BD(AI) Tutorial

Using PG and python for images

Author: Pedro Furtado, professor at University of Coimbra, Portugal.

What to load after class:

Show the steps that you did. Include every command/query/statement that you had to ran in any tool to do the work. That way you will show that you did it. You can also include screenshots of results when graphically.

Objective:

In this tutorial we learn how to connect to the database and how to show images from the database.

Software

PostgreSQL (latest) – http://www.postgresql.org/download/
pgAdmin– https://www.pgadmin.org/

Anaconda -> Jupyter notebook, or jupyter notebook only

What are the software parts involved?

PostgresSQL (PG) is a relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

PGadmin ia an open source administration and management tools for the PostgreSQL database. Includes a graphical administration interface, an SQL query tool, a procedural code editor, ERD tool, Schema comparison, and much more. pgAdmin is designed to answer the needs of most users, from writing simple SQL queries to developing complex databases. Runs as a desktop application on most common operating systems, or can be deployed as a multi-user web application on a server or in a container.

Jupyter notebooks is an interactive python development environment where you add code to cells, run cells and get results. It allows you to evolve your work step-by-step as you also see the results.

(EXTRA, OPTIONAL, OUT-OF-CLASS) Tableau is a well-known OLAP data analysis tool. It allows you to analyze your dataset with a no-code fast productive approach.

1. Ensure you have Anaconda (or jupyter notebooks) and postgres installed

If not, just search for “postgres download” and “anaconda download” (or jupyter notebooks if you prefer) in google and install those. You may have difficulties in certain windows versions, let’s try our best against the beast!

2. In case you did not yet, get three pictures of rooms in apartments and store them in your disk somewhere you can find

Also, get the path where that is (go to the folder where you have the files, then look at the path header in the explorer window, or if in mac, right-click and see information and copy the path from there (ask the teacher for help).

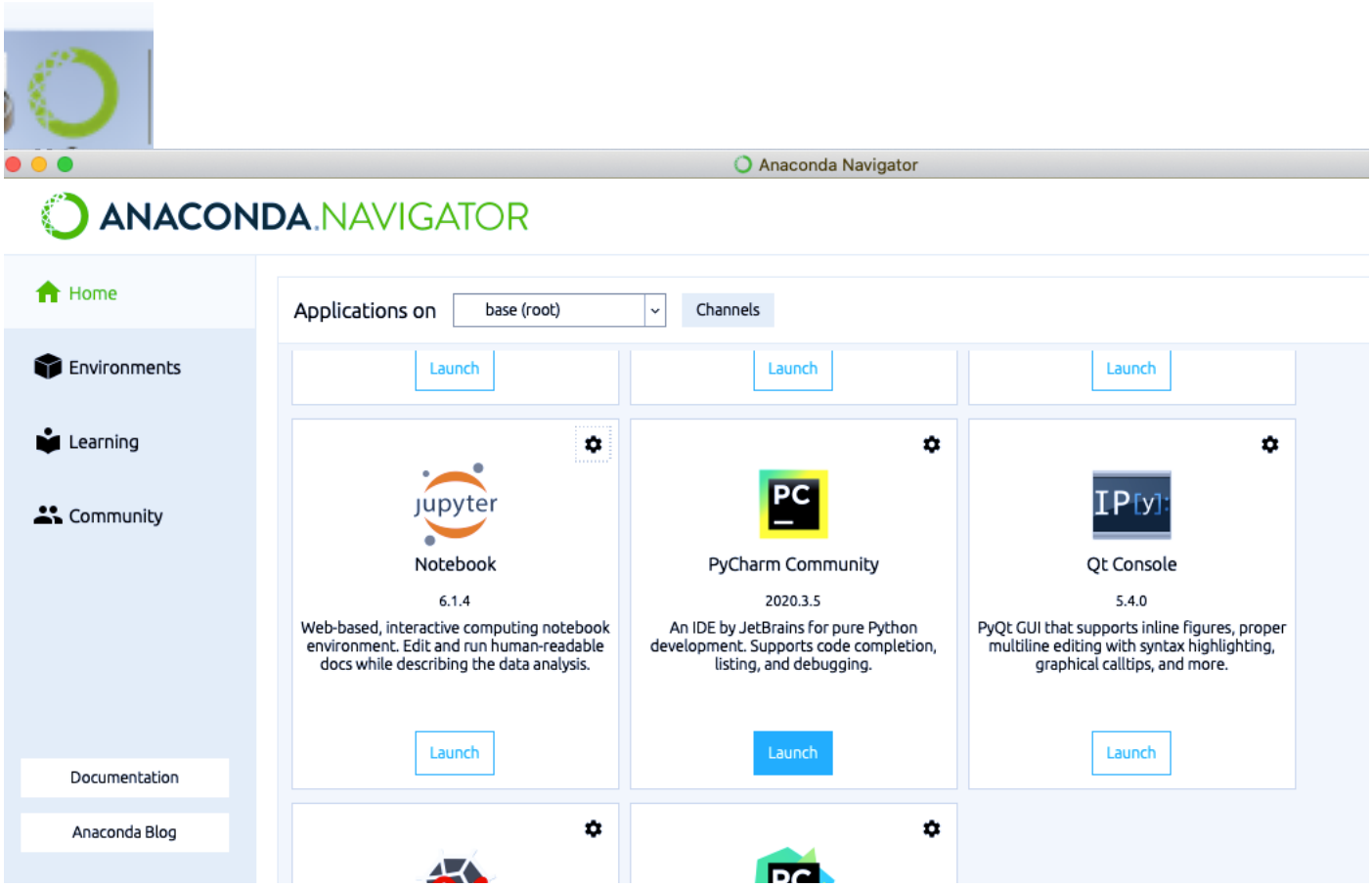
3. Make sure you have imoveis installed and with data in it

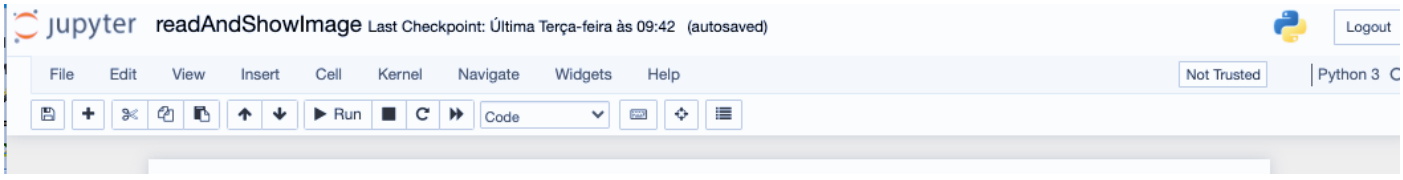
In case you don't…but do not forget to change the paths in the inserts and to have the pictures in the correct folders you point at using the paths…

```
CREATE TABLE public.imovel
(
    ref integer NOT NULL DEFAULT nextval('imovel_ref_seq'::regclass),
    type character varying(20) COLLATE pg_catalog."default",
    area numeric(8,2),
    nwc integer,
    address character varying(512) COLLATE pg_catalog."default",
    zipcode character(8) COLLATE pg_catalog."default",
    location character varying(512) COLLATE pg_catalog."default",
    place character varying(200) COLLATE pg_catalog."default",
    nif numeric(12,0),
    CONSTRAINT imovel_pkey PRIMARY KEY (ref)
)

CREATE TABLE public.photo
(
    id integer NOT NULL DEFAULT nextval('photo_id_seq'::regclass),
    photo character varying(4096) COLLATE pg_catalog."default",
    ref integer,
    CONSTRAINT photo_pkey PRIMARY KEY (id)
)


CREATE TABLE public.proprietario
(
    nif numeric(12,0) NOT NULL,
    lname character varying(100) COLLATE pg_catalog."default",
    fname character varying(100) COLLATE pg_catalog."default",
    tel numeric(9,0),
    countrycode character(4) COLLATE pg_catalog."default",
    email character varying(100) COLLATE pg_catalog."default",
    address character varying(512) COLLATE pg_catalog."default",
    CONSTRAINT proprietario_pkey PRIMARY KEY (nif)
)


insert into imovel values
(default,'T3',120,2,'Rua da Alegria','3000-242','Coimbra','Coimbra',1),
(default,'T2',100,1,'Rua da Contente','3000-252','Coimbra','Coimbra',1),
(default,'T4',140,1,'Rua da Tristeza','4000-252','Lisboa','Lisboa',2);

delete from proprietario;
insert into proprietario values
(1, 'Pedro','Correia',999999999,'+351','a@gmail.com','Rua 1'),
(2, 'Pedro','Moreira',111111111,'+351','m@gmail.com','Rua 2');


insert into photo values
(default,'/Users/pedro/Documents/Aulas/BD/BD2023 - 2024/pratica/aula 4_joins/Small_Apartment.jpeg',4),
```

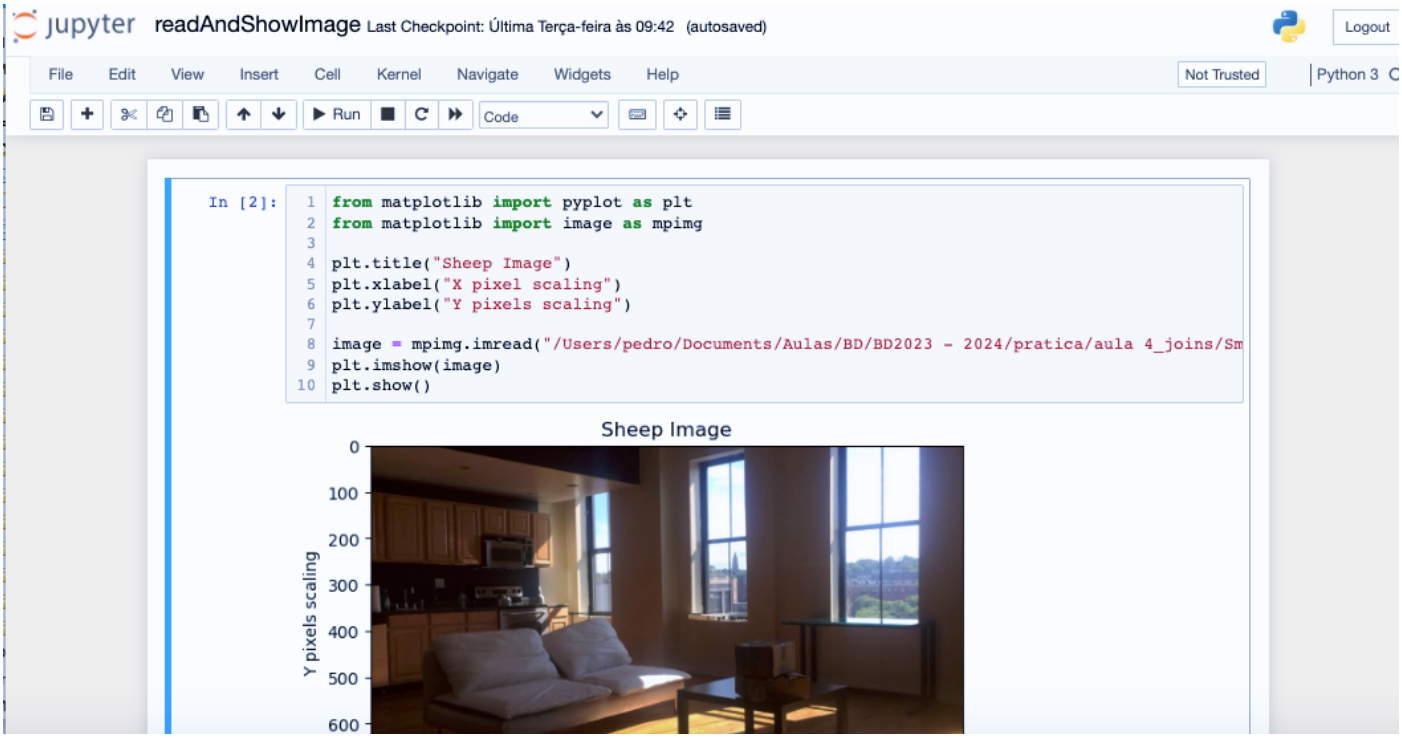
(default,'/Users/pedro/Documents/Aulas/BD/BD2023 - 2024/pratica/aula 4_joins/bathroom.jpeg',4),
(default,'/Users/pedro/Documents/Aulas/BD/BD2023 - 2024/pratica/aula 4_joins/bathroom2.jpeg',5);

4. Create a jupyter notebook code…



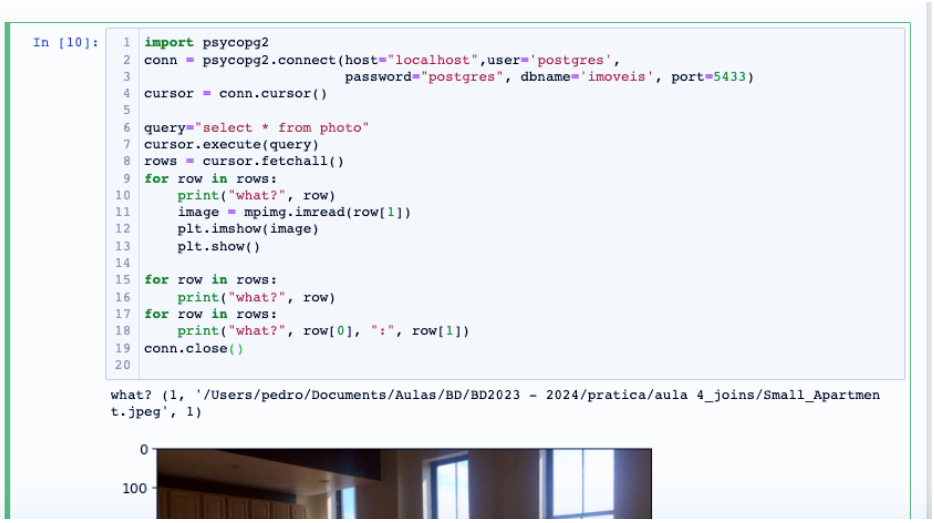


5. Show the image:




```
from matplotlib import pyplot as plt
from matplotlib import image as mpimg

plt.title("Sheep Image")
plt.xlabel("X pixel scaling")
plt.ylabel("Y pixels scaling")

image = mpimg.imread("/Users/pedro/Documents/Aulas/BD/BD2023 - 2024/pratica/aula 4_joins/Small_Apartment.jpeg")
plt.imshow(image)
plt.show()
```

6. Show the images that are in the database stored as a paths:




```
import psycopg2
conn = psycopg2.connect(host="localhost",user='postgres',
                        password="postgres", dbname='imoveis', port=5433)
```

```
cursor = conn.cursor()

query="select * from photo"
cursor.execute(query)
rows = cursor.fetchall()
for row in rows:
    print("what?", row)
    image = mpimg.imread(row[1])
    plt.imshow(image)
    plt.show()

for row in rows:
    print("what?", row)
for row in rows:
    print("what?", row[0], ":", row[1])
conn.close()
```

7. Now describe what each line of code is doing…

THE END