

NAME

ALTER_USER_MAPPING – change the definition of a user mapping

SYNOPSIS

```
ALTER USER MAPPING FOR { user_name | USER | CURRENT_ROLE | CURRENT_USER | SESSION_USER | PUBLIC }  
    SERVER server_name  
    OPTIONS ( [ ADD | SET | DROP ] option ['value'] [, ... ] )
```

DESCRIPTION

ALTER USER MAPPING changes the definition of a user mapping.

The owner of a foreign server can alter user mappings for that server for any user. Also, a user can alter a user mapping for their own user name if USAGE privilege on the server has been granted to the user.

PARAMETERS

user_name

User name of the mapping. CURRENT_ROLE, CURRENT_USER, and USER match the name of the current user. PUBLIC is used to match all present and future user names in the system.

server_name

Server name of the user mapping.

```
OPTIONS ( [ ADD | SET | DROP ] option ['value'] [, ... ] )
```

Change options for the user mapping. The new options override any previously specified options.

ADD, SET, and DROP specify the action to be performed. ADD is assumed if no operation is explicitly specified. Option names must be unique; options are also validated by the server's foreign-data wrapper.

EXAMPLES

Change the password for user mapping bob, server foo:

```
ALTER USER MAPPING FOR bob SERVER foo OPTIONS (SET password 'public');
```

COMPATIBILITY

ALTER USER MAPPING conforms to ISO/IEC 9075–9 (SQL/MED). There is a subtle syntax issue: The standard omits the FOR key word. Since both CREATE USER MAPPING and DROP USER MAPPING use FOR in analogous positions, and IBM DB2 (being the other major SQL/MED implementation) also requires it for ALTER USER MAPPING, PostgreSQL diverges from the standard here in the interest of consistency and interoperability.

SEE ALSO

CREATE USER MAPPING (**CREATE_USER_MAPPING(7)**), DROP USER MAPPING (**DROP_USER_MAPPING(7)**)