Linguaggio SQL: costrutti avanzati

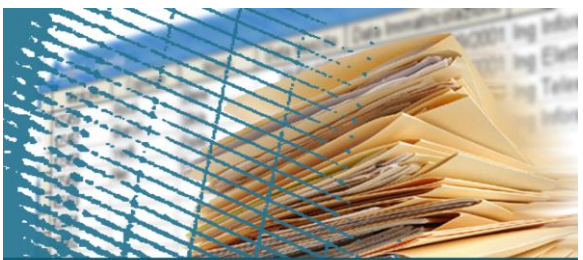
SQL per le applicazioni

DBMG

SQL per le applicazioni

- Introduzione
- Concetto di cursore
- Aggiornabilità
- SQL statico e dinamico
- Embedded SQL
- Call Level Interface (CLI)
- Stored Procedure
- Confronto tra le alternative

DBMG 2



SQL per le applicazioni

Introduzione

DBMG

Esempio applicativo

- Operazioni bancarie
  - operazione di prelievo dal proprio conto corrente mediante bancomat

DBMG 4

Esempio applicativo

- Operazioni bancarie
  - operazione di prelievo dal proprio conto corrente mediante bancomat
  - operazione di prelievo dal proprio conto corrente presso uno sportello bancario

DBMG 5

Prelievo mediante bancomat

- Operazioni svolte
  - verificare la validità del bancomat e del codice PIN
  - selezionare l'operazione di prelievo
  - specificare l'importo richiesto
  - verificare la disponibilità
  - memorizzare il movimento
  - aggiornare il saldo
  - erogare la somma richiesta

DBMG 6

Prelievo mediante bancomat

- Per svolgere molte delle operazioni indicate è necessario accedere alla base di dati
  - esecuzione di istruzioni SQL

DBMG 7

Prelievo mediante bancomat

- Per svolgere molte delle operazioni indicate è necessario accedere alla base di dati
  - esecuzione di istruzioni SQL
- Le operazioni devono essere svolte nell'ordine corretto

DBMG 8

Prelievo presso uno sportello bancario

- Operazioni svolte
  - verificare l'identità dell'utente
  - comunicare l'intenzione di effettuare un prelievo
  - verificare la disponibilità
  - memorizzare il movimento
  - aggiornare il saldo
  - erogare la somma richiesta

DBMG 9

Prelievo presso uno sportello bancario

- Per svolgere molte delle operazioni indicate è necessario accedere alla base di dati
  - esecuzione di istruzioni SQL
- Le operazioni devono essere svolte nell'ordine corretto

DBMG 10

Esempio: operazioni bancarie

- Le operazioni bancarie richiedono di accedere alla base di dati e di modificarne il contenuto
  - esecuzione di istruzioni SQL

DBMG 11

Esempio: operazioni bancarie

- Le operazioni bancarie richiedono di accedere alla base di dati e di modificarne il contenuto
  - esecuzione di istruzioni SQL
    - i clienti e il personale della banca non eseguono direttamente le istruzioni SQL

DBMG 12

Esempio: operazioni bancarie

- Le operazioni bancarie richiedono di accedere alla base di dati e di modificarne il contenuto
  - esecuzione di istruzioni SQL
    - i clienti e il personale della banca non eseguono direttamente le istruzioni SQL
  - un'applicazione nasconde l'esecuzione delle istruzioni SQL

DBMG 13

Esempio: operazioni bancarie

- Le operazioni bancarie richiedono di accedere alla base di dati e di modificarne il contenuto
  - esecuzione di istruzioni SQL
    - i clienti e il personale della banca non eseguono direttamente le istruzioni SQL
  - un'applicazione nasconde l'esecuzione delle istruzioni SQL
- La corretta gestione delle operazioni bancarie richiede di eseguire una sequenza precisa di passi
  - un'applicazione permette di specificare l'ordine corretto di esecuzione delle operazioni

DBMG 14

Applicazioni e SQL

- Per risolvere problemi reali non è quasi mai sufficiente eseguire singole istruzioni SQL

DBMG 15

Applicazioni e SQL

- Per risolvere problemi reali non è quasi mai sufficiente eseguire singole istruzioni SQL
- Servono applicazioni per
  - acquisire e gestire i dati forniti in ingresso
    - scelte dell'utente, parametri
  - gestire la logica applicativa
    - flusso di operazioni da eseguire
  - restituire i risultati all'utente in formati diversi
    - rappresentazione non relazionale dei dati
      - documento XML
    - visualizzazione complessa delle informazioni
      - grafici, report

DBMG 16

Integrazione tra SQL e applicazioni

- Le applicazioni sono scritte in linguaggi di programmazione tradizionali di alto livello
  - C, C++, Java, C#, ...
  - il linguaggio è denominato *linguaggio ospite*
- Le istruzioni SQL sono usate nelle applicazioni per accedere alla base di dati
  - interrogazioni
  - aggiornamenti

DBMG 17

Integrazione tra SQL e applicazioni

- È necessario integrare il linguaggio SQL e i linguaggi di programmazione
  - SQL
    - linguaggio dichiarativo
  - linguaggi di programmazione
    - tipicamente procedurali

DBMG 18

Conflitto di impedenza

- Conflitto di impedenza
  - le interrogazioni SQL operano su una o più tabelle e producono come risultato una tabella
    - approccio set oriented

DBMG 19

Conflitto di impedenza

- Conflitto di impedenza
  - le interrogazioni SQL operano su una o più tabelle e producono come risultato una tabella
    - approccio set oriented
  - i linguaggi di programmazione accedono alle righe di una tabella leggendole *una a una*
    - approccio tuple oriented

DBMG 20

Conflitto di impedenza

- Conflitto di impedenza
  - le interrogazioni SQL operano su una o più tabelle e producono come risultato una tabella
    - approccio set oriented
  - i linguaggi di programmazione accedono alle righe di una tabella leggendole *una a una*
    - approccio tuple oriented
- Soluzioni possibili per risolvere il conflitto
  - uso di cursori
  - uso di linguaggi che dispongono in modo naturale di strutture di tipo "insieme di righe"

DBMG 21

SQL e linguaggi di programmazione

- Tecniche principali di integrazione
  - Embedded SQL
  - Call Level Interface (CLI)
    - SQL/CLI, ODBC, JDBC, OLE DB, ADO.NET, ..
  - Stored procedure

DBMG 22

SQL e linguaggi di programmazione

- Tecniche principali di integrazione
  - Embedded SQL
  - Call Level Interface (CLI)
    - SQL/CLI, ODBC, JDBC, OLE DB, ADO.NET, ..
  - Stored procedure
- Classificabili in
  - client side
    - embedded SQL, call level interface
  - server side
    - stored procedure

DBMG 23

Approccio client side

- L'applicazione
  - è esterna al DBMS
  - contiene tutta la logica applicativa
  - richiede al DBMS di eseguire istruzioni SQL e di restituirne il risultato
  - elabora i dati restituiti

DBMG 24

Approccio server side

- L'applicazione (o una parte di essa)
  - si trova nel DBMS
  - tutta o parte della logica applicativa si sposta nel DBMS

DBMG 25

Approccio client side vs server side

- Approccio client side
  - maggiore indipendenza dal DBMS utilizzato
  - minore efficienza
- Approccio server side
  - dipendente dal DBMS utilizzato
  - maggiore efficienza

DBMG 26

SQL per le applicazioni

Concetto di cursore

DBMG

Conflitto di impedenza

- Principale problema di integrazione tra SQL e linguaggi di programmazione
  - le interrogazioni SQL operano su una o più tabelle e producono come risultato una tabella
    - approccio set oriented
  - i linguaggi di programmazione accedono alle righe di una tabella leggendole *una a una*
    - approccio tuple oriented

DBMG 28

Cursori

- Se un'istruzione SQL restituisce una sola riga
  - è sufficiente specificare in quale variabile del linguaggio ospite memorizzare il risultato dell'istruzione

DBMG 29

Cursori

- Se un'istruzione SQL restituisce una sola riga
  - è sufficiente specificare in quale variabile del linguaggio ospite memorizzare il risultato dell'istruzione
- Se un'istruzione SQL restituisce una tabella (insieme di tuple)
  - è necessario un metodo per leggere (e passare al programma) una tupla alla volta dal risultato dell'interrogazione
    - uso di un *cursore*

DBMG 30

DB 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

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

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

Esempio n.1

- Visualizzare nome e numero di soci del fornitore con codice F1

Esempio n.1

- Visualizzare nome e numero di soci del fornitore con codice F1

```
SELECT NomeF, NSoci
FROM   F
WHERE CodF='F1';
```


Esempio n.1

- Visualizzare nome e numero di soci del fornitore con codice F1

```
SELECT NomeF, NSoci
FROM   F
WHERE CodF='F1';
```

- L'interrogazione restituisce *al massimo* una tupla

NomeF	NSoci
Andrea	2

Esempio n.1

- Visualizzare nome e numero di soci del fornitore con codice F1

```
SELECT NomeF, NSoci
FROM   F
WHERE CodF='F1';
```

- L'interrogazione restituisce *al massimo* una tupla

NomeF	NSoci
Andrea	2

- È sufficiente specificare in quali variabili del linguaggio ospite memorizzare la tupla selezionata

Esempio n.2

- Visualizzare nome e numero di soci dei fornitori di Torino

Esempio n.2

- Visualizzare nome e numero di soci dei fornitori di Torino

```
SELECT NomeF, NSoci
FROM   F
WHERE Sede='Torino';
```


Esempio n.2

- Visualizzare nome e numero di soci dei fornitori di Torino

```
SELECT NomeF, NSoci
FROM   F
WHERE Sede='Torino';
```

- L'interrogazione restituisce un insieme di tuple

NomeF	NSoci
Andrea	2
Gabriele	2

Esempio n.2

- Visualizzare nome e numero di soci dei fornitori di Torino

```
SELECT NomeF, NSoci
FROM   F
WHERE Sede='Torino';
```

- L'interrogazione restituisce un insieme di tuple

NomeF	NSoci
Andrea	2
Gabriele	2

- È necessario definire un *cursore* per leggere separatamente le tuple del risultato

Esempio n.2

- Definizione del cursone mediante la sintassi del linguaggio PL/SQL di Oracle

```
CURSOR FornitoriTorino IS
SELECT NomeF, NSoci
FROM   F
WHERE Sede='Torino';
```


Cursore

- Il cursore permette di leggere singolarmente le tuple che fanno parte del risultato di un'interrogazione
  - deve essere associato a un'interrogazione specifica
- Ogni interrogazione SQL che può restituire un insieme di tuple *deve essere associata* a un cursore

Cursore

- Non necessitano di cursori
  - le interrogazione SQL che restituiscono al massimo una tupla
    - selezioni sulla chiave primaria
    - operazioni di aggregazione senza clausola GROUP BY
  - i comandi di aggiornamento e di DDL
    - non generano tuple come risultato

SQL per le applicazioni

SQL statico e dinamico

DBMG

SQL statico

- Le istruzioni SQL da eseguire sono note durante la scrittura dell'applicazione
  - è nota la definizione di ogni istruzione SQL
  - le istruzioni possono contenere variabili
    - il valore delle variabili è noto solo durante l'esecuzione dell'istruzione SQL

DBMG 44

SQL statico

- La definizione delle istruzioni SQL avviene durante la scrittura dell'applicazione
  - semplifica la scrittura dell'applicazione
    - è nota a priori la struttura di interrogazioni e risultati

DBMG 45

SQL statico

- La definizione delle istruzioni SQL avviene durante la scrittura dell'applicazione
  - semplifica la scrittura dell'applicazione
    - è nota a priori la struttura di interrogazioni e risultati
  - rende possibile l'ottimizzazione a priori delle istruzioni SQL
    - durante la fase di compilazione dell'applicazione, l'ottimizzatore del DBMS
      - compila l'istruzione SQL
      - crea il piano di esecuzione

DBMG 46

SQL statico

- La definizione delle istruzioni SQL avviene durante la scrittura dell'applicazione
  - semplifica la scrittura dell'applicazione
    - è nota a priori la struttura di interrogazioni e risultati
  - rende possibile l'ottimizzazione a priori delle istruzioni SQL
    - durante la fase di compilazione dell'applicazione, l'ottimizzatore del DBMS
      - compila l'istruzione SQL
      - crea il piano di esecuzione
    - queste operazioni non sono più necessarie durante l'esecuzione dell'applicazione
      - esecuzione più efficiente

DBMG 47

SQL dinamico

- Le istruzioni SQL da eseguire *non* sono note durante la scrittura dell'applicazione
  - le istruzioni SQL sono definite dinamicamente dall'applicazione in fase di esecuzione
    - dipendono dal flusso applicativo eseguito
  - le istruzioni SQL possono essere fornite in ingresso dall'utente

DBMG 48

SQL dinamico

- La definizione a tempo di esecuzione delle istruzioni SQL
  - permette di definire applicazioni più complesse
    - offre una maggiore flessibilità

DBMG 49

SQL dinamico

- La definizione a tempo di esecuzione delle istruzioni SQL
  - permette di definire applicazioni più complesse
    - offre una maggiore flessibilità
  - rende più difficile la scrittura delle applicazioni
    - durante la scrittura non è noto il formato del risultato dell'interrogazione

DBMG 50

SQL dinamico

- La definizione a tempo di esecuzione delle istruzioni SQL
  - permette di definire applicazioni più complesse
    - offre una maggiore flessibilità
  - rende più difficile la scrittura delle applicazioni
    - durante la scrittura non è noto il formato del risultato dell'interrogazione
  - rende l'esecuzione meno efficiente
    - durante ogni esecuzione dell'applicazione, è necessario compilare e ottimizzare ogni istruzione SQL

DBMG 51

SQL dinamico

- Se la stessa interrogazione dinamica deve essere eseguita più volte nella *stessa sessione* di lavoro
  - è possibile ridurre i tempi di esecuzione
    - si effettua una sola volta la compilazione e la scelta del piano di esecuzione
    - si esegue l'interrogazione più volte (con valori diversi delle variabili)

DBMG 52