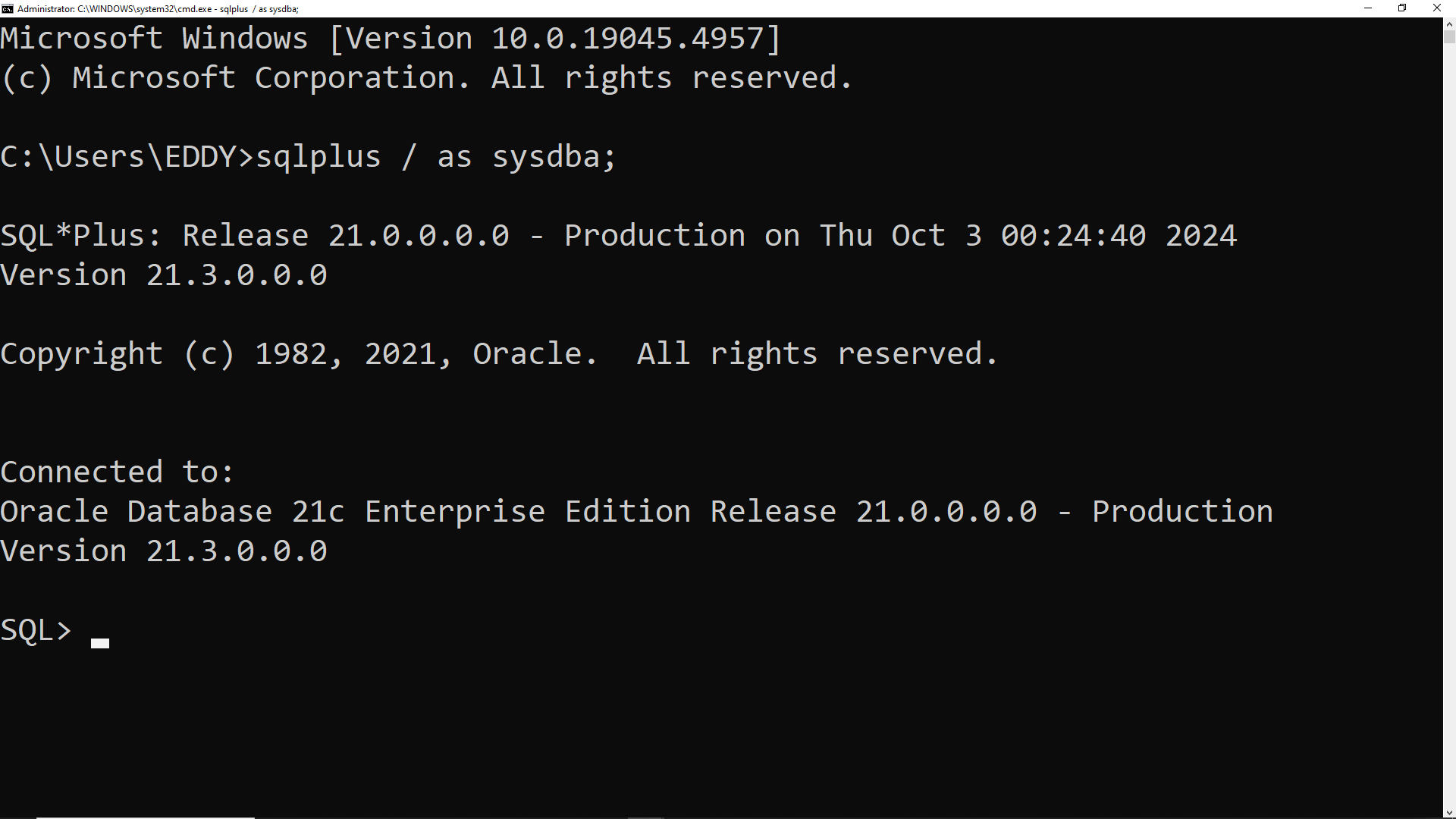
**PL/SQL ASSIGNMENT II**

**Database Creation, Deletion & EOM**

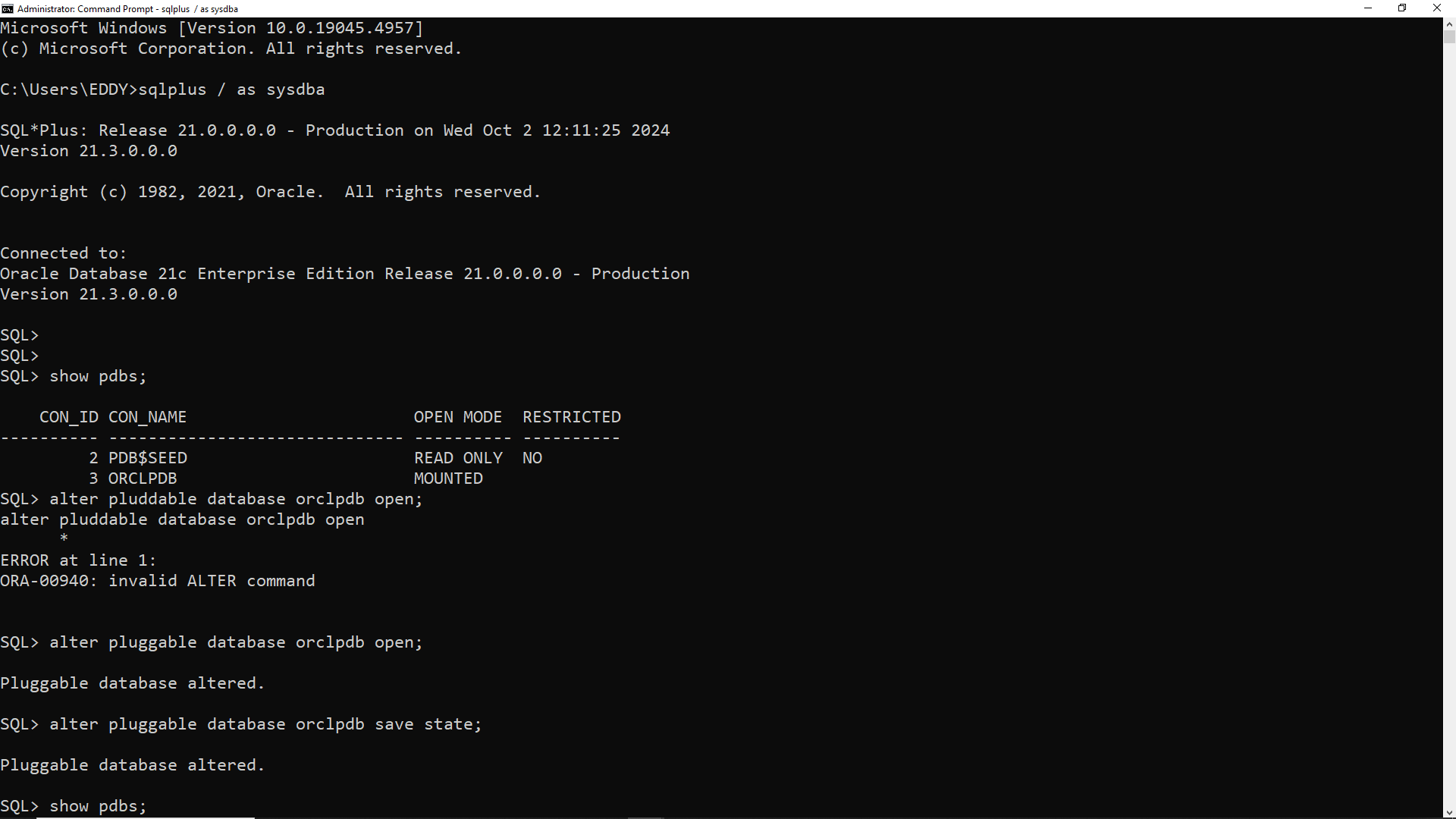
To create or delete a new pluggable database, we first open CMD and connect to an Oracle database using SQL\*Plus as the SYS user with SYSDBA privileges.

By Simply typing ‘sqlplus sys as sysdba;’or ‘sqlplus / as sysdba;’, as shown in a picture below.

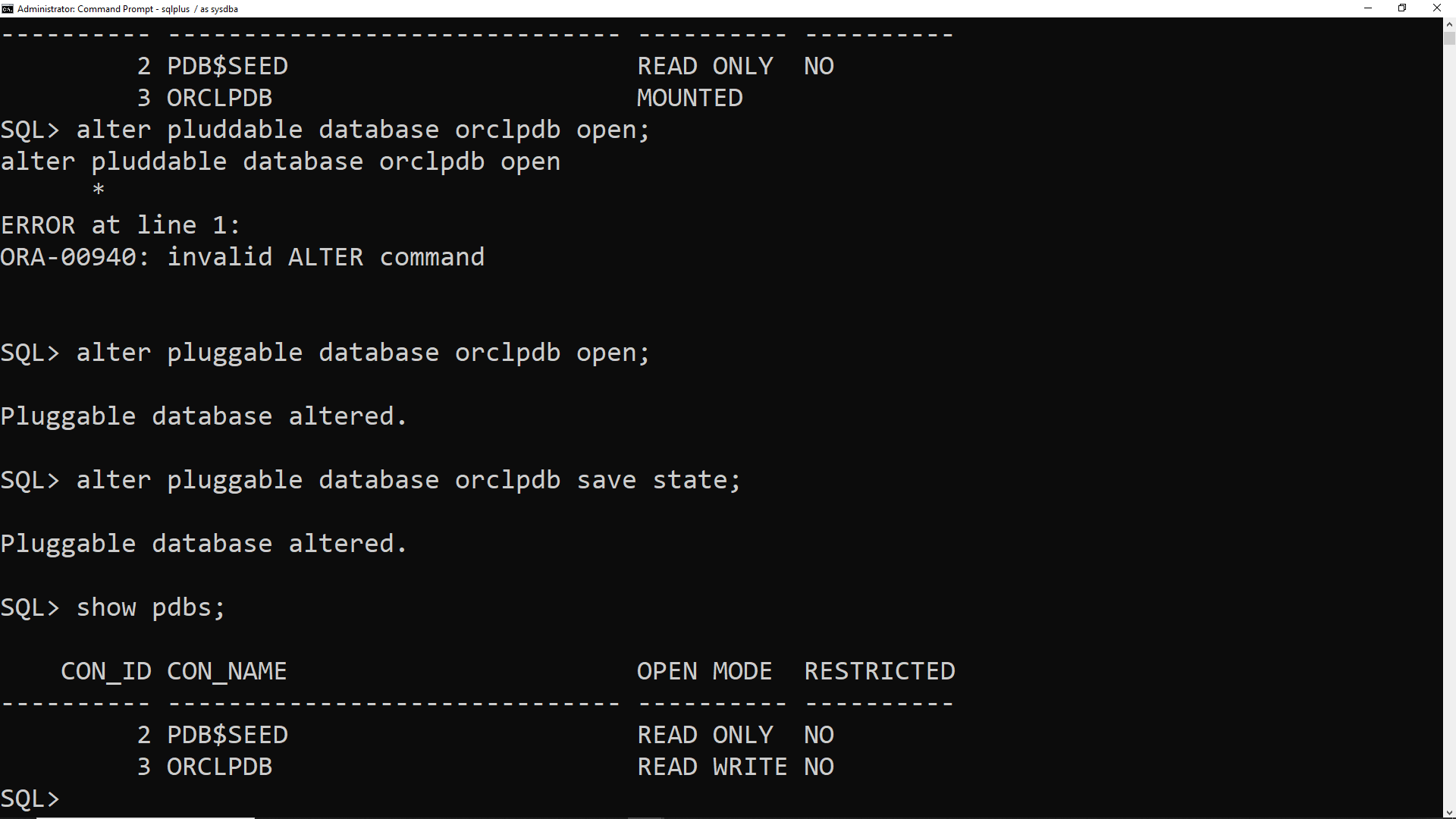


**1.CREATING A NEW PLUGGABLE DATABASES.**

Before creating a new pluggable database(pdb), we fist need to know some important information as container databases name, pluggable database name, and the path for our container database, and other information by typing: show pdbs; (please always remember to add a semi coloron at the end of your query).



You may ask yourself that why do ORCLPDB is mounted? It means that pdb created is not yet open for use. To open it we use “alter pluggable database orclpdb open;” then you save the state that time it automatically opens the orclpdb is read and write in open mode as shown below.



After this we need to display the path of our container database in order to create our new Pluggable database and we use pdbseed .

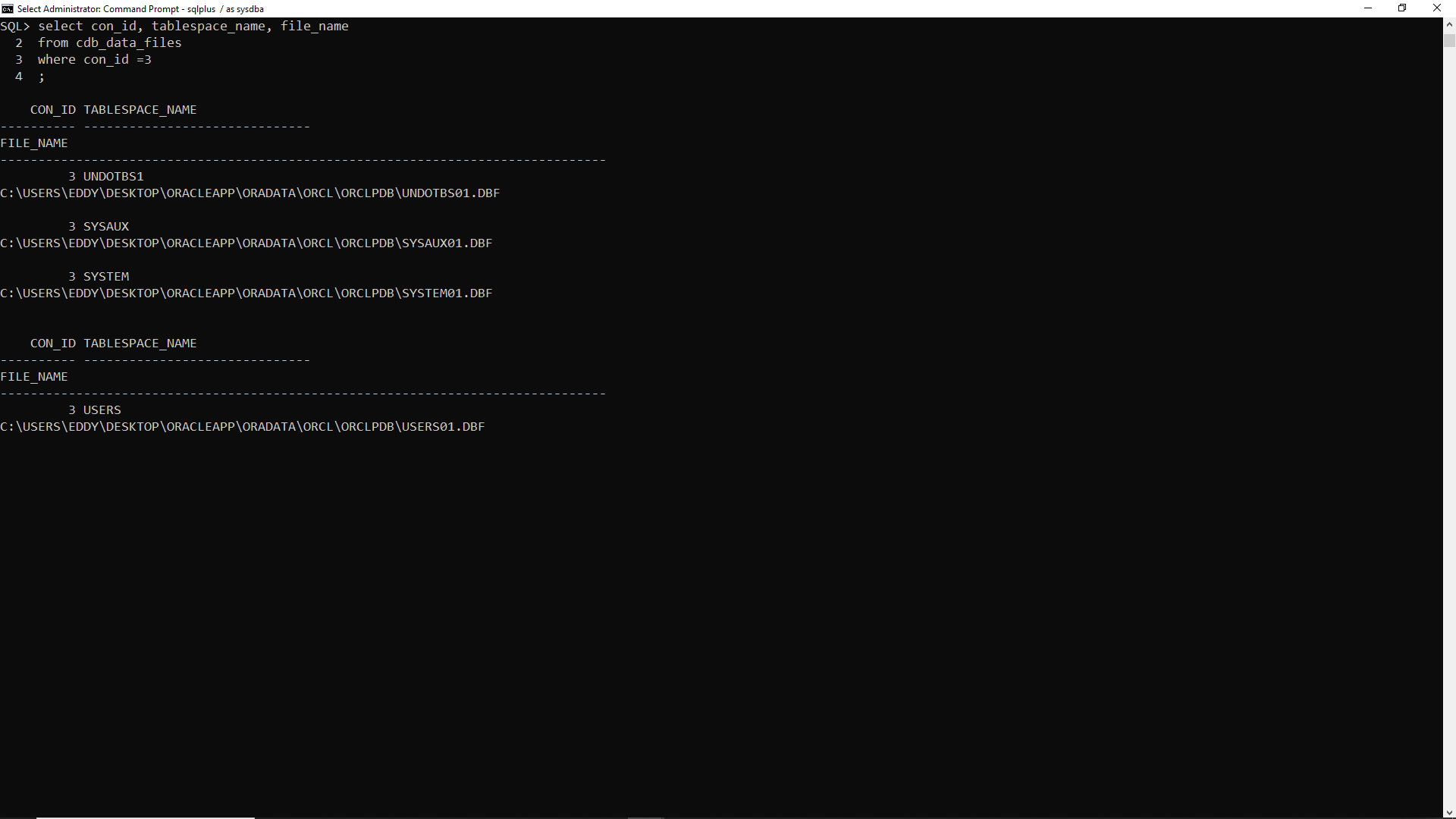
**Pdbseed**: this is the template we base one when creating our new pluggable database.

This query retrieves the tablespace names and file names of data

files for the Pluggable Database (PDB).

. We use this query “SELECT CON\_ID, TABLESPACE\_NAME, FILE\_NAME FROM

CDB\_DATA\_FILES WHERE CON\_ID = 3;” the result is shown below

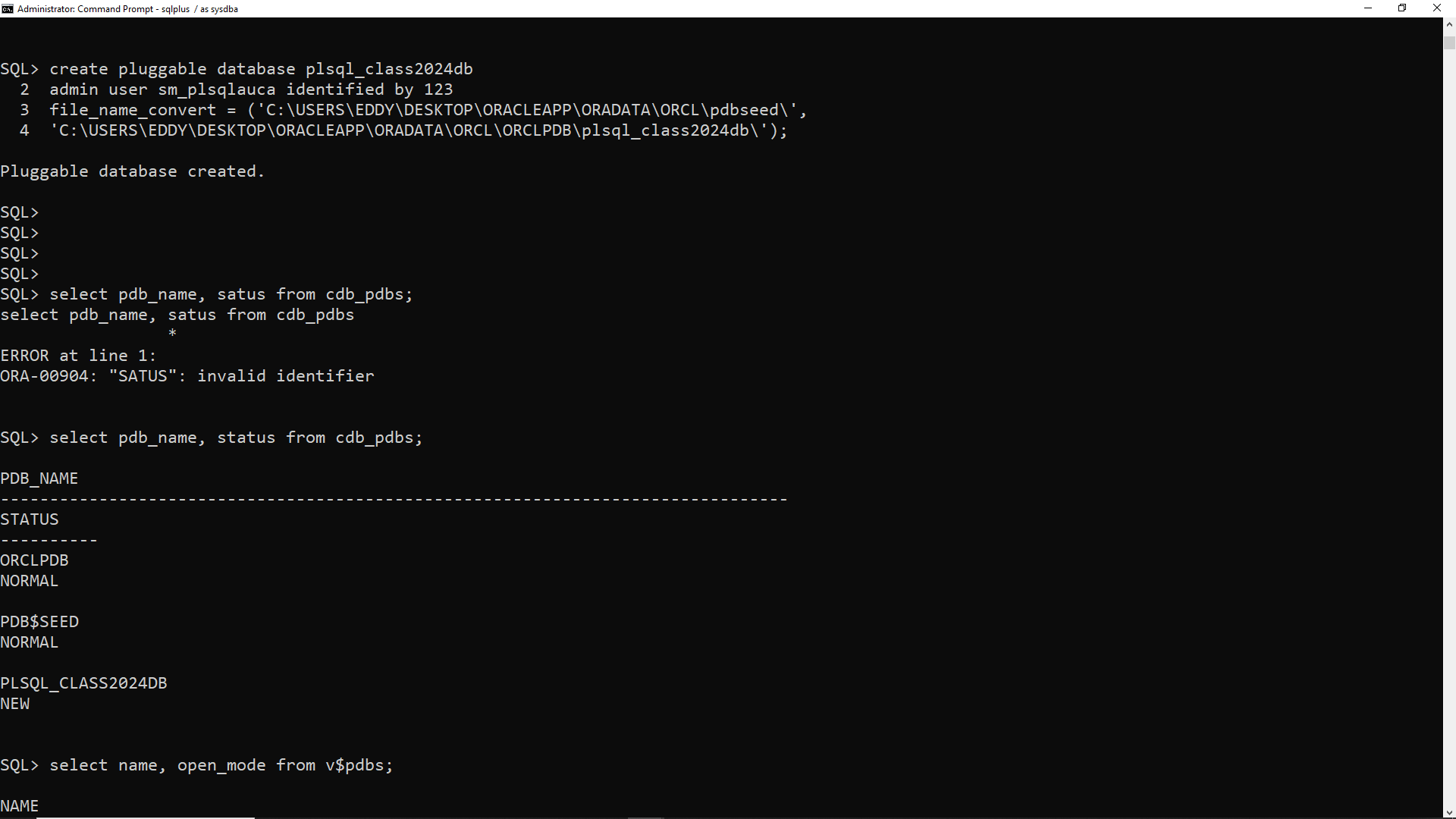


At this point we can now create our pluggable database using pdbseed, we can successfully achieve this by using this query : create pluggable database plsql\_class2024db

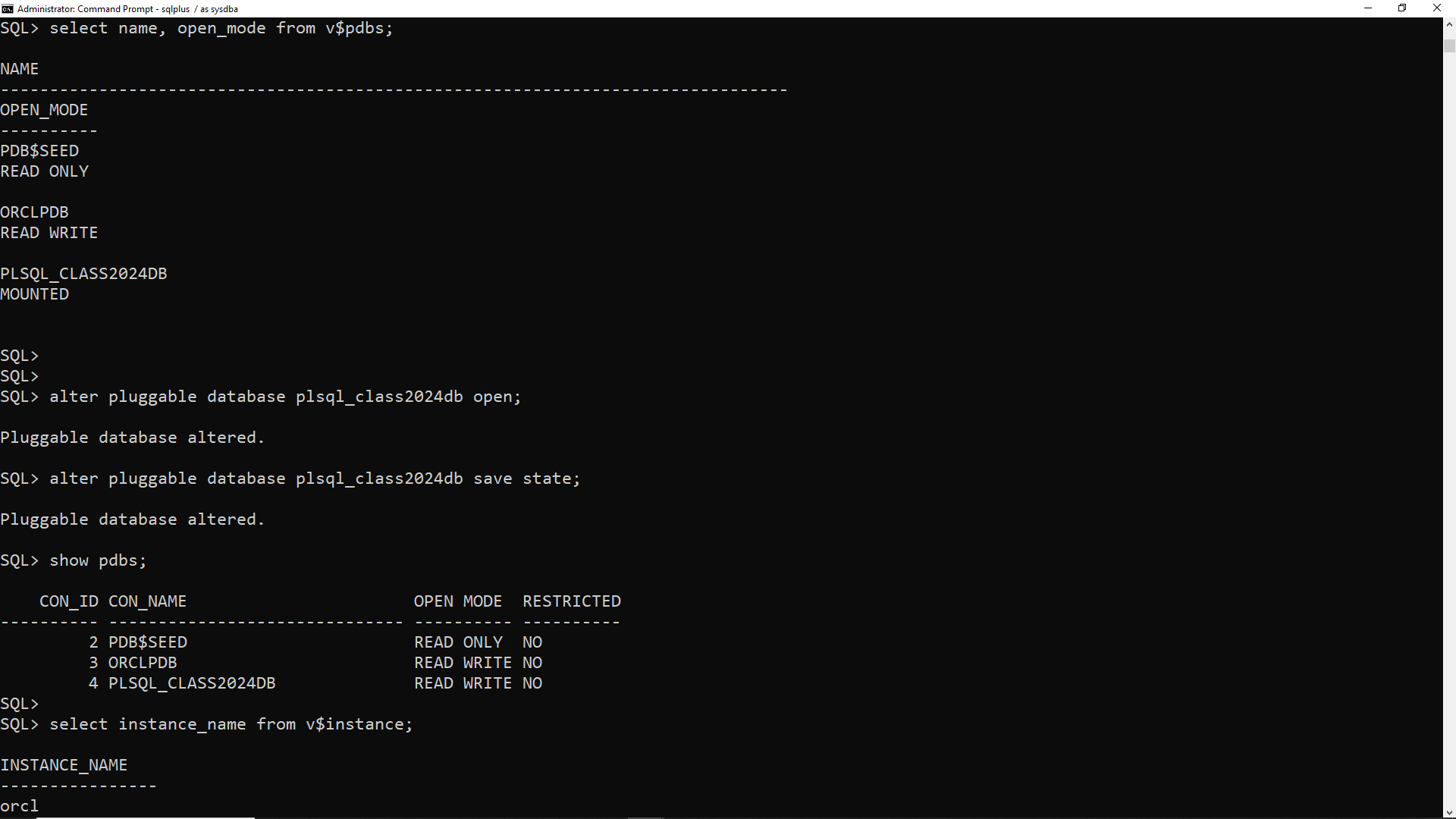
admin user sm\_plsqlauca identified by 123

file\_name\_convert = ('C:\USERS\EDDY\DESKTOP\ORACLEAPP\ORADATA\ORCL\pdbseed\',

'C:\USERS\EDDY\DESKTOP\ORACLEAPP\ORADATA\ORCL\ORCLPDB\plsql\_class2024db’);



As you can see the pluggable database that we created has status new because it has not yet opened.



To open it we repeat the same thing like” alter pluggable database plsql\_class2024db open;” then you save the state that time it automatically opens the plsql\_class2024db is read and write in open mode as shown in above image.

**CHECKING THE CURRENT INSTANCE NAME**

Checking the current instance name by using this query “alter session set container =PLSQL\_CLASS2024DB;”, and we use the name of our pluggable database.

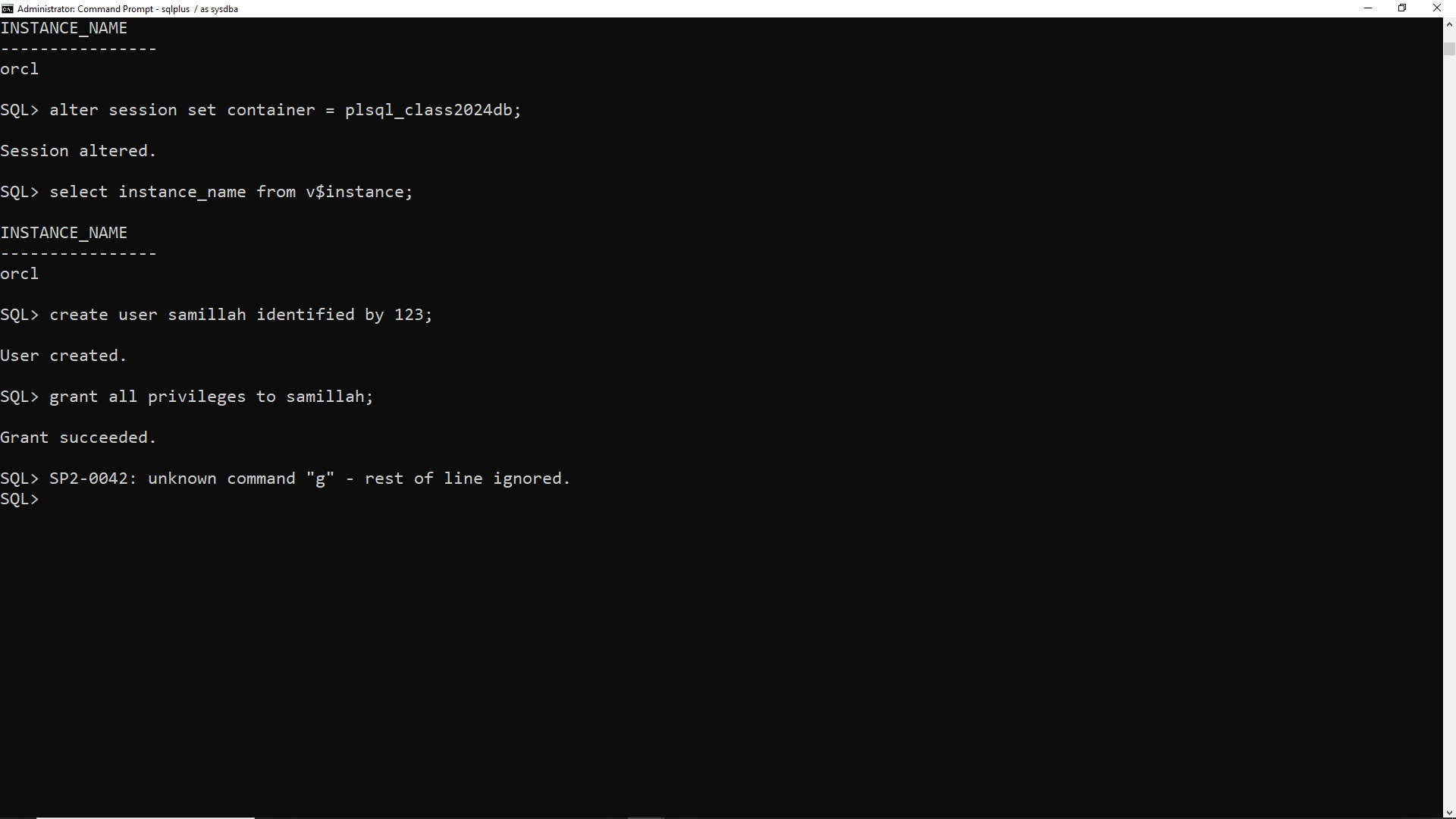
At this step we can create a new and later we grant (give) the created use all privileges,

To create a user, we use this query: (for example)

create user samillah identified by 123;

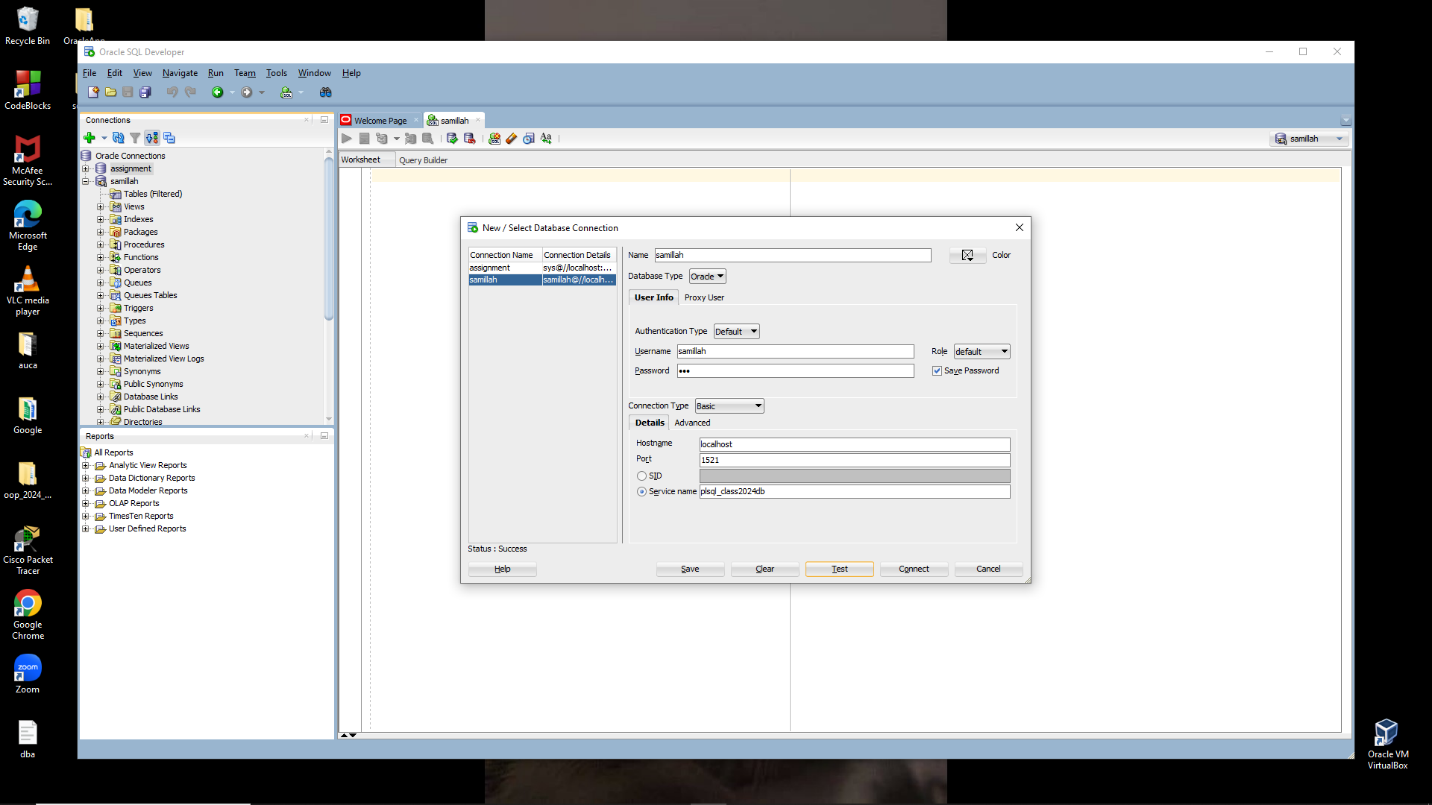
grant all privileges to samillah;

This enables us to be able to create new connection in sql developer by changing the service name. All have been shown in the picture below.



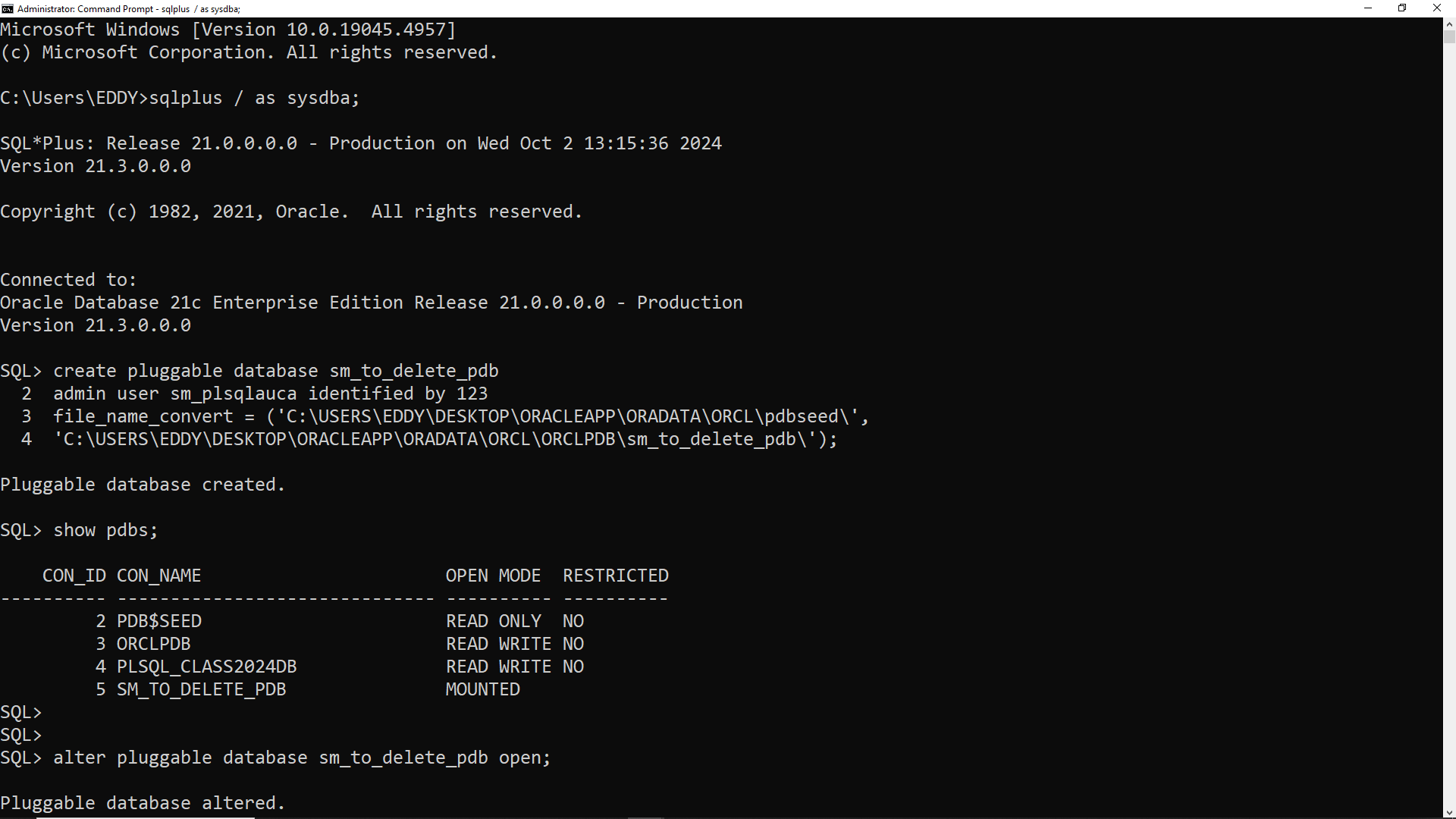
**Connecting to Oracle PDB through Oracle SQL Developer for the Newly Created User**

To check if our new pluggable databse can connect to Oracle PDB SQL Developer through PLSQL\_CLASS2024DB, we first open SQL Developer >create a new connection > ..

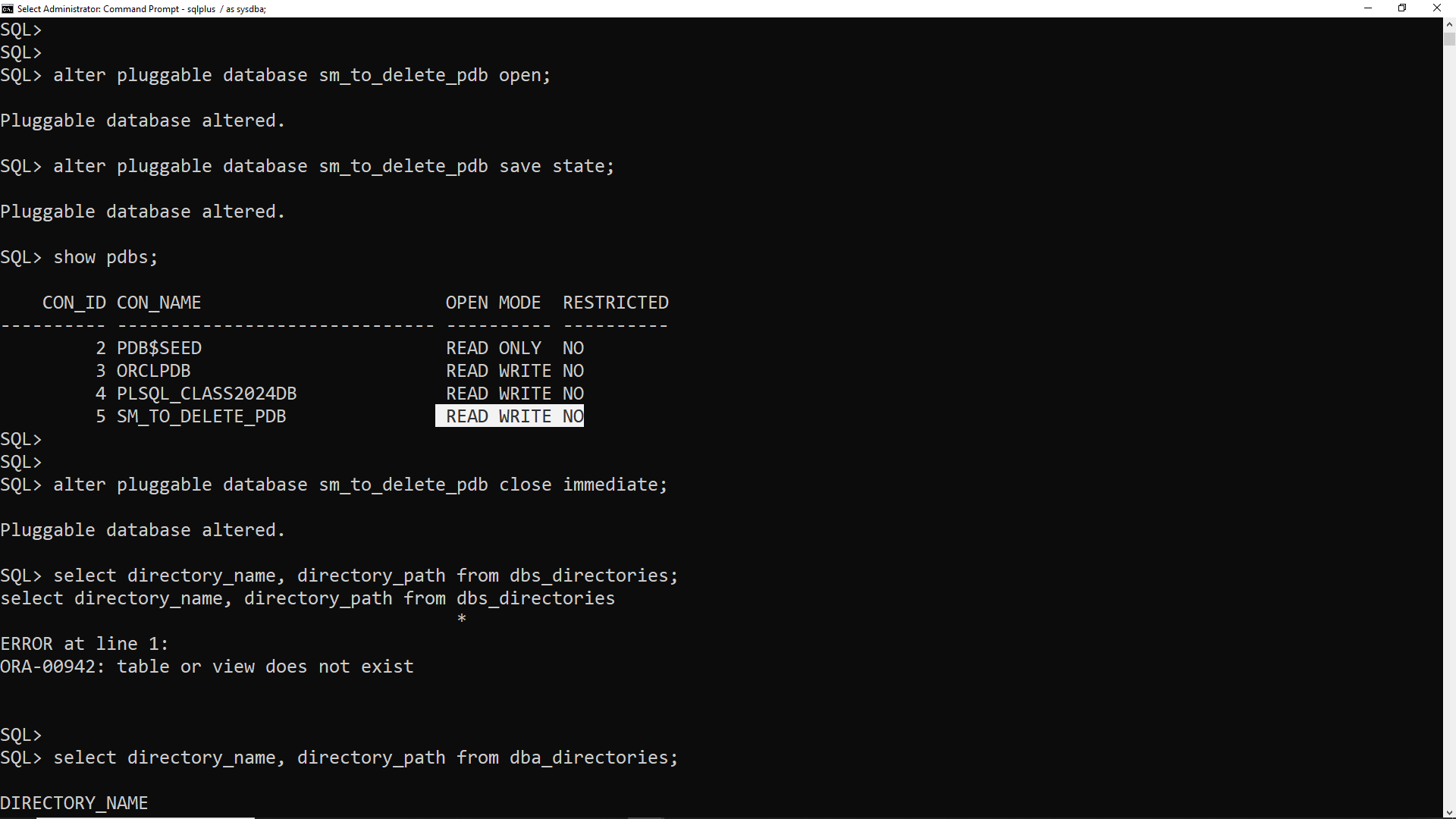
On name : samillah , Authentication type is default , username : samillah (the user we created) , password :123 , then select service name and type PLSQL\_CLASS2024DB then test it if the status is success, save it and connect.

**2.Delete the pluggable databases**

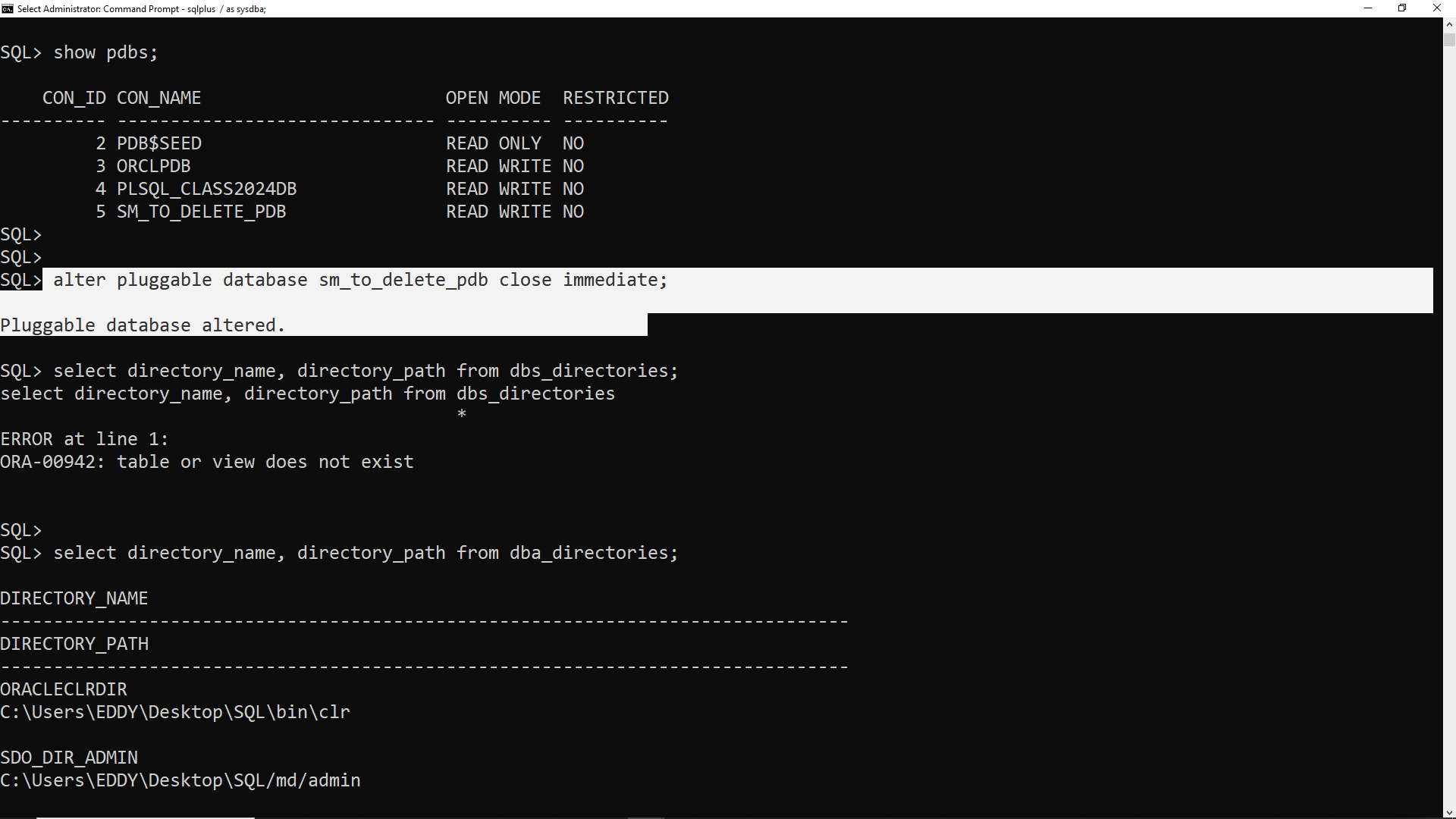
I am going to create the pluggable database that I will delete because the pluggable database I have I can delete them because I will use them. To create pluggable database we repeat what we have done earlier but change the name of the pluggable database . I will name it sm\_to\_delete\_pdb. As show below.



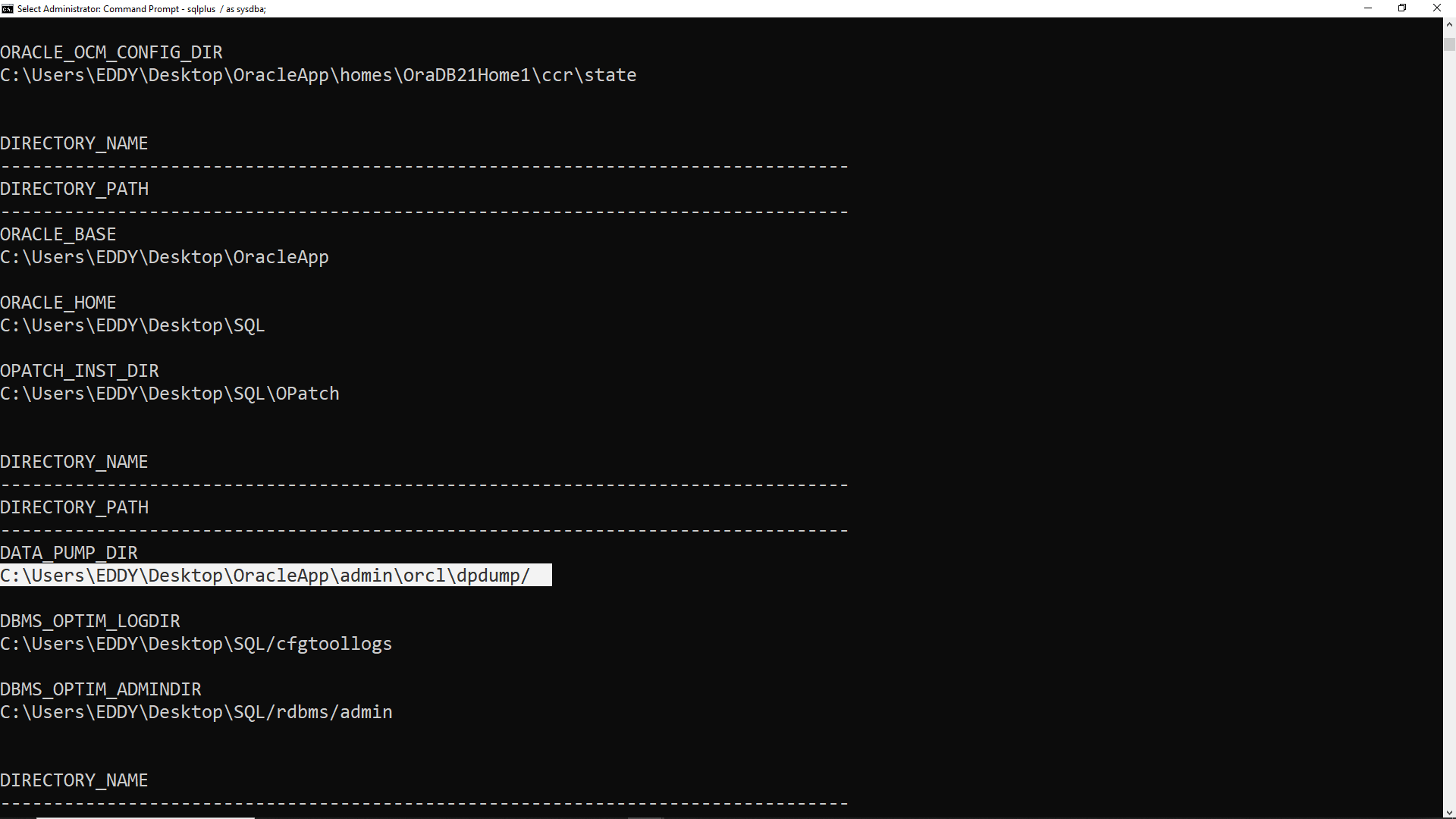
We all know what mounted mean. To open it, you can run the command ALTER PLUGGABLE DATABASE sm\_to delete\_pdb OPEN; When you do this, the database will transition to an open state, allowing read and write access as shown below



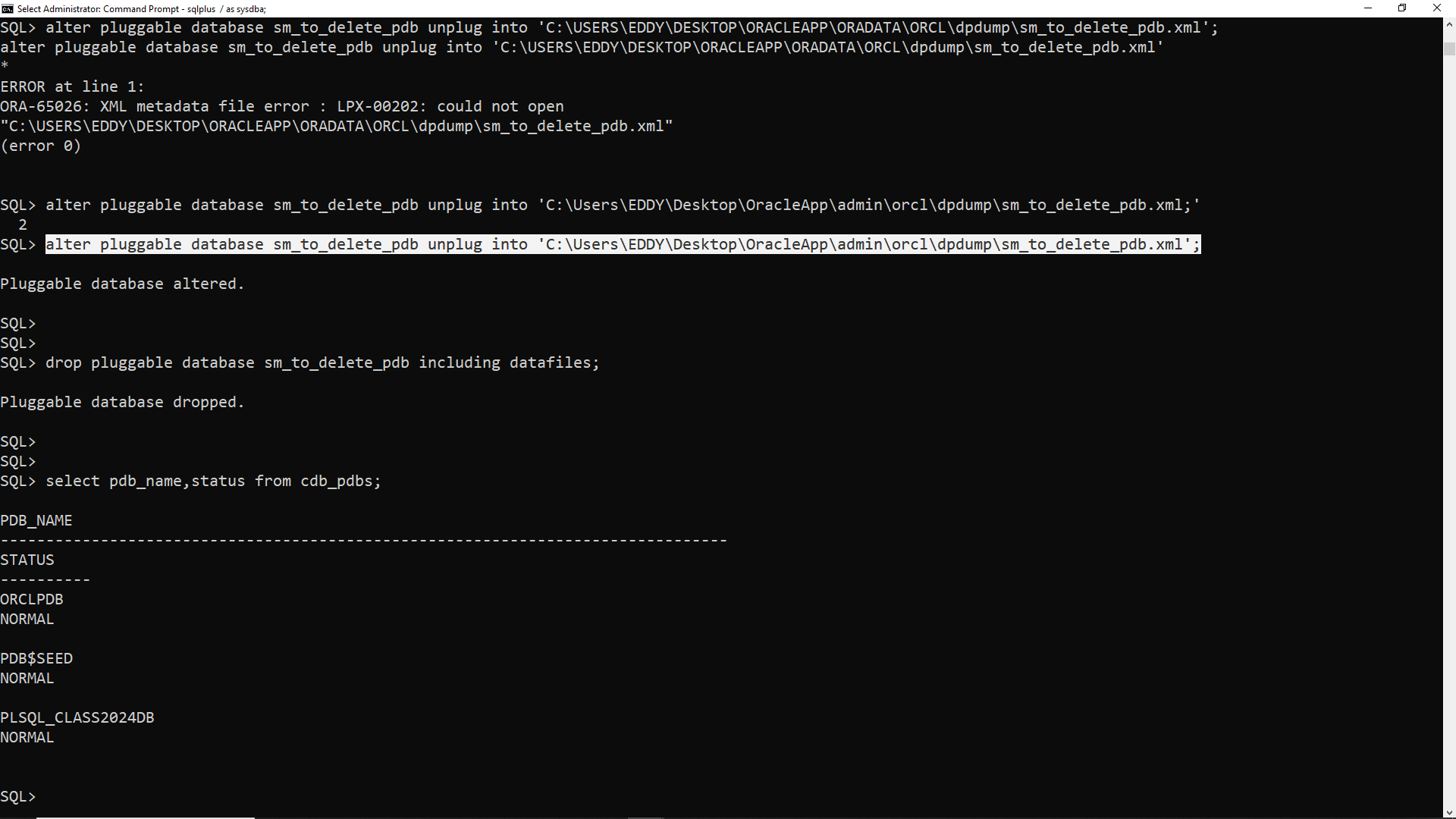
Before deleting or dropping a pluggable database , it needs to be closed by using this query:” alter pluggable database sm\_to\_delete\_pdb close immediate;”



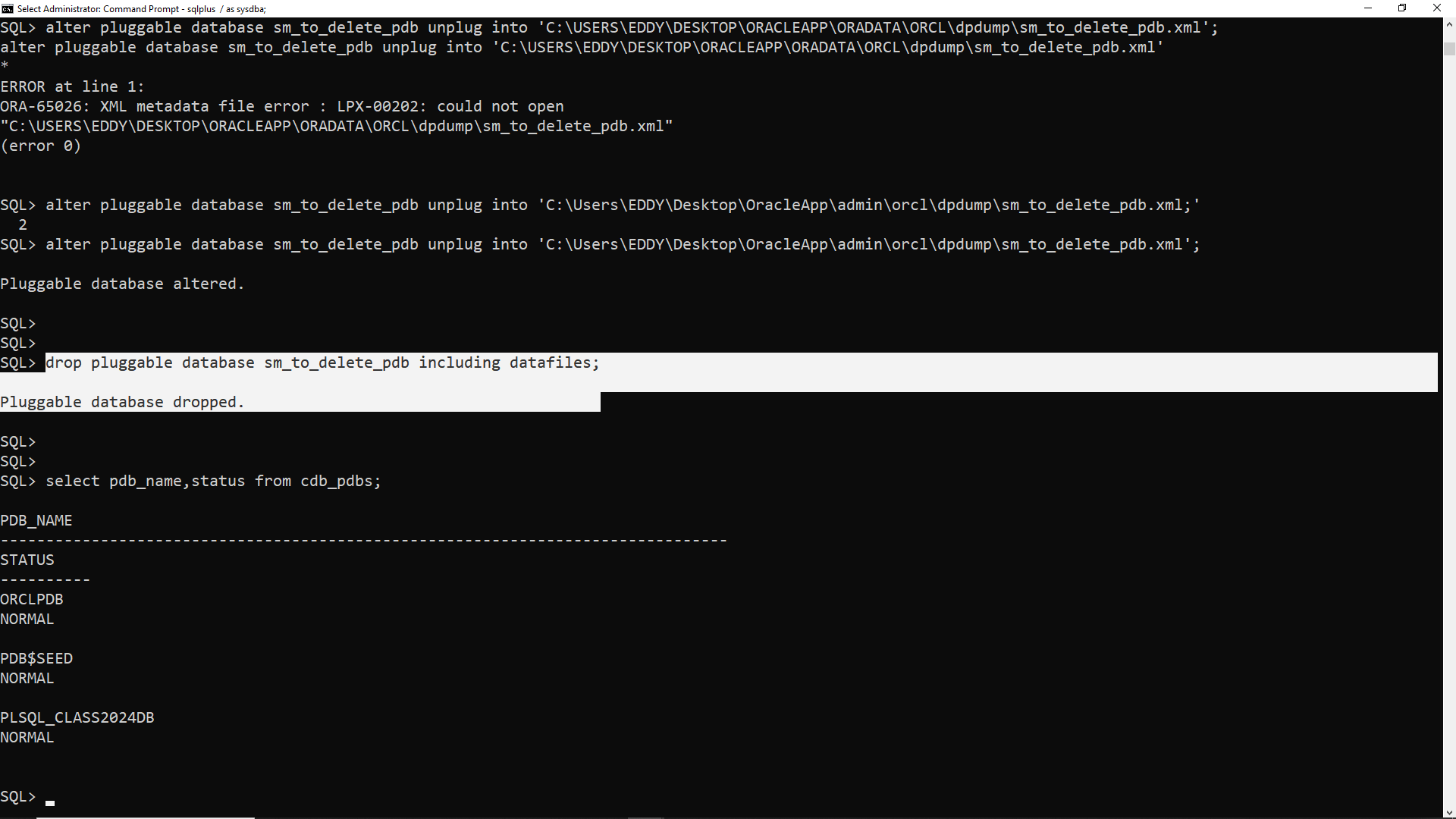
Then find or set the correct directory path for unplugging the sm\_to delete\_pdb using this query: select directory\_name, directory\_path from dba\_directories;



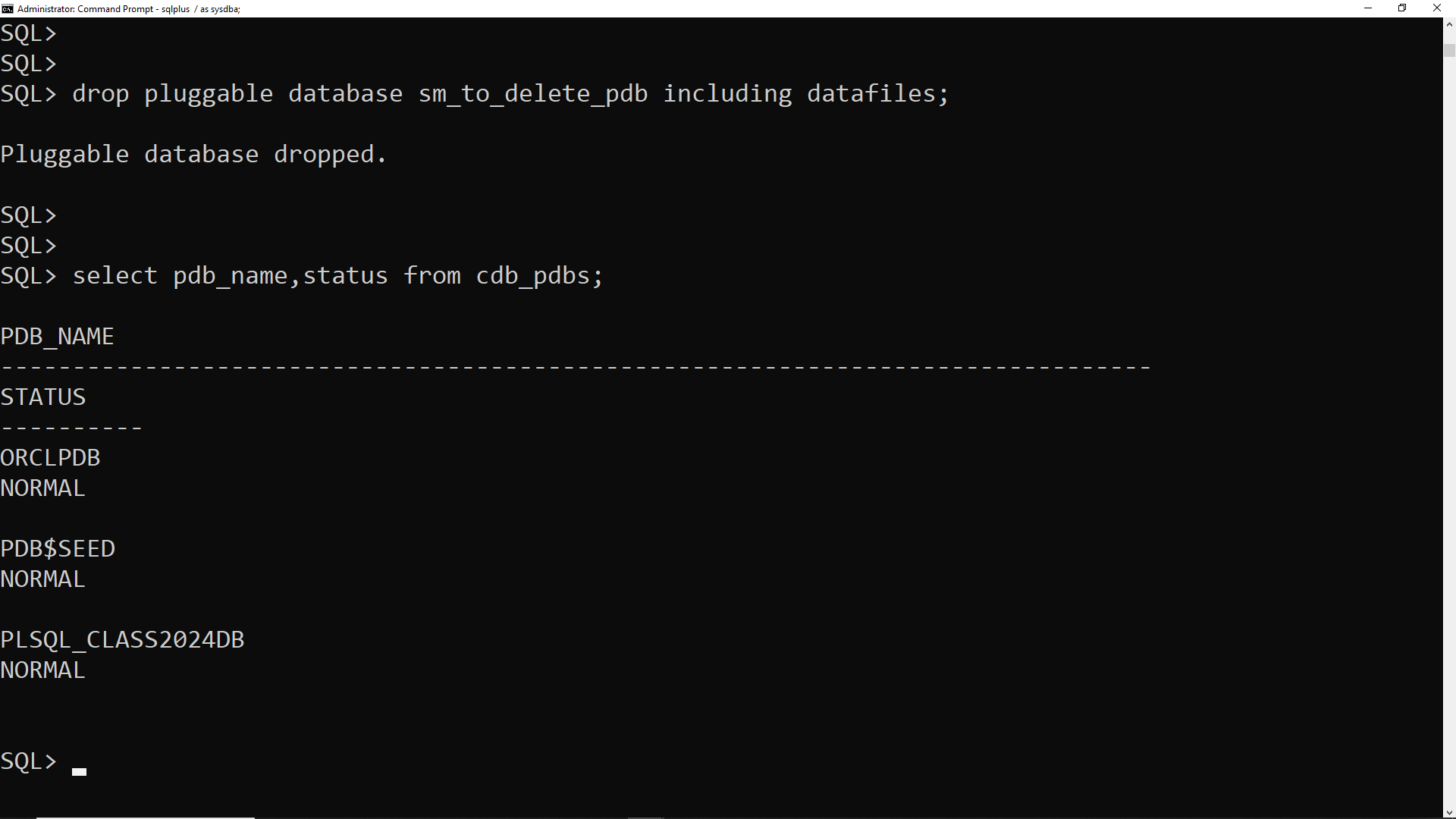
Then use this path :'C:\Users\EDDY\Desktop\OracleApp\admin\orcl\dpdump\ ‘to unplug sm\_to\_delete\_pdb including all datafiles. Using :”alter pluggable databse sm\_to\_delete\_pdb unplug into :'C:\Users\EDDY\Desktop\OracleApp\admin\orcl\dpdump\sm\_to\_delete\_pdb.xml;’”



Finally you can drop sm\_to\_delete\_pdbs with all its datafiles using :”drop pluggable sm\_to\_delete\_pdb including datafiles;”.



Lastly check if it is successfully dropped using :“select pdb\_name, status from cdb\_pdbs;”



As you see it is successfully dropped.

**3. Configure Oracle Enterprise Manager**

To check if I am connected to the root container we use “show con\_name;” If not, we should use:

alter session set container = CDB$ROOT; and Check the current container with

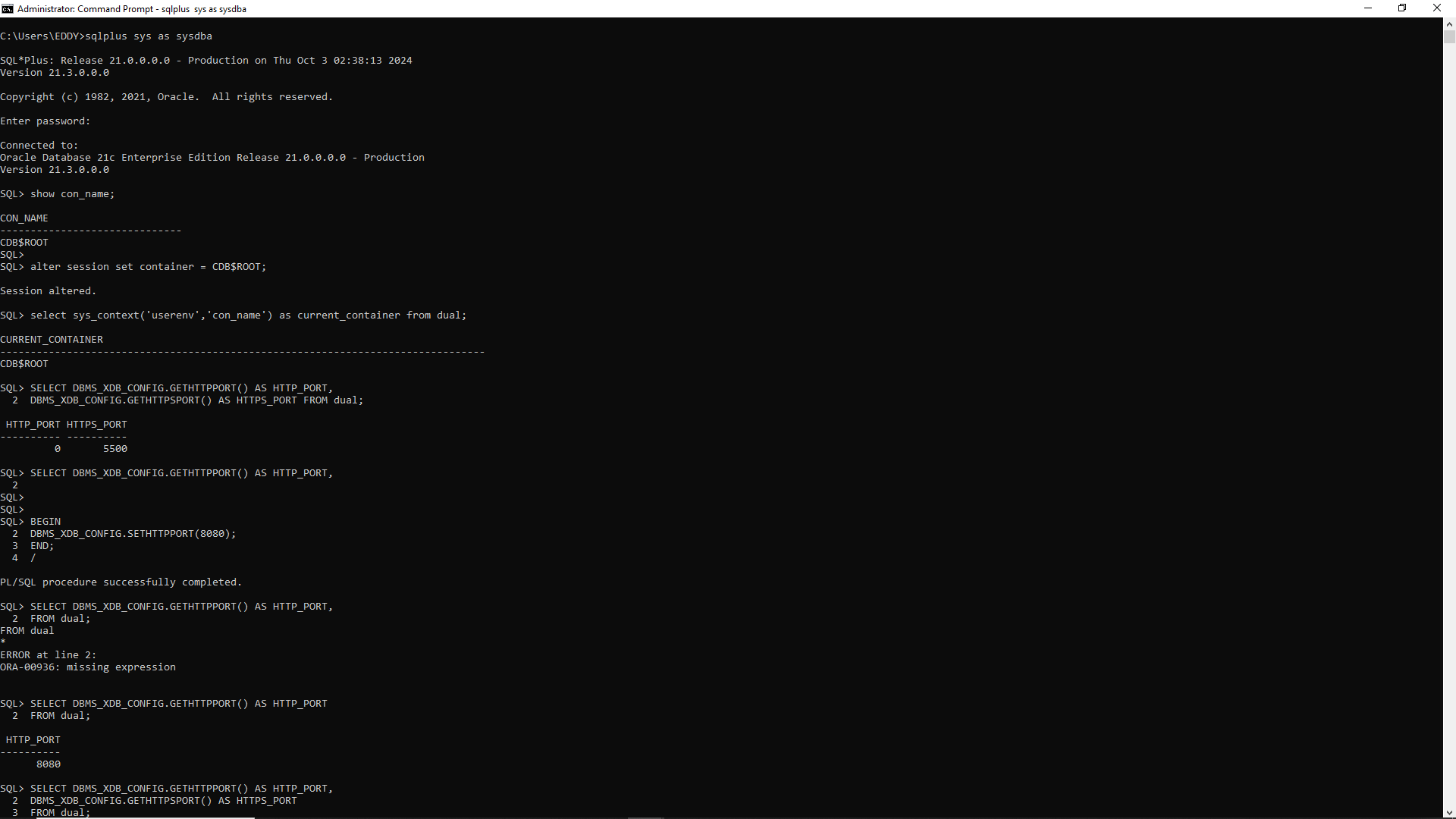
select sys\_context('userenv','con\_name') as current\_container from dual;

Then verify Configuration: After restarting, run the initial query again to

confirm the new port settings with

SELECT DBMS\_XDB\_CONFIG.GETHTTPPORT() AS HTTP\_PORT,

DBMS\_XDB\_CONFIG.GETHTTPSPORT() AS HTTPS\_PORT FROM dual;



Since HTTP\_PORT is returning 0, it confirms that the HTTP server is currently not enabled. Here’s how to proceed: Set New HTTP Port: Let's set the HTTP port to a specific value.

Choose ports that are not in use 8080 for with:

BEGIN

DBMS\_XDB\_CONFIG.SETHTTPPORT(8080);

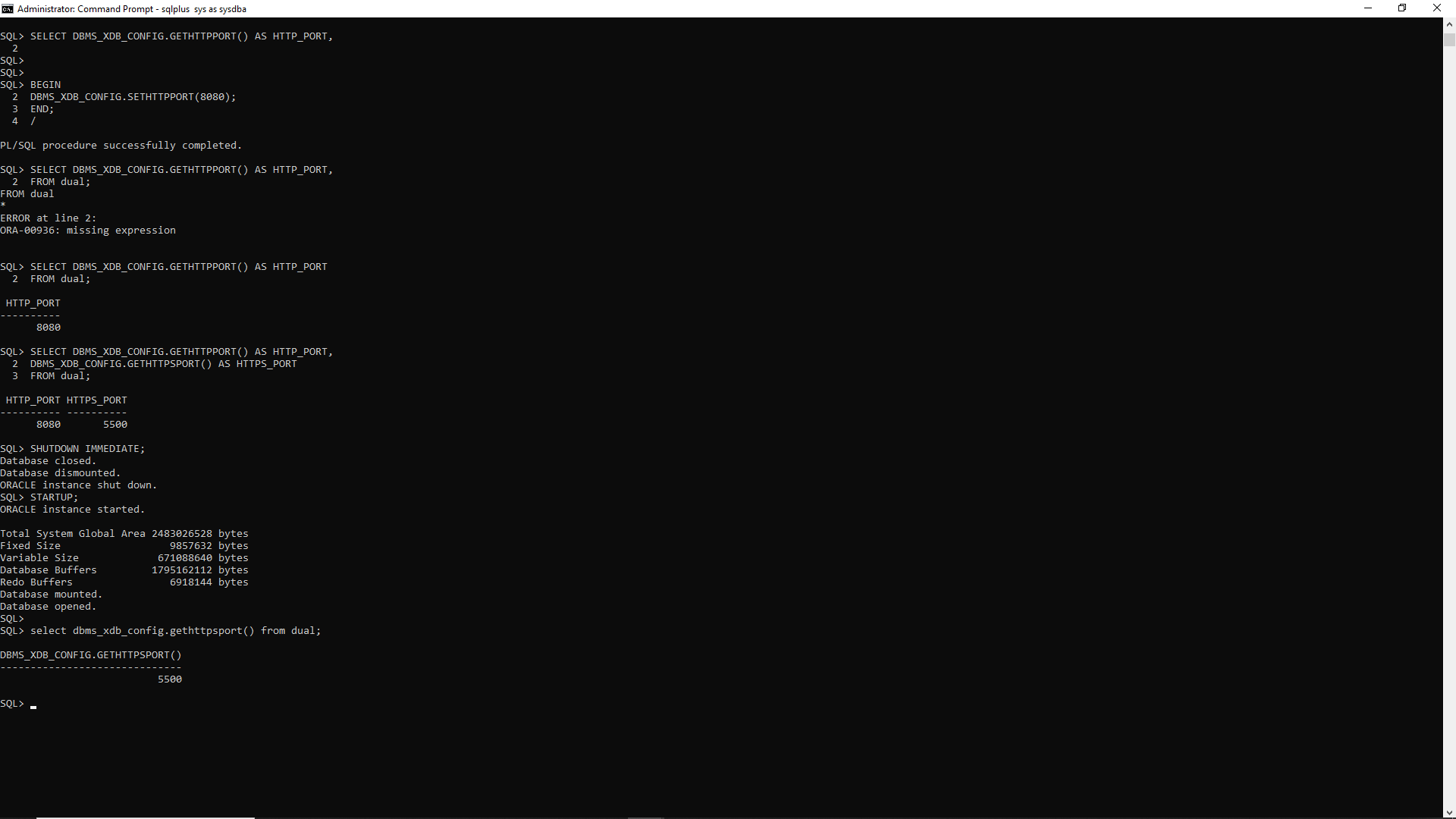
END; /

After executing the above, check the port settings again:

SELECT DBMS\_XDB\_CONFIG.GETHTTPPORT() AS HTTP\_PORT,

DBMS\_XDB\_CONFIG.GETHTTPSPORT() AS HTTPS\_PORT

FROM dual;



If the ports were successfully set, restart the database to apply changes:

SHUTDOWN IMMEDIATE;

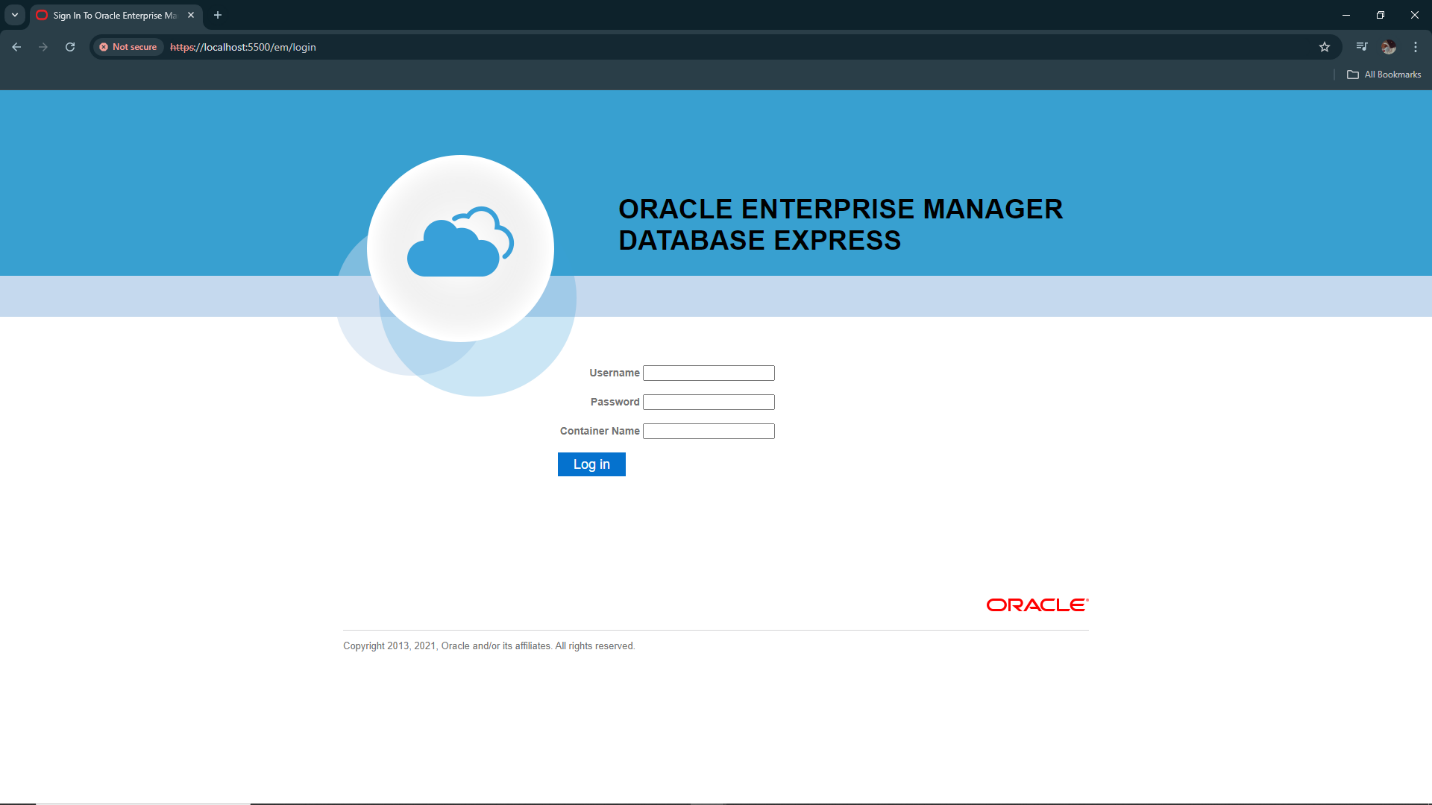
Then start it up again:

STARTUP;

Then this will display both port numbers in one result set. As shown in the above image.

select dbms\_xdb\_config.gethttpsport() from dual;

Later we access our ORACLE ENTREPRISE MANAGER, we can use any desired browser for us we are using chrome. Where we write in URL ‘https://localhost:5500/em/’. We click on advance and click on “continue to localhost(unsafe)”



We use Sys as username and leave container name empty and the password we use the one configured during installation.

