Corso sul linguaggio SQL

Modulo L2B (SQL)

6 – Controlli e sicurezza

Prerequisiti

- Problemi di controllo degli accessi
- Problemi di sicurezza delle **transazioni**

Introduzione

In questa Unità si prendono in esame i problemi relativi alla sicurezza di un database. Questi problemi, molto più evidenti nella gestione in rete (che verrà affrontata successivamente) sono affrontati mediante il **controllo degli accessi** del personale al DB e attraverso **tecniche che rendono sicure le operazioni complesse** (gestioni bancarie, controllo aereo,…).

Informazioni generali

N.B. – A solo scopo didattico:

- i caratteri MAIUSCOLI indicano parole chiave del linguaggio;
- i caratteri *corsivi* indicano elementi che dovranno essere specificati dal programmatore;
- le parentesi quadre indicano opzione
- la barra verticale "|" indica alternativa.

Controllo dell'accesso

In SQL è possibile specificare per un DB i **diritti di accesso** (**privilegi**), che stabiliscono:

- **chi lo usa** (il tipo di utente, operatore, amministratore, …);
- **per cosa lo usa** (le operazioni ammesse, lettura, scrittura, …).

Oggetto dei **privilegi** possono essere:

- tabelle
- singoli attributi
- viste
- domini

Un utente predefinito **_system** (amministratore della base di dati) *ha tutti i privilegi*.

Il creatore di una risorsa ha sempre tutti i privilegi su di essa

Privilegi

Un privilegio è caratterizzato:

- dalla **risorsa** cui si riferisce
- dall'**utente concedente**, ossia colui che accorda il privilegio
- dall'**utente ricevente**, che riceve il privilegio
- dall'azione che viene **permessa**
- dalla **trasmissibilità** del privilegio

Tipi di privilegi offerti da SQL

Normalmente, i privilegi sono assegnati sulle tabelle e possono essere una qualunque combinazione dei seguenti.

Il privilegio…	**permette di…**
Select	…eseguire query sulla tabella mediante SELECT
Insert	…inserire nuove tuple con INSERT
Update	…modificare le righe con UPDATE
Delete	…cancellare righe con DELETE
References	…definire vincoli di integrità referenziale verso la risorsa (può limitare la possibilità di modificare la risorsa)
Alter	…modificare la struttura della tabella
Index	…creare un indice con CREATE INDEX

Comandi relativi ai privilegi

I privilegi possono essere gestiti mediante due comandi:

- **GRANT** (concede privilegi);
- **REVOKE** (revoca privilegi).

Comando **GRANT**

Il comando **GRANT** ha la seguente sintassi:

GRANT *lista_privilegi* **ON** *oggetto* **to** *user*;

dove:

- *lista_privilegi* è una lista dei privilegi indicati nella tabella precedente;
- *oggetto* indica l'oggetto sui cui vengono applicati;
- *user* indica l'utente che li riceve.

Comando **GRANT**

Esempi:

- **GRANT** select, insert, update, delete **ON** articoli **TO** Stefano;

(consente i comandi specificati sulla tabella *articoli* all'utente Stefano)

- **GRANT** all **ON** articoli **TO** Stefano;
  (consente tutti i comandi sulla tabella *articoli* all'utente Stefano)

- **GRANT** select **ON** articoli **TO** public;
  (consente il comando **SELECT** sulla tabella *articoli* a tutti gli utenti)

Comando **REVOKE**

Il comando **REVOKE** ha la seguente sintassi:

REVOKE *lista_privilegi* **ON** *oggetto* **FROM** *user*;

dove:

- *lista_privilegi* è una lista dei privilegi indicati nella tabella precedente;
- *oggetto* indica l'oggetto a cui vengono revocati;
- *user* indica l'utente che viene privato dei privilegi.

Comando **REVOKE**

Esempi:

- **REVOKE** delete **ON** articoli **FROM** Anna;

(revoca il comando **Delete** sulla tabella *articoli* all'utente Anna)

- **REVOKE** all **ON** articoli **TO** Anna;
  (revoca tutti i comandi sulla tabella *articoli* all'utente Anna)

- **REVOKE** all **ON** articoli **TO** public;
  (revoca tutti i comandi sulla tabella *articoli* a tutti gli utenti)

Concetto di transazione

Un'operazione complessa, nel senso che risulta composta da un insieme di operazioni più semplici, da considerare indivisibile, prende il nome di **transazione**.

In pratica, una transazione è un blocco di istruzioni strettamente correlate, tale che, in caso di errore (nel programma, oppure dovuto ad un problema esterno, per esempio se il sistema si blocca) non verrà annullata solo l'istruzione interessata dall'errore, ma l'intera transazione.

In tal modo si è certi che l'integrità dei dati verrà garantita e solo quando una transazione viene eseguita, produce effetti definitivi (esempio operazioni bancarie, prenotazioni aeree, ecc).

Proprietà delle transazioni

Le **proprietà delle transazioni** sono:

- **A**tomicità
- **C**onsistenza
- **I**solamento
- **D**urabilità (persistenza)

L'acronimo che descrive le proprietà delle transazioni è **ACID**.

Proprietà delle transazioni
- atomicità

La transazione deve essere **indivisibile** (**atomica**) nel senso che la sequenza di operazioni sulla base di dati viene eseguita **per intero o per niente**.

Esempio: il trasferimento di fondi da un conto A ad un conto B prevede il prelevamento da A e il versamento su B, oppure nessuno dei due.

Proprietà delle transazioni
- consistenza

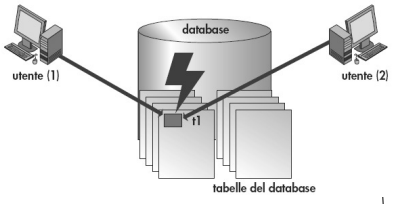
La transazione deve risultare **consistente**, ossia al termine della sua esecuzione, i vincoli di integrità debbono essere soddisfatti.

Durante l'esecuzione ci possono essere violazioni ai vincoli, ma se restano alla fine allora la transazione deve essere annullata per intero (*abortita*)

Proprietà delle transazioni
- isolamento

Le transazioni devono essere **isolate**, in particolare l'esecuzione di transazioni **concorrenti** deve essere *equivalente* alla loro esecuzione separata.

Esempio: se due assegni emessi sullo stesso conto corrente vengono incassati contemporaneamente, occorre trattare le due operazioni in modo sequenziale.



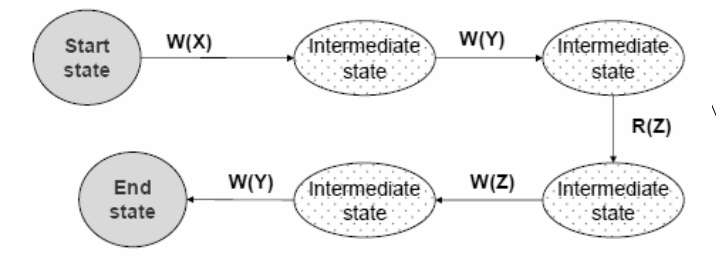
Proprietà delle transazioni
- durabilità

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.

Esempio: le transazioni bancarie devono essere registrate e ricostruibili in ogni momento.

Transazioni in SQL

Una transazione può essere vista come una sequenza di operazioni elementari di lettura (R) e scrittura (W) di oggetti (tuple) del DB che, a partire da uno stato iniziale consistente del DB, porta il DB in un nuovo stato finale consistente.



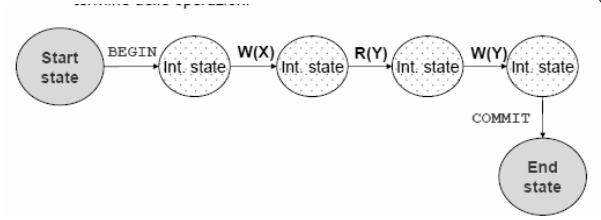
Terminazione di transazioni

Una transazione può terminare:

- **correttamente**, tutte le operazioni previste dalla transazione sono eseguite completamente e l'esito sui dati è definitivo;
- **non correttamente**, un'operazione tra quelle previste dalla transazione non ha avuto successo e tutta la transazione non deve produrre effetto sui dati.

Terminazione di transazioni
- terminazione corretta

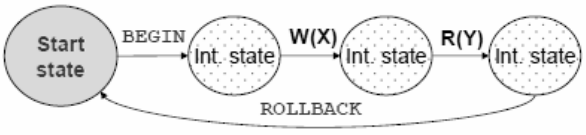
L'applicazione esegue la transazione fino a trovare una particolare istruzione SQL, detta **COMMIT** (o **COMMIT WORK**) che comunica "ufficialmente" al DBMS il termine delle operazioni



Terminazione di transazioni
- terminazione non corretta

Può darsi che:

- la transazione, per qualche motivo, non può essere continuata e quindi "abortisce" eseguendo l'istruzione SQL **ROLLBACK** (o **ROLLBACK WORK**)
- il sistema non è in grado (ad es. guasto, violazione di un vincolo) di garantire la corretta prosecuzione della transazione, che viene quindi abortita



Se la transazione non può terminare, il DBMS deve *disfare* (**UNDO**) tutte le modifiche apportate.

Operare transazioni in SQL

I comandi fondamentali di SQL per gestire le transazioni sono:

- **BEGIN TRANSACTION** *nomeTransazione*
  - specifica l'inizio della transazione (le operazioni non vengono eseguite sulla base di dati);
- **COMMIT** *nomeTransazione*
  - le operazioni specificate a partire dal *begin transaction* vengono eseguite materialmente sui dati;
- **ROLLBACK** *nomeTransazione*
  - si rinuncia all'esecuzione delle operazioni specificate dopo l'ultimo *begin transaction* e si ripristinano i dati originali;
- **END TRANSACTION** *nomeTransazione*
  - indica il termine della transazione.

COMMIT e ROLLBACK

T1	T2
R(X)	
W(X)	
Commit	
	R(Y)
	W(Y)
	Commit

Il DBMS può consentire l'accesso a dati condivisi attraverso la serializzazione delle transazioni.

T1	T2
R(X)	
	R(Y)
	W(Y)
	Commit
W(X)	
Commit	

In alcuni casi, il DBMS può eseguire due transazioni concorrenti in modalità *interleaving*, alternando le operazioni.

Utilizzo di transazioni in SQL

Un utilizzo tipico delle transazioni è il seguente:

- prima di eseguire una transazione, si esegue un'istruzione di inizio transazione (*begin transaction*);
- si eseguono le operazioni di interrogazione e modifica dei dati;
- se si riscontra qualche anomalia, si esegue l'istruzione *rollback work*, per abortire la transazione;
- se si sono eseguite tutte le operazioni senza riscontrare anomalie, si esegue un'istruzione di conferma (*commit work*), per confermare la transazione.

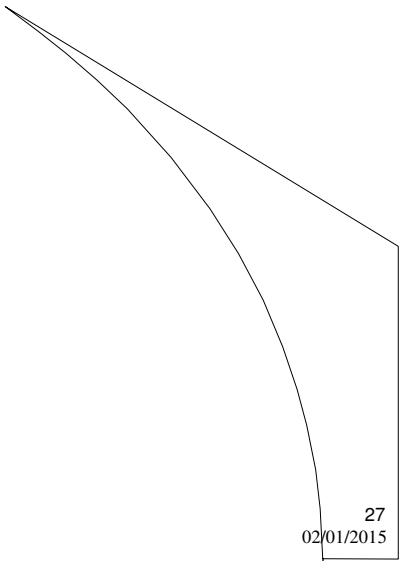
Caratteristiche delle transazioni

Alcune caratteristiche importanti sono:

- se il DBMS riscontra internamente qualche anomalia, o al riavvio dopo un improvviso black-out, esegue *automaticamente* una **ROLLBACK** delle transazioni in corso al momento dell'interruzione;.
- le transazioni sono implementate su un'apposita area d'appoggio del disco fisso in cui vengono copiati i dati originali *prima di essere modificati*. Quando viene eseguita una **COMMIT**, i dati originali copiati vengono eliminati, mentre se viene eseguita una **ROLLBACK**, si ricopiano indietro i dati originali copiati.
- una possibile causa del fallimento di una transazione è *l'insufficienza di spazio d'appoggio* per copiare i dati originali.

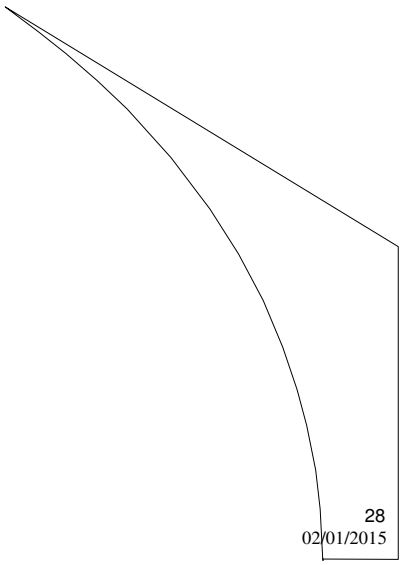
Esempi di transazioni in SQL

```
BEGIN  TRANSACTION;
UPDATE ContoCorrente
   SET Saldo = Saldo + 100
   WHERE NumeroConto = 12345 ;
UPDATE ContoCorrente
   SET Saldo = Saldo - 100
   WHERE NumeroConto = 54321 ;
COMMIT WORK;
```



Esempi di transazioni in SQL

```
BEGIN TRANSACTION;
DELETE FROM ElencoChiamate
        WHERE CodChiamata = '0';
INSERT INTO ElencoChiamate
        VALUES ('S', 'order status');
COMMIT WORK;
```



Argomenti

- Controllo dell'accesso
- Privilegi
- Tipi di privilegi offerti da SQL
- Comandi relativi ai privilegi
- Comando **GRANT**
- Comando **REVOKE**
- Concetto di transazione
- Proprietà delle transazioni
  - atomicità
  - consistenza
  - isolamento
  - durabilità
- Transazioni in SQL
- Terminazione di transazioni
  - corretta
  - non corretta
- Operare transazioni in SQL
- COMMIT e ROLLBACK
- Utilizzo di transazioni in SQL
- Caratteristiche delle transazioni
- Esempi di transazioni in SQL

Altre fonti di informazione

- Atzeni, Ceri, Paraboschi, Torlone, Basi di dati - McGraw-Hill, 1996-2002