

NAME

CREATE_FOREIGN_DATA_WRAPPER – define a new foreign-data wrapper

SYNOPSIS

```
CREATE FOREIGN DATA WRAPPER name
[ HANDLER handler_function | NO HANDLER ]
[ VALIDATOR validator_function | NO VALIDATOR ]
[ OPTIONS ( option 'value' [, ... ] ) ]
```

DESCRIPTION

CREATE FOREIGN DATA WRAPPER creates a new foreign-data wrapper. The user who defines a foreign-data wrapper becomes its owner.

The foreign-data wrapper name must be unique within the database.

Only superusers can create foreign-data wrappers.

PARAMETERS

name

The name of the foreign-data wrapper to be created.

HANDLER *handler_function*

handler_function is the name of a previously registered function that will be called to retrieve the execution functions for foreign tables. The handler function must take no arguments, and its return type must be fdw_handler.

It is possible to create a foreign-data wrapper with no handler function, but foreign tables using such a wrapper can only be declared, not accessed.

VALIDATOR *validator_function*

validator_function is the name of a previously registered function that will be called to check the generic options given to the foreign-data wrapper, as well as options for foreign servers, user mappings and foreign tables using the foreign-data wrapper. If no validator function or NO VALIDATOR is specified, then options will not be checked at creation time. (Foreign-data wrappers will possibly ignore or reject invalid option specifications at run time, depending on the implementation.) The validator function must take two arguments: one of type text[], which will contain the array of options as stored in the system catalogs, and one of type oid, which will be the OID of the system catalog containing the options. The return type is ignored; the function should report invalid options using the **ereport(ERROR)** function.

OPTIONS (*option* '*value*' [, ...])

This clause specifies options for the new foreign-data wrapper. The allowed option names and values are specific to each foreign data wrapper and are validated using the foreign-data wrapper's validator function. Option names must be unique.

NOTES

PostgreSQL's foreign-data functionality is still under active development. Optimization of queries is primitive (and mostly left to the wrapper, too). Thus, there is considerable room for future performance improvements.

EXAMPLES

Create a useless foreign-data wrapper dummy:

```
CREATE FOREIGN DATA WRAPPER dummy;
```

Create a foreign-data wrapper file with handler function file_fdw_handler:

```
CREATE FOREIGN DATA WRAPPER file HANDLER file_fdw_handler;
```

Create a foreign-data wrapper mywrapper with some options:

```
CREATE FOREIGN DATA WRAPPER mywrapper
```

OPTIONS (debug 'true');

COMPATIBILITY

CREATE FOREIGN DATA WRAPPER conforms to ISO/IEC 9075-9 (SQL/MED), with the exception that the HANDLER and VALIDATOR clauses are extensions and the standard clauses LIBRARY and LANGUAGE are not implemented in PostgreSQL.

Note, however, that the SQL/MED functionality as a whole is not yet conforming.

SEE ALSO

ALTER FOREIGN DATA WRAPPER (**ALTER_FOREIGN_DATA_WRAPPER(7)**), DROP FOREIGN DATA WRAPPER (**DROP_FOREIGN_DATA_WRAPPER(7)**), CREATE SERVER (**CREATE_SERVER(7)**), CREATE USER MAPPING (**CREATE_USER_MAPPING(7)**), CREATE FOREIGN TABLE (**CREATE_FOREIGN_TABLE(7)**)