NAME

dblink_build_sql_insert - builds an INSERT statement using a local tuple, replacing the primary key field values with alternative supplied values

SYNOPSIS

```
dblink_build_sql_insert(text relname,
                        int2vector primary_key_attnums,
                        integer num_primary_key_atts,
                        text[] src_pk_att_vals_array,
                        text[] tgt_pk_att_vals_array) returns text
```

DESCRIPTION

dblink_build_sql_insert can be useful in doing selective replication of a local table to a remote database. It selects a row from the local table based on primary key, and then builds an SQL **INSERT** command that will duplicate that row, but with the primary key values replaced by the values in the last argument. (To make an exact copy of the row, just specify the same values for the last two arguments.)

ARGUMENTS

relname

Name of a local relation, for example foo or myschema.mytab. Include double quotes if the name is mixed-case or contains special characters, for example "FooBar"; without quotes, the string will be folded to lower case.

primary_key_attnums

Attribute numbers (1-based) of the primary key fields, for example 1 2.

num_primary_key_atts

The number of primary key fields.

src_pk_att_vals_array

Values of the primary key fields to be used to look up the local tuple. Each field is represented in text form. An error is thrown if there is no local row with these primary key values.

tgt_pk_att_vals_array

Values of the primary key fields to be placed in the resulting **INSERT** command. Each field is represented in text form.

RETURN VALUE

Returns the requested SQL statement as text.

NOTES

As of PostgreSQL 9.0, the attribute numbers in *primary_key_attnums* are interpreted as logical column numbers, corresponding to the column's position in SELECT * FROM relname. Previous versions interpreted the numbers as physical column positions. There is a difference if any column(s) to the left of the indicated column have been dropped during the lifetime of the table.

EXAMPLES

```
SELECT dblink_build_sql_insert('foo', '1 2', 2, '{"1", "a"}', '{"1", "b''a"}');
          dblink_build_sql_insert
-------------------------------------------------
 INSERT INTO foo(f1,f2,f3) VALUES('1','b''a','1')
(1 row)
```