

Le funzionalità di un DBMS



Dario Maio
<http://bias.csr.unibo.it/maio/>

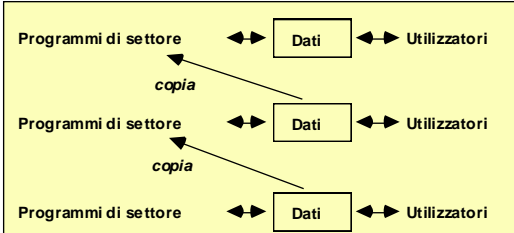
Funzionalità di un DBMS

1

Sistemi informatici settoriali

- In un'organizzazione dotata di un sistema informativo, è normalmente possibile individuare alcuni **settori** nei quali l'organizzazione stessa si articola. Ad esempio, in un'azienda manifatturiera settori tipici sono quelli della **produzione, delle vendite, degli acquisti, del personale, del magazzino**, ecc.
- **Il flusso di informazioni intra-settoriale è più intenso rispetto al flusso inter-settoriale.** Questa considerazione, unitamente ad aspetti di natura tecnologica, ha fatto sì che prima dell'avvento dei DBMS l'automatizzazione della gestione nei vari settori fosse realizzata con programmi applicativi ad hoc pensati per i dati specifici del settore.

Nei **sistemi settoriali** il flusso di dati da un settore all'altro era di solito gestito mediante creazione di **copie** elettroniche e/o cartacee.

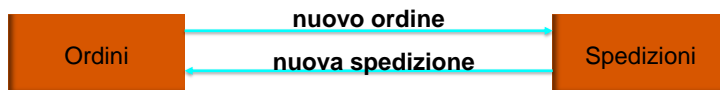


Funzionalità di un DBMS

2

Problemi dei sistemi informatici settoriali (1)

- La presenza di copie dello stesso dato dà luogo a **ridondanze**. Di conseguenza si ha un inutile **spreco di memoria**.
- Inoltre, la ridondanza (non introdotta in questo caso per scopi di protezione da guasti) può dar luogo a problemi di **inconsistenza delle copie** e, in ogni caso, comporta la **necessità di propagare le modifiche**, con ulteriore **spreco di risorse** (CPU e rete).
 - » Il settore **Ordini** di un'azienda manifatturiera archivia i propri dati in un file, **non condiviso con gli altri settori aziendali**. Ogni volta che arriva un ordine, i dati relativi devono essere trasmessi al settore **Spedizioni**, affinché l'ordine possa essere evaso. A spedizione eseguita, i dati relativi devono essere ritrasmessi al settore **Ordini**.



Funzionalità di un DBMS

3

Problemi dei sistemi informatici settoriali (2)

Settore	Evento	Dati Generati	Dati usati e non generati dal settore
Accettazione ordini	Ricezione ordine	Codice ordine Codice cliente Codice articolo Quantità ordinata Prezzo unitario Data ordine	Quantità spedita*
Spedizioni	Spedizione	Codice bolla Data spedizione Quantità spedita	Codice ordine Codice cliente Codice articolo Quantità ordinata Data ordine

❖ Si considera il caso generale in cui, per ragioni legate alle effettive giacenze in magazzino, un ordine possa non essere soddisfatto con una singola spedizione. Si rende pertanto necessario un feedback dal reparto spedizioni al reparto accettazione ordini.

Funzionalità di un DBMS

4

Problemi dei sistemi informatici settoriali (3)

- ❑ La **progettazione degli archivi** di settore è fatta sulla base di considerazioni locali. La **mancaanza di standard a livello globale** complica la gestione dei flussi inter-settoriali e può creare problemi di incompatibilità nelle rappresentazioni adottate.
- ❑ I dati sono soggetti a diversi **vincoli di integrità**, che riflettono la conoscenza della realtà specifica rappresentata. Se i vincoli vengono a dipendere solo da specifiche considerazioni (**viste**) settoriali si possono generare inconsistenze.

Funzionalità di un DBMS

5

DBMS: principali funzionalità

- Le caratteristiche fondamentali di un DBMS possono essere così sintetizzate:
- un DBMS è un sistema software che gestisce **grandi quantità di dati persistenti e condivisi**, e che offre supporto per almeno un **modello dei dati**
- La gestione di **grandi quantità di dati** richiede particolare attenzione ai problemi di **efficienza** (ottimizzazione delle richieste, ma non solo!)
- La **persistenza** e la **condivisione** richiedono che un DBMS fornisca meccanismi per garantire l'**affidabilità** dei dati (**fault tolerance**), per il **controllo degli accessi** e per il **controllo della concorrenza**
- Un **modello dei dati** consente agli utenti di disporre di un'astrazione di alto livello attraverso cui interagire con il DB.
- Diverse altre funzionalità vengono messe a disposizione per motivi di **efficacia**, ovvero per semplificare la descrizione delle informazioni, lo sviluppo delle applicazioni, l'amministrazione di un DB, ecc.

Funzionalità di un DBMS

6

Perché non i file system?

- ✦ Per gestire grandi quantità di dati in modo persistente e condiviso, sarebbe anche possibile ricorrere all'uso di un **file system**, ma ciò presenta molteplici inconvenienti, tra cui:
 - ✦ Povertà dell'astrazione offerta per modellare i dati
 - ✦ I meccanismi di condivisione sono limitati
 - ✦ Non sufficienti i meccanismi di protezione da guasti
 - ✦ L'accesso a file condivisi richiede una descrizione degli stessi nel codice delle applicazioni, con **rischi di descrizioni errate e quindi inconsistenti**
 - ✦ Non sono disponibili i servizi aggiuntivi offerti da un DBMS
- ✦ N.B. tuttavia, la gestione dei dati mediante file system può risultare più efficiente rispetto all'uso di un DBMS, proprio grazie alla maggiore semplicità dei meccanismi offerti.

Funzionalità di un DBMS



7

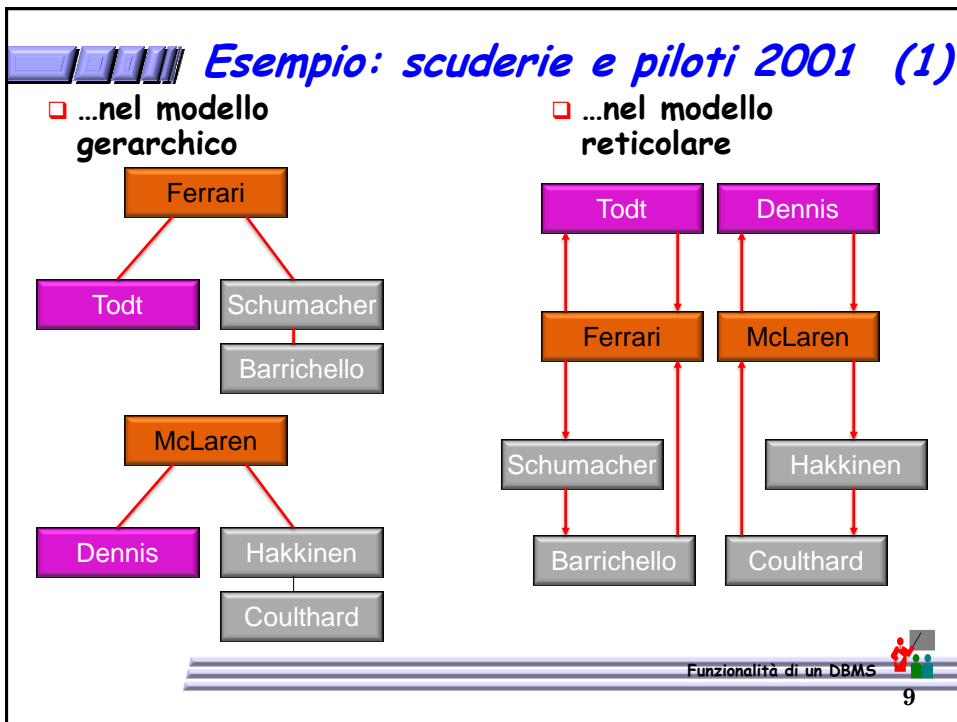
Il modello dei dati

- Dal punto di vista utente un DB è visto come una collezione di dati che modellano una certa porzione della realtà di interesse.
- L'**astrazione logica** con cui i dati vengono resi disponibili all'utente definisce un **modello dei dati**; più precisamente:
 - ✦ un **modello dei dati** è una collezione di concetti utilizzati per descrivere i dati, le loro associazioni, e i vincoli che questi devono rispettare.
- Un ruolo di primaria importanza nella definizione di un modello dei dati è svolto dai **meccanismi che possono essere usati per strutturare i dati** (cfr. i costruttori di tipo in un linguaggio di programmazione).
 - ✦ Ad es. esistono modelli in cui i dati sono descritti (solo) rispettivamente sotto forma di alberi (**modello gerarchico**), di grafi (**modello reticolare**), di oggetti complessi (**modello a oggetti**).

Funzionalità di un DBMS



8



Schemi e istanze

- In ogni DB si hanno due componenti:
 - lo **schema**, che descrive la **struttura dei dati** (la cosiddetta **parte intensionale** del DB);
 - l'**istanza**, ovvero **i dati veri e propri** (la **parte estensionale** del DB).
- Lo schema permette di fatto di **interpretare** i dati dell'istanza.

Scuderia	D_S	Scuderia	Pilota
Ferrari	Todt	Ferrari	Schumacher
McLaren	Dennis	Ferrari	Barrichello
		McLaren	Hakkinen
		McLaren	Coulthard

- In generale, mentre un'istanza varia nel tempo, lo schema tende a restare invariato, e variazioni dello schema comportano una ristrutturazione più o meno semplice del DB.

Funzionalità di un DBMS



11

Indipendenza fisica e logica

- Un importante obiettivo di un DBMS consiste nel fornire caratteristiche di:

Indipendenza fisica

- L'**organizzazione fisica** dei dati dipende da considerazioni legate all'efficienza delle organizzazioni adottate. La riorganizzazione fisica dei dati non deve comportare effetti collaterali sui programmi applicativi

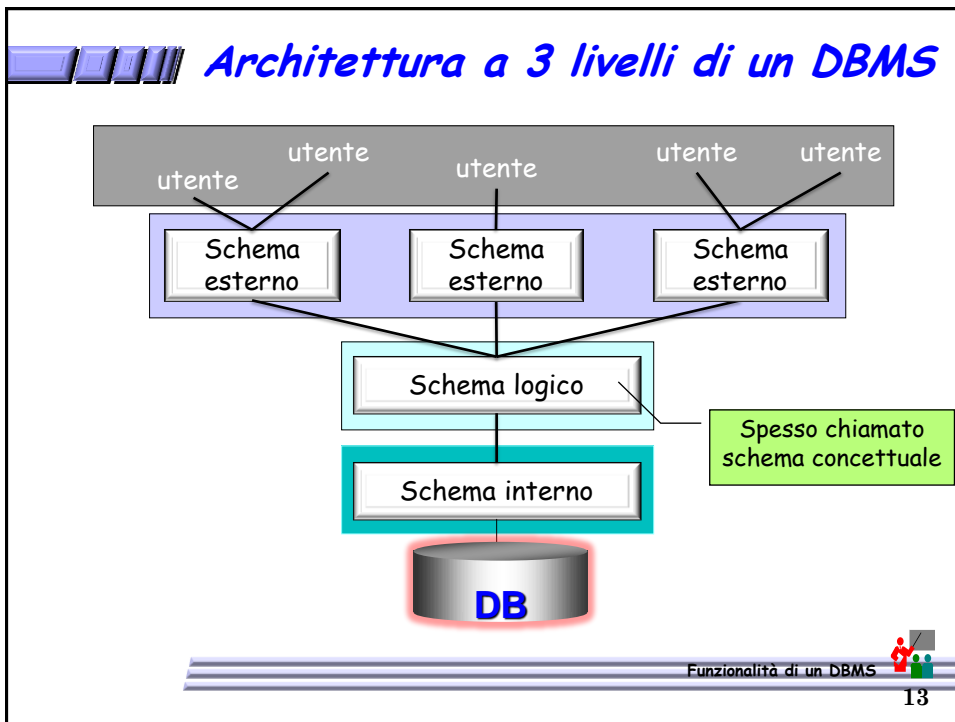
Indipendenza logica

- Pur in presenza di uno schema logico integrato non è utile o conveniente che ogni utente ne abbia una visione uniforme; è desiderabile che modifiche allo schema logico non comportino aggiornamenti degli schemi esterni e delle applicazioni
- La soluzione porta a quella che è comunemente nota come...

Funzionalità di un DBMS



12




Il livello fisico (o interno)

- Il **"DB fisico"** consiste di una serie di file, residenti su memorie di massa che contengono dati, indici e altro.
- Lo **schema fisico** descrive come il DB logico è rappresentato a livello fisico (ad es.: in quale/i file è memorizzata una relazione).

Ferrari	Barrichello	McLaren	Coulthard
Todt		Dennis	
Schumacher		Hakkinen	

File: c://apps/DB/data/scuderie

- La gestione del DB fisico è a carico di chi amministra il DBMS e non degli utenti, che possono quindi concentrarsi su aspetti di più alto livello per lo sviluppo delle loro applicazioni.

Funzionalità di un DBMS 

14

Il livello delle viste (o esterno)

- Il **livello esterno** è costruito a partire dallo schema logico integrato mediante la **definizione di viste ad hoc**, che descrivono parte dello schema logico secondo le esigenze dei diversi utenti.
 - Ad es. è possibile definire una vista che combina i dati di più relazioni:

Pilota	D_S
Schumacher	Todt
Barrichello	Todt
Hakkinen	Dennis
Coulthard	Dennis

- La distinzione tra livello esterno e logico può, in molti casi, risultare trasparente agli utenti, che, ad es., in un RDBMS vedono semplicemente un insieme di tabelle.

Funzionalità di un DBMS

15

Utilità delle viste

- Oltre a fornire una visione "personalizzata" del DB, le viste possono svolgere un ruolo importante anche per diversi altri motivi:
 - una **ristrutturazione dello schema integrato** può, in alcuni casi, essere opportunamente "mascherata" facendo uso di viste;
 - mediante le viste è possibile regolare meglio il **controllo degli accessi** al DB, ad es. mascherando dati riservati.
 - le viste possono essere usate per "**calcolare dinamicamente**" nuovi dati a partire da quelli memorizzati nel DB, senza per questo introdurre ridondanza.

Pilota	DataNascita		Pilota	Età
Schumacher	03/01/1969	Il 28/09/2001 →	Schumacher	32
Hakkinen	28/09/1968		Hakkinen	33

Funzionalità di un DBMS

16



I 3 livelli: una semplice analogia

- Per fissare meglio le idee, si considerino due classi Java, Pilota e Scuderia:
 - » il **livello logico** corrisponde (in un certo senso) all'interfaccia delle classi;
 - » il **livello interno** corrisponde alla loro implementazione concreta;
 - » il **livello esterno** può includere (un metodo di) una classe che usa i servizi delle classi Pilota e Scuderia.

```
public class Pilota {  
    private String nome;  
    private Date datanascita; ...  
    public Date getDataNascita() {...} ...}
```

```
public int Eta(Pilota unPilota) {  
    int eta =  
        diffYears(new Data(), unPilota.getDataNascita());  
    return eta;  
}
```

```
public class Scuderia {  
    private Pilota[] iPiloti;  
    ...  
    public Scuderia() {  
        iPiloti = new Pilota[2];  
    }  
    public LinkedList getPiloti() {  
        ...  
    }  
    ...}
```

Funzionalità di un DBMS

17



I linguaggi dei DBMS

- Un DBMS mette a disposizione diversi linguaggi per interagire con le BD. Il livello di astrazione di tali linguaggi dipende fortemente dal modello dei dati cui ci si riferisce.
- Una comune distinzione classifica i linguaggi sulla base delle funzioni svolte:
 - **DDL (Data Definition Language)**
 - » serve per definire gli schemi (logici, esterni, interni)
 - **DML (Data Manipulation Language)**
 - » serve per interrogare e modificare le istanze delle BD
 - **DCL (Data Control Language)**
 - » include comandi di vario tipo, ad es. per il controllo degli accessi

Il linguaggio **SQL** riunisce in sé istruzioni di tutte le tre tipologie (per cui si parla del DDL di SQL, del DML di SQL e del DCL di SQL).

Funzionalità di un DBMS

18

Linguaggi per lo sviluppo di applicazioni

- Oltre alle istruzioni necessarie per la manipolazione della base di dati, per realizzare applicazioni è necessario disporre di linguaggi general purpose. In generale sono disponibili diverse alternative, tra cui:

- ✦ **linguaggio nativo**

- ✦ Il DBMS dispone di un linguaggio Turing-completo (es. **Oracle PL/SQL**, **Postgres PLpg/SQL**)

- ✦ **linguaggio ospite con procedure esterne**

- ✦ Le istruzioni per manipolare i dati vengono invocate all'interno di un linguaggio convenzionale (es. **JDBC** per **Java**, **ODBC** per altri linguaggi di programmazione)

- ✦ **linguaggio ospite con precompilatore**

- ✦ "Embedded statements" in un linguaggio convenzionale che richiedono la presenza di un precompilatore (es. **PRO*C** per **SQL in ambiente Oracle**)

Funzionalità di un DBMS

19

Condivisione: regolamentare gli accessi

- ✦ Gli utenti di un DB sono naturalmente suddivisibili in diverse tipologie, a cui vanno pertanto associate **autorizzazioni** distinte, ad esempio:
 - ✦ uno **studente** può leggere i propri dati, ma non quelli di altri studenti; inoltre non può modificare l'elenco degli esami sostenuti;
 - ✦ un **docente** può leggere i dati dei soli studenti del proprio corso; non può modificare l'elenco degli esami già sostenuti da uno studente, ma può registrare esami del proprio corso;
 - ✦ la **segreteria** può leggere i dati di tutti e può registrare nuovi studenti.
- ✦ La gestione delle autorizzazioni può essere oltremodo complessa, per questo motivo sono previste specifiche figure di **Data Base Administrator (DBA)** che conferiscono agli utenti i "giusti" privilegi.
- ✦ **Mediante il DCL di SQL** la concessione dei privilegi (ad es. leggere i dati di una tabella) a un utente è molto semplificata; inoltre un utente può concedere a sua volta privilegi ad altri utenti.

Funzionalità di un DBMS

20



Condivisione: gestire la concorrenza (1)

- Un DBMS deve garantire che gli accessi ai dati, da parte di diverse applicazioni, non interferiscano tra loro. Al fine di conservare l'integrità dei dati è pertanto necessario far ricorso a opportuni meccanismi di **controllo della concorrenza**.
- Es.: si considerino due richieste, **T1** e **T2**, che cercano di eseguire in contemporanea un'operazione di prelievo, rispettivamente di **Q=80 €** e **Q=50 €**, dal conto corrente numero **123** che ha una disponibilità iniziale di **100 €**:

```
CC = read(C_Correnti; 123);  
if CC.disponibile >= Q  
then {CC.disponibile = CC.disponibile - Q;  
      eroga la somma Q;  
      write(C_Correnti; CC)}  
else print("operazione non possibile");
```

Nucleo della
transazione

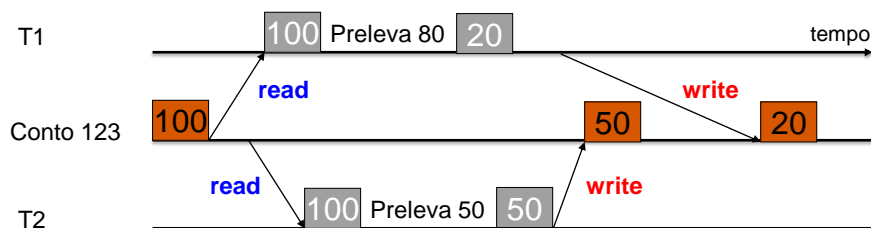
Funzionalità di un DBMS

21



Condivisione: gestire la concorrenza (2)

- Una possibile esecuzione (**non corretta!**) è la seguente:



- È compito del DBMS evitare tali esecuzioni non corrette:
 - » ad es. impedendo a T1 di scrivere il nuovo saldo (20) e forzando la rilettura del saldo aggiornato (50)

Funzionalità di un DBMS

22

Persistenza: protezione dai guasti

- Quando un'applicazione deve operare più operazioni di modifica, è possibile che per qualche motivo (guasti, intervento dell'utente, ecc.) solo una parte di queste venga effettivamente eseguita.
- In questo caso, per garantire l'integrità dei dati, il DBMS deve provvedere ad **annullare** tali modifiche.
- Es.: se **T1** volesse trasferire **50 €** dal conto **123** (saldo **100 €**) al conto **235** (saldo **30 €**), si potrebbe verificare la seguente situazione (**indesiderata!**):

Funzionalità di un DBMS 23

Moduli di un DBMS (1)

□ A grandi linee si possono già individuare alcuni moduli fondamentali di un DBMS, che realizzano le funzionalità sin qui descritte:

Query Manager

Transaction Manager

DBA

DDL Compiler

Storage Manager

Logging & Recovery Manager

Concurrency Manager

QM : analizza, autorizza, ottimizza ed esegue le richieste

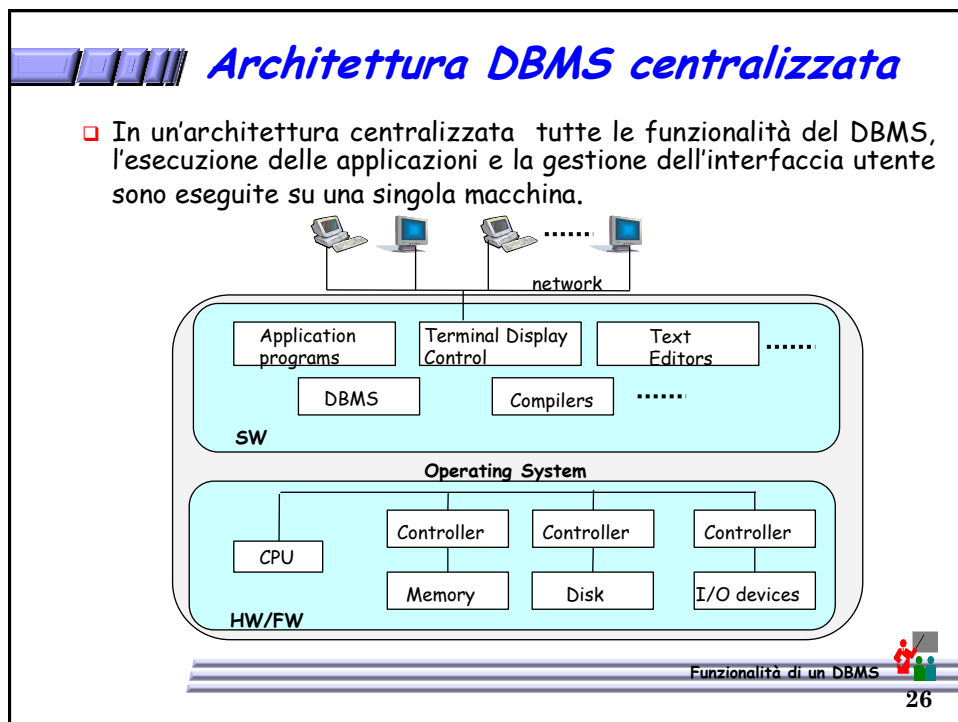
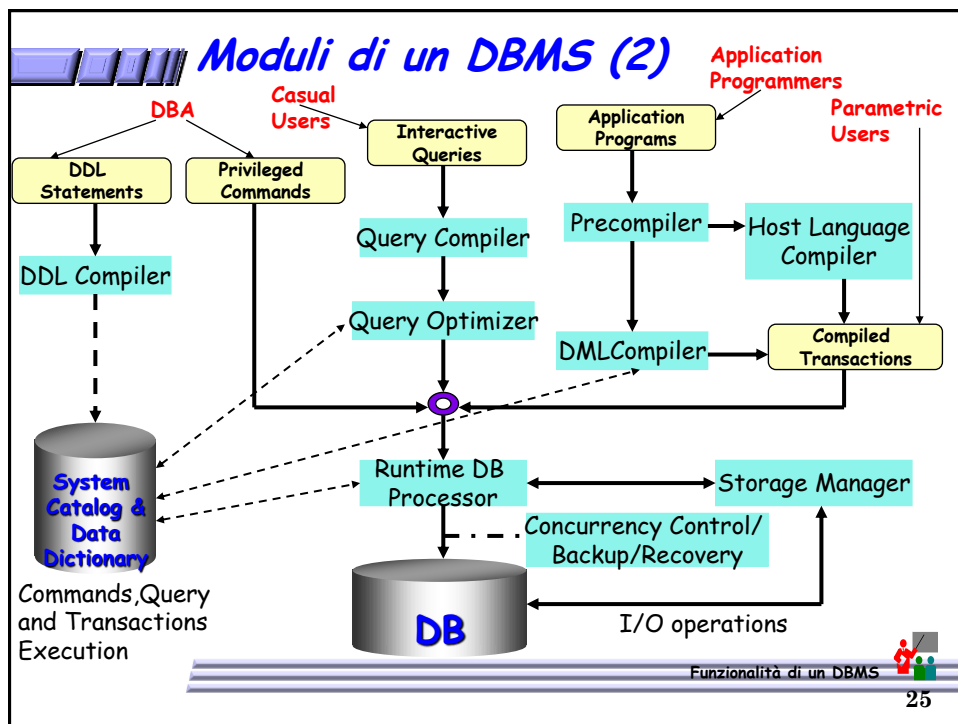
TM : coordina le richieste

Log&RM : protegge dai guasti

CM : gestisce accessi concorrenti

SM : gestisce la memoria di massa

Funzionalità di un DBMS 24

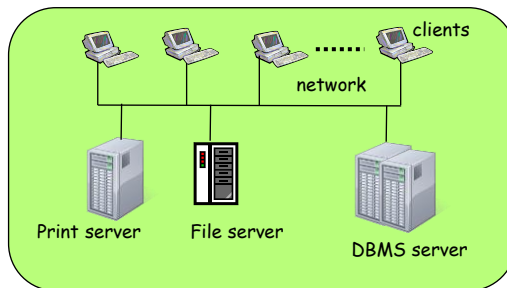




Architettura Client/Server

- ✦ L'architettura **client/server** è stata sviluppata per gestire in modo efficiente ambienti in cui sono presenti un gran numero di postazioni PC, workstation, file server, ecc.
- ✦ In questo contesto un client è tipicamente una macchina utente in grado di fornire funzionalità d'interfaccia e di elaborazione locale.
- ✦ un server è un sistema in grado di fornire servizi di varia natura alle macchine client.

Logical two-tier
client/server
architecture



Funzionalità di un DBMS

27

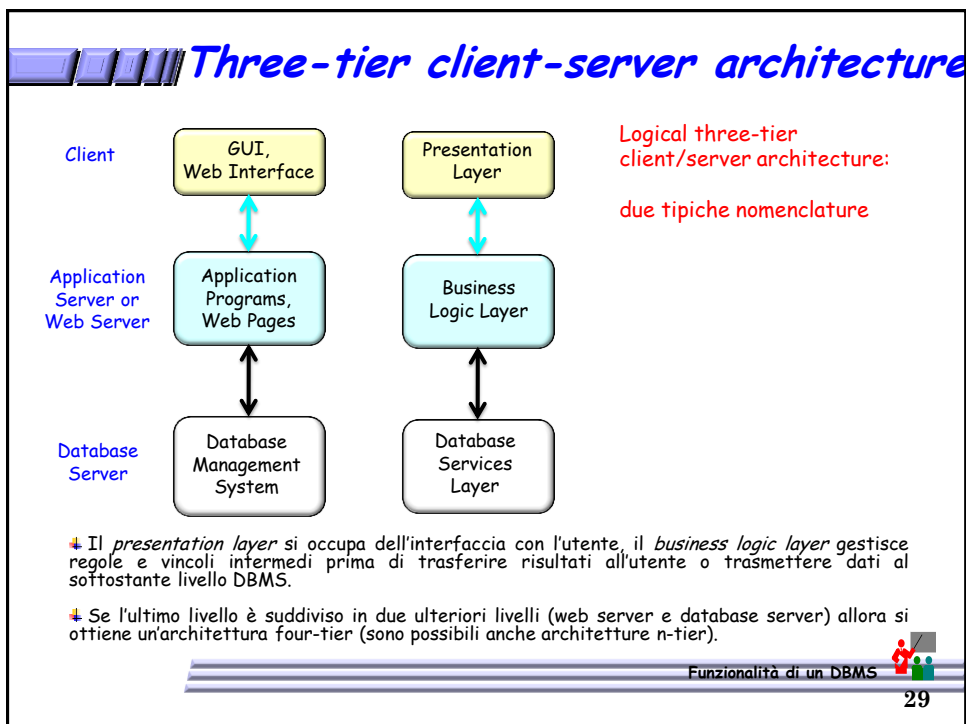


Two-tier vs Three-tier architecture

- ✦ In un'architettura **two-tier** (a due livelli) le componenti software sono distribuite su due sistemi: client e server.
- ✦ La diffusione del Web ha modificato sostanzialmente i ruoli dei client e dei server, richiedendo il passaggio ad architetture a tre livelli (**three-tier architecture**) in cui è presente un livello intermedio (spesso chiamato application server o web server).
- ✦ Nell'architettura three-tier è possibile garantire un migliore grado di sicurezza rispetto agli accessi, infatti:
 1. al database server si accede solo attraverso il livello intermedio;
 2. i client non possono accedere direttamente al database server.

Funzionalità di un DBMS

28



Sommario:

- Un **DBMS** è un sistema software (complesso!) che gestisce **grandi quantità di dati persistenti e condivisi**, garantendone l'**integrità** e la **sicurezza** mediante l'**esecuzione coordinata delle richieste**, la **protezione da malfunzionamenti** e la **realizzazione di una politica degli accessi**.
- Un **modello dei dati** è una collezione di concetti che vengono utilizzati per descrivere i dati, le loro associazioni, e i vincoli che questi devono rispettare.
- Una **base di dati** si compone di uno **schema**, che ne descrive la struttura logica, e di un'**istanza**, che consiste dei dati memorizzati.
- Mediante un'**organizzazione a 3 livelli**, un DBMS permette di ottenere gradi di **indipendenza fisica e logica dei dati**.

Funzionalità di un DBMS 
30

Un po' di storia (1)

Anni 60

- ✓ Sviluppo di file system generalizzati
- ✓ Modello gerarchico
- ✓ IBM introduce IMS
- ✓ CODASYL DBTG introduce il modello reticolare

Anni 70

- ✓ DBMS basati sul modello reticolare (IDMS, TOTAL)
- ✓ Codd sviluppa il modello relazionale (1970)
- ✓ Chen introduce il modello ER (1976)
- ✓ Primi prototipi di RDBMS (System R, Ingres)

Anni 80

- ✓ Diffusione sul mercato di RDBMS
- ✓ Sviluppo di modelli semantici
- ✓ Sviluppo della ricerca nelle aree OODBMS, KBMS, DDBMS

Funzionalità di un DBMS



31

Un po' di storia (2)

Anni 90

- ✓ Diffusione di architetture client-server
- ✓ OODBMS appaiono sul mercato
- ✓ Nuove aree di impiego di DBMS
(design, multimedia, applicazioni scientifiche, datawarehouse, process management, ...)
- ✓ Consapevolezza dei problemi dei legacy DB
- ✓ Fine anni 90 : Internet e Web favoriscono lo sviluppo di protocolli di interazione e cooperazione tra sistemi diversi

Anni 2000 (alcune aree di ricerca e sviluppo)

- ✓ DBMS architecture
- ✓ Database design
- ✓ Query Languages
- ✓ Application development
- ✓ Database Administration
- ✓ Data Warehousing & Data Mining
- ✓ Web DB, Image DB, Spatial DB, Multimedia DB, ..

Funzionalità di un DBMS



32