




## Linguaggio SQL

### Introduzione Istruzione SELECT: Fondamenti



## Il linguaggio SQL


- ⊃ Linguaggio per gestire le basi di dati relazionali
  - Structured Query Language
- ⊃ SQL possiede istruzioni per
  - definire lo schema di una base di dati relazionale
  - leggere e scrivere i dati
  - definire lo schema di tabelle derivate
  - definire i privilegi di accesso degli utenti
  - gestire le transazioni



2

## Il linguaggio SQL


- ⊃ Il linguaggio SQL è un linguaggio a *livello di set*
  - gli operatori operano su relazioni
  - il risultato è sempre una relazione
- ⊃ Il linguaggio SQL è *dichiarativo*
  - descrive *cosa fare* e non come fare
  - si pone ad un livello di astrazione superiore rispetto ai linguaggi di programmazione tradizionali



3

## Il linguaggio SQL

- ⊃ Può essere diviso in
  - DML (Data Manipulation Language)
    - linguaggio di manipolazione dei dati
  - DDL (Data Definition Language)
    - linguaggio di definizione della struttura della base di dati



4

## Data Manipulation Language


- ⊃ Interrogazione di una base dati per estrarre i dati di interesse
  - SELECT
- ⊃ Modifica dell'istanza di una base dati
  - inserimento di nuove informazioni in una tabella
    - INSERT
  - aggiornamento di dati presenti nella base dati
    - UPDATE
  - cancellazione di dati obsoleti
    - DELETE



5

## Data Definition Language

- ⊃ Definizione dello schema di una base di dati
  - creazione, modifica e cancellazione di tabelle
    - CREATE, ALTER, DROP TABLE
- ⊃ Definizione di tabelle derivate
  - creazione, modifica e cancellazione di tabelle il cui contenuto è ottenuto da altre tabelle della base dati
    - CREATE, ALTER, DROP VIEW
- ⊃ Definizione di strutture dati accessorie per recuperare efficientemente i dati
  - creazione e cancellazione di indici
    - CREATE, DROP INDEX



6

### BD forniture prodotti

**P**

CodP	NomeP	Colore	Taglia	Magazzino
P1	Maglia	Rosso	40	Torino
P2	Jeans	Verde	48	Milano
P3	Camicia	Blu	48	Roma
P4	Camicia	Blu	44	Torino
P5	Gonna	Blu	40	Milano
P6	Bermuda	Rosso	42	Torino

**FP**

CodF	CodP	Qta
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

**F**

CodF	NomeF	NSoci	Sede
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

### SELECT base (n.1)

↳ Trovare il codice di tutti i prodotti

```
SELECT CodP
FROM P;
```

**P**

CodP	NomeP	Colore	Taglia	Magazzino
P1	Maglia	Rosso	40	Torino
P2	Jeans	Verde	48	Milano
P3	Camicia	Blu	48	Roma
P4	Camicia	Blu	44	Torino
P5	Gonna	Blu	40	Milano
P6	Bermuda	Rosso	42	Torino

**R**

CodP
P1
P2
P3
P4
P5
P6

### SELECT base (n.2)

↳ Trovare il codice dei prodotti forniti da almeno un fornitore

```
SELECT CodP
FROM FP;
```

**FP**

CodF	CodP	Qta
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

**R**

CodP
P1
P2
P3
P4
P5
P6

### SELECT base (n.2)

↳ Trovare il codice dei prodotti *diversi* forniti da almeno un fornitore

```
SELECT DISTINCT CodP
FROM FP;
```

**FP**

CodF	CodP	Qta
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

**R**

CodP
P1
P2
P3
P4
P5
P6

### SELECT (base n.3)

↳ Trovare il codice e il numero di soci dei fornitori di Milano

```
SELECT CodF, NSoci
FROM F
WHERE Sede='Milano';
```

**F**

CodF	NomeF	NSoci	Sede
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

**R**

CodF	NSoci
F2	1
F3	3

### Ordinamento del risultato (n.1)

↳ Trovare il codice dei prodotti e la loro taglia ordinando il risultato in ordine decrescente di taglia

```
SELECT CodP, Taglia
FROM P
ORDER BY Taglia DESC;
```

**P**

CodP	NomeP	Colore	Taglia	Magazzino
P1	Maglia	Rosso	40	Torino
P2	Jeans	Verde	48	Milano
P3	Camicia	Blu	48	Roma
P4	Camicia	Blu	44	Torino
P5	Gonna	Blu	40	Milano
P6	Bermuda	Rosso	42	Torino

**R**

CodP	Taglia
P2	48
P3	48
P4	44
P6	42
P1	40
P5	40

### Join (n.1)

⇒ Trovare il nome dei fornitori che forniscono il prodotto P2

```
SELECT NomeF
FROM F, FP
WHERE F.CodF=FP.CodF
```

NomeTabella.NomeAttributo

DBG

### Join (n.1)

⇒ Trovare il nome dei fornitori che forniscono il prodotto P2

```
SELECT NomeF
FROM F, FP
WHERE F.CodF=FP.CodF
```

← Condizione di join

DBG

### Join (n.1)

⇒ Trovare il nome dei fornitori che forniscono il prodotto P2

```
SELECT NomeF
FROM F, FP
WHERE F.CodF=FP.CodF AND
      CodP='P2';
```

DBG

### Join (n.1)

⇒ Trovare il nome dei fornitori che forniscono il prodotto P2

```
SELECT NomeF
FROM F, FP
WHERE F.CodF=FP.CodF
      AND CodP='P2';
```

↔

```
SELECT NomeF
FROM F, FP
WHERE CodP='P2' AND
      F.CodF=FP.CodF;
```

⇒ Il risultato e l'efficienza sono indipendenti dall'ordine dei predicati nella clausola WHERE

DBG

### Funzioni aggregate

⇒ Una funzione aggregata

- opera su un insieme di valori
- produce come risultato un unico valore (aggregato)
- è indicata nella clausola SELECT
  - non si possono indicare anche attributi non aggregati
  - possono essere richieste più funzioni aggregate contemporaneamente

DBG

### Funzioni aggregate

⇒ Funzioni aggregate disponibili in SQL-2

- COUNT: conteggio degli elementi in un attributo
- SUM: somma dei valori di un attributo
- AVG: media dei valori di un attributo
- MAX: massimo valore di un attributo
- MIN: minimo valore di un attributo

DBG

### Funzione COUNT (n.1)

➤ Trovare il numero di fornitori

```
SELECT COUNT(*)
FROM F;
```

CodF	NomeF	NSoci	Sede
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

➔

R
5

DBG

### Funzione COUNT (n.2)

➤ Trovare il numero di fornitori che hanno almeno una fornitura

```
SELECT COUNT(*)
FROM FP;
```

CodF	CodP	Qta
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

➔

R
12

➤ Conta il numero di forniture, non di fornitori

DBG

### Funzione COUNT (n.2)

➤ Trovare il numero di fornitori che hanno almeno una fornitura

```
SELECT COUNT(CodF)
FROM FP;
```

CodF	CodP	Qta
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

➔

R
12

➤ Conta il numero di forniture, non di fornitori

DBG

### Funzione COUNT (n.2)

➤ Trovare il numero di fornitori che hanno almeno una fornitura

```
SELECT COUNT(DISTINCT CodF)
FROM FP;
```

CodF	CodP	Qta
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

➔

R
4

➤ Conta il numero di fornitori diversi

DBG

### Funzione COUNT

➤ Conteggio del numero di elementi di un insieme

- righe di una tabella
- valori (eventualmente distinti) di uno o più attributi

```
COUNT (<[*] [DISTINCT | ALL] ListaAttributi >)
```

➤ Se l'argomento della funzione è preceduto da **DISTINCT**, conta il numero di valori distinti dell'argomento

DBG

### Funzioni SUM, MAX, MIN, AVG

➤ SUM, MAX, MIN e AVG

- ammettono come argomento un attributo o un'espressione

➤ SUM e AVG

- ammettono solo attributi di tipo numerico o intervallo di tempo

➤ MAX e MIN

- richiedono che l'espressione sia ordinabile
- possono essere applicate anche su stringhe di caratteri e istanti di tempo

DBG

### Raggruppamento

⇒ Per ogni prodotto, trovare la quantità totale di pezzi forniti

FP		
CodF	CodP	Qta
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	100
F4	P6	400
F4	P5	400
F4	P6	100

FP		
CodF	CodP	Qta
F1	P1	300
F2	P1	300
F1	P2	200
F2	P2	400
F3	P2	200
F1	P3	400
F4	P3	200
F1	P4	200
F4	P4	300
F1	P5	100
F4	P5	400
F4	P6	100

R	
CodP	Qta
P1	600
P2	800
P3	600
P4	500
P5	500
P6	100

### Raggruppamento

⇒ Per ogni prodotto, trovare la quantità totale di pezzi forniti

```
SELECT CodP, SUM(Qta)
FROM FP
GROUP BY CodP;
```

### GROUP BY

⇒ Clausola di raggruppamento

GROUP BY *ElencoAttributiDiRaggruppamento*

- l'ordine degli attributi di raggruppamento è ininfluente
- ⇒ Nella clausola SELECT possono comparire solo
  - attributi presenti nella clausola GROUP BY
  - funzioni aggregate

### GROUP BY e SELECT

⇒ Per ogni prodotto, trovare il codice, il nome e la quantità totale fornita

```
SELECT P.CodP, NomeP, SUM(Qta)
FROM P, FP
WHERE P.CodP=FP.CodP
GROUP BY P.CodP, NomeP
```

⇒ Artificio sintattico

- gli attributi univocamente determinati da attributi già presenti nella clausola GROUP BY possono essere aggiunti *senza alterare il risultato*

### Condizione di selezione sui gruppi (n.1)

⇒ Trovare la quantità totale di pezzi forniti per i prodotti per cui sono forniti *in totale* almeno 600 pezzi

```
SELECT CodP, SUM(Qta)
FROM FP
GROUP BY CodP
HAVING SUM(Qta)>=600;
```

⇒ La clausola HAVING permette di specificare condizioni su funzioni aggregate

### Condizione di selezione sui gruppi (n.2)

⇒ Trovare il codice dei prodotti rossi forniti da più di un fornitore

```
SELECT FP.CodP
FROM FP, P
WHERE FP.CodP=P.CodP AND Colore='Rosso'
GROUP BY FP.CodP
HAVING COUNT(*)>1;
```

**Struttura dell'istruzione SELECT**

```
SELECT [DISTINCT] ElencoAttributiDaVisualizzare  
FROM ElencoTabelleDaUtilizzare  
[WHERE CondizioniDiTupla ]  
[GROUP BY ElencoAttributiDiRaggruppamento ]  
[HAVING CondizioniSuAggregati ]  
[ORDER BY ElencoAttributiDiOrdinamento ];
```

