NAME

dblink_open - opens a cursor in a remote database

SYNOPSIS

dblink_open(text cursorname, text sql [, bool fail_on_error]) returns text
dblink_open(text connname, text cursorname, text sql [, bool fail_on_error]) returns text

DESCRIPTION

dblink_open() opens a cursor in a remote database. The cursor can subsequently be manipulated with **dblink_fetch()** and **dblink_close()**.

ARGUMENTS

connname

Name of the connection to use; omit this parameter to use the unnamed connection.

cursorname

The name to assign to this cursor.

sql

The **SELECT** statement that you wish to execute in the remote database, for example select * from pg_class.

fail_on_error

If true (the default when omitted) then an error thrown on the remote side of the connection causes an error to also be thrown locally. If false, the remote error is locally reported as a NOTICE, and the function’s return value is set to ERROR.

RETURN VALUE

Returns status, either OK or ERROR.

NOTES

Since a cursor can only persist within a transaction, **dblink_open** starts an explicit transaction block (**BEGIN**) on the remote side, if the remote side was not already within a transaction. This transaction will be closed again when the matching **dblink_close** is executed. Note that if you use **dblink_exec** to change data between **dblink_open** and **dblink_close**, and then an error occurs or you use **dblink_disconnect** before **dblink_close**, your change *will be lost* because the transaction will be aborted.

EXAMPLES

```
SELECT dblink_connect(’dbname=postgres options=-csearch_path=’);
```

```
 dblink_connect
----------------
 OK
(1 row)

SELECT dblink_open('foo', 'select proname, prosrc from pg_proc');
 dblink_open
-------------
 OK
(1 row)
```