

I sistemi informativi aziendali: l'importanza delle informazioni

di Roberta Molinari

Dati e Informazioni

- ▶ Il termine **dato** significa letteralmente "*fatto*".
- ▶ I dati sono una rappresentazione dei fatti.
- ▶ Un dato costituisce un'informazione se fornisce una nuova conoscenza attraverso una chiave di interpretazione
- ▶ **L'informazione** è l'incremento di conoscenza che può essere acquisita dai dati.

es. il dato 12, senza chiave di interpretazione, non costituisce informazione. Con la chiave di interpretazione "Quantità disponibile" assume l'informazione di "quantità disponibile per un determinato articolo"

Le Informazioni

- ▶ In ogni sistema di vita dell'uomo vengono trattate **informazioni**.
- ▶ Costituiscono un grande patrimonio, sono considerate risorse preziosissime grazie alle quali lo stesso sistema umano sopravvive.
- ▶ Individuate e raccolte devono essere memorizzate in modo che si possano facilmente eseguire le operazioni **CRUD**:
 1. *Create* - **AGGIUNGERE**
 2. *Read* - **RECUPERARE**
 3. *Update* - **MODIFICARE**
 4. *Delete* - **CANCELLARE**

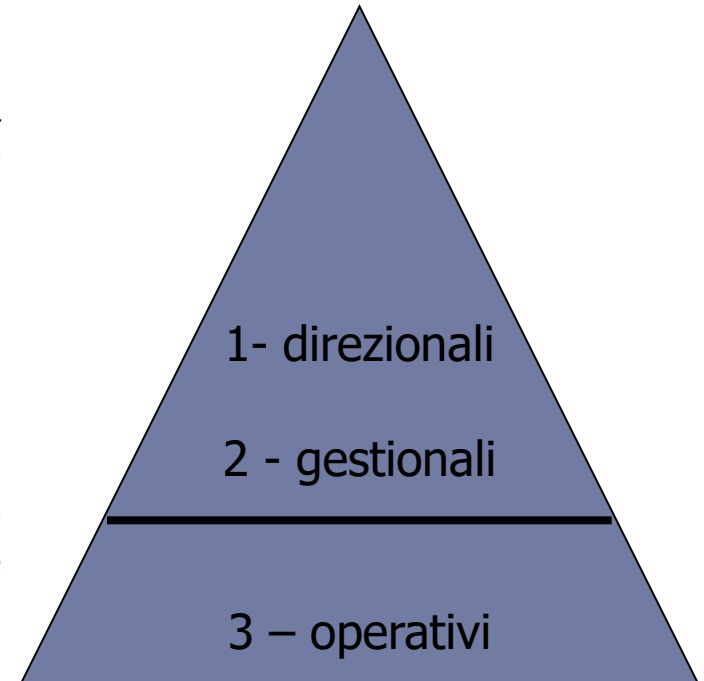
- ▶ Ogni impresa si organizza in base:
 - ▶ alla sua **missione** o **mission** (lo scopo per cui è nata: produrre scarpe)
 - ▶ ai suoi **obiettivi generali** o **target** a breve o a lungo termine (aumentare il fatturato)
- ▶ Al suo interno si possono individuare:
 - ▶ **Risorse**: tutto ciò con cui opera (materiale, immateriale, persone, interne o esterne)
 - ▶ **Processi**: insieme di attività (decisioni e azioni) svolte per il raggiungimento di mission e target tramite un risultato definito e misurabile (prodotto)

Classificazioni dei processi

Piramide di Anthony

Processi organizzativi

1. concorrono alla definizione degli obiettivi strategici Definiscono i piani a medio e lungo termine, progettano l'organizzazione dell'intera azienda (filiali) (decisioni strategiche)
2. concorrono alla traduzione degli obiettivi in criteri di gestione-programmazione ed effettuano il controllo del raggiungimento di tali obiettivi Controllano il corretto utilizzo delle risorse e definiscono piani a breve e medio termine (raggiungimento di un certo budget) (decisioni tattiche)



Processi operativi

3. concorrono all'attuazione concreta degli obiettivi (produzione delle scarpe)

Dati- Informazioni - Conoscenza

- ▶ Un caso molto particolare di risorsa su cui operano tutte le aziende è l'informazione. L'informazione è infatti una risorsa che riguarda tutte le altre risorse.
- ▶ I **dati** sono una materia prima in continua crescita
- ▶ Le **informazioni** sono il valore aggiunto ai dati
- ▶ Possedere la **conoscenza** è un'esigenza fondamentale per poter prendere decisioni



Sistema informativo aziendale (S.I.)

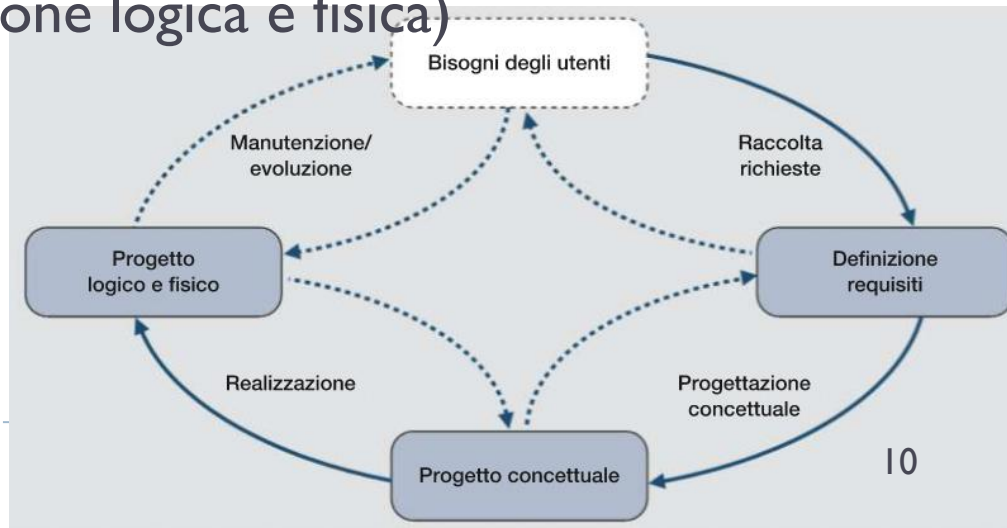
- ▶ Il **sistema informativo SI** è l'insieme delle persone, dei mezzi e delle procedure che riguardano la *raccolta*, la *produzione*, l'*archiviazione*, l'*elaborazione*, la *distribuzione* dei dati al fine di ottenere informazioni che servono da supporto alle funzioni operative, ai processi decisionali e al controllo di una organizzazione.

Sistema informatico aziendale

- ▶ Quella parte del sistema informativo in cui le informazioni sono raccolte, elaborate, archiviate, scambiate mediante l'uso dell'**ICT** (**I**nformation & **C**ommunication **T**echnology, sono le tecnologie della informazione e della comunicazione) costituisce il **sistema informatico**, anche chiamato **EDP** (**E**lectronic **D**ata **P**rocessing).
- ▶ È il sottoinsieme del sistema informativo formato da una componente:
 - ▶ **Software**: archivi e programmi di gestione (applicazioni)
 - ▶ **Hardware**: supporti fisici, computer, rete, infrastruttura,...

Ciclo di vita del sistema informatico

- L'informatizzazione di un SI deve essere occasione di razionalizzazione delle attività per renderle più efficaci e efficienti.
- Ci sono metodologie differenti, ma in generale è un processo ciclico e sono previste tre fasi
 - Raccolta delle richieste degli utenti
 - Progettazione concettuale
 - Realizzazione (progettazione logica e fisica)



Le basi di dati

Gli archivi

Gli **archivi** servono per conservare i dati in modo permanente, per poter essere reperiti e utilizzati in seguito.

Gli archivi sono un insieme di dati i cui elementi hanno le seguenti caratteristiche:

- sono legati tra loro da un *nesso logico* (si riferiscono ad uno stesso argomento Es. rubrica telefonica)
- sono rappresentati secondo un certo *formato* per permetterne l'interpretazione (Es. nominativo, indirizzo, tel)
- sono registrati in modo permanente su un certo *supporto* su cui è possibile leggere e scrivere (Es. la rubrica cartacea)
- sono *organizzati* per permetterne la consultazione (Es. ci sono le etichette delle lettere)

Se il supporto è di tipo informatico si parla di **Archivi elettronici** o **File di dati**

Gli archivi

Record logici

I dati, in generale, sono raggruppati in unità logiche ognuna riferita ad un singolo soggetto della “realtà” memorizzata.

Ogni unità è un **record logico**, che a sua volta è suddiviso in **campi**, i cui valori caratterizzano il soggetto. L'elenco dei campi e il relativo tipo costituiscono il **tracciato record**.

Es. di tracciato record per la rubrica telefonica

Cognome-Nome	Indirizzo	Numero Telefono
Alfabetico	Alfanumerico	Numerico

Es. di record

Rossi Mario	Via Roma 1	5556346
-------------	------------	---------

Gli archivi

Record logici

Campo: spazio riservato per memorizzare il dato relativo ad un'informazione di senso compiuto (es. indirizzo). Singolo dato

Record: insieme di campi correlati tra loro formanti un'unità informativa più ampia (es. persona).

Insieme di dati di uno stesso soggetto



Nome	Indirizzo	Telefono
Mario Rossi	Via Roma 1	0171-66666
Bianchi Luisa	P.zza Italia 14	011-999999
Verdi Ugo	Via Po 12	011-444444

Archivio: insieme di record omogenei (es. rubrica).

Insieme di dati di un archivio di più soggetti

Gli archivi

Operazioni

- **Creazione dell'archivio:** si definisce il supporto, il tracciato record, il nome e l'organizzazione
- **Manipolazione dei dati:**
 - *Inserimento*
 - *Modifica o aggiornamento*
 - *Cancellazione*
- **Interrogazione o consultazione dei dati:** reperimento di informazioni
- **Distruzione**

Archivi

Limiti archivi tradizionali file-based

- ▶ I programmi sono legati al linguaggio utilizzato: modifiche della struttura record comportano modifiche ai programmi che la utilizzano
- ▶ L'accesso ai dati è determinato dal tipo di organizzazione degli archivi (sequenziale, diretto) e si devono comunque leggere record per record
- ▶ Alcuni dati si presentano più volte nello stesso file o in file differenti
- ▶ L'accesso ai dati avviene solo tramite l'applicazione. Nuove interrogazioni richiedono la modifica delle applicazioni

Archivi

Limiti archivi tradizionali file-based

- ▶ In una gestione tradizionale ogni applicazione opera sui suoi dati la cui struttura è definita all'interno del programma stesso. È complicata la condivisione di dati da parte di più applicazioni.
- ▶ Se i dati sono utilizzati da più applicazioni, spesso vengono duplicati, con spreco di memoria e rischio di inconsistenza dei dati

Archivi

Limiti archivi tradizionali file-based

Ordini							
CodOrd	Codacc	Descr	Prezzo	Qt	Codcli	Nome	Indirizzo
01	M03	Batteria	100,00	3	010	Rossi Mario	Via Roma 1
01	M12	Antenna	25,00	1	010	Rossi Mario	Via Roma 1
02	M03	Batteria	100,00	2	020	Verdi Luca	C.so Italia 10

Archivi

Limiti archivi tradizionali file-based

Ridondanza

- ▶ Gli stessi dati appaiono in più punti

Incongruenza

- ▶ Conseguenza della ridondanza in caso di modifiche parziali delle occorrenze dei dati ripetuti

Inconsistenza

- ▶ Conseguenza della incongruenza: i dati non sono più affidabili

Database

Definizione

I **database** nascono per superare i limiti degli archivi tradizionali.

Sono una raccolta di dati (archivi) organizzati per essere usati in modo efficiente da differenti applicazioni e da utenti diversi. (è sicuramente su supporto informatico)

Nei database è presente sia la definizione della struttura dei dati (tracciato record: numero nome e tipo) che i dati stessi. La struttura non fa più parte dell'applicazione, è ora indipendente da essa.

Database

Teoria delle basi di dati

- ▶ Nell'informatica la **teoria delle basi di dati** studia come organizzare al meglio grandi quantità di informazioni, per poterle gestire in modo:
 - ▶ **Semplice**: per utenti e applicazioni
 - ▶ **Efficiente**: in tempo e spazio
 - ▶ **Efficace**: rappresentano realmente la realtà che si vuole gestire
 - ▶ **Sicuro**: da utenti non autorizzati
 - ▶ **Solido**: resistente a guasti o errori accidentali degli operatori
 - ▶ **Condiviso**: permettere l'accesso simultaneo

Database

Caratteristiche

I **database** garantiscono:

- ▶ **Indipendenza dalla struttura fisica dei dati:** si possono modificare i supporti senza modificare la struttura logica e quindi le applicazioni
- ▶ **Indipendenza dalla struttura logica dei dati:** si possono modificare le definizioni delle strutture senza modificare le applicazioni
- ▶ **Condivisione:** utilizzo da parte di più utenti o più applicazioni, è consentito accesso concorrente e viste parziali

Database

Caratteristiche

- ▶ **Eliminazione della ridondanza:** non si devono duplicare dati perché gli archivi sono integrati
- ▶ **Sicurezza dei dati:** protezione da accessi non autorizzati e guasti, si effettua tramite autenticazione, autorizzazione e controllo integrità
- ▶ **Integrità e recupero dei dati:** vi sono controlli per recuperare anomalie causate da programmi o utenti autorizzati
- ▶ **Garantita la consistenza dei dati:** contro il pericolo dovuto ad accessi concorrenti di lettura/scrittura
- ▶ **Facilità di accesso:** accesso semplice e veloce
- ▶ **Interrogazioni:** richieste di dati che verifichino un certo criterio di ricerca
- ▶ **Garantita l'integrità dei dati** a 3 livelli:
 - ▶ di campo (tipo e vincoli espliciti)
 - ▶ di tabella (integrità sull'entità: non duplicati e pk non nulla)
 - ▶ di associazione (integrità referenziale)

Garantire l'**integrità** significa garantire la consistenza, validità dei dati. Un **vincolo di integrità** è una proprietà che deve essere soddisfatta dalle istanze di un database

DataBase Management System

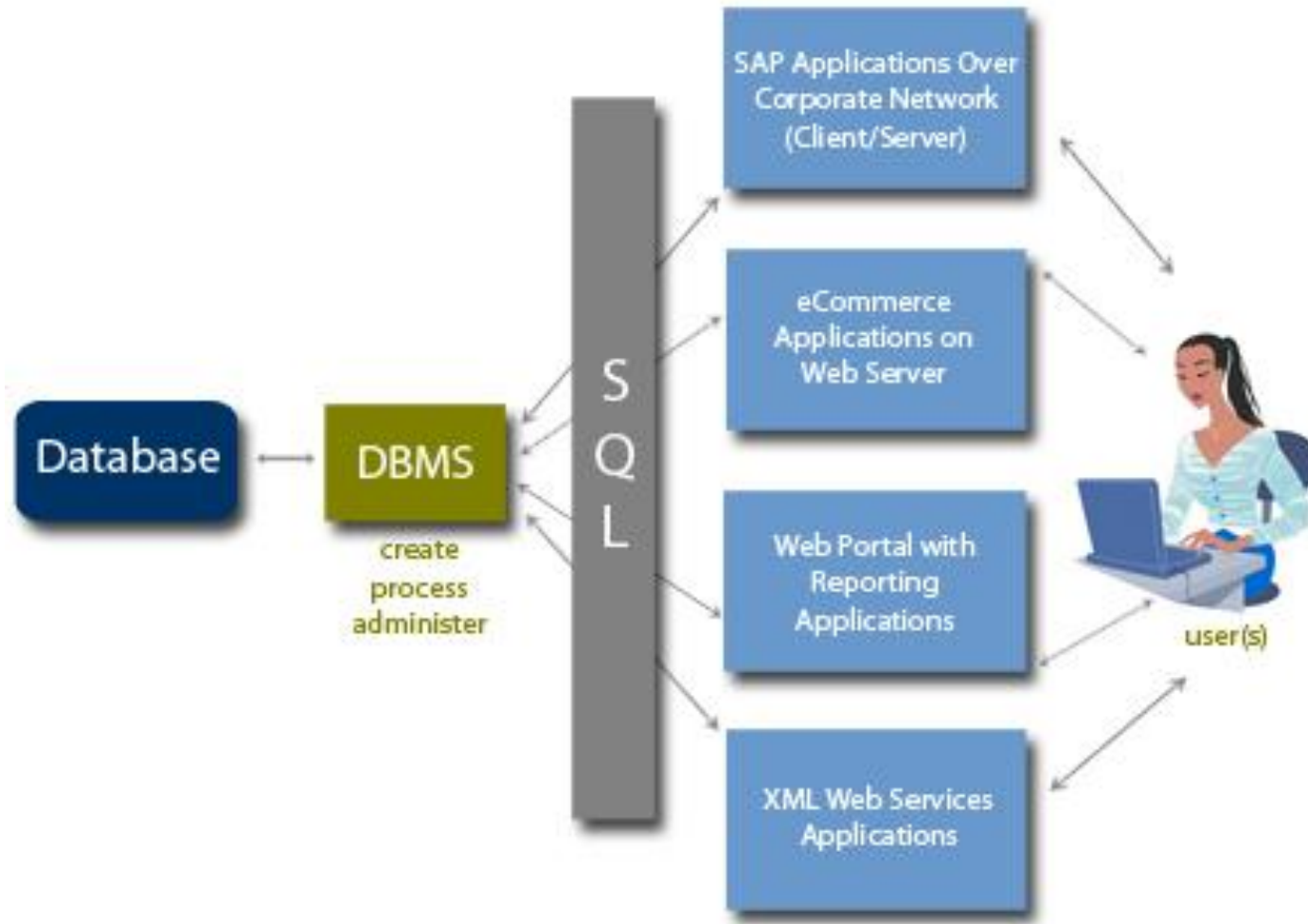
DBMS

I prodotti software per la gestione dei DB si chiamano **DBMS - DataBase Management System**.
Costituiscono un interfaccia utente/applicazione-DB.
Permettono di:

- ▶ Definire la struttura dei dati (modello logico)
- ▶ Manipolare ed interrogare il DB
- ▶ Controllare l'integrità dei dati
- ▶ Permettere la condivisione
- ▶ Garantire sicurezza e protezione e persistenza
- ▶ Gestire il modo in cui fisicamente sono archiviati i dati

DataBase Management System

DBMS



DBMS: modifica e interrogazione

- I linguaggi di interrogazione del database mediante *query* (**interrogazioni**) e i generatori di *report* permettono agli utenti di interrogare in maniera interattiva il database e di analizzarne i dati.
- Il DBMS fornisce un modo per **aggiornare e immettere** nuovi dati nel database, oltre che per interrogarlo, questa capacità permette di gestire database personali.

DBMS: integrità

Il DBMS può mantenere **l'integrità** del database:

- non consentendo a più utenti di modificare lo stesso record contemporaneamente (blocco del record).
- Il database può impedire l'immissione di due record duplicati; per esempio può essere impedita l'immissione nel database di due clienti con lo stesso numero identificativo ("campi chiave").
- L'integrità è garantita dalla proprietà "ACID" delle transizioni.

DBMS: gestione autorizzazioni

Il sistema di sicurezza dei dati impedisce agli utenti non autorizzati di visualizzare o aggiornare il database. Mediante l'uso di *password* (parole d'ordine) agli utenti è permesso l'accesso all'intero database o a un suo sottoinsieme: in questo secondo caso si parla di ***subschema o vista***. Per esempio un database di impiegati può contenere tutti i dati riguardanti un singolo soggetto e un gruppo di utenti può essere autorizzato a vedere solamente i dati riguardanti lo stipendio, mentre altri utenti possono essere autorizzati a vedere solamente le informazioni che riguardano la sua storia lavorativa e la situazione sanitaria.

DataBase Management System

I linguaggi di un DBMS

I DBMS permettono tramite linguaggi (comandi) specifici di:

- ▶ Definire la struttura dati logica, definire le maschere video e i prospetti (Data Definition Language-**DDL**)
- ▶ Definire la struttura fisica relativamente ad una specifica MM (Data Media Control Language-**DMCL** o Storage Definition Language-**SDL**)
- ▶ Manipolazione dei dati (Data Manipulation Language-**DML**)
- ▶ Definire i vincoli di sicurezza, le autorizzazioni agli accessi e tipi di operazioni consentite agli utenti (Data Control Language-**DCL**)
- ▶ Interrogazione del DB (**Data Query Language – DQL**)
- ▶ TCL (**Transaction Control Language**): consente di avviare, concludere e gestire le transazioni.

DataBase Management System

I linguaggi di un DBMS in SQL



DataBase Management System

Architettura a 3 livelli

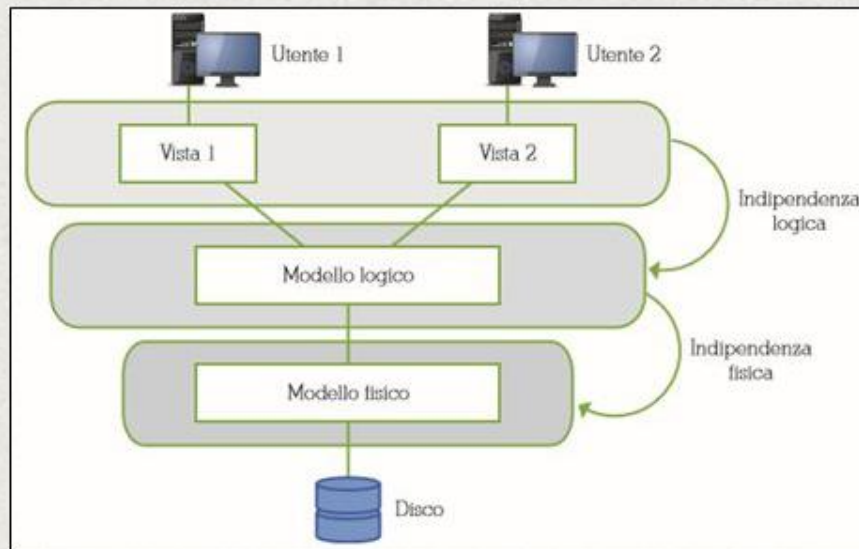
I DBMS permettono di interagire con il DB su 3 livelli:

- ▶ **Livello esterno:** usato dagli utenti del DB. Il **DBA** (*DB Administrator*) ha realizzato per essi delle **viste** differenti in base a permessi definiti con il **DCL** e compiti degli utenti. A questo livello è possibile modificare i dati con il **DML** o fare interrogazioni con il **DQL**.
- ▶ **Livello concettuale o logico:** viene definito lo schema dei dati indipendentemente dalla implementazione fisica. Si usa il **DDL**.
- ▶ **Livello interno o fisico:** a questo livello il DBA decide i supporti di memorizzazione, l'organizzazione, i metodi di accesso per il DB,... Si usa il **DMCL**.

DataBase Management System

Architettura a 3 livelli

- il **livello fisico** gestisce i *file* che dovranno essere memorizzati sul disco,
- il **livello logico** gestisce le tabelle relazionali,
- il **livello di interfaccia verso l'esterno** si occupa di quali dati far vedere ("vista") agli utenti e in che modalità.



DataBase Management System

Architettura a 3 livelli

L'architettura a tre livelli dei db, descrive i dati secondo 3 differenti livelli di astrazione, mediante opportuni schemi. Il DBMS realizza questi meccanismi di astrazione dei dati e assicura l'**indipendenza dei dati**: i livelli superiori non sono influenzati , entro certi limiti, dai cambiamenti che avvengono in quelli inferiori:

- ▶ **Indipendenza dalla struttura fisica dei dati**: si possono modificare i supporti, spostare tabelle, ecc. senza modificare il livello logico e quindi quello esterno realizzato dalle applicazioni
- ▶ **Indipendenza dalla struttura logica dei dati**: si possono modificare le definizioni delle strutture fatte a livello logico, senza modificare le viste esterne utilizzate dalle applicazioni

DataBase Management System

Tipi di architettura

- ▶ **Stand-alone:** insieme di dati di piccole dimensioni risiedente su un PC (Access, SQLite).
- ▶ **Terminal server:** architettura diffusa negli anni passati in cui in un mainframe erano presenti dati e procedure che venivano utilizzati da più utenti ai vari terminali (DB2 della IBM per AS400 e Oracle)
- ▶ **Client-server:** su un computer server risiedono i dati e il programma database server in grado di rispondere alle richieste del sw database client residente su altri computer collegati in rete. La comunicazione tra client e server prevede: instaurazione della connessione e dell'autenticazione, richiesta dal client che resta in attesa della risposta del server (DBMS MySQL, MariaDB).

DataBase Management System

Transazioni

- ▶ **Transazione:** insieme di operazioni che devono essere eseguite in modo atomico, come un unico blocco: **L'atomicità** delle transazioni implica che o vengono eseguite completamente o non vengono eseguite. Se una transazione è annullata, per ripristinare la situazione iniziale si devono annullare tutte le operazioni eseguite fino a quel momento.
- ▶ Si classificano in
 - ▶ **Implicite**, create in automatico dal DBMS quando esegue operazioni di aggiornamento, inserimento e cancellazione
 - ▶ **Esplicite**, dichiarate dal programmatore

Es.

- ▶ Trasferimento di denaro tra conti correnti.
- ▶ Prenotazione di un posto in aereo

DataBase Management System

Transazioni – Esempio

```
begin_transaction;  
read (saldoX);  
saldoX = saldoX - importo;  
write (saldoX);  
read (saldoY);  
saldoY = saldoY + importo;  
write (saldoY);  
if(saldoX < fido) then  
    rollback;  
else  
    commit;  
end_if;  
end_transaction;
```

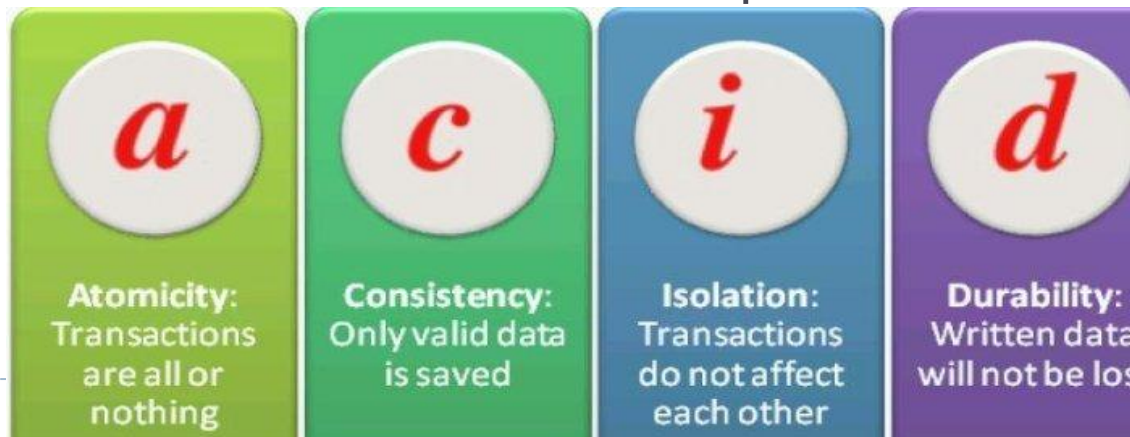
Annulla tutte
le modifiche
dall'inizio della
transazione

Conferma tutte
le modifiche
apportate nella
transazione

DataBase Management System

Transazioni - Proprietà ACID

- ▶ Il DBMS deve garantire alle transazioni le seguenti proprietà
 - ▶ **Atomicity (atomicità):** tutte le istruzioni sono eseguite come un'unica unità (o tutto o niente)
 - ▶ **Consistency (consistenza):** il DB si deve trovare in uno stato consistente sia all'inizio che alla fine di una transazione, non deve violare eventuali vincoli di integrità
 - ▶ **Isolation (isolamento):** in caso di transazioni concorrenti solo una può modificare i dati, se una fallisce le altre non devono fallire
 - ▶ **Durability (persistenza):** una volta giunta a termine (commit), le modifiche effettuate dalla transazione sono permanenti nel DB



DataBase Management System

Transazioni – Transaction Log

- ▶ Per garantire l'atomicità il DBMS non esegue gli aggiornamenti direttamente sul DB, ma registra le operazioni da effettuare in un file di sistema chiamato **transaction log**. Ad ogni operazione sul DB si aggiunge un record nel file contenente:
 - ▶ Before image: dato prima della modifica
 - ▶ After image: dato dopo aggiornamento
 - ▶ User: chi ha richiesto la modifica
- ▶ Se la transazione è portata a termine (**committed**), allora viene effettivamente eseguita sul DB e si segna il punto in cui è arrivato nell'esecuzione delle operazioni presenti nel log, con un check point
- ▶ Se la transazione è abortita (**aborted**), non si aggiornano i file e il record corrispondente nel file log viene cancellato

DataBase Management System

Transazioni – Lock

- ▶ Nel caso di transazioni concorrenti che vogliono modificare gli stessi dati, il DBMS per garantire la consistenza imposta dei blocchi (**lock**) alle informazioni in modo che solo una transazione possa modificarli.
- ▶ Questi blocchi possono essere impostati a livello di tabella, di riga e di campo.
- ▶ Si distinguono
 - ▶ **Blocchi ottimistici**: in cui le informazioni anche se bloccate da un'altra transazione, possono essere letti da altre transazioni
 - ▶ **Blocchi pessimistici**: in cui le informazioni bloccate non possono neanche essere letti da altre transazioni

DataBase Management System

Backup

Uno dei compiti fondamentali di un DBMS è quello del salvataggio periodico dei dati (**backup**) e dell'eventuale loro ripristino (**restore**). Per poter ripristinare il DB, oltre ai file di backup, utilizza anche il **transaction log**.

Si parla di backup a **caldo** (o Hot backup) quando è effettuato mentre il database è in linea. I dati possono quindi essere modificati mentre il backup è in corso.

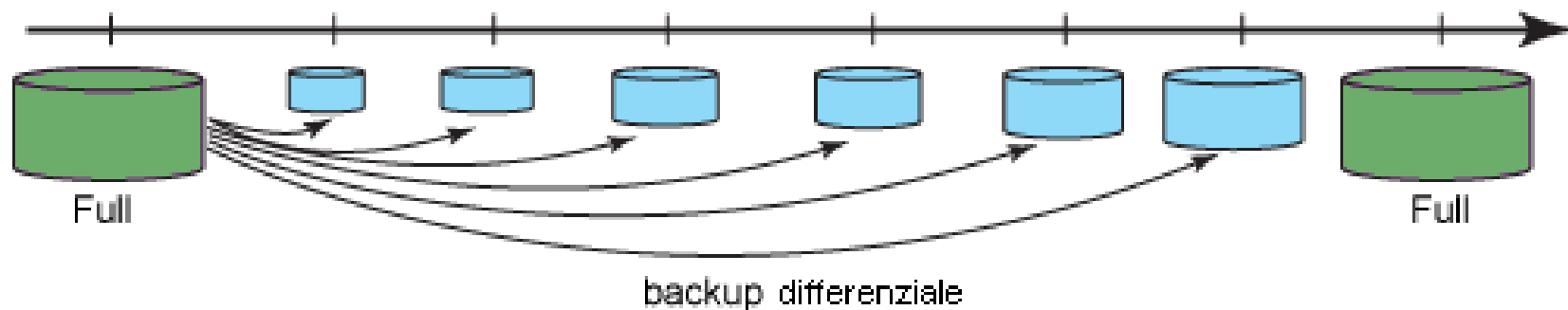
I backup si classificano in:

1. **Completo** (o Full backup): backup di tutti i files sul sistema. A differenza della disk image, un full backup non include le tavole di allocazione, le partizioni ed i settori di boot.

DataBase Management System

Backup

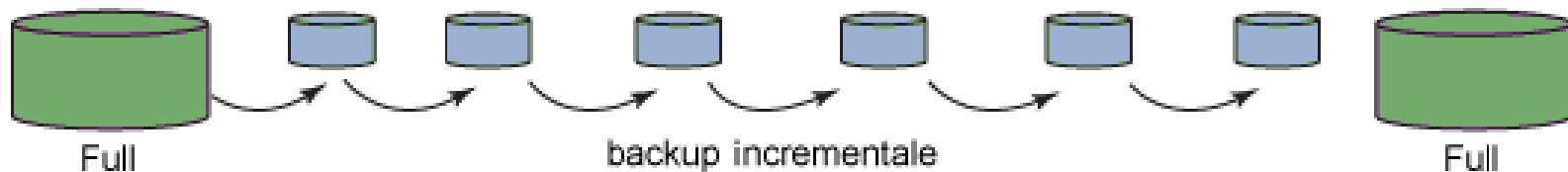
- 2. Differenziale:** Backup cumulativo di tutti i cambiamenti effettuati a partire dall'ultimo backup completo effettuato. Il vantaggio è il minor tempo necessario rispetto ad un backup completo. Lo svantaggio è che i dati da salvare aumentano per ogni giorno trascorso dall'ultimo backup.



DataBase Management System

Backup

- 3. Incrementale:** Backup che contiene tutti i files cambiati dal precedente backup (completo o incrementale). Il backup incrementale è più rapido di quello differenziale, ma richiede tempi di restore più lunghi poiché è necessario partire dall'ultimo backup completo e poi aggiungere in sequenza tutti i backup incrementali.



DataBase Management System

Backup: modalità

- ▶ Se un DB è di dimensioni limitate si farà
 - ▶ Un backup completo ogni sera
 - ▶ Ogni giorno si crea un nuovo transaction log
- ▶ Se un DB è di grandi dimensioni si farà
 - ▶ Un backup completo ogni fine settimana
 - ▶ Un backup incrementale ogni sera
 - ▶ Ogni giorno si crea un nuovo transaction log

In caso di malfunzionamento si ripristinerà il DB all'ultimo full backup, quindi all'eventuale backup differenziale della sera precedente e tramite le transaction log si ricostruiranno le operazioni eseguite nella giornata