Interrogazioni semplici

- Le interrogazioni esprimono cosa si cerca e non come si cerca
- Il come è prodotto dall'interprete SQL del DBMS che traduce l'interrogazione SQL nelle operazioni interne sui dati che producono la risposta
- L'utente non deve quindi preoccuparsi di come si trova il risultato
- Le operazioni di interrogazione vengono specificate con l'istruzione **`select`**

select

Matricola	Cognome	Nome	Data di nascita
A80198760	Bianchi	Anna	22/03/1967
A80293450	Rossi	Andrea	13/04/1968
A80198330	Neri	Luca	04/08/1970
A80295640	Rossi	Lorenzo	25/02/1969
A80197456	Melli	Mara	17/10/1966

Studenti

Campi considerati

Tabelle da cui si selezionano

```
select Matricola as NumeroMatricola
from Studenti
where Cognome = 'Rossi'
```

Condizioni

Nuovo nome per l'attributo

NumeroMatricola
A80293450
A80295640

Risultato: una relazione!

select *

Matricola	Cognome	Nome	Data di nascita
A80198760	Bianchi	Anna	22/03/1967
A80293450	Rossi	Andrea	13/04/1968
A80198330	Neri	Luca	04/08/1970
A80295640	Rossi	Lorenzo	25/02/1969
A80197456	Melli	Mara	17/10/1966

Studenti

Tutti i campi sono considerati

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

Matricola	Cognome	Nome	Data di nascita
A80293450	Rossi	Andrea	13/04/1968
A80295640	Rossi	Lorenzo	25/02/1969

select (espressioni)

Prodotto	Cliente	Quantita	PrezzoUnitario
Chiodo 35mm	Bianchi	100	15
Vite 45mm	Rossi	50	20
Dado 5mm	Neri	200	8
Vite 8mm	Rossi	20	25
Bulloni 5cm	Melli	40	50

```
select Prodotto, Quantita*PrezzoUnitario as PrezzoTotale
from Ordini
where Cliente = 'Rossi'
```

Espressione che combina più attributi

Prodotto	PrezzoTotale
Vite 45mm	1000
Vite 8mm	500

Funzioni

SQL mette a disposizione funzioni predefinite

- stringhe (SUBSTR, TRIM, LENGTH, | ,….)
- date, intervalli (+, -, DATE, DAYOFWEEK,...)
- matematiche (* , + , - , / , TAN, SQRT, SIN,...)
- informazioni di sistema (USER, CURRENT_DATE,...)

Esempi

Nome e cognome degli studenti

```
select  cognome | ‘ ‘ | nome as nomeCompleto
from studenti
```

Matricola	Cognome	Nome
A80198760	Bianchi	Anna
A80293450	Rossi	Andrea
A80198330	Neri	Luca
A80295640	Rossi	Lorenzo
A80197456	Melli	Mara

NomeCompleto
Bianchi Anna
Rossi Andrea
Neri Luca
Rossi Lorenzo
Melli Mara

Esempi

Mese di nascita degli studenti

```
select  monthname("data di nascita") as meseDiNascita
from studenti
```

Cognome	Nome	Data di nascita
Bianchi	Anna	22/03/1967
Rossi	Andrea	13/04/1968
Neri	Luca	04/08/1970
Rossi	Lorenzo	25/02/1969
Melli	Mara	17/10/1966

meseDiNascita
marzo
aprile
agosto
febbraio
ottobre

Valori distinti

- A differenza del calcolo o dell'algebra relazionale, i risultati delle interrogazioni SQL contengono duplicati
- I duplicati possono essere rimossi usando **distinct**

```
select  distinct nome from studenti
```

Cognome	Nome
Bianchi	Anna
Rossi	Andrea
Neri	Anna
Rossi	Andrea
Melli	Mara

Con **distinct**

Nome
Anna
Andrea
Mara

Senza **distinct**

Nome
Anna
Andrea
Anna
Andrea
Mara

Ordinamento

- La clausola `order by` definisce l'ordine con cui i record devono essere restituiti
- Senza la clausola, l'ordinamento dipende dal sistema, e potrebbe anche cambiare ad ogni interrogazione

Studenti ordinati per cognome (ordine inverso) e nome

```
select cognome, nome from studenti
order by cognome desc, nome asc
```

Cognome	Nome
Rossi	Andrea
Rossi	Lorenzo
Neri	Luca
Melli	Mara
Bianchi	Anna

Select … where

- La clausola **`where`** ammette come argomento una condizione costruita da
  - condizioni semplici (ad esempio confronti con =, <> ,<,>,<=,>= fra costanti o valori degli attributi)
  - espressioni ottenute usando gli operatori **`and`**, **`or`** e **`not`**
- Si possono usare le funzioni predefinite
- Vengono selezionate solo le righe per cui la condizione è vera

Esempi

Ricerca degli impiegati con reddito maggiore di 100 milioni e età non compresa fra 40 e 50 anni

```
select *
from impiegati
where reddito >= 100 and not (eta <=50 and eta>=40)
```

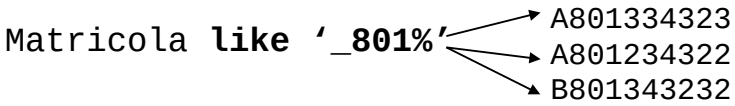
La precedenza fra gli operatori **and** e **or** non è definita dallo standard

Ricerca degli studenti nati di domenica

```
select *
from studenti
where dayofweek("data di nascita") = 1
```


Ricerca di stringhe

- Per il confronto di stringhe si può usare l'operatore like che supporta i caratteri speciali _ (qualsiasi carattere) e % (qualsiasi sequenza di caratteri)



Ricerca degli studenti il cui indirizzo inizia per "via"

```
select *
from studenti
where indirizzo like 'via%'
```

Valori nulli

I valori nulli possono significare

- valore non conosciuto
- valore non applicabile
- non si sà se il valore è sconosciuto o non applicabile

- SQL-89 usa una logica a due valori
  - un confronto con **null** produce **FALSE**
- SQL-2 usa una logica a tre livelli
  - un paragone con **null** produce **UNKNOWN**

select ….. from

- Per accedere a righe appartenenti a più di una tabella, si usa come argomento della clausola **`from`** la lista della tabelle a cui si vuole accedere
- Sul prodotto cartesiano delle tabelle elencate vengono applicate le condizioni della clausola **`where`**
- Se si vuole effettuare un join occorre scrivere in modo esplicito le condizioni che esprimono il legame tra le diverse tabelle

Select e prodotto cartesiano

```
select  * from tab1, tab2 where attr1='A'
```

tab1

Attr1	Attr2
A	B
B	A

tab2

Attr3	Attr4
C	B
D	B

Prodotto cartesiano

Attr1	Attr2	Attr3	Attr4
A	B	C	B
B	A	C	B
A	B	D	B
B	A	D	B

Selezione

Attr1	Attr2	Attr3	Attr4
A	B	C	B
A	B	D	B

Il join con select

Matricola	Cognome	Nome	Data di nascita
A80198760	Bianchi	Anna	22/03/1967
A80293450	Rossi	Andrea	13/04/1968
A80295640	Felici	Lorenzo	25/02/1969

Studenti

Studente	Voto	Corso
A80198760	28	Analisi I
A80293450	30	Basi di Dati
A80295640	27	Analisi I
A80198760	30L	Fisica I
A80293450	21	Chimica

Esami

Attributi di tabelle

```
select Studenti.Cognome,Studenti.Nome,Esami.Voto
from Studenti,Esami
where (Studenti.Matricola = Esami.Studente) AND
      (Esami.Corso = 'Analisi I')
```

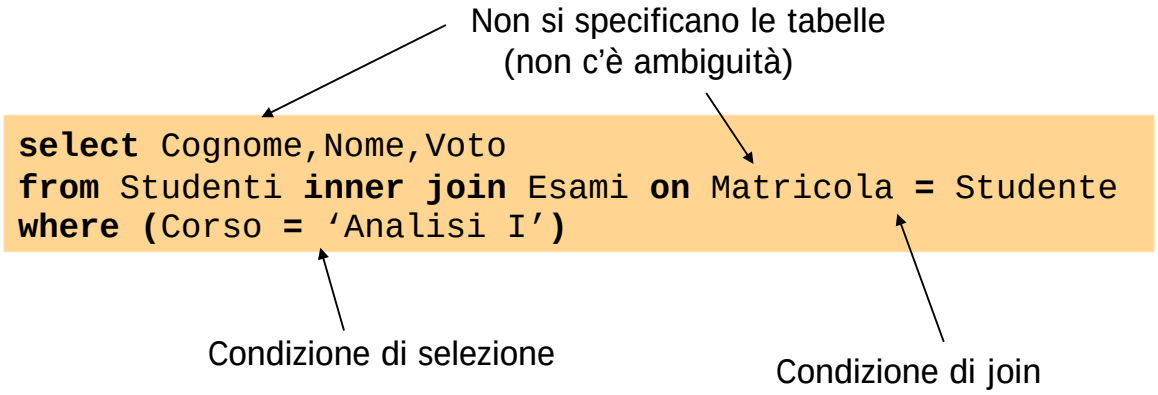
prodotto cartesiano

Condizione di join

Cognome	Nome	Voto
Bianchi	Anna	28
Felici	Lorenzo	27

join interno

- Modo alternativo per indicare il join di due tabelle che distingue le condizioni di join da quelle di selezione delle righe



join esterno

- Quando si fa il join può capitare che alcune delle righe di una tabella non vengano selezionate perché non c'è nessuna riga dell'altra tabella che soddisfa la condizione di join
- Il join esterno esegue il join interno mantenendo però tutte le righe che fanno parte di una o di entrambe le tabelle coinvolte come segue:

❶ outer left sono aggiunte le righe della relazione a sinistra del join che non hanno righe corrispondenti nella tabella di destra

❷ outer right sono aggiunte le righe della relazione a destra del join che non hanno righe corrispondenti nella tabella di sinistra

❸ outer full compaiono tutte le righe delle due tabelle

outer left join

Guidatore

Cognome	Nome	NroPatente
Bianchi	Anna	VR 2030020Y
Rossi	Andrea	PZ 1012436B
Felici	Lorenzo	AP 4544442R

Automobile

Targa	Marca	Modello	NroPatente
AB574WW	Fiat	Punto	VR2030020Y
AA652FF	Renault	Clio	VR2030020Y
BJ747XX	Ford	Focus	PZ1012436B
BB421JJ	Renault	Megane	MI2020030U

Associa un altro nome alla tabella (alias)

```
select *
from Guidatore G left join Automobile A
       on (G.NroPatente = A.NroPatente)
```

ci sarebbe stata ambiguità

Cognome	Nome	NroPatente	Targa	Marca	Modello
Bianchi	Anna	VR 2030020Y	AB574WW	Fiat	Punto
Bianchi	Anna	VR 2030020Y	AA652FF	Renault	Clio
Rossi	Andrea	PZ 1012436B	BJ747XX	Ford	Focus
Felici	Lorenzo	AP 4544442R	NULL	NULL	NULL

Uso di variabili

- Permette di far riferimento a più esemplari della stessa tabella

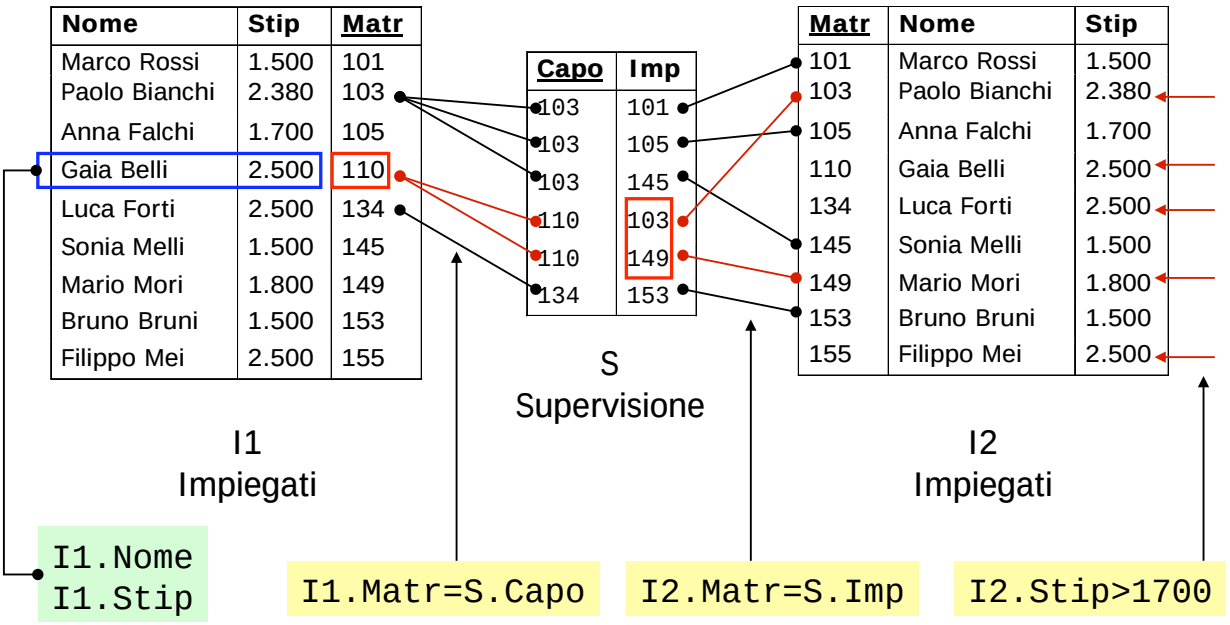
Trovare nome e stipendio dei capi degli impiegati che guadagnano più di 1.700 Euro

```
select I1.Nome,I1.Stip
from Impiegati I1, Supervisione S, Impiegati I2
where I1.Matr = S.Capo and
      I2.Matr = S.Imp and
      I2.Stip > 1700
```

Copia I1: usata per ricavare le informazioni sul capo

Copia I2: usata per riferirsi agli impiegati

Esempio di uso di variabili



Operatori aggregati

- Permettono di costruire interrogazioni che coinvolgono più tuple contemporaneamente
- Ad esempio: contare il numero di studenti che hanno superato un esame, trovare il massimo voto preso da uno studente ad un esame, calcolare la media di uno studente
- Si definiscono gli operatori aggregati che combinano fra loro più tuple

count	conteggio righe
sum	somma
max	massimo
min	minimo
avg	media

Conteggio, massimo, media...

Studente	Voto	Corso
A80198760	28	01
A80293450	30	04
A80198760	27	03
A80293450	25	03
A80293450	21	05

```
select count(*) from esami
where Studente = 'A80293450'
```

```
select max(Voto) from esami
where Studente = 'A80293450'
```

```
select avg(Voto) from esami
where Studente = 'A80293450'
```

Informazioni sui voti dello studente A80293450

```
select count(distinct Studente)
from esami
```

Conteggio di quanti studenti hanno fatto almeno un esame

Interrogazioni con raggruppamento

- Il raggruppamento permette di applicare gli operatori aggregati distintamente a sottoinsiemi di righe
  - Prima si esegue la query senza `group by` e operatori aggregati
  - Poi si esegue il raggruppamento in accordo agli attributi in `group by`
  - Infine si calcolano gli operatori aggregati

Calcolare la media di ogni studente

```
select Studente,avg(Voto) as Media
from Esami
group by Studente
```

Interrogazioni con raggruppamento II

Calcolare la media di ogni studente

Studente	Voto	Corso
A80198760	28	01
A80293450	30	04
A80198760	27	03
A80293450	25	03
A80293450	21	05

```
select Studente,avg(Voto) as Media
from Esami
group by Studente
```

Studente	Voto
A80198760	28
A80198760	27
A80293450	30
A80293450	25
A80293450	21

Studente	Media
A80198760	27.500
A80293450	25.333

Restrizioni della sintassi

- Nelle interrogazioni con raggruppamento, nella clausola **select** possono comparire solo gli attributi che compaiono anche nella clausola `group by`

Una interrogazione scorretta

```
select cognome, nome, "data di nascita"
from studenti left join Esami on Matricola = Studente
group by cognome, nome
```

Una interrogazione corretta

```
select cognome, nome, "data di nascita"
from studenti left join Esami on Matricola = Studente
group by cognome, nome,"data di nascita"
```

Interrogazioni con raggruppamento e selezione

- Si possono selezionare solo alcuni raggruppamenti in base a condizioni di tipo aggregato

Trovare la media degli studenti che hanno sostenuto almeno 3 esami

Studente	Voto	Corso
A80198760	28	01
A80293450	30	04
A80198760	27	03
A80293450	25	03
A80293450	21	05

```
select Studente,avg(Voto) as Media
from Esami
group by Studente
having count(*)>=3
```

Studente	Voto
A80198760	28
A80198760	27
A80293450	30
A80293450	25
A80293450	21

Studente	Media
A80293450	25.333

Having vs where

- La clausola `having`
  - è applicata dopo il raggruppamento
  - si applica solo ad attributi presenti nel `group by` e ad operatori aggregati
- La clausola where si applica prima del raggruppamento

```
select Studente,avg(Voto)
from Esami group by Studente
having avg(Voto)>25
```

Gli studenti con media maggiore di 25

Fare la media usando solo i voti maggiori di 25

```
select Studente,avg(Voto)
from Esami group by Studente
where voto>25
```