



PEGASO
Università Telematica



Indice

1. RAGGRUPPAMENTI.....	3
2. PREDICATI	6
3. INSIEMI	7
4. NOTAZIONE POSIZIONALE.....	10
BIBLIOGRAFIA	12
SITOGRAFIA	13

1. Raggruppamenti

Le funzioni possono essere applicate a partizioni delle relazioni, attraverso la clausola GROUP BY.

GROUP BY listaAttributi

Esempio di una query¹:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

La clausola GROUP BY viene spesso utilizzata con funzioni aggregate (COUNT, MAX, MIN, SUM, AVG) per raggruppare il set di risultati in base a una o più colonne. Vediamo un esempio pratico. Data la tabella Paternita di Figura 1, vogliamo sapere il numero di figli per ciascun padre.

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Figura 1: tabella paternità.

La conseguente query è la seguente:

```
SELECT Padre, count(*) AS NumFigli
FROM Paternita
GROUP BY Padre
```

Il risultato di Figura 2 è il result-set della query.

¹ https://www.w3schools.com/sql/sql_groupby.asp

Padre	NumFigli
Sergio	1
Luigi	2
Franco	2

Figura 2: result-set della query con gruppi.

Viene prima prodotto il result-set normale. Successivamente vengono raggruppati per attributo ed infine agisce la funzione count(*).

Vediamo una seconda interrogazione:

Interrogazione 27: estrarre la somma degli stipendi di tutti gli impiegati dello stesso dipartimento dalla tabella di Figura 3.

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giovanni	Verde	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Gialli	Direzione	7	73	Genova
Paola	Rosati	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Figura 3: tabella impiegato.

La query è la seguente:

- SELECT Dipart, SUM(Stipendio)
- FROM Impiegato
- **GROUP BY Dipart**

Vediamo passo-passo come lavora tale query.

- *Passo 1:* L'interrogazione viene eseguita come se la clausola GROUP BY non esistesse:
 - SELECT Dipart, Stipendio
 - FROM Impiegato
- *Passo 2:* la tabella ottenuta viene analizzata, dividendo le righe in insiemi caratterizzati dagli stessi valori degli attributi che compaiono nella clausola GROUP BY, ottenendo la tabella di Figura 4.

Dipart	Stipendio
Amministrazione	45
Amministrazione	40
Amministrazione	40
Produzione	36
Produzione	46
Distribuzione	45
Direzione	80
Direzione	73

Figura 4: passo 1 della query.

- *Passo 3:* agisce l'operatore aggregato su ogni sottoinsieme, producendo la tabella finale di Figura 5.

Dipart	SUM(Stipendio)
Amministrazione	125
Produzione	82
Distribuzione	45
Direzione	153

Figura 5: tabella finale.

2. Predicati

Abbiamo visto che con la clausola GROUP BY, le righe possono venire raggruppate in sottoinsiemi.

Si può avere bisogno però di considerare solo i sottoinsiemi che soddisfino alcune condizioni. In questo caso si hanno due possibilità:

- Clausola WHERE (se singole righe)
- Clausola **HAVING** se le condizioni sono delle condizioni di tipo aggregato:
 - Descrive le condizioni che si devono applicare al termine dell'esecuzione di una interrogazione che fa uso della clausola GROUP BY

Vediamo adesso un esempio.

Interrogazione 31: estrarre dalla tabella Impiegato i dipartimenti che erogano più di 100 in stipendi.

La query è la seguente:

- SELECT Dipart, SUM(Stipendio) AS Sommastipendi
- FROM Impiegato
- GROUP BY Dipart
- **HAVING** SUM(Stipendio) >100

Si procede con gli stessi passi descritti per le interrogazioni con GROUP BY. Dopo aver raggruppato le righe in base al valore dell'attributo Dipart, viene valutato il predicato argomento della clausola HAVING che seleziona i dipartimenti in base a quanto richiesto. Il risultato è in Figura 6.

Dipart	SommaStipendi
Amministrazione	125
Direzione	153

Figura 6: risultato della query con la clausola HAVING.

3. Insiemi

Il linguaggio SQL mette a disposizione una serie di operatori insiemistici simili a quelli dell'algebra relazionale:

- Unione (Union)
- Intersezione (Intersect)
- Differenza (Minus o Except)

La sintassi è la seguente:

SelectSQL{<union | intersect | except> [all] SelectSQL}

Tali operatori agiscono sui result-set delle singole SELECT coinvolte (i duplicati vengono eliminati).

Infatti, la SELECT da sola non permette di fare unioni; serve quindi un costrutto esplicito, ovvero:

SELECT ...

UNION [ALL]

SELECT ...

Attenzione perché i duplicati vengono eliminati (a meno che si usi *all*); anche dalle proiezioni!

Vediamo di costruire una query.

Interrogazione 33: estrarre i nomi ed i cognomi degli impiegati dalla tabella Impiegato di Figura 3.

La query è la seguente:

- SELECT Nome
- FROM Impiegato
 - **Union**
- SELECT Cognome
- FROM Impiegato

Il risultato è quello illustrato in Figura 7.

Nome
Mario
Carlo
Giovanni
Franco
Lorenzo
Paola
Marco
Rossi
Bianchi
Verdi
Neri
Gialli
Rosati

Figura 7: risultato dell'operatore Union.

Vediamo un'altra interrogazione.

Interrogazione 34: Se si vogliono mantenere i duplicati. Estrarre i nomi e i cognomi di tutti gli impiegati, eccetto quelli appartenenti al dipartimento Amministrazione, mantenendo i duplicati.

La query è la seguente:

- SELECT Nome
- FROM Impiegato
- WHERE Dipart <>'Amministrazione'
- » **Union all**
- SELECT Cognome
- FROM Impiegato
- WHERE Dipart<>'Amministrazione'

Il risultato è quello illustrato in Figura 8.

Nome
Carlo
Franco
Carlo
Lorenzo
Marco
Bianchi
Neri
Rossi
Gialli
Franco

Figura 8: risultato con mantenimento dei duplicati.

4. Notazione posizionale

Per concludere questa UDA, come ultima cosa, ci chiediamo come funziona l'ordine degli attributi con il quale la SELECT produce le tabelle. Vediamo un esempio alla luce degli operatori spiegati in questa dispensa. Date le due tabelle Maternità e Paternità di Figura 9 e Figura 10.

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Figura 9: tabella Maternità.

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Figura 10: tabella paternità.

Ci chiediamo cosa produca la query:

```
SELECT padre, figlio
FROM paternita
UNION
SELECT madre, figlio
FROM maternita
```

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Figura 11: risultato posizionale della query.

In Oracle-SQL viene messo il nome del primo operando.

La seguente istruzione non è semanticamente corretta poiché metterebbe la colonna dell'attributo madre sotto quella della colonna figlio.

```
SELECT padre as genitore, figlio
FROM paternita
UNION
SELECT figlio, madre as genitore
FROM maternita
```

Bibliografia

- Atzeni P., Ceri S., Fraternali P., Paraboschi S., Torlone R. (2018). Basi di Dati. McGraw-Hill Education.
- Batini C., Lenzerini M. (1988). Basi di Dati. In Cioffi G. and Falzone V. (Eds). Calderini. Seconda Edizione.

Sitografia

- https://www.w3schools.com/sql/sql_groupby.asp