

NAME

`dblink_fetch` – returns rows from an open cursor in a remote database

SYNOPSIS

`dblink_fetch(text cursorname, int howmany [, bool fail_on_error])` returns setof record

`dblink_fetch(text connname, text cursorname, int howmany [, bool fail_on_error])` returns setof record

DESCRIPTION

`dblink_fetch` fetches rows from a cursor previously established by **`dblink_open`**.

ARGUMENTS

connname

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

cursorname

The name of the cursor to fetch from.

howmany

The maximum number of rows to retrieve. The next *howmany* rows are fetched, starting at the current cursor position, moving forward. Once the cursor has reached its end, no more rows are produced.

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 returns no rows.

RETURN VALUE

The function returns the row(s) fetched from the cursor. To use this function, you will need to specify the expected set of columns, as previously discussed for **`dblink`**.

NOTES

On a mismatch between the number of return columns specified in the FROM clause, and the actual number of columns returned by the remote cursor, an error will be thrown. In this event, the remote cursor is still advanced by as many rows as it would have been if the error had not occurred. The same is true for any other error occurring in the local query after the remote **FETCH** has been done.

EXAMPLES

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

```
dblink_connect
```

```
-----
```

```
OK
```

```
(1 row)
```

```
SELECT dblink_open('foo', 'select proname, prosrc from pg_proc where proname like "bytea%");
```

```
dblink_open
```

```
-----
```

```
OK
```

```
(1 row)
```

```
SELECT * FROM dblink_fetch('foo', 5) AS (funcname name, source text);
```

```
funcname | source
```

```
-----+-----
```

```
byteacat | byteacat
```

```
byteacmp | byteacmp
```

```
byteaeq  | byteaeq
```

```
byteage  | byteage
```

```
byteagt  | byteagt
```

```
(5 rows)
```

```
SELECT * FROM dblink_fetch('foo', 5) AS (funcname name, source text);
```

funcname	source
byteain	byteain
byteale	byteale
bytealike	bytealike
bytealt	bytealt
byteane	byteane

(5 rows)

SELECT * FROM dblink_fetch('foo', 5) AS (funcname name, source text);

funcname	source
byteanlike	byteanlike
byteaout	byteaout

(2 rows)

SELECT * FROM dblink_fetch('foo', 5) AS (funcname name, source text);

funcname	source
----------	--------

(0 rows)