Transazioni

In informatica, una transazione è una sequenza di operazioni, che può concludersi con un successo o un insuccesso; in caso di successo, il risultato delle operazioni deve essere permanente, mentre in caso di insuccesso si deve tornare allo stato precedente all'inizio della transazione. Una transazione, per essere tale, deve godere delle cosiddette proprietà **ACID,** particolarmente significative nei sistemi in cui possono essere eseguite più transazioni contemporaneamente. ACID è un acronimo per: ***A**tomicity, **C**onsistency, **I**solation, e **D**urability (**A**tomicità, **C**onsistenza, **I**solamento e **D**urabilità)*.

atomicità: la transazione è indivisibile nella sua esecuzione e la sua esecuzione deve essere o totale o nulla, non sono ammesse esecuzioni intermedie.

consistenza: quando inizia una transazione il database si trova in uno stato consistente e quando la transazione termina il database deve essere in uno stato consistente, ovvero non deve violare eventuali vincoli di integrità, quindi non devono verificarsi contraddizioni (inconsistency) tra i dati archiviati nel DB.

isolamento: ogni transazione deve essere eseguita in modo isolato e indipendente dalle altre transazioni, l'eventuale fallimento di una transazione non deve interferire con le altre transazioni in esecuzione.

durabilità: detta anche persistenza, si riferisce al fatto che una volta che una transazione ha richiesto un commit work, i cambiamenti apportati non dovranno essere più persi. Per evitare che nel lasso di tempo fra il momento in cui la base di dati si impegna a scrivere le modifiche e quello in cui li scrive effettivamente si verifichino perdite di dati dovuti a malfunzionamenti, vengono tenuti dei registri di log, dove sono annotate tutte le operazioni sul DB.

Un utilizzo tipico delle transazioni è il seguente:

1) Prima di eseguire una transazione, si esegue un'istruzione di "inizio transazione".
2) Si eseguono le operazioni di interrogazione e modifica dei dati.
3) Se si riscontra qualche anomalia, si esegue un'istruzione detta di "rollback", per abortire la transazione.
4) Se si sono eseguite tutte le operazioni senza riscontrare anomalie, si esegue un'istruzione detta di "commit", per confermare la transazione.

Se il DBMS riscontra internamente qualche anomalia, esegue automaticamente una rollback. Se il DBMS stesso termina bruscamente, per intervento esterno, o per un bug, o per spegnimento improvviso del computer, il DBMS, quando viene riattivato, esegue automaticamente la rollback delle transazioni che erano in corso al momento del crash.

Per implementare una transazione, tipicamente si usa un'apposita area d'appoggio del disco fisso in cui vengono copiati i dati originali appena prima di essere modificati. Quando viene eseguita una commit, i dati originali copiati vengono eliminati. Quando 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.

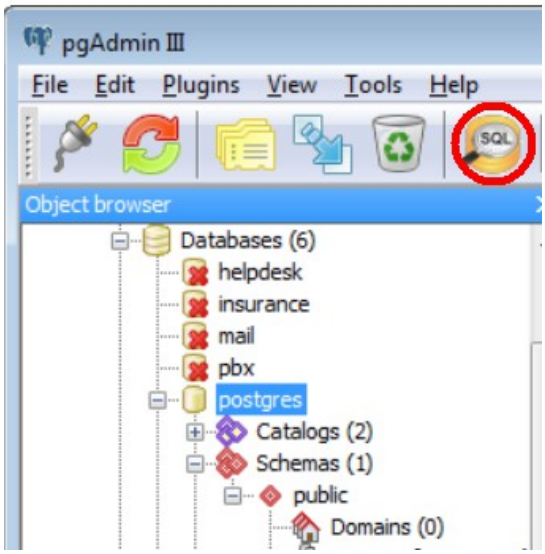
Esempio di transazione: PRELEVAMENTO E VERSAMENTO:

```
/* DROP TABLE conto */

CREATE TABLE conto (
                        codcli char(4) primary key,
                        saldo float
                    );
INSERT INTO  conto   VALUES ('A023',350.5);
INSERT INTO  conto   VALUES ('A024',875.0);
INSERT INTO  conto   VALUES ('A025',-100.0);



/* ESERCIZIO 1 Transazione corretta **********/
BEGIN TRANSACTION;

UPDATE conto SET saldo = saldo - 100.00
   WHERE codcli = 'A023';


UPDATE conto SET saldo = saldo + 100.00
   WHERE codcli = 'A024';
COMMIT;                          /* è opzionale in postgres ****/
END TRANSACTION;

/* ESERCIZIO 2 Transazione annullata **********/

BEGIN TRANSACTION;
```

```
UPDATE conto SET saldo = saldo - 100.00
   WHERE codcli = 'A023';

UPDATE conto SET saldo = saldo + 100.00
   WHERE codcli = 'A099';  /* INESISTENTE */
/* mi accorgo che il codice A99 non esiste, ad esempio la pg_exec in PHP fallisce */
ROLLBACK;              /* … rimetto tutto come prima ***/
END TRANSACTION;
```

Abbiamo già provato centinaia di volte il meccanismo di transazione, magari senza saperlo.

Esempio: quando eseguiamo un file .sql da PgAdmin (ad esempio la create table e le insert della tabella articoli)



come fa il programma PgAdmin a lasciare la base di dati invariata, se incontra un errore nel file sql, magari dopo aver eseguito un centinaio di istruzioni sql?

Risposta: **agisce in transazione** ed al primo errore richiama **ROLLBACK** e la base di dati ritorna nello stato iniziale, stato che aveva prima del lancio dell'intero file .sql.