

ARCHIVI TRADIZIONALI e BASI DI DATI

Un **file** (archivio) è una raccolta di **record logici**. La gestione dei **file** è un servizio reso dal Sistema Operativo.

Ogni record logico contiene una sequenza di bit ed è suddiviso in **campi**. Ogni campo contiene una differente informazione.

Esempio:

ROSSINI GIORGIO	Via Nappi 6 - Milano	2372445
<i>campo1:</i> nome	<i>campo2:</i> indirizzo	<i>campo3:</i> telefono

Su questi file esistono meccanismi per l'**inserimento**, la **cancellazione**, la **modifica** e la **ricerca** dei dati.

Organizzazioni dei file:

- **sequenziale** I record sono letti o scritti uno dopo l'altro. Inserimento, cancellazione e modifica di un file sequenziale presentano problemi, in quanto richiedono di norma lo spostamento di tutti i record che seguono il punto di intervento.
- **ad accesso diretto** L'allocazione dei record in memoria di massa è definita da un algoritmo che genera un indirizzo a partire dal valore assunto da un campo del record detto **chiave**.
Il vantaggio delle strutture ad accesso diretto è la rapidità della ricerca: in assenza di conflitti ogni ricerca richiede una sola operazione di ingresso/uscita.
- **a indice** In queste strutture, come in quelle ad accesso diretto, è il valore di un campo **chiave** che determina il record a cui accedere. La corrispondenza fra la chiave e l'indirizzo è mantenuta esplicitamente mediante un file ausiliario detto **file indice**.
Su uno stesso archivio è possibile avere diversi file indice relativi a chiavi diverse.

Le **basi di dati** sono sorte alla fine degli anni '60. Prima i sistemi informativi si basavano sull'uso di **files** separati.

I programmi applicativi accedevano ai **files** individualmente, utilizzando procedure del Sistema Operativo.

L'accesso ai files tradizionali comportava una serie di **problemi**:

1. **Ridondanza**: informazioni ripetute
2. **Incongruenza**: le modifiche non sempre vengono apportate in tutti gli archivi dove compaiono le stesse informazioni
3. **Inconsistenza**: se ad esempio uno stesso articolo si ritrova con marche diverse quale sarà quella giusta?
4. **Dipendenza dei programmi dai dati**: se cambia il tracciato record o la cartella di un archivio devo cambiare l'applicativo
5. **Difficoltà nel gestire l'integrità dei dati**: va scritto codice ad hoc nell'applicativo
6. **Difficoltà nel gestire la concorrenza**: in un file condiviso se due utenti tentano la modifica, si possono generare informazioni non corrette
7. **Limitata sicurezza e privacy**: non tutti gli utenti hanno stessi permessi sui dati e la privacy dei dati è limitata perché gestita solo dal sistema operativo
8. **Scarsa protezione** dei dati da guasti accidentali

Questi inconvenienti si possono superare se tutti i dati vengono organizzati in una **base di dati** e gestiti da un **DBMS** (*Data Base Management System*).

Confronto tra sistemi di archiviazione tradizionali e DBMS

ARCHIVI TRADIZIONALI	DBMS	
<i>Sistemi di archiviazione</i>	<i>Sistemi per basi di dati</i>	<i>Benefici</i>
In generale ogni applicazione ha propri archivi, con dati parzialmente duplicati.	I dati sono organizzati per poter essere usati da diverse applicazioni.	Integrazione dei dati.
Gli archivi sono modificati in tempi diversi e copie dello stesso dato possono risultare diverse.	Gli aggiornamenti vengono visti subito da tutte le applicazioni, a causa della condivisione dei dati.	Consistenza dei dati.
Ogni applicazione deve garantire l'integrità dei dati.	I DBMS offrono meccanismi per il controllo centralizzato dell'integrità dei dati.	Integrità dei dati.
Cambiamenti nella definizione dei record devono essere riportati in ogni applicazione.	Con la descrizione centralizzata dei dati solo le viste logiche devono essere modificate.	Indipendenza logica.
Le modalità di accesso ai dati dipendono dalla loro organizzazione fisica che, in generale, va programmata.	Si può accedere ai dati indipendentemente dalla loro organizzazione fisica, che viene scelta fra le organizzazioni previste dal DBMS.	Indipendenza fisica.
I programmi usano nomi diversi per riferirsi agli stessi dati.	I programmi usano i nomi definiti nello schema.	Standardizzazione dei dati.
I dati sono accessibili solo da programmi.	I dati sono accessibili da programmi e interattivamente.	Facilità d'uso dei dati.
La sicurezza può essere assicurata solo scrivendo opportuni programmi.	I DBMS offrono meccanismi per garantire la sicurezza dei dati.	Sicurezza dei dati.
Ogni applicazione deve definire le proprie procedure per proteggere i dati da malfunzionamenti.	I DBMS offrono meccanismi per la protezione dei dati da malfunzionamenti.	Affidabilità dei dati.
I dati sono usati in modo esclusivo da ogni applicazione.	I dati sono accessibili in modo concorrente.	Condivisione dei dati.

Tabella 1.1. Confronto fra i sistemi di archiviazione e DBMS.

BASE DI DATI

E' una raccolta di dati progettati per essere fruiti in maniera **ottimizzata** da differenti applicazioni e utenti diversi



Semplice: facilmente ritrovabili
Efficiente: rispetto al tempo CPU e spazio RAM
Efficace: informazioni rappresentative della realtà in esame
Sicuro: operazioni consentite a soggetti identificabili e sicuri

SISTEMA DI GESTIONE DI UNA BASE DI DATI o DBMS



Prodotti software che permettono di creare e di interagire con una base di dati, consentendo opportune **operazioni** agli **utenti** autorizzati, nel rispetto delle **regole** prestabilite. Le richieste degli utenti non devono violare alcun vincolo sui dati.

FUNZIONI DI UN DBMS

Permettere la **creazione** di una nuova base di dati



DDL
Data Definition Language



Facilitare gli utenti nell'inserimento, cancellazione, **modifica**



DML
Data Manipulation Language



Rendere possibile l'**estrazione** di informazioni interrogando la base di dati



QL
Query Language



Superare i limiti descritti nelle pagine precedenti:

Non ridondanza integrità indipendenza concorrenza sicurezza