1. **Connection in a database :**

* We can create several **connection name** with **the same user** but with **different schema** , to access immediately in our schema .

1. **Execute a SQL statement :**

* If we execute several SQL statement together , it’ll generate a different result tab.
  1. **Snippet**
* It’s there where we save our **query,trigger** that we can run anytime we want .
  1. **Schema navigator .**
* **Select Rows - Limit 1000**: Pulls up to 1000 rows of table data from the live server into a Results tabsheet, and enables editing. Data can be saved directly to the live server.
* **Table Inspector**: Displays table information, similar to the Schema Inspector. This also has a simpler and easier to use interface for analyzing and creating indexes for tables.
* **Table Data Export**: Opens the table export wizard to export the table's data to *JSON* or customized CSV.
* **Table Data Import**: Opens the table import wizard to import *JSON* or CSV formatted data to the selected or new table.
* **Copy to Clipboard**: There are various submenus, each of which copies information to the clipboard:

1. **Relationship model**

* We can create a new model and generate its equivalent of database ; by **Databases🡺Forward engineering .**
* We can generate a relationship model from an existing database by : **Databases🡺Reverse engineering .**
* To display a model in a correct way ; just click on : **Arrange🡺AutoLayout .**
* In the **Bird’s eye** , we can drag the view of the diagram as we want ; or zoom in whatever level we want .
* We can edit the layout of the object in the diagram via **Properties editor**
* We can edit the property of the **metadata** of the tables directly via the interface .
  1. **Synchronise the model with the database .**
* Because our model is reverse engineering from an existing database , we can synchronize of the database to make our changes in the model applied in the database .  
   **Database🡺Synchronise model.**
  1. **Forward engineering**
* We can create our diagram and generate our database by **Databases🡺Forward engineering .**
* We can edit our existing diagram and generate the new database separate of our existing by the same way .
* For database with many tables , we might create several diagrams each one showing only a subset of objects in the database. ; this kind of diagrams called “software article” , it represent a specific part of the database ;   
  example : ***Customer address information..*** *(which represent only : address linkend to city (\*-1) , then city linked to country (\*-1) , and address linked to the customer (1-\*) .*Having create the diagram , we can rename it to reflect its content .
* Once you’ve created the model , you can save it as an MWB file which you can share with others .
* Saving as MWB file also make it appear in the home top as a stored model .
* Because reverse engineering is so common , there’s a shortcut to do it straight from the home screen .

**<!** **The best way is to layout the diagrams with layers and colors ; to make it easier to use** **!>**

1. **Migration from OS to OS** 
   1. **Exemple : Migration from database Windows to Linux** *(very useful)*- To do migration from databse based on windows to Linux , you go to :   
      **Databases🡺Migration Wizard** , and specify the source host , and then the target host . when complete ; just do **Next** ; and then your databases will migrate to the Target host .