



Linguaggio SQL: costrutti avanzati

Gestione delle transazioni

Gestione delle transazioni

- Introduzione
- Transazioni in SQL
- Proprietà delle transazioni



Gestione delle transazioni

Introduzione

Esempio applicativo



- Operazioni bancarie
 - operazione di prelievo dal proprio conto corrente mediante bancomat

Esempio applicativo



➤ Operazioni bancarie

- operazione di prelievo dal proprio conto corrente mediante bancomat
- operazione di versamento di denaro contante sul proprio conto corrente





➤ Operazioni svolte

- specificare l'importo richiesto
- verificare la disponibilità
- memorizzare il movimento
- aggiornare il saldo
- abilitare l'erogazione della somma richiesta



➤ Operazioni svolte

- specificare l'importo richiesto
- verificare la disponibilità
- memorizzare il movimento
- aggiornare il saldo
- abilitare l'erogazione della somma richiesta

➤ Tutte le operazioni devono essere eseguite correttamente, altrimenti il prelievo non va a buon fine

➤ *Cosa succede se un cointestatario diverso del conto fa un altro prelievo?*

- *Cosa succede se un cointestatario diverso del conto fa un altro prelievo?*
- *Cosa succede in caso di malfunzionamento?*

Versamento



➤ Operazioni svolte

- verificare l'importo versato
- memorizzare il movimento
- aggiornare il saldo

➤ Tutte le operazioni devono essere eseguite correttamente, altrimenti il versamento non va a buon fine

Versamento

➤ *Cosa succede se un'altra persona fa un versamento sullo stesso conto?*

Versamento

- *Cosa succede se un'altra persona fa un versamento sullo stesso conto?*
- *Cosa succede in caso di malfunzionamento?*

Esempio: operazioni bancarie

- La base di dati bancaria è un ambiente multiutente
 - diversi operatori possono operare contemporaneamente sulla stessa porzione di dati

Esempio: operazioni bancarie

- La base di dati bancaria è un ambiente multiutente
 - diversi operatori possono operare contemporaneamente sulla stessa porzione di dati
- La gestione corretta delle informazioni richiede
 - meccanismi per la gestione *dell'accesso concorrente* alla base di dati

Esempio: operazioni bancarie

- La base di dati bancaria è un ambiente multiutente
 - diversi operatori possono operare contemporaneamente sulla stessa porzione di dati
- La gestione corretta delle informazioni richiede
 - meccanismi per la gestione *dell'accesso concorrente* alla base di dati
 - meccanismi per il *ripristino* (recovery) dello stato corretto della base di dati in caso di guasti

Gestione delle transazioni

- Necessaria quando più utenti possono accedere contemporaneamente ai dati
- Offre meccanismi efficienti per
 - gestire l'accesso concorrente ai dati
 - effettuare il recovery a seguito di un malfunzionamento

- Una transazione è una sequenza di operazioni che
- rappresenta un'unità elementare di lavoro
 - può concludersi con un successo o un insuccesso
 - in caso di successo, il risultato delle operazioni eseguite deve essere memorizzato in modo permanente nella base di dati
 - in caso di insuccesso, la base di dati deve ritornare allo stato precedente l'inizio della transazione

Sistema transazionale

- Un sistema che mette a disposizione un meccanismo per la definizione e l'esecuzione di transazioni viene detto *sistema transazionale*
- I DBMS contengono blocchi architetturali che offrono servizi di gestione delle transazioni



Gestione delle transazioni

Transazioni in SQL

➤ Una transazione è

- un'unità logica di lavoro, non ulteriormente scomponibile
- una sequenza di operazioni (istruzioni SQL) di modifica dei dati, che porta la base di dati da uno stato consistente a un altro stato consistente
 - non è necessario conservare la consistenza negli stati intermedi

Inizio di una transazione

- Per definire l'inizio di una transazione, il linguaggio SQL prevede l'istruzione
 - `START TRANSACTION`

Inizio di una transazione

- Per definire l'inizio di una transazione, il linguaggio SQL prevede l'istruzione
 - **START TRANSACTION**
- Di solito l'istruzione di inizio della transazione è omessa
 - l'inizio è implicito
 - prima istruzione SQL del programma che accede alla base di dati
 - prima istruzione SQL successiva all'istruzione di termine della transazione precedente

Fine di una transazione

- Il linguaggio SQL prevede istruzioni per definire la fine di una transazione
 - Transazione terminata con successo
 - COMMIT [WORK]
 - l'azione associata all'istruzione si chiama *commit*

Fine di una transazione

➤ Il linguaggio SQL prevede istruzioni per definire la fine di una transazione

- Transazione terminata con successo
 - COMMIT [WORK]
 - l'azione associata all'istruzione si chiama *commit*
- Transazione terminata con insuccesso
 - ROLLBACK [WORK]
 - l'azione associata all'istruzione si chiama *abort*

- Azione eseguita quando una transazione termina con successo

- Azione eseguita quando una transazione termina con successo
- La base di dati è in un nuovo stato (finale) corretto
- Le modifiche dei dati effettuate dalla transazione divengono
 - permanenti
 - visibili agli altri utenti

- Azione eseguita quando una transazione termina a causa di un errore
 - per esempio, di un errore applicativo

- Azione eseguita quando una transazione termina a causa di un errore
 - per esempio, di un errore applicativo
- Tutte le operazioni di modifica dei dati eseguite durante la transazione sono “annullate”
- La base di dati ritorna nello stato precedente l’inizio della transazione
 - i dati sono nuovamente visibili agli altri utenti

- Trasferire la somma 100
- dal conto corrente bancario
IT92X0108201004300000322229
 - al conto corrente bancario
IT32L0201601002410000278976

- Trasferire la somma 100
- dal conto corrente bancario
IT92X0108201004300000322229
 - al conto corrente bancario
IT32L0201601002410000278976

START TRANSACTION;

➤ Trasferire la somma 100

- dal conto corrente bancario
IT92X0108201004300000322229
- al conto corrente bancario
IT32L0201601002410000278976

```
START TRANSACTION;  
UPDATE Conto-Corrente  
SET Saldo = Saldo - 100  
WHERE IBAN='IT92X0108201004300000322229';
```

➤ Trasferire la somma 100

- dal conto corrente bancario
IT92X0108201004300000322229
- al conto corrente bancario
IT32L0201601002410000278976

```
START TRANSACTION;  
UPDATE Conto-Corrente  
  SET Saldo = Saldo - 100  
  WHERE IBAN='IT92X0108201004300000322229';  
UPDATE Conto-Corrente  
  SET Saldo = Saldo + 100  
  WHERE IBAN= 'IT32L0201601002410000278976';
```

➤ Trasferire la somma 100

- dal conto corrente bancario
IT92X0108201004300000322229
- al conto corrente bancario
IT32L0201601002410000278976

```
START TRANSACTION;  
UPDATE Conto-Corrente  
  SET Saldo = Saldo - 100  
  WHERE IBAN='IT92X0108201004300000322229';  
UPDATE Conto-Corrente  
  SET Saldo = Saldo + 100  
  WHERE IBAN= 'IT32L0201601002410000278976';  
COMMIT;
```



Gestione delle transazioni

Proprietà delle transazioni

Proprietà delle transazioni

- Le proprietà principali delle transazioni sono
- **A**tomicity – atomicità

Proprietà delle transazioni

- Le proprietà principali delle transazioni sono
- Atomicity – atomicità
 - Consistency – consistenza

Proprietà delle transazioni

- Le proprietà principali delle transazioni sono
- **A**tomicity – atomicità
 - **C**onsistency – consistenza
 - **I**solation – isolamento

Proprietà delle transazioni

- Le proprietà principali delle transazioni sono
- **A**tomicity – atomicità
 - **C**onsistency – consistenza
 - **I**solation – isolamento
 - **D**urability – persistenza (o durabilità)

Proprietà delle transazioni

- Le proprietà principali delle transazioni sono
 - **A**tomicity – atomicità
 - **C**onsistency – consistenza
 - **I**solation – isolamento
 - **D**urability – persistenza (o durabilità)
- Sono riassunte dall'acronimo (inglese) ***ACID***

- Una transazione è un'unità indivisibile (atomo) di lavoro
- devono essere eseguite tutte le operazioni contenute nella transazione
 - oppure nessuna delle operazioni contenute nella transazione deve essere eseguita
 - la transazione non ha nessun effetto sulla base di dati

- Una transazione è un'unità indivisibile (atomo) di lavoro
 - devono essere eseguite tutte le operazioni contenute nella transazione
 - oppure nessuna delle operazioni contenute nella transazione deve essere eseguita
 - la transazione non ha nessun effetto sulla base di dati
- La base di dati non può rimanere in uno stato intermedio assunto durante l'esecuzione di una transazione

- L'esecuzione di una transazione deve portare la base di dati
- da uno stato iniziale consistente (corretto)
 - a uno stato finale consistente

- L'esecuzione di una transazione deve portare la base di dati
 - da uno stato iniziale consistente (corretto)
 - a uno stato finale consistente
- La correttezza è verificata dai vincoli di integrità definiti sulla base di dati

- L'esecuzione di una transazione deve portare la base di dati
 - da uno stato iniziale consistente (corretto)
 - a uno stato finale consistente
- La correttezza è verificata dai vincoli di integrità definiti sulla base di dati
- Quando si verifica la violazione di un vincolo di integrità il sistema interviene
 - per annullare la transazione
 - oppure, per modificare lo stato della base di dati eliminando la violazione del vincolo

- L'esecuzione di una transazione è indipendente dalla contemporanea esecuzione di altre transazioni

- L'esecuzione di una transazione è indipendente dalla contemporanea esecuzione di altre transazioni
- Gli effetti di una transazione non sono visibili dalle altre transazioni fino a quando la transazione non è terminata
 - si evita la visibilità di stati intermedi non stabili
 - uno stato intermedio può essere annullato da un rollback successivo
 - in caso di rollback, è necessario effettuare rollback delle altre transazioni che hanno osservato lo stato intermedio (effetto domino)

- L'effetto di una transazione che ha effettuato il commit è memorizzato in modo permanente
 - le modifiche dei dati eseguite da una transazione terminata con successo sono permanenti dopo il commit

- L'effetto di una transazione che ha effettuato il commit è memorizzato in modo permanente
 - le modifiche dei dati eseguite da una transazione terminata con successo sono permanenti dopo il commit
- Garantisce l'affidabilità delle operazioni di modifica dei dati
 - i DBMS offrono meccanismi di ripristino dello stato corretto della base di dati dopo che si è verificato un guasto