Linguaggi di Programmazione per il Web – Parte 6

Estensione MySQLi

Le Transazioni

Autore

Prof. **Rio Chierego**
riochierego@libero.it

Siti Utili

http://www.riochierego.it/mobile

http://www.html.it/

http://www.mrwebmaster.it

http://www.php.net/

http://php.html.it/

http://mysql.it

Estensione MySQLi: Le transazioni

- Una **transazione** è una successione di query che si conclude con un successo o un insuccesso. Nel primo caso gli effetti prodotti dalle query diventano permanenti, altrimenti il database torna nello stato precedente l'inizio della transazione.

N.B. Le tabelle di tipo MyISAM non supportano le transazioni, possibili in MySQL solo con tabelle di tipo InnoDB e BDB.

- La vecchia **estensione nativa mysql non offre un supporto** nativo **alle transazioni** che devono essere eseguite utilizzando direttamente delle query. Questo approccio è ovviamente ancora praticabile, pertanto risulta immediato convertire un vecchio script per l'utilizzo di **mysqli (mysql improved)**. Vediamo quindi come effettuare una transazione facendo uso esclusivamente di query.

Estensione MySQLi: Le transazioni

Di default **MySQL** funziona in modalità **AUTOCOMMIT**, ovvero tutte le query che modificano il contenuto del database (esempio **INSERT**, **DELETE** ed **UPDATE**) hanno un effetto immediato, duraturo ed non possono essere annullate.

Per effettuare una transazione:

- occorre disabilitare l'AUTOCOMMIT ed utilizzare i comandi **COMMIT** e **ROLLBACK** per confermare o annullare gli effetti delle query eseguite;

oppure in alternativa

- lasciando abilitato l'AUTOCOMMIT, **è possibile lanciare il comando START TRANSACTION** (o il suo alias **BEGIN TRANSACTION**) per indicare l'inizio della transazione ed impiegare i i comandi **COMMIT** e **ROLLBACK** per confermare o annullare gli effetti delle query eseguite in caso di successo o di insuccesso.

E' possibile anche utilizzare dei punti di ripristino (SAVEPOINT) definiti arbitrariamente dal programmatore ai quali è possibile ritornare selettivamente mediante un *ROLLBACK*

Estensione MySQLi: Le transazioni

E' possibile anche utilizzare dei punti di ripristino (**SAVEPOINT**) definiti arbitrariamente dal programmatore ai quali è possibile ritornare selettivamente mediante un *ROLLBACK.*

Vediamo di seguito come creare dei *savepoint* e come effettuare *rollback* selettivi:

```
START TRANSACTION;
... primo blocco di istruzioni ...
SAVEPOINT p1;
... secondo blocco di istruzioni ...
ROLLBACK TO SAVEPOINT p1;
 ...terzo blocco di istruzioni...
COMMIT;
```

Nel nostro esempio viene creato un *savepoint* dopo un primo blocco di istruzioni, il secondo blocco, invece, è seguito da un *rollback* che lo annulla riportando la situazione al *savepoint* "p1".
Infine abbiamo un terzo blocco di istruzioni seguito da un *COMMIT*. In pratica le modifiche apportate dal primo e terzo blocco saranno effettive, mentre quelle del secondo blocco no.

Estensione MySQLi: LE TRANSAZIONI

Approccio Procedurale

START TRANSACTION: approccio procedurale 1/2

```
<?php
// provo a connettermi al server MySQL
$connessione = mysqli_connect ('localhost', 'root', 'password', 'db_name');

// Avvio la transazione
$risultato = mysqli_query ($connessione, 'START TRANSACTION');

// eseguo alcune query
$risultato = mysqli_query ($connessione, "INSERT INTO Fornitore VALUES ('TR-1',
    'Verdi', 'Asdrubale', '1975/08/09');"
$risultato = mysqli_query ($connessione, "UPDATE Fornitore SET Cognome = 'HAPPY
    HIPPO' WHERE Nome='Asdrubale';");

// chiudo la transazione con esito positivo ...eseguo il COMMIT
$risultato = mysqli_query ($connessione, 'COMMIT');
```

START TRANSACTION: approccio procedurale 2/2

```
// Avvio una nuova transazione
$risultato = mysqli_query ($connessione, 'START TRANSACTION');
// eseguo una nuova query
$risultato = mysqli_query ($connessione, "DELETE FROM Fornitore;");
// affected_rows funziona anche con le transazioni
// anche se non è detto che l'effetto della query sarà duraturo
echo 'Righe cancellate: '. mysqli_affected_rows($risultato);
// questa volta chiudo la transazione con esito negativo forzando il ROLLBACK
$risultato = mysqli_query ($connessione, 'ROLLBACK');

// Chiudo la connessione al server MySQL
mysqli_close($connessione)
?>
```

SET AUTOCOMMIT = 0: approccio procedurale 1/2

```
<?php
// provo a connettermi al server MySQL
$connessione = mysqli_connect ('localhost', 'root', 'password', 'db_name');

// Avvio la transazione
$risultato = mysqli_query ($connessione, 'SET AUTOCOMMIT = 0;');

// eseguo alcune query
$risultato = mysqli_query ($connessione, "INSERT INTO Fornitore VALUES ('TR-1',
    'Verdi', 'Asdrubale', '1975/08/09');"
$risultato = mysqli_query ($connessione, "UPDATE Fornitore SET Cognome = 'HAPPY
    HIPPO' WHERE Nome='Asdrubale';");

// chiudo la transazione con esito positivo ...eseguo il COMMIT
$risultato = mysqli_query ($connessione, 'COMMIT');
```

SET AUTOCOMMIT = 0: approccio procedurale 2/2

```
// Adesso inizia implicitamente una nuova transazione
// poiché l'AUTOCOMMIT è disabilitato

// eseguo una query
$risultato = mysqli_query ($connessione, "DELETE FROM Fornitore;");

// chiudo la transazione con esito negativo
$risultato = mysqli_query ($connessione, 'ROLLBACK');

// Chiudo la connessione al server MySQL
mysqli_close($connessione);
?>
```

Estensione MySQLi: LE TRANSAZIONI

Approccio Object-Oriented

by Prof. Rio Chierego

START TRANSACTION: approccio O-O 1/2

```
<?php
// provo a connettermi al server MySQL
$mysqli = new mysqli('localhost', 'root', 'password', 'db_name');

// Avvio la transazione
$mysqli->query('START TRANSACTION');

// eseguo le stesse query di prima
$mysqli->query ("INSERT INTO Fornitore VALUES ('TR-1', 'Verdi', 'Asdrubale',
    '1975/08/09');"
$mysqli->query ("UPDATE Fornitore SET Cognome = 'HAPPY HIPPO' WHERE
    Nome='Asdrubale';");

// chiudo la transazione con esito positivo
$mysqli->query('COMMIT');
```

START TRANSACTION: approccio O-O 2/2

```
// Avvio una nuova transazione
$mysqli->query('START TRANSACTION');
// eseguo una query
$mysqli->query("DELETE FROM Fornitore;");
// affected_rows funziona anche con le transazioni
// anche se non è detto che l'effetto della query sarà duraturo
echo 'Righe cancellate:', $mysqli->affected_rows;
// chiudo la transazione con esito negativo
$mysqli->query('ROLLBACK');
// chiudo la connessione
$mysqli->close();
?>
```

SET AUTOCOMMIT = 0: approccio O-O 1/2

```
<?php
// provo a connettermi al server MySQL
$mysqli = new mysqli('localhost', 'root', 'password', 'db_name');
// Disabilito l'AUTOCOMMIT delle query
$mysqli->query('SET AUTOCOMMIT=0');
// eseguo le stesse query di prima
$mysqli->query ("INSERT INTO Fornitore VALUES ('TR-1', 'Verdi', 'Asdrubale',
    '1975/08/09');");
$mysqli->query ("UPDATE Fornitore SET Cognome = 'HAPPY HIPPO' WHERE
    Nome='Asdrubale';");
// chiudo la transazione con esito positivo
$mysqli->query('COMMIT');
```

SET AUTOCOMMIT = 0: approccio O-O 2/2

```
// Adesso inizia implicitamente una nuova transazione
// poiché l'AUTOCOMMIT è disabilitato

// eseguo una query
$mysqli->query("DELETE FROM Fornitore");

// chiudo la transazione con esito negativo
$mysqli->query('ROLLBACK');

// chiudo la connessione
$mysqli->close();
?>
```

Uso dei metodi specifici della classe MySQLi: approccio O-O 1/2

```
<?php
// provo a connettermi al server MySQL
$mysqli = new mysqli('localhost', 'root', 'password', 'db_name');
// Disabilito l'AUTOCOMMIT delle query
$mysqli->autocommit (false);
// eseguo alcune query
$mysqli->query("INSERT INTO mia_tabella VALUES (NULL, 'Alberto', 'Rossi');");
$mysqli->query("UPDATE mia_tabella SET cognome='Bianchi' WHERE nome='Alberto‘ ;");
// chiudo la transazione con esito positivo
$mysqli->commit ();
```

Uso dei metodi specifici della classe MySQLi: approccio O-O 2/2

```
// Adesso inizia implicitamente una nuova transazione
// poiché l'AUTOCOMMIT è disabilitato
// eseguo una query
$mysqli->query("DELETE FROM Fornitore;");
// chiudo la transazione con esito negativo
$mysqli->rollback();
?>
```

Nota Bene

Anche se non ci sono importanti benefici provenienti dall'uso dei metodi autocommit(), commit() e rollback(), il codice risulta comunque più chiaro ed evidenzia esclusivamente le query appartenenti alla transazione.