

Vediamo adesso i costrutti della parte **DML** (*Data Manipulation Language*), dell'SQL.

Inserimento

L'inserimento consiste nel, considerata una nuova tabella, inserire un nuovo record i cui valori degli attributi possono essere sia statici che ricavati. Inseriamo un nuovo paziente nel database, notiamo i dati (statici):

- Nome: Elvira
- Cognome: Passerotti
- Sesso: F
- Data di nascita: 27 Ottobre 1964
- Città: Pisa
- Reddito: 1500 Euro
- Codice Fiscale: PSSLVR65R67G702U

```
INSERT INTO Paziente
VALUES ('PSSLVR65R67G702U', 'Passerotti', 'Elvira', 'F', '
    1965-10-27', 'Pisa', 1500);
```

Dove è importante rispettare l'ordine di definizione degli attributi della tabella. Facciamo un'altro esempio: vogliamo inserire il paziente Edoardo Lepre, visitato tre giorni fa, di codice fiscale "slq6". Poniamo però di non sapere se la visita era stata mutuata o meno:

```
INSERT INTO Visita(Medico, Paziente, Data)
VALUES(010, 'slq_6', CURRENT_DATE - INTERVAL 3 DAY);
```

Questo mi permette di specificare solo alcuni valori, lasciando gli altri a NULL o al loro valore di default. Notiamo che, in generale, la sintassi:

```
INSERT INTO Tabella [(Attributo_1, Attributo_2, ...)]
VALUES (Valore_1, Valore_2, ...);
```

Ci permette di definire qualsiasi attributo in qualsiasi ordine, anche diverso da quello di definizione.

Inserimento con valori ricavati

E' possibile definire un'inserimento del tipo:

```
INSERT INTO Tabella [(Attributo_1, Attributo_2, ...,
   Attributo_N)]
Query_di_selezione;
```

Attraverso una query "Query_di_selezione" arbitraria che restituisca valori per tutti gli attributi. Poniamo di voler archiviare tutte i record delle visite effettuate prima di due anni fa in una nuova tabella, VisitaVecchia(CognomePaziente, CognomeMedico, DataVisita, Specializzazione):

```
INSERT INTO VisitaVecchia
SELECT P.Cognome AS CognomePaziente,
  M.Cognome AS CognomeMedico,
  V.Data AS DataVisita,
  M.Specializzazione
FROM Visita V INNER JOIN Medico M ON M.Matricola = V.Medico
  INNER JOIN Paziente P ON P.CodFiscale = V.Paziente
WHERE YEAR(V.Data) <= YAER(CURRENT_DATE) - 2;
```

Come vediamo, basterà definire una query che restituisce tutti i record da inserire nella tabella nell'ordine di definizione (dato nel SELECT).

Aggiornamento

Possiamo anche aggiornare record già definiti. Dovremo però usare una clausola WHERE per specificare quali record aggiornare:

```
UPDATE Tabella
SET Attributo_i = Valore_i, ecc...
WHERE Condizione
```

Mutuiamo tutte le visite del mese corrente effettuate da pazienti nati prima del 1925:

```
UPDATE Visita
SET Mutuata = 1
WHERE MONTH(Data) = MONTH(CURRENT_DATE)
  AND YEAR(Data) = YEAR(CURRENT_DATE)
  AND Paziente IN (
    SELECT CodFiscale
    FROM Paziente
    WHERE YEAR(DataNascita) < 1925
  );
```

Cancellazione

Vediamo come cancellare record dalla tabella. La sintassi è analoga a quella dell'aggiornamento:

```
DELETE FROM Tabella
WHERE Condizione
```

Poniamo di voler licenziare tutti i medici di Pisa che non hanno effettuato visite mutate il mese scorso. Scriviamo la query:

```
DELETE FROM Medico
WHERE Matricola IN (
  SELECT M_1.Matricola
  FROM Medico M_1 LEFT OUTER JOIN (
    SELECT V_2.Medico AS MedicoMutuato
    FROM Visita V_2 INNER JOIN Medico M_2 ON M_2.Matricola =
   V_2.Medico
    WHERE M_2.Citta = 'Pisa'
      AND V_2.Mutuata = TRUE
      AND V_2.Data > CURRENT_DATE - INTERVAL 1 MONTH
    ) AS D
    ON M_1.Matricola = D.MedicoMutuato
    WHERE D.MedicoMutuato IS NULL
);
```

Il DBMS riceve la query e restituisce un errore. Questo perchè, nel momento in cui dichiariamo l'operazione di cancellazione con DELETE FROM Medico; stiamo effettivamente bloccando la tabella. Non possiamo quindi definire una query che legga tale tabella: il DBMS ce lo impedirà in quanto potremmo generare loop infiniti. Possiamo risolvere questa situazione in più modi.

- Trasformare la query in una derived table:

```
DELETE FROM Medico
WHERE Matricola IN (
  SELECT * FROM (
    SELECT M_1.Matricola
    FROM Medico M_1 LEFT OUTER JOIN (
     SELECT V_2.Medico AS MedicoMutuato
     FROM Visita V_2 INNER JOIN Medico M_2 ON M_2.
   Matricola = V_2.Medico
      WHERE M_2.Citta = 'Pisa'
        AND V_2.Mutuata = TRUE
        AND V_2.Data > CURRENT_DATE - INTERVAL 1 MONTH
      ) AS D
      ON M_1.Matricola = D.MedicoMutuato
      WHERE D.MedicoMutuato IS NULL
  ) AS D_2
);
```

Questo assicura che il valore della derived table venga calcolato prima dell'operazione di cancellazione, eludendo quindi il blocco imposto dal DBMS. La stessa cosa potrà essere fatta, anziché con una derived table, usando una CTE.

- Un'altro approccio, meno intuitivo, è quello di usare un join:

```
DELETE M_1.*
FROM Medico M_1 LEFT OUTER JOIN (
  SELECT V_2.Medico AS MedicoMutuato
  FROM Visita V_2 INNER JOIN Medico M_2 ON M_2.Matricola
   = V_2.Medico
  WHERE M_2.Citta = 'Pisa'
    AND V_2.Mutuata = TRUE
    AND V_2.Data > CURRENT_DATE - INTERVAL 1 MONTH
 ) AS D
ON M_1.Matricola = D.MedicoMutuato
WHERE D.MedicoMutuato IS NULL;
```