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

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

SQL: DML

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	[illegible]e	Micheli	Associato	Fisica	20000
125				Fisica	30000
126				Informatica	32000
127				Informatica	15000
129				Fisica	20000

RISULTATO

Dipartimento	Numero
Chimica	1
Fisica	3
Informatica	2

SQL: DML

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

STRUTTURATI

Codice	Nome	Cognome	Tipo	Dipartimento	Stipendio

Soluzione **1**:

```
SELECT COUNT(*) AS NUMERO
FROM STRUTTURATI
```

NON FA QUANTO RICHIESTO!!!

Numero
6

SQL: DML

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

STRUTTURATI

Codice	**Nome**	**Cognome**	**Tipo**	Dipartimento	Stipendio

Soluzione **2**:

```
SELECT COUNT(*) AS NUMERO,DIPARTIMENTO
FROM STRUTTURATI
```

QUERY ERRATA!!!

SQL: DML

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

STRUTTURATI

Codice	Nome	Cognome	Tipo	Dipartimento	Stipendio

Soluzione **3**:

```
SELECT COUNT(*) AS NUMERO
FROM STRUTTURATI
WHERE (Dipartimento='Fisica')
```

Numero
3

Come faccio a generalizzare questa query?

SQL: DML

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**.

Q. Possibilità di **combinare i due approcci**?

SQL: DML

Operatori di query visti fin qui:

- `SELECT ATTRIBUTI FROM WHERE` →
  Valuta i valori di ciascuna riga **in isolamento**.

- `SELECT OP(ATTRIBUTI) FROM WHERE` →
  Va…ti alle
  co…

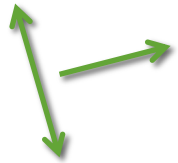
Estrarre informazioni aggregate da gruppi di righe…

Q. Possibilità di **combinare i due approcci**?

SQL: DML

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!

SQL: DML

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

DIP	Numero
Chimica	1
Fisica	3
Informatica	2

SQL: DML

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

STEP1: Partizionamento della tabella

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

SQL: DML

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

STEP1: 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	Tipo	Dipartimento	Stipendio
126	Dario	Rossi	Ordinario	**Informatica**	32000
127	Mario	Rossi	Ricercatore	**Informatica**	15000

SQL: DML

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

STEP2: Si applica la select su ciascun gruppo

Codice	Nome	C	ento	Stipendio
123	Marco	M		20000

Dipartimento	Numero
Chimica	1

Codice	Nome	Cognome	Tipo	Dipartimento	Stipendio
124	Michele	M			20000
125	Lucia	D			30000
129	Michele	Bianchi	Associato	**Fisica**	20000

Dipartimento	Numero
Fisica	3

Codice	Nome	Cognome	Tipo	Dipartimento	Stipendio
126	Dario	R		ica	32000
127	Mario	R		ica	15000

Dipartimento	Numero
Informatica	2

SQL: DML

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

STEP3: Si costruisce il risultato finale

Dip	Numero
Chimica	1
Fisica	3
Informatica	2

SQL: DML

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

SQL: DML

```
SELECT DIPARTIMENTO AS DIP, AVG(STIPENDIO) AS
STIPENDIOMEDIO
FROM STRUTTURATI
GROUPBY DIPARTIMENTO
```

STEP3: Si costruisce il risultato finale

Dip	StipendioMedio
Chimica	20000
Fisica	23333
Informatica	23500

SQL: DML

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

Come faccio ad estrarre 1 sola riga??

STRUTTURATI

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
```

ERRORE!

SQL: DML

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

```
SELECT ListaAttributi1
…
GROUPBY ListaAttributi2
HAVING Condizione
```

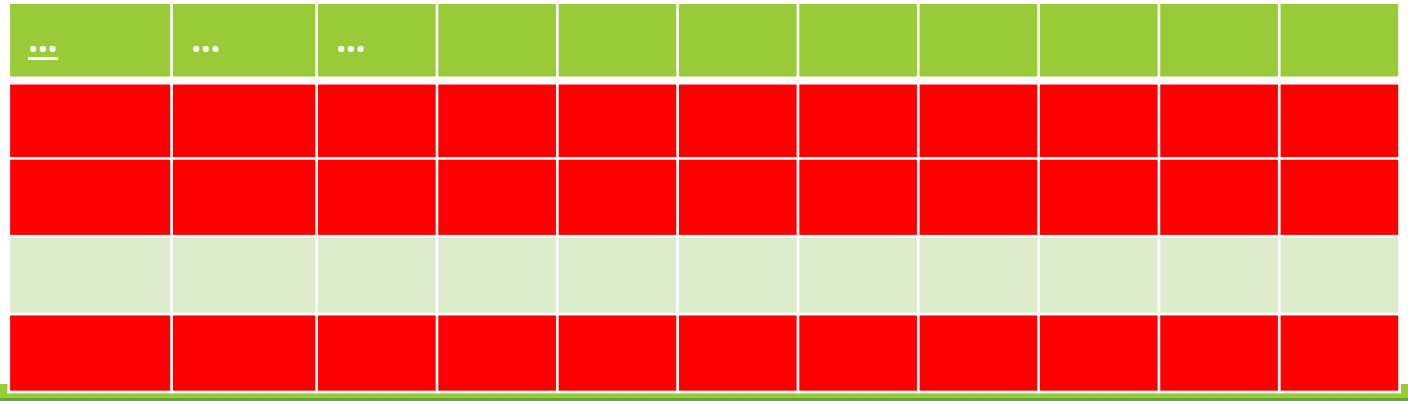
- clausola `where` → valutata riga per riga.
- clausola `having` → valutata su ciascun gruppo, contiene operatori aggregati o condizioni su `ListaAttributi2`.

SQL: DML

Sintassi Generale:

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

STEP0: Prodotto cartesiano delle tabelle + Estrazione delle righe che rispettano la condizione della clausola `WHERE`

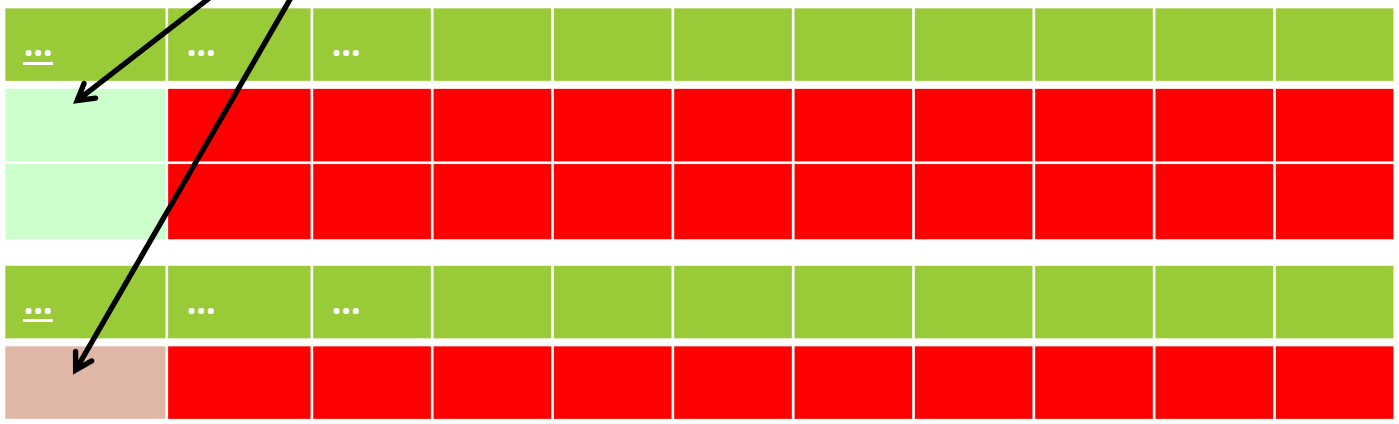


SQL: DML

Sintassi Generale:

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

STEP1: Partizionamento della tabella



SQL: DML

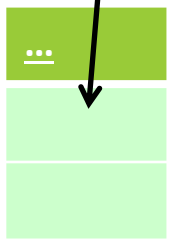
Sintassi Generale:

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

SQL: DML

Sintassi Generale:

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



STEP3: Selezione dei valori delle colonne o esecuzione degli operatori aggregati su ciascuno dei gruppi, e composizione della tabella finale.

SQL: DML

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

SQL: DML

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

DIP
Fisica
Informatica

SQL: DML

Costrutto `select` nella sua forma più generale.

```
SELECT ListaAttributi
FROM ListaTabelle
WHERE Condizione
LIMIT Number
GROUPBY  AttributiRaggruppamento
HAVING CondizioniGruppi
ORDERBY ListaAttributiOrdinamento
```

SQL: DML

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

- `UNION [ALL]`
- `INTERSECT [ALL]`
- `EXCEPT [ALL]`

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

SQL: DML

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

SQL: DML

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
…	…

SQL: DML

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

SQL: DML

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

SQL: DML

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**.

SQL: DML

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]
…
```

SQL: DML

Esempio di **query con utilizzo dell'inner join**.

GUIDATORI

NrPatente	Nome	Cognome
1243242	Sara	Bianchi
2656565	Michele	Rossi

VEICOLI

Targa	Modello	NrPatente
BO2121	Panda	1243242
BO4567	Punto	1243242
BO4896	Yaris	5687876

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

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

SQL: DML

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]
…
```

SQL: DML

Esempio di **query con utilizzo del left join**.

GUIDATORI

NrPatente	Nome	Cognome
1243242	Sara	Bianchi
2656565	Michele	Rossi

VEICOLI

Targa	Modello	NrPatente
BO2121	Panda	1243242
BO4567	Punto	1243242
BO4896	Yaris	5687876

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

SQL: DML

Esempio di **query con utilizzo del left join**.

GUIDATORI

NrPatente	Nome	Cognome
1243242	Sara	Bianchi
2656565	Michele	Rossi

VEICOLI

Targa	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

SQL: DML

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

- **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]
…
```

SQL: DML

Esempio di **query con utilizzo del right join**.

GUIDATORI

NrPatente	Nome	Cognome
1243242	Sara	Bianchi
2656565	Michele	Rossi

VEICOLI

Targa	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

SQL: DML

Es. Estrarre il codice dello strutturato che riceve lo stipendio più alto.

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	35000
127	Mario	Rossi	Ricercatore	Informatica	15000
129	Michele	Bianchi	Associato	Fisica	20000

SQL: DML

Es. Estrarre il codice dello strutturato che riceve lo stipendio più alto.

```
SELECT CODICE, MAX(STIPENDIO)
FROM STRUTTURATI
```

ERRORE!

- `SELECT MAX(STIPENDIO)` restituisce solo un valore!!
- `SELECT CODICE` restituisce più di un valore!!

SQL: DML

Es. Estrarre il codice dello strutturato che riceve lo stipendio più alto.

```
SELECT CODICE
FROM STRUTTURATI
WHERE STIPENDIO= MAX(STIPENDIO)
```

ERRORE!

- L'operatore aggregato MAX si applica sulla SELECT e viene valutato dopo la WHERE …

SQL: DML

Nella clausola `where`, oltre ad espressioni semplici, possono comparire espressioni complesse in cui il valore di un attributo viene confrontato con il risultato di un'altra query (**query annidate**).

NOTA: Si sta confrontando un singolo valore con il risultato di una query (quindi potenzialmente una tabella).

```
SELECT
FROM
WHERE (Attributo expr SELECT
                FROM
                WHERE)
```

SQL: DML

Es. Estrarre il codice dello strutturato che riceve lo stipendio più alto.

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	35000
127	Mario	Rossi	Ricercatore	Informatica	15000
129	Michele	Bianchi	Associato	Fisica	20000

SQL: DML

Es. Estrarre il codice dello strutturato che riceve lo stipendio più alto.

QUERY **ESTERNA**

```
SELECT CODICE
FROM STRUTTURATI
WHERE (STIPENDIO = SELECT MAX(STIPENDIO)
                   FROM STRUTTURATI)
```

QUERY **INTERNA**

CODICE
126

SQL: DML

Nel caso precedente, la query interna restituisce solo un valore … Cosa accade se la query interna restuisce più di un valore?

Gli operatori di confronto <,=,> non si possono utilizzare in questo caso !

Es. Estrarre nome e cognome degli strutturati del dipartimento di Informatica che guadagnano quanto un loro collega di Fisica.

SQL: DML

Nel caso precedente, la query interna restituisce solo un valore … Cosa accade se la query interna restuisce più di un valore?

```
SELECT NOME, COGNOME
FROM STRUTTURATI
WHERE (DIPARTIMENTO="INFORMATICA") AND
    (STIPENDIO = (SELECT STIPENDIO
            FROM STRUTTURATI
            WHERE (DIPARTIMENTO="FISICA")))
```

NON FUNZIONA!

SQL: DML

Esistono **operatori speciali di confronto** nel caso di interrogazioni annidate:

- `any` → la riga soddisfa la condizione se è vero il confronto tra il valore dell' attributo ed ALMENO UNO dei valori ritornati dalla query annidata.

- `all` → a riga soddisfa la condizione se eè vero il confronto tra il valore dell' attributo e TUTTI i valori ritornati dalla query annidata.

SQL: DML

Il costrutto **`in`** restituisce `true` se un certo valore è contenuto nel risultato di una interrogazione nidificata, **`false`** altrimenti.

Nel caso di più di 1 valore si utilizza il costrutture di tuple.

```
SELECT ListaAttributi
FROM TabellaEsterna
WHERE Valore/i IN SELECT ListaAttributi2
        FROM TabellaInterna
        WHERE Condizione
```

SQL: DML

Il costrutto **exists** restituisce true se l'interrogazione nidificata restituisce un risultato non vuoto (>=1 elemento trovato).

Controlla se il numero di righe della query interna>0

```
SELECT ListaAttributi
FROM TabellaEsterna
WHERE EXISTS SELECT ListaAttributi2
       FROM TabellaInterna
       WHERE Condizione
```

SQL: DML

Es. Estrarre nome e cognome degli strutturati del dipartimento di Informatica che guadagnano **quanto un loro collega** di Fisica.

```
SELECT NOME, COGNOME
FROM STRUTTURATI
WHERE (DIPARTIMENTO="INFORMATICA") AND
    (STIPENDIO = ANY (SELECT STIPENDIO
            FROM STRUTTURATI
            WHERE (DIPARTIMENTO="FISICA")))
```

SQL: DML

Es. Estrarre nome e cognome degli strutturati del dipartimento di Informatica che guadagnano **più di tutti** i loro colleghi di Fisica.

```
SELECT NOME, COGNOME
FROM STRUTTURATI
WHERE (DIPARTIMENTO="INFORMATICA") AND
    (STIPENDIO > ALL (SELECT STIPENDIO
            FROM STRUTTURATI
            WHERE (DIPARTIMENTO="FISICA")))
```

SQL: DML

Le interrogazioni **nidificate** possono essere:

- **Semplici** → non c'è passaggio di binding tra un contesto all'altro. Le interrogazioni vengono valutate dalla più interna alla più esterna.

- **Complesse** → c'è passaggio di binding attraverso variabili condivise tra le varie interrogazioni. In questo caso, le interrogazioni più interne vengono valutate su ogni tupla.

SQL: DML

STEP1: Viene valutata la **query più interna**…

```
SELECT NOME, COGNOME
FROM STRUTTURATI
WHERE (DIPARTIMENTO=“INFORMATICA”) AND
    (STIPENDIO > ALL (SELECT STIPENDIO
            FROM STRUTTURATI
            WHERE (DIPARTIMENTO=“FISICA”)))
```

Stipendio
20000
30000
20000

SQL: DML

STEP2: Viene confrontata ciascuna riga della tabella più esterna con il risultato della query interna …

```
SELECT NOME, COGNOME
FROM STRUTTURATI
WHERE (DIPARTIMENTO="INFORMATICA") AND
    (STIPENDIO > ALL (SELECT STIPENDIO
                      …
```

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	35000
127	Mario	Rossi	Ricercatore	Informatica	15000
129	Michele	Bianchi	Associato	Fisica	20000

Stipendio
20000
30000
20000

SQL: DML

STEP2: Viene confrontata ciascuna riga della tabella più esterna con il risultato della query interna …

```
SELECT NOME, COGNOME
FROM STRUTTURATI
WHERE (DIPARTIMENTO="INFORMATICA") AND
    (STIPENDIO > ALL (SELECT STIPENDIO
                      …
```

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	35000
127	Mario	Rossi	Ricercatore	Informatica	15000
129	Michele	Bianchi	Associato	Fisica	20000

Stipendio
20000
30000
20000

SQL: DML

STEP2: Viene confrontata ciascuna riga della tabella più esterna con il risultato della query interna …

```
SELECT NOME, COGNOME
FROM STRUTTURATI
WHERE (DIPARTIMENTO=“INFORMATICA”) AND
    (STIPENDIO > ALL (SELECT STIPENDIO
            FROM STRUTTURATI
            WHERE (DIPARTIMENTO=“FISICA”)))
```

Nome	Cognome
Dario	Rossi

SQL: DML

Le interrogazioni **nidificate** possono essere:

- **Semplici** → non c'è passaggio di binding tra un contesto all'altro. Le interrogazioni vengono valutate dalla più interna alla più esterna.

- **Complesse** → c'è passaggio di binding attraverso variabili condivise tra le varie interrogazioni. In questo caso, le interrogazioni piu' interne vengono valutate su ogni tupla.

SQL: DML

Es. Estrarre nome/cognome degli impiegati che hanno omonimi (stesso nome/cognome di altri impiegati).

IMPIEGATI

Codice	Nome	Cognome	Ufficio
1	Marco	Marchi	A
2	Dario	Rossi	B
3	Lucia	Di Lucia	C
4	Dario	Rossi	C
5	Mario	Rossi	A
6	Marco	Marchi	B

SQL: DML

Es. Estrarre nome/cognome degli impiegati che hanno omonimi (stesso nome/cognome di altri impiegati).

```
SELECT NOME, COGNOME
FROM IMPIEGATI AS I
WHERE (I.NOME,I.COGNOME) =
    (SELECT NOME, COGNOME
     FROM IMPIEGATI AS I2
   WHERE          (I.NOME=I2.NOME)          AND
(I.COGNOME=I2.COGNOME)   AND   (I.CODICE   <>
I2.CODICE))
```

SQL: DML

Funzionamento: La query più interna viene valutata su ciascuna tupla della query più esterna...

I

Codice	Nome	Cognome	Ufficio
1	**Marco**	**Marchi**	A
2	Dario	Rossi	B
3	Lucia	Di Lucia	C
4	Dario	Rossi	C
5	Mario	Rossi	A
6	Marco	Marchi	B

I2

Codice	Nome	Cognome	Ufficio
1	**Marco**	**Marchi**	A
2	Dario	Rossi	B
3	Lucia	Di Lucia	C
4	Dario	Rossi	C
5	Mario	Rossi	A
6	Marco	Marchi	B

SQL: DML

Funzionamento: La query più interna viene valutata su ciascuna tupla della query più esterna...

I

Codice	Nome	Cognome	Ufficio
1	**Marco**	**Marchi**	A
2	Dario	Rossi	B
3	Lucia	Di Lucia	C
4	Dario	Rossi	C
5	Mario	Rossi	A
6	Marco	Marchi	B

I2

Codice	Nome	Cognome	Ufficio
1	Marco	Marchi	A
2	**Dario**	**Rossi**	**B**
3	Lucia	Di Lucia	C
4	Dario	Rossi	C
5	Mario	Rossi	A
6	Marco	Marchi	B

SQL: DML

In alcuni casi, le **query annidate** possono essere riscritte usando costrutti di join tra tabelle o self-join (prodotto cartesiano + selezione).

```
SELECT NOME, COGNOME
FROM IMPIEGATI AS I,IMPIEGATI AS I2
WHERE (I.NOME=I2.NOME) AND
(I.COGNOME=I2.COGNOME) AND (I.CODICE <>
I2.CODICE))
```

SQL: DML

In maniera equivalente, usando l'operatore `in` ed i **costruttori di tupla**:

```
SELECT CODICE
FROM IMPIEGATI AS I
WHERE (I.NOME,I.COGNOME) NOT IN
          (SELECT NOME, COGNOME
            FROM IMPIEGATI AS I2
           WHERE      (I.NOME=I2.NOME)      AND
(I.COGNOME=I2.COGNOME)    AND    (I.CODICE    <>
I2.CODICE))
```

SQL: DML

Es. Estrarre nome/cognome degli impiegati che **NON** hanno omonimi (stesso nome/cognome di altri impiegati).

IMPIEGATI

Codice	Nome	Cognome	Ufficio
1	Marco	Marchi	A
2	Dario	Rossi	B
3	Lucia	Di Lucia	C
4	Dario	Rossi	C
5	Mario	Rossi	A
6	Marco	Marchi	B

SQL: DML

```
SELECT NOME, COGNOME
FROM IMPIEGATI AS I
WHERE NOT EXISTS (SELECT *
                  FROM IMPIEGATI AS I2
                WHERE    (I.NOME=I2.NOME)    AND
(I.COGNOME=I2.COGNOME) AND (I.CODICE <>
I2.CODICE))
```

Q. E’ possibile scrivere la stessa query senza usare interrogazioni annidate?