

TECNOLOGIA DBMS

È un Sistema di Gestione di Basi di Dati è un Sistema Software in Grado di Gestire Collezioni di Dati che Siano: **GRANDI**

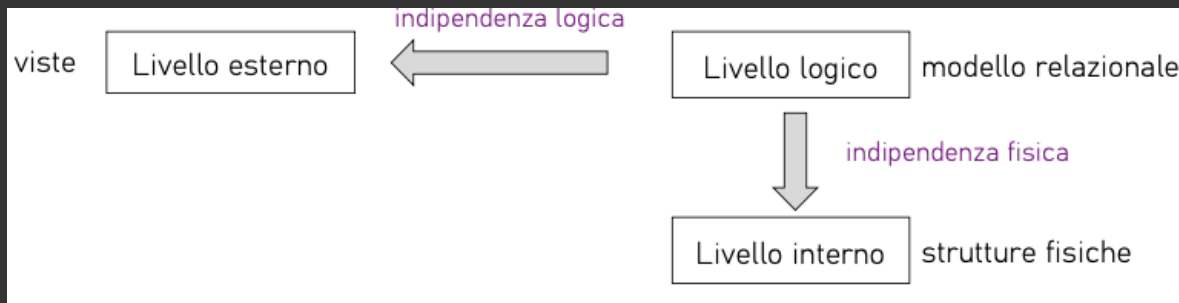
CONDIVISE

PERSISTENTI

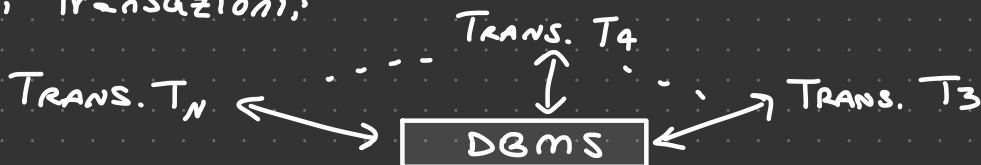
Il DBMS deve Essere: **EFFICIENTE** } Dipendenza dalla
EFFICACE } Progettazione della
Base di Dati

BASI DI DATI = Collezioni di Dati Gestite da un DBMS

Architettura su 3 livelli dei DBMS:



Un **DBMS Basato Sul Modello Relazionale** è Nella maggior parte dei casi anche un **Sistema Transazionale**. Quindi fornisce un Meccanismo per la definizione ed Esecuz. di Transazioni:



TRANSAZIONE

È un **Unità di lavoro** Svolta da un programma Applicativo (che interagisce con una Base di Dati) per la quale si vogliono garantire proprietà di

- ❖ CORRETTEZZA
- ❖ ROBUSTEZZA
- ❖ ISOLAMENTO

Una Transazione è ATOMICA, o **tutto** o **Niente**

Ha Effetto sulla Base di dati → Non ha Effetto

e NON Sono Ammesse Esecuzioni parziali

SINTASSI

<Transazione> ⇒ INIZIO
 <Programma>
 FINE

<Programma> ⇒ <istruzione | commit Work | RollBack work>

La Transazione va a **Buon Fine** all'Esecuzione di un **COMMIT WORK**

La Transazione **NON** ha alcun Effetto se Viene Eseguito un **ROLLBACK WORK** (si ripristina la Situazione com'era prima di Eseguire la Transazione)

È Ben formata se:

- ❧ Inizia con **BEGIN TRANSACTION**
- ❧ Termina con un **END TRANSACTION**
- ❧ dopo un **commit/rollback** non si Esegono Altri accessi Alla Base di Dati

ESEMPIO TRANSAZIONE BEN FORMATA;

INIZIO

```
update CONTO set SALDO = SALDO + 10  
where filiale = '005' and Numero = 15  
commit Work
```

FINE

PROPRIETÀ DELLE TRANSAZIONI

- ❧ ATOMICITÀ
- ❧ CONSISTENZA
- ❧ ISOLAMENTO
- ❧ PERSISTENZA

} **ACID** ⇒ Proprietà da Garantire per
Ogni Transazione

DBMS dovrebbero Garantire **QUESTE** Proprietà

ATOMICITÀ:

È un Unità di **Esecuzione INDIVISIBILE** (è ATOMICA)

Se una Transazione Viene Interrotta prima l'esecuzione del Commit il lavoro fatto deve Essere disfatto ripristinando

La Situazione della Base di dati prima della Transazione.

CONSISTENZA:

Non deve Violare i Vincoli di Integrità

Verifica Immediata:

- ❖ Fatta nel Corso della Transazione
- ❖ Applicazione può Reagire alla Violazione
- ❖ Viene Abortita solo l'ultima operazione e il Sistema restituisce un ERRORE

Verifica differita:

se al Commit un Vincolo d'Integrità viene violato la Transazione viene Abortita Senza possibilità da Parte dell'Applicazione di Reagire alla Violazione

ISOLAMENTO:

L'esecuzione di una Transazione deve ESSERE INDIPENDENTE dalla contemporanea Esecuzione di Altre Transazione

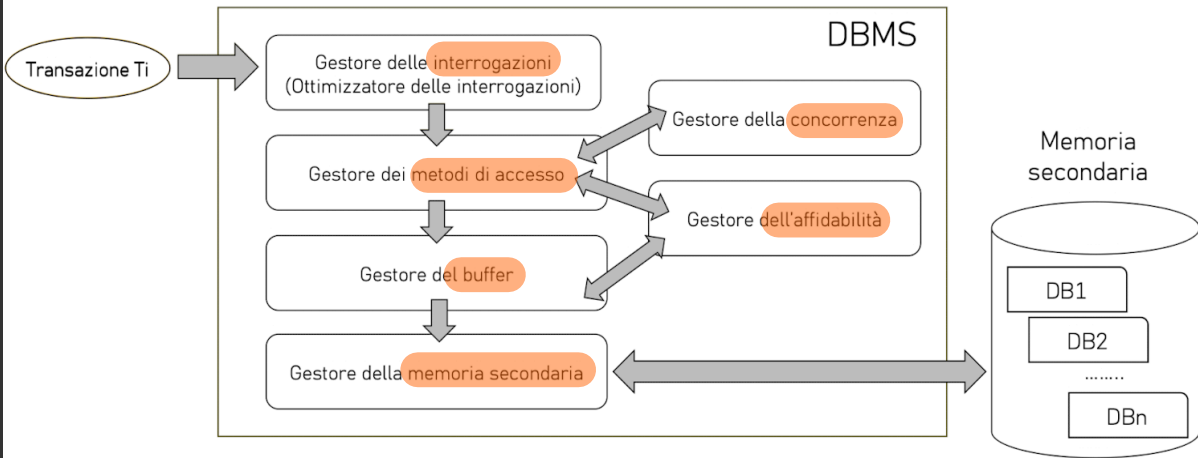
IL ROLLBACK non deve Creare ROLLBACK a Catena di Altre Transazioni in Esecuzione

PERSISTENZA:

Effetto devo di una Transazione che ha Eseguito il Commit Non deve Andare Perso

IL Sistema deve Essere in Grado, in caso di Guasto, di Garantire gli Effetti delle Transazioni che Al Momento del Guasto avevano Già Eseguito un Commit

ARCHITETTURA:



MODULI e PROPRIETÀ:

Gestore Metodi D'accesso → CONSISTENZA

Gestore Exec. Concorrente → ATOMICITÀ e ISOLAMENTO

Gestore Affidabilità → ATOMICITÀ e PERSISTENZA