



UNIVERSITÀ DEGLI STUDI DI SALERNO

Atzeni, Ceri, Fraternali, Paraboschi, Torlone
Basi di dati *Quinta edizione*
McGraw-Hill Education, 2018

BASI DI DATI
SQL-DML Avanzato
Accesso e Transazioni - terza parte



DIEM
DIPARTIMENTO DI
INGEGNERIA DELL'INFORMAZIONE ED ELETTRICA E MATEMATICA APPLICATA

Matteo Gaeta
Full Professor – Senior Member IEEE

Controllo dell'accesso

- In SQL è possibile specificare chi (utente) e come (lettura, scrittura, ...) può utilizzare la base di dati (o parte di essa)
- Oggetto dei privilegi (diritti di accesso) sono di solito le tabelle, ma anche altri tipi di risorse, quali singoli attributi, viste o domini
- Un utente predefinito `_system` (amministratore della base di dati) ha tutti i privilegi
- Il creatore di una risorsa ha tutti i privilegi su di essa

Privilegi

- Un privilegio è caratterizzato da:
 - ✓ la risorsa cui si riferisce
 - ✓ l'utente che concede il privilegio
 - ✓ l'utente che riceve il privilegio
 - ✓ l'azione che viene permessa
 - ✓ la trasmissibilità del privilegio

Tipi di privilegi offerti da SQL

- **insert:** permette di inserire nuovi oggetti (ennuple)
- **update:** permette di modificare il contenuto
- **delete:** permette di eliminare oggetti
- **select:** permette di leggere la risorsa
- **references:** permette la definizione di vincoli di integrità referenziale verso la risorsa (può limitare la possibilità di modificare la risorsa)
- **usage:** permette l'utilizzo in una definizione (per esempio, di un dominio)

Grant e Revoke

➤ Concessione di privilegi:

**grant < Privileges | all privileges > on Resource
to Users [with grant option]**

- ✓ **grant option** specifica se il privilegio può essere
trasmesso ad altri utenti

grant select on Department to Stefano

➤ Revoca di privilegi

**revoke Privileges on Resource from Users
[restrict | cascade]**

Autorizzazioni e Commenti

- La gestione delle autorizzazioni deve
“nascondere” gli elementi cui un utente
non può accedere, senza sospetti

➤ Esempio:

- **Impiegati** non esiste (esiste **Impiegati**)
- **ImpiegatiSegreti** esiste, ma l'utente non
è autorizzato

❖ L'utente deve ricevere lo stesso messaggio

Autorizzazioni e Commenti

- Come autorizzare un utente a vedere solo alcune ennuple di una relazione?
- **Attraverso una vista:**
 - ✓ Definiamo la vista con una condizione di selezione
 - ✓ Attribuiamo le autorizzazioni sulla vista, anziché sulla relazione di base

Autorizzazioni e Commenti

- (Estensioni di SQL:1999)
- Concetto di ruolo, cui si associano privilegi (anche articolati), poi concessi agli utenti attribuendo il ruolo

Transazioni

- Lo standard tradizionalmente adottato per sistemi dedicati all'elaborazione di transazioni affidabili, gli OLTP è riassumibile con l'acronimo **ACID** (Atomicità, Consistenza, Isolamento, Durabilità).
- I DSS richiedono standard meno rigidi quando si utilizzino dati storici e statici, generati da sistemi OLTP ed importati. Questo standard è riassumibile con la parola **BASE**.

Transazioni

- Insieme di operazioni da considerare indivisibile ("atomico"), corretto anche in presenza di concorrenza e con effetti definitivi
- Proprietà ("acide"):
 - ❖ **Atomicità**
 - ❖ **Consistenza**
 - ❖ **Isolamento**
 - ❖ **Durabilità(Persistenza)**

Transazioni: Atomicità

- **Atomicità:** significa che la transazione è indivisibile nella sua esecuzione, e che tale esecuzione deve essere totale, completa oppure nulla o annullata, come se non fosse mai stata iniziata.
- In altre parole, di una transazione non sono accettate esecuzioni parziali.

Transazioni: Atomicità

- **Una transazione è una unità atomica di elaborazione**
- **Non può lasciare la base di dati in uno stato intermedio**
 - ✓ un guasto o un errore prima del commit debbono causare l'annullamento (UNDO) delle operazioni eventualmente svolte
 - ✓ un guasto o errore dopo il commit non deve avere conseguenze; se necessario vanno ripetute (REDO) le operazioni
- **Esito**
 - ✓ Commit = caso "normale" e più frequente
 - ✓ Abort (o rollback)
 - richiesto dall'applicazione = suicidio
 - richiesto dal sistema (violazione dei vincoli, concorrenza, incertezza in caso di fallimento) = omicidio

Transazioni: Consistenza

- **Consistenza (o coerenza):** significa che prima di iniziare una transazione il sistema è in uno stato internamente coerente e quando la transazione termina tale sistema deve trovarsi in un altro stato internamente coerente, cioè senza violazione di vincoli di integrità della base dati stessa che genererebbero inconsistenza tra i dati distribuiti tra le varie tabelle.
- La transazione rispetta i vincoli di integrità
- Conseguenza: **se lo stato iniziale è corretto anche lo stato finale è corretto**

Transazioni: Isolamento

- **Isolamento:** significa che ogni transazione deve essere eseguita in modo isolato e indipendente da tutte le altre transazioni.
- Ogni transazione è indipendente dalle altre.
- L'eventuale fallimento di una transazione non deve influire con altre transazioni in esecuzione nello stesso lasso di tempo.
- La transazione non risente degli effetti delle altre transazioni concorrenti: ***l'esecuzione concorrente di una collezione di transazioni deve produrre un risultato che si potrebbe ottenere con una esecuzione sequenziale***
- Conseguenza: una transazione non espone i suoi stati intermedi
 - ✓ si evita l' "effetto domino"

Transazioni: Isolamento

- **Durabilità (o persistenza):** significa che una volta che la transazione è stata marcata come completata, i cambiamenti che essa ha apportato alla base dati non dovranno più essere persi, salvandoli quindi su supporto non volatile.
- Tali modifiche ai dati devono essere scritti in modo che ne venga garantita la leggibilità anche in caso di guasto del sistema.
- Gli effetti di una transazione andata in commit non vanno perduti ("durano per sempre"), anche in presenza di guasti
 - ✓ "commit" significa "impegno"

Le transazioni sono ... atomiche

- La sequenza di operazioni sulla base di dati viene eseguita per intero o per niente:
 - trasferimento di fondi da un conto A ad un conto B: o si fanno il prelevamento da A e il versamento su B o nessuno dei due

Le transazioni sono ... consistenti

- Al termine dell'esecuzione di una transazione, i vincoli di integrità debbono essere soddisfatti
- "Durante" l'esecuzione ci possono essere violazioni, ma se restano alla fine allora la transazione deve essere annullata per intero ("abortita")

Le transazioni sono ... isolate

- L'effetto di transazioni concorrenti deve essere coerente (ad esempio "equivalente" all'esecuzione separata)
 - se due assegni emessi sullo stesso conto corrente vengono incassati contemporaneamente si deve evitare di trascurarne uno



I risultati delle transazioni sono durevoli

- La conclusione positiva di una transazione corrisponde ad un impegno (in inglese **commit**) a mantenere traccia del risultato in modo definitivo, anche in presenza di guasti e di esecuzione concorrente

Definizione di Transazione

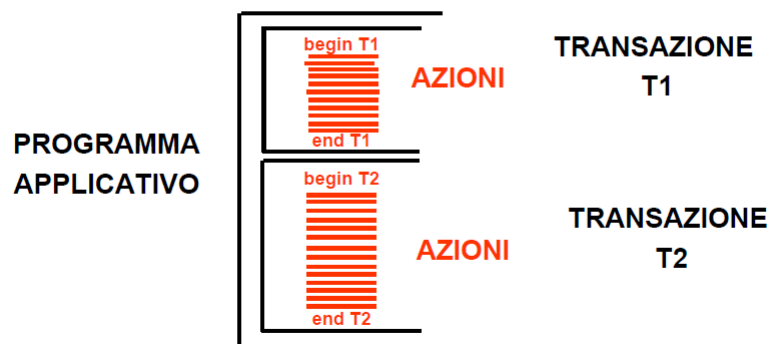
- Transazione: parte di programma caratterizzata da un inizio (**begin-transaction**, **start transaction** in SQL, non sempre esplicitata), una fine (**end-transaction**, non esplicitata in SQL) e al cui interno deve essere eseguito una e una sola volta uno dei seguenti comandi
 - **commit work** per terminare correttamente
 - **rollback work** per abortire la transazione
- la transazione va eseguita per intero o per niente

Transazioni in SQL

- Una transazione inizia al primo comando SQL dopo la "connessione" alla base di dati oppure alla conclusione di una precedente transazione (lo standard indica anche un comando **start transaction**, non obbligatorio, e quindi non previsto in molti sistemi)
- Conclusione di una transazione
 - **commit [work]**: le operazioni specificate a partire dall'inizio della transazione vengono eseguite sulla base di dati
 - **rollback [work]**: si rinuncia all'esecuzione delle operazioni specificate dopo l'inizio della transazione
- Molti sistemi prevedono una modalità **autocommit**, in cui ogni operazione forma una transazione

Le Transazioni

DIFFERENZA FRA APPLICAZIONE E TRANSAZIONE



Una transazione in SQL

```
start transaction          (opzionale)
update ContoCorrente
  set Saldo = Saldo - 10
  where NumeroConto = 12345 ;
update ContoCorrente
  set Saldo = Saldo + 10
  where NumeroConto = 55555 ;
commit work;
```

Una transazione con decisioni

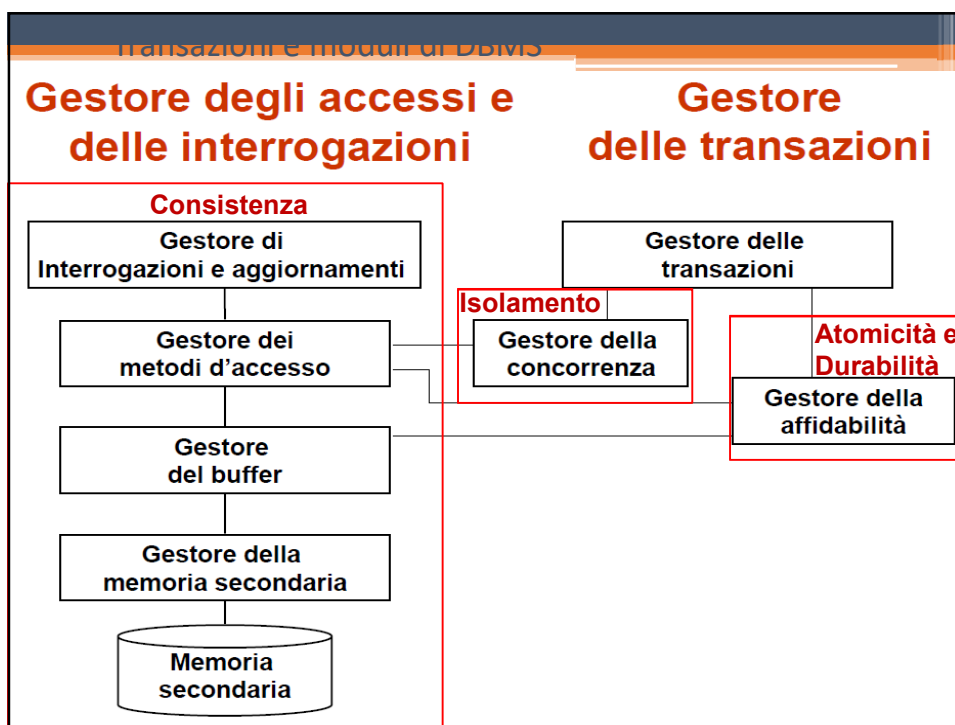
```
start transaction; (opzionale)
update ContoCorrente
set Saldo = Saldo + 10 where NumConto = 12202;
update ContoCorrente
set Saldo = Saldo - 10 where NumConto = 42177;
select Saldo into A
  from ContoCorrente
  where NumConto = 42177;
if (A >= 0)
  then commit work
else rollback work;
```

Una transazione in JDBC

- Scelta della modalità delle transazioni: un metodo definito nell'interfaccia **Connection**:
setAutoCommit(boolean autoCommit)
- **con.setAutoCommit(true)**
 - ✓ (default) "autocommit": ogni operazione è una transazione
- **con.setAutoCommit(false)**
 - ✓ gestione delle transazioni da programma
 - **con.commit()**
 - **con.rollback()**
 - ✓ non c'è **start transaction**

Transazioni e moduli di DBMS

- **Atomicità e durabilità**
 - ✓ Gestore dell'affidabilità (Reliability manager)
- **Isolamento:**
 - ✓ Gestore della concorrenza
- **Consistenza:**
 - ✓ Gestore dell'integrità a tempo di esecuzione (con il supporto del compilatore del DDL)



BASI DI DATI

Materiale utilizzato e bibliografia

➤ Le slide utilizzate dai docenti per le attività frontali sono in gran parte riconducibili e riprese dalle slide originali (con alcuni spunti parziali ripresi dai libri indicati) realizzate da:

- ✓ autori del libro *Basi di Dati* (Atzeni e altri) testo di riferimento del corso *Basi di Dati* e sono reperibili su internet su molteplici link oltre che laddove indicato dagli stessi autori del libro;
- ✓ Prof.ssa Tiziana Catarci e dal dott. Ing. Francesco Leotta – corso di *Basi di Dati* dell'Università degli Studi La Sapienza di Roma al seguente link ed altri: <http://www.dis.uniroma1.it/~catarci/basidatGEST.html> (molto Interessanti anche le lezioni su YouTube).
- ✓ Proff. Luca Allulli e Umberto Nanni, *Libro Fondamenti di basi di dati*, editore HOEPLI (testo di facile lettura ed efficace).

➤ Diverse slide su specifici argomenti utilizzate dai docenti per le attività frontali sono anche in parte riconducibili e riprese dalle slide originali – facilmente reperibili e accessibili su internet - realizzate da:

Prof.ssa Roberta Aiello – corso *Basi di Dati* dell'Università di Salerno

Prof. Dario Maio – corso *Basi di Dati* dell'Università di Bologna al seguente link ed altri: <http://bias.csr.unibo.it/maio>

Prof. Marco Di Felice - corso *Basi di Dati* dell'Università di Bologna al seguente link ed altri: <http://www.cs.unibo.it/difelice/dbsi/>

Prof. Marco Maggini e prof. Franco Scarselli - corso *Basi di Dati* dell'Università di Siena ai seguenti link ed altri: [http://staff.icar.cnr.it/pontieri/didattica/LabSI/lezioni/_preliminari-DB1%20\(Maggini\).pdf](http://staff.icar.cnr.it/pontieri/didattica/LabSI/lezioni/_preliminari-DB1%20(Maggini).pdf)

Prof.ssa Raffaella Gentilini - corso *Basi di Dati* dell'Università di Perugia al seguente link ed altri: <http://www.dmi.unipg.it/raffaella.gentilini/BD.htm>

Prof. Enrico Giunchiglia - corso *Basi di Dati* dell'Università di Genova al seguente link ed altri: <http://www.star.dist.unige.it/~enrico/BasiDiDati/>

Prof. Maurizio Lenzerini - corso *Basi di Dati* dell'Università degli Studi La Sapienza di Roma al seguente link ed altri: <http://didattica.info.altervista.org/Quinta/Database2.pdf>

- The PostgreSQL Global Development Group - PostgreSQL nn.xx Documentation
- PostgreSQL (appendice - scaricabile dal sito del libro (area studenti) e www.postgresql.org)