Commit-Rollback

Ctrl-z?

Transaction

It is often wise to consider changes to a database as an atomic action that either is successful or fails miserably.

This is sometimes called transactions - we start a transaction and if everything we wanted to do seems right, we commit to that transaction and make the changes permanent (so to speak).

If we are not pleased with the whole, we can instead roll back to the previous state of the database.

Artie Shaw (perhaps playing Begin the Beguine)

BEGIN TRANSACTION - ROLLBACK

```
sqlite> BEGIN TRANSACTION;
sqlite> UPDATE municipalities SET HTTPS=0;
sqlite> SELECT HTTPS, count(*) FROM municipalities GROUP BY HTTPS;
0|290
sqlite> -- Oh crap. I forgot the WHERE...
sqlite> ROLLBACK;
sqlite> SELECT HTTPS, count(*) FROM municipalities GROUP BY HTTPS;
0|222
1|68
sqlite> -- Pheeeew!
```

https://www.flickr.com/photos/eepaul/12991719735 Paul Wilkinson CC-BY

COMMIT

```
sqlite> BEGIN TRANSACTION;
sqlite> UPDATE municipalities SET HTTPS=0 WHERE name='Ale kommun';
sqlite> SELECT HTTPS, count(*) FROM municipalities GROUP BY HTTPS;
0|223
1|67
sqlite> COMMIT;
```

What about JDBC?

In order to use commit and rollback in jdbc, the connection must not do autocommits.

```
con.setAutoCommit(false);
```

With autocommit set to false, we can use the connection to either commit or rollback a transaction:

```
con.commit(); // Commit current transaction

con.rollback(); // Rollback current transaction
```

Example Java code

```
MunicipalityDB db = new MyMunicipalities();
db.setAutoCommit(false);
String name="' or 1=1;--";
System.out.println("Setting HTTPS to false for the name: " + name);
int numRowsAffected = db.updateHTTPSbyName(name, false);
if(numRowsAffected == 1){
  System.out.println("One row affected. Commit.");
  db.commit();
}else{
  System.out.println(numRowsAffected + " rows affected. Not good. Rollback!");
  db.rollback();
}
```

```
$ javac */*/*.java && java -cp .:driver/sqlite-jdbc-3.8.11.2.jar db.main.TierMain
Setting HTTPS to false for the name: ' or 1=1;--
DEBUG: SQL: UPDATE municipalities SET HTTPS=0 WHERE name='' or 1=1;--'
290 rows affected. Not good. Rollback!
Issuing rollback
```

Example Java code

```
MunicipalityDB db = new MyMunicipalities();
db.setAutoCommit(false);
String name="Ale kommun";
System.out.println("Setting HTTPS to false for the name: " + name);
int numRowsAffected = db.updateHTTPSbyName(name, false);
if(numRowsAffected == 1){
  System.out.println("One row affected. Commit.");
  db.commit();
}else{
  System.out.println(numRowsAffected + " rows affected. Not good. Rollback!");
  db.rollback();
}
```

```
$ javac */*/*.java && java -cp .:driver/sqlite-jdbc-3.8.11.2.jar db.main.TierMain
Setting HTTPS to false for the name: Ale kommun
DEBUG: SQL: UPDATE municipalities SET HTTPS=0 WHERE name='Ale kommun'
One row affected. Commit.
Issuing commit
```