Interval App User Manual

Authors
Adam Chojan 109723
Piotr Przybyłowski 109728

Table of content

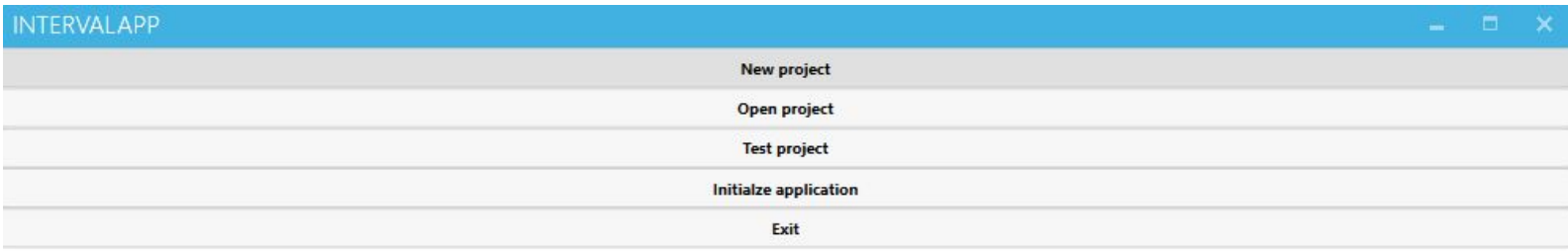
1. Startup app

1. Create file connection.txt with your connection string.
2. In the main menu click *Initialize project* button. This will create required tables, triggers and sequence in your database and test project with *TES* will be created.

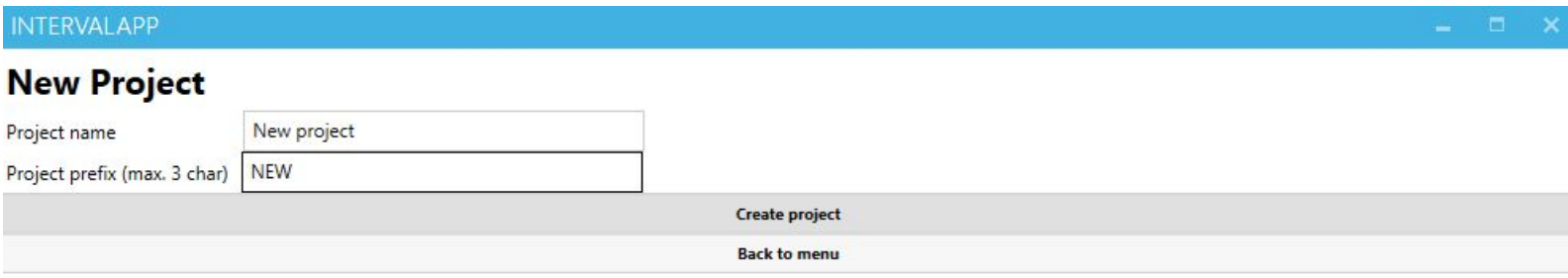
3. App is ready to use

2. Create new project

1. In the main menu choose *New project* button.



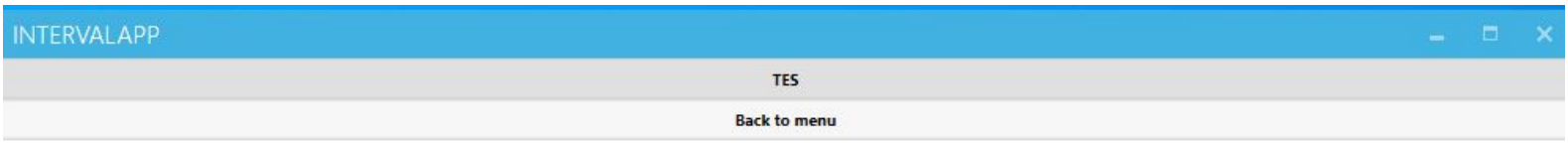
2. Fill with unique values name and prefix fields.



3. Open existing project

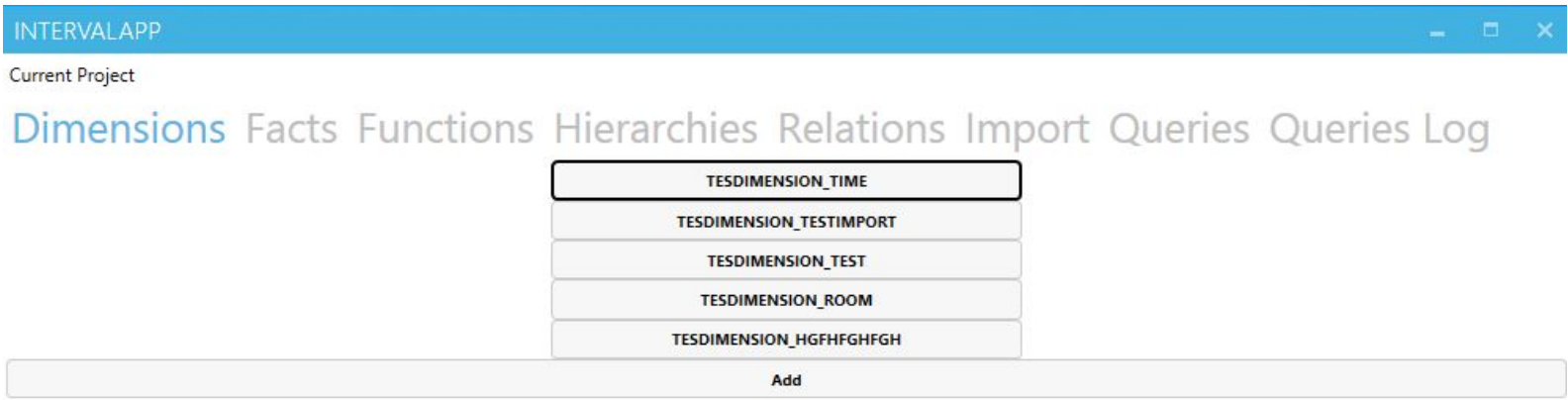
1. In the main menu choose *Open project* button.

2. Click button with selected project.



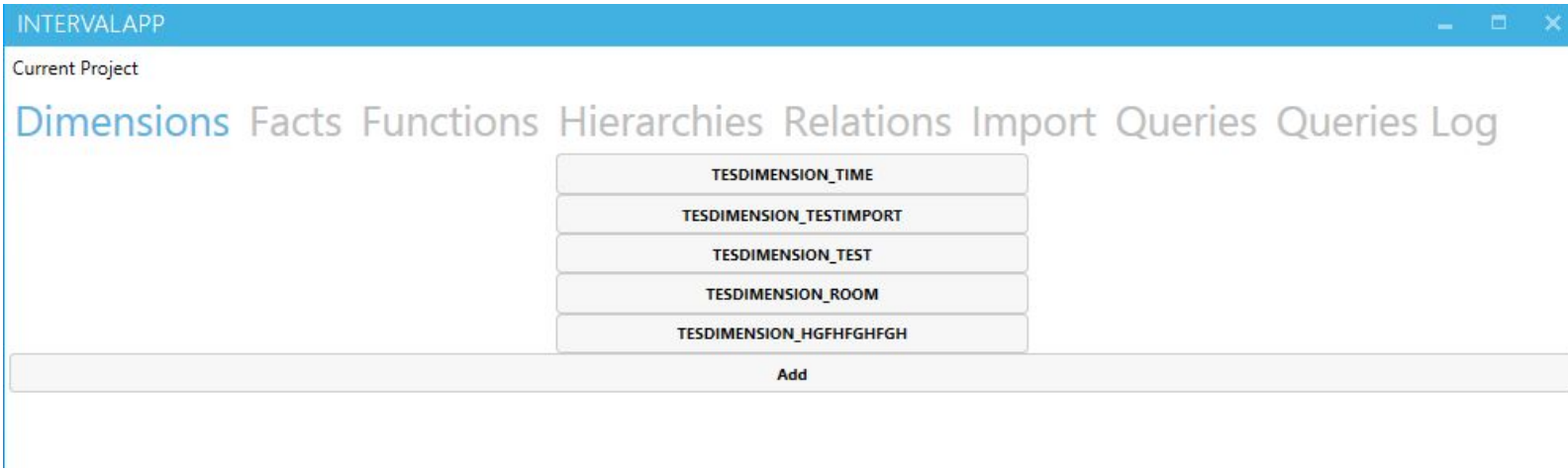
4. Main view - project page

- Contains tabs with all features needed to create your own data warehouse and to create interval functions. App contains
  - creating dimension, facts, relations, hierarchies,
  - creating function table with calculated slopes and intercepts,
  - importing data to tables,
  - executing queries,
  - saving queries, executing time and comments,
  - creating bar plots with selected queries logs

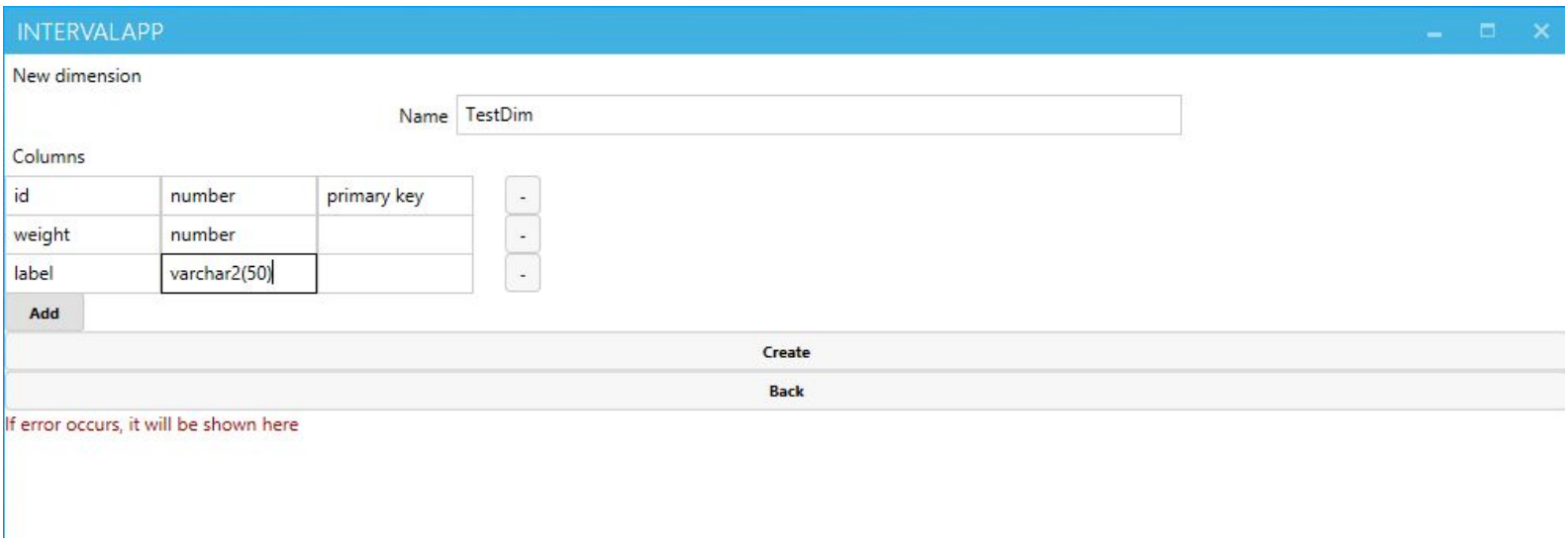


5. Create dimension

1. Go to *Dimensions* tab
2. Click *Add* button



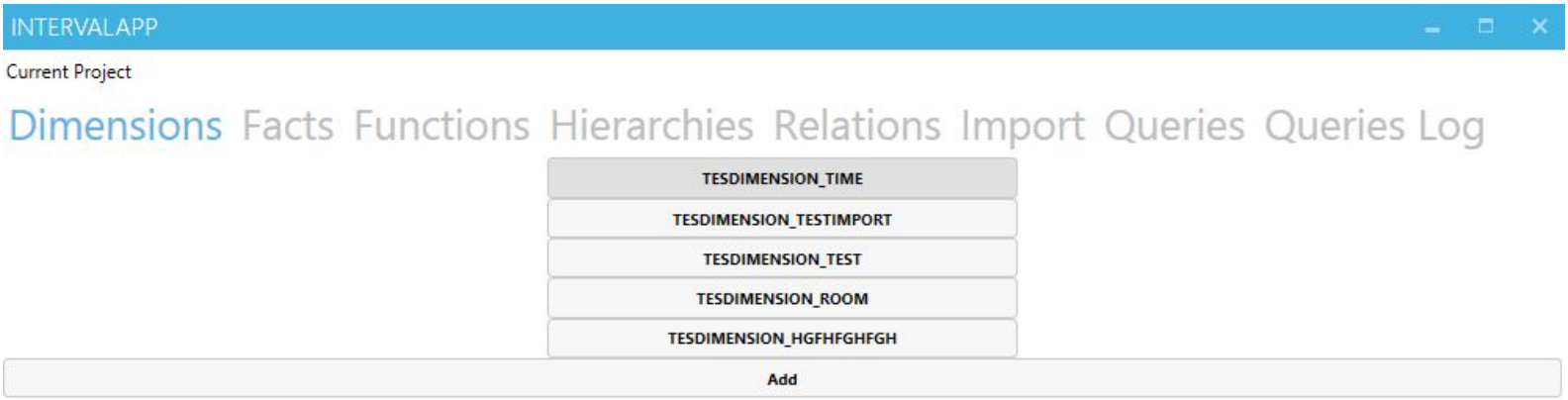
3. Put table name then columns name, columns type and column constraint.



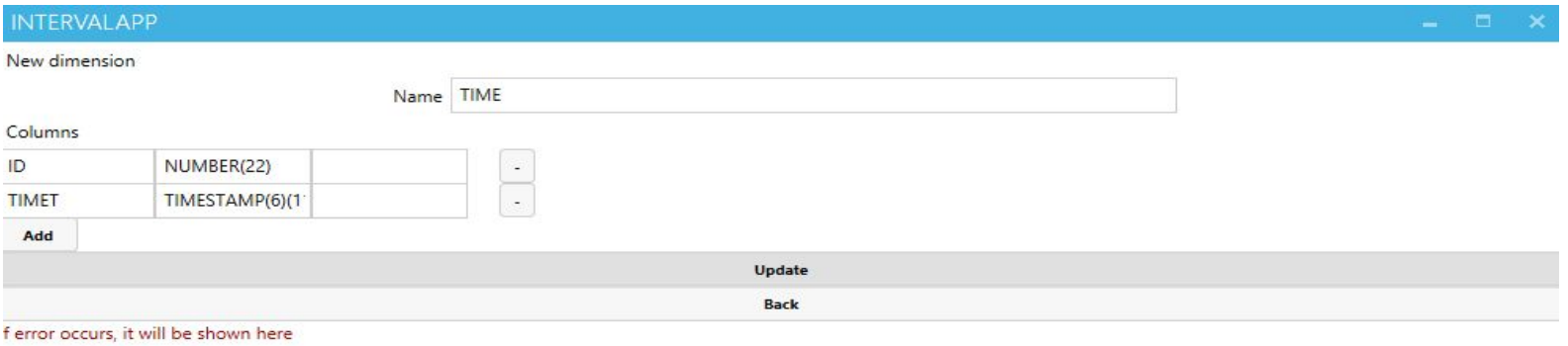
You can add new column with *Add* button or delete extra columns with **-** button

6. Update dimension

1. Go to *Dimensions* tab
2. Click dimension table button

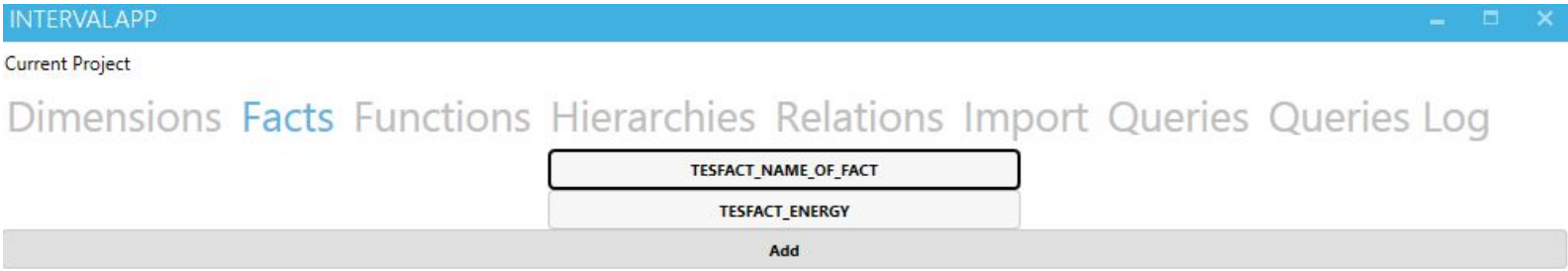


3. Change table name, columns name, columns type and column constraint.
4. Click *Update* button.
   You can add new column with *Add* button or delete extra columns with - button
   Important thing is that table will be dropped before creating updated version.

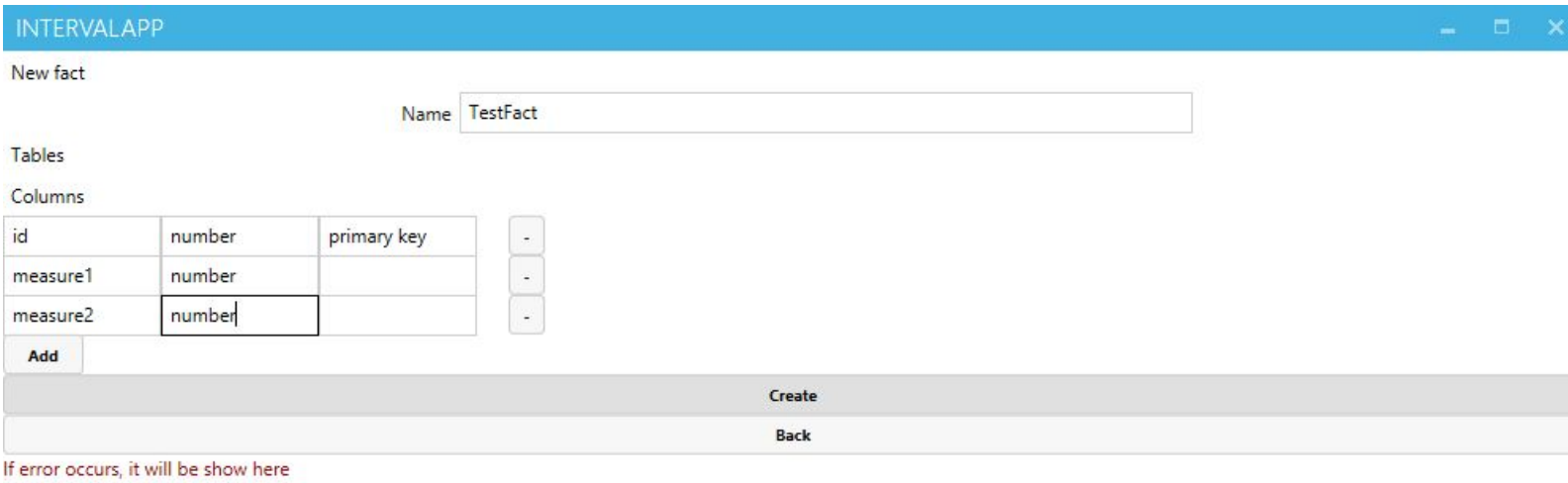


7. Create fact

1. Go to *Facts* tab.
2. Click *Add* button.

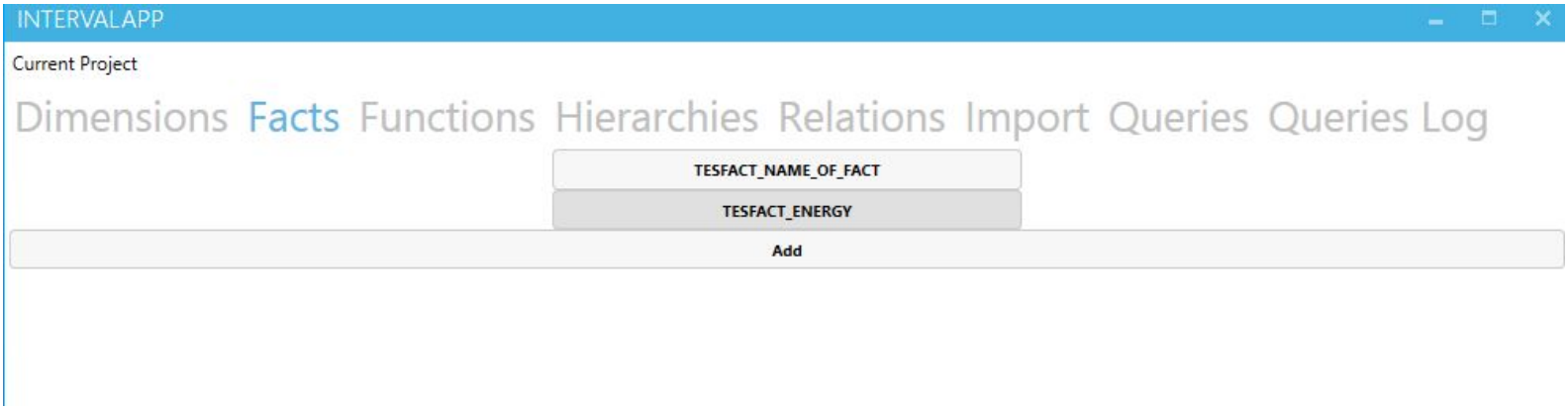


3. Fill table name, columns name, columns type and column constraint fields. You can add new column with *Add* button or delete extra columns with - button

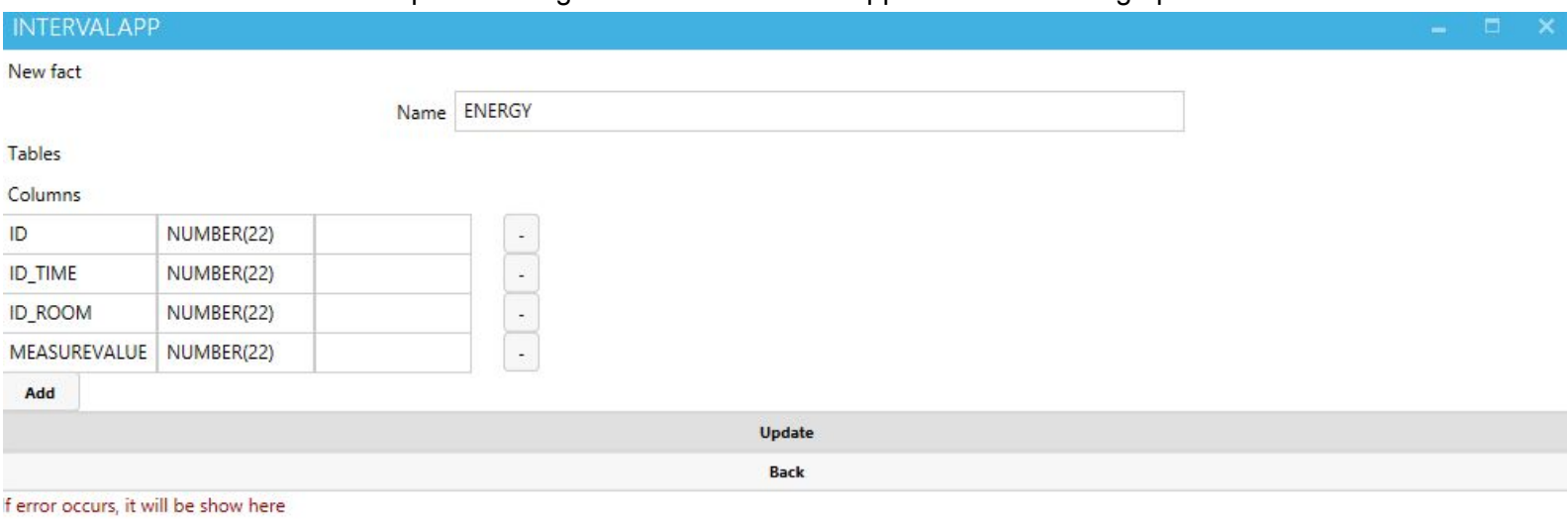


8. Update fact

1. Go to *Facts* tab.
2. Click fact table button.

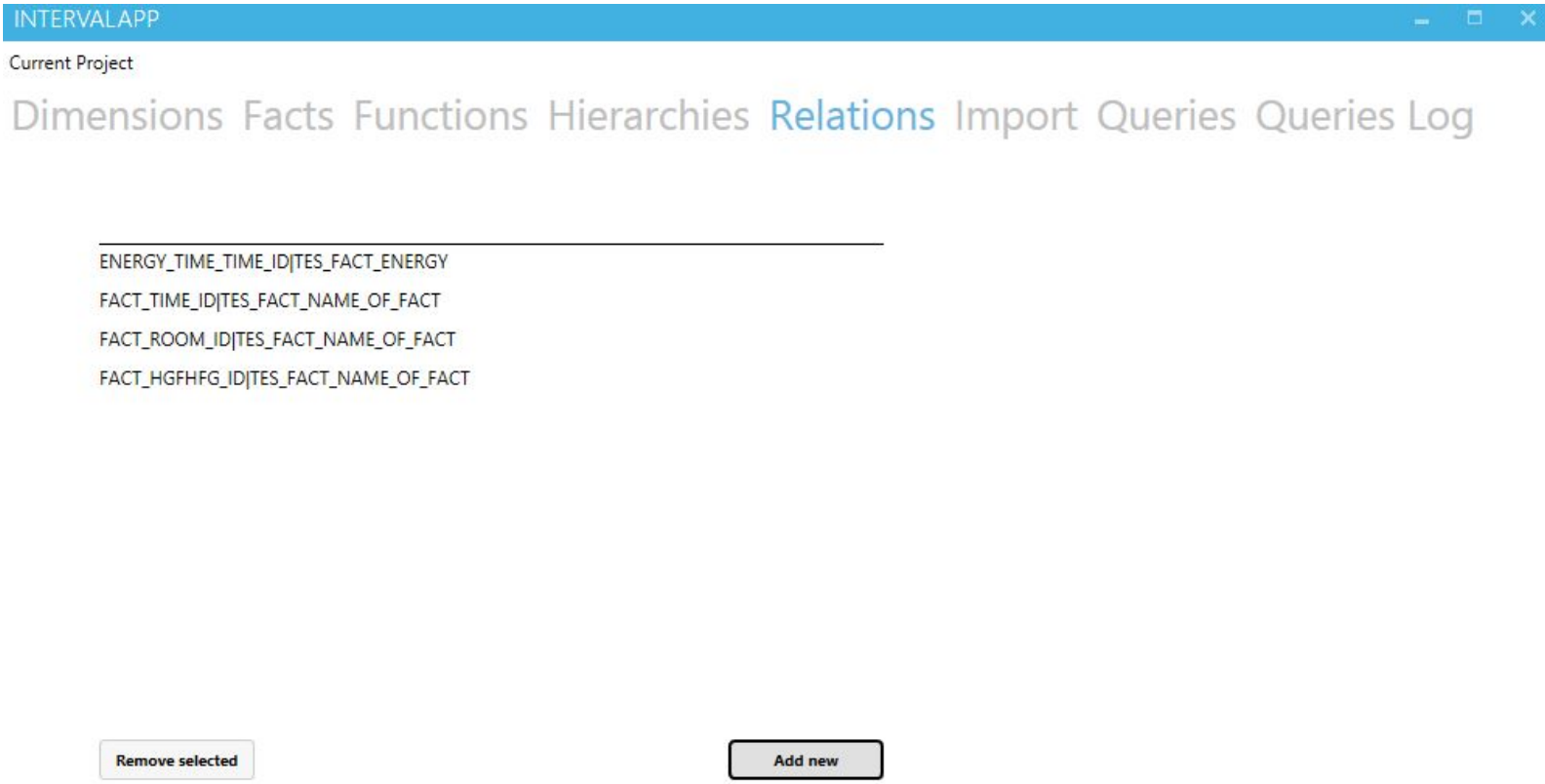


3. Change table name, columns name, columns type or column constraint.
4. Click *Update* button.
   You can add new column with *Add* button or delete extra columns with - button.
   Important thing is that table will be dropped before creating updated version.

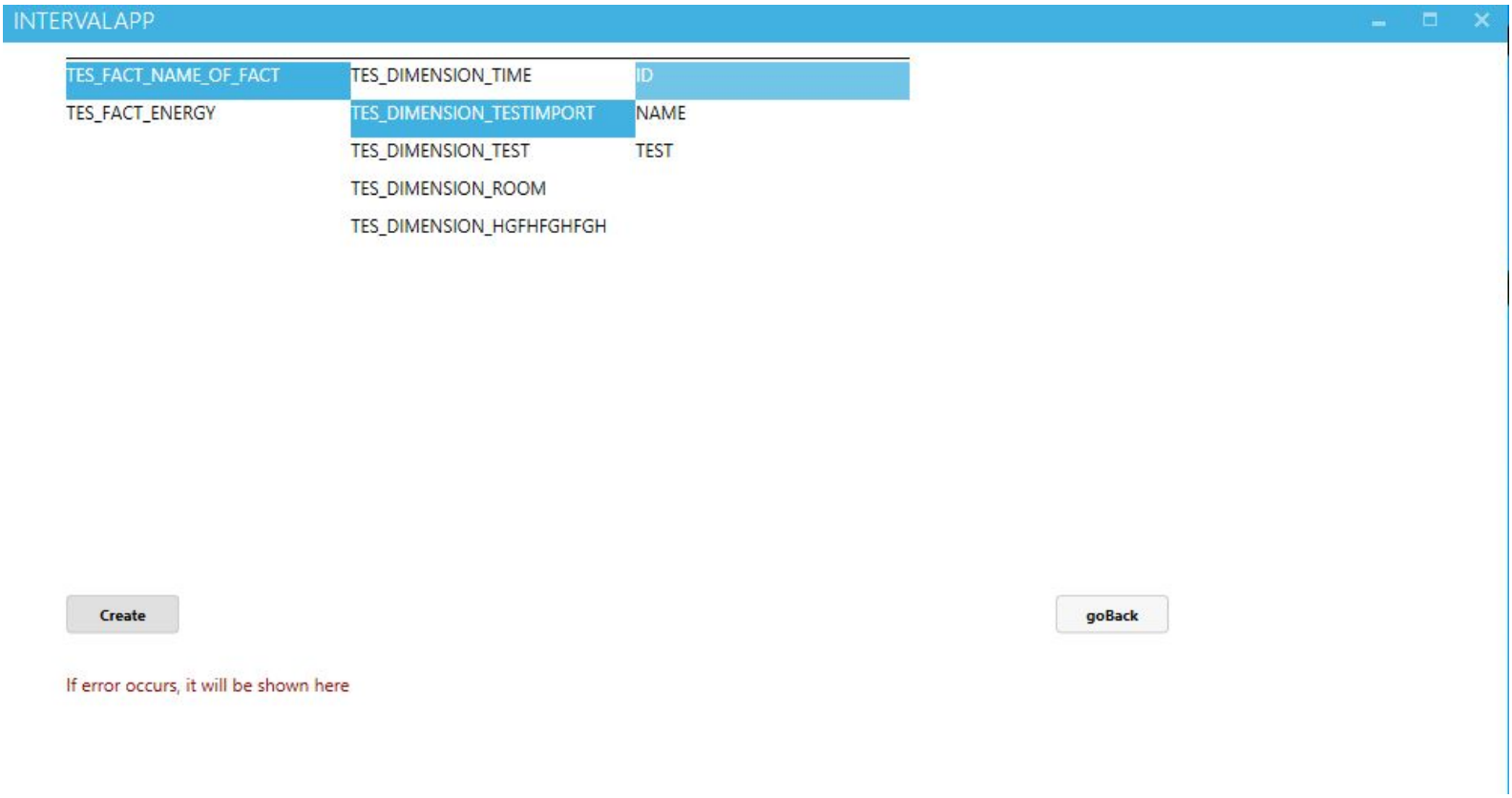


9. Create relation

1. Go to *Relations* tab.
2. Click *Add new* button.

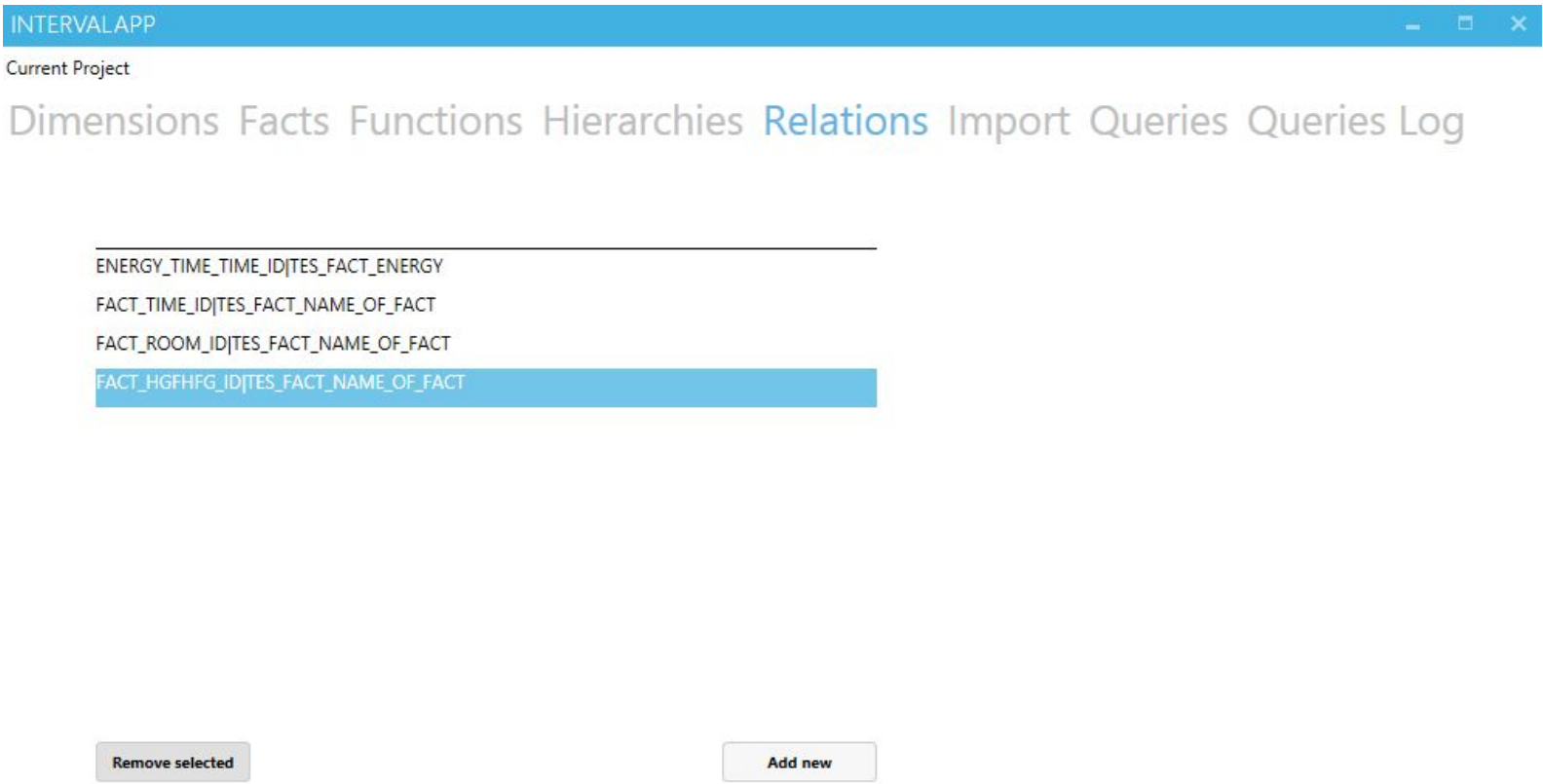


3. In the first column select FACT in which you want to create foreign key. In example its TES_FACT_NAME_OF_FACT.
4. In the second column select DIMENSION table which contains column you want you want to reference in the created foreign key. In example it's TES_DIMENSION_TESTIMPORT
5. Finally in the third column select column you want to reference - it should be column with either UNIQUE or PRIMARY KEY constraint otherwise there will be an error.
6. Click *Create* button.


INTERVALAPP
TES_FACT_NAME_OF_FACT
TES_FACT_ENERGY
TES_DIMENSION_TIME
TES_DIMENSION_TESTIMPORT
TES_DIMENSION_TEST
TES_DIMENSION_ROOM
TES_DIMENSION_HGFHFGHFGH
ID
NAME
TEST
Create
goBack
If error occurs, it will be shown here

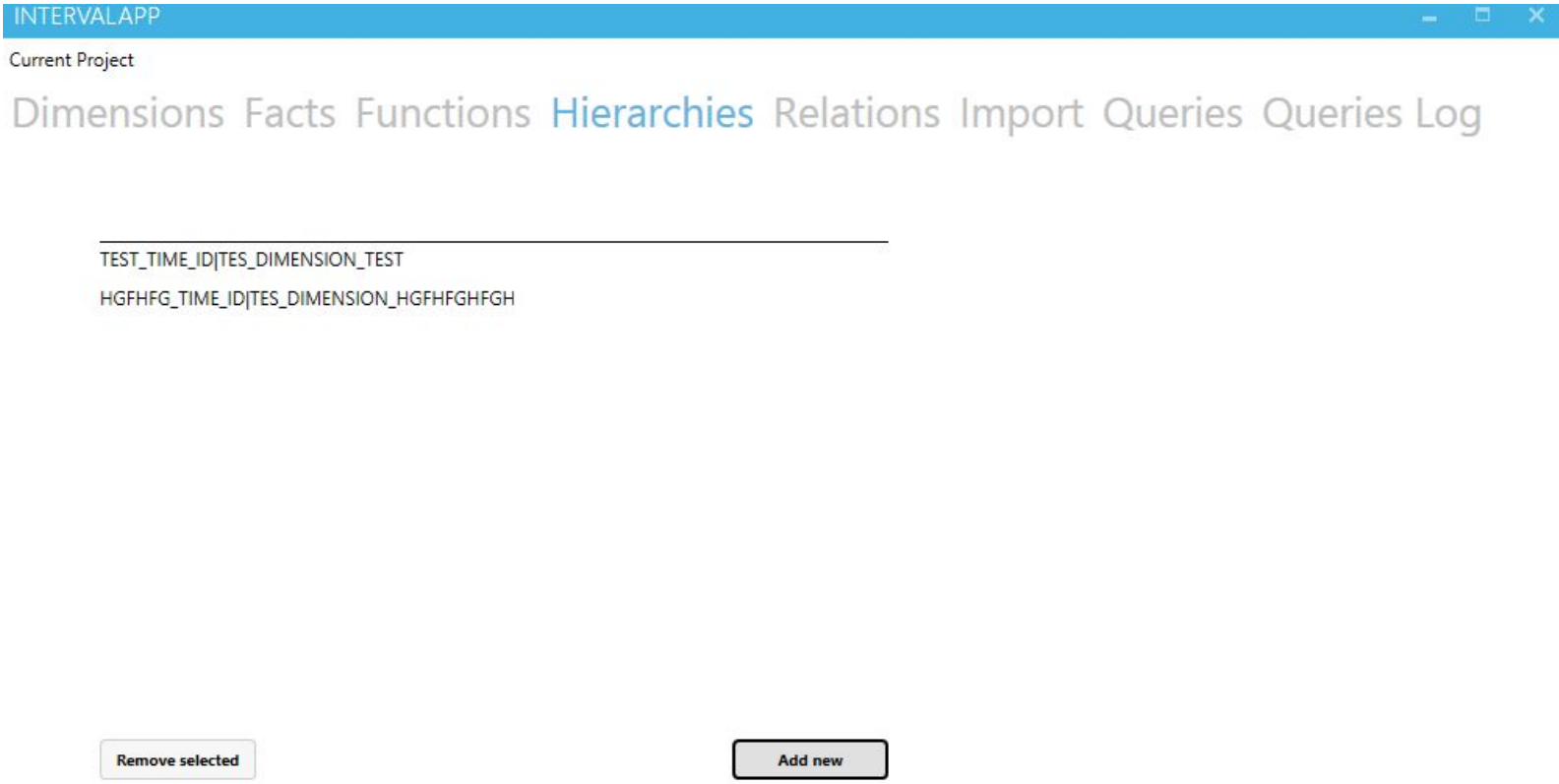
10. Delete relation

1. Go to *Relations* tab
2. Select relation.
3. Click *Remove selected* button.

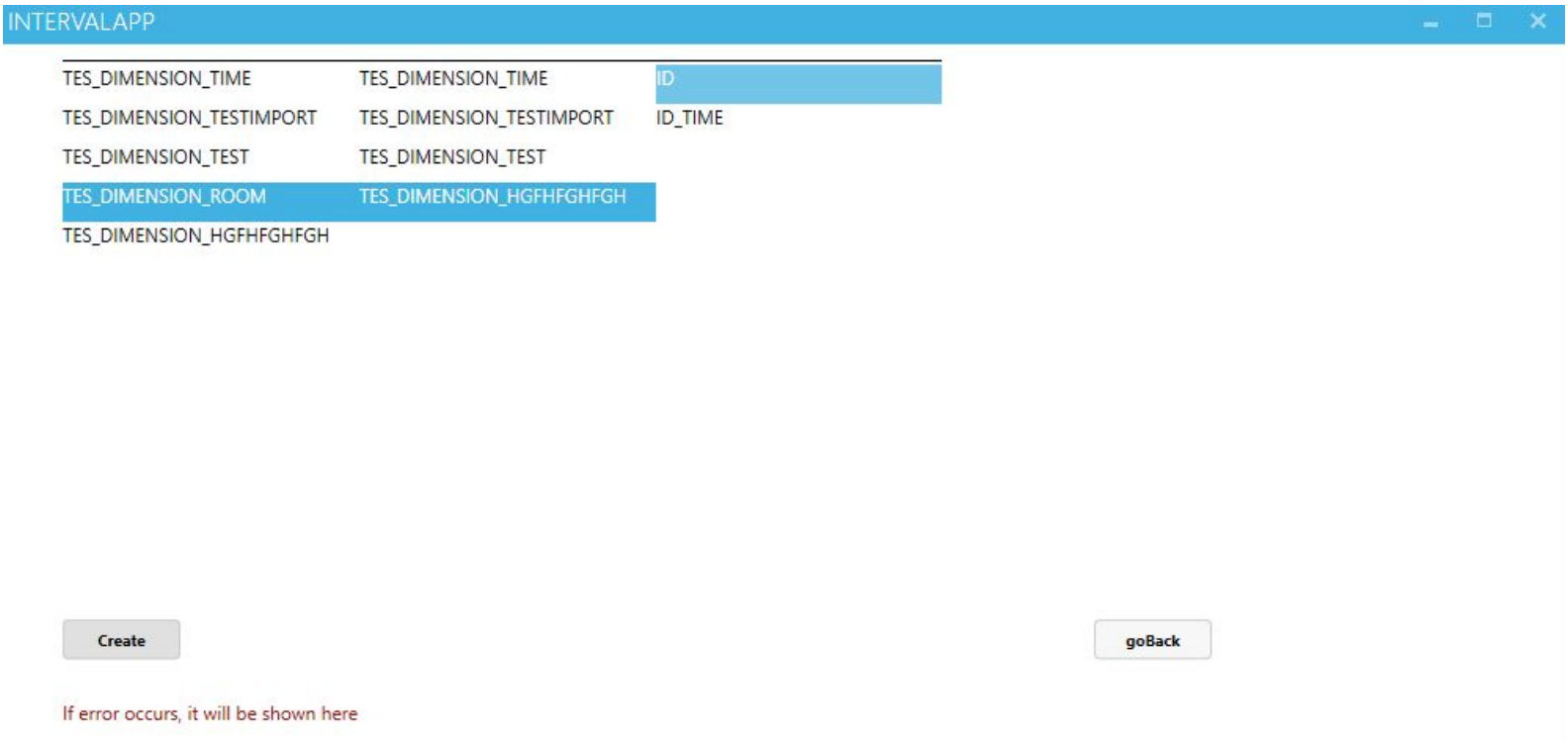


11. Create hierarchy

1. Go to *Hierarchies* tab.
2. Click *Add new* button.

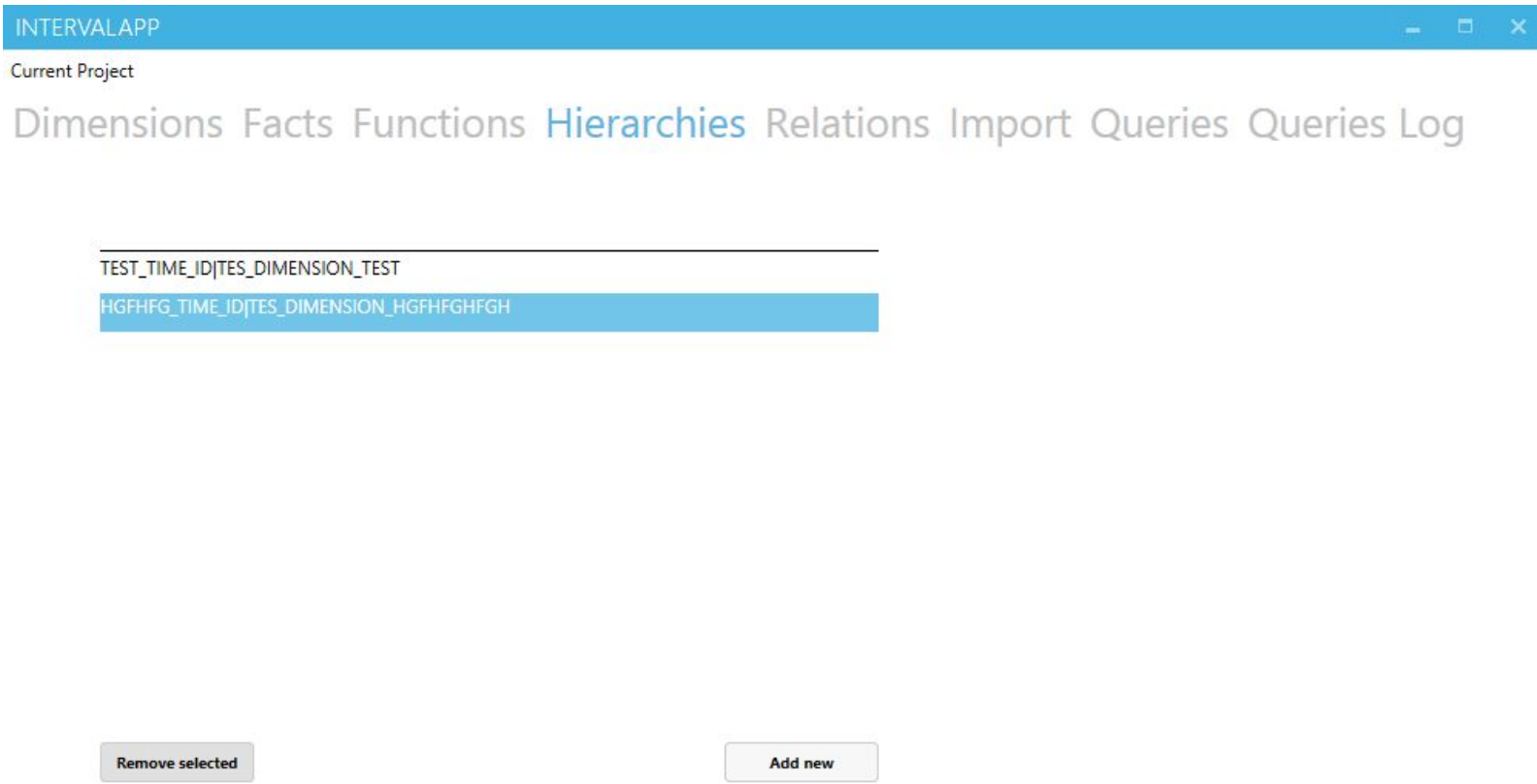


3. In the first column select DIMENSION in which you want to create foreign key. In example its TES_DIMENSION_ROOM.
4. In the second column select DIMENSION table which contains column you want you want to reference in the created foreign key. In example it's TES_DIMENSION_HFHFHGHFH
5. Finally in the third column select column you want to reference - it should be column with either UNIQUE or PRIMARY KEY constraint otherwise there will be an error.
6. Click *Create* button.


INTERVALAPP
TES_DIMENSION_TIME
TES_DIMENSION_TESTIMPORT
TES_DIMENSION_TEST
TES_DIMENSION_ROOM
TES_DIMENSION_HGFHFGHFGH
TES_DIMENSION_TIME
TES_DIMENSION_TESTIMPORT
TES_DIMENSION_TEST
TES_DIMENSION_HGFHFGHFGH
ID
ID_TIME
Create
goBack
If error occurs, it will be shown here

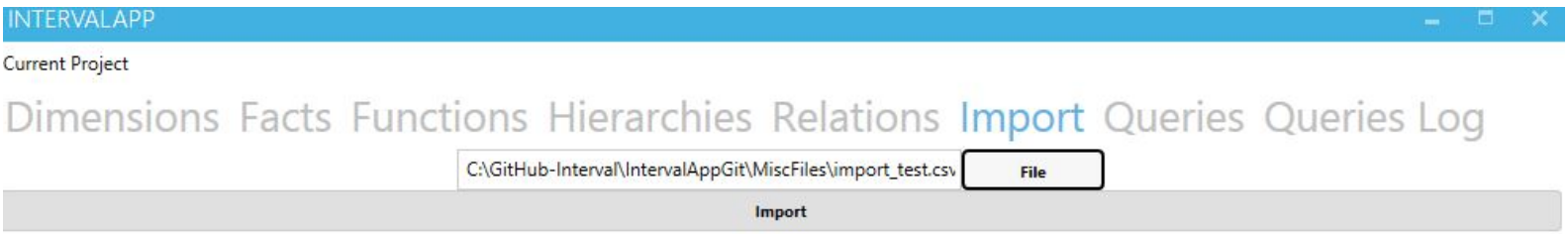
12. Delete hierarchy

1. Go to *Hierarchies* tab
2. Select hierarchy.
3. Click *Remove selected* button.



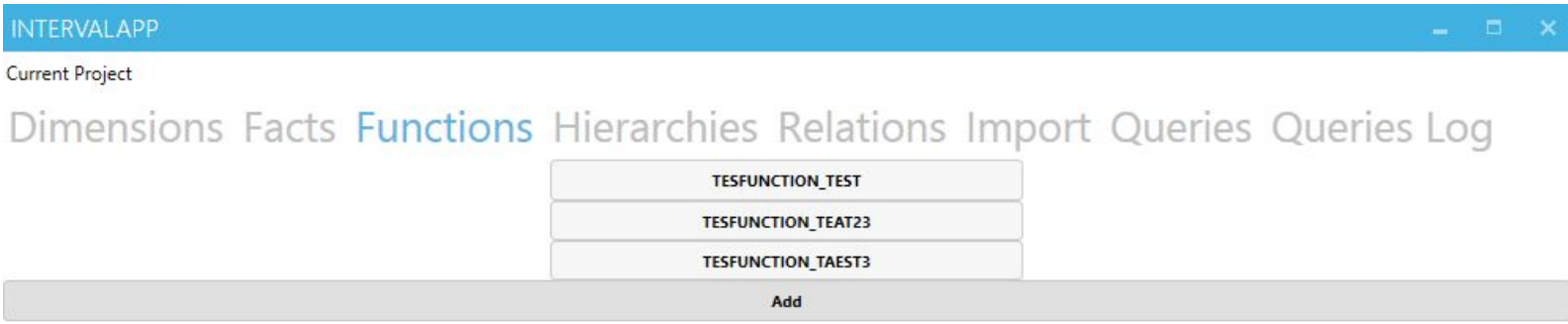
13. Import data

1. Go to *Import* tab.
2. Fill path field or use *Windows* file dialog to choose *csv* file.
3. Click *Import* button.

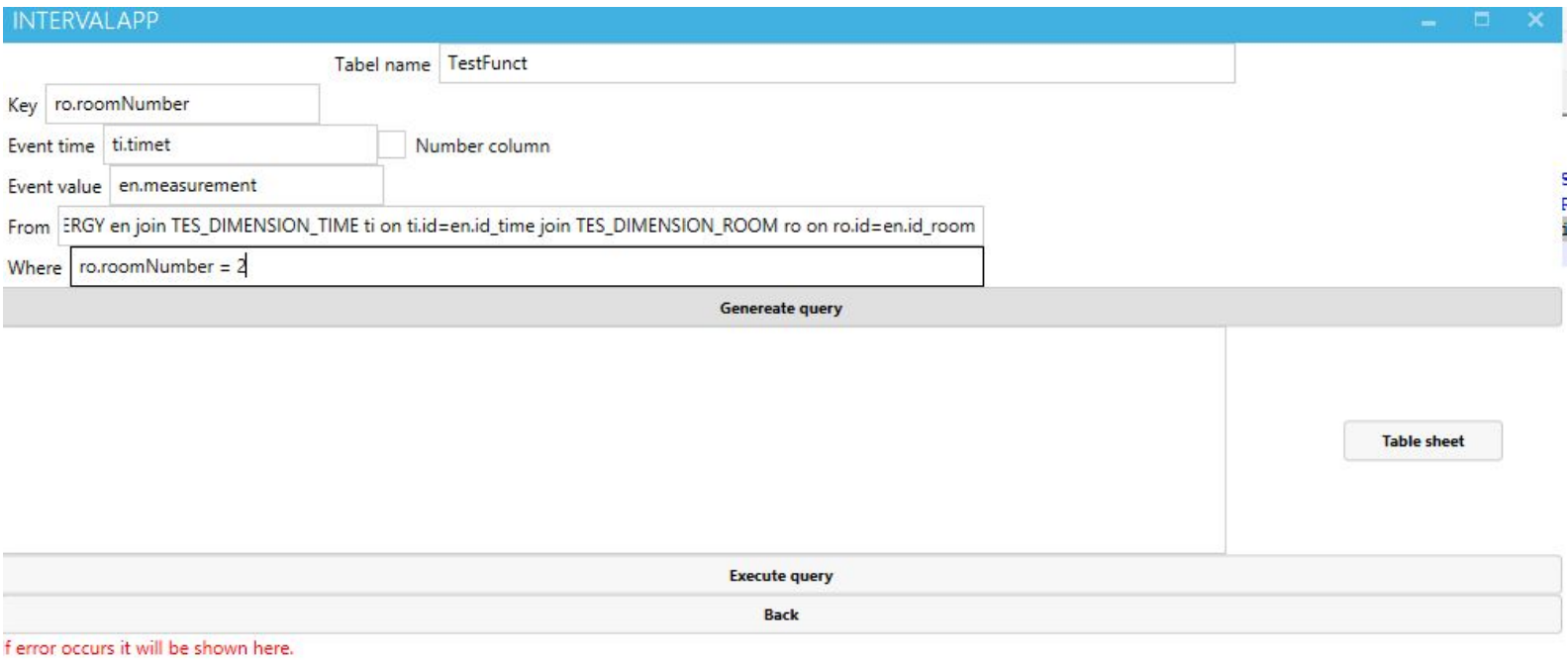


14. Create function

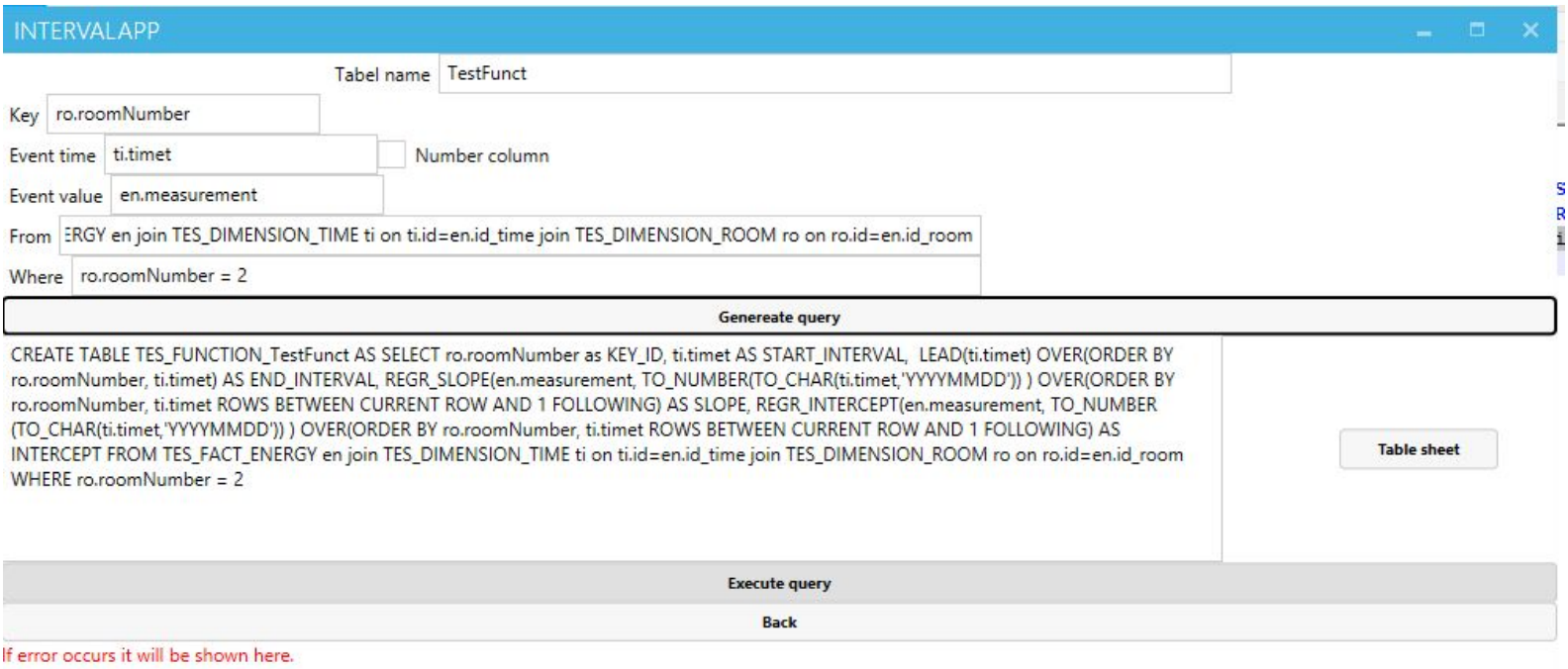
1. Go to *Functions* tab
2. Click *Add* button



3. Fill all required fields with columns names with prefixes if needed(*Table name, Key* - information to identify row, *Event time* - *Datetime* or *Number*, *Event value, From*).
4. Fill required field - *From* with tables names and joins.
5. Fill optional field *Where* if needed.
6. Click *Generate query* button.

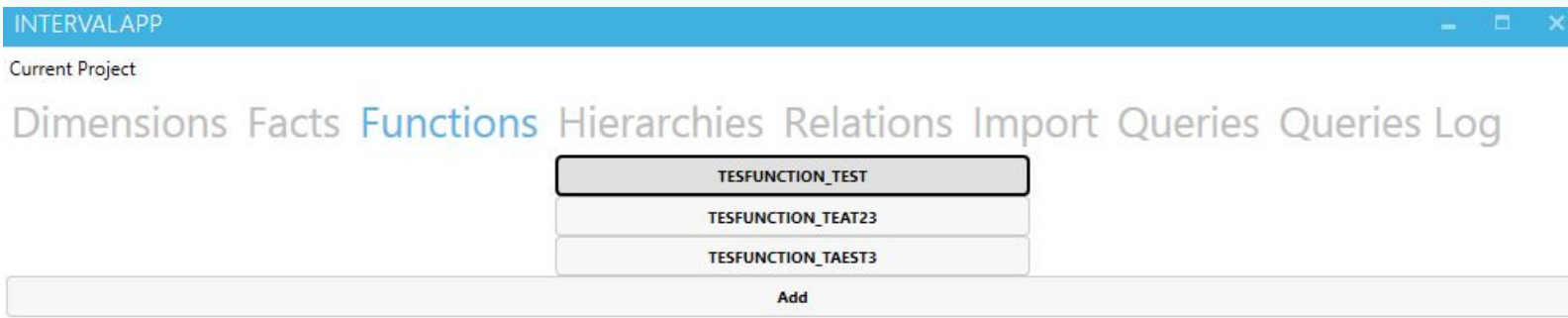


7. App will generate query to create function table. User can modify it if needed.
8. Click *Execute query* if everything is fine
   User can use *Table sheet* button to open new window which contains all projects tables with columns name



15. Delete function

1. Go to *Functions* tab
2. Select function table button



3. Click *Delete function* button.

INTERVALAPP

Table name TEAT23

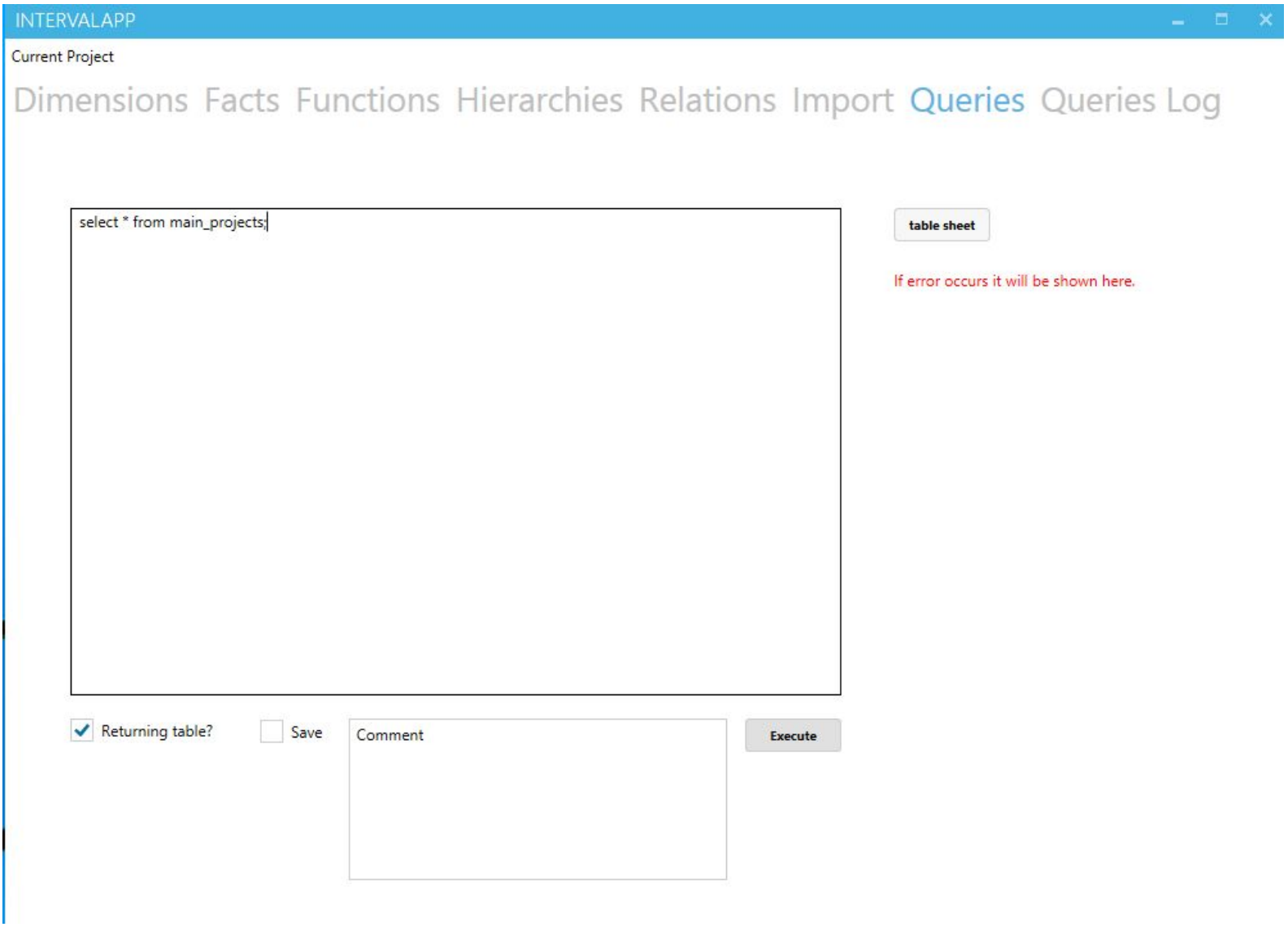
Delete function

KEYID	STARTINTERVAL	ENDINTERVAL	SLOPE	INTERCEPT
1	1/1/2012 10:00:00 AM	1/2/2012 10:00:00 AM	1	-20120100
1	1/2/2012 10:00:00 AM	1/3/2012 10:00:00 AM	3	-60360304
1	1/3/2012 10:00:00 AM	1/4/2012 10:00:00 AM	4	-80480407
1	1/4/2012 10:00:00 AM	1/5/2012 10:00:00 AM	9	-181080927
1	1/5/2012 10:00:00 AM	1/6/2012 10:00:00 AM	-8	160960858
1	1/6/2012 10:00:00 AM	1/7/2012 10:00:00 AM	12	-241441262
1	1/7/2012 10:00:00 AM	1/8/2012 10:00:00 AM	12	-241441262
1	1/8/2012 10:00:00 AM	1/9/2012 10:00:00 AM	1	-20120074

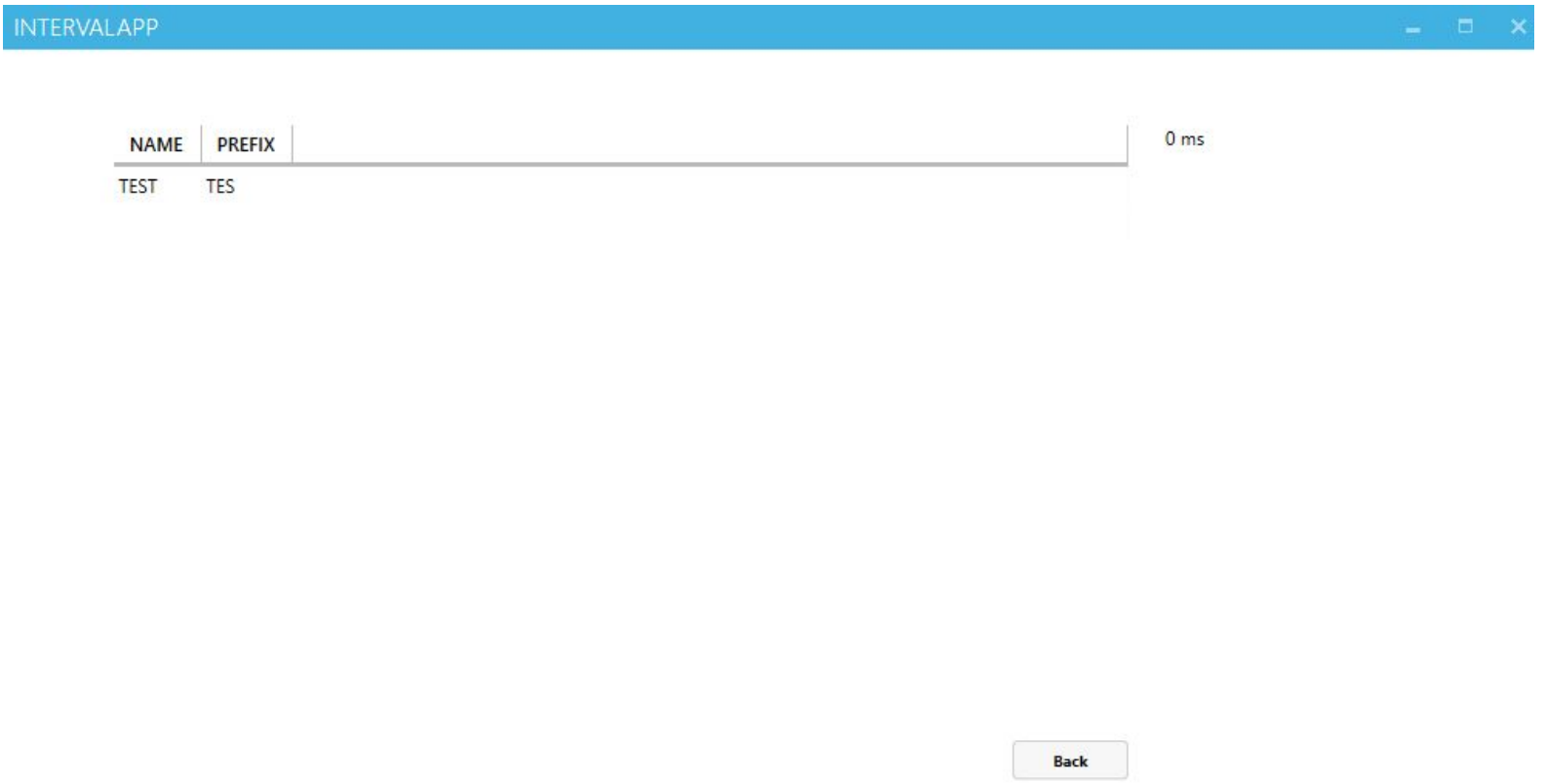
Back

16. Execute query

1. Go to *Queries* tab.
2. Write your query into main field.
3. Choose if your query should return table (checkbox *Returning table?* selected) or no.
4. Click *Execute* button.

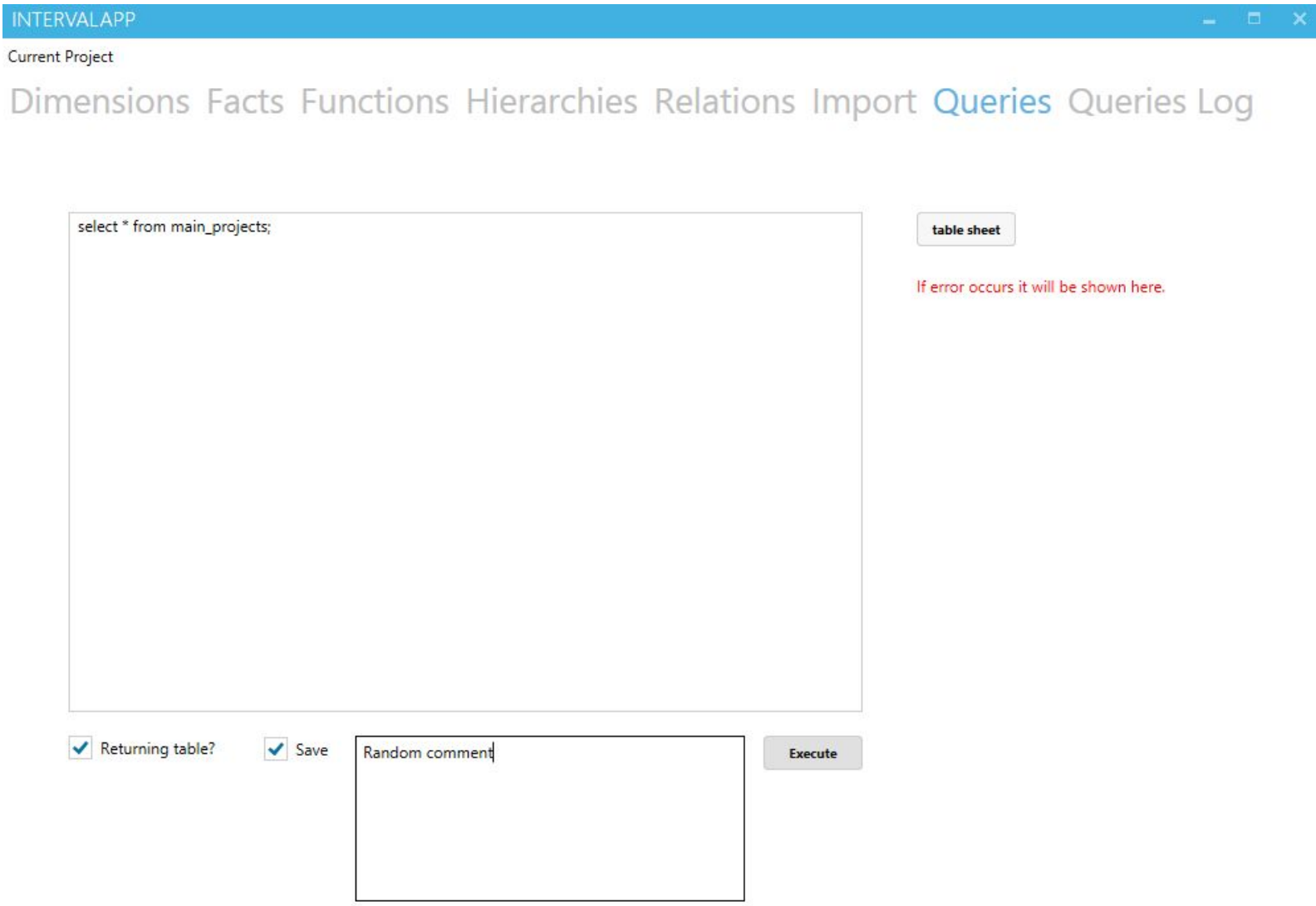


5. If in 16.3 checkbox was select, app will open new window with result



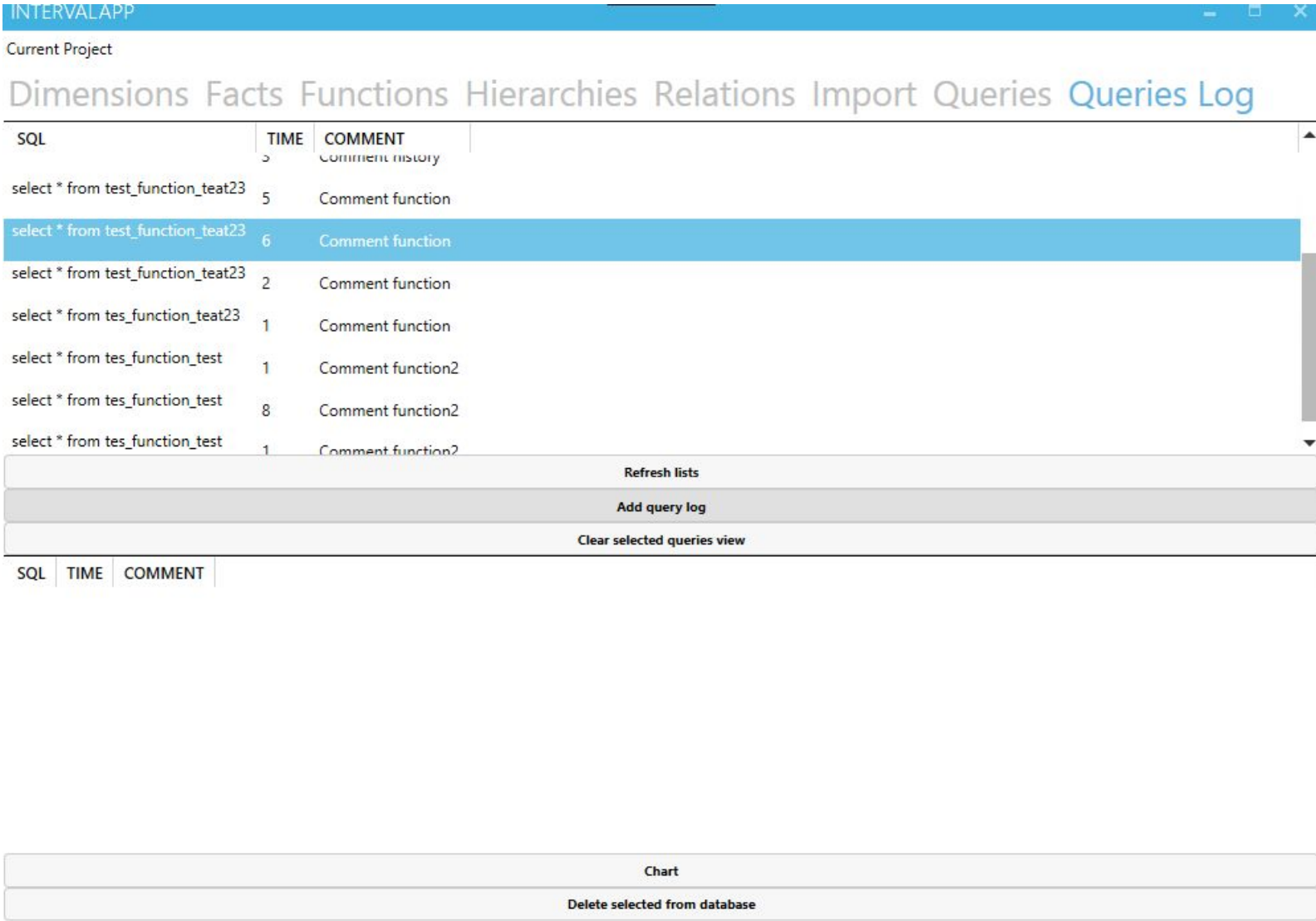
17. Save query

1. Go to *Queries* tab.
2. Write your query into main field.
3. Choose if your query return table (checkbox *Returning table?* selected) or no.
4. Click *Save* checkbox.
5. Fill *Comment* textbox if needed.
6. Click *Execute* button.
7. If in 16.3 checkbox was select, app will open new window with result
8. Query will be save.


INTERVALAPP
Current Project
Dimensions Facts Functions Hierarchies Relations Import Queries Queries Log
select * from main_projects;
table sheet
If error occurs it will be shown here.
Returning table?
Save
Random comment
Execute

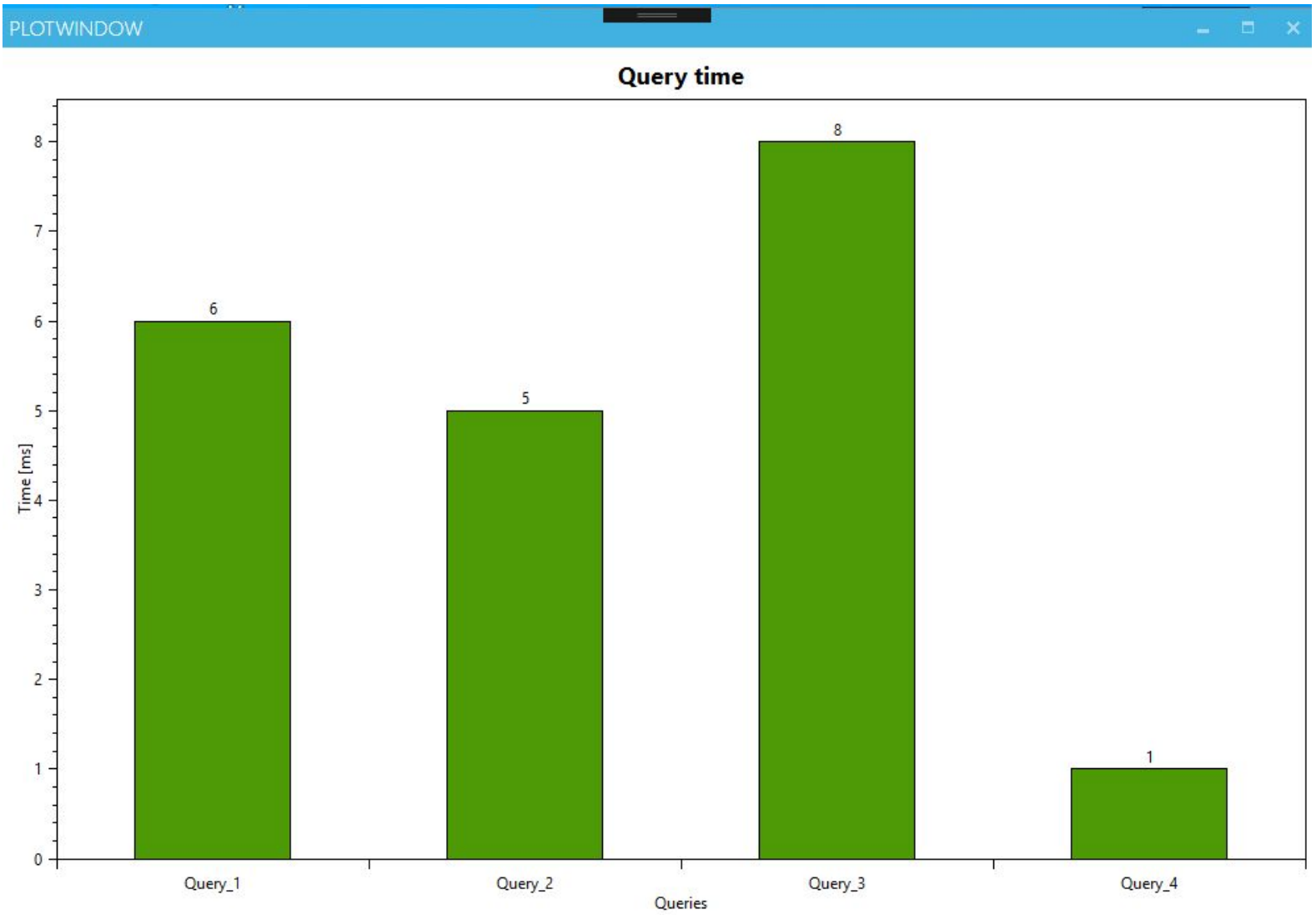
18. Select saved queries and create plot

1. Go to *Queries Log.*
2. Select query log form first list view.
3. Click *Add query log* button to add it to plot.



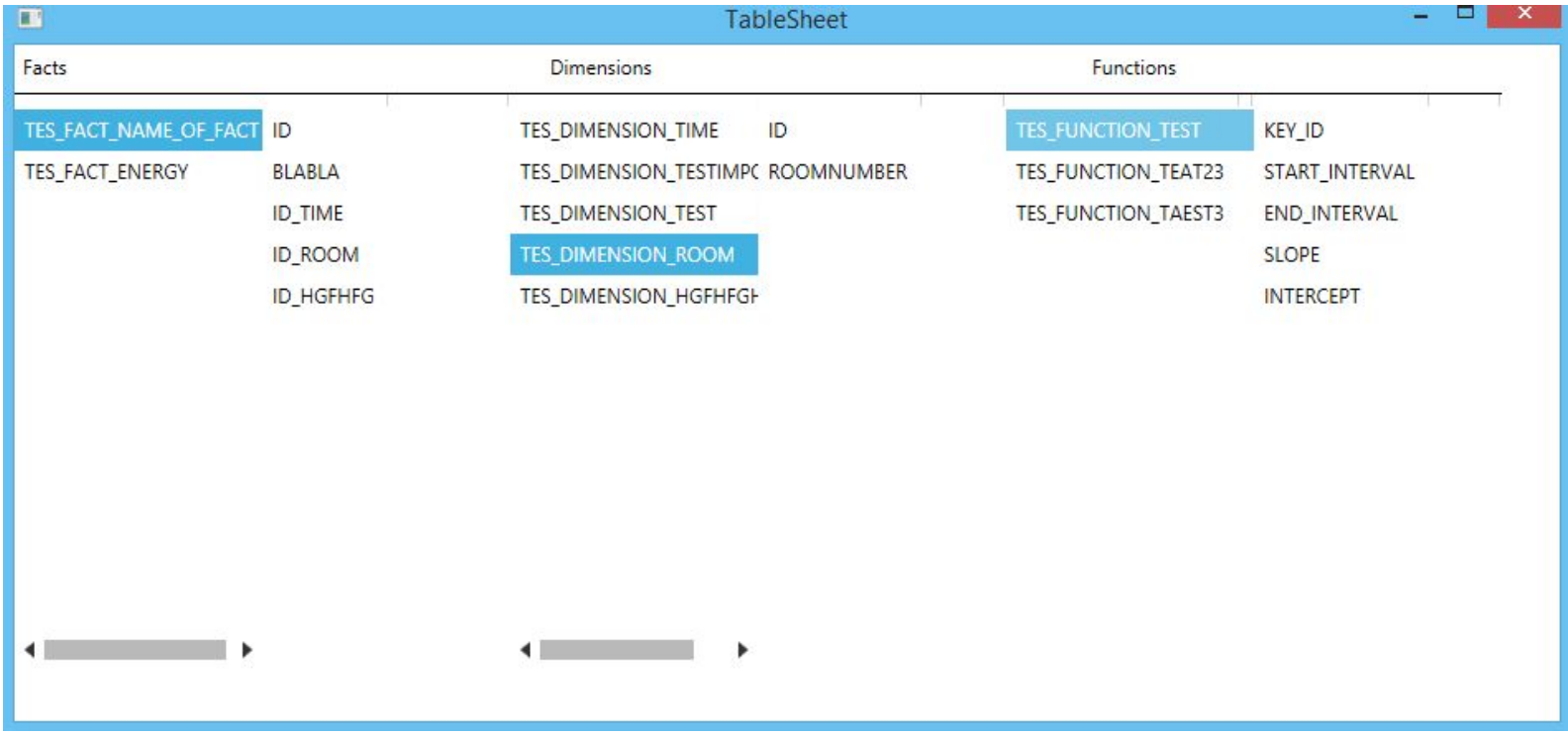
4. Click *Chart* button to how window with bar chart.

- Button *Refresh lists* will download for database all logs.
- Button *Clear selected queries view* to clean second list view.
- Button *Delete selected form database* will delete all query logs in second list view in database.


PLOTWINDOW
Query time
Time [ms]
0
1
2
3
4
5
6
7
8
6
5
8
1
Query_1
Query_2
Query_3
Query_4
Queries

19. Table sheet

- Some views contains *Table sheet* button. This button open new window.
- Window contains tables name in three columns (*Dimensions, Facts, Functions*).
- Click on table name to show columns



20. Import file examples

- Import file must be *csv* file
- Required markers
  - table config row - table; <table_name>;<FACT,FUNCTION,DIMENSION>
  - column config row - columns;<column_name1>|<column type1> <primarykey>;<column_name2>|<column type2> <primarykey>;(...)
  - data - after this next rows must be contains data to insert, this marker must be the last one config markers.
  - insert row - <data1>;<data2>;(..)<dataN>
- Optional markers
  - onlydata - if table exists rows after data marker will be add to database, otherwise you will get MessageBox. **File must contain this marker if table exists.**
  - drop - drop old table if exists and create new one
- Example 1 - create table + insert

```
table;testimport;dimension;
columns;id|int;name|varchar2(50);test|int;
data;
1;test;1;
2;test2;2;
3;test3;3;
4;test4;4;
```

- Example 2 - only data

```
onlydata
table;testimport;dimension;
columns;id|int;name|varchar2(50);test|int;
data;
5;test5;5;
6;test6;6;
7;test7;7;
8;test8;8;
```

- Example 3 - drop and create table + insert

```
drop
table;testimport;dimension;
columns;id|int;name|varchar2(50);test|int;
data;
8;test8;8;
9;test9;9;
```

21. GitHub link

- https://github.com/Adek2kk/IntervalAppGit