DBO

Generated by Doxygen 1.8.9.1

Wed Dec 16 2015 12:37:29

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 constraint Class Reference

object used to define a simple constraint

Public Member Functions

- __construct ($column, $operand, $operator=constraint::EQ)
- **sqlColumn** ()
- **sqlOperand** ($store)
- **sqlOperator** ()

Public Attributes

- const EQ = 1

  Equal.
- const LT = 2

  Less.
- const GT = 3

  Greater.
- const LE = 4

  Less or equal.
- const LTE = 5

  Less or equal.
- const GE = 6

  Greater or equal.
- const GTE = 7

  Greater or equal.
- const NE = 8

  Not equal.
- const IN = 9

  IN.

4.1.1 Detailed Description

object used to define a simple constraint

4.1.2 Constructor & Destructor Documentation

4.1.2.1 constraint::__construct (*$column, $operand, $operator* = constraint::EQ)

Parameters

$column	Name of the table column
$operand	Actual value to be used in the comparison
$operator	One of the known operators (default to EQ)

The documentation for this class was generated from the following file:

- lib/dbo.php

4.2 dbo_result Class Reference

Public Member Functions

- **__construct** ($result)
- **fetch_assoc** ()
- **__construct** ($result)
- **fetch_assoc** ()

The documentation for this class was generated from the following files:

- lib/mysql.php
- lib/pgsql.php

4.3 entity Class Reference

object used to represent either a connection to a database table or the result of a join

Public Member Functions

- __construct ($store, $t, $d=false)
- setValue ($c, $v)

  Set a column value Use this method to set a column value to be used in a create or modify operation. Set as many values as you need before issuing the create or modify.
- clearValues ()

  Clear any values that have been set for columns in the entity Call this on an entity that has been used for a create or modify operation already and is going to be reused for a different operation.
- addFilter ($f)

  Add a filter to this entity An entity can be associated with any number of filters. Data visible in the entity must satisfy at least one of the filters. Filters apply to data, modify and delete operations.
- clearFilters ()

  Remove all filters registered on this entity.
- create ($forceful=false)

  Create a new record in the database The record will have its column values set using any setValue operations that have been executed since the entity was created or since clearValues was last issued. Note that any columns not populated in this way must either allow null values or have a default defined in the table definition.
- modify ($forceful=false)

Modify a database record All records matching any filters put on the entity will be modified. If no filters exist, the modification will apply to all records in the base table. This will cause an exception to be raised unless the $forceful parameter is set true. The columns to be changed and their new values should be set using one setValue operation for each column.

- remove ($forceful=false)

  Remove records from the database table This method will remove all records from the entity's base table that match any of the filters that have been applied to it. If no filters exist, the modification will apply to all records in the base table. This will cause an exception to be raised unless the $forceful parameter is set true.

- data ($showKeys=false)

  Gets all applicable entity data This method returns an array containing all data in the base table that match any of the filters currently applied to the entity. The return value is an array of arrays of the following structure:

- join ($t, $j)

  Create an entity representing a join This method creates a new entity containing sufficient information to join the data relevant to two other entities. The data encapsulated in the resultant entity can be viewed using the data method and joined with another entity. Note that for a joined entity to be joined again, the second join must be over columns in the main entity in the first join, not the entity with which it was joined (see Join Chaining).

Public Attributes

- $name

  Database table name.

- $schema

  Name of table schema (not pgsql database)

- $columns = false

  Array containing table column information.

- $filters = array()

  Array containing registered filters.

4.3.1 Detailed Description

object used to represent either a connection to a database table or the result of a join

4.3.2 Constructor & Destructor Documentation

4.3.2.1 entity::__construct (*$store, $t, $d* = `false`)

Parameters

$store	DBO store object containing database connection
$t	Database table name
$d	Database table schema

4.3.3 Member Function Documentation

4.3.3.1 entity::addFilter (*$f*)

Add a filter to this entity An entity can be associated with any number of filters. Data visible in the entity must satisfy at least one of the filters. Filters apply to data, modify and delete operations.

Parameters

$f	The filter to be added

4.3.3.2 entity::data (*$showKeys* = `false`)

Gets all applicable entity data This method returns an array containing all data in the base table that match any of the filters currently applied to the entity. The return value is an array of arrays of the following structure:

```
Array ( [<rownum>] => Array ( [<table name>] => Array ( [id] => <value>
[name] => <column name> ) ) )
```

Parameters

$showKeys	If set true, the return array will contain all the primary and foreign key values in the base table (these will be hidden otherwise)

Returns

An array of data from the base table that match the currently defined entity

4.3.3.3 entity::join (*$t*, *$j*)

Create an entity representing a join This method creates a new entity containing sufficient information to join the data relevant to two other entities. The data encapsulated in the resultant entity can be viewed using the data method and joined with another entity. Note that for a joined entity to be joined again, the second join must be over columns in the main entity in the first join, not the entity with which it was joined (see Join Chaining).

Parameters

$t	The entity to be joined with this one
$j	An array defining the join condition (each element of the array should contain the name of the column in this table and the name of the column in the join table that need to match)

Returns

The joined entity

4.3.3.4 entity::modify (*$forceful* = `false`)

Modify a database record All records matching any filters put on the entity will be modified. If no filters exist, the modification will apply to all records in the base table. This will cause an exception to be raised unless the $forceful parameter is set true. The columns to be changed and their new values should be set using one setValue operation for each column.

Parameters

$forceful	Allow all records to be modified by a single call

4.3.3.5 entity::remove (*$forceful* = `false`)

Remove records from the database table This method will remove all records from the entity's base table that match any of the filters that have been applied to it. If no filters exist, the modification will apply to all records in the base table. This will cause an exception to be raised unless the $forceful parameter is set true.

Parameters

$forceful	Allow all records to be removed by a single call

4.3.3.6 entity::setValue (*$c, $v*)

Set a column value Use this method to set a column value to be used in a create or modify operation. Set as many values as you need before issuing the create or modify.

Parameters

$c	The column name $v The new value

The documentation for this class was generated from the following file:

- lib/dbo.php

4.4 filter Class Reference

object containing a set of constraints

Public Member Functions

- __construct ($c=false)
- add ($c)

  add a constraint to this filter
- clear ()

  remove all constraints from this filter

Public Attributes

- **$constraints** = array()

4.4.1 Detailed Description

object containing a set of constraints

4.4.2 Constructor & Destructor Documentation

4.4.2.1 filter::__construct (*$c* = `false`)

Parameters

$c	one constraint - more can be added using add method - default none

4.4.3 Member Function Documentation

4.4.3.1 filter::add (*$c*)

add a constraint to this filter

Parameters

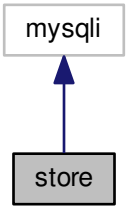
$c	the constraint to be added

The documentation for this class was generated from the following file:

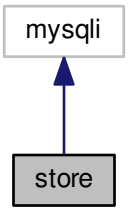
- lib/dbo.php

4.5 store Class Reference

Inheritance diagram for store:



Collaboration diagram for store:



Public Member Functions

- **__construct** ($u, $p, $h= 'localhost', $d=false)
- **query** ($sql)
- **real_escape_string** ($s)
- **describe** ($t)
- **__construct** ($u, $p, $h= 'localhost', $d=false)
- **describe** ($t)
- **__construct** ($u, $p, $d=false, $h= 'localhost')
- **describe** ($t)
- **query** ($sql)
- **real_escape_string** ($s)

Public Attributes

- **$error**

The documentation for this class was generated from the following files:

- lib/mysql.php
- lib/pgsql.php
- lib/mysqli.php

Chapter 5

File Documentation

5.1 lib/dbo.php File Reference

DBO Class definitions.

Classes

- class entity

  object used to represent either a connection to a database table or the result of a join
- class constraint

  object used to define a simple constraint
- class filter

  object containing a set of constraints

5.1.1 Detailed Description

DBO Class definitions.