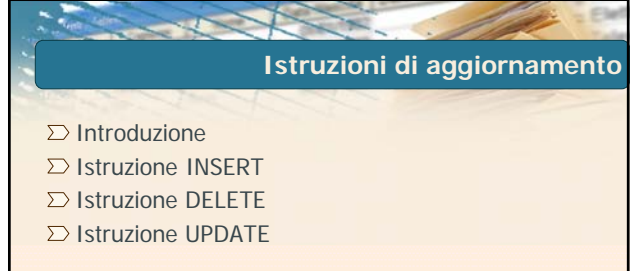




Linguaggio SQL: fondamenti

Istruzioni di aggiornamento

DBG  
M




Istruzioni di aggiornamento

- ⊃ Introduzione
- ⊃ Istruzione INSERT
- ⊃ Istruzione DELETE
- ⊃ Istruzione UPDATE

DBG  
M

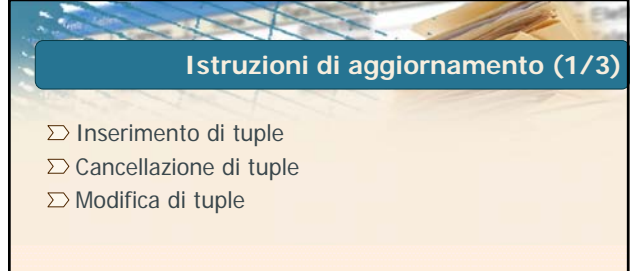
2



Istruzioni di aggiornamento

Introduzione

DBG  
M

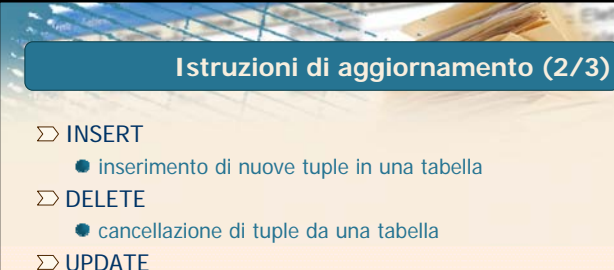


Istruzioni di aggiornamento (1/3)

- ⊃ Inserimento di tuple
- ⊃ Cancellazione di tuple
- ⊃ Modifica di tuple

DBG  
M

4

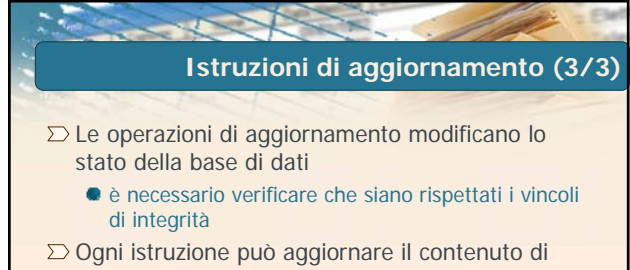


Istruzioni di aggiornamento (2/3)

- ⊃ INSERT
  - inserimento di nuove tuple in una tabella
- ⊃ DELETE
  - cancellazione di tuple da una tabella
- ⊃ UPDATE
  - modifica del contenuto di tuple in una tabella

DBG  
M

5

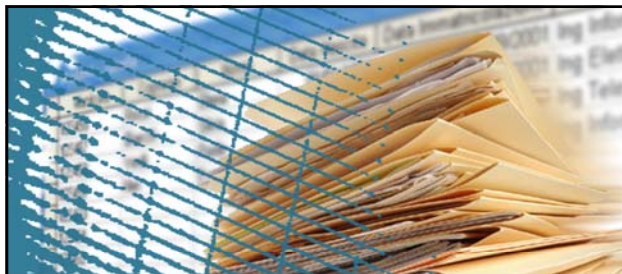


Istruzioni di aggiornamento (3/3)

- ⊃ Le operazioni di aggiornamento modificano lo stato della base di dati
  - è necessario verificare che siano rispettati i vincoli di integrità
- ⊃ Ogni istruzione può aggiornare il contenuto di una sola tabella


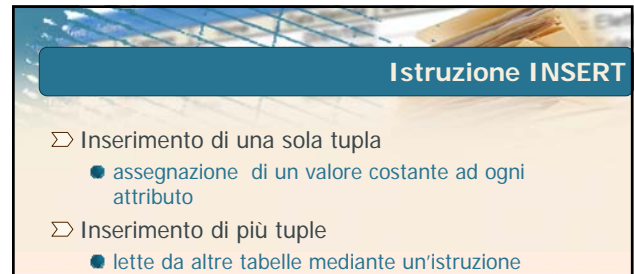
DBG  
M

6




## Istruzioni di aggiornamento

### Istruzione INSERT





## Istruzione INSERT

- ⊃ Inserimento di una sola tupla
  - assegnazione di un valore costante ad ogni attributo
- ⊃ Inserimento di più tuple
  - lette da altre tabelle mediante un'istruzione SELECT




8

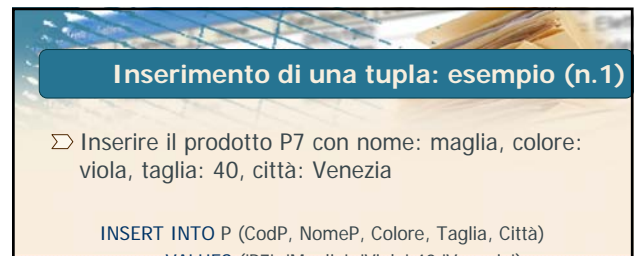


## Inserimento di una tupla

```
INSERT INTO NomeTabella
      [(ElencoColonne)]
VALUES (ElencoCostanti);
```



9




## Inserimento di una tupla: esempio (n.1)

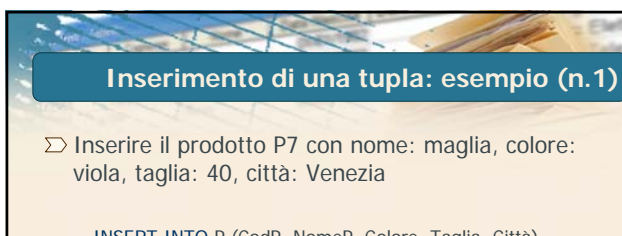
- ⊃ Inserire il prodotto P7 con nome: maglia, colore: viola, taglia: 40, città: Venezia

```
INSERT INTO P (CodP, NomeP, Colore, Taglia, Città)
VALUES ('P7', 'Maglia', 'Viola', 40, 'Venezia');
```

- ⊃ È inserita nella tabella P una nuova tupla con i valori specificati



10




## Inserimento di una tupla: esempio (n.1)

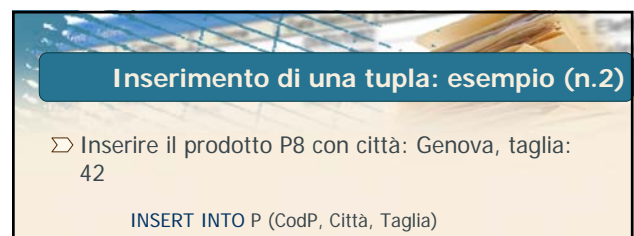
- ⊃ Inserire il prodotto P7 con nome: maglia, colore: viola, taglia: 40, città: Venezia

```
INSERT INTO P (CodP, NomeP, Colore, Taglia, Città)
VALUES ('P7', 'Maglia', 'Viola', 40, 'Venezia');
```

- ⊃ Omettere la lista dei campi equivale a specificare tutti i campi secondo l'ordine di creazione delle colonne nella tabella
  - Se la tabella cambia schema, l'istruzione non è più applicabile



11




## Inserimento di una tupla: esempio (n.2)

- ⊃ Inserire il prodotto P8 con città: Genova, taglia: 42

```
INSERT INTO P (CodP, Città, Taglia)
VALUES ('P8', 'Genova', 42);
```

- ⊃ È inserita nella tabella P una nuova tupla con i valori specificati
  - a NomeP e Colore è assegnato il valore NULL
- ⊃ Per tutti gli attributi il cui valore non è specificato, il dominio dell'attributo deve consentire il valore NULL



12

### Inserimento: integrità referenziale

- ⇒ Inserire una nuova fornitura relativa al fornitore F20, prodotto P20 e quantità 1000

```
INSERT INTO FP (CodF, CodP, Qta)
VALUES ('F20', 'P20', 1000);
```

- ⇒ Vincolo di integrità referenziale
  - è necessario che P20 e F20 siano già presenti rispettivamente in P e F
  - se il vincolo non è soddisfatto, l'inserimento non deve essere eseguito

### Inserimento di più record

```
INSERT INTO NomeTabella
[(ElencoColonne)]
Interrogazione;
```

- ⇒ Sono inserite in *NomeTabella* tutte le tuple selezionate dall'interrogazione *Interrogazione*
- ⇒ *Interrogazione* è un'istruzione SELECT arbitraria
  - non può contenere la clausola ORDER BY

### Inserimento di più record: esempio

FORNITURE-TOTALI (CodP, TotQta)

- ⇒ Per ogni prodotto, inserire nella tabella FORNITURE-TOTALI la quantità totale fornita
  - dati aggregati estratti dalla tabella FP

```
SELECT CodP, SUM(Qta)
FROM FP
GROUP BY CodP
```

### Inserimento di più record: esempio

FORNITURE-TOTALI (CodP, TotQta)

- ⇒ Per ogni prodotto, inserire nella tabella FORNITURE-TOTALI la quantità totale fornita

```
INSERT INTO FORNITURE-TOTALI (CodP, TotQta)
(SELECT CodP, SUM(Qta)
FROM FP
GROUP BY CodP);
```

### Istruzioni di aggiornamento

#### Istruzione DELETE

### Istruzione DELETE

```
DELETE FROM NomeTabella
[WHERE predicato];
```

- ⇒ Cancellazione dalla tabella *NomeTabella* di tutte le tuple che soddisfano il predicato
- ⇒ Occorre sempre verificare che la cancellazione non causi la violazione di vincoli di integrità referenziale

## Istruzione DELETE: esempio (n.1)

- ⇒ Cancellare tutte le forniture

```
DELETE FROM FP;
```

- ⇒ Senza clausola WHERE tutte le tuple soddisfano il predicato di selezione
  - elimina il contenuto della tabella FP
  - la tabella *non* è eliminata

## Istruzione DELETE: esempio (n.2)

- ⇒ Cancellare la tupla corrispondente al fornitore con codice F1

```
DELETE FROM F
WHERE CodF='F1';
```

- ⇒ Se in FP esistono forniture che fanno riferimento ai fornitori cancellati, la base di dati perde la propria integrità
  - avviene una violazione del vincolo di integrità referenziale tra FP e F
  - occorre propagare la cancellazione

## Istruzione DELETE: esempio (n.2)

- ⇒ Cancellare la tupla corrispondente al fornitore con codice F1

```
DELETE FROM F
WHERE CodF='F1';
```

```
DELETE FROM FP
WHERE CodF='F1';
```

- ⇒ Per mantenere la coerenza è necessario completare le operazioni su entrambe le tabelle

## Istruzione DELETE: esempio complesso

- ⇒ Cancellare i fornitori di Milano

```
DELETE FROM F
WHERE Sede='Milano';
```

- ⇒ Se in FP esistono forniture che fanno riferimento ai fornitori cancellati, è violato il vincolo di integrità referenziale tra FP e F
  - occorre cancellare anche tali forniture in FP

## Istruzione DELETE: esempio complesso

- ⇒ Cancellare i fornitori di Milano

```
DELETE FROM F
WHERE Sede='Milano';
```

```
DELETE FROM FP
WHERE CodF IN (SELECT CodF
FROM F
WHERE Sede='Milano');
```

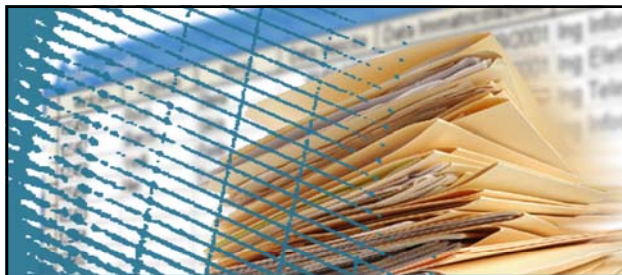
- ⇒ In quale ordine devono essere eseguite le due operazioni di cancellazione?

## Istruzione DELETE: esempio complesso

- ⇒ Ordine corretto di esecuzione

```
DELETE FROM FP
WHERE CodF IN (SELECT CodF
FROM F
WHERE Sede='Milano');
```

```
DELETE FROM F
WHERE Sede='Milano';
```



Istruzioni di aggiornamento

Istruzione UPDATE

DBG

Istruzione UPDATE

```
UPDATE NomeTabella
SET colonna = espressione
{, colonna=espressione}
[WHERE predicato];
```

⊃ Tutti i record della tabella *NomeTabella* che soddisfano il predicato sono modificati in base alle assegnazioni *colonna=espressione* nella clausola SET

DBG

26

Aggiornamento di una tupla

⊃ Aggiornare le caratteristiche del prodotto P1: assegnare giallo al colore, incrementare la taglia di 2 e assegnare NULL a città

```
UPDATE P
SET Colore = 'Giallo',
    Taglia=Taglia+2,
    Città = NULL
WHERE CodP='P1';
```

⊃ È aggiornata la tupla individuata dal codice P1

DBG

27

Aggiornamento multiplo

⊃ Aggiornare il numero dei soci al doppio del valore per tutti i fornitori di Milano

```
UPDATE F
SET NSoci=2*NSoci
WHERE Città='Milano';
```

⊃ Sono aggiornate tutte le tuple individuate dal predicato nella clausola WHERE

DBG

28

Aggiornamento con sottointerrogazione

⊃ Aggiornare a 10 la quantità fornita per tutti i fornitori di Milano

```
UPDATE FP
SET Qta = 10
WHERE CodF IN (SELECT CodF
                FROM F
                WHERE Sede='Milano');
```

DBG

29

Aggiornamento di più tabelle

⊃ Modificare con il valore F9 il codice del fornitore F2

```
UPDATE F
SET CodF='F9'
WHERE CodF='F2';
```

⊃ Se in FP esistono forniture che fanno riferimento ai codici dei fornitori aggiornati, è violato il vincolo di integrità referenziale

- occorre aggiornare anche tali forniture in FP

DBG

30

### Aggiornamento di più tabelle

- ⇒ Modificare con il valore F9 il codice del fornitore F2

```
UPDATE F
SET CodF='F9'
WHERE CodF='F2';
```

```
UPDATE FP
SET CodF='F9'
WHERE CodF='F2';
```

- ⇒ Per mantenere la coerenza è necessario completare l'aggiornamento di entrambe le tabelle