Transaction Tracking worksheet.

You are working on a program

1. BEGIN the Transaction.
2. Update, insert or delete entries in the.
3. If you like the changes to the database, then you COMMIT to the Transaction.
4. If you do not like the changes then you ROLLBACK the changes to the original condition of the database.

Note: You must use InnoDB type tables or Transactions, will not work.

Lets create a basic table in your database called "trans", and make the table type "innodb".

CREATE TABLE trans  
(  
id int not null auto\_increment,  
item varchar(30) not null,  
quantity varchar(10) not null,  
primary key(id)   
)type=innodb;

Heres the construction of the table:

mysql> DESC trans;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| id | int(11) |  | PRI | NULL | auto\_increment |
| item | varchar(30) |  |  |  |  |
| quantity | varchar(10) |  |  |  |  |

3 rows in set (0.00 sec)

Insert some data into the table:

mysql> INSERT INTO trans (id,item,quantity) VALUES (NULL,'computer’,'5');

Query OK, 1 row affected (0.00 sec)

Begin the Transaction by using the BEGIN command and update the entry:

mysql> BEGIN;

Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE trans SET quantity ='4' WHERE id=1;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

View the results of the update:

mysql> SELECT \* FROM trans;

|  |  |  |
| --- | --- | --- |
| **id** | **Item** | **Quantity** |
| 1 | Computer | 4 |

1 row in set (0.00 sec)

If you do not like the changes, then use the ROLLBACK command to revert back to the original version of the table.

mysql> ROLLBACK;

Query OK, 0 rows affected (0.00 sec)

Notice the table has reverted back to the original insertion:

mysql> SELECT \* FROM trans;

|  |  |  |
| --- | --- | --- |
| **id** | **item** | **Quantity** |
| 1 | Computer | 5 |

1 row in set (0.01 sec)

Lets update the table using another Transaction and commit to the changes:

mysql> BEGIN;

Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE trans SET quantity ='2' WHERE id=1;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT \* FROM trans;

|  |  |  |
| --- | --- | --- |
| **id** | **item** | **quantity** |
| 1 | Computer | 2 |

1 row in set (0.00 sec)

mysql> COMMIT;

Query OK, 0 rows affected (0.00 sec)

After you use the COMMIT command, the table will take on the changes and remain that way until they are modified.

mysql> SELECT \* FROM trans;

|  |  |  |
| --- | --- | --- |
| **id** | **item** | **quantity** |
| 1 | Computer | 2 |

1 row in set (0.00 sec)

Now that you understand the basics of Transactions lets create a PHP script that will insert new data into the table.

Here is the code for the transaction script:

<?php  
// trans.php  
function begin()   
{  
@mysql\_query("BEGIN");  
}  
function commit()  
{  
@mysql\_query("COMMIT");  
}  
function rollback()  
{  
@mysql\_query("ROLLBACK");  
}  
@mysql\_connect("localhost","username", "password") or die(mysql\_error());  
@mysql\_select\_db("test") or die(mysql\_error());  
$query = "INSERT INTO trans (id,item,quantity)  
values (null,'Baseball',4)";  
begin(); // transaction begins  
$result = @mysql\_query($query);  
if(!$result)  
{  
rollback(); // transaction rolls back  
echo "you rolled back";  
exit;  
}  
else   
{  
commit(); // transaction is committed  
echo "your insertion was successful";  
}  
?>

**Explanation of the script**

1. Functions are created for the BEGIN, COMMIT and ROLLBACK commands.
2. The script connects to the server and runs the query of inserting data into the table.
3. If the query is successful then it COMMITS the Transaction.
4. If the query is unsuccessful then the Transaction will ROLLBACK.

Here is the table after the script executes:

mysql> SELECT \* FROM trans;

|  |  |  |
| --- | --- | --- |
| **id** | **item** | **quantity** |
| 1 | Computer | 2 |
| 2 | Baseball | 4 |

2 rows in set (0.00 sec)