SQL

SQL come Data Manipulation Language

-

(SELECT FROM WHERE ORDER BY)

Esempio 1

Nome	***Eta***	***Reddito***
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Persone

Maternita

Madre	***Figlio***
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Padre	***Figlio***
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Paternita

Interrogazione semplice

Nome e Reddito delle persone con meno di 30 anni.

Nome	*Eta*	*Reddito*
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Persone

Interrogazione semplice

Attributi su cui proiettare il risultato

Nome e **Reddito** delle **persone** con **meno di 30 anni**.

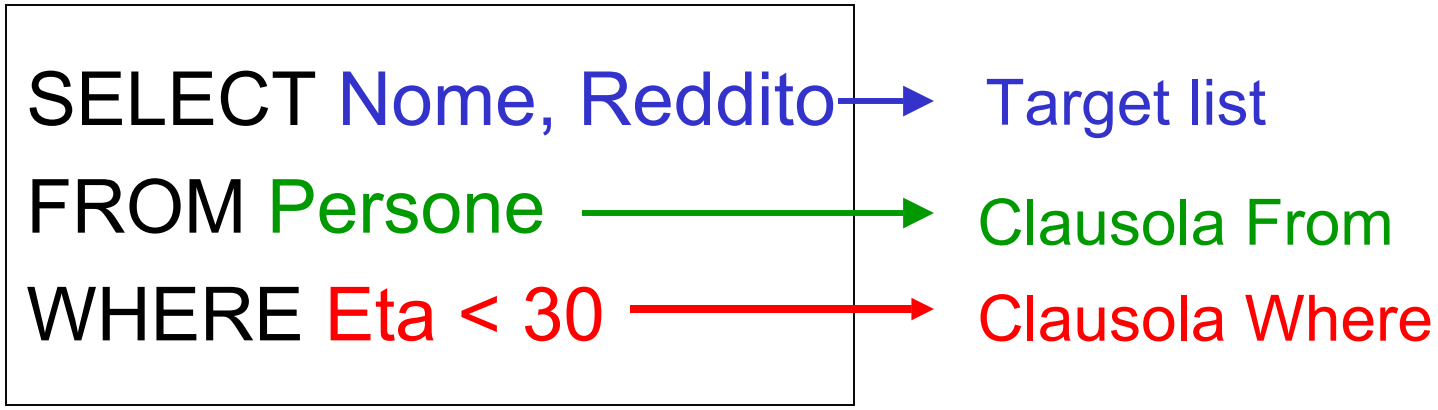
Condizione

Tabella/e da utilizzare

Nome	*Eta*	*Reddito*
Andrea	27	21
Aldo	25	15
~~Maria~~	~~55~~	~~42~~
~~Anna~~	~~50~~	~~35~~
Filippo	26	30
~~Luigi~~	~~50~~	~~40~~
~~Franco~~	~~60~~	~~20~~
~~Olga~~	~~30~~	~~41~~
~~Sergio~~	~~85~~	~~35~~
~~Luisa~~	~~75~~	~~87~~

Persone

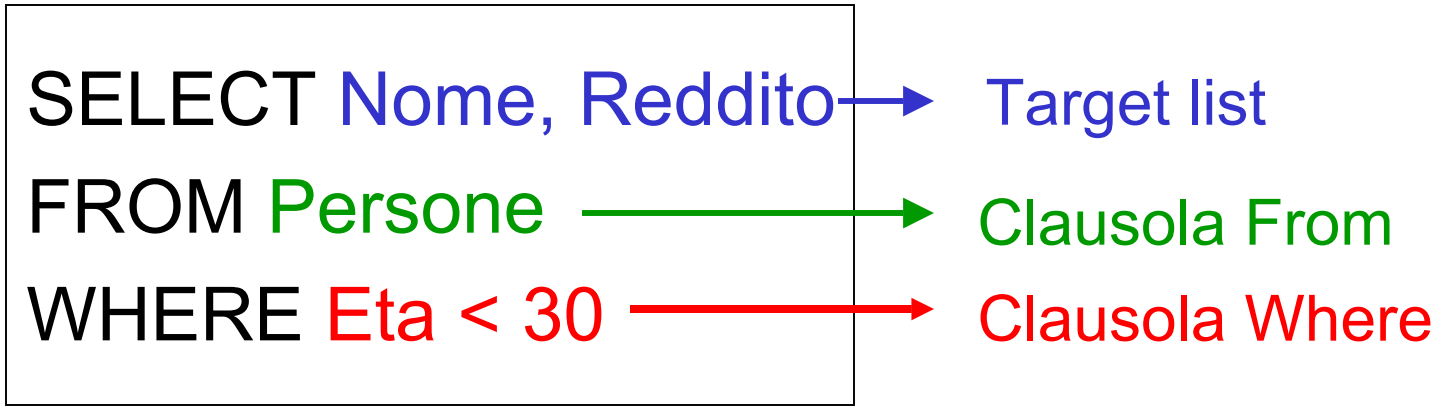
Interrogazione semplice (SQL e algebra)



$$\pi_{\text{Nome, Reddito}} (\sigma_{\text{Eta} < 30} (\text{Persone}))$$

Interrogazione semplice (SQL e algebra)

- Si esegue il prodotto cartesiano delle tabelle coinvolte (in questo caso, essendoci solo una tabella, il p.c. non viene effettuato).
- Si selezionano le righe (tuple) sulla base del predicato della *clausola Where*.
- Si proietta sugli attributi della *target list*.



Prodotto cartesiano

```
SELECT Nome, Reddito
FROM Persone
WHERE Eta < 30
```

Nome	*Eta*	*Reddito*
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Persone

Selezione

```
SELECT Nome, Reddito
FROM Persone
WHERE Eta < 30
```

Nome	*Eta*	*Reddito*
Andrea	**27**	**21**
Aldo	**25**	**15**
Maria	55	42
Anna	50	35
Filippo	**26**	**30**
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Persone

Proiezione

```
SELECT Nome, Reddito
FROM Persone
WHERE Eta < 30
```

Nome	*Eta*	*Reddito*
Andrea	27	**21**
Aldo	25	**15**
Maria	55	42
Anna	50	35
Filippo	26	**30**
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

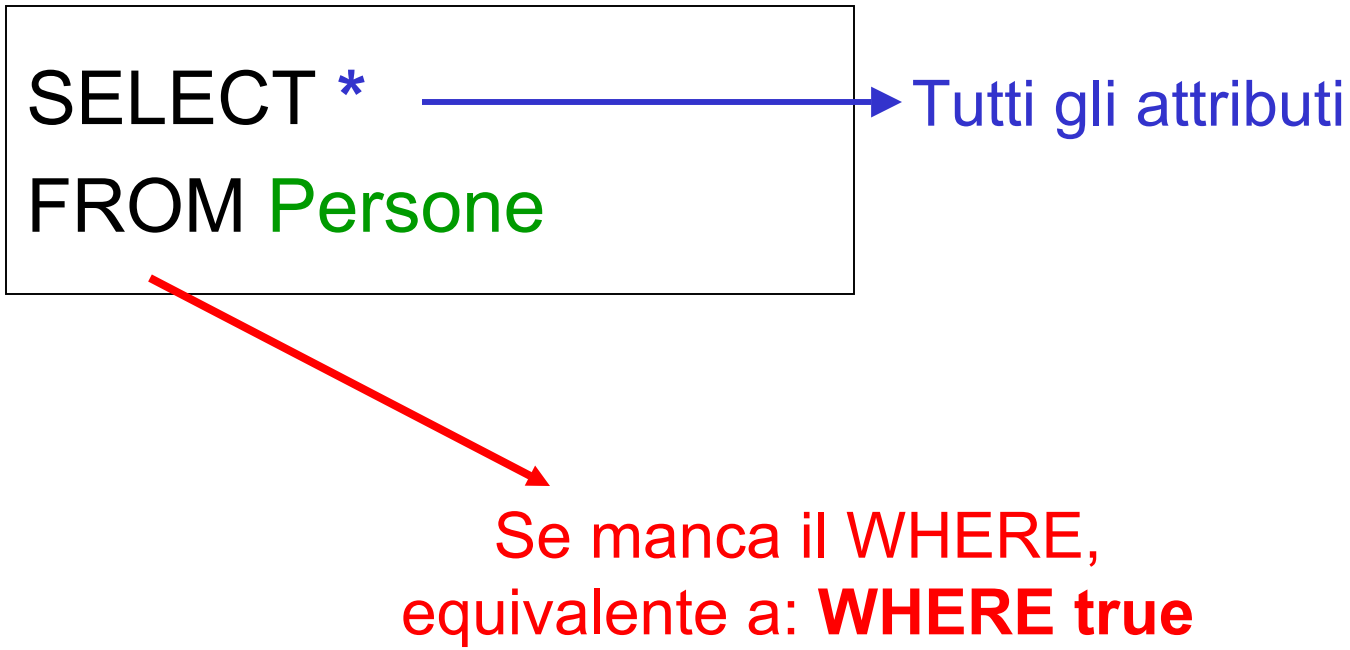
Persone

Risultato

```
SELECT Nome, Reddito
FROM Persone
WHERE Eta < 30
```

Nome	*Reddito*
Andrea	21
Aldo	15
Filippo	30

SELECT: abbreviazioni



SQL

Nome	***Cognome***	***Dipart***	***Ufficio***	***Stipendio***
Mario	Rossi	Amministrazione	10	45
Carlo	Bianchi	Produzione	20	36
Giuseppe	Verdi	Amministrazione	20	40
Franco	Neri	Distribuzione	16	45
Carlo	Rossi	Direzione	14	80
Lorenzo	Lanzi	Direzione	7	73
Paola	Borroni	Amministrazione	75	40
Marco	Franco	Produzione	20	46

Impiegato

Dipartimento

Nome	***Indirizzo***	***Citta***
Amministrazione	Via Tito Livio	Milano
Produzione	Piazza Lavater	Torino
Distribuzione	Via Segre	Roma
Direzione	Via Tito Livio	Milano
Ricerca	Via Morone	Milano

Target list: selezione senza proiezione

```
select *
from Impiegato
where Cognome = 'Rossi'
```

- $(\sigma_{Cognome=Rossi}(Impiegato))$

Nome	***Cognome***	***Dipart***	***Ufficio***	***Stipendio***
Mario	Rossi	Amministrazione	10	45
Carlo	Rossi	Direzione	14	80

Target List: selezione con proiezione

```
select Nome, Cognome, Stipendio
from Impiegato
where Cognome = 'Rossi'
```

- $\pi_{\text{Nome, Cognome, Stipendio}}(\sigma_{\text{Cognome=Rossi}}(\text{Impiegato}))$

Nome	***Cognome***	***Stipendio***
Mario	Rossi	45
Carlo	Rossi	80

Target List: proiezione senza selezione

select Nome, Cognome
from Impiegato

- $\pi_{\text{Nome, Cognome}}$ (Impiegato))

Nome	***Cognome***
Mario	Rossi
Carlo	Bianchi
Giuseppe	Verdi
Franco	Neri
Carlo	Rossi
Lorenzo	Lanzi
Paola	Borroni
Marco	Franco

Target List: proiezione con/senza duplicati

select Cognome
from Impiegato

Cognome
Rossi
Bianchi
Verdi
Neri
Rossi
Lanzi
Borroni
Franco

select distinct Cognome
from Impiegato

Cognome
Rossi
Bianchi
Verdi
Neri
Lanzi
Borroni
Franco

Target List: espressioni

```
select Stipendio/12 AS StipendioMensile
from Impiegato
where Cognome = 'Bianchi'
```

StipendioMensile
3.00

Clausola Where: disgiunzione

```
select Nome, Cognome
from Impiegato
where Dipart = 'Amministrazione' OR
      Dipart = 'Produzione'
```

Nome	***Cognome***
Mario	Rossi
Carlo	Bianchi
Giuseppe	Verdi
Paola	Borroni
Marco	Franco

Clausola Where: condizione complessa

```
select Nome
from Impiegato
where Cognome= 'Rossi' AND
        (Dipart = 'Amministrazione' OR
        Dipart = 'Produzione')
```

Nome
Mario

Clausola Where: operatore IN

```
select *
from Impiegato
where Cognome = 'Rossi' AND
      Dipart IN ('Amministrazione',
                 'Produzione')
```

Nome
Mario

Clausola Where: operatore LIKE

```
select *
from Impiegato
where Cognome LIKE '_o%i'
```

un carattere qualsiasi

stringa qualsiasi

Nome	***Cognome***	***Dipart***	***Ufficio***	***Stipendio***
Mario	Rossi	Amministrazione	10	45
Carlo	Rossi	Direzione	14	80
Paola	Borroni	Amministrazione	75	40

Clausola Where: operatore BETWEEN

```
select *
from Impiegato
where Stipendio BETWEEN 40 AND 45
```

Nome	*Cognome*	*Dipart*	*Ufficio*	*Stipendio*
Mario	Rossi	Amministrazione	10	45
Giuseppe	Verdi	Amministrazione	20	40
Franco	Neri	Distribuzione	16	45
Paola	Borroni	Amministrazione	75	40

Clausola Where: valori nulli

"Impiegati che hanno o potrebbero avere uno stipendio minore di 50 milioni"

Nome	*Cognome*	*Dipart*	*Ufficio*	*Stipendio*
Mario	Rossi	Amministrazione	10	45
Carlo	Rossi	Direzione	14	80
Paola	Borroni	Amministrazione	75	***null***

Imp

Clausola Where: valori nulli

```
select *
from Imp
where Stipendio < 50 or Stipendio IS NULL
```

Nome	***Cognome***	***Dipart***	***Ufficio***	***Stipendio***
Mario	Rossi	Amministrazione	10	45
Paola	Borroni	Amministrazione	75	***null***

Join

- Il Join e' un operatore fondamentale, in quanto permette di utilizzare congiuntamente le informazioni contenute in piu' tabelle.
- In SQL un join si puo' formulare utilizzando i costrutti visti finora (From – Where), che permettono di compiere prodotti cartesiani e selezioni.
- Esistono anche operatori specifici.
- Ripassiamo la semantica del join.

Padre	*Figlio*
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Join naturale

Madre	*Figlio*
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Padre	*Figlio*	*Madre*
Luigi	Olga	Anna
Luigi	Filippo	Anna
Franco	Andrea	Maria
Franco	Aldo	Maria

Left Outer Join

Padre	*Figlio*
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Madre	*Figlio*
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Padre	*Figlio*	*Madre*
Sergio	Franco	***null***
Luigi	Olga	Anna
Luigi	Filippo	Anna
Franco	Andrea	Maria
Franco	Aldo	Maria

Full Outer Join

Padre	*Figlio*
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Madre	*Figlio*
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Padre	*Figlio*	*Madre*
Sergio	Franco	***null***
Luigi	Olga	Anna
Luigi	Filippo	Anna
Franco	Andrea	Maria
Franco	Aldo	Maria
null	Maria	Luisa
null	Luigi	Luisa

Join naturale

- "Padre e Madre di ogni persona"
- Paternita ⋈ Maternita

```
select Paternita.Figlio, Padre, Madre
from Paternita, Maternita
where Paternita.Figlio = Maternita.Figlio
```

Selezione, Proiezione e Join

"I padri di persone che guadagnano più di venti milioni"

- $\pi_{Padre}(Paternita \underset{Figlio = Nome}{\bowtie} (\sigma_{Reddito>20}(Persone)))$

```
select distinct Padre
from Paternita, Persone
where Figlio = Nome AND Reddito > 20
```

Join di una relazione con se stessa (algebra)

"Le persone che guadagnano più dei rispettivi padri. Mostrare nome, reddito e reddito del padre"

$$\pi_{\text{Nome, Reddito, RP}} (\sigma_{\text{Reddito>RP}}$$

$$(\rho_{\text{NP,EP,RP} \leftarrow \text{Nome,Eta,Reddito}}(\text{persone}) \bowtie_{\text{NP=Padre}}$$

$$(\text{paternita} \bowtie_{\text{Figlio = Nome}} \text{persone})))$$

Join di una relazione con se stessa (algebra)

$$\pi_{\text{Nome, Reddito, RP}} (\sigma_{\text{Reddito>RP}}$$

$$(\rho_{\text{NP,EP,RP} \leftarrow \text{Nome,Eta,Reddito}}(\text{persone}) \bowtie_{\text{NP=Padre}}$$

$$(\text{paternita} \bowtie_{\text{Figlio = Nome}} \textbf{persone})))$$

Padre	***Figlio***	***Nome***	***Eta***	***Reddito***
Sergio	Franco	Franco	60	20
Luigi	Olga	Olga	30	41
Luigi	Filippo	Filippo	26	30
Franco	Andrea	Andrea	27	21
Franco	Aldo	Aldo	25	15

Join di una relazione con se stessa (algebra)

$$\pi_{\text{Nome, Reddito, RP}} (\sigma_{\text{Reddito>RP}}$$

$$(\rho_{\text{NP,EP,RP} \leftarrow \text{Nome,Eta,Reddito}}(\textbf{persone}) \bowtie_{\text{NP=Padre}}$$

$$(\text{paternita} \bowtie_{\text{Figlio = Nome}} \textbf{persone})))$$

NP	*EP*	*RP*	*Padre*	*Figlio*	*Nome*	*Eta*	*Reddito*
Sergio	85	35	Sergio	Franco	Franco	60	20
Luigi	50	40	Luigi	Olga	Olga	30	41
Luigi	50	40	Luigi	Filippo	Filippo	26	30
Franco	60	20	Franco	Andrea	Andrea	27	21
Franco	60	20	Franco	Aldo	Aldo	25	15

Join di una relazione con se stessa (algebra)

$$\pi_{\text{Nome, Reddito, RP}} (\sigma_{\text{Reddito>RP}}$$

$$(\rho_{\text{NP,EP,RP} \leftarrow \text{Nome,Eta,Reddito}}(\text{persone}) \bowtie_{\text{NP=Padre}}$$

$$(\text{paternita} \bowtie_{\text{Figlio = Nome}} \text{persone})))$$

NP	*EP*	*RP*	*Padre*	*Figlio*	*Nome*	*Eta*	*Reddito*
Sergio	85	35	Sergio	Franco	Franco	60	20
Luigi	50	40	Luigi	Olga	Olga	30	41
Luigi	50	40	Luigi	Filippo	Filippo	26	30
Franco	60	20	Franco	Andrea	Andrea	27	21
Franco	60	20	Franco	Aldo	Aldo	25	15

Join di una relazione con se stessa (algebra)

$$\pi_{\text{Nome, Reddito, RP}} (\sigma_{\text{Reddito>RP}}$$

$$(\rho_{\text{NP,EP,RP} \leftarrow \text{Nome,Eta,Reddito}}(\text{persone}) \underset{\text{NP=Padre}}{\bowtie}$$

$$(\text{paternita} \underset{\text{Figlio = Nome}}{\bowtie} \text{persone})))$$

Nome	***Reddito***	***RP***
Olga	41	40
Andrea	21	20

Join di una relazione con se stessa (SQL)

"Le persone che guadagnano più dei rispettivi padri. Mostrare nome, reddito e reddito del padre"

$\pi_{\text{Nome, Reddito, RP}}(\sigma_{\text{Reddito>RP}}$

$(\rho_{\text{NP,EP,RP} \leftarrow \text{Nome,Eta,Reddito}}(\textbf{persone}) \bowtie_{\text{NP=Padre}}$

(paternita $\bowtie_{\text{Figlio = Nome}}$ **persone**)))

```
select F.Nome, F.Reddito, P.Reddito
from Paternita, Persone P, Persone F
where Figlio = F.Nome AND P.Nome=Padre
      AND F.Reddito > P.Reddito
```

Stessa cosa, con ridenominazione risultato

"Le persone che guadagnano più dei rispettivi padri. Mostrare nome, reddito e reddito del padre"

```
select  Figlio,
        F.Reddito AS Reddito,
        P.Reddito AS RedditoPadre
from Paternita, Persone P, Persone F
where Figlio = F.Nome AND P.Nome=Padre
        AND F.Reddito > P.Reddito
```

Join

- Vediamo ora una sintassi specifica per il join.
- E' possibile specificare il join interamente all'interno della clausola From.
- Vari tipi di join possono essere formulati:
  - Naturale (esplicito o implicito).
  - Right outer join.
  - Left outer join.
  - Full outer join.

Join esplicito

- "Padre e Madre di ogni persona"
- Paternita ⋈ Maternita

```
select Paternita.Figlio, Padre, Madre
from Paternita join Maternita
  on Paternita.Figlio = Maternita.Figlio
```

Join naturale implicito

- "Padre e Madre di ogni persona"
- Paternita ⋈ Maternita

```
select Paternita.Figlio, Padre, Madre
from Paternita natural join Maternita
```

Left Outer Join esplicito

- "Padre e, se nota, Madre di ogni persona"
- Paternita $\overset{\text{LEFT}}{\underset{\text{Figlio = Nome}}{\bowtie}}$ Maternita

```
select Paternita.Figlio, Padre, Madre
from Paternita left join Maternita
  on Paternita.Figlio = Maternita.Figlio
```

Left Outer Join implicito

- "Padre e, se nota, Madre di ogni persona"
- Paternita $\overset{\text{LEFT}}{\bowtie}$ Maternita

```
select Paternita.Figlio, Padre, Madre
from Paternita left natural join Maternita
```

Full Outer Join

- "Padre e, se nota, Madre di ogni persona"
- Paternita $\overset{\text{FULL}}{\bowtie_{\text{Figlio = Nome}}}$ Maternita

```
select Paternita.Figlio, Padre, Madre
from Paternita full join Maternita
  on Paternita.Figlio = Maternita.Figlio
```

Join esplicito con alias

- "Nome cognome e città lavorativa di ogni impiegato"

```
select I.Nome, Cognome, Citta
from Impiegato I join Dipartimento D
        on Dipart = D.Nome
```

Join esplicito con alias: risultato

Nome	*Cognome*	*Citta*
Mario	Rossi	Milano
Carlo	Bianchi	Torino
Giuseppe	Verdi	Milano
Franco	Neri	Roma
Carlo	Rossi	Milano
Lorenzo	Lanzi	Milano
Paola	Borroni	Milano
Marco	Franco	Torino

Ordinamento del risultato

- A differenza del modello relazionale, in cui le tuple non sono ordinate, le righe di una tabella possono esserlo.
- Talvolta la possibilità di ordinare il risultato di un'interrogazione è importante. Ad esempio, se si vogliono gli stipendi in ordine dal minore al maggiore.
- SQL mette a disposizione la clausola ORDER BY.

Ordinamento del risultato

```
select Cognome, Nome, Stipendio
from Impiegato
where Dipartimento LIKE 'Amm%'
ORDER BY Stipendio DESC, Cognome ASC
```

Cognome	***Nome***	***Stipendio***
Rossi	Mario	45
Borroni	Paola	40
Verdi	Giuseppe	40

Default