

PILAR

Usage of a Database to Store Risk Analysis Projects

13.11.2009

Table of Contents

1	INTRODUCTION	1
2	ASSUMPTIONS	2
3	DATABASE TABLES	3
4	INPUT DATA	4
4.1	MODEL (A.K.A. PROJECT).....	4
4.1.1	<i>Model information</i>	5
4.2	SOURCES OF INFORMATION	5
4.3	DOMAINS	5
4.4	DOMAIN VALUATION.....	5
4.5	LAYERS.....	6
4.6	ASSETS	7
4.6.1	<i>Asset information</i>	7
4.6.2	<i>Sources of information</i>	8
4.6.3	<i>Asset dependencies</i>	8
4.6.4	<i>Asset valuation</i>	8
4.7	THREATS.....	9
4.8	PHASES.....	9
4.9	SAFEGUARDS	9
4.9.1	<i>Maturity</i>	9
4.9.2	<i>Flags</i>	10
4.9.3	<i>Comments</i>	10
4.9.4	<i>References</i>	11
4.10	SAFEGUARD CODES.....	12
4.11	VALUE SCALE.....	12
5	CALCULATED VALUES	13
5.1	RISK ANALYSIS: IMPACT AND RISK.....	13
5.1.1	<i>Accumulated values</i>	13
5.1.2	<i>Deflected values</i>	13
5.2	BUSINESS CONTINUITY: IMPACT AND RISK.....	14
5.2.1	<i>Accumulated values</i>	14
5.2.2	<i>Deflected values</i>	14
5.3	SECURITY PROFILES.....	15
6	LIBRARY	16
6.1	CORE ELEMENTS.....	16
6.2	EXTENSIONS.....	17

1 Introduction

Traditionally, PILAR saves the projects in a single file, .MGR. This approach has a number of benefits:

- ❑ files are encrypted (protected by a pass phrase): key management policy is under full control of the analyst
- ❑ files are very compact
- ❑ file structure is not public, in order to simplify its design and evolution
- ❑ it is very easy to work on a single portable computer
- ❑ it is very easy to share, exchanging messages over email

But the approach introduces a number of limits for working on big organizations:

- ❑ it is not easy to collaborate with other tools, in particular
 - inventories
 - service data bases (e.g. ITIL CMDBs)
- ❑ it is not easy to exploit results
- ❑ it is not easy to work in teams, where different people care after different parts of the analysis

The introduction of a database as an external repository of information addresses those concerns.

2 Assumptions

There are many database products available, either freeware, or commercial ones. PILAR should work with any, to the largest extend.

The access to the database will be over the network, so PILAR may be used on any computer that has access to the server.

The permissions to access data are not longer under control of PILAR, but will be transferred to the administrator of the database.

One project in PILAR is mapped into one database. If there are N models, there will be N databases. The administrator shall configure the access rights of the users.

PILAR will use JDBC to connect to the database. So, the administrator is expected to provide java classes to access the database over the net using JDBC.

Perhaps this may help:

<http://developers.sun.com/product/jdbc/drivers>

3 Database tables

```
mysql> show tables;
+-----+
| apps          |
| assetinfo    |
| assetkb      |
| assets       |
| assetsafeguards |
| assetsources |
| bcmdown1     |
| bcmdown2     |
| bcmup1       |
| bcmup2       |
| comments     |
| domainkb     |
| domains      |
| domainsafeguards |
| domainvaluebc |
| domainvaluera |
| layers       |
| links        |
| maturity     |
| model        |
| modelinfo    |
| phases       |
| profiles     |
| riskdown1    |
| riskdown2    |
| riskup1      |
| riskup2      |
| sources      |
| threatdegradation |
| threats      |
| valuebc      |
| valuera      |
| valuescale   |
+-----+
| history      |
| preferences  |
+-----+
```

As a general rule,

- codes are VARCHAR(32),
- names are VARCHAR(128), and
- long texts are TEXT.

Table summary

Input Data	Calculated Values
model	riskdown1
modelinfo	riskdown2
sources	riskup1
domains	riskup2
domainvaluera	bcmdown1
domainvaluebc	bcmdown2
layers	bcmup1
assets	bcmup2
assetinfo	
assetsources	27002_2005_apps
links	27002_2005_value
valuera	
valuebc	
threats	
threatdegradation	
phases	
maturity	
apps	
comments	
domainsafeguards	
assetsafeguards	
profiles	
domainkb	
assetkb	
safeguards	
valuescale	

4 Input data

4.1 Model (a.k.a. project)

```
mysql> describe model;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code           | varchar(32)   | YES  |     | NULL    |       |
| name           | varchar(128)  | YES  |     | NULL    |       |
| description    | text          | YES  |     | NULL    |       |
| lib            | varchar(32)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

There is a single row in this table.

4.1.1 Model information

```
mysql> describe modelinfo;
+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| mykey      | varchar(32)    | YES  |     | NULL    |       |
| val        | varchar(128)  | YES  |     | NULL    |       |
| position   | int(11)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as administrative lines in the project.

POSITION is a relative ordering of the rows.

4.2 Sources of information

```
mysql> describe sources;
+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code       | varchar(32)    | NO   | PRI | NULL    |       |
| name       | varchar(128)  | YES  |     | NULL    |       |
| description | text           | YES  |     | NULL    |       |
| position   | int(11)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as sources.

POSITION is a relative ordering of the rows.

4.3 Domains

```
mysql> describe domains;
+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code       | varchar(32)    | NO   | PRI | NULL    |       |
| next       | varchar(32)    | YES  |     | NULL    |       |
| name       | varchar(128)  | YES  |     | NULL    |       |
| qualifiers | text           | YES  |     | NULL    |       |
| description | text           | YES  |     | NULL    |       |
| position   | int(11)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as domains.

POSITION is a relative ordering of the rows.

QUALIFIERS is a comma-separated list of domain qualifier codes.

4.4 Domain valuation

There are two tables,

- domainvaluera for normal risk analysis
- domainvaluebc for business continuity

```
mysql> describe domainvaluera;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| domain     | varchar(32)   | YES  |     | NULL    |       |
| dimension  | varchar(8)    | YES  |     | NULL    |       |
| level      | int(11)       | YES  |     | NULL    |       |
| value      | float         | YES  |     | NULL    |       |
| criteria   | text          | YES  |     | NULL    |       |
| comment    | text          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as value entries.

CRITERIA is a comma-separated list of criteria codes.

```
mysql> describe domainvaluebc;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| domain     | varchar(32)   | YES  |     | NULL    |       |
| step       | int(11)       | YES  |     | NULL    |       |
| level      | int(11)       | YES  |     | NULL    |       |
| value      | float         | YES  |     | NULL    |       |
| criteria   | text          | YES  |     | NULL    |       |
| comment    | text          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as value entries.

CRITERIA are encoded as above.

4.5 Layers

```
mysql> describe layers;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code       | varchar(32)   | NO   | PRI | NULL    |       |
| name       | varchar(128)  | YES  |     | NULL    |       |
| description | text          | YES  |     | NULL    |       |
| position   | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as layers.

POSITION is a relative ordering of the layers. Layers with a lower position shall appear before (in a previous row in PILAR GUI) than those with a higher row. It is a loose ordering of rows.

4.6 Assets

```
mysql> describe assets;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code       | varchar(32)   | NO   | PRI | NULL    |      |
| name       | varchar(128)  | YES  |     | NULL    |      |
| description | text         | YES  |     | NULL    |      |
| domain     | varchar(32)   | YES  |     | NULL    |      |
| layer      | varchar(32)   | YES  |     | NULL    |      |
| father     | varchar(32)   | YES  |     | NULL    |      |
| position   | int(11)       | YES  |     | NULL    |      |
| classes    | text         | YES  |     | NULL    |      |
| extra      | text         | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as assets, either groups or real assets.

Assets have a FATHER if they belong to a group.

POSITION is a relative ordering of the assets. Assets with a lower position shall appear before (in a previous row in PILAR GUI) than those with a higher row. It is a loose ordering of rows.

CLASSES is a comma-separated list of asset classes.

EXTRA encodes additional flags that qualify the asset. The format is a semicolon-ended list of key-value pairs. For instance

```
E=false;V=false;
```

EXTRA is expected to grow in the future. Currently, the following keys are recognized:

key	type	default	meaning
E	boolean	true	if the asset exists
TH	boolean	true	if the asset has threats
V	boolean	true	if the asset is visible
AV	boolean	true	if the asset is available

4.6.1 Asset information

```
mysql> describe assetinfo;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| asset      | varchar(32)   | YES  |     | NULL    |      |
| mykey      | varchar(32)   | YES  |     | NULL    |      |
| val        | varchar(128)  | YES  |     | NULL    |      |
| position   | int(11)       | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as lines of information.

POSITION is a relative ordering of the rows.

4.6.2 Sources of information

```
mysql> describe assetsources;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| asset | varchar(32)   | YES  |     | NULL    |       |
| source | varchar(32)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as sources.

4.6.3 Asset dependencies

```
mysql> describe links;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| asset      | varchar(32)   | YES  |     | NULL    |       |
| below      | varchar(32)   | YES  |     | NULL    |       |
| expression | text          | YES  |     | NULL    |       |
| comment    | text          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as connections between two assets.

The EXPRESSION is encoded as in PILAR GUI:

encoding	meaning
	100% over all dimensions
50%	50% over all dimensions
C:10%/I:100%	10% for [C], 100% for [I], 0% for those not mentioned.

4.6.4 Asset valuation

There are two tables involved:

- assetvaluera for normal risk analysis
- assetvaluebc for business continuity

```
mysql> describe valuera;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| asset      | varchar(32)   | YES  |     | NULL    |       |
| dimension  | varchar(8)    | YES  |     | NULL    |       |
| level      | int(11)       | YES  |     | NULL    |       |
| value      | float         | YES  |     | NULL    |       |
| criteria   | text          | YES  |     | NULL    |       |
| comment    | text          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as value entries.

The CRITERIA are encoded as in domains.

```
mysql> describe valuebc;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| asset      | varchar(32)   | YES  |     | NULL    |       |
| step       | int(11)       | YES  |     | NULL    |       |
| level      | int(11)       | YES  |     | NULL    |       |
| value      | float         | YES  |     | NULL    |       |
| criteria   | text          | YES  |     | NULL    |       |
| comment    | text          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

There shall be as many rows as value entries.

The CRITERIA are encoded as in domains.

4.7 Threats

```
mysql> describe threats ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| asset      | varchar(32)   | YES  |     | NULL    |       |
| threat     | varchar(32)   | YES  |     | NULL    |       |
| freq       | float         | YES  |     | NULL    |       |
| step       | int(11)       | YES  |     | NULL    |       |
| comment    | text          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

```
mysql> describe threatdegradation ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| asset      | varchar(32)   | YES  |     | NULL    |       |
| threat     | varchar(32)   | YES  |     | NULL    |       |
| dimension  | varchar(8)    | YES  |     | NULL    |       |
| deg        | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

4.8 Phases

```
mysql> describe phases ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code       | varchar(32)   | NO   | PRI | NULL    |       |
| prev       | varchar(32)   | YES  |     | NULL    |       |
| name       | varchar(128)  | YES  |     | NULL    |       |
| description | text          | YES  |     | NULL    |       |
| position   | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

POSITION is a relative ordering of the rows.

4.9 Safeguards

4.9.1 Maturity

Maturity is encoded as follows

char(1)	maturity
0	L0
1	L1
2	L2
3	L3
4	L4
5	L5
n	n.a.
?	...?
-	no info

```
mysql> describe maturity;
```

Field	Type	Null	Key	Default	Extra
reference	int(11)	YES		NULL	
phase	varchar(32)	YES		NULL	
maturity	char(1)	YES		NULL	

REFERENCE: see 4.9.4.

4.9.2 Flags

In this section:

- applies is a boolean: TRUE if applies (default is true)
- active is a boolean: TRUE if it is on (default is true)
- doubts is a boolean: TRUE if there are doubts (default is false)

```
mysql> describe apps;
```

Field	Type	Null	Key	Default	Extra
reference	int(11)	NO	PRI	NULL	
applies	tinyint(1)	YES		NULL	
active	tinyint(1)	YES		NULL	
doubt	tinyint(1)	YES		NULL	

REFERENCE: see 4.9.4.

4.9.3 Comments

```
mysql> describe comments;
```

Field	Type	Null	Key	Default	Extra
reference	int(11)	NO	PRI	NULL	
comment	text	YES		NULL	

REFERENCE: see 4.9.4.

4.9.4 References

The references in previous sections are established as follows.

Safeguards per domain:

```
mysql> describe domainsafeguards;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code      | varchar(32)   | YES  |     | NULL    |      |
| domain     | varchar(32)   | YES  |     | NULL    |      |
| reference  | int(11)       | NO   | PRI | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

CODE stands for safeguard code (users code).

Safeguards per asset:

```
mysql> describe assetsafeguards;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code      | varchar(32)   | YES  |     | NULL    |      |
| asset      | varchar(32)   | YES  |     | NULL    |      |
| reference  | int(11)       | NO   | PRI | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

CODE stands for safeguard code (users code).

Security profiles:

```
mysql> describe profiles;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| profile    | varchar(32)   | YES  |     | NULL    |      |
| code       | varchar(32)   | YES  |     | NULL    |      |
| domain     | varchar(32)   | YES  |     | NULL    |      |
| reference  | int(11)       | NO   | PRI | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

Additional protections per domain:

```
mysql> describe domainkb;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kb         | varchar(32)   | YES  |     | NULL    |      |
| code       | varchar(32)   | YES  |     | NULL    |      |
| domain     | varchar(32)   | YES  |     | NULL    |      |
| reference  | int(11)       | NO   | PRI | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

Additional protections per asset:

```
mysql> describe assetkb;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kb         | varchar(32)   | YES  |     | NULL    |       |
| code       | varchar(32)   | YES  |     | NULL    |       |
| asset      | varchar(32)   | YES  |     | NULL    |       |
| reference  | int(11)       | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

4.10 Safeguard codes

Safeguards are presented with dynamically generated codes. Pilar uses an internal code for long-term recovery. The following table establishes the relation. Most external users may safely ignore it.

```
mysql> describe safeguards;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code     | varchar(32)   | YES  |     | NULL    |       |
| icode    | varchar(32)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

CODE stands for safeguard code (users code).

ICODE stands for safeguard code (internal code).

4.11 Value scale

These are the default values to map qualitative levels to / from quantitative values:

```
mysql> describe valuescale;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| level | int(11)   | YES  |     | NULL    |       |
| value | float     | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

5 Calculated values

5.1 Risk analysis: Impact and risk

5.1.1 Accumulated values

```
mysql> describe riskdown1 ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| asset      | varchar(32)   | YES  |     | NULL    |       |
| threat     | varchar(32)   | YES  |     | NULL    |       |
| dimension  | varchar(8)    | YES  |     | NULL    |       |
| level      | int(11)       | YES  |     | NULL    |       |
| value      | float         | YES  |     | NULL    |       |
| deg        | int(11)       | YES  |     | NULL    |       |
| freq       | float         | YES  |     | NULL    |       |
| reference  | int(11)       | YES  | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+

mysql> describe riskdown2 ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| reference  | int(11)       | YES  |     | NULL    |       |
| phase     | varchar(32)   | YES  |     | NULL    |       |
| impact    | float         | YES  |     | NULL    |       |
| risk      | float         | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

REFERENCE in riskdown1 is linked to REFERENCE in riskdown2.

5.1.2 Deflected values

```
mysql> describe riskup1 ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| above     | varchar(32)   | YES  |     | NULL    |       |
| below     | varchar(32)   | YES  |     | NULL    |       |
| threat     | varchar(32)   | YES  |     | NULL    |       |
| dimension  | varchar(8)    | YES  |     | NULL    |       |
| level      | int(11)       | YES  |     | NULL    |       |
| value      | float         | YES  |     | NULL    |       |
| deg        | int(11)       | YES  |     | NULL    |       |
| freq       | float         | YES  |     | NULL    |       |
| reference  | int(11)       | YES  | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+

mysql> describe riskup2 ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| reference  | int(11)       | YES  |     | NULL    |       |
| phase     | varchar(32)   | YES  |     | NULL    |       |
| impact    | float         | YES  |     | NULL    |       |
| risk      | float         | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

REFERENCE in riskup1 is linked to REFERENCE in riskup2.

5.2 Business continuity: Impact and risk

5.2.1 Accumulated values

```
mysql> describe bcmdown1 ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| asset      | varchar(32)   | YES  |     | NULL    |       |
| threat     | varchar(32)   | YES  |     | NULL    |       |
| reference  | int(11)       | YES  | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+

mysql> describe bcmdown2 ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| reference  | int(11)       | YES  |     | NULL    |       |
| phase      | varchar(32)   | YES  |     | NULL    |       |
| step       | int(11)       | YES  |     | NULL    |       |
| freq       | float         | YES  |     | NULL    |       |
| impact     | float         | YES  |     | NULL    |       |
| risk       | float         | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

REFERENCE in bcmdown1 is linked to REFERENCE in bcmdown2.

5.2.2 Deflected values

```
mysql> describe bcmup1 ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| above      | varchar(32)   | YES  |     | NULL    |       |
| below      | varchar(32)   | YES  |     | NULL    |       |
| threat     | varchar(32)   | YES  |     | NULL    |       |
| reference  | int(11)       | YES  | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+

mysql> describe bcmup2 ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| reference  | int(11)       | YES  |     | NULL    |       |
| phase      | varchar(32)   | YES  |     | NULL    |       |
| step       | int(11)       | YES  |     | NULL    |       |
| freq       | float         | YES  |     | NULL    |       |
| impact     | float         | YES  |     | NULL    |       |
| risk       | float         | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

REFERENCE in bcmup1 is linked to REFERENCE in bcmup2.

5.3 Security profiles

Two tables are generated for each security profile. The names of the tables are derived from the code of the profile.

```
mysql> describe 27002_2005_apps;
+-----+-----+-----+-----+-----+-----+
| Field  | Type          | Null  | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| control | varchar(32)   | YES   |      | NULL    |       |
| domain  | varchar(32)   | YES   |      | NULL    |       |
| applies | char(16)      | YES   |      | NULL    |       |
| comment | text          | YES   |      | NULL    |       |
+-----+-----+-----+-----+-----+-----+

mysql> describe 27002_2005_value;
+-----+-----+-----+-----+-----+-----+
| Field  | Type          | Null  | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| control | varchar(32)   | YES   |      | NULL    |       |
| domain  | varchar(32)   | YES   |      | NULL    |       |
| phase   | varchar(32)   | YES   |      | NULL    |       |
| value   | varchar(16)   | YES   |      | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

APPLIES encodes the applicability of the control:

- yes
- no
- mandatory

VALUE encodes the evaluation of the control

- a maturity level (L0 – L5) if the control has further detail (tree expansion into sub-controls or safeguards)
- n.a. if it does not apply
- empty string if unknown

6 Library

Library elements go to another database that is expected to be shared between different projects.

The full library may be exported for report generation. This part cannot be imported back into PILAR.

The extensions may be exported, and may be later imported on start-up.

6.1 Core elements

There are 5 tables:

```
mysql> show tables;
+-----+
| Tables_in_bgr |
+-----+
| classes        |
| criteria       |
| safeguards     |
| threats        |
| threatmodifiers |
+-----+
```

Each of them has the same structure:

```
mysql> describe classes;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code       | varchar(32)   | NO   | PRI | NULL    |       |
| father     | varchar(32)   | YES  |     | NULL    |       |
| position   | int(11)       | YES  |     | NULL    |       |
| name       | varchar(128)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

POSITION provides a relative index between the children of the same mother.

6.2 Extensions

There are 3 tables:

```
mysql> show tables;
+-----+
| Tables_in_bgr          |
+-----+
| extended_classes      |
| extended_criteria     |
| extended_threats     |
+-----+
```

Each of them has the same structure:

```
mysql> describe extended_classes;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code       | varchar(32)   | NO   | PRI | NULL    |       |
| father     | varchar(32)   | YES  |     | NULL    |       |
| position   | int(11)       | YES  |     | NULL    |       |
| name       | varchar(128)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

POSITION provides a relative index between the children of the same mother.