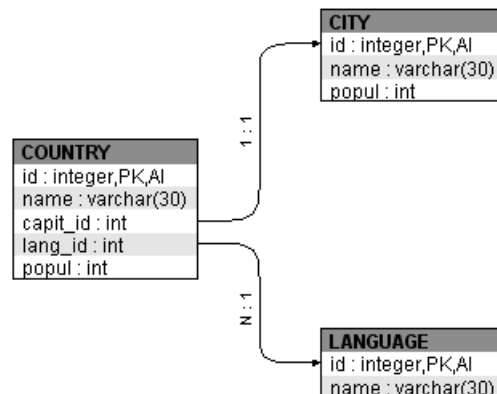


## Tutorial for: **DBASE DIAGRAMS** Version: 0.1

Filip Král  
vydramail <at> quick <dot> cz  
www.filipkral.wz.cz

The usage of functions in `dbase_diagram.r` is presented in this tutorial on a simple example. We are trying to create the following picture. Note that it is wise to write all the following commands as a new script. It is impossible to undo any steps when writing directly into console. When writing commands into a script, you can more easily run all the commands over and over any time you make a mistake.



To load the functions for drawing diagrams, place the file `dbase_diagram.r` into your R working directory and type into R console (or script):

```
source("diagrams.r")
```

We need to create vectors of items for each box. Note that the first item will be used as a caption later.

```
cnt<-c("COUNTRY", "id : integer,PK,AI", "name : varchar(30)", "capit_id : int", "lang_id :  
int", "popul : int")  
cty<-c("CITY", "id : integer,PK,AI", "name : varchar(30)", "popul : int")  
lng<-c("LANGUAGE", "id : integer,PK,AI", "name : varchar(30)")
```

Now, let's create graphical objects – the boxes. You have to specify on what coordinates they have to appear. Coordinates must be from 0 to 1. The coordinates you set are the coordinates of the centre of the box.

```
country<-boxGrob(cnt, x=0.3, y=0.5)  
city<-boxGrob(cty, x=0.7, y=0.7)  
language<-boxGrob(lng, x=0.7, y=0.3)
```

Finally, we can draw our boxes:

```
grid.draw(country); grid.draw(city); grid.draw(language)
```

At this point, one can simply stop using R and finish the diagram in some graphical editor (e.g., Inkscape – free vector editor, or GIMP – free raster editor). You should preferably use these tools to draw connections, lines and other stuff, if your schema is complicated. However, there are functions in **DBASE DIAGRAMS** that allow you to draw some lines and add some text to your schema.

You can connect items on different boxes very quickly by using the “connect” function:

```
connect(box1, box2, lab1, lab2, arr=arrow(type="closed", angle=15, length=unit(2,"mm")), dir="up")
```

The arguments are: `box1`, `box2` are names of the boxes to connect; `lab1`, `lab2` are item numbers to be connected, `lab1` is on `box1`, `lab2` is on `box2`, caption has item number of zero. You can define your own arrow to appear near the `box2`. If your line should go from up to down, set the parameter `dir="down"`. The function `connect` will work properly only if the boxes are in appropriate relative positions. The user should be able to figure out quickly, when this function can not be used.

In our example, we type:

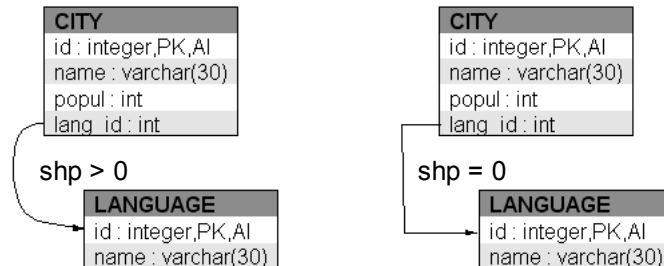
```
connect(country, city, 3, 1)
connect(country, language, 4, 1, dir="down")
```

When you can not achieve acceptable results with the function “connect”, there is another possibility.

Function “drawLine” is more flexible (at least in some way):

```
drawLine(snap=0,shp=1,arr=arrow(type="closed", angle=15, length=unit(2,"mm")))
```

This function allows you to draw a line by hand. You can define your own ending arrow. Setting parameter snap to some float (recommended are values around 0.02) will cause snapping of vertexes to an imaginary grid (the snap value is the spacing of the rectangular grid). Snapping does not affect the first and the last point. You will end the line by clicking twice at the same place. Different values of parameter shp with snap=0 are shown on the following figure (negative shp is possible, but not recommended):



Sometimes, it is useful to add some text, e.g., just as we do in our example, to emphasize the relations between tables. Use function “addText” for this purpose:

```
addText(txt="1:N", font="", fsize=1, angle=0, x=-1, y=-1)
```

Parameters are: txt is a string representing the text you want to display. Parameter font can be set to “b”, “bold”, “i” or “italic”. Font size can be modified by setting the fsize parameter to a multiple of the default font size. Parameters x and y are the coordinates of the centre of the text. If either x, or y is specified, you must specify the second coordinate by clicking on the canvas. If both, x and y are not set, than you have to specify the exact location of the text manually by clicking on the canvas. The parameter angle should be clear from the context.

We have added the description of relations by the following commands:

```
addText("1 : 1", font="b", fsize=0.8, angle=90, x=0.48, y=0.6)
addText("N : 1", font="b", fsize=0.8, angle=90, x=0.48, y=0.37)
```

Presented functions don not allow you to automatically generate a schema of a complicated database. However, with DBASE DIAGRAM functions, schemas of databases with a few tables can be drawn in a very short time. Hence, for small projects, like students projects etc., DBASE DIAGRAM can be beneficial. By using some additional free software, like Inkscape, to draw the connections, user can achieve a diagram with pretty professional look. Furthermore, the code can be extended by the user to satisfy special requirements.

Big part of code of DBASE DIAGRAM has been derived from an article:

Murrel, P. (2009): Drawing Diagrams with R, The R Journal Vol. 1/1 May 2009. ISSN 2073-4859

R-project:

<http://www.r-project.org/> [on-line: 2009/06/22 ]

Inkscape – free vector graphical editor:

<http://www.inkscape.org/> [on-line: 2009/06/22 ]

GIMP – free raster graphical editor:

<http://www.gimp.org/> [on-line: 2009/06/22 ]