UnLock Statement

Description

Releases a lock set by the station in a previous Lock statement. A table or row can only be locked again (by another user, for example) after the UnLock statement has been issued.

Syntax

UnLock *[table_var* | All | *Cursor cursor_var*] [,*key [,locktype*]] Then | *Else statements*

Parameters

The UnLock statement accepts arguments for the following parameters.

Parameter	Description
table_var	Refers to a table variable that has been locked in a previous Lock statement.
cursor_var	Contains a cursor variable. Cursor variables are initialized with a Select...By statement.
key	If a row is to be unlocked, key specifies which row within a table is to be unlocked. To unlock a previously locked table, pass a null in key. To UnLock all locks previously set by this workstation during the current logon session, use the UnLock All statement.
locktype	Specifies the type of unlocking that is to take place. A locktype can be passed as a mnemonic code (not a literal) or as a two-digit code. For details on the possible values for locktype, refer to the Lock statement. ***Note: The code n0 is used to indicate an UnLock All.***
Then	The statement(s) following Then are executed if the file or record is unlocked successfully.
Else	The statement(s) following Else are executed if the file or record cannot be unlocked. The Status() function indicates the severity of the error, and the system variable @FILE_ERROR contains details about the nature of the error.

Remark

Use UnLock to release each lock. Logging off the system will release all locks.

See also

Lock

Example

```
/* The following function fragment locks a record, then reads it,
does some unspecified processing, and then unlocks the record, before concluding. */

Equate TRUE$  To 1
Equate FALSE$ To 0
table = "SAMPLE_CUSTOMERS"
key = 8
status = ""
Open table To tablevar Then
  Locked = FALSE$
  Loop
    Lock tablevar, key Then
      locked = TRUE$
    End Else
      /* set up delay for network server access */
      For ctr = 1 To 1000
      Next
    End
  Until Locked Repeat
  * we're locked, now read
  Read @RECORD From tablevar,key Then
    /* do record processing here */
  End
  * now, unlock the record we've written back
  UnLock tablevar,key Else
    * error processing here
  End  ;* successfully unlocked
End Else  ;* table fails to open
  status = Set_FSError()
End  ;* table taken care of

Return status
```