

[Indietro](#)[Avanti](#)[Ripeti](#)[Inizio](#)[Aiuto](#)[Esci](#)

Lezione 1.1 - Gli archivi ed il database

In questa lezione si impareranno i concetti fondamentali sui database

Gli **archivi** sono insiemi organizzati di informazioni, conservati su appositi supporti e disponibili per l'uso.

Organizzare i dati non è una prerogativa dell'informatica, chiunque di noi può usare certi criteri per dare una sistemazione alla sua collezione di libri, di CD o di videocassette.

Se però dovessimo organizzare ad esempio una biblioteca scolastica, con il relativo servizio prestiti, certamente dovremmo usare qualche accorgimento più complesso.

A causa della complessità crescente rapidamente con l'aumento del numero di dati da gestire, il settore della conservazione e della elaborazione dei dati è stato uno dei primi in cui l'informatica si è affermata, proprio per l'affinità fra le caratteristiche intrinseche dell'archivistica e le possibilità offerte dai nuovi strumenti.

Gli archivi possono essere **statici** oppure **dinamici** a seconda che contengano dati che non vengono praticamente mai aggiornati (i cosiddetti archivi storici, utili solo alla conservazione di dati non in uso), o invece dati dinamici, continuamente variati.

Naturalmente un archivio non può essere un insieme disordinato di dati, dai quali pescare faticosamente le informazioni desiderate: esso deve avere una struttura ed una forma ben definiti, divenendo quindi un sistema organizzato per la conservazione ed il trattamento dei dati. Tale struttura è quindi caratterizzata da una organizzazione che si distingue in :

- Organizzazione fisica (supporto), ad es. nastri e dischi
- Organizzazione logica (modalità di gestione), ad es. sequenziale, sequenziale ad indice e diretta.

Il Data Base

Dal punto di vista del software, una innovazione paragonabile a quella che nell'hardware è stata rappresentata dai PC può essere considerata la nascita dei Data Base.

I costrutti teorici dei database risalgono agli anni '70, ma solo dalla metà degli anni '80 la loro diffusione è stata tale da rivoluzionare i metodi di conservazione e manipolazione dei dati.

Tale massiccia diffusione è legata anche alla comparsa di PC, diffusi in tutti gli ambienti di lavoro: l'interazione di aspetti hardware e software ha creato una sinergia che a sua volta ha favorito l'utilizzo dei database anche da parte di utenti non esperti di informatica.

Per capire meglio che cosa è un database, e quali sono i vantaggi legati al suo impiego diamo alcune definizioni :

- **Data Base** Un insieme di informazioni strettamente correlate, memorizzate su un supporto di massa, costituenti un tutt'uno che può essere manipolato da più programmi applicativi.
- **DBMS (Data Base Management System)** Un sistema software per la gestione di basi di dati: si occupa dell'aggiornamento, della manutenzione e della consultazione di un insieme di registrazioni contenute su un supporto di massa.

Talora però si parla di Data Base intendendo sia la parte passiva (i dati), sia la parte attiva (l'insieme dei programmi rivolti alla gestione dei dati stessi)

Operando con un DBMS, esso può ricevere comandi:

- Direttamente dall'utente in modo interattivo, tramite un linguaggio di interrogazione.
- Attraverso un programma scritto in linguaggio di manipolazione dei dati, proprio

del Data Base.

- Tramite un programma scritto in linguaggio tradizionale (COBOL, Pascal, RPG ecc.)

Per il colloquio fra Data Base (o meglio i suoi programmi di gestione) e l'utente sono disponibili appositi linguaggi specializzati nello svolgimento di determinate funzioni diverse:

- **DDL (Data Description Language)** che definisce la struttura del Data Base.
- **DML (Data Management Language)** che serve per impartire comandi di elaborazione dei dati.
- **QL (Query Language)** o linguaggio di interrogazione, che presenta forti caratteristiche di interattività.

L'attività del DBMS consiste nel ricevere comandi, metterli in relazione con il modello dei dati, verificarne l'eseguibilità e tradurli in operazioni sugli archivi; nell'esecuzione di tali operazioni il DBMS (come d'altronde tutti i programmi applicativi) dipende dal Sistema Operativo per l'utilizzo concorrente delle risorse, per l'Input/Output a video e in stampa, per gli interventi sulle memorie di massa ecc.

La caratteristica principale del DBMS consiste nel mantenimento di un **modello dei dati**, ovvero di un'astrazione che permette all'utente di vedere i dati non in termini di record, files o campi, ma in termini di unità informative, direttamente riferibili agli oggetti della realtà descritti dall'informazione stessa.

Le informazioni vengono rappresentate a livello astratto da relazioni, ovvero insiemi di dati in connessione fra di loro e riferiti ad un soggetto di informazione.

Tramite il modello dei dati il Data Base permette la progettazione e la gestione degli archivi in modo totalmente svincolato :

- dalle applicazioni
- dall'organizzazione fisica dei dati

Storicamente i primi database sono stati quello **gerarchico** e quello **reticolare**, ma la nascita di quelli **relazionali** li ha fatti praticamente finire nel dimenticatoio.

Rispetto ad altri tipi di organizzazione dei dati un database relazionale presenta i seguenti vantaggi:

- non presenta ridondanza (cioè duplicazione di dati) che è causa di inutile occupazione di spazio e inoltre può causare:
- inconsistenza potenziale, cioè la possibilità che se un dato è presente più volte, un aggiornamento od una cancellazione vada ad interessare solo una di queste presenze.
- "navigazione" fra i dati possibile sia nel senso padre-figlio che all'inverso.

Il modello relazionale prevede le **entità**, cioè gli oggetti di cui ci si occupa, ed i relativi **attributi**, cioè le loro caratteristiche che ci interessano, organizzati in **relazioni** (tabelle).

Consideriamo ad esempio l'entità STUDENTI, di cui ci interessano gli attributi Nome, Cognome, Classe e Sezione :

STUDENTI(Nome, Cognome, Classe, Sezione)

Normalmente ogni attributo può assumere solo una serie di possibili valori (ad es. la Classe può essere solo 1-2-3-4-5, la Sezione solo A-B-C-D ecc.), tale gamma di variabilità viene detta **dominio** dell'attributo stesso.

Possiamo allora avere la relazione (Tabella) :

Nome	Cognome	Classe	Sezione
Franco	Rossi	1	C
Mario	Bianchi	1	A
Giorgio	Benetti	2	A
Rosa	Cellini	2	B
Federica	Assoni	3	C

ogni colonna rappresenta un attributo, ogni riga quella che si definisce una **istanza** degli attributi stessi; l'insieme degli attributi costituisce lo **schema** della relazione.

Quella che abbiamo appena visto è una **relazione base**, creata a partire da attributi base, attraverso il disegno di uno schema; vi sono però anche **relazioni virtuali** che nascono da relazioni base, o da altre relazioni virtuali, applicando appositi **operatori relazionali** (ad es. selezione, proiezione ecc.).

In una relazione non ha alcuna importanza l'ordine delle singole righe, né quello delle varie colonne; due righe si distinguono fra loro solo per i valori assunti dai loro attributi, quindi due righe con attributi eguali sono in realtà la stessa riga.

In particolare nel progettare una relazione occorre scegliere uno o più attributi caratteristici per ogni entità, detti **attributi chiave**.

Una chiave è quindi una serie di attributi tale che i valori da essi assunti identifichino in modo univoco ogni riga della relazione; essa garantisce l'identificazione di ogni riga, e per questo deve avere:

- **univocità** : non debbono poter esistere due righe con gli stessi valori per l'attributo o gli attributi assunti come chiave.
- **minimalità** : l'insieme degli attributi chiave non può essere ridotto neppure di uno, senza che la chiave perda la sua univocità.

Si dice infatti che il modello relazionale è **orientato ai valori**, la struttura del database dipende infatti dai valori assunti, e non dalla loro posizione e disposizione all'interno della base dati.

Le relazioni base rispondono ai seguenti requisiti :

- in una relazione base non esistono righe con dati tutti eguali, ciò è invece possibile per le relazioni virtuali.

- tutte le relazioni base hanno sempre una chiave **primaria** che risponde ai requisiti di univocità e minimalità e garantisce contemporaneamente la caratteristica di cui sopra.
- l'insieme delle righe e quello degli attributi non sono fisicamente ordinati: non vi è quindi nessuna corrispondenza fra disposizione dei dati in memoria e la loro presentazione.
- tutti i valori degli attributi sono **atomici**, cioè non possono essere ulteriormente suddivisi (ad es. Nome-Cognome non deve essere un attributo unico).

Una relazione con le caratteristiche sopra ricordate si dice in **Prima Forma Normale (1FN)**, esistono poi ulteriori vincoli ancora più restrittivi che portano alla seconda ed alla terza forma normale.

Mentre in linguaggi come il COBOL si usava una struttura di campi e sotto-campi :

Codice Denominazione Indirizzo

Cognome Nome Via e n° CAP Città Prov. Regione

per strutturare un Database si usano campi singoli; in questo caso per rispettare la atomicità degli attributi useremo non Codice-Denominazione-Indirizzo, ma Codice-Cognome- Nome-Via e n°-CAP-Città-Provincia-Regione.

Procedendo ancora nell'opera di "normalizzazione" della tabella possiamo osservare che alcuni attributi **dipendono** da altri: è evidente che una volta fissata la Città risultano univocamente definite anche Provincia e Regione, una corretta progettazione di un Database consiglierebbe allora di sbinare la tabella in due, la prima con Codice-Cognome-Nome-Via e n°-CAP-Città e la seconda con Città-Provincia-Regione.

In questo modo abbiamo evitato molte duplicazioni di dati in quanto ogni Provincia ed ogni Regione viene scritta una sola volta invece che decine o centinaia di volte, ripetendole nei vari record anagrafici.

Sulle relazioni possono agire (generando altre relazioni) operatori relazionali come **Unione, Intersezione, Differenza, Complemento, Selezione, Proiezione e Join**.

Le più note sono la Selezione, che consente di ottenere da una relazione base una relazione virtuale costituita solo dalle righe per cui uno o più attributi rispondono a certe condizioni; ad esempio Imponendo che il Cognome inizi con la lettera "B" otterremo per selezione dalla tabella vista prima, la nuova tabella:

Nome	Cognome	Classe	Sezione
Mario	Bianchi	1	A
Giorgio	Benetti	2	A

La Proiezione invece è costituita dall'estrazione solo di determinati attributi di tutte le righe; ad esempio dalla tabella STUDENTI per Proiezione potremmo estrarre solo Nome e Cognome ottenendo la tabella che segue:

Nome	Cognome
Franco	Rossi
Mario	Bianchi
Giorgio	Benetti
Rosa	Cellini
Federica	Assoni

Il Join è un operatore più complesso, lo definiremo perciò tramite un esempio. Supponiamo di avere oltre la tabella STUDENTI una seconda tabella CLASSI che come attributi ha solo Classe e Descrizione :

Classe	Descrizione
1	Prima
2	Seconda
3	Terza
4	Quarta
5	Quinta

Il Join fra le due tabelle, fatto sull'eguaglianza dell'attributo "Classe", consente di ottenere una relazione virtuale come la seguente :

Nome	Cognome	Descrizione	Sezione
Franco	Rossi	Prima	C
Mario	Bianchi	Prima	A
Giorgio	Benetti	Seconda	A
Rosa	Cellini	Seconda	B
Federica	Assoni	Terza	C

In particolare sui Join e sulle caratteristiche dei vari tipi esistenti torneremo molto più diffusamente nella parte dedicata al linguaggio SQL.

[Indietro](#)

[Avanti](#)

[Ripeti](#)

[Inizio](#)

[Aiuto](#)

[Esci](#)