

L'organizzazione degli archivi e le basi di dati

ITS Academy - Fabbrica digitale 4.0

Docente: Fedeli Massimo

AS 2025/26

Gli archivi

- L'uso degli archivi deriva dalla necessità di conservare dati e informazioni in modo permanente perché potranno essere utili in momenti successivi.

Quali dati necessitano di essere memorizzati?

- Questo vale per dati di tipo personale, come quelli contenuti in una normale rubrica di contatti, ma anche per i documenti funzionali alla vita di uno Stato, di un'azienda o di un ente.

Chi ne ha bisogno?

- In un'**azienda** l'esecuzione delle normali attività amministrative e operative, così come la definizione e la scelta delle politiche commerciali, finanziarie e relative al personale, è strettamente legata all'elaborazione di insiemi di dati che sono raccolti e conservati in archivi.

Def. un archivio può essere definito come un **insieme organizzato di informazioni** caratterizzate da alcune proprietà fondamentali:

- esiste un **nesso logico** tra di esse (cioè sono in qualche modo inerenti a un medesimo argomento);
- sono rappresentate secondo un **formato** che ne rende possibile l'interpretazione;
- sono registrate con un supporto su cui è possibile scrivere e rileggere informazioni anche a distanza di tempo;
- sono organizzate in modo che siano **facilmente consultabili**.

Correlazione tra dati - esempio dell'archivio telefonico

- L'elenco telefonico è un archivio di dati in cui le informazioni riguardano gli abbonati al telefono di una provincia.
- Per ogni abbonato sono riportati nell'ordine: generalità indirizzo numero di telefono



- La struttura delle informazioni nelle **righe** (formato), ne rende facile la lettura e l'interpretazione da parte della persona che consulta l'elenco.
- Il supporto è dal disco che contiene i dati.
- Gli abbonati sono stampati seguendo l'ordine alfabetico dei cognomi, all'interno della suddivisione per comune, per permettere un veloce reperimento del numero di telefono che corrisponde alla persona cercata (organizzazione dei dati).



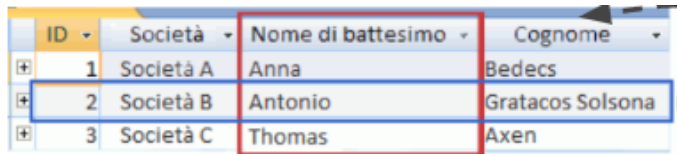
Figura: Gestione automatizzata degli archivi

I record e i campi

In un archivio le informazioni, in genere, sono raggruppate secondo un'unità logica: nel caso dell'elenco telefonico i dati relativi a ogni abbonato, in un archivio notarile i dati contenuti nell'incartamento relativo a un cliente. Questi insiemi di informazioni logicamente organizzate

e riferite a un unico soggetto sono chiamati con il termine **record**;

- Le singole informazioni che compongono il record si chiamano **campi**;
- L'elenco dei campi che lo compongono viene detto **tracciato del record**.



	ID ▾	Società ▾	Nome di battesimo ▾	Cognome ▾
+	1	Società A	Anna	Bedecs
+	2	Società B	Antonio	Gratacos Solsona
+	3	Società C	Thomas	Axen

Figura: Record

Si consideri un archivio con le informazioni anagrafiche degli studenti.

- Una determinata collezione di record omogenei è detto **file**.
- Un **archivio** si compone solitamente di più file.
- Ogni record del file contiene le informazioni anagrafiche di un dato studente ed è composto da un insieme di campi.
- Nel caso
di un archivio anagrafico il record sarà composto, per esempio, da campi per rappresentare il numero di matricola, il cognome, il nome, la data e il luogo di nascita, l'indirizzo e il numero

studente descritto nel record.

Definizione

Un **file** è una collezione di record, cioè di informazioni logicamente omogenee che descrivono i singoli elementi di una realtà considerata.

Ogni record è composto da un insieme di campi che contengono i valori assunti dalle caratteristiche scelte per descrivere la realtà.

La creazione di un archivio

La creazione di un archivio richiede la definizione preliminare delle seguenti specifiche:

- il **nome dell'archivio**, che lo identifica e serve a ricordarne il contenuto; per esempio “archivio fornitori” oppure “archivio anagrafico”;
- il **tracciato record**, in altre parole quali informazioni compongono il record;
- il **supporto** da usare per archiviare i dati (fogli di carta, dischi o nastri magnetici, dischi ottici);
- la **dimensione massima** dell'archivio: per esempio il numero massimo di scaffali occupati in un archivio cartaceo o di abbonati in un elenco telefonico;
- il modo con cui i dati sono strutturati e collegati tra loro, cioè l'**organizzazione dell'archivio**(**Legami tra le tabelle**)

Legami tra tabelle

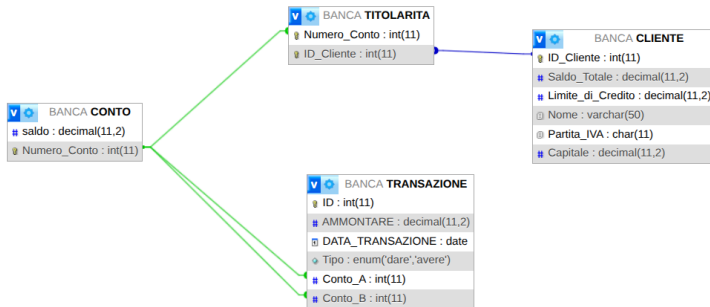


Figura: Legami tra tabelle

Dopo aver creato l'archivio, su di esso si possono effettuare operazioni di:

- **manipolazione**, cioè inserimento di nuovi dati o variazione dei dati registrati;
- **interrogazione**, cioè reperimento all'interno dell'archivio delle informazioni necessarie.

- Gli archivi memorizzati su tali supporti vengono detti **file** perchè in inglese la parola archivio viene tradotta con file.
- In informatica questo termine serve a indicare in modo generico qualsiasi informazione che può essere registrata sui supporti di memoria, come un testo, un programma, un comando del sistema operativo o un grafico.
- Nelle **applicazioni** con il computer, un archivio può contenere qualsiasi tipo di informazione, non solo di tipo testuale ma anche di tipo multimediale.
- Nelle procedure gestionali comunque, nella maggior parte dei casi, gli archivi sono costituiti da insiemi di record omogenei, nel senso che ciascun archivio possiede un tracciato predefinito e uguale per tutti i record in esso contenuti, che si dicono **record logici** (file di record).

- I supporti per registrare i dati prendono il nome di **memorie di massa** perché possono contenere notevoli quantità di dati.
- Si chiamano anche **memorie ausiliarie**, perché costituiscono un'estensione della memoria centrale di un computer e consentono, a differenza della memoria centrale che è una memoria volatile, la permanenza delle registrazioni nel tempo.
- Tali apparecchiature sono esterne all'unità centrale e quindi sono **unità periferiche di memoria** (o semplicemente periferiche).

Le memorie di massa

Le memorie di massa sono caratterizzate da alcuni parametri usati abitualmente per illustrarne le prestazioni:

- il **tipo di accesso ai dati**, che può essere diretto (o random) come nei dischi oppure sequenziale come nei nastri;
- la **capacità**, in pratica la quantità di dati che il supporto è in grado di contenere; si misura in Megabyte, Gigabyte e Terabyte;
- il **tempo medio di accesso** misurato in millisecondi (ms), costituito dal tempo medio necessario per ritrovare i dati e per trasferirli nell'unità centrale;
- la **velocità di trasferimento** dei dati dalla memoria di massa alla memoria centrale misurata in KB/s o MB/s, in altre parole il numero di byte trasferiti dal supporto alla memoria del computer in un secondo.

I file e le memorie di massa



Il file system

- Il modulo del sistema operativo che svolge le funzioni di gestore dei file viene chiamato **file system**.
- Esso è costituito dall'insieme delle routine (funzioni) che consentono all'utente-programmatore di usufruire degli archivi sulle memorie di massa, senza preoccuparsi dei dettagli delle operazioni di input/output (I/O) e facendo riferimento ai file solo con nomi simbolici.
- Il file system regola l'organizzazione, l'assegnamento, la protezione e il ritrovamento di insiemi di dati, cioè di file. In particolare esso svolge le seguenti funzioni:

Organizzazione degli archivi

Organizzazione sequenziale

L'organizzazione sequenziale

- L'**organizzazione sequenziale** consiste nel registrare i record uno di seguito all'altro, in modo sequenziale, intervallati da sequenze di caratteri che indicano la fine del record.
- Essa consente l'uso di **record a lunghezza variabile** e ha come modello di file il pacco di schede o un file su nastro magnetico dove è possibile accedere a un record solo dopo aver visitato tutti i record che lo precedono.
- Si parla in tal caso di **accesso sequenziale**.



Accesso sequenziale



Organizzazione sequenziale

Figura: Accesso sequenziale

L'organizzazione sequenziale

- Questo tipo di organizzazione, semplice da gestire, consente l'uso di record aventi lunghezza diversa l'uno dall'altro.
- L'organizzazione sequenziale presenta dei **limiti** in fase di ritrovamento dei dati quando il numero dei record diventa elevato.
- Risulta invece un'organizzazione **efficace per file di piccole dimensioni**, per esempio file di testo, e che comunque possono essere pensati come un flusso di caratteri intervallati da sequenze di caratteri che indicano la fine del record.

L'organizzazione sequenziale

- Un file a organizzazione sequenziale può essere utilizzato solo per **scrivere nuovi record**, per **leggere record** o per aggiungere record in coda a quelli già registrati.
- La scrittura di record a partire da una qualsiasi posizione, eseguita su un file già esistente, provoca di norma la cancellazione di tutti i record che lo seguono; perciò la riscrittura dei record non è permessa.

Organizzazione ad accesso diretto

L'accesso diretto al record

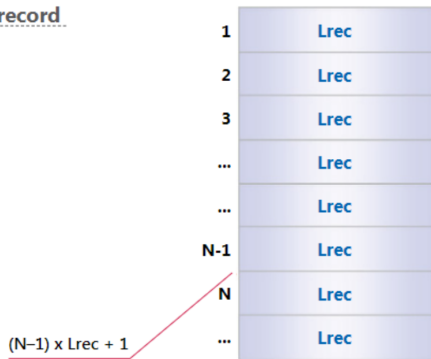


Figura:

L'organizzazione ad accesso diretto

- I file con *record di lunghezza fissa*, cioè aventi tutti la stessa lunghezza, sono di uso comune nelle applicazioni gestionali (record del cliente, record dell'articolo di magazzino, record anagrafico dello studente)
- In questo caso si può calcolare la posizione del primo carattere di un qualsiasi record, nota la sua posizione nel file.

Organizzazione ad indici

L'organizzazione ad indici (accesso associativo)

Nelle applicazioni gestionali non basta poter accedere a un record mediante la posizione che occupa in un file, ma bisogna riuscire a identificare un record in base a informazioni contenute nel record stesso (**accesso associativo**).

Read('Anagrafe', 'ROS2', Buffer);

Anagrafe

CARR	Carrara	Alessandro
ANZA	Anzani	Antonio
CATT	Cattaneo	Mirella
GANA	Ganapini	Walter
ROSS	Rossi	Giuliano
BIAN	Blanchi	Francesca
CAVA	Cavallotti	Ennio
ERMO	Ermolli	Marco
ROS2	Rossi	Piercarlo
BERG	Bergantini	Mario
MAGN	Magnani	Gianni
DOTT	Dotti	Laura
MARE	Marenzi	Giuliana
LORE	Lorenzetti	Carla
VE NE	Venezian	Luca

Figura: Organizzazione ad indice

Esempio Pratico: Archivio Clienti

Immagina di avere un database di 10.000 clienti. Vuoi recuperare i dati di Giovanni Bianchi, ma non sai in quale posizione si trova il suo record nell'archivio.

Come Funziona l'Accesso Associativo?

- **Indice Creato:** Un sistema crea un “indice” che associa ogni nome cliente alla posizione del suo record.
- **Ricerca Diretta:** Inserisci “Giovanni Bianchi”, l'indice ti restituisce istantaneamente la posizione 892.
- Il sistema accede direttamente al record senza scorrere tutto l'archivio.

Anche se devi “consultare” l'indice, lo fai su un dataset molto più piccolo e con una struttura ottimizzata per la ricerca. Questo trasforma un'operazione lenta (scansione completa) in una quasi istantanea.

- L'**accesso associativo** si ottiene usando file ad accesso diretto e metodi per calcolare la posizione occupata dal record che contiene le informazioni cercate.
- Negli archivi con organizzazione ad indice i record sono identificati attraverso un elemento caratteristico (**chiave**) che di norma è una variabile alfanumerica, per esempio la matricola del dipendente di un'azienda, il codice di un articolo, il codice del cliente.

L'organizzazione ad indici

1. La chiave (il “nome”):

Ogni record (foglio) ha un elemento unico:

- Matricola del dipendente (ID123)
- Codice articolo (ART456)
- Codice cliente (CLI789)

2. L'indice (l'“indirizzario”):

Il computer crea una tabella che associa ogni chiave alla posizione del record.

3. La ricerca:

Quando chiedi “trovami il cliente CLI789”:

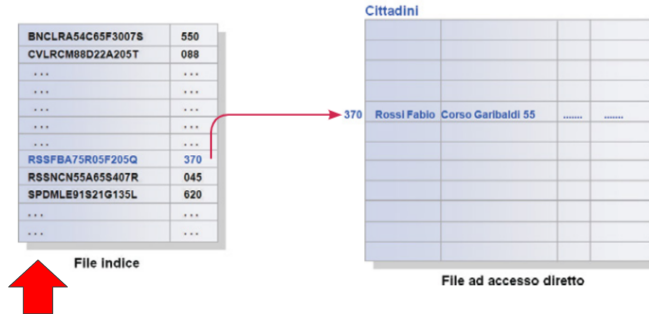
- Il computer guarda l'indice → trova “CLI789 → Posizione 300”.
- Salta direttamente al record 300 → nessuna scansione totale!

L'organizzazione ad indici

- Organizzazione ad indici implica la possibilità di leggere e scrivere record in base al valore della chiave identifica l'**accesso a chiave**.
- I clienti sono identificati in modo univoco per mezzo di un codice e i record sono ordinati secondo l'ordine di arrivo in archivio.
- Per permettere l'**accesso a chiave** l'organizzazione dell'archivio deve essere simile a quella di un libro dotato di indice analitico: in tale indice sono elencate in ordine alfabetico le parole chiave, che richiamano gli argomenti e i concetti trattati nel testo, affiancate dalla pagina corrispondente del libro.
- Il lettore cerca la parola nell'indice analitico secondo un metodo di **ricerca binaria** (o ricerca dicotomica): essendo l'elenco ordinato, il numero di pagina funziona da puntatore della pagina dove viene trattato l'argomento.
- Basta allora costruire un **indice ordinato** in base al valore della chiave e abbinare a ogni valore del codice un puntatore al record posto in un file ad accesso diretto.

- In un file con **organizzazione sequenziale a indici**, accanto alla zona con i record, memorizzati in un file ad accesso diretto, è gestita una tabella delle chiavi o **file indice** con l'elenco ordinato delle chiavi.
- La ricerca di un record avviene con una ricerca binaria nella tabella delle chiavi per recuperare la posizione del record cercato nel file ad accesso diretto.
- In questo modo si riesce ad accedere al record specificandone il valore della chiave (accesso a chiave).

Read ('Cittadini', 'RSSFBA75R05F205Q', Buffer);



Le chiavi sono ordinate in ordine alfabetico.

Figura: Costruzione dell'indice - Chiave primaria

È possibile costruire un **indice**:

- sia sul campo che identifica univocamente un record, detto **chiave primaria** (per esempio, il numero di matricola di un dipendente)
- sia su altri campi anche non univoci (per esempio, il cognome del dipendente).

In questo caso, accanto all'indice della chiave primaria, devono essere costruiti gli indici per tutti gli altri campi chiave considerati, in modo da consentire l'accesso a un record sia tramite la chiave primaria, sia tramite il valore di altri campi.

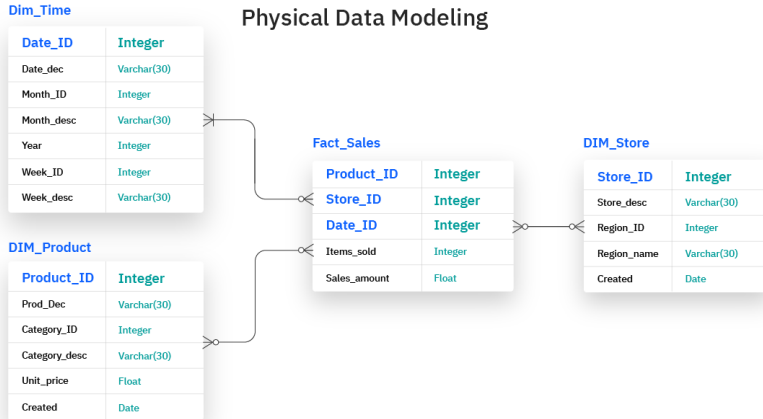
Le basi di dati

Con il termine **basi di dati** (o database) si indicano in informatica gli archivi di dati:

- organizzati in modo integrato
- progettati con tecniche di modellazione dei dati
- gestiti sulle memorie di massa dei computer utilizzando appositi software

con l'obiettivo di manipolare e trovare in modo efficiente i dati memorizzati, superando i limiti presenti nelle organizzazioni tradizionali degli archivi.

Figura: Basi di dati



Quando si parla di **efficienza** e di **produttività** dell'organizzazione degli archivi si intende la possibilità di:

- **ritrovare** facilmente le informazioni desiderate, anche con diversi criteri di ricerca e per scopi diversi;
- gestire i dati con una buona **velocità di elaborazione**;
- garantire la **sicurezza dei dati e l'integrità** delle registrazioni.

Tutto questo diventa rilevante quando la gestione si riferisce a una mole considerevole di dati. Devono anche essere offerte misure di sicurezza per impedire che il database venga danneggiato da interventi accidentali o non autorizzati, garantendo l'integrità dei dati e la consistenza del database.

- **Integrità** significa garantire che le operazioni effettuate sul database da utenti autorizzati non provochino una perdita di consistenza ai dati.
- **Consistenza** degli archivi significa assicurare che i dati in essi contenuti siano significativi e siano effettivamente utilizzabili nelle applicazioni.

- I database favoriscono gli utenti nel manipolare i dati e liberandolo dagli aspetti riguardanti la collocazione fisica delle registrazioni sui supporti degli archivi.
- Viene così superata la visione tradizionale dell'organizzazione degli archivi basata su file che obbliga il programmatore ad accedere ai dati con comandi espliciti di apertura, chiusura, lettura, scrittura e posizionamento sui file contenenti i dati degli archivi da gestire.

Il database, è una collezione di archivi (file) di dati ben organizzati e ben strutturati, che sono gestiti in modo integrato e che costituiscono una base di lavoro per utenti diversi con programmi diversi.

I prodotti software per la gestione dei database sono indicati con il termine **DBMS**, acronimo di **DataBase Management System**.

I DBMS consentono all'utente di focalizzare l'attenzione solo sull'applicazione che utilizza i dati dell'archivio consentendone la gestione anche a utenti che nulla conoscono dell'hardware sottostante.

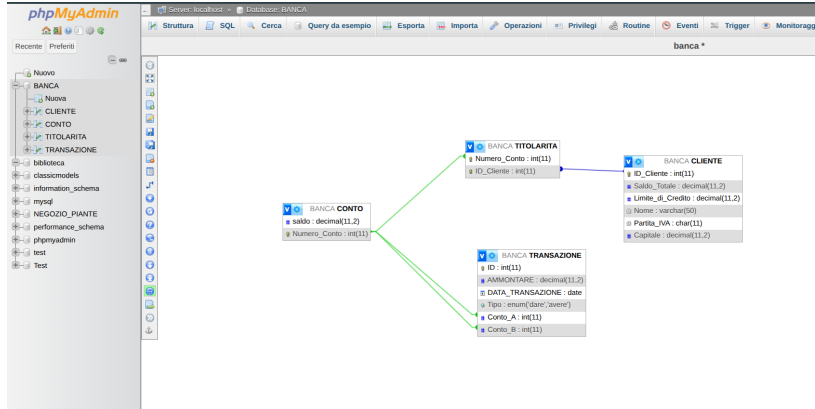


Figura: DBMS

In questo modo l'utente del database può concentrare la sua attenzione sul:

- progetto degli archivi
- sulla gestione e sul ritrovamento delle informazioni

senza preoccuparsi del modo con il quale avviene l'organizzazione fisica dei dati sulle memorie di massa, compito che rimane a carico del software di gestione della base di dati.

Questo crea una distinzione tra:

- **utenti finali** che utilizzano le informazioni contenute nel database
- **amministratori del database** che si occupano della progettazione e della manutenzione degli archivi

Questa distinzione corrisponde alla differenza tra:

- **struttura concettuale** → modo attraverso il quale l'utente pensa all'organizzazione e al ritrovamento dei dati
- **struttura fisica dei dati** → tecniche utilizzate dal sistema operativo per registrare e leggere negli archivi

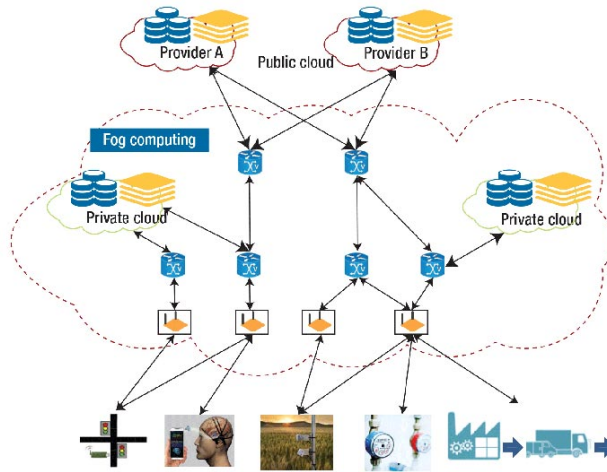
Database vs DBMS

A questo proposito si tenga ben presente anche la differenza tra:

- **Database** come insieme di dati
- **DBMS** come sistema per la gestione del database

così come nell'uso degli archivi tradizionali c'è una distinzione tra archivi di dati e file system.

- Gli utenti della base di dati elaborano in modo locale gli archivi che hanno a disposizione nel proprio sistema e nello stesso tempo accedono in modo remoto a sistemi centrali attraverso le linee di comunicazione.
- Gli archivi integrati che costituiscono la base di dati aziendale possono risiedere su un unico computer oppure possono essere distribuiti sulle memorie di massa di computer diversi, facenti parte di una rete aziendale, i cui nodi possono essere anche fisicamente lontani: in questo caso si parla di **database distribuiti**.



Ridondanza e inconsistenza dei dati.

- La ridondanza nei dati è frequente nell'approccio basato su file indipendenti.
- La duplicazione dei dati richiede di inserirli più volte, con l'aumento dello spazio occupato dai dati, ma, soprattutto, la ridondanza può portare all'**incongruenza** nel caso in cui un dato sia aggiornato in un archivio e non in un altro, oppure siano presenti valori diversi per lo stesso dato.

- L'incongruenza porta a sua volta all'**inconsistenza** dei dati, cioè i dati a disposizione non sono più affidabili, perché non si sa quale sia quello corretto.

Difficoltà nell'accesso ai dati

L'accesso ai dati dipende dall'organizzazione scelta per gli archivi. Di conseguenza il programmatore deve accedere agli archivi con le modalità previste per l'organizzazione scelta e deve limitare le operazioni ammissibili sugli stessi.

Isolamento dei dati, dati di differente formato.

I dati sono dispersi tra diversi file e sono rappresentati con differenti formati a causa, per esempio, dell'uso di differenti linguaggi nello sviluppo di diverse parti di un'applicazione: di conseguenza diventa difficile collegare i dati tra di loro e integrarli.

Dipendenza dai dati.

I programmi sono dipendenti dagli archivi che gestiscono, perché i linguaggi di programmazione richiedono di specificare, all'interno di ogni programma, quali sono gli archivi utilizzati e la struttura dei loro record.

Difficoltà nel gestire l'integrità dei dati.

I vincoli di integrità si possono rappresentare solo scrivendo un apposito codice nei programmi che manipolano i dati.

Di conseguenza i vincoli di integrità non sono dichiarati esplicitamente ma sono impliciti, distribuiti e nascosti nel software e di difficile documentazione.

Interrogazioni predefinite.

È possibile accedere ai dati solo tramite un numero limitato di applicazioni sviluppate ad hoc. Per ogni altra esigenza informativa bisogna sviluppare nuove applicazioni.

Complessità nell'aggiornamento del software e delle strutture dati

Ogni aggiornamento del software e ogni nuova applicazione deve tenere conto dei vincoli d'integrità inserendoli nei programmi mediante specifiche istruzioni.

Qualsiasi modifica alla struttura del record richiede la modifica di tutti i programmi che utilizzano quel record.

Modelli per i database

- Il database è un modello della realtà considerata: i contenuti della base di dati rappresentano gli stati in cui si trova la realtà da modellare.
- I cambiamenti che vengono apportati alla base di dati rappresentano gli eventi che avvengono nell'ambiente in cui opera l'azienda. Per descrivere i dati, il loro significato, come sono correlati e i vincoli definiti su di essi si fa uso di **modelli dei dati**.

Essi sono classificabili secondo diversi livelli di generalità:

- **modello concettuale** (o modello a livello di oggetti)
- **modello logico dei dati** (o modello a livello di record)
- **modello fisico dei dati**, che si occupa delle modalità e delle tecniche usate nella registrazione sulle memorie di massa

L'uso efficace dei dati organizzati in un database presuppone un attento lavoro di progettazione iniziale, che viene fatto con riferimento ai dati che si vogliono memorizzare e successivamente elaborare.

Tra i numerosi modelli proposti per la progettazione concettuale, il più noto è il **modello Entità/Associazioni**, indicato come **modello E/R** dal termine inglese Entity/Relationship. Nella costruzione del modello E/R di una realtà si individuano:

- gli oggetti che la compongono, detti **entità**
- poi gli **attributi**, che rappresentano le caratteristiche delle entità individuate
- e infine le **associazioni**, che descrivono le correlazioni logiche tra entità

Entità, attributi, associazioni sono rappresentati graficamente in un **diagramma E/R**.

Il modello concettuale

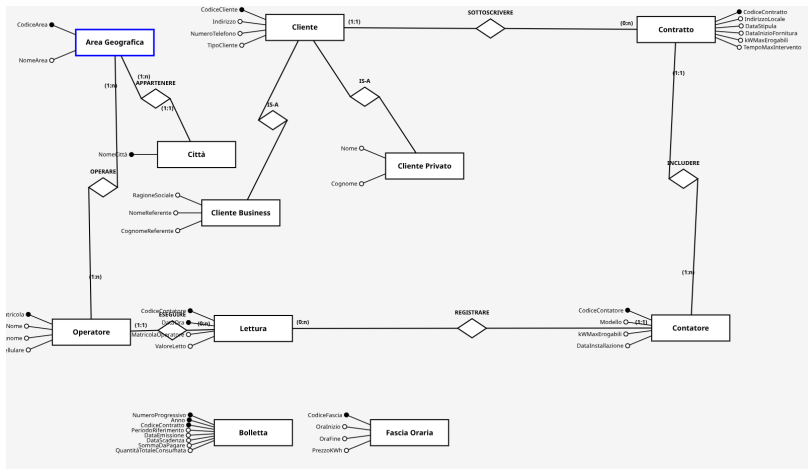


Figura: Modello concettuale

A partire dallo schema concettuale Entità/Associazioni, un database può essere progettato e realizzato passando al **modello logico**, cioè alle strutture che organizzano i dati, in modo da consentire le operazioni di manipolazione e di interrogazione.

Nello sviluppo della teoria dei database, a partire dagli anni Sessanta, sono emersi tre diversi tipi di modelli a livello di record per le basi di dati:

- il modello gerarchico
- il modello reticolare
- il modello relazionale

- I modelli gerarchico e reticolare sono stati definiti attraverso un processo di astrazione da sistemi già implementati, mentre il modello relazionale è stato definito a livello teorico prima di qualsiasi implementazione sul computer.
- Il modello relazionale nasce nel 1970, proposto da **Edgar F. Codd**, ricercatore IBM, come idea di un modello logico molto semplice e nello stesso tempo in grado di superare i limiti degli altri modelli utilizzati.

Modello relazionale

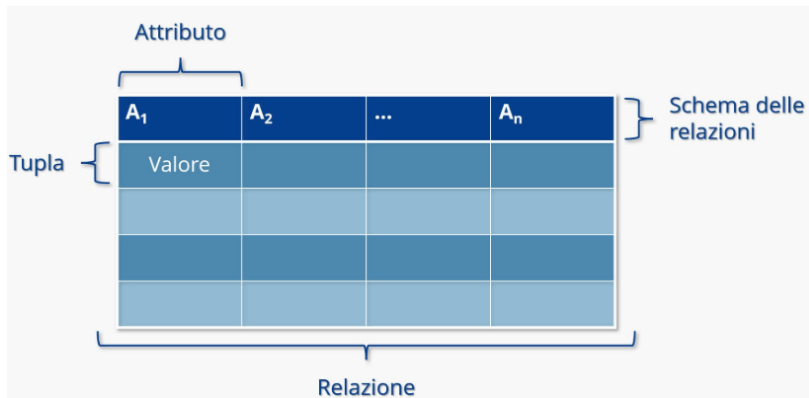


Figura:

Il modello relazionale

Il modello relazionale si basa su alcuni concetti fondamentali tipicamente matematici e assegna grande importanza all'uso rigoroso del linguaggio matematico, con due obiettivi importanti:

- utilizzare un linguaggio conosciuto a livello universale, quale è il linguaggio matematico;
- eliminare i problemi di ambiguità nella terminologia e nella simbologia

Il modello relazionale è un **modello basato sui valori**. Le associazioni tra entità sono descritte solamente tramite i valori assunti da campi, nelle righe delle tabelle che modellano le entità stesse, senza fare uso di puntatori.

- La diffusione del **modello relazionale** è cresciuta nel tempo in coincidenza con la crescita della capacità delle memorie centrali e della potenza di calcolo dei processori, che hanno permesso di manipolare le tabelle con sempre maggior rapidità.
- Invece i database costruiti sul **modello gerarchico e reticolare**, meno esigenti in termini di prestazioni dell'hardware, erano più adeguati alle limitate prestazioni delle macchine degli anni Settanta.

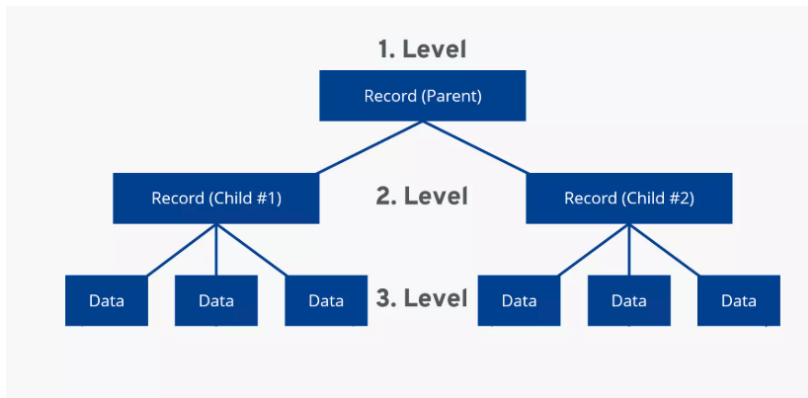


Figura: Modello gerarchico

- Le operazioni sui database gerarchici e reticolari sono complesse, poiché agiscono su singoli record che vengono identificati specificando i percorsi per ritrovarli.
- Si dice in tale caso che l'approccio adottato nell'elaborazione dei dati è di tipo **procedurale**, ovvero bisogna indicare al computer come deve fare per trovare i dati.
- I sistemi relazionali adottano invece un approccio di tipo **dichiarativo** nell'elaborazione delle informazioni, in quanto viene specificato cosa si vuole trovare, mentre i percorsi per trovare i dati sono a carico del sistema.

Il modello relazionale

- Tra i modelli di database illustrati, il modello relazionale è il più diffuso nella quasi totalità dei prodotti commerciali DBMS, nei sistemi di elaborazione grandi, medi e piccoli.
- Proprio questo modello ha determinato un'ampia diffusione dei programmi per la gestione di database anche per i personal computer.

Il **modello fisico** dei dati ha la funzione di descrivere il modo con il quale un dato modello logico è realizzato concretamente sulle memorie di massa del computer.

Nel caso del modello relazionale, il modello fisico precisa come sono realizzate le tabelle, il modo per implementare i vincoli sui dati che le compongono, come rappresentare le associazioni tra tabelle, come costruire gli indici sui campi di una tabella e così via.

Il modello fisico si occupa di problemi trattati dai progettisti di uno specifico DBMS. Esempi di prodotti DBMS che gestiscono i modelli fisici dei dati sono Oracle, DB2, MySQL, SQL Server, Access.

Architettura a tre livelli e indipendenza dei dati

I software DBMS seguono, di fatto, l'impostazione concettuale dell'**architettura a tre livelli ANSI-SPARC**, proposta nel 1975 da un comitato (Standard Planning And Requirements Committee) dell'ANSI (American National Standards Institute).

Secondo questa impostazione i dati sono descritti secondo tre differenti livelli di astrazione, mediante opportuni schemi. I tre livelli sono indicati come:

- il livello esterno
- il livello logico
- il livello interno

Il **livello esterno (viste)** rappresenta la visione del database da parte dell'utente. Descrive la parte del database che è importante per il singolo utente o per un gruppo di utenti.

Architettura a tre livelli e indipendenza dei dati

In generale in un database ci sono tanti schemi esterni quante sono le classi di utenti con differenti esigenze e visione dei dati.

Il **livello logico** rappresenta la visione complessiva del database dal punto di vista logico, indipendentemente dalle modalità e dalle tecniche di memorizzazione dei dati.

Il livello logico descrive, con le convenzioni del modello di database scelto, entità, attributi, associazioni e vincoli. Il livello logico è la cerniera tra ciò che vedono gli utenti e quello che viene effettivamente memorizzato nel computer.

Architettura a tre livelli e indipendenza dei dati

Naturalmente c'è un solo schema logico di database.

Il **livello interno (fisico)** riguarda la rappresentazione fisica del database nel computer e fornisce informazioni in merito alla realizzazione concreta del livello logico.

Il livello interno entra nel merito dei dettagli implementativi delle forme di memorizzazione.

Esempio.

Una tabella di un database relazionale può essere rappresentata con un file ad accesso diretto e con una struttura a indice sequenziale per rappresentare la sua chiave primaria; nel caso di un database gerarchico o reticolare, è di competenza del livello interno la realizzazione, mediante puntatori, degli archi che collegano i nodi.

Architettura a tre livelli e indipendenza dei dati

In un semplice database:

- il livello esterno è composto dalle cinque differenti visioni dei dati che hanno gli impiegati dei cinque dipartimenti;
- il livello logico è costituito dalla coppia di tabelle Dipartimenti e Studenti;
- il livello interno descrive come sono memorizzate le tabelle Studenti e Dipartimenti e dipende dal DBMS scelto;
- al di sotto del livello interno ci sono i bit, o meglio i blocchi di byte, su disco che sono di competenza del sistema operativo.

Il DBMS mantiene al proprio interno la descrizione delle viste logiche, ovvero degli schemi esterni, dello schema logico e dello schema interno del database. In base a queste informazioni, esso è in grado di attuare la corrispondenza (mapping) tra le viste logiche e lo schema logico del database e tra lo schema logico e il livello interno.

Si supponga per esempio che un impiegato di Ingegneria desideri accedere tramite un'apposita interrogazione alle informazioni anagrafiche di uno studente di nome Nino Verdi.

- Il DBMS traduce la richiesta nella ricerca dei record di Studenti con quel nome e cognome, ma con il campo CodDip di valore Ing, identificando il record di Studenti con Matricola = 2340.

Architettura a tre livelli e indipendenza dei dati

- In base alle informazioni sull'organizzazione degli archivi contenute nello schema interno del database, il DBMS riesce a localizzare il record corrispondente allo studente con quel valore di matricola. Il risultato di questa operazione è del tipo: la riga cercata si trova nel record di posto 25 del file ad accesso diretto.
- L'abbinamento tra numero di record e numero di blocco del file e la successiva ricerca su disco vengono attuati dal DBMS e dal sistema operativo con modalità che dipendono dal DBMS usato e dallo specifico sistema operativo.

Architettura a tre livelli e indipendenza dei dati

L'architettura a tre livelli ha il vantaggio di semplificare la visione del database da parte degli utenti, in quanto essi vedono solo le informazioni alle quali sono interessati, secondo la propria visione del mondo reale.

Inoltre l'architettura a tre livelli limita le possibilità di errore e permette di affrontare i problemi di riservatezza in modo puntuale.

Architettura a tre livelli e indipendenza dei dati

L'architettura a tre livelli dei database realizza meccanismi di astrazione dei dati e assicura la cosiddetta **indipendenza dei dati**.

Con questo termine si vuole indicare il fatto che i livelli superiori non sono influenzati, entro certi limiti, dai cambiamenti che avvengono nei livelli inferiori dell'architettura.

Si identificano due livelli di indipendenza dei dati:

- l'indipendenza logica
- l'indipendenza fisica

Il DBMS è il software che consente di costruire e gestire una base di dati, realizzandola nella pratica su memoria di massa, a partire da un progetto e da uno schema dei dati definiti a livello concettuale e tradotti poi in un modello logico dei dati.

Le funzioni che il DBMS, attraverso i suoi moduli software, è in grado di offrire agli utenti del database si possono raggruppare in sei categorie.

1. L'implementazione del modello logico sul sistema di elaborazione.

- Definizione dei dati e delle strutture dati derivate dallo schema logico (tipicamente le tabelle del modello relazionale), con produzione della documentazione sul modello.

Definizione dei sottoschemi (viste logiche) che consentono agli utenti di accedere ai soli dati ai quali sono interessati in base alle proprie esigenze applicative.

Una vista logica è una tabella virtuale, mentre le tabelle del database sono indicate come tabelle primarie. Le viste logiche sono finestre dinamiche sulle tabelle del database, in quanto ogni modifica ai dati delle tabelle primarie si riflette sulle viste corrispondenti e viceversa.

2. La manipolazione e l'interrogazione sulla base di dati

- Inserimento dei dati nel database e trattamento dei dati già registrati con operazioni di modifica o cancellazione.
- Interfaccia tra i programmi degli utenti (scritti con i tradizionali linguaggi di programmazione) e la base di dati, utilizzando le funzionalità del DBMS per migliorare l'organizzazione dei dati e le prestazioni dei programmi nelle operazioni di ritrovamento dei dati.
- Accesso ai dati del database attraverso interfacce grafiche e semplici comandi che facilitano l'utente non specialista.

3. Il controllo dell'integrità dei dati

- Integrità sui dati, in relazione ai valori che possono assumere e alle interdipendenze tra dati di differenti tabelle.
- Integrità definite dall'utente, cioè vincoli che sono specifici per un particolare database, come conseguenza di politiche commerciali dell'impresa oppure di norme legislative e fiscali.

4. La sicurezza e la protezione

- Garanzia di sicurezza dei dati contro i danni causati da malfunzionamenti di componenti hardware o software o da interventi dolosi.
- Protezione dei dati da eventuali danneggiamenti per garantire l'integrità dei dati, offrendo anche la possibilità di attivare procedure di recovery in caso di perdita dei dati.
- Autorizzazione degli utenti che accedono alla base di dati e protezione dei dati dagli accessi non autorizzati.
- Controllo degli accessi al database in modo concorrente da parte di più utenti.

5. Il supporto alle transazioni

- Garanzia che tutte le operazioni che compongono la transazione siano completamente e correttamente eseguite oppure che non ne sia eseguita alcuna.

Una **transazione** consiste in un insieme di operazioni di interrogazione o di modifica del database che devono essere eseguite come se fossero un'unica operazione.

Un esempio di transazione è un'operazione di trasferimento di fondi tra conti correnti bancari. L'importo trasferito deve essere tolto da un conto e aggiunto a un altro. Entrambe le operazioni devono essere eseguite per poter affermare di avere effettuato il trasferimento e non è tollerabile che sia eseguita una sola delle due operazioni. Il supporto alle transazioni deve garantirne il corretto completamento anche in caso di esecuzione concorrente.

6. L'ottimizzazione delle prestazioni

- Uso delle tecniche che ottimizzano l'occupazione della memoria di massa e i tempi di accesso ai dati registrati sui supporti di memorizzazione.
- Implementazione delle opportune strategie per ottimizzare le interrogazioni minimizzando i corrispondenti tempi di risposta o, più in generale, i costi dell'interrogazione.

Per poter attuare questo ulteriore livello di ottimizzazione, il DBMS deve disporre di informazioni sulla numerosità dei record nelle tabelle e sulla distribuzione dei dati di uno specifico campo o di un dato indice: questi dati informativi costituiscono le **statistiche del database** o dati statistici gestiti dal DBMS.

L'esempio mostra come le statistiche possano essere usate dal modulo di ottimizzazione delle interrogazioni per costruirle nel modo più efficiente. Poiché la distribuzione dei dati in un campo o in un indice è un valore che cambia nel tempo (si pensi, per esempio, alla distribuzione degli studenti nei diversi dipartimenti), le statistiche sono un tipo di metadato dinamico gestito dal DBMS.

I DBMS dispongono spesso di generatori di rapporti, cioè componenti specializzate nella produzione di report.

Si tratta, in pratica, di applicazioni progettate per presentare i dati estratti da un database in formati adatti alla consultazione da parte di una specifica categoria di utenti.

Un DBMS gestisce anche il **dizionario dei dati** (o catalogo del database), contenente informazioni su:

- nomi delle tabelle e delle colonne;
- associazioni;
- viste logiche;
- vincoli di integrità;
- utenti e proprietari;
- autorizzazioni degli accessi.

Il dizionario contiene i **metadati**, cioè i dati che descrivono i dati contenuti nel database. Anche le informazioni del dizionario sono organizzate in modo relazionale, cioè come valori in tabelle. Gli utenti autorizzati possono quindi accedere alle informazioni del dizionario con le stesse modalità con le quali operano per ritrovare i dati nel database.

Le prestazioni del DBMS vengono attivate dall'utente usando appositi comandi, che costituiscono a tutti gli effetti un linguaggio attraverso il quale l'utente può comunicare con il sistema di elaborazione che gestisce il database.

I comandi che il DBMS mette a disposizione possono essere classificati nelle seguenti categorie di linguaggi:

Linguaggio per la descrizione dei dati, delle tabelle e delle viste, delle associazioni tra tabelle, dei vincoli di integrità e dei controlli relativi alla sicurezza, detto **DDL** (Data Definition Language).

Il DDL rappresenta lo strumento attraverso il quale l'utente, facendo riferimento al proprio schema logico, ordina al DBMS la creazione della struttura fisica del database.

Il DDL possiede inoltre specifici comandi per definire i sottoschemi relativi alle applicazioni contenute nei programmi dei singoli utenti, oltre che per eliminare tabelle e viste già esistenti.

Linguaggio per il trattamento (o manipolazione) dei dati contenuti nel database, detto **DML** (Data Manipulation Language), che consente le usuali operazioni di accesso per inserimenti, modifiche o cancellazioni.

Linguaggio per le interrogazioni alla base di dati, detto **QL** (Query Language), che consente il ritrovamento dei dati che interessano, sulla base dei criteri di ricerca richiesti dall'utente. Il linguaggio denominato DDL comprende anche la possibilità di generare in modo automatico, a partire dalle tabelle e dai loro attributi, le maschere video che servono a facilitare l'utente nell'inserimento, nella modifica e nella consultazione dei dati, e i prospetti di output per la rappresentazione ordinata e facilmente leggibile dei dati estratti dalla base di dati.

I linguaggi per database relazionali

Lo sviluppo e il raffinamento delle tecniche di gestione delle basi di dati hanno dato vita a linguaggi formati da comandi specifici, per consentire agli utenti un facile uso delle prestazioni del DBMS per basi di dati relazionali (detti RDBMS, cioè Relational DBMS).

Accanto alla possibilità di usare questi comandi, richiamandoli dall'interno di un programma scritto con i tradizionali linguaggi di programmazione (Cobol, C), è molto comune l'uso di linguaggi orientati alla gestione delle basi di dati, con caratteristiche di linguaggio a sé stanti.

I linguaggi per database relazionali

La diffusione del modello relazionale ha poi favorito l'uso prevalente di linguaggi non procedurali ma **dichiarativi**: in questo modo l'utente non ha la necessità di conoscere né le modalità con le quali le informazioni sono state fisicamente registrate, né i cammini per ritrovare le informazioni contenute nella base di dati.

I linguaggi per database relazionali

Si parla allora di linguaggio per basi di dati, intendendo un insieme completo di comandi che consente e facilita le operazioni di definizione del database, di manipolazione dei dati e di interrogazione da parte degli utenti: vengono cioè unificate in un unico linguaggio le funzioni dei linguaggi DDL, DML e QL.

Tipici linguaggi non procedurali per database relazionali sono:

- il linguaggio **SQL** (Structured Query Language, linguaggio strutturato di interrogazione)
- **QBE** (Query By Example, interrogazioni per esempi)

I linguaggi per database relazionali

QBE permette di eseguire interrogazioni dichiarando, in un'apposita maschera, come si dovrà presentare la risposta alla query che si vuole realizzare. Nel descrivere l'output desiderato si possono effettuare calcoli partendo dai valori dei campi, oppure esprimere condizioni che devono essere verificate dai record estratti. Inoltre il QBE permette non solo di costruire interrogazioni, ma anche di modificare e variare i dati.

Con il linguaggio SQL si può interrogare un database per estrarne informazioni. Ma non solo: nonostante il nome, SQL è un linguaggio per database completo. Con SQL si possono infatti sia definire che manipolare dati.

L'approccio dichiarativo del linguaggio è evidenziato dagli esempi introduttivi.

I linguaggi per database relazionali

I linguaggi per database relazionali si basano sulla visione tabellare dei dati. I comandi del linguaggio relazionale operano inoltre su gruppi di righe o sull'intera tabella, anziché su una riga per volta: con una sola richiesta possono essere trattati o ritrovati molti record e non solo un record per volta, come avviene con i tradizionali linguaggi di programmazione. Sono semplificate le operazioni per mettere in connessione tra loro tabelle diverse e per presentare sul video o stampare i risultati delle interrogazioni, ben impaginati, in modo da facilitarne la lettura e la comprensione.

I linguaggi per database relazionali

Nei software DBMS, soprattutto nelle implementazioni di prodotti su personal computer, sono disponibili interfacce utente, con le quali si può interagire usando la lingua nazionale, che permettono di selezionare i comandi del linguaggio attraverso menu, sottomenu e il puntamento a icone sul video, con le modalità caratteristiche di un'interfaccia grafica. Sono inoltre disponibili messaggi di aiuto, che possono essere richiamati sul video in modo contestuale, cioè nel momento in cui serve una breve spiegazione sul tipo di operazione che si vuole attivare.

Un database viene utilizzato da persone diverse, per funzioni e per applicazioni diverse.

1. L'Amministratore della Base di Dati (DBA, DataBase Administrator) con i seguenti compiti:

- implementazione del modello logico del database nel sistema di elaborazione sui supporti fisici delle memorie di massa;
- gestione e trattamento dei dati (controllo di inserimenti, modifiche, cancellazioni);
- autorizzazione degli accessi;
- definizione delle viste per accessi parziali di utenti alla base di dati;
- controllo dei programmi applicativi che richiedono l'uso del database;

- manutenzione del database nel tempo, in termini di efficienza e di ottimizzazione delle risorse;
- controllo sugli interventi di recupero, nel caso di cattivi funzionamenti, e sulle copie di salvataggio periodiche;
- controllo della disponibilità degli spazi su memoria di massa.

Il DBA utilizza, per svolgere le sue funzioni, le prestazioni e i linguaggi del DBMS e collabora con le altre figure (sistemista, analista, responsabile dello sviluppo software, programmatore) nelle diverse fasi: progettazione del database, costruzione delle applicazioni, traduzione dello schema del database nel modello fisico da creare su memoria di massa, manutenzione dei dati nel tempo.

2. I programmatori, che intendono utilizzare per le loro applicazioni i dati organizzati in un database, sfruttano un linguaggio DML, oppure comandi che sono un'estensione dei tradizionali linguaggi di programmazione, oppure un linguaggio specifico per basi di dati. In ogni caso il programmatore deve agire sotto il controllo del gruppo di progetto e di produzione del software applicativo a cui partecipa anche l'Amministratore del database.

3. Gli utenti finali possono accedere alla base di dati attraverso i comandi di un linguaggio di interrogazione, di un linguaggio QBE, oppure (per utenti finali ancora meno esperti) attraverso interfacce software che presentano sul video finestre, menu e icone.

Le transazioni

Le proprietà acide delle transazioni

L'importo trasferito deve essere tolto da un conto e aggiunto all'altro. Entrambe le operazioni devono essere eseguite per poter affermare di avere concluso il trasferimento e non è tollerabile che venga eseguita una sola delle due operazioni.

Ne segue che il sistema di gestione delle transazioni deve intervenire per effettuare un **rollback** automatico nel caso che, per una qualsiasi ragione, una delle due operazioni di aggiornamento della base di dati non vada a buon fine.

Questo comportamento caratterizza un'importante proprietà delle transazioni e prende il nome di **atomicità**. Essa è una delle cosiddette proprietà acide delle transazioni, dall'acronimo inglese **ACID** (Atomicity, Consistency, Isolation, Durability).

Le proprietà acide delle transazioni

- **Atomicità:** una transazione è un'entità atomica indivisibile. È compito del sistema di gestione della sicurezza garantire l'atomicità e riuscire a ripristinare la situazione preesistente quando necessario;
- **Consistenza:** le transazioni non devono violare i vincoli di integrità dei dati. La consistenza è gestita dal DBMS con procedure opportune;
- **Isolamento:** gli effetti di una transazione devono essere indipendenti da quello di tutte le altre transazioni eseguite in concorrenza. L'isolamento è gestito dal controllore della concorrenza.

- **Persistenza:** le informazioni in un database devono essere memorizzate in modo persistente, cioè per sempre. Gli effetti di una transazione eseguita con esito positivo (dopo l'esecuzione di un comando commit) devono essere memorizzati permanentemente nel database a cura del sistema di gestione della sicurezza e del ripristino dei dati.

Grazie per l'attenzione!