

In questo modulo saranno presentate le caratteristiche principali teoriche propedeutiche allo studio delle basi di dati e della loro gestione.

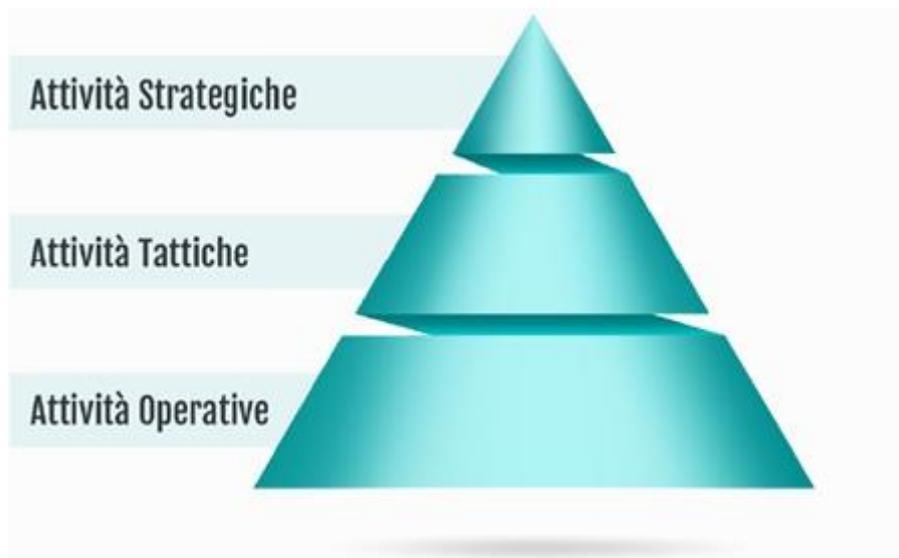
# Concetti teorici delle basi di dati

*Prof. Michele Tarantino*

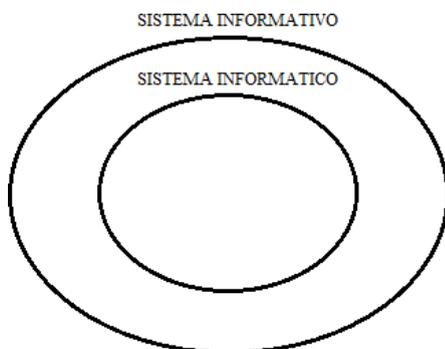
*Tutti i diritti riservati.*

*Il presente testo può essere utilizzato liberamente per motivi di studio, didattica e attività di ricerca purché sia presente il riferimento bibliografico.*

Per poter capire le caratteristiche degli accessi all'interno di una base di dati, bisogna innanzitutto andare ad enunciare e differenziare i termini dato ed informazione. Si va a definire come **dato** la misura di un fenomeno che si è interessati ad osservare mentre si può definire l'**informazione** come risultato ottenuto dall'elaborazione di dati. Prendendo come modello di base la *piramide di Anthony* ci troviamo di fronte tre livelli principali, quello **operativo**, quello **tattico** ed infine quello **strategico**.



Analizzando il livello **operativo**, troviamo una sotto piramide, anch'essa sviluppata su tre livelli: il primo sono i **dati**, quelli che entrano nell'azienda (come può essere un modulo da compilare o una registrazione effettuata da un utente), infatti spesso si parla di **flusso di dati**, ovvero una generazione continua di quest'ultimi che generalmente inoltrano **record** di dati (una struttura di dati svariata fatta da dati composti, contenente cioè un insieme di campi o elementi, ciascuno dei quali identificato da un nome univoco e da un tipo di dato contemporaneamente e in quantità minori). Questi flussi e dati vengono correlati al **sistema informativo** cioè l'insieme delle tecniche o procedure che gestisce i dati che entrano in



un'azienda; infatti, il sistema informativo è collegato al **sistema informatico**, cioè un suo sott'insieme che automatizza i dati del sistema informativo.

Un sistema informativo, su cui ci si può trovare a lavorare per un'organizzazione potrebbe già essere esistente, questo sistema prende il nome **legacy**.

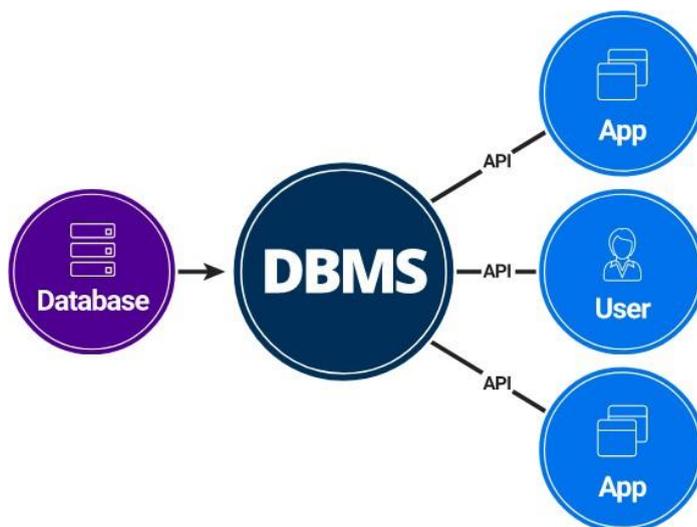
Quindi si può definire legacy un sistema informativo preesistente, che non soddisfa più le richieste di ottimizzazione o di lavoro per la ricerca dei dati e quindi si cerca di integrarlo in un nuovo sistema; questo è un problema diffuso nell'ambito pratico e ci si troverebbe di fronte alla ricreazione da 0 del sistema ma avrebbe un costo troppo elevato, quindi si cerca di integrare il sistema richiesto dal committente con il sistema legacy.

Successivamente si trova **l'informazione** ovvero il risultato dall'elaborazione dell'insieme di dati e come ultimo punto, di questa sotto piramide, troviamo gli investimenti dell'azienda stessa. I dati, invece sono a loro volta correlati tra loro e prendono il nome di **file o archivio**, dove vengono memorizzati su un supporto fisico come può essere un disco rigido in un server di un'azienda, che lavorano in modo indipendente dal resto del sistema, infatti un file creato da un programma A può essere gestito da un programma B, quindi non essendoci alcuna correlazione tra essi.

Rianalizzando il *livello tattico*, lo possiamo intendere come la disponibilità a medio termine, mentre *il livello strategico* lo possiamo intendere come le politiche di investimento a lungo termine, quest'ultime sono gestite da persone fisiche (uomo) mentre i due livelli precedenti sono automatizzate.

Di seguito l'analisi della piramide di Anthony e dei vari livelli che ne conseguono, si va ad analizzare i database o basi di dati, definiti come una grossa quantità di dati correlati tra loro.

### DATABASE e DBMS



Si va a definire come **Database** l'insieme di informazioni correlate e memorizzate su un supporto di memoria di massa, che costituiscono un tutt'uno e che possono essere manipolate da più programmi applicativi, dove quest'ultimi vengono gestiti da un **DBMS**, ovvero un sistema di software che come visto, oltre a permetterne la gestione ne

permette la privatizzazione e la resa in sicurezza di essi per l'azienda.

Andando in analisi approfondita sui DBMS ci troviamo innanzitutto di fronte alla differenza tra il DBMS stesso e i file system, dove per quest'ultimo lo si definisce come un nucleo, costituito da programmi, presente in ogni sistema operativo. La sua funzione è quella di gestire le varie operazioni sui file; questa gestione non è visibile all'utente, infatti il file system opera direttamente al servizio di altri programmi.

Il DBMS è per l'appunto un software che poggiando sul sistema operativo utilizza il file system di quest'ultimo. In questo caso, dunque, la maggiore "distanza" del DBMS dall'hardware, rispetto al file system, permette un grado di interazione maggiore. Questo significa che l'utente (programmatore, amministratore di sistema, ecc.) non dovrà più avere a che fare con record e file come visto in precedenza, ma bensì con entità astratte che rappresentano la realtà.

I DBMS possono ricevere dei comandi in molteplici modalità; la prima è sicuramente la più comune è direttamente dall'utente stesso in modo interattivo, con dei comandi particolari che appartengono alla cerchia dei linguaggi che un particolare DBMS può accettare.

Il secondo metodo è quello attraverso un programma che viene a sua volta scritto da uno o più linguaggi come accennato poco fa ed infine tramite un programma scritto tramite un linguaggio algoritmico che ingloba a sua volta alcuni comandi che appartengono a quelli accettati da DBMS.

Un termine ampiamente usato per descrivere le funzioni e i requisiti delle transazioni di un sistema di gestione di base di dati è **ACID**, acronimo di *atomicity, consistency, isolation e durability*.

- **Atomicity:** Garantisce che le operazioni di una transazione siano eseguite in modo definito "atomico" o tutte o nessuna. Questa proprietà è fondamentale per l'integrità dei dati, dove se l'operazione ha esito negativo il sistema deve essere in grado di annullare tutti i cambiamenti effettuati a partire dall'inizio della transazione.
- **Consistency:** Assicura che l'esecuzione della transazione trasformi la base di dati da uno stato iniziale coerente, ovvero senza violazioni di eventuali vincoli di integrità, ad un altro stato coerente finale.
- **Isolation:** Garantisce che l'esecuzione di una transazione sia indipendente dall'esecuzione contemporanea di altre transazioni in modo tale che transazioni concorrenti non influenzino l'una con l'altra (anche l'eventuale fallimento deve essere

tenuto in considerazione, dove una transazione non deve interferire con le altre transazioni in esecuzione).

- **Durability:** Assicura che tutti i dati vengono memorizzati in modo permanente nel DBMS, anche dopo che una transazione è stata completata con successo. Ciò vale anche o soprattutto in caso di errori di sistema o di problemi del DBMS. I log delle transazioni, che registrano tutti i processi nel DBMS, sono essenziali per la durabilità.

## LINGUAGGI di un DBMS



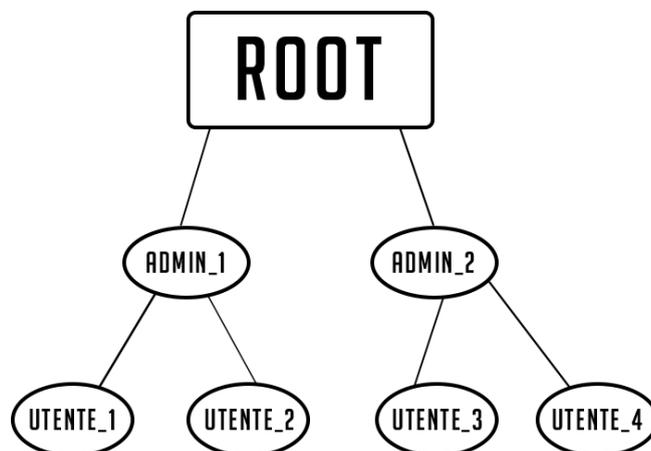
In genere i DBMS sono definiti come multiutente dove l'accesso sia in scrittura che lettura sono abilitate tramite un'interfaccia denominata SQL, acronimo di (Structured Query Language), è un linguaggio strutturato che deriva dalla logica del I° ordine ed insiemi, viene incluso nel linguaggio di programmazione

con la modalità embedded ovvero integrato/incluso e si può anche definire *chiuso* ovvero la funzione prende almeno una o l'insieme di tabelle che elaborandole restituisce UNA ed UNA SOLA tabella.

Il linguaggio SQL svolge più compiti e funzioni come la definizione di dati, aggiornamento dei dati e interrogazione o query dei dati ma vengono svolti da 4 sotto linguaggi che ora andremo a definire.

- **DDL** (Data Definition Language), è il primo sotto linguaggio dove possiamo creare tabelle, attributi, vincoli e check. Il DDL lavora quindi sulla struttura del database che però è pericoloso per l'integrità dei dati, per questo la creazione di tabelle e la manipolazione dei dati viene fatta dalla software house e non dall'azienda per non evitare la perdita dei dati o la loro cancellazione involontaria. Per questo possiamo definire un albero con i ruoli che parte dal **Root** ovvero la così detta radice, insomma colui che gestisce tutto il database e la struttura della tabella.

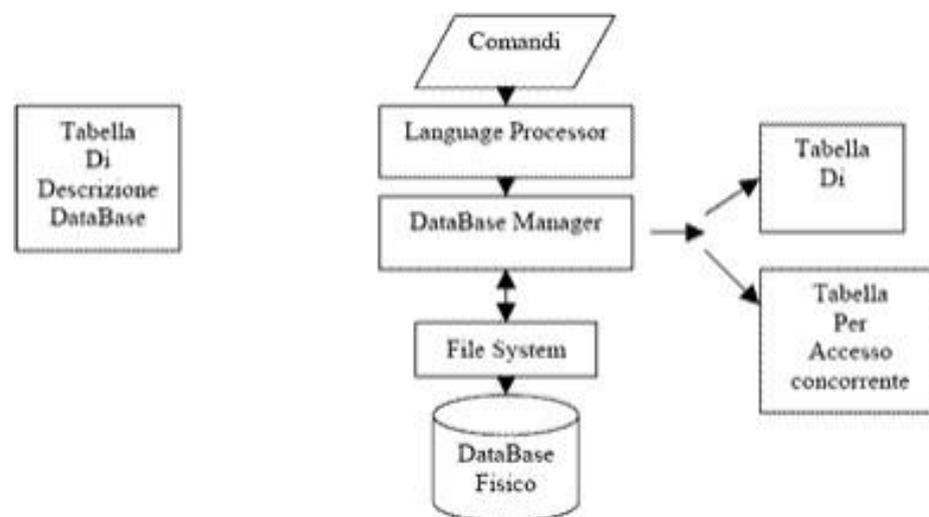
- Al di sotto del Root possiamo trovare gli amministratori del database che fanno parte del **DCL** (Data Control Language) quindi la parte di gestione e revoca dei permessi, all'interno del database. Il DCL è il più intuitivo dei 4 livelli dell'SQL poiché comprende solamente 3 comandi essenziali quali: **DENY**, **REVOKE** e **GRANT**.
- Al di sotto del livello di Amministrazione troviamo gli Utenti che vengono integrati nel **DML** (Data Manipulation Language) che lavora sui dati effettivi e permette tre operazioni principali, ovvero l'inserimento, l'eliminazione e l'aggiornamento dei dati anche se quest'ultima operazione in informatica non esiste ma avviene una sostituzione del dato precedente.
- Come ultimo e quarto sotto livello troviamo il **QL** (Query Language) ovvero quel livello che permette la SOLA visualizzazione dei dati ed eventualmente il suo ordinamento. Di seguito si allega la figura dell'albero di Root.



I DBMS a loro volta sono disposti di determinati componenti funzionali che troviamo qui di seguito:

- *Tabella di descrizione database*, è una tabella in cui è descritto il modello e le caratteristiche del database.

- *Tabella delle autorizzazioni*, è una tabella in cui sono presenti le informazioni dei vari utenti riguardo ai loro permessi di accesso ai vari dati.
- *Tabella per accesso concorrente*, ha la funzione di permettere la gestione del traffico di più richieste operati contemporaneamente sui medesimi dati.
- *Language processor*, i comandi vengono ricevuti da questa unità, che ha il compito di metterli in relazione con le specifiche di definizione del modello (contenute nella tabella di descrizione del database) e trasmetterli al Database Manager
- *Database Manager*, ha il ruolo di ricevere i comandi espressi a livello concettuale ossia operanti sul modello astratto dei dati e tradurli in comandi a livello fisico trasmettendoli al file system del sistema operativo non prima di aver fatto gli opportuni controlli sulla *tabella di autorizzazione* e sulla *tabella di accesso concorrente*.



All'interno di una base di dati, un DBA (Database Administrator), grazie ai comandi specifici del DCL è possibile fornire agli amministratori stessi flessibilità all'interno del database stesso.

**Grant** è il primo dei tre comandi base del DCL, questo comando permette di fornire uno o più permessi ad un dato utente su un particolare e specifico oggetto presente nel database.

**GRANT** <object privileges>

**ON** <object\_name>

**TO** <User\_Name>

Analizzando l'immagine allegata sopra, si può evincere la sintassi del comando stesso, dove all'interno dell'Object Privileges vi si può concedere all'utente che ritroviamo nell'ultimariga "To <>", la possibilità ad esempio di selezionare una tabella che ritroviamo nella seconda riga "On <>".

**Revoke** è il secondo comando che si andrà ad analizzare, e permette la rimozione di accesso al database da un utente che però in precedenza vi poteva accedere; Questo comando è l'esatto opposto del comando visto precedentemente e la sua sintassi è la medesima.

**REVOKE** <object privileges>

**ON** <object\_name>

**TO** <User\_Name>

**Deny** è il terzo comando che si andrà ad analizzare, quest'ultimo impedisce in maniera esplicita di ricevere delle particolari autorizzazioni come ad esempio se l'utente A è membro di un gruppo a cui è stata concessa una particolare autorizzazione all'interno del DBMS, con questo comando si va ad impedire che il singolo utente B possa ereditare autorizzazioni dall'utente A all'interno del DBMS stesso. La sintassi riportata di seguito è la medesima per gli altri due comandi visti in precedenza.

**DENY** <object privileges>

**ON** <object\_name>

**TO** <User\_Name>

All'interno di un DBMS, è necessario garantire un ottimo livello di **sicurezza e protezione** dei dati stessi che vi sono all'interno. La sicurezza delle basi di dati coinvolge diversi aspetti tra i quali i principali sono quelli etici e legali, le politiche di gestione dei dati, le componentistiche Hardware e Software del sistema e la definizione di diversi livelli disicurezza poiché il DBMS stesso deve garantire **Integrità, Disponibilità e Riservatezza** dei dati. Questi 3 termini prendono parte del **RID**, dove:

- **Riservatezza:** è la protezione dei dati trasmessi o conservati per evitarne l'intercettazione e la visione da parte di soggetti terzi non autorizzate. La riservatezza risulta necessaria per la trasmissione dei dati sensibili ed è dunque uno dei requisiti che garantiscono il rispetto della vita privata degli utenti.

- **Disponibilità:** è l'esigenza che i dati siano sempre accessibili e che i servizi funzionino anche nel caso di interruzioni dovute, ad esempio, alla cessazione dell'energia elettrica, a eventi disastrosi naturali, eventi imprevedibili e/o ad attacchi di pirateria informatica. È un requisito di primaria importanza nei casi in cui l'indisponibilità di una rete di comunicazione può generare disfunzioni rispetto all'erogazione del servizio.
- **Integrità:** la si può definire come il mantenimento invariato della struttura del database, ovvero poter gestire il database e far in modo che al suo interno vi siano sempre informazioni corrette ed esaustive.

L'integrità a sua volta differisce in molteplici definizioni che di seguito prendiamo in esame:

- **Integrità Referenziale:** è un insieme di regole del modello relazionale che garantiscono l'integrità dei dati quando si hanno relazioni associate tra loro attraverso la chiave esterna: queste regole servono per rendere valide le associazioni tra le tabelle e per eliminare gli errori di inserimento, cancellazione o modifica di dati collegati tra loro
- **Integrità Fisica:** consiste nel proteggere il database dai possibili danneggiamenti all'hardware che possono essere causati da un disastro naturale o da un attacco hacker che pregiudicano le funzionalità del database stesso.

Come definito in precedenza, le basi di dati sono multiutente ed è perciò consentito l'accesso allo stesso dato da più utenti contemporaneamente ecco perché il ruolo del DBA diventa fondamentale poiché gestisce i privilegi e permessi ovvero la parte di "Vista" del Database.

Il DBA però, non svolge solamente i compiti principali enunciati sopra ma svolge anche altre mansioni importanti per il mantenimento del DBMS stesso come: la configurazione e installazione di software necessari per quell'indicato database, la verifica delle prestazioni o il controllo della sicurezza per non compromettere i dati che il database possa contenere, permette e gestisce i piani di backup del server, esegue un'analisi delle performance e manutenzione del database stesso però questi punti li andremo ad analizzare successivamente.

Come spiegato in prima istanza il DBA, va a definire la parte di *vista* del DBMS, ma andiamo ad



analizzarla più nel dettaglio; questo ruolo è molto importante in un'azienda perché, gli utenti potrebbero erroneamente danneggiare e compromettere l'intero database che a quel punto sarebbe inutilizzabile ed ecco perché vi è una differenziazione dei ruoli che vengono assegnati. Grazie ai ruoli differenziati, il DBA può gestire i vari accessi che vengono effettuati all'interno del database, quindi che la visualizzazione dei dati sia possibile solo ad utenti specifici e riducendo di conseguenza altri rischi che si potrebbero presentare, come la violazione anche della privacy e sensibilità dei dati stessi. Di certo un database per offrire le prestazioni migliori deve anche rimanere prestante e stabile nel tempo, andando ad alleggerire la curva di guasti che si potrebbero presentare e quindi di spesa che l'azienda andrebbe ad investire, per questo un altro compito del DBA è quello di garantire, grazie a dei controlli e test periodici, il corretto funzionamento del database e in caso di non riuscita per problemi terzi come un problema hardware si cerca di recuperare tutti i dati possibili dalla macchina e/o eventualmente in un esempio comune, come un blackout la maggior parte dei server hanno dei sistemi che ne impediscono la rottura per esempio un secondo alimentatore che in caso di rottura del principale, il sistema fa uno "swap" automatico al secondo, stessa cosa succede per la memoria principale che in queste macchine è definita ECC ovvero che corregge i possibili errori che vi si possano presentare; Per quanto concerne la memoria secondaria lì può dipendere dalle ore di lavoro che il dispositivo ha, da un possibile RAID di configurazione che vi è stato inizializzato e che quindi possa compromettere l'integrità dei dati e file conservati.

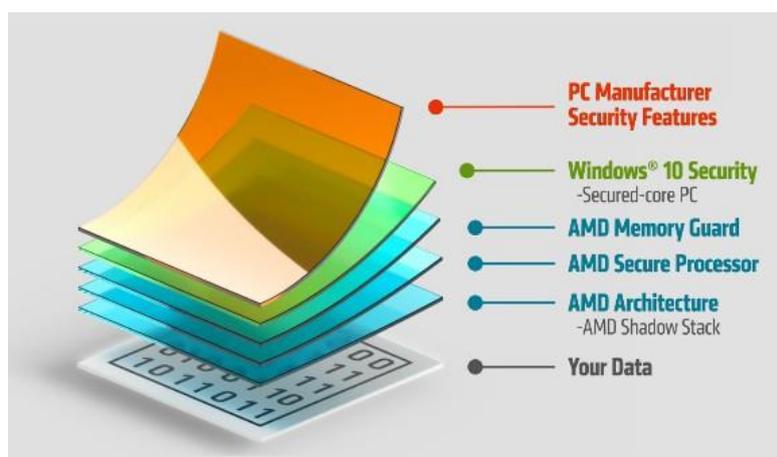
Analizzando e portando ad esempio una realtà di medie ma anche grandi aziende, il cosiddetto server fisico è diventato sempre più obsoleto da mantenere nella sede stessa; ecco perché si prendono in esame diverse opzioni ora sempre più presenti, i cosiddetti server cloud come AWS di Amazon o Aruba che sono i più conosciuti e importanti nel settore. In questi casi il ruolo fisico del DBA va a concentrarsi sulla procedura in rete, tutto tramite un'interfaccia apposita, dove l'Amministratore può connettersi ed eseguire le eventuali diagnostiche o procedure da fare.



Tornando ad analizzare alcuni punti trattati sopra, il Backup del database è una procedura molto importante da eseguire su un server in un'azienda ma in generale in un dispositivo dove vi sono allocati dei dati importanti, per questo viene eseguito solitamente ogni giorno ad una specifica

ora e solitamente questa procedura viene automatizzata tramite linea di comando o uno script, ma vi può essere sempre la possibilità che il backup venga danneggiato di proposito da software terzi o malware nonostante il livello di sicurezza che in un server possa essere elevato ed è per questo che si consiglia il backup su un supporto terziario non connesso alla rete e su macchine o calcolatori privi di qualsiasi software che possa tracciare dei dati che vengano inseriti all'interno del calcolatore stesso.

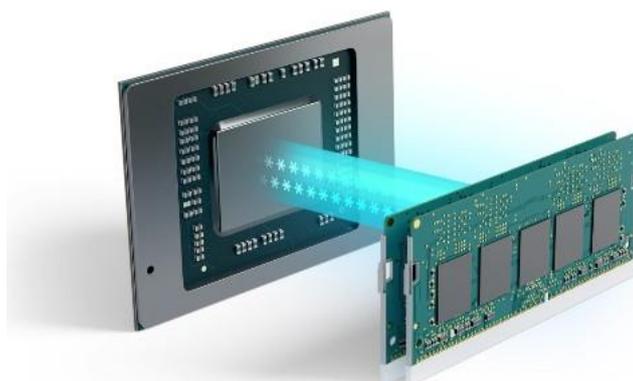
Ecco perché il ruolo del DBA gioca un ruolo chiave quando si tratta il tema della sicurezza all'interno delle basi di dati.



Per quanto concerne la **protezione** all'interno di una base di dati si possono implementare diversi schemi. Il più comune è quello di utilizzare hardware appositamente progettato per database, dalle CPU che implementano delle funzioni tramite l'architettura delle stesse,

a poter proteggere anche la memoria del calcolatore tra cui quella principale e quella secondaria che si pone come contenitore dei dati stessi.

Volendo applicare tutte le norme di protezione, a tutto l'hardware ma specialmente ai contenitori dei dati, in questo caso il disco rigido, si può evincere però che per mantenere un alto livello di protezione non sia sufficiente e che quindi si debba ricorrere alla protezione di ogni singolo file e/o documento presente all'interno anche perché è di norma condividere file o messaggi importanti per una data azienda in maniera non sicura, per esempio attraverso un supporto USB non cifrato oppure via mail, ed ecco perché la cifratura e crittografia diventano quasi indispensabili.





Resta connesso e informato sui prossimi eventi, corsi e seminari:

## **Web**

[www.profmicheletarantino.com](http://www.profmicheletarantino.com)

## **Email**

[profmicheletarantino@gmail.com](mailto:profmicheletarantino@gmail.com)

## **Telefono**

349 83 54 521

## **Facebook**

[@micheletarantinodocente](https://www.facebook.com/micheletarantinodocente)

## **Instagram**

[@profmicheletarantino](https://www.instagram.com/profmicheletarantino)

Hai bisogno di un modulo personalizzato? Non esitare a contattarmi!