Sample Transaction Processing Subroutine

The programming example below **(TESTTRANS)** demonstrates transaction processing. The purpose is to write (and/or delete) 10 rows, with record IDs from **90** to **99**, in the **CUSTOMERS** table. (You can use the **CUSTOMERS** table in the **EXAMPLES** application, if you wish.) Before each row is written, it is locked. Only if each row is successfully locked, written, and unlocked will all the rows be committed to the database. If the **FAIL** argument is passed with a value of 1, row 96 is locked before the loop writing the 10 rows, causing the write to row 96 to fail. This will cause the entire transaction to be rolled back, so that none of the 10 rows is written to the database, thus returning the database to its condition before the transaction started.

If passed with the **DEL** argument, the 10 rows are deleted as a group, unless the **FAIL** argument is passed with a value of **1**.

Create the **TESTTRANS** stored procedure by pasting the code below. See the comments for how to run this stored procedure.

```
compile subroutine testtrans(branch,fail)
declare subroutine transact, control_on, control_off
* run from the exec line.
* write records with no fail - run testtrans "WRITE",0 - this should create 10 new records
* write records with fail - run testtrans "WRITE",1 - this should fail and rollback
* delete records no fail - run testtrans "DEL",0 - should delete 10 records
* delete records with fail - run testtrans "DEL", 1,  - shoud fail and rollback all records.
control_on("CUSTOMERS" , 0 ) /* turn control on for the table */
error = 0
open "CUSTOMERS" to datafile then
sqlstate = ''
transact( 2 , sqlstate)    /*  begin transaction */
if fail = 1 then
  lock datafile,96 then
  * cool, we are holding a lock to make the batch of transactions fail.
  end
end
for i = 90 to 99
  lock datafile,i else
    error = 1
  end
  while error = 0
    if branch = "WRITE" then
      write "THIS IS A TEST" on datafile,i else
      error = 1
    end
  end else
    delete datafile,i else
    error = 1
  end
end
unlock datafile,i else
  error = 1
end
next i
branch = error
if error = 0 then
  * commit transaction
  transact( 1 , sqlstate) /* commit */
  end else
  * rollback
    transact( 0 , sqlstate) /* rollback */
  end
end
if fail = 1 then
  unlock datafile,96 then
  * cool, unlock as required.
  end
end
control_off("CUSTOMERS" , 0) /* control is turned off */
return
```