Contents

1 Introduction

DBMS stands for *database management system*. It may be a simple function library or even a larger system composed by several programs and processes running separately and in parallel, whose main function is to manage one or several databases hosted in a server. It has the responsibility to manipulate and to keep the consistency of data, allowing the software developers to focus on functionalities. Thus, practically any modern system that manages data utilizes some kind of DBMS, regardless of the amount of stored information.

OmniDB's first version was created as an undergrad final project in the Computer Science Course from the Federal University of Paraná. The objective was to trace a common line between popular DBMS, and to study deeply their *metadata*. The result was a tool capable of connecting and identifying the main structures (tables, keys, indexes and constraints), in a generic way, from several DBMS:

- Firebird

- MariaDB / MySQL
- Oracle
- PostgreSQL
- SQLite
- Microsoft SQL Server

Since early development, OmniDB was designed as an web app. Consequently, it runs in any browser, from any operational system. It can be accessed by several computers and multiple users, each one of them with his/her own group of connections. It also may host any operational system, without the need of install any dependencies. We will see further details on installation in the next chapters.

OmniDB's main objective is to offer an unified workspace with all functionalities needed to manipulate different DMBS. DBMS specific tools aren't required: in OmniDB, the context switch between different DBMS is done with a simple connection switch, without leaving the same page. The end-user's sensation is that there is no difference when he/she manipulates different DBMS, it just feels like different connections.

2 Migration from .NET / Mono to Python

OmniDB was rewritten to Python using the Django framework. Starting from version `2.0`, OmniDB Python version will receive new features and will be actively maintained.

The source code for the ASP.NET/C# version is in the branch `csharp`. The next release of OmniDB C# version is 1.7, and it will only receive bug fixes.

OmniDB source code is hosted on GitHub and there are 3 main branches:

- **master**: Contains the current beta release of OmniDB Python version
- **dev**: Contains the current development release of OmniDB Python version
- **csharp**: Contains the .NET / Mono version of OmniDB

Besides being written in Python, initial version of OmniDB 2.0 contains the following main differences from the C# version:

- Support to HTTPS;
- It allows query execution in background and cancellation through the use of websockets;
- Initially, only an improved support of PostgreSQL is implemented. More RDBMS support coming soon;
- There is a new Snippet feature;
- Log capabilities and a test suite are being developed;
- You don't need to install dependencies and web servers any more. Everything OmniDB needs is now bundled in a single executable.

3 Installation

In order to run OmniDB, you don't need to install any additional piece of software. Just head to omnidb.org and download the latest zip package for your specific operating system and architecture:

- Linux 32 bits / 64 bits
- Windows 32 bits / 64 bits
- Mac OSX

Extract the zip file somewhere in your computer. It will create a folder called `omnidb`. Get inside this folder and run the `omnidb` executable.

```
user@machine:~$ cd omnidb
user@machine:~/omnidb$ ./omnidb
Starting OmniDB at http://localhost:12000
Open OmniDB in your favorite browser
```

Now that the web server is running, you may access OmniDB web app on your favorite browser. Type in address bar: `localhost:12000` and hit `Enter`. If you are running in any port other than `12000`, utilize the appropriate port number. If everything went fine, you shall see a page like this:

Now you know that OmniDB is running correctly. In the next chapters, we will see how to login for the first time, how to create an user and to utilize OmniDB.

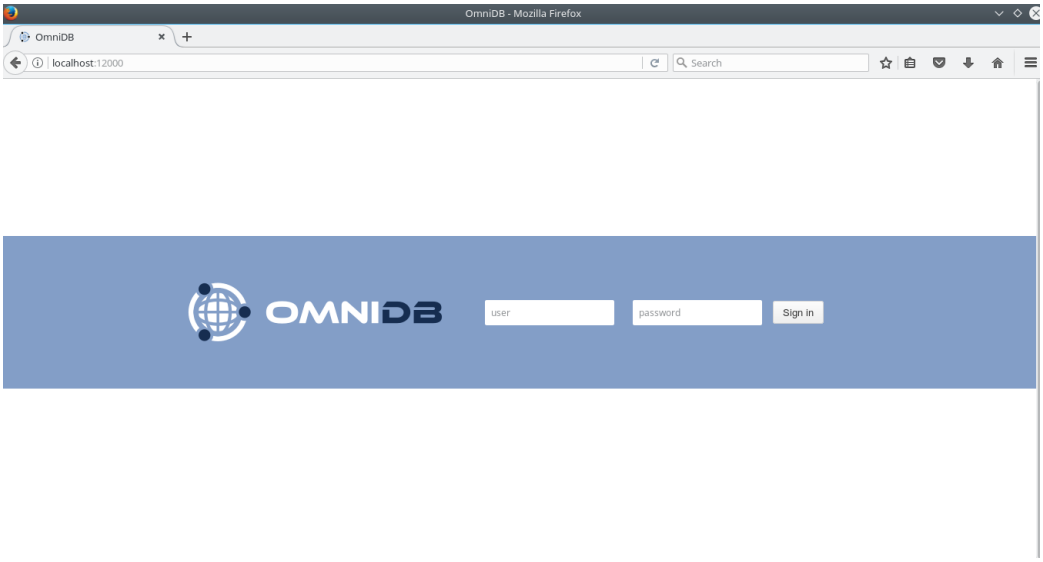

Figure 1: First execution

4 Creating Users and Connections

4.1 Logging in as user *admin*

OmniDB comes only with the user *admin*. The first thing to do is sign in as admin, the default password is admin.

The next window is the **Connections** window. We will talk about it later.

4.2 Creating another user

Click on the *Users* tab.

After clicking on the *New User* button the tool inserts a new user called *user2* (if that is the first user after *admin*).

You will have to change the *username* and *password*. Check if you want this new user to be a *super user*. This user management window is only seem by super users. When you are done, click on the *Save Data* button.

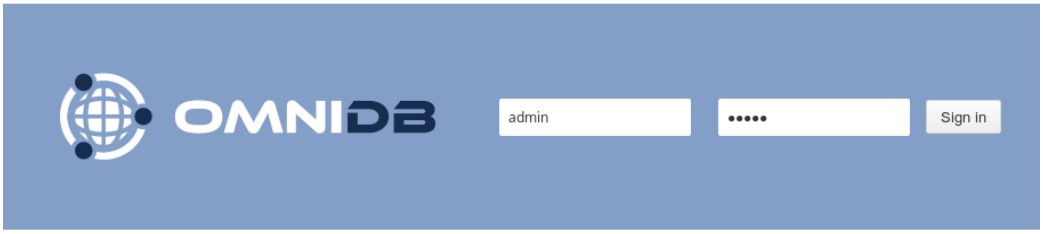

Figure 2: Signing in as admin

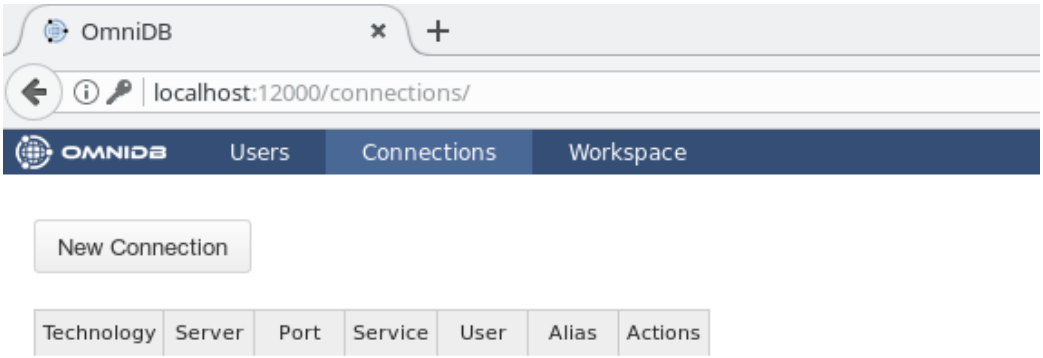

Figure 3: Connections window as admin

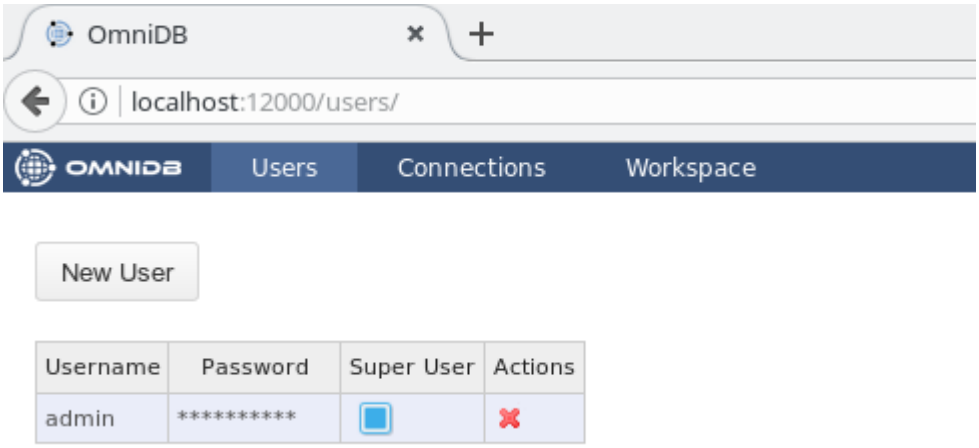

Figure 4: User management window

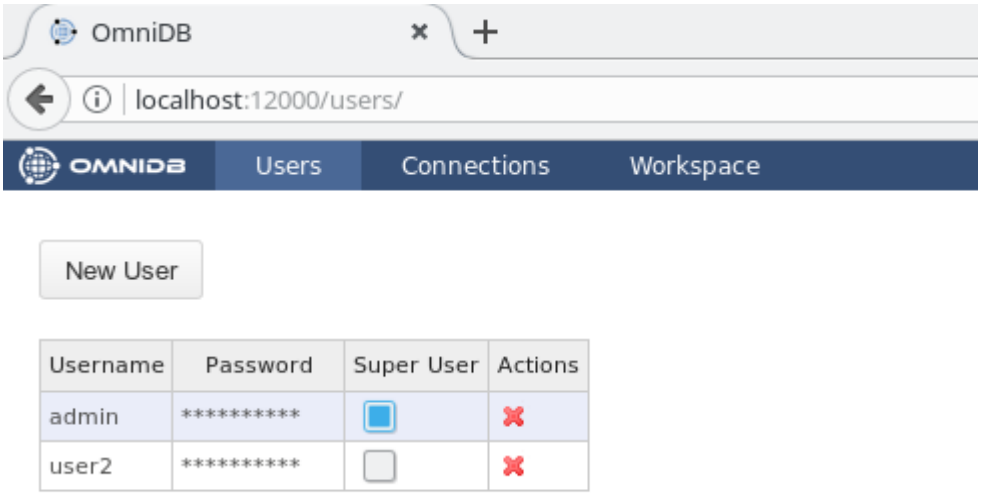

Figure 5: Creating new user

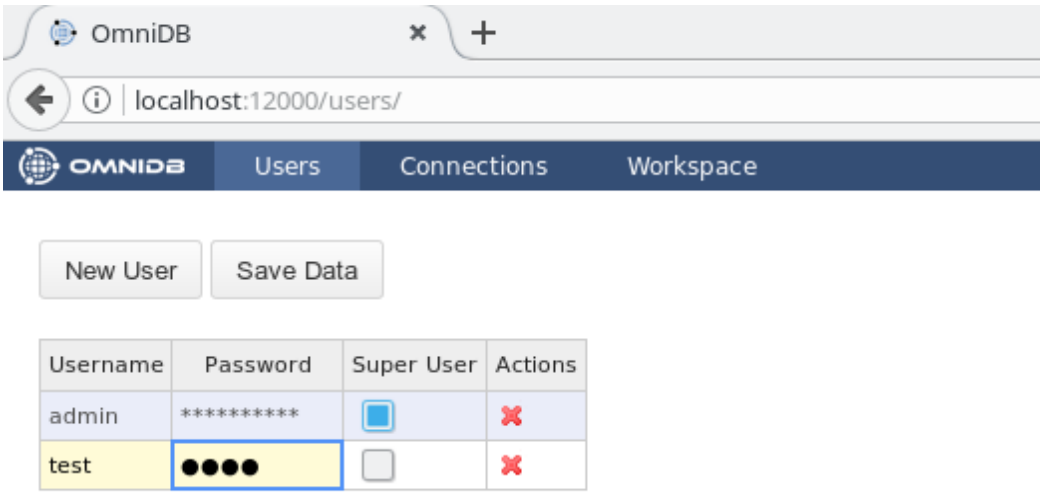

Figure 6: Changing new user attributes

You can create as many users as you want, edit existing users and also delete users by clicking on the red cross at the actions column. Now you can logout.

4.3 Signing in as the new user

Let us sign in as the user we just created.

Figure 7: Signing in as the new user

And we can see the **Connections** window again. Note that now there is no *Users* tab, because the *test* user is not a super user.

4.4 Creating connections

OmniDB C# version supported several DBMS. At the moment, OmniDB Python version, or `OmniDB 2.0`, supports only PostgreSQL. More DBMS support is being added as you read this.

We will now create two connections to PostgreSQL databases. To create the connections you have to click on the button *New Connection* and then choose the connection and fill the other fields. After filling all the fields for both connections, click on the *Save Data* button.

For each connection there is an *Actions* column where you can delete and test them. Go ahead and test one of the connections.

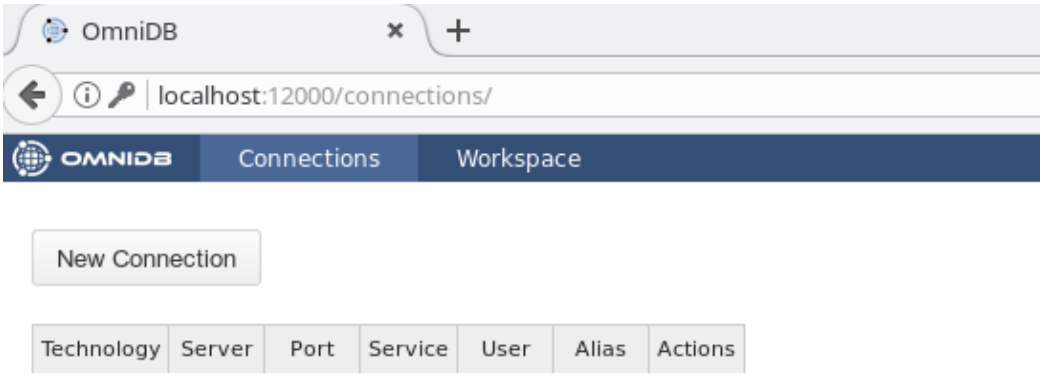

Figure 8: Connections window

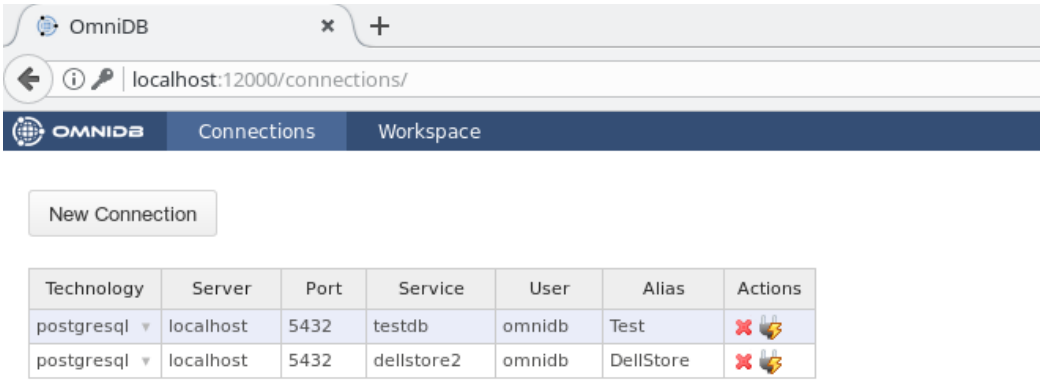

Figure 9: Some connections created

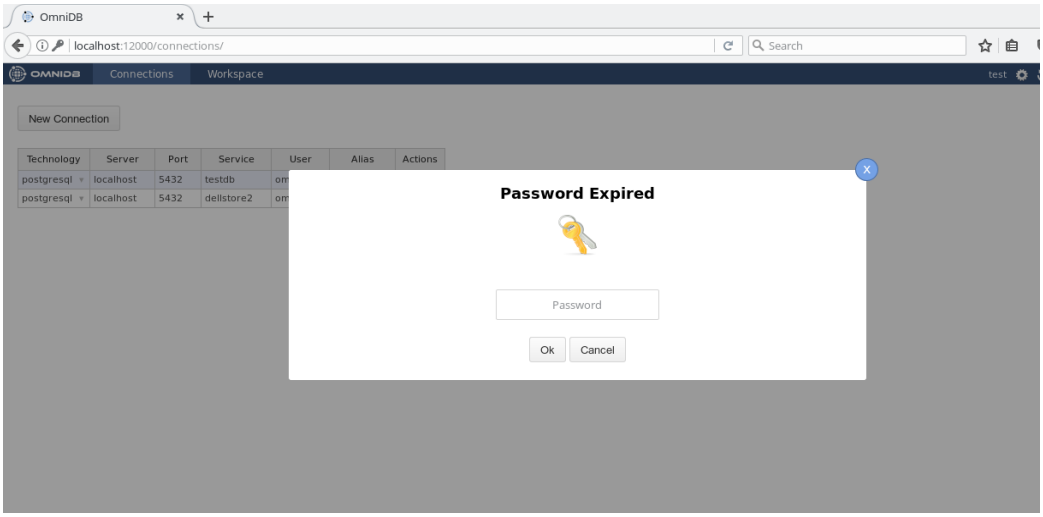

Figure 10: Some connections created

Notice the *Password Expired* popup. This is happening because OmniDB does not store the database user password on disk. When the user types a password in this popup, the password is encrypted and stored in memory.

After you type the password and hit *Enter*, if the connection to the database is successful you will see a confirmation popup.

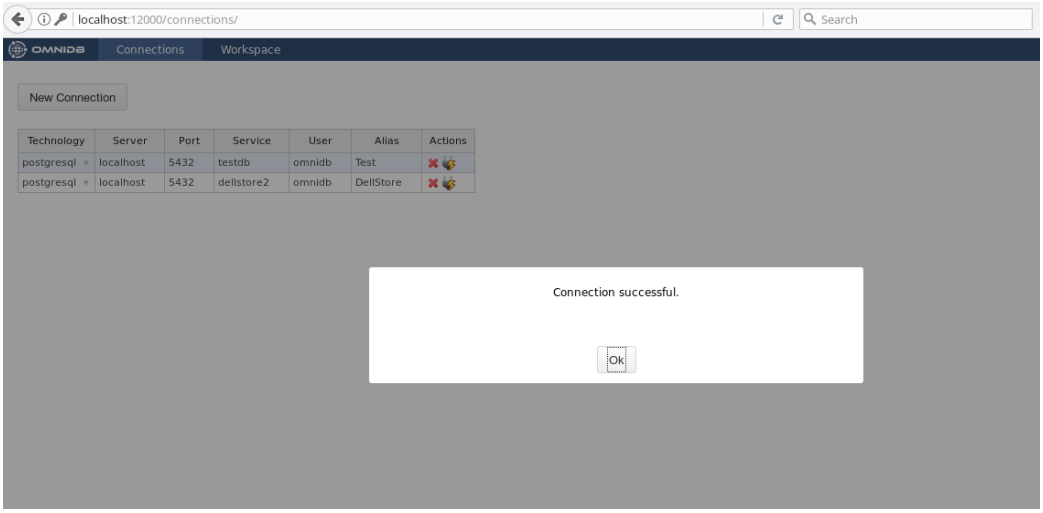

Figure 11: Connection successful

But, if you have trouble of any kind connecting to your PostgreSQL database, the *Password Expired* popup will remain showing the error OmniDB got.

5 Workspace

After creating at least one connection the user can enter the *Workspace*.

5.1 Sections of the *Workspace* window

This interface has several elements:

- **1) Links**: Enables the user to navigate between OmniDB windows

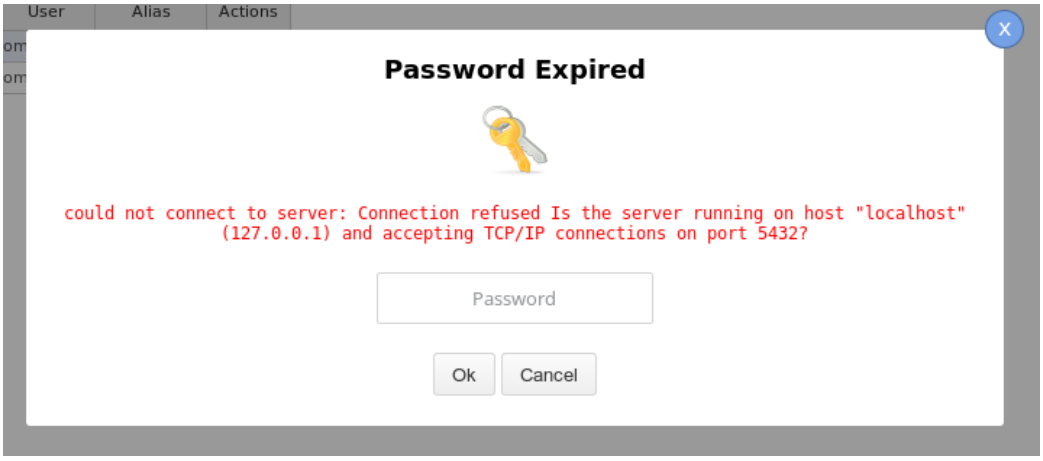

Figure 12: Connection unsuccessful

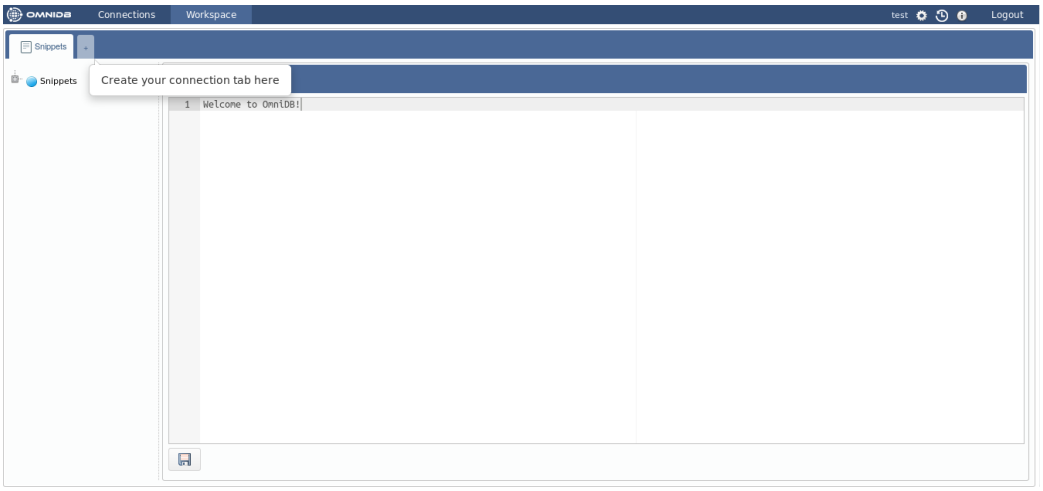

Figure 13: Workspace window

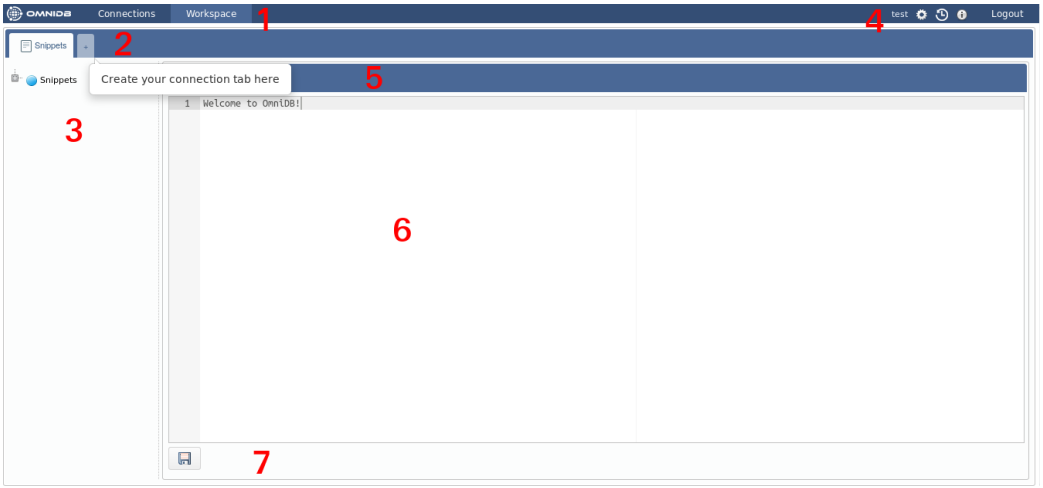

Figure 14: Sections of the Workspace window

- **2) Outer Tabs**: OmniDB lets you work with several databases at the same time. Each database will be accessible through an *outer tab*. Outer tabs also can host miscellaneous features, like the *Snippets* feature
- **3) Tree View**: Depending on the feature, the *tree view* can show different kinds of elements arranged in a hierarchical tree
- **4) Options**: Shows the current user logged in, and also links for *user settings*, *query history*, *information* and *logout*.
- **5) Inner Tabs**: For each *outer tab* you are working on at the moment, you can open several *inner tabs* to perform several kinds of activities
- **6) Inner Tab Content**: The content of a *inner tab* can be an *editor* or several kinds of forms
- **7) Inner Tab Actions and Results**: If the *inner tab* hosts and *editor*, you may see different actions and results

5.2 Connection Outer Tab

Notice the little tab with a cross besides the *Snippet Outer Tab*. This allows you to create a new outer tab that will automatically be a *Connection Outer Tab*. However, the *Snippet Outer Tab* is fixed and will always be the first.

Now create a new *Connection Outer Tab*.

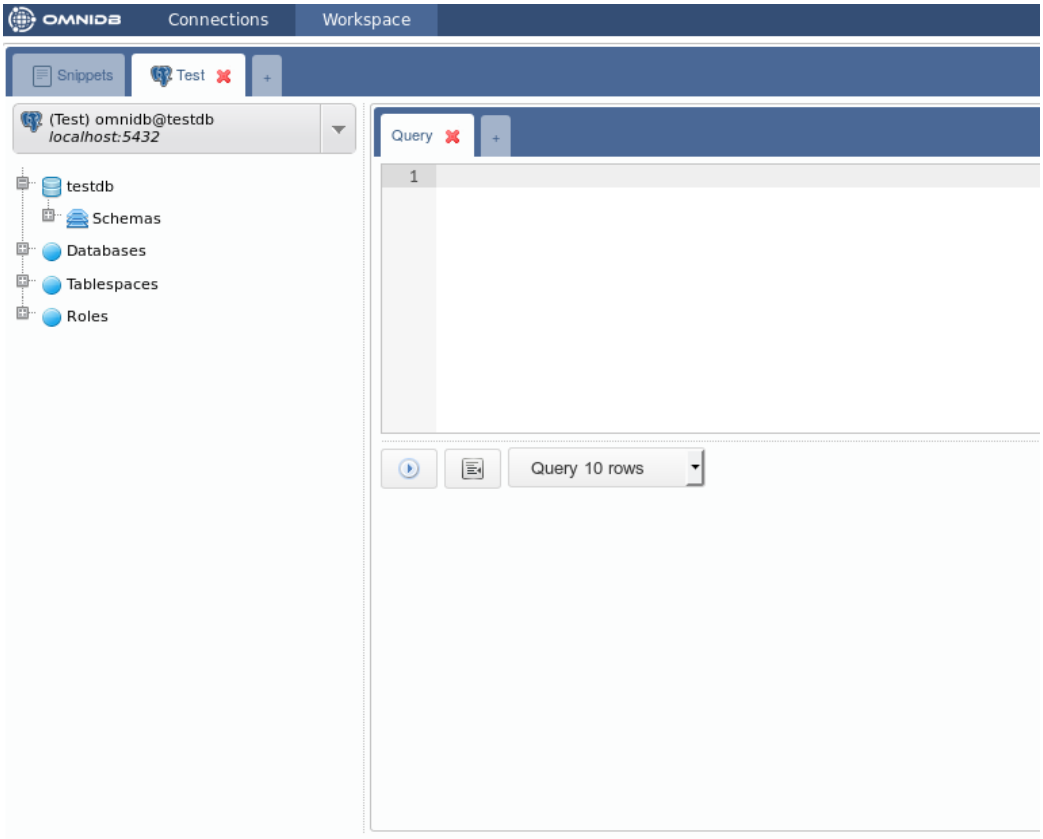

Figure 15: Connection outer tab

A new *Connection Outer Tab* will always automatically point to the first connection on your list of database connections. Observe the elements inside of this tab:

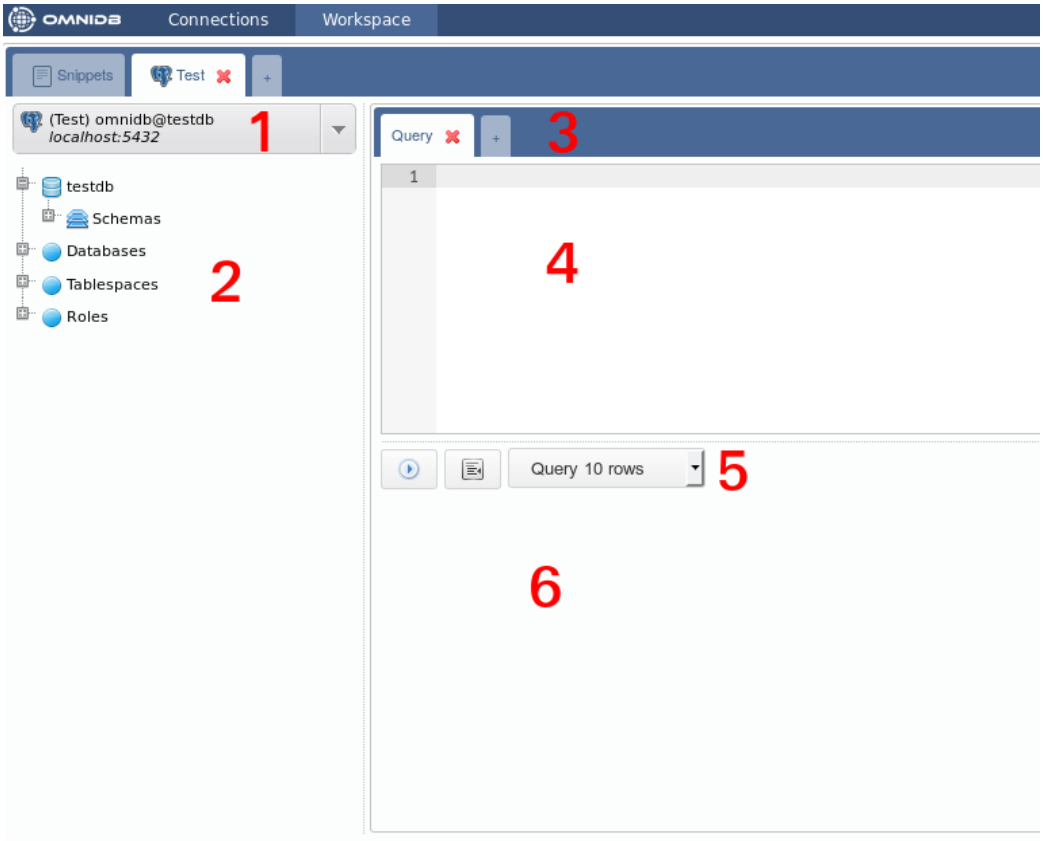

Figure 16: Sections of the Connection outer tab

- **1) Connection Selector**: Shows all connections and lets the user select the current one
- **2) Tree of Structures**: Displays a hierarchical tree where you can navigate through the database elements
- **3) Inner Tabs**: Allows the user to execute actions in the current database. There are several kinds of inner tabs for the current database. By clicking on the last small tab with a cross, you can add a new tab. A new tab always will be a *Query Tab*, where you can write any kind of SQL statement

- **4) Inner Tab Content**: Can vary depending on the kind of inner tab. The figure shows a *Query Tab* and in this case the content will be an *SQL Editor*, with syntax highlight and autocomplete
- **5) Inner Tab Actions**: Can vary depending on the kind of inner tab. For a *Query Tab*, they are *Execute Button*, *Format Button* and *Editor Mode* (script, execute or query)
- **6) Inner Tab Results**: A *Query Tab* in query mode, after you click in the *Execute Button* or type the execute shortcut (`Alt-Q`), will show a grid with the query results. All modes will show error messages, if any.

5.3 Working with databases

Take a look at your connections selector. OmniDB always points to the first available connection but you can change it by clicking on the selector.

Select the *DellStore* connection. Now go to the tree right below the selector and click to expand the node *Schemas*.

Bear in mind that every 10 minutes you keep without performing actions on the database, will trigger a *Password Expired* popup. As explained before, this is important for your database security. After you type the correct password, you will see all schemas in your database (in case of PostgreSQL, `TOAST` and temp schemas are not shown).

Now click to expand the schema `public`. You will see different kinds of elements contained in this schema.

Now click to expand the node *Tables*, and you will see all tables contained in the schema `public`. Expand any table and you will see its columns, primary key, foreign keys, unique constraints and indexes. Each column is also expansible, displaying data type and nullable constraint.

In order to view records inside a table, right click it and choose *Data Actions > Query Data*.

Notice that OmniDB fills the current SQL editor with a simple query to list table records. The records are displayed in a grid right below the editor. This grid can be controlled with keyboard as if you were using a spreadsheet manager. You can also copy data from single cells or block of cells (that

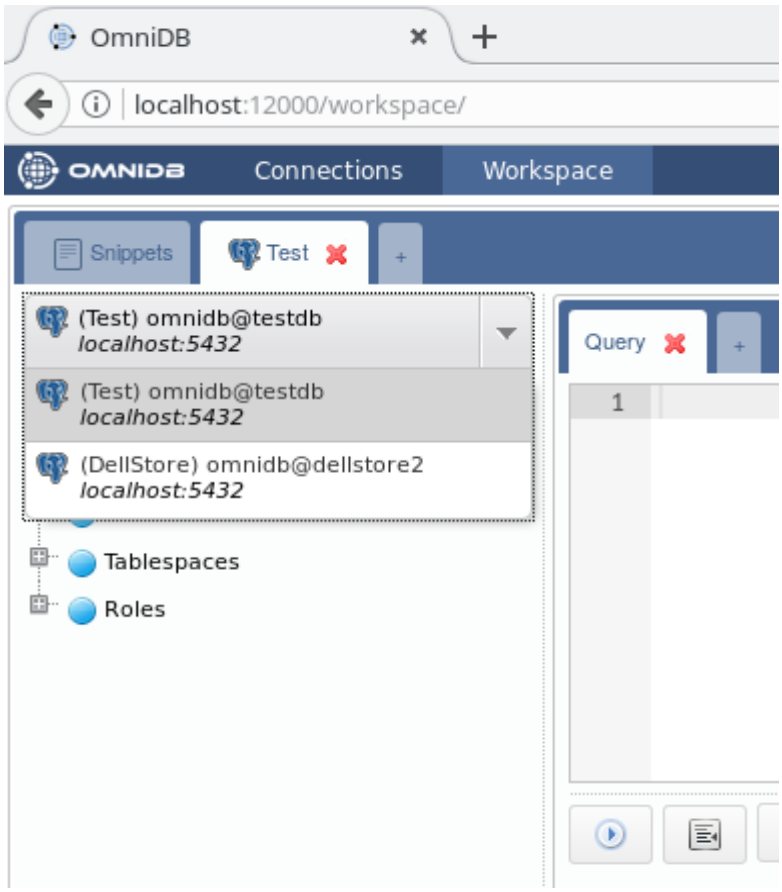

Figure 17: Changing current Connection

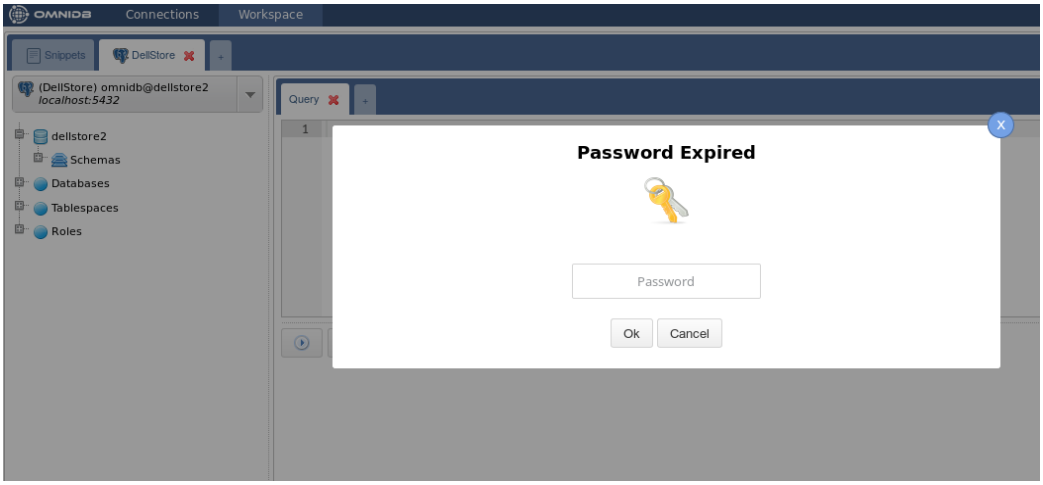

Figure 18: Password expired

can be selected with the keyboard or mouse) and paste on any spreadsheet manager.

You can edit the query on the SQL editor, writing simple or more complex queries and clicking on the action button. You can control how many records should be displayed (10, 100, 1000 or all rows). More details in the next chapters.

5.4 Working with multiple tabs inside the same connection

Inside a single connection, you can create several inner tabs by clicking on the last little tab with a cross. Each new inner tab will be a *Query Tab*.

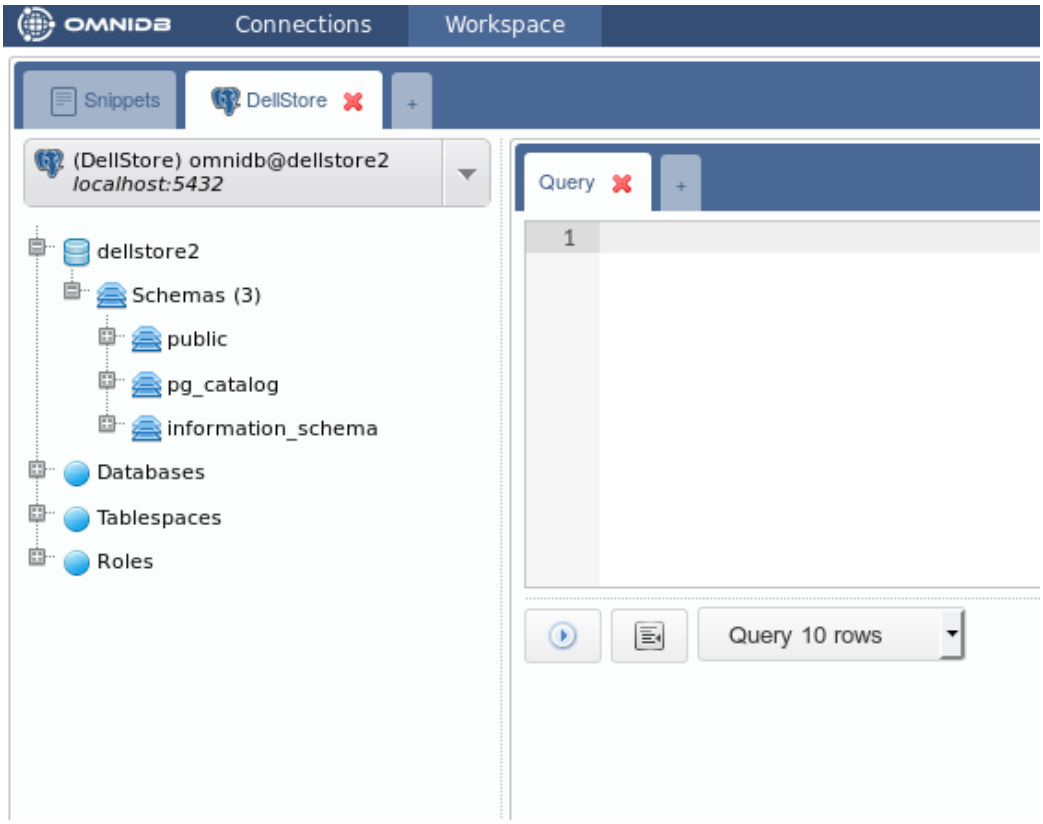

Figure 19: Schemas

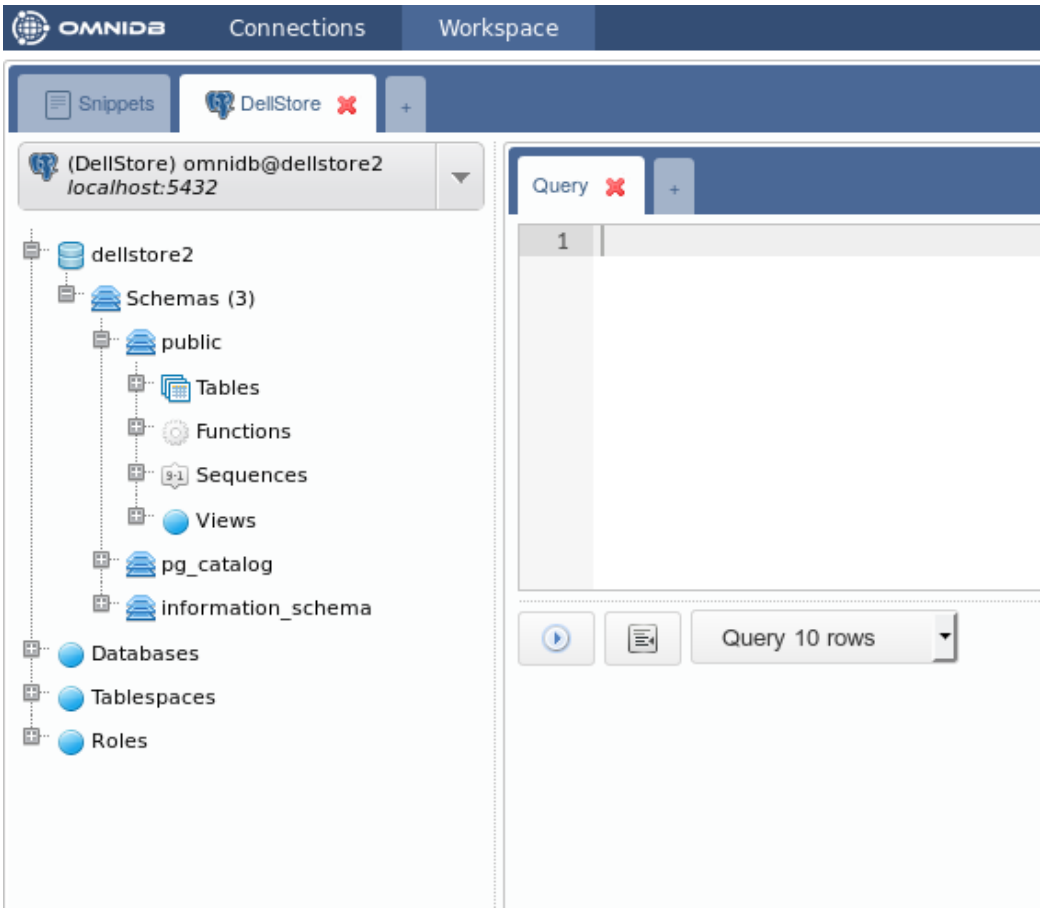

Figure 20: Schema public

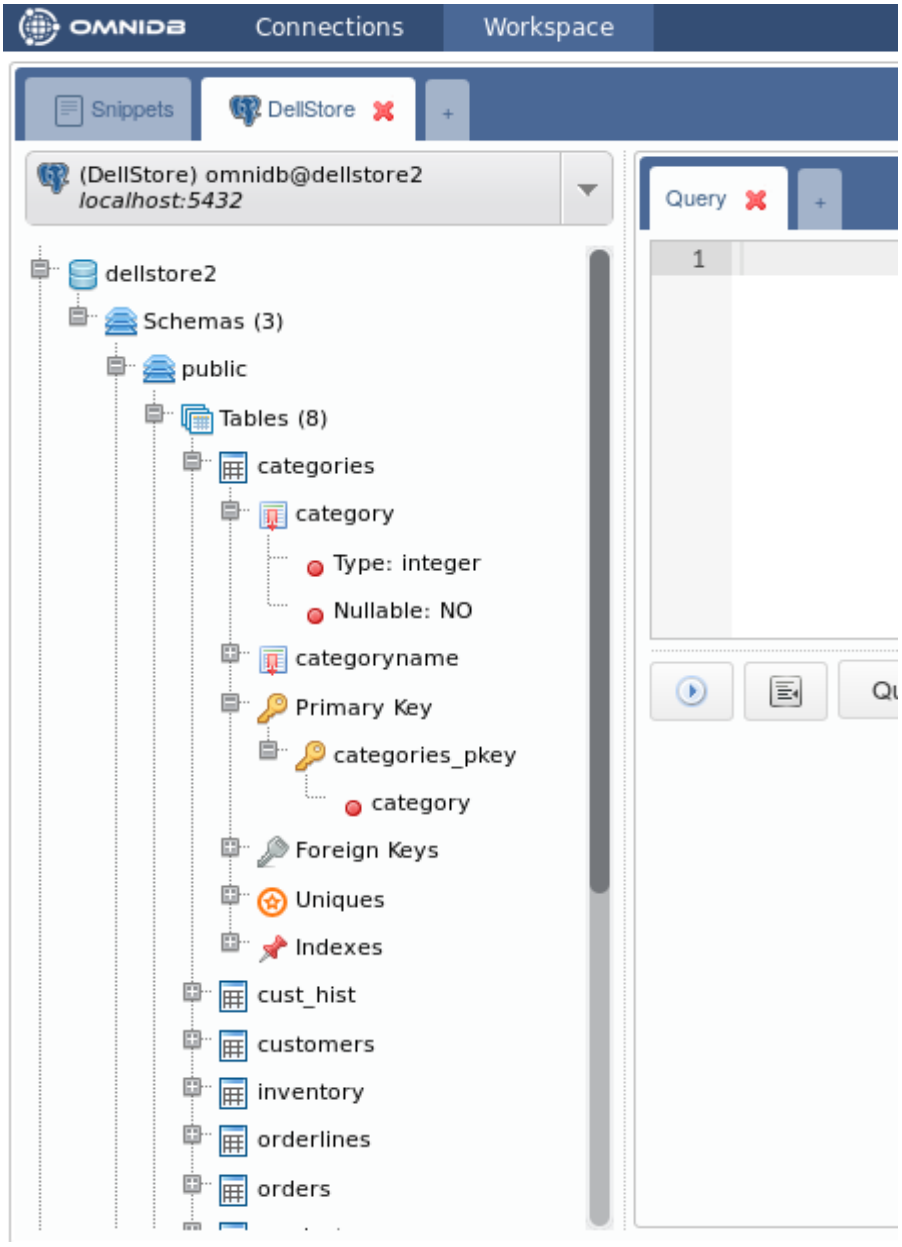

Figure 21: Tables

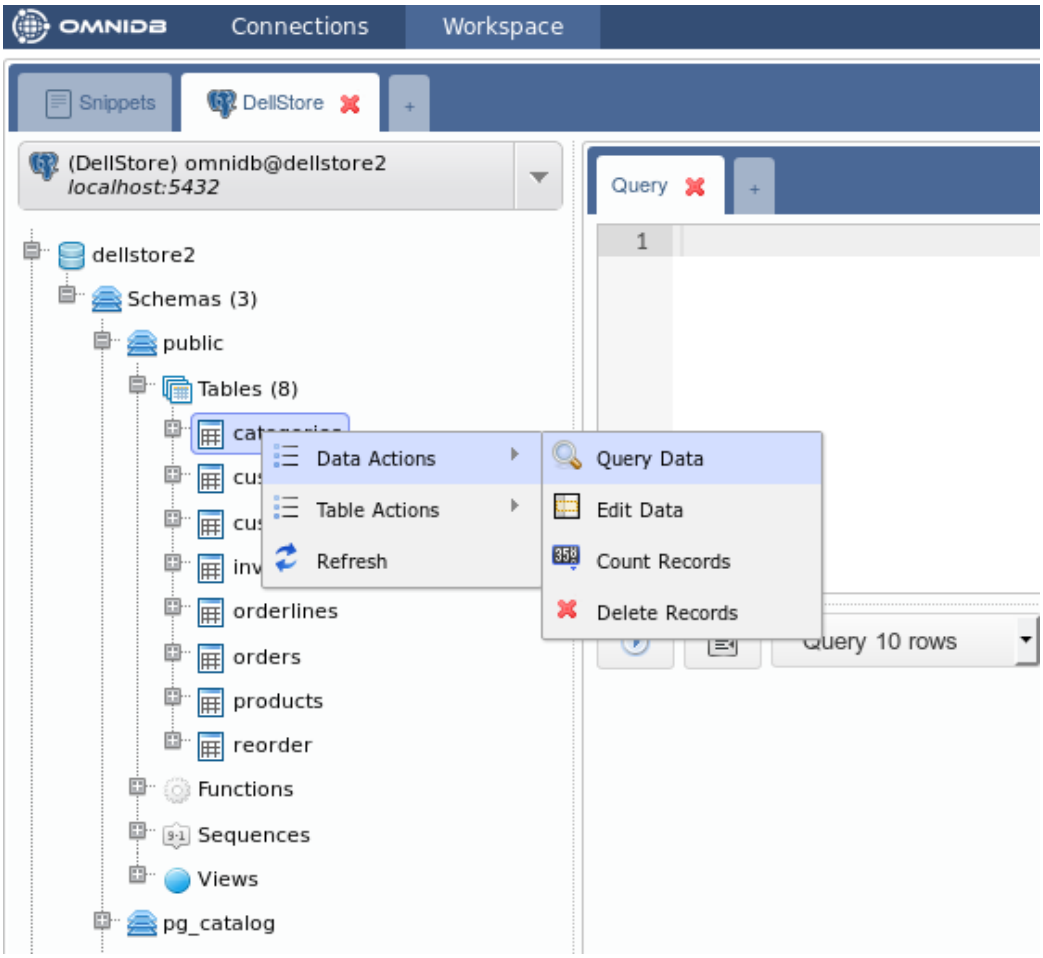

Figure 22: Query Data context menu

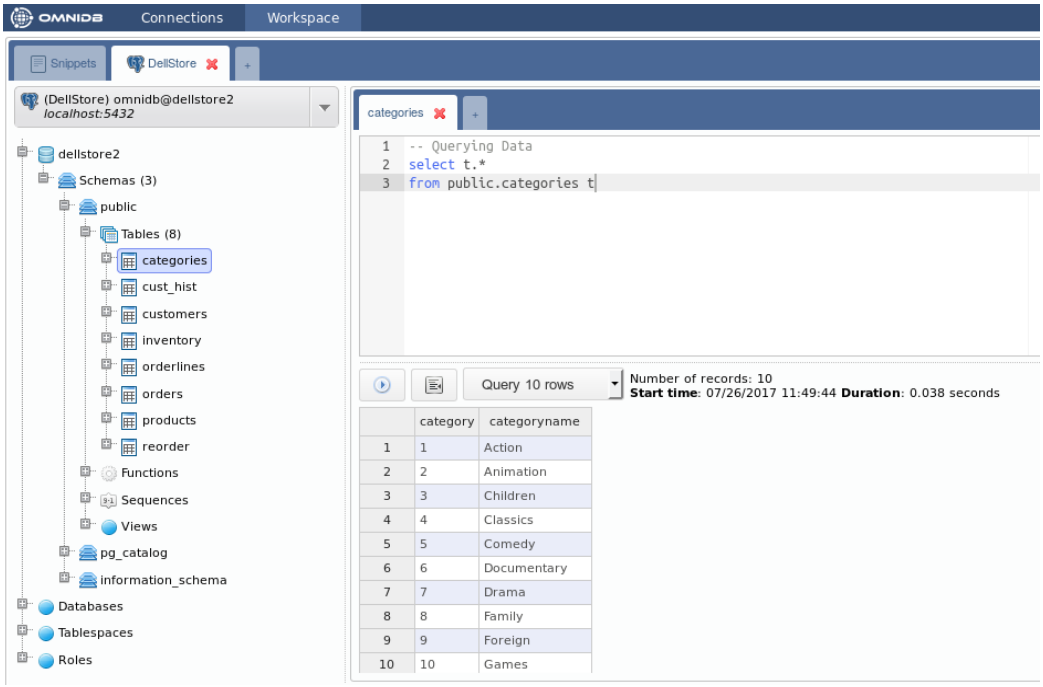

Figure 23: Query Data

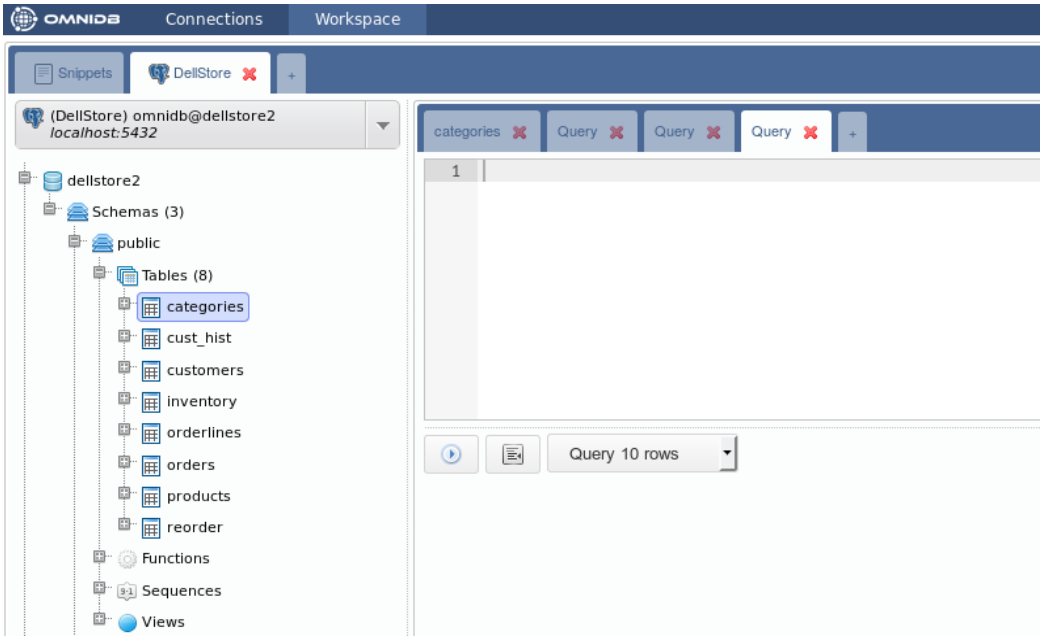

Figure 24: Several inner tabs

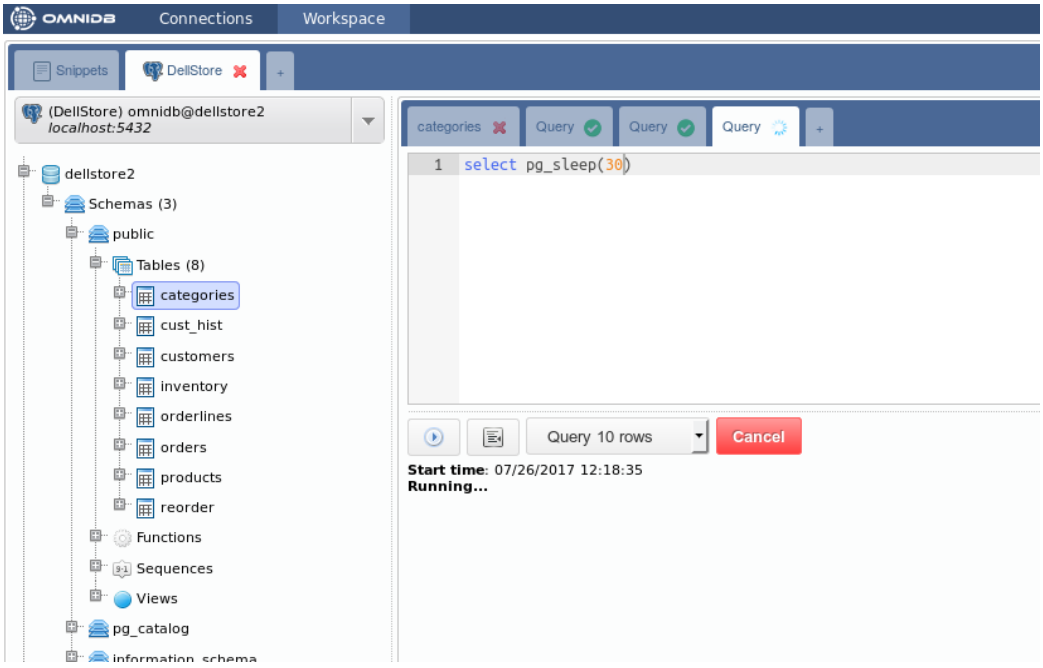

Figure 25: Several inner tabs