

Introduzione ai database


Database – una definizione

Un database può essere definito come una **collezione organizzata di dati correlati** che modellano alcuni aspetti del mondo reale.

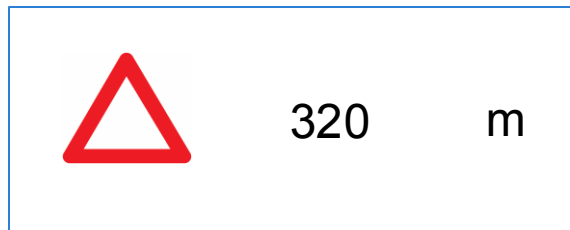
I database sono componenti fondamentali di moltissime applicazioni informatiche.

Database – perché collezione organizzata di dati?

La memorizzazione dei soli dati spesso non è sufficiente, in quanto i dati sono valori singoli, una entità atomiche:

- «Mario», «123456», «00010101100101»  sono dati

La correlazione dei dati permette invece di ottenere informazioni di maggiore interesse.



Dati



Informazione

Database - Il valore della correlazione

- Le informazioni sono delle risorse. Sono allo stesso livello delle materie prime, del capitale, delle persone...
- Non hanno tutte lo stesso valore:
 - Molti dati grezzi hanno meno valore di pochi dati ben correlati.
 - Il valore aumenta cioè all'aumentare della correlazione.



Database - Esempio

Creare un database per memorizzare tutti i risultati di tutte le edizioni dei giochi olimpici.

Dati:

- Atleti
- Discipline olimpiche
- Edizioni

Correlazioni:

- Quale atleta ha partecipato a quali discipline e in quali giochi?

Database – File di testo?

L'utilizzo di uno o più file di testo (per esempio csv) può essere considerato un database?

```
Atleti.csv (nome, sesso, nazione, anno, sport)
Federica Pellegrini, F, ITA, 2004, Swimming
Gianmarco Tamberi, M, ITA, 2020, Athletics
```

```
import csv

with open('atleti.csv', 'r') as file:
    csv_reader = csv.reader(file)

    for row in csv_reader:
        if row[4] == "Swimming":
            print(row[0])
```

Database – Limitazioni dei file di testo

L'utilizzo di uno o più file di testo comporta numerose limitazioni:

- Integrità dei dati:
 - Come assicuriamo che i dati siano sempre corretti?
 - Cosa succede se viene scritto un anno non olimpico?
 - Cosa succede se nel campo dell'anno olimpico inserisco una stringa?
- Disponibilità:
 - Cosa succede se la macchina va in crash durante un aggiornamento?
 - Come gestisco la replicazione su più macchine?
- Riservatezza:
 - Come controllo chi può accedere ai dati?
- Implementazione:
 - Come effettuo la ricerca? E se il file è di 10TB? Se è su server remoto?

Database Management System

Un Database Management System (DBMS) è un software che gestisce la creazione dei database e il successivo accesso. Si occupa della memorizzazione dei dati, delle ricerche, delle modifiche, delle correlazioni e di tutti gli aspetti amministrativi.

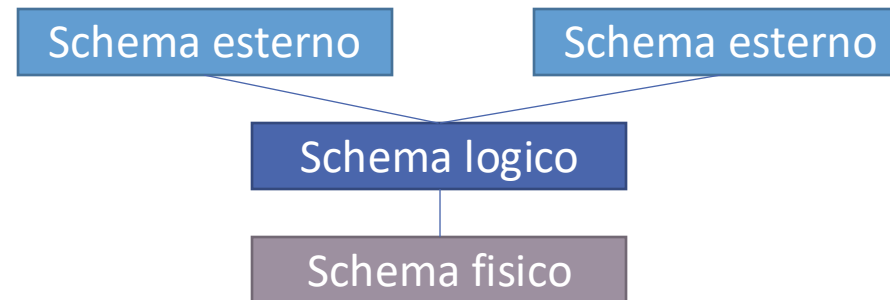
Caratteristiche principali di un DBMS:

- Gestisce grandi quantità di dati con particolare attenzione all'efficienza.
- Gestisce la persistenza dei dati garantendo la fault tolerance.
- Gestisce la condivisione dei dati garantendo il controllo degli accessi e della concorrenza.
- Definisce uno o più linguaggi di programmazione per interagire con i dati
- Divide lo schema fisico dallo schema logico

Schema fisico vs schema logico

Nei primi DBMS la modalità di memorizzazione dei dati (schema fisico) influiva fortemente sull'utilizzo degli stessi, portando quindi ad applicazioni complesse da realizzare e mantenere.

I DBMS moderni permettono di definire uno schema logico, indipendente da quello fisico, creato per rendere più semplice la realizzazione delle applicazioni.



Tipologie di DBMS

Nel corso degli anni sono emersi numerosi modelli per la definizione di database, ovvero per la definizione degli schemi, dei dati e delle operazioni:

- Modello Relazionale
- Key / Value
- Graph
- Document based
- Column Oriented

Il modello relazionale

Il modello relazionale

- E' un modello per la definizione dello schema logico di un database.
- Nasce nel 1970 con lo scopo di aumentare l'indipendenza dello schema logico da quello fisico.
- Si pone l'obiettivo di essere sia efficace che semplice da comprendere e utilizzare
- Si pone l'obiettivo di creare una teoria formale per la progettazione dei database (teoria relazionale)
- Nei modelli pre-esistenti (principalmente gerarchico e reticolare) c'era un forte legame con lo schema fisico e la comprensione del modello non era semplice.

La relazione

Il modello relazionale utilizza il termine relazione nella sua eccezione matematica:

- Si considerino n insiemi D_1, D_2, \dots, D_n , non necessariamente distinti
- Si consideri il prodotto cartesiano $D_1 \times D_2 \times \dots \times D_n$, ovvero l'insieme di tutte le n -ple ordinate (d_1, d_2, \dots, d_n) tali che $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$
- Una relazione su D_1, D_2, \dots, D_n è un qualunque sottoinsieme del prodotto cartesiano $D_1 \times D_2 \times \dots \times D_n$

Esempio:

- $D_1 = \{a, b, c\}, D_2 = \{1, 2\}$
- $D_1 \times D_2 = \{ (a, 1), (a, 2), (b, 1), (b, 2), (c, 1), (c, 2) \}$
- $r = \{ (a, 1), (c, 2) \}$ è una relazione in quanto $r \subseteq D_1 \times D_2$

La relazione

Alcuni termini:

- $D_1 \times D_2 \times \dots \times D_n$ sono i domini della relazione
- Il valore di n è il grado della relazione
- Il numero di n -ple viene chiamato cardinalità

Ricapitolando:

- Una relazione è un insieme di n -ple distinte tra di loro, senza un ordinamento: $\{ (a,1),(c,2) \} = \{ (c,2),(a,1) \}$
- L'ordine con cui si considerano i domini è rilevante: $D_1 \times D_2 \neq D_2 \times D_1$
- I domini non devono essere necessariamente distinti: $D_1 \times D_1$

La relazione

Per comodità le relazioni sono rappresentate in forma tabellare: relazione \simeq tabella

| | | |
|-----------------|---|---|
| { (a,1),(c,2) } | a | 1 |
| | c | 2 |

| | | | |
|---------------------|---|---|---|
| { (a,1,b),(c,2,c) } | a | 1 | b |
| | c | 2 | c |

Ad ogni occorrenza di dominio si associa un nome univoco, detto attributo. Questo permette di aumentare la leggibilità, di esplicitare il collegamento con i concetti rappresentati e di rendere irrilevante l'ordine con cui si considerano i domini.

Ad ogni relazione è inoltre associato un nome univoco.

| | | |
|-----------------|------------|-------|
| { (a,1),(c,2) } | Relazione1 | |
| | attr1 | attr2 |
| | a | 1 |
| | c | 2 |

| | | | |
|---------------------|------------|-------|-------|
| { (a,1,b),(c,2,c) } | Relazione2 | | |
| | attr1 | attr2 | attr3 |
| | a | 1 | b |
| | c | 2 | c |

La relazione - esempio

Schema

Nome della relazione → **Anagrafica**

Attributi →

| Codice | Cognome | Nome | DataNascita | Email |
|--------|---------|----------|-------------|--------------------------|
| 1 | Rossi | Luigi | 01/07/1980 | luigi.rossi@gmail.com |
| 2 | Bianchi | Marco | 23/04/1977 | marco.bianchi@gmail.com |
| 3 | Verdi | Giuseppe | 01/09/2000 | giuseppe.verdi@gmail.com |

Tupla o record →

Istanza

Lo schema di una relazione può anche essere rappresentato con la seguente notazione:

Anagrafica(Codice, Cognome, Nome, DataNascita, Email), sottolineando gli attributi che partecipano alla chiave primaria.

Vincoli di integrità

- Una relazione non contiene dati arbitrari.
- E' necessario inserire una serie di vincoli da rispettare per poter considerare validi i dati.
- In assenza di vincoli non può essere garantita la piena operatività sulle informazioni.
- Un vincolo di integrità è una proprietà che deve essere soddisfatta dalle istanze.
- Possiamo avere più tipologie di vincoli:
 - Vincoli di dominio
 - Vincoli di tupla
 - Vincoli di chiave
 - Vincoli di integrità relazionale
- Tutti i vincoli sono definiti a livello di schema, e quindi validi per tutte le tuple.

Vincoli di dominio

- E' un vincolo che limita i valori ammissibili per un singolo attributo.
- E' un vincolo sempre presente in quanto imposto dal DBMS utilizzato. Ogni DBMS infatti gestisce solamente alcune tipologie di dati (interi, stringhe, date...).
- Possono essere previsti vincoli più stringenti rispetto ai tipi di dato del DBMS: per esempio solo numeri positivi, oppure numeri compresi in un certo range, oppure stringhe di n caratteri...

Vincoli di tupla

- Sono vincoli che interessano più attributi della tupla. Per esempio: se $\text{Voto} < 30 \Rightarrow \text{Lode} = \text{"NO"}$
- I vincoli di dominio sono dei casi particolari di vincoli di tupla.

Vincoli di chiave

- Sono vincoli che impediscono l'esistenza di più tuple con gli stessi valori sugli attributi identificati come chiave. Per esempio: matricola studente, codice articolo, codice gara sportiva.
- La chiave deve essere composta dal numero minimo di attributi necessari (altrimenti si parla di superchiave).
- Data una relazione possono esistere più chiavi, ugualmente valide.
- E' sempre possibile identificare almeno una chiave, quella composta da tutti gli attributi.
- **ATTENZIONE:** Il fatto che un attributo possa essere chiave dipende dal dominio (per esempio il codice fiscale).
- Le chiavi ricoprono un ruolo fondamentale in quanto permettono di mettere in correlazione i dati di relazioni differenti

Valori NULL

- Spesso dobbiamo gestire dati incompleti, dei quali non conosciamo il valore. Per esempio il codice fiscale per un cittadino estero non residente in Italia.
- I DBMS relazionali gestiscono questo caso introducendo il valore NULL (nullo) utilizzabile in tutti i domini.
- L'applicabilità del valore NULL su ogni attributo va specificata nello schema della relazione
- Un valore NULL può avere più significati:
 - Valore non applicabile
 - Valore applicabile ma ignoto
 - Applicabilità ignota
- NULL non è mai un valore di dominio e pertanto $\text{NULL} \neq \text{NULL}$
- La chiave che non accetta valori NULL è detta chiave primaria

Vincoli di integrità referenziale

- Sono vincoli che coinvolgono più relazioni e permettono di metterle in correlazione.
- Impongono che l'insieme dei valori validi per un attributo sia un sottoinsieme della chiave primaria di un'altra relazione (detta relazione secondaria)
- La colonna che referencia la chiave della relazione secondaria è detta chiave esterna o foreign key

Prodotti

| <u>Codice</u> | CodCateg | Descrizione |
|---------------|----------|-------------------------|
| 1 | 1 | Succo mela 125ml |
| 2 | 1 | Succo pesca 125ml |
| 3 | 2 | Confettura fragole 200g |

Categorie

| <u>Codice</u> | Descrizione |
|---------------|------------------|
| 1 | Succhi di frutta |
| 2 | Confetture |

- CodCateg è chiave esterna di Prodotti

Introduzione al DBMS Relazionali

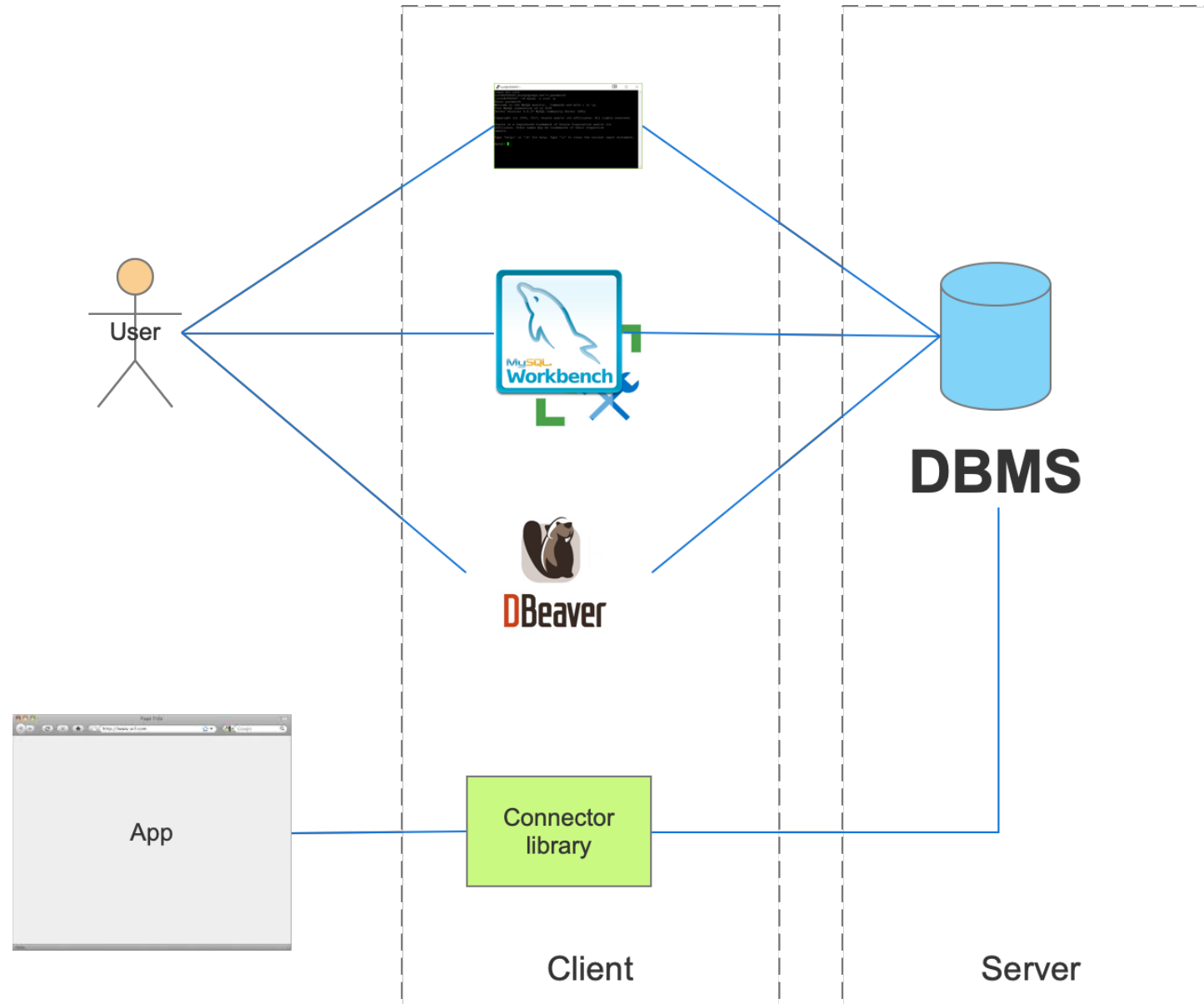
RDBMS - Architettura

La maggior parte dei DBMS relazionali adotta una architettura Client-Server. Il RDBMS gira come servizio su un server ed espone le proprie funzionalità ad uno o più client, utilizzati da uno o più utenti (anche applicazioni).

I client possono essere di vario tipo:

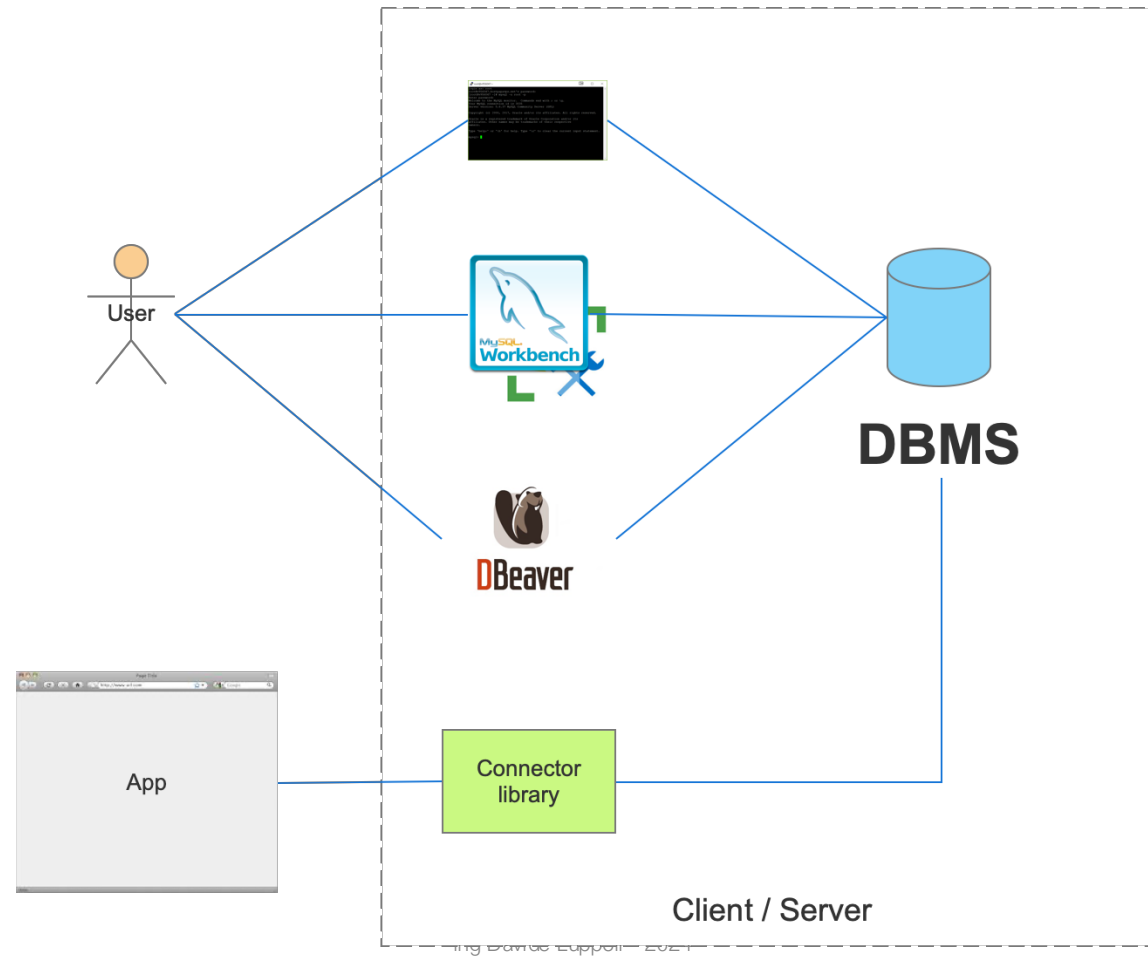
- A riga di comando (CLI – Command Line Interface)
- Con interfaccia grafica
 - Ufficiali
 - Di terze parti
- Librerie software

RDBMS - Architettura



RDBMS - Architettura

Client e server possono essere installati entrambi nella stessa macchina.



Alcuni RDBMS

Esistono numerosi DBMS relazionali, diversi tra di loro, ma con le stesse caratteristiche di base:

- Architettura Client Server
- Multisessione
- Multiutente
- Disponibilità di client da riga di comando, grafici e librerie
- Presenza del linguaggio SQL

- MySql, è un RDBMS di proprietà di Oracle (fino al 2009 la proprietà era di Sun Microsystems).
- E' un software open source liberamente scaricabile ed installabile.
- Ne esiste una versione closed-source a pagamento per il mondo enterprise, con migliorie specifiche per la gestione di database di grandi dimensioni.
- Possiede un client da riga di comando
- Possiede un client grafico chiamato MySql Workbench (da installare a parte)
- Possiede librerie per l'utilizzo con i principali linguaggi di programmazione (elenco completo su <https://www.mysql.com/it/products/connector/>)
- La connessione al server avviene specificando indirizzo ip, porta (default 3306), nome utente e password

MariaDB



MariaDB è un RDBMS nato nel 2009 come fork di MySQL per garantire che il progetto MySQL rimanesse open source nonostante l'acquisizione da parte di Oracle.

MariaDB oggi è una versione migliorata di MySQL, che offre nuove funzionalità e numerosi miglioramenti in termini di sicurezza e velocità di esecuzione, pur rimanendo completamente retrocompatibile con MySQL.

Essendo un sostituto di MySQL, completamente compatibile, l'adozione è molto semplice e richiede solamente il porting dei dati, senza nessuna modifica a query e applicazioni.

Molte applicazioni stanno quindi migrando a MariaDB per sfruttarne i miglioramenti

Microsoft Sql Server



- Microsoft SQL Server, o più semplicemente SQL Server, è un RDBMS di proprietà Microsoft.
- E' un software closed source, disponibile in varie versioni una delle quali utilizzabile gratuitamente (SQL Server Express Edition).
- SQL Server nasce come applicazione windows ma, nelle ultime versioni, può anche essere installato in alcune versioni di Linux.
- Possiede un client da riga di comando
- Possiede un client grafico chiamato SSMS - Sql Server Management Studio (da installare a parte)
- Possiede librerie per l'utilizzo con i principali linguaggi di programmazione (elenco completo su <https://docs.microsoft.com/it-it/sql/connect/sql-connection-libraries>)
- La connessione al server avviene specificando indirizzo ip, porta (default 1433), nome utente e password

Microsoft Sql Server

- La connessione al server avviene specificando:
 - Identificativo del server (per esempio indirizzo ip o nome della macchina Windows)
 - Identificativo dell'istanza di SQL Server (un server potrebbe contenere più installazioni di SQL Server)
 - Porta di ascolto del server (default 1433)
 - Credenziali di accesso
 - Autenticazione di windows
 - Autenticazione di sql server

Terminologia adottata da MySQL / MariaDB / SqlServer

- Relazione o Entità → Tabella
- Associazione → Relazione
- Attributo → Campo
- Tupla → Record o riga
- Schema → Schema
- Vincolo di record / di tupla → Vincoli