Abstract

PostgreSQL has been touted as the "world's most advance open source database", and offers various features that make it stand out from other SQL databases such as MySQL and MariaDB. Although users might not immediately make use of those features, it is good to start with a database that has those features available if needed in the future. With a database chosen, the next step is to fill the database with data. It doesn't really matter if you choose a database first or the data first, what is important is that you choose the tool that gets the job done. Scraping data from a website is a way to take information available online, and then either manipulate it immediately, or store it in a database for future functionality.

In this project, the data will come from an online shopping website. Depending on how much data you want, and the goals of your project, you can choose to scrape all the data available, or just a small subset to make the job quicker. Here the goal will be to scrape all the physical attributes of models, and then put those attributes into a PostgreSQL database, using Python.

Introduction

With the development of the Internet, network data covers various fields, but with the increasing amount of network data and diversified data formats, it becomes more and more difficult for users to obtain valuable data from massive data. At present, data collection technology has been studied at home and abroad, and has found that network resources can be automatically obtained through network crawler technology(1).

Web scraping is the process of creating a computer program to download, parse, and organize data from the web in an automated manner(2).

Before scraping you need to consider and check some features. If there's no bulk download available, check to see whether the website has an application programming interface (API). An API lets software interact with a website's data directly, rather than requesting the HTML. This can be much less burdensome than scraping individual web pages, but there might be a fee associated with API access (see, for example, Google's Map API). Some websites don't make their data available directly in the HTML and might require some more advanced techniques. Other websites include protections like captchas and anti-denial-of-service (DoS) measures that can make scraping difficult. A few websites simply don't want to be scraped and are built to discourage it. Be sure to check for licensing or copyright restrictions on the extracted data. You might be able to use what you scrape, but it's worth checking that you can also legally share it. Ideally, the website content license will be readily available.(3)

The most popular libraries used by web scraping developers in python are Beautifulsoup, Scrapy, and Selenium but every library has its own pros and cons as you can see in Figure1.

BeautifulSoup is really a beautiful tool for web scrappers because of its core features. It can help the programmer to quickly extract the data from a certain web page. This library will help us to pull the data out of HTML and XML files. But the problem with Beautiful Soup is it can't able to do the entire job on its own. this library requires specific modules to work done.

you should need to remember that Selenium is designed to automate test for Web Applications. It provides a way for the developer to write tests in a number of popular programming languages such as C#, Java, Python, Ruby, etc. This framework is developed to perform browser automation. Let's have a look at the sample code that automates the browser.

Scrapy is an open source collaborative framework for extracting the data from the websites what we need. Its performance is ridiculously fast and it is one of the most

powerful libraries available out there. One of the key advantages of scrapy is that it is built on top of Twisted, an asynchronous networking framework, that means scrapy uses the non-blocking mechanism while sending the requests to the users.

Figure1. Compare web scraping methods

In this article, a number of tools for crawling websites are presented, and an example using online shopping website has been adopted in order to specifically show how these can be extracted from a http platform. For this purpose, Python with the library “Scrapy” is used. Other program packages include “Response” and “psycopg2”, with which more complex applications can be realized. In addition to the technical aspects of web scraping, the legal framework of this process will also be discussed.

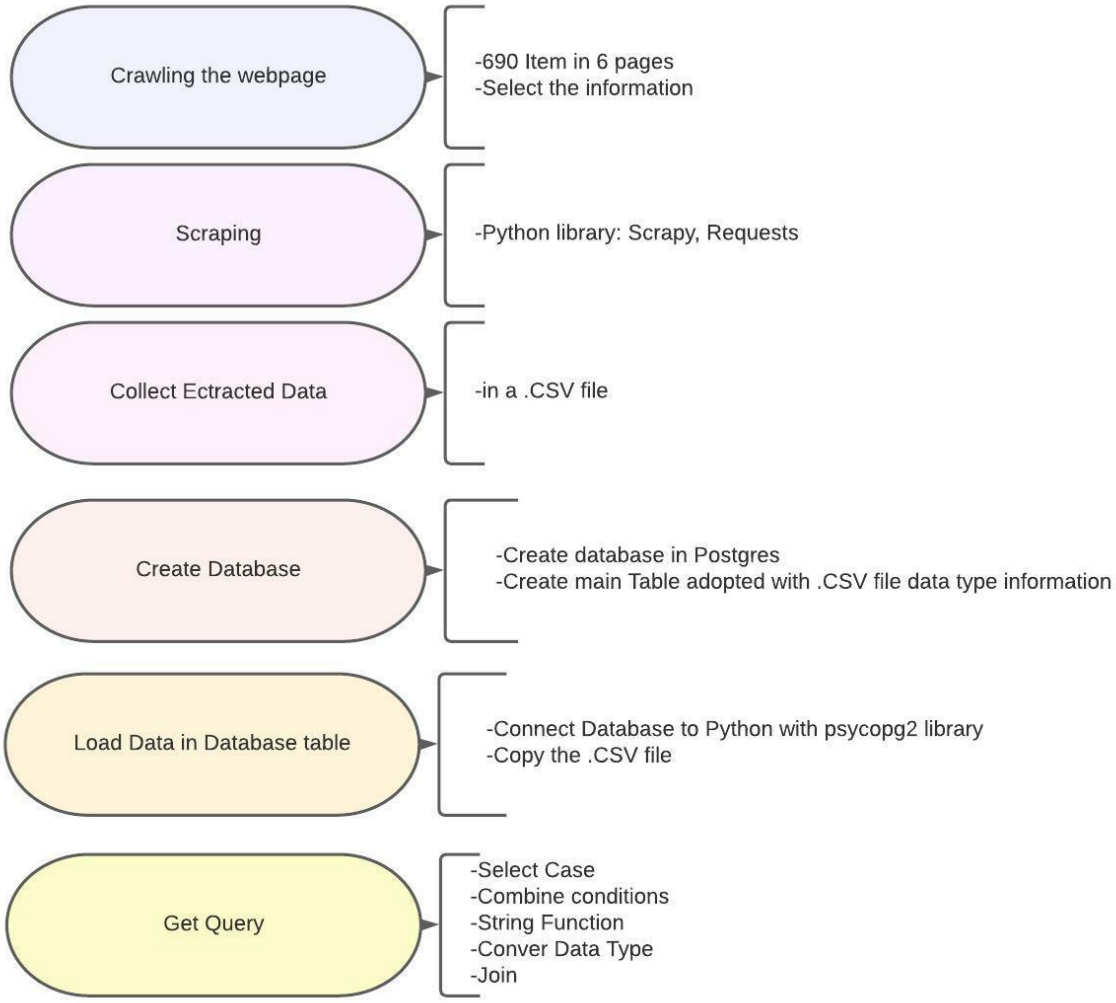


Figure2.Overview of project

Course Relevance:

1- Connect Database to Python:

Connecting to database postgres psql is an interactive terminal program provided by PostgreSQL. It allows you to interact with the PostgreSQL database server such as executing SQL statements and managing database objects. connect to the PostgreSQL database server via the following tools:

- psql – a terminal-based front-end to PostgreSQL database server.
- pgAdmin – a web-based front-end to PostgreSQL database server.

we need to understand what it requires to connect to the database. The basic parameters on any of the platforms are as follows:

Server [localhost]: This is the address for the server. You can use an IP address or the hostname of the machine on which database server is running. If you do not give any value, by default it is localhost.

Database [postgres]: The name of the database with which you want to connect. The default name of the database is the same as that of the user. (If you are using Advanced Server, the default name would be edb.)

Port [5432]: This is the port on which you have configured your instance while installing or initializing. The default port is 5432. (If you are using Advanced Server this would be 5444.)

Username [postgres]: This is the username that is created while the installation takes place. The default username for postgres is postgres. (If you are using Advanced Server it is enterprisedb.)

On a Mac or Windows, you are able to connect to the default instance by simply hitting enter at the shell or command prompt when trying to run psql and keying in the password. we will use the library psycopg2 to connect the database to Python. The syntax would be:

```
conn = psycopg2.connect(
                        host="localhost",
                        database="suppliers",
                         user="postgres",
                        password="Abcd1234")
```

2- Join

A **join** clause in SQL – corresponding to a join operation in relational algebra – combines columns from one or more tables into a new table. ANSI-standard SQL specifies five types of JOIN: INNER, LEFT OUTER, RIGHT OUTER, FULL OUTER and CROSS.

Here are the different types of the JOINs in SQL:

- `(INNER) JOIN`: Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN`: Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN`: Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN`: Returns all records when there is a match in either left or right table

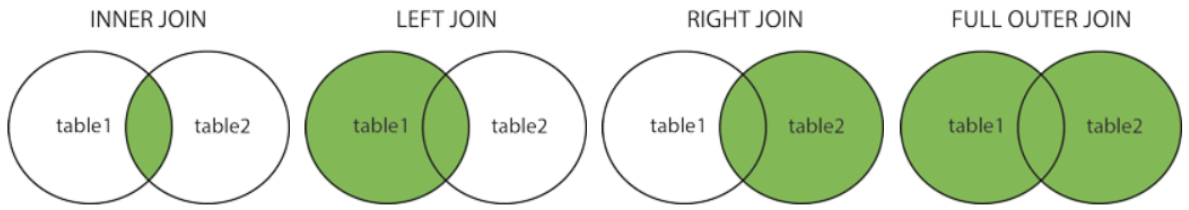


Figure3. Join Clause

Thus far, our queries have only accessed one table at a time. Queries can access multiple tables at once, or access the same table in such a way that multiple rows of the table are being processed at the same time. Queries that access multiple tables (or multiple instances of the same table) at one time are called *join* queries. They combine rows from one table with rows from a second table, with an expression specifying which rows are

to be paired. For example, to return all the weather records together with the location of the associated city, the database needs to compare the city column of each row of the weather table with the name column of all rows in the cities table, and select the pairs of rows where these values match.

3- **Attribute Constraint**

constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY - Prevents actions that would destroy links between tables
- CHECK - Ensures that the values in a column satisfies a specific condition
- DEFAULT - Sets a default value for a column if no value is specified
- CREATE INDEX - Used to create and retrieve data from the database very quickly

Statistic information:

We state that web scraping combined with big data techniques will allow estimating more individualized and efficient metrics comparable in quality to official statistics. Web scraping technologies empower civil society and small research groups alike by allowing them gather and interpret socioeconomic data. It also helps to create new dimensions of analysis by allowing changes in frequency and focus on specific groups of products and services(4). When evaluating a query, it is often useful to capture metainformation about the result of a query, along with the result itself. The meta-information may indicate where the query result comes from, how it was computed, how many times each result was produced, what probability each result has, etc (5).

We can easily identify the statistical analysis about the price, number of items in each range, number of specific age items.

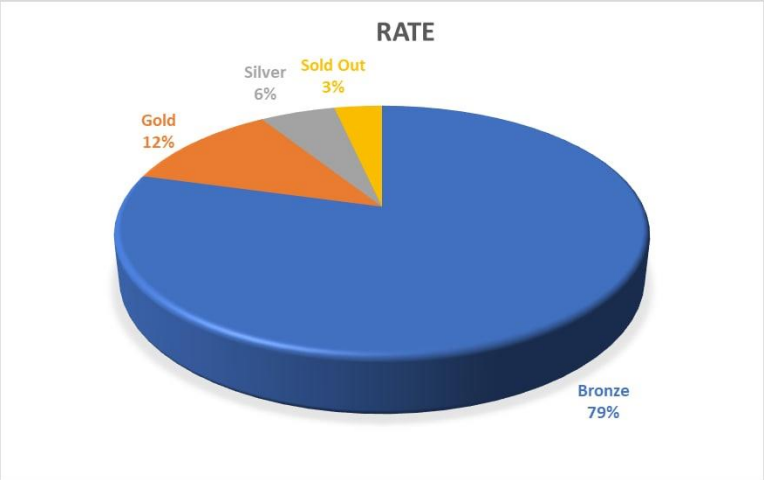


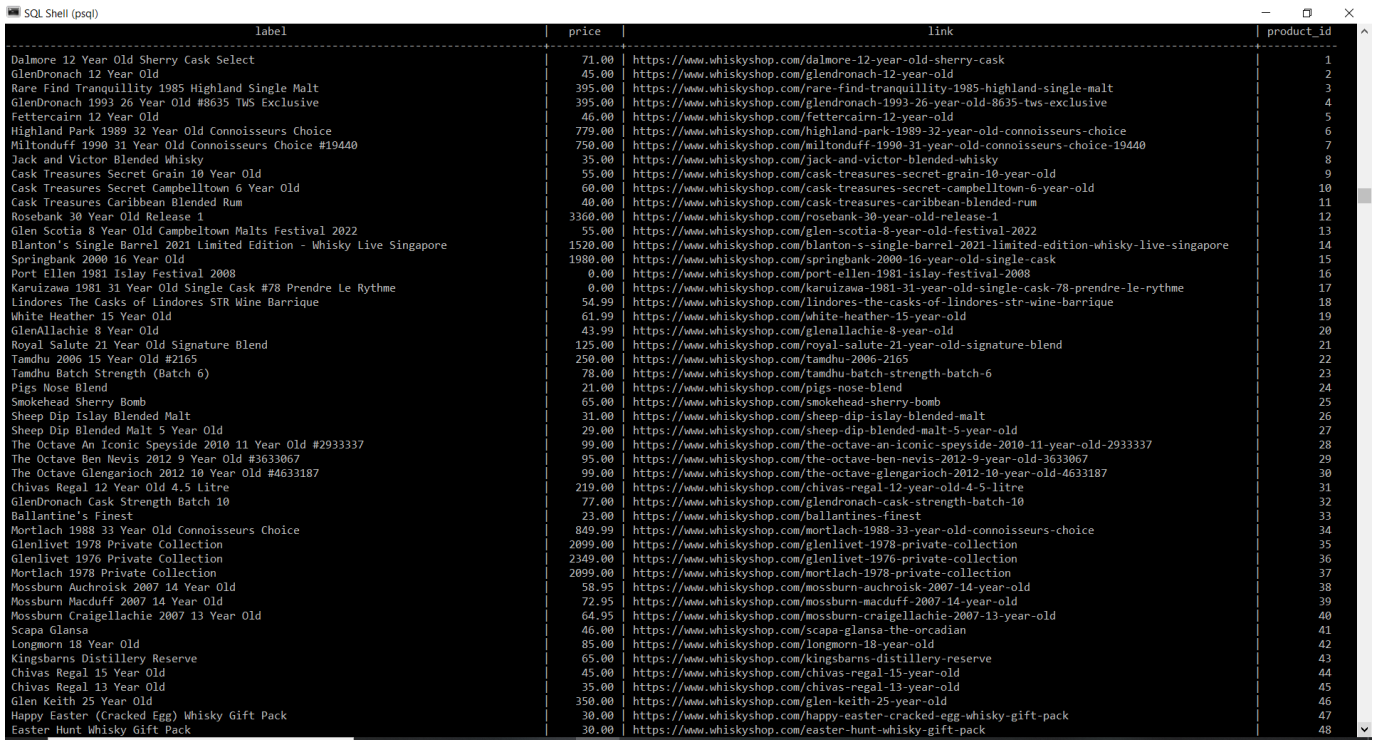
Figure4. Rating Statistic

Implementation:

I created a database and named Scrap.

In this database, I created the main table which contained all dataset that I extracted.

The main Table is named “product” and product_id is the Primary_Key.



label	price	link	product_id
Dalmore 12 Year Old Sherry Cask Select	71.00	https://www.whiskyshop.com/dalmore-12-year-old-sherry-cask	1
GlenDronach 12 Year Old	45.00	https://www.whiskyshop.com/glendronach-12-year-old	2
Rare Find Tranquillity 1985 Highland Single Malt	395.00	https://www.whiskyshop.com/rare-find-tranquillity-1985-highland-single-malt	3
GlenDronach 1993 26 Year Old #8635 TWS Exclusive	395.00	https://www.whiskyshop.com/glendronach-1993-26-year-old-8635-tws-exclusive	4
Fettercairn 12 Year Old	46.00	https://www.whiskyshop.com/fettercairn-12-year-old	5
Highland Park 1989 32 Year Old Connoisseurs Choice	779.00	https://www.whiskyshop.com/highland-park-1989-32-year-old-connoisseurs-choice	6
Miltonduff 1990 31 Year Old Connoisseurs Choice #19440	750.00	https://www.whiskyshop.com/miltonduff-1990-31-year-old-connoisseurs-choice-19440	7
Jack and Victor Blended Whisky	35.00	https://www.whiskyshop.com/jack-and-victor-blended-whisky	8
Cask Treasures Secret Grain 10 Year Old	55.00	https://www.whiskyshop.com/cask-treasures-secret-grain-10-year-old	9
Cask Treasures Secret Campbelltown 6 Year Old	60.00	https://www.whiskyshop.com/cask-treasures-secret-campbelltown-6-year-old	10
Cask Treasures Caribbean Blended Rum	40.00	https://www.whiskyshop.com/cask-treasures-caribbean-blended-rum	11
Rosebank 30 Year Old Release 1	3360.00	https://www.whiskyshop.com/rosebank-30-year-old-release-1	12
Glen Scotia 8 Year Old Campbeltown Malts Festival 2022	55.00	https://www.whiskyshop.com/glen-scotia-8-year-old-festival-2022	13
Blanton's Single Barrel 2021 Limited Edition - Whisky Live Singapore	1520.00	https://www.whiskyshop.com/blanton-s-single-barrel-2021-limited-edition-whisky-live-singapore	14
Springbank 2000 16 Year Old	1980.00	https://www.whiskyshop.com/springbank-2000-16-year-old-single-cask	15
Port Ellen 1981 Islay Festival 2008	0.00	https://www.whiskyshop.com/port-ellen-1981-islay-festival-2008	16
Karuizawa 1981 31 Year Old Single Cask #78 Prendre Le Rythme	0.00	https://www.whiskyshop.com/karuizawa-1981-31-year-old-single-cask-78-prendre-le-rythme	17
Lindores The Casks of Lindores STR Wine Barrique	54.99	https://www.whiskyshop.com/lindores-the-casks-of-lindores-str-wine-barrique	18
White Heather 15 Year Old	61.99	https://www.whiskyshop.com/white-heather-15-year-old	19
GlenAllachie 8 Year Old	43.99	https://www.whiskyshop.com/glenallachie-8-year-old	20
Royal Salute 21 Year Old Signature Blend	125.00	https://www.whiskyshop.com/royal-salute-21-year-old-signature-blend	21
Tamdhu 2006 15 Year Old #2165	250.00	https://www.whiskyshop.com/tamdhu-2006-2165	22
Tamdhu Batch Strength (Batch 6)	78.00	https://www.whiskyshop.com/tamdhu-batch-strength-batch-6	23
Pigs Nose Blend	21.00	https://www.whiskyshop.com/pigs-nose-blend	24
Smokehead Sherry Bomb	65.00	https://www.whiskyshop.com/smokehead-sherry-bomb	25
Sheep Dip Islay Blended Malt	31.00	https://www.whiskyshop.com/sheep-dip-islay-blended-malt	26
Sheep Dip Blended Malt 5 Year Old	29.00	https://www.whiskyshop.com/sheep-dip-blended-malt-5-year-old	27
The Octave An Iconic Speyside 2010 11 Year Old #2933337	99.00	https://www.whiskyshop.com/the-octave-an-iconic-speyside-2010-11-year-old-2933337	28
The Octave Ben Nevis 2012 9 Year Old #3633067	95.00	https://www.whiskyshop.com/the-octave-ben-nevis-2012-9-year-old-3633067	29
The Octave Glengarioch 2012 10 Year Old #4633187	99.00	https://www.whiskyshop.com/the-octave-glengarioch-2012-10-year-old-4633187	30
Chivas Regal 12 Year Old 4.5 Litre	219.00	https://www.whiskyshop.com/chivas-regal-12-year-old-4-5-litre	31
GlenDronach Cask Strength Batch 10	77.00	https://www.whiskyshop.com/glendronach-cask-strength-batch-10	32
Ballantine's Finest	23.00	https://www.whiskyshop.com/ballantines-finest	33
Mortlach 1988 33 Year Old Connoisseurs Choice	849.99	https://www.whiskyshop.com/mortlach-1988-33-year-old-connoisseurs-choice	34
Glenlivet 1978 Private Collection	2099.00	https://www.whiskyshop.com/glenlivet-1978-private-collection	35
Glenlivet 1976 Private Collection	2349.00	https://www.whiskyshop.com/glenlivet-1976-private-collection	36
Mortlach 1978 Private Collection	2099.00	https://www.whiskyshop.com/mortlach-1978-private-collection	37
Mossburn Auchroisk 2007 14 Year Old	58.95	https://www.whiskyshop.com/mossburn-auchroisk-2007-14-year-old	38
Mossburn Macduff 2007 14 Year Old	72.95	https://www.whiskyshop.com/mossburn-macduff-2007-14-year-old	39
Mossburn Craigellachie 2007 13 Year Old	64.95	https://www.whiskyshop.com/mossburn-craigellachie-2007-13-year-old	40
Scapa Glansa	46.00	https://www.whiskyshop.com/scapa-glansa-the-orcadian	41
Longmorn 18 Year Old	85.00	https://www.whiskyshop.com/longmorn-18-year-old	42
Kingsbarns Distillery Reserve	65.00	https://www.whiskyshop.com/kingsbarns-distillery-reserve	43
Chivas Regal 15 Year Old	45.00	https://www.whiskyshop.com/chivas-regal-15-year-old	44
Chivas Regal 13 Year Old	35.00	https://www.whiskyshop.com/chivas-regal-13-year-old	45
Glen Keith 25 Year Old	350.00	https://www.whiskyshop.com/glen-keith-25-year-old	46
Happy Easter (Cracked Egg) Whisky Gift Pack	30.00	https://www.whiskyshop.com/happy-easter-cracked-egg-whisky-gift-pack	47
Easter Hunt Whisky Gift Pack	30.00	https://www.whiskyshop.com/easter-hunt-whisky-gift-pack	48

In This query I categorized items based on their price by SELECT CASE().

I defined as:

Price=0 Sold out , Price>1000 Gold, Price>500 Silver, price>0 Bronze

SQL Shell (psql)

price	product_id	rate
71.00	1	Bronze
45.00	2	Bronze
395.00	3	Bronze
395.00	4	Bronze
46.00	5	Bronze
779.00	6	Silver
750.00	7	Silver
35.00	8	Bronze
55.00	9	Bronze
60.00	10	Bronze
40.00	11	Bronze
3360.00	12	Gold
55.00	13	Bronze
1520.00	14	Gold
1980.00	15	Gold
0.00	16	Sold Out
0.00	17	Sold Out
54.99	18	Bronze
61.99	19	Bronze
43.99	20	Bronze
125.00	21	Bronze
250.00	22	Bronze
78.00	23	Bronze
21.00	24	Bronze
65.00	25	Bronze
31.00	26	Bronze
29.00	27	Bronze
99.00	28	Bronze
95.00	29	Bronze
99.00	30	Bronze
219.00	31	Bronze
77.00	32	Bronze
23.00	33	Bronze
849.99	34	Silver
2099.00	35	Gold
2349.00	36	Gold
2099.00	37	Gold
58.95	38	Bronze
72.95	39	Bronze
64.95	40	Bronze
46.00	41	Bronze
85.00	42	Bronze
65.00	43	Bronze
45.00	44	Bronze
35.00	45	Bronze
350.00	46	Bronze
30.00	47	Bronze
30.00	48	Bronze

Query of only Gold items AND price>1500:

SQL Shell (psql)

product_id	price	rate
12	3360.00	Gold
14	1520.00	Gold
15	1980.00	Gold
35	2099.00	Gold
36	2349.00	Gold
37	2099.00	Gold
62	2995.00	Gold
63	2695.00	Gold
91	2295.00	Gold
105	3999.00	Gold
106	2999.00	Gold
115	3500.00	Gold
120	2250.00	Gold
126	4999.00	Gold
134	3299.00	Gold
139	1750.00	Gold
158	3999.00	Gold
160	3850.00	Gold
186	3750.00	Gold
202	3500.00	Gold
208	3295.00	Gold
213	7999.00	Gold
237	2100.00	Gold
247	2500.00	Gold
248	3999.00	Gold
256	8500.00	Gold
276	6995.00	Gold
282	2499.00	Gold
290	3750.00	Gold
294	2900.00	Gold
324	20995.00	Gold
326	40000.00	Gold
329	3000.00	Gold
333	5500.00	Gold
334	5900.00	Gold
335	3500.00	Gold
336	3500.00	Gold
337	3999.00	Gold
338	3999.00	Gold
339	4400.00	Gold
340	10850.00	Gold
354	1600.00	Gold
366	2995.00	Gold
379	3000.00	Gold
383	8000.00	Gold
385	2750.00	Gold
398	1800.00	Gold
422	2999.00	Gold

String Function(): Extract “XX Year Old” AND “X Year Old” from “Label” column.

SQL Shell (psql)

product_id	year
1	12 Year Old
2	12 Year Old
4	26 Year Old
5	12 Year Old
6	32 Year Old
7	31 Year Old
9	10 Year Old
10	6 Year Old
12	30 Year Old
13	8 Year Old
15	16 Year Old
17	31 Year Old
19	15 Year Old
20	8 Year Old
21	21 Year Old
22	15 Year Old
27	5 Year Old
28	11 Year Old
29	9 Year Old
30	10 Year Old
31	12 Year Old
34	33 Year Old
38	14 Year Old
39	14 Year Old
40	13 Year Old
42	18 Year Old
44	15 Year Old
45	13 Year Old
46	25 Year Old
55	22 Year Old
56	21 Year Old
57	12 Year Old
63	33 Year Old
65	15 Year Old
66	12 Year Old
67	7 Year Old
70	35 Year Old
76	6 Year Old
77	13 Year Old
78	5 Year Old
80	13 Year Old
82	28 Year Old
83	14 Year Old
84	12 Year Old
87	12 Year Old
88	10 Year Old
90	13 Year Old
91	39 Year Old

Convert New extracted column from String to integer, in order to get query from Year column.

SQL Shell (psql)

```
               Table "public.tbl_year"
   Column   |  Type   | Collation | Nullable | Default
------------+---------+-----------+----------+---------
 product_id | integer |           |          |
 year       | integer |           |          |


scrap=#  SELECT * FROM tbl_year;
 product_id | year
------------+------
          1 |   12
          2 |   12
          4 |   26
          5 |   12
          6 |   32
          7 |   31
          9 |   10
         10 |    6
         12 |   30
         13 |    8
         15 |   16
         17 |   31
         19 |   15
         20 |    8
         21 |   21
         22 |   15
         27 |    5
         28 |   11
         29 |    9
         30 |   10
         31 |   12
         34 |   33
         38 |   14
         39 |   14
         40 |   13
         42 |   18
         44 |   15
         45 |   13
         46 |   25
         55 |   22
         56 |   21
         57 |   12
         63 |   33
         65 |   15
         66 |   12
         67 |    7
         70 |   35
         76 |    6
         77 |   13
         78 |    5
```

Result:

This query is created of the result of two previous queries.

Two query combined: 1- price>1500 AND year>10.

product_id	price	year
63	2695.00	33
91	2295.00	39
106	2999.00	36
125	1500.00	40
134	3299.00	40
139	1750.00	40
158	3999.00	40
202	3500.00	43
237	2100.00	38
247	2500.00	59
248	3999.00	51
256	8500.00	46
290	3750.00	56
294	2900.00	41
326	40000.00	35
354	1600.00	40
379	3000.00	40
385	2750.00	40
422	2999.00	40
489	17995.00	50
490	34250.00	50
491	28850.00	54
496	100000.00	80
500	3595.00	36
518	5500.00	50
615	2750.00	50
618	42500.00	31
642	1350.00	32
650	7995.00	35
663	2000.00	50

Figure 5- Result query

Conclusion

PostgreSQL has gained a great reputation as a powerful, feature-rich choice for relational data. Valuing stability, functionality, and standards conformance, PostgreSQL checks all of the right boxes for many projects. Similarly, if you require flexibility in how you can represent data and want to be able to use a variety of tools and languages, PostgreSQL is also a good choice. PostgreSQL is notable for offering excellent implementation of core relational features while not limiting itself to the boundaries of traditional RDBMSs. While no database can serve every need, PostgreSQL is an excellent option that is versatile enough to suit many use cases. When we consider web scraping, in terms of speed and efficiency Scrapy is a better choice. Scrapy is a powerful web-scraping framework that can be used for scraping huge volumes of data from different websites.

References:

1. Kaiying, D., Senpeng, C., & Jingwei, D. (2020). On optimisation of web crawler system on Scrapy framework. International Journal of Wireless and Mobile Computing, 18(4), 332-338.
2. Kaiying, D., Senpeng, C., & Jingwei, D. (2020). On optimisation of web crawler system on Scrapy framework. *International Journal of Wireless and Mobile Computing*, *18*(4), 332-338.
3. DeVito, Richards, G. C., & Inglesby, P. (2020). How we learnt to stop worrying and love web scraping. Nature (London), 585(7826), 621–622. https://doi.org/10.1038/d41586-020-02558-0
4. Uriarte, Ramírez Muñoz De Toro, G. R., & Larrosa, J. M. . (2020). Web scraping based online consumer price index: The “iPC Online” case. Journal of Economic and Social Measurement, 44(2-3), 141–159. https://doi.org/10.3233/JEM-190464
5. Senellart, Jachiet, L., Maniu, S., & Ramusat, Y. (2018). ProvSQL: Provenance and probability management in PostgreSQL. Proceedings of the VLDB Endowment, 11(12), 2034–2037. https://doi.org/10.14778/3229863.3236253