

Transazioni con PHP - MySQLi

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.

Le tabelle di tipo MyISAM non supportano le transazioni, possibili in MySQL solo con tabelle di tipo InnoDB e BDB. Per i nostri esempi useremo sempre la tabella `mia_tabella` costruita nel [primo articolo](#), convertendola però in InnoDB attraverso la seguente query:

```
ALTER TABLE mia_tabella type = InnoDB
```

La vecchia estensione `ext/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 `ext/mysql`. Vediamo quindi come effettuare una transazione facendo uso esclusivamente di query.

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

Per effettuare una transazione è possibile disabilitare l'AUTOCOMMIT ed utilizzare i comandi COMMIT e ROLLBACK per confermare o annullare gli effetti delle query eseguite. Oppure, lasciando l'impostazione AUTOCOMMIT abilitata, è possibile lanciare il comando START TRANSACTION (o il suo alias BEGIN) per indicare l'inizio della transazione ed impiegare i comandi COMMIT e ROLLBACK per confermare o annullare gli effetti delle query eseguite.

Vediamo un esempio che avvia una transazione con il comando START TRANSACTION:

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

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

// 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->query('COMMIT');
```

```

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

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

// 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');

?>

```

Adesso vediamo invece uno script che implementa le transazioni disabilitando l'AUTOCOMMIT:

```

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

// Disabilito l'AUTOCOMMIT delle query
$mysqli->query('SET AUTOCOMMIT=0');

// 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->query('COMMIT');

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

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

```

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

?>
```

Con l'estensione *ext/mysqli* abbiamo a disposizione i metodi `commit()` e `rollback()` della classe *mysqli* che, insieme al metodo `autocommit()` permettono di eseguire delle transazioni inviando al server esclusivamente le query costituenti la transazione stessa. La logica da seguire coincide con quella dell'ultimo esempio visto poiché questi metodi hanno la stessa valenza dei relativi comandi SQL:

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

// 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();

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

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

// chiudo la transazione con esito negativo
mysqli->rollback();

?>
```

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