

Angel David Cuellar Bautista
22 de mayo de 2021
Prueba de conocimientos BAM

Prueba de conocimientos BAM

Repositorio del proyecto: [AngelloDavincii/BAM_EVALUACION: Evaluación para aplicación para ingeniero de datos en el BAM \(github.com\)](https://github.com/AngelloDavincii/BAM_EVALUACION)

Explicación en vídeo de todo lo realizado:
https://drive.google.com/drive/folders/1uhYpl60d0aokLdft3hGg_Uq4QW3z8U8H?usp=sharing

Parte 1

Particularmente escogí PostgreSQL para crear mi base de datos local porque cuando trabajé programando en Django este era el motor de bases de datos que utilizaba, además me parece que presenta más potencia frente a otros motores como MySQL, mariaDB, etc. Esto sin tomar en cuenta que es más API friendly y más versátil. También porque más adelante voy a desarrollar en Django una pequeña página web para mostrar los resultados.

Importando los datos

Create - Table

General

Columns

Advanced

Constraints

Partitions

Parameters

Security

SQL

Name

datos_base_clientes

Owner

postgres

Schema

public

Tablespace

Select an item...

Partitioned table?

No

Comment

i

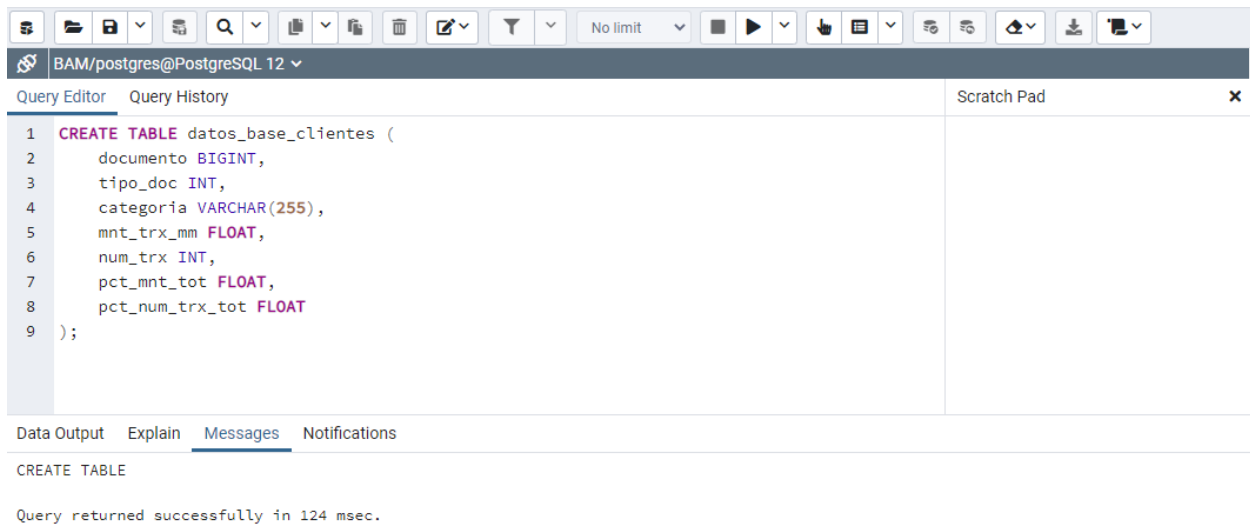
?

Cancel

Reset

Save

Angel David Cuellar Bautista
22 de mayo de 2021
Prueba de conocimientos BAM



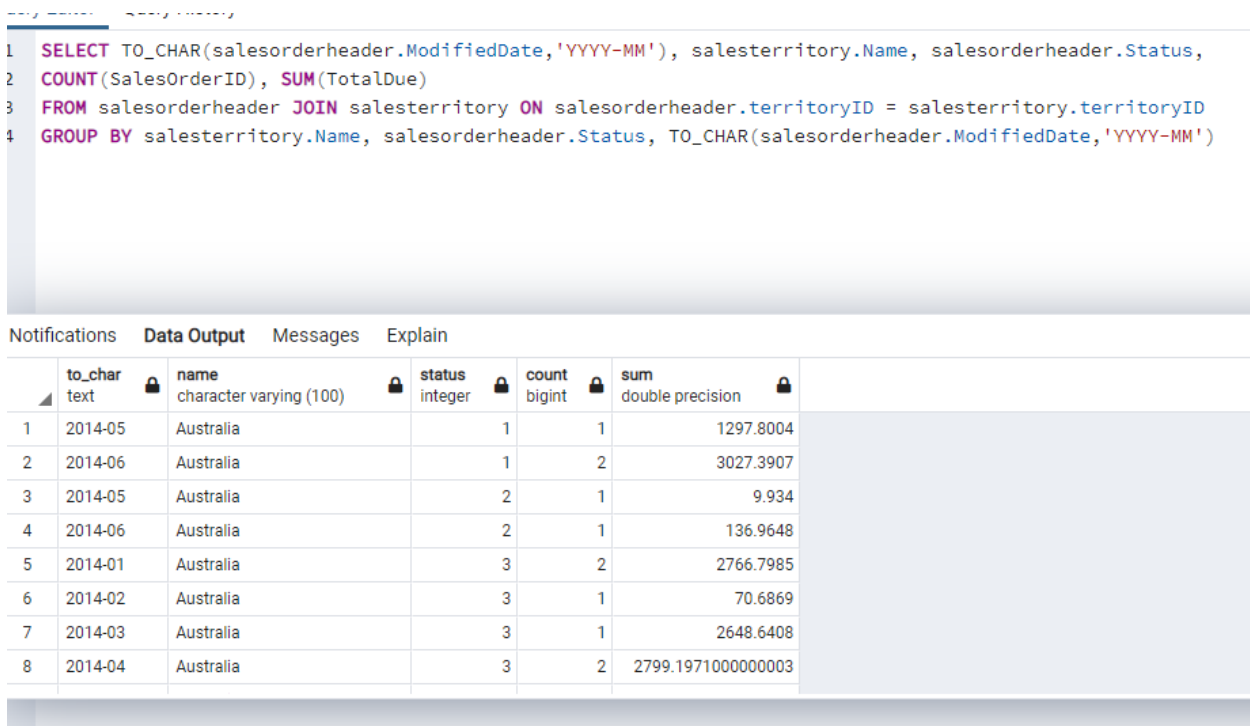
The screenshot shows a PostgreSQL query editor interface. The top toolbar includes icons for file operations, search, and execution. The main window is titled "BAM/postgres@PostgreSQL 12" and contains a "Query Editor" tab. The query being executed is a CREATE TABLE statement for a table named "datos_base_clientes". The query is as follows:

```
1 CREATE TABLE datos_base_clientes (  
2     documento BIGINT,  
3     tipo_doc INT,  
4     categoria VARCHAR(255),  
5     mnt_trx_mm FLOAT,  
6     num_trx INT,  
7     pct_mnt_tot FLOAT,  
8     pct_num_trx_tot FLOAT  
9 );
```

Below the query editor, the "Messages" tab is active, displaying the following message:

```
CREATE TABLE  
  
Query returned successfully in 124 msec.
```

Inciso 1



The screenshot shows a PostgreSQL query editor interface. The main window is titled "BAM/postgres@PostgreSQL 12" and contains a "Query Editor" tab. The query being executed is a SELECT statement that joins salesorderheader and salesterritory tables. The query is as follows:

```
1 SELECT TO_CHAR(salesorderheader.ModifiedDate,'YYYY-MM'), salesterritory.Name, salesorderheader.Status,  
2 COUNT(SalesOrderID), SUM(TotalDue)  
3 FROM salesorderheader JOIN salesterritory ON salesorderheader.territoryID = salesterritory.territoryID  
4 GROUP BY salesterritory.Name, salesorderheader.Status, TO_CHAR(salesorderheader.ModifiedDate,'YYYY-MM')
```

Below the query editor, the "Data Output" tab is active, displaying the following results:

	to_char text	name character varying (100)	status integer	count bigint	sum double precision
1	2014-05	Australia		1	1297.8004
2	2014-06	Australia		1	3027.3907
3	2014-05	Australia		2	9.934
4	2014-06	Australia		2	136.9648
5	2014-01	Australia		3	2766.7985
6	2014-02	Australia		3	70.6869
7	2014-03	Australia		3	2648.6408
8	2014-04	Australia		3	2799.1971000000003

Inciso 2

Angel David Cuellar Bautista
22 de mayo de 2021
Prueba de conocimientos BAM

BAM/postgres@PostgreSQL 12

Query Editor Query History

```
1 SELECT TO_CHAR(salesorderheader.OrderDate,'YYYY-MM'),salesorderheader.customerid, salesorderdetail.productID,  
2 COUNT(salesorderdetail.productID), SUM(salesorderdetail.LineTotal)  
3 FROM salesorderheader JOIN salesorderdetail ON salesorderheader.salesorderid = salesorderdetail.salesorderid  
4 WHERE salesorderheader.status = 2  
5 GROUP BY salesorderdetail.productID,TO_CHAR(salesorderheader.OrderDate,'YYYY-MM'),  
6 salesorderheader.customerid  
7 ORDER BY COUNT(salesorderdetail.productID) DESC, SUM(salesorderdetail.LineTotal) DESC  
8 LIMIT 3
```

Notifications Data Output Messages Explain

	to_char text	customerid integer	productid integer	count bigint	sum double precision
1	2014-05	16560	794	1	2443.35
2	2014-05	28076	968	1	2384.07
3	2014-05	18413	780	1	2319.99

Inciso 4

BAM/postgres@PostgreSQL 12

Query Editor Query History

```
1 select distinct l1.codigo as ConsecutiveNums  
2 from  
3 secuencial l1,  
4 secuencial l2,  
5 secuencial l3  
6 where l1.secuencial = l2.secuencial-1  
7 and l2.secuencial = l3.secuencial-1  
8 and l1.codigo=l2.codigo  
9 and l2.codigo=l3.codigo
```

Notifications Data Output Messages

	consecutivenums integer
1	3600
2	3602
3	3604
4	3605
5	3607
6	3608
7	3609
8	3613

Inciso 5

Angel David Cuellar Bautista
22 de mayo de 2021
Prueba de conocimientos BAM

Query Editor

Query History

```
1 SELECT TO_CHAR(T.OrderDate,'YYYY-MM') AS MES,T.CustomerID AS CLIENTE, MAX(T.OrderDate) AS FechaUltCompra,
2         (MAX(T.OrderDate) -
3         (SELECT MAX(N.OrderDate) FROM salesorderheader N WHERE
4         N.customerid = T.customerid AND
5         N.OrderDate < MAX(T.OrderDate))) AS DIASUC_PC
6 FROM salesorderheader T
7 GROUP BY TO_CHAR(T.OrderDate,'YYYY-MM'), T.CustomerID
```

Notifications

Data Output

Messages

	<div>mes</div> <div>text</div>	<div>cliente</div> <div>integer</div>	<div>fechaultcompra</div> <div>date</div>	<div>diasuc_pc</div> <div>integer</div>
1	2014-06	26460	2014-06-18	[null]
2	2013-08	27791	2013-08-02	[null]
3	2012-09	13024	2012-09-16	[null]
4	2014-01	15167	2014-01-14	570
5	2013-07	28248	2013-07-24	[null]
6	2014-04	26423	2014-04-11	535
7	2014-03	27848	2014-03-30	[null]
8	2014-03	25537	2014-03-24	[null]
-	-	-	-	-

Parte 2

Angel David Cuellar Bautista
22 de mayo de 2021
Prueba de conocimientos BAM

```
File Edit Format Run Options Window Help
# Angel David Cuellar
# 23 de Mayo de 2021

# Importamos la libreria
import psycopg2

# Definimos funcion que ejecuta la query en postgres
def ejecutar(db, file, dest):
    # Nos conectamos a la base de datos autenticandonos con los datos
    conn = psycopg2.connect(user="postgres",
                             password="belylch0-",
                             host="127.0.0.1",
                             port="5432",
                             database=db)

    cur = conn.cursor()
    # Leemos el archivo
    read = open(file, "r").read()
    # Customizamos la query
    query = "CREATE TABLE %s AS (%s)" % (dest, read)
    # Ejecutamos la query
    cur.execute(query)
    # Hacemos los cambios en la base de datos
    conn.commit()

def menu():
    # Creamos un menu basico con prints e inputs para interactuar con el usuario
    print("BIENVENIDO A ESTE PROGRAMA")
    print("")
    name = input("Por favor ingrese el nombre de la base de datos:")
    print("")
    print("Ingrese el nombre del archivo .sql a ejecutar. Porfavor fijese que el")
    file = input("Archivo este en el mismo directorio que este programa: ")
    print("")
    destino = input("Ingrese el nombre de la base de datos de destino: ")

    ejecutar(name, file, destino)

    print("")
    print("Tu programa ha sido exitosamente ejecutado y tu resultado se guardo en "+ destino)
    print("GRACIAS POR USAR ESTE PROGRAMA")
```

Puntos extra

Angel David Cuellar Bautista
22 de mayo de 2021
Prueba de conocimientos BAM

