

# Database - Parte 2

## Le applicazioni gestionali e le basi di dati

Prof. Francesco Gobbi

I.I.S.S. Galileo Galilei - Ostiglia (MN)  
Materia: Informatica

September 17, 2024

# Introduzione alle Applicazioni Gestionali

Le applicazioni gestionali sono software progettati per aiutare le aziende, studi professionali o enti pubblici a risolvere problemi gestionali.

- ▶ I dati vengono raccolti e organizzati in archivi.
- ▶ Vengono forniti strumenti di interrogazione e stampa di report.

# Tipi di Archivi in Applicazioni Gestionali

Gli archivi in una gestione informatica sono solitamente suddivisi in tre categorie:

- ▶ **Anagrafiche:** contengono informazioni sui soggetti, persone o prodotti.
- ▶ **Movimenti:** riguardano eventi relativi ai soggetti descritti nelle anagrafiche.
- ▶ **Parametri:** dati che rimangono costanti per un determinato periodo di tempo.

# Esempio di Applicazione Gestionale: CRM per un'Azienda

Un esempio concreto di applicazione gestionale è l'uso di un CRM (Customer Relationship Management) per un'azienda:

- ▶ **Anagrafiche:** memorizza informazioni sui clienti (nome, indirizzo, contatti).
- ▶ **Movimenti:** registra tutte le interazioni con i clienti, come ordini o richieste di supporto.
- ▶ **Parametri:** include sconti fissi applicati ai clienti fedeli.

Questo permette di:

- ▶ Migliorare la gestione delle relazioni con i clienti.
- ▶ Automatizzare la registrazione degli ordini e delle richieste.
- ▶ Analizzare i dati per migliorare le vendite e il servizio clienti.

# Vantaggi delle Applicazioni Gestionali

Utilizzare applicazioni gestionali porta diversi vantaggi:

- ▶ Maggiore efficienza operativa.
- ▶ Riduzione degli errori manuali.
- ▶ Migliore capacità di analisi e decisioni basate sui dati.

# Gestione dell'Archivio: Creazione e Manutenzione

La gestione degli archivi richiede un processo ben strutturato per garantire l'efficienza e l'integrità dei dati nel tempo.

- ▶ **Creazione degli Archivi:**

- ▶ Definizione della struttura dei dati (tabelle, campi, relazioni).
- ▶ Implementazione dei sistemi di sicurezza per l'accesso ai dati.
- ▶ Esempio: Creare un archivio clienti con campi come nome, indirizzo, numero di telefono.

- ▶ **Manutenzione nel Tempo:**

- ▶ Verifica periodica dell'integrità dei dati.
- ▶ Aggiornamento delle informazioni obsolete.
- ▶ Esempio: Controllo mensile per eliminare o aggiornare i clienti che non hanno interagito da un anno.

# Procedure di Interrogazione degli Archivi

L'interrogazione degli archivi consente di estrarre informazioni rilevanti in modo rapido e preciso.

- ▶ **Query di Selezione:**

- ▶ Utilizzate per estrarre dati specifici dall'archivio.
- ▶ Esempio: "Trova tutti i clienti che hanno effettuato un acquisto negli ultimi 6 mesi".

- ▶ **Query di Aggregazione:**

- ▶ Utilizzate per ottenere statistiche o riassunti dei dati.
- ▶ Esempio: "Calcola il totale delle vendite mensili per ogni regione".

- ▶ **Query con Condizioni:**

- ▶ Utilizzate per filtrare i dati in base a determinate condizioni.
- ▶ Esempio: "Trova tutti i clienti con età superiore a 30 anni e che vivono in una città specifica".

# Esempi di Linguaggi di Interrogazione: SQL

Il linguaggio SQL (Structured Query Language) è uno dei più comuni per l'interrogazione degli archivi.

- ▶ **SELECT**: per selezionare dati specifici.
  - ▶ Esempio: 'SELECT nome, indirizzo FROM clienti WHERE città = 'Roma';'
- ▶ **UPDATE**: per aggiornare i dati.
  - ▶ Esempio: 'UPDATE clienti SET telefono = '1234567890' WHERE id\_cliente = 1;'
- ▶ **DELETE**: per cancellare dati non più necessari.
  - ▶ Esempio: 'DELETE FROM clienti WHERE id\_cliente = 5;'



# Concetto di Basi di Dati

Le **basi di dati** (o database) sono insiemi organizzati di dati che possono essere:

- ▶ Archiviati in modo strutturato per un facile accesso.
- ▶ Gestiti tramite software appositi per manipolare, recuperare e memorizzare dati in modo efficiente.

**Obiettivo:** superare i limiti degli archivi tradizionali, garantendo l'integrità e la sicurezza delle informazioni.

# Importanza delle Basi di Dati nell'Organizzazione

Le basi di dati sono fondamentali per l'**efficienza** e la **produttività** aziendale:

- ▶ **Accesso rapido:** permettono di ritrovare facilmente le informazioni desiderate.
- ▶ **Organizzazione ottimale:** consentono di gestire grandi quantità di dati con velocità ed efficacia.
- ▶ **Sicurezza:** proteggono i dati sensibili da accessi non autorizzati.
- ▶ **Supporto decisionale:** forniscono informazioni precise e aggiornate, migliorando il processo decisionale.

Esempio: i dati relativi agli articoli del magazzino possono essere utilizzati per generare listini e report personalizzati.

# Integrità dei Dati

L'integrità dei dati è un principio fondamentale per il funzionamento di un database:

- ▶ **Integrità:** garantisce che le operazioni effettuate dagli utenti autorizzati non compromettano la consistenza del database.
- ▶ **Consistenza:** i dati inseriti devono essere significativi e utilizzabili correttamente nelle applicazioni.
- ▶ **Sicurezza:** devono essere implementate misure per prevenire la perdita o il danneggiamento accidentale dei dati.

Esempio: se un dipendente effettua un aggiornamento su un articolo di magazzino, il sistema garantisce che l'informazione sia coerente e riflessa in tutti i listini e report connessi.

# Cos'è un DBMS

**DBMS** sta per *DataBase Management System*, ed è un sistema software utilizzato per la gestione delle basi di dati. Le sue funzioni principali includono:

- ▶ Organizzare i dati in maniera strutturata.
- ▶ Permettere l'accesso, la manipolazione e l'aggiornamento dei dati da parte degli utenti.
- ▶ Mantenere l'integrità e la sicurezza delle informazioni.
- ▶ Consentire la gestione degli utenti e dei loro permessi.

# Come si compone un DBMS

Un DBMS è formato da diverse componenti che lavorano insieme per gestire le basi di dati:

- ▶ **Motore di archiviazione:** responsabile dell'organizzazione fisica dei dati su memoria di massa.
- ▶ **Linguaggio di interrogazione (SQL):** consente agli utenti di accedere e modificare i dati con comandi specifici.
- ▶ **Sistema di gestione delle transazioni:** garantisce che le operazioni sui dati siano sicure e coerenti, anche in presenza di errori o interruzioni.
- ▶ **Sistema di controllo degli accessi:** gestisce gli utenti, i permessi e le politiche di sicurezza.
- ▶ **Sistema di backup e ripristino:** permette di salvaguardare i dati e ripristinarli in caso di perdita o danno.

# Differenza tra DBMS e Database

È importante distinguere tra il concetto di **DBMS** e **database**:

- ▶ **Database**: è una collezione di dati strutturati, organizzati per essere facilmente accessibili, gestiti e aggiornati.
- ▶ **DBMS**: è il software che gestisce i database. Si occupa della manipolazione, archiviazione e sicurezza dei dati, senza richiedere che l'utente abbia conoscenze dettagliate della struttura fisica dei dati.

In sintesi:

- ▶ Il **database** è l'insieme dei dati.
- ▶ Il **DBMS** è il sistema che gestisce quei dati.

# Utenti del DBMS

Il DBMS interagisce con due principali categorie di utenti:

- ▶ **Utenti finali:** accedono alle informazioni contenute nel database tramite applicazioni specifiche per svolgere operazioni quotidiane (ad esempio, inserire nuovi dati o consultare informazioni esistenti).
- ▶ **Amministratori di database (DBA):** sono responsabili della progettazione e della manutenzione del database, della gestione delle performance e dell'integrità dei dati, e della configurazione del sistema per garantire la sicurezza.

# Vantaggi dell'uso di un DBMS

L'adozione di un DBMS porta numerosi vantaggi rispetto ai sistemi di archiviazione tradizionali:

- ▶ **Migliore organizzazione:** i dati sono strutturati e facilmente accessibili.
- ▶ **Sicurezza:** il DBMS permette la gestione avanzata degli accessi e dei permessi.
- ▶ **Consistenza dei dati:** il sistema garantisce l'integrità e la coerenza delle informazioni, riducendo il rischio di dati duplicati o contraddittori.
- ▶ **Backup e ripristino:** il DBMS include funzionalità di backup automatico e procedure di ripristino in caso di guasti.



# Problemi dell'Approccio File-Based

L'approccio **file-based**, cioè basato su archivi indipendenti, presenta i seguenti inconvenienti:

- ▶ **Ridondanza e inconsistenza dei dati:** La duplicazione dei dati può causare incongruenze tra diversi archivi, rendendo difficile determinare i dati corretti.
  - ▶ **Esempio:** Se un cliente cambia indirizzo, questo deve essere aggiornato in più file. Se l'aggiornamento avviene solo in un file e non negli altri, si crea un'incongruenza.
- ▶ **Difficoltà nell'accesso ai dati:** Il programmatore deve seguire le modalità di accesso stabilite, limitando le operazioni ammissibili.
  - ▶ **Esempio:** In un sistema file-based, per trovare i dati di un cliente, il programmatore deve sapere esattamente in quale file e come sono organizzati, rendendo il processo complesso e manuale.

# Isolamento dei Dati e Dipendenza dai Dati

- ▶ **Isolamento dei dati:** I dati possono essere rappresentati in formati diversi, rendendo difficile l'integrazione tra archivi e programmi.
  - ▶ **Esempio:** Un file potrebbe contenere informazioni sui clienti in formato CSV, mentre un altro file con dati sui pagamenti potrebbe essere in formato JSON. Collegare questi file richiede un'integrazione manuale complessa.
- ▶ **Dipendenza dai dati:** Ogni programma dipende dalla struttura dell'archivio, richiedendo modifiche a tutti i programmi se la struttura cambia.
  - ▶ **Esempio:** Se viene aggiunto un nuovo campo, come il "numero di cellulare" di un cliente, tutti i programmi che accedono a quel file devono essere aggiornati per gestire il nuovo campo.

# Integrità dei Dati e Interrogazioni Predefinite

- ▶ **Difficoltà nel gestire l'integrità dei dati:** I vincoli di integrità sono spesso nascosti nei programmi, rendendo difficile mantenere la coerenza dei dati.
  - ▶ **Esempio:** Se un cliente viene eliminato da un file, non c'è alcuna garanzia che i suoi ordini vengano eliminati anche in un altro file collegato, violando l'integrità referenziale.
- ▶ **Interrogazioni predefinite:** Le interrogazioni sui dati sono limitate a quelle previste dalle applicazioni sviluppate, limitando la flessibilità.
  - ▶ **Esempio:** Un'applicazione per la gestione di magazzino permette solo di vedere il numero di prodotti disponibili, ma non consente di effettuare query avanzate come il calcolo del valore complessivo del magazzino.

# Aggiornamento del Software e Difficoltà d'Accesso

- ▶ **Complessità nell'aggiornamento del software e delle strutture:** Modificare la struttura di un record richiede la modifica di tutti i programmi che utilizzano tale record.
  - ▶ **Esempio:** Aggiungendo un nuovo campo "data di nascita" nel file clienti, tutti i programmi che accedono al file devono essere modificati per gestire correttamente il nuovo campo.
- ▶ **Difficoltà nell'accesso ai dati:** L'accesso è rigido e dipendente dalla struttura del file, complicando il lavoro del programmatore per accedere ai dati.
  - ▶ **Esempio:** Se un file contiene i dati di ordini in un formato tabellare non standard, l'accesso ai dati richiede la scrittura di codice personalizzato per leggere e manipolare il formato specifico.

# Difficoltà nel Gestire la Concorrenza

L'approccio file-based soffre di problemi legati alla concorrenza, quando più utenti accedono e modificano i dati contemporaneamente.

- ▶ **Esecuzione concorrente:** Lo stesso programma può essere eseguito contemporaneamente da più utenti.
- ▶ **Perdita di aggiornamento:** Questo può portare a problemi come la "perdita di aggiornamento", dove uno degli utenti sovrascrive accidentalmente le modifiche fatte da un altro.

**Esempio:** Se due utenti aggiornano lo stesso record cliente contemporaneamente, solo l'ultima modifica sarà salvata, causando la perdita di aggiornamenti importanti.

# Salvataggio e Ripristino Dati Limitato

I sistemi file-based hanno limitazioni nel gestire il salvataggio e il ripristino dei dati.

- ▶ **Backup:** Il backup è l'unico mezzo di difesa per prevenire la perdita accidentale di dati causata da malfunzionamenti hardware o software.
- ▶ **Ripristino inefficace:** Tuttavia, il ripristino da backup può risultare inefficace, soprattutto se i dati vengono persi o modificati in modo frequente.

**Esempio:** Se un sistema subisce un crash e i dati non sono stati salvati di recente, l'utente potrebbe perdere modifiche importanti apportate subito prima del malfunzionamento.

# Limiti nella Gestione della Sicurezza

Nei sistemi tradizionali, la gestione della sicurezza presenta diverse limitazioni:

- ▶ **Riserbo:** Il sistema deve garantire che solo utenti autorizzati possano accedere ai dati e che lo facciano solo nei limiti dei loro permessi.
- ▶ **Operazioni limitate:** Nei sistemi file-based, le operazioni di controllo sono limitate a livello di file system, non a livello di singolo record.

**Esempio:** Se un utente ha accesso a un intero file, non è possibile limitare l'accesso a specifici campi all'interno di quel file, compromettendo la sicurezza.