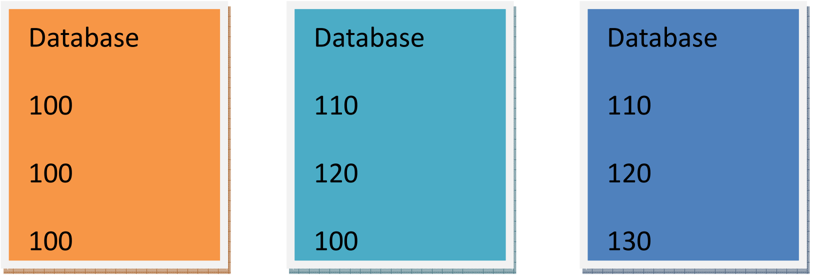
Database Tutorials

Hello everyone… welcome to database tutorials. These are going to be very basic tutorials about using the database to create simple applications, hope you enjoy it. If you have any notes about it, please send them to **notes@mka-soft.com.** Finally if you find these tutorials are useful, it would be nice from you to send a small donation via PayPal to **donation@mka-soft.com**.

The work with this tutorial started on 2011-FEBREWARY-26.

Understanding Transactions

This tutorial is about Transactions. Basically transactions are a number of sql commands that when executed change the database from one consistent state into another. Consider the following example:



Database	Database	Database
100	110	110
100	120	120
100	100	130

Each box represents the same database at some point in time. The one on the most left is your database in its original state – its consistent state. Each number represents a salary for someone. What you want to do is add 10 to the first salary, add 20 to the second salary, and 30 to the third salary so that the final database state would be the box on the right. To do this the update commands should work on the rows, and this update is not done it one step, instead it happens in stages one after another. The middle box shows how might the database look like during the execution.

Now the box in the middle represent an inconsistent state of the database since all the records should be increased, and it its current state the last record is not increased yet. So the transaction is happening between the box on the left and the box on the right so that it changes the database from the one status (the box on the left) to another status (the box on the right). The box in the middle is just an intermediate state that happens during the execution of a transaction.

Now why do we use them? Basically because transactions either get executed to the end or they don't. In other words the database status is always consistent if we use transactions. Let us say for the example you have a database having account balances for people in a bank. And your code should calculate the interest for each account. If you have 60,000 customers, and you are updating their information, and while you are updating the 30,000 account, an error happens in the system. This will leave your database with 30,000 people having an interest calculated for them, and the rest did not get any interest. If you are using transaction, then what happens is that all the updates are rolled back, and the account balances are they way they were before the transaction started. In other words in case of an error, a transaction rollbacks to the latest consistent database status.

Transactions also allows for multiple access of database table in a shared environment. So if you are working on a table updating it, others could still see the table before you started updating it – unless you want to lock it – and when you finish the transaction people will see updates.

One last thing about transactions is that it doesn't have to be an error for a rollback to happen. You can write a completely correct code that runs successfully, and instead of saving the result you could cancel everything by issuing a rollback command. Transactions gives you the option to either commit a transaction or rollback whenever you want.

The best way to understand transaction is by running actual code. The source file is included, just modify the database path in the connection string, and try different combinations and experiment with what is happening:

Test A:

1- Check values using access

2- Open the vb program

3- open the connection

4- Begin a transaction

5- Perform an update

6- Read values

7- Check values using access

8- Commit

9- Check value in access.

Test B:

1- Check values using access

2- Open the vb program

3- open the connection

4- Begin a transaction

5- Perform an update

6- Read values

7- Check values using access

8- Commit

9- Read values

http://www.mka-soft.com

Next tutorial we start working with insert, update and delete and see perform actual transactions with them. So this will be all for today. If you have questions or notes, send them to notes@mka-soft.com.

Thank you.

mkaatr