

- INDICE: struttura di accesso secondaria → rende più veloce l'accesso ai dati memorizzati

- INDICE PRIMARIO: - ORDINATO

- DIM. FISSA

- DUE CAMPI :   
CAMPO CHIAVE  
PULITORES PROTO DEL DISCO

- SEMPRE SPARSO: Contiene un solo per alcune dei blocchi memorizzati nel file → (INDICE PRIMARIO)

- INDICE SECONDARIO: ULTERIORE struttura d'accesso dove c'è GIA INDICE PRIMARIO

- ORDINATO, NON ORDINATO, HASH

- CAMPO INDICIZZATO PUÒ ESSERE CHIAVE o NO

↳ SE CHIAVE → DISUO: contiene val. per ogni  
contiene VOCE per ogni VALORE  
↳ SE NON CHIAVE E RECORD A LUNGHEZZA  
VARIABILE → SPARSO

- SUPERCHIAVE: Parte del VINCULO DI CHIAVE che è un VINCULO DI INSERITA!  
INTERAZIONALI (I PIÙ IMPORTANTI)

Una CHIAVE è un insieme di attributi utilizzati per identificare UNIVOCAMENTE le tuple di una RELAZIONE.

- INSIEME di attributi K è SUPERCHIAVE di una relazione

se r non contiene due tuple distinte  
 $t_1$  e  $t_2$  con  $t_1[K] = t_2[K]$ .

- PUÒ ESSERE MINIMA o NON MINIMA

CHIAVE contiene in sé altre SUPERCHIAVI.

- NULL non ammesso su CHIAVE PRIMARIA, ma generalmente AMMESSI sulle altre CHIAVI.

- CAVALLO DI TROIA

Esempio di attacco sofisticato che richiede quindi POLITICHE MANDATORIE per la sicurezza della base di dati.

ABBIANO DUE USR: X e Y

X proprietario di file A

Y proprietario di file B e file P, tuttavia codice malvagio

Y concede ~~esecuzione~~ PRIVILEGIO di esecuzione su P e PRIVILEGIO di scrittura su B ad

PRIVILEGIO

X.

X attiva su P il permesso di esecuzione su B, Y ora può leggere contenuto di B senza avere il PERMESSO.

- **LOST UPDATE**: Problema dovuto ad accesso concorrente di utenti/applicazioni sui dati.

(UTSUTI)

Si verifica quando due CLIENT accedono agli stessi DATI ed eseguono le proprie OPERAZIONI in tempi tali che il VALORE FINALE risulta errato.

Esempio: U1:  $x := \text{read}(X) + 100$      $x \approx 50 + 100$

U2:  $x := \text{read}(X) + 200$      $x \approx 50 + 200$

U1:  $\text{write}(x)$      $\approx 250$     (Ho preso  $x=150$ )

U2:  $\text{write}(x)$      $\approx 250$

- **DIRTY READ**: Problema dovuto ad accesso concorrente di utenti/applicazioni sui dati.

(UTSUTI)

Si verifica quando un CLIENT U1 aggiorna un dato X e successivamente prima che il dato X temporaneamente aggiornato da U1 viene letto da U2 prima che X venga ripristinato al valore originale.

Esempio: U1:  $x := \text{read}(X) + 100$

U1:  $\text{write}(x)$

U2:  $x := \text{read}(X) + 200$      $\leftarrow$  X potenzialmente errato.

U2:  $\text{write}(x)$

U1: Abort

- **INCORRECT SUMMARY**: Problema dovuto ad accesso concorrente di utenti/applicazioni sui dati.

Se un CLIENT calcola una funzione aggregata su un insieme di record mentre un altro client effettua aggiornamenti sui record, il primo client può calcolare considerando alcuni VALORI PRIMA dell'aggiornamento e altri DOPPO aggiornamento.

Esempio: U2:  $\text{sum} := 0$

U1:  $X := \text{read}(X) - 200$

U1:  $\text{write}(X)$

U2:  $\text{sum} := \text{sum} + \text{read}(X)$

U2:  $\text{sum} := \text{sum} + \text{read}(Y)$

U1:  $Y := \text{read}(Y) + 200$

U1:  $\text{write}(Y)$

**EVITARLI**: Atti avendo l'indicatore del livello di ISOLAMENTO

- READ COMMITTED: evita dirty READ;

- REPEATABLE READ: evita: DIRTY READ, LOST UPDATE, INCORRECT SUMMARY;

- SERIALIZABLE: evita: DIRTY READ, LOST UPDATE, INCORRECT SUMMARY.

ATOMICITY: l'esecuzione di una transazione deve essere totale o nucca, non può ammettere esecutori parziali.

Se si verifica un errore prima del COMMIT → effetti parziali non memorizzati nel DB.

ROLLBACK: cancellazione di tutti gli effetti parziali di una transazione.

CONSISTENCY:

Ogni transazione:  
1) ASSUME DI LAVORARE SU DB con tutti i vincoli di INTEGRITÀ VERIFICATI.  
2) DEVE LASCIARE UN DB che soddisfa tutti VINCOLI INTEGRITÀ

I vincoli possono essere specificati COME:

- IMMEDIATE: verifica eseguita DURANTE l'esecuzione.

UNDO dell'istruzione che causa VIOLAZIONI, senza imporre ABORT

- DEFERRED: verifica al TERMINE della Transazione, DOPO il COMMIT.

- Se ci sono VINCOLI VIOLATI → ROLLBACK transazione con UNDO di intera sequenza di comandi.