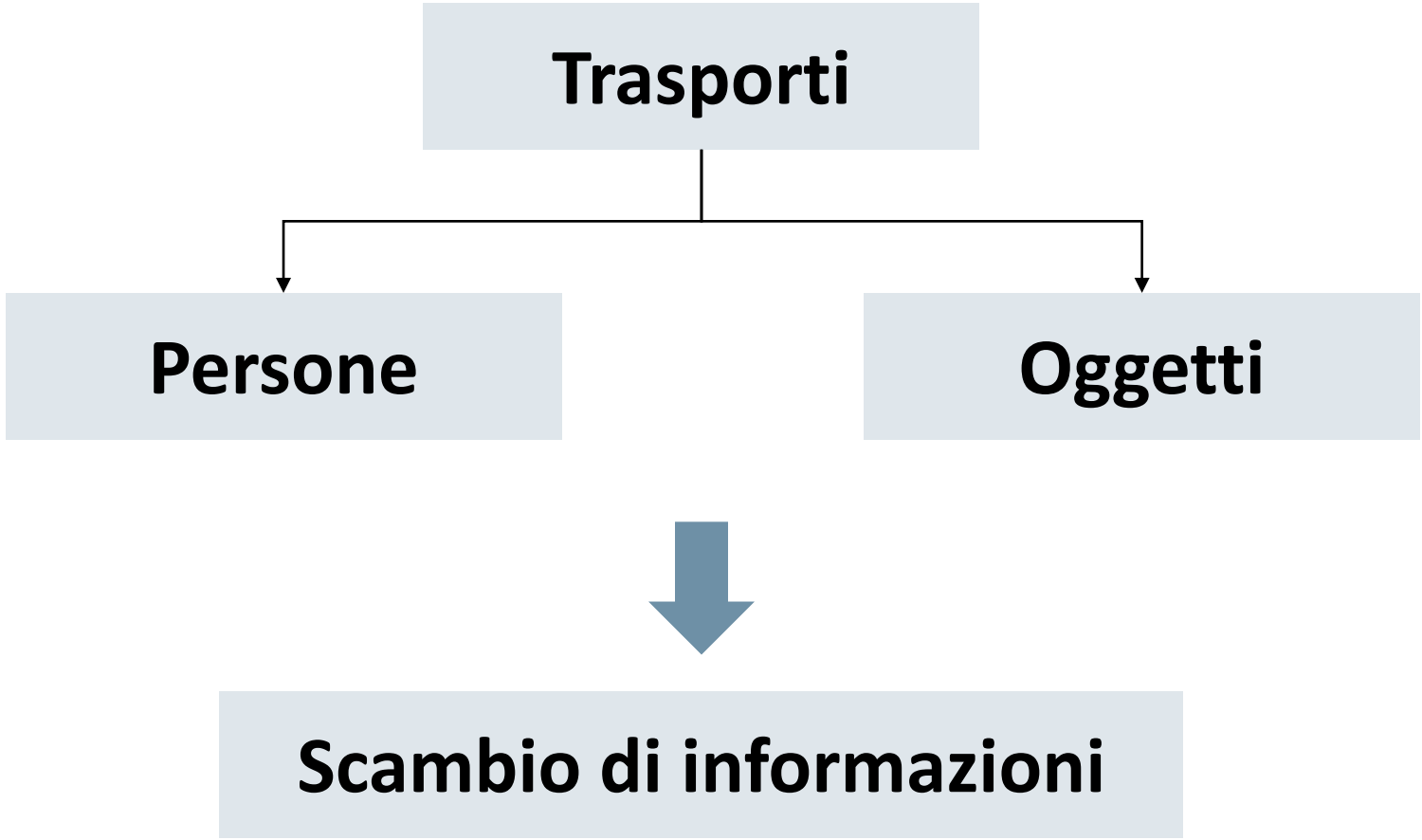
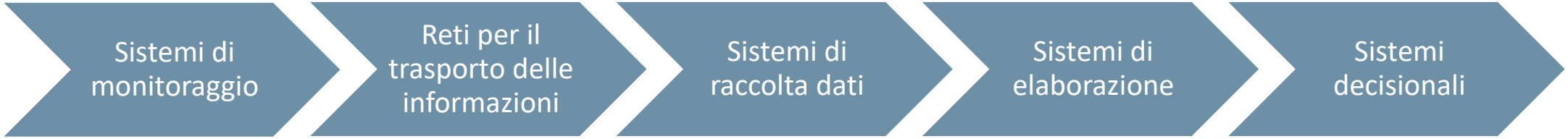
Sistemi informativi e basi di dati

Sistemi informatici per i Trasporti



Sistemi informatici per i Trasporti

Sistemi informatici per lo scambio di informazioni



Sistemi informatici per i Trasporti

Es. logistica: una azienda di trasporto vuole conoscere in tempo reale lo stato di avanzamento delle spedizioni per ottimizzare la propria attività riducendo i costi.

Ha bisogno di:

- Sistemi automatici di rilevamento della merce.
- Conoscere lo spostamento dei mezzi.
- Sistemi per la memorizzazione delle informazioni e monitoraggio real-time dei processi.
- Sistemi di elaborazione dei dati raccolti per ottimizzazione percorsi e carichi.

Sistemi informatici per i Trasporti

Es. guida autonoma: una macchina in grado di guidare autonomamente in qualsiasi contesto capace di sostituire completamente l'attività dell'uomo.

Ha bisogno di:

- Sistemi di posizionamento, sistemi automatici di emergenza e geolocalizzazione.
- Connettività.
- Sistema di scambio di informazioni tra autovetture.
- Sistemi di raccolta dati per ottimizzazione percorsi e ricarica elettrica.

Definizioni - Sistema

Nell'ambito scientifico, qualsiasi oggetto di studio che, pur essendo costituito da diversi elementi reciprocamente interconnessi e interagenti tra loro o con l'ambiente esterno, reagisce o evolve come un tutto, con proprie leggi generali.

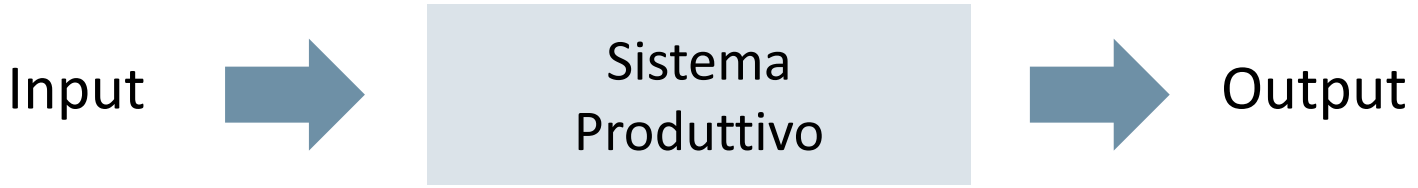
Da https://www.treccani.it/vocabolario/sistema/

Un organismo, un ente, un dispositivo,… eventualmente composto da più parti, che svolge una funzione (per perseguire degli obiettivi) e reagisce a specifici ingressi (condizioni esterne che influiscono sul sistema) con opportune uscite (rilevabili dall'esterno).

Definizioni – Sistema

Esempi:

- Sistemi naturali: ecosistemi, organismi viventi, …
- Sistemi artificiali: elaboratore digitale, macchine automatiche, …
- Sistemi sociali: aziende, enti pubblici, …



L'obiettivo di un «sistema azienda» è produrre valore (output) tramite processi produttivi che consumano risorse (input).

Definizioni – Sistema Informativo

Sistema organizzativo: insieme delle regole e delle risorse per lo svolgimento coordinato delle attività e l'utilizzo delle risorse finalizzato al perseguimento degli obiettivi aziendali.

Sistema informativo: componente di una organizzazione che gestisce le informazioni di interesse, cioè quelle utilizzate per il perseguimento degli scopi dell'organizzazione

Ogni organizzazione ha un sistema informativo, eventualmente non esplicitato nella struttura.

Definizioni – Sistema Informativo

Vanno gestite:

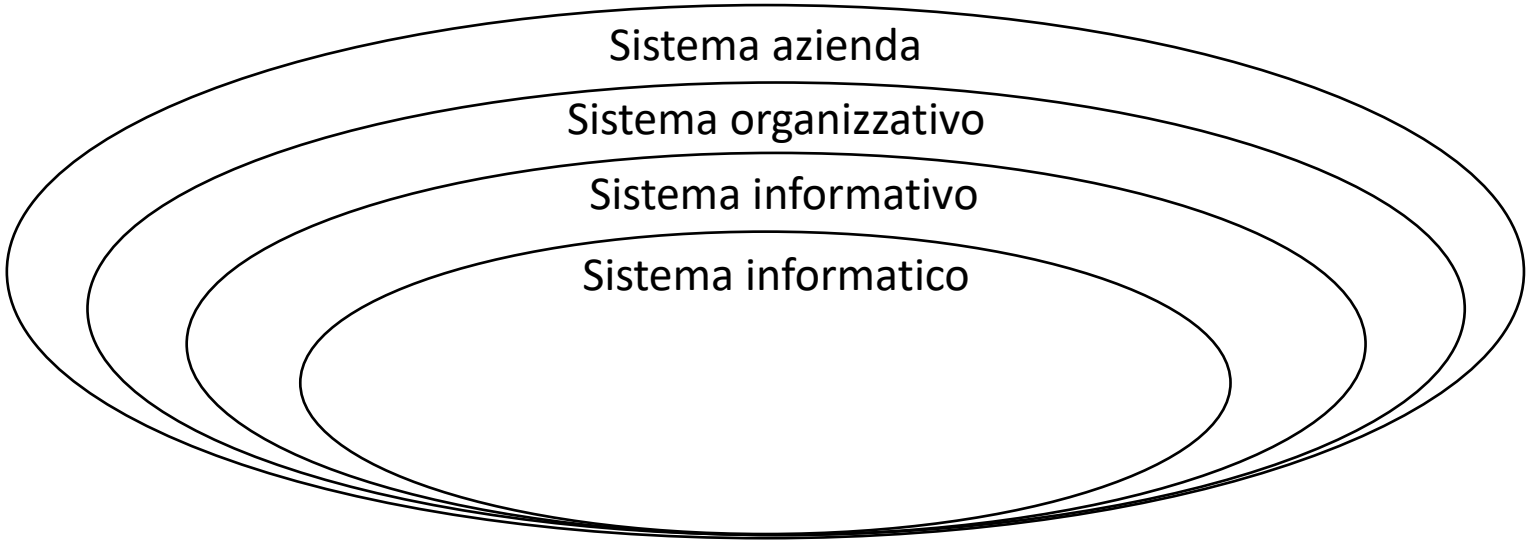
- Raccolta, acquisizione e archiviazione dell'informazione
- Elaborazione, trasformazione e produzione di informazioni
- Comunicazione e scambio di informazioni

Il concetto di «sistema informativo» è indipendente da qualsiasi automatizzazione. Es.: la gestione di servizi anagrafici.

Definizioni – Sistema Informatico

Sistema informatico (sistema di elaborazione dati): insieme di componenti hardware e software interconnesse tra di loro e preposte all'elaborazione dell'informazione in maniera automatica e dinamica.

E' la parte automatizzata del sistema informativo, cioè quella che gestisce le informazioni in maniera «automatica», con la tecnologia informatica.



Gestione delle informazioni

Informazioni in varia forma:

- Idee informali, linguaggio naturale (es. libri)
- Disegni, schemi, grafici
- …

«Memorizzata» in modo vario:

- A mente o su carta
- Su dispositivi elettronici
- …

Gestione delle informazioni

In un sistema informatico, le informazioni vengono rappresentate (come 0 e 1), in maniera essenziale, mediante dati

Informazione: notizia dato o elemento che consente di avere conoscenza più o meno esatta di fatti, situazioni, modi di essere, concetti.

Dato: ciò che è immediatamente presente alla conoscenza, prima di ogni elaborazione. In un sistema informatico sono simboli da elaborare

Gestione delle informazioni – Esempio

- Dato: rosso, giallo, verde
- Informazione: obbligo di arresto, preavviso di arresto, libera circolazione

Gestione delle informazioni – Esempio

- Dato: Foglio di carta con scritto «`Ferrari 8`»
- Informazione:
  - Se il foglio è dagli appunti di un giornalista sportivo, potrebbe essere la posizione di arrivo di una monoposto ad un gran premio
  - Se il foglio è dal blocco delle ordinazioni di un albergo, potrebbe essere una bottiglia di spumante per la camera 8

Dato come «risorsa»

In molte applicazioni, i dati hanno caratteristiche più stabili rispetto a quelle delle procedure che operano su di essi. Esempio:

- Dati relativi alle applicazioni bancarie (anagrafica, saldo conto corrente…)
- Procedure cambiano e si aggiornano frequentemente (per fornire un servizio migliore, per rispondere a cambiamenti legislativi…)

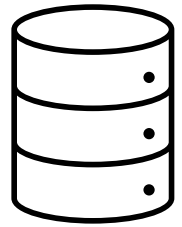
I dati costituiscono una risorsa, un patrimonio per l'organizzazione che li gestisce.

Definizioni – Base di dati

Base di dati (database): collezione organizzata di dati utilizzata per rappresentare le informazioni di interesse di un sistema informativo.

E' l'insieme organizzato di dati utilizzati per il supporto allo svolgimento delle attività di un'azienda, un ente, …

In un sistema informatico, è l'insieme dei dati gestito da un DataBase Management System (DBMS)



Gestione dei dati

Gestione dei dati

I sistemi informatici hanno necessità di gestire dati in maniera persistente, cioè memorizzando i dati su memoria secondaria. Si individuano due «approcci»:

Approccio convenzionale

Basato su files

Approccio strutturato

Basato su software di gestione dati

Approccio «convenzionale»

L'approccio più semplice, convenzionale sfrutta la presenza di file per memorizzare i dati in memoria secondaria

- Un archivio o file consente memorizzazione e ricerca
- Meccanismi di accesso e condivisione non avanzati (es. più utenti che devono accedere ad uno stesso file)
- I dati di interesse per più programmi o procedure sono replicati, causando spesso eccessiva ridondanza e possibilità di incoerenza

Approccio «convenzionale» - Esempio

Informazioni relative ai docenti in università

- L'ufficio personale elabora le informazioni relative alla «carriera» (es. distinzione tra ricercatore, professore associato, professore ordinario)
- Il dipartimento assegna e mantiene le informazioni sugli incarichi di insegnamento
- Il servizio di gestione del sito web pubblica le informazioni sugli insegnamenti
- L'ufficio stipendi usa le informazioni sulla carriera e gli insegnamenti, le leggi e i regolamenti e calcola i compensi

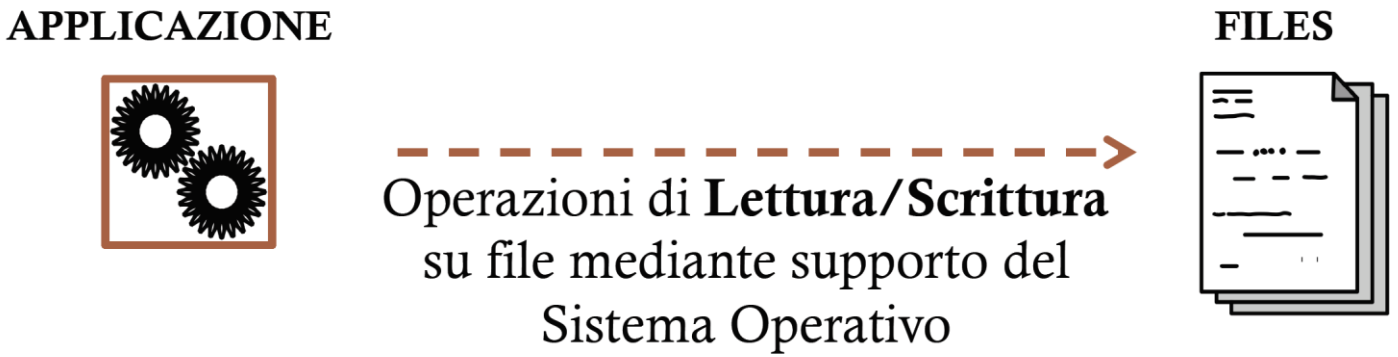
Approccio «convenzionale» - Esempio

Che succede se ognuno dei soggetti elencati gestisce le informazioni separatamente con l'approccio «convenzionale»?

Approccio «convenzionale»

Problemi:

- Gestione di grandi quantità di dati
- Problemi di scalabilità ed efficienza
- Condivisione ed accesso concorrente



Approccio strutturato

Le basi di dati sono state concepite per risolvere problemi di questa natura

- Gestendo in maniera integrata e flessibile le informazioni di interesse per diversi soggetti
- Limitando ridondanza (che però può essere utile per scopi di «backup») e incoerenza

Gestire le basi di dati mediante DataBase Management System (DBMS)

Approccio strutturato – DBMS

DBMS: sistema software in grado di gestire collezioni di dati grandi, condivise e persistenti in maniera efficiente e sicura

- Creazione di una base di dati e memorizzazione su memoria secondaria
- Accesso in lettura/scrittura ad i dati
- Condivisione di dati tra diversi utenti/applicazioni
- Protezione dei dati da accessi non autorizzati
- Disponibilità dei dati in caso di guasti (hardware/software)

Approccio strutturato – DBMS

Un «sistema di gestione di basi di dati» (DataBase Management System – DBMS) è un sistema software in grado di gestire collezioni di dati

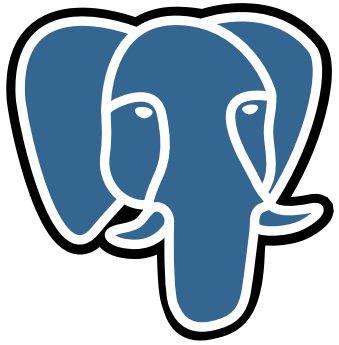
- Grandi
- Condivise
- Persistenti

Garantendo

- Privatezza
- Affidabilità
- Efficienza
- Efficacia

Approccio strutturato – DBMS

Un DBMS è un <u>prodotto software complesso</u>



Approccio strutturato – DBMS

La basi di dati sono grandi:

- Dimensioni molto maggiori della memoria centrale di un elaboratore digitale
- Il DBMS deve funzionare indipendentemente dalla quantità, l'unico limite deve essere quello fisico dei dispositivi usati
- Scale di grandezze:
  - 500 GB (Dati transazionali)
  - 10 TB (Dati decisionali)
  - 500 TB (Dati scientifici)

Approccio strutturato – DBMS

La basi di dati sono persistenti:

- Il tempo di vita è indipendente dalle singole esecuzioni dei programmi che le utilizzano (e anche dalle procedure che le utilizzano… es. dati bancari).

La basi di dati sono condivise:

- Le organizzazioni, pubbliche o private sono spesso divise in settori, dipartimenti, attività…
- Ciascun settore ha un (sotto)sistema informativo
- Vanno eliminate le ridondanze non necessarie e le incoerenze

Approccio strutturato – DBMS

Una base di dati può essere quindi una risorsa condivisa tra applicazioni diverse

- Le applicazioni svolgono attività diverse su dati condivisi: servono meccanismi di autorizzazione
- Più utenti accedono ai dati condivisi: servono meccanismi di controllo della concorrenza

Approccio strutturato – DBMS

Una base di dati può essere quindi una risorsa condivisa tra applicazioni diverse

- Le applicazioni svolgono attività diverse su dati condivisi: servono meccanismi di autorizzazione
- Più utenti accedono ai dati condivisi: servono meccanismi di controllo della concorrenza

Approccio strutturato – DBMS

I DBMS devono garantire affidabilità per le basi di dati, cioè resistere a malfunzionamenti hardware e software.

In particolare, le transazioni devo essere gestite garantendo le proprietà «ACID»

- Atomicità (*Atomicity*)
- Coerenza (*Consistency*)
- Isolamento (*Isolation*)
- Durabilità (*Durability*)

Approccio strutturato – DBMS

Transazione: insieme di operazioni da considerare indivisibile, corretto anche in presenza di concorrenza e con effetti definitivi

Atomicità: l'esecuzione di una transazione deve essere totale o nulla. Non sono ammesse esecuzioni parziali.

- Es. trasferimento di denaro da un conto X ad un conto Y
- O si fa sia il prelievo da X che il versamento in Y o nessuna delle due operazioni

Approccio strutturato – DBMS

Transazione: insieme di operazioni da considerare indivisibile, corretto anche in presenza di concorrenza e con effetti definitivi

Coerenza: i dati devo essere «integri» sia a inizio che a fine transazione, anche in presenza di concorrenza.

- Es. due utenti accedono contemporaneamente alla prenotazione di un posto in treno
- Il posto deve essere assegnato al massimo ad un utente

Approccio strutturato – DBMS

Transazione: insieme di operazioni da considerare indivisibile, corretto anche in presenza di concorrenza e con effetti definitivi

Isolamento: ogni transazione deve essere eseguita in modo isolato e indipendente dalle altre transazioni, l'eventuale fallimento di una transazione non deve interferire con le altre transazioni in esecuzione.

Approccio strutturato – DBMS

Transazione: insieme di operazioni da considerare indivisibile, corretto anche in presenza di concorrenza e con effetti definitivi

Durabilità: la conclusione positiva di una transazione corrisponde ad un impegno («commit work») a mantenere traccia del risultato in modo definitivo, anche in caso di malfunzionamento.

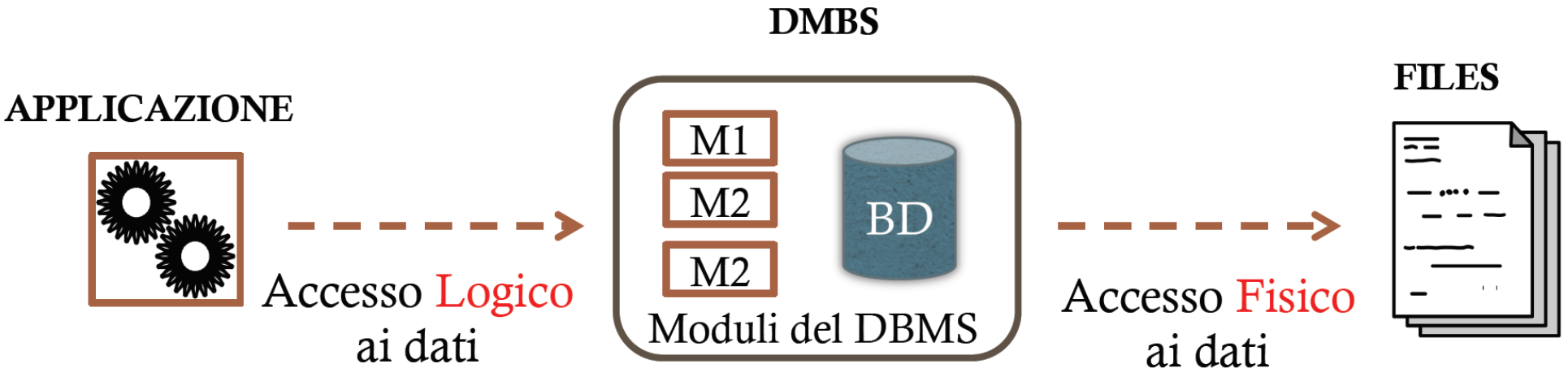
In altre parole, i risultati delle transazioni devono essere permanenti.

Approccio strutturato – DBMS

I DBMS dovrebbero garantire efficienza ed efficacia

- Dovrebbero essere efficienti nel senso di garantire di usare al meglio le risorse in termini di spazio di memoria (principale e secondaria) e tempo (di esecuzione e di risposta quando interrogati)
- Dovrebbero essere efficaci nel senso di rendere produttive le attività dei loro utenti

Approccio strutturato – DBMS



Tramite i DBMS, è possibile implementare un paradigma di separazione di dati ed applicazioni. Le applicazioni non necessitano di conoscere la struttura fisica dei dati (es. come e dove sono memorizzati su disco) ma solo la struttura logica (cosa rappresentano).

Approccio strutturato – DBMS

Rappresentazioni dei dati a livelli diversi permetto l'indipendenza dei dati dalla rappresentazione fisica.

I programmi fanno riferimento alla struttura a livello più alto e le rappresentazioni sottostanti possono essere modificate senza necessità di modifica dei programmi

Approccio strutturato – DBMS

Esempio: dati relativi ai mezzi di trasporto di un'azienda

Schema

Targa	Nome	Tipo	Portata
AB001AB	Ducato	Autocarro	1450
XF002RT	Daily	Autocarro	4900
CD003EF	Transit	Autocarro	5000
MP004DC	Punto	Autovettura	445
PD005AL	Panda Van	Autovettura	885

Istanza

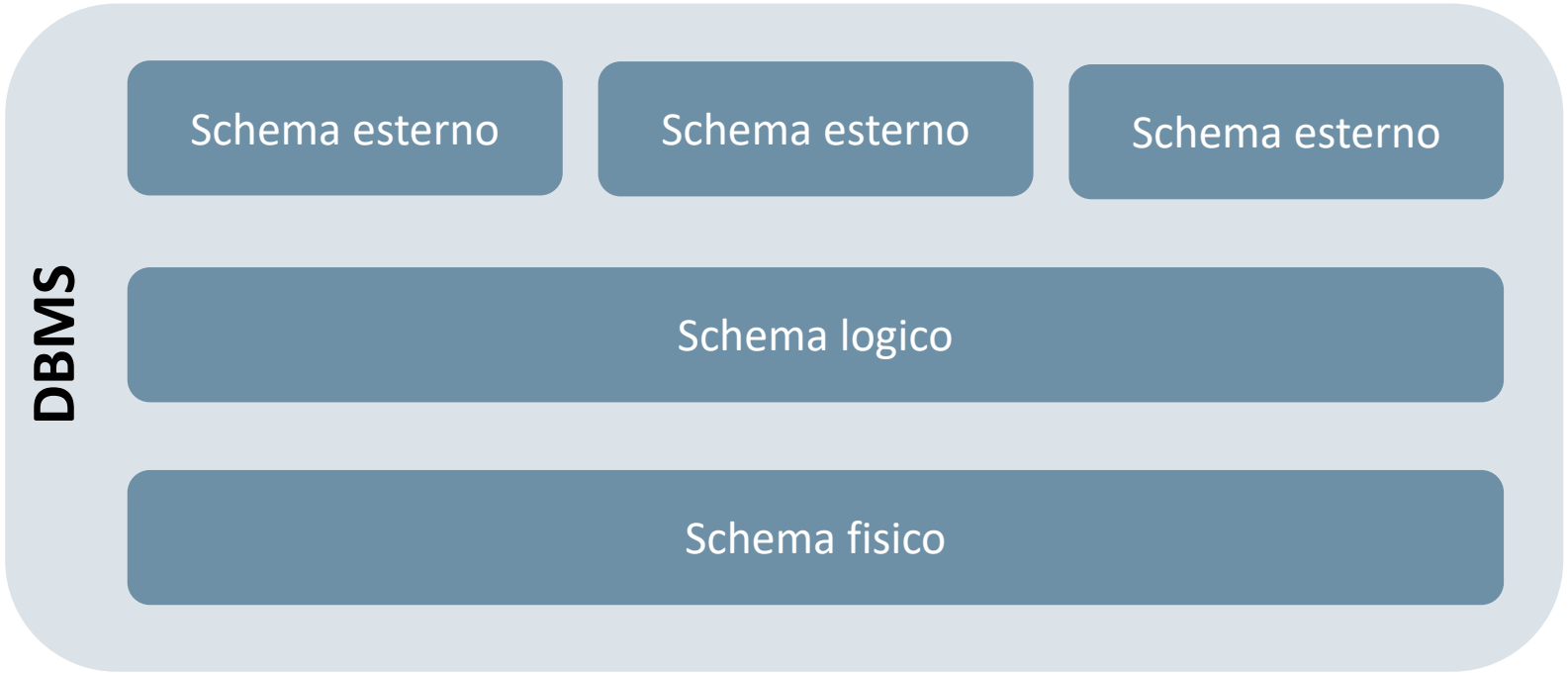
Approccio strutturato – DBMS

- Lo schema della base di dati ne descrive la struttura, che è generalmente invariante nel tempo
- L'istanza è costituita dai valori attuali, che possono cambiare molto rapidamente (es. cessione o acquisto di mezzi)

In una tabella lo schema è l'intestazione, l'istanza è costituita dalle righe di dati che la compongono.

Architettura di un DBMS

Un DBMS ha un'architettura software a 3 livelli



Descrive come si presenta il DB

Descrive cosa rappresenta il DB

Descrive come/dove sono memorizzati

Architettura di un DBMS

- Schema logico: descrizione della base di dati (es. struttura tabella)
- Schema fisico (schema interno): rappresentazione dello schema logico per mezzo di strutture di memorizzazione (es. file)
- Schema esterno: descrizione di parti della base di dati presenti nello schema logico (es. diverse categorie di utenti vedono parti diverse)

Schema esterno - Esempio

Mezzi

Targa	Nome	Tipo	Portata
AB001AB	Ducato	Autocarro	1450
XF002RT	Daily	Autocarro	4900
CD003EF	Transit	Autocarro	5000
MP004DC	Punto	Autovettura	445
PD005AL	Panda Van	Autovettura	885

Autisti

Nome	Targa
Rossi	AB001AB
Bianchi	PD005AL
Verdi	CD003EF

Assegnamento mezzi

Nome Autista	Nome Mezzo	Portata
Rossi	Ducato	1450
Bianchi	Panda Van	885
Verdi	Transit	5000

Architettura di un DBMS

Indipendenza modello logico – modello fisico

- L'organizzazione logica dei dati non dipende dalle strutture dati usate per l'effettiva memorizzazione su disco.
- In pratica, le applicazioni accedono al DBMS specificando i concetti logici del modello dei dati, piuttosto che i dettagli relativi alla loro memorizzazione.

Indipendenza dai dati

L'accesso avviene solo dal livello «esterno», che può coincidere con il livello logico

Indipendenza fisica: il livello logico e quello esterno sono indipendenti da quello fisico. Es.: le informazioni sull'assegnamento dei mezzi sono usate allo stesso modo qualunque sia la realizzazione fisica.

Indipendenza logica: il livello esterno è indipendente da quello logico (aggiunte o modifiche allo schema esterno non modificano il livello logico).

DBMS - Vantaggi

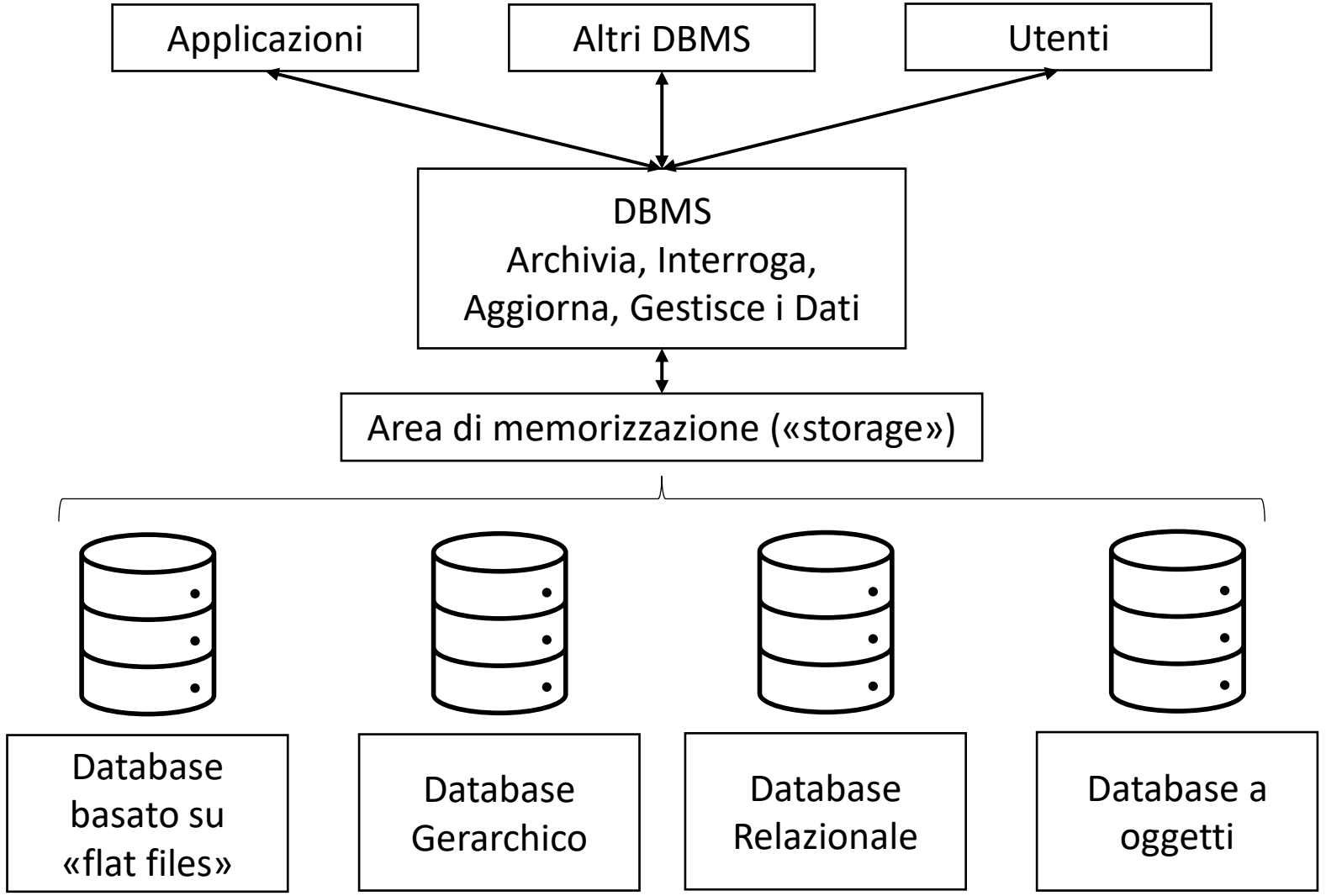
- L’indipendenza dei dati facilita manutenzione e sviluppo delle applicazioni
- Standardizzazione
- Riduzione inconsistenze
- I dati diventano una risorsa a disposizione di tutte le componenti di un’organizzazione/azienda
- La base di dati fornisce un modello unificato della realtà di interesse per l’organizzazione/azienda

DBMS - Svantaggi

- Il costo dei DBMS è significativo e ancor di più lo è la transizione verso un DBMS
- Alcune funzionalità non necessarie non possono essere scorporate (perdita di efficienza)

Quale DBMS?

DBMS


Applicazioni
Altri DBMS
Utenti
DBMS
Archivia, Interroga,
Aggiorna, Gestisce i Dati
Area di memorizzazione («storage»)
Database
basato su
«flat files»
Database
Gerarchico
Database
Relazionale
Database a
oggetti

DBMS – Modello logico

I DBMS differiscono per il modello logico usato per i database:

Database relazionali («SQL databases») *vs* Database non relazionali («NoSQL databases»)

Structured Query Language (SQL): linguaggio standardizzato per database relazionali (creazione e modifica dello schema del database, creazione, modifica e cancellazione dei dati, interrogazione del database)

Modello Relazionale

Basato sui concetti di relazione (formale) e tabella (informale)

Relazione:

Dati $n > 0$ insiemi $D_1, D_2, \dots, D_n$ (non necessariamente distinti), il prodotto cartesiano di $D_1, D_2, \dots, D_n$ indicato con $D_1 \times D_2 \times \cdots \times D_n$ è costituito dalle n-uple $(v_1, v_2, \dots, v_n)$ tali che v_i appartiene a D_i per $1 \leq i \leq n$.

Una relazione è un sottoinsieme del prodotto cartesiano $D_1 \times D_2 \times \cdots \times D_n$

Modello Relazionale

Basato sui concetti di relazione (formale) e tabella (informale)

Relazione, es:

Se A e B sono i due insiemi $A = \{1, 2, 4\}$ e $B = \{a, b\}$, allora il prodotto cartesiano $A \times B$ è

$$\{(1, a), (1, b), (2, a), (2, b), (4, a), (4, b)\}$$

Una possibile relazione è il sottoinsieme

$$\{(1, b), (2, b), (4, b)\}$$

Modello Relazionale

Basato sui concetti di relazione (formale) e tabella (informale)

Mezzi

Targa	Nome	Portata
AB001AB	Ducato	1450
XF002RT	Daily	4900
CD003EF	Transit	5000
MP004DC	Punto	445
PD005AL	Panda Van	885

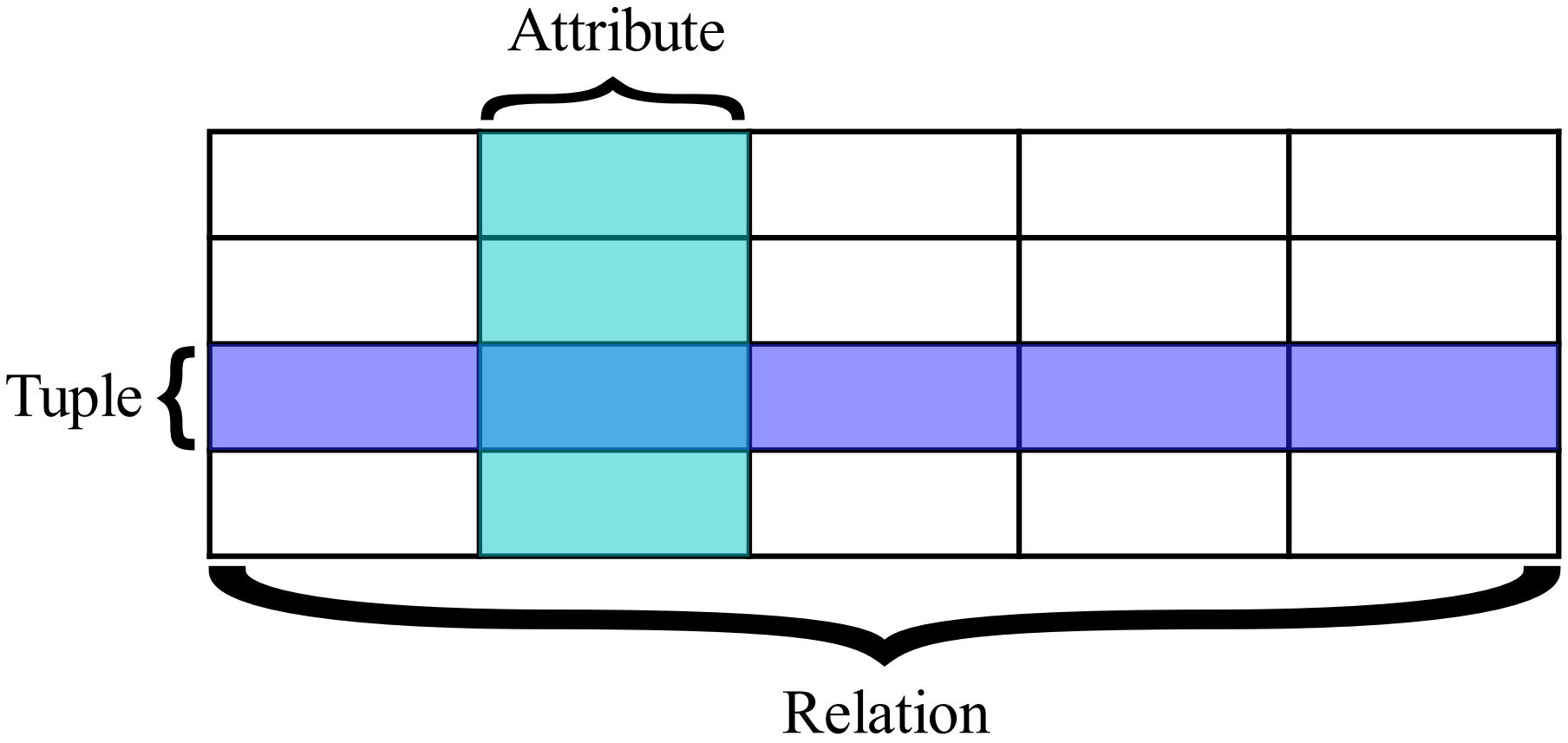
Autisti

Matricola	Nome	Cognome	Data di nascita
001024	Mario	Rossi	16/12/1979
001198	Sara	Bianchi	07/03/1981
008736	Chiara	Verdi	08/08/1989

Responsabilità

MatricolaAut	TargaMezzo
008736	XF002RT
008736	MP004DC
001024	AB001AB

Modello Relazionale



Modello Relazionale

- Modello più diffuso (proposto da E.F. Codd nel 1970, disponibile in DBMS reali a partire dal 1981)
- Schema dei dati preciso e rigido
- La maggior parte delle implementazioni garantiscono le proprietà ACID anche su operazioni «multi-record»
- Inconsistenze e ridondanze ridotte al minimo (modello tipicamente *centralizzato*)
- Combinare tuple da tabelle diverse («join») potrebbe essere oneroso
- Scalabile «verticalmente» (cioè aumentando la capacità di calcolo del singolo server)

Modelli Non Relazionali («NoSQL»)

Tutti i modelli di dati non basati su relazioni, ad es.

- Chiave-valore
- Database «document-oriented»
- Basati su grafi
- …

Ci si riferisce a queste basi di dati anche come memorizzazione strutturata («structured storage»).

Modelli Non Relazionali («NoSQL»)

Database chiave-valore («key-value»)

- I record di dati sono memorizzati in corrispondenza ad una chiave univoca, in un «dizionario»
- Non c'è schema fisso dei dati, ogni record può avere struttura diversa dagli altri

Chiave	Valore
K1	Mario, Rossi, 11/07/1983
K2	35,67 €
K3	Iveco Daily

Modelli Non Relazionali («NoSQL»)

Database chiave-valore («key-value»)

- Lettura/scrittura veloci
- Nessun problema in caso di cambiamento dello schema che descrive possibili record
- Ricerca lenta
- Mancanza di standardizzazione

Modelli Non Relazionali («NoSQL»)

Database orientati ai documenti («document-oriented»)

- Sono una sottoclasse dei database chiave-valore.
- In corrispondenza della chiave viene memorizzato un «documento» di cui la struttura è definita (a differenza del database chiave-valore).
- Permette la ricerca non solo per chiave, ma anche sulla base del contenuto di specifici campi del documento

Modelli Non Relazionali («NoSQL»)

Database orientati ai documenti («document-oriented»), esempio

```
{
    _id = “AB001AB”,
    nome: “Ducato”,
    tipo: “Autocarro,
    portata: 1450
}
```

Modelli Non Relazionali («NoSQL»)

Vantaggi (rispetto al modello relazionale)

- Utile quando lo schema dei dati non è rigido
- Mapping diretto con il concetto di «oggetto» della programmazione orientata agli oggetti
- Scalabilità «orizzontale» (si può distribuire più facilmente il «carico» su più macchine)
- Non ci sono relazioni: documenti vengono annidati dentro altri documenti, velocizzando il tempo di «interrogazione»

Modelli Non Relazionali («NoSQL»)

Svantaggi (rispetto al modello relazionale)

- Molte implementazioni non garantiscono le proprietà ACID nelle operazioni che coinvolgono più documenti
- I dati possono essere duplicati (documenti all'interno di altri documenti)
- Più complesso gestire inconsistenze

«SQL» vs «NoSQL»

- Coerenza
- Dati già strutturati
- «ACID»
- Necessità di centralizzazione

Database relazionali

- Tempo di interrogazione
- Duplicati non problematici
- Schema dei dati flessibile
- Mapping diretto con «oggetti»

Database non relazionali

«SQL» vs «NoSQL»

Altri «parametri di confronto»:

- Integrazioni
  - Con gestionali/software già in uso
  - Con linguaggi di programmazione per customizzazioni
- Licenza
  - Soluzioni proprietarie vs Soluzioni open source
- Disponibilità di documentazione/community

«SQL» vs «NoSQL»

Altri «parametri di confronto»:

- Dimensioni massime gestibili
- Operazioni ammesse sui dati (unioni, intersezioni, join, …)
- Controllo accesso
- Indicizzazione
- …

DBMS «commerciali»

	Oracle database	MySQL	Microsoft SQL Server	PostgreeSQL	MongoDB	Redis	IBM Db2
Sviluppatore	Oracle	Oracle	Microsoft	PostgreSQL	MongoDB	Redis	IBM
Primo rilascio	1979	1995	1989	1989	2009	2009	1983
Ultimo rilascio	2019	2022	2019	2022	2022	2022	2021
Modello logico	Relazionale	Relazionale	Relazionale	Relazionale	Document-oriented	Chiave-valore	Relazionale
ACID	Sì	Sì	Sì	Sì	Sì	No	Sì
Sistemi Operativi	Windows, Mac OS, Linux	Windows, Mac OS, Linux	Windows	Windows, Mac OS, Linux	Windows, Mac OS, Linux	Unix-like	Windows, Mac OS, Linux
Licenza	Proprietario	GPL v2	Proprietario	Open source	Open source	Open source	Proprietario

«Most popular DBMS»: https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/