



# LINGUAGGIO SQL

**Modifiche all'istanza di una relazione**

1



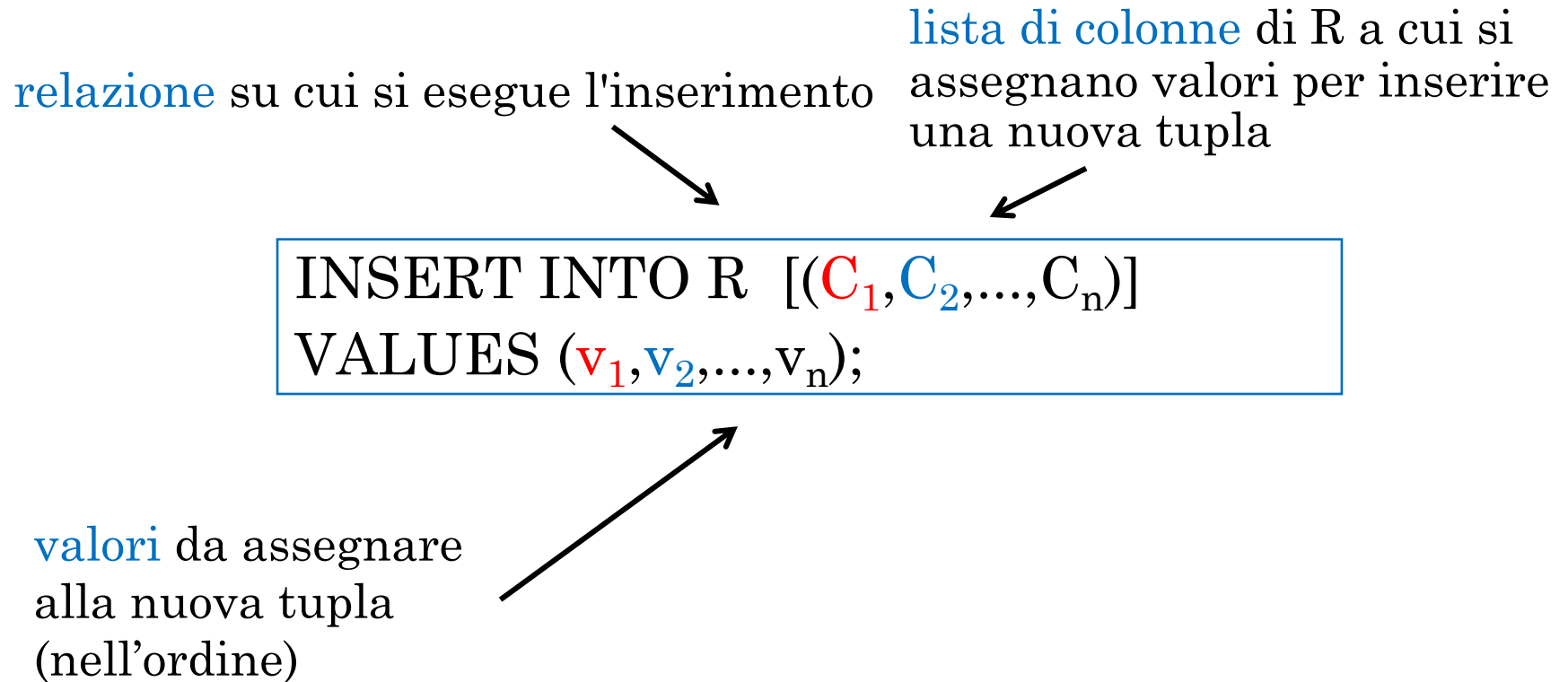
# MODIFICA DI TUPLE

- **INSERT** - inserimento
- **UPDATE** - modifica
- **DELETE** – cancellazione

Si applicano a una o più tuple selezionate con clausole analoghe a quelle usate per la ricerca



# INSERIMENTO CON VALORI ESPLICITI



# ESEMPIO

Inserire un nuovo film, con

- titolo → ``La tigre e la neve''
- regista → Roberto Benigni
- anno di produzione → 2005
- genere → commedia
- valutaz → 3

```
INSERT INTO Film(titolo, regista, anno, genere, valutaz)
VALUES ('la tigre e la neve', 'roberto benigni', 2005, 'commedia', 3);
```

oppure

```
INSERT INTO Film(anno, titolo, genere, regista, valutaz)
VALUES (2005, 'la tigre e la neve', 'commedia', 'roberto benigni', 3);
```



# INSERIMENTO CON VALORI ESPLICITI: ECCEZIONE 1

- Se la lista delle colonne manca, equivale alla lista di tutte le colonne di R nell'ordine dato dal comando CREATE TABLE

Inserire un nuovo film, con

- titolo → "La tigre e la neve"
- regista → Roberto Benigni
- anno di produzione → 2005
- genere → commedia
- valutaz → 3

INSERT INTO Film

VALUES ('la tigre e la neve', 'roberto benigni', 2005, 'commedia', 3);

# INSERIMENTO CON VALORI ESPLICITI:

## ECCEZIONE 2

- Se una colonna di R non compare nella lista, l'attributo corrispondente viene inizializzato con il valore di default se specificato nel comando di creazione di R, con NULL altrimenti
- Se non c'è default e una colonna ha vincolo di obbligatorietà ma non abbiamo messo la colonna si genera errore

## ESEMPIO: ECCEZIONE 2

Inserire un nuovo film, con

- titolo → ``La tigre e la neve``
- regista → Roberto Benigni
- anno di produzione → 2005
- genere → commedia

Manca valutaz

valutazione non è specificata  
quindi assume il valore di  
default, se presente, oppure  
il valore nullo

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

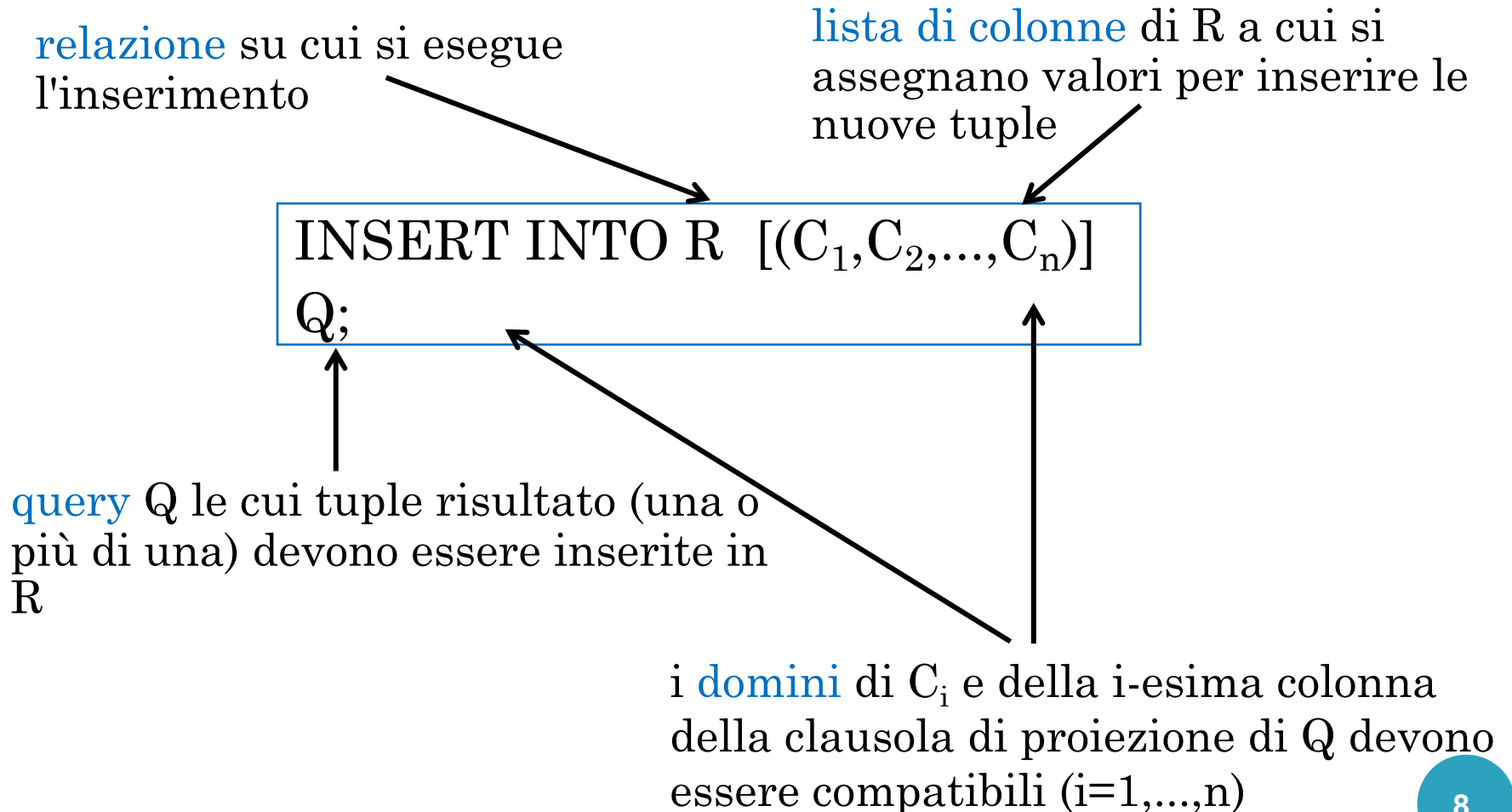
Equivalentemente

```
INSERT INTO Film
VALUES ('la tigre e la neve', 'roberto benigni', 2005, 'commedia', NULL);
```

Se nello schema valutaz è  
dichiarato con vincolo NOT  
NULL e non c'è default →  
errore

Non rispetcia perfettamente la specifica  
perché assegna esplicitamente  
NULL alla valutazione

# INSERIMENTO MEDIANTE QUERY





# ESEMPIO

Inserire in data odierna noleggi relativi al **cliente 6635** e a tutti i video contenenti film di Gabriele Salvatores che non aveva ancora noleggiato

```
INSERT INTO Noleggio(colloc, codCli)
  SELECT colloc, 6635 FROM Video
  WHERE regista = 'gabriele salvatores' AND (colloc,6635)
    NOT IN (SELECT colloc, codCli FROM Noleggio);
```

Se ci sono più video per uno stesso film?  
Li noleggia tutti!

Le tuple inserite sono:

- (1124, CURRENT\_DATE, 6635, NULL) e
- (1126, CURRENT\_DATE, 6635, NULL)

Default stabiliti al momento  
della creazione della tabella

Se alcuni video  
sono già  
noleggiati?

Li inserisce lo  
stesso!

# INSERIMENTO MEDIANTE QUERY: ECCEZIONI

- Valgono le eccezioni già discusse per l'inserimento con valori espliciti:
  1. **se la lista delle colonne manca**, equivale alla lista di tutte le colonne di R nell'ordine dato dal comando `CREATE TABLE`
  2. **se una colonna di R non compare nella lista**, l'attributo corrispondente viene inizializzato con il valore di default se specificato nel comando di creazione di R, con null altrimenti
  3. **se non c'è default e una colonna ha vincolo di obbligatorietà** ma non abbiamo messo la colonna si genera errore

# CANCELLAZIONE

**relazione** su cui si esegue la cancellazione

da usare eventualmente per riferire a tuple di R in una qualche sotto-interrogazione di F

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

**clausola di qualificazione** che specifica le tuple da cancellare  
si possono usare sottointerrogazioni

se manca la clausola di qualificazione, vengono cancellate tutte le tuple di R

## ESEMPI

- Cancellare il film “La tigre e la neve”

```
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 dataNol > (CURRENT_DATE - 1 YEAR));
```

# MODIFICA

**relazione** su cui si esegue la cancellazione

da usare eventualmente per riferire a tuple di R in una qualche sotto-interrogazione di F

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

alla colonna C<sub>i</sub> deve essere assegnato il valore dell'espressione e<sub>i</sub> o null

e<sub>i</sub> può contenere sottointerrogazioni

se manca la clausola di qualificazione, vengono modificate tutte le tuple di R

**clausola di qualificazione** che specifica le tuple da modificare si possono usare sottointerrogazioni

## ESEMPI

- Raddoppiare le valutazioni dei film (ad esempio in corrispondenza ad un cambio di 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 e segnala che il noleggio è iniziato ieri (e non come erroneamente diversamente registrato)

```
UPDATE Noleggio  
SET dataRest = CURRENT_DATE,  
    dataNol = (CURRENT_DATE - 1 DAY)  
WHERE codCli = 6635 AND dataRest IS NULL;
```

## ESEMPIO

Aumentare del 10% la valutazione dei film horror usciti dopo il 1999 che sono stati noleggiati da almeno due clienti

```
UPDATE Film
SET valutaz = valutaz * 1.1
WHERE genere = 'horror' AND anno > 1999
AND (titolo, regista) IN
  (SELECT titolo, regista
   FROM Noleggio NATURAL JOIN Video
   GROUP BY titolo, regista
   HAVING COUNT(DISTINCT codCli) >= 2);
```

## ESEMPIO

Assegnare ad ogni film **che è stato noleggiato almeno una volta nell'ultimo mese**, una valutazione pari al **110%** della valutazione media dei film dello stesso anno e genere

```
UPDATE Film X
SET valutaz = (
    SELECT 1.1 * AVG(valutaz)
    FROM Film
    WHERE anno = X.anno AND genere = X.genere)
WHERE EXISTS (
    SELECT *
    FROM Noleggio NATURAL JOIN Video
    WHERE dataNoI > CURRENT_DATE - 1 MONTH
    AND titolo = X.titolo AND regista = X.regista
)
```



# MODIFICA DI PIÙ TUPLE

## ORDINE DI VALUTAZIONE

- Le modifiche in SQL sono eseguite in modo set-oriented
  - la clausola WHERE e il valore dell'espressione SET vengono valutati (un'unica volta)
  - gli assegnamenti vengono effettuati su tutte le tuple **contemporaneamente**

# ESEMPIO

la valutazione media viene calcolata sulle tuple inizialmente contenute in Film

La stessa valutazione viene assegnata a tutti i film dello stesso anno e genere, noleggiati almeno una volta nell'ultimo mese

```
UPDATE Film X
SET valutaz = (
    SELECT 1.1 * AVG(valutaz)
    FROM Film
    WHERE anno = X.anno AND genere = X.genere)
WHERE EXISTS (
    SELECT *
    FROM Noleggio NATURAL JOIN Video
    WHERE dataNoI > CURRENT_DATE - 1 MONTH
    AND titolo = X.titolo AND regista = X.regista
)
```

- si evitano feedback e diventa deterministico il risultato