Gestione dei dati

Dato, definizione: 'In informatica, la singola informazione codificabile o codificata'.

- Le informazioni vengono rappresentate tramite dati.

L'evoluzione della tecnologia permette oggi di raccogliere una mole di dati, continuamente e in maniera pervasiva (e.g. smart-watch, cellulare). Questo ha solo portato alle estreme conseguenze una necessità, la corretta e efficiente gestione dati, che è sempre stata fondamentale per la buona realizzazione di qualsiasi sistema informativo.

- Prima avvertita solo nelle grandi compagnie

master copy COPISTERIA

Sistema Informativo

Componente di una organizzazione che *gestisce le informazioni* di interesse (cioé utilizzate per il perseguimento degli scopi dell'organizzazione).

Ogni organizzazione ha un sistema informativo, eventualmente non esplicitato nella struttura

Gestione delle informazioni:

- Raccolta, acquisizione
- Archiviazione, conservazione
- Elaborazione, trasformazione, produzione
- Distribuzione, comunicazione, scambio

Sistema Informativo e Automazione

Il Sistema Informativo è un sistema atto a organizzare e gestire i dati in maniera tale da permettere la loro facile elaborazione/memorizzazione per ottenere informazioni (normale operatività/dati aggregati).

I sistemi informativi venivano realizzati anche prima dell'invenzione e diffusione dei calcolatori elettronici, infatti esistono organizzazioni la cui ragion d'essere è la gestione di informazioni (Banche, servizi anagrafici) ed esistono da prima dei calcolatori.

Per la porzione automatizzata del sistema informativo, al giorno d'oggi viene usato il termine Sistema Informatico che gestisce informazioni con tecnologia informatica.

Database Management System - DBMS

All'interno del sistema informativo, la collezione dei dati, utilizzati per rappresentare le informazioni di interesse, è chiamata Base di Dati o Database.

- Compito della base di dati è non solo di memorizzare i dati ma di rappresentare le relazioni tra di essi (ad esempio fattura e il suo importo).

All'interno del sistema informativo/informatico il software atto specificatamente a gestire i dati è detto Sistema di Gestione della basi di dati o Database Management System.

Tradizionalmente adottato nei sistemi informativi di grandi dimensioni (solitamente composti da più programmi), oggi adottato anche da sistemi più semplici:

- Mole dati crescenti;
- DBMS anche a basso prezzo/semplicità del loro utilizzo.

master COPY

Prospettiva storica

Memorizzare e manipolare dati è stato uno degli obiettivi principali dai primi giorni dei computer.

Il primo DBMS per uso generico (Integrated Data Store), fu progettato da Charles Bachman alla General Electric nei primi anni '60 e costituì la base per il *modello di dati reticolare.*

Sul finire degli anni '60, l'IBM sviluppò il DBMS Information Management System (IMS) che fornì la base per una struttura alternativa di rappresentazione dei dati chiamata *modello di dati gerarchico.*

Nel 1970 Edgar Codd propose una nuova struttura di rappresentazione dei dati chiamata *modello di dati relazionale.* Negli anni '80 il modello relazionale consolidò la sua posizione di dominanza.

Il linguaggio di interrogazione *SQL* per basi di dati relazionali, sviluppato come parte del progetto System R dell'IBM, è divenuto il linguaggio di interrogazione standard.

Archivio basato su Files / I

L'approccio classico usato dal/dai programma/i che compongono il sistema informativo per la gestione delle informazioni è un archivio basato su files. Ogni programma ha accesso al file system gestito dal sistema operativo per creare uno o più files (archivi).

• Ogni file è un insieme di registrazioni (record) all'interno dei quali sono memorizzati i dati elementari (attributi e campi).

La condivisione di dati tra più programmi può essere fatto tramite l'uso di file condivisi.

Uso file - Svantaggi

I file possono avere diversi formati incompatibili tra di loro, i programmi si devono adeguare a diverse convenzioni anche a distanza di parecchio tempo. Questo rende la condivisione dei dati attraverso applicazioni differenti difficoltosa.

I dati se non memorizzati su file condivisi sono replicati con spreco di risorse di memorizzazione e possibili problemi legati a inconsistenze.

L'accesso a file in condivisione porta a dover gestire la concorrenza con soluzioni ad-hoc (specialmente se due o più programmi vogliono modificarne il contenuto).

Approccio basato su DBMS

L'approccio basato su DBMS invece va oltre l'uso di file locali gestiti dalle singole applicazioni tramite l'adozione di un sistema di gestione dei dati che risulta indipendente dalle applicazioni e specializzato in tale funzione.

I dati non sono gestiti (memorizzati) dalle singole applicazioni ma da un DBMS che offre un'interfaccia comune a tutte le applicazioni e che si interpone fra le applicazioni e la memoria di massa.

I dati non appartengono ad una singola applicazione, ma esse vi accedono attraverso il DBMS.

DBMS - Vantaggi

Le basi di dati sono condivise: le applicazioni e gli utenti accedono a dati comuni evitando la realizzazione adhoc di soluzioni come la condivisione di file tra applicazioni diverse. Questo aiuta a:

• Ridurre la ridondanza:

una base di dati centralizzata permette di ridurre la replica della stessa informazione che si avrebbe se le diverse applicazioni gestissero i dati tramite file locali;

• Ridurre l'inconsistenza:

l'eliminazione della presenza di varie copie dello stesso dato elimina la possibilità di inconsistenze, la gestione attraverso una componente specializzata permette di introdurre controlli sui dati per garantirne la consistenza (Vincoli di consistenza).

DBMS - Vantaggi

DBMS sono componenti software specializzati nel gestire grandi quantità di dati e implementano procedure basate sulle best-practices (solitamente non implementate nelle soluzioni basate su file) per la gestione di:

• *Efficacia e efficienza:*

Le tecniche di memorizzazione adottate permettono di migliorare le prestazioni di memorizzazione (spazio e tempo) e accesso alle informazioni (che altrimenti dovrebbero essere implementate in ogni programma) (efficienza) e consentono di semplificare lo sviluppo di applicazioni (efficacia);

• *Grosse dimensioni dei dati:*

Le BD possono avere grosse dimensioni, per cui i dati risiedono in memoria secondaria e devono essere acceduti in modo efficiente (indici, strutture di memorizzazione e architetture hardware);

• *Persistenza*:

Le BD sono persistenti, ossia hanno un tempo di vita che non è limitato a quello delle applicazioni che le utilizzano, a differenza dei dati delle applicazioni. Il DBMS garantisce la persistenza dei dati a prescindere dall'utilizzo delle applicazioni;

• *Affidabilità:*
Tecniche di salvaguardia e verifica dell'integrità dei dati in caso di malfunzionamenti hardware e software (crash recovery) sono solitamente implementate (Backup e informazioni di controllo);
• *Concorrenza:*
I sistemi DBMS implementano delle metodologie per garantire un corretto accesso concorrente ai dati, minimizzandone l'impatto sulle prestazioni di accesso (esempio limitando i tempi di attesa in seguito alla mutua esclusione su un dato che non è implementata su file);
• *Privatezza:*
Tecniche di sicurezza per garantire accesso ristretto sono implementate in modo da garantire a ciascun utente accesso solo al sottoinsieme dei dati a cui è autorizzato.

L'utilizzo di una componente per la gestione dei dati di diverse applicazioni permette di migliorare il processo di realizzazione delle applicazioni:
• *Riduzione del tempo di sviluppo*: invece di implementare le funzionalità di gestione dei dati ogni applicazione si appoggia su quelle fornite dal DBMS;

• *Semplificazione e standardizzazione dello sviluppo:* il processo di realizzazione delle applicazioni viene semplificato dato che la memorizzazione e la gestione dei dati è demandato ad una componente con la quale l'applicazione interagisce tramite un'interfaccia standard:
* L'interfaccia di interazione col DB è standard
*La struttura dell'applicazione segue tale standard
*Facilità di modifiche
*Non occorre, ad esempio, conoscere la codifica dei dati su file.

DBMS - Svantaggi

• *Costo*
Specie i DBMS commerciali sono costosi e a questi si aggiungono costi indiretti
(Risorse hardware, Sviluppo/conversione di applicazioni, Formazione del personale/ Acquisizione nuovo personale (DBA), Riorganizzazione dei processi aziendali (es: cartaceo -> elettronico)).

• *Integrazione di servizi non sempre utili*

Tali servizi possono non essere scorporabili ed implicare aumentate competenze rispetto alle esigenze e riduzione di prestazioni.
In generale, per applicazioni con pochi utenti, senza accessi concorrenti e stabili nel tempo possono essere utilizzate soluzioni con file ordinari e non DBMS, tuttavia la disponibilità di DBMS a basso costo (nullo) ne sta estendendo l'uso.

DBMS – Prodotti commerciali

I DBMS non sono software ad-hoc ma sono solitamente prodotti commerciali.
Le aziende produttrici solitamente coinvolgono decine di sviluppatori con progetti che durano svariati anni al fine di garantire la solidità del software richiesta dagli utenti finali.
Esempi commerciali: Oracle, IBM DB2, Microsoft SQL Server.
In aggiunta esistono DBMS open-source che vengono sviluppati e gestiti da una comunità di sviluppatori e sono liberamente accessibili. Questi DBMS pur avendo molte delle funzionalità offerte dai prodotti commerciali non sono adatti a gestire grandi quantità di dati, ma possono invece essere usati in progetti più piccoli.
Esempi open-source: MySQL, MariaDB, Firebird SQL, PostgreSQL.

Schema e Modello dei Dati

I DBMS non sono progettati per gestire un unico caso d'uso, al contrario sono software in grado di gestire dati eterogenei (differenti basi di dati). Dopo aver installato il DBMS, al fine di creare e gestire la corrispondente base di dati, uno schema dei dati deve essere fornito al DBMS.

Lo schema viene costruito secondo un modello di dati ben definito.
• Un Modello di Dati è una collezione di costrutti usati per descrivere i dati di interesse e la loro struttura (relazioni), in modo che essa risulti *comprensibile* ad un calcolatore.
Con il modello dei dati si descrivono lo schema dei dati, le loro relazioni e i vincoli di consistenza che devono essere applicati sugli stessi.
• Il modello dei dati specifica come i dati rappresentano l'informazione.
(Ad esempio tramite un determinato modello di dati si può descrivere una base di dati dell'amministrazione universitaria nella quale ci starà il dato che descrive gli esami e il relativo voto).
Tramite questo schema dei dati si fornisce al DBMS una rappresentazione dei dati (della base di dati), in modo tale da permetterne la gestione.

Modello dei Dati Relazionale

Esistono diverse tipologie di modelli (logici) definiti nel tempo: Modello Gerarchico, Modello a Oggetti, Modello Reticolare.
Il modello attualmente più diffuso è il *Modello Relazionale.*
Il modello relazionale descrive lo schema di una base di dati attraverso la specifica delle relazioni che i dati stessi hanno tra di loro. Il costrutto base del modello relazionale è la ***relazione o tabella relazionale*** che può essere pensata come un insieme di record.
Nel modello relazionale *ciascuna relazione* viene identificata da un *nome*. I *campi della relazione* vengono identificati attraverso un *nome e il tipo*. I *legami fra tabelle* vengono espressi per *valori*.

Schemi e Istanze

La struttura di un database, descritta tramite il modello dei dati è detta schema.
Lo schema è frutto del processo di progettazione della base di dati e raramente cambia durante la vita del DBMS. Un cambiamento dello schema è frutto o di una errata progettazione, o di un cambiamento della "realtà".
L'istanza di un database invece è l'insieme delle informazioni contenute in un database in un certo istante.
L'istanza cambia continuamente nel tempo
L'inserimento/modifica/cancellazione dei dati non cambia comunque la struttura generale (lo schema).

Astrazioni sui dati

Altro vantaggio dei DBMS è il loro supporto su astrazioni sui dati.
La descrizione dello schema di una base di dati attraverso il modello relazionale ad esempio fornisce un'astrazione sui dati rispetto a come, ad esempio, questi verranno memorizzati realmente su file. Tramite diversi livelli di astrazioni, il DBMS maschera dettagli implementativi, al fine di semplificare l'interazione di diversi tipi di utente (alcuni anche non esperti) con il sistema.

Livelli di astrazione

I sistemi DBMS sono caratterizzati da tre livelli di astrazione: Fisico, Logico e delle Viste. Per ciascun livello di astrazione esiste uno schema.
• *LIVELLO LOGICO:*
È il livello di astrazione rappresentato dal modello dei dati (e.g. dal modello relazionale). Associato al livello logico abbiamo lo schema logico che fornisce una descrizione dell'intera base di dati per mezzo del modello logico adottato dal DBMS. Sono tralasciati tutti i dettagli implementativi.
• *LIVELLO FISICO:*
È il livello più basso di astrazione, quello interno del database. Lo schema interno associato alla rappresentazione dei dati al livello fisico descrive come i dati rappresentati nello schema logico sono memorizzati nelle strutture fisiche di memorizzazione.
• *LIVELLO ESTERNO (DELLE VISTE):*
È il livello di astrazione più alto che permette di esporre agli utenti finali delle visioni (viste) parziali o successivamente elaborate del livello logico. Vengono definite quando non tutti gli utenti hanno necessità

(o diritto) di conoscere tutta la struttura completa logica del database ma solo parti di effettivo interesse.
Lo schema esterno costituisce la descrizione della porzione della base di dati di interesse.
Le viste possono essere usate per regolare meglio il controllo degli accessi o per calcolare dinamicamente nuovi dati a partire da quelli memorizzati nel DB evitando ridondanza.

Indipendenza dei dati

I livelli di astrazione implementati nell'architettura dal DBMS permettono di garantire l'indipendenza dei dati.
Gli utenti e i programmatori che utilizzano una base di dati interagiscono con essa indipendentemente dai suoi dettagli implementativi. In particolare abbiamo:
• *Indipendenza Fisica*:
gli utenti interagiscono con il DBMS indipendentemente dalla struttura fisica usata per memorizzare i dati. Questo permette di cambiare la struttura di memorizzazione fisica in maniera trasparente rispetto agli utenti o i programmi (senza alterare lo schema logico o esterno).
• *Indipendenza Logica:*
gli utenti interagiscono con le viste (livello esterno) in maniera indipendente dal livello logico. Questo permette di aggiungere nuove viste per accomodare i requisiti di nuovi utenti senza modificare lo schema logico. In aggiunta è possibile cambiare lo schema logico mantenendo inalterate le viste degli utenti.

Linguaggi

Il DBMS mette a disposizione diversi linguaggi per effettuare operazioni di vario tipo sulla base di dati.
Questi linguaggi si distinguono in categorie in base allo scopo:
• *Linguaggi di definizione dei dati o Data Definition Language (DDL),* utilizzati dai Data Base Administrators per definire gli schemi (logici, fisici e esterni) e le autorizzazioni per l'accesso ai vari utenti.
• *Linguaggi di manipolazione dei dati o Data Manipulation Language (DML)*, utilizzati dai programmatori e dagli utenti finali per interrogare o aggiornare/modificare un'istanza della base di dati. La parte per il recupero dell'informazione si indica anche come LINGUAGGIO DI INTERROGAZIONE (QL).
• *Linguaggi di definizione delle strutture di memorizzazione o Data Storage definition Language (SDL),* utilizzati per specificare le strutture di memorizzazione ed i metodi di accesso della base di dati.

L'accesso ai dati può essere effettuato tramite delle interfacce offerte dal DBMS oppure direttamente nei vari programmi scritti in diversi linguaggi utilizzando apposite librerie.
In entrambi i casi le operazioni da svolgere sono descritte tramite DDL o DML.

- Più che di differenti linguaggi si tratta di differenti funzionalità/ruoli presenti in un unico linguaggio: SQL contiene tutti i ruoli (per cui si parla del DDL di SQL, del DML di SQL ...);
- L'utilizzo delle diverse funzionalità avviene in momenti diversi, e con privilegi e ruoli diversi;
- Una volta definito lo schema logico della base di dati, la progettazione della base di dati avviene ricorrendo ai DDL;
- Il caricamento dei dati avviene tramite i DML;
- La ricerca e la manipolazione delle informazioni avviene tramite le query (ossia i linguaggi di interrogazione QL/DML);
- L'ottimizzazione avviene tramite i Data Storage Language.

Riassumendo, l'accesso ai dai può avvenire:
• Tramite una interfaccia interattiva, solitamente fornita dal DBMS, in cui si inseriscono comandi SQL (Sql+ in oracle, workbench in mysql);
• Tramite comandi simili a quelli interattivi, immersi (embedded) in un linguaggio di programmazione (Java, C, C++). Ad esempio: API jdbc, odbc;
• Tramite interfacce grafiche amichevoli, offerte da DBMS, che permettono di sintetizzare interrogazioni senza usare un linguaggio testuale (Ad esempio Microsoft Access, PHPmyadmin).

Sia le interfacce interattive che i programmi possono risiedere su macchine diverse dal DBMS, connesse in rete e per motivi di prestazioni.