



# Linguaggio SQL: costrutti avanzati

Gestione delle viste

# Gestione delle viste

- Introduzione
- Creazione e gestione delle viste in SQL
- Aggiornabilità delle viste
- Gestione della privatezza

- La vista è una *tabella "virtuale"*
  - il contenuto (tuple) è definito mediante un'interrogazione SQL sulla base di dati
    - il contenuto della vista dipende dal contenuto delle altre tabelle presenti nella base di dati
  - il contenuto *non* è memorizzato fisicamente nella basi di dati
    - è ricalcolato tutte le volte che si usa la vista eseguendo l'interrogazione che la definisce
- La vista è un oggetto della base di dati
  - è utilizzabile nelle interrogazioni come se fosse una tabella

# DB forniture prodotti

P

| <u>CodP</u> | <u>NomeP</u> | <u>Colore</u> | <u>Taglia</u> | <u>Magazzino</u> |
|-------------|--------------|---------------|---------------|------------------|
| P1          | Maglia       | Rosso         | 40            | Torino           |
| P2          | Jeans        | Verde         | 48            | Milano           |
| P3          | Camicia      | Blu           | 48            | Roma             |
| P4          | Camicia      | Blu           | 44            | Torino           |
| P5          | Gonna        | Blu           | 40            | Milano           |
| P6          | Bermuda      | Rosso         | 42            | Torino           |

FP

| <u>CodF</u> | <u>CodP</u> | <u>Qta</u> |
|-------------|-------------|------------|
| F1          | P1          | 300        |
| F1          | P2          | 200        |
| F1          | P3          | 400        |
| F1          | P4          | 200        |
| F1          | P5          | 100        |
| F1          | P6          | 100        |
| F2          | P1          | 300        |
| F2          | P2          | 400        |
| F3          | P2          | 200        |
| F4          | P3          | 200        |
| F4          | P4          | 300        |
| F4          | P5          | 400        |

F

| <u>CodF</u> | <u>NomeF</u> | <u>NSoci</u> | <u>Sede</u> |
|-------------|--------------|--------------|-------------|
| F1          | Andrea       | 2            | Torino      |
| F2          | Luca         | 1            | Milano      |
| F3          | Antonio      | 3            | Milano      |
| F4          | Gabriele     | 2            | Torino      |
| F5          | Matteo       | 3            | Venezia     |

## Esempio n.1

- Definizione della vista *piccoli fornitori*
  - i fornitori che hanno meno di 3 soci sono considerati "piccoli fornitori"
- La vista piccoli fornitori
  - contiene il codice, il nome, il numero di soci e la sede dei fornitori che hanno meno di 3 soci

# Esempio n.1: definizione della vista

## ➤ Definizione della vista piccoli fornitori

- contiene il codice, il nome, il numero di soci e la sede dei fornitori che hanno meno di 3 soci

```
SELECT CodF, NomeF, NSoci, Sede  
FROM F  
WHERE Nsoci<3
```

# Esempio n.1: definizione della vista

## ➤ Definizione della vista piccoli fornitori

- contiene il codice, il nome, il numero di soci e la sede dei fornitori che hanno meno di 3 soci

```
SELECT CodF, NomeF, NSoci, Sede  
FROM F  
WHERE Nsoci<3
```

Interrogazione associata alla vista

# Esempio n.1: definizione della vista

- Definizione della vista piccoli fornitori
- contiene il codice, il nome, il numero di soci e la sede dei fornitori che hanno meno di 3 soci

```
CREATE VIEW PICCOLI_FORNITORI AS  
SELECT CodF, NomeF, NSoci, Sede  
FROM F  
WHERE Nsoci<3;
```

## Esempio n.1: interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino
- L'interrogazione può essere risolta senza l'uso di viste

```
SELECT *  
FROM F  
WHERE NSoci<3 AND  
Sede='Torino';
```

## Esempio n.1: interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino
- L'interrogazione può essere risolta usando la vista definita in precedenza

```
SELECT *  
FROM PICCOLI_FORNITORI  
WHERE Sede='Torino';
```

- La vista PICCOLI\_FORNITORI è usata come se fosse una tabella

# Riscrittura delle interrogazioni

- Se l'interrogazione fa riferimento a una vista, deve essere riscritta dal DBMS prima dell'esecuzione
- La riscrittura è svolta automaticamente
  - si sostituiscono i riferimenti alla vista con la sua definizione

## Esempio n.1: riscrittura dell'interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino

```
SELECT *  
FROM PICCOLI_FORNITORI  
WHERE Sede='Torino';
```

## Esempio n.1: riscrittura dell'interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino

```
SELECT CodF, NomeF, Sede, NSoci  
FROM PICCOLI_FORNITORI  
WHERE Sede='Torino';
```

## Esempio n.1: riscrittura dell'interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino

```
SELECT CodF, NomeF, Sede, NSoci  
FROM PICCOLI_FORNITORI  
WHERE Sede='Torino';
```

- Riscrittura della clausola SELECT
  - si rendono espliciti gli attributi presenti nella definizione della vista

## Esempio n.1: riscrittura dell'interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino

```
SELECT CodF, NomeF, Sede, NSoci  
FROM F  
WHERE NSoci<3 AND  
Sede='Torino';
```

- Introduzione della definizione della vista
- nella clausola FROM
  - nella clausola WHERE

## Esempio n.2: definizione della vista

- Definizione della vista numero di fornitori per prodotto
- per ogni prodotto, contiene il codice prodotto e il numero di fornitori diversi che l'hanno fornito

```
SELECT CodP, COUNT(*)  
FROM FP  
GROUP BY CodP
```



DBMG Interrogazione associata alla vista

## Esempio n.2: definizione della vista

- Definizione della vista numero di fornitori per prodotto
- per ogni prodotto, contiene il codice prodotto e il numero di fornitori diversi che l'hanno fornito

Nome della vista  
←

```
CREATE VIEW NUMFORNITORI_PER_PRODOTTO  
(CodP, NumFornitori) AS  
SELECT CodP, COUNT(*)  
FROM FP  
GROUP BY CodP;
```

## Esempio n.2: definizione della vista

- Definizione della vista numero di fornitori per prodotto
- per ogni prodotto, contiene il codice prodotto e il numero di fornitori diversi che l'hanno fornito

```
CREATE VIEW NUMFORNITORI_PER_PRODOTTO  
(CodP, NumFornitori) AS  
SELECT CodP, COUNT(*)  
FROM FP  
GROUP BY CodP;
```

Attributi della vista

## Esempio n.2: interrogazione

- Visualizzare il codice dei prodotti forniti dal maggior numero di fornitori
- Senza l'uso di viste

```
SELECT CodP
FROM FP
GROUP BY CodP
HAVING COUNT(*)=(SELECT MAX(NumFornitori)
                  FROM (SELECT COUNT(*) AS NumFornitori
                        FROM FP
                        GROUP BY CodP));
```

## Esempio n.2: interrogazione

- Visualizzare il codice dei prodotti forniti dal maggior numero di fornitori
- Usando la vista NUMFORNITORI\_PER\_PRODOTTO

```
SELECT CodP
FROM NUMFORNITORI_PER_PRODOTTO
WHERE NumFornitori=(SELECT MAX(NumFornitori)
                    FROM
                    NUMFORNITORI_PER_PRODOTTO);
```

## Considerazioni sugli esempi

- L'uso delle viste semplifica la scrittura delle interrogazioni
- La vista PICCOLI\_FORNITORI nasconde la definizione del concetto di piccolo fornitore
  - è possibile ridefinire il concetto di piccolo fornitore cambiando solo la definizione della vista
    - non è necessario modificare le interrogazioni che la usano
- La vista NUMFORNITORI\_PER\_PRODOTTO permette di evitare l'uso di table function

# Vantaggi offerti dalle viste

## ➤ Semplificazione delle interrogazioni

- espressioni molto complesse possono essere definite in modo più semplice mediante viste
  - scomposizione di un'interrogazione complessa in sottointerrogazioni associate alle viste
    - utile in presenza di sottointerrogazioni (complesse) ripetute

# Vantaggi offerti dalle viste

- Estensione del potere espressivo del linguaggio SQL
  - in assenza di table function, alcune tipologie di interrogazioni possono essere definite solo mediante l'uso di viste
    - in alternativa all'uso di codice procedurale

# Vantaggi offerti dalle viste

## ➤ Gestione della sicurezza

- è possibile introdurre meccanismi di protezione della privatezza diversi per ogni utente o gruppo
  - la vista diviene l'elemento a cui sono associate le autorizzazioni di accesso
  - ogni utente, o gruppo, accede alla base di dati solo mediante viste appropriate per le operazione che è abilitato a svolgere

# Vantaggi offerti dalle viste

## ➤ Evoluzione della base di dati

- in caso di ristrutturazione di una base di dati, è possibile definire viste che corrispondono a tabelle eliminate
  - la vista sostituisce la tabella presente nella base di dati pre-ristrutturazione e ora non più presente
    - non si devono riscrivere le interrogazioni scritte prima della ristrutturazione e presenti nelle applicazioni già sviluppate

## Creazione di una vista

```
CREATE VIEW NomeVista [(ElencoAttributi) ]  
AS InterrogazioneSQL;
```

# Creazione di una vista

- Se i nomi degli attributi della vista non sono specificati
  - sono utilizzati quelli presenti nella select dell'interrogazione SQL
- I nomi degli attributi devono essere specificati se
  - rappresentano il risultato di una funzione interna
  - rappresentano il risultato di un'espressione
  - sono costanti
  - due colonne (provenienti da tabelle diverse) hanno lo stesso nome

# Cancellazione di una vista

DROP VIEW *NomeVista*;

# Effetto della cancellazione di tabelle

- La cancellazione di una tabella a cui fa riferimento una vista può avere effetti diversi
  - eliminazione automatica delle viste associate
  - invalidazione automatica delle viste associate
  - divieto di esecuzione dell'operazione di cancellazione della tabella
- L'effetto dipende dal DBMS utilizzato

## Modifica della definizione di una vista

```
ALTER VIEW NomeVista [(ElencoAttributi)]  
AS InterrogazioneSQL;
```

## Aggiornabilità delle viste

- È possibile eseguire operazioni di aggiornamento dei dati presenti in una vista *solo* per alcune tipologie di viste

# Aggiornabilità delle viste

- È possibile eseguire operazioni di aggiornamento dei dati presenti in una vista *solo* per alcune tipologie di viste
- Standard SQL-92
  - sono aggiornabili le viste in cui una sola riga di ciascuna tabella di base corrisponde a una sola riga della vista
    - corrispondenza univoca tra le tuple della vista e le tuple della tabella su cui è definita
    - è possibile propagare senza ambiguità le modifiche apportate sulla vista verso ogni tabella su cui è definita

# Aggiornabilità delle viste

- *Non è aggiornabile* una vista che, nel blocco più esterno dell'interrogazione che la definisce
- non contiene la chiave primaria della tabella su cui è definita
  - contiene join che rappresentano corrispondenze uno a molti o molti a molti
  - contiene funzioni aggregate
  - contiene DISTINCT

# Esempio n.1

## ➤ Vista FORNITORE\_SEDE

```
CREATE VIEW FORNITORE_SEDE AS  
SELECT CodF, Sede  
FROM F;
```

## Esempio n.1: inserimento

➤ Inserimento in FORNITORE\_SEDE di  
(`F10`, `Roma`)

- corrisponde all'inserimento in F di  
(`F10`, NULL, NULL, `Roma`)
- gli attributi NomeF, NSoci devono ammettere il valore NULL

## Esempio n.1: cancellazione

➤ Cancellazione da FORNITORE\_SEDE di  
(`F1', `Torino')

- cancellazione da F di  
(`F1', `Andrea',2,`Torino')
- l'identificazione della tupla da cancellare è  
permessa dalla chiave primaria

## Esempio n.1: modifica

➤ Modifica in FORNITORE\_SEDE di  
(`F1`, `Torino`) in (`F1`, `Milano`)

- modifica in F di  
(`F1`, `Andrea`,2,`Torino`) in (`F1`, `Andrea`,2,`Milano`)
- l'identificazione della tupla da modificare è permessa dalla chiave primaria

## Esempio n.1: aggiornabilità

- La vista FORNITORE\_SEDE *è aggiornabile*
- ogni tupla della vista corrisponde a una sola tupla della tabella F
  - le operazioni di modifica effettuate sulla vista possono essere propagate alla tabella su cui è definita

### ➤ Vista NUMSOCI\_SEDE

```
CREATE VIEW NUMSOCI_SEDE AS  
SELECT DISTINCT NSoci, Sede  
FROM F;
```

## Esempio n.2: inserimento

➤ Inserimento in NUMSOCI\_SEDE di  
(40, 'Napoli')

- ❖ impossibile inserire in F  
(NULL, NULL, 40, 'Napoli')
- manca il valore della chiave primaria

## Esempio n.2: cancellazione

➤ Cancellazione da NUMSOCI\_SEDE di  
(2, 'Torino')

- più tuple sono associate alla coppia (2, 'Torino')
  - quale tupla deve essere cancellata da F?

## Esempio n.2: modifica

➤ Modifica in NUMSOCI\_SEDE di  
(2, 'Torino') in (3, 'Milano')

- più tuple sono associate alla coppia (2, 'Torino')
  - quale tupla deve essere modificata in F?

## Esempio n.2: aggiornabilità

- La vista NUMSOCI\_SEDE *non è aggiornabile*
- non è presente la chiave primaria della tabella F nella vista
    - l'inserimento di nuove tuple nella vista non è propagabile a F
  - alcune tuple della vista corrispondono a più tuple della tabella F
    - l'associazione tra tuple nella vista e tuple nella tabella è ambigua
    - non è possibile propagare le modifiche effettuate su tuple della vista alle tuple della tabella su cui è definita

# Aggiornabilità delle viste

- Alcune viste non aggiornabili possono diventare aggiornabili modificando l'espressione SQL associata alla vista
  - può essere necessario ridurre il contenuto informativo della vista

## Esempio n.3: vista non aggiornabile

```
CREATE VIEW FORNITORI_TORINO AS  
SELECT *  
FROM F  
WHERE Sede='Torino';
```

- La vista non è aggiornabile
  - non seleziona in modo esplicito la chiave primaria della tabella F
- È sufficiente sostituire al simbolo "\*" il nome degli attributi

## Esempio n.3: vista modificata

```
CREATE VIEW FORNITORI_TORINO AS  
SELECT CodF, NomeF, NSoci, Sede  
FROM F  
WHERE Sede='Torino';
```

➤ La vista è aggiornabile

## Esempio n.4: vista non aggiornabile

```
CREATE VIEW FORNITORI_IMPORTANTI (CodF, NomeF) AS
SELECT DISTINCT CodF, NomeF
FROM F, FP
WHERE F.CodF=FP.CodF AND
      Qta>100;
```

⇒ La vista non è aggiornabile

- è presente un join
- è presente la parola chiave DISTINCT

## Esempio n.4: vista modificata

```
CREATE VIEW FORNITORI_IMPORTANTI (CodF, NomeF) AS
SELECT CodF, NomeF
FROM F
WHERE CodF IN (SELECT CodF
               FROM FP
               WHERE Qta>100);
```

➤ La vista è aggiornabile

- il join è stato realizzato mediante IN
- la parola chiave DISTINCT non è più necessaria

## Esempio n.5: vista non aggiornabile

```
CREATE VIEW FORNITORI_MOLTO_IMPORTANTI (CodF, NomeF,  
TotQta) AS  
SELECT CodF, NomeF, SUM(Qta)  
FROM F, FP  
WHERE F.CodF=FP.CodF  
GROUP BY CodF, NomeF  
HAVING SUM(Qta)>500;
```

➤ La vista non è aggiornabile

- è presente una funzione aggregata
- è presente un join

## Esempio n.5: vista modificata

```
CREATE VIEW FORNITORI_MOLTO_IMPORTANTI (CodF, NomeF)
AS
SELECT CodF, NomeF
FROM F
WHERE CodF IN (SELECT CodF FROM FP
               GROUP BY CodF
               HAVING SUM(Qta)>500);
```

## Esempio n.5: vista modificata

```
CREATE VIEW FORNITORI_MOLTO_IMPORTANTI (CodF, NomeF)
AS
SELECT CodF, NomeF
FROM F
WHERE CodF IN (SELECT CodF FROM FP
               GROUP BY CodF
               HAVING SUM(Qta)>500);
```

➤ La vista è aggiornabile

- la group by è stata spostata nell'interrogazione nidificata

➤ Il *contenuto informativo è cambiato*

# Viste e gestione della privacy

- Le viste permettono di individuare sottoinsiemi di dati
  - individuati da un'espressione SELECT
- Assegnando a un utente l'accesso a specifiche viste si limitano
  - la sua visibilità sulle tabelle esistenti
  - le operazioni che può eseguire

## Esempio n.1

```
CREATE VIEW FORNITORI_TORINO(CodF, NomeF, NSoci) AS
SELECT CodF, NomeF, Nsoci
FROM F
WHERE Sede='Torino'
WITH CHECK OPTION;
```

- La vista FORNITORI\_TORINO seleziona solo i dati dei fornitori di Torino
- Un utente che ha accesso *solo* a questa vista
  - non può accedere alla tabella F
    - non può operare sui fornitori con sede diversa da Torino

## Esempio n.2

```
CREATE VIEW CODICE_NOME_FORNITORI(CodF, NomeF) AS  
SELECT CodF, NomeF  
FROM F;
```

- La vista CODICE\_NOME\_FORNITORI seleziona solo il codice e il nome dei fornitori
- Un utente che ha accesso *solo* a questa vista
  - non può accedere alla tabella F
  - non può operare sugli attributi NSoci e Sede

- Il dizionario dei dati contiene i metadati di una base di dati relazionale
  - i metadati sono informazioni (dati) sui dati
  - descrivono gli oggetti della base di dati (tabelle, viste, ...)
- Nel dizionario dei dati sono definite viste che limitano la visibilità dei singoli utenti sui metadati del dizionario
  - ogni utente può vedere solo le informazioni relative a oggetti della base di dati definiti da se stesso

## Esempio: Oracle

- Il DBMS Oracle rende disponibili numerose viste che descrivono i dati creati da un utente
- `USER_TABLES` contiene metadati relativi alle tabelle dell'utente
  - `USER_TAB_STATISTICS` contiene le statistiche calcolate sulle tabelle dell'utente
  - `USER_TAB_COL_STATISTICS` contiene le statistiche calcolate sulle colonne delle tabelle dell'utente