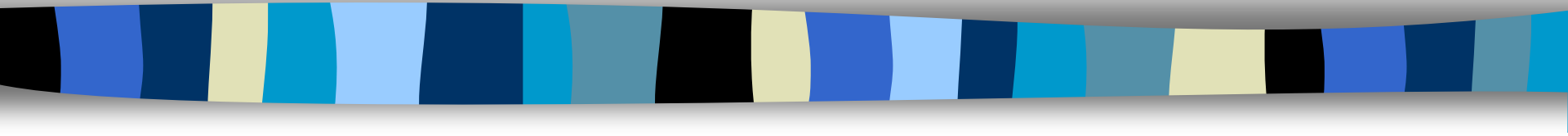


Introduzione ai Sistemi Informativi



Il sistema informativo

- ❑ I processi, per poter operare, hanno bisogno di conoscere la risorsa o le risorse su cui agiscono, in altri termini hanno bisogno di informazioni.
 - Si definisce **Information Processing Capacity** – capacità elaborativa “adeguatezza di un’organizzazione rispetto alle necessità di elaborare informazioni a essa imposte dai propri obiettivi e dal contesto in cui opera”.
- ❑ L’insieme delle informazioni gestite - generate, utilizzate, elaborate - dai processi aziendali e le modalità per la loro gestione costituisce il **sistema informativo**.
- ❑ La porzione del sistema informativo in cui le informazioni sono raccolte, elaborate, archiviate e scambiate mediante l’uso di tecnologie informatiche costituisce il **sistema informatico**.

Sistemi Informativi VS Basi di Dati

- ❑ I concetti di **Sistema Informativo** e di **Base di Dati** non devono essere considerati dei sinonimi.
- ❑ La definizione e la gestione di un Sistema Informativo richiede la perfetta conoscenza di:
 - Processi aziendali
 - Informazioni necessarie ai processi
 - Struttura aziendale
- ❑ Progettare un Sistema Informativo richiede competenze in ambito di:
 - Organizzazione aziendale
 - Economia
 - Psicologia
- ❑ Un Sistema Informativo da un lato si adatta alle caratteristiche dell'azienda in cui è costruito, dall'altro ne modifica struttura e processi ottimizzandone la struttura.
- ❑ Le **basi di dati** sono una sottoporzione del Sistema Informativo atta alla memorizzazione strutturata delle informazioni. La loro funzione è solo quella di fornire un supporto informatico per la memorizzazione dei dati; essendo questi il nucleo di tutti i sistemi informativi le basi di dati rivestono un ruolo centrale.

Una modellazione imprecisa della base di dati implica lo scorretto funzionamento del Sistema Informativo.

DBMS

- ❑ Un **DBMS** (Data Base Management System) è un sistema software in grado di gestire efficientemente le informazioni necessarie a un sistema informativo, rappresentandone i dati in forma integrata e garantendone la persistenza.
- ❑ Le caratteristiche fondamentali possono essere così sintetizzate:
 - La gestione di grandi quantità di dati richiede particolare attenzione ai problemi di efficienza
 - 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 del DB, ecc.

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 che vengono 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
 - Ad esempio, esistono modelli in cui i dati sono descritti (solo) sotto forma di alberi (*modello gerarchico*), di grafi (*modello reticolare*) e di oggetti complessi (*modello a oggetti*)
- ❑ Un **RDBMS** è un DBMS che offre come modello logico dei dati il modello relazionale

Esempio: scuderie e piloti

modello relazionale

Ferrari	Todt
McLaren	Dennis

Ferrari	Schumacher
Ferrari	Barrichello
McLaren	Hakkinen
McLaren	Coulthard



ATTENZIONE: le tabelle sono solo “un modo”
(conveniente) di rappresentare le relazioni

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

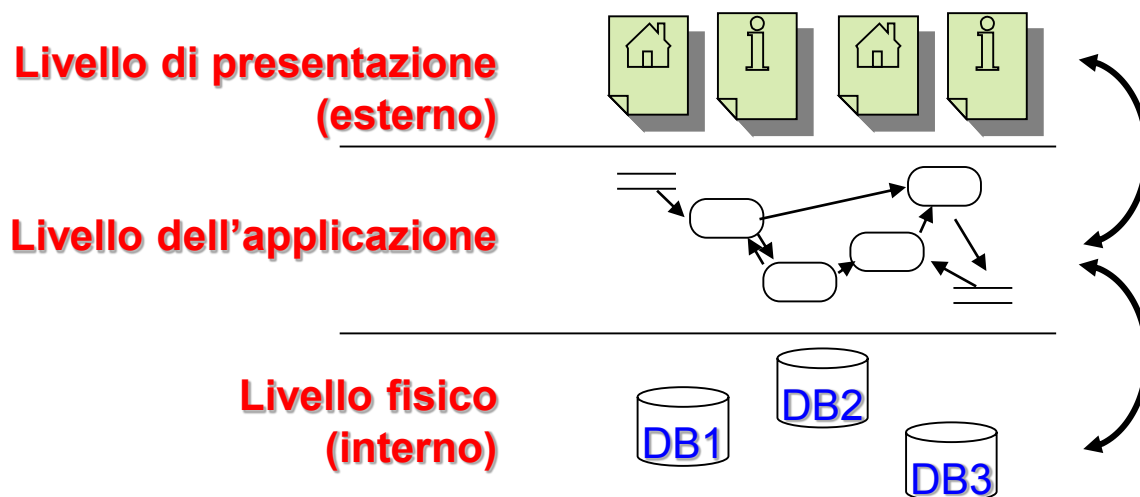
Scuderia	D_S
Ferrari	Todt
McLaren	Dennis

Scuderia	Pilota
Ferrari	Schumacher
Ferrari	Barrichello
McLaren	Hakkinen
McLaren	Coulthard

- ❑ In generale, mentre un'istanza varia nel tempo, lo schema tende a restare invariato

Architettura del DBMS

- Un importante obiettivo di un DBMS consiste nel fornire le seguenti caratteristiche
 - **Indipendenza fisica:** la riorganizzazione fisica dei dati non deve comportare effetti collaterali sui programmi applicativi.
 - **Indipendenza logica:** le modifiche allo schema logico non devono comportare aggiornamenti degli schemi esterni e delle applicazioni.
- La soluzione porta a quella che è comunemente nota come **architettura a 3 livelli** di un DBMS.



I linguaggi dei DBMS

- ❑ Un DBMS mette a disposizione diversi linguaggi per interagire con il DB. 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 esempio per il controllo degli accessi
- ❑ Il linguaggio SQL riunisce in sé istruzioni delle suddette tre tipologie.

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, la concessione di privilegi (ad es., leggere i dati di una tabella) ad un utente è molto semplificata; inoltre, un utente può concedere a sua volta i privilegi ad altri utenti.

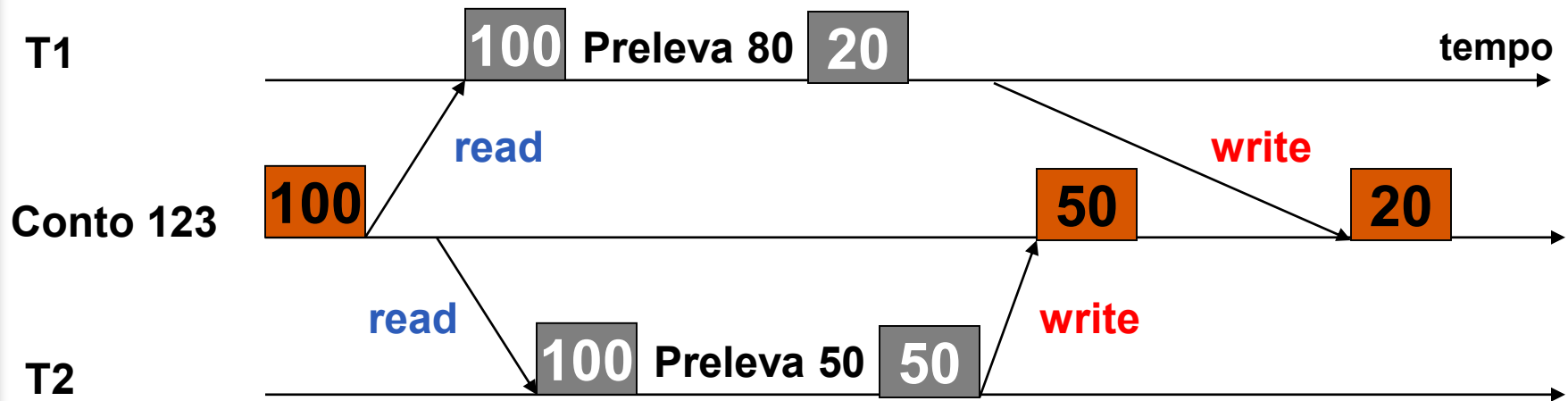
Condivisione: gestire la concorrenza

- ❑ 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
- ❑ Esempio: si considerino due richieste, **T1** e **T2**, che cercano di eseguire in contemporanea un'operazione di prelievo, rispettivamente di $Q = 80 \text{ €}$ e $Q = 50 \text{ €}$, 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");
```

Condivisione: gestire la concorrenza

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



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

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
 - Esempio: se T1 volesse trasferire 50€ dal conto 123 (saldo 100€) al conto 235 (saldo 30€), si potrebbe verificare la seguente situazione (*indesiderata!*):

