A1

Sistemi informativi e sistemi informatici

Da sempre le attività umane si sono basate su scambi di informazioni che derivano da flussi di dati che caratterizzano specifici scenari reali. L'introduzione dell'informatica, e successivamente di Internet, ha determinato una significativa accelerazione in questo senso: non solo le aziende pubbliche e private ma anche i singoli individui sono costantemente alla ricerca di informazioni e al tempo stesso produttori di quantità sempre maggiori di dati.

1 Dati e informazione

Nella pratica quotidiana spesso si usano in modo interscambiabile i termini «dato» e «informazione»: in effetti non è immediato distinguerli.

In prima approssimazione possiamo dire che un **dato** è la misura di un fenomeno che siamo interessati a osservare.

ESEMPIO Si possono utilizzare un metro per misurare l'altezza di una persona, oppure un termometro per misurare la sua temperatura corporea. Due valori come 180 cm e 39 °C sono dati che quantificano i fenomeni osservati.

L'**informazione** è ciò che si ottiene dall'elaborazione di un insieme di dati e che accresce lo stato di conoscenza relativo a un fenomeno.

ESEMPIO Nel caso della temperatura corporea, il rilevamento di un dato che superi la soglia di 37 °C fornisce l'informazione di un'alterazione febbrile in corso.

In generale è ragionevole affermare che maggiore è la quantità di dati di cui si dispone rispetto a un fenomeno, migliore sarà l'apporto informativo rispetto a esso.

ESEMPIO Così come un medico incrocia tra loro i sintomi e i risultati delle analisi (dati) per formulare una diagnosi, allo stesso modo un analista aziendale esamina i dati di un'impresa per determinarne l'andamento economico ed, eventualmente, individuare i correttivi idonei per ottenerne una redditività adeguata.

Non sempre, osservando lo scenario di un fenomeno, tutti i dati analizzabili risultano utili alla sintesi dell'informazione.

ESEMPIO Vi sono situazioni in cui un paziente presenta sintomi che non solo non sono utili alla formulazione della diagnosi, ma che anzi rischiano di portare a commettere errori di valutazione.

Il conseguimento di informazioni utili alla comprensione e all'interazione con un fenomeno studiato discende da aspetti sia quantitativi sia qualitativi dei suoi dati: è necessario rilevare più dati possibili rispetto a esso filtrando quelli non necessari allo scopo dell'analisi.

Frequentemente, inoltre, le informazioni reali hanno una validità limitata nel tempo.

ESEMPIO L'andamento dei titoli in borsa: nel giro di pochi minuti la loro valutazione può infatti cambiare anche notevolmente.

2 Sistemi informativi e sistemi informatici

Nel tempo, per raccogliere, archiviare ed elaborare dati per gestire e comunicare informazioni, sono stati utilizzati strumenti che vanno dalle incisioni su pietra ai moderni elaboratori elettronici.

OSSERVAZIONE Il concetto di sistema informativo è indipendente dagli strumenti utilizzati per la gestione delle informazioni che esso gestisce.

Un **sistema informativo** viene utilizzato da un'organizzazione pubblica o privata per il conseguimento di specifici obiettivi; si può affermare che gli scopi per i quali è necessario disporre di adeguate informazioni sono sostanzialmente due: *operativo* e *decisionale*.

- **Scopo operativo**. Tutte le organizzazioni hanno la necessità di gestire dati funzionali relativi alle loro attività operative (infrastrutture, dipendenti, materiali, strumenti, ecc.) e da cui derivano informazioni che pertanto possiamo definire *di servizio*.

ESEMPI Un'amministrazione comunale gestisce l'anagrafe dei cittadini per fornire servizi come l'emissione di certificati; un'azienda di produzione deve provvedere alla fatturazione delle merci e alle paghe dei propri dipendenti; uno studio commerciale deve compilare le dichiarazioni dei redditi dei propri clienti, ecc.

- **Scopo decisionale**. Per poter prendere decisioni relative alle attività di programmazione, controllo e valutazione un'organizzazione deve basarsi su informazioni comunemente denominate *di governo*.

ESEMPI Un'amministrazione comunale decide di localizzare un asilo nido o un centro di assistenza per gli anziani in funzione delle fasce di età e della zona di residenza dei propri cittadini; un'azienda commerciale decide di rivedere il prezzo di ▶

un prodotto in base al costo di produzione, all'andamento delle vendite e alla situazione del mercato su cui opera, ecc.

OSSERVAZIONE Esistono informazioni che sono al tempo stesso sia di servizio sia di governo. I costi e i ricavi di un'azienda sono informazioni di servizio per la contabilità fiscale e di governo per gli organi interni preposti all'assunzione di decisioni sulle future strategie aziendali.

Un **sistema informativo** è un insieme strutturato di procedure e di risorse umane e materiali finalizzate alla raccolta, all'archiviazione, all'elaborazione e alla comunicazione di dati, allo scopo di ottenere le informazioni necessarie a un'organizzazione per gestire sia le attività operative sia quelle di governo.

OSSERVAZIONE Le risorse materiali di un sistema informativo non necessariamente sono costituite da componenti informatiche. Prima dell'avvento degli elaboratori elettronici gli archivi di banche o servizi anagrafici si sono basati per secoli solo su schedari e registri cartacei.

Un **sistema informatico** è il sottoinsieme di un sistema informativo dedicato al trattamento «automatico» di informazioni derivanti dalla gestione di dati archiviati in formato digitale.

OSSERVAZIONE La presenza di tecnologie informatiche non significa necessariamente una completa automatizzazione del sistema informativo: esistono aspetti di un sistema informativo che può non valere la pena trattare informaticamente, per questioni sia pratiche sia economiche (come, per esempio, le comunicazioni verbali tra impiegati di uno stesso ufficio).

3 Ciclo di vita di un sistema informatico

La progettazione di un sistema informatico generalmente passa attraverso un processo piuttosto complesso che deve essere condotto da personale professionalmente qualificato. Le conseguenze di un progetto gestito male possono portare a inefficienza, perdita di dati, alti costi di manutenzione, eventuali costi di riprogettazione, blocco delle attività operative.

OSSERVAZIONE Un errore tipico nella progettazione di un sistema informatico consiste nel concepirlo come una banale automazione delle procedure eseguite manualmente. L'informatizzazione di un sistema informativo deve essere un'occasione per razionalizzarne le attività, garantendone comunque la coerenza, al fine di eliminare inefficienze preesistenti e migliorarne l'efficienza e l'efficacia.

Nella pratica non c'è una metodologia di progettazione standard, ma esistono diverse tecniche, con livelli di formalizzazione differenziati, impiegate dalle aziende produttrici di software e dai professionisti del settore.

In generale la progettazione di un sistema informatico è un processo ciclico permanente durante tutto il tempo di vita del sistema stesso. Tale processo si articola su un insieme di attività raggruppabili in almeno tre macrofasi concettualmente distinte:

- **raccolta delle richieste degli utenti**;
- **progettazione concettuale**;
- **realizzazione** (**progettazione logica e fisica**).

In FIGURA **1** è rappresentato graficamente il «ciclo di vita» di un sistema informatico (gli archi tratteggiati indicano attività di verifica – *feedback* – che devono essere eseguite prima di passare alla fase successiva).

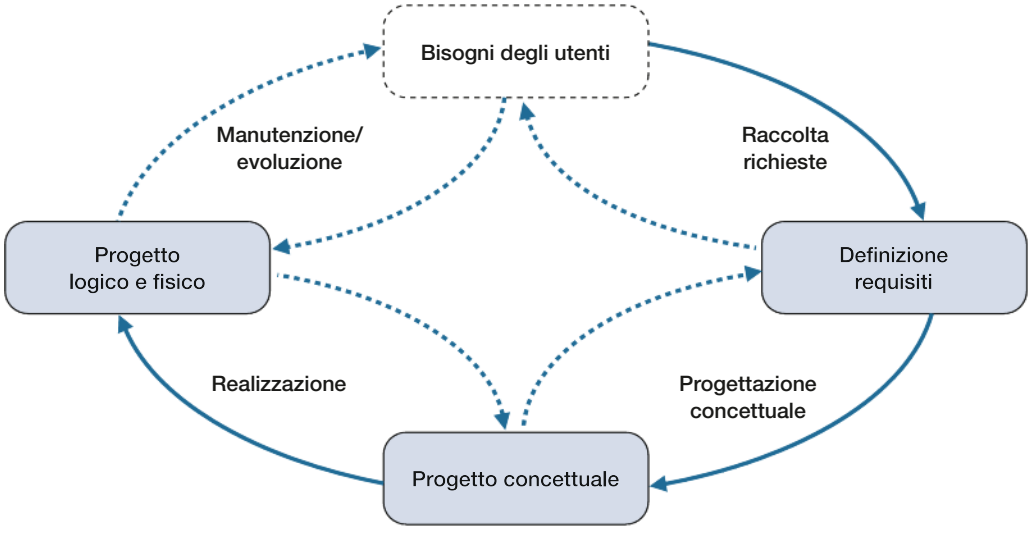


FIGURA **1**

Il processo di progettazione è ciclico perché con l'uso del sistema gli utenti e i committenti avanzano richieste correttive o evolutive del software. Inoltre si possono creare nuove esigenze per le dinamiche proprie di ogni organizzazione.

3.1 Raccolta delle richieste degli utenti

Nel corso della prima fase è necessario raccogliere tutti quegli elementi che servono a definire le caratteristiche che il sistema informatico dovrà avere per supportare adeguatamente i bisogni degli utenti: il tempo e il costo di tutto il progetto dipendono dalla qualità del lavoro svolto in questa fase. La prima fase, in generale, prevede almeno le seguenti attività.

Indagine preliminare relativa all'introduzione del sistema informatico nell'organizzazione. Tale attività dovrebbe essere svolta a cura del personale dell'organizzazione stessa al fine di:

- individuare i settori dell'organizzazione potenzialmente interessati all'introduzione delle tecnologie informatiche;
- valutare gli impatti dovuti all'introduzione delle tecnologie informatiche sull'organizzazione del lavoro (riqualificazione del personale, ecc.) e sul sistema informativo esistente;
- valutare le risorse disponibili (budget economico) per l'introduzione delle tecnologie informatiche.

Analisi del sistema informativo esistente. Quest'attività viene normalmente svolta da una specifica figura professionale, denominata **analista**, che ha lo scopo di raccogliere, catalogare e sistematizzare le conoscenze relative al sistema informativo esistente nei settori da informatizzare. Gli analisti interagiscono con gli utenti del sistema informativo, dai dirigenti al personale operativo, per rilevare i loro bisogni: informazioni necessarie, procedure in essere, tempistica delle elaborazioni, privatezza delle informazioni, possibili esigenze future, ecc. Si tratta di un'attività delicata perché condiziona il resto della progettazione.

Definizione dei requisiti del nuovo sistema. L'analista produce una documentazione dettagliata descrivendo elementi quali:

- la classificazione dei dati utilizzati (nome, tipo, uso, ecc.);
- i vincoli di integrità dei dati, cioè le condizioni che essi devono rispettare per essere significativi e validi;

ESEMPIO Un vincolo semplice può essere quello relativo ai componenti di una data (giorno, mese, anno) per essere considerata valida; una situazione più complessa può essere quella di una compagnia aerea che non accetta prenotazioni di passeggeri minorenni se non accompagnati da almeno un passeggero maggiorenne.

- la descrizione delle procedure da automatizzare che metta in evidenza per ciascuna di esse i dati coinvolti, la relazione tra dati d'ingresso e in uscita, la modalità d'interazione con l'utenza, i vincoli sui tempi di risposta del sistema, la frequenza d'uso, ecc.;
- il volume iniziale e la previsione di crescita dei dati nel tempo;
- il grado di privatezza dei dati differenziato a seconda degli utenti e del tipo di utilizzazione.

OSSERVAZIONE Rispetto alla privatezza dei dati devono essere valutati due aspetti: uno sotto il profilo della praticità e uno sotto quello della sicurezza. Circa l'aspetto pratico è opportuno che in un'organizzazione complessa i vari utenti del sistema informatico possano accedere solo ai dati di propria competenza per evitare che si possano creare situazioni confuse da «sovraccarico informativo» (*information overloading*). Per quanto riguarda l'aspetto relativo alla sicurezza, la regola generale è «non tutti devono conoscere tutto»: normalmente, per esempio, solo pochi utenti necessitano di conoscere dettagli relativi a future strategie

aziendali, o a progetti realizzativi di nuovi prodotti, per evitare potenziali situazioni di spionaggio industriale.

3.2 Progettazione concettuale

La documentazione prodotta nella fase di raccolta delle richieste costituirà l'input della successiva fase di progettazione concettuale. Questa prevede la definizione di un modello astratto del sistema informatico (progetto concettuale) come elemento di riferimento per la successiva fase di realizzazione. Molto spesso il progetto concettuale costituisce anche il tramite di verifica fra il committente e i progettisti.

I termini «astratto» e «concettuale» implicano che in questa fase si tende a definire l'organizzazione dei dati senza affrontare dettagli relativi alle tecnologie hardware e software che saranno utilizzate successivamente.

Di solito il progetto concettuale è un insieme di documenti, schemi e diagrammi che descrivono in modo organico almeno i seguenti aspetti:

- la struttura dei dati in termini di insiemi e relazioni fra insiemi;
- i vincoli di ammissibilità dei dati;
- l'integrità e la riservatezza delle informazioni.

3.3 Realizzazione (progettazione logica e fisica)

La progettazione logica e fisica consiste nella realizzazione effettiva del sistema informatico nelle varie componenti:

- **infrastrutturali**: individuazione e acquisizione delle piattaforme hardware, software e di comunicazione;
- **applicative**: acquisizione di prodotti software già disponibili sul mercato, eventualmente da personalizzare e integrare, e sviluppo di software specifico.

La centralità dei dati ha sempre caratterizzato le applicazioni informatiche, ma solo a partire dalla fine degli anni Sessanta del secolo scorso sono stati sviluppati ambienti software specificamente dedicati alla loro gestione.

In assenza di tali ambienti, ancora oggi alcuni sistemi informatici vengono realizzati con un approccio basato su *file system*, nel senso che l'archiviazione dei dati avviene mediante file memorizzati nella memoria persistente di un elaboratore mediante le funzionalità rese disponibili dal sistema operativo.

OSSERVAZIONE Un file consente la memorizzazione e la ricerca di dati, ma fornisce solo semplici meccanismi di accesso e di condivisione. Seguendo questo approccio le procedure implementate mediante un linguaggio di programmazione sono autonome: ognuna di esse, infatti, definisce e utilizza uno o più file privati ed eventuali dati di interesse per procedure distinte sono spesso replicati comportando un'inaccettabile ridondanza che può causare situazioni di incoerenza.

Linguaggi di modellizzazione

La progettazione concettuale può essere effettuata ricorrendo a un linguaggio di modellizzazione che consenta la descrizione della realtà analizzata e, al tempo stesso, la definizione del sistema informatico da realizzare per implementarla.

I diagrammi E/R (*Entity/Relationship*) sono utilizzati da molto tempo come formalismo grafico per la documentazione di strutture di dati organizzate secondo il modello «relazionale», affermatosi nel corso degli anni Settanta del secolo scorso e tuttora dominante.

Nello stesso periodo sono emersi i linguaggi grafici della famiglia IDEF (*Integration Definition*); il più noto – IDEF1X – è un formalismo di documentazione dei dati.

Successivamente si è imposto come strumento utilizzato da moltissimi analisti UML (*Unified Modeling Language*): si tratta di un insieme di diagrammi standard e di formalismi grafici per descrivere vari aspetti di un sistema informatico.

Da UML deriva SysML (*System Modeling Language*), un linguaggio di modellizzazione specifico per la descrizione dei sistemi complessi.

Infine BPMN (*Business Process Model and Notation*) è un linguaggio grafico per la descrizione e la definizione dei processi aziendali.

ESEMPIO In un'azienda l'archivio dei prodotti è gestito dall'ufficio fatturazione, che ha il compito di fatturare le merci vendute dal magazzino, che deve controllare il carico e lo scarico delle merci, e dall'ufficio vendite, che intende utilizzare i dati per proporre azioni di vendita promozionale. Se questi diversi soggetti gestissero separatamente i dati dei prodotti si avrebbero numerose duplicazioni dei dati e, a lungo andare, si potrebbero avere dati ridondanti non sincronizzati.

Fin dagli anni Ottanta del secolo scorso l'approccio basato su file system è stato gradualmente sostituito dall'approccio basato sui **Sistemi di gestione delle basi di dati** (DBMS, *DataBase Management System*), sistemi software in grado di gestire grandi collezioni di dati integrate, condivise e persistenti.

4 Aspetti intensionale ed estensionale dei dati

Come abbiamo già notato, un'informazione è l'incremento di conoscenza acquisita o dedotta dai dati mediante la loro elaborazione: da ciò deriva il fatto che i dati sono utili solo quando si dispone di una chiave interpretativa che consenta di comprenderne il significato (*semantica*), cioè i fatti che essi rappresentano. Per esempio, i dati dell'insieme schematizzato in FIGURA **2** non sono significativi fino a quando non viene specificato che cosa effettivamente rappresentano e le eventuali relazioni che li legano. In altre parole, il contenuto informativo dei dati, cioè il loro significato, non può nascere solo dai valori specifici dei dati (**aspetto estensionale dei dati**) ma anche e necessariamente dalla loro interpretazione (**aspetto intensionale dei dati**). Questo è evidente se rappresentiamo l'insieme di FIGURA **2** nella seguente forma:

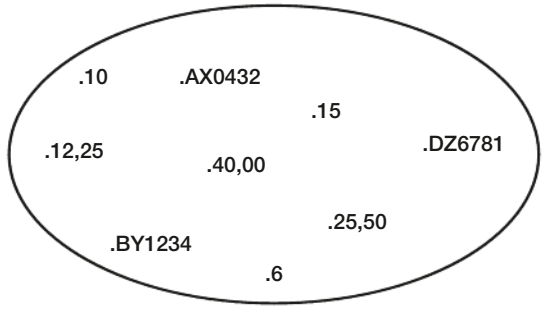


FIGURA **2**

TABELLA **1**

Codice prodotto	Prezzo	Disponibilità
AX0432	25,50	10
BY1234	12,25	15
DZ6781	40,00	6

Questa deve essere interpretata come una delle possibili **istanze** (o *estensioni*) (TABELLA **1**) del seguente **schema** (o *intensione*) (TABELLA **2**).

TABELLA **2**

Codice prodotto	Prezzo	Disponibilità

La distinzione fra **intensione** ed **estensione** è molto importante nella teoria della gestione degli insiemi di dati e a essa si farà spesso riferimento nel seguito.

Nel linguaggio naturale il significato intensionale dei dati che possono comparire in una frase viene generalmente descritto nella frase stessa.

ESEMPIO

Nella frase «la fattura numero 254 ha un importo di 125,00 euro»:

- «fattura numero» «254»;
- «importo euro» rappresenta il significato intensionale del dato «125,00»;
- il verbo «ha» stabilisce un'associazione fra i due dati «254» e «125,00».

Quando si è in presenza di gruppi consistenti di dati aventi una stessa interpretazione, è conveniente fornire l'interpretazione a livello di gruppo piuttosto che per ogni singolo dato.

ESEMPIO

Dovendo elencare gli importi di diverse fatture viene immediato organizzare una tabella di coppie del tipo (X, Y) il cui significato intensionale è «la fattura numero X ha un importo di Y euro». Essendo tale significato comune a tutte le coppie esso può essere specificato una sola volta.

OSSERVAZIONE Questa distinzione fra i dati e la loro interpretazione si ritrova in molti ambiti della vita quotidiana: gli orari degli autobus e dei treni o la classifica di una gara sportiva, così come tutte le rappresentazioni tabellari in cui viene fornita insieme ai dati stessi anche la descrizione che consente di interpretarli.

Un insieme di molti dati aventi la stessa interpretazione è uno degli elementi più ricorrenti in ambito informatico: definendo *categoria* una collezione di dati aventi la stessa interpretazione, possiamo affermare che i sistemi informatici sono caratterizzati dall'avere un numero di categorie relativamente basso e costante rispetto ai dati contenuti in ciascuna categoria. In altri termini, la dimensione delle *tabelle* è prevalente rispetto al numero delle tabelle stesse. Caratteristica, questa, che consente di progettare soluzioni semplici ed efficienti. La situazione complementare in cui si abbia un significativo numero variabile di categorie, ognuna delle quali ha pochi dati, ricade nell'ambito dei linguaggi naturali, dove per ogni dato viene fornita una specifica interpretazione.

Con il termine «linguaggio naturale» ci si riferisce, in generale, a un qualsiasi linguaggio, scritto o parlato, formatosi ed evolutosi naturalmente attraverso il suo uso continuo da parte degli esseri umani. Il linguaggio naturale è complesso, stratificato, intricato e spesso ambiguo.

OSSERVAZIONE La realizzazione di sistemi informatici che gestiscano situazioni con un numero elevato di categorie richiede tecniche e capacità elaborative tipiche dell'area dell'intelligenza artificiale (interpretazione e traduzione dei linguaggi naturali, sistemi esperti, ecc.).

5 File di dati

Un **archivio di dati**, o **file**, è un insieme di dati correlati identificato da un nome, memorizzato permanentemente su un supporto di memoria persistente di un elaboratore e avente vita indipendente dal/dai programma/i utilizzato/i per la sua creazione e/o modifica (un file creato con un programma può essere successivamente elaborato sia dallo stesso programma sia da altri).

Il modulo software di base che consente la gestione dei file è denominato *file system*: è un componente fondamentale del sistema operativo e permette di utilizzare gli archivi memorizzati sulle memorie persistenti dell'elaboratore riferendoli mediante nomi simbolici.

Il file system del sistema operativo svolge le seguenti funzioni di base:

- mantiene traccia dei file, del loro stato e della loro posizione sul supporto di memorizzazione utilizzando tabelle denominate *directory*;
- controlla, in base alle richieste effettuate dai programmi in esecuzione, le protezioni e i diritti di accesso (lettura, scrittura, ecc.);
- rende o meno disponibili i dati contenuti nei singoli file ai programmi che ne fanno richiesta.

In un file i dati sono normalmente raggruppati in unità logiche denominate **registrazioni** o **record**.

ESEMPIO Nell'elenco telefonico i dati dei vari abbonati sono costituiti da un insieme di elementi (cognome, nome, numero telefonico, indirizzo, ecc.) che costituisce la struttura del record (aspetto intensionale del dato). Ogni singolo elemento all'interno di questa struttura è noto come **campo** o ***field***. Prendendo in considerazione uno specifico abbonato, l'insieme dei dati che lo rappresentano in accordo con la struttura predefinita («Bianchi», «Giovanni», «0634162567», «Via Roma 14», ecc.) costituisce uno specifico record. L'insieme di tutti i record di un file costituisce l'aspetto estensionale dei dati.

I tipi classici di organizzazione dei file (modalità in cui vengono memorizzate ed elaborate le registrazioni) sono tre.

- **Sequenziale**. Le registrazioni vengono memorizzate in sequenza, nell'ordine d'inserimento. Per accedere a una specifica registrazione è necessario «scorrere» tutte quelle che la precedono.
- **Accesso diretto (o casuale, *random*)**. Ogni singola registrazione è individuata da un numero che ne rappresenta la posizione all'interno del file. L'accesso al file per operazioni di lettura/scrittura dati avviene direttamente specificando il numero di posizione interessata.
- **Indicizzata (o *indexed*)**. Un insieme di uno o più campi del record del file i cui valori identificano univocamente le singole registrazioni viene designato come chiave. Oltre a un file primario ad accesso diretto in cui sono memorizzate le registrazioni in ordine d'inserimento è previsto un file indice dei valori delle chiavi gestito da un modulo software dedicato

con una tecnica di albero binario di ricerca. Ogni elemento dell'indice contiene, oltre al valore di una chiave, il numero di posizione della registrazione relativa nell'archivio primario. La ricerca avviene con accesso diretto specificando il valore di una chiave. A partire dalla registrazione individuata, è poi possibile elaborare sequenzialmente quelle precedenti/successive seguendo l'ordinamento precostituito.

6 Basi di dati e sistemi di gestione delle basi di dati

Alla fine degli anni Sessanta del secolo scorso furono introdotti i primi DBMS: da allora questo settore dell'informatica ha conosciuto uno sviluppo costante, sia dal punto di vista della diffusione, interessando aree applicative sempre più vaste, sia teorico, dando luogo a proposte di sistemi anche molto diversi tra loro per gli aspetti logici e le capacità operative.

Un **DBMS** è un sistema software in grado di gestire grandi collezioni di dati integrate, condivise e persistenti assicurando loro affidabilità e privatezza.

Un **database** (o **base di dati**) è una collezione di dati gestita da un DBMS (un DBMS può gestire diverse basi di dati distinte).

Schematicamente si può affermare che gli scopi che hanno portato allo sviluppo di questi sistemi siano fondamentalmente i seguenti:

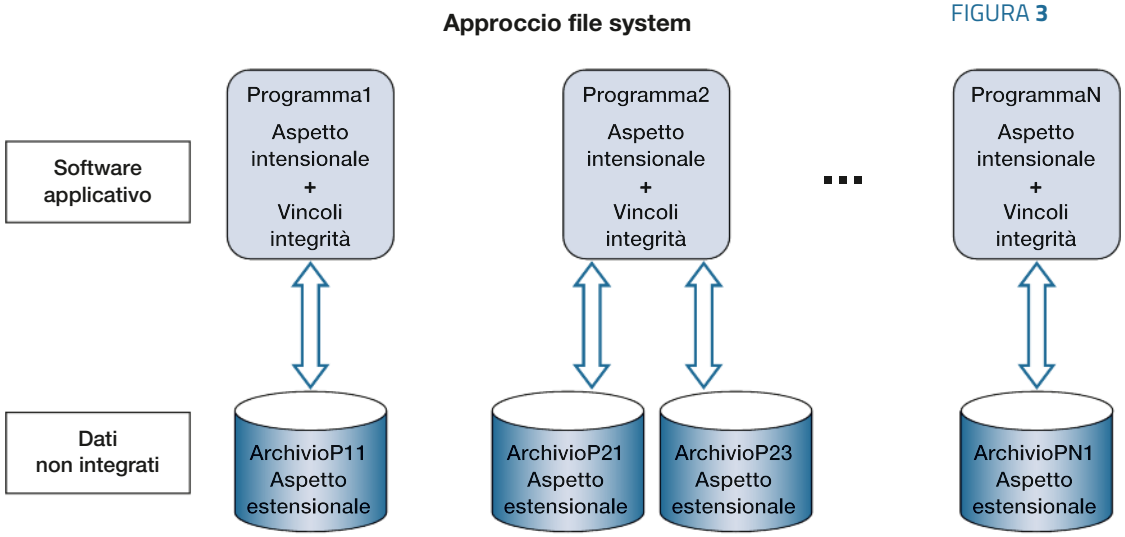
- rendere possibile una ricca definizione degli aspetti strutturali dei dati;
- rendere possibili convenienti interazioni tra gli aspetti dinamici relativi all'elaborazione dei dati e gli aspetti statici relativi alla loro struttura.

Questi due aspetti, che sono alla base delle differenze tra l'approccio basato su file system e quello fondato su DBMS, sono illustrati nelle FIGURE **3** e **4** (a pagina seguente).

DBMS

I DBMS più diffusi sono dei RDBMS (*Relational Data Base Management System*) in quanto adottano una tecnologia di modellazione dei dati definita «relazionale»: Oracle DB, Microsoft SQL-server, IBM DB2, Postegre-SQL, MySQL e il suo fork MariaDB.
Gli RDBMS sono designati come DBMS SQL dal nome del linguaggio di interrogazione standard da essi utilizzato.
Altri tipi di DBMS detti, per contrapposizione ai precedenti, NoSQL sono disponibili in una varietà di tipologie con diversi modelli di dati differenti da quello relazionale. I tipi principali sono quelli orientati ai documenti, chiave-valore, wide-column e a grafo. Essi forniscono schemi flessibili adatti a gestire grandi quantità di dati non rigidamente strutturabili, facilmente scalabili e in grado di far fronte a carichi elevati di utenza.
Appartengono a questa categoria prodotti come: MongoDB, DynamoDB, DocumentDB, Neptune, Cassandra, ecc.

FIGURA **3**



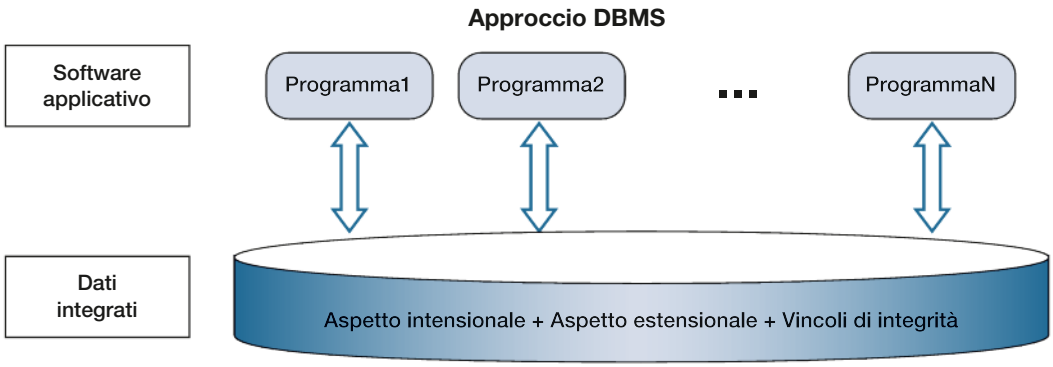


FIGURA **4**

Nel superato approccio basato su file system ogni programma dispone dei propri file di dati anche se alcuni di essi sono condivisi allo scopo di ridurre la ridondanza dei dati: il problema di fondo è che l'aspetto intensionale dei dati e le regole di integrità a cui devono soddisfare sono implementati nel codice dei programmi che gestiscono i file.

OSSERVAZIONE Se si ha la necessità di modificare la struttura dei dati (per esempio semplicemente per aggiungere nuovi campi) o le regole di integrità, è necessario modificare tutti i programmi che sono coinvolti nella loro elaborazione.

Nell'approccio fondato su DBMS i dati sono memorizzati una sola volta in modo integrato: il database contiene sia l'aspetto estensionale sia quello intensionale dei dati, compresi i vincoli di integrità. I programmi non prevedono al loro interno la definizione della struttura dei dati, ma fanno riferimento ai soli nomi dei campi che devono elaborare: in questo modo è possibile modificare la struttura dei dati senza che questo implichi modifiche ai programmi applicativi (o, eventualmente, solo a un numero limitato di questi).

OSSERVAZIONE Nell'approccio basato su file system il significato dei dati e delle relazioni che «legano» i vari file tra di loro (per esempio i clienti e le relative fatture, oppure i docenti e i loro studenti) sono immersi – e quindi «nascosti» – nel codice delle procedure di gestione, rendendo difficile l'interpretazione della struttura informativa dello scenario a cui si riferiscono.
Nei DBMS questi aspetti sono invece esplicitati a priori come parte integrante della struttura dei dati che modella una certa realtà, in modo indipendente dalle procedure di elaborazione.

I principali punti che qualificano il ricorso all'approccio fondato su DBMS sono i seguenti.

- **Integrazione**. Una base di dati è un insieme integrato di dati strutturati e permanenti memorizzati senza ridondanze superflue e organizzati in modo tale da poter essere usati da applicazioni diverse senza dipendere da alcuna di esse.
- **Indipendenza logica**. I dati sono definiti indipendentemente dalle

Approccio fondato su DBMS con file system

La diffusione universale dell'approccio fondato su DBMS alla gestione dei dati da parte di applicazioni software ha portato a soluzioni che ne fanno uso anche in assenza di un DBMS vero e proprio.
SQL lite, per esempio, è una libreria software che può essere inclusa in programmi scritti in linguaggio C/C++ o PHP e che consente di «vedere» i dati contenuti in un unico file come se fossero gestiti da un DBMS vero e proprio.

procedure che li gestiscono: in questo modo la struttura logica della base di dati può essere ampliata senza la necessità di modificare i programmi applicativi.

- **Indipendenza fisica**. Descrive la struttura dei dati astraendo da quella che è la loro implementazione fisica (organizzazione della memorizzazione, modalità di accesso, ecc.) in modo tale che si possa modificare quest'ultima senza modificare la struttura logica dei dati e, di conseguenza, i programmi applicativi.
- **Integrità**. È il DBMS, e non le procedure di gestione, a dover prevedere meccanismi per controllare che i dati inseriti o modificati soddisfino ai vincoli di integrità specificati.

È possibile classificare l'utenza di un sistema di gestione della basi di dati nel seguente modo.

- **Programmatori di applicazioni**. Utenti professionali che hanno il compito di sviluppare applicazioni software che si interfacciano con una base di dati mediando l'accesso ai dati per gli utenti finali.
- **Utenti finali**. Utenti non professionali che interagiscono con la base di dati esclusivamente mediante programmi applicativi realizzati per svolgere determinate attività operative di gestione ed elaborazione dei dati nell'ambito di un'organizzazione (per esempio: addetti a sportelli postali o bancari, magazzinieri, utenti web, ecc.) senza conoscere la struttura dei dati con cui interagiscono.
- **Utenti avanzati**. Utenti che conoscono la struttura dei dati e sono in grado di operare attività di indagine utilizzando un linguaggio generico di interrogazione della base di dati senza alterarla.
- **Utenti amministratori (DBA, *DataBase Administrator*)**. Utenti professionali a cui è demandata la manutenzione della base di dati nel tempo: l'amministratore, utilizzando gli strumenti di amministrazione resi disponibili dal sistema DBMS, è l'unico soggetto che può, in funzione delle necessità, modificare la struttura della base di dati (aspetto intensionale), definire o modificare i diritti di accesso ai dati per ogni singolo utente del DBMS, ecc.

L'interazione con un database è resa possibile attraverso l'uso di specifici linguaggi offerti dal DBMS:

- **DDL** (*Data Definition Language*): linguaggi per la definizione della struttura dei dati; sono usati principalmente dall'amministratore del sistema e sono costituiti da comandi che consentono di definire e modificare la struttura della base di dati e dei vincoli di integrità;
- **DML** (*Data Manipulation Language*): linguaggi per la gestione e l'utilizzazione dei dati contenuti in una base di dati; sono costituiti da comandi che permettono di accedere ai dati per effettuare operazioni di interrogazione o manipolazione (inserimento, eliminazione, modifica).

  In generale, i linguaggi DML possono essere utilizzati in modalità:

  – **ospitata**: vengono «immersi» in un linguaggio di programmazione e

Modello dei dati

Si può affermare che la teoria delle basi di dati come disciplina informatica è nata con la nozione di modello dei dati.
Intuitivamente un modello dei dati è uno strumento concettuale non immediato da definire e che assume sfumature diverse a seconda degli autori.
Il termine **modello dei dati** fu introdotto alla fine degli anni Sessanta del secolo scorso quando fu evidente che esistevano diversi modelli cui ispirarsi per esprimere la semantica dei dati: più precisamente il termine fu proposto nel 1970 da Edgar Codd, al tempo ricercatore dell'IBM, in coincidenza con la presentazione del modello relazionale dei dati (a cui è dedicata un'ampia parte in questo libro) che è divenuto in seguito dominante e che consente al progettista di attribuire un certo significato (o interpretazione) ai dati.
Il significato viene attribuito principalmente assegnando una struttura ai dati mediante specifici meccanismi di strutturazione previsti dal modello stesso: infatti, come abbiamo già osservato, i dati di per sé non producono informazione in assenza di un'interpretazione.
Possiamo quindi dire che un modello dei dati è uno strumento concettuale mediante il quale si può acquisire conoscenza da un insieme di dati altrimenti insignificanti.

utilizzati dagli sviluppatori software per codificare le applicazioni che interagiscono con le basi di dati di un DBMS;
- **autonoma**: vengono utilizzati costrutti e/o strumenti atti a consentire un accesso prevalentemente interattivo ai dati.

L'approccio fondato su DBMS prevede che la fase di realizzazione possa essere scomposta in due sottofasi distinte:

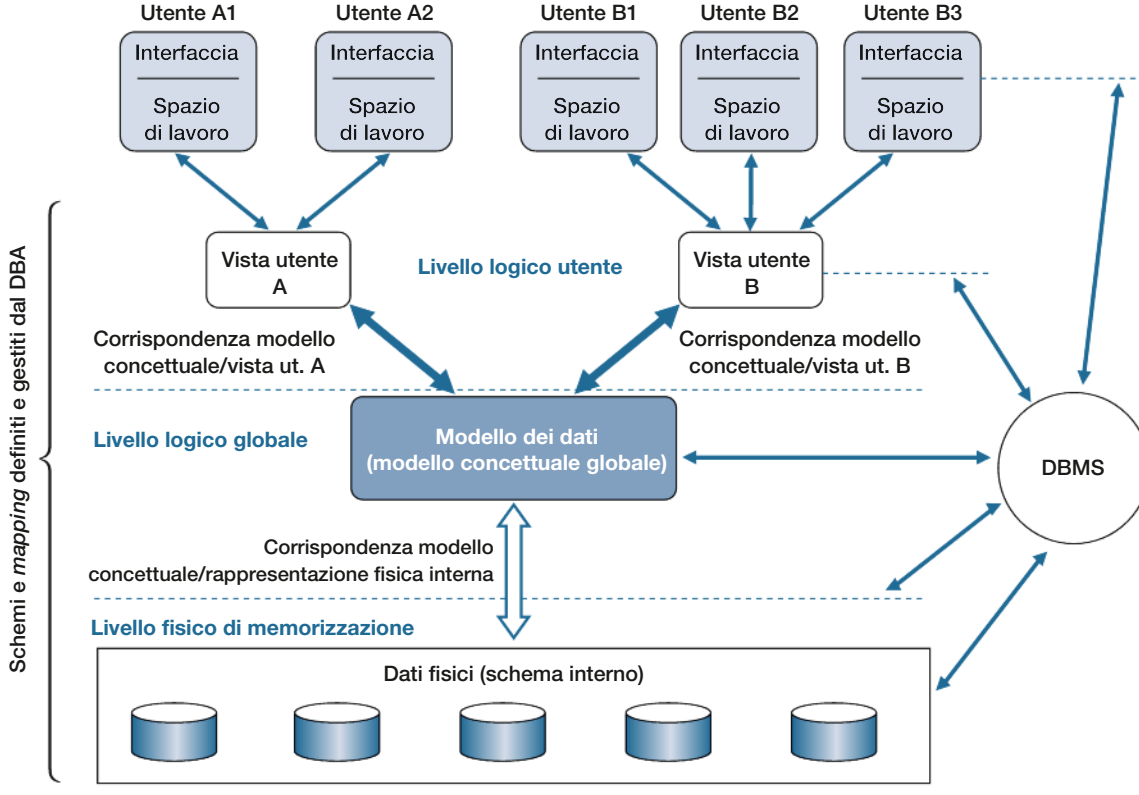
- **progettazione logica**: consiste nella conversione del progetto concettuale in un «progetto logico», cioè nella strutturazione dei dati e nella formalizzazione delle procedure applicative;
- **progettazione fisica**: consiste nella descrizione dell'organizzazione fisica dei dati che realizzano la base di dati del sistema informatico utilizzando il linguaggio DDL e gli operatori previsti a tale scopo dal DBMS.

OSSERVAZIONE La definizione del dominio dei dati dei campi è uno degli obiettivi della progettazione logica; aspetti della progettazione fisica sono invece la definizione delle strutture di memorizzazione (i file fisici, il loro tipo, ecc.), gli indici (per velocizzare l'accesso ai dati), le partizioni, gli shard, ecc...

7 Architettura logica di un sistema di gestione delle basi di dati

L'architettura di un sistema di gestione di database può essere schematizzata come in FIGURA **5**.

FIGURA **5**



Seguendo un modello ormai classico (ANSI-SPARC) essa è strutturata su tre livelli.

- **Livello logico utente**. Ogni utente ha un proprio spazio di lavoro (cioè un'area di input/output che permette di ricevere/trasmettere i dati della base di dati nelle operazioni di lettura/scrittura) che si interfaccia con la base di dati mediante un'applicazione o direttamente tramite il linguaggio DML. A questo livello sono definite dall'amministratore utilizzando il linguaggio DDL le *viste utente*, ovvero dei sottoinsiemi del modello logico globale dell'intera base di dati: le viste utente realizzano i meccanismi fondamentali della privatezza dei dati.
- **Livello logico globale**. È relativo alla definizione della struttura logica generale (aspetto intensionale dei dati, vincoli di integrità, ecc.) della base di dati: viene definito dall'amministratore utilizzando il linguaggio DDL.
- **Livello fisico di memorizzazione**. È il livello più basso dove si trovano i dati fisici e pertanto relativo ai dispositivi di memorizzazione, all'organizzazione fisica dei dati e alle relative modalità di accesso.

OSSERVAZIONE Nello schema i collegamenti tra i livelli sono realizzati dal DBMS che gestisce l'intera struttura mediante tecniche di corrispondenza (mapping), cioè procedure software che associano gli elementi di un livello ai corrispondenti elementi di quello contiguo.

Riguardo allo sviluppo di software per la gestione di dati con DBMS (nello specifico quelli basati sul modello di dati relazionale o RDBMS), un settore interessante è quello relativo ai prodotti *Object-Relational Mapping* (ORM). Questi introducono una tecnica di programmazione per l'integrazione di applicazioni software orientate agli oggetti con sistemi RDBMS astraendo dalle caratteristiche specifiche di questi ultimi. Gli ORM permettono la gestione dei servizi relativi alla persistenza degli oggetti realizzandone il mapping con i corrispondenti insiemi di dati nei database (FIGURA **6**).

Esempi di ambienti ORM sono: Hibernate, Doctrine, Propel, ecc.

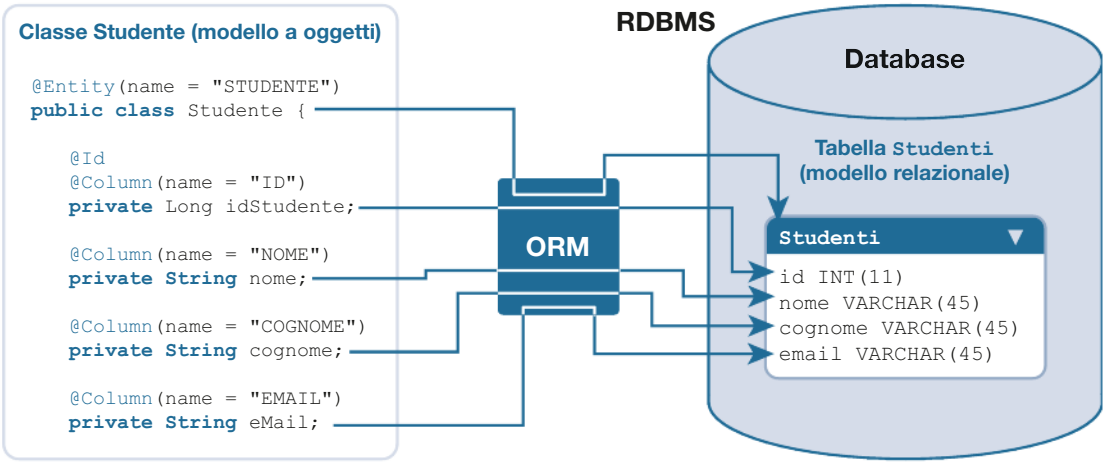


FIGURA **6**

I CONCETTI CHIAVE

DATO. Misura del fenomeno che siamo interessati a osservare.

INFORMAZIONE. Si ottiene dall'elaborazione di un insieme di dati; è in grado di accrescere lo stato di conoscenza di un fenomeno. La qualità dell'informazione è legata agli aspetti qualitativo e quantitativo dei dati elaborati e ha una validità limitata nel tempo.

SISTEMA INFORMATIVO. Insieme strutturato di procedure e di risorse umane e materiali finalizzate alla raccolta, all'archiviazione, all'elaborazione e alla comunicazione di dati, allo scopo di ottenere le informazioni necessarie a un'organizzazione per gestire sia le attività operative sia quelle di governo.

SISTEMA INFORMATICO. È il sottoinsieme di un sistema informativo dedicato al trattamento «automatico» di informazioni derivanti dalla gestione di dati archiviati in formato digitale.

CICLO DI VITA DI UN SISTEMA INFORMATICO. Processo ciclico permanentemente in vita relativo alla progettazione di un sistema informatico; normalmente prevede una serie di attività raggruppabili in almeno tre macrofasi concettualmente distinte: *raccolta delle richieste degli utenti*, *progettazione concettuale* e *realizzazione* (*progettazione logica e fisica*). Le tre fasi producono rispettivamente come output: la definizione dei requisiti a cui il sistema informatico dovrà essere conforme, il progetto concettuale e il progetto logico e fisico che rappresenta l'insieme delle componenti software che implementano il sistema informatico stesso.

APPROCCIO BASATO SU FILE SYSTEM. È l'approccio obsoleto alla gestione dei dati in un sistema informatico: prevede il ricorso a linguaggi di programmazione per la gestione di file di dati basati sulle organizzazioni rese disponibili dal file system del sistema operativo.

APPROCCIO FONDATO SU DBMS. È l'approccio attualmente più utilizzato per la gestione dei dati in un sistema informatico: prevede l'impiego di DBMS (*DataBase Management System*), sistemi software in grado di gestire grandi collezioni di dati integrate, condivise e persistenti.

ASPETTI INTENSIONALE ED ESTENSIONALE DEI DATI. L'aspetto intensionale dei dati è relativo alle informazioni necessarie per la loro corretta interpretazione. Nel caso di un file l'aspetto intensionale è dato dalla struttura del record. L'aspetto estensionale dei dati è invece relativo al valore dei dati stessi: in un file fa quindi riferimento al suo contenuto.

FILE (ARCHIVIO). Un file di dati (o archivio) è un insieme di dati correlati identificato da un nome, memorizzato permanentemente su un supporto di memoria persistente e avente vita indipendente dal/dai programma/i utilizzato/i per la sua creazione e/o modifica.

RECORD (REGISTRAZIONE). Unità logiche in cui i dati sono raggruppati in un file.

ORGANIZZAZIONE DI UN FILE. Quest'espressione indica sia il modo in cui il file è memorizzato sul supporto fisico della memoria persistente, sia il modo in cui il contenuto del file viene elaborato. Le organizzazioni classiche sono: *sequenziale*, in cui le registrazioni vengono memorizzate in sequenza, nell'ordine di inserimento; *ad accesso diretto* (*o casuale*, *random*), ogni singola registrazione è individuata da un numero che ne rappresenta la posizione all'interno del file; *indicizzata* (o *indexed*), in base alla quale un insieme di uno o più campi del record del file i cui valori identificano univocamente le singole registrazioni viene designato come chiave. Oltre a un file primario ad accesso diretto in cui sono memorizzate le registrazioni in ordine di inserimento è previsto un file indice dei valori delle chiavi.

MODALITÀ DI ACCESSO A UN FILE. Questa espressione indica la modalità con cui l'organizzazione del file consente l'accesso ai singoli record di dati. L'organizzazione sequenziale consente esclusivamente l'accesso sequenziale; l'organizzazione casuale (random) consente sia l'accesso diretto, specificando il numero di posizione; l'organizzazione indicizzata (indexed) consente sia l'accesso diretto specificando il valore di una chiave, sia l'accesso sequenziale seguendo l'ordinamento precostituito.

DBMS (*DataBase Management System*). Sistema software in grado di gestire grandi collezioni di dati integrate, condivise e persistenti assicurando loro affidabilità e privatezza.

BASE DI DATI (DATABASE). È una collezione di dati gestita da un DBMS (un DBMS può gestire diverse basi di dati).

INTEGRAZIONE DEI DATI. Una base di dati è un insieme integrato di dati strutturati e permanenti memorizzati senza ridondanze superflue e organizzati in modo tale da poter essere usati da applicazioni diverse senza dipendere da alcuna di esse.

INDIPENDENZA LOGICA E FISICA DEI DATI. In un database il fatto che i dati siano descritti indipendentemente dalle procedure di gestione comporta che la struttura logica della base di dati può essere ampliata

senza dover modificare i programmi applicativi (*indipendenza logica*). Inoltre, la possibilità di descrivere la struttura dei dati astraendo da quella che è la loro rappresentazione fisica (organizzazione di memorizzazione, modalità di accesso, ecc.) permette la loro indipendenza fisica, ovvero la possibilità di modificare i supporti di memorizzazione senza dover modificare la struttura logica dei dati.

INTEGRITÀ DEI DATI. In un DBMS è il sistema stesso, e non le procedure gestionali, a dover prevedere meccanismi per controllare che i dati inseriti o modificati soddisfino ai vincoli di integrità specificati.

AMMINISTRATORE (DBA, *DataBase Administrator*). È la figura professionale a cui è demandata la manutenzione della base di dati nel tempo. L'amministratore, utilizzando specifici strumenti di amministrazione resi disponibili dal DBMS, è l'unico che può, in funzione delle necessità, modificare la struttura della base di dati (aspetto intensionale), definire o modificare i diritti di accesso ai dati per ogni utente del DBMS, ecc.

DDL (*Data Definition Language*). Linguaggio per la definizione della struttura dei dati usato dal DBA; è costituito da comandi che permettono di definire e modificare la struttura della base di dati e dei vincoli di integrità.

DML (*Data Manipulation Language*). Linguaggio per la gestione e l'uso dei dati; è costituito da comandi che permettono di accedere ai dati per effettuare operazioni di interrogazione o manipolazione. Un linguaggio DML solitamente può essere utilizzato in modalità: *ospitata*, quando viene «immerso» in un linguaggio di programmazione e utilizzato per codificare le applicazioni che interagiscono con le basi di dati di un DBMS; oppure *autonoma*, se utilizzato sfruttando costrutti e/o strumenti atti a consentire un accesso prevalentemente interattivo ai dati.

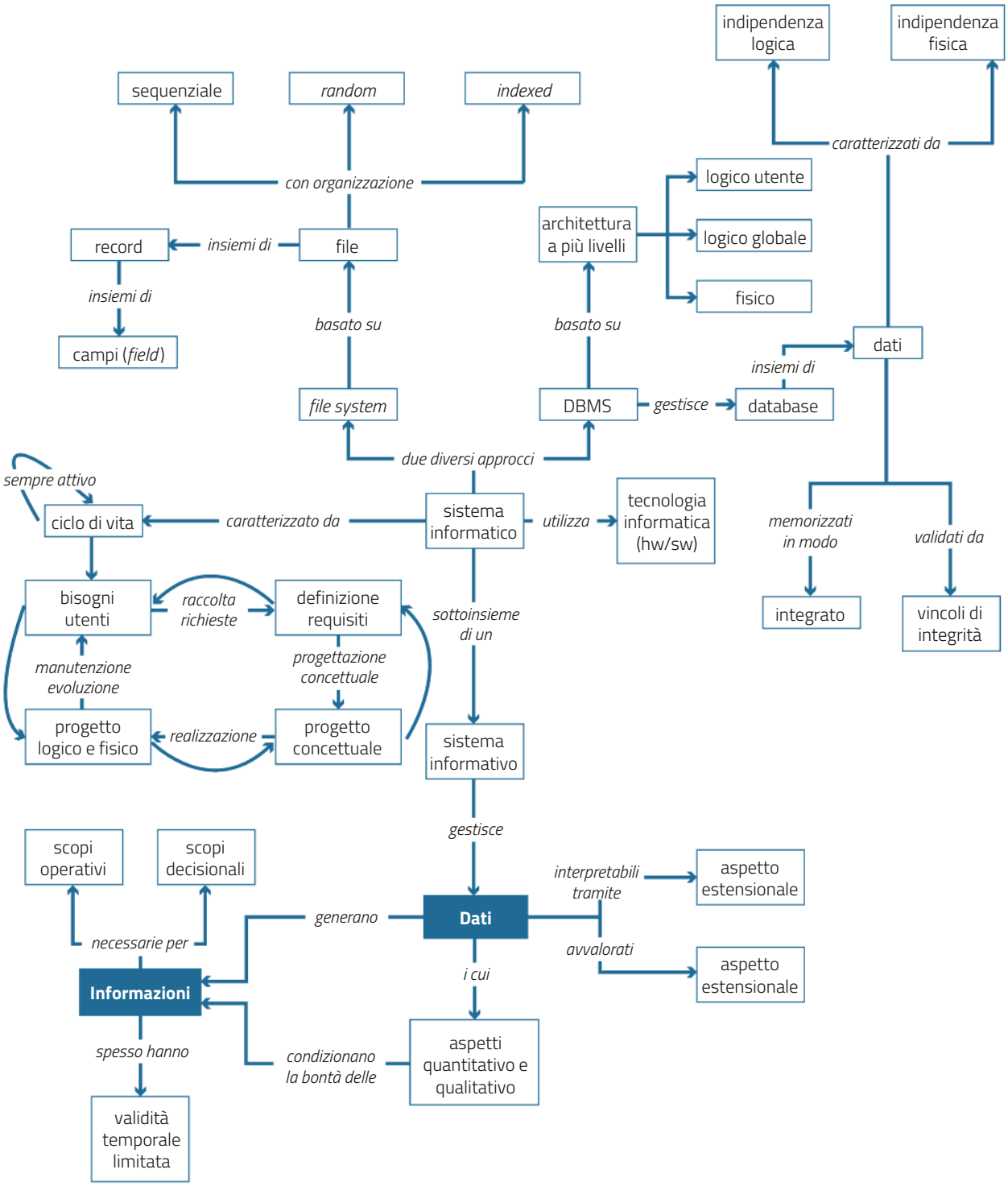
ARCHITETTURA LOGICA DI UN SISTEMA DI GESTIONE DELLE BASI DI DATI. Seguendo un modello classico essa è strutturata su tre livelli: *livello logico utente*, *livello logico globale* e *livello fisico di memorizzazione*.

LIVELLO LOGICO UTENTE. Ogni utente ha un proprio spazio di lavoro (cioè un'area di input/output che permette di ricevere/trasmettere i dati della base di dati nelle operazioni di lettura/scrittura) che si interfaccia con la base di dati mediante un'applicazione o direttamente tramite il linguaggio DML. A questo livello sono definite dal DBA utilizzando il linguaggio DDL le *viste utente*, ovvero dei sottoinsiemi del modello logico globale dell'intera base di dati: le viste utente realizzano i meccanismi fondamentali della privatezza dei dati.

LIVELLO LOGICO GLOBALE. È relativo alla definizione della struttura logica generale (aspetto intensionale dei dati, vincoli di integrità, ecc.) della base di dati: viene definito dal DBA utilizzando il linguaggio DDL.

LIVELLO FISICO DI MEMORIZZAZIONE. È il livello più basso dove si trovano i dati fisici e pertanto relativo ai dispositivi di memorizzazione, all'organizzazione fisica dei dati e alle relative modalità di accesso.

RIPASSA CON LA MAPPA



QUESITI

1 **Un dato è...**

A l'interpretazione di un fenomeno che siamo interessati a osservare.
B la misura di un fenomeno che siamo interessati a osservare.
C la stessa cosa di un'informazione.
D una componente variabile di un sistema informativo.

2 **Un'informazione è...**

A la stessa cosa di un dato.
B la misura di un fenomeno che siamo interessati a osservare.
C ciò che si ottiene dall'elaborazione di un insieme di dati e che accresce lo stato di conoscenza relativo a un fenomeno che siamo interessati a osservare.
D una componente variabile di un sistema informativo.

3 **Un sistema informativo è...**

A un sottoinsieme di un sistema informatico.
B un insieme strutturato di procedure e di risorse umane e materiali finalizzate, tramite uso di tecnologia informatica, alla raccolta, all'archiviazione, all'elaborazione e alla comunicazione di dati allo scopo di ottenere le informazioni necessarie per gestire sia le attività operative sia quelle di governo di un'azienda.
C un insieme strutturato di procedure e di risorse umane e materiali finalizzate alla raccolta, all'archiviazione, all'elaborazione e alla comunicazione di dati allo scopo di ottenere le informazioni necessarie per gestire sia le attività operative sia quelle di governo di un'azienda.
D un insieme di prodotti software.

4 **Quali dei seguenti sono scopi fondamentali di un sistema informativo aziendale?**

A Sviluppo delle componenti software.
B Operatività dell'azienda.
C Raccolta e catalogazione dei dati.
D Scelte strategiche aziendali.

5 **Quali delle seguenti sono macrofasi del ciclo di vita di un sistema informatico?**

A Raccolta delle richieste degli utenti.
B Acquisto delle componenti hardware.
C Progettazione concettuale.
D Acquisto delle componenti software.

6 **Qual è il compito dell'analista?**

A Effettuare la manutenzione di un sistema informatico nel tempo.
B Supervisionare al funzionamento di un sistema informatico.
C Sviluppare specifici moduli software di un sistema informatico.
D Studiare le caratteristiche di un sistema informativo in modo da poter definire le caratteristiche di un sistema informatico di supporto.

7 **Quali dei seguenti sono elementi che caratterizzano il progetto concettuale di un sistema informatico?**

A Linguaggio/i da utilizzare per sviluppare il software.
B La struttura dei dati in termini di insiemi e relazioni fra insiemi.
C Caratteristiche dell'hardware da utilizzare.
D I vincoli di ammissibilità dei dati.

8 **Quali delle seguenti affermazioni relative all'approccio file system sono vere?**

A. La struttura dei dati viene definita internamente ai programmi. V F
B. È possibile ampliare la struttura dei dati senza dover modificare i programmi preesistenti. V F
C. La ridondanza dei dati è ridotta al minimo indispensabile. V F
D. I vincoli di validità dei dati sono memorizzati insieme alla struttura dei dati. V F

9 **L'aspetto intensionale dei dati è relativo...**

A alle caratteristiche dei programmi che li gestiscono.
B a elementi che servono ad attribuire una corretta interpretazione del significato dei dati.
C al valore stesso dei dati.
D a categorie di insiemi secondo le quali possono essere catalogati.

10 **Quali dei seguenti non sono punti fondamentali di un progetto concettuale?**

A I vincoli di ammissibilità dei dati.
B Il dimensionamento dell'hardware.
C L'integrità e la riservatezza delle informazioni.
D Il tipo di DBMS utilizzato.

11 **Quale delle seguenti sequenze è logicamente corretta in senso gerarchicamente crescente?**

A record, file, field.
B record, field, file.
C file, field, record.
D field, record, file.

12 **Quali dei seguenti sono aspetti fondamentali dell'approccio DBMS?**

A La scelta del linguaggio di programmazione con cui sviluppare gli applicativi.
B L'integrazione dei dati.
C La scelta dei tipi di file da utilizzare.
D L'integrità dei dati.

13 **Quali delle seguenti informazioni sono vere rispetto all'approccio DBMS?**

A. I vincoli di ammissibilità dei dati sono specificati attraverso il codice delle procedure applicative. V F
B. Il codice delle procedure applicative determina il dimensionamento hardware. V F
C. La struttura dei dati viene specificata esternamente alle procedure applicative. V F
D. La struttura dei vincoli di integrità dei dati viene specificata e memorizzata insieme alla loro struttura. V F

14 **I due approcci file system e DBMS nella realizzazione di un sistema informatico sono due tecniche...**

A sostanzialmente identiche nella gestione dei dati.
B diverse per la gestione dei dati di cui la prima è un'evoluzione della seconda.
C diverse per la definizione degli aspetti statici dei dati, la loro integrazione e i vincoli che questi debbono rispettare per essere validi.
D distinte di sviluppo delle componenti software del sistema.

15 **Una base di dati è...**

A una base su cui definire una struttura dati.
B una grande collezione di dati integrati, condivisi e persistenti.
C una grande collezione di dati non necessariamente integrati, condivisi o persistenti.
D un insieme minimo di dati su cui lavorare per ottenere informazioni.

16 **Un DBMS è...**

A l'equivalente di una base di dati.
B un ambiente software che permette di definire e gestire basi di dati.
C uno qualsiasi dei programmi applicativi che elaborano una base di dati.
D l'equivalente di un sistema informativo.

17 **I due termini integrità e integrazione riferiti a una base di dati...**

A sono equivalenti.
B il primo si riferisce al fatto che i dati sono memorizzati senza ridondanze superflue, mentre il secondo ai vincoli cui i dati devono soddisfare per essere validi.
C il primo si riferisce ai vincoli cui i dati debbono soddisfare per essere validi, mentre il secondo al fatto che i dati sono memorizzati senza ridondanze superflue.
D il primo si riferisce ai vincoli cui i dati debbono soddisfare per essere validi, mentre il secondo alla possibilità costante di integrare la base di dati con nuovi elementi.

18 **I due termini DDL e DML riferiti a un DBMS...**

A sono equivalenti e si riferiscono al personale addetto alla manutenzione del sistema in oggetto.
B il primo indica un linguaggio per la definizione della struttura dei dati di un database, mentre il secondo si riferisce a un linguaggio per l'uso dei dati in esso contenuti.
C il primo indica un linguaggio per l'uso dei dati contenuti in un database, mentre il secondo si riferisce a un linguaggio per la definizione della sua struttura dei dati.
D sono equivalenti e fanno riferimento a insiemi di comandi per interrogare un database.

19 **Relativamente a un database, con il termine indipendenza fisica dei dati si intende la possibilità di...**

A descrivere la struttura dei dati astraendo da quella che è la loro implementazione fisica per definire vari livelli di privatezza dei dati.

B ampliare la struttura logica della base di dati senza la necessità di modificare i programmi applicativi.

C descrivere la struttura dei dati astraendo da quella che è la loro implementazione fisica (organizzazione della memorizzazione, modalità di accesso, ecc.) in modo tale che si possa modificare quest'ultima senza modificare la struttura logica dei dati e, di conseguenza, i programmi applicativi.

D memorizzare i dati liberamente senza nessun vincolo.

20 **Indicare quale delle sequenze riportate di seguito indica l'ordine corretto dei livelli dell'architettura logica di una base di dati a partire dal livello più esterno.**

A Livello logico utente, modello concettuale globale, livello fisico di memorizzazione.

B Livello fisico di memorizzazione, livello logico utente, modello concettuale globale.

C Livello fisico di memorizzazione, modello concettuale globale, livello logico utente.

D Modello concettuale globale, livello logico utente, livello fisico di memorizzazione.

to accrue
accumulare, maturare

to arise
presentarsi/sorgere

asset
bene, attività

to carry out
effettuare

corollary
corollario

to devote
dedicare

elsewhere
altrove

fairly
abbastanza

the foregoing
la cosa precedente

to point out
evidenziare

1.2 Why database?

Why should an enterprise choose to store its operational data in an integrated database? There are many answers to this question. One general answer is that it provides the enterprise with *centralized control* of its operational data [...] is its most valuable asset. This is in sharp contrast to the situation [...], where typically each application has its own private files [...] too-so that the operational data is widely dispersed, and there is little or no attempt to control it in a systematic way.

The foregoing implies that in an enterprise with a database system there will be some one identifiable person – the *database administrator*, or DBA – who has this central responsibility for the operational data. In fact, we may consider the DBA as part of the database system (that is, the system is more than just data plus software plus hardware – it includes people, too). We shall be discussing the role of the DBA in more detail later; for the time being, it is sufficient to note that the job will involve both a high degree of technical expertise and the ability to understand and interpret management requirements at a senior level. (In practice the DBA may consist of a team of people instead of just one person.) It is important to realize that the position of the DBA within the enterprise is a very senior one.

Let us now consider the advantages that accrue from having centralized control of the data, as discussed above.

- **The amount of redundancy in the stored data can be reduced.**

In most current systems each application has its own private files. This can often lead to considerable redundancy in stored data, with resultant waste in storage space. For example, a personnel application and an education-records application may each own a file containing a name, number, and department for every employee. With central control, however, the DBA can identify the fact that the two applications require essentially the same data, and hence can integrate the two files; that is, the data can be stored once only and can be shared by the two applications.

- **Problems of inconsistency in the stored data can be avoided (to a certain extent).**

This is really a corollary of the previous point. If the "same" fact about the real world – say, the fact that employee E3 works in department D8 – is represented by two distinct entries in the database, then at some time the two entries will not agree (i.e., when one and only one has been updated). The database is then inconsistent. If, on the other hand, the information is represented by a single entry – if the redundancy has been removed – such an inconsistency cannot arise.

- **The stored data can be shared.**

This point has already been made in passing but is worth stating here as an important advantage in its own right. It means not only that all the files of existing applications are integrated, but also that new applications may be developed to operate against the existing database.

- **Standards can be enforced.**

With central control of the database, the DBA can ensure that installation and industry standards are followed in the representation of the data. This simplifies problems of maintenance and data interchange between installations.

- **Security restrictions can be applied.**

Having complete jurisdiction over the operational data, the DBA (a) can ensure that the only means of access to the database is through the proper channels, and hence (b) can define authorization checks to be carried out whenever access to sensitive data is attempted. Different procedures can be established for each type of access (retrieve,

update, delete, etc.) to each type of data field in the database. [Perhaps it should also be pointed out that without such procedures the security of the data may actually be more at risk in a database system than in a traditional (dispersed) filing system.]

- **Data integrity can be maintained.**

The problem of integrity is the problem of ensuring that the database contains only accurate data. Inconsistency between two entries representing the same "fact" is an example of lack of integrity (which of course can only occur if redundancy exists in the stored data). Even if redundancy is eliminated, however, the database may still contain incorrect data. For example, an employee may be shown as having worked 200 hours in the week, or a list of employee numbers for a given department may include the number of a non-existent employee. Centralized control of the database helps in avoiding these situations (in so far as they can be avoided) by permitting the DBA to define validation procedures to be carried out whenever any storage operation is attempted. (Here, as elsewhere, we use the term "storage operation" to cover all the operations of updating, inserting, and deleting.)

- **Conflicting requirements can be balanced.**

Knowing the overall requirements of the enterprise – as opposed to the requirements of any individual user – the DBA can structure the database system to provide an overall service that is "best for the enterprise". For example, a representation can be chosen for the data in storage that gives fast access for the most important applications at the cost of poor performance in some other applications.

Most of the advantages listed above are fairly obvious. However, one other point, which is not so obvious – although it is implied by several of the foregoing – must be added to the list, namely, the provision of data independence. (Strictly speaking, this is an objective rather than an advantage.) This concept is so important that we devote a separate section to it.

[C.J. Date, "An Introduction to Database Systems",
Addison Wesley Publishing Company, 1979]

QUESTIONS

a What does the acronym DBA stand for?
b Briefly describe the difficulties associated with data integrity.
c What are security restrictions?
d Briefly describe the difficulties associated with data inconsistency.
e Describe the relationship between data inconsistency and data redundancy.