



Il Linguaggio SQL DML/DQL

Basi di Dati

Corso di Laurea in Informatica per il Management

Alma Mater Studiorum - Università di Bologna

Prof. Marco Di Felice

Dipartimento di Informatica – Scienza e Ingegneria marco.difelice3@unibo.it

Il Linguaggio SQL

LINGUAGGI supportati dai RDBMS

3. SQL (Structured Query Language)

Diverse **versioni** del linguaggio:

- SQL-86 → Costrutti base
- \circ SQL-89 \rightarrow (SQL1) Integrità referenziale
- SQL-92 (SQL2) → SQL Interattivo, sistema tipi
- SQL:1999 (SQL3) → Modello ad oggetti
- SQL:2003 (SQL3) → Nuove parti: SQL/JRT, SQL/XML
- SQL:2006 (SQL3) → Estensione di SQL/XML
- SQL:2008 (SQL3) \rightarrow Lievi aggiunte
- 0

http://troels.arvin.dk/db/rdbms/

Il Linguaggio SQL

Due componenti principali:

DDL (Data Definition Language)
 Contiene i costrutti necessari per la creazione e modifica dello schema della base di dati.



DML/DQL (Data Manipulation/Query Language)
 Contiene i costrutti per le interrogazioni e di inserimento, eliminazione e modifica di dati.

Esempio di **interrogazione** (**query**) \rightarrow Recuperare nome e cognome dello studente con numero di matricola pari a 4678...

STUDENTI

Matricola	Nome	Cognome	DataNascita
4566	Marco	Rossi	3/5/1989
4678	Michele	Bianchi	2/5/1989
4900	Antonio	Rossi	14/3/1990



Nome	Cognome
Michele	Bianchi

Le operazioni di **interrogazione** vengono implementate dal costrutto di select.

SEMANTICA: Effettua il **prodotto cartesiano** delle Tabella₁, ..., Tabella_N. Da queste, **estrai le righe** che rispettano la **Condizione**. Di quest'ultime, preleva solo le colonne corrispondenti a: **Attributo₁**, ..., **Attributo_M**.

Nel caso di una sola tabella:



Nel caso di una sola tabella:



Nel caso di una sola tabella:

select

from Tabella

where Condizione

Attributo_i, Attributo_{j,} ... Attributo_m

STEP 3: Si costruisce la tabella risultato ...



Numero di colonne definito dalla clausola SELECT



Numero di righe definito dalla clausola WHERE

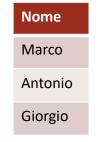


Attributo ₁	Attributo _i	Attributo _m

IMPIEGATI

Esempio 1. Selezionare i nomi degli impiegati che lavorano nell'ufficio A.

Codice	Nome	Cognome	Ufficio	Stipendio	
123	Marco	Marchi	А	15000	
125	Michele	Monti	В	18000	
134	Antonio	Verdi	А	25000	
156	Giorgio	Rossi	Α	32000	



SELECT NOME FROM IMPIEGATI WHERE (UFFICIO="A")

Esempio 2. Selezionare i nomi degli impiegati che guadagnano più di 20000 euro annui.

IMPIEGATI

Codice	Nome	Cognome	Ufficio	Stipendio	Nome
123	Marco	Marchi	А	15000	Antonio
125	Michele	Monti	В	18000	Giorgio
134	Antonio	Verdi	А	25000	diolgio
156	Giorgio	Rossi	А	32000	

SELECT NOME FROM IMPIEGATI WHERE (STIPENDIO>20000)

Esempio 3. Selezionare nomi e cognomi degli impiegati che lavorano nell'ufficio B e guadagnano più di 20000 euro annui.

IMPIEGATI				
Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	А	15000
125	Michele	Monti	В	18000
134	Antonio	Verdi	А	25000
156	Giorgio	Rossi	А	32000

SELECT NOME, COGNOME FROM IMPIEGATI WHERE ((STIPENDIO>20000) AND (UFFICIO="B"))

La clausola **where** specifica quali righe della/e tabella/e devono comparire nel risultato finale.

La condizione della clausola può contenere un'espressione booleana, o una combinazione di espressioni mediante gli operatori and, or, not.

```
SELECT CODICE

FROM IMPIEGATI

WHERE NOT((NOME="Marco") AND (UFFICIO="A"))
```

Nella clausola **where**, è possibile fare **confronti tra stringhe** usando l'operatore **like** e l'utilizzo di **wildcard**:

→ carattere arbitrario

% → sequenza di caratteri arbitraria.

In questo modo, è possibile trovare tutte le stringhe che rispettano un certo pattern. Es: selezionare il codice di tutti gli impiegati il cui nome inizi per 'M', abbia una 'r' come terzo carattere, e termini per 'o'.

SELECT CODICE FROM IMPIEGATI WHERE (NOME LIKE 'M_R%O')

Nella clausola where, l'operatore between consente di verificare l'appartenenza ad un certo insieme di valori.

Es. Trovare i codici degli impiegati il cui stipendio sia compreso tra i 24000 ed i 34000 euro annui.

IMPIEGATI				
Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	Α	15000
125	Michele	Monti	В	18000
134	Antonio	Verdi	А	25000
156	Giorgio	Rossi	A	32000

SELECT NOME FROM IMPIEGATI WHERE (STIPENDIO BETWEEN 24000 AND 34000)

D. Cosa accade nella clausola **where** in caso di valori NULL... Vengono inclusi nel risultato finale? **NO**!

IMPIEGATI				
Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	А	15000
125	Michele	Monti	В	18000
134	Antonio	Verdi	А	NULL
156	Giorgio	Rossi	А	32000

SELECT NOME FROM IMPIEGATI WHERE (STIPENDIO > 20000)

In generale, SQL utilizza una logica a tre valori: true (T), false (F), unknown (U). Esistono gli operatori IS NULL ed IS NOT NULL.

IMPIEGATI					
Codice	Nome	Cognome	Ufficio	Stipendio	
123	Marco	Marchi	Α	15000	
125	Michele	Monti	В	18000	
134	Antonio	Verdi	А	NULL	
156	Giorgio	Rossi	Α	32000	

SELECT NOME FROM IMPIEGATI WHERE ((STIPENDIO > 20000)
OR (STIPENDIO IS NULL))

La clausola **select** specifica quali **colonne** delle righe selezionate devono comparire nel risultato finale.

L'asterisco (*) indica tutte le colonne della tabella.

```
SELECT * FROM IMPIEGATI
WHERE (NOME="Marco") AND (UFFICIO="A")
```



Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	Α	15000

E' possibile **ridenominare** le colonne del risultato di una query attraverso il costrutto as.

```
SELECT NOME as Name, Cognome as LastName FROM IMPIEGATI
WHERE (NOME="Marco")
```

Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	Α	12000
145	Marco	Bianchi	В	24000
167	Lucia	Di Lucia	С	36000



Name	LastName
Marco	Marchi
Marco	Bianchi

E' possibile usare *espressioni aritmetiche* (semplici) sui valori degli attributi di una **SELECT**.

SELECT NOME as Name, Stipendio/12 as SalaryM FROM IMPIEGATI
WHERE (NOME="Marco")

Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	Α	12000
145	Marco	Bianchi	В	24000
167	Lucia	Di Lucia	С	36000



Name	SalaryM
Marco	1000
Marco	2000

La clausola **from** specifica la lista delle tabelle cui si deve accedere (nel caso #tabelle>1, si effettua il **prodotto cartesiano** delle stesse).

E' possibile specificare degli alias per i nomi delle tabelle, mediante il costrutto as:

```
SELECT CODICE
FROM IMPIEGATI AS I
WHERE (I.NOME="MICHELE")
```

Vediamo come funziona la SELECT su più tabelle.

Es. Selezionare il numero di telefono dell'impiegato con codice 145

IMPIEGAT	1			
Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	Α	12000
145	Marco	Bianchi	В	24000
167	Lucia	Di Lucia	Α	36000
187	Giorgio	Rossi	В	12000

SEDI	
UffNum	Telefono
Α	2034333
В	2035434

SELECT TELEFONO AS TEL FROM IMPIEGATI, SEDI WHERE (UFFICIO=UFFNUM) AND (CODICE=145)

... COSA FA QUESTA QUERY??

IMPIEGATI

Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	Α	12000
145	Marco	Bianchi	В	24000
167	Lucia	Di Lucia	Α	36000
187	Giorgio	Rossi	В	12000

SEDI

UffNum	Telefono
Α	2034333
В	2035434

SELECT TELEFONO AS TEL FROM IMPIEGATI, SEDI WHERE (UFFICIO=UFFNUM) AND (CODICE=145)

STEP 1. Si effettua il prodotto cartesiano delle due tabelle ...

Codice	Nome	Cognome	Ufficio	Stipendio	UffNum	Telefono
123	Marco	Marchi	А	12000	Α	2034333
145	Marco	Bianchi	В	24000	Α	2034333
167	Lucia	Di Lucia	Α	36000	А	2034333
187	Giorgio	Rossi	В	12000	Α	2034333
123	Marco	Marchi	Α	12000	В	2035434
145	Marco	Bianchi	В	24000	В	2035434
167	Lucia	Di Lucia	Α	36000	В	2035434
187	Giorgio	Rossi	В	12000	В	2035434

BASI DI DATI

SELECT TELEFONO AS TEL FROM IMPIEGATI, SEDI WHERE (HEETCTO=HEENLIM)

STEP 2. Si selezionano le righe con valori comuni nelle tue tabelle ...

WHERE (UFFICIO=UFFNUM) AND (CODICE=145)

Codice	Nome	Cognome	Ufficio	Stipendio	UffNum	Telefono
123	Marco	Marchi	А	12000	Α	2034333
145	Marco	Bianchi	В	24000	A	2034333
167	Lucia	Di Lucia	А	36000	А	2034333
187	Giorgio	Rossi	В	12000	А	2034333
123	Marco	Marchi	Α	12000	В	2035434
145	Marco	Bianchi	В	24000	В	2035434
167	Lucia	Di Lucia	A	36000	В	2035434
187	Giorgio	Rossi	В	12000	В	2035434

BASI DI DATI

PROF. MARCO DI FELICE – CORSO DI LAUREA IN INFORMATICA PER IL MANAGEMENT

SELECT TELEFONO AS TEL FROM IMPIEGATI, SEDI WHERE (UFFICTO=UFFNUM)

STEP 3. Si selezionano le righe relative all'impiegato 145 ...

WHERE (UFFICIO=UFFNUM) AND (CODICE=145)

Codice	Nome	Cognome	Ufficio	Stipendio	UffNum	Telefono
123	Marco	Marchi	A	12000	А	2034333
145	Marco	Bianchi	В	24000	Α	2034333
167	Lucia	Di Lucia	А	36000	А	2034333
187	Giorgio	Rossi	В	12000	Α	2034333
123	Marco	Marchi	А	12000	В	2035434
145	Marco	Bianchi	В	24000	В	2035434
167	Lucia	Di Lucia	Α	36000	В	2035434
187	Giorgio	Rossi	В	12000	В	2035434

BASI DI DATI

SELECT TELEFONO AS TEL FROM IMPIEGATI, SEDI WHERE (UFFTCTO=UFFNUM)

STEP 4. Si seleziona la colonna dell'attributo Telefono ...

WHERE (UFFICIO=UFFNUM) AND (CODICE=145)

Codice	Nome	Cognome	Ufficio	Stipendio	UffNum	Telefono
123	Marco	Marchi	Α	12000	А	2034333
145	Marco	Bianchi	В	24000	Α	2034333
167	Lucia	Di Lucia	А	36000	Α	2034333
187	Giorgio	Rossi	В	12000	Α	2034333
123	Marco	Marchi	А	12000	В	2035434
145	Marco	Bianchi	В	24000	В	2035434
167	Lucia	Di Lucia	А	36000	В	2035434
187	Giorgio	Rossi	В	12000	В	2035434

BASI DI DAT

SELECT TELEFONO AS TEL FROM IMPIEGATI, SEDI WHERE (UFFICIO=UFFNUM) AND (CODICE=145)



TEL

2035434

STEP 5. Si costruisce il risultato finale ...

D. Che accade se le tabelle della clausola from hanno attributi con nomi uguali?

SELECT TELEFONO AS TEL ???? ERRORE!!!
FROM IMPIEGATI, SEDI
WHERE (UFFICIO=UFFICIO) AND (CODICE=145)

Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	Α	12000
145	Marco	Bianchi	В	24000
167	Lucia	Di Lucia	Α	36000
187	Giorgio	Rossi	В	12000

SEDI	
Ufficio	Telefono
А	2034333
В	2035434

In questi casi, si può utilizzare la notazione **NomeTabella.NomeAttributo** per far riferimento ad un attributo in maniera non ambigua.

```
SELECT TELEFONO AS TEL
FROM IMPIEGATI, SEDI
WHERE (IMPIEGATI.UFFICIO=SEDI.UFFICIO) AND (CODICE=145)

SELECT TELEFONO AS TEL
FROM IMPIEGATI AS I, SEDI AS S
WHERE (I.UFFICIO=S.UFFICIO) AND (CODICE=145)
```

ATTENZIONE: Il risultato di una query SQL potrebbe avere righe duplicate!

SELECT NOME AS NAME FROM IMPIEGATI AS I WHERE (STIPENDIO <25000)

IMPIEGATI						
Codice	Nome	Cognome	Ufficio	Stipendio		Nam
123	Marco	Marchi	Α	12000		
145	Marco	Bianchi	В	24000		Mar
167	Lucia	Di Lucia	C	36000		Marc
107	Lucia	Di Lucia		30000		

Il costrutto distinct (nella select) consente di rimuovere i duplicati nel risultato.

Il costrutto all (nella select) NON rimuove i duplicati (comportamento di default).

SELECT **DISTINCT** NOME AS NAME FROM IMPIEGATI AS I WHERE (STIPENDIO >20000)





Nella clausola **where** possono comparire più istanze della stessa tabella mediante il meccanismo degli **alias** ...

Es. Selezionare i nomi dei nonni di Matteo Bianchi.

GENITORI			
Nome	Cognome	NomeGen	Cognome Gen
Matteo	Bianchi	Michele	Bianchi
Michele	Bianchi	Gianni	Bianchi
Matteo	Bianchi	Lucia	Rossi
Lucia	Rossi	Sara	Rossi
Nicola	Verdi	Simone	Verdi

```
SELECT NOME, COGNOME
FROM GENITORI, GENITORI
WHERE (GENITORI.NOME=GENITORI.NOMEGEN) ...
```

GENITORI

Nome	Cognome	NomeGen	CognomeGen
Matteo	Bianchi	Michele	Bianchi
Michele	Bianchi	Gianni	Bianchi
Matteo	Bianchi	Lucia	Rossi
Lucia	Rossi	Sara	Rossi
Nicola	Verdi	Simone	Verdi

SELECT G2.NOMEGEN, G2.COGNOMEGEN

FROM GENITORI AS G1, GENITORI AS G2

WHERE (G1.NOMEGEN=G2.NOME) AND (G1.COGNOMEGEN=G2.COGNOME)

AND (G1.NOME="MATTEO") AND (G1.COGNOME="BIANCHI")

GENITORI				
Nome	Cognome	NomeGen	CognomeGen	
Matteo	Bianchi	Michele	Bianchi	
Michele	Bianchi	Gianni	Bianchi	
Matteo	Bianchi	Lucia	Rossi	
Lucia	Rossi	Sara	Rossi	
Nicola	Verdi	Simone	Verdi	

Il costrutto **order by** consente di **ordinare le righe** del risultato di un'interrogazione in base al valore di un attributo specificato.

```
order by Attributo_1 [asc|desc], ..., Attributo_N [asc|desc]
```

```
SELECT *
FROM IMPIEGATI
WHERE (UFFICIO="A")
ORDER BY STIPENDIO
```

Deve comparire sempre dopo la clausola where!

Supponiamo di voler scrivere una **query** per contare il numero di Impiegati che lavorano nell'ufficio A.

Problema: La SELECT vista fin qui opera a livello di tuple, e non a livello di colonne ..

IMPIEGATI				
Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	А	12000
145	Marco	Bianchi	В	24000
167	Lucia	Di Lucia	А	36000
187	Giorgio	Rossi	В	12000

Da questa colonna dovremmo estrarre un solo valore!

Gli operatori **aggregati** si applicano a gruppi di tuple (e non tupla per tupla), e producono come risultato un solo valore.

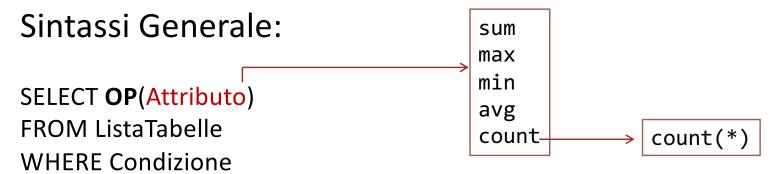
Vengono in genere inseriti nella select, e valutati <u>DOPO</u> la clausola where e from.

```
count (* | [distinct|all] Lista Attributi)
```

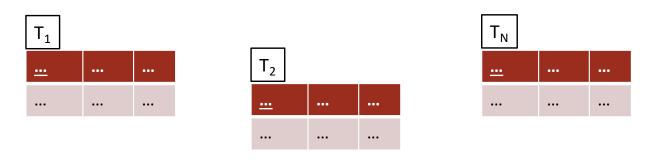
* > si applica su tutti gli attributi, in pratica conta il numero di righe

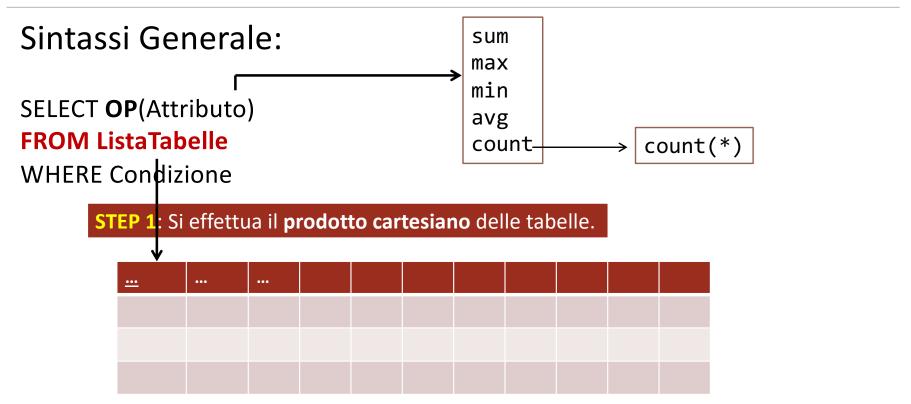
Gli operatori **aggregati** si applicano a gruppi di tuple (e non tupla per tupla), e producono come risultato un solo valore.

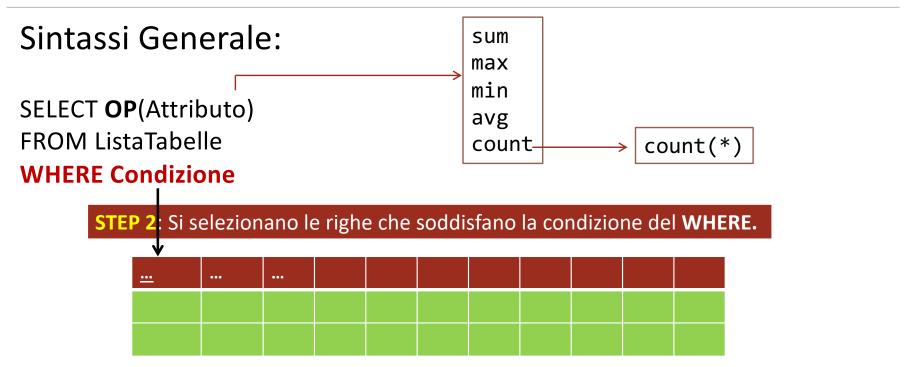
sum (Lista Attributi)avg (Lista Attributi)min (Lista Attributi)max (Lista Attributi)

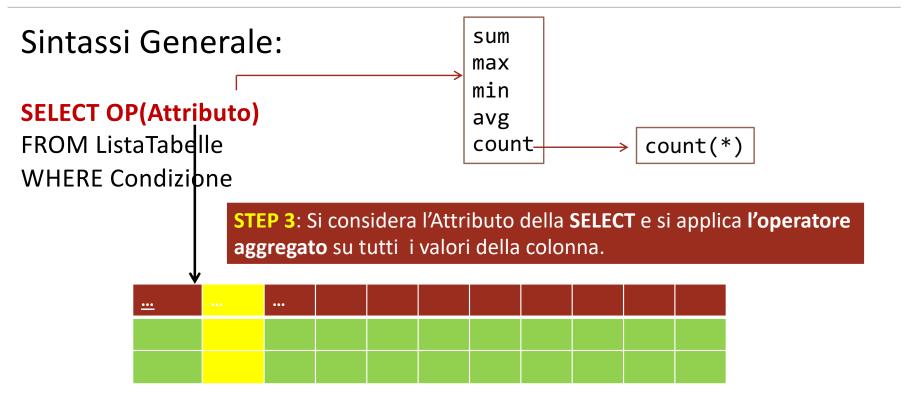


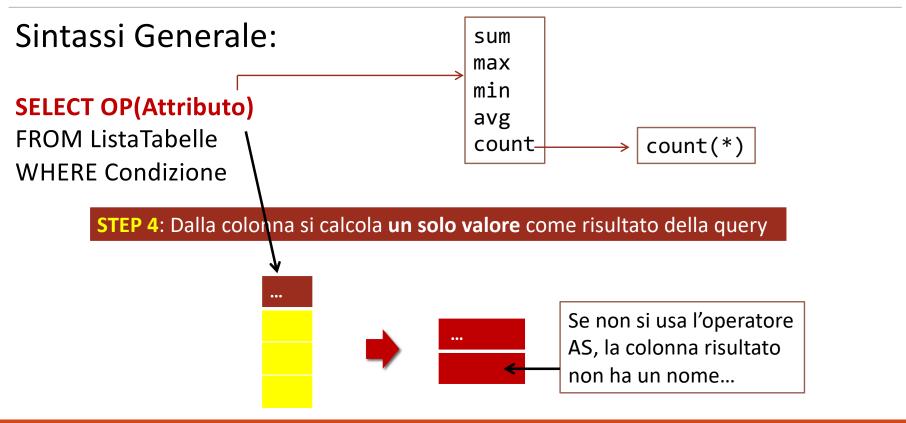
STEP 0: Si considerano le tabelle indicate nella clausola FROM









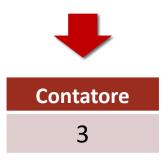


Es. Contare il numero di strutturati che lavorano nel *Dipartimento di Fisica*.

STRUTTURATI

<u>Codice</u>	Nome	Cognome	Tipo	Dipartimento	Stipendio
123	Marco	Marchi	Associato	Chimica	20000
124	Michele	Micheli	Associato	Fisica	20000
125	Lucia	Di Lucia	Ordinario	Fisica	30000
126	Dario	Rossi	Ordinario	Informatica	32000
127	Mario	Rossi	Ricercatore	Informatica	15000
129	Michele	Bianchi	Associato	Fisica	20000

SELECT COUNT(*) AS CONTATORE
FROM STRUTTURATI
WHERE (DIPARTIMENTO="FISICA")

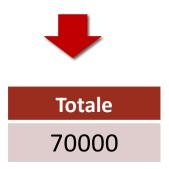


Es. Contare la somma complessiva degli stipendi degli strutturati *del dipartimento di Fisica*.

STRUTTURATI

<u>Codice</u>	Nome	Cognome	Тіро	Dipartimento	Stipendio
123	Marco	Marchi	Associato	Chimica	20000
124	Michele	Micheli	Associato	Fisica	20000
125	Lucia	Di Lucia	Ordinario	Fisica	30000
126	Dario	Rossi	Ordinario	Informatica	32000
127	Mario	Rossi	Ricercatore	Informatica	15000
129	Michele	Bianchi	Associato	Fisica	20000

SELECT SUM(STIPENDIO) AS TOTALE FROM STRUTTURATI
WHERE (DIPARTIMENTO="FISICA")



Es. Determinare il valore dello *stipendio più alto* tra i professori associati.

STRUTTURATI

<u>Codice</u>	Nome	Cognome	Tipo	Dipartimento	Stipendio
123	Marco	Marchi	Associato	Chimica	20000
124	Michele	Micheli	Associato	Fisica	50000
125	Lucia	Di Lucia	Ordinario	Fisica	30000
126	Dario	Rossi	Ordinario	Informatica	32000
127	Mario	Rossi	Ricercatore	Informatica	15000
129	Michele	Bianchi	Associato	Fisica	20000

SELECT MAX(STIPENDIO) AS MAXSTIPENDIO FROM STRUTTURATI WHERE (TIPO="ASSOCIATO")



Es. Estrarre codice e stipendio del professore associato che ha lo stipendio più alto ...

ERRORE!

```
SELECT CODICE, MAX(STIPENDIO)

FROM STRUTTURATI
WHERE (TIPO="ASSOCIATO")
```

L'operatore aggregato restituisce un solo valore, mentre la prima parte della SELECT restituisce un valore per ogni tupla selezionata!!!

COME FARE? Con interrogazioni annidate (vedi prossimi blocchi ...)

Es. Contare il numero di strutturati (ricercatori + professori) di ciascun dipartimento.

STRUTTURATI

Codice	Nome	Cognome	Tipo	Dipartimento	Stipendio
123	Marco	Marchi	Associato	Chimica	20000
124	Michele	Micheli	Associato	Fisica	20000
125	Lucia	Di Lucia	Ordinario	Fisica	30000
126	Dario	Rossi	Ordinario	Informatica	32000
127	Mario	Rossi	Ricercatore	Informatica	15000
129	Michele	Bianchi	Associato	Fisica	20000

Es. Contare il numero di strutturati (ricercatori + professori) di ciascun dipartimento.

STRUTTURATI

Codice	Nome	Cog	nome	Тіро	Dipartimento	Stipendio
123	Marco	Mar	chi	Associato	Chimica	20000
124	RISULTATO 3	Micl	heli	Associato	Fisica	20000
125	Dipartimer	nto	N	umero	Fisica	30000
126	Chimica			1	Informatica	32000
127	Fisica			3	Informatica	15000
129	Information	a		2	Fisica	20000

Es. Contare il numero di strutturati (ricercatori + professori) di ciascun dipartimento.

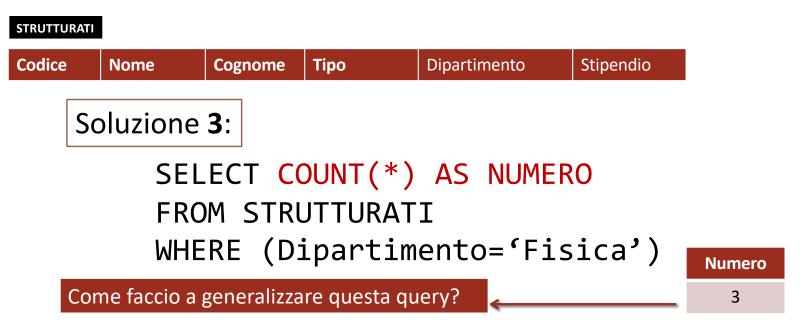


Es. Contare il numero di strutturati (ricercatori + professori) di ciascun dipartimento.



QUERY ERRATA!!!

Es. Contare il numero di strutturati (ricercatori + professori) di ciascun dipartimento.



Operatori di query visti fin qui:

- SELECT ATTRIBUTI FROM WHERE → Valuta i valori di ciascuna riga in isolamento.
- SELECT OP(ATTRIBUTI) FROM WHERE → Valuta i valori delle righe corrispondenti alle colonne della SELECT in modo aggregato.
- D. Esiste la possibilità di combinare i due approcci?

Operatori di query visti fin qui:

- SELECT ATTRIBUTI FROM WHERE → Valuta i valori di ciascuna riga in isolamento.
- Estrarre informazioni aggregate da
 SE gruppi di righe... i valori delle righe corrispondenti alle colonne della SELECT in modo aggregato.
- D. Esiste la Possibilità di combinare i due approcci?

L'operatore di **raggruppamento** consente di dividere la tabella in **gruppi**, ognuno caratterizzata da un valore comune dell'attributo specificato nell'operatore.

SELECT ListaAttributi1
FROM ListaTabelle
WHERE Condizione
GROUPBY ListaAttributi2

ListaAttributi1 deve essere un sottoinsieme di ListaAttributi2, puo' contenere operatori aggregati!

Ogni gruppo produce una sola riga nel risultato finale!

SELECT DIPARTIMENTO AS DIP, COUNT(*) AS NUMERO FROM STRUTTURATI
GROUPBY DIPARTIMENTO



DIP	Numero
Chimica	1
Fisica	3
Informatica	2

SELECT DIPARTIMENTO AS DIP, COUNT(*) AS NUMERO FROM STRUTTURATI GROUPBY DIPARTIMENTO

STRUTTURATI

STEP 1: Partizionamento della tabella

Codice	Nome	Cognome	Tipo	Dipartimento	Stipendio
123	Marco	Marchi	Associato	Chimica	20000
124	Michele	Micheli	Associato	Fisica	20000
125	Lucia	Di Lucia	Ordinario	Fisica	30000
126	Dario	Rossi	Ordinario	Informatica	32000
127	Mario	Rossi	Ricercatore	Informatica	15000
129	Michele	Bianchi	Associato	Fisica	20000

SELECT DIPARTIMENTO AS DIP, COUNT(*) AS NUMERO

FROM STRUTTURATI

GROUPBY DIPARTIMENTO

STEP 1: Partizionamento della tabella

Codice	Nome	Cognome	Tipo	Dipartimento	Stipendio
123	Marco	Marchi	Associato	Chimica	20000
Codice	Nome	Cognome	Tipo	Dipartimento	Stipendio
124	Michele	Micheli	Associato	Fisica	20000
125	Lucia	Di Lucia	Ordinario	Fisica	30000
129	Michele	Bianchi	Associato	Fisica	20000
Codice	Nome	Cognome	Тіро	Dipartimento	Stipendio
126	Dario	Rossi	Ordinario	Informatica	32000
127	Mario	Rossi	Ricercatore	Informatica	15000

PROF. MARCO DI FELICE – CORSO DI LAUREA IN INFORMATICA PER IL MANAGEMENT

SELECT DIPARTIMENTO AS DIP, COUNT(*) AS NUMERO

FROM STRUTTURATI
GROUPBY DIPARTIMENTO

STEP 2: Si applica la select su ciascun gruppo

Codice	Nome	C Dipartime	ento	Nun	nero	ento	Stipendio
123	Marco	№ Chimica		1			20000
Codice	Nome	Cognome	Tipo		Dipartim	ento	Stipendio
124	Michele	M		Nun	nero		20000
125	Lucia	D Fisica		3			30000
129	Michele	Bianchi	Associato	_	Fisica		20000
Codice	Nome	Cognome	Tipo		Dipartim	ento	Stipendio
126	Dario	R _i Dipartime	R _I Dipartimento		nero	ica	32000
127	Mario	R Informati	ca	2		ica	15000

SELECT **DIPARTIMENTO AS DIP, COUNT(*) AS NUMERO**FROM STRUTTURATI
GROUPBY DIPARTIMENTO

STEP 3: Si costruisce il risultato finale



Dip	Numero
Chimica	1
Fisica	3
Informatica	2

Es. Calcolare, per ogni dipartimento, lo stipendio medio degli strutturati.

STRUTTURATI

Codice	Nome	Cognome	Tipo	Dipartimento	Stipendio
123	Marco	Marchi	Associato	Chimica	20000
124	Michele	Micheli	Associato	Fisica	20000
125	Lucia	Di Lucia	Ordinario	Fisica	30000
126	Dario	Rossi	Ordinario	Informatica	32000
127	Mario	Rossi	Ricercatore	Informatica	15000
129	Michele	Bianchi	Associato	Fisica	20000

SELECT DIPARTIMENTO AS DIP, AVG(STIPENDIO) AS

STIPENDIOMEDIO

FROM STRUTTURATI
GROUPBY DIPARTIMENTO



Dip	StipendioMedio
Chimica	20000
Fisica	23333
Informatica	23500

Attenzione! Nella SELECT possono comparire solo un sottoinsieme degli attributi della clausola GROUPBY oppure operatori aggregati.



Codice	Nome	Cognome	Tipo	Dipartimento	Stipendio
123	Marco	Marchi	Associato	Fisica	20000
124	Michele	Micheli	Associato	Fisica	20000

SELECT NOME, COUNT(*) AS NUMERO FROM STRUTTURATI
GROUPBY DIPARTIMENTO



E' possibile **filtrare** i gruppi in base a determinate condizioni, attraverso il costrutto having.

SELECT ListaAttributi1

•••

GROUPBY ListaAttributi2

HAVING Condizione

- clausola where → valutata riga per riga.
- clausola having → valutata su ciascun gruppo, contiene operatori aggregati o condizioni su ListaAttributi2.

Sintassi Generale:

SELECT ListaAttributi1

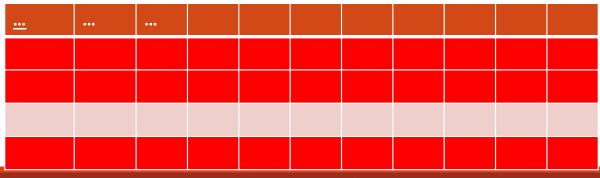
FROM ListaTabelle

WHERE Condizione

GROUPBY ListaAttributi2

HAVING Condizione

STEP 0: Prodotto cartesiano delle tabelle+ Estrazione delle righe che rispettanola condizione della clausola WHERE



Sintassi Generale:

SELECT ListaAttributi1
FROM ListaTabelle

WHERE Condizione

GROUPBY ListaAtţributi2

HAVING Condizione

STEP 1: Partizionamento della tabella



Sintassi Generale:

```
SELECT ListaAttributi1
FROM ListaTabelle
WHERE Condizione
GROUPBY ListaAttributi2
```

HAVING Condizione

STEP 2: Selezione dei gruppi



Sintassi Generale:

SELECT ListaAttributi1

FROM Lista Tabelle

WHERE Condizione

GROUPBY ListaAttributi2

HAVING Condizione



STEP 3: Selezione dei valori delle colonne o esecuzione degli operatori aggregati su ciascuno dei gruppi, e composizione della tabella finale.

Es. Estrarre il nome dei dipartimenti che hanno almeno due strutturati nel suo organico.

STRUTTURATI

Codice	Nome	Cognome	Tipo	Dipartimento	Stipendio
123	Marco	Marchi	Associato	Chimica	20000
124	Michele	Micheli	Associato	Fisica	20000
125	Lucia	Di Lucia	Ordinario	Fisica	30000
126	Dario	Rossi	Ordinario	Informatica	32000
127	Mario	Rossi	Ricercatore	Informatica	15000
129	Michele	Bianchi	Associato	Fisica	20000

SELECT DIPARTIMENTO AS DIP FROM STRUTTURATI GROUPBY DIPARTIMENTO HAVING COUNT(*) >= 2



DIP

Fisica

Informatica

Costrutto select nella sua forma più generale.

```
SELECT ListaAttributi
FROM ListaTabelle
WHERE Condizione
```

LIMIT Number

GROUPBY AttributiRaggruppamento
HAVING CondizioniGruppi
ORDERBY ListaAttributiOrdinamento

In SQL, è possibile effettuare **operazioni insiemistiche** tra tabelle o in generale tra risultati di SELECT:

- O UNION [ALL]
- O INTERSECT [ALL]
- O EXCEPT [ALL]

Gli attributi della SELECT devono avere tipi di dato compatibili e (possibilmente) gli stessi nomi.

Es. Estrarre nome e cognome di tutto il personale universitario (strutturati + tecnici).

STRUTTURATI

Codice	Nome	Cognome	Ruolo
123	Marco	Marchi	Associato
124	Michele	Micheli	Ordinario
125	Lucia	Di Lucia	Ricercatore
126	Dario	Rossi	Ordinario
127	Mario	Rossi	Ordinario
129	Michele	Bianchi	Associato

TECNICI

Codice	Nome	Cognome	Livello
445	Michele	Marini	5
356	Daniele	Marini	6
154	Giovanna	Bianchi	5
156	Lucia	Di Lucia	4

Es. Estrarre nome e cognome di tutto il personale universitario (strutturati + tecnici).

SELECT NOME, COGNOME FROM STRUTTURATI UNION

SELECT NOME, COGNOME FROM TECNICI

Nome	Cognome
Marco	Marchi
Michele	Micheli
Lucia	Di Lucia
Dario	Rossi
Mario	Rossi
Michele	Bianchi
Michele	Marini
	•••

Es. Estrarre nome e cognome degli strutturati che hanno degli omonimi che lavorano come tecnici.

SELECT NOME, COGNOME FROM STRUTTURATI
INTERSECT
SELECT NOME, COGNOME FROM TECNICI

Nome	Cognome
Lucia	Di Lucia

Es. Estrarre nome e cognome degli strutturati che NON hanno degli omonimi che lavorano come tecnici ...

SELECT NOME, COGNOME FROM STRUTTURATI

EXCEPT

SELECT NOME, COGNOME FROM TECNICI

Nome	Cognome
Marco	Marchi
Michele	Micheli
Dario	Rossi
Mario	Rossi
Michele	Bianchi
Michele	Marini

Attenzione. Gli attributi delle SELECT nelle due tabelle devono avere tipi compatibili ...

SELECT RUOLO
FROM STRUTTURATI
UNION
SELECT LIVELLO
FROM TECNICI

ERRORE!STRUTTURATO.Ruolo e' una **stringa**TECNICI.Livello e' un **intero**.

Oltre ad i comandi di interrogazione, la parte DML definisce anche le operazioni per la **modifca dell'istanza** della base di dati.

- o insert → inserisce una o più righe.
- o delete → cancella una o più righe.
- o update → aggiorna un attributo o più.

E' possibile **inserire una riga** esplicitando i valori degli attributi oppure estraendo le righe da altre tabelle del database.

insert into NomeTabella [ListaAttributi]values (ListaValori)

```
INSERT INTO IMPIEGATI(Codice, Nome, Cognome,
Ufficio) values ('8', 'Vittorio', 'Rossi', 'A')
```

E' possibile **inserire una riga esplicitando** i valori degli attributi oppure estraendo le righe da altre tabelle del database.

insert into NomeTabella [ListaAttributi]values (ListaValori)

```
INSERT INTO IMPIEGATI(Codice, Nome, Cognome)
values('8','Vittorio','Rossi')
```

Ufficio → non specificato, NULL o default

E' possibile **inserire una riga** esplicitando i valori degli attributi oppure estraendo le righe da altre tabelle del database.

```
o insert into NomeTabella SQLSelect
INSERT INTO IMPIEGATI
(Codice,Nome,Cognome,Ufficio) VALUES (
    SELECT *
    FROM IMPIEGATICOMUNE)
```

E' possibile **cancellare** tutte le righe che soddisfano una condizione (cancella tutto se non specificata).

o delete from Tabella where Condizione

```
DELETE FROM IMPIEGATI

DELETE FROM IMPIEGATI WHERE (UFFICIO="A")

DELETE FROM TABELLA WHERE NOME IN (

SELECT NOME FROM IMPIEGATICOMUNE)
```

E' possibile **aggiornare** il contenuto di uno o più attributi di una tabella che rispettano una certa condizione.

```
o update NomeTabella
set attributo = expr|SELECT|null|default
[where Condizione]

UPDATE IMPIEGATI
SET NOME="Mario"
WHERE (CODICE=5)
```

E' possibile **aggiornare** il contenuto di uno o più attributi di una tabella che rispettano una certa condizione.

- O UPDATE IMPIEGATI SET NOME='MARCO' WHERE
 (CODICE=5)
- UPDATE IMPIEGATI SET NOME=(SELECT NOME FROM IMPIEGATICOMUNE WHERE CODICE=5) WHERE (CODICE=5)

E' possibile implementare il **join** tra tabelle in due modi distinti (ma equivalenti nel risultato):

- Inserendo le condizioni del JOIN direttamente nella clausola del WHERE
- Attraverso l'utilizzo dell'operatore di inner JOIN nella clausola FROM

```
SELECT ListaAttributi
FROM Tabella JOIN Tabella ON CondizioneJoin
[WHERE Condizione]
```

• • •

Esempio di query con utilizzo dell'INNER join.

GUIDATORI

<u>NrPatente</u>	Nome	Cognome
1243242	Sara	Bianchi
2656565	Michele	Rossi

SELECT Modello
FROM GUIDATORI, VEICOLI
WHERE (GUIDATORI.NrPatente=
 VEICOLI.NrPatente) AND
 (Nome="Sara")

VEICOLI

<u>Targa</u>	Modello	NrPatente
BO2121	Panda	1243242
BO4567	Punto	1243242
BO4896	Yaris	5687876

SELECT Modello
FROM GUIDATORI JOIN VEICOLI
ON GUIDATORI.NrPatente
=VEICOLI.NrPatente
WHERE (Nome="Sara")

Esistono altre **tre** varianti (poco usate) dell'operatore di JOIN

 ○ LEFT join → risultato dell'inner join + righe della tabella di sinistra che non hanno un corrispettivo a destra (completate con valori NULL)

```
SELECT ListaAttributi
FROM Tabella LEFT JOIN Tabella ON CondizioneJoin
[WHERE Condizione]
```

Esempio di query con utilizzo del LEFT join.

GU	IDATC	DRI

<u>NrPatente</u>	Nome	Cognome
1243242	Sara	Bianchi
2656565	Michele	Rossi

VEICOLI

<u>Targa</u>	Modello	NrPatente
BO2121	Panda	1243242
BO4567	Punto	1243242
BO4896	Yaris	5687876

SELECT Modello
FROM GUIDATORI LEFT JOIN VEICOLI ON
GUIDATORI.NrPatente =VEICOLI.NrPatente

Esempio di query con utilizzo del LEFT join.

GUIDATORI

<u>NrPatente</u>	Nome	Cognome
1243242	Sara	Bianchi
2656565	Michele	Rossi

VEICOLI

<u>Targa</u>	Modello	NrPatente
BO2121	Panda	1243242
BO4567	Punto	1243242
BO4567	Yaris	5687876

NrPatente	Nome	Cognome	Targa	Modello	NrPatente
1243242	Sara	Bianchi	BO2121	Panda	1243242
1243242	Sara	Bianchi	BO4567	Panda	1243242
2656565	Michele	Rossi	NULL	NULL	NULL

Esistono altre **tre** varianti (poco usate) dell'operatore di

 ○ RIGHT join → risultato dell'inner join + righe della tabella di destra che non hanno un corrispettivo a destra (completate con valori NULL)

```
SELECT ListaAttributi
FROM Tabella RIGHT JOIN Tabella ON CondizioneJoin
[WHERE Condizione]
```

• • •

Esempio di query con utilizzo del RIGHT join.

GU	ID	ΔΤ	o	RI
)	4		7	

<u>NrPatente</u>	Nome	Cognome	
1243242	Sara	Bianchi	
2656565	Michele	Rossi	

VEICOLI

<u>Targa</u>	Modello	NrPatente
BO2121	Panda	1243242
BO4567	Punto	1243242
BO4896	Yaris	5687876

NrPatente	Nome	Cognome	Targa	Modello	NrPatente
1243242	Sara	Bianchi	BO2121	Panda	1243242
1243242	Sara	Bianchi	BO4567	Panda	1243242
NULL	NULL	NULL	BO4896	Yaris	5687876

Esistono altre **tre** varianti (poco usate) dell'operatore di

 FULL join → risultato dell'inner join + righe della tabella di sinistra/destra che non hanno un corrispettivo a destra/sinistra (completate con valori NULL)

```
SELECT ListaAttributi
FROM Tabella FULL JOIN Tabella ON CondizioneJoin
[WHERE Condizione]
```

Esempio di query con utilizzo del FULL join.

GUIDATORI

<u>NrPatente</u>	Nome	Cognome
1243242	Sara	Bianchi
2656565	Michele	Rossi

VEICOLI

<u>Targa</u>	Modello	NrPatente
BO2121	Panda	1243242
BO4567	Punto	1243242
BO4896	Yaris	5687876

NrPatente	Nome	Cognome	Targa	Modello	NrPatente
1243242	Sara	Bianchi	BO2121	Panda	1243242
1243242	Sara	Bianchi	BO4567	Panda	1243242
NULL	NULL	NULL	BO4896	Yaris	5687876
2656565	Michele	Rossi	NULL	NULL	NULL