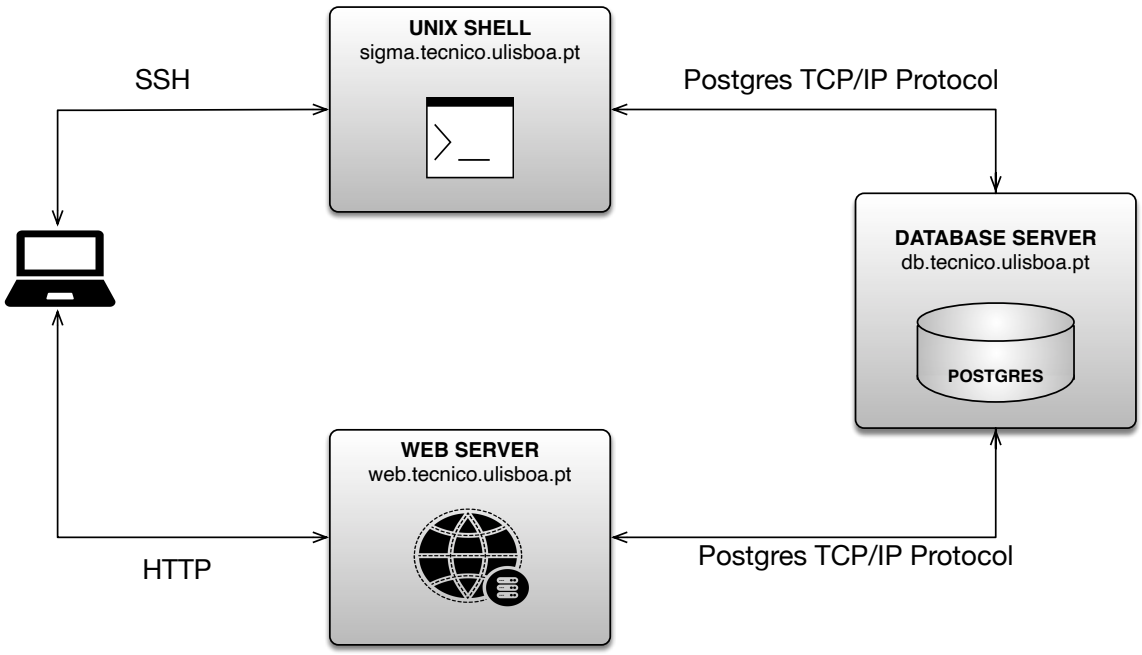
Sistemas de Informação e Bases de Dados

Lab 1: Introduction to the environment

The **PostgreSQL** database management system, as well as the rest of the Shell Unix (Linux) services and the Web server System (where Python code is executed) are available in the IST infrastructure, as shown in the following picture:



Before doing this lab, you should review the following concepts:

- Shell Unix
- SSH e SCP

Settings and Postgres system login

In order to start using the **Postgres** system available in the IST infrastructure, you should proceed with the following steps:

1. To be able to complete this lab, you must first access the self-service page at DSI:

   https://selfservice.dsi.tecnico.ulisboa.pt/

   and turn on ("*Activar*") the shell, web and cgi services.

2. To access IST's Cloud infrastructure you will have to use a program colloquially known as "Secure Shell", that enables you to establish a secure session with the host **sigma.tecnico.ulisboa.pt**.

   - **Windows 10, Linux or Mac OS:** Open a terminal and use the command

     `ssh istxxxxxx@sigma.tecnico.ulisboa.pt`[1] ↵

   - **Windows 8.1 and earlier versions**: You can use a "SSH Secure Shell" program or "PuTTY" (which you must download) and connect to host **sigma.tecnico.ulisboa.pt**

   Note 1: Sigma login is always done using your Fénix credentials (Fénix username and password). For security reasons, when you write your password the **characters do not show on screen**.

   Note 2: The first time you establish a secure connection with a new server (host) you will be requested to accept the ssh key. You can accept it. In most cases this will mean writing 'yes' or pressing the 'Ok' button.

3. When you login to **sigma** you should see a Shell Unix command line with a prompt similar to:

   ```
   ist123456@sigma01:~$
   ```

4. To get your **Postgres** account password, and after logging into the sigma cluster, execute the following command:

   `psql_reset` ↵

[1] If you get the **ssh: connect to host sigma.tecnico.ulisboa.pt port 22: Connection timed out error**, make sure your firewall is not blocking SSH connections

5. To log into **Postgres** use command:

   psql -h db.tecnico.ulisboa.pt -U istxxxxxx↵

   Where **istxxxxxx** should be replaced with your Fénix username. You should enter the password you got from the previous command when prompted. You should write this command with the spaces as shown.

6. You will get a **Postgres** session, which will be displayed by the change in the command line prompt to:

   ist123456=>

7. If you wish to update the **Postgres** password you can use the following command:

   ALTER USER istxxxxxx WITH PASSWORD 'mypassxxxx'; ↵

   where **mypassxxxx** is the new Postgres password.

8. Use command **\q ↵** to exit the system and return to the terminal.

9. Note that the command line prompt return to:

   ist123456@sigma01:~$

Transferring files to Sigma

10. Download the Zip file that comes with this lab.

11. To copy files from your PC to your file area within sigma.tecnico.ulisboa.pt you should use a specific Secure Copy program.

    - **Windows 10, Linux or Mac OS:** Open a terminal and use command "scp":

    `scp <file-path> istxxxxx@sigma.tecnico.ulisboa.pt:`

    Windows 8.1 and earlier: Download and install "WinSCP" and establish a connection with sigma

    Note 1: If you are downloading on a Linux PC in the Lab, copying will, in principle, not be required since the download will already leave the files in your network volume (same that is mounted in your area in sigma).

    Note 2: Optionally you can install "FileZilla" to make the file transfers easier. This software runs on Windows, Linux and Mac OS.

    Note 3: On Windows, do not run **scp** from inside a directory path that has spaces. The **scp** command will not work! If your **scp** commands are not working on Windows, try moving the downloaded files into a C:\TMP directory and run and **scp** commands from there.

    Note 4: Please make sure you are typing the **scp** (or any other command) in the correct terminal window.

12. Copy the *`bank.sql`* file to the root directory (~) of your work area in **sigma**.

13. Copy the *`test.cgi`* file to the *`web`* folder of your work area in **sigma**.

Using the command line

As a rule of thumb, every database management system has a command line interface through which you can execute SQL instructions, along with other administration and system maintenance commands.

Note: All administration commands start with \ (backwards bar). All commands starts with an SQL instruction (**select**, **insert**, **update**, **delete**, **create**, …) and end with ; (semicolon).

14. Log into Postgres again, as shown in step 5.

15. Once in the system use command **\h** ↵ to get information on all available SQL commands.

16. Use command **\?** ↵ to get information on all available administration commands.

17. If at any point you need to exit the program you can use command **\q** ↵**.**

Creating the example database – 'Bank'

The *bank.sql* file contains a set of SQL instructions to create the example database shown in Figure 1.

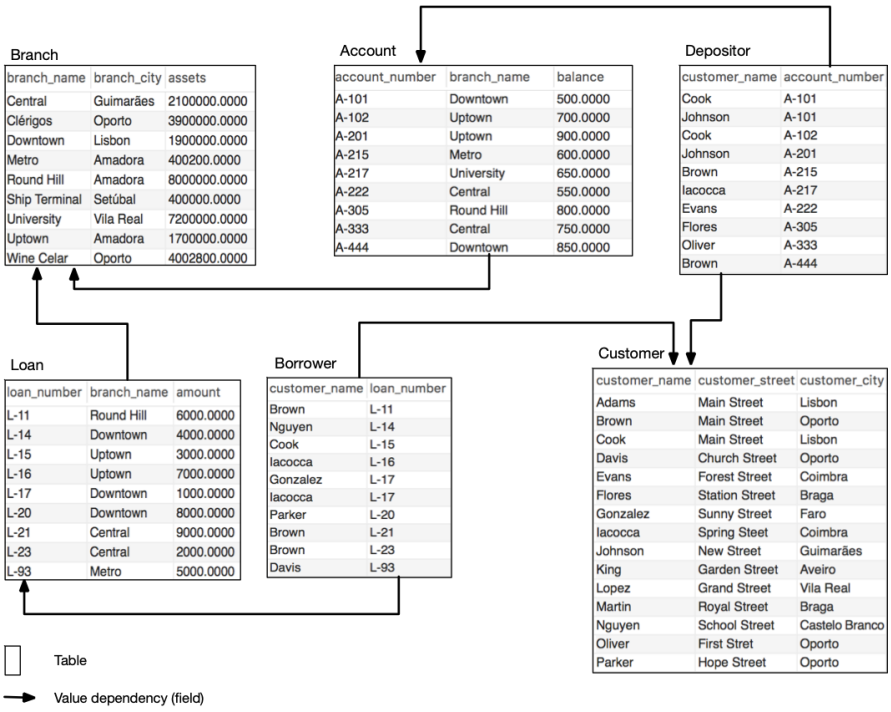


Figure 1. Table organization in the example database

In order to create the database, you need to create the tables and load the data into each table. The table creation is done using the **CREATE TABLE** Instruction. For example, the customer table can be created using the following instruction:

```
CREATE TABLE customer
   (customer_name        VARCHAR(80)  NOT NULL,
    customer_street      VARCHAR (255 NOT NULL,
    customer_city        VARCHAR (30) NOT NULL,
    CONSTRAINT pk_customer PRIMARY KEY(customer_name));
```

This instruction specifies the table name, the name of the three columns, and the type of each column. It also specifies constraints, such as the values cannot be **NULL** and the fact that the table's primary key is the customer name.

The data of each table is loaded through **INSERT** instructions. For example:

```
INSERT INTO customer VALUES ('Luis', 'Rua do Cima', 'Musgueira');
```

This instruction specifies the values for each column in the same order they were defined during the table creation[2]. This instruction creates a new row in the customer table.

Note that *bank.sql* contains instructions to insert more rows than the ones shown in Figure 1. These rows will be used to run various tests over the database. In future lab classes we will use this database to demonstrate various concepts within the course.

Note: Although you can use the graphical interface **DataGrip**[3] to interact with the **Postgres** database management system, in the lab classes we will mostly use the command line interface.

18. Log into your sigma area, as shown in step 2 (if needed, move to the directory where you saved the *bank.sql* file).

19. Log into Postgres, as shown in steps 4—6.

20. Execute this command to create the bank example database:

    \i bank.sql ↵

    To load and execute the SQL instructions in *bank.sql*.

    Postgres outputs some messages while it executes the instruction in the file. Once it is over, the example database will have been created.

21. To list the tables in the database, use command **\d** ↵

[2] There are other variations of the INSERT instruction where you can list the values in another order, or list only certain values, leaving the rest as NULL or with the default value (which was not specified in this case)

[3] The `DataGrip` interface is available at `https://www.jetbrains.com/datagrip`

22. How long the system takes to reply to certain queries is an important factor when the data volume is substantial. Execute command:

    `\timing` ↲ to turn the timing of SQL commands on and off.

23. Once in you Postgres session, you can make some queries with SQL commands, namely:

    - see the full list of clients:

      ```
      SELECT * FROM customer; ↲
      ```

    - see the full list of accounts:

      ```
      SELECT * FROM account; ↲
      ```

    - check balance of account A-101:

      ```
      SELECT balance FROM account WHERE account_number='A-101'; ↲
      ```

    - see all clients that are not depositors (i.e. have no accounts):

      ```
      SELECT * FROM customer
      WHERE customer_name NOT IN (
      SELECT customer_name FROM depositor); ↲
      ```

    In future classes you will learn how to get answers to more complex queries.

Obtaining information on the database schema

When a database is already in the system but there is no documentation on it, it is possible to use special instructions to obtain information on the respective tables. Usually there are proprietary mechanisms and differ for each system. In Postgres these functionalities are available through command **\d** and variants.

24. Use command **\l** (lower case 'L') to get information on all the databases in the system.

25. Get information on customer and account tables:

    \d customer ↲

    \d account ↲

    Confirm that the tables follow the structure established in *bank.sql*.

26. Use command **\q** to exit the system and return to the terminal.

Summary of used Postgres commands

\h	List all SQL commands available and their information.
\q	Exit Postgres command line.
\?	List all administration commands available and their information.
\i <*filename*>	Execute commands contained in the given file.
\c <*databasename*>	Connect to a given *database*.
\d	List tables in the current database.
\timing	Show time to execute queries.
\l	List databases in the system.
\d *table*	Get information on the table structure.

Developing applications with Postgres databases: Python (psycopg) example

The following steps are intended to test connecting to the database using a Python script.

27. Make sure you have an appropriate text editor installed. `Sublime Text`[4] or `Visual Studio Code`[5] are advised.

28. Edit *test.cgi* and place your username and password (the Postgres ones, password was given by `psql_reset`) in variables IST_ID and password, respectively.

29. Using an SCP client ("WinSCP" in Windows; "scp" in Linux) place *test.cgi* in the *web* folder of your sigma.tecnico.ulisboa.pt cluster area.

30. In the command line in Sigma, go to the *web/* folder and run the command **chmod 755 test.cgi** to give it run permissions.

31. Open your browser and access the URL:

    `http://web2.tecnico.ulisboa.pt/istxxxxxx/test.cgi`

    where **`istxxxxxx`** is your Fénix username.

32. Confirm the page shows up without any errors.

    Note: If the page is not properly shown in the address, this is most likely due to incorrect *username* or *password* in *test.cgi*.

[4] https://www.sublimetext.com

[5] https://code.visualstudio.com

33. Open test.cgi in an editor and locate the calls to the following functions:

    - `psycopg2.connect(…)`
    - `connection.cursor()`
    - `cursor.execute(…)`
    - `cursor.fetchall()`
    - `cursor.close()`
    - `connection.close()`

    This is usually the sequence of calls to interact with a database using Python with the **psycopg** module.

34. In file `test.cgi`, find the database query:

    ```
    SELECT * FROM account;
    ```

    Compare the result of this query, show on the browser, with the result you got previously through the command line.

35. In file `test.cgi`, locate the place where an HTML table is built to show the results:

    - <**table**> and </**table**> define the beginning and the end of the HTML table
    - <**tr**> and </**tr**> define the beginning and the end of a line in the HTML table
    - <**td**> and </**td**> define the beginning and the end of a cell in the HTML table
    - This is the usual way of building tables in HTML. Notice that the table is built within a loop that goes through the results of **`cursor.fetchall`**()

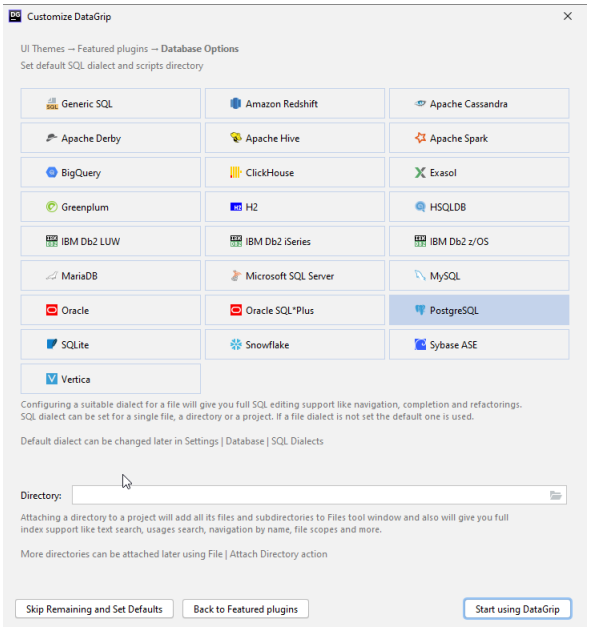
36. In the browser, open the source-code of the page and compare that HTML code with the Python code inside the script file `test.cgi`.

Installing and configuring DataGrip

37. Install the most recent version of **DataGrip**:

- On **Windows**: go **to https://www.jetbrains.com/datagrip**, click on download, and download the installer for Windows. Double click on the executable and follow the instructions.
- On **Mac**: go to **https://www.jetbrains.com/datagrip**, click on download, and download the .dmg for Mac, double click on the file, and drag to the Applications folder.
- On **Linux**: if you have **Ubuntu** as operative system, or have the **snapd** package installed, you can install by running a single command: **sudo snap install datagrip --classic**. Else, go to **https://www.jetbrains.com/datagrip**, click on download, and download the **.tar.gz** for Linux. Extract the file with **tar xvzf [package_name].tar.gz**. Navigate to the package directory and follow the instructions in the file **Install-Linux-tar.txt**.

After opening the program and prompted to configure it, pick PostgreSQL as the default SQL dialect.

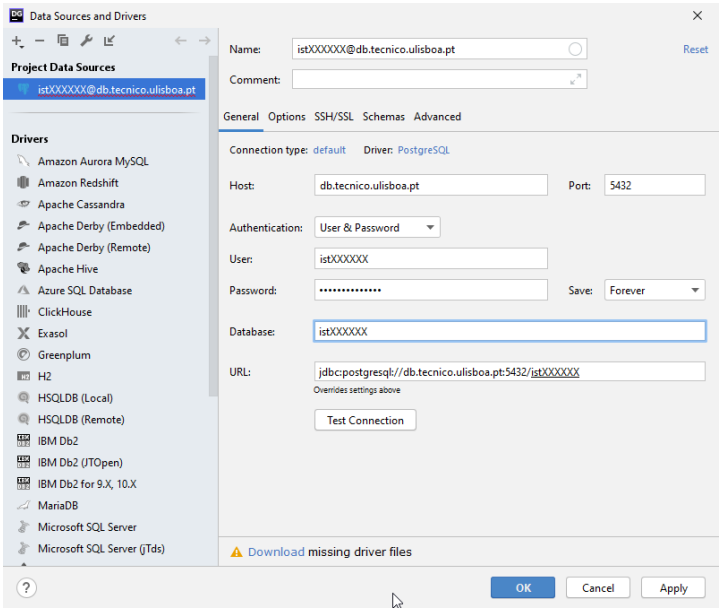


You may be prompted to login to your Jetbrains account. If you do, simply login to your Jetbrains account.

If you're not prompted, go to **Help > Registe**r and login with your account, that you should have previously associated to your Técnico e-mail address.

38. In Linux, Windows or Mac OS open **DataGrip**, if you haven't already. Create a new Data Source connection using **PostgreSQL** driver.

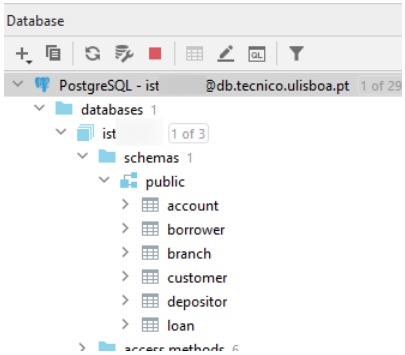
    - Go to **File > New > Data Source > PostgreSQL** …
    - When prompted, enter:
        i. **Host name** = `db.tecnico.ulisboa.pt`
        ii. **Port** = 5432
        iii. ***User*** and ***Password*** your istXXXX username and the database password (⚠ this is not the Fénix password!)
        iv. In database introduce your istXXXX username again



    - If there is an indication to download missing drivers, click on it before proceeding. Test the connection, if successful, click OK.
    - Click **Test Connection**

39. Click on Refresh.

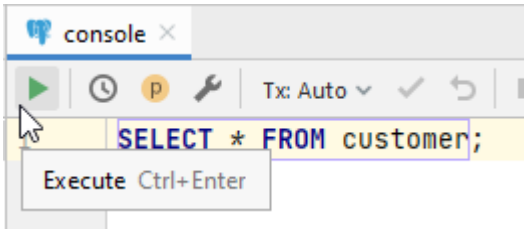
40. Expand the connection tree **istXXXXXX > schemas > public**, where you can see the tables of the example database you previously created.



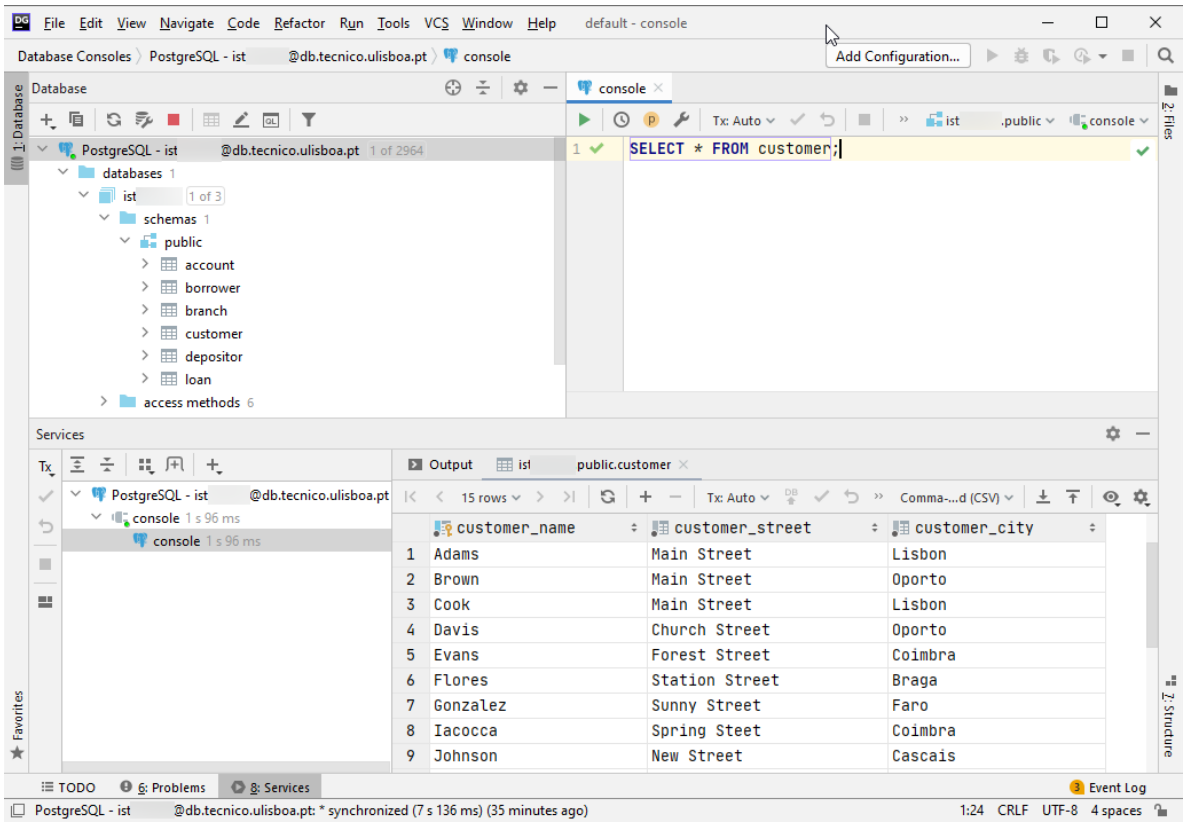
41. Open an SQL console by clicking on the Source Editor button

42. Type in and execute a query to get the complete list of customers:

SELECT * FROM customer;



43. You will see the result of the query at the bottom


File Edit View Navigate Code Refactor Run Tools VCS Window Help default - console
Database Consoles PostgreSQL - ist @db.tecnico.ulisboa.pt console
Add Configuration...
Database
console
PostgreSQL - ist @db.tecnico.ulisboa.pt 1 of 2964
databases 1
ist 1 of 3
schemas 1
public
account
borrower
branch
customer
depositor
loan
access methods 6
SELECT * FROM customer;
Services
PostgreSQL - ist @db.tecnico.ulisboa.pt
console 1 s 96 ms
console 1 s 96 ms
Output ist public.customer
15 rows
customer_name customer_street customer_city
1 Adams Main Street Lisbon
2 Brown Main Street Oporto
3 Cook Main Street Lisbon
4 Davis Church Street Oporto
5 Evans Forest Street Coimbra
6 Flores Station Street Braga
7 Gonzalez Sunny Street Faro
8 Iacocca Spring Steet Coimbra
9 Johnson New Street Cascais
TODO 6: Problems 8: Services Event Log
PostgreSQL - ist @db.tecnico.ulisboa.pt: * synchronized (7 s 136 ms) (35 minutes ago)
1:24 CRLF UTF-8 4 spaces