DMIF, Università di Udine

# Tecnologie Digitali per il Cibo e la Ristorazione

## *Basi di Dati Relazionali - Argomenti avanzati*

Andrea Brunello
andrea.brunello@uniud.it

A.A. 2021–2022

# Transazioni

- Una transazione è un'*unità* fondamentale di un programma in esecuzione la quale accede ed eventualmente modifica degli elementi in un database
- La transazione è composta da un insieme di operazioni che, dal punto di vista logico, possono essere considerate come un'unica operazione
  - Esempio: trasferimento di fondi da un conto corrente all'altro
- Attraverso le transazioni, un DBMS è in grado di fornire garanzie per quanto riguarda proprietà desiderabili relative alla gestione dei fallimenti e alla gestione dell'accesso concorrente

# Transazioni
## Esempio

- Transazione che opera il trasferimento di \$50 dal conto $A$ al conto $B$:
  1. read($A$)
  2. $A = A - 50$
  3. write($A$)
  4. read($B$)
  5. $B = B + 50$
  6. write($B$)
- Cosa succede
  - In caso di un crash del sistema?
  - In caso di letture/scritture concorrenti?

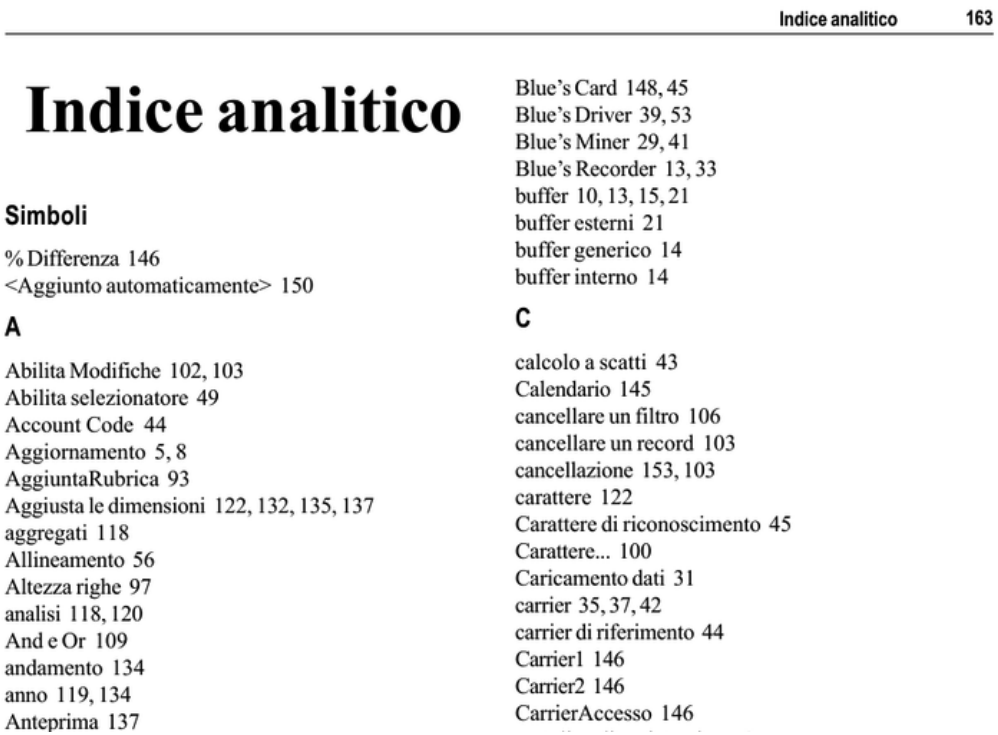
- **Atomicità**: tutte le operazioni in una transazione hanno successo, oppure viene fatto il roll-back di ogni operazione (come se la transazione non fosse mai stata eseguita)
- **Consistenza**: la transazione trova e lascia la base di dati in uno stato consistente (relativamente ai vincoli)
- **Isolamento**: le transazioni non interferiscono l'una con l'altra. Il risultato che si ottiene dall'esecuzione concorrente di un insieme di transazioni è equivalente al risultato che si otterrebbe eseguendole in sequenza secondo un dato ordine
- **Persistenza (Durability)**: una volta che la transazione ha terminato la sua esecuzione, i suoi risultati sono permanentemente registrati nella base di dati, a prescindere dall'insorgere di fallimenti del sistema

# Accesso ai dati e indici

- Supponiamo di avere, nella nostra base di dati, una tabella *Cliente*(*Nome*, *Cognome*, *Codice*, *Citta*, *Telefono*)
- Supponiamo inoltre di richiedere alla base di dati il numero di telefono di tutti i clienti di Udine
- Intuitivamente, ciò richiede di scansionare la tabella determinando, riga per riga, la città del cliente
- Tale operazione, effettuata su una tabella contenente centinaia di migliaia o milioni di righe, può essere estremamente dispendiosa
- Ciò è particolarmente vero se la tabella è troppo grande per essere caricata interamente nella memoria principale. Si rendono in tal caso necessari diversi (e costosi) trasferimenti fra disco e memoria

# Accesso ai dati e indici

- Al fine di rendere più veloci operazioni come quella descritta, è possibile ricorrere a delle particolari strutture, dette **indici**
- Un indice ha molte affinità con l'indice analitico che si trova alla fine di un libro





## Indice analitico

# Accesso ai dati e indici

- Un indice in ambito DBMS è composto da due campi:
  - **Chiave/campo di ricerca** (es., *Citta*)
  - **Puntatore**: identifica l'area/le aree del disco dove è possibile recuperare i record con lo specifico valore della chiave di ricerca
- Considerazioni
  - Definire indici sui campi di una tabella accelera le interrogazioni che fanno uso di tali campi
  - Tuttavia, le operazioni di inserimento / cancellazione / aggiornamento della tabella risultano rallentate, in quanto vi è anche il costo di aggiornamento degli indici