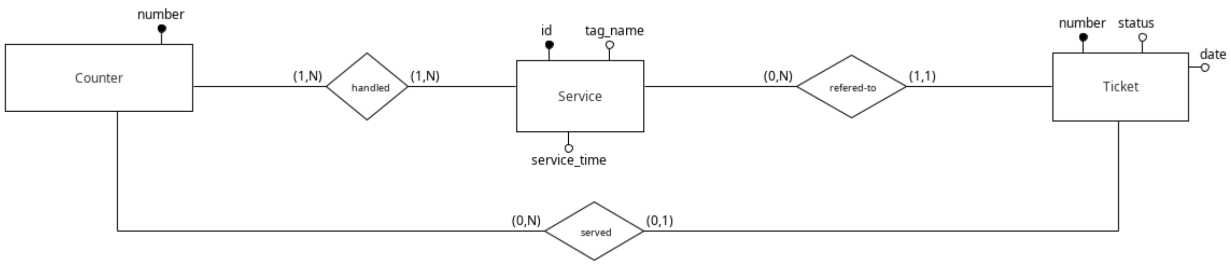
Office Queue Management System

Conceptual design

In this phase, the entity-relationship model of the system was developed, where it was necessary to represent all the entities involved and the relationships required between them. Specifically, in this sprint, the system required the presence of two particular types of users, namely, Customer and Officer, each of whom interacts with the system differently. Furthermore, customer registration was deemed redundant as it was considered "anonymous," as it is closely tied to the ticket entity, and thus, only the ticket would be sufficient to represent the action associated with a user. We can tell the same for the officer, because each counter is managed by an officer so it would be implicit that counter has an officer (it’s not required, in this moment, to access as an officer).



Logical design

Translation of entities and associations

- Counter(**number**)
- Handled(***counter_number***, ***service_id***)
- Service(**id**, tag_name, service_time)
- Ticket(**number**, status, date, *service_id*, *counter_number**)

attribute: in bold => primary key

attribute: in italics => foreign key

attribute*: * => it can be null

Referential constraints

- Handled(counter_number) ⊆ Counter(number)
- Handled(service_id) ⊆ Service(id)
- Ticket(service_id) ⊆ Service(id)
- Ticket(counter_number) ⊆ Counter(number)

Counter

Attribute	Tipology	Description
number	Integer	The number attribute describes the number that is assigned to that particular counter, and being a primary key is unique to each counter.

Handled

This particular table traces all the correspondences between service and counter, that is identifies for every counter all the services that can manage.

Attribute	Tipology	Description
counter_number	Integer	Reference to the counter number
service_id	Integer	Reference to the service id

Service

Attribute	Tipology	Description
id	Integer	Unique indentifier for a service
tag_name	String	Descriptive name for that service
service_time	Integer	Indicates in minutes how approximately you need to complete this service

Ticket

Contains all tickets generated by the system. There is a particular field called "status" that has the responsibility to track the current status of the ticket that can take one of the following values:

- 0: waiting
- 1: being served
- 2: served

This field is very important for the implementation of the queue, because it would be enough to update this state in order to update the queue for that service.

Attribute	**Tipology**	**Description**	**Can be Null?**
number	Integer	Number assigned to that ticket	F
status	Integer	Status of the ticket	F
date	Timestamp	When the ticket was generated	F
service_id	Integer	Service id reference	F
counter_number	Integer	Number of the counter that served the ticket	T (is valued when associated with a counter)

Installation

For installation, it is recommended to use the latest version of PostgreSQL. A highly recommended tool for advanced and functional database management is DBeaver. This tool provides a wide range of features for working with the respective database.

How to install

It is necessary to change the password of the "postgres" user to establish a successful connection between the JavaScript backend and the database on your machine. The user should be configured with the following credentials:

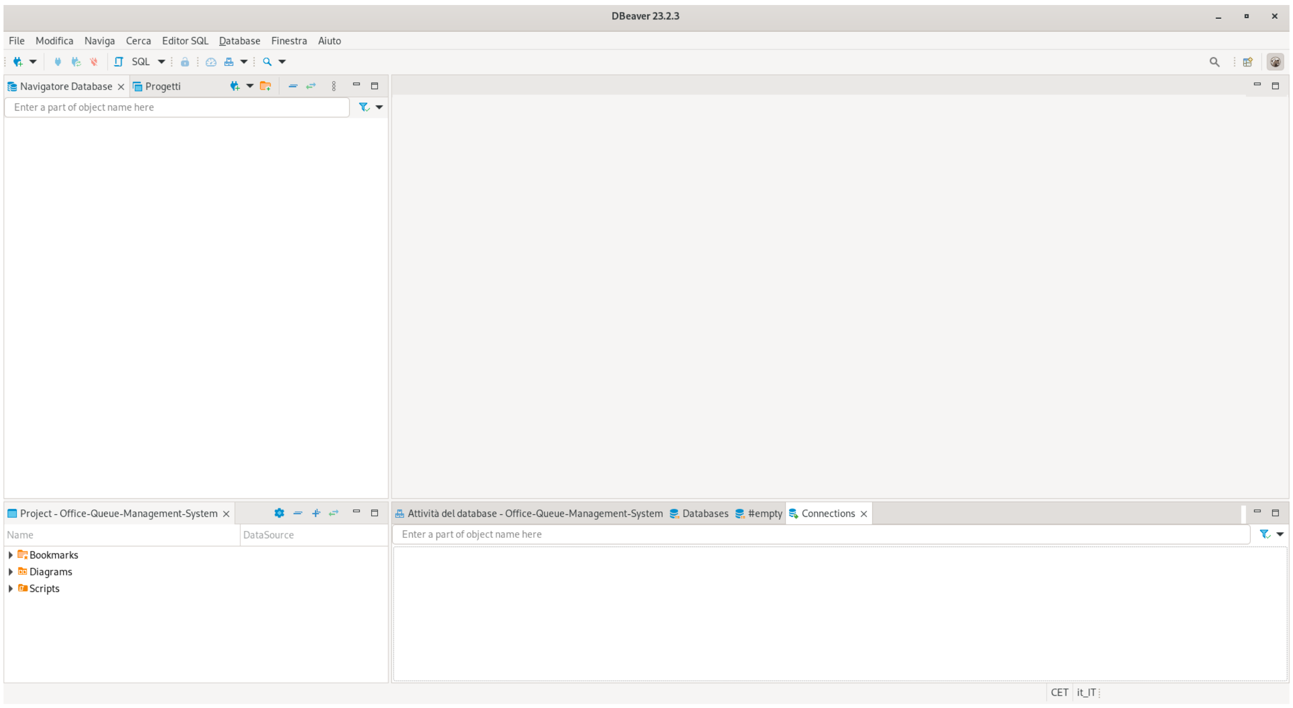
- User: "postgres"
- Password: "group15-postgres"

Once you have completed these steps, you are ready to import the database on your machine. There are two methods to import the dump file

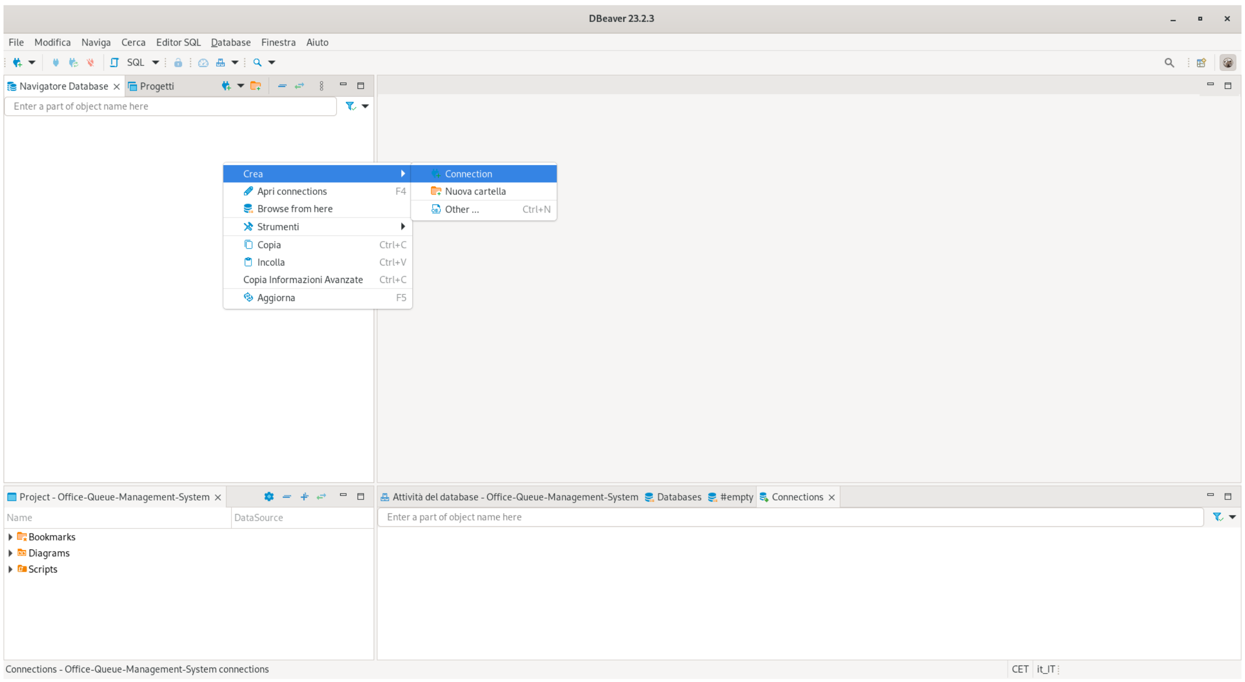
- Import it with Dbeaver
- Import it from terminal

Dbeaver Method

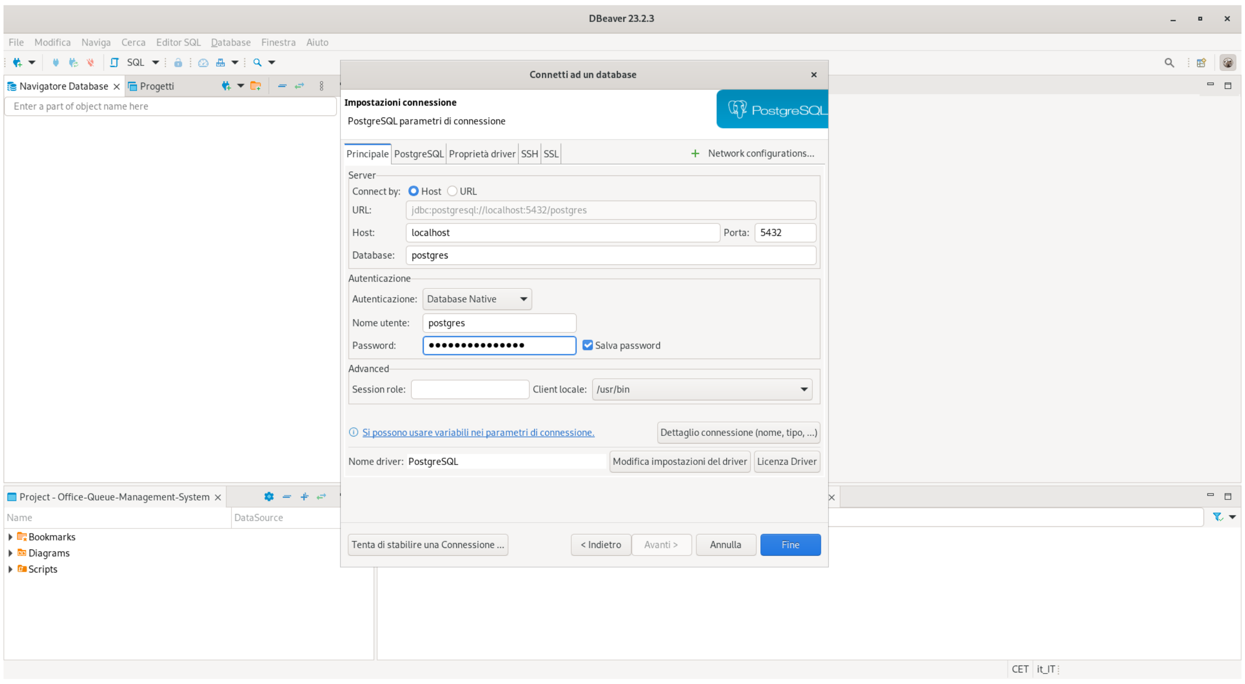
1. Open Dbeaver


DBeaver 23.2.3
File Modifica Naviga Cerca Editor SQL Database Finestra Aiuto
Navigatore Database
Progetti
Enter a part of object name here
Project - Office-Queue-Management-System
Name
DataSource
Bookmarks
Diagrams
Scripts
Attività del database - Office-Queue-Management-System
Databases
#empty
Connections
CET
it_IT

2. Create a new connection, choose PostgreSQL and click on Next


DBeaver 23.2.3
File Modifica Naviga Cerca Editor SQL Database Finestra Aiuto
Navigatore Database
Progetti
Enter a part of object name here
Crea
Apri connections F4
Browse from here
Strumenti
Copia Ctrl+C
Incolla Ctrl+V
Copia Informazioni Avanzate Ctrl+C
Aggiorna F5
Connection
Nuova cartella
Other ... Ctrl+N
Project - Office-Queue-Management-System
Name
DataSource
Bookmarks
Diagrams
Scripts
Attività del database - Office-Queue-Management-System
Databases
#empty
Connections
Connections - Office-Queue-Management-System connections
CET
it_IT

3. Insert “postgres” as username and “group15-postgres” as password (important: modify the password of the user “postgres” to “group15-postgres” if not previously done) and click on “End”


DBeaver 23.2.3
File Modifica Naviga Cerca Editor SQL Database Finestra Aiuto
Navigatore Database
Progetti
Connetti ad un database
Principale PostgreSQL Proprietà driver SSH SSL
Network configurations...
Si possono usare variabili nei parametri di connessione.
Dettaglio connessione (nome, tipo, ...)
Modifica impostazioni del driver
Licenza Driver
Tenta di stabilire una Connessione ...
< Indietro
Avanti >
Annulla
Fine
CET it_IT

Impostazioni connessione

PostgreSQL parametri di connessione

Server

Connect by: [✓ Host] URL

URL: jdbc:postgresql://localhost:5432/postgres

Host: localhost Porta: 5432

Database: postgres

Autenticazione

Autenticazione: Database Native

Nome utente: postgres

Password: ●●●●●●●●●●●●●●● [✓ Salva password]

Advanced

Session role:

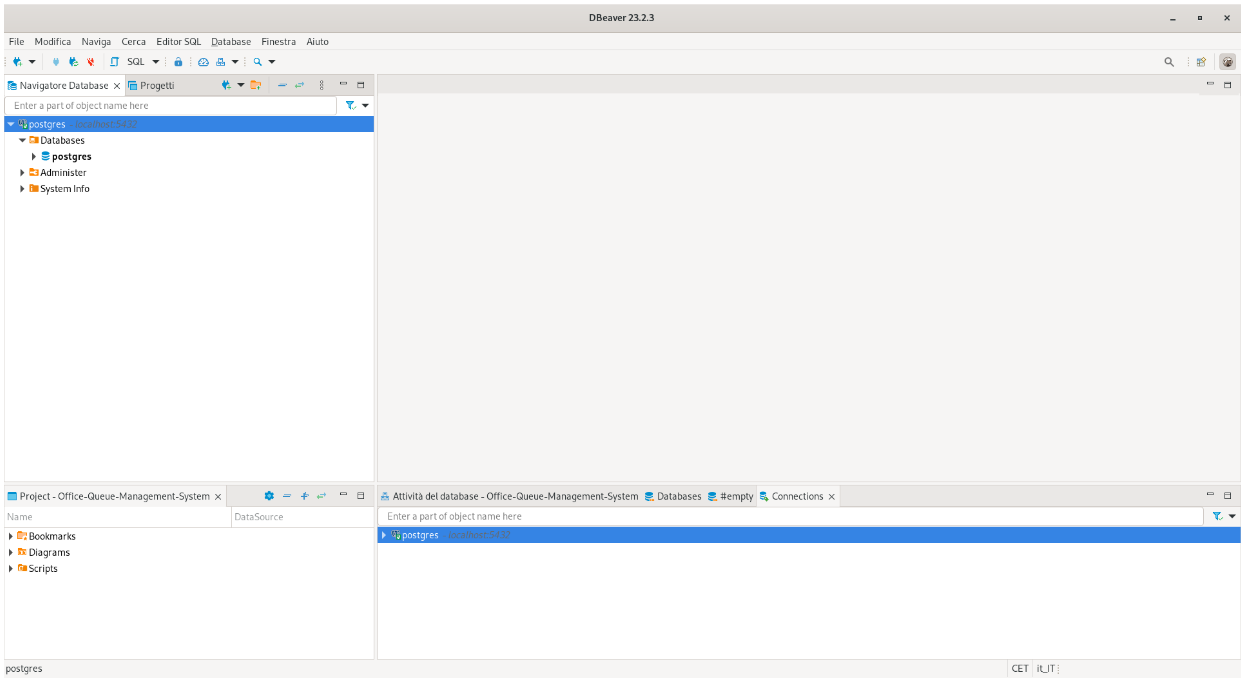
Client locale: /usr/bin

Nome driver: PostgreSQL

Project - Office-Queue-Management-System

Name	DataSource
Bookmarks	
Diagrams	
Scripts	

4. Will displayed this page


DBeaver 23.2.3
File Modifica Naviga Cerca Editor SQL Database Finestra Aiuto
Navigatore Database
Progetti
Enter a part of object name here
Attività del database - Office-Queue-Management-System
Databases
#empty
Connections
CET it_IT

- postgres localhost:5432
  - Databases
    - postgres
  - Administer
  - System Info

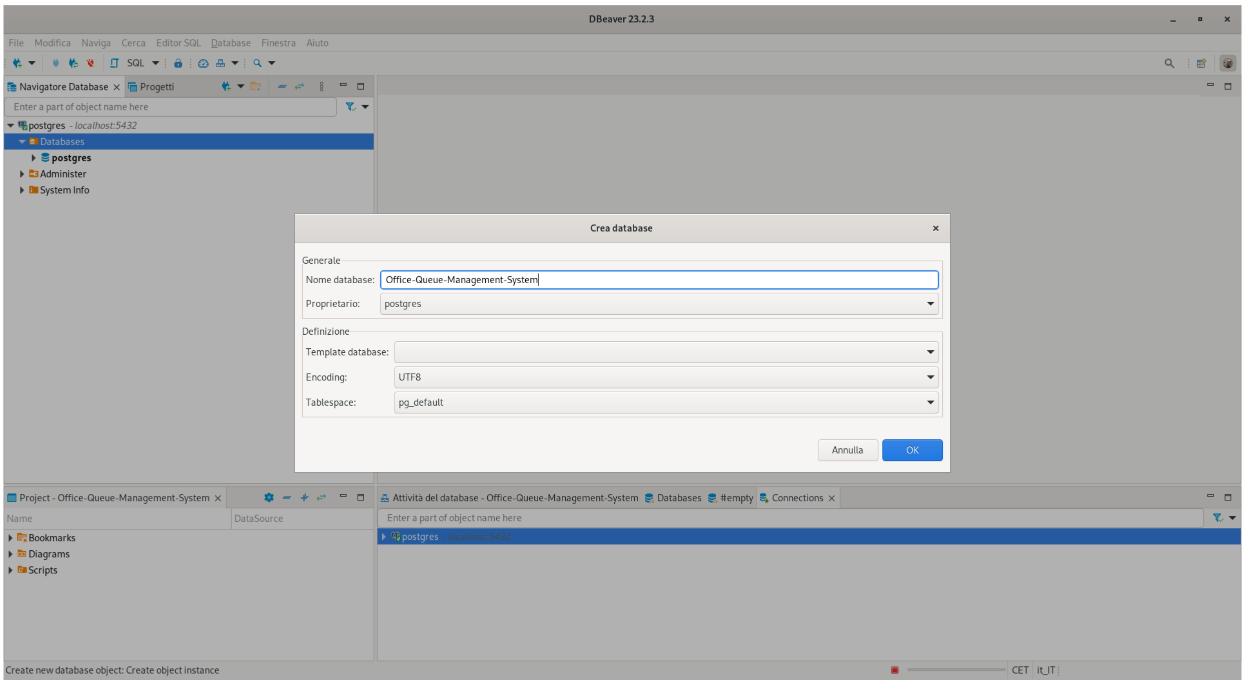
Project - Office-Queue-Management-System

Name	DataSource
Bookmarks	
Diagrams	
Scripts	

- postgres localhost:5432

postgres

5. Click on Databases, click on Create a new Database and call it “Office-Queue-Management-System”.


Crea database

Generale

Nome database: Office-Queue-Management-System

Proprietario: postgres

Definizione

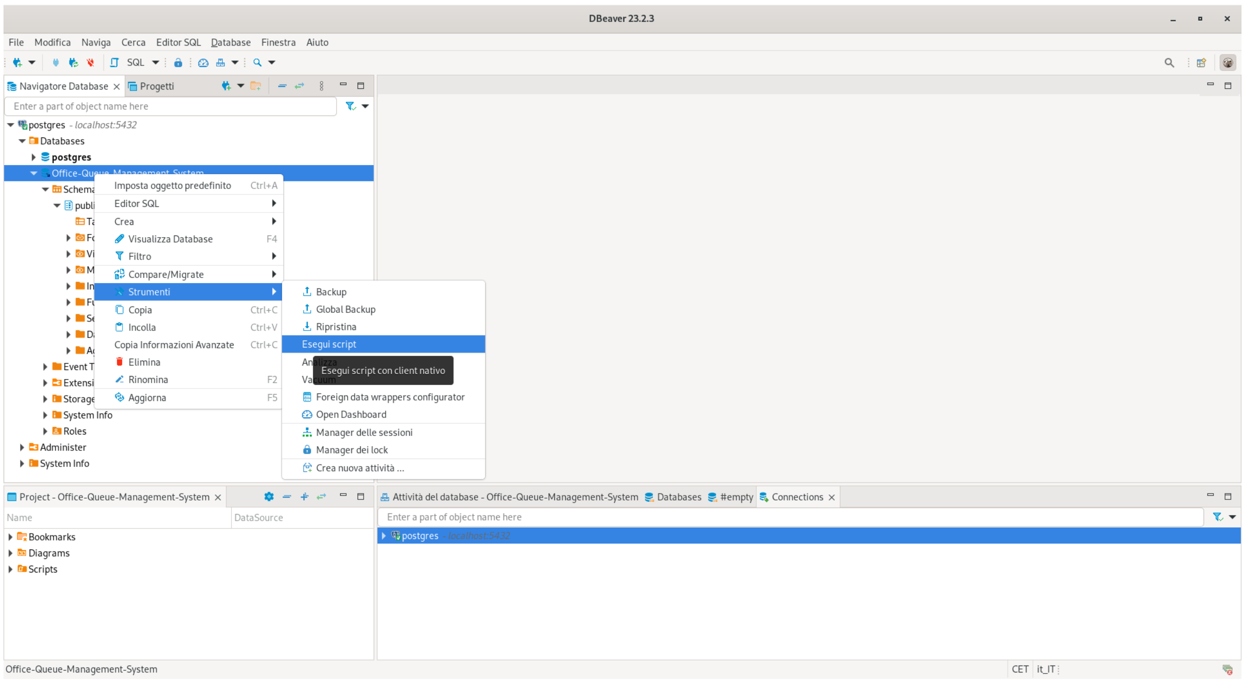
Template database:

Encoding: UTF8

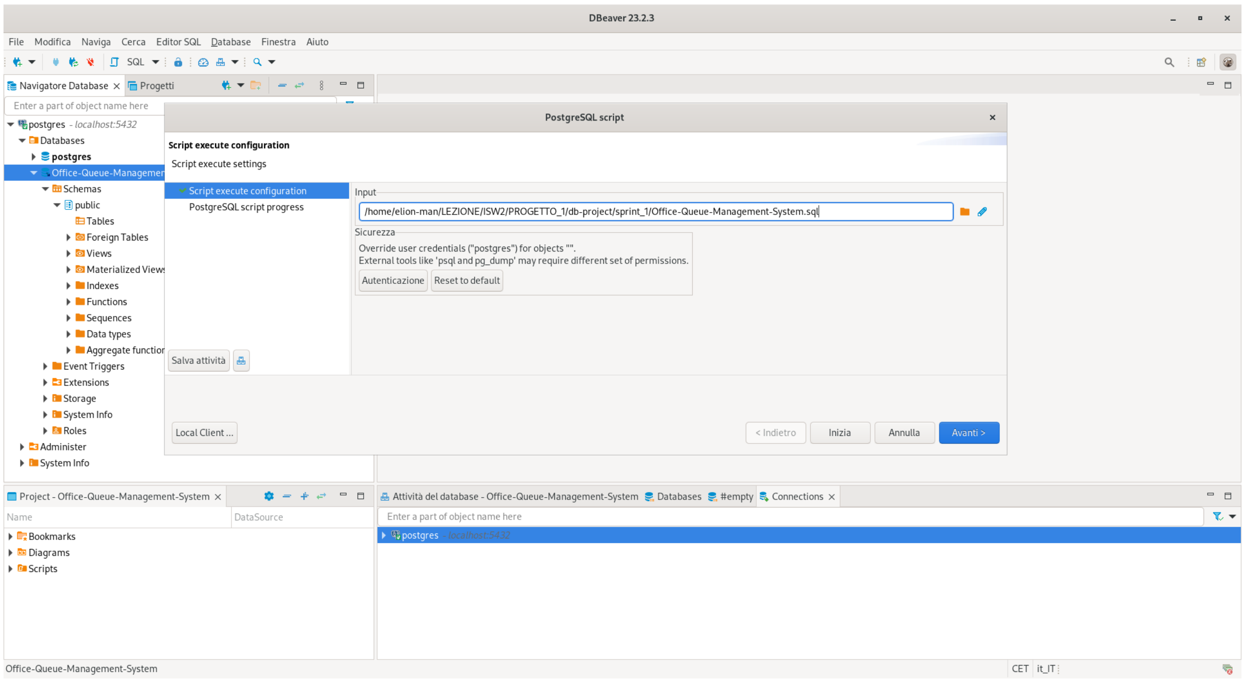
Tablespace: pg_default

DBeaver 23.2.3
File Modifica Naviga Cerca Editor SQL Database Finestra Aiuto
Navigatore Database
Progetti
Enter a part of object name here
postgres - localhost:5432
Databases
postgres
Administer
System Info
Annulla
OK
Project - Office-Queue-Management-System
Name
DataSource
Bookmarks
Diagrams
Scripts
Attività del database - Office-Queue-Management-System
Databases
#empty
Connections
Create new database object: Create object instance
CET it_IT

6. Once created the database, we have to run the sql file. So right click on the created database, click on tools and finally run scripts.



7. Insert as input the path of the file “Office-Queue-Management-System.sql” in the Input field requested. And finally, click on Start.



PostgreSQL script

Script execute configuration

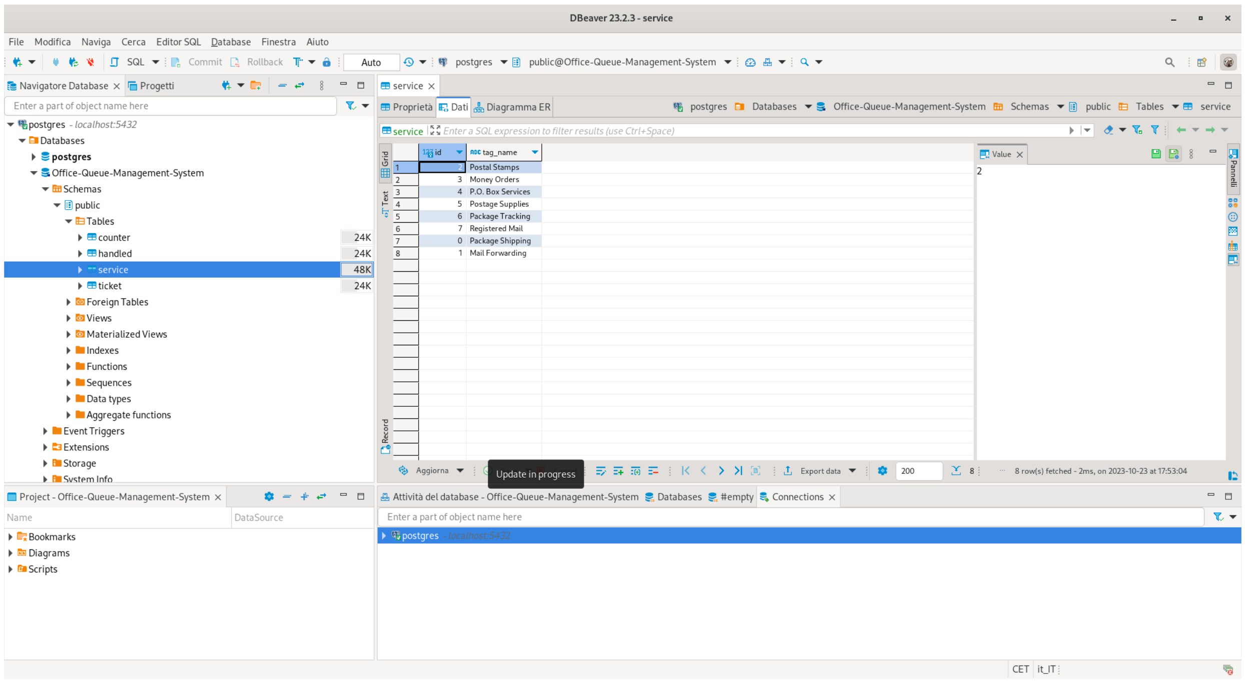
Script execute settings

Input: /home/elion-man/LEZIONE/ISW2/PROGETTO_1/db-project/sprint_1/Office-Queue-Management-System.sql

Sicurezza

Override user credentials ("postgres") for objects "".
External tools like 'psql and pg_dump' may require different set of permissions.

8. If at the end of the execution an error returns "Task execution failed, reason: ERROR: you cannot delete the currently opened database." you can ignore it. The database has been successfully loaded:



	id	tag_name
1	2	Postal Stamps
2	3	Money Orders
3	4	P.O. Box Services
4	5	Postage Supplies
5	6	Package Tracking
6	7	Registered Mail
7	0	Package Shipping
8	1	Mail Forwarding

8 row(s) fetched - 2ms, on 2023-10-23 at 17:53:04

Terminal Method

To perform these steps, you need to have PostgreSQL installed on your computer to use the provided commands. Additionally, please note that these steps have only been tested on Linux (Debian). In the same repository folder, a file named 'load_db.sh' has been uploaded, which, when executed, allows you to load the database automatically. You will only need to enter the password "group15-postgres" when prompted.

If the file is not in executable form, type the command “sudo chmod 744 load_db.sh” to make it executable.

To run the file, you only need to execute the following command: “./load_db.sh”.

Database Access via Terminal

To access the database via the command line, we use the “psql -U postgres -W” command, and it will prompt us to enter the password. Once we are granted access, we can type the command “\l” to view the list of all databases in the system, including “Office-Queue-Management-System”.

Using the command '\c Office-Queue-Management-System', we can select the desired database (this may prompt us to enter the password again). With the command '\d', we can retrieve a list of all available tables.

Alternatively, we can log out using “\q” and then enter the command “psql -d Office-Queue-Management-System -U postgres -W”, where, after entering the correct password, we will be directly logged into the specified database.