ADBMS Experiment 4

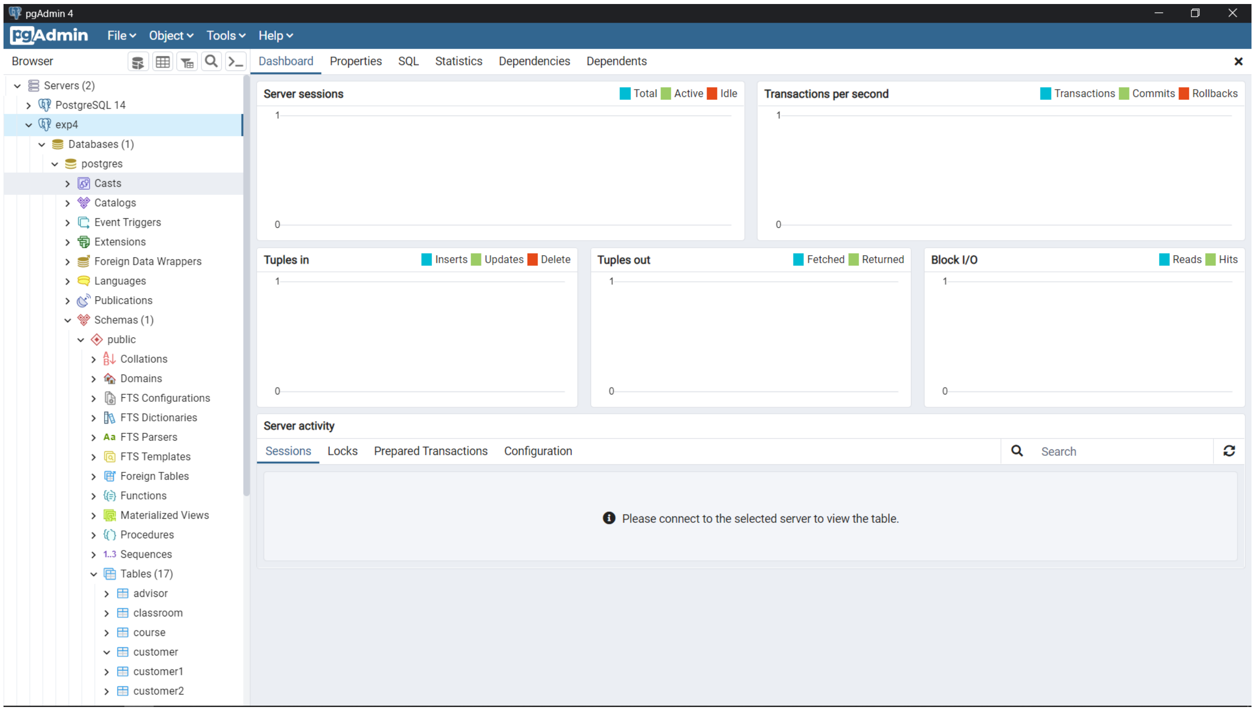
Ojas Patil
2019130048
TE COMPS

Aim: To implement the replication technique for a given scenario
Scenario:

A relation named customer belonging to a certain firm contains details of a customer (CUST_CODE, CUST_NAME, CUST_CITY, WORKING_AREA, CUST_COUNTRY, PAYMENT_AMT, PHONE_NO). The schema is replicated on two different nodes. The data consistency is achieved through a python script which approaches both the database through the url they are hosted on and the port no. Changes on one replica are made on another replica as well and even if one replica fails to make changes, both the replicas initialise rollbacks and get back to their recent consistent state.

Procedure:

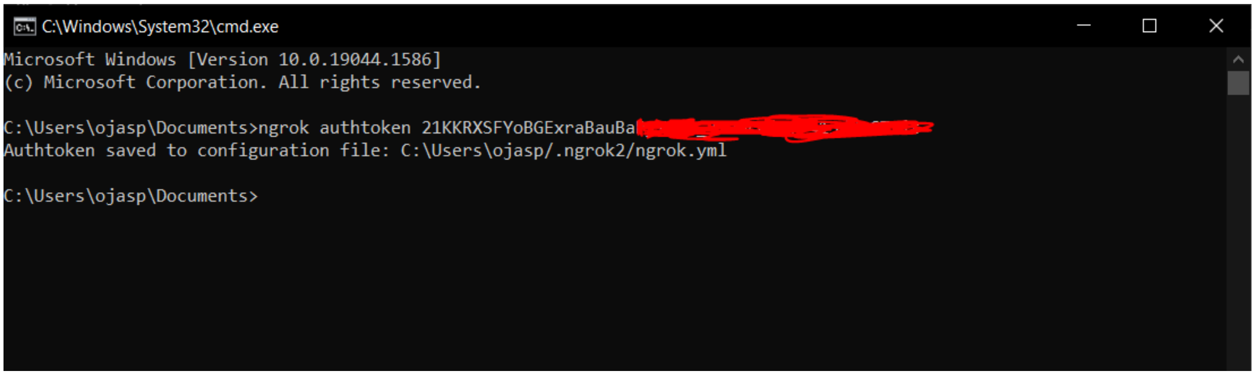
1. PostgreSQL and pgAdmin14 are installed on both the machines which are acting as nodes.



Please connect to the selected server to view the table.

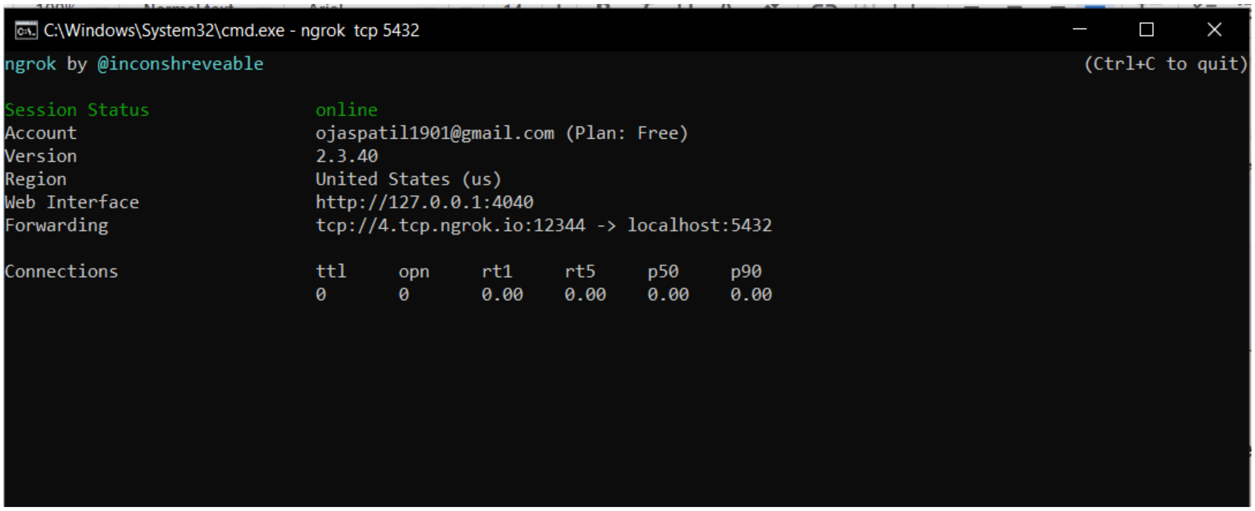
pgAdmin4 + postgreSQL 14

2. Download and extract ngrok from [ngrok - download](). Log in and get your authorization token.
   On Extracting ngrok, open cmd where the ngrok is extracted and run the command ‘ngrok authtoken <token>’



```
Microsoft Windows [Version 10.0.19044.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ojasp\Documents>ngrok authtoken 21KKRXSFYoBGExraBauBa[redacted]
Authtoken saved to configuration file: C:\Users\ojasp/.ngrok2/ngrok.yml

C:\Users\ojasp\Documents>
```

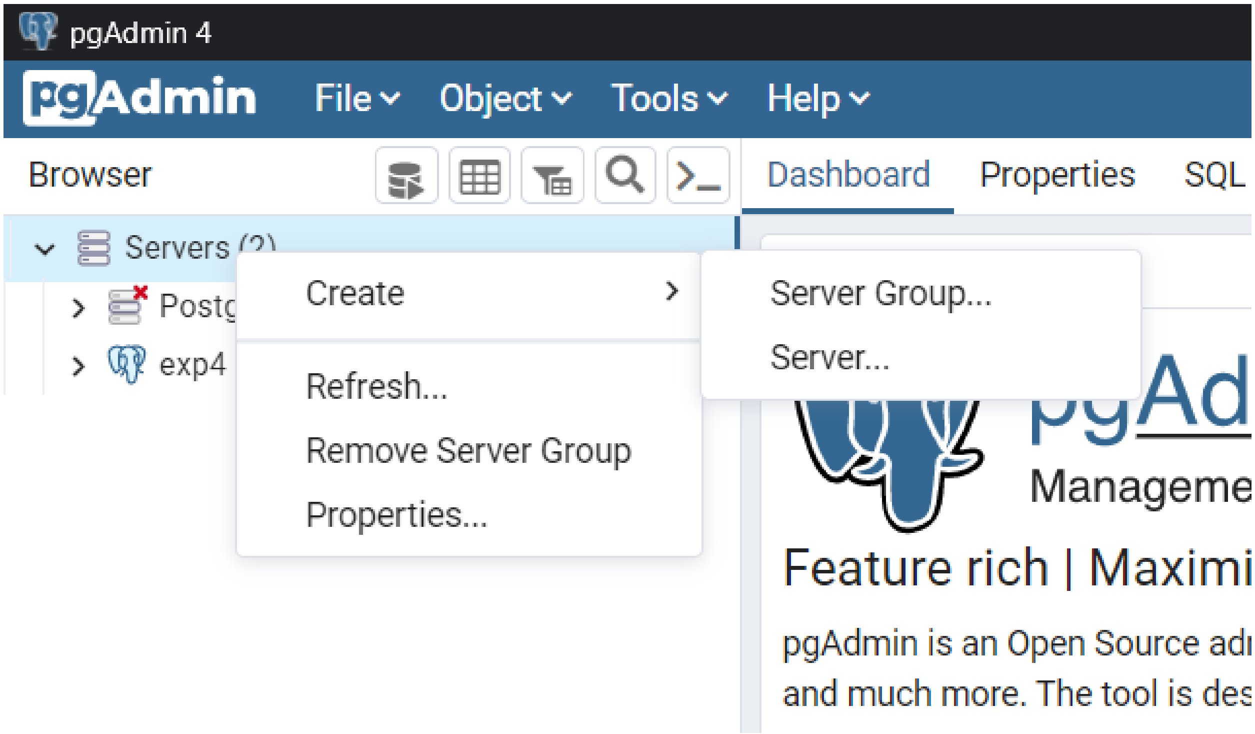
3. Now, run command ‘ngrok tcp 5432’ to expose port 5432 where the postgres server runs.



C:\Windows\System32\cmd.exe - ngrok tcp 5432

```
ngrok by @inconshreveable                                                                          (Ctrl+C to quit)

Session Status                online
Account                       ojaspatil1901@gmail.com (Plan: Free)
Version                       2.3.40
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://4.tcp.ngrok.io:12344 -> localhost:5432

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

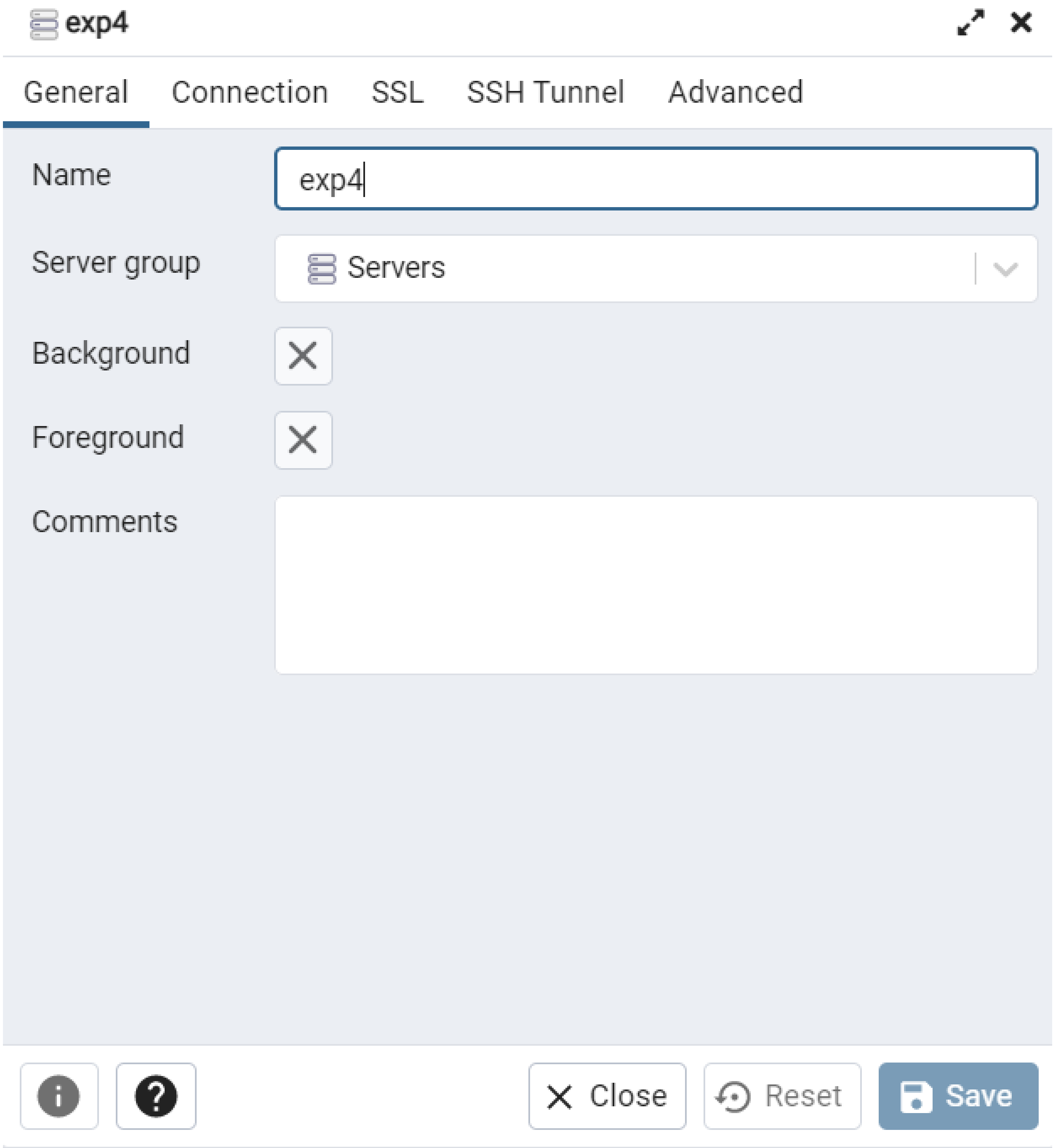
4. Open pgAdmin4 and create a new server.
Right click on Servers and goto Create -> Server.



Feature rich | Maximi

pgAdmin is an Open Source adı
and much more. The tool is des

5. Provide a suitable name to server



Name: exp4

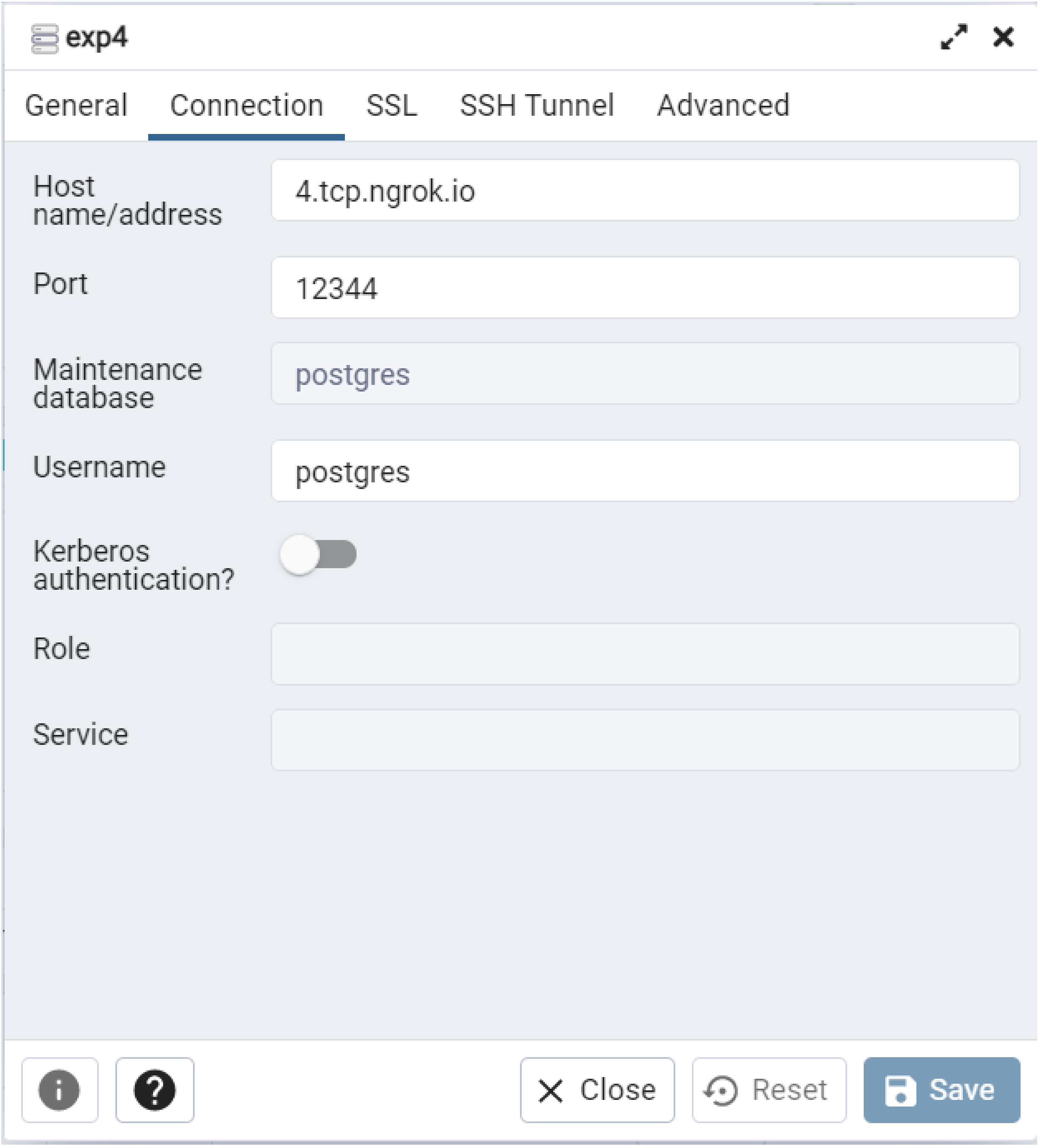
Server group: Servers

Background

Foreground

Comments

6. Paste the url and port no (Obtained in step 3; select mandatory part of url only. Refer image below) in appropriate boxes



Host name/address: 4.tcp.ngrok.io

Port: 12344

Maintenance database: postgres

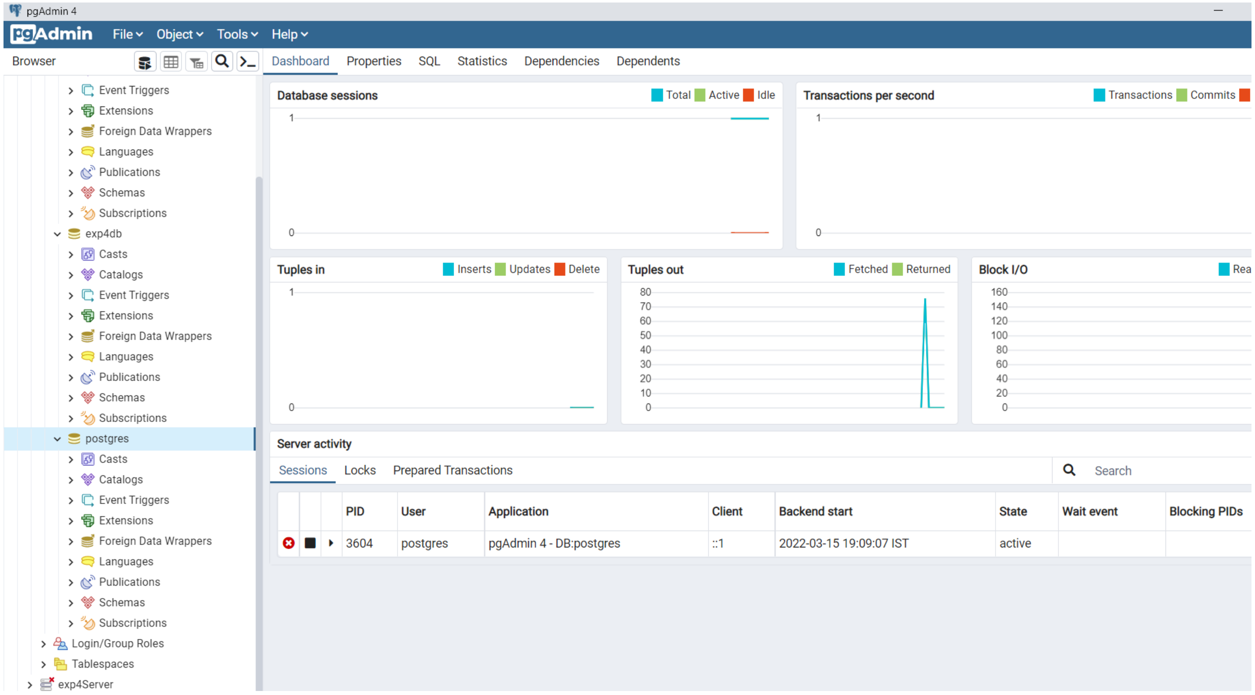
Username: postgres

Kerberos authentication? ○

Role:

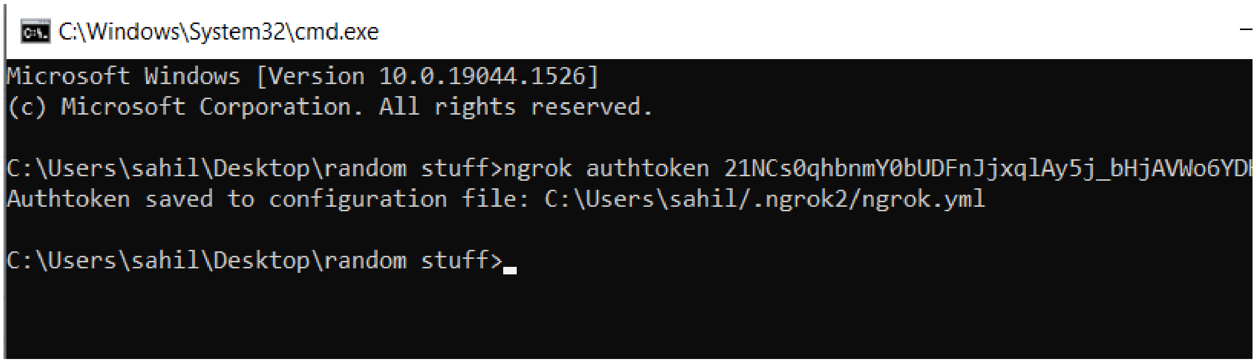
Service:

7. Repeat steps 1-6 on another node as well.



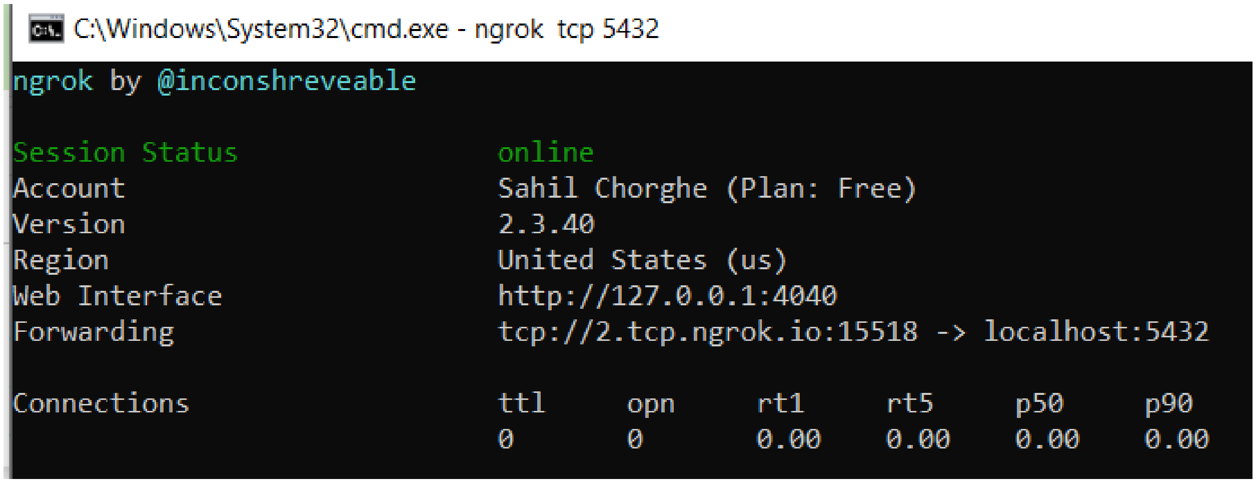
			PID	User	Application	Client	Backend start	State	Wait event	Blocking PIDs
⊗	■	▸	3604	postgres	pgAdmin 4 - DB:postgres	::1	2022-03-15 19:09:07 IST	active		

Repeated step 1



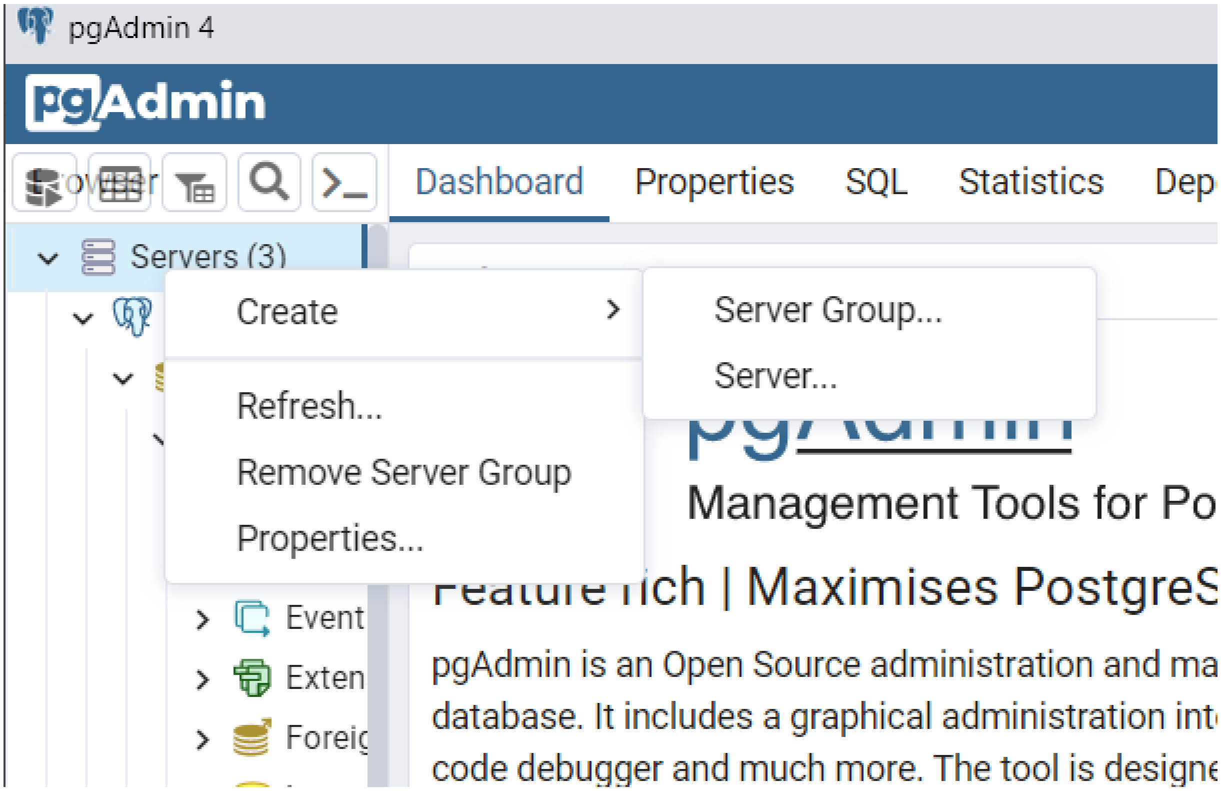
```
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sahil\Desktop\random stuff>ngrok authtoken 21NCs0qhbnmY0bUDFnJjxqlAy5j_bHjAVWo6YD
Authtoken saved to configuration file: C:\Users\sahil/.ngrok2/ngrok.yml

C:\Users\sahil\Desktop\random stuff>
```

Repeated step 2



```
C:\Windows\System32\cmd.exe - ngrok  tcp 5432
ngrok by @inconshreveable

Session Status                online
Account                       Sahil Chorghe (Plan: Free)
Version                       2.3.40
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://2.tcp.ngrok.io:15518 -> localhost:5432

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Repeated Step 3

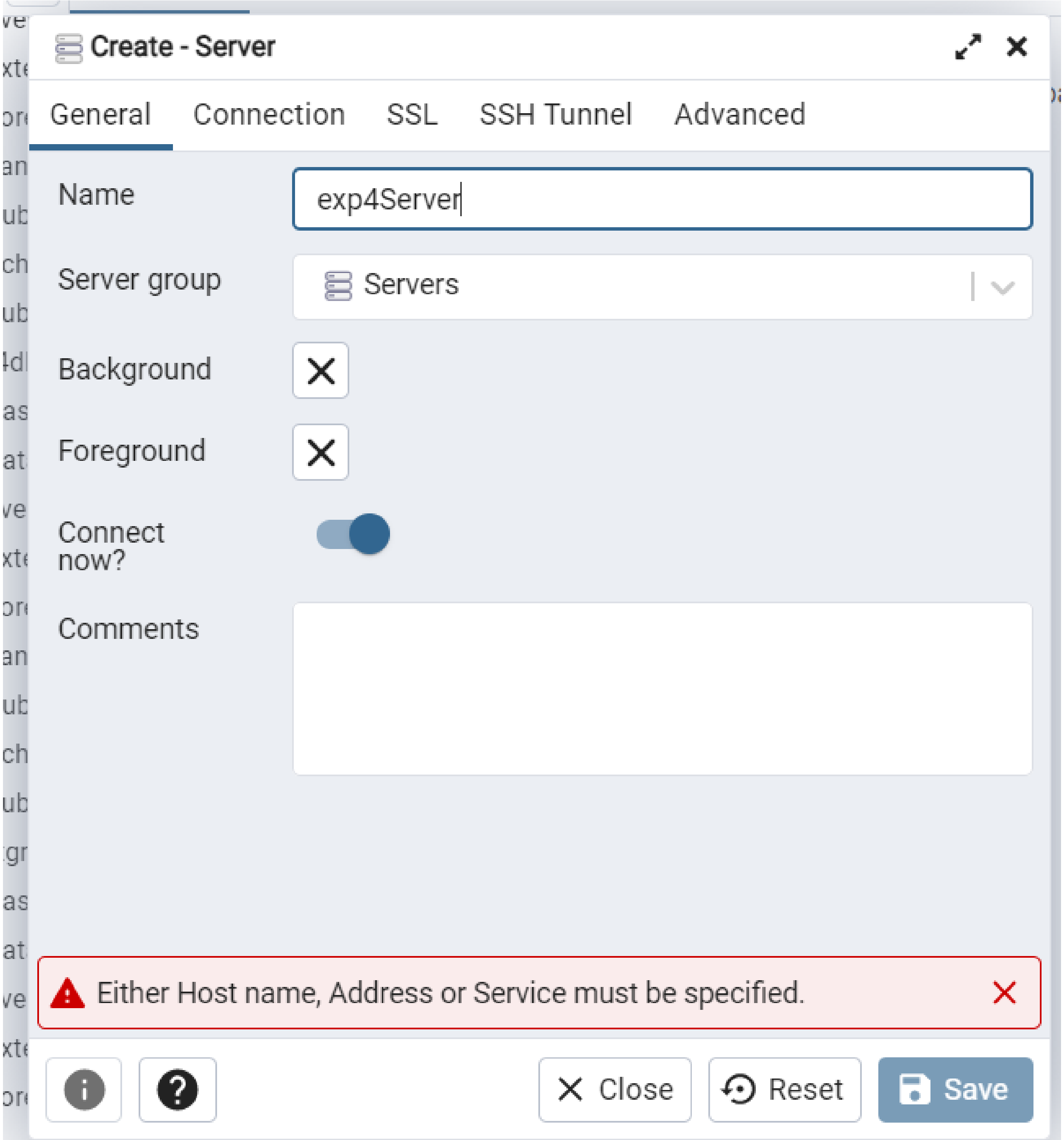


Management Tools for Po

Feature rich | Maximises PostgreS

pgAdmin is an Open Source administration and ma
database. It includes a graphical administration int
code debugger and much more. The tool is designe

Repeated Step 4



Create - Server

Name: exp4Server

Server group: Servers

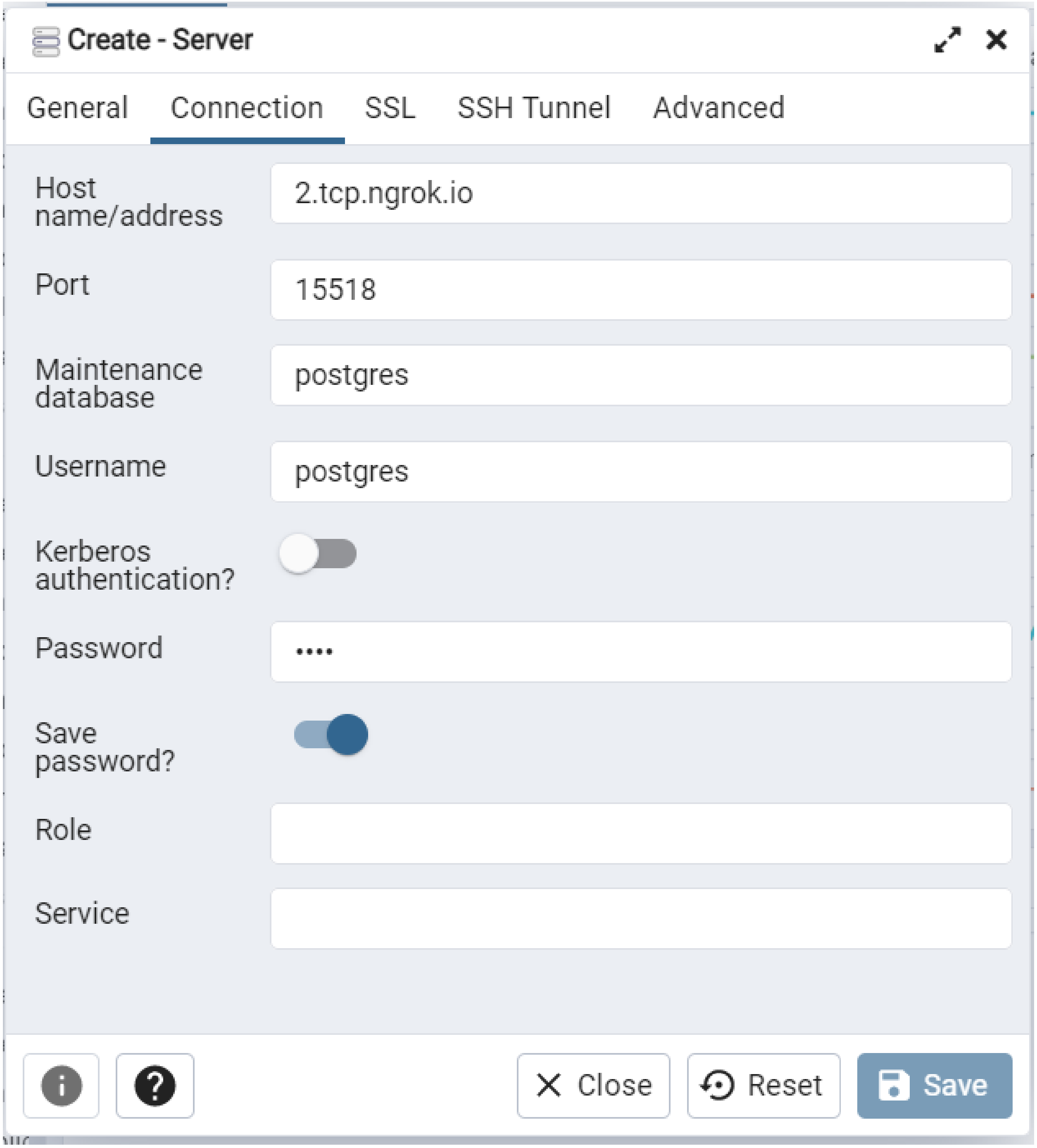
Background: ☒

Foreground: ☒

Connect now? ●

Comments:

Either Host name, Address or Service must be specified.



Host name/address: 2.tcp.ngrok.io

Port: 15518

Maintenance database: postgres

Username: postgres

Kerberos authentication? ○

Password: ••••

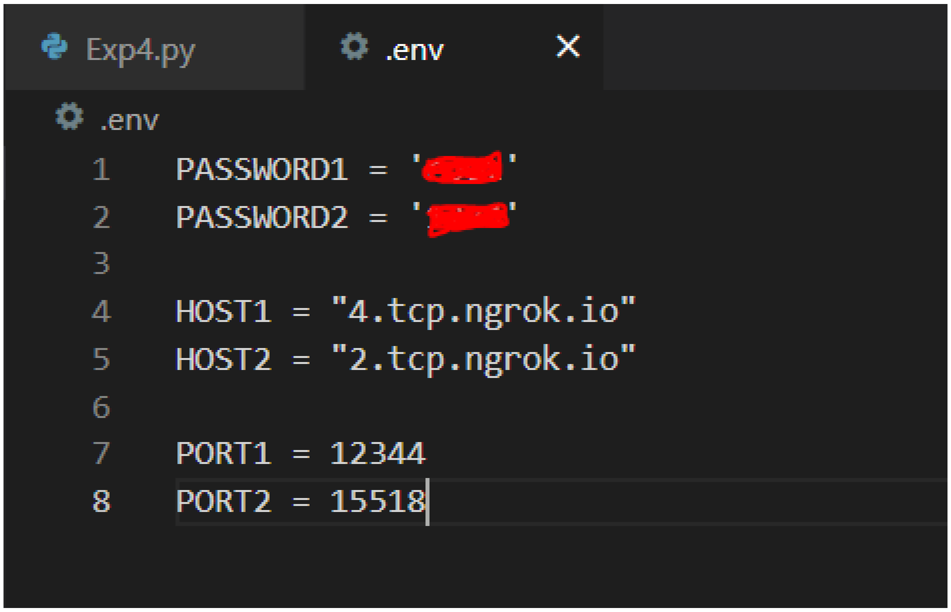
Save password? ●

Role:

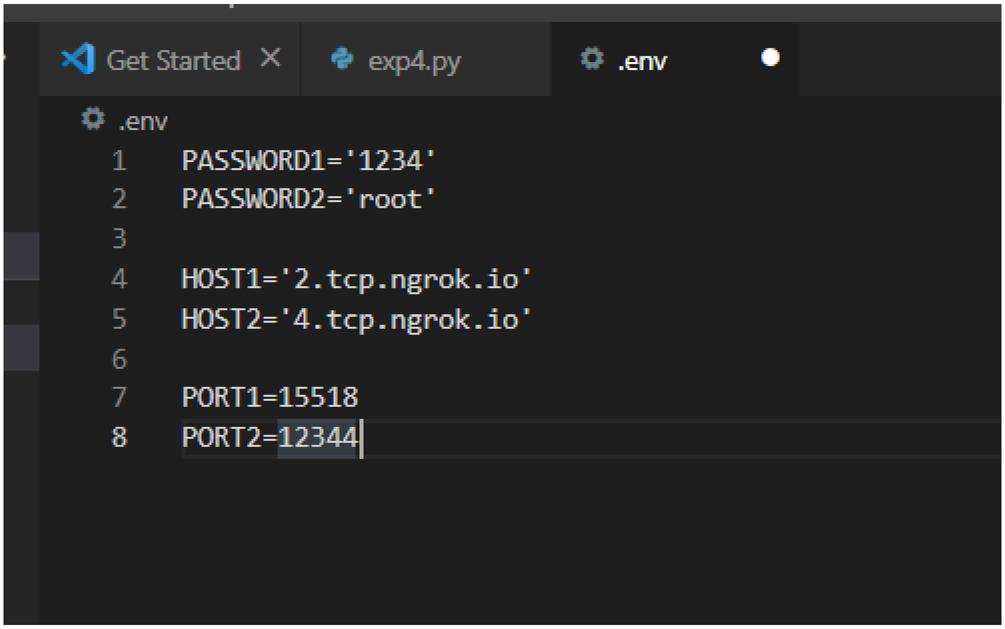
Service:

Repeated step 5 & 6

8. Provide appropriate credentials in the .env file which will later be accessed by python script.



```
PASSWORD1 = '[redacted]'
PASSWORD2 = '[redacted]'

HOST1 = "4.tcp.ngrok.io"
HOST2 = "2.tcp.ngrok.io"

PORT1 = 12344
PORT2 = 15518
```



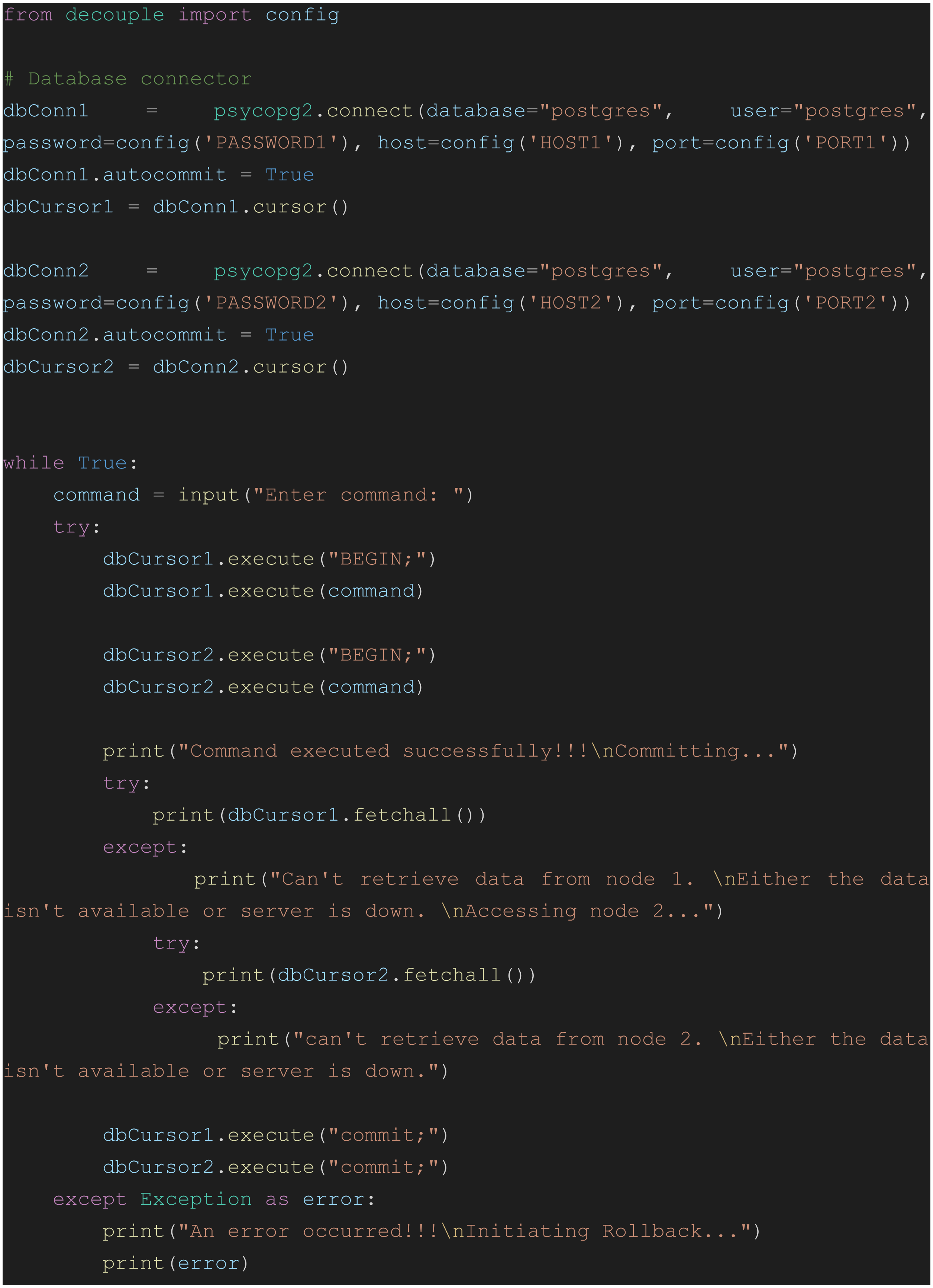
```
PASSWORD1='1234'
PASSWORD2='root'

HOST1='2.tcp.ngrok.io'
HOST2='4.tcp.ngrok.io'

PORT1=15518
PORT2=12344
```

Repeating the same step for second node

9. Run the python script. And when asked to enter a command, enter any command to manipulate data or schema. Observe the changes happening on both the nodes.

Python Script:

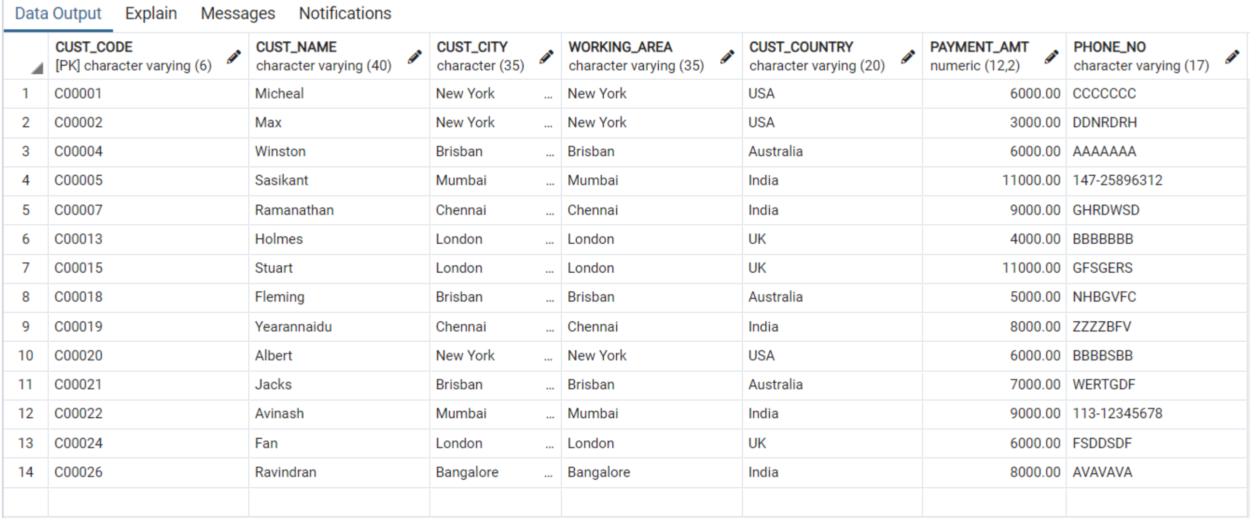
```
import psycopg2
```



```
from decouple import config

# Database connector
dbConn1 = psycopg2.connect(database="postgres", user="postgres",
password=config('PASSWORD1'), host=config('HOST1'), port=config('PORT1'))
dbConn1.autocommit = True
dbCursor1 = dbConn1.cursor()

dbConn2 = psycopg2.connect(database="postgres", user="postgres",
password=config('PASSWORD2'), host=config('HOST2'), port=config('PORT2'))
dbConn2.autocommit = True
dbCursor2 = dbConn2.cursor()


while True:
    command = input("Enter command: ")
    try:
        dbCursor1.execute("BEGIN;")
        dbCursor1.execute(command)

        dbCursor2.execute("BEGIN;")
        dbCursor2.execute(command)

        print("Command executed successfully!!!\nCommitting...")
        try:
            print(dbCursor1.fetchall())
        except:
            print("Can't retrieve data from node 1. \nEither the data
isn't available or server is down. \nAccessing node 2...")
            try:
                print(dbCursor2.fetchall())
            except:
                print("can't retrieve data from node 2. \nEither the data
isn't available or server is down.")

        dbCursor1.execute("commit;")
        dbCursor2.execute("commit;")
    except Exception as error:
        print("An error occurred!!!\nInitiating Rollback...")
        print(error)
        dbCursor1.execute("rollback;")
        dbCursor2.execute("rollback;")
```

Output:

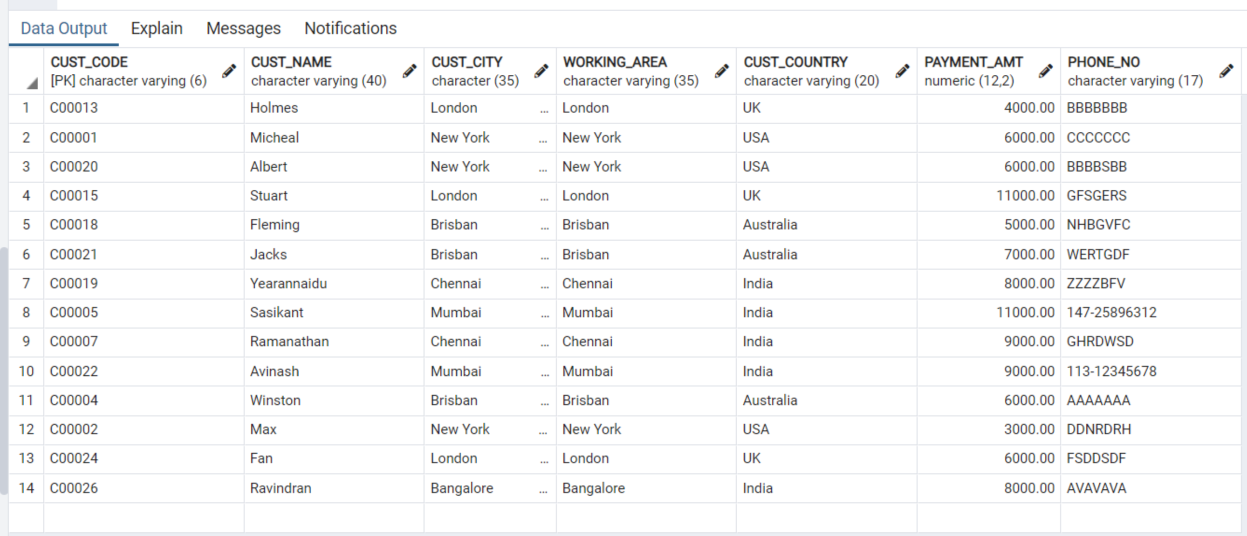
Data at node 1(original):



Data Output Explain Messages Notifications

	CUST_CODE [PK] character varying (6)	CUST_NAME character varying (40)	CUST_CITY character (35)	WORKING_AREA character varying (35)	CUST_COUNTRY character varying (20)	PAYMENT_AMT numeric (12,2)	PHONE_NO character varying (17)
1	C00001	Micheal	New York	New York	USA	6000.00	CCCCCCC
2	C00002	Max	New York	New York	USA	3000.00	DDNRDRH
3	C00004	Winston	Brisban	Brisban	Australia	6000.00	AAAAAAA
4	C00005	Sasikant	Mumbai	Mumbai	India	11000.00	147-25896312
5	C00007	Ramanathan	Chennai	Chennai	India	9000.00	GHRDWSD
6	C00013	Holmes	London	London	UK	4000.00	BBBBBBB
7	C00015	Stuart	London	London	UK	11000.00	GFSGERS
8	C00018	Fleming	Brisban	Brisban	Australia	5000.00	NHBGVFC
9	C00019	Yearannaidu	Chennai	Chennai	India	8000.00	ZZZZBFV
10	C00020	Albert	New York	New York	USA	6000.00	BBBBSBB
11	C00021	Jacks	Brisban	Brisban	Australia	7000.00	WERTGDF
12	C00022	Avinash	Mumbai	Mumbai	India	9000.00	113-12345678
13	C00024	Fan	London	London	UK	6000.00	FSDDSDF
14	C00026	Ravindran	Bangalore	Bangalore	India	8000.00	AVAVAVA

Data at node 2(original):

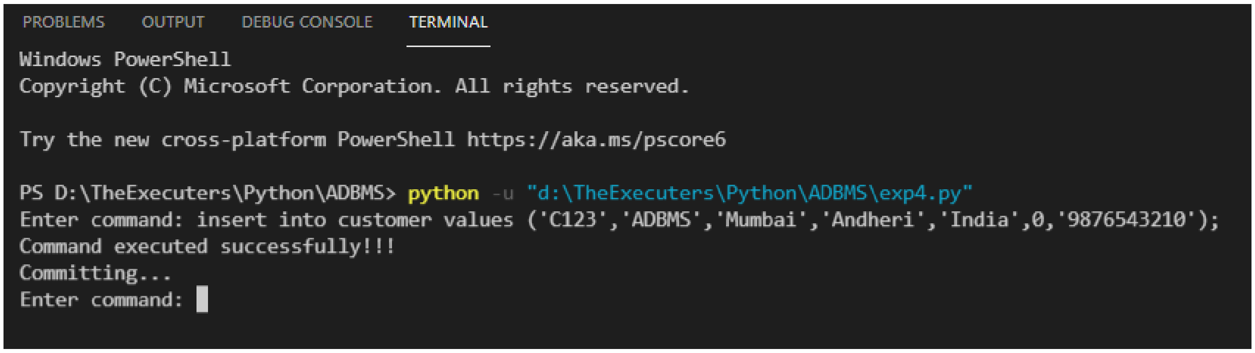


Data Output Explain Messages Notifications

	CUST_CODE [PK] character varying (6)	CUST_NAME character varying (40)	CUST_CITY character (35)	WORKING_AREA character varying (35)	CUST_COUNTRY character varying (20)	PAYMENT_AMT numeric (12,2)	PHONE_NO character varying (17)
1	C00013	Holmes	London	London	UK	4000.00	BBBBBBB
2	C00001	Micheal	New York	New York	USA	6000.00	CCCCCCC
3	C00020	Albert	New York	New York	USA	6000.00	BBBBSBB
4	C00015	Stuart	London	London	UK	11000.00	GFSGERS
5	C00018	Fleming	Brisban	Brisban	Australia	5000.00	NHBGVFC
6	C00021	Jacks	Brisban	Brisban	Australia	7000.00	WERTGDF
7	C00019	Yearannaidu	Chennai	Chennai	India	8000.00	ZZZZBFV
8	C00005	Sasikant	Mumbai	Mumbai	India	11000.00	147-25896312
9	C00007	Ramanathan	Chennai	Chennai	India	9000.00	GHRDWSD
10	C00022	Avinash	Mumbai	Mumbai	India	9000.00	113-12345678
11	C00004	Winston	Brisban	Brisban	Australia	6000.00	AAAAAAA
12	C00002	Max	New York	New York	USA	3000.00	DDNRDRH
13	C00024	Fan	London	London	UK	6000.00	FSDDSDF
14	C00026	Ravindran	Bangalore	Bangalore	India	8000.00	AVAVAVA

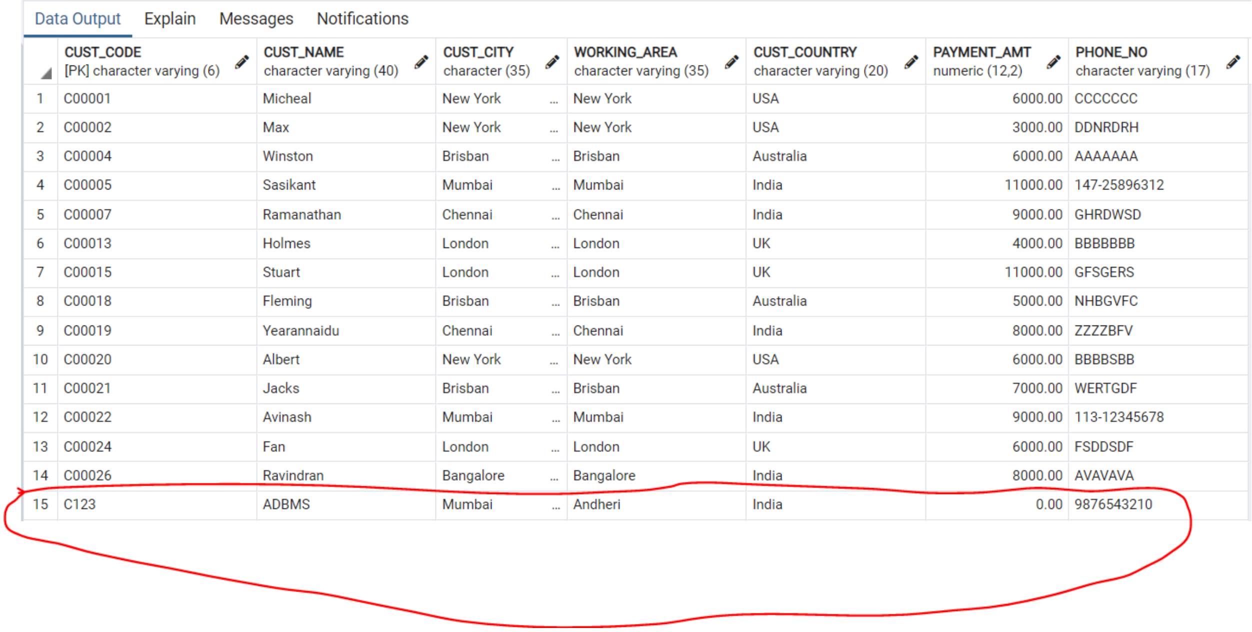
Insert command executed at node 2:

insert into customer (‘C123’, ‘ADBMS’, ‘Mumbai’, ‘Andheri’, ‘India’, 0, ‘9876543210’);



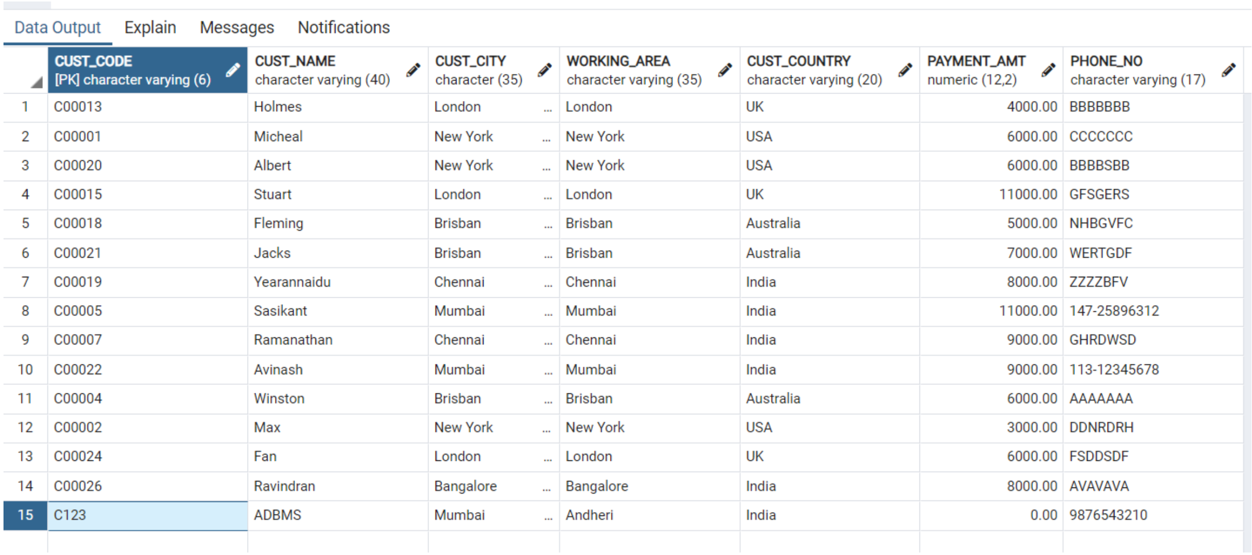
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\TheExecuters\Python\ADBMS> python -u "d:\TheExecuters\Python\ADBMS\exp4.py"
Enter command: insert into customer values ('C123','ADBMS','Mumbai','Andheri','India',0,'9876543210');
Command executed successfully!!!
Committing...
Enter command: 
```

@ node1:



	CUST_CODE [PK] character varying (6)	CUST_NAME character varying (40)	CUST_CITY character (35)	WORKING_AREA character varying (35)	CUST_COUNTRY character varying (20)	PAYMENT_AMT numeric (12,2)	PHONE_NO character varying (17)
1	C00001	Micheal	New York ...	New York	USA	6000.00	CCCCCCC
2	C00002	Max	New York ...	New York	USA	3000.00	DDNRDRH
3	C00004	Winston	Brisban ...	Brisban	Australia	6000.00	AAAAAAA
4	C00005	Sasikant	Mumbai ...	Mumbai	India	11000.00	147-25896312
5	C00007	Ramanathan	Chennai ...	Chennai	India	9000.00	GHRDWSD
6	C00013	Holmes	London ...	London	UK	4000.00	BBBBBBB
7	C00015	Stuart	London ...	London	UK	11000.00	GFSGERS
8	C00018	Fleming	Brisban ...	Brisban	Australia	5000.00	NHBGVFC
9	C00019	Yearannaidu	Chennai ...	Chennai	India	8000.00	ZZZZBFV
10	C00020	Albert	New York ...	New York	USA	6000.00	BBBBSBB
11	C00021	Jacks	Brisban ...	Brisban	Australia	7000.00	WERTGDF
12	C00022	Avinash	Mumbai ...	Mumbai	India	9000.00	113-12345678
13	C00024	Fan	London ...	London	UK	6000.00	FSDDSDF
14	C00026	Ravindran	Bangalore ...	Bangalore	India	8000.00	AVAVAVA
15	C123	ADBMS	Mumbai ...	Andheri	India	0.00	9876543210

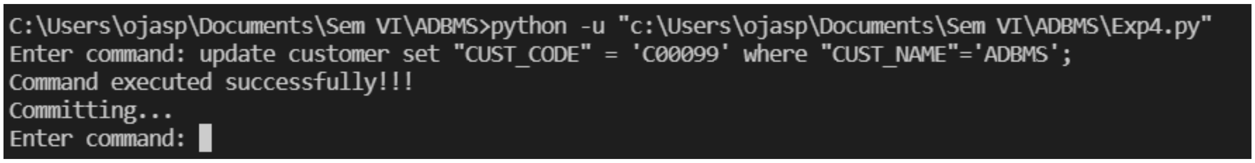
@ node2:



	CUST_CODE [PK] character varying (6)	CUST_NAME character varying (40)	CUST_CITY character (35)	WORKING_AREA character varying (35)	CUST_COUNTRY character varying (20)	PAYMENT_AMT numeric (12,2)	PHONE_NO character varying (17)
1	C00013	Holmes	London ...	London	UK	4000.00	BBBBBBB
2	C00001	Micheal	New York ...	New York	USA	6000.00	CCCCCCC
3	C00020	Albert	New York ...	New York	USA	6000.00	BBBBSBB
4	C00015	Stuart	London ...	London	UK	11000.00	GFSGERS
5	C00018	Fleming	Brisban ...	Brisban	Australia	5000.00	NHBGVFC
6	C00021	Jacks	Brisban ...	Brisban	Australia	7000.00	WERTGDF
7	C00019	Yearannaidu	Chennai ...	Chennai	India	8000.00	ZZZZBFV
8	C00005	Sasikant	Mumbai ...	Mumbai	India	11000.00	147-25896312
9	C00007	Ramanathan	Chennai ...	Chennai	India	9000.00	GHRDWSD
10	C00022	Avinash	Mumbai ...	Mumbai	India	9000.00	113-12345678
11	C00004	Winston	Brisban ...	Brisban	Australia	6000.00	AAAAAAA
12	C00002	Max	New York ...	New York	USA	3000.00	DDNRDRH
13	C00024	Fan	London ...	London	UK	6000.00	FSDDSDF
14	C00026	Ravindran	Bangalore ...	Bangalore	India	8000.00	AVAVAVA
15	C123	ADBMS	Mumbai ...	Andheri	India	0.00	9876543210

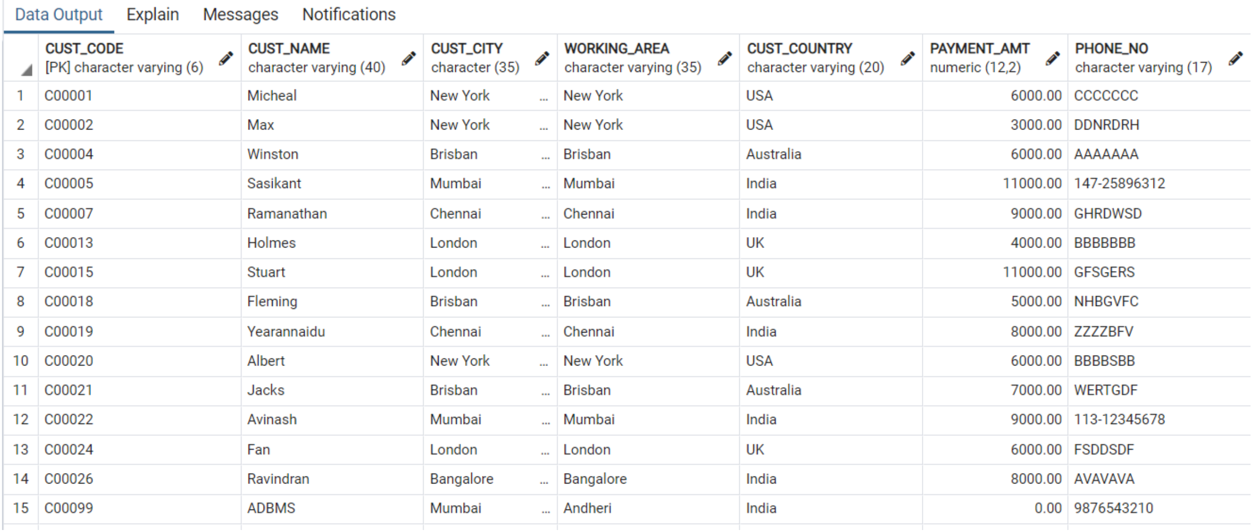
<u>Update command executed at node 1</u>:

update customer set "CUST_CODE" = 'C00099' where "CUST_NAME"='ADBMS';



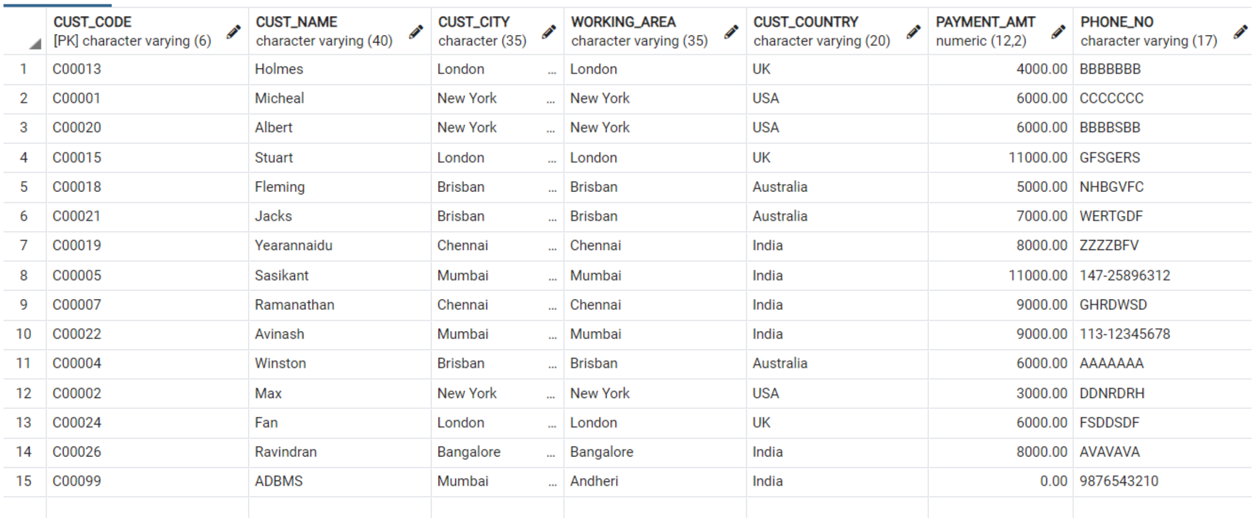
```
C:\Users\ojasp\Documents\Sem VI\ADBMS>python -u "c:\Users\ojasp\Documents\Sem VI\ADBMS\Exp4.py"
Enter command: update customer set "CUST_CODE" = 'C00099' where "CUST_NAME"='ADBMS';
Command executed successfully!!!
Committing...
Enter command: 
```

@ node1



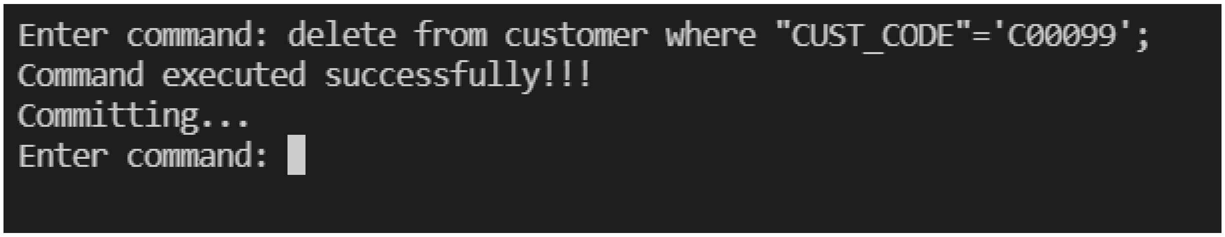
	CUST_CODE [PK] character varying (6)	CUST_NAME character varying (40)	CUST_CITY character (35)	WORKING_AREA character varying (35)	CUST_COUNTRY character varying (20)	PAYMENT_AMT numeric (12,2)	PHONE_NO character varying (17)
1	C00001	Micheal	New York ...	New York	USA	6000.00	CCCCCCC
2	C00002	Max	New York ...	New York	USA	3000.00	DDNRDRH
3	C00004	Winston	Brisban ...	Brisban	Australia	6000.00	AAAAAAA
4	C00005	Sasikant	Mumbai ...	Mumbai	India	11000.00	147-25896312
5	C00007	Ramanathan	Chennai ...	Chennai	India	9000.00	GHRDWSD
6	C00013	Holmes	London ...	London	UK	4000.00	BBBBBBB
7	C00015	Stuart	London ...	London	UK	11000.00	GFSGERS
8	C00018	Fleming	Brisban ...	Brisban	Australia	5000.00	NHBGVFC
9	C00019	Yearannaidu	Chennai ...	Chennai	India	8000.00	ZZZZBFV
10	C00020	Albert	New York ...	New York	USA	6000.00	BBBBSBB
11	C00021	Jacks	Brisban ...	Brisban	Australia	7000.00	WERTGDF
12	C00022	Avinash	Mumbai ...	Mumbai	India	9000.00	113-12345678
13	C00024	Fan	London ...	London	UK	6000.00	FSDDSDF
14	C00026	Ravindran	Bangalore ...	Bangalore	India	8000.00	AVAVAVA
15	C00099	ADBMS	Mumbai ...	Andheri	India	0.00	9876543210

@node2



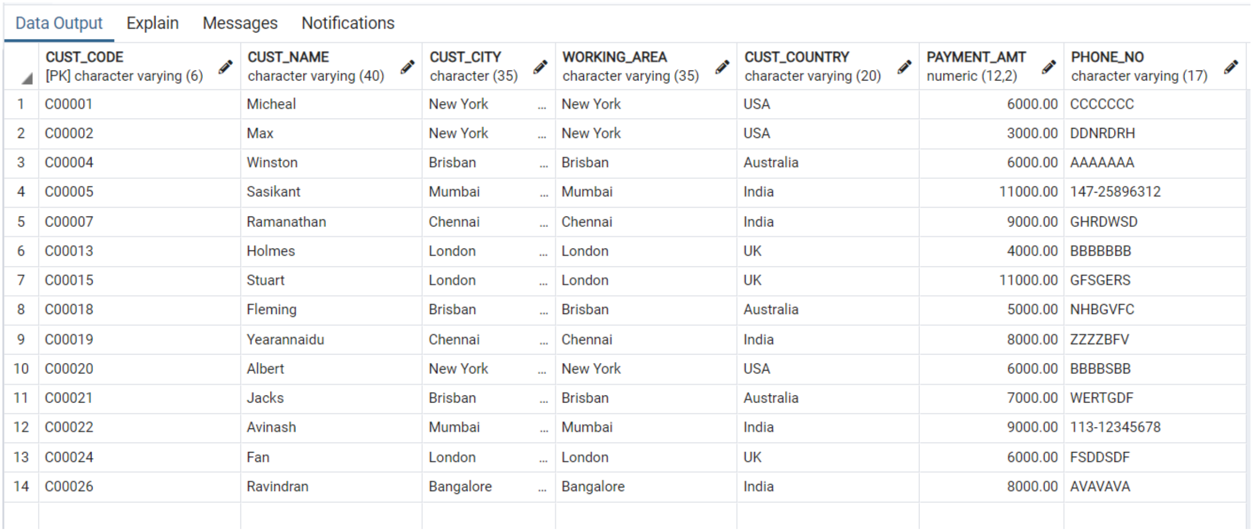
	CUST_CODE [PK] character varying (6)	CUST_NAME character varying (40)	CUST_CITY character (35)	WORKING_AREA character varying (35)	CUST_COUNTRY character varying (20)	PAYMENT_AMT numeric (12,2)	PHONE_NO character varying (17)
1	C00013	Holmes	London ...	London	UK	4000.00	BBBBBBB
2	C00001	Micheal	New York ...	New York	USA	6000.00	CCCCCCC
3	C00020	Albert	New York ...	New York	USA	6000.00	BBBBSBB
4	C00015	Stuart	London ...	London	UK	11000.00	GFSGERS
5	C00018	Fleming	Brisban ...	Brisban	Australia	5000.00	NHBGVFC
6	C00021	Jacks	Brisban ...	Brisban	Australia	7000.00	WERTGDF
7	C00019	Yearannaidu	Chennai ...	Chennai	India	8000.00	ZZZZBFV
8	C00005	Sasikant	Mumbai ...	Mumbai	India	11000.00	147-25896312
9	C00007	Ramanathan	Chennai ...	Chennai	India	9000.00	GHRDWSD
10	C00022	Avinash	Mumbai ...	Mumbai	India	9000.00	113-12345678
11	C00004	Winston	Brisban ...	Brisban	Australia	6000.00	AAAAAAA
12	C00002	Max	New York ...	New York	USA	3000.00	DDNRDRH
13	C00024	Fan	London ...	London	UK	6000.00	FSDDSDF
14	C00026	Ravindran	Bangalore ...	Bangalore	India	8000.00	AVAVAVA
15	C00099	ADBMS	Mumbai ...	Andheri	India	0.00	9876543210

Drop command executed at a 3rd user's end:
Delete from customer where "CUST_CODE"='C00099';



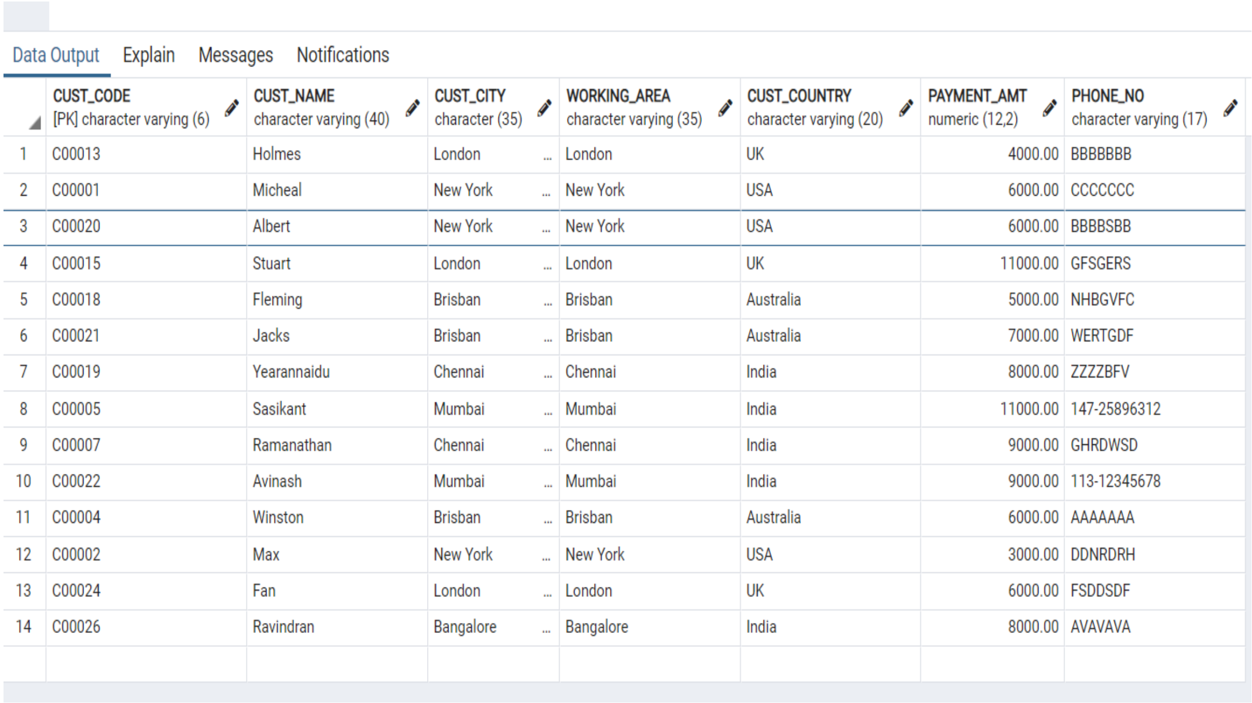
```
Enter command: delete from customer where "CUST_CODE"='C00099';
Command executed successfully!!!
Committing...
Enter command: █
```

@ node 1:



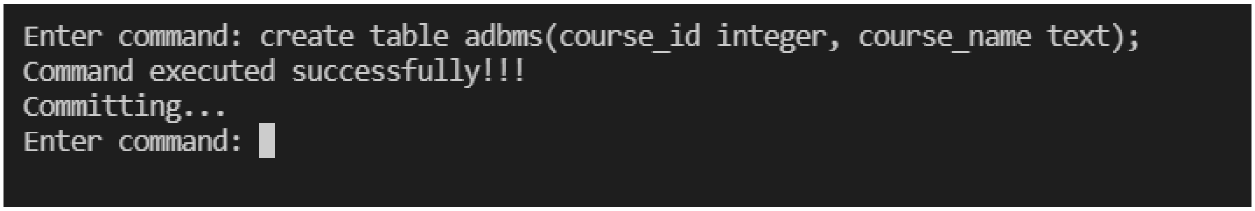
	CUST_CODE [PK] character varying (6)	CUST_NAME character varying (40)	CUST_CITY character (35)	WORKING_AREA character varying (35)	CUST_COUNTRY character varying (20)	PAYMENT_AMT numeric (12,2)	PHONE_NO character varying (17)
1	C00001	Micheal	New York ...	New York	USA	6000.00	CCCCCCC
2	C00002	Max	New York ...	New York	USA	3000.00	DDNRDRH
3	C00004	Winston	Brisban ...	Brisban	Australia	6000.00	AAAAAAA
4	C00005	Sasikant	Mumbai ...	Mumbai	India	11000.00	147-25896312
5	C00007	Ramanathan	Chennai ...	Chennai	India	9000.00	GHRDWSD
6	C00013	Holmes	London ...	London	UK	4000.00	BBBBBBB
7	C00015	Stuart	London ...	London	UK	11000.00	GFSGERS
8	C00018	Fleming	Brisban ...	Brisban	Australia	5000.00	NHBGVFC
9	C00019	Yearannaidu	Chennai ...	Chennai	India	8000.00	ZZZZBFV
10	C00020	Albert	New York ...	New York	USA	6000.00	BBBBSBB
11	C00021	Jacks	Brisban ...	Brisban	Australia	7000.00	WERTGDF
12	C00022	Avinash	Mumbai ...	Mumbai	India	9000.00	113-12345678
13	C00024	Fan	London ...	London	UK	6000.00	FSDDSDF
14	C00026	Ravindran	Bangalore ...	Bangalore	India	8000.00	AVAVAVA

@node2

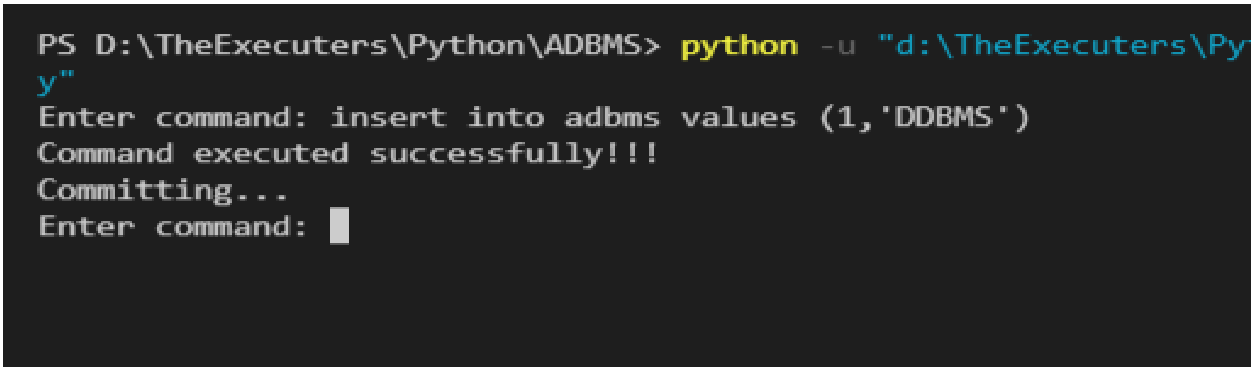


	CUST_CODE [PK] character varying (6)	CUST_NAME character varying (40)	CUST_CITY character (35)	WORKING_AREA character varying (35)	CUST_COUNTRY character varying (20)	PAYMENT_AMT numeric (12,2)	PHONE_NO character varying (17)
1	C00013	Holmes	London	London	UK	4000.00	BBBBBBB
2	C00001	Micheal	New York	New York	USA	6000.00	CCCCCCC
3	C00020	Albert	New York	New York	USA	6000.00	BBBBSBB
4	C00015	Stuart	London	London	UK	11000.00	GFSGERS
5	C00018	Fleming	Brisban	Brisban	Australia	5000.00	NHBGVFC
6	C00021	Jacks	Brisban	Brisban	Australia	7000.00	WERTGDF
7	C00019	Yearannaidu	Chennai	Chennai	India	8000.00	ZZZZBFV
8	C00005	Sasikant	Mumbai	Mumbai	India	11000.00	147-25896312
9	C00007	Ramanathan	Chennai	Chennai	India	9000.00	GHRDWSD
10	C00022	Avinash	Mumbai	Mumbai	India	9000.00	113-12345678
11	C00004	Winston	Brisban	Brisban	Australia	6000.00	AAAAAAA
12	C00002	Max	New York	New York	USA	3000.00	DDNRDRH
13	C00024	Fan	London	London	UK	6000.00	FSDDSDF
14	C00026	Ravindran	Bangalore	Bangalore	India	8000.00	AVAVAVA

Create new relation and insert data:

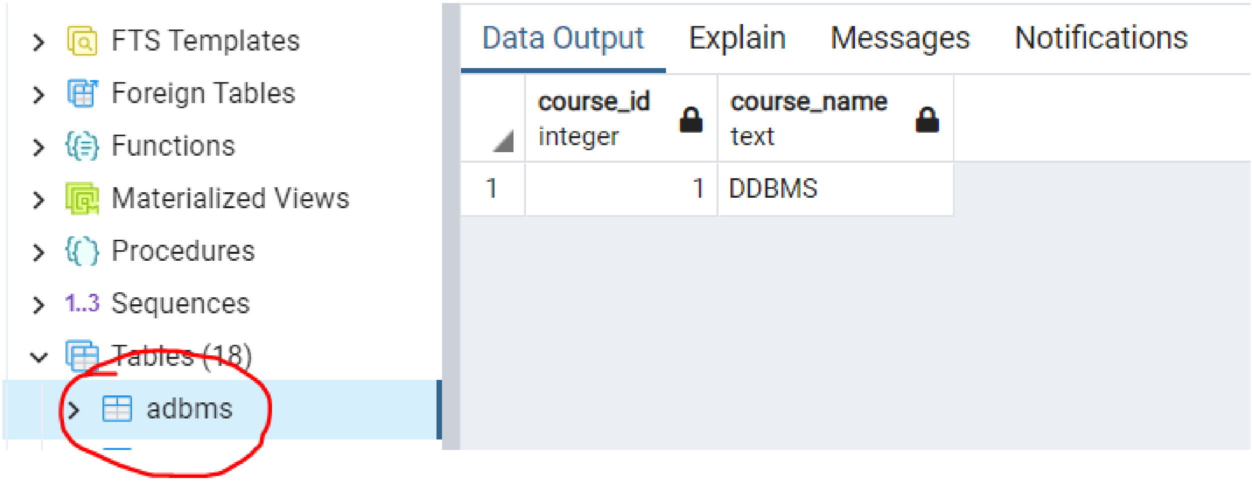


```
Enter command: create table adbms(course_id integer, course_name text);
Command executed successfully!!!
Committing...
Enter command:
```



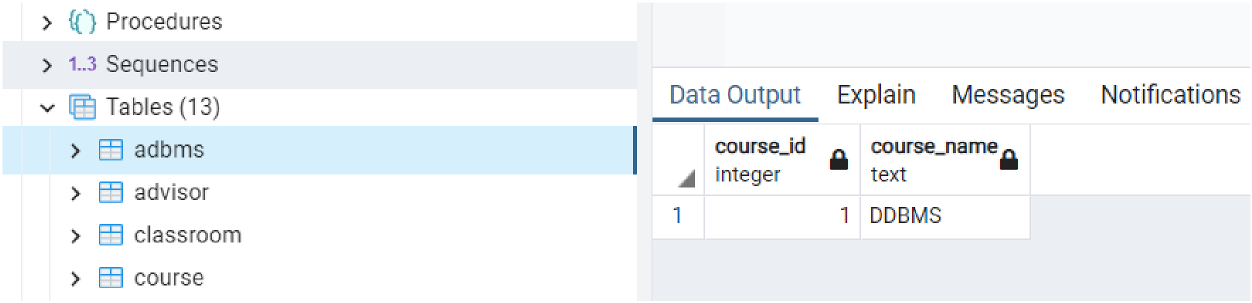
```
PS D:\TheExecuters\Python\ADBMS> python -u "d:\TheExecuters\Py
y"
Enter command: insert into adbms values (1,'DDBMS')
Command executed successfully!!!
Committing...
Enter command:
```

@ node 1:



	course_id integer	course_name text
1	1	DDBMS

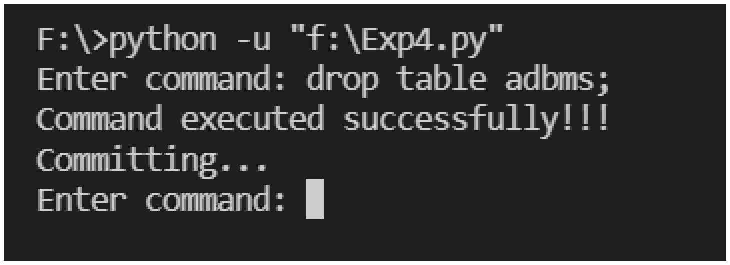
@ node 2:



	course_id integer	course_name text
1	1	DDBMS

Drop created relation:

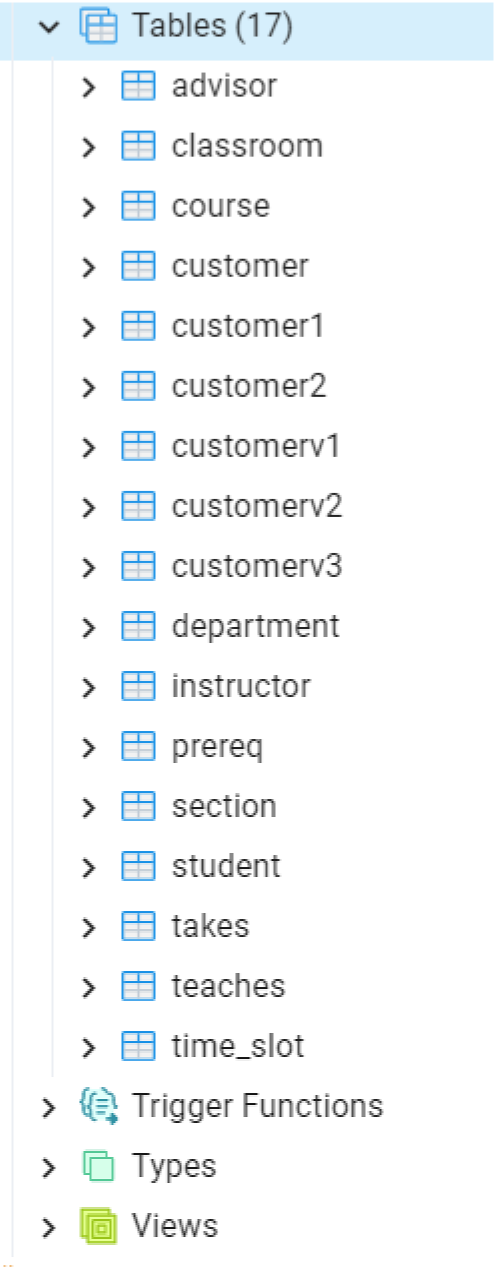
Drop table adbms;



```
F:\>python -u "f:\Exp4.py"
Enter command: drop table adbms;
Command executed successfully!!!
Committing...
Enter command: 
```

This shows that the table has been dropped successfully.

@ node 1:



Tables (17)

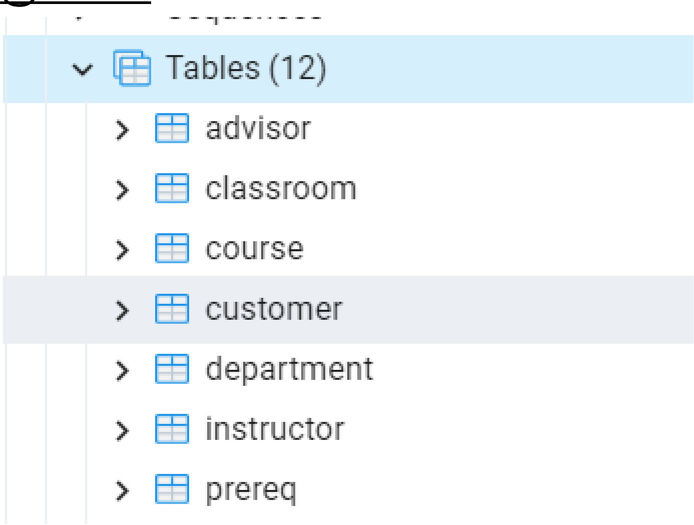
- advisor
- classroom
- course
- customer
- customer1
- customer2
- customerv1
- customerv2
- customerv3
- department
- instructor
- prereq
- section
- student
- takes
- teaches
- time_slot

Trigger Functions

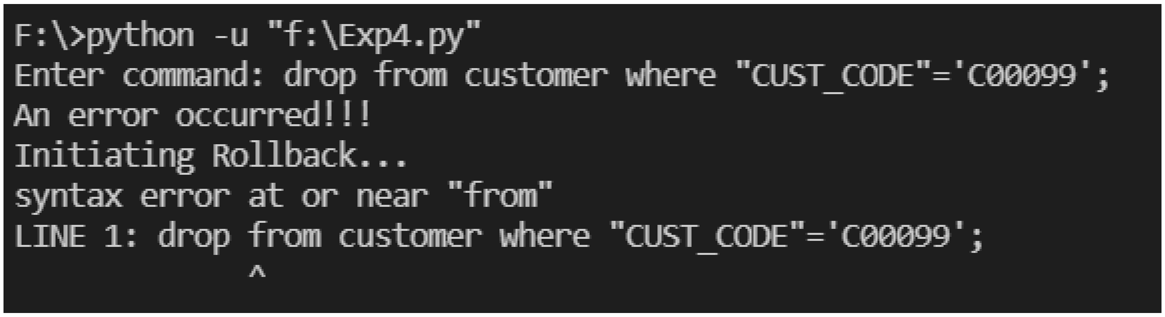
Types

Views

@node2:



Error display:



```
F:\>python -u "f:\Exp4.py"
Enter command: drop from customer where "CUST_CODE"='C00099';
An error occurred!!!
Initiating Rollback...
syntax error at or near "from"
LINE 1: drop from customer where "CUST_CODE"='C00099';
             ^
```

Conclusion:

The data replication ensures reliability and availability of the data. Even if one node fails, it can access the data from another node. Also, for successful implementation of replication, data has to be securely exposed to the internet. This ensures availability of data to authorised people anytime anywhere. Maintaining consistency is a challenge while replicating the database.