

Apricot DB

The User Guide

v 0.2



Preface

The “Apricot DB” is a desktop application, which purpose is to help users with the design, generation, analysis and maintenance of the structure of the complex relational databases.

The main audience of this software are presumable the software developers, who’s been working with the multi- tables databases and need a tool for graphic representation of existing and newly created database- structures.

The database structure is a set of tables, connected with each other by relationships (that’s been making the database “relational”). Tables also can be called “entities”. The entity-relationship diagram (ER-diagram, or ERD) is a graphical representation of the database structure.

Running the “Apricot DB”

The runnable file of the current edition of “Apricot DB” is a Java- archive, which can be started by command:

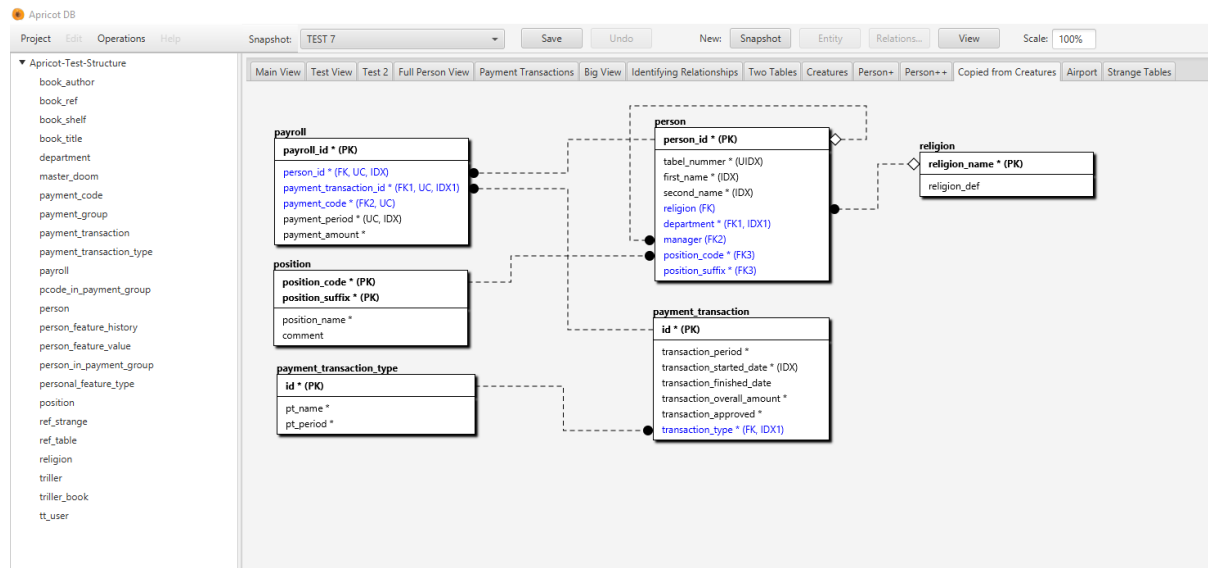
```
$ java -jar apricot-ui-xx.jar
```

The “apricot-ui-xx.jar” is a runnable Java archive. You need the Java Machine of version 8.x to be installed on your system.

Introduction

Here I'll explain the important basics of "Apricot DB".

The "Apricot DB" application is similar to the "classic" IDEs like Eclipse, NetBeans. The top element of the interface is the Project. There might be any number of the Projects created and maintained in "Apricot DB", but only one Project can be opened at any moment of time.



Ideally, a Project is dedicated to one database (or schema, or any other stable database structure depending on the type of the database). In my example (see the screenshot) the project named "Apricot-Test-Structure" represents the SQL Server database - "APRICOT_TEST".

The Project contains Snapshots and Views.

The Snapshot

The Snapshot is a container of tables and relationships between them. Project has to have at least one Snapshot. There might be any number of Snapshots in one Project. The different Snapshots in the Project might have not equivalent collections of tables/relationships. The Snapshot might be considered as a primitive version control system, which stores a graphical representation of the database structure in different moments of time. Technically, the Snapshots can have completely different and not overlapping sets of tables in them. It makes sense, though, to stick with the logically the same database structure in all Snapshots of the Project. It allows to perform efficient comparison between different Snapshots in one Project.

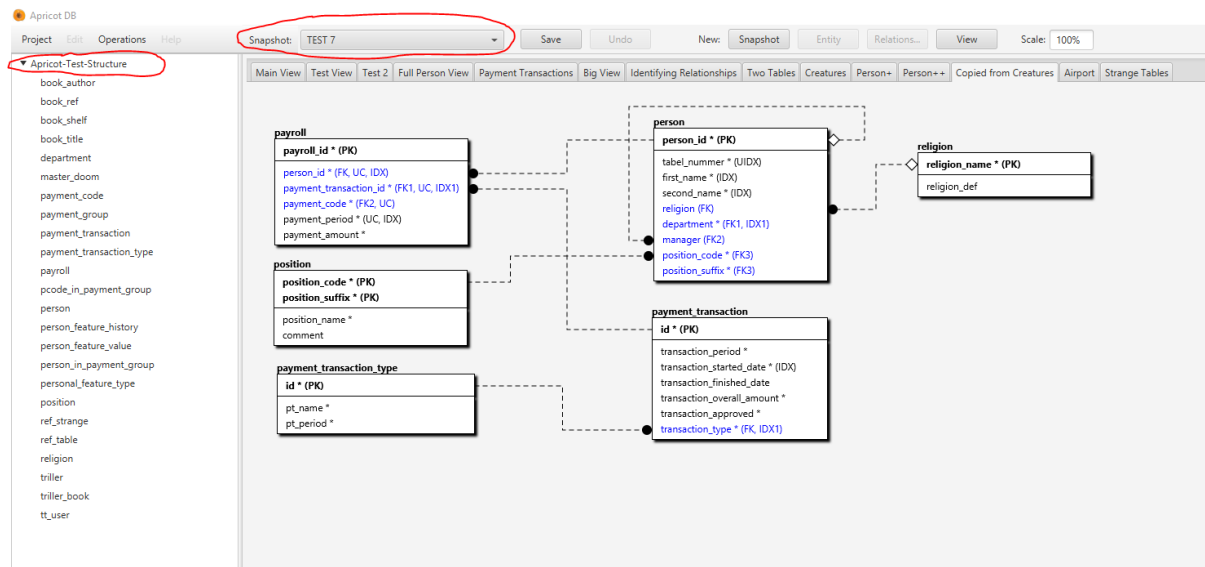
The View

The view contains the graphical representation of entities/relationships (see the screen shot above). There always one mandatory View included in any Project: the "Main View". It shows all tables of the chosen Snapshot. This view cannot be removed.

For the big database structures, which might include hundreds of tables and relationships between them, it is more convenient to "split" the whole set of the tables of the current Snapshot into the logical subsets: Views (do not confuse the "Apricot DB View" with the database view which has been created by CREATE VIEW... DDL command!).

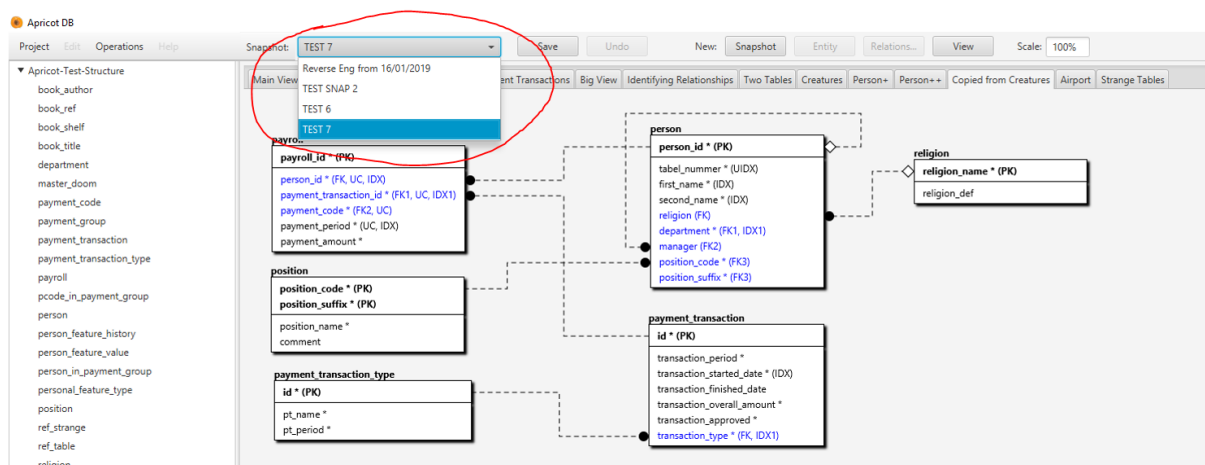
The Elements of the “Apricot DB”- main screen

The current Project and Snapshot: the current Project is shown on the left pane. The Snapshot inside the current Project is presented by the drop-down list.

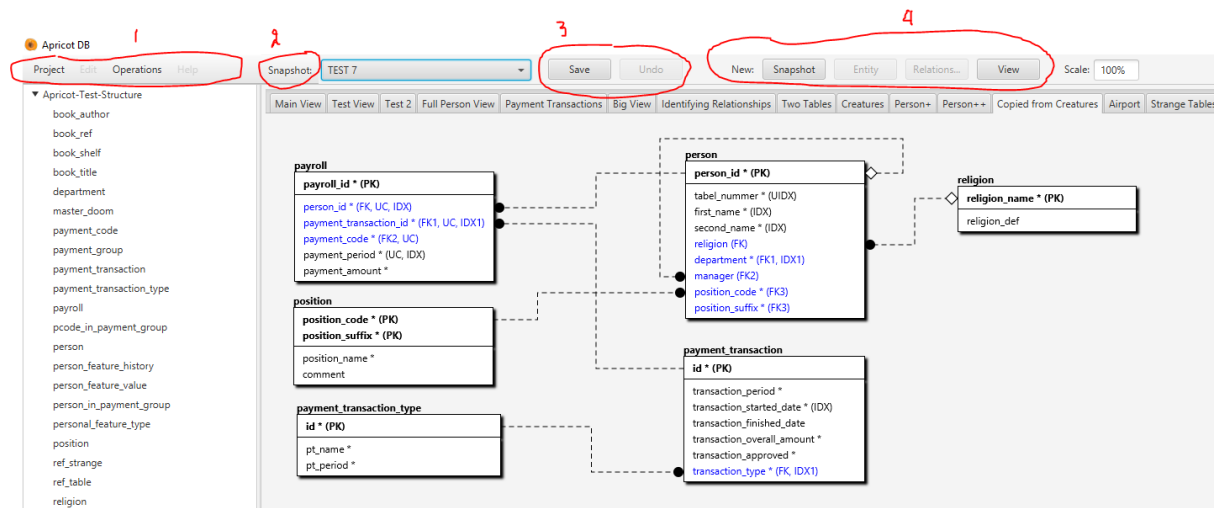


A collection of tables included into the current snapshot is shown under the current Project- name (see the left pane).

To change the current snapshot use the drop down “Snapshot:”, see below:



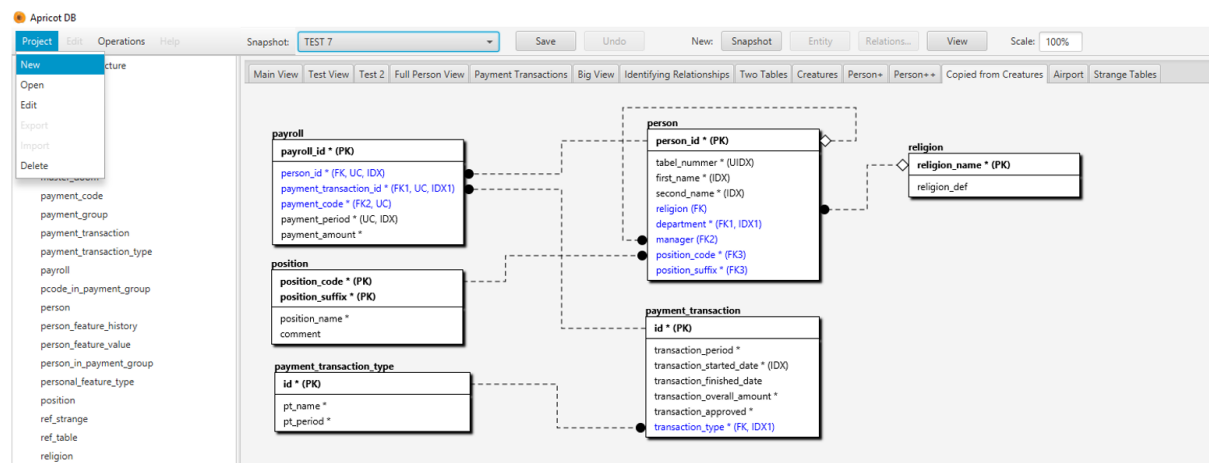
Menus and tool bar



- 1) the main menu is located right above the current project (see the screen shot). It includes some important “Apricot DB” commands;
- 2) menu “Snapshot” – allows edit and delete the current Snapshot;
- 3) menu “Save/Undo” – saves or undoes the recent changes in the selected view;
- 4) menu “New” – allows to create new Snapshots, Views, Entities and Relationships.

Create a new Project

Select Project/New in main menu.



The form of creation of the new Project is shown below.

The Project Name

▼ Apricot-Test-Structure

- book_author
- book_ref
- book_shelf
- book_title
- department
- master_doom
- payment_code
- payment_group
- payment_transaction
- payment_transaction_type
- payroll
- pcode_in_payment_group
- person
- person_feature_history
- person_feature_value
- person_in_payment_group
- personal_feature_type
- position
- ref_strange
- ref_table
- religion
- triller
- triller_book
- tt_user

