

Universita' di Milano Bicocca
Corso di Basi di dati 1 in eLearning
C. Batini
7. SQL DML
7.12 Operazioni di aggiornamento
e vincoli di integrita'

Fino ad ora

- Abbiamo potuto interrogare la base di dati, ma non abbiamo potuto modificarla, aggiornarla.
- Ma il mondo cambia, e la base di dati deve seguire questi cambiamenti.

Operazioni di aggiornamento

- Sono operazioni di
 - inserimento: insert
 - eliminazione: delete
 - modifica: update

Sono eseguite

a. su una o più ennuple di una relazione R_1

b. sulla base di una condizione che può coinvolgere anche altre relazioni R_2, \dots, R_n

Inserimento

```
INSERT INTO Tabella [ ( Attributi ) ]  
VALUES( Valori )
```

che permette di inserire negli Attributi della Tabella singoli valori.

oppure

```
INSERT INTO Tabella [ ( Attributi ) ]  
SELECT ...
```

che permette di inserire negli Attributi della Tabella insiemi di ennuple estratte dalla base dati

Inserimento, primo insieme di regole

1. Se qualche attributo non e' specificato, e' inserito valore **nullo**.
2. Se e' violato un vincolo di **not null**, l'insert e' rifiutato.
3. l'**ordinamento** degli attributi (se presente) e dei valori è significativo

Inserimento, altre regole

4. le due liste (Attributi, Valori) debbono avere lo stesso numero di elementi.
5. se la lista di attributi è omessa, si fa riferimento a tutti gli attributi della relazione, secondo l'ordine con cui sono definiti
6. se la lista di attributi non contiene tutti gli attributi della relazione, per gli altri viene inserito un valore di default, o, in assenza di questo, il valore nullo (che deve essere permesso).

Esempi

- Consideriamo la relazione
- Persone (Nome, Eta, Reddito)

Esempi di INSERT INTO

Relazione Persone (Nome, Eta', Reddito)

```
INSERT INTO Persone (Nome, Eta, Reddito)  
VALUES ('Pino', 25, 52)
```

```
INSERT INTO Persone VALUES ('Mario',  
25, 52)
```

```
INSERT INTO Persone (Nome, Reddito)  
VALUES ('Lino', 55)
```


Esempio con Select

```
INSERT INTO Persone ( Nome )  
SELECT Padre  
FROM Paternita  
WHERE Padre NOT IN (SELECT Nome  
                      FROM Persone)
```

Domanda 7.12.1 - Esprimere a parole quale aggiornamento esegue questa interrogazione

Esempio con Select

```
INSERT INTO Persone ( Nome )  
SELECT Padre  
FROM Paternita  
WHERE Padre NOT IN (SELECT Nome  
                      FROM Persone)
```

Risposta

Inserisce in Persone i padri che non vi compaiono già'

Eliminazione di ennuple

DELETE FROM Tabella
[WHERE Condizione]

elimina le ennuple che soddisfano la
condizione

Eliminazione, regole

- 1. la eliminazione può causare eliminazioni in altre relazioni, se ci sono vincoli di integrità referenziale, ed e' adottata una politica di reazione **cascade**
- 2. se la **where** viene omessa, si intende **where true**, quindi vengono eliminate tutte le **ennuple**

Esempi di DELETE FROM - Domanda 7.12.2

Cosa calcolano la seconda e la terza
(la prima e' ovvia)?

Risposta

```
DELETE FROM Persone  
WHERE Eta < 35
```

Cancella dai padri coloro
che non sono nonni

```
DELETE FROM Paternita  
WHERE Figlio NOT in (SELECT Padre
```

F
Cancella tutti i padri

```
DELETE FROM Paternita
```

Differenza tra DELETE (DML) e DROP (DDL)

La DELETE from DIPARTIMENTO

elimina tutte le n-ple della istanza di

DIPARTIMENTO, ed eventuali altre per la politica cascade, ma lascia immutato lo schema

La DROP Table DIPARTIMENTO cascade

elimina dallo schema della base di dati la relazione

DIPARTIMENTO e tutti gli schemi di relazione che fanno riferimento ad esso.

La DROP Table DIPARTIMENTO restrict fallisce se vi sono tuple nella istanza di relazione.

Modifica di ennuple: UPDATE

UPDATE NomeTabella

SET Attributo = <Espressione |
SELECT ... |
NULL |
DEFAULT>

Significa:
oppure

[WHERE Condizione]

Modifica le n-ple che rispettano la (eventuale) condizione. Se non c'è la condizione, le modifica tutte

Il nuovo valore puo' essere

1. **Espressione**, il risultato della valutazione di una espressione sugli attributi della tabella
2. **SELECT**, il risultato di una interrogazione
3. **NULL**, il valore nullo
4. **DEFAULT**, il valore di default

Esempi di update

```
UPDATE Persone  
  SET Reddito = 45  
  WHERE Nome = 'Piero'
```

```
UPDATE Persone  
  SET Reddito = Reddito * 1.1  
  WHERE Eta < 30
```

Attenzione!!!

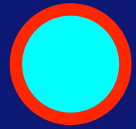
L'esecuzione di:

```
UPDATE Persone
  SET Stipendio = Stipendio * 1.1
  WHERE Stipendio <= 30

UPDATE Persone
  SET Stipendio = Stipendio * 1.15
  WHERE Stipendio > 30
```

provoca, ad es., per un impiegato con Stipendio = 30, due aumenti, e non uno. Cio' per la natura "set oriented" di SQL.

Il problema si risolve scrivendo un programma in un linguaggio ad alto livello (if stipendio <= then SET .. else SET ...)



Vincoli di integrita'

- Vincoli intrarelazionale di attributo o tabella: CHECK
- Vincoli di schema, cioe' su piu' relazioni: CREATE ASSERTION

● Vincoli di integrità generici: CHECK

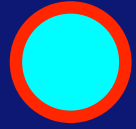
Specifica di vincoli di ennumera (e anche vincoli più complessi) di un attributo o di una tabella.

Sintassi

CHECK (Condizione)

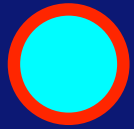
dove Condizione puo' essere una condizione che puo' comparire in una clausola WHERE

Significato: La Condizione che compare nella CHECK deve essere sempre verificata.



Proprieta' della CHECK

- La CHECK permette anche di esprimere tutti i vincoli predefiniti, ma in questi casi, rispetto ad essi, e' meno naturale e comprensibile (vedi libro)



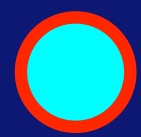
Check, esempi

```
create table Impiegato  
(Matricola character(6) primary key,  
Cognome character(30),  
Nome character(30),  
Sesso character(1) not null,  
Stipendio integer,  
Superiore character(6),
```

Verifica che lo stipendio dell'impiegato sia minore o uguale allo stipendio dell'impiegato che ne e' superiore.

Vincolo non esprimibile senza CHECK

```
check (Stipendio <= (select Stipendio  
from Impiegato J  
where Superiore = J.Matricola))
```



Vincoli di integrità generici: asserzioni

- Specificano vincoli a livello di schema

Sintassi:

```
CREATE ASSERTION NomeAsserzione  
CHECK ( Condizione )
```

```
CREATE ASSERTION AlmenoUnImpiegato  
CHECK (1 <= (select count(*)  
from Impiegato ))
```

Verifica se nella relazione Impiegato vi è almeno una ennupla, cioè se c'è almeno un impiegato

Politiche di verifica dei vincoli di integrita'

- Vincolo con politica immediata - sono verificati immediatamente dopo ogni modifica della base di dati
- Vincolo con politica differita - sono verificati dopo l'esecuzione di una serie di operazioni che costituiscono una transazione (vedi corso di complementi di basi di dati)

Esito della verifica

Se un vincolo immediato non è soddisfatto il sistema "disfa" l'operazione di modifica (operazione chiamata anche **rollback**), cioè riporta la base di dati a uno stato in cui l'operazione non è stata eseguita

Per i vincoli differiti, vedi corso di Basi di dati 2.

Esercizio 7.12.1

Esegui l'esercizio 4.5 del testo di riferimento

Esercizio 7.12.2

- Esegui l'esercizio 4.6 del testo di riferimento

Esercizio 7.12.3

- Esegui l'esercizio 4.9 del testo di riferimento.

Esercizio 7.12.4

- Esegui l'esercizio 4.11 del testo di riferimento.

Concetti introdotti - 1

- Operatore di aggiornamento
- Operatore Insert
- Operatore Delete
- Operatore Update
- Vincoli di integrità'
 - Vincoli intrarelazionale di attributo o tabella: CHECK
 - Vincoli di schema: CREATE ASSERTION