PSTricks Macros for Databases*

Wolfgang May†

2006/11/26

Abstract

The *pst-dbicons* package provides some useful macros in the database area. It focusses on typesetting ER-Diagrams in a declarative style, i.e., by positioning some nodes and defining the position of all other nodes relative to them by using the standard database terminology. The PSTricks package is required for using pst-dbicons, but there is no deep knowledge of PSTricks commands required (although this is useful for exploiting the full functionality).

1 Commands

1.1 ER-Diagrams

ER-Diagrams are a widely used graphical representation formalism for conceptual modeling; especially used in the database community. Their main notions are *entities* (e.g., persons, cities, or countries), *attributes* of entities (e.g., name, id-number, age), and relationships between entities (e.g., belongs_to, is_capital_of).

`\seticonparams` With

`\seticonparams{`⟨*icon-type*⟩`}{`⟨*graphics parameters*⟩`}`

the graphical layout of icons (entities, relationships, and attributes) can be specified (by giving the optional argument for PSTricks' boxes). Default is `[fillstyle=none]` for all of them. in this documentation, we modify it to

```
\seticonparams{entity}{shadow=true,fillcolor=lightgray,fillstyle=solid}
\seticonparams{attribute}{fillcolor=lightgray,fillstyle=solid}
\seticonparams{relationship}{shadow=true,fillcolor=lightgray,fillstyle=solid}
```

From the user's point of view, every ER-icon (entities, relationships, attributes) has a name which should be typeset in the respective box. Additionally, there are *internal* identifiers of the boxes/nodes to allow for referencing by graphics commands in the standard PSTricks way. These *internal id*'s must start with a letter and must contain only letters and digits. In general, id's must be unique on every page.[1] Often, the *name* of the entity/relationship/attribute satisfies these conditions; in these cases, the name is also used as id. in other cases, name and id must be specified. Thus, for all commands for typesetting ER-icons, the node id *must* be given, whereas the name to be typeset in the box is optional; if no name is specified, the id is typeset as the name.

*This file has version number 0.2, last revised 2006/11/26.

†`may@informatik.uni-goettingen.de`

[1] people who are familiar with PSTricks know under which conditions non-unique node identifiers can be used.

1.2 Entities and Attributes

`\entity` With

`\entity[`⟨*property*⟩`]{`⟨*id*⟩`}[`⟨*text*⟩`]`

an entity type is set as a rectangular node. If the optional argument ⟨*text*⟩ is not given, ⟨*id*⟩ also provides the entity's text. ⟨*property*⟩ is used for *weak* entitiy types which are denoted by double lines:
Here is a simple entity `\entity{Person}` 

where `Person` serves also as node text; `\entity[weak]{cty}[City]` 

makes up a weak entity type where the displayed name is different from the internal name.

`\attribute` Attribute icons are set as an oval nodes by

`\attribute[`⟨*property*⟩`]{`⟨*id*⟩`}[`⟨*text*⟩`]`

Here, the optional argument ⟨*property*⟩ can take the values `mv` (multivalued; resulting in a double-lined oval) or `key` (key attribute; resulting in underlining the attribute name). Here are three attributes:

- an ordinary one, `\attribute{phone}[phone\_no]` 

,
- a multivalued attribute, `\attribute[mv]{nickname}` 

,
- a key attribute, `\attribute[key]{pid}[person\_id]` 

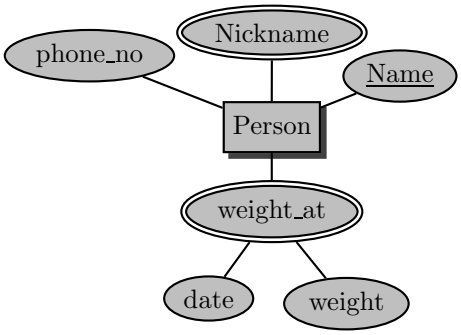
.

Note that with *phone_no*, the optional argument is used for the node text (in case that the node text contains stuff that is not allowed in internal postscript ids – as a rough rule, only characters (and since dbicons 1.14, underscores) are allowed in ids).

`\attributeof` `\attrdist` In Section 1.3, relationship nodes are defined analogously. At first, it is described how to attach attributes with entities:

`\attributeof{`⟨*id*⟩`}[`⟨*dist*⟩`]{`⟨*angle*⟩`}[`⟨*property*⟩`]{`⟨*id*$_2$⟩`}[`⟨*text*⟩`]`

typesets an attribute node at angle ⟨*angle*⟩ in distance ⟨*dist*⟩ from the node which is identified by ⟨*id*⟩ (which can be either an entity node or a relationship node). The attribute is made a node named ⟨*id*$_2$⟩. ⟨*id*$_2$⟩ and ⟨*text*⟩ work as for `\entity`. The argument ⟨*dist*⟩ is optional, it has not to be given with *every* attribute. By `\attrdist{`⟨*dist*⟩`}`, this value can be set to a default (as startup default, 2em is set).




```
\entity{Person}
\attributeof{Person}{30}[key]{Name}
\attributeof{Person}{90}[mv]{Nickname}
\attributeof{Person}[4em]{150}
            {phone}[phone\_no]
\attributeof{Person}{270}[mv]{wt}[weight\_at]
\attributeof{wt}{220}{date}
\attributeof{wt}{300}{weight}
```

Note that *phone_no* is set with a bigger distance to *person*. Additionally, the example shows how complex attributes can be set with these commands.

1.3 Relationships

`\relationship` With `\relationship[`⟨*property*⟩`]{`⟨*id*⟩`}[`⟨*text*⟩`]`, a relationship type is set as a diamond-shaped node. Here, the optional argument ⟨*property*⟩ is used to represent *identifying* relationships, used for *weak* entities – thus ⟨*property*⟩ can be equivalently `weak` or `ident` which results in a double-lined relationship type.

`lationshipbetween` For declaratively specifying nodes representing relationships between entities,

`\relationshipbetween[`⟨*property*⟩`]{`⟨*entity-is*$_1$⟩`}{`⟨*entity-id*$_2$⟩`}`
`{`⟨*relationship-id*⟩`}[`⟨*relationship-name*⟩`]`

is used (which can be augmented with several optional arguments). In the simplest version, as given above, a relationship node is set in-between two entity nodes:

```
\entity{Person} \hspace*{6cm} \entity{Company}
\relationshipbetween{Person}{Company}{worksat}[works\_at]
```

Additionally, the *roles* of the entities in the relationship, and the cardinalities can be given (both as independent optional arguments):

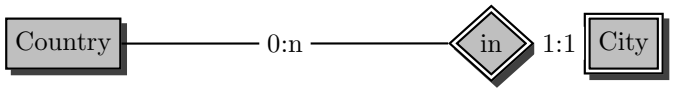
`\relationshipbetween[`⟨*property*⟩`]`
`{`⟨*entity-id*$_1$⟩`}(`⟨*role*$_1$⟩`)[`⟨*card*$_1$⟩`]`
`{`⟨*entity-id*$_2$⟩`}(`⟨*role*$_2$⟩`)[`⟨*card*$_2$⟩`]`
`{`⟨*relationship-id*⟩`}[`⟨*relationship-name*⟩`]`

```
\entity{Country} \hspace*{6cm} \entity{City}
\relationshipbetween{Country}(of)[1:1]{City}(is)[0:1]{capital}
```

Moreover, the placement of the relationship node wrt. the entities can be specified: above, the relationship node was put in the middle of an imaginary line in-between the entity nodes.
As a first, small, extension, the placement ratio of the diamond between the entities can be changed (default: 0.5):

```
\relationshipbetween[⟨property⟩]{⟨entity-id1⟩}...{⟨entity-id2⟩}...
        {⟨relationship-id⟩}[⟨relationship-name⟩](⟨placement-ratio⟩)
```

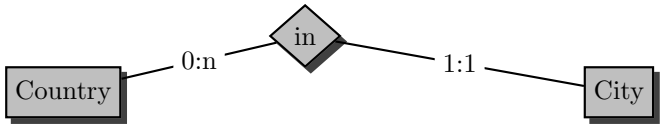
```
\entity{Country} \hspace*{6cm} \entity[weak]{City}
\relationshipbetween[ident]{Country}[0:n]{City}[1:1]{in}(0.8)
```



If this is still not enough, instead of an imaginary straight line, any other PSTricks node connection command can be used, most likely `\ncarc[...]` with suitable optional arguments:

```
\relationshipbetween[⟨property⟩]
     {⟨entity-id1⟩}...{⟨entity-id2⟩}...
     {⟨relationship-id⟩}[⟨relationship-name⟩]
     /⟨connection⟩/(⟨placement-ratio⟩)
```

```
\entity{Country} \hspace*{6cm} \entity{City}
\relationshipbetween{Country}[0:n]{City}[1:1]
      {in}/\ncarc[arcangle=20]/(0.4)
```

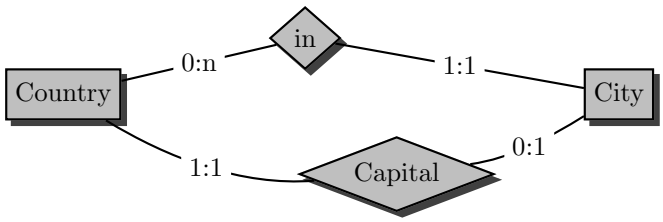


For TEX-insiders: Note that the use of `/.../` as argument delimiter avoids collisions with the delimiters used by PSTricks which thus can be used inside `/.../`.
In the above example, although the relationship is placed on an imaginary arc, the *connections* are straight lines. For these lines, PSTricks commands can be given, too. With this, the *full* syntax is

```
\relationshipbetween[⟨property⟩]
   {⟨entity-id1⟩}(⟨role1⟩)[⟨card1⟩]/⟨connection1⟩/
   {⟨entity-id2⟩}(⟨role2⟩)[[⟨card2⟩]/⟨connection2⟩/
   {⟨relationship-id⟩}[⟨relationship-name⟩]/ ⟨connection⟩/(⟨placement-ratio⟩)
```

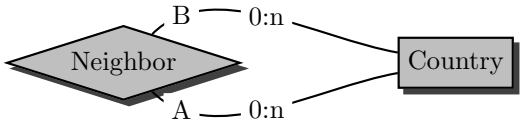
where all arguments embraced with `(...)`, `[...]`, or `/.../` are optional.
With this, an example can be given where two different relationships can hold between a pair of entity types:

```
\entity{Country} \hspace*{6cm} \entity{City}
\relationshipbetween{Country}[0:n]{City}[1:1]{in}
/\ncarc[arcangle=20]/(0.4)
\relationshipbetween{Country}[1:1]/\ncarc[arcangle=-18]/
     {City}[0:1]/\ncarc[arcangle=-12]/
     {Capital}/\ncarc[arcangle=-30]/(0.6)
```



Note that for *capital*, `\ncarc` is used for the lines, where the `arcangle` of the whole arc is divided suitably to the placement ratio of the diamond node, and that the `arcangle` of the second entity is negative (since the connection always points from the entity to the relationship).
There is one more interesting special case of binary relationships: recursive relationships. There, the entity node cannot be set "between" the nodes – it has to be placed relative to the entity type which is involved in the relationship.
In this case, the argument ⟨*connection*⟩ has no effect, and last argument (⟨*placement ratio*⟩) has the syntax `(`⟨*distance*⟩`,`⟨*angle*⟩`)`: the relationship node is set at angle ⟨*angle*⟩ in distance ⟨*dist*⟩ from the entity node. The default for ⟨*connection*$_1$⟩ and ⟨*connection*$_2$⟩ is set to `\ncarc[arcangleA=10,arcangleB=45]}` and
`\ncarc[arcangleA=-10,arcangleB=-45]`, respectively.

```
\hspace*{6cm}\entity{Country}
\relationshipbetween{Country}(A)[0:n]{Country}(B)[0:n]
         {Neighbor}(8em,180)
```



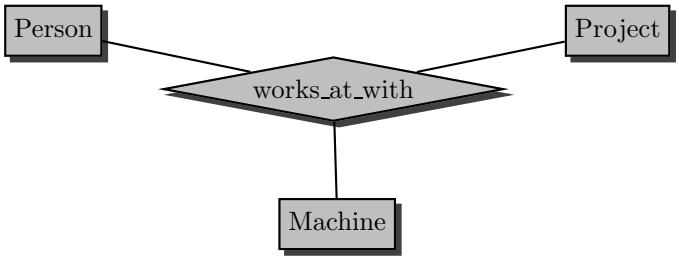
With the above commands, only binary relationships can be represented. Thus, there is one more macro, relating entity nodes with already existing relationship nodes:

`\inrelationship`

```
\inrelationship[⟨property⟩]{⟨entity-id⟩}(⟨role⟩)[⟨card⟩]
    /⟨connection⟩/{⟨relationship-id⟩}
```

With this, n-ary relations can be represented:

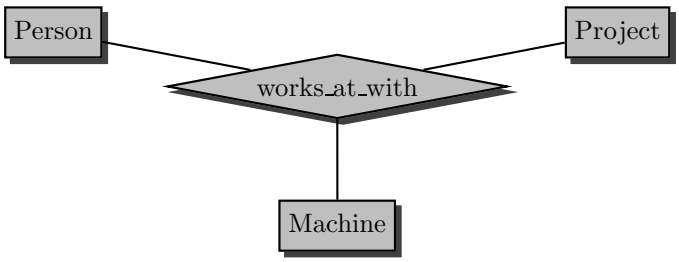
```
\begin{tabular}{c}
\entity{Person} \hspace*{6cm} \entity{Project} \\[2cm]
\entity{Machine}
\end{tabular}
\relationshipbetween{Person}{Project}{worksatwith}[works\_at\_with]
      /\ncarc[arcangle=-20]/
\inrelationship{Machine}{worksatwith}
```



Additionally, there is a straightforward extension of `\inrelationship` to relationship nodes which do not already exist but are set by the command:

```
\inrelationship{⟨entity-id⟩}(⟨role⟩)[⟨card⟩]/⟨connection⟩/
    {⟨relationship-id⟩}[⟨relationship-name⟩](⟨distance⟩,⟨angle⟩)
```

```
\begin{tabular}{c}
\entity{Person} \hspace*{6cm} \entity{Project} \\[2cm]
\entity{Machine}
\end{tabular}
\inrelationship{Machine}{worksatwith}[works\_at\_with](4em,90)
\inrelationship{Person}{worksatwith}
\inrelationship{Project}{worksatwith}
```



`\rolepos` `\cardpos` The position where roles and cardinalities are placed on the node connection is determined by `\rolepos` and `\cardpos` which can be set by `\rolepos{`⟨*number*⟩`}` and `\cardpos{`⟨*number*⟩`}`, where number underlies the same constraints as for `npos` in PSTricks (i.e., for `\ncline` and `\ncarc`, it has to be between 0 and 1, other values are allowed e.g. for `\ncbar`). The default setting is `\cardpos{0.5}` and `\rolepos{0.85}`.

1.4 Annotations to Objects

`\annote` With

```
\annote{⟨id⟩}{⟨text⟩}(⟨distance⟩,⟨angle⟩)
```

a node ⟨*id*⟩ can be annoted with a comment ⟨*text*⟩.

1.5 Usage

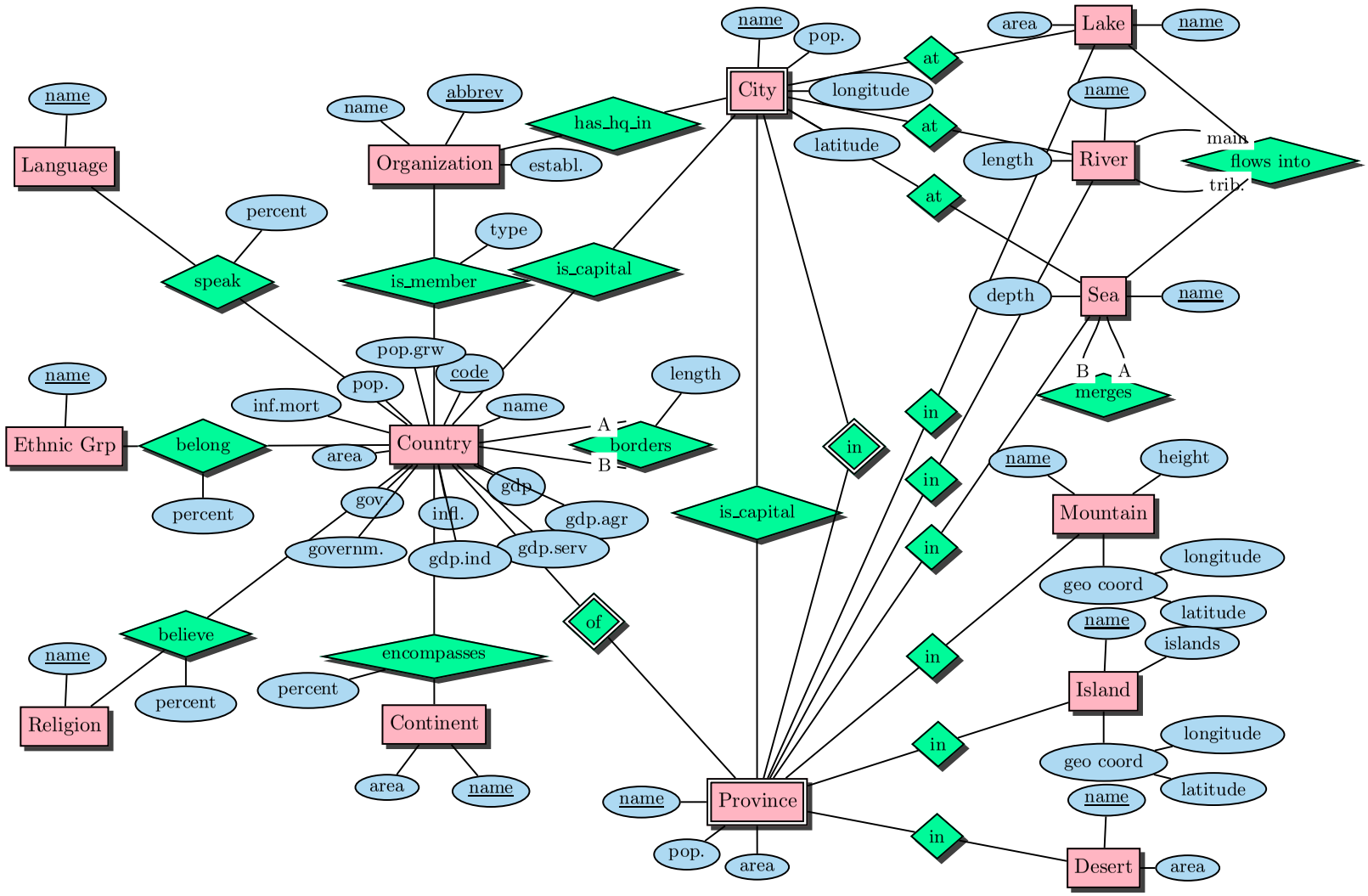
The package is intended to be used for two purposes:

- Typesetting small fragments of ER-diagrams, e.g., for lecture notes and slides: as shown in the above documentation, fragments of ER-diagrams can easily be integrated with the running text.
- Typesetting ER-Diagrams for project documentation: here the standard procedure is to design the ER-diagram based on a `tabular` environment in which the entity nodes are positioned; Then, attributes and relationships are positioned in the above declarative way. Using basic PSTricks commands, additional entity nodes can also be placed declaratively.

There are the following advantages compared to using graphics tools, e.g., xfig:

- the source code of the diagrams is written into the running source code of the document, thus there is no need for a bunch of separate `.fig` and `.eps` files.
- more flexibility wrt. renaming and layout changes.
- given the positions of some basic objects, the other objects are set with declarative commands, i.e. they automatically adjust when the positions change.

1.6 Example


name
pop.
City
area
Lake
name
at
longitude
name
has_hq_in
at
latitude
length
River
main
flows into
trib.
at
name
abbrev
Organization
establ.
Language
percent
type
speak
is_member
is_capital
depth
Sea
name
pop.grw
code
B
A
merges
name
pop.
length
inf.mort
name
A
borders
Ethnic Grp
belong
area
Country
B
in
in
in
in
in
in
in
in
gdp
gov
infl.
gdp.agr
percent
is_capital
name
height
Mountain
governm.
gdp.ind
gdp.serv
longitude
geo coord
latitude
name
islands
Island
believe
of
name
encompasses
percent
percent
Religion
Continent
longitude
geo coord
latitude
name
area
name
name
Province
Desert
pop.
area
area

1.7 Miscellaneous

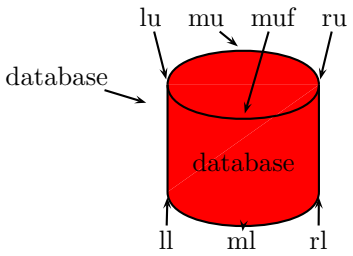
\nodeconnections With `\nodeconnections{`⟨*PSTricks-node-connections*⟩`}`, the PSTricks-node connections and label commands do not require any extra horizontal or vertical space.

\database The command

`\database[`⟨*pos*⟩`]{`⟨*width*⟩`}{`⟨*height*⟩`}[`⟨*graph_params*⟩`]{`⟨*text*⟩`}[`⟨*id*⟩`]`

defines a database barrel as a tabular which is vertically centered as given in the optional argument ⟨*pos*⟩ (Default: `[c]`). ⟨*text*⟩ is the text to be written on the front of the barrel, ⟨*width*⟩ and ⟨*height*⟩ give the width and height of the barrel; both must be a dimension, e.g., 2cm. With the optional graphics parameter ⟨*graph_params*⟩, the pstricks parameters fillstyle and fillcolor for the database can be set (see example below). The optional argument ⟨*id*⟩ is used as an internal name, it must start with a letter and must contain only letters and digits. ⟨*id*⟩ is used for `\rnode` definitions: ⟨*id*⟩ is the whole barrel `lu`⟨*id*⟩, `ru`⟨*id*⟩, `mu`⟨*id*⟩, and `muf`⟨*id*⟩ denote the points at the upper left corner, upper right corner, upper middle behind and aupper middle in front of the barrel. Analogously, `ll`⟨*id*⟩, `rl`⟨*id*⟩, and `ml`⟨*id*⟩, denote the lower left and lower right corner and lower middle in front of the barrel.

```
\psset{nodesep=2pt}
\begin{tabular}{lc}
  &\rnode{lu}{lu}\quad\rnode{mu}{mu}\quad
   \rnode{muf}{muf}\quad\rnode{ru}{ru}\\[1em]
  \rnode{db}{database}
   &\database[t]{2cm}{1cm}
       [fillstyle=solid,fillcolor=red]
       {database}[example] \\[5em]
  &\rnode{ll}{ll}\qquad\rnode{ml}{ml}
    \qquad\rnode{rl}{rl}
\end{tabular}
\nodeconnections{%
\ncline{->}{lu}{luexample}
\ncline{->}{mu}{muexample}
\ncline{->}{muf}{mufexample}
\ncline{->}{ru}{ruexample}
\ncline{->}{ll}{llexample}
\ncline{->}{ml}{mlexample}
\ncline{->}{db}{example}
\ncline{->}{rl}{rlexample}}
```



2 Code Documentation

Required packages from the PSTricks bundle

```
\RequirePackage{pstricks}
\RequirePackage{pst-node}
\typeout{Style `\basename', Version \fileversion\space <\filedate>}
\ProvidesPackage{pst-dbicons}[\filedate \space\fileversion]
```

\seticonparams `\seticonparams{#1}{#2}`

```
\def\seticonparams#1#2{\expandafter\def\csname #1@parm\endcsname{[#2]}}
\seticonparams{entity}{fillstyle=none}
\seticonparams{relationship}{fillstyle=none}
\seticonparams{attribute}{fillstyle=none}
```

\purifylabel Macro `\@purifylabel` expands the string given in `#2` and stores the result in command given in `#1`.
Supported TeX commands in string `#2`: `\_`, `\textunderscore` Example: `\@purifylabel{\templabel}{has\_hq\_in}`, then use `\templabel` as nodelabel for pstricks.

```
\def\dbi@purifylabel#1#2{%
  \begingroup
    \edef\_{\string_}%
    \edef\textunderscore{\string_}%
    \edef\x{\endgroup
      \def\noexpand#1{#2}%
    }%
  \x
}
```

\entity `\entity[#1a]{#1b}[#2]`

```
\def\entity{%
 \@ifnextchar[{\entity@i}{\entity@i[normal]}}
\def\entity@i[#1]#2{%
 \@ifnextchar[{\entity@ii{#1}{#2}}{\entity@ii{#1}{#2}[#2]}}
\def\entity@ii#1#2[#3]{\csname entity@#1\endcsname{#2}{#3}}
\def\entity@normal#1#2{%
  \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
  \rnode{\dbi@prfd@nodename}{\expandafter\psframebox\entity@parm{\strut#2}}}
\def\entity@weak#1#2{%
  {\psset{doubleline=true}%
   \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
   \rnode{\dbi@prfd@nodename}{\expandafter\psframebox\entity@parm{\strut#2}}}}
```

\attribute `\attribute[#1]{#2}[#3]`

```
\def\attribute{%
 \@ifnextchar[{\attribute@i}{\attribute@i[sv]}}
\def\attribute@i[#1]#2{%
 \@ifnextchar[{\attribute@ii{#1}{#2}}{\attribute@ii{#1}{#2}[#2]}}
\def\attribute@ii#1#2[#3]{\csname attribute@#1\endcsname{#2}{#3}}
\def\attribute@mv#1#2{{\psset{doubleline=true}%
  \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
  \expandafter\ovalnode\attribute@parm{\dbi@prfd@nodename}{#2}}}
\def\attribute@sv#1#2{%
  \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
  \expandafter\ovalnode\attribute@parm{\dbi@prfd@nodename}{#2}}
\def\attribute@key#1#2{
  \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
  \expandafter\ovalnode\attribute@parm{\dbi@prfd@nodename}{\underline{#2}}}
```

\attributeof \attributeof{#1}[#2]{#3}[#4]{#5}[#6]

```
\newdimen\@attrdist
\@attrdist2em  % Default value for distance of attribute from entity
\def\attrdist#1{\@attrdist#1} % set default distance

\def\attributeof#1{%
 \@ifnextchar[{\attributeof@i{#1}}{\attributeof@i{#1}[\attrdist]}}
\def\attributeof@i#1[#2]#3{%
 \@ifnextchar[{\attributeof@ii{#1}[#2]{#3}}{\attributeof@ii{#1}[#2]{#3}[sv]}}
\def\attributeof@ii#1[#2]#3[#4]#5{%
 \@ifnextchar[{\attributeof@iii{#1}[#2]{#3}[#4]{#5}}%
              {\attributeof@iii{#1}[#2]{#3}[#4]{#5}[#5]}}
\def\attributeof@iii#1[#2]#3[#4]#5[#6]{%
  \SpecialCoor
  \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
  \uput{#2}[#3]{0}(\dbi@prfd@nodename){\attribute[#4]{#5}[#6]}%
  \NormalCoor
  \ncline{-}{#1}{#5}}
```

\relationship \relationship[#1a]{#1b}[#2]

```
\def\relationship{%
 \@ifnextchar[{\relationship@i}{\relationship@i[normal]}}
\def\relationship@i[#1]#2{%
 \@ifnextchar[{\relationship@ii{#1}{#2}}{\relationship@ii{#1}{#2}[#2]}}
\def\relationship@ii#1#2[#3]{\csname relationship@#1\endcsname{#2}{#3}}
\def\relationship@normal#1#2{%
   \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
   \expandafter\dianode\relationship@parm{\dbi@prfd@nodename}{#2}}
\def\relationship@weak#1#2{%
  {\psset{doubleline=true}%
   \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
   \expandafter\dianode\relationship@parm{\dbi@prfd@nodename}{#2}}}
\let\relationship@ident\relationship@weak
```

elationshipbetween \relationshipbetween[#1a]{#1b}(#2)[#3]/#4/{#5}(#6)[#7]/#8/{#9}[#10]/#11/(#12)

```
\newtoks\rolepos
\rolepos{0.85}%
\newtoks\cardpos
\cardpos{0.5}%
\newif\ifdbi@recursive
\def\relationshipbetween{%
 \@ifnextchar[{\relationshipbetween@type}{\relationshipbetween@type[normal]}}
\def\relationshipbetween@type[#1]#2{\dbi@recursivefalse%
 \def\relationtype{#1}%
 \@ifnextchar({\relationshipbetween@i@role{#2}}%
              {\relationshipbetween@i@role{#2}(\relax)}}
\def\relationshipbetween@i@role#1(#2){%
 \@ifnextchar[{\relationshipbetween@i@card{#1}{#2}}%
              {\relationshipbetween@i@card{#1}{#2}[\relax]}}
\def\relationshipbetween@i@card#1#2[#3]{%
```

```
 \@ifnextchar/{\relationshipbetween@i@linetype{#1}{#2}{#3}}%
              {\relationshipbetween@i@linetype{#1}{#2}{#3}/\relax/}}%
\def\relationshipbetween@i@linetype#1#2#3/#4/#5{%
 \def\dbi@linecmd@i{#4}%
 \def\dbi@tempa{#1}\def\dbi@tempb{#5}%
 \ifx\dbi@tempa\dbi@tempb\dbi@recursivetrue\fi
 \ifx#4\relax
    \ifdbi@recursive
       \def\dbi@linecmd@i{\ncarc[arcangleA=10,arcangleB=45]}%
      \else\def\dbi@linecmd@i{\ncline}\fi\fi
 \@ifnextchar({\relationshipbetween@ii@role{#1}{#2}{#3}{#5}}%
              {\relationshipbetween@ii@role{#1}{#2}{#3}{#5}(\relax)}}%
\def\relationshipbetween@ii@role#1#2#3#4(#5){%
 \@ifnextchar[{\relationshipbetween@ii@card{#1}{#2}{#3}{#4}{#5}}%
              {\relationshipbetween@ii@card{#1}{#2}{#3}{#4}{#5}[\relax]}}
\def\relationshipbetween@ii@card#1#2#3#4#5[#6]{%
 \@ifnextchar/{\relationshipbetween@ii@linetype{#1}{#2}{#3}{#4}{#5}{#6}}%
              {\relationshipbetween@ii@linetype{#1}{#2}{#3}{#4}{#5}{#6}%
               /\relax/}}%
\def\relationshipbetween@ii@linetype#1#2#3#4#5#6/#7/#8{%
 \def\dbi@linecmd@ii{#7}%
 \ifx#7\relax
    \ifdbi@recursive
       \def\dbi@linecmd@ii{\ncarc[arcangleA=-10,arcangleB=-45]}%
      \else\def\dbi@linecmd@ii{\ncline}\fi\fi
 \@ifnextchar[{\relationshipbetween@optname{#1}{#2}{#3}{#4}{#5}{#6}{#8}}%
              {\relationshipbetween@optname{#1}{#2}{#3}{#4}{#5}{#6}{#8}[#8]}}%
\def\relationshipbetween@optname#1#2#3#4#5#6#7[#8]{%
 \@ifnextchar/{\relationshipbetween@linetype{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}}%
              {\relationshipbetween@linetype{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}%
               /\ncline/}}%
\def\relationshipbetween@linetype#1#2#3#4#5#6#7#8/#9/{%
 \def\dbi@linecmd{#9}%
 \@ifnextchar({\relationshipbetween@pos{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}}%
              {\relationshipbetween@pos{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}(0.5)}}%
\def\relationshipbetween@pos#1#2#3#4#5#6#7#8(#9){%
   \ifdbi@recursive
     \relationshipbetween@rec{#1}{#2}{#3}{#5}{#6}{#7}{#8}(#9)\else
     \relationshipbetween@nonrec{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}(#9)\fi}

\def\relationshipbetween@nonrec#1#2#3#4#5#6#7#8(#9){%
 \dbi@purifylabel{\dbi@prfd@nodename@i}{#1}%
 \dbi@purifylabel{\dbi@prfd@nodename@ii}{#4}%
 \dbi@purifylabel{\dbi@prfd@nodename@iii}{#7}%
 {\psset{linestyle=none}%
  \dbi@linecmd{-}{\dbi@prfd@nodename@i}{\dbi@prfd@nodename@ii}}%
 \ncput[npos=#9]{\relationship[\relationtype]{#7}[#8]}%
 \dbi@linecmd@i{-}{\dbi@prfd@nodename@i}{\dbi@prfd@nodename@iii}%
  \ifx#3\relax\else\ncput*[npos=\the\cardpos]{#3}\fi
  \ifx#2\relax\else\ncput*[npos=\the\rolepos]{#2}\fi
 \dbi@linecmd@ii{-}{\dbi@prfd@nodename@ii}{\dbi@prfd@nodename@iii}%
```

```
  \ifx#6\relax\else\ncput*[npos=\the\cardpos]{#6}\fi
  \ifx#5\relax\else\ncput*[npos=\the\rolepos]{#5}\fi}

\def\relationshipbetween@rec#1#2#3#4#5#6#7(#8,#9){%
  \dbi@purifylabel{\dbi@prfd@nodename@i}{#1}%
  \dbi@purifylabel{\dbi@prfd@nodename@ii}{#6}%
  \SpecialCoor
  \uput{#8}[#9]{0}(\dbi@prfd@nodename@i)%
         {\relationship[\relationtype]{#6}[#7]}%
  \NormalCoor
  \dbi@linecmd@i{-}{\dbi@prfd@nodename@i}{\dbi@prfd@nodename@ii}%
   \ifx#3\relax\else\ncput*[npos=\the\cardpos]{#3}\fi
   \ifx#2\relax\else\ncput*[npos=\the\rolepos]{#2}\fi
  \dbi@linecmd@ii{-}{\dbi@prfd@nodename@i}{\dbi@prfd@nodename@ii}%
   \ifx#5\relax\else\ncput*[npos=\the\cardpos]{#5}\fi
   \ifx#4\relax\else\ncput*[npos=\the\rolepos]{#4}\fi}
```

\inrelationship \inrelationship[#1a]{#1b}(#2)[#3]/#4/{#5}[#6](#7,#8)

```
\def\inrelationship{%
 \@ifnextchar[{\inrelationship@type}{\inrelationship@type[normal]}}
\def\inrelationship@type[#1]#2{%
 \def\relationtype{#1}%
 \@ifnextchar({\inrelationship@role{#2}}%
              {\inrelationship@role{#2}(\relax)}}
\def\inrelationship@role#1(#2){%
 \@ifnextchar[{\inrelationship@card{#1}{#2}}%
              {\inrelationship@card{#1}{#2}[\relax]}}
\def\inrelationship@card#1#2[#3]{%
 \@ifnextchar/{\inrelationship@linetype{#1}{#2}{#3}}%
              {\inrelationship@linetype{#1}{#2}{#3}/\ncline/}}%
\def\inrelationship@linetype#1#2#3/#4/#5{%
 \dbi@purifylabel{\dbi@prfd@nodename@i}{#1}%
 \dbi@purifylabel{\dbi@prfd@nodename@ii}{#5}%
 \def\dbi@linecmd{#4}%
 \@ifnextchar[{\inrelationship@newrel{#1}{#2}{#3}{#5}}%
              {\@ifnextchar({%
                \inrelationship@newrel{#1}{#2}{#3}{#5}[#5]}%
               {\dbi@linecmd{-}{\dbi@prfd@nodename@i}{\dbi@prfd@nodename@ii}
                \ifx#3\relax\else\ncput*[npos=\the\cardpos]{#3}\fi
                \ifx#2\relax\else\ncput*[npos=\the\rolepos]{#2}\fi}}%

\def\inrelationship@newrel#1#2#3#4[#5]{%
 \@ifnextchar({\inrelationship@newrel@pos{#1}{#2}{#3}{#4}{#5}}%
              {\PackageError{\basename}{Position of relationship #4
                undefined.}\@eha}}
\def\inrelationship@newrel@pos#1#2#3#4#5(#6,#7){%
  \SpecialCoor
  \dbi@purifylabel{\dbi@prfd@nodename@i}{#1}%
  \dbi@purifylabel{\dbi@prfd@nodename@ii}{#4}%
  \uput{#6}[#7]{0}(\dbi@prfd@nodename@i){\relationship[\relationtype]{#4}[#5]}%
  \NormalCoor
```

```
  \dbi@linecmd{-}{\dbi@prfd@nodename@i}{\dbi@prfd@nodename@ii}
  \ifx#3\relax\else\ncput*[npos=\the\cardpos]{#3}\fi
  \ifx#2\relax\else\ncput*[npos=\the\rolepos]{#2}\fi}
```

\annote

```
\def\annote#1#2(#3,#4){%
  \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
  \SpecialCoor
  \uput{#3}[#4]{0}(\dbi@prfd@nodename){#2}%
  \NormalCoor}
```

\nodeconnections

```
\def\nodeconnections#1{\hbox to 0cm{\vbox to 0cm {#1}}}
```

\database \database[#1]{#2}{#3}[#4]{#5}[#6]

```
\def\database{\@ifnextchar[{\database@i}{\database@i[c]}}
\def\database@i[#1]#2#3{%
 \@ifnextchar[{\database@ii{#1}{#2}{#3}}%
              {\database@ii{#1}{#2}{#3}[fillstyle=none]}}
\def\database@ii#1#2#3[#4]#5{%
 \@ifnextchar[{\database@iii{#1}{#2}{#3}{#4}{#5}}%
              {\database@iii{#1}{#2}{#3}{#4}{#5}[#5]}}
\def\database@iii#1#2#3#4#5[#6]{%
 \psset{nodesep=0pt}%
 \dbi@purifylabel{\dbi@prfd@nodename}{#6}%
 \rnode{\dbi@prfd@nodename}{%
 \begin{tabular}[#1]{c}%
 \rnode{lu\dbi@prfd@nodename}{}\hspace{#2}\rnode{ru\dbi@prfd@nodename}{}\\[#3]
 \rnode{ll\dbi@prfd@nodename}{}\hspace{#2}\rnode{rl\dbi@prfd@nodename}{}%
 \end{tabular}}%
 \nodeconnections{%
 \ncbar[linestyle=none,#4,
        angleA=180,angleB=180,armB=0]{ru\dbi@prfd@nodename}{ll\dbi@prfd@nodename}
 \ncbar[linestyle=none,#4,
        angleA=0,angleB=0,armA=0]{ru\dbi@prfd@nodename}{ll\dbi@prfd@nodename}
 \nccurve[#4,angleA=90,angleB=90,ncurv=.6]{lu\dbi@prfd@nodename}{ru\dbi@prfd@nodename}%
      \ncput{\rnode{mu\dbi@prfd@nodename}{}}%
 \nccurve[angleA=-90,angleB=-90,ncurv=.6]{lu\dbi@prfd@nodename}{ru\dbi@prfd@nodename}%
      \ncput{\rnode{muf\dbi@prfd@nodename}{}}%
 \nccurve[#4,angleA=-90,angleB=-90,ncurv=.6]{ll\dbi@prfd@nodename}{rl\dbi@prfd@nodename}%
      \ncput{\rnode{ml\dbi@prfd@nodename}{}}%
 \ncline{lu\dbi@prfd@nodename}{ll\dbi@prfd@nodename}%
 \ncline{ru\dbi@prfd@nodename}{rl\dbi@prfd@nodename}%
 \ncline[linestyle=none]{muf\dbi@prfd@nodename}{ml\dbi@prfd@nodename}\ncput[npos=0.4]{#5}}}
```

2.1 Acknowledgements

Thanx to Heiko Oberdiek for adding the handling of ids for internal use in postcsript (used in the `purifylabel` macro).

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.