

NAME

CREATE_OPERATOR_CLASS – define a new operator class

SYNOPSIS

```
CREATE OPERATOR CLASS name [ DEFAULT ] FOR TYPE data_type
  USING index_method [ FAMILY family_name ] AS
  { OPERATOR strategy_number operator_name [ ( op_type, op_type ) ] [ FOR SEARCH | FOR ORDER BY sort_family_name ]
    | FUNCTION support_number [ ( op_type [ , op_type ] ) ] function_name ( argument_type [ , ... ] )
    | STORAGE storage_type
  } [ , ... ]
```

DESCRIPTION

CREATE OPERATOR CLASS creates a new operator class. An operator class defines how a particular data type can be used with an index. The operator class specifies that certain operators will fill particular roles or “strategies” for this data type and this index method. The operator class also specifies the support functions to be used by the index method when the operator class is selected for an index column. All the operators and functions used by an operator class must be defined before the operator class can be created.

If a schema name is given then the operator class is created in the specified schema. Otherwise it is created in the current schema. Two operator classes in the same schema can have the same name only if they are for different index methods.

The user who defines an operator class becomes its owner. Presently, the creating user must be a superuser. (This restriction is made because an erroneous operator class definition could confuse or even crash the server.)

CREATE OPERATOR CLASS does not presently check whether the operator class definition includes all the operators and functions required by the index method, nor whether the operators and functions form a self-consistent set. It is the user's responsibility to define a valid operator class.

Related operator classes can be grouped into operator families. To add a new operator class to an existing family, specify the FAMILY option in **CREATE OPERATOR CLASS**. Without this option, the new class is placed into a family named the same as the new class (creating that family if it doesn't already exist).

Refer to Section 36.16 for further information.

PARAMETERS

name

The name of the operator class to be created. The name can be schema-qualified.

DEFAULT

If present, the operator class will become the default operator class for its data type. At most one operator class can be the default for a specific data type and index method.

data_type

The column data type that this operator class is for.

index_method

The name of the index method this operator class is for.

family_name

The name of the existing operator family to add this operator class to. If not specified, a family named the same as the operator class is used (creating it, if it doesn't already exist).

strategy_number

The index method's strategy number for an operator associated with the operator class.

operator_name

The name (optionally schema-qualified) of an operator associated with the operator class.

op_type

In an OPERATOR clause, the operand data type(s) of the operator, or NONE to signify a prefix operator. The operand data types can be omitted in the normal case where they are the same as the

operator class's data type.

In a FUNCTION clause, the operand data type(s) the function is intended to support, if different from the input data type(s) of the function (for B-tree comparison functions and hash functions) or the class's data type (for B-tree sort support functions, B-tree equal image functions, and all functions in GiST, SP-GiST, GIN and BRIN operator classes). These defaults are correct, and so *op_type* need not be specified in FUNCTION clauses, except for the case of a B-tree sort support function that is meant to support cross-data-type comparisons.

sort_family_name

The name (optionally schema-qualified) of an existing btree operator family that describes the sort ordering associated with an ordering operator.

If neither FOR SEARCH nor FOR ORDER BY is specified, FOR SEARCH is the default.

support_number

The index method's support function number for a function associated with the operator class.

function_name

The name (optionally schema-qualified) of a function that is an index method support function for the operator class.

argument_type

The parameter data type(s) of the function.

storage_type

The data type actually stored in the index. Normally this is the same as the column data type, but some index methods (currently GiST, GIN, SP-GiST and BRIN) allow it to be different. The STORAGE clause must be omitted unless the index method allows a different type to be used. If the column *data_type* is specified as anyarray, the *storage_type* can be declared as anyelement to indicate that the index entries are members of the element type belonging to the actual array type that each particular index is created for.

The OPERATOR, FUNCTION, and STORAGE clauses can appear in any order.

NOTES

Because the index machinery does not check access permissions on functions before using them, including a function or operator in an operator class is tantamount to granting public execute permission on it. This is usually not an issue for the sorts of functions that are useful in an operator class.

The operators should not be defined by SQL functions. An SQL function is likely to be inlined into the calling query, which will prevent the optimizer from recognizing that the query matches an index.

EXAMPLES

The following example command defines a GiST index operator class for the data type `_int4` (array of `int4`). See the `intarray` module for the complete example.

```
CREATE OPERATOR CLASS gist__int4_ops
  DEFAULT FOR TYPE _int4 USING gist AS
  OPERATOR      3      &&,
  OPERATOR      6      = (anyarray, anyarray),
  OPERATOR      7      @>,
  OPERATOR      8      <@,
  OPERATOR     20      @@ (_int4, query_int),
  FUNCTION      1      g_int_consistent (internal, _int4, smallint, oid, internal),
  FUNCTION      2      g_int_union (internal, internal),
  FUNCTION      3      g_int_compress (internal),
  FUNCTION      4      g_int_decompress (internal),
  FUNCTION      5      g_int_penalty (internal, internal, internal),
  FUNCTION      6      g_int_picksplit (internal, internal),
```

```
FUNCTION      7      g_int_same (_int4, _int4, internal);
```

COMPATIBILITY

CREATE OPERATOR CLASS is a PostgreSQL extension. There is no **CREATE OPERATOR CLASS** statement in the SQL standard.

SEE ALSO

ALTER OPERATOR CLASS (**ALTER_OPERATOR_CLASS(7)**), DROP OPERATOR CLASS (**DROP_OPERATOR_CLASS(7)**), CREATE OPERATOR FAMILY (**CREATE_OPERATOR_FAMILY(7)**), ALTER OPERATOR FAMILY (**ALTER_OPERATOR_FAMILY(7)**)