Start the PostgreSQL database WITH SHELL SCRIPT.

The table users will have the following columns:

uname

uid

home

You will connect to template1 database which is already available by default. To connect to this database, run the following command at the ‘postgres=#’ prompt.

```
postgres=# \c template1

psql (14.19 (Ubuntu 14.19-0ubuntu0.22.04.1), server 13.2)
You are now connected to database "template1" as user "postgres".

template1=#

template1=# create table users(username varchar(50),userid int,homedirectory varchar(100));
CREATE TABLE
```

Exercise Loading data into a PostgreSQL table.

> ***postgreCLI to terminal***
> ***In the terminal, run the following command to create a new shell script named csv2db.sh.***

```
theia@theiadocker-rajendraabro:/home/project$ touch csv2db.sh
```

TOUCH IS USED TO CREATE NEW FILE OR CHANGE PERMISSIONS

> *You need to add lines of code to the script that will Extract user name (field 1), user id (field 3), and home directory path (field 6) from /etc/passwd file using the cut command.*
> *bash csv2db.sh*

# Extract phase

```
[[
echo "Extracting data"

# Extract the columns 1 (user name), 2 (user id) and
# 6 (home directory path) from /etc/passwd

cut -d":" -f1,3,6 /etc/passwd ]  ( saved in csv2db file)]

theia@theiadocker-rajendraabro:/home/project$ bash csv2db.sh  ( executes file by bash cmd)

Extracting data
root:0:/root
daemon:1:/usr/sbin
bin:2:/bin
sys:3:/dev
sync:4:/bin
games:5:/usr/games
man:6:/var/cache/man
lp:7:/var/spool/lpd
mail:8:/var/mail
news:9:/var/spool/news
uucp:10:/var/spool/uucp
proxy:13:/bin
www-data:33:/var/www
backup:34:/var/backups
list:38:/var/list
irc:39:/run/ircd
gnats:41:/var/lib/gnats
nobody:65534:/nonexistent
_apt:100:/nonexistent
systemd-network:101:/run/systemd
systemd-resolve:102:/run/systemd
messagebus:103:/nonexistent
systemd-timesync:104:/run/systemd
sshd:105:/run/sshd
theia:1000:/home/theia
cassandra:106:/var/lib/cassandra
mongodb:107:/home/mongodb
```

7. Verify that the output contains the three fields, that you extracted.
8. Change the script to redirect the extracted data into a file named extracted-data.txt

```
cut -d":" -f1,3,6 /etc/passwd > extracted-data.txt
```

```
theia@theiadocker-rajendraabro:/home/project$ bash csv2db.sh
        [Extracting data
        # Transform phase
        echo "Transforming data"
        # read the extracted data and replace the colons with commas.
        [saved in csv2db]
        tr ":" "," < extracted-data.txt  > transformed-data.csv]
```

13. Run the command below to verify that the file transformed-data.csv is created, and has the content.

```
theia@theiadocker-rajendraabro:/home/project$ cat extracted-data.txt

root:0:/root
daemon:1:/usr/sbin
bin:2:/bin
sys:3:/dev
sync:4:/bin
games:5:/usr/games
man:6:/var/cache/man
lp:7:/var/spool/lpd
mail:8:/var/mail
news:9:/var/spool/news
uucp:10:/var/spool/uucp
proxy:13:/bin
www-data:33:/var/www
backup:34:/var/backups
list:38:/var/list
irc:39:/run/ircd
gnats:41:/var/lib/gnats
nobody:65534:/nonexistent
_apt:100:/nonexistent
systemd-network:101:/run/systemd
systemd-resolve:102:/run/systemd
messagebus:103:/nonexistent
systemd-timesync:104:/run/systemd
sshd:105:/run/sshd
theia:1000:/home/theia
cassandra:106:/var/lib/cassandra
mongodb:107:/home/mongodb
```

<u>The extracted columns are separated by the original “:” delimiter. You need to convert this into a “,” delimited file. Add the below lines at the end of the script and save the file.</u>

```
theia@theiadocker-rajendraabro:/home/project$ bash csv2db.sh

Extracting data
Transforming data

theia@theiadocker-rajendraabro:/home/project$ cat transformed-data.csv
root,0,/root
daemon,1,/usr/sbin
bin,2,/bin
sys,3,/dev
sync,4,/bin
games,5,/usr/games
man,6,/var/cache/man
lp,7,/var/spool/lpd
mail,8,/var/mail
news,9,/var/spool/news
uucp,10,/var/spool/uucp
proxy,13,/bin
www-data,33,/var/www
backup,34,/var/backups
list,38,/var/list
irc,39,/run/ircd
gnats,41,/var/lib/gnats
nobody,65534,/nonexistent
_apt,100,/nonexistent
systemd-network,101,/run/systemd
systemd-resolve,102,/run/systemd
messagebus,103,/nonexistent
systemd-timesync,104,/run/systemd
sshd,105,/run/sshd
theia,1000,/home/theia
cassandra,106,/var/lib/cassandra
mongodb,107,/home/mongodb
```

<u>To load data from a shell script, you will use the psql client utility in a non-interactive manner. This is done by sending the database commands through a command pipeline to psql with the help of echo command.</u>

PostgreSQL command to copy data from a CSV file to a table is COPY.

The basic structure of the command which we will use in our script is,

COPY table_name FROM 'filename' DELIMITERS 'delimiter_character' FORMAT;

Now, add the lines below to the end of the script ‘csv2db.sh’ and save the file.

```
theia@theiadocker-rajendraabro:/home/project$ bash csv2db.sh
Extracting data
Transforming data
Loading data

You are now connected to database "template1" as user "postgres".
COPY 27
# Load phase
echo "Loading data"
# Set the PostgreSQL password environment variable.
# Replace <yourpassword> with your actual PostgreSQL password.
export PGPASSWORD=<yourpassword>;
# Send the instructions to connect to 'template1' and
# copy the file to the table 'users' through command pipeline.
echo "\c template1;\COPY users  FROM '/home/project/transformed-data.csv' DELIMITERS ','
CSV;" | psql --username=postgres --host=postgres
```

Exercise 6 - Execute the final script

```
theia@theiadocker-rajendraabro:/home/project$ bash csv2db.sh
Extracting data
Transforming data
Loading data
You are now connected to database "template1" as user "postgres".
COPY 27
    username     | userid |   homedirectory
-----------------+--------+--------------------
 root            |      0 | /root
 daemon          |      1 | /usr/sbin
 bin             |      2 | /bin
 sys             |      3 | /dev
 sync            |      4 | /bin
 games           |      5 | /usr/games
 man             |      6 | /var/cache/man
 lp              |      7 | /var/spool/lpd
 mail            |      8 | /var/mail
 news            |      9 | /var/spool/news
```

```
uucp             |    10 | /var/spool/uucp
proxy            |    13 | /bin
www-data         |    33 | /var/www
backup           |    34 | /var/backups
list             |    38 | /var/list
irc              |    39 | /run/ircd
gnats            |    41 | /var/lib/gnats
nobody           | 65534 | /nonexistent
_apt             |   100 | /nonexistent
systemd-network  |   101 | /run/systemd
systemd-resolve  |   102 | /run/systemd
messagebus       |   103 | /nonexistent
systemd-timesync |   104 | /run/systemd
sshd             |   105 | /run/sshd
theia            |  1000 | /home/theia
cassandra        |   106 | /var/lib/cassandra
mongodb          |   107 | /home/mongodb
root             |     0 | /root
daemon           |     1 | /usr/sbin
bin              |     2 | /bin
sys              |     3 | /dev
sync             |     4 | /bin
games            |     5 | /usr/games
man              |     6 | /var/cache/man
lp               |     7 | /var/spool/lpd
mail             |     8 | /var/mail
news             |     9 | /var/spool/news
uucp             |    10 | /var/spool/uucp
proxy            |    13 | /bin
www-data         |    33 | /var/www
backup           |    34 | /var/backups
list             |    38 | /var/list
irc              |    39 | /run/ircd
gnats            |    41 | /var/lib/gnats
nobody           | 65534 | /nonexistent
_apt             |   100 | /nonexistent
systemd-network  |   101 | /run/systemd
systemd-resolve  |   102 | /run/systemd
messagebus       |   103 | /nonexistent
systemd-timesync |   104 | /run/systemd
sshd             |   105 | /run/sshd
theia            |  1000 | /home/theia
cassandra        |   106 | /var/lib/cassandra
mongodb          |   107 | /home/mongodb
```

(54 rows)
Practice exercises
Copy the data in the file ‘web-server-access-log.txt.gz’ to the table ‘access_log’ in the PostgreSQL database ‘template1’.

The file is available at the location :
https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/ETL%20using%20shell%20scripting/web-server-access-log.txt.gz

The following are the columns and their data types in the file:

a. timestamp - TIMESTAMP
b. latitude - float
c. longitude - float
d. visitorid - char(37)
e. accessed_from_mobile - boolean
f. browser_code - int
The columns which we need to copy to the table are the first four coumns : timestamp, latitude, longitude and visitorid.

NOTE: The file comes with a header. So use the ‘HEADER’ option in the ‘COPY’ command.

The problem may be solved by completing the following tasks:

Go to the SkillsNetwork Tools menu and start the Postgres SQL server if it is not already running.

Create a table named access_log to store the timestamp, latitude, longitude and visitorid.

Click here for Hint
Click here for Solution
Step 1: Open the Postgres SQL CLI, if it is not already open.

Step 2: At the postgres=# prompt, run the following command to connect to the database ‘template1’.

1
```
\c template1;
```

Copied!

Wrap Toggled!
Step 3: Once you connect to the database, run the command to create the table called ‘access_log’:

```
CREATE TABLE access_log(timestamp TIMESTAMP, latitude float, longitude float, visitor_id char(37));
```

Copied!

Wrap Toggled!
Task 3. Create a shell script named cp-access-log.sh and add commands to complete the remaining tasks to extract and copy the data to the database.

Create a shell script to add commands to complete the rest of the tasks.

Click here for Hint
Task 4. Download the access log file.

Add the wget command to the script to download the file.

```
wget "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/ETL%20using%20shell%20scripting/web-server-access-log.txt.gz"
```

Copied!

Wrap Toggled!
Task 5. Unzip the gzip file.

Add the code, to run the gunzip command to unzip the .gz file and extract the .txt file, to the script.

```
# Unzip the file to extract the .txt file.
gunzip -f web-server-access-log.txt.gz
```

Copied!

Wrap Toggled!
The -f option of gunzip is to overwrite the file if it already exists.

Task 6. Extract required fields from the file.

Extract timestamp, latitude, longitude and visitorid which are the first four fields from the file using the cut command.

The columns in the web-server-access-log.txt file is delimited by ‘#’.

Click here for Hint
Click here for Solution
Step 1: Copy the following lines and add them to the end of the script.

```
# Extract phase
echo "Extracting data"
# Extract the columns 1 (timestamp), 2 (latitude), 3 (longitude) and
# 4 (visitorid)
cut -d"#" -f1-4 web-server-access-log.txt
```

Copied!

Wrap Toggled!
Step 2: Save the file.

Step 3: Run the script.

```
bash cp-access-log.sh
```

Copied!

Wrap Toggled!

Executed!
Verify that the output contains all the four fields that we extracted.

Task 7. Redirect the extracted output into a file.

Redirect the extracted data into a file named extracted-data.txt

Click here for Hint
Click here for Solution
Step 1: Replace the cut command at end of the script with the following command and save the file.

```
cut -d"#" -f1-4 web-server-access-log.txt > extracted-data.txt
```

Copied!

Wrap Toggled!
Step 2: Run the script.

```
bash cp-access-log.sh
```

Copied!

Wrap Toggled!
Step 3: Run the command below to verify that the file extracted-data.txt is created, and has the content.

```
cat extracted-data.txt
```

Copied!

Wrap Toggled!
Task 8. Transform the data into CSV format.

The extracted columns are separated by the original “#” delimiter.

We need to convert this into a “,” delimited file.

Click here for Hint
Click here for Solution
Step 1: Add the lines below at the end of the script.

```
# Transform phase
echo "Transforming data"
# read the extracted data and replace the colons with commas and
# write it to a csv file
tr "#" "," < extracted-data.txt > transformed-data.csv
```

Copied!

Wrap Toggled!

Step 2: Save the file.

Step 3: Run the script.

```
bash cp-access-log.sh
```

Copied!

Wrap Toggled!

Step 4: Run the command below to verify that the file ‘transformed-data.csv’ is created, and has the content.

```
cat transformed-data.csv
```

Copied!

Wrap Toggled!

Task 9. Load the data into the table access_log in PostgreSQL

PostgreSQL command to copy data from a CSV file to a table is COPY.

The basic structure of the command is,

```
COPY table_name FROM 'filename' DELIMITERS 'delimiter_character' FORMAT;
```

Copied!

Wrap Toggled!

The file comes with a header. So use the ‘HEADER’ option in the ‘COPY’ command.

Invoke this command from the shellscript, by sending it as input to ‘psql’ filter command.

Click here for Hint
Click here for Solution
Step 1: Add the lines below to the end of the script ‘cp-access-log.sh’ and save the file.

```
# Load phase
echo "Loading data"
# Send the instructions to connect to 'template1' and
# copy the file to the table 'access_log' through command pipeline.
echo "\c template1;\COPY access_log  FROM '/home/project/transformed-data.csv'
DELIMITERS ',' CSV HEADER;" | psql --username=postgres --host=localhost
```

Copied!

Wrap Toggled!
Task 10. Execute the final script.

Run the final script.

Click here for Solution
Run the following command at the terminal:

```
bash cp-access-log.sh
```

Copied!

Wrap Toggled!
The completed bash script would be as below.

1
2
3
4
5
6
7
8

```
# cp-access-log.sh
# This script downloads the file 'web-server-access-log.txt.gz'
# from
"https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetw
ork/labs/Bash%20Scripting/ETL%20using%20shell%20scripting/".
# The script then extracts the .txt file using gunzip.
# The .txt file contains the timestamp, latitude, longitude
# and visitor id apart from other data.
# Transforms the text delimeter from "#" to "," and saves to a csv file.
# Loads the data from the CSV file into the table 'access_log' in PostgreSQL database.
# Download the access log file
```

```
wget
"https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetw
ork/labs/Bash%20Scripting/ETL%20using%20shell%20scripting/web-server-access-log.txt.gz"
# Unzip the file to extract the .txt file.
gunzip -f web-server-access-log.txt.gz
# Extract phase
echo "Extracting data"
# Extract the columns 1 (timestamp), 2 (latitude), 3 (longitude) and
# 4 (visitorid)
cut -d"#" -f1-4 web-server-access-log.txt > extracted-data.txt
# Transform phase
echo "Transforming data"
# read the extracted data and replace the colons with commas.
tr "#" "," < extracted-data.txt > transformed-data.csv
# Load phase
echo "Loading data"
# Send the instructions to connect to 'template1' and
# copy the file to the table 'access_log' through command pipeline.
echo "\c template1;\COPY access_log  FROM '/home/project/transformed-data.csv'
DELIMITERS ',' CSV HEADER;" | psql --username=postgres --host=localhost
```

Copied!

Wrap Toggled!
Task 11. Verify by querying the database.

Click here for Hint
Click here for Solution
Run the command below at Postgres SQL CLI prompt.

```
SELECT * from access_log;
```

Copied!

Wrap Toggled!
You should see the records displayed on screen.