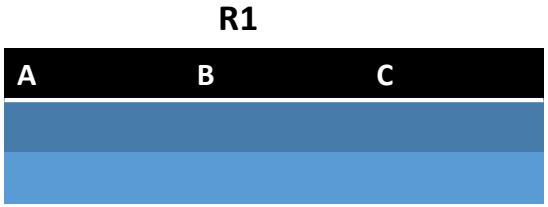
SQL query

VIDEOCONFERENZA 22/04/2020: JOIN

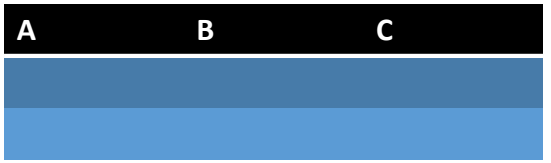
Docente: CHIARA DAMIANI chiara.damiani@unimib.it

L'istruzione SELECT

finora



Istanza della tabella

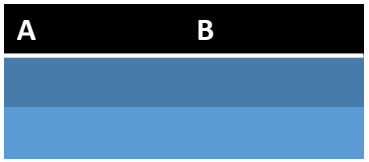


SELECT *
FROM R1

Selezione

SELECT *
FROM R1
WHERE A>B

Proiezione

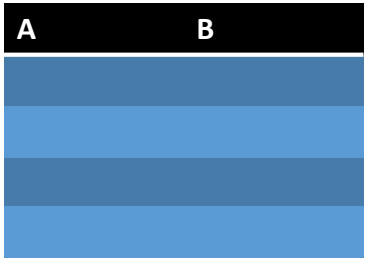


SELECT **A,B**
FROM R1

Il risultato di una select è sempre una tabella

Una tabella è sempre ottenuta con un'istruzione di select

Unione di due tabelle



SELECT A,B
FROM R1
UNION
SELECT B,C
FROM R1

Date la seguente istanza di relazione:

Personale_docente

[Cardinalità 5]

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	14
218301	QUAGLIA	LAURA	Ordinario	10

Poll 1: quale è la cardinalità della relazione prodotta dalla seguente espressione AR?

$$\Pi_{Ruolo}(\textbf{Personale\_docente})$$

Ruolo
Associato
Ordinario
Ricercatore

La relazione prodotta ha cardinalità 3 perché una **Relazione** intesa in senso matematico **non ammette duplicati**

Date la seguente istanza di relazione:

Personale_docente

[Cardinalità 5]

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	14
218301	QUAGLIA	LAURA	Ordinario	10

La relazione prodotta dalla seguente query:

```
SELECT Ruolo
FROM Personale_docente
```

Ruolo
Associato
Ricercatore
Ricercatore
Ordinario
Ordinario

ha cardinalità 5 perché **SQL** (per ragioni di efficienza**) non elimina i duplicati**

Algebra relazionale

- Linguaggio prettamente formale che forma la base per linguaggi ‘reali’.
- Linguaggio procedurale: si specifica l’algoritmo con cui ottenere il risultato.
- Istruzioni equivalenti possono differire in termini di efficienza.
- Relazioni intese in senso matematico => Insiemi di tuple, definite su attributi
- **Negli insiemi non ci possono essere elementi uguali.**

SQL

- Linguaggio più usato per basi di dati relazionali.
- Linguaggio (parzialmente) dichiarativo: si specifica il risultato da ottenere senza preoccuparsi di specificare l’algoritmo.
- Istruzioni equivalenti differiscono solo per leggibilità.
- Relazioni intese come tabelle.
- **Possono esserci righe uguali**

Date la seguente istanza di relazione:

Personale_docente

[Cardinalità 5]

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	14
218301	QUAGLIA	LAURA	Ordinario	10

Esercizio 3: formulare una query SQL che produca i diversi ruoli ricoperti dal personale docente

La clausola **DISTINCT** permette di eliminare i duplicati come in AR

```
SELECT DISTINCT Ruolo
FROM Personale_docente
```

Ruolo
Associato
Ricercatore
Ordinario

Personale_docente(matricola_d,Cognome,Nome,Ruolo, Classe_stipendio)

Stipendio(Classe**,**Valore**)**

Select su più tabelle

ESEMPIO DI THETA-JOIN (EQUI-JOIN):

Selezionare l'ammontare dello stipendio di ciascun docente

```
SELECT Valore
FROM  Personale_docente, Stipendio
WHERE Classe_stipendio=Classe
```

Prodotto cartesiano di Personale_docente e Stipendio

Selezione sul prodotto cartesiano nella clausola WHERE. Il predicato Classe_stipendio= S.Classe è detto **predicato di join**, in quanto stabilisce il criterio con cui le tuple di Stipendio e Personale docente devono essere combinate

Viene eseguito il prodotto cartesiano a prescindere che le due tabelle abbiano o meno attributi in comune.

Vengono combinate solo le tuple che soddisfano il **criterio esplicitato**.

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Personale_docente(matricola_d,Cognome,Nome,Ruolo, Classe_stipendio)
Stipendio(Classe**,**Valore**)**

Esercizio 4: formulare una query SQL che produca il personale docente che guadagna almeno 60.000

QUERY IN AR (Es. 7 - selezione congiuntiva su prodotto cartesiano)

$\sigma_{\text{valore>=60000 AND Classe\_stipendio=classe}}$ **(**Personale_docente ⋈ Stipendio**)**

QUERY IN SQL

?

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Personale_docente(matricola_d,Cognome,Nome,Ruolo, Classe_stipendio)
Stipendio(Classe,Valore)

Esercizio 4: formulare una query SQL che produca il personale docente che guadagna almeno 60.000

QUERY IN AR (Es. 7 - selezione congiuntiva su prodotto cartesiano)

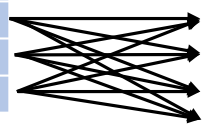
$\sigma_{\text{valore}>=60000 \text{ AND Classe\_stipendio=classe}}$ (Personale_docente ▷◁ Stipendio)

QUERY IN SQL

```
SELECT *
FROM  Personale_docente, Stipendio
WHERE Classe_stipendio=Classe AND Valore>=60.000
```

Personale_docente [Cardinalità 3]

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	6
774002	BUZZI	UMBERTO	Ricercatore	3



Stipendio [Cardinalità 4]

Classe	Valore
3	30000
5	50000
6	60000
14	140000

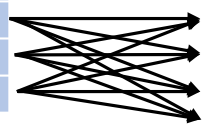
SELECT *
FROM Personale_docente, Stipendio

[Cardinalità 12]

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio	Classe	
485801	BELOTTI	GIOVANNI	Associato	5	3	30000
485801	BELOTTI	GIOVANNI	Associato	5	5	50000
485801	BELOTTI	GIOVANNI	Associato	5	6	60000
485801	BELOTTI	GIOVANNI	Associato	5	14	140000
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	6	3	30000
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	6	5	50000
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	6	6	60000
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	6	14	140000
774002	BUZZI	UMBERTO	Ricercatore	3	3	30000
774002	BUZZI	UMBERTO	Ricercatore	3	5	50000
774002	BUZZI	UMBERTO	Ricercatore	3	6	60000
774002	BUZZI	UMBERTO	Ricercatore	3	14	140000

Personale_docente [Cardinalità 3]

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	6
774002	BUZZI	UMBERTO	Ricercatore	3



Stipendio [Cardinalità 4]

Classe	Valore
3	30000
5	50000
6	60000
14	140000

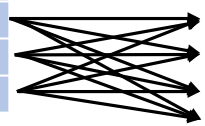
SELECT *
FROM Personale_docente, Stipendio
WHERE
Classe_stipendio=Classe AND Valore>=60.000

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio	Classe	
~~485801~~	~~BELOTTI~~	~~GIOVANNI~~	~~Associato~~	~~5~~	~~3~~	~~30000~~
~~485801~~	~~BELOTTI~~	~~GIOVANNI~~	~~Associato~~	~~5~~	~~5~~	~~50000~~
~~485801~~	~~BELOTTI~~	~~GIOVANNI~~	~~Associato~~	~~5~~	~~6~~	~~60000~~
~~485801~~	~~BELOTTI~~	~~GIOVANNI~~	~~Associato~~	~~5~~	~~14~~	~~140000~~
~~512601~~	~~CAMPIGLIA~~	~~GIUSEPPE~~	~~Ricercatore~~	~~6~~	~~3~~	~~30000~~
~~512601~~	~~CAMPIGLIA~~	~~GIUSEPPE~~	~~Ricercatore~~	~~6~~	~~5~~	~~50000~~
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	6	6	60000
~~512601~~	~~CAMPIGLIA~~	~~GIUSEPPE~~	~~Ricercatore~~	~~6~~	~~14~~	~~140000~~
~~774002~~	~~BUZZI~~	~~UMBERTO~~	~~Ricercatore~~	~~3~~	~~3~~	~~30000~~
~~774002~~	~~BUZZI~~	~~UMBERTO~~	~~Ricercatore~~	~~3~~	~~5~~	~~50000~~
~~774002~~	~~BUZZI~~	~~UMBERTO~~	~~Ricercatore~~	~~3~~	~~6~~	~~60000~~
~~774002~~	~~BUZZI~~	~~UMBERTO~~	~~Ricercatore~~	~~3~~	~~14~~	~~140000~~

[Cardinalità 1]

Personale_docente [Cardinalità 3]

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	6
774002	BUZZI	UMBERTO	Ricercatore	3



Stipendio [Cardinalità 4]

Classe	Valore
3	30000
5	50000
6	60000
14	140000

SELECT *
FROM Personale_docente, Stipendio
WHERE
Classe_stipendio=Classe AND Valore>=60.000

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio	Classe	
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	6	6	60000

[Cardinalità 1]

Dato il seguente schema relazionale:

Personale_docente(matricola_d,Cognome,Nome,Ruolo, Classe_stipendio)
Stipendio(Classe,Valore)

Esercizio 4: formulare una query SQL che produca i ricercatori che guadagna almeno 60000

QUERY IN AR (Es. 8 - selezione congiuntiva su prodotto cartesiano)

$$\sigma_{\text{Ruolo='Ricercatore'}}(\text{Personale\_docente}) \rhd\lhd_{\text{Classe\_stipendio=classe}} \sigma_{\text{valore>=60000}}(\text{Stipendio})$$

QUERY IN SQL

?

Dato il seguente schema relazionale:

Personale_docente(matricola_d,Cognome,Nome,Ruolo, Classe_stipendio)
Stipendio(Classe,Valore)

Esercizio 4: formulare una query SQL che produca i ricercatori che guadagna almeno 60000

QUERY IN AR (Es. 8 - selezione congiuntiva su prodotto cartesiano)

$$\sigma_{\text{Ruolo='Ricercatore'}}(\text{Personale\_docente}) \rhd\lhd_{\text{Classe\_stipendio=classe}} \sigma_{\text{valore>=60000}}(\text{Stipendio})$$

QUERY IN SQL

```
SELECT *
FROM  Personale_docente, Stipendio
WHERE Classe_stipendio=Classe AND Valore>=60000 AND
Ruolo='Ricercatore'
```

DICHIARATIVITA' DELL'SQL, non c'è ragione di anticipare la selezione rispetto al JOIN

Algebra relazionale

- Il **theta join** ha senso solo per relazioni che non hanno attributi in comune perché si combinano sempre le tuple che hanno gli stessi valori per gli attributi comuni

SQL

- Il **theta join** ha sempre senso, perché si combinano solo le tuple che hanno lo stesso valore per gli attributi specificati nella condizione di join

Il **Natural Join** è stato solo recentemente introdotto nei DBMS ma si ottiene con un costrutto dedicato

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Studente(matricola_st,Cognome,Nome, corso_di_Laurea)

NOTA: non possiamo escludere che ci siano matricole ripetute tra le diverse entità

Esercizio 5: Formulare una query SQL che produca il personale non docente che è anche iscritto ad un corso di laurea

ESPERESSIONE AR (Es 3)

$$\Pi_{matricola\_d,Cognome,Nome}(Personale\_non\_docente) \cap \rho_{matricola\_d <-- matricola\_st}(\Pi_{matricola\_st,Cognome,Nome}(Studente))$$

QUERY SQL

```
SELECT Matricola_d  AS Matr, Cognome, Nome
FROM Personale_non_docente
INTERSECT
SELECT Matricola_st, Cognome,Nome
FROM studente
```

NB: l'intersezione e la differenza non sono supportate da mysql

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Studente(matricola_st,Cognome,Nome, corso_di_Laurea)

Esercizio 6: formulare una query SQL che produca il personale non docente che è anche iscritto ad un corso di laurea senza usare l'operatore di INTERSEZIONE né di DIFFERENZA

ESPERESSIONE AR (Es AR.9)

$$\rho_{\text{Matr <-- Matricola\_d}}(\Pi_{\text{Matricola\_d,Cognome,Nome}}(\text{Personale\_non\_docente})) \bowtie \rho_{\text{Matr<-- Matricola\_st}}(\Pi_{\text{Matricola\_st,Cognome,Nome}}(\text{Studente}))$$

QUERY SQL

SELECT ?
FROM ?
WHERE ?

Se due relazioni sono definite sugli stessi attributi il join naturale equivale all'intersezione delle due relazioni.

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Studente(matricola_st,Cognome,Nome, corso_di_Laurea)

Esercizio 6: formulare una query SQL che produca il personale non docente che è anche iscritto ad un corso di laurea senza usare l'operatore di INTERSEZIONE né di DIFFERENZA

ESPERESSIONE AR (Es AR.9)

$$\rho_{\text{Matr <-- Matricola\_d}}(\Pi_{\text{Matricola\_d,Cognome,Nome}}(\text{Personale\_non\_docente})) \bowtie \rho_{\text{Matr<-- Matricola\_st}}(\Pi_{\text{Matricola\_st,Cognome,Nome}}(\text{Studente}))$$

QUERY SQL

```
SELECT ?
FROM ?
WHERE ?
```

IN AR avevamo ridenominato Matricola_st per avere le due relazioni definite sugli stessi attributi
IN SQL possiamo/dobbiamo semplicemente accoppiare tutti gli attributi omologhi nel predicato di join

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Studente(matricola_st,Cognome,Nome, corso_di_Laurea)

Esercizio 6: formulare una query SQL che produca il personale non docente che è anche iscritto ad un corso di laurea senza usare l'operatore di INTERSEZIONE né di DIFFERENZA

ESPERESSIONE AR (Es AR.9)

$$\rho_{\text{Matr <-- Matricola\_d}}(\Pi_{\text{Matricola\_d,Cognome,Nome}}(\text{Personale\_non\_docente})) \bowtie \rho_{\text{Matr<-- Matricola\_st}}(\Pi_{\text{Matricola\_st,Cognome,Nome}}(\text{Studente}))$$

QUERY SQL

SELECT ?
FROM ?
WHERE ?

E' una buona strategia costruire una query partendo dalla clausola FROM

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Studente(matricola_st,Cognome,Nome, corso_di_Laurea)

Esercizio 6: formulare una query SQL che produca il personale non docente che è anche iscritto ad un corso di laurea senza usare l'operatore di INTERSEZIONE né di DIFFERENZA

ESPERESSIONE AR (Es AR.9)

$$\rho_{\text{Matr <-- Matricola\_d}}(\Pi_{\text{Matricola\_d,Cognome,Nome}}(\text{Personale\_non\_docente})) \bowtie \rho_{\text{Matr<-- Matricola\_st}}(\Pi_{\text{Matricola\_st,Cognome,Nome}}(\text{Studente}))$$

QUERY SQL

SELECT ?
FROM Personale_non_docente, Studente
WHERE ?

E' una buona strategia costruire una query partendo dalla clausola FROM

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Studente(matricola_st,Cognome,Nome, corso_di_Laurea)

Esercizio 6: formulare l'espressione in AR che produca il personale non docente che è anche iscritto ad un corso di laurea senza usare l'operatore di INTERSEZIONE né di DIFFERENZA

ESPERESSIONE AR (Es AR.9)

$$\rho_{\text{Matr <-- Matricola\_d}}(\Pi_{\text{Matricola\_d,Cognome,Nome}}(\text{Personale\_non\_docente})) \bowtie \rho_{\text{Matr<-- Matricola\_st}}(\Pi_{\text{Matricola\_st,Cognome,Nome}}(\text{Studente}))$$

QUERY SQL

```
SELECT ?
FROM Personale_non_docente, Studente
WHERE ?
```

Per poi passare alla clausola WHERE

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Studente(matricola_st,Cognome,Nome, corso_di_Laurea)

Esercizio 6: formulare l'espressione in AR che produca il personale non docente che è anche iscritto ad un corso di laurea senza usare l'operatore di INTERSEZIONE né di DIFFERENZA

ESPERESSIONE AR (Es AR.9)

$$\rho_{\text{Matr} \leftarrow \text{Matricola\_d}}(\Pi_{\text{Matricola\_d,Cognome,Nome}}(\text{Personale\_non\_docente})) \bowtie \rho_{\text{Matr} \leftarrow \text{Matricola\_st}}(\Pi_{\text{Matricola\_st,Cognome,Nome}}(\text{Studente}))$$

QUERY SQL

```
SELECT ?
FROM Personale_non_docente, Studente
WHERE Matricola_d=Matricola_st AND ?
```

L'attributo Cognome genera AMBIGUITA'. Ci riferiamo a Cognome in Studente o Cognome in Personale Docente?
Questo problema non si poneva con AR perché attributi comuni erano automaticamente combinati

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Studente(matricola_st,Cognome,Nome, corso_di_Laurea)

Esercizio 6: formulare l'espressione in AR che produca il personale non docente che è anche iscritto ad un corso di laurea senza usare l'operatore di INTERSEZIONE né di DIFFERENZA

ESPERESSIONE AR (Es AR.9)

$$\rho_{\text{Matr <-- Matricola\_d}}(\Pi_{\text{Matricola\_d,Cognome,Nome}}(\text{Personale\_non\_docente})) \triangleright\triangleleft \rho_{\text{Matr<-- Matricola\_st}}(\Pi_{\text{Matricola\_st,Cognome,Nome}}(\text{Studente}))$$

QUERY SQL

```
SELECT ?
FROM Personale_non_docente, Studente
WHERE Matricola_d=Matricola_st AND
Personale_non_docente.Nome=Studente.Nome AND
Personale_non_docente.Cognome=Studente.Cognome
```

La clausola **PUNTO .** permette di specificare a che relazione appartiene un attributo
(o anche una tabella a un database)

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Studente(matricola_st,Cognome,Nome, corso_di_Laurea)

Esercizio 6: formulare l'espressione in AR che produca il personale non docente che è anche iscritto ad un corso di laurea senza usare l'operatore di INTERSEZIONE né di DIFFERENZA

ESPERESSIONE AR (Es AR.9)

$$\rho_{\text{Matr <-- Matricola\_d}}(\Pi_{\text{Matricola\_d,Cognome,Nome}}(\text{Personale\_non\_docente})) \bowtie \rho_{\text{Matr<-- Matricola\_st}}(\Pi_{\text{Matricola\_st,Cognome,Nome}}(\text{Studente}))$$

QUERY SQL

```
SELECT ?
FROM Personale_non_docente, Studente
WHERE Matricola_d=Matricola_st AND
Personale_non_docente.Nome=Studente.Nome and
Personale_non_docente.Cognome=Studente.Cognome
```

E infine pensiamo alla target list

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Studente(matricola_st,Cognome,Nome, corso_di_Laurea)

Esercizio 6: formulare l'espressione in AR che produca il personale non docente che è anche iscritto ad un corso di laurea senza usare l'operatore di INTERSEZIONE né di DIFFERENZA

ESPERESSIONE AR (Es AR.9)

$$\rho_{\text{Matr} \leftarrow \text{Matricola\_d}}(\Pi_{\text{Matricola\_d,Cognome,Nome}}(\text{Personale\_non\_docente})) \bowtie \rho_{\text{Matr} \leftarrow \text{Matricola\_st}}(\Pi_{\text{Matricola\_st,Cognome,Nome}}(\text{Studente}))$$

QUERY SQL

```
SELECT Matricola_d, Personale_non_docente.Nome,
Personale_non_docente.Cognome
FROM Personale_non_docente, Studente
WHERE Matricola_d=Matricola_st AND
Personale_non_docente.Nome=Studente.Nome and
Personale_non_docente.Cognome=Studente.Cognome
```

NB: DICHIARATIVITA' DELL'SQL, non è possibile anticipare la proiezione rispetto al join

```
SELECT Matricola_d, Personale_non_docente.Nome, Personale_non_docente.Cognome
FROM Personale_non_docente, Studente
WHERE Matricola_d=Matricola_st AND Personale_non_docente.Nome=Studente.Nome
AND Personale_non_docente.Cognome=Studente.Cognome
```

- SQL permette (e a volte richiede) di ridenominare non solo le colonne ma anche le tabelle
- In casi come questi è comodo **ridenominare le tabelle** con un alias più breve
- Gli alias esistono solo «per la durata della query»

```
SELECT Matricola_d, A.Nome, A.Cognome
FROM Personale_non_docente AS A, Studente AS B
WHERE Matricola_d=Matricola_st AND A.Nome=B.Nome and A.Cognome=B.Cognome
```

Il costrutto join

Sintassi diversa, semantica equivalente

```
SELECT *
FROM  Personale_docente, Stipendio
WHERE Classe_stipendio=Classe AND
Valore>=60000 AND Ruolo='Ricercatore'
```

```
SELECT *
FROM  Personale_docente JOIN  Stipendio ON
Classe_stipendio=Classe
WHERE Valore>=60000 AND Ruolo='Ricercatore'
```

Si può esplicitare il **predicato di join** direttamente nella clausola FROM utilizzando il costrutto **JOIN ON** anche detto join esplicito

Il costrutto join permette di specificare il tipo di join

```
SELECT *
FROM  Personale_docente INNER JOIN Stipendio ON Classe_stipendio=Classe
```

```
SELECT *
FROM  Personale_docente LEFT OUTER JOIN Stipendio ON Classe_stipendio=Classe
```

```
SELECT *
FROM  Personale_docente RIGHT OUTER JOIN Stipendio ON Classe_stipendio=Classe
```

```
SELECT *
FROM  Personale_docente FULL OUTER JOIN Stipendio ON Classe_stipendio=Classe
```

```
SELECT *
FROM  Personale_docente NATURAL JOIN Stipendio
```

Il signficato è identico a quello dell'algebra relazionale

Dato il seguente schema relazionale:

Personale_non_docente(Matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Personale_docente(Matricola_d,Cognome,Nome,Ruolo, Classe_stipendio)
Studente(Matricola_st,Cognome,Nome, corso_di_Laurea)

Esercizio 7: formulare la query SQL che produca il personale non docente che NON è iscritto a nessun corso di laurea

AR (Es 4)

Operatore di **DIFFERENZA**

$$\rho_{\text{Matr} \leftarrow \text{Matricola\_d}}(\Pi_{\text{matricola\_d,Cognome,Nome}}(\text{Personale\_non\_docente})) - \rho_{\text{Matr} \leftarrow \text{Matricola\_st}}(\Pi_{\text{matricola\_st,Cognome,Nome}}(\text{Studente}))$$

SQL

```
SELECT Matricola_d  AS Matr
FROM Personale_non_docente
MINUS
SELECT Matricola_st
FROM studente
```

Dato il seguente schema relazionale:

Personale_non_docente(matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Personale_docente(matricola_d,Cognome,Nome,Ruolo, Classe_stipendio)
Stipendio(Classe,Valore**)**

Esercizio 8: formulare la query SQL che produca le classi di stipendio che non sono attribuite a nessun personale docente – senza utilizzare l'operatore Differenza

AR (Es 7)

Operatore di **IS NULL**

Π_{Classe} ($\sigma_{Matricola\_d\ \textbf{IS NULL}}$ (Personale_docente $\triangleright\triangleleft RIGHT_{Classe\_stipendio=Classe}$ Stipendio))

SQL

```
SELECT Classe
FROM Personale_docente RIGHT OUTER JOIN Stipendio ON Classe_stipendio=Classe
WHERE Matricola_d IS NULL
```

La seguente espressione AR produce i Ricercatori che hanno la stessa classe di stipendio di professori ordinari (es 9 AR)

$$\sigma_{\text{ruolo=ricercatore}}(\text{Personale\_docente}) \bowtie \Pi_{\text{classe\_stipendio}}(\sigma_{\text{ruolo=ordinario}}(\text{Personale\_docente}))$$

Esercizio 9: formulare la corrispondente query SQL

In caso di Self Join l'ALIAS è obbligatorio

```
SELECT A.*
FROM Personale_Docente AS A, Personale_Docente  AS B
WHERE A.Ruolo='Ricercatore' AND B.Ruolo='Ordinario' AND
A.Classe_stipendio=B.Classe_stipendio
```

Il self join

Si effettua un join di una tabella con (una copia di) se stessa per:

- **Confrontare tuple** di una relazione tra loro
- Trovare elementi duplicati
- Quando una tupla fa **riferimento** a dati nella stessa tabella. Esempi tipici:
  - **Relazioni gerarchiche** (genitore-figlio, dipendente-capo)
  - **Relazioni sequenziali**
  - **Relazioni a grafo**

Assumendo che i docenti possano essere supervisionati da altri docenti

Supervisione

Dipendente	Supervisore
485801	512601
512601	774002
774002	512601
94302	774002
218301	512601

Personale_docente

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	5
218301	QUAGLIA	LAURA	Ordinario	10

Esercizio 10: formulare la query SQL per trovare le matricole dei docenti che hanno la stessa classe di stipendio dei loro sottoposti

Poll 2: Quali matricole hanno la stessa classe di stipendio dei loro sottoposti?

RISOLVIAMO COME PRIMA COSA LA QUERY A MENTE

Assumendo che i docenti possano essere supervisionati da altri docenti

Supervisione

Dipendente	Supervisore	Classe_dip
485801	512601	5
512601	774002	10
774002	512601	3
94302	774002	5
218301	512601	10

Personale_docente

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	5
218301	QUAGLIA	LAURA	Ordinario	10

Esercizio 10: formulare la query SQL per trovare le matricole dei docenti che hanno la stessa classe di stipendio dei loro sottoposti

Poll 2: Quali matricole hanno la stessa classe di stipendio dei loro sottoposti?

Cerchiamo la corrispondenza di valori tra Supervisione.Dipendente e Personale_docente.matricola_d per sapere la classe di stipendio del dipendente

Assumendo che i docenti possano essere supervisionati da altri docenti

Supervisione

Dipendente	Supervisore	Classe_dip	Classe_sup
485801	512601	5	10
512601	774002	10	3
774002	512601	3	10
94302	774002	14	3
218301	512601	10	10

Personale_docente

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	5
218301	QUAGLIA	LAURA	Ordinario	10

Esercizio 10: formulare la query SQL per trovare le matricole dei docenti che hanno la stessa classe di stipendio dei loro sottoposti

Poll 2: Quali matricole hanno la stessa classe di stipendio dei loro sottoposti?

Cerchiamo la corrispondenza tra Supervisione.Supervisore e Supervisione.Matricola_d per sapere la classe di stipendio del supervisore

La Matricola **512601** (Campiglia Giuseppe) ha la stessa classe di stipendio della sua sottoposta (Quaglia Laura Matr. **218301**)

Assumendo che i docenti possano essere supervisionati da altri docenti

Supervisione

Dipendente	Supervisore
485801	512601
512601	774002
774002	512601
94302	774002
218301	512601

Personale_docente

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	5
218301	QUAGLIA	LAURA	Ordinario	10

Esercizio 10: formulare la query SQL per trovare le matricole dei docenti che hanno la stessa classe di stipendio dei loro sottoposti.

```
SELECT ?
FROM ?
WHERE ?
```

CLAUSOLA FROM: Quali tabelle sono coinvolte?

Assumendo che i docenti possano essere supervisionati da altri docenti

Supervisione

Dipendente	Supervisore
485801	512601
512601	774002
774002	512601
94302	774002
218301	512601

Personale_docente

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	5
218301	QUAGLIA	LAURA	Ordinario	10

Esercizio 10: formulare la query SQL per trovare le matricole dei docenti che hanno la stessa classe di stipendio dei loro sottoposti.

```
SELECT ?
FROM Supervisione, Personale_docente AS Dipendente, Personale_docente AS Supervisore
WHERE ?
```

CLAUSOLA WHERE

Step 1: correliamo le informazioni sui dipendenti con l'informazione sul personale_docente

Assumendo che i docenti possano essere supervisionati da altri docenti

Supervisione

Dipendente	Supervisore
485801	512601
512601	774002
774002	512601
94302	774002
218301	512601

Personale_docente

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	5
218301	QUAGLIA	LAURA	Ordinario	10

Esercizio 10: formulare la query SQL per trovare le matricole dei docenti che hanno la stessa classe di stipendio dei loro sottoposti.

```
SELECT ?
FROM Supervisione, Personale_docente AS Dipendente, Personale_docente AS Supervisore
WHERE Supervisione.Dipendente = Dipendente.Matricola_d …
```

CLAUSOLA WHERE

Step 2: correliamo le informazioni sui supervisori con l'informazione sul personale_docente

Assumendo che i docenti possano essere supervisionati da altri docenti

Supervisione

Dipendente	Supervisore
485801	512601
512601	774002
774002	512601
94302	774002
218301	512601

Personale_docente

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	5
218301	QUAGLIA	LAURA	Ordinario	10

Esercizio 10: formulare la query SQL per trovare le matricole dei docenti che hanno la stessa classe di stipendio dei loro sottoposti.

```
SELECT ?
FROM Supervisione, Personale_docente AS Dipendente, Personale_docente AS Supervisore
WHERE Supervisione.Dipendente = Dipendente.Matricola_d AND
Supervisione.Supervisore=Supervisore.Matricola_d ...
```

Supervisione

Dipendente	Supervisore
485801	512601
512601	774002
774002	512601
94302	774002
218301	512601

Personale_docente

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	5
218301	QUAGLIA	LAURA	Ordinario	10

SELECT *
FROM Supervisione, Personale_docente **AS** Dipendente, Personale_docente **AS** Supervisore
WHERE Supervisione.Dipendente = Dipendente.Matricola_d **AND** Supervisione.Supervisore=Supervisore.Matricola_d

Dipendente	Supervisore	Matricola_d	Cognome	Nome	Codice_fiscale	Citta_di_nascita	Ruolo	Classe_stipendio	Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
94302	774002	94302	QUERCINI	PIETRO	CF	14	Ordinario	5	774002	BUZZI	UMBERTO	Ricercatore	3
218301	512601	218301	QUAGLIA	LAURA	CF	190	Ordinario	10	512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
485801	512601	485801	BELOTTI	GIOVANNI	CF	2	Associato	5	512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
512601	774002	512601	CAMPIGLIA	GIUSEPPE	CF	5	Ricercatore	10	774002	BUZZI	UMBERTO	Ricercatore	3
774002	512601	774002	BUZZI	UMBERTO	CF	11	Ricercatore	3	512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10

Assumendo che i docenti possano essere supervisionati da altri docenti

Supervisione

Dipendente	Supervisore
485801	512601
512601	774002
774002	512601
94302	774002
218301	512601

Personale_docente

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	5
218301	QUAGLIA	LAURA	Ordinario	10

Esercizio 10: formulare la query SQL per trovare le matricole dei docenti che hanno la stessa classe di stipendio dei loro sottoposti.

```
SELECT ?
FROM Supervisione, Personale_docente AS Dipendente, Personale_docente AS Supervisore
WHERE Supervisione.Dipendente = Dipendente.Matricola_d AND
Supervisione.Supervisore=Supervisore.Matricola_d ...
```

CLAUSOLA WHERE

Step 3: seleziona le n-uple che soddisfano il criterio sul confronto tra classi di stipendio

Assumendo che i docenti possano essere supervisionati da altri docenti

Supervisione

Dipendente	Supervisore
485801	512601
512601	774002
774002	512601
94302	774002
218301	512601

Personale_docente

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	5
218301	QUAGLIA	LAURA	Ordinario	10

Esercizio 10: formulare la query SQL per trovare le matricole dei docenti che hanno la stessa classe di stipendio dei loro sottoposti.

```
SELECT ?
FROM Supervisione, Personale_docente AS Dipendente, Personale_docente AS Supervisore
WHERE Supervisione.Dipendente = Dipendente.Matricola_d AND
Supervisione.Supervisore=Supervisore.Matricola_d
AND Dipendente.Classe_stipendio=Supervisore.Classe_stipendio
```

TARGET LIST

Infine eseguiamo una proiezione sugli attributi che ci servono

Assumendo che i docenti possano essere supervisionati da altri docenti

Supervisione

Dipendente	Supervisore
485801	512601
512601	774002
774002	512601
94302	774002
218301	512601

Personale_docente

Matricola_d	Cognome	Nome	Ruolo	Classe_stipendio
485801	BELOTTI	GIOVANNI	Associato	5
512601	CAMPIGLIA	GIUSEPPE	Ricercatore	10
774002	BUZZI	UMBERTO	Ricercatore	3
94302	QUERCINI	PIETRO	Ordinario	5
218301	QUAGLIA	LAURA	Ordinario	10

Esercizio 10: formulare la query SQL per trovare le matricole dei docenti che hanno la stessa classe di stipendio dei loro sottoposti.

```
SELECT Supervisore.Matricola_d
FROM Supervisione, Personale_docente AS Dipendente, Personale_docente AS Supervisore
WHERE Supervisione.Dipendente = Dipendente.Matricola_d AND
Supervisione.Supervisore=Supervisore.Matricola_d
AND Dipendente.Classe_stipendio=Supervisore.Classe_stipendio
```

Dato il seguente schema relazionale:

Personale_non_docente(Matricola_d,Cognome,Nome, Ruolo, Classe_stipendio)
Personale_docente(Matricola_d,Cognome,Nome,Ruolo, Classe_stipendio)
Supervisione(Dipendente,Supervisore)
Stipendio(Classe,Valore**)**

Esercizio 11: formulare la query SQL per trovare le matricole che guadagnano più dei loro supervisori

```
SELECT Dipendente.Matricola_d
FROM Supervisione, Personale_docente AS Dipendente, Personale_docente AS Supervisore,
Stipendio AS Stip_dip, Stipendio AS Stip_sup
WHERE Supervisione.Dipendente = Dipendente.Matricola_d AND
Supervisione.Supervisore=Supervisore.Matricola_d AND
Stip_dip.Classe=Dipendente.Classe_stipendio AND
Stip_sup.Classe=Supervisore.Classe_stipendio
AND Stip_dip.Valore> Stip_sup.Valore
```

Esercizi per casa

Formulare le espressioni in SQL che producano:

- **gli studenti che hanno dipendenti (non docenti) omonimi**
- **i docenti che hanno dipendenti (non docenti) omonimi**
- **i docenti che hanno docenti omonimi**