Sistemi di Basi di Dati

Gestione dei dati

Dato = "In informatica, la singola informazione codificabile o codificata". Il dato è la minima entità di informazione codificabile o codificata. Le informazioni vengono rappresentate tramite i dati.
L'interpretazione dei dati avviene tramite le codifiche e le decodifiche.
Se conosco il contesto (cioè come codificare/decodificare un dato) e il legame fra i dati, posso trarne informazioni. Ad esempio:
- 8 Ferrari. Contesto albergo→ stanza 8 ha ordinato una bottiglia di Ferrari. Contesto giornalismo sportivo→ la Ferrari è arrivata ottava.

L'evoluzione della tecnologia permette oggi di raccogliere una mole di dati. Questo ha portato alla necessità di avere una corretta e efficiente gestione dei dati, che è sempre stata fondamentale per la buona realizzazione di un sistema informativo.

Sistema informativo

Componente di una organizzazione che gestisce le informazioni di interesse cioè quelle utilizzate per il raggiungimento degli scopi dell'organizzazione.
Ciascuna organizzazione ha un sistema informativo, eventualmente non esplicitato nella struttura.
Sistema atto a organizzare e gestire i dati in modo tale da permettere la loro facile elaborazione/ memorizzazione per ottenere informazioni →Consente di gestire i dati in modo efficiente.
I sistemi informativi venivano realizzati anche prima dell'invenzione dei calcolatori. Esistono delle organizzazioni il cui compito è la gestione di informazioni (Banche, servizi anagrafici).
La porzione automatizzata del sistema Informativo è detta Sistema informatico, gestisce le informazioni con tecnologia informatica.

Database Management System – DBMS

La collezione dei dati utilizzati per rappresentare le informazioni di interesse è chiamata Base di dati o Database. Compito della base di dati è memorizzare i dati e rappresentare le relazioni tra di essi.
Il software atto a gestire i dati è detto Sistema di Gestione della base di dati o Database Management System.
Tradizionalmente era adottato nei sistemi informativi di grandi dimensioni, oggi adottato anche da sistemi più semplici→ mole dati crescenti, DBMS anche a basso prezzo e semplici da utilizzare.

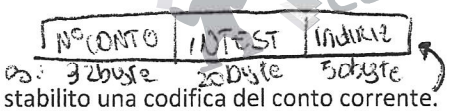
Un po' di storia..

Il primo DBMS per uso generico fu progettato da Charles Bachman nei primi anni '60: costituì la base per il modello di dati reticolare. Nel 1970 Edgar Cood propose una nuova struttura di rappresentazione di dati chiamata modello di dati relazionale. Negli anni '80 il modello relazione consolidò la sua posizione di dominanza. Il linguaggio di interrogazione SQL per basi di dati relazioni, è divenuto il linguaggio di interrogazione standard.

Archivio basato su Files

L'approccio classico usato dai programmi 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 (insieme di dati = archivio). Files = astrazione del sistema operativo del contenuto del disco.
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.
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 sono memorizzati su file condivisi sono replicati → 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 modificare il contenuto.
Esempio
Due applicazioni:
applicazione 1, manipolata da un operatore in banca. L'operatore deve poter accedere ai dati anagrafici, al conto corrente e alle operazioni effettuabili (prelievo,deposito,calcolo interessi).
Conto corrente è costituito da :



ho stabilito una codifica del conto corrente.
Applicazione 2, manipolata da cliente da web. Il cliente deve poter accedere al conto corrente e alle operazioni che può effettuare. Devo conoscere la codifica dell'applicazione 1, e questo può non essere facile da conoscere perché il programmatore può non avermi dato informazioni sulla codifica del programma. Le due applicazioni devono poter accedere agli stessi file: svantaggi. La condivisione dei file è rischiosa: ci deve essere un meccanismo per gestire le scritture concorrenti.

Approccio basato su DBMS

Questo approccio va oltre l'uso di file locali gestiti dalle singole applicazioni, tramite l'adozione di un sistema di gestione di dati che risulta indipendente dalle applicazioni e specializzato in tale funzione. I dati non sono gestiti/memorizzati dalle singole applicazioni ma da un DMBS che offre un'interfaccia comune a tutte le applicazioni. Il DMBS si interpone tra le applicazioni e la memoria di massa. I dati dunque non appartengono alla singola applicazione ma esse vi accedono tramite il DBMS. Il DMBS si mette tra i programmi e il DB in modo da risolvere i problemi di gestione e concorrenza.
Vantaggi
1-Le basi di dati sono condivise→ le applicazioni e gli utenti possono accedere a dati comuni evitando la realizzazione ad hoc di soluzioni come la condizione di file tra applicazioni diverse. Questo aiuta a :
-ridurre la ridondanza: una base di dati centralizzata permette di ridurre la replica delle stesse informazioni 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.
2-I DBMS sono componenti software specializzati nella gestione di grandi quantità di dati e implementano procedure basate sulle best-practices per la gestione di :
- efficacia e efficienza → efficienza: le tecniche di memorizzazione adottate permettono di migliorare le prestazioni di memorizzazione (tempo e spazio) e accesso alle informazioni (che altrimenti dovrebbero essere implementate in ogni programma). Efficacia: consentono di semplificare lo sviluppo di applicazioni.
- grosse dimensioni dei dati
- persistenza→ le BD sono persistenti, ovvero 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→ implementa tecniche per recuperare dati in caso di rottura(crash) dei dispositivi
-concorrenza→ implementa metodologie per garantisce un corretto accesso concorrente ai dati minimizzandone l'impatto sulle prestazioni di accesso. (esempio: limitando i tempi di attesa in seguito alla mutua esclusione di un dato).

- privatezza→ tecniche di sicurezza per garantire accesso ristretto, in modo da garantire a ciascun utente accesso solo al sottoinsieme dei dati a cui è autorizzato.

3. 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 → la memorizzazione e la gestione dei dati sono compiti del DBMS e dunque il processo di realizzazione delle applicazioni è semplificato. L'applicazione interagisce con il DB tramite un'interfaccia standard→ la struttura delle applicazioni segue tale standard.

Svantaggi:

Costo→ i DBMS commerciai sono costosi. Inoltre si aggiungo i costi indiretti: risorse hardware, sviluppo/conversione di applicazioni, formazione del personale, riorganizzazione dei processi aziendali.

Integrazione di servizi non sempre utili→ tali servizi possono 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 utilizzare soluzioni con file ordinari e non DBMS (es. excel). Tuttavia la diffusione di DMBS a basso costo ne sta estendendo l'uso.

Prodotti commerciali

I DBMS non sono software ad hoc ma sono solitamente prodotti commerciali. Le aziende produttrici 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, Microsoft SQL server.

In aggiunta esistono DMBS open-source che vengono sviluppati e gestiti da una comunità di sviluppatore e sono liberamente accessibili. Pur avendo molte funzionalità offerte dai prodotti commerciali, non sono adatti a gestire grandi quantità di dati ma possono essere usati in progetti più piccoli. Esempi open-source: MySQL, MariaDB.

Schema e modello dei dati

I DMBS sono software in grado di gestire dati eterogenei(differenti basi di dati).

Un modello di dati è una collezione di costrutti usati per descrivere i dati di interesse e la loro struttura in modo che essa risulti comprensibile ad un calcolatore. Con il modello di 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. Tramite questo schema dei dati si fornisce al DBMS una rappresentazione dei dati in modo tale da permetterne la gestione.

Modello di dati relazionale

è il modello attualmente più diffuso. 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 relazionale o tabella relazionale che può essere pensata come un insieme di record.

Modello relazionale: collezione di tabelle che rappresentano la relazione matematica fra i suoi elementi. Ogni relazione viene identificata da un nome; i campi della relazione vengono identificati attraverso un nome e un tipo.

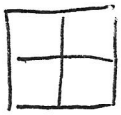
Schemi e istanze

La struttura di una database è detta **schema**:

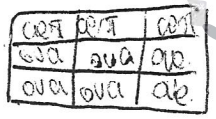
- è frutto del processo di progettazione della base di dati e raramente cambia durante la vita del DBMS.
- un cambiamento della schema è frutto di una errata progettazione o di un cambiamento della "realtà"

→ lo schema è un'entità statica

Scheletro vuoto:



L'**istanza** di un database è l'insieme delle informazioni contenute in un database in un certo istante:
-cambia continuamente nel tempo
-inserimento/modifica/cancellazione dei dati non cambia la struttura generale,ovvero lo schema.
Scheletro riempito ad un certo istante:



Astrazioni sui dati

Altro vantaggio dei DBMS è il loro supporto su astrazioni sui dati. I DBMS permettono di sviluppare le applicazioni estraendo al modo in cui i dati sono stati memorizzati. Tramite diversi livelli di astrazioni il DMBS maschera dettagli implementativi, al fine di semplificare l'interazione di diversi tipi di utenti 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 rappresentato dal modello dei dati. Ad esso è associato uno schema logico che fornisce una descrizione dell'intera base di dati per mezzo del modello logico adottato dal DMBS. Sono tralasciati tutti i dettagli implementativi.

Livello fisico→ livello più basso di astrazione, quello interno del database. Lo schema interno descrive come i dati rappresentati nello schema logico sono memorizzati nelle strutture fisiche di memorizzazione.

Livello esterno(delle viste)→ livello più alto di astrazione che permette di esporre agli utenti finali delle visioni(viste) parziali o successivamente elaborate dal livello logico. Vengono definite quando non tutti gli utenti hanno necessità/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 ridondanze.

Linguaggi

Il DBMS mette a disposizione diversi linguaggi per effettuare operazioni di vario tipo sulla base di dati. Si distinguono in categorie in base allo scopo:

-Linguaggi di definizione dei dati o Data Definition Language (DDL) → usati per definire gli schemi e le autorizzazioni per l'accesso ai vari utenti

-Linguaggi di manipolazione dei dati o Data Manipulation Language (DML) → usati per interrogare o aggiornare/modificare un'istanza della base di dati.

-Linguaggi di definizione delle strutture di memorizzazione o Data Storage definition Language (SDL)→ usati per specificare le strutture di memorizzazione e i metodi di accesso della base di dati.

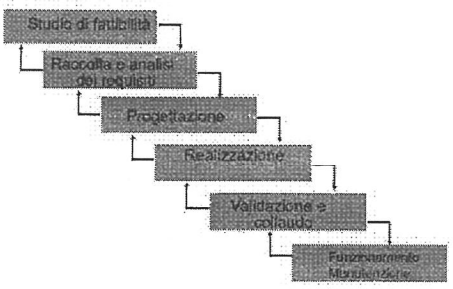
Indipendenza dei dati

I livelli di astrazione 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.

-Indipendenza fisica→ gli utenti interagiscono con il DBMS indipendentemente dalla struttura fisica usata per memorizzare i dati. Questo permette di cambiare 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) indipendentemente dal livello logico. Questo permette di aggiungere nuove viste per accomodare i requisiti di nuovi utenti senza modificare lo schema logico. È possibile,inoltre,cambiare lo schema logico mantenendo inalterate le viste degli utenti.

Workflow nella progettazione di una base di dati

Ciclo di vita dei sistemi informativi
-insieme delle attività svolte da analisti,progettisti,utenti nello sviluppo e nell'uso dei sistemi informativi
-attività iterativa dunque ciclo.
Fasi tecniche del ciclo di vita
Fasi sequenziali: si passa da una all'altra



Studio di fattibilità→ si analizza se vale la pensa realizzare questo particolare sistema informativo. Stima dei costi. Valutazione di diverse soluzioni: effettuare o meno lo sviluppo, informatizzare in parte il sistema, soluzioni più convenienti.
Raccolta e analisi dei requisiti→ quali funzioni devo implementare e quali dati devo utilizzare. Azione svolta parlando con gli utenti del sistema stesso.
Progettazione → di dati e applicazioni. Definizione dell'architettura del sistema.
Realizzazione→ dei dati e delle applicazioni
Validazione e collaudo→ quando l'applicazione è realizzata, devo verificare che ciò che è stato fatto effettivamente risponde ai requisiti del sistema. Se nella validazione lo sviluppatore si accorge che non è corretto, si torna alla fase della progettazione. Per questo si creano i cicli. Se la validazione è stata soddisfatta si passa al collaudo presso il cliente.
Funzionamento e manutenzione→ Se attraverso il collaudo si verifica che tutti i requisiti sono soddisfatti, il sistema va in funzionamento. Eventualmente nel funzionamento si possono scoprire nuovi errori, e si ritorna alla fase di realizzazione, correzione del progetto, validazione e collaudo e rimessa in funzione. Nella fase di manutenzione, si può verificare che l'utente chieda di introdurre nuovi servizi. Oppure questa fase si può riferire all'hardware che è sottoposto ad obsolescenza (invecchia velocemente, in termini di prestazioni. I processi nuovi in generale, più performanti). Poiché i costi energetici possono essere rilevanti, effettuo manutenzione per adeguare l'hardware con i nuovi requisiti del sistema (altra problematica possibile con hardware: aumento numero di utenze).

Progettazione
Riguarda due aspetti:
-progettazione dei dati
-progettazione delle applicazioni.