01/03/2022: Introduzione alle basi di dati e concetti base/Capitolo 1

Le *basi di dati* sono un insieme consistente, persistente, organizzato e condiviso dei dati utilizzati. Al di là di questo, si parla di dati, entità presente prima di ogni elaborazione e che devono essere interpretate sotto forma di informazioni. I dati stessi codificano le informazioni dando rappresentazioni più precise di informazione e conoscenza, magari per analisi o altri scopi.

Essi sono gestiti dai *DBMS (Database Management System)* garantendo privacy, affidabilità, efficienza ed efficacia. Essa è inoltre persistente (in grado di resistere a vari problemi e perdite di dati) e condiviso tra più utenti, dando ai singoli utenti permessi diversi a seconda dello scopo. In generale possono contenere molti dati, ammettiamo per esempio terabyte di dati o miliardi di record (*grande*), indipendente dalla singole esecuzione di un programma (*persistente*), dando appunto agli utenti varie porzioni della base di dati acceduta (*condivisa*).

Naturalmente si deve garantire ridondanza, informazioni ripetute e che devono essere mantenute in maniera sicura per evitare incoerenze. A tale scopo, attenzione al controllo della concorrenza, magari anche con utilizzo di strutture come lock o semafori, assieme a meccanismi specifici di autenticazione.

L’uso dei DBMS cerca di garantire *l’affidabilità*, resistendo a malfunzionamenti hardware/software, gestendo le transazioni, quindi un insieme di operazioni che modificano lo stato di una base di dati. Prendiamo l’esempio di trasferimento di fondi, classico esempio del prelievo/versamento, quindi rimozione e/o aggiunta tra basi di dati. L’ordine è fondamentale per non invalidare quanto effettivamente fatto.

Naturalmente le risorse devono essere bastevoli tali che il sistema informativo stesso possa essere considerato efficiente. A noi interessa di più *l’efficacia* rispetto all’efficienza.

I dati sono organizzati normalmente secondo modelli di dati utilizzati per organizzare dati di interesse e quindi descriverne la dinamica, quindi con il modello relazionale, creando degli insiemi di record omogenei. Ogni riga è una *tupla* (prodotto cartesiano di n domini, matematicamente parlando), combinando tra di loro tutti i possibili elementi. Un esempio è:

Immagine che contiene tavolo

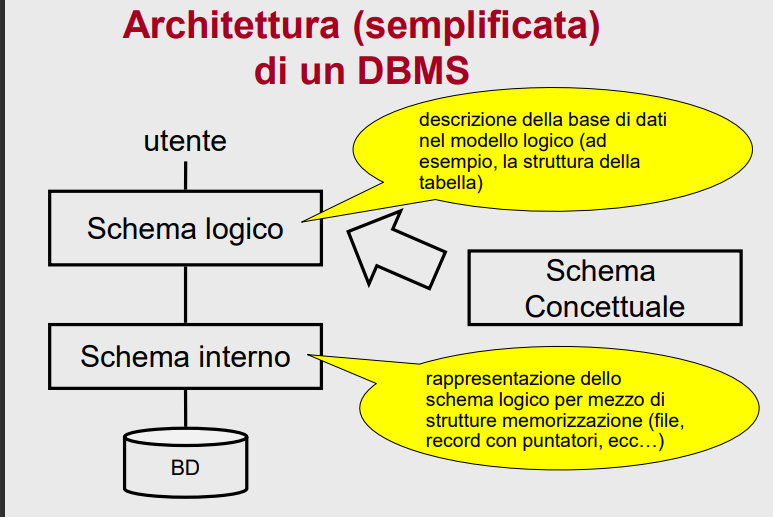
Descrizione generata automaticamente

In ogni base di dati vi sono lo *schema,* che descrive la struttura di un database (aspetto intensionale, quindi che non cambia nel tempo, es. intestazioni delle tabelle) e l’*istanza*, con i valori attualmente utilizzati e che possono cambiare anche molto rapidamente (ad es. il corpo di ciascuna tabella).

In generale avremo due tipi di modelli:

1. *logici*, adottati dai DBMS organizzando i dati a livello logico, sono utilizzati dai programmi e sono indipendenti dalle strutture fisiche. Possiamo per esempio enunciare modelli di tipo relazionale, reticolare, gerarchico, ad oggetto, XML, ecc.
2. *concettuali*, rappresenta i dati in modo indipendente, cercando di descrivere i concetti reali e spesso usati nelle fasi preliminari di progettazione. Il modello in questo senso più diffuso è il cosiddetto *Entity-Relationship.*

A tale scopo definiamo così l’archittettura, intendendo come utente ciò che viene creato dalla macchina finale per gestire tutto ciò; più in generale segue lo schema ANSI/SPARC; buon esempio al link https://simonebortolin.gitbooks.io/informatica-quinta/content/modello\_ansi\_sparc\_dei\_dbms.html



A noi interessa soprattutto il livello logico piuttosto che quello fisico, entrambi indipendente uno dall’altro. Lo scherma esterno (schema logico e interno visibile da altre entità). Se io volessi verificare una serie di informazioni, esso viene fatto tramite una *vista*, vedendo una certa porzione di database in un certo momento con una query, ad esempio, e ciò viene fatto proprio per semplificarle.

Altro contributo all’efficacia delle basi di dati tramite linguaggi testuali interattivi (SQL), comandi immersi in un linguaggio ospite (Pascal, Java, C, interagendo con SQL), tramite interfacce amichevoli (quindi non testuali).

Un esempio di interrogazione fatta su SQL:



Esempi anche di apertura connessione ed esecuzione query, caso Java, può essere il seguente frammento di codice:

Connection conn=

DriverManager.getConnection(“jdbc:mysql://localhost/EMP“,

“user“,“pass“);

Statement stmt= conn.createStatement();

ResultSet rs =

stmt.executeQuery("SELECT id, first, last, age FROM Employees");

La nostra interazione grafica viene fatta con POSTGRES. Nel caso pratico, anche cambiando sistemi di interazioni con le basi di dati, è possibile agire ugualmente sul sistema. La gestione è centralizzata e ha possibilità di standardizzazione purché si usino linguaggi simili al SQL standard. Questo funziona bene nelle grandi organizzazioni, ma in caso di pochi utenti vanno anche bene file semplici.

Le operazioni possibili su una base di dati avvengono solitamente tramite DML (Data Manipulation Language), interrogando e/o aggiornando istanze delle basi di dati e DDL (Data Definition Language), definendo schermi (logixi, esterni, fisici) per varie operazioni.