

Linguaggio SQL: comandi del DML

1 Operazioni di aggiornamento

SQL fornisce tre comandi per la manipolazione del contenuto di una relazione:

- **INSERT**: per l'inserimento di tuple;
- **UPDATE**: per la modifica di tuple;
- **DELETE**: per la cancellazione di tuple.

Le tuple oggetto di tali comandi sono determinate tramite condizioni che possono coinvolgere anche altre relazioni. Ogni istruzione, però, può aggiornare il contenuto di una sola tabella.

Siccome le operazioni di aggiornamento modificano lo stato della base di dati, il DBMS deve verificare che siano rispettati i vincoli di integrità.

2 Inserimento

È possibile inserire:

- una singola tupla, assegnando un valore costante (cioè specificato direttamente nel comando) a ciascun attributo;
- più tuple, lette da altre tabelle mediante un'istruzione **SELECT**.

La sintassi è

```
INSERT INTO S [(C1, C2, ..., Cn)] {VALUES (v1, v2, ..., vn) | sq};
```

dove:

- *S* è il nome della relazione su cui si esegue l'inserimento.
- *C*₁, *C*₂, ..., *C*_{*n*} è la lista delle colonne a cui si assegnano valori:
 - ogni colonna non elencata riceve il proprio valore di default, se esso è stato specificato nella creazione di *S*, oppure il valore nullo, se ciò è ammesso (cioè se la colonna non ha un vincolo **NOT NULL**), altrimenti il DBMS rifiuta l'inserimento e segnala un errore;

- se non si specifica esplicitamente questa lista, vengono considerate tutte le colonne di S , nell'ordine dato dal comando `CREATE TABLE`.
- La clausola `VALUES` permette di specificare esplicitamente i valori v_1, v_2, \dots, v_n da assegnare alla singola tupla che viene inserita:
 - i valori sono assegnati in base a una corrispondenza posizionale, cioè v_i (con $i = 1, \dots, n$) viene assegnato alla colonna C_i ;
 - ogni valore v_i può essere:
 - * un valore appartenente al dominio di C_i ;
 - * il valore `NULL`, se ammesso per la colonna C_i ;
 - * la parola chiave `DEFAULT`, per assegnare a C_i il suo valore di default, se esiste.
- Al posto di `VALUES`, si può specificare una sotto-interrogazione, sq , i cui risultati (un numero qualsiasi di tuple) vengono inseriti nella relazione S .

sq è un'interrogazione arbitraria, ma non può contenere la clausola `ORDER BY`, dato che essa ha senso solo per la visualizzazione dei dati, e non permette invece di controllare l'ordine delle tuple nella tabella.

Come per `VALUES`, si ha una corrispondenza posizionale: il dominio dell' i -esima colonna/espressione specificata nella clausola `SELECT` di sq deve essere compatibile con quello di C_i .

2.1 Vincoli d'integrità

I vincoli d'integrità che vengono controllati quando si effettua un inserimento sono quelli:

- sui singoli attributi (ad esempio, `NOT NULL`);
- di chiave (`PRIMARY`, `UNIQUE`);
- di integrità referenziale (se le tuple inserite contengono chiavi esterne, i loro valori devono corrispondere a delle tuple esistenti nelle tabelle riferite).

2.2 Esempi

Dato lo schema

Film(titolo, regista, anno, genere, valutaz)

si vuole inserire un nuovo film, tale che:

- il titolo “La tigre e la neve”;
- il regista Roberto Benigni;
- l’anno di produzione è 2005;
- il genere è commedia.

```
INSERT INTO Film (titolo, regista, anno, genere)
VALUES ('la tigre e la neve', 'roberto benigni', 2005, 'commedia');
```

Siccome non è specificata una valutazione, questa assume valore nullo.

Dato lo schema

Video(colloc, titolo^{Film}, regista^{Film}, tipo)

il comando seguente inserisce un nuovo video, assegnando valori a tutte le colonne:

```
INSERT INTO Video
VALUES (1130, 'le iene', 'quentin tarantino', 'd');
```

Il comando seguente inserisce nella tabella `ProdottiMilanesi` il codice e la descrizione di tutti i prodotti provenienti da Milano:

```
INSERT INTO ProdottiMilanesi
SELECT codice, descrizione
FROM Prodotti
WHERE LuogoProd = 'Milano';
```

3 Cancellazione

La sintassi del comando di cancellazione è

```
DELETE FROM S [<alias>] [WHERE F];
```

dove:

- S è il nome della relazione su cui si esegue la cancellazione.
- F è la clausola di qualificazione (condizione) che specifica quali tuple cancellare; se è assente, vengono cancellate tutte.
- Il nome della relazione può avere un alias, che permette l’uso di sotto-interrogazioni correlate all’interno di F .

3.1 Vincoli d'integrità

Quando si effettua una cancellazione, il DBMS controlla i vincoli di integrità referenziale. Se viene cancellata una tupla riferita, il comportamento dipende dalla specifica **ON DELETE** (**NO ACTION**, **CASCADE**, **SET NULL**, **SET DEFAULT**) della chiave esterna nella tabella referente. In particolare, se questa è **NO ACTION**, il DBMS rifiuta l'operazione: in tal caso, l'unico modo di cancellare una tupla riferita è eliminare prima le tuple referenti.

3.2 Esempi

Cancellare il film “La tigre e la neve”, inserito in uno degli esempi precedenti:

```
DELETE FROM Film
WHERE titolo = 'la tigre e la neve'
AND regista = 'roberto benigni';
```

Cancellare i clienti che non hanno effettuato noleggi nell'ultimo anno:

```
DELETE FROM Cliente
WHERE codCli NOT IN (
  SELECT codCli
  FROM Noleggio
  WHERE dataNo1 > CURRENT_DATE - '1' YEAR
);
```

4 Modifica

Il comando di modifica ha la sintassi

```
UPDATE S [<alias>]
SET C1 = {e1 | NULL}, ..., Cn = {en | NULL}
[WHERE F];
```

dove:

- S è il nome della relazione su cui si esegue la modifica.
- S può avere un alias, che permette l'uso di sotto-interrogazioni correlate.
- $C_i = \{e_i \mid \text{NULL}\}$ (con $i = 1, \dots, n$) è un'espressione di assegnamento, la quale specifica che alla colonna C_i deve essere assegnato:
 - il valore dell'espressione e_i , che può essere una costante, un'espressione aritmetica / di stringa / ecc., o una sotto-interrogazione;

- oppure il valore nullo.
- F è la clausola di qualificazione che specifica le tuple da modificare; se è omessa, vengono modificate tutte le tuple presenti in S .

Nel comando `UPDATE`, le sotto-interrogazioni possono essere quindi usate:

- nella clausola di qualificazione, per determinare le tuple da modificare;
- nella clausola di assegnamento, per determinare i nuovi valori da assegnare alle tuple.

4.1 Esempi

Raddoppiare la valutazione di tutti i film (supponendo, ad esempio, che si sia deciso di cambiare la scala da 0–5 a 0–10):

```
UPDATE Film
SET valutaz = valutaz * 2;
```

Il cliente 6635 restituisce tutti i video che ha attualmente in noleggio:

```
UPDATE Noleggio
SET dataRest = CURRENT_DATE
WHERE codCli = 6635
AND dataRest IS NULL;
```

Diminuire del 10 % il costo dei noleggi dei film usciti prima del 2011 che sono stati noleggiati da meno di 10 clienti:

```
UPDATE Film
SET costo = costo * 0.9
WHERE anno < 2011
AND (titolo, regista) IN (
  SELECT titolo, regista
  FROM Video
  JOIN Noleggio NATURAL JOIN Video
  GROUP BY titolo, regista
  HAVING COUNT(DISTINCT codCli) < 10
);
```

Dato lo schema

```
Impiegato(Nome, CF, Indirizzo, Stipendio, RespImpiegato, DipDipartimento)
Dipartimento(Nome, IdD, ManagerImpiegato, ManagerData_In)
```

si vuole assegnare ai manager di ogni dipartimento uno stipendio pari al 110 % della media degli stipendi del dipartimento che dirigono. Si ipotizza che ogni manager appartenga allo stesso dipartimento che dirige.

```
UPDATE Impiegato I
SET Stipendio = (
    SELECT 1.1 * AVG(Stipendio)
    FROM Impiegato
    WHERE Dip = I.Dip
)
WHERE CF IN (
    SELECT Manager
    FROM Dipartimento
);
```