

Linguaggi per Database

Prof. Francesco Gobbi

I.I.S.S. Galileo Galilei - Ostiglia (MN)
Materia: Informatica

25 settembre 2024

Linguaggi per Database

Le potenzialità e peculiarità del DBMS vengono attivate dall'utente usando appositi comandi, che costituiscono a tutti gli effetti un **linguaggio attraverso il quale l'utente può comunicare con il sistema di elaborazione che gestisce il database.**

Tipi di Linguaggi

I comandi che il DBMS mette a disposizione possono essere classificati nelle seguenti categorie di linguaggi:

- ▶ **Linguaggio per la descrizione dei dati (DDL - Data Definition Language):**
 - ▶ Descrive i dati, le tabelle, le viste, le associazioni tra tabelle, i vincoli di integrità, e i controlli relativi alla sicurezza.
 - ▶ Permette di ordinare al DBMS la creazione della struttura fisica del database.
 - ▶ Consente di definire sottoschemi relativi alle applicazioni e eliminare tabelle o viste già esistenti.
- ▶ **Linguaggio per il trattamento o manipolazione dei dati (DML - Data Manipulation Language):**
 - ▶ Consente le operazioni di accesso per inserimenti, modifiche o cancellazioni.
- ▶ **Linguaggio per le interrogazioni (QL - Query Language):**
 - ▶ Permette di ritrovare i dati di interesse sulla base di criteri di ricerca richiesti dall'utente.

Dettagli sul DDL

- ▶ Il **DDL** comprende la possibilità di **generare in modo automatico**:
 - ▶ Tabelle e la loro attribuzione
 - ▶ Maschere video per facilitare inserimento, modifica e consultazione dei dati
- ▶ L'output generato dal DDL è memorizzato e reso leggibile dai programmi che effettuano l'accesso ai dati.

Linguaggi per Database Relazionali

Lo sviluppo delle tecniche di gestione delle basi di dati ha dato vita a linguaggi formati da comandi specifici, per consentire un facile uso delle prestazioni del DBMS per basi di dati relazionali (**RDBMS**).

Linguaggi Procedurali e Non Procedurali

- ▶ Con i linguaggi tradizionali di programmazione (Cobol, C), è possibile richiamare comandi DBMS all'interno di programmi.
- ▶ Con i linguaggi **non procedurali** ma dichiarativi, come **MySQL**, l'utente non deve conoscere dettagli tecnici sull'archiviazione dei dati.
- ▶ MySQL combina le funzioni di **DDL**, **DML**, e **QL**.

Linguaggio MySQL: DDL - Data Definition Language

DDL permette di definire la struttura della base di dati.

► Esempio di creazione di una tabella:

```
1 CREATE TABLE studenti (  
2     id INT PRIMARY KEY,  
3     nome VARCHAR(100),  
4     cognome VARCHAR(100),  
5     data_nascita DATE  
6 );
```

Linguaggio MySQL: DML - Data Manipulation Language

DML consente l'inserimento, la modifica e la cancellazione dei dati.

- ▶ Esempio di inserimento di un record:

```
1  INSERT INTO studenti (id, nome, cognome,  
    data_nascita)  
2  VALUES (1, 'Mario', 'Rossi', '2005-04-23');
```

- ▶ Esempio di aggiornamento di un record:

```
1  UPDATE studenti  
2  SET nome = 'Marco'  
3  WHERE id = 1;
```

- ▶ Esempio di cancellazione di un record:

```
1  DELETE FROM studenti  
2  WHERE id = 1;
```


Linguaggio MySQL: QL - Query Language

QL permette di interrogare il database per estrarre informazioni.

► Esempio di interrogazione:

```
1 SELECT nome, cognome
2 FROM studenti
3 WHERE data_nascita > '2005-01-01';
```

Linguaggi per i Database

- ▶ I **linguaggi per database relazionali** si basano su una **visione tabellare dei dati**, operando su **gruppi di righe o sull'intera tabella** con una sola richiesta.
- ▶ Le **operazioni sono semplificate per mettere in connessione tabelle diverse e per presentare i risultati delle interrogazioni** in modo chiaro e comprensibile.
- ▶ Nei software DBMS, esistono interfacce utente grafiche che permettono l'interazione tramite menu, sottomenù e icone.
- ▶ Sono disponibili messaggi di aiuto contestuale per fornire spiegazioni immediate sulle operazioni da eseguire.

Utenti di un DBMS

I principali utenti di un Sistema di Gestione di Basi di Dati (DBMS) sono:

- ▶ **DBA (Database Administrator):**

Responsabile della gestione, manutenzione e sicurezza del database. *Esempio:* Configurare backup e ripristino dei dati.

- ▶ **Programmatori:**

Sviluppano le applicazioni che interagiscono con il database, scrivendo query SQL e interfacce per l'accesso ai dati.

Esempio: Creare un'applicazione web che consente agli utenti di consultare un catalogo di prodotti.

- ▶ **Utenti finali:**

Utilizzano le applicazioni per interagire con i dati, eseguendo operazioni di inserimento, aggiornamento o consultazione.

Esempio: Un cliente che consulta i prodotti disponibili su un sito di e-commerce.

Definizione di Transazione

Transazione: Un insieme di operazioni di interrogazione o di modifica del database che devono essere eseguite unitariamente, come se fossero un'unica operazione.

Esempi di Transazioni:

- ▶ Trasferimento di fondi da un conto a un altro
- ▶ Prelievo da uno sportello Bancomat
- ▶ Acquisto con carta di credito
- ▶ Prenotazione di un volo

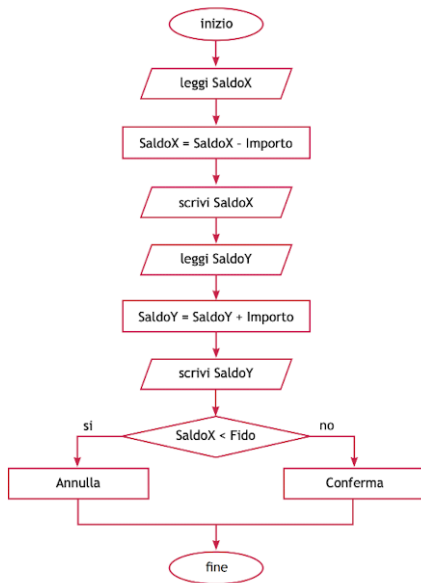
Esempio di Transazione

Si consideri il trasferimento di fondi dal conto di X al conto di Y.
Variabili utilizzate:

- ▶ **SaldoX e SaldoY:** valore del saldo dei conti X e Y
- ▶ **Importo:** entità del trasferimento
- ▶ **Fido:** valore negativo che rappresenta il credito concesso su un conto

Esempio di Transazione

Diagramma di flusso:



Codice della Transazione

Il comportamento della transazione può essere rappresentato con il seguente pseudocodice:

```
1      begin\_transaction; \\  
2          read (SaldoX; \\  
3          SaldoX = SaldoX - Importo; \\  
4          write (SaldoX; \\  
5          read (SaldoY; \\  
6          SaldoY = SaldoY + Importo; \\  
7          write (SaldoY; \\  
8          if( SaldoX < Fido ) then \\  
9              rollback; \\  
10         else \\  
11             commit; \\  
12         endif; \\  
13     end\_transaction \\  

```

Proprietà ACID("acide") delle transazioni

Atomicità:

- ▶ Una transazione è un'entità atomica indivisibile.
- ▶ Il sistema di gestione della sicurezza garantisce che, se una parte della transazione fallisce, tutto il resto fallisce e si ripristina lo stato precedente.

Consistenza:

- ▶ Le transazioni non devono violare i vincoli di integrità dei dati.
- ▶ La consistenza è gestita dal DBMS con procedure opportune.

Isolamento:

- ▶ Gli effetti di una transazione devono essere indipendenti da altre transazioni eseguite in contemporanea.
- ▶ L'isolamento è gestito dal controllo di concorrenza nel DBMS.

Persistenza:

- ▶ I risultati di una transazione, una volta confermati, devono essere memorizzati in modo permanente nel database.
- ▶ Anche in caso di guasto del sistema, i dati devono rimanere disponibili.