

# CROSS DATABASE FIELDS



## Disclaimer

This SOFTWARE PRODUCT is provided by El Condor "as is" and "with all faults." El Condor makes no representations or warranties of any kind concerning the safety, suitability, lack of viruses, inaccuracies, typographical errors, or other harmful components of this SOFTWARE PRODUCT. There are inherent dangers in the use of any software, and you are solely responsible for determining whether this SOFTWARE PRODUCT is compatible with your equipment and other software installed on your equipment. You are also solely responsible for the protection of your equipment and backup of your data, and El Condor will not be liable for any damages you may suffer in connection with using, modifying, or distributing this SOFTWARE PRODUCT.

You can use this SOFTWARE PRODUCT freely, if you would you can credit me in program comment:

El Condor – CONDOR INFORMATIQUE – Turin

Comments, suggestions and criticisms are welcomed: mail to [rossati@libero.it](mailto:rossati@libero.it)

## Conventions

Commands syntax, instructions in programming language and examples are with font **COURIER NEW**. The optional parties of syntactic explanation are contained between [square parentheses], alternatives are separated by | and the variable parties are in *italics*.

## Contents

1	CrossData.....	3
1.1	Syntax.....	3
1.2	Compatibilities.....	3
1.3	Usage.....	3
1.3.1	Functions.....	3
1.3.2	Properties.....	4
1.4	Examples.....	4
1.4.1	Simple example.....	4
1.4.2	Calculations on numeric fields.....	4
1.4.3	Rows percentage.....	5
1.4.4	Example of FIELD operation.....	5
1.5	Advanced.....	5
1.5.1	Calculated field.....	5
1.5.2	Use with JOOMLA!.....	5
1.5.3	Use with WordPress.....	6
1.5.4	Style the table.....	6
1.5.5	Use of callback function.....	7

## 1 CrossData

CrossData is a PHP object for creating an HTML table or an array of crossed data from two fields of a table on PDO<sup>1</sup> supported Data Base; this package has been tested whit MySql and SQLite Data Bases.

Crossing is from simple count of occurrence and percentage to some group functions on a third field of the table.

### 1.1 Syntax

The syntax of the command for crossing mimics SQL (there is not case sensitivity):

```
CROSS data1 [label1] BY data2 [label2] [operation [data3] [labelOp] FROM dataTable
operation ::= % | ([%]SUM | AVG | MIN | MAX | % (ROWS | COLS)) field
SUM, AVG, MIN and MAX are the homonym SQL group functions, FIELD lists the different values
which a third field assume for every combination of data1 and data2.
dataTable ::= table | join clause | SELECT ...
```

The optional labels are used for headers in place of the field name. These are particularly significant for the header of the calculated field. Labels can contain spaces, in this case they must be delimited by apostrophes.

% ROWS and % COLS calculate a percentage relative to a row or column of a sum of third field.

After FROM there is no syntax control, it is possible to put all which SQL can accept, in particular there must be a table name (or a JOIN clause or a SELECT) and possibly a WHERE clause.

### 1.2 Compatibility

PHP 5.2.3

### 1.3 Usage

**Important:** Crossdata assumes that the database has already been opened (IE a new PDO object is created).

For using Crossdata we must include `crossdata.php`, create an object, tell the DataBase handle and call the crossing function:

```
include 'crossadata.php';
$crObj = new CrossData;
$dbh = new PDO('mysql:host=localhost;dbname=myDBName', "user", "password");
$crObj->dbh = $dbh;
echo $crObj->Cross(crossDataCommand);
```

The result is obtained by a function `Cross`:

```
$answer = $crObj->Cross(command, title, onlydata)
```

CrossData is a PHP object, it exposes some functions and properties.

#### 1.3.1 Functions

**Cross**(command, title, onlydata)

command is the cross command,

title is optional, if omitted is generated by createTitle function.

onlydata can be true or false (default). If true the function returns the data array.

CrossData returns an HTML table or a matrix depending on `onlydata` argument, if this is true, the array has the form:

first row contains the column names,

first cell of all rows but the first, contains the row name,

the required calculation or a possibly percentage occupy the last one rows and columns.

**createTitle**(f1, f2, op, f3)

This is a simple function for create a title from operation and table fields involved, somewhat like this:

1 See PDO Drivers at <http://php.net/manual/en/pdo.drivers.php>

% SUM Total of Product by Town

### 1.3.2 Properties

Most of these properties, directly or indirectly, control the presentation of the data.

<code>callBack</code>	The name of a possible function invoked before the creation of a cell in the table.
<code>callBackRowHeader</code>	Call Back for rows headers
<code>callBackColHeader</code>	Call Back for columns headers
<code>dbh</code>	Required. Is the handle to Data Base.
<code>decPoint</code>	Is the decimal separator of numbers, the default is point.
<code>IDClass</code>	The possible ID assigned to a table.
<code>ifEmpty</code>	Controls what printed when cell contains 0, by default is empty.
<code>noStyle</code>	If <code>true</code> the table doesn't contains any styling; the default is <code>false</code> .
<code>precision</code>	Controls the number of decimals printed, if the request is percentage or average and no value is supplied it is forced to 2.
<code>tableClass</code>	The class assigned to a table, the default is <code>CDTable</code> .
<code>thousandsSep</code>	Is the separator for thousands, the default is comma.

### 1.4 Examples

The examples presuppose a table like this:

```
CREATE TABLE IF NOT EXISTS `orders` (  
  `Town` varchar(25) NOT NULL,  
  `Seller` varchar(20) NOT NULL,  
  `Product` varchar(50) DEFAULT NULL,  
  `Price` int(7) DEFAULT NULL,  
  `Qty` decimal(5,2) DEFAULT NULL,  
  `Sold` int(10) DEFAULT NULL  
)
```

In the samples below is omitted the access to a Data Base, something like:

```
$dbh = new PDO('sqlite:dbName');
```

or `$dbh = new PDO('mysql:host=localhost;dbname=dbName','user','password');`

#### 1.4.1 Simple example

```
$crObj = new CrossData;  
$crObj->dbh = $dbh;  
echo $crObj->Cross("CROSS Product BY Seller FROM orders","Product by  
Seller");
```

Note the last row and column are headed by `COUNT`, if we would change this head we must write something like this: `CROSS Product BY Seller Seller 'Total number' FROM orders`, where the second `Seller` is a label for the column names.

For receive only the data:

```
$crObj = new CrossData;  
$crObj->dbh = $dbh;  
$arrayRes = $crObj->Cross("CROSS Product BY Seller FROM orders","",true);
```

#### 1.4.2 Calculations on numeric fields

```
$cross = "CROSS Product BY Seller AVG Sold 'Average sell' FROM orders";  
$crObj = new CrossData;  
$crObj->dbh = $dbh;  
$crObj->precision = 0;  
echo $crObj->Cross($cross);
```

### 1.4.3 Rows percentage

```
$cross = "CROSS Product BY Seller % ROWS Sold 'Total Sold' FROM orders";  
$crObj = new CrossData;  
$crObj->dbh = $dbh;  
echo $crObj->Cross($cross);
```

% on rows of Sold of Product by Seller					
Seller Product	Clark	Giobert	Niels	Smith	Total Sold
Chai			99.53	0.47	100.00
Chang	10.65	0.17	76.32	12.86	100.00
Filo Mix				100.00	100.00
Gravad lax				100.00	100.00
Guarana		100.00			100.00
Gudbrandsdalsost	27.41	7.78	14.43	50.38	100.00
Gula Malacca			21.26	78.74	100.00
Ikura	100.00				100.00
Inlagd Sill		10.98	4.13	84.88	100.00
Konbu				100.00	100.00
Maxilaku	100.00				100.00
Pavlova				100.00	100.00
Red table wine		100.00			100.00
Tofu				100.00	100.00

### 1.4.4 Example of FIELD operation

```
$cross = "CROSS Product BY Seller FIELD Town FROM orders WHERE Seller <>  
'Smith'";  
$crObj = new CrossData;  
$crObj->dbh = $dbh;  
$crObj->ifEmpty = "-"; // if empty print -  
echo $crObj->Cross($cross,"Cities where were sold the products from the  
seller");
```

## 1.5 Advanced

### 1.5.1 Calculated field

We can use calculate DataBase fields, but, given the scanning (rudimentary) of the command, they must be enclosed in apostrophes or written without spaces e.g.:

```
CROSS Product BY Seller SUM 'sold * 0.2' Fees FROM orders  
CROSS Product BY Seller SUM sold*0.2 Fees FROM orders
```

### 1.5.2 Use with JOOMLA!

CrossData needs the Sourcerer® plug-in which permits to insert JavaScript and PHP scripts, HTML tags, and CSS in an article; below is the sample for insert a CrossData in Joomla! 3.

```

{source}
<style>
.CDTable td, th {border: 1px solid black;padding:3px}
.CDTable td {font: normal 10pt Arial}
.CDTable th {font: bold 11pt Arial;text-align:
center;padding:3px;vertical-align:top;background-color:#ddd;}
.CDTable tr:nth-child(2n+1) {background-color:#eee;}
</style>
<?php
include 'crossdata.php';
// access data base
$dbh = new PDO('mysql:host=localhost;dbname=joomla3', "root", "");
// get table prefix
$app = JFactory::getApplication();
$prefix = $app->getCfg('dbprefix');
$cross = "CROSS Product BY Seller % Sold FROM $prefix"."orders";
$crObj = new CrossData;
$crObj->dbh = $dbh;
$crObj->ifEmpty = "-"; // if empty print -
echo $crObj->Cross($cross,"");
$dbh = null;
?>
{/source}

```

Note which we can't use the JOOMLA! functions for handle data, but we need to open the Data Base and possibly take the table prefix.

### 1.5.3 Use with WordPress

We can add PHP instruction in WordPress pages, by the plugin Exec-PHP.

```

<style>
.CDTable td, th {border: 1px solid black;padding:3px}
.CDTable td {font: normal 10pt Arial}
.CDTable th {font: bold 11pt Arial;text-align: center;padding:3px;vertical-
align:top;background-color:#ddd;}
.CDTable tr:nth-child(2n+1) {background-color:#eee;}
</style>
<?php
include 'crossdata.php';
// access data base
$dbh = new PDO('sqlite:orders.sqlite');
$cross = "CROSS Product BY Town SUM Sold FROM orders";
$crObj = new CrossData;
$crObj->dbh = $dbh;
$crObj->ifEmpty = "-"; // if empty print -
echo $crObj->Cross($cross,"");
$dbh = null;
?>

```

### 1.5.4 Style the table

In order to style the HTML table that has the class name CDTable, we can modify this name by property tableClass; moreover the table has a caption which contains the title and the first row is a THEAD row.

CrossData inserts a style text-align:right if the data are numerics and vertical-align:top;text-align:center for the THEAD row. If we won't any style we must set the property noStyle=true.

Here is a sample of style table by CSS.

```

<style>
.CDTable td, th {border: 1px solid black;padding:3px}
.CDTable td {font: normal 10pt Arial}
.CDTable th, caption {font: bold 11pt Arial;text-align: center;padding:3px
3px 5px 3px}
.CDTable tr:nth-child(2n+2) {background-color:#eee;}
.CDTable tr:nth-child(2n+3) {background-color:#ffffff;}
.CDTable tr:nth-child(1) {background-color:#ccc;}

```

</style>

### 1.5.5 Use of callback function

The callback function allows to intervene on the presentation of the data: the function must have this arguments:

- \$r cell row coordinate
- \$c cell column coordinate
- \$aData array of data
- \$hr row head (\*)
- \$hc column head (\*)

(\*) The array doesn't contains the heads of rows and columns.

```
<?PHP
function colorize($r,$c,$aData,$hr,$hc) {
    $hTot = $aData[$r][count($aData[$r])-1]; // last row cell
    if ($hTot > 0 && $c != count($aData[$r])-1 && $r != count($aData)-1) {
        $perc = floor($aData[$r][$c]*100/$aData[$r][count($aData[$r])-1]);
        if ($perc > 60) return "<span style='color:blue'>".$aData[$r][$c]."</span>";
        if ($perc < 30) return "<span style='color:red'>".$aData[$r][$c]."</span>";
    }
    return "&nbsp;".$aData[$r][$c];
}
include 'crossdata.php';
$dbh = new PDO('sqlite:orders.sqlite');
$cross = "CROSS Seller BY Town SUM Sold/1000 'Thousands sold' FROM orders";
$crObj = new CrossData;
$crObj->dbh = $dbh;
$crObj->precision = 0;
$crObj->callBack = "colorize";
echo $crObj->Cross($cross);
?>
```

Sum Sold/1000 of Seller by Town					
Town	London	Paris	Rome	Turin	Thousands sold
Seller					
Clark	53	1	16	0	70
Giobert	19	6	27	18	70
Niels	2	2	34	81	119
Smith	244	459	560	418	1,681
Thousands sold	318	468	637	517	1,940

The callbacks for rows and columns headers receives a cell.