

CONTINUAZIONE OPERATORI AGGREGAZIONE:

■ GROUP BY \Rightarrow Suddividere il Risultato in GRUPPI \rightarrow RAGGRUPPAMENTO
↓
Che Condividono lo Stesso Valore di un Attributo

ESEMPIO: tutti gli Studenti di Verona

SELECT
From
[WHERE...]
GROUP BY
[HAVING...]

Dato: STUDENTE (matricola, nome, cognome, Indirizzo, città, Media)

■ Contare Tutti gli Studenti SELECT count(*) FROM STUDENTE

1
≠

count(città) perché NON so se città sia NOT NULL o meno

e a sua volta
≠

count(DISTINCT città)

■ Contare tutti gli Studenti Per Ogni Città:

SELECT Città, count(*)
From Studente

Date le RIGHE delle Tab Studenti; si
Creano Gruppi Per Ogni Città e Solamente

Group By città

Poi Viene fatto il COUNT A GRUPPO

Con GROUP BY, Nella Select Posso Indicare Solo Attributi che Compiono Nella Clauses GROUP BY + operatori di AGGREGAZIONE visto che il Risultato avrà 1 Riga per Ogni GRUPPO

Con CLAUSOLA WHERE:

```
SELECT Studente  
FROM ①  
WHERE ② matricola <> 'VR001'  
GROUP BY ③ città
```

- ① determino Righe d'Interesse
- ② Reggruppo \Rightarrow GROUP BY
- ③ Filtra gli Attributi
- ④

WHERE mi permette di Selezionare le RIGHE sul Quale Calcolare il Risultato.

GROUP BY \Rightarrow Reggruppare le Righe Restituite dal WHERE

Voglio Una Selezione Anche Sui Gruppi \Rightarrow HAVING

■ Solo CITTÀ Con 10 STUDENTI:

con Aggiunta:

■ Attributi GROUP BY

```
SELECT città, COUNT(*) ④ Restituisco  
FROM Studente ③ Seleziona Righe di Partenza per il RISULTATO  
GROUP BY città ② Calcola i Gruppi;  
HAVING count(*)  $\geq$  10 ③ Seleziona i Gruppi
```

■ Operatori

SOTTOLINEARE DIFFERENZA TRA HAVING e WHERE

SELECT Lower(città), avg(media) :: DECIMAL(5,2)

From Studente

CASTING IN SOLA

WHERE

VISUALIZZAZIONE

GROUP BY Lower(città)

HAVING avg(media) > 23.5

ho Raggruppato in
modo CASE-SENSITIVE

Deve Esserci COERENZA

anche Nella SELECT.



'Verona'
'VERONA'
'verona'

} stesso
Gruppo

Se NON Avessi Messo lower(città)
ci Sarebbero ECCEZIONI → ERRORE
visto che Avrei Colonne diverse MA
Nella Stesso Gruppo

Risultato del TIPO \Rightarrow

CITTÀ	MEDIA
Verona	24.00
Padova	23.50

Non Aspettavo... Ma il

CASTING può IMBROGLIARE infatti

Padova poteva Avere 23.50g e Quindi facendo CAST a 2 cifre tornerà 23.50

OPERATORE DI JOIN:

METODO BASE:

SELECT *
From a,b,c

Mi porta ad un PRODOTTO CARTESIANO
che poi filtra con un WHERE

UTILIZZO CLAUSOLA DI JOIN:

SELECT *

From t₁ Tipo JOIN t₂ ON

Tipi Di JOIN:

- Inner \Leftrightarrow con il WHERE
 - Full
 - Right
 - Left
- NON OTTENIBILI ALTRIMENTI

INNER-JOIN: θ -Join dell'Algebra Relazionale

Combine Ciascuna Riga di t₁ con Ogni Riga di t₂ che Soddisfano la Condizione di JOIN

IMPIEGATO (matricola, Nome, Cognome, nomeReperto)

REPARTO (nome, indirizzo, tel, città)

SELECT I.nome, I.cognome, I.nomeReperto, R.tel
FROM Impiegato AS I, Reparto AS R
WHERE I.nomeReperto = R.nome

Oppure:

From Impiegato JOIN REPARTO ON I.nomeReparto = R.nome

Voglio tutti gli Impiegati e per quelli Assegnati ad un Reparto Anche il Numero di Telefono

SELECT I.nome, I.cognome, I.nomeReparto, R.tel

FROM Impiegato I LEFT JOIN Reparto R ON I.nomeRep=R.nome



Esegue un INNER-JOIN e per ciascuna riga di Tab₁ (impiegato) che non ha un corrispondente in Tab₂ (reparto) aggiunge la riga rassegnando a NULL gli attributi di Tab₂

Quindi & Impiegati che non hanno un Reparto Assegnato aggiungono al risultato una riga per quell'Impiegato con tel=NULL

C'è anche il DUALE ovvero il RIGHT [OUTER] JOIN dove andrà a tenere le righe di Tab₂ che non hanno un corrispondente dove tutti Attrib. di Tab₁=NULL

T₁ FULL OUTER JOIN T₂ dove calcola ① INNER }
NB! ② LEFT } Includo
} ③ RIGHT } Tutti

→ diverso da PRODOTTO CARTESIANO visto che prima Combinò solo ciò che ha corrispondenza, altrimenti posso

Usare il CROSS-JOIN

ESEMPIO:

T ₁	A ₁	A ₂	T ₂	A ₃	A ₄
	1	a		a	10
	2	b		a	14
	3	c		b	18
				f	20

■ SELECT * FROM T₁, T₂ \Rightarrow TUTTE COMBINAZIONI

■ SELECT * FROM T₁ JOIN T₂ ON t₁.A₂ = t₂.A₃ \Rightarrow ESCLUDERÒ
L'ULTIMA RIGA DI ENTRAMBE LE TABELLE

■ SELECT * FROM T₁ LEFT-JOIN T₂ ON t₁.A₂ = t₂.A₃ \Rightarrow ALLORA
AVRÒ TUTTE LE RIGHE DI T₁ MA PER L'ULTIMA RIGA AVRÒ (A₃, A₄)
CHE SONO = NULL

■ SELECT * FROM T₁ RIGHT-JOIN T₂ ON t₁.A₂ = t₂.A₃ \Rightarrow ALLORA
AVRÒ TUTTE LE RIGHE DI T₂ MA PER L'ULTIMA RIGA AVRÒ (A₁, A₂)
CHE SONO = NULL

■ SELECT * FROM T₁ FULL-JOIN T₂ ON t₁.A₂ = t₂.A₃ \Rightarrow ALLORA
AVRÒ TUTTE LE RIGHE DI T₁ MA PER L'ULTIMA RIGA AVRÒ (A₃, A₄)
CHE SONO = NULL MA ANCHE TUTTE LE RIGHE DI T₂ MA L'ULTIMA
NON AVENDO CORRISPONDENZA HA ATTRIBUTI (A₁, A₂) = NULL

