PDBF Documentation

Patrick Bender
s9pabend@stud.uni-saarland.de

Website: https://github.com/uds-datalab/PDBF
Version 1.2.4

Contents

1 Bugs, Suggestions, Feature requests

If you encounter bugs, have suggestions or have a feature request, then please go to the issue page open a new issue if necessary and explain your concern. You can also write us an email (ichbinkeinreh at t-online.de or jens.dittrich at cs.uni-saarland.de).

2 Abstract

PDBF documents are a hybrid format. They are a valid PDF and a valid HTML page at the same time. You can now optionally add an Open Virtual Appliance (OVA) file with a complete operating system to the PDBF document. Yes, this means that the resulting file is a valid PDF, HTML, and OVA file at the same time. If you change the file extension to PDF and open it with an PDF viewer, you can see the static part of the document. If you change the file extension to HTML and open it with a Browser (currently Chrome/Firefox/Safari/IE 10 supported), you can see the dynamic part of the document. And if an ova file is attached you can also change the file extension to OVA and install and run the attached operating system.
The difference between the PDF and the HTML version is that the PDF version contains static version of all PDBF elements, whereas the HTML version is dynamic. For example you can zoom into graphs, temporarly remove dataseries from the graph, inspect and change the underling query of the PDBF element and see the result of the change directly in the browser.
This approach works completely offline. No internet connection is required, neither at compile time, nor at viewing time.
PDBF files are created from LaTeX source code and a relational database. The raw data can either be a SQL statement string, a file with SQL statements, or contained in a database (currently PostgreSQL/MySQL/MariaDB supported). In the LaTeX code you can then specify how the PDBF element (currently charts/pivot tables/multiplot charts/sql statements/dataTexts/dataTables are supported) is created from the raw data with options and an SQL query. Read more in the documentation, which is itself is a PDBF document.

PDBF toolkit is written in Java and LaTeX and can be used to compile documents on Windows, Mac, and Linux. PDBF documents are also platform independent and run on any desktop OS (Windows, Linux, Mac) with a browser/PDF viewer.

A demo paper of our tool appeared at VLDB 2015.

3 License

4 Getting started

4.1 Normal usage

- Make sure you have a Java Runtime (version >= 1.7) and a LaTeX distribution installed
- Download the latest version
- Extract zip and change workingdir to extracted folder
- Adjust config.cfg
- Try to compile minimal.tex file with this command: `java -jar pdbf.jar minimal.tex`
- Open minimal.html, this is the final output of the compilation process, if you rename it to ".pdf" it is also a valid pdf-document

4.2 Attach a Open Virtual Appliance (OVA) file

- Optionally you can attach the included vldb-Invaders.ova (Space invaders clone) or download the dsl.ova (Damn small linux) Open Virtual Appliance (OVA) file and attach it to the compiled PDBF file with this command: `java -jar pdbf.jar --vm minimal.html vldb-Invaders.ova`.
- Open minimal.ova, this is the final output with the attached ova file. Its still a valid pdf and html file at the same time.

4.3 Attach a tar archive

Note: compressed tar.gz archives are not supported. You can only wrap your tar.gz file in another tar and then attach it.

- Optionally you can attach a tar archive file to the compiled PDBF file with this command: `java -jar pdbf.jar --tar minimal.html TAR_file.tar`
- Open minimal.tar, this is the final output with the attached tar file. Its still a valid pdf and html file at the same time.

4.4 Learn how to use the PDBF framework

- You can play around with the minimal.tex file. It contains a small example on how to specify PDBF elements in LaTeX
- For further information take a look at the documentation

5 Features

5.1 Automatic generation of Charts, Multiplot Charts, Pivot tables

With PDBF you don't need to manually generate these kinds of elements. The PDBF compiler automatically generates a static version for the pdf and the dynamic version for the html part of the PDBF document. This also means that your document is always up to date! If you change something in the underlying data that generate your PDBF document and then recompile the document, then the data in the document is up to date. No need to manually update externally generated charts or pivot tables.

5.2 Generate your document directly from the results of your experiment

The idea of PDBF is to store the results of the experiment directly in the document and to make it more transparent how this chart, pivot table, etc. was generated from the result data. Therefore we currently support CSV files, SQL files, and SQL servers as data sources and use SQL as description language for the transformation of the raw result data to the final representation in the document.

5.3 Compile LaTeX to single HTML file

You can also use the PDBF compiler to compile your LaTeX files to a single HTML file. To do so just run the compiler as on any other document (you dont need to include the pdbf package in your tex file):

```
java -jar pdbf.jar sometexfile.tex
```

The resulting HTML file is saved in the same folder with the same name but with ".html" filename extension.

6 Build Instructions

- Run "mvn package" if you only want to compile pdbf.jar
- Run "mvn verify" if you want to compile pdbf.jar and run integration tests.

7 Thanks to

phantomJS (https://github.com/ariya/phantomjs)
Apache Commons (http://commons.apache.org/)
Apache PDFBox (https://pdfbox.apache.org/)
jTar (https://github.com/kamranzafar/jtar)
google-gson (https://github.com/google/gson)
postgreSQL JDBC Driver (https://jdbc.postgresql.org/)
MariaDB JDBC Driver (https://mariadb.com/kb/en/mariadb/about-the-mariadb-java-client/)
AlaSQL (https://github.com/agershun/alasql)
C3 (https://github.com/masayuki0812/c3)
D3 (https://github.com/mbostock/d3)
Codemirror (https://github.com/codemirror/codemirror)
google-diff-match-patch (https://code.google.com/p/google-diff-match-patch/)
explorercanvas (https://code.google.com/p/explorercanvas/)
DataTables (https://github.com/DataTables/DataTables)
jQuery (https://github.com/jquery/jquery)
jQuery-UI (https://github.com/jquery/jquery-ui)
jStat (https://github.com/jstat/jstat)
pivottable (https://github.com/nicolaskruchten/pivottable)
PDF.js (https://github.com/mozilla/pdf.js)
google-closure-compiler (https://github.com/google/closure-compiler)
lz-string (https://github.com/pieroxy/lz-string)
YUI Compressor (https://github.com/yui/yuicompressor)

8 How to use the PDBF compiler

– The standart mode which compiles a tex file into the janiform PDBF format:
`java -jar PDBF.jar LaTeX_file`

Note: The compiled document can be found in the same folder as the tex file and has the same name as the tex file but ".html" as filename extension.

– The VM mode which attaches a Open Virtual Appliance (OVA) file to an existing PDBF document:
`java -jar PDBF.jar --vm PDBF_File.html VM_File.ova`

Note: The compiled document can be found in the same folder as the html file and has the same name as the html file but ".ova" as filename extension.

– The TAR mode which attaches a tar archive file to an existing PDBF document:
`java -jar PDBF.jar --tar PDBF_File.html TAR_File.tar`

Note: The compiled document can be found in the same folder as the html file and has the same name as the html file but ".tar" as filename extension.

Note: compressed tar.gz archives are not supported. You can only wrap your tar.gz file in another tar and then attach it.

9 Requirements

The following LaTeX packages are required for PDBF documents:

- `zref`
- `xcolor`
- `graphicx`
- `xstring`
- `xparse`
- `geometry`
- `array`

Furthermore it is needed that your LaTeX document uses the geometry package to specify the page size.

Warning: There are problems with **inputenc** and **pgfpages** package. If you really have to use the inputenc package with UTF8 option then you cant use non-ASCII characters inside queries. The pgfpages package is currently incompatible to this package.

10 SQL specifics and special functions

PDBF uses AlaSQL as database engine. For all available features of the database visit https://github.com/agershun/alasql. Some functionality might not yet be available because we currently don't use the latest version.

Note: Mostly sql syntax is standart sql, but for example attributes with spaces in their name have to be sourrounded by square brackets. E.g. SELECT attribute AS [attribute with spaces] FROM test;

We extended AlaSQL with the following statistical sql functions:

`GRUBBS_FILTER(arr, alpha, max)`:

Desc.: Filters an array of values with a two-sided grubbs test. The grubbs test is performed on the whole array. If a significant outlier is detected (Z ¿ ZCrit) and the relative margin of error is not too high (margin_of_error / avg ¿ 0.025) then the outlier is removed. This procedure is repeated until one of these two conditions fails.

Returns the filtered array.

arr	
Desc.:	Array on which the operation is performed.
Argument:	JavaScript array
Default:	No default value. Value must always be set!

alpha	
Desc.:	significance level
Argument:	Number > 0 AND Number < 1
Default:	`0.05`

`MEAN(arr)`:

Desc.: Calculates the arithmetic mean of an array of values.

Returns the arithmetic mean of the array.

arr	
Desc.:	Array on which the operation is performed.
Argument:	JavaScript array
Default:	No default value. Value must always be set!

`STDDEV_SAMP(arr)`:

Desc.: Calculates the sample standard deviation of an array of values.

Returns the sample standard deviation of the array.

arr	
Desc.:	Array on which the operation is performed.
Argument:	JavaScript array
Default:	No default value. Value must always be set!

`MARGIN_OF_ERROR(arr, alpha)`:

Desc.: Calculates the margin of error of an array of values.

Returns the margin of error of the array.

arr	
Desc.:	Array on which the operation is performed.
Argument:	JavaScript array
Default:	No default value. Value must always be set!

alpha	
Desc.:	significance level
Argument:	Number > 0 AND Number < 1
Default:	`0.05`

`CONF_INT(arr, alpha)`:

Desc.: Calculates the confidence interval of an array.

Returns the confidence interval of the array.

arr	
Desc.:	Array on which the operation is performed.
Argument:	JavaScript array
Default:	No default value. Value must always be set!

alpha	
Desc.:	significance level
Argument:	Number > 0 AND Number < 1
Default:	`0.05`

`T_TEST(arr1, arr2, alpha)`:

Desc.: Performs a Student's t-test on two arrays.

Returns true if they are indistinguishable, and returns false otherwise.

arr1, arr2	
Desc.:	Arrays on which the operation is performed.
Argument:	JavaScript array
Default:	No default value. Value must always be set!

alpha	
Desc.:	significance level
Argument:	Number > 0 AND Number < 1
Default:	`0.05`

`WELCH_TEST(arr1, arr2, alpha)`:

Desc.: Performs a Welch's t-test on two arrays.

Returns true if they are indistinguishable, and returns false otherwise.

arr1, arr2	
Desc.:	Arrays on which the operation is performed.
Argument:	JavaScript array
Default:	No default value. Value must always be set!

alpha	
Desc.:	significance level
Argument:	Number > 0 AND Number < 1
Default:	`0.05`

11 Data sources

11.1 SQL

Macros:

`\dbSQLText{sqlQueryString}`
`\dbSQLFile{fileWithSQLQueries}`
Note: Relative filepaths are relative to the tex file.
`\dbSQLJDBC{jdbcConnectionURL}{user}{password}{commaSeperatedListOfTableNames}`
Note: jdbcConnectionURL consist of jdbc followed by the name of the dbms (only postgresql and mysql are currently supported) followed by the url of the dbms followed by the database name (e.g. jdbc:postgresql://localhost:5432/postgres).
Note: jdbcConnectionURL, user, and password are not stored in the output documents.

11.2 CSV

Note: By default the first line of the CSV file is used as attribute names for the table. You can use the header option to manually specify attribute names.

`\dbCSVFile[options]{fileWithCSVData}{tableName}`

Note: Relative filepaths are relative to the tex file.

Options:

headers	
Desc.:	Defines the names of columns for the CSV file. If not set the first line of the CSV file is used as header.
Argument:	Javascript-Array of Strings
Default:	`[]`

quote	
Desc.:	Defines the quote character for this CSV file.
Argument:	String
Default:	`"`

seperator	
Desc.:	Defines the seperator character for this CSV file.
Argument:	String
Default:	`,`

12 PDBF elements

Note: All PDBF elements use the font size, font family, and font style that was active at the point of their definition. You can for example surround all commands with `\textbf{...}`, `\emph{...}` or any other font command.

12.1 Chart

Examples:

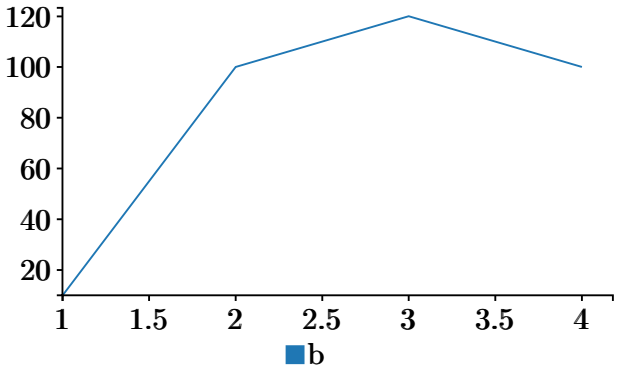

Figure 1: Line Chart

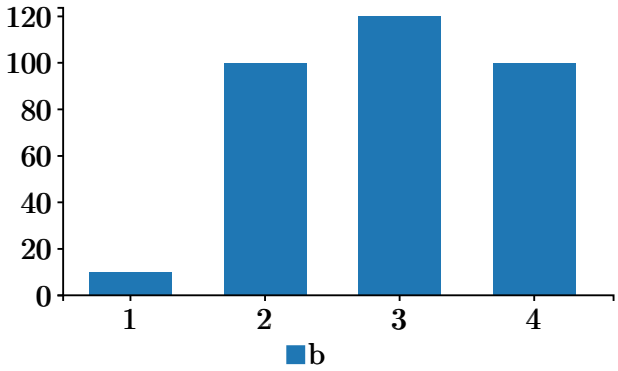

Figure 2: Bar Chart

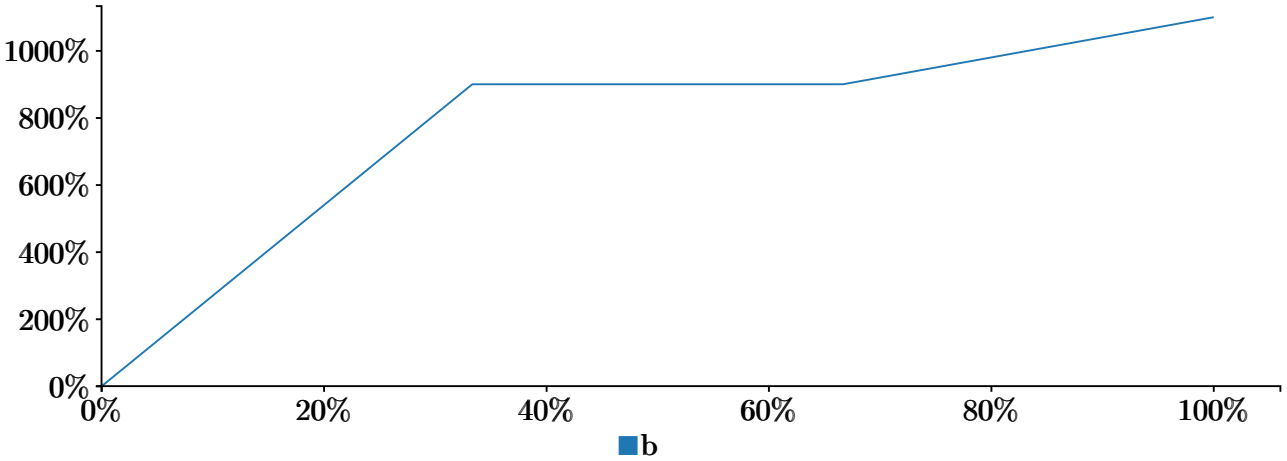

Figure 3: compareToBest Chart

Macros:

`\chart[options][queryForOverlay]{queryForPage}`
Note: if `queryForOverlay` is omitted the `queryForPage` is used for the overlay.
Note: The first column of the query result is used for the x-Axis, all other columns are used for y-Axis.

Options:

name	
Desc.:	Sets the name of this element. This is only useful for linking Elements. See DataText.
Argument:	String
Default:	`undefined`

width	
Desc.:	Sets the width of the chart.
Argument:	LaTeX length
Default:	No default value. Value must always be set!

height	
Desc.:	Sets the height of the chart.
Argument:	LaTeX length
Default:	No default value. Value must always be set!

quality	
Desc.:	Sets the quality for the image version of the chart in the pdf. 1.0 corresponds roughly to 240 pixels per inch. Can also be redefined globally (`pdbfQuality`).
Argument:	Number > 0
Default:	`1.0`

chartType	
Desc.:	Sets the type of the chart.
Argument:	`line` or `bar` or `compareToBest`
Default:	`line`

Note: A chart or multiplot chart with option "chartType" set to "compareToBest" expects either an array of numbers or an array of an array of numbers as input. It plots the slowdown compared to the best value in the array. In the second case the inner array of numbers contains values for repetitions of the same experiment.

xunit	
Desc.:	Sets the a label for the x-axis.
Argument:	String
Default:	`""` (which means hide)

yunit	
Desc.:	Sets the a label for the y-axis.
Argument:	String
Default:	`""` (which means hide)

options	
Desc.:	Sets options that are directly passed to the c3 chart library (→ Documentation). You need to wrap the JSON-String with {} if you want to use `[` or `]` or `,`.
Argument:	JSON-String
Default:	{} (which means empty object)

includeZero	
Desc.:	Set the minimum of the range of the y-axis to zero.
Argument:	`true` or `false`
Default:	`false`

drawPoints	
Desc.:	Whether to show each point in line.
Argument:	`true` or `false`
Default:	`false`

fillGraph	
Desc.:	Whether to fill the area below the graph.
Argument:	`true` or `false`
Default:	`false`

showRangeSelector	
Desc.:	Whether to show a range selector for the x-axis below the chart.
Argument:	`true` or `false`
Default:	`false` for in-page chart, `true` for overlay chart

logscale	
Desc.:	If set, the y-axis uses log scale. Can also be redefined globally (`pdbfLogscale`).
Argument:	`true` or `false`
Default:	`false`

12.2 Multiplot Chart

Example:

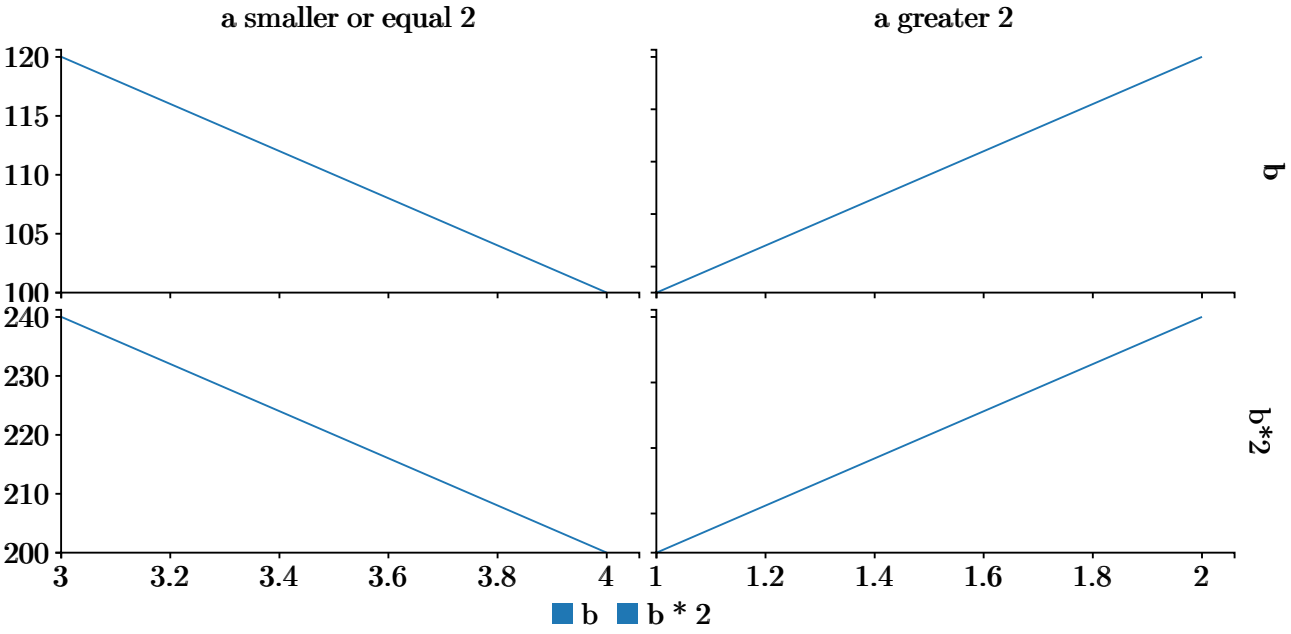

Figure 4: Multiplot Line Chart

Macros:

`\multiplotChart[options][queryForOverlay]{queryForPage}`
Note: If `queryForOverlay` is omitted the `queryForPage` is used for the overlay
Note: Both `queryForOverlay` and `queryForPage` must contain exactly two occurences of the ? character. These are later replaced with values from the xValues/yValues option.

Options:

Note: For multiplot charts all options of chart are also valid options.

xCount	
Desc.:	Sets number of columns.
Argument:	Number > 0
Default:	No default value. Value must always be set!

yCount	
Desc.:	Sets number of rows.
Argument:	Number > 0
Default:	No default value. Value must always be set!

leftArr	
Desc.:	Sets the labels for the left side.
Argument:	Either a JavaScript array of Strings where each string corresponds to exactly one row (e.g {"row1", "row2"}) or a JavaScript array of Objects with a c property which corresponds to the number of rows this text should span and a text property which corresponds to the text-string.
Default:	If this option is not set, then the value of the xunit option is used spanning over the whole site.

rightArr	
Desc.:	Sets the labels for the right side.
Argument:	Either a JavaScript array of Strings where each string corresponds to exactly one row (e.g {"row1", "row2"}) or a JavaScript array of Objects with a c property which corresponds to the number of rows this text should span and a text property which corresponds to the text-string.
Default:	If this option is not set, then the value that is used for the query in this row is used.

bottomArr	
Desc.:	Sets the labels for the bottom side.
Argument:	Either a JavaScript array of Strings where each string corresponds to exactly one column (e.g {"column1", "column2"}) or a JavaScript array of Objects with a c property which corresponds to the number of columns this text should span and a text property which corresponds to the text-string.
Default:	If this option is not set, then the value of the yunit option is used spanning over the whole site.

topArr	
Desc.:	Sets the labels for the top side.
Argument:	Either a JavaScript array of Strings where each string corresponds to exactly one column (e.g {"column1", "column2"}) or a JavaScript array of Objects with a c property which corresponds to the number of columns this text should span and a text property which corresponds to the text-string.
Default:	If this option is not set, then the value that is used for the query in this column is used.

xValues	
Desc.:	The values that replace the first ? character in the query. If yFirst is set, they replace the second ? character in the query instead.
Argument:	A JavaScript array that has as much entries as the xCount options value
Default:	No default value. Value must always be set!

yValues	
Desc.:	The values that replace the second ? character in the query. If yFirst is set, they replace the first ? character in the query instead.
Argument:	A JavaScript array that has as much entries as the yCount options value
Default:	No default value. Value must always be set!

yFirst	
Desc.:	If this option is set, then first ? character is replaced with values from yValues option and second ? character is replaced with values from xValues. If this option is not set, then first ? character is replaced with values from xValues option and second ? character is replaced with values from yValues.
Argument:	`true` or `false`
Default:	`false`

forceXequal	
Desc.:	If this option is set, then all columns have the same x-axis range as the uppermost chart. If this option is not set, then all charts have individual x-axis ranges.
Argument:	`true` or `false`
Default:	`false`

forceYequal	
Desc.:	If this option is set, then all rows have the same y-axis range as the leftmost chart. If this option is not set, then all charts have individual y-axis ranges.
Argument:	`true` or `false`
Default:	`false`

12.3 Pivot Table

Example:

a	Totals
1	10.00
2	100.00
3	120.00
4	100.00
Totals	10.00

Figure 5: Pivot Table

Macros:

`\pivotTable[options][queryForOverlay]{queryForPage}`
Note: if `queryForOverlay` is omitted the `queryForPage` is used for the overlay

Options:

width	
Desc.:	Sets the width of the table.
Argument:	LaTeX length
Default:	No default value. Value must always be set!

height	
Desc.:	Sets the height of the table.
Argument:	LaTeX length
Default:	No default value. Value must always be set!

quality	
Desc.:	Sets the quality for the image version of the table in the pdf. 1.0 corresponds roughly to 240 pixels per inch. Can also be redefined globally (`pdbfQuality`).
Argument:	Number > 0
Default:	`1.0`

aggregation	
Desc.:	Sets the aggregation function for the page.
Argument:	`Count` or `Count Unique Values` or `List Unique Values` or `Sum` or `Integer Sum` or `Average` or `Minimum` or `Maximum` or `Sum over Sum` or `80% Upper Bound` or `80% Lower Bound` or `Sum as Fraction of Total` or `Sum as Fraction of Rows` or `Sum as Fraction of Columns` or `Count as Fraction of Total` or `Count as Fraction of Rows` or `Count as Fraction of Columns`
Default:	`Minimum`

aggregationBig	
Desc.:	Sets the aggregation function for the overlay.
Argument:	`Count` or `Count Unique Values` or `List Unique Values` or `Sum` or `Integer Sum` or `Average` or `Minimum` or `Maximum` or `Sum over Sum` or `80% Upper Bound` or `80% Lower Bound` or `Sum as Fraction of Total` or `Sum as Fraction of Rows` or `Sum as Fraction of Columns` or `Count as Fraction of Total` or `Count as Fraction of Rows` or `Count as Fraction of Columns`
Default:	If this option is not set, then the value of the aggregation option is used

aggregationattribute	
Desc.:	Sets the attribute for the aggregation function in the page.
Argument:	The name of an attribute that is present in the result of the sql query for the page
Default:	No default value. Value must always be set!

aggregationattributeBig	
Desc.:	Sets the attribute for the aggregation function in the overlay.
Argument:	The name of an attribute that is present in the result of the sql query for the overlay
Default:	If this option is not set, then the value of the aggregationattribute option is used

cols	
Desc.:	The attributes for the columns in the pivot table. **Note:** This feature is currently broken!
Argument:	JavaScript array of strings (e.g. [”a”, ”b”, ”c”]). The name of attributes have to be present in the result of the sql query for the overlay and page
Default:	`[]` (which means empty array)

rows	
Desc.:	The attributes for the rows in the pivot table.
Argument:	JavaScript array of strings (e.g. [”a”, ”b”, ”c”]). The name of attributes have to be present in the result of the sql query for the overlay and page
Default:	`[]` (which means empty array)

12.4 SQL

Clickable SQL-statements

Note: Currently SQL is limited to one line of text. No line breaks are possible.
Example:

SELECT * FROM test;

Macros:

`\sql[options][textForPage]{queryForOverlay}`
Note: if `textForPage` is omitted the `queryForOverlay` is used as text for the page.

Options:

color	
Desc.:	The textcolor of the displayed text.
Argument:	`white`, `black`, `red`, `green`, `blue`, `cyan`, `magenta`, `yellow` or xcolor syntax (see here). Can also be redefined globally (`pdbfDynamicTextColor`)
Default:	`blue`

12.5 DataText

Write text that is the result of a SQL-statement.
Note: Currently DataText is limited to one line of text. No line breaks are possible.
Note: DataText can be linked to other PDBF elements (Currently only charts supported). This means that if you move the mouse over this DataText then the specified data in the linked PDBF element is highlighted.

Example:

The table "test" contains 4 tuples.
The table "test" contains the following tuples: (1, 10), (2, 100), (3, 120), (4, 100).

Macros:

`\dataText[options]{query}`

Options:

color	
Desc.:	The textcolor of the displayed text.
Argument:	`white`, `black`, `red`, `green`, `blue`, `cyan`, `magenta`, `yellow` or xcolor syntax (see here). Can also be redefined globally (`pdbfDynamicTextColor`)
Default:	`blue`

linkTo	
Desc.:	When this option is set then this DataText and the element with name equal to the value of `linkTo` are linked. When the cursor is moved above the DataText then in the linked element the specified `linkSelector` specifies which part of the element is highlighted and annotated with the value of `linkLabel`.
Argument:	String which value is the name of some other element. **Currently only chart elements can be used for `linkTo`.**
Default:	`undefined`

linkSelector	
Desc.:	The selector that specifies which part of the `linkTo` element is highlighted.
Argument:	`Number`, `Date` or `String` depending on the type of the `linkTo` element. For charts the type depends on the datatype of the xAxis and the value specifies which part of the xAxis is highlighted.
Default:	`undefined`

linkLabel	
Desc.:	The text that is displayed near the highlighted area.
Argument:	String
Default:	`undefined`

12.6 DataTable

Write a LaTeX table that is the result of a SQL-statement.

Example:

a	b
1	10
2	100
3	120
4	100

Macros:

`\dataTable[options]{query}`

Options:

color	
Desc.:	The textcolor of the displayed text.
Argument:	`white`, `black`, `red`, `green`, `blue`, `cyan`, `magenta`, `yellow` or xcolor syntax (see here). Can also be redefined globally (`pdbfDynamicTextColor`)
Default:	`blue`

verticalLines	
Desc.:	Specifies the vertical lines for the table.
Argument:	`a` for all, `i` for inner, `o` for outer, `n` for none, or you can use the default tabular syntax (e.g. `|l|c|r|`)
Default:	`a`

horizontalLinesHeader	
Desc.:	Specifies the horizontal lines for the table header.
Argument:	`a` for all, `i` for inner, `o` for outer, `n` for none, or you can specify a custom pattern that is repeatedly applied with the following format: h means line, b means no line, the space character delimits format specifiers. E.g. "hh b" means: double line followed by no line followed by double line followed by no line ... and so on
Default:	`a`

horizontalLinesBody	
Desc.:	Specifies the horizontal lines for the table body, excluding the border above the first tuple, which is controlled by horizontalLinesHeader.
Argument:	`a` for all, `i` for inner, `o` for outer, `n` for none, or you can specify a custom pattern that is repeatedly applied with the following format: h means line, b means no line, the space character delimits format specifiers. E.g. "hh b" means: double line followed by no line followed by double line followed by no line ... and so on
Default:	`a`

13 F.A.Q.

- Q.: The overlay is not on the right position!
  A.: Most likely you use pages with different sizes in your document or you use the pgfpages package. This is currently not supported by PBDF

- Q.: I get `Error: paperwidth value missing! Did you forgot to specify the papersize via the geometry package?` when compiling my document.
  A.: You need to explicitly specify the papersize of your document via the geometry package (e.g. `\usepackage[letterpaper]{geometry}`)

- Q.: I get `\unhbox \voidb@x \bgroup \let \unhbox \voidb@x \setbox \@tempboxa \hbox {u\global \mathchardef \accent@spacefactor \spacefactor }\accent 127 u\egroup \spacefactor \accent@spacefactor` or a similar string in a error message.
  A.: Chances are high that you use the `\usepackage[utf8]{inputenc}` package or a similar package. These are not fully compatible with PDBF. If you really have to use the utf8 package then you cant use special characters inside options and queries of the PDBF package.

- Q.: I get strange errors when running "maven verify".
  A.: If you are using TexLive 2014 or earlier and get the following errors when running "maven verify":

  ```
  The file is not valid, error(s) :
  ```

```
1.2.1 : Body Syntax error, Single space expected [offset=2786901; key
   =2786901; line=5 0 obj <<; object=COSObject{5, 0}]
1.2.1 : Body Syntax error, Single space expected [offset=2786635; key
   =2786635; line=11 0 obj <<; object=COSObject{11, 0}]
1.2.1 : Body Syntax error, EOL expected before the 'endobj' keyword
   at offset 2786894
1.2.1 : Body Syntax error, Single space expected [offset=2761071; key
   =2761071; line=3 0 obj <<; object=COSObject{3, 0}]
1.2.1 : Body Syntax error, Single space expected [offset=2761433; key
   =2761433; line=8 0 obj <<; object=COSObject{8, 0}]
1.2.1 : Body Syntax error, Single space expected [offset=2787764; key
   =2787764; line=12 0 obj <<; object=COSObject{12, 0}]
```

Then you can safely ignore them. They are non critical errors. You can get rid of these errors by upgrading to TexLive 2015 or later.