



► Result Arkon Test

Data Engineer

Anabel Rodríguez Rodríguez

04/12/2024

Resume

- ▶ My github repositories:

https://github.com/anitarr/Arkon_Test

- ▶ Python Code:

https://github.com/anitarr/Arkon_Test/blob/master/src/my_code.ipynb

- ▶ SQL Code

https://github.com/anitarr/Arkon_Test/blob/master/sql/test5-7.sql

- ▶ PostgreSQL DB in Docker

https://github.com/anitarr/Arkon_Test/blob/master/Readme.md

- ▶ Connecting postgres with Python

https://github.com/anitarr/Arkon_Test/blob/master/postgresDB.ipynb

Initialize a new Git repository

▶ `git init`

Add files to the staging area

▶ `git add -A`

Commit the changes

▶ `git commit -m "First commit"`

Add the GitHub remote

▶ `git remote add origin
https://github.com/anitarr/Arkon_test.git`

Push the changes to GitHub

▶ `git push -u origin main`

Git and Github

Python code

Question 1

#Importing the pandas and numpy libraries

- ▶ import pandas as pd
- ▶ import numpy as np

#Read data from “.parquet” files

- ▶ df_parquet =
pd.read_parquet(r"D:\DATA_Analysis\Arkon\data\data2
(1).parquet")

#Read data from “.csv” files

- ▶ df_data1 =
pd.read_csv(r"D:\DATA_Analysis\Arkon\data\Data1.csv")

#Question 1. Join the two datasets into one

- ▶ df_union = pd.concat([df_parquet, df_data1])

#Delete NAN and duplicate data

- ▶ df_union_dna = df_union.dropna().drop_duplicates()

Question 2. Unique 'starships' values

- ▶ `valores_unicos = df_union["starships"].unique()`

Question 3. Generate a record count on the group [Skin_color, eye_color]

- ▶ `count_s_e = df_union.groupby(['skin_color', 'eye_color']).size().reset_index(name='Count')`

Question 4. Generate a table with the duplicate 'Names' and how many times they are repeated.

- ▶ `name_counts = df_union['name'].value_counts().reset_index()`
- ▶ `name_counts.columns = ['Name', 'Count']`
- ▶ `duplicates = name_counts[name_counts['Count'] > 1]`

Python code

Question 3,4,5

Python code

Questions 5,6,7

#Question 5. Filter in python

```
▶ filtered_df = df_union[
    (df_union['height'] >= 180) &
    (df_union['height'] <= 190) &
    (df_union['sex'] == 'male') &
    (df_union['hair_color'] != 'none')]
```

#Question 6. Record count on the group ['skin_color', 'eye_color']

```
▶ avg = df_union['mass'].mean()
▶ # create new column 'flat' and show results
▶ df_union['flat'] = df_union['mass'].apply(lambda x: 1 if x > avg
    else 0)
▶ mass_flat = df_union[['name', 'mass', 'flat']]
```

#Question 7. Metrics column 'species'

```
▶ result = df_union.groupby('species').agg(
    avg_height=('height', 'mean'),
    max_height=('height', 'max'),
    min_height=('height', 'min')).reset_index()
```

/*Question 2 Unique 'starships' values */

▶ SELECT DISTINCT starships FROM data_union;

/* Question 3 Generate a record count on the group
[Skin_color, eye_color] */

▶ SELECT COUNT(*), skin_color, eye_color FROM data_union
group by skin_color, eye_color;

/* Question 4 Generate a table with the duplicate 'Names' and
how many times they are repeated.*/

▶ SELECT name, COUNT(*) AS cantidad_duplicados
FROM data_union
GROUP BY name
HAVING COUNT(*) > 1;

SQL code

Questions 2,3,4

SQL code

Questions 5,6,7

/* Question 5 . Filter in SQL*/

▶ SELECT name FROM testdata.data_union
WHERE height BETWEEN 180 AND 190 AND sex = 'male' AND
hair_color!='none';

/* Question 6 Record count on the group ['skin_color', 'eye_color']*/

▶ SELECT name, mass,
CASE
WHEN mass > (SELECT AVG(mass) FROM data_union) THEN 1
ELSE 0
END AS bandera
FROM data_union;

/* Question 7 Metrics column 'species' */

▶ SELECT species,
AVG(height) AS altura_promedio,
MAX(height) AS altura_maxima,
MIN(height) AS altura_minima
FROM data_union
GROUP BY species;

PostgreSQL DB in Docker

Pull/Download Official Postgres Image From Docker Hub

- ▶ `docker pull postgres`

Create and Run Postgres Container

- ▶ `docker run -d --name arkon_data -p 5432:5432 -e POSTGRES_PASSWORD=pass1234 postgres`

Initialize the database connection

```
▶ db = PostgresDB(  
    host="localhost",  
    database="postgres",  
    user="postgres",  
    password="pass1234"  
)  
db.connect()
```

Postgres DB with python

Generate SQL for creating the table

- ▶ `table_name = "data_union"`
- ▶ `schema_name = "arkon_data"`
- ▶ `columns = []`
- ▶ `for column_name, dtype in df_union.dtypes.items():`
 - `sql_type = map_dtype_to_sql(dtype)`
 - `columns.append(f"{column_name} {sql_type}")`
- ▶
`create_table_query = f"""`
`CREATE SCHEMA IF NOT EXISTS {schema_name};`
`CREATE TABLE IF NOT EXISTS`
`{schema_name}.{table_name} (`
`id SERIAL PRIMARY KEY,`
`{', '.join(columns)}`
`);`
`"""`
- ▶ `db.execute_query(create_table_query)`

Postgres DB with python

Insert data into the table

```
▶ insert_query_template = f"""  
    INSERT INTO {schema_name}.{table_name} ({',  
    '.join(df_union.columns)}) VALUES ({',  
    '.join(['%s'] * len(df_union.columns))});  
    ""
```

```
▶ cursor = db.connection.cursor()  
▶ for index, row in df_union.iterrows():  
▶ cursor.execute(insert_query_template,  
    tuple(row))  
▶ db.connection.commit()  
▶ cursor.close()
```

Close the database connection

```
▶ db.close_connection()
```

Postgres DB with python

DBeaver queries in Postgres

Question 5 . Filter in Postgres

DBeaver 24.2.4 - <postgres> Script-1

Archivo Editar Navegar Search Editor SQL Base de Datos Ventana Ayuda

SQL Commit Rollback Auto postgres public@postgres

Navega... x Proyec... x *<postgres> Script-1 x arkon_data

Ingrese parte del nombre de un ob

Esquemas

arkon_data

Tablas

data_union

Columnas

- 123 id (serial4)
- A-Z name (text)
- 123 height (float)
- 123 mass (float)
- A-Z skin_color (text)
- A-Z eye_color (text)
- 123 birth_year (integer)
- A-Z hair_color (text)
- A-Z sex (text)
- A-Z gender (text)
- A-Z species (text)
- A-Z homeworld (text)
- A-Z films (text)
- A-Z vehicles (text)

```
SELECT name FROM arkon_data.data_union
WHERE height BETWEEN 180 AND 190 AND sex = 'male' AND hair_color != 'none';
```

data_union 1 x

SELECT name FROM arkon_data.data_union WHERE

Grilla

	A-Z name
1	Jango Fett
2	Raymus Antilles
3	Cliegg Lars
4	Biggs Darklighter
5	Obi-Wan Kenobi
6	Anakin Skywalker
7	Wilhuff Tarkin
8	Han Solo
9	Boba Fett
10	Ric Olié
11	Quarsh Panaka
12	Cliegg Lars

Record

Refresh Save Cancel Exportar datos ... 200 12

CST es Writable Smart Insert 3 : 1 : 115 Sel: 0 | 0

Question 6 Record count on the group ['skin_color', 'eye_color']

DBEaver 24.2.4 - <postgres> Script-1

Archivo Editar Navegar Search Editor SQL Base de Datos Ventana Ayuda

SQL Commit Rollback Auto postgres public@postgres

Navega... Proyec... *<postgres> Script-1 x arkon_data

ALTER TABLE arkon_data.data_union ADD COLUMN flat INT;

CREATE TEMPORARY TABLE temp_avg_mass AS
SELECT AVG(mass) AS avg_mass FROM arkon_data.data_union;

-- Ejecutar la actualización
UPDATE arkon_data.data_union
SET flat = CASE
WHEN mass > (SELECT avg_mass FROM temp_avg_mass) THEN 1
ELSE 0
END;

select name, mass, flat from arkon_data.data_union

data_union 1 Estadísticas 2 Estadísticas 3 Estadísticas 4 data_union 5 data_union 6 x

select name, mass, flat from arkon_data.data_un

Grilla

	A-Z name	123 mass	123 flat
44	Shaak Ti	57	0
45	Grievous	159	0
46	Mace Windu	84	0
47	Ki-Adi-Mundi	82	0
48	Kit Fisto	87	0
49	Mace Windu	84	0
50	Ki-Adi-Mundi	82	0
51	Kit Fisto	87	0
52	Luke Skywalker	77	0
53	C-3PO	75	0
54	R2-D2	32	0
55	Darth Vader	136	0
56	Leia Organa	49	0
57	Owen Lars	120	0

Record

Refresh Save Cancel Exportar datos ... 200 116

CST es Writable Smart Insert 13:52 [51] Sel: 51 | 1

Question 7 Metrics column 'species'

DBeaver 24.2.4 - <postgres> Script-1

Archivo Editar Navegar Search Editor SQL Base de Datos Ventana Ayuda

SQL Commit Rollback Auto postgres public@postgres

Navega... x Proyec... x *<postgres> Script-1 x arkon_data

Ingresa parte del nombre de un ob

- 123 mass (flo
- A-Z skin_colo
- A-Z eye_color
- 123 birth_yea
- A-Z hair_colo
- A-Z sex (text)
- A-Z gender (t
- A-Z species (t
- A-Z homewo
- A-Z films (tex
- A-Z vehicles (
- A-Z starships
- 123 flat (int4)

Restriccione

Claves forán

Índices

Dependenci

Referencias

Particiones

Project - General x

Name DataSource

- > Bookmarks
- > Dashboards
- > Diagrams
- > Scripts

```
SELECT
  species,
  AVG(height) AS avg_height,
  MAX(height) AS max_height,
  MIN(height) AS min_height
FROM
  arkon_data.data_union
GROUP BY
  species;
```

select name, mass, flat from arkon_data.data_union

data_union 1 Estadísticas 2 Estadísticas 3 Estadísticas 4 data_union 5 data_union 6 x

SELECT species, AVG(height) AS avg_height, MA Enter a SQL expression to filter results (use Ctrl+Space)

	A-Z species	123 avg_height	123 max_height	123 min_height
20	Iktotchi	188	188	188
21	Zabrak	172,3333333333	175	171
22	Skakoan	193	193	193
23	Mon Calamari	180	180	180
24	Cerean	198	198	198
25	Geonosian	183	183	183
26	Twilek	179	180	178
27	Chagrian	196	196	196
28	Toydarian	137	137	137
29	Tholothian	184	184	184
30	Xexto	122	122	122
31	Toong	163	163	163

Refresh Save Cancel Exportar datos ... 200 39

39 row(s) fetched - 0,005s (0,001s fetch), on 2024-12-04 at 16:54:25

CST es Writable Smart Insert 8:10:161 Sel: 0 | 0

Questions

The image features a large, three-dimensional gold question mark as the central focus. It is surrounded by a dense field of smaller, dark grey question marks that create a textured, almost infinite background. The composition is framed by blue geometric elements: a solid light blue triangle in the top-left corner and a series of overlapping, semi-transparent blue triangles on the right side. The word "Questions" is written in a clean, white, sans-serif font, positioned in the upper-middle section of the image.



Thanks