

Raggruppamento

1 Raggruppamento

Le funzioni di gruppo possono essere applicate a *partizioni delle relazioni* (invece che a tutte le tuple insieme). Per fare ciò, si specificano nell'apposita clausola **GROUP BY** le colonne in base alle quali effettuare il raggruppamento. Ogni tupla della relazione risultante rappresenta un gruppo di tuple della relazione originale che hanno gli stessi valori nelle colonne specificate nella clausola.

L'operazione di raggruppamento viene eseguita dopo la selezione (**WHERE**), ma prima della proiezione (**SELECT**), e quindi prima della valutazione delle funzioni di gruppo.

In un'interrogazione con raggruppamento, la clausola di proiezione può contenere solo:

- colonne specificate nella clausola **GROUP BY**;
- funzioni di gruppo.

Tutti gli altri attributi non sono utilizzabili, perché possono assumere più valori diversi nello stesso gruppo, ma a ogni gruppo corrisponde una sola tupla del risultato (e quindi un solo valore per ciascun attributo). Invece, le colonne specificate nella clausola **GROUP BY** hanno, per definizione, valori uguali in tutte le tuple di un gruppo.

1.1 Esempi

Per ogni regista, determinare quanti suoi film sono presenti in catalogo, di quanti generi diversi, e la valutazione minima, media e massima di tali film:

```
SELECT
  regista,
  COUNT(*) as numFilm,
  COUNT(DISTINCT genere) as numGeneri,
  MIN(valutaz) as minVal,
  AVG(valutaz) as avgVal,
  MAX(valutaz) as maxVal
FROM Film
GROUP BY regista;
```

Determinare il numero di figli di ciascun padre:

```

SELECT Padre, COUNT(*) AS NumFigli
FROM Paternita
GROUP BY Padre;

```

Per ogni padre, determinare il suo nome, il suo reddito, e la media dei redditi dei figli:

```

SELECT Padre, p.Reddito, AVG(f.Reddito)
FROM Paternita
  JOIN Persone p ON p.Nome = Padre
  JOIN Persone f ON f.Nome = Figlio
GROUP BY Padre, p.Reddito;

```

In quest'ultima interrogazione, la colonna rispetto alla quale si vorrebbe effettuare il raggruppamento è `Padre`. Così facendo, però, non sarebbe possibile includere nella `SELECT` l'attributo `p.Reddito`. Si ricorre allora a un artificio sintattico: siccome per ogni valore di `Padre` c'è un solo valore di `p.Reddito`, cioè `p.Reddito` è univocamente determinato da `Padre`, si può aggiungere `p.Reddito` alla `GROUP BY` *senza alterare il risultato*, ma permettendo l'uso di quest'ultimo attributo nella `SELECT`.

2 Condizioni su valori aggregati

Siccome l'operazione di raggruppamento e la valutazione delle funzioni di gruppo avvengono dopo l'esecuzione della clausola `WHERE`, non è possibile usare quest'ultima per imporre condizioni sui valori aggregati.

Esiste invece un'ulteriore clausola, `HAVING`, la quale permette di specificare un predicato che coinvolge (obbligatoriamente) funzioni di gruppo, oppure una combinazione booleana di tali predicati.

2.1 Esempi

Per ogni regista che ha girato almeno due film prima del 2000, determinare quanti sono tali film, di quanti generi diversi, e la valutazione minima, media e massima di tali film:

```

SELECT
  regista,
  COUNT(*) as numFilm,
  COUNT(DISTINCT genere) as numGeneri,
  MIN(valutaz) as minVal,
  AVG(valutaz) as avgVal,
  MAX(valutaz) as maxVal
FROM Film
WHERE anno < 2000

```

```
GROUP BY regista
HAVING COUNT(*) >= 2;
```

Selezionare i padri i cui figli sotto i 30 anni hanno un reddito medio maggiore di 25:

```
SELECT Padre
FROM Paternita
  JOIN Persone ON Nome = Figlio
WHERE Eta < 30
GROUP BY Padre
HAVING AVG(Reddito) > 25;
```

Selezionare i padri che hanno almeno due figli:

```
SELECT Padre
FROM Paternita
GROUP BY Padre
HAVING COUNT(*) >= 2;
-- oppure, senza raggruppamento e COUNT:
SELECT DISTINCT p.Padre
FROM Paternita p
  JOIN Paternita x ON p.Padre = x.Padre
WHERE p.Figlio <> x.Figlio;
```