



Fondamenti di Informatica

- Accademia di Belle Arti di Verona
- Università degli Studi di Verona
- A.A. 2020-2021

- Docente - Vincenzo Giannotti

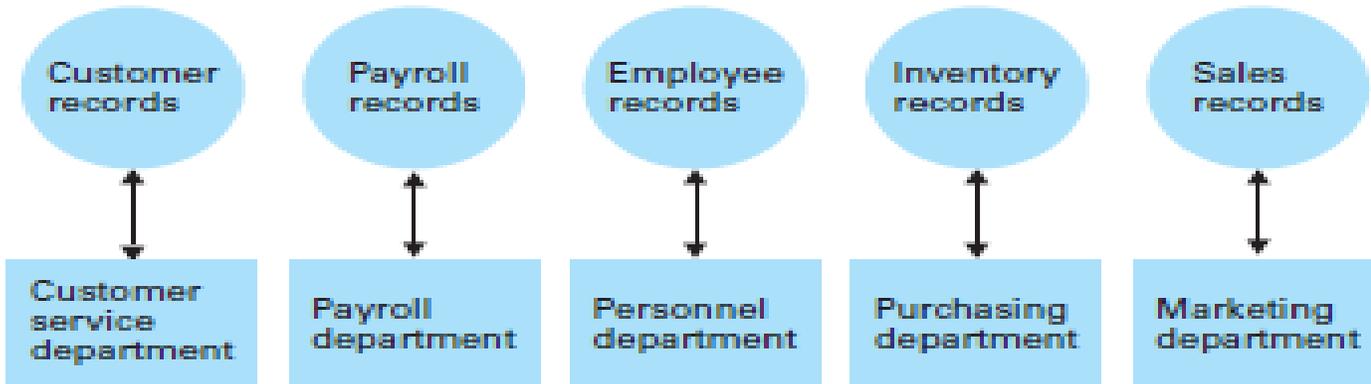
CAPITOLO 6 – BASI DI DATI

Basi di dati

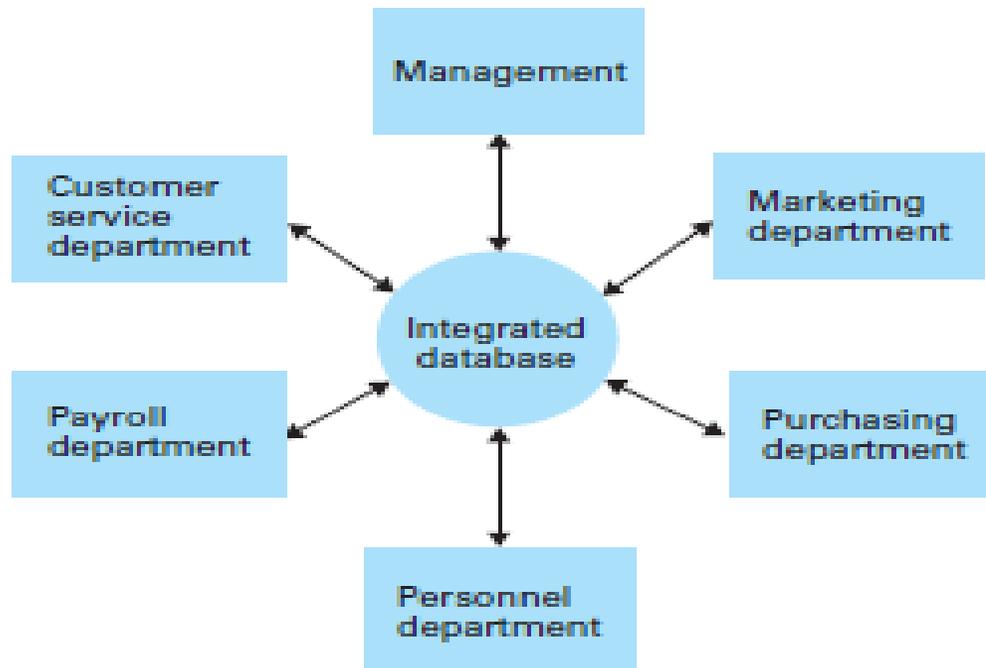
Il termine «Base di Dati» o più comunemente «**Database**» in inglese, si riferisce a una collezione di dati organizzati in maniera che la loro ricerca e il loro accesso possa avvenire in maniera efficiente secondo diverse modalità e utilizzando delle applicazioni dedicate.

Diversamente dal «**file**» che è un sistema di archiviazione delle informazioni di tipo mono-dimensionale, ossia in grado di rappresentarli secondo un unico punto di vista, il data base è un sistema multi-dimensionale che consente di rappresentare i dati in diversi modi e secondo diversi punti di vista.

a. File-oriented information system



b. Database-oriented information system

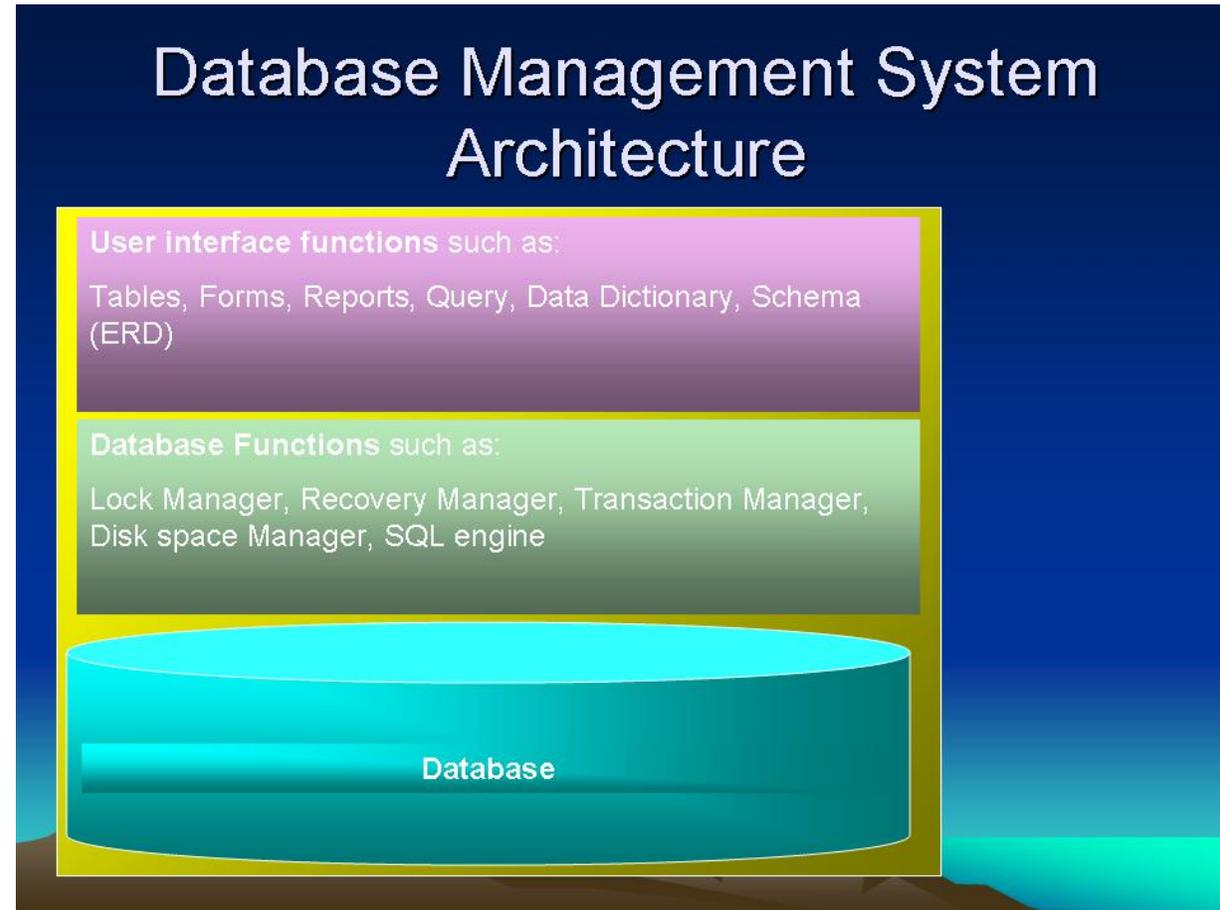


Basi di Dati

- Nella gestione dei dati per File (che spesso avviene nelle piccole organizzazioni) le analisi avvengono attraverso la comunicazione delle persone.
- La gestione con Data Base consente di automatizzare molte attività di analisi dei dati, mettendo in relazione, per esempio, i risultati dei diversi dipartimenti aziendali e consentendo una condivisione profonda delle informazioni.

Il Data Base Management System (DBMS)

- Come detto, un database è una collezione di dati organizzati in maniera che la loro ricerca e il loro accesso possa avvenire in maniera efficiente secondo diverse modalità.
- Lo strumento che consente di effettuare queste operazioni è il «**DataBase Management System**» (DBMS), un sistema software progettato per consentire la creazione, la manipolazione e l'interrogazione di una collezione di dati strutturati. Spesso col termine «database» ci si riferisce sia al DBMS che alla collezione di dati da esso gestita.

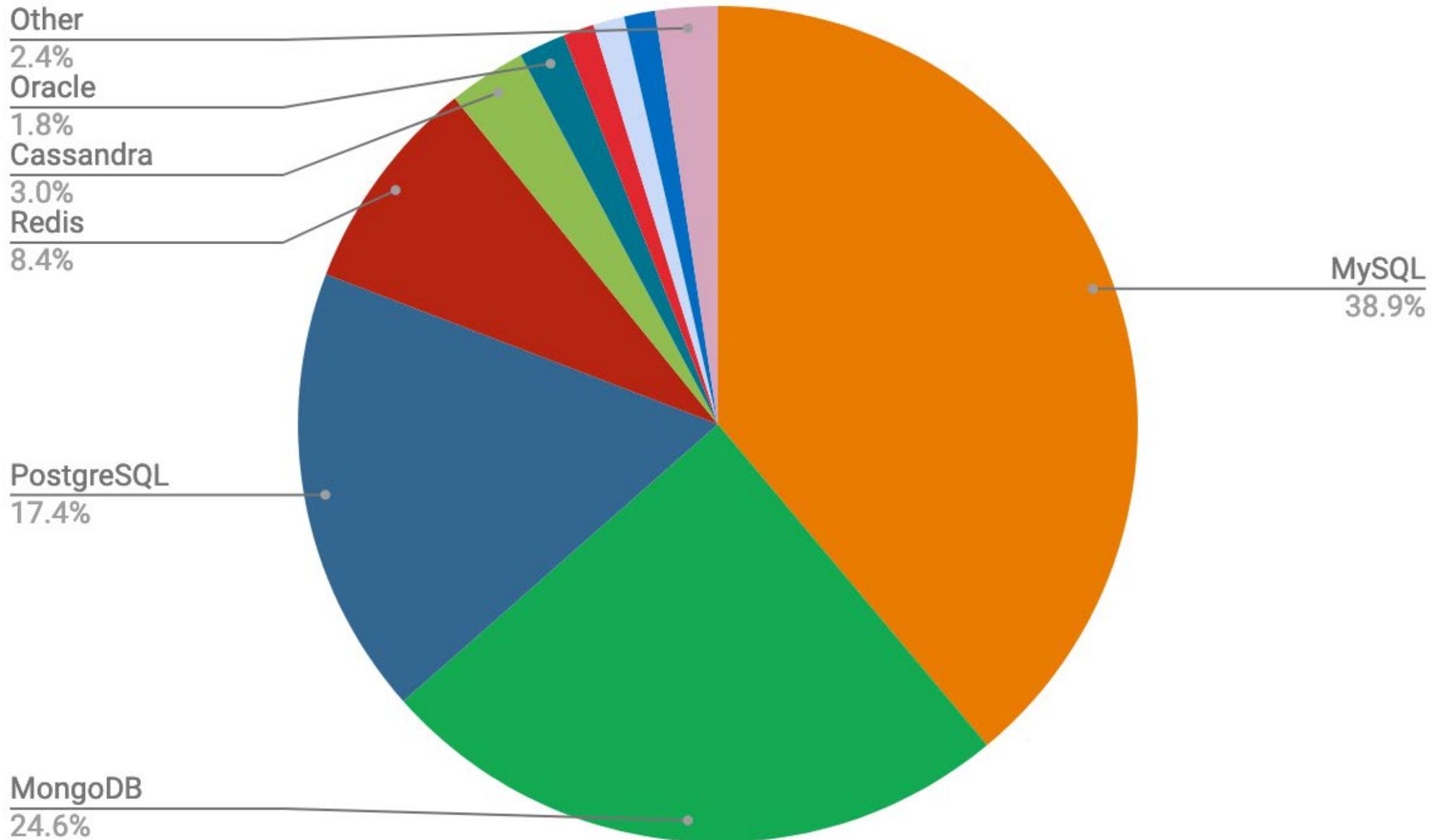


Il Data Base Management System (DBMS)

- Il DBMS è dunque una applicazione software che normalmente possiede le seguenti caratteristiche di gestione dell'informazione:
 - È in grado di gestire collezioni organizzate di dati di grandi dimensioni. Con l'affermazione «**di grandi dimensioni**» intendiamo collezioni di dimensioni molto maggiori di quelle che sono normalmente gestibili nella memoria centrale del computer
 - È in grado di gestire informazioni «**persistenti**», le quali hanno un ciclo di vita del tutto indipendente dalle applicazioni che le utilizzano (che possono essere molte, anche contemporaneamente: pensiamo ai DB di Google)
 - È capace di trattare informazioni «**condivise**» che possono quindi essere utilizzate e manipolate da applicazioni diverse.

Il Data Base Management System (DBMS)

- Inoltre il DBMS deve possedere le seguenti ulteriori caratteristiche di garanzia, strettamente connesse alla sicurezza e alle performance:
 - È «**affidabile**». Ciò significa che è in grado di resistere ad eventuali malfunzionamenti del sistema sia dal punto di vista Hardware che Software.
 - Garantisce la «**privacy**» dell'informazione, quando richiesta, con adeguati strumenti e procedure di controllo degli accessi e verifica delle identità.
 - È «**efficiente**», ossia è in grado di ottimizzare l'utilizzazione delle risorse (in particolare di memorie e di tempi di CPU) che gli vengono messe a disposizione dal sistema operativo.
 - È «**efficace**» e questo significa che riesce effettivamente ad aumentare la produttività di chi lo utilizza.



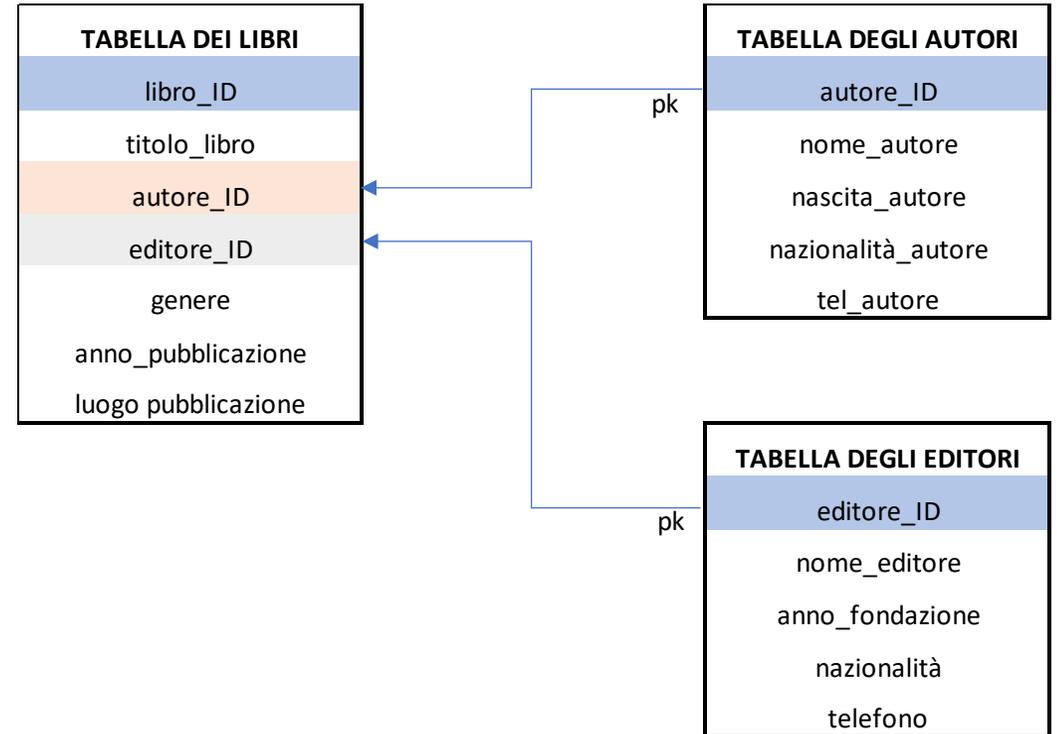
Il Database Relazionale (SQL)

Nel **database relazionale** tutto ruota attorno al concetto di **tabella**. Ne viene creata una per ogni tipo di informazione da trattare; per esempio, se vogliamo realizzare un archivio di libri: la tabella dei libri, quella degli autori, quella dei generi etc...

A sua volta ogni tabella è costituita da **colonne** che descrivono quel tipo di informazione: la tabella dei libri per esempio conterrà le colonne autore, genere, data di pubblicazione, editore etc...; a sua volta quella degli autori conterrà nome, cognome, data di nascita e di morte, luogo di nascita e così via.

Il Database Relazionale (SQL)

Nel database relazionale le tabelle possono essere messe in **relazione**. Ciò significa che una riga della tabella 1 può far riferimento ad una riga della tabella 2. Per esempio, si associa ad un dato libro della tabella 1 un determinato autore della tabella 2 col vantaggio che quel determinato autore sarà scritto una volta sola nel database ma potrà essere richiamato n volte in associazione a diversi libri, cioè tutti quelli che lui ha scritto.



Il Database Relazionale (SQL)

Il linguaggio utilizzato per impartire comandi ad un DBMS relazionale è il **SQL (Structured Query Language)**. Il database relazionale è molto efficiente ma necessita di una impegnativa attività di disegno della struttura interna costituita di tabelle e di relazioni.

```
select titolo_libro  
from tabella dei libri  
where anno_publicazione < «anno»  
order by «criterio»
```

Il Database Non Relazionale (NoSQL)

Nei **database NoSQL** manca la strutturazione rigida dei contenuti, tipica dei database relazionali, il che comporta alcuni vantaggi e alcuni svantaggi.

Le informazioni non sono più organizzate in tabelle composte di righe e colonne, ma in oggetti diversi che non sono necessariamente strutturati; per esempio in documenti archiviati in collezioni. Il Database MongoDB, un Open Source tra i più utilizzati, è un database NoSQL basato sui documenti.

Evitiamo di addentrarci nei meccanismi di collegamento delle informazioni tipici di questo tipo di database; piuttosto ci concentriamo sul fatto quella NoSQL è una soluzione ideale quando abbiamo a che fare con dati scorrelati e che tendono ad evolvere nel tempo (come spesso accade nel Web). L'assenza di una struttura fissa permette di aggiungere nuovi tipi di dati (o modificare i dati esistenti) senza dover riprogettare il database.

Database distribuiti

Un **database distribuito** è un database che si trova sotto il controllo di un DBMS nel quale gli archivi di dati sono memorizzati su diversi computer anche fisicamente molto distanti tra loro ma **interconnessi in rete**.

La funzione del DBMS serve a garantire, oltre alle usuali funzioni:

- Che la distribuzione dei dati avvenga in maniera «**trasparente**» agli utenti, che debbono poter interagire col sistema come se si trattasse di un tutt'uno anche dal punto di vista prestazionale; non è necessario conoscere nulla sulla dislocazione dei dati ma semplicemente accedervi come se il database fosse centralizzato.
- Che sia garantito l'accesso «**concorrente**» ai dati: è necessario che il DBMS gestisca la possibilità che più utenti dislocati non si sa dove, possano accedere contemporaneamente ai dati, mantenendo l'integrità di questi ultimi.

Vantaggi dei database distribuiti

Maggiore **affidabilità** e disponibilità e più facile espandibilità (scalabilità)

Può riflettere la **struttura organizzativa** con porzioni di database memorizzati all'interno dei servizi cui si riferiscono

Maggiore protezione dei **dati sensibili** che non risultano accentrati in un unico luogo

Miglioramento delle **prestazioni** accentrando i dati nei pressi del sito di maggiore domanda

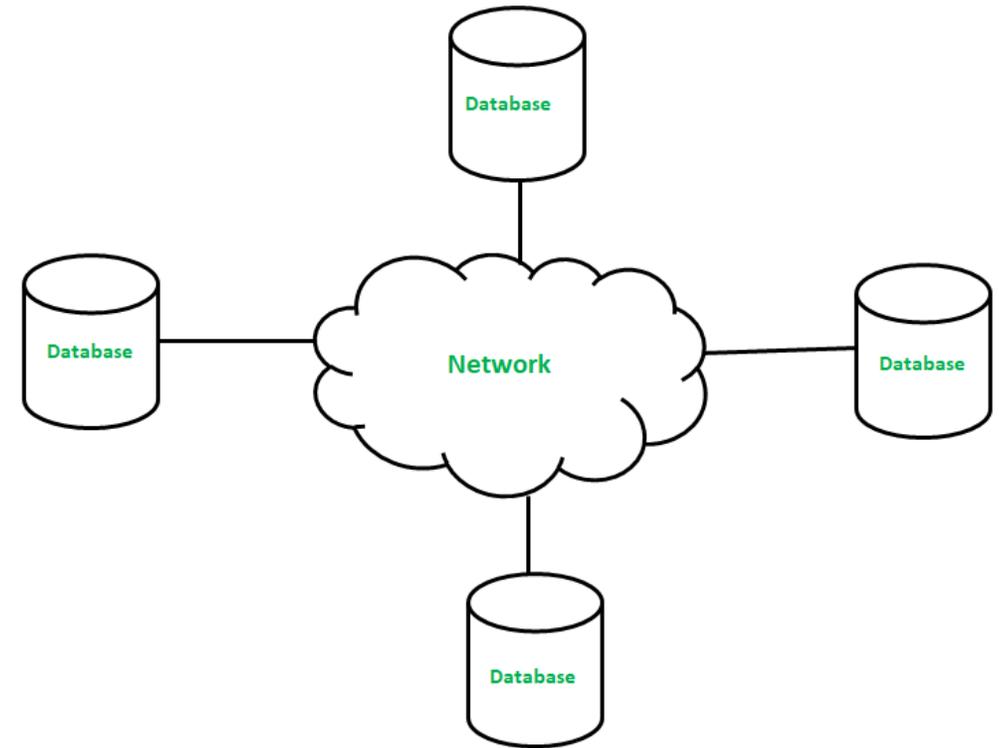
Può essere più **economico**, per esempio creando una rete di piccoli DB con la potenza di uno più grande

Consente maggiore **modularità**: i sistemi possono essere modificati, aggiunti e rimossi dal database distribuito senza influenzare altri moduli.

Vantaggi dei database distribuiti

Consente transazioni più affidabili e un **funzionamento continuo**, anche se alcuni nodi vanno offline

Maggiore **solidità** complessiva: il fallimento di un singolo sito non influisce sulle prestazioni complessive del sistema.



Svantaggi dei database distribuiti

- Maggiore **complessità** del sistema che si riflette sul lavoro degli Amministratori che debbono garantire la massima trasparenza della natura distribuita del sistema e mantenere più sistemi eterogenei.
- Serve un maggior lavoro di **progettazione** del database
- Minore **economicità** se riferita ai costi del lavoro
- Maggiore complessità delle **procedure di sicurezza**; può essere necessario, per esempio, crittografare i collegamenti di rete tra i diversi siti remoti
- È più difficile mantenere l'**integrità** dei dati, per esempio da modifiche accidentali o dolose e garantire l'accesso concorrente
- Minori **skills** disponibili; i database distribuiti sono più difficili da lavorare e non si trova facilmente personale

Prossimo Capitolo – Sicurezza informatica

Nel prossimo capitolo vedremo i concetti basilari della sicurezza informatica e come possiamo anche noi, proteggere il nostro computer di casa da virus, malware e attacchi informatici.