

## MANUAL OF ShERML

ShEx Rule Mapping Language (ShERML) is a tool developed to export data from relational databases to the Resource Description Framework (RDF), which is a graph data model that describes the semantics of web resources, constrained by a shapes schema. The exporting process ends obtaining a graph that satisfies the shapes schema and is ready to be queried by another application.

The beginning step consists of loading a database (sql file) and a shapes schema (json file). Then, the tool shows the relational schema of the database and the shapes schema as class diagrams. Then the mapping process is done by drawing arrows between anchors that are next to attributes of a table in the relational schema to properties of a type name of a shapes schema. Once the set of mappings is defined, the tool materializes a graph from the instance of the relational schema following the mappings. Finally, the graph is ready to be exported.

A ShEx or SHACL are languages for constraining the structure of a graph. Here in this tool, we capture those languages with the formalism of shapes schema. A shapes schema constrains a graph in the number of outgoing edges of a node if the node is typed. We allow the constraining of edges to be one (1), at most one (?), at least one (+) and many (\*).

For instance, a graph below in Turtle format:

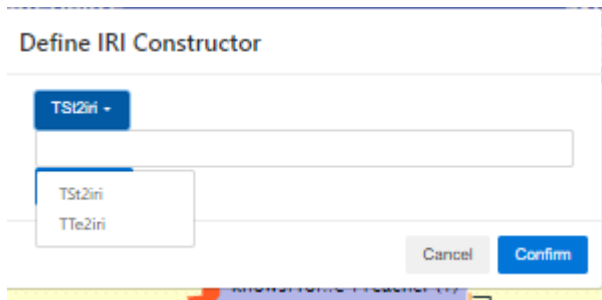
```
<https://inria.fr/TStudent/100>
  <http://example.com/course> "Math" ;
  <http://example.com/knownProf> "Pamela" ;
  <http://example.com/name> "Ana" .
```

If the node <https://inria.fr/TStudent/100> is typed with shape TStudent. The shape TStudent defines the following constraints:  
:course :: Literal (1);  
:knownProf :: TTeacher (+)

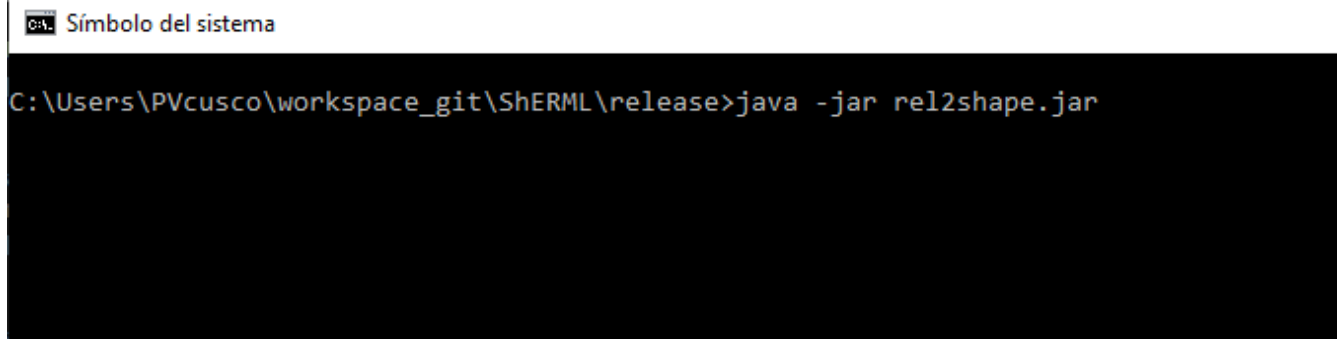
where : is the prefix for <http://example.com/>

This graph satisfies constraint with course but not with knownProf because the type is a Literal and not a Tteacher.

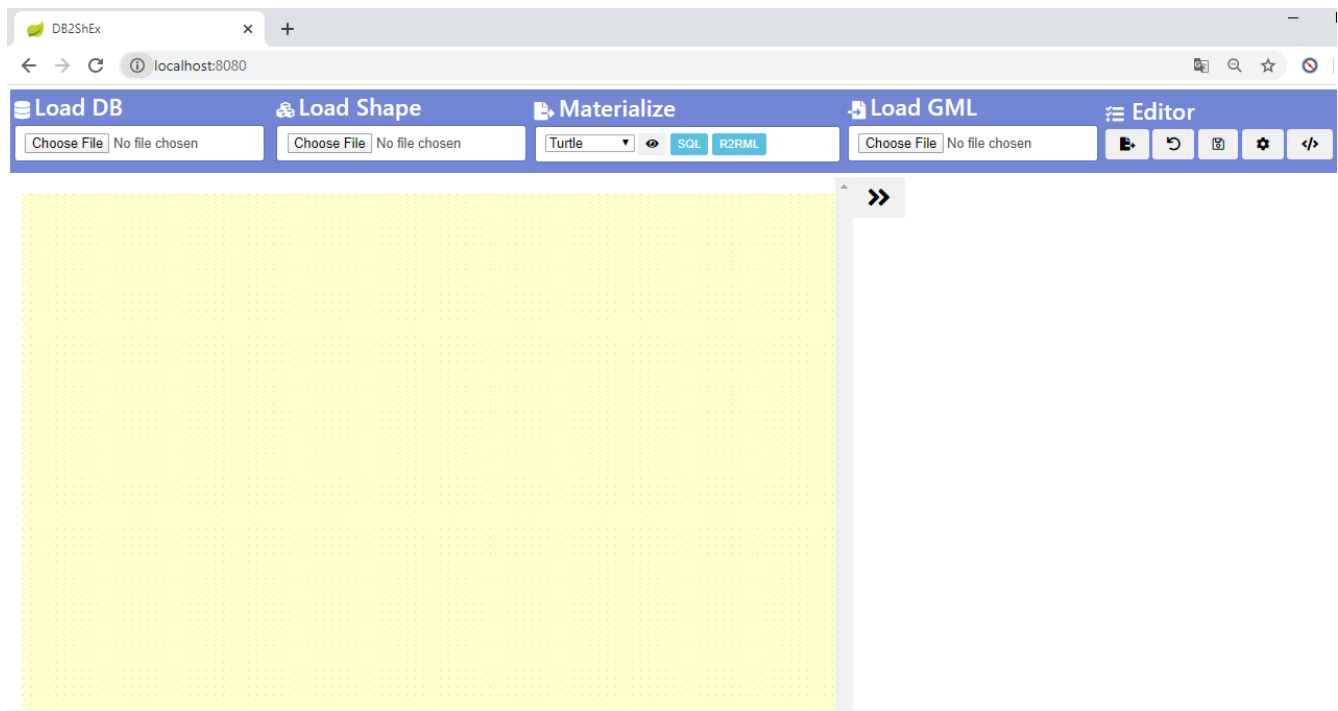
For the mapping process, the tool by default for each shape creates an IRI constructor that is unique to each shape. Therefore, when mapping to constraints that has target type a shape as knownProf :: TTeacher, user has to select the IRI constructor that corresponds to the target type in the window that the tool demands to specify it. For instance, in this message, user select the IRI constructor.



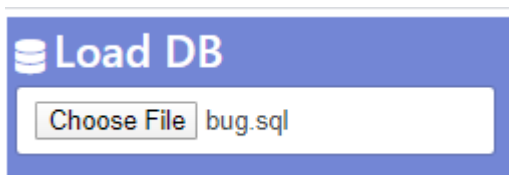
1. Launch application from terminal with command  
\$ java -jar rel2shape.jar



2. Open an explorer and go to localhost:8080



3. Open an sql file by clicking in Choosing file of Load DB



4. The system shows the diagram of the schema database. For instance of bug.sql shows the following picture.

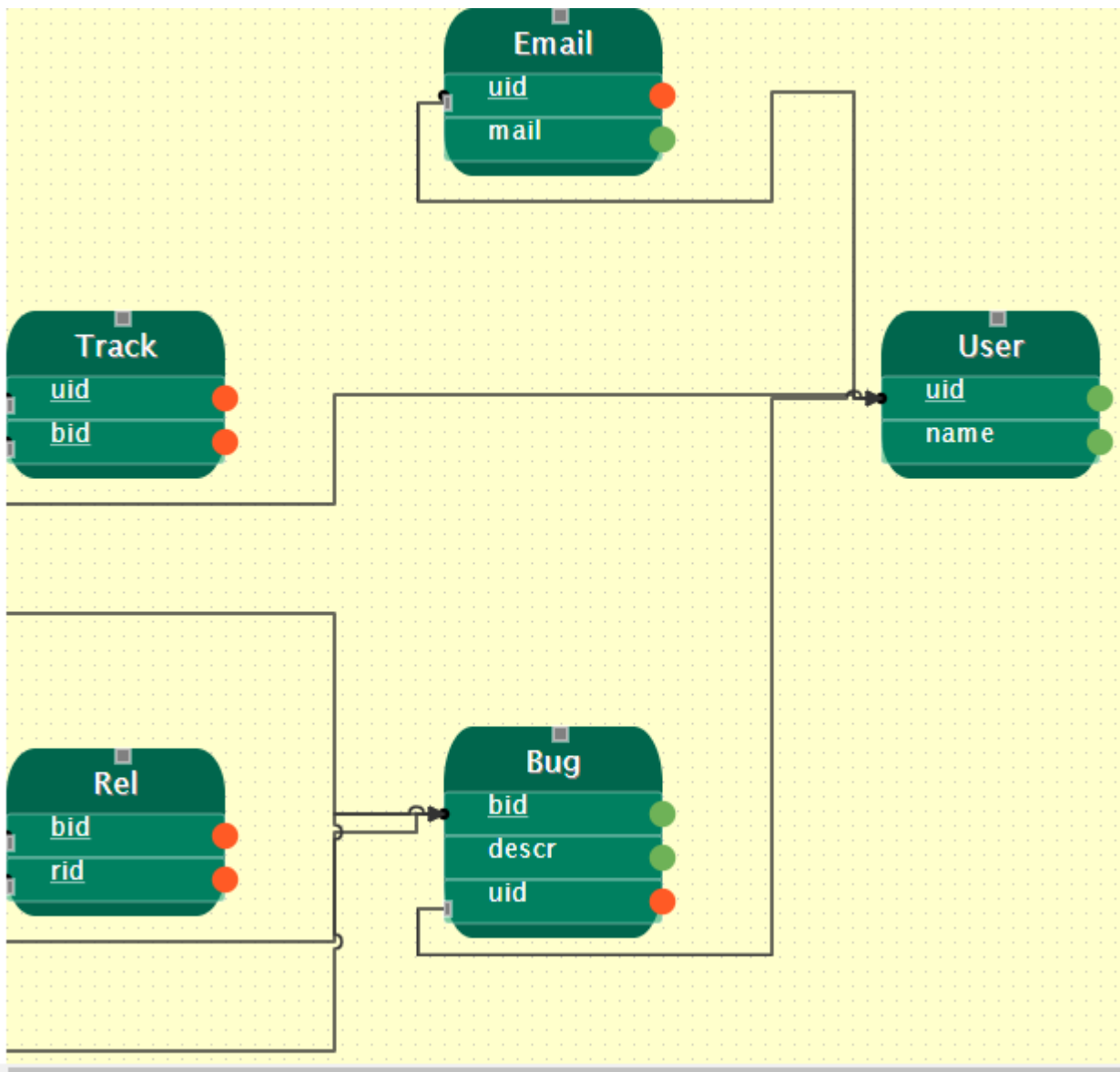


Figure1. Relational schema.

5. Then, open an ShEx or SHACL file by clicking in Choosing file of Load Shape



6. The system shows the diagram of the ShEx or SHACL file as follows.

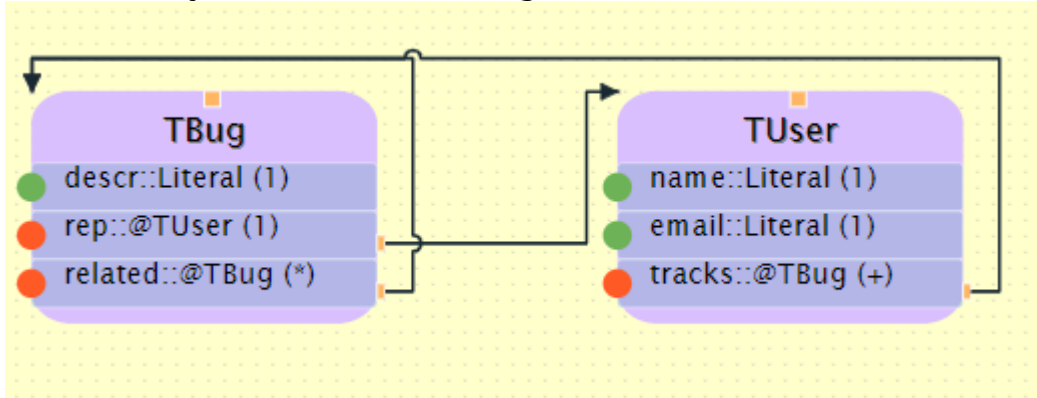


Figure 2. Shapes schema.

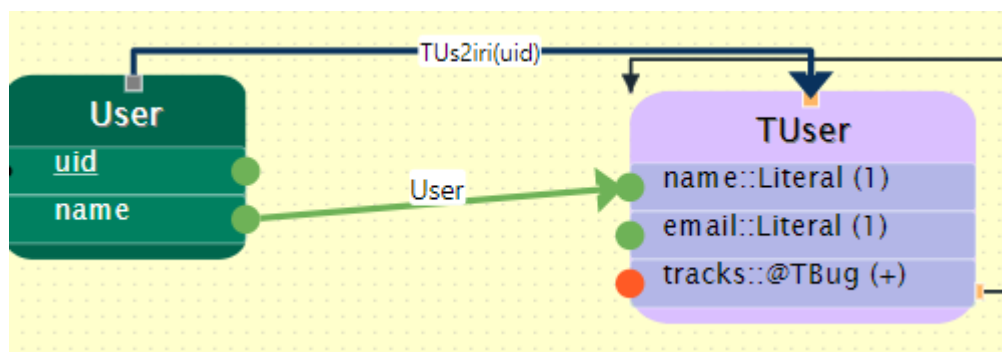
We observe that there are two kind of colors in the bullets next to the attributes. In the database (Figure 1), orange color for attributes that are foreign keys and green the rest. In the shape schema (Figure 2), orange color for properties that constraint the target node to be of another shape name and green for properties that constraint a literal.

All properties with (1 or +) must be linked because that indicates that the graph must contains an edge with a node whose subject is typed with Tuser and the type of the object is a Literal or Type that is the diagram.

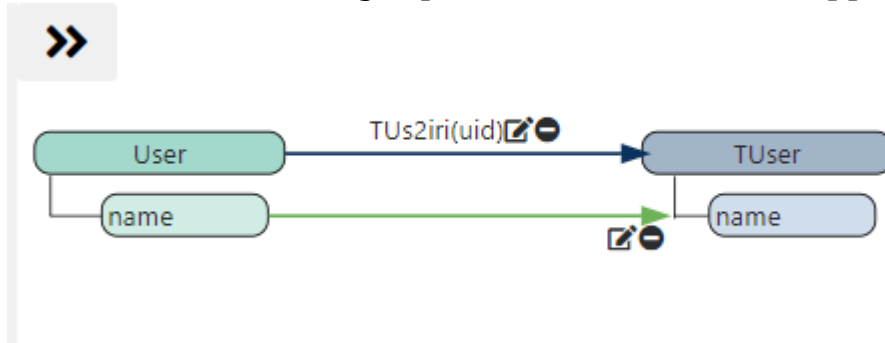
7. Start with the mapping process by dragging from bullets from the relational schema to bullets of the shapes schema.

For instance, we map the following rule:

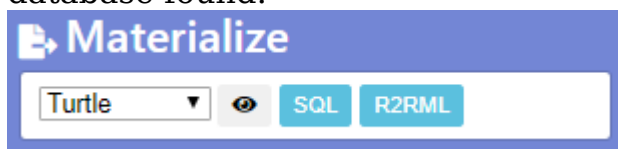
Map name of User to shape Tuser property name, the result will be.



The user drags from green bullet of name User to green buller of property Tuser. If the table is related with other tables, the system asks you to choose the table from where it is connected the main type. User can see in the right panel of ShERML the mappings done in the editor.

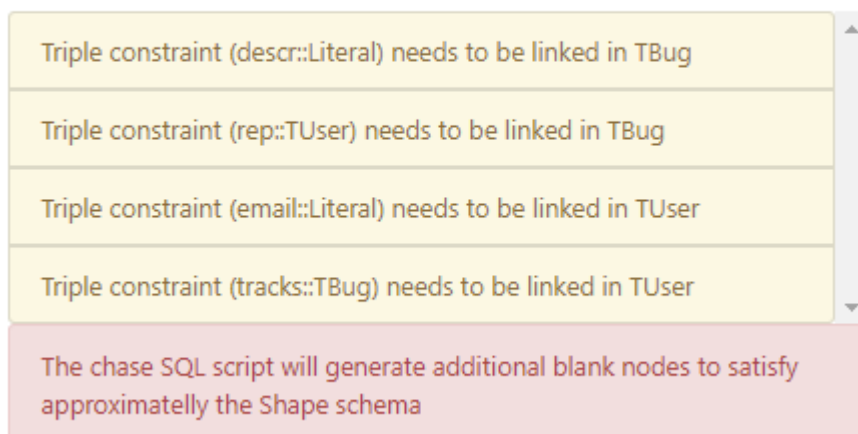


We can materialize our result by clicking the button of an eye. It is required to have at least a blue arrow and a green arrow. Otherwise, the tool shows No database found.



Since there are propertys that must be linked. Systems shows this message

### Warning Messages



Click on Continue and then we obtain the following RDF graph.

```
<http://example.com/TBug/@@@>
  <http://example.com/descr> "@@@" ;
  <http://example.com/rep> <http://example.com/TUser/@@@> .

<https://inria.fr/TUser/2>
  <http://example.com/email> "@@@" ;
  <http://example.com/name> "Edith" ;
  <http://example.com/tracks> <http://example.com/TBug/@@@> .

<https://inria.fr/TUser/1>
  <http://example.com/email> "@@@" ;
  <http://example.com/name> "Jose" ;
  <http://example.com/tracks> <http://example.com/TBug/@@@> .

<http://example.com/TUser/@@@>
  <http://example.com/email> "@@@" ;
  <http://example.com/name> "@@@" ;
  <http://example.com/tracks> <http://example.com/TBug/@@@> .
```

The symbols @@@ denote a node that has null value.