

Fondamenti di Informatica e Basi di Dati a.a. 2019/2020

DOCENTE: DOTT.SSA VALERIA FIONDA

BASATE SUL MATERIALE DEL PROF. **MARCO DI FELICE**

Linguaggi per DBMS

SQL



Il Linguaggio SQL

Due **componenti** principali:

➤ **DDL** (*Data Definition Language*)

Contiene i costrutti necessari per la creazione/modifica dello **schema** della base di dati.

➤ **DML** (*Data Manipulation Language*)

Contiene i costrutti per le interrogazioni e di inserimento/eliminazione/modifica di **dati**.

SQL: DML

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

<u>Matricola</u>	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

SQL: DML

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

```
select  Attributo1, ... AttributoM  
from    Tabella1, ... ,TabellaN  
where   Condizione
```

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**.

SQL: DML

Nel caso di una sola tabella:

select Attributo_i, Attributo_j, ... Attributo_m
from Tabella
where Condizione

STEP1: Si selezionano le ennuple della tabella che soddisfano la condizione ...

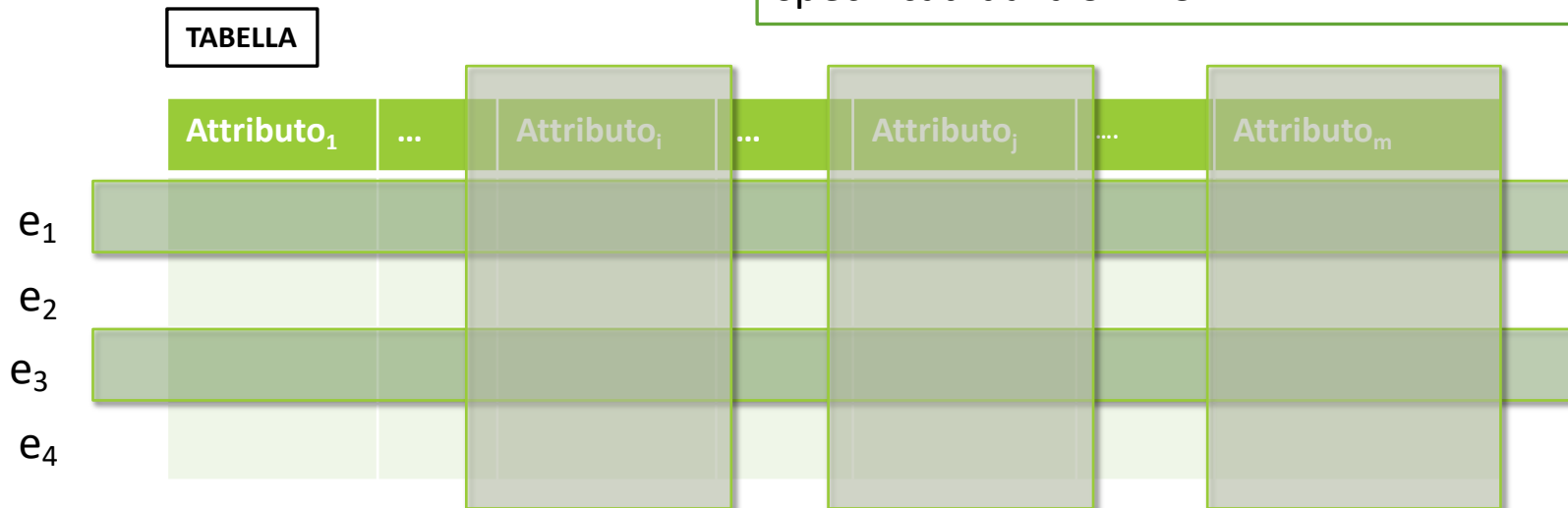
	TABELLA						
	Attributo ₁	...	Attributo _i	...	Attributo _j	...	Attributo _m
e ₁							
e ₂							
e ₃							
e ₄							

SQL: DML

Nel caso di una sola tabella:

select Attributo_i, Attributo_j, ... Attributo_m
from Tabella
where Condizione

STEP2: Si selezionano le colonne/attributi specificati dalla SELECT ...



SQL: DML

Nel caso di una sola tabella:

select *Attributo_i, Attributo_j, ... Attributo_m*
from *Tabella*
where *Condizione*

STEP3: Si costruisce la tabella risultato ...

Numero di colonne definito dalla clausola **SELECT**



Numero di righe
definito dalla
clausola **WHERE**



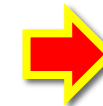
Attributo ₁	Attributo _i	Attributo _m

SQL: DML

Esempio1. Selezionare i nomi degli impiegati che lavorano nell'ufficio A.

IMPIEGATI

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



Nome
Marco
Antonio
Giorgio

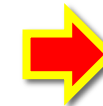
```
SELECT NOME  
FROM IMPIEGATI  
WHERE (UFFICIO="A")
```

SQL: DML

Esempio2. Selezionare i nomi degli impiegati che guadagnano più di 20000 euro annui.

IMPIEGATI

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



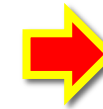
Nome
Antonio
Giorgio

```
SELECT NOME  
FROM IMPIEGATI  
WHERE (STIPENDIO>20000)
```

SQL: DML

Esempio3. 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	A	15000
125	Michele	Monti	B	18000
134	Antonio	Verdi	A	25000
156	Giorgio	Rossi	A	32000



Nome	Cognome
------	---------

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

SQL: DML

La clausola **where** specifica quali **righe delle tabelle 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"))
```

SQL: DML

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'.

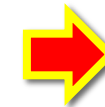
```
SELECT CODICE  
FROM IMPIEGATI  
WHERE (NOME LIKE 'M_R%O')
```

SQL: DML

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	A	15000
125	Michele	Monti	B	18000
134	Antonio	Verdi	A	25000
156	Giorgio	Rossi	A	32000



Nome
Antonio
Giorgio

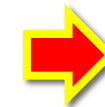
```
SELECT NOME  
FROM IMPIEGATI  
WHERE STIPENDIO BETWEEN(24000,34000)
```

SQL: DML

Q. 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	A	15000
125	Michele	Monti	B	18000
134	Antonio	Verdi	A	NULL
156	Giorgio	Rossi	A	32000



Nome
Giorgio

```
SELECT NOME  
FROM IMPIEGATI  
WHERE (STIPENDIO > 20000)
```

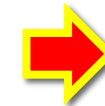
SQL: DML

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	A	15000
125	Michele	Monti	B	18000
134	Antonio	Verdi	A	NULL
156	Giorgio	Rossi	A	32000



Nome
Antonio
Giorgio


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


SQL: DML

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")
```



IMPIEGATI				
Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	A	15000

SQL: DML

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")
```

IMPIEGATI

Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	A	12000
145	Marco	Bianchi	B	24000
167	Lucia	Di Lucia	C	36000



Name	LastName
Marco	Marchi
Marco	Bianchi

SQL: DML

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")
```

IMPIEGATI

Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	A	12000
145	Marco	Bianchi	B	24000
167	Lucia	Di Lucia	C	36000



Name	SalaryM
Marco	1000
Marco	2000

SQL: DML

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 (NOME="MARCO")
```

SQL: DML

Vediamo come funziona la **SELECT** su più tabelle.

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

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

SEDI	
UffNum	Telefono
A	2034333
B	2035434

SQL: DML

```
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	A	12000
145	Marco	Bianchi	B	24000
167	Lucia	Di Lucia	A	36000
187	Giorgio	Rossi	B	12000

SEDI

UffNum	Telefono
A	2034333
B	2035434

SQL: DML

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

STEP1. Si effettua il **prodotto cartesiano** delle due tabelle ...

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

SQL: DML

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

STEP2. Si selezionano le **righe con valori comuni** nelle tue tabelle ...

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

SQL: DML

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

STEP3. Si selezionano le righe relative all'impiegato 145 ...

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

SQL: DML

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

STEP4. Si seleziona la colonna dell'attributo Telefono ...

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

SQL: DML

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



TEL
2035434

STEP5. Si costruisce il risultato finale ...

SQL: DML

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

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

???? ERRORE!!!

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

SEDI	
Ufficio	Telefono
A	2034333
B	2035434

SQL: DML

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)
```

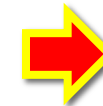
SQL: DML

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

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

IMPIEGATI

Codice	Nome	Cognome	Ufficio	Stipendio
123	Marco	Marchi	A	12000
145	Marco	Bianchi	B	24000
167	Lucia	Di Lucia	C	36000



Name
Marco
Marco

SQL: DML

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)
```



Name
Marco

SQL: DML

Nella clausola **where** possono comparire **più istanze della stessa tabella mediante gli 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

SQL: DML

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

???? ERRORE!!!



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

SQL: DML

```
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

SQL: DML

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

order by Attributo₁ [asc|desc], ...,
Attributo_N [asc|desc]

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

Deve comparire sempre
dopo la clausola where!



SQL: DML

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	A	12000
145	Marco	Bianchi	B	24000
167	Lucia	Di Lucia	A	36000
187	Giorgio	Rossi	B	12000

Da questa colonna dovremmo estrarre un solo valore!

SQL: DML

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 DOPO la clausola `where` e `from`.

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

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

SQL: DML

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)

SQL: DML

Sintassi Generale:

SELECT **OP**(Attributo)
FROM ListaTabelle
WHERE Condizione

sum
max
min
avg
count

count(*)

STEP 0: Si considerano le tabelle indicate nella clausola **FROM**

T ₁		
...
...

T ₂		
...
...

T _N		
...
...

SQL: DML

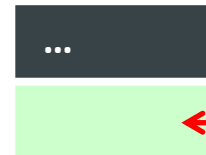
Sintassi Generale:

SELECT OP(Attributo)
FROM ListaTabelle
WHERE Condizione

sum
max
min
avg
count

count(*)

STEP 4: Dalla colonna si calcola **un solo valore** come risultato della query



Se non si usa l'operatore AS, la colonna risultato non ha un nome...

SQL: DML

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

SQL: DML

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



Contatore
2

SQL: DML

Es. Contare la somma complessiva degli stipendi degli strutturati *del 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

SQL: DML

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



Totale
70000

SQL: DML

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

SQL: DML

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



MaxStipendio

50000

SQL: DML

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

```
SELECT CODICE, MAX(STIPENDIO)
FROM STRUTTURATI
WHERE (TIPO="ASSOCIATO")
```



ERRORE!

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 dopo ...)