

SQL

2 lezione

interrogazioni

Francesca Gasparini
gasparini@disco.unimib.it

Generalità

SQL permette di effettuare interrogazioni (query) in modo **dichiarativo**: si specifica obiettivo di interrogazione, non il modo per effettuarla.

Ogni interrogazione SQL è passata da DBMS a **Query Optimizer**, che la traduce in linguaggio procedurale interno al sistema per eseguirla in modo efficiente (passaggio trasparente a utente).

Le query devono essere costruite per essere leggibili e modificabili, non necessariamente efficienti.

2

Istruzione SELECT

Le interrogazioni avvengono usando l'istruzione SELECT:

SELECT *ListaAttributi* — TARGET LIST
Cosa si vuole come risultato

FROM *ListaTabelle* — Dove reperire le informazioni

[WHERE Condizioni] — Che condizioni devono essere soddisfatte

3

Istruzione SELECT

SELECT *AttrEspr* [[**AS**] *Alias*] { , *AttrEspr* [[**AS**] *Alias*] }

FROM *Tabella* [[**AS**] *Alias*] { , *Tabella* [[**AS**] *Alias*] }

[WHERE Condizione]

Nome attributo o
Espressione

Permette di definire dei
nomi-alias per l'elemento

4

Istruzione SELECT : Esempio 1

Query: Individuare lo stipendio degli impiegati di cognome "Rossi"

```
SELECT Stipendio
FROM Impiegato
WHERE Cognome="Rossi"
```



Stipendio
45
80

Impiegato

ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
1	Mario	Rossi	Amministrazione	10	45		6 Ingegnere	Milano	
2	Carlo	Bianchi	Produzione	20	36		2 Tecnico	Torino	
3	Giuseppe	Venti	Amministrazione	25	40		3 amministratore	Roma	
4	Franco	Neri	Distribuzione	16	45		5 segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	60		0 dirigente	Milano	
6	Lorenzo	Lanzi	Direzione	7	73		1	Genova	
7	Paola	Bonetti	Amministrazione	75	40		1 segretaria	Venezia	
8	Marco	Franco	Produzione	20	46		ingegnere	Roma	
9	Francesca	Rivoli		60	60		3 ricercatore	Milano	
10	Giovanni	Orsini		7			segretaria	Genova	

5

Istruzione SELECT : Esempio 2

Query: Selezionare nome, cognome e mansione degli impiegati dell'ufficio 10

```
SELECT Nome, Cognome, Mansione
FROM Impiegato
WHERE ufficio='10'
```

Nome	Cognome	mansione
Mario	Rossi	Ingegnere

Impiegato

ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
1	Mario	Rossi	Amministrazione	10	45		6 Ingegnere	Milano	
4	Franco	Neri	Distribuzione	16	45		5 segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	60		0 dirigente	Milano	
6	Lorenzo	Lanzi	Direzione	7	73		1	Genova	
8	Marco	Franco	Produzione	20	46		ingegnere	Roma	
9	Francesca	Rivoli		60	60		3 ricercatore	Milano	

6

Istruzione SELECT : Esempio 3

Query: Estrarre lo stipendio mensile relativo agli impiegati di cognome Rivoli

```
SELECT stipendio/12 as StipendioMensile
FROM Impiegato
WHERE cognome='Rivoli'
```

Impiegato									
ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
1	Mario	Rossi	Amministrazione	10	45		6 Ingegnere	Milano	7
4	Franco	Neri	Distribuzione	16	45		5 segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	80		0 dirigente	Milano	
6	Lorenzo	Lanzi	Direzione	7	73		1	Genova	
8	Marco	Franco	Produzione	20	46		Ingegnere	Roma	9
9	Francesca	Rivoli		60	60		3 ricercatore	Milano	5

StipendioMensile
5

7

Istruzione SELECT e Algebra Relazionale

Query: Individuare le informazioni degli impiegati con stipendio > 40

```
SELECT *
FROM Impiegato
WHERE Stipendio>40
```



$\sigma_{\text{Stipendio}>40}(\text{Impiegato})$

Impiegato

ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
1	Mario	Rossi	Amministrazione	10	45		6 Ingegnere	Milano	7
4	Franco	Neri	Distribuzione	16	45		5 segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	80		0 dirigente	Milano	
6	Lorenzo	Lanzi	Direzione	7	73		1	Genova	
8	Marco	Franco	Produzione	20	46		Ingegnere	Roma	9
9	Francesca	Rivoli		60	60		3 ricercatore	Milano	5

8

Istruzione SELECT e Algebra Relazionale

Query: Selezionare nome, cognome e mansione degli impiegati dell'ufficio 10

$\pi_{\text{Nome,Cognome,Mansione}}(\sigma_{\text{Ufficio}=10}(\text{Impiegato}))$

3 SELECT Nome, Cognome, Mansione
1 FROM Impiegato
2 WHERE ufficio='10'

9

Istruzione SELECT : Target List

Query: Selezionare tutte le informazioni sui dipendenti

```
SELECT id_impiegato, Nome, Cognome, Dipart, Ufficio,
stipendio, premioprod, Mansione, Città, idcapo
FROM Impiegato
```

```
SELECT *
FROM Impiegato
```

Manca la clausola WHERE

Attenzione nome attributo con spazi

ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
1	Mario	Rossi	Amministrazione	10	45		6 Ingegnere	Milano	7
2	Carlo	Bianchi	Produzione	20	36		2 tecnico	Torino	6
3	Giuseppe	Verdi	Amministrazione	20	40		3 amministrativo	Roma	7
4	Franco	Neri	Distribuzione	16	45		5 segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	80		0 dirigente	Milano	
6	Lorenzo	Lanzi	Direzione	7	73		1	Genova	
7	Paola	Borroni	Amministrazione	75	40		1 segretaria	Venezia	
8	Marco	Franco	Produzione	20	46		Ingegnere	Roma	9
9	Francesca	Rivoli		60	60		3 ricercatore	Milano	5
10	Giovanni	Oscar		7			segreteria	Genova	

0

Istruzione Select : Target List

Query: Selezionare cognome, stipendio e premio di produzione per gli impiegati che hanno uno stipendio compreso tra 38 e 60

```
SELECT Cognome, stipendio, premioprod AS premio_di_produzione
FROM Impiegato
WHERE ( Stipendio>=38 AND Stipendio <=60 )
```

Cognome	stipendio	premio_di_produzione
Rossi	45	6
Verdi	40	3
Neri	45	5
Borroni	40	1
Franco	46	
Rivoli	60	3

11

Istruzione SELECT : Clausola FROM

Specifica la (le) tabella(e) su cui operare per determinare il risultato.

Query: Selezionare i nomi, i cognomi e le città di provenienza degli impiegati

```
SELECT Nome, Cognome, Città AS città_provenienza
FROM Impiegato
```

Nome	Cognome	città_provenienza
Mario	Rossi	Milano
Carlo	Bianchi	Torino
Giuseppe	Verdi	Roma
Franco	Neri	Napoli
Carlo	Rossi	Milano
Lorenzo	Lanzi	Genova
Paola	Borroni	Venezia
Marco	Franco	Roma

12

Istruzione SELECT : Clausola FROM

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Manione	Città	Id capo
Dip	NomeDip	Indirizzo	CittàDip							

Query: Selezionare i nomi e i cognomi degli impiegati, le città di provenienza e le città in cui lavorano

SELECT Nome, Cognome, CittàDip AS città_lavoro,
Città AS città_provenienza

FROM Impiegato, Dip

WHERE Dipart = NomeDip

I dati da recuperare sono contenuti in due tabelle distinte

condizione che lega le due diverse tabelle

13

Istruzione SELECT : Clausola FROM

SELECT Nome, Cognome, CittàDip AS città_lavoro,
Città AS città_provenienza

FROM Impiegato, Dip

WHERE Dipart = NomeDip

Specificare più tabelle su cui operare per determinare il risultato significa: eseguire il prodotto cartesiano delle tabelle e poi selezionare le righe che soddisfano clausola WHERE.

=

Join di Algebra Relazionale

14

Istruzione SELECT : Clausola FROM

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Manione	Città	Id capo
Dip	NomeDip	Indirizzo	CittàDip							

SELECT Nome, Cognome, CittàDip AS città_lavoro,
Città AS città_provenienza

FROM Impiegato, Dip

WHERE Dipart = NomeDip

π Nome, Cognome, CittàDip, Città (Dip JOIN NomeDip=Dipart Impiegato)

π Nome, Cognome, CittàDip, Città (Dip JOIN (Dip JOIN (Dipart=NomeDip Impiegato)))

15

Istruzione SELECT : Clausola FROM

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Manione	Città	Id capo
Dip	NomeDip	Indirizzo	CittàDip							

Query: Selezionare i nomi e i cognomi degli impiegati, le città di provenienza e le città in cui lavorano

SELECT Nome, Cognome, CittàDip AS città_lavoro,
Città AS città_provenienza

FROM Impiegato, Dip

WHERE Dipart = NomeDip

Nome	Cognome	città lavoro	città provenienza
Mario	Rossi	Milano	Milano
Carlo	Bianchi	Torino	Torino
Giuseppe	Verdi	Milano	Roma
Franco	Neri	Roma	Napoli
Carlo	Rossi	Milano	Milano
Lorenzo	Lanzi	Milano	Genova
Paola	Borroni	Milano	Venezia
Marco	Franco	Torino	Roma

16

Clausola FROM : Notazione punto

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Manione	Città	Id capo
Dip	Nome	Indirizzo	Città							

Query: Selezionare i nomi e i cognomi degli impiegati, le città di provenienza e le città in cui lavorano

SELECT Nome, Cognome, Città AS città_lavoro,

Città AS città_provenienza

FROM Impiegato, Dip

WHERE Dipart = NomeDip

17

Clausola FROM : Notazione punto

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Manione	Città	Id capo
Dip	Nome	Indirizzo	Città							

Si usano Alias per ridenominare le tabelle:

- 1) Abbreviano riferimento a tabelle
- 2) Risolvono ambiguità di riferimento

SELECT I.Nome, Cognome, D.Città AS città_lavoro,

I.Città AS città_provenienza

FROM Impiegato I, Dip D

WHERE Dipart = D.Nome

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

18

Clausola FROM : Notazione punto

Impiegato

ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
--------------	------	---------	--------	---------	-----------	------------	----------	-------	--------

Dip

Nome	Indirizzo	Città
------	-----------	-------

Query: Selezionare i nomi e i cognomi degli impiegati, le città di provenienza e le città in cui lavorano

SELECT Impiegato.Nome, Impiegato.Cognome,
Dip.Città AS città_lavoro, Impiegato.Città AS città_provenienza

FROM Impiegato, Dip
WHERE Impiegato.Dipart = Dip.Nome

19

Istruzione SELECT : Clausola WHERE

Specifica condizione di selezione.

L'argomento di WHERE è un'espressione booleana:

[NOT] *PredicatoSemplice* { < AND | OR >
[NOT] *PredicatoSemplice* }

Predicati semplici: Espressioni di valutazione su attributi mediante operatori di confronto

Operatori di confronto: =, <>, <, >, <=, >=, LIKE, BETWEEN, IS NULL, IS NOT NULL

20

Istruzione SELECT : Clausola WHERE

Si selezionano per il risultato solo le tuple per le quali l'espressione è **vera**.

Precedenza di valutazione tra operatori booleani:

- 1) NOT
- 2) AND, OR

Per stabilire la precedenza usare sempre le parentesi! Es:

(Exp1 AND (Exp2 OR NOT Exp3)) OR (Exp3 AND Exp4)

21

Istruzione SELECT : Clausola WHERE

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
-----------	--------------	------	---------	--------	---------	-----------	------------	----------	-------	--------

Query: Estrarre nome, cognome e mansione degli impiegati che lavorano nell'ufficio 20 del dipartimento Amministrazione

$\pi_{\text{Nome, cognome, mansione}}(\sigma_{\text{ufficio}=20 \wedge \text{dipartimento}=\text{'amministrazione'}}(\text{Impiegato}))$

SELECT Nome, Cognome, mansione
FROM Impiegato
WHERE (Ufficio = '20') AND (Dipart = 'Amministrazione')

Nome	Cognome	mansione
Giuseppe	Verdi	amministrativo

22

Istruzione SELECT : Clausola WHERE

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
-----------	--------------	------	---------	--------	---------	-----------	------------	----------	-------	--------

Query: Estrarre nome e cognome degli impiegati che lavorano nel dipartimento Amministrazione o nel dipartimento Produzione

$\pi_{\text{nome, cognome}}(\sigma_{\text{dipartimento}=\text{'amministrazione'} \vee \text{dipartimento}=\text{'produzione'}}(\text{Impiegato}))$

SELECT Nome, Cognome
FROM Impiegato
WHERE (Dipart = 'Amministrazione')
OR (Dipart = 'Produzione')

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

23

Istruzione SELECT : Clausola WHERE

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
-----------	--------------	------	---------	--------	---------	-----------	------------	----------	-------	--------

Query: Estrarre i nomi propri degli impiegati di cognome 'Rossi' che lavorano nei dipartimenti Amministrazione o Produzione

SELECT Nome
FROM Impiegato
WHERE (Cognome='Rossi') AND
(Dipart = 'Amministrazione' OR Dipart = 'Produzione')

Nome
Mario

24

Clausola WHERE : Operatore LIKE

L'operatore **LIKE** permette di esprimere dei "pattern" su stringhe mediante "wildcard":

- _ (un carattere arbitrario) (in ACCESS ?)
- % (una stringa arbitraria) (in ACCESS *)

Query: Nomi degli impiegati che finiscono con una 'i' e hanno una 'o' in seconda posizione

25

Clausola WHERE : Operatore LIKE

Impiegato ID Impiegato Nome Cognome Dipart Ufficio Stipendio premioprod Mandione Città Idcapo

Query: Nomi degli impiegati che finiscono con una 'i' e hanno una 'o' in seconda posizione

```
SELECT Cognome
FROM Impiegato
WHERE Cognome LIKE '_o%i'
```

in ACCESS:

```
SELECT cognome
FROM Impiegato
WHERE Cognome LIKE "?o*i"
```

Cognome
Rossi
Rossi
Borroni

26

Clausola WHERE : Operatore BETWEEN

Impiegato ID Impiegato Nome Cognome Dipart Ufficio Stipendio premioprod Mandione Città Idcapo

L'operatore **BETWEEN** permette di esprimere condizioni di appartenenza a un intervallo

Query: Nome e stipendio degli impiegati che hanno uno stipendio compreso tra 38 e 60 (estremi inclusi)

```
SELECT Nome, Stipendio
FROM Impiegato
WHERE Stipendio BETWEEN 38 AND 60
```

```
SELECT Nome, Stipendio
FROM Impiegato
WHERE Stipendio >= 38 AND Stipendio <= 60
```

Cognome	stipendio
Rossi	45
Verdi	40
Neri	45
Borroni	40
Franco	46

27

Clausola WHERE : Operatore IN

Impiegato ID Impiegato Nome Cognome Dipart Ufficio Stipendio premioprod Mandione Città Idcapo

L'operatore **IN** permette di esprimere condizioni di appartenenza a un insieme

Query: Cognome e ufficio degli impiegati degli uffici 10 e 20

```
SELECT Cognome, ufficio
FROM Impiegato
WHERE ufficio IN ('10','20')
```

```
SELECT Cognome, ufficio
FROM Impiegato
WHERE ufficio = '10' OR ufficio = '20'
```

Cognome	ufficio
Rossi	10
Bianchi	20
Verdi	20
Franco	20

28

Clausola WHERE : Valori nulli

SQL-2 usa una logica a tre valori per valutare il valore di verità di una clausola WHERE: True (T), False (F), Unknown (?)

- 1) Un predicato semplice valutato su un attributo a valore nullo dà come risultato della valutazione ?.
- 2) Una tupla per cui il valore di verità è ? non viene restituita dalla query.
- 3) Se la valutazione del predicato di un constraint è ? il constraint non è violato

29

Clausola WHERE : Valori nulli

Il valore di verità di un predicato complesso viene calcolato in base alle seguenti tabelle di verità:

AND	T	F	?
T	T	F	?
F	F	F	F
?	?	F	?

OR	T	F	?
T	T	T	T
F	F	F	?
?	T	?	?

NOT	T	F
T	F	T
F	T	F
?	?	?

30

Clausola WHERE : Valori nulli

	A	B	C
t1	a	?	c1
t2	a1	b	c2
t3	a2	?	?

SELECT * FROM Tabella WHERE A=a OR B=b

Il valore di verità della condizione per ogni tupla è il seguente:

t1:	T OR ?	→ T
t2:	F OR T	→ T
t3:	F OR ?	→ ?

31

Clausola WHERE : Valori nulli

	A	B	C
t1	a	?	c1
t2	a1	b	c2
t3	a2	?	?

SELECT * FROM Tabella WHERE A=a AND B=b

Il valore di verità della condizione per ogni tupla è il seguente:

t1:	T AND ?	→ ?
t2:	F AND T	→ F
t3:	F AND ?	→ F

Nessuna tupla verifica la query.

32

Clausola WHERE : Valori nulli

	A	B	C
t1	a	?	c1
t2	a1	b	c2
t3	a2	?	?

SELECT * FROM Tabella WHERE NOT C=c1

Il valore di verità della condizione per ogni tupla è il seguente:

t1:	NOT T	→ F
t2:	NOT F	→ T
t3:	NOT ?	→ ?

33

Clausola WHERE : Operatore IS [NOT] NULL

Come valutare attributi nulli?

Attributo IS [NOT] NULL

IS NULL: vero se attributo ha valore nullo, falso altrimenti

IS NOT NULL: vero se attributo ha valore specificato, falso altrimenti

34

Clausola WHERE : Operatore IS NULL

	A	B	C
t1	a	?	c1
t2	a1	b	c2
t3	a2	?	?

SELECT * FROM Tabella WHERE B IS NULL

Restituisce le tuple t1 e t3

SELECT * FROM Tabella WHERE B IS NULL AND C IS NULL

Restituisce la tupla t3

SELECT * FROM Tabella WHERE B IS NULL OR C IS NULL

Restituisce le tuple t1 e t3

35

Clausola WHERE : Operatore IS NULL

	A	B	C
t1	a	?	c1
t2	a1	b	c2
t3	a2	?	?

SELECT * FROM Tabella WHERE B IS NOT NULL

Restituisce la tupla t2

Si noti che le interrogazioni

SELECT * FROM Tabella WHERE B = 'b' OR B <> 'b'

SELECT * FROM Tabella WHERE B IS NOT NULL

sono equivalenti

36

Clausola WHERE : Operatore IS NULL

Impiegato

ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcaga
--------------	------	---------	--------	---------	-----------	------------	----------	-------	--------

Query: Gli Impiegati che sono o potrebbero essere ingegneri.

```
SELECT *  
FROM impiegato  
WHERE mansione = 'Ingegnere' OR mansione IS NULL
```

ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcaga
1	Mario	Rossi	Amministrazione	10	45		6 Ingegnere	Milano	7
6	Lorenzo	Lanzi	Direzione	7	73	1		Genova	
8	Marco	France	Produzione	20	46		ingegnere	Roma	9

37

Istruzione SELECT : DISTINCT

In algebra relazionale i risultati delle interrogazioni non contengono elementi duplicati.

In SQL, le tabelle prodotte dalle interrogazioni possono contenere più righe identiche tra loro.

I duplicati possono essere rimossi usando la parola chiave **DISTINCT**.

```
SELECT [ DISTINCT ] AttrExpr  
FROM Tabella  
[ WHERE Cond ]
```

38

Istruzione SELECT : DISTINCT

Impiegato

ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcaga
--------------	------	---------	--------	---------	-----------	------------	----------	-------	--------

Query: Ritornare i cognomi degli impiegati e le città di provenienza.

```
SELECT DISTINCT Cognome, Città  
FROM Impiegato  
WHERE stipendio>40
```

Il risultato della query SQL contiene righe duplicate:

Cognome	Città
Franco	Roma
Lanzi	Genova
Neri	Napoli
Rossi	Milano

39

Espressioni e Funzioni

I predicati usati nelle interrogazioni possono coinvolgere, oltre a nomi di colonna, anche **espressioni**.

Le espressioni sono formulate applicando operatori ai valori delle colonne delle tuple.

Esempi di espressioni e funzioni sono quelle aritmetiche, su stringhe, su date e tempi.

Le espressioni possono comparire nelle clausole SELECT, WHERE e UPDATE (si vedrà più avanti)

40

Espressioni e Funzioni

Una espressione usata nella clausola SELECT dà luogo ad una colonna, detta **virtuale**, non presente nella relazione su cui si effettua l'interrogazione.

Le colonne virtuali non sono fisicamente memorizzate, ma sono **materializzate** (esistono) come risultato delle interrogazioni

Anche alle colonne virtuali è possibile assegnare un alias.

41

Espressioni e Funzioni

SQL mette a disposizione **funzioni predefinite**:

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

42

Funzioni Aritmetiche

Impiegato ID Impiegato Nome Cognome Dipart Ufficio Stipendio premioprod Mansione Città Idcaga

Query: Ritornare lo stipendio mensile degli impiegati che guadagnano più di 40

```
SELECT stipendio/12 AS stipendio_mensile
FROM Impiegato
WHERE stipendio>40;
```

stipendio_mensile
3.75
3.75
6.666666666666667
6.083333333333333
3.833333333333333

43

Funzioni Aritmetiche

Impiegato ID Impiegato Nome Cognome Dipart Ufficio Stipendio premioprod Mansione Città Idcaga

Query: trovare il nome, lo stipendio, il premio di produzione, e la somma dello stipendio e del premio di produzione di tutti gli ingegneri.

```
SELECT Nome, Stipendio, Premioprod,
       Stipendio+Premioprod AS stipendio_totale
FROM Impiegato
WHERE Mansione = 'Ingegnere'
```

Nome	Stipendio	Premioprod	stipendio_totale
Mario	45	6	51

44

Funzioni Aritmetiche

Impiegato ID Impiegato Nome Cognome Dipart Ufficio Stipendio premioprod Mansione Città Idcaga

Query: trovare il nome, lo stipendio, il premio di produzione, e la somma di stipendio e premio di produzione di tutti gli impiegati del dipartimento amministrazione per cui la somma di stipendio e premio di produzione è maggiore di 42.

```
SELECT Nome, Stipendio, Premioprod,
       Stipendio+Premioprod AS stipendio_totale
FROM Impiegato
WHERE dipart='amministrazione' AND
       Stipendio+Premioprod >42
```

Nome	Stipendio	Premioprod	stipendio_totale
Mario	45	6	51
Giuseppe	40	3	43

45

Funzioni per stringhe

Gli operatori più comuni per le stringhe sono:

|| (in ACCESS +) Operatore di concatenazione

UPPER, UCASE Trasforma la stringa in caratteri maiuscoli

LOWER, LCASE Trasforma la stringa in caratteri minuscoli

46

Funzioni per stringhe

```
SELECT Cognome || " " || Nome || " " || città
FROM Impiegato
```

Restituisce un'unica stringa che contiene cognome, nome ed indirizzo separati da uno spazio bianco.

```
SELECT nome, cognome, città, mansione
FROM impiegato
WHERE UCASE(mansione)='INGEGNERE'
```

Nome	Cognome	città	mansione
Mario	Rossi	Milano	Ingegnere
Marco	Franco	Roma	ingegnere

47

Operatore Join

Operatore di JOIN

L'operatore di JOIN rappresenta un'importante funzionalità in quanto permette di correlare dati in relazioni diverse.

In SQL il JOIN è espresso tramite un prodotto cartesiano tra tabelle a cui sono applicati uno o più predicati di join.

Un predicato di join esprime una condizione che deve essere verificata dalle tuple del risultato dell'interrogazione.

49

Tabelle di Esempio

Dip

Nome	Indirizzo	Città
Amministrazione	via Tito Livio 27	Milano
direzione	via Tito Livio 27	Milano
distribuzione	Via Segre 9	Roma
produzione	P.le Lavater 3	Torino
ricerca	via Morone 6	Milano

Impiegato

ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
1	Mario	Rossi	Amministrazione	10	45		6. Ingegnere	Milano	7
2	Carlo	Bianchi	Produzione	20	36		2. tecnico	Torino	5
3	Giuseppe	Verdi	Amministrazione	20	40		3. amministrativo	Roma	7
4	Franco	Neri	Distribuzione	15	45		5. segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	80		0. dirigente	Milano	
6	Lorenzo	Lanzi	Direzione	7	73		1	Genova	
7	Paola	Borroni	Amministrazione	75	40		1. segretaria	Venezia	
8	Marco	Franco	Produzione	20	45		ingegnere	Roma	9
9	Federica	Rivoli	Vendite	60	60		3. ricercatore	Milano	5
10	Giovanni	Oscar	Amministrazione	7			segreteria	Genova	

50

Operatore di JOIN

Impiegato	ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
Dip	Nome	Indirizzo	Città							

Query: Estrarre tutte le informazioni sugli impiegati e sui dipartimenti in cui lavorano.

```
SELECT Impiegato.*,Dip.*
FROM Impiegato, Dip
WHERE Impiegato.Dipart = Dip.Nome ;
```

ID impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Impiegato	Migro	dip.Nome	Indirizzo	dip.Città
1	Mario	Rossi	Amministrazione	10	45		6. Ingegnere	Milano		7. Amministrazione	via Tito Livio 27	Milano
2	Carlo	Bianchi	Produzione	20	36		2. tecnico	Torino		5. produzione	P.le Lavater 3	Torino
3	Giuseppe	Verdi	Amministrazione	20	40		3. amministrativo	Roma		7. Amministrazione	via Tito Livio 27	Milano
4	Franco	Neri	Distribuzione	15	45		5. segretario	Napoli		7. distribuzione	Via Segre 9	Roma
5	Carlo	Rossi	Direzione	14	80		0. dirigente	Milano		direzione	via Tito Livio 27	Milano
6	Lorenzo	Lanzi	Direzione	7	73		1	Genova		1. direzione	via Tito Livio 27	Milano
7	Paola	Borroni	Amministrazione	75	40		1. segretaria	Venezia		Amministrazione	via Tito Livio 27	Milano
8	Marco	Franco	Produzione	20	45		ingegnere	Roma		9. produzione	P.le Lavater 3	Torino
10	Giovanni	Oscar	Amministrazione	7			segreteria	Genova		Amministrazione	via Tito Livio 27	Milano

51

Operatore di JOIN

Studenti

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

Esami

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

Query: Estrarre Nome, Cognome e Voto di tutti gli studenti che hanno sostenuto l'esame di 'Analisi I'.

```
SELECT Studenti.Cognome,Studenti.Nome, Esami.Voto
FROM Studenti, Esami
WHERE (Studenti.Matricola = Esami.Studente) AND
(Esami.Corso = 'Analisi I')
```

Cognome	Nome	Voto
Bianchi	Anna	28
Felici	Lorenzo	27

52

Operatore di JOIN

```
SELECT Impiegato.*,Dip.*
FROM Impiegato, Dip
WHERE Impiegato.Dipart = Dip.Nome

SELECT Studenti.Cognome,Studenti.Nome, Esami.Voto
FROM Studenti, Esami
WHERE (Studenti.Matricola = Esami.Studente) AND
(Esami.Corso = 'Analisi I')
```

Attributi di tabelle
Prodotto Cartesiano
Predicati di JOIN

53

Operatore di JOIN: join esplicito

Sintassi alternativa per Join, introdotta da SQL-2:

```
SELECT AttrExpr [ [AS] Alias ] { , AttrExpr [ [AS] Alias ] }
FROM Tabella [ [AS] Alias ]
{ [ TipoJoin ] JOIN Tabella [ [AS] Alias ] ON CondizJoin }
[ WHERE Cond ]
```

Esplícita il JOIN e separa le condizioni di JOIN dalle condizioni di query.

TipoJoin può essere inner, RIGHT [OUTER], LEFT [OUTER] oppure FULL [OUTER], consentendo la rappresentazione dei join esterni.

54

Operatore di INNER JOIN

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Manifone	Città	Idcapo
Dip	Nome	Indirizzo	Città							

SELECT I.Nome, Cognome, Dip.Città
FROM Impiegato **AS** I, Dip
WHERE Dipart = Dip.Nome

SELECT I.Nome, Cognome, Dip.Città
FROM Impiegato **AS** I **[INNER] JOIN** Dip
ON Dipart = Dip.Nome

Non si ha traccia delle tuple di Impiegato che non corrispondono ad alcuna tupla di Dip.

55

Operatore di OUTER JOIN

L'operatore di **OUTER JOIN** aggiunge al risultato le tuple di Impiegato e Dip che non hanno partecipato al **JOIN**, completandole con **NULL**.

Esistono tre tipologie di **OUTER JOIN**:

LEFT [OUTER] JOIN

RIGHT [OUTER] JOIN

FULL [OUTER] JOIN

56

Operatore di OUTER JOIN

Consideriamo:

(...) Impiegato **[FULL | LEFT | RIGHT] OUTER JOIN** Dip (...)

FULL: Sia le tuple di Impiegato che quelle di Dip che non partecipano al join vengono completate (con valori **NULL**) ed inserite nel risultato.

LEFT: Le tuple di Impiegato che non partecipano al join vengono completate (con valori **NULL**) ed inserite nel risultato.

RIGHT: Le tuple di Dip che non partecipano al join vengono completate (con valori **NULL**) ed inserite nel risultato.

57

Operatore di JOIN

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Manifone	Città	Idcapo
Dip	Nome	Indirizzo	Città							

Query: Estrarre le informazioni sui dipendenti e sui dipartimenti di appartenenza.

SELECT [id impiegato], I.Nome, Cognome, I.Dipart, I.Città,
dip.Nome, dip.Indirizzo, dip.Città
FROM impiegato **AS** I **INNER JOIN** dip
ON dip.Nome = I.Dipart;

ID impiegato	Nome	Cognome	Dipart	I.Città	dip.Nome	Indirizzo	dip.Città
7	Paola	Bortoni	Amministrazione	Venezia	Amministrazione	via Tito Livio 27	Milano
10	Giovanni	Oscar	Amministrazione	Genova	Amministrazione	via Tito Livio 27	Milano
1	Mario	Rossi	Amministrazione	Milano	Amministrazione	via Tito Livio 27	Milano
3	Giuseppe	Vendi	Amministrazione	Roma	Amministrazione	via Tito Livio 27	Milano
6	Lucrezio	Lanzi	Direzione	Genova	direzione	via Tito Livio 27	Milano
5	Carlo	Rossi	Direzione	Milano	direzione	via Tito Livio 27	Milano
4	Franco	Neri	Distribuzione	Napoli	distribuzione	Via Segre 9	Roma
2	Carlo	Bianchi	Produzione	Torino	produzione	P.le Lavater 3	Torino
9	Marco	Franco	Produzione	Roma	produzione	P.le Lavater 3	Torino

58

Operatore di JOIN

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Manifone	Città	Idcapo
Dip	Nome	Indirizzo	Città							

Query: Estrarre le informazioni sui dipendenti e sugli eventuali dipartimenti di appartenenza.

SELECT [id impiegato], I.Nome, Cognome, I.Dipart, I.Città,
dip.Nome, dip.Indirizzo, dip.Città
FROM Impiegato **AS** I **LEFT JOIN** dip **ON** dip.Nome = I.Dipart

ID impiegato	Nome	Cognome	Dipart	I.Città	dip.Nome	Indirizzo	dip.Città
7	Paola	Bortoni	Amministrazione	Venezia	Amministrazione	via Tito Livio 27	Milano
10	Giovanni	Oscar	Amministrazione	Genova	Amministrazione	via Tito Livio 27	Milano
1	Mario	Rossi	Amministrazione	Milano	Amministrazione	via Tito Livio 27	Milano
3	Giuseppe	Vendi	Amministrazione	Roma	Amministrazione	via Tito Livio 27	Milano
6	Lucrezio	Lanzi	Direzione	Genova	direzione	via Tito Livio 27	Milano
5	Carlo	Rossi	Direzione	Milano	direzione	via Tito Livio 27	Milano
4	Franco	Neri	Distribuzione	Napoli	distribuzione	Via Segre 9	Roma
2	Carlo	Bianchi	Produzione	Torino	produzione	P.le Lavater 3	Torino
9	Marco	Franco	Produzione	Roma	produzione	P.le Lavater 3	Torino
9	Francesca Rivoli	Vendite	Milano				

59

Operatore di JOIN

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Manifone	Città	Idcapo
Dip	Nome	Indirizzo	Città							

Query: Estrarre le informazioni sui dipartimenti e sugli eventuali dipendenti.

SELECT [id impiegato], I.Nome, Cognome, I.Dipart, I.Città,
dip.Nome, dip.Indirizzo, dip.Città
FROM Impiegato **AS** I **RIGHT JOIN** dip
ON dip.Nome = I.Dipart;

ID impiegato	Nome	Cognome	Dipart	I.Città	dip.Nome	Indirizzo	dip.Città
7	Paola	Bortoni	Amministrazione	Venezia	Amministrazione	via Tito Livio 27	Milano
3	Giuseppe	Vendi	Amministrazione	Roma	Amministrazione	via Tito Livio 27	Milano
1	Mario	Rossi	Amministrazione	Milano	Amministrazione	via Tito Livio 27	Milano
6	Lucrezio	Lanzi	Produzione	Genova	produzione	P.le Lavater 3	Torino
2	Carlo	Bianchi	Produzione	Torino	produzione	P.le Lavater 3	Torino
4	Franco	Neri	Distribuzione	Napoli	distribuzione	Via Segre 9	Roma
6	Lucrezio	Lanzi	Direzione	Genova	direzione	via Tito Livio 27	Milano
6	Carlo	Rossi	Direzione	Milano	direzione	via Tito Livio 27	Milano
					ricerca	via Moro 6	Milano

60

Operatore di JOIN

Guidatore			Auto			
Nome	Cogn	Patente	Targa	Marca	Modello	Patente
Mario	Rossi	VR 2030020Y	AB 574 WW	Fiat	Punto	VR 2030020Y
Carlo	Bianchi	PZ 1012436B	AA 652 FF	Fiat	Brava	VR 2030020Y
Marco	Neri	AP 4544442R	BJ 747 XX	Lancia	Delta	PZ 1012436B
			BB 421 JJ	Fiat	Uno	MI 2020030U

Query: Estrarre i guidatori con le automobili loro associate, mantenendo nel risultato anche i guidatori senza auto.

SELECT Nome, Cogn, G.Patente, Targa, Marca, Modello
FROM Guidatore **AS** G **LEFT JOIN** Auto
ON G.Patente = Auto.Patente

Nome	Cogn	Patente	Targa	Marca	Modello
Mario	Rossi	VR2030020Y	AB 574 WW	Fiat	Punto
Mario	Rossi	VR2030020Y	AA 652 FF	Fiat	Brava
Carlo	Bianchi	PZ1012436B	BJ 747 XX	Lancia	Delta
Marco	Neri	AP4544442R	NULL	NULL	NULL

61

Operatore di JOIN

Guidatore			Auto			
Nome	Cogn	Patente	Targa	Marca	Modello	Patente
Mario	Rossi	VR 2030020Y	AB 574 WW	Fiat	Punto	VR 2030020Y
Carlo	Bianchi	PZ 1012436B	AA 652 FF	Fiat	Brava	VR 2030020Y
Marco	Neri	AP 4544442R	BJ 747 XX	Lancia	Delta	PZ 1012436B
			BB 421 JJ	Fiat	Uno	MI 2020030U

Query: Estrarre tutti i guidatori e tutte le automobili mostrando tutte le relazioni esistenti tra di essi.

SELECT Nome, Cogn, G.Patente, Targa, Marca, Modello
FROM Guidatore **AS** G **FULL JOIN** Auto
ON G.Patente = Auto.Patente

Nome	Cogn	Patente	Targa	Marca	Modello
Mario	Rossi	VR2030020Y	AB 574 WW	Fiat	Punto
Mario	Rossi	VR2030020Y	AA 652 FF	Fiat	Brava
Carlo	Bianchi	PZ1012436B	BJ 747 XX	Lancia	Delta
Marco	Neri	AP4544442R	NULL	NULL	NULL
NULL	NULL	NULL	BB 421 JJ	Fiat	Uno

62

Self JOIN

Genitori

Genitore	Figlio
Luca	Anna
Maria	Anna
Giorgio	Luca
Silvia	Maria
Enzo	Maria

Query: Chi sono i nonni di Anna?

63

Self JOIN

Query: Chi sono i nonni di Anna?

SELECT G1.Genitore **AS** Nonno
FROM Genitori G1, Genitori G2
WHERE G1.Figlio = G2.Genitore
AND G2.Figlio = 'Anna'

E' come avere una copia della tabella.

Genitori G1

Genitore	Figlio
Luca	Anna
Maria	Anna
Giorgio	Luca
Silvia	Maria
Enzo	Maria

Genitori G2

Genitore	Figlio
Luca	Anna
Maria	Anna
Giorgio	Luca
Silvia	Maria
Enzo	Maria

64

Self JOIN

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
Dip	Nome	Indirizzo	Città							

Query: Estrarre cognome, nome e dipartimento di tutti gli impiegati che hanno lo stesso cognome (ma diverso nome) di impiegati del dipartimento Amministrazione.

ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
1	Mario	Rossi	Amministrazione	10	45	6	Ingegnere	Milano	7
2	Carlo	Bianchi	Produzione	20	36	2	tecnico	Torino	5
3	Giuseppe	Verdi	Amministrazione	20	40	3	amministrativo	Roma	7
4	Franco	Neri	Distribuzione	15	45	5	segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	80	0	dirigente	Milano	
6	Lorenzo	Lenzi	Direzione	7	73	1		Genova	
7	Paola	Borroni	Amministrazione	75	40	1	segretaria	Venezia	8
8	Marco	Franca	Produzione	20	46		ingegnere	Roma	9
9	Francesca	Rivoli		60	60	3	ricercatore	Milano	5
10	Giovanni	Oscar		7			segreteria	Genova	

65

Self JOIN

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
Dip	Nome	Indirizzo	Città							

Query: Chi sono i dipendenti di Paola Borroni?

ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	Idcapo
1	Mario	Rossi	Amministrazione	10	45	6	Ingegnere	Milano	7
2	Carlo	Bianchi	Produzione	20	36	2	tecnico	Torino	5
3	Giuseppe	Verdi	Amministrazione	20	40	3	amministrativo	Roma	7
4	Franco	Neri	Distribuzione	15	45	5	segretario	Napoli	7
5	Carlo	Rossi	Direzione	14	80	0	dirigente	Milano	
6	Lorenzo	Lenzi	Direzione	7	73	1		Genova	
7	Paola	Borroni	Amministrazione	75	40	1	segretaria	Venezia	8
8	Marco	Franco	Produzione	20	46		ingegnere	Roma	9
9	Francesca	Rivoli		60	60	3	ricercatore	Milano	5
10	Giovanni	Oscar		7			segreteria	Genova	

dobbiamo cioè cercare gli impiegati che hanno capo con idcapo uguale all'ID impiegato che si chiama Paola Borroni

66

Self JOIN

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	idcapo
Dip	Nome	Indirizzo	Città							

Query: Chi sono i dipendenti di Paola Borroni?

```
SELECT I1.Nome + ' ' + I1.Cognome as impiegato, I1.idcapo,
I1.IDImpiegato, I2.Nome + ' ' + I2.Cognome AS capo
FROM Impiegato AS I1, Impiegato AS I2
WHERE I1.idcapo=I2.IDImpiegato AND
      I2.Nome='Paola' AND I2.Cognome='Borroni';
```

impiegato	idcapo	ID impiegato	capo
Mario Rossi	7	1	Paola Borroni
Giuseppe Vardi	7	3	Paola Borroni
Franco Neri	7	4	Paola Borroni

67

Ordinamento del risultato : ORDER BY

```
SELECT AttrExpr [ [AS] Alias ] { , AttrExpr [[AS] Alias ] }
FROM Tabella [ [AS] Alias ] { , Tabella [[AS] Alias ] }
[ WHERE Cond ]
ORDER BY Attr [ ASC | DESC ] { , Attr [ ASC | DESC ] }
```

Le righe ordinate in base al primo attributo. Se i valori sono uguali, viene ordinata in base al secondo attributo, etc.

Ordinamento: **ascendente (ASC)**, **discendente (DESC)** (ascendente è default e può essere omissa).

68

Ordinamento del risultato

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	idcapo
Dip	Nome	Indirizzo	Città							

Query: Elencare lo stipendio, la mansione, il nome e il cognome di tutti gli impiegati che hanno stipendio > 40, ordinando il risultato in ordine crescente in base allo stipendio.

```
SELECT Stipendio, Mansione, Nome, Cognome
FROM Impiegato
WHERE stipendio>40
ORDER BY Stipendio
```

Stipendio	Mansione	Nome	Cognome
45	segretario	Franco	Neri
45	Ingegnere	Mario	Rossi
46		Marco	Franco
73		Lorenzo	Lanzi
80	dirigente	Carlo	Rossi

69

Ordinamento del risultato

Impiegato	ID Impiegato	Nome	Cognome	Dipart	Ufficio	Stipendio	premioprod	Mansione	Città	idcapo
Dip	Nome	Indirizzo	Città							

Query: Elencare mansione, nome e stipendio di tutti gli impiegati, ordinando il risultato in base al cognome in ordine crescente ed in base allo stipendio in ordine decrescente.

```
SELECT Mansione, Stipendio, Nome, Cognome
FROM Impiegato
ORDER BY cognome,
          Stipendio DESC
```

Mansione	Stipendio	Nome	Cognome
tecnico	36	Carlo	Bianchi
segretaria	40	Paola	Borroni
	46	Marco	Franco
	73	Lorenzo	Lanzi
segretario	45	Franco	Neri
dirigente	80	Carlo	Rossi
Ingegnere	45	Mario	Rossi
amministrativo	40	Giuseppe	Verdi

70