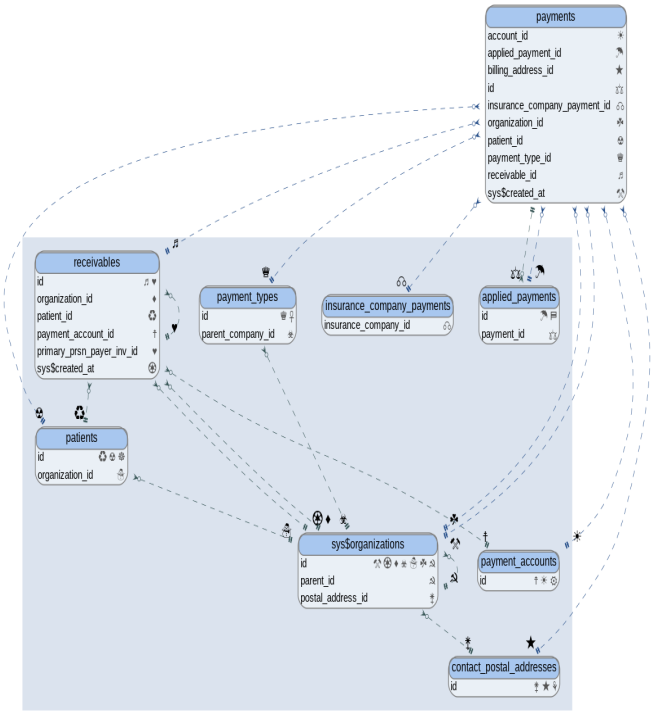
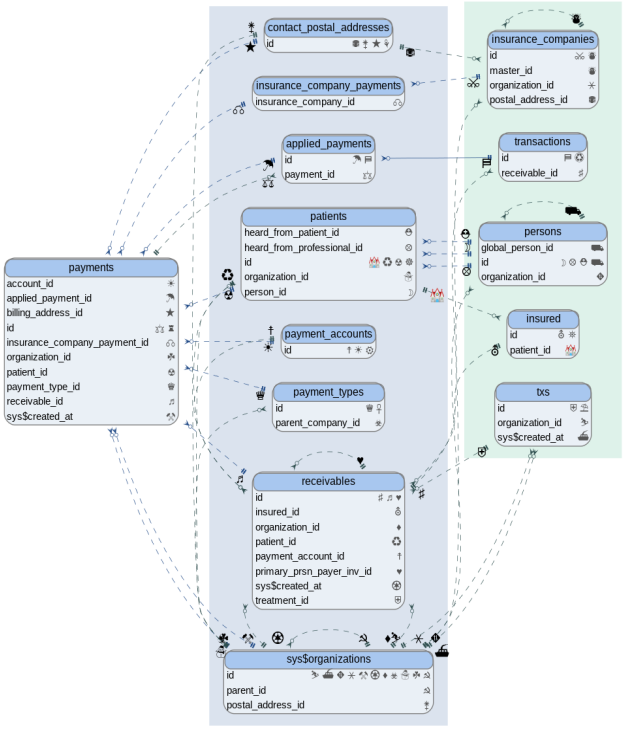
ER schema visualization tool ERvizauto v0.2.

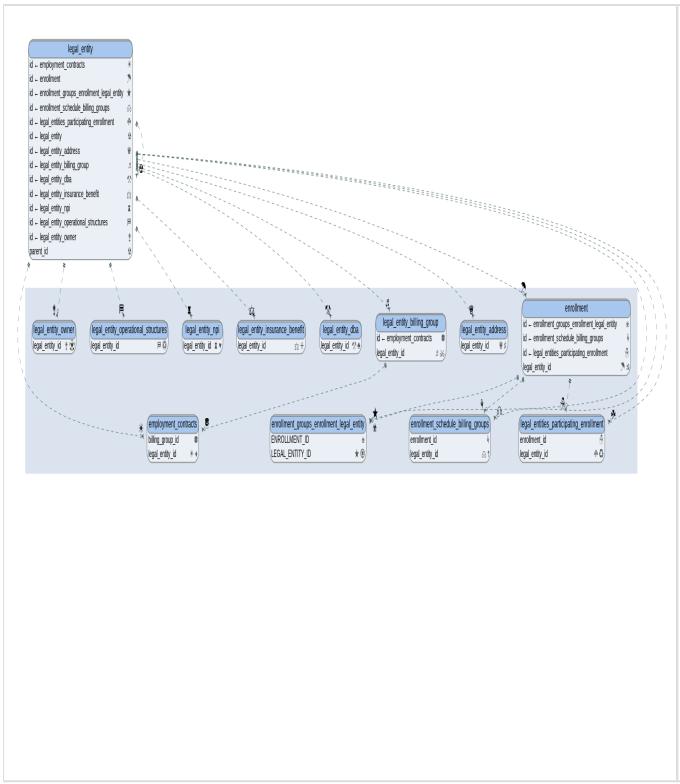
Does:

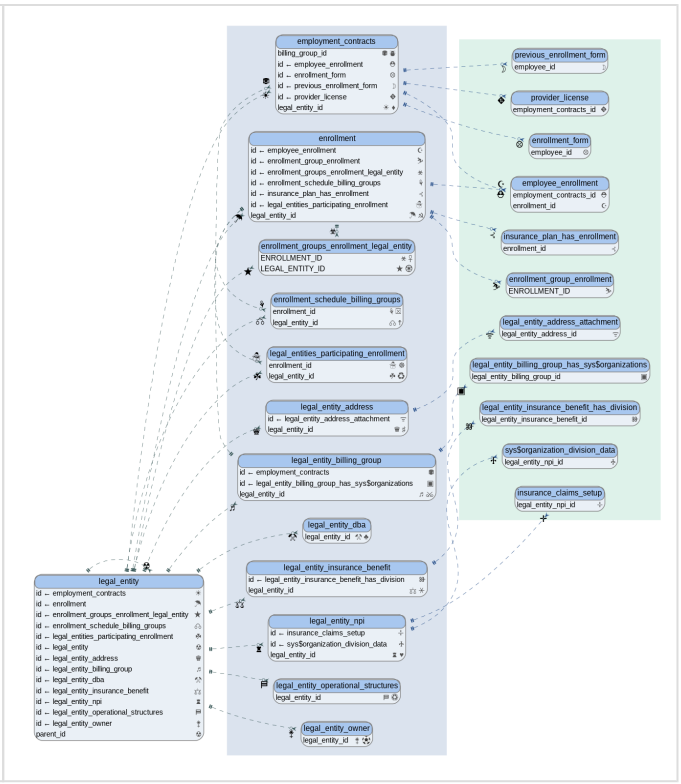
> 1. *Generates all 1st and 2nd level ERs for a given DB: forward, i.e. from child to parents, and backward, i.e. from parent to children.*
> 2. *Computes connection paths between 2 given tables.*

Results:

Examples

1st level FK-connections (child to parents):	2nd level FK-connections (child to parents):
	
1st level backward FK-connections (parent to children):	**2nd level backward FK-connections (parent to children):**





Prerequisites:

```
sudo apt-get install libmysqlclient-dev
sudo apt-get install python-dev
```

— these are possible prerequisites for debian/ubuntu, mysql-server & mysql-client supposed to be installed;
packages like build-essential autoconf libtool pkg-config might be required as well

```
sudo pip --trusted-host pypi.org --trusted-host files.pythonhosted.org install mysql
sudo pip --trusted-host pypi.org --trusted-host files.pythonhosted.org install mysqlclient (optional)
sudo pip install --upgrade --force-reinstall --trusted-host pypi.org --trusted-host files.pythonhosted.org
mysql-python
in mysql root console:
    CREATE DATABASE vaxiom; -- unless DB exists
    GRANT ALL PRIVILEGES ON *.* TO 'beewhiz'@'localhost' IDENTIFIED BY 'flowers2b';
    FLUSH PRIVILEGES;
sudo mysql vaxiom < structure_only.sql  (unless DB exists, load it's structure or make a full dump
restore)
sudo apt install graphviz
sudo pip install graphviz
sudo pip install pytablewriter

DBsql.py
 stored in the same folder
config.py
 stored in the same folder, contains configuration to connect to the DB and necessary paths:
```

```
DB = {'host': 'localhost', 'user': 'beewhiz', 'password': 'flowers2b',
'dbname': 'vaxiom'}
SQLPATH = '/path_to_sql/sql/'
PNGPATH = '/path_to_png/png/'
PDFPATH = '/path_to_pdf/pdf/'
```

```
vzdia.py
 stored in the same folder, contains diagram_draw() method and all related ones
```

Commits:

On bitbucket, see there DBsql.py, config.py and schemaviz.py

Location and main files:

```
in /axpm/data-mine/:
schemaviz.py
vzdia.py
DBsql.py
config.py
dataservices.py
```

Uses:

`SQL query from /axpm/data-mine/sql/fk_list.sql`

Options:

arguments:	default:	making:
`--level,-l [number 1 or 2]`	1	to generate PNG images of all tables along with the tables connected to a given table via foreign keys directly (1) or directly + next such connections (2)
`--way, -w [number 1 or -1]`	0	the direction of relationship: 1: for child to parent (from FK to parent column/table); -1: for parent to child 0: draws nothing, outputs db basic statistics
`--tables, -t [two comma separated tables]`	-	draws full relationship path(s) between two given tables and outputs basic statistics on the database
`--number, -n [number]`	10	no greater than this maximum number of joints in the computed paths between given tables
`--depth, -d [depth]`	10	this value is opting for clustering, by default it assumes that a cluster's center has no less than 10 connections (with parents and children tables) , the less is this number, the harder is load for the clusters graph calculations
`--clean, -c [cleaning a full DB snapshot ]`	0	if set 0, then draws full schema, if set 1 — removes the annotations from cluster core tables, if set 2 — excludes the most connected table from the schema (to make other connections more perceptible)

Example of usage:

```
$ python schemaviz.py --level 2 --way -1
or
$ python schemaviz.py -l 2 -w -1
$ python schemaviz.py -t table_A,table_B -n5 -d8 -c2
```

TODO & Remarks:

1. Icon set verification when a relationships number is too big (like parent children relationships for sys$organizations)
2. code optimization (a separate clase for icons set assignment at least)
3. make a package