



Introduzione ai DB

Prof. Viglietti Francesco
Www.in4matika.altervista.org

CARATTERISTICHE DI UN DB

Un DB è una collezione di archivi di dati ben organizzati e ben strutturati, gestiti in modo integrato da un DBMS, in modo da essere utilizzati da utenti diversi con programmi diversi.

Deve possedere le seguenti caratteristiche:

Efficienza: facilità nel reperire le informazioni desiderate.

Consistenza: deve contenere dati significativi ed utilizzabili.

Sicurezza ed integrità: la prima protegge il DB da eventi accidentali o non autorizzati, la seconda protegge il DB dalle operazioni degli utenti autorizzati impedendo perdite di consistenza di dati.

Limiti degli archivi convenzionali 1

Es: Si vogliono gestire le informazioni dei clienti di una banca. Per ciascun cliente vengono gestiti: il numero di conto, il nome, intestatario, indirizzo, cap, e saldo. Il tutto è stato organizzato in due file, uno con anagrafica e numero conto e l'altro con numero conto e saldo.

| Nome | indirizzo | città | cap | n.conto |
|---------|------------|---------|-------|---------|
| Rossi | Dante | Sassari | 07100 | 10340 |
| Verdi | XXV aprile | Napoli | 80100 | 8190 |
| Bianchi | Pascoli | Roma | 00100 | 1300 |
| Neri | Sardegna | Milano | 20100 | 23452 |
| Viola | Colombo | Torino | 10100 | 77818 |
| Rosa | Spagna | Roma | 00100 | 1300 |

La gestione dati mediante Cobol. Inserimento e modifica dell'anagrafica, versamento prelievo, elenco clienti e saldo

| n.conto | saldo |
|---------|---------|
| 77818 | 32000 |
| 10340 | 10000 |
| 1300 | 200000 |
| 23452 | 1003000 |
| 8190 | 1230 |

Limiti degli archivi convenzionali 2

Solo successivamente viene inserita una funzionalità di gestione dei singoli movimenti che deve integrarsi con i precedenti.

Problemi:

- campi identici con diverso nome (*nconto* e *codice*)
- stessa tipologia ma differente Lunghezza per gli Stessi campi (nome e cliente)
- differente tipologia per campi che rappresentano la stessa grandezza (saldo e importo)
- ridondanza dei dati presenza del campo (cliente e nome nelle due tabelle)
- aggiunta successiva di altri campi, vincoli di integrità, interrogazioni,...

| cliente | codice | data | causale | importo |
|---------|--------|---------|---------|---------|
| Bianchi | 1300 | 1/1/10 | Vers | 1400,00 |
| Neri | 23452 | 1/3/10 | prel | 1750,00 |
| ... | ... | ... | ... | ... |
| Neri | 23452 | 15/7/10 | vers | 1200,00 |
| Rossi | 10340 | 5/7/10 | prel | 775,00 |

Limiti degli archivi convenzionali 3

Perciò l'approccio su archivi tradizionali (file based) ha portato ai seguenti inconvenienti:

Dipendenza dei dati → i programmi saranno vincolati agli archivi che gestiscono (il linguaggio necessita delle informazioni sugli archivi)

Default Query → unico modo di accesso ai dati

Differenze di formato → difficoltà di collegamento

Ridondanza e inconsistenza → problemi in fase di update o differenti valori in diversi file (*incongruenza*), e quindi porta all'inconsistenza (quale dei valori è corretto?).

Integrità dei dati → di difficile attuazione sugli aggiornamenti

Concorrenza → difficile applicazione nei sistemi tradizionali, servono meccanismi che rendano esclusivi gli accessi

Sicurezza → nei sistemi tradizionali dobbiamo fermarci a livello di file, non a livello di campi.

Salvataggio → non esiste una gestione del backup, quindi in caso di restore del sistema i dati saranno solo parzialmente ripristinati...

DataBase Management System

È un prodotto software in grado di gestire collezioni di dati:

- **Grandi**, di dimensioni molto maggiori della memoria centrale dei sistemi di calcolo utilizzati
- **Persistenti**, con un periodo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano
- **Condivise**, utilizzate da applicazioni e utenti diversi

garantendo **affidabilità** (resistenza a malfunzionamenti HW e SW) e **privatezza** (con una disciplina e un controllo degli accessi). Come ogni prodotto informatico, un DBMS deve essere **efficiente** (utilizzando al meglio le risorse di spazio e tempo del sistema) ed **efficace** (rendendo produttive le attività dei suoi utilizzatori).

DataBase Management System

Un DBMS deve essere in grado di:

- creare un DB mediante il **Data Definition Language**
- modificare i dati nel DB mediante il **Data Manipulation Language**
- interrogare il DB mediante il **Query Language**

Inoltre deve risolvere i problemi visti in precedenza degli archivi tradizionali.

Alcuni prodotti software DBMS disponibili sul mercato:

MS Access, DB2, Oracle, Sybase, MS SQL Server,
MySQL

Proprietà: Consistenza

Ridondanza → informazioni ripetute

Rischio di inconsistenza → le versioni possono non coincidere



CCS Ingegneria Informatica Orario - Netscape

File Edit View Go Communicator Help

UNIVERSITA' DEGLI STUDI DI CHISSADOVE

Corso di Studi in Ingegneria Informatica

ORARIO DELLE LEZIONI PER L'ANNO
ACCADEMICO 1999-2000

| INSEGNAMENTO | Docente | Aula | Orario |
|----------------------|-------------|------|-------------|
| Analisi matematica I | Luigi Neri | N1 | 8:00-9:30 |
| Basi di dati | Piero Rossi | N2 | 9:45-11:15 |
| Chimica | Nicola Mori | N1 | 9:45-11:30 |
| Fisica I | Mario Bruni | N1 | 11:45-13:00 |
| Fisica II | Mario Bruni | N3 | 9:45-11:15 |
| Sistemi informativi | Piero Rossi | N3 | 8:00-9:30 |

Document: Done



Orari di ricevimento - Netscape

File Edit View Go Communicator Help

UNIVERSITA' DEGLI STUDI CHISSADOVE

Corso di Studi in Ingegneria Informatica

Orario di ricevimento dei docenti

| DOCENTE | INSEGNAMENTI | ORARIO |
|-------------|-------------------------------------|----------------|
| Mario BRUNI | Fisica I Fisica II | Martedì' 10-12 |
| Luigi NERI | Analisi matematica I | Lunedì' 12-13 |
| Piero ROSSI | Basi di dati Sistemi informativi | Giovedì' 11-13 |
| Nicola MORI | Chimica | Martedì' 16-18 |

Document: Done

I DB sono condivisi e garantiscono la Privacy

Un DB è una risorsa integrata, condivisa fra applicazioni

Conseguenze:

Attività diverse su dati condivisi → meccanismi di autorizzazione

Accessi di più utenti ai dati condivisi → controllo della concorrenza

Si possono definire meccanismi di autorizzazione

l'utente A è autorizzato a leggere tutti i dati e a modificare quelli sul ricevimento

l'utente B è autorizzato a leggere X e a modificare Y

I DBMS sono affidabili e le transazioni sono atomiche

Affidabilità → resistenza a malfunzionamenti HW e SW
Una base di dati è una risorsa pregiata e quindi deve essere conservata a lungo termine

Tecnica fondamentale → gestione delle transazioni

Transazioni → insieme di operazioni da considerare indivisibile (atomico), corretto anche in presenza di concorrenza e con effetti definitivi

La sequenza di operazioni sulla base di dati viene eseguita per intero o per niente:

trasferimento di fondi da un conto A ad un conto B: o si fa il prelievo da A **e** il versamento su B **o** nessuno

Le transazioni sono concorrenti e permanenti

L'effetto di transazioni concorrenti deve essere coerente (ad esempio "equivalente" all'esecuzione separata) → se due assegni emessi sullo stesso c/c vengono incassati contemporaneamente si deve evitare di trascurarne uno.

La conclusione positiva di una transazione corrisponde ad un impegno (in inglese commit) a mantenere traccia del risultato in modo definitivo, anche in presenza di guasti e di esecuzione concorrente. (Transazione permanente)

I DBMS devono essere...efficienti

- Cercano di utilizzando al meglio le risorse di spazio di memoria (principale e secondaria) e tempo (di esecuzione e di risposta)
- I DBMS, con tante funzioni, rischiano l'inefficienza e per questo ci sono grandi investimenti e competizione
- L'efficienza è anche il risultato della qualità delle applicazioni

Descrizioni dei dati nei DBMS

Rappresentazioni dei dati a livelli diversi permettono l'indipendenza dei dati dalla rappresentazione fisica:
i programmi fanno riferimento alla struttura a livello più alto, e le rappresentazioni sottostanti possono essere modificate senza necessità di modifica dei programmi
Precisiamo attraverso il concetto di modello dei dati

Modello dei dati

Insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica

Tre tipi (principali) di modelli

- modelli concettuali → indipendenti dal DBMS sono vicini alla rappresentazione della realtà.
- modelli logici → dipendono dal DBMS e ne rappresentano l'implementazione a livello software
- Modelli fisici → sono a basso livello, vicini al metodo di memorizzazione dei dati.

I modelli logici dei dati

Sono adottati nei DBMS esistenti per l'organizzazione dei dati, sono utilizzati dai programmi e sono indipendenti dalle strutture fisiche

Tre modelli logici tradizionali

- Gerarchico
- Reticolare



utilizzano riferimenti espliciti (puntatori) fra record

- Relazionale → "è basato su valori" anche i riferimenti fra dati in strutture (relazioni) diverse sono rappresentati per mezzo dei valori stessi

Più recente (e poco diffuso)

- a oggetti

Il modello relazionale

Proposto da E.F. Codd nel '70 per favorire l'indipendenza dei dati. Disponibile in DBMS reali nel 1981 (non è facile implementare l'indipendenza con efficienza e affidabilità!)
Si basa sulla rappresentazione delle relazioni (concetto matematico) fra i dati per mezzo di **tabelle**

In ogni DB esistono:

lo schema, sostanzialmente invariante nel tempo, che ne descrive la struttura (aspetto intensionale) es.: le intestazioni delle tabelle

l'istanza, i valori attuali, che possono cambiare anche molto rapidamente (aspetto estensionale) es.: il "corpo" di ciascuna tabella

I Linguaggi per DB

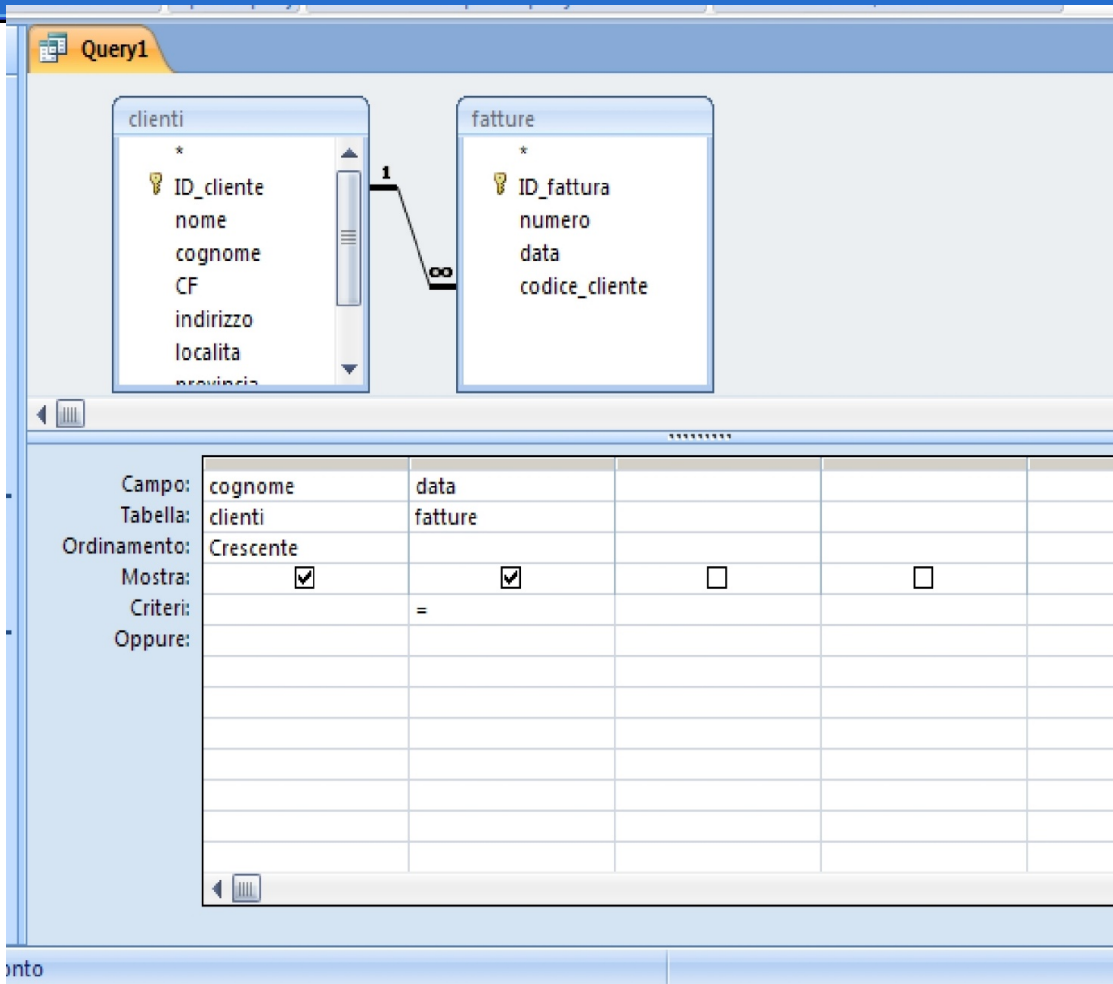
Data definition language (DDL) → per la definizione di schemi (logici, esterni, fisici) e altre operazioni generali (permessi agli utenti, creazione, eliminazione di tabelle, viste, report, ...)

Data manipulation language (DML) → per l'aggiornamento delle istanze del DB

Query language (QL) → interrogazione del DB mediante criteri di ricerca forniti dall'utente

La diffusione del modello relazionale ha dato vita a dei linguaggi specifici per DB, raggruppando in un unico linguaggio le funzioni del DDL, DML e del QL.

Linguaggi di interrogazione



SQL (Structured Query Language): parzialmente dichiarativo (reale).

QBE (Query by Example): dichiarativo (reale).

Permette di effettuare interrogazioni riempiendo interattivamente una maschera per interrogazioni.

Particolarmente usato in MS Access.