

Interrogazioni in SQL

Dichiaratività di SQL

- E' un linguaggio di **IV generazione**: in SQL l'utente specifica **QUALE** informazione è di suo interesse ma non **COME** estrarla dai dati
- il sistema costruisce ed esegue una strategia di accesso ottima (**QUERY OPTIMIZATION**)
- è l'aspetto più qualificante delle basi di dati relazionali

Struttura di SQL

- **Basata sulla composizione di blocchi**
SELECT
FROM
WHERE
- **Ogni blocco ha il potere espressivo di una qualunque combinazione di selezioni, proiezioni e join**

Esempio: gestione degli esami universitari

studente

MATR	NOME	CITTA'	C-DIP
123	Carlo	Bologna	Inf
415	Paola	Torino	Inf
702	Antonio	Roma	Log

esame

MATR	COD-CORSO	DATA	VOTO
123	1	1997-09-07	30
123	2	1998-01-08	28
702	2	1997-09-07	20

corso

COD-CORSO	TITOLO	DOCENTE
1	matematica	Barozzi
2	informatica	Natali

Interrogazione semplice

```
SELECT *  
FROM STUDENTE
```

MATR	NOME	CITTA'	C-DIP
123	Carlo	Bologna	Inf
415	Paola	Torino	Inf
702	Antonio	Roma	Log

```
SELECT NOME  
FROM STUDENTE  
WHERE C-DIP = 'Log'
```

NOME
Antonio

Sintassi nella clausola SELECT

SELECT *

SELECT NOME, CITTÀ

SELECT DISTINCT CITTÀ

SELECT CITTÀ AS LUOGO-DI-RESIDENZA

SELECT REDDITO-CATASTALE * 0.05

AS TASSA-ICI

SELECT SUM (SALARIO)

Sintassi della clausola FROM

FROM STUDENTE

FROM STUDENTE **AS** X

FROM STUDENTE X

FROM STUDENTE, ESAME

FROM STUDENTE **JOIN** ESAME

ON STUDENTE.MATR=ESAME.MATR

Sintassi del predicato di selezione

espressione booleana di predicati semplici

operazioni booleane :

- AND (P1 AND P2)
- OR (P1 OR P2)
- NOT (P1)

predicati semplici :

- TRUE, FALSE
- termine
comparatore
termine

comparatore :

- =, !=, <, <=, >, >=

termine :

- costante, attributo
- espressione aritmetica di costanti e attributi

Sintassi della clausola WHERE

- **Espressione Booleana di predicati semplici (come in algebra)**
- **Alcuni predicati aggiuntivi:**
 - **BETWEEN (estremi inclusi!) :**
DATA BETWEEN '1990-01-01' AND '1999-12-31'
 - **LIKE:**
C-DIP LIKE 'log%'
TARGA LIKE 'MI_777_8%'

Selezione

```
SELECT *  
FROM STUDENTE  
WHERE NOME='Paola'
```

Il risultato è una tabella (priva di nome) con

- **schema :**

lo stesso schema di STUDENTE

- **istanze :**

**le tuple di STUDENTE che soddisfano
il predicato di selezione**

MATR	NOME	CITTA'	C-DIP
415	Paola	Torino	Inf

Esempio di selezione

```
SELECT *  
FROM STUDENTE  
WHERE (CITTÀ='Torino') OR  
((CITTÀ='Roma') AND NOT (C-DIP='log'))
```

MATR	NOME	CITTÀ	C-DIP
123	Carlo	Bologna	Inf
415	Paola	Torino	Inf
702	Antonio	Roma	Log

gestione duplicati (proiezione)

```
SELECT  
DISTINCT C-DIP  
FROM STUDENTE
```

C-DIP
Inf
Log

(relazione)

```
SELECT C-DIP  
FROM STUDENTE
```

C-DIP
Inf
Inf
Log

(tabella)

Valori nulli

```
SELECT *  
FROM STUDENTE  
WHERE CITTÀ IS [NOT] NULL
```

attenzione :

se CITTÀ ha valore NULL
il risultato per (CITTÀ = 'Milano')
ha valore 'UNKNOWN'

Composizione di predicati con valore nullo

- logica
a tre valori (V,F,U)

V AND U = U

V OR U = V

F AND U = F

F OR U = U

NOT U = U

- **P =**
(CITTÀ IS NOT NULL)
AND (C-DIP LIKE '%Inf')

CITTA'	C-DIP	P	TUPLA SELEZ.
Milano	Inf	V	si
Milano	NULL	U	no
NULL	Inf	F	no
Milano	Log	F	no

JOIN su due tabelle

```
SELECT NOME  
FROM STUDENTE, ESAME  
WHERE STUDENTE.MATR = ESAME.MATR  
AND C-DIP LIKE 'In%' AND VOTO = 30
```



NOME
Carlo

variante sintattica:

```
SELECT NOME  
FROM STUDENTE JOIN ESAME  
  ON STUDENTE.MATR = ESAME.MATR  
WHERE C-DIP LIKE 'In%' AND VOTO = 30
```


Join fra tre tabelle

```
SELECT NOME  
FROM STUDENTE, ESAME, CORSO  
WHERE STUDENTE.MATR = ESAME.MATR  
AND CORSO.COD-CORSO = ESAME.COD-CORSO  
AND TITOLO LIKE 'info%' AND VOTO < 24
```



NOME
Antonio

Interrogazioni con variabili relazionali

chi sono i dipendenti “non-pendolari”?

MATR	NOME	CITTÀ	SALARIO	MATR-MGR
1	Piero	BO	3 M	2
2	Giorgio	MO	2,5 M	4
3	Giovanni	FE	2 M	2
....

impiegato

dipartimento

DNO	NOME	CITTÀ	MATR-MGR
1	AMMINISTRAZIONE	BO	2
2	SPEDIZIONI	FE	7
...

Variabili relazionali (alias)

```
SELECT X.NOME  
FROM IMPIEGATO AS X, DIPARTIMENTO AS Y  
WHERE X.MATR-MGR = Y.MATR_MGR  
AND Y.CITTÀ = X.CITTÀ
```



X.NOME
Piero

```
AND Y.CITTÀ != X.CITTÀ
```



X.NOME
Giovanni

Variabili relazionali nel self-join

Chi sono i dipendenti di Giorgio?

```
SELECT X.NOME  
FROM IMPIEGATO AS X, IMPIEGATO AS Y  
WHERE X.MATR-MGR = Y.MATR  
AND Y.NOME = 'Giorgio'
```

X.NOME	X.MATR-MGR	Y.MATR	Y.NOME
Piero	2	2	Giorgio
Giovanni	2	2	Giorgio

Blocchi SQL per la modifica

tre operazioni elementari

- 1 cancellazione: **DELETE**
- 2 inserimento: **INSERT**
- 3 modifica: **UPDATE**

1 Cancellazione

```
DELETE FROM STUDENTE  
WHERE MATR = '678678'
```

2 Inserimento

```
INSERT INTO STUDENTE  
VALUES ('456878', 'Giorgio Rossi',  
'Bologna', 'Logistica e Produzione')
```

```
INSERT INTO BOLOGNESI  
(SELECT *  
FROM STUDENTE  
WHERE CITTÀ = 'Bologna')
```

3 Modifica

```
UPDATE ESAME  
SET VOTO = 30  
WHERE DATA = '1997-04-01'
```

```
UPDATE ESAME  
SET VOTO = VOTO + 1  
WHERE MATR = '787989'
```

Esercizi

- Dato un DB per la gestione del personale esprimere in SQL le **interrogazioni** seguenti:
 - in quali tipi di progetti lavora Giovanni?
 - chi è il manager di Piero?
 - in quali progetti lavora Piero?
 - quali impiegati lavorano nel progetto “IDEA”?
 - quali impiegati lavorano al 100% del loro tempo nel progetto “WIDE”?
- E le **modifiche**:
 - inserire la tupla <4, Luca, 1997-01-01, 2M, 1>
 - modificare il salario di Piero in 4M
 - aumentare il salario di Giorgio del 5%
 - cancellare i dati di Giovanni

Esempio : gestione personale

impiegato

MATR	NOME	DATA-ASS	SALARIO	MATR-MGR
1	Piero	1995-01-01	3 M	2
2	Giorgio	1997-01-01	2,5 M	null
3	Giovanni	1996-07-01	2 M	2

assegnamento

MATR	NUM-PROG	PERC
1	3	50
1	4	50
2	3	100
3	4	100

progetto

NUM-PROG	TITOLO	TIPO
3	Idea	Esprit
4	Wide	Esprit

Esercizi

- in quali tipi di progetti lavora Giovanni?

```
SELECT TIPO
FROM IMPIEGATO AS I,
     ASSEGNAMENTO AS A,
     PROGETTO AS P
WHERE I.MATR=A.MATR
      AND A.NUM-PROG=P.NUM-PROG
      AND NOME='Giovanni'
```

TIPO
Esprit

- chi è il manager di Piero?

```
SELECT Y.NOME
FROM IMPIEGATO AS X, IMPIEGATO AS Y
WHERE X.MATR-MGR=Y.MATR
      AND X.NOME='Piero'
```

NOME
Giorgio

Esercizi

- **modificare il salario di Piero in 4M**

```
UPDATE IMPIEGATO  
SET SALARIO = 4M  
WHERE NOME='Piero'
```

- **aumentare il salario di Giorgio del 5%**

```
UPDATE IMPIEGATO  
SET SALARIO = SALARIO * 1.05  
WHERE NOME='Giorgio'
```