

Internet of Things: Assignment 3

Name: Ajay Choudhury

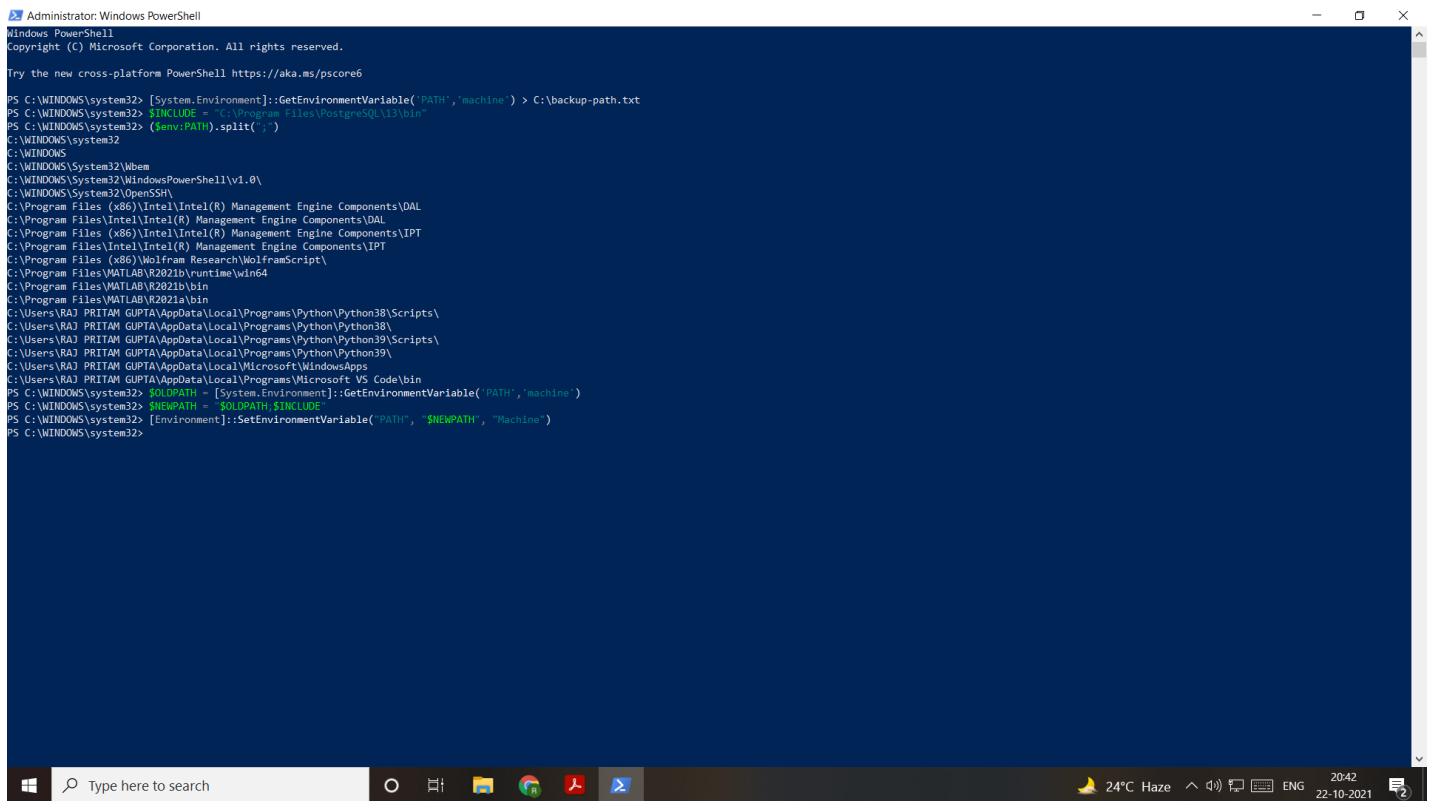
Roll No.: 18018

Date: 22nd October 2021

Installing TimescaleDB:

Here are the steps and codes used for installing TimescaleDB on Windows PC:

1. Installation of PostgreSQL 13.4 and added its path to environment variables.



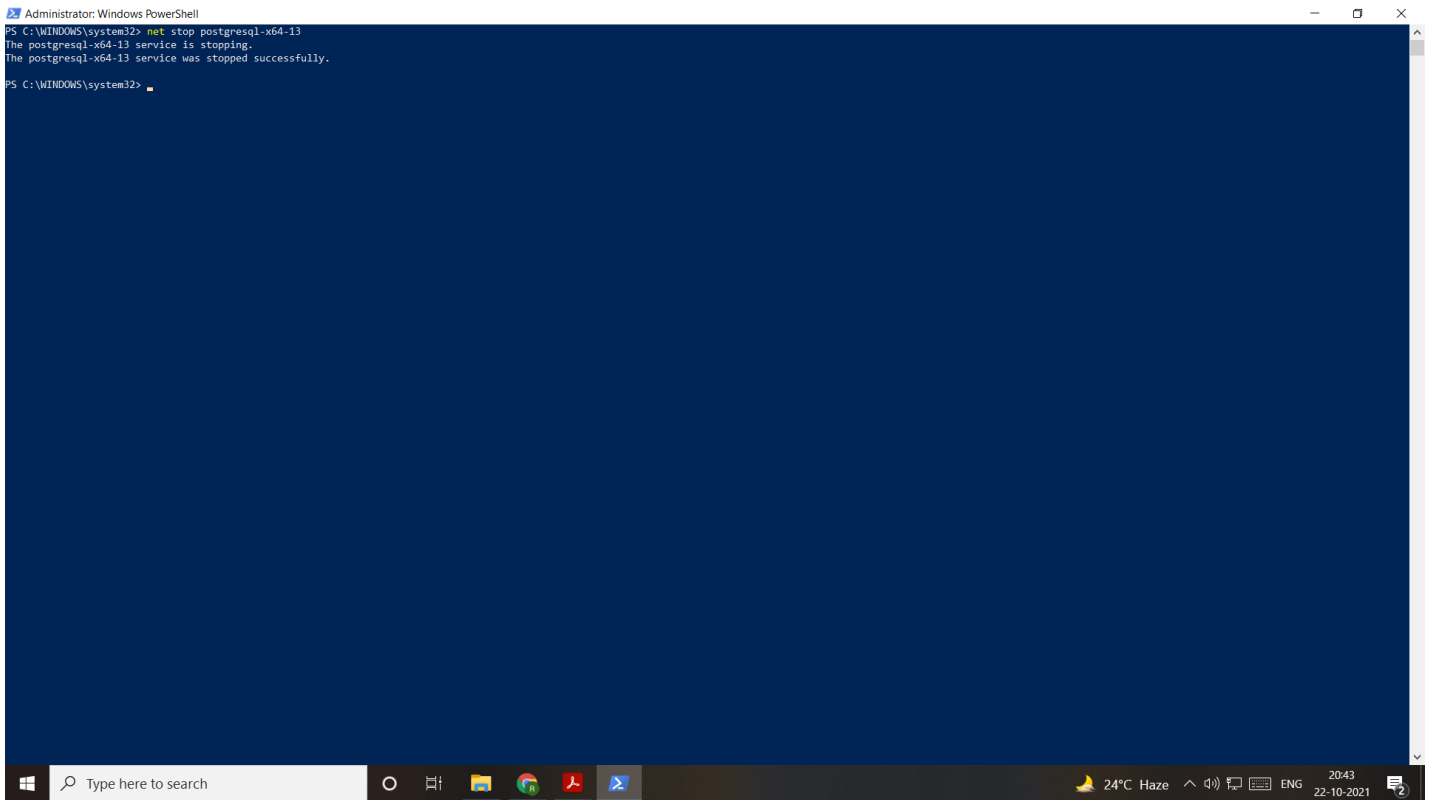
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\WINDOWS\system32> [System.Environment]::GetEnvironmentVariable("PATH","machine") > C:\backup-path.txt
PS C:\WINDOWS\system32> $INCLUDE = "C:\Program Files\PostgreSQL\13\bin"
PS C:\WINDOWS\system32> ($env:PATH).split(";")
C:\WINDOWS\system32
C:\WINDOWS
C:\WINDOWS\System32\Wbem
C:\WINDOWS\System32\WindowsPowerShell\v1.0\
C:\WINDOWS\System32\OpenSSH\
C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\DAL
C:\Program Files\Intel\Intel(R) Management Engine Components\DAL
C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\IPT
C:\Program Files\Intel\Intel(R) Management Engine Components\IPT
C:\Program Files (x86)\Wolfram Research\WolframScript\
C:\Program Files\WATLAB\R2021b\runtime\win64
C:\Program Files\WATLAB\R2021b\bin
C:\Program Files\WATLAB\R2021a\bin
C:\Users\RAJ PRITAM GUPTA\AppData\Local\Programs\Python\Python38\Scripts\
C:\Users\RAJ PRITAM GUPTA\AppData\Local\Programs\Python\Python38\
C:\Users\RAJ PRITAM GUPTA\AppData\Local\Programs\Python\Python39\Scripts\
C:\Users\RAJ PRITAM GUPTA\AppData\Local\Programs\Python\Python39\
C:\Users\RAJ PRITAM GUPTA\AppData\Local\Microsoft\WindowsApps
C:\Users\RAJ PRITAM GUPTA\AppData\Local\Programs\Microsoft VS Code\bin
PS C:\WINDOWS\system32> $OLDPATH = [System.Environment]::GetEnvironmentVariable("PATH","machine")
PS C:\WINDOWS\system32> $NEWPATH = "$OLDPATH;$INCLUDE"
PS C:\WINDOWS\system32> [Environment]::SetEnvironmentVariable("PATH", "$NEWPATH", "Machine")
PS C:\WINDOWS\system32>
```

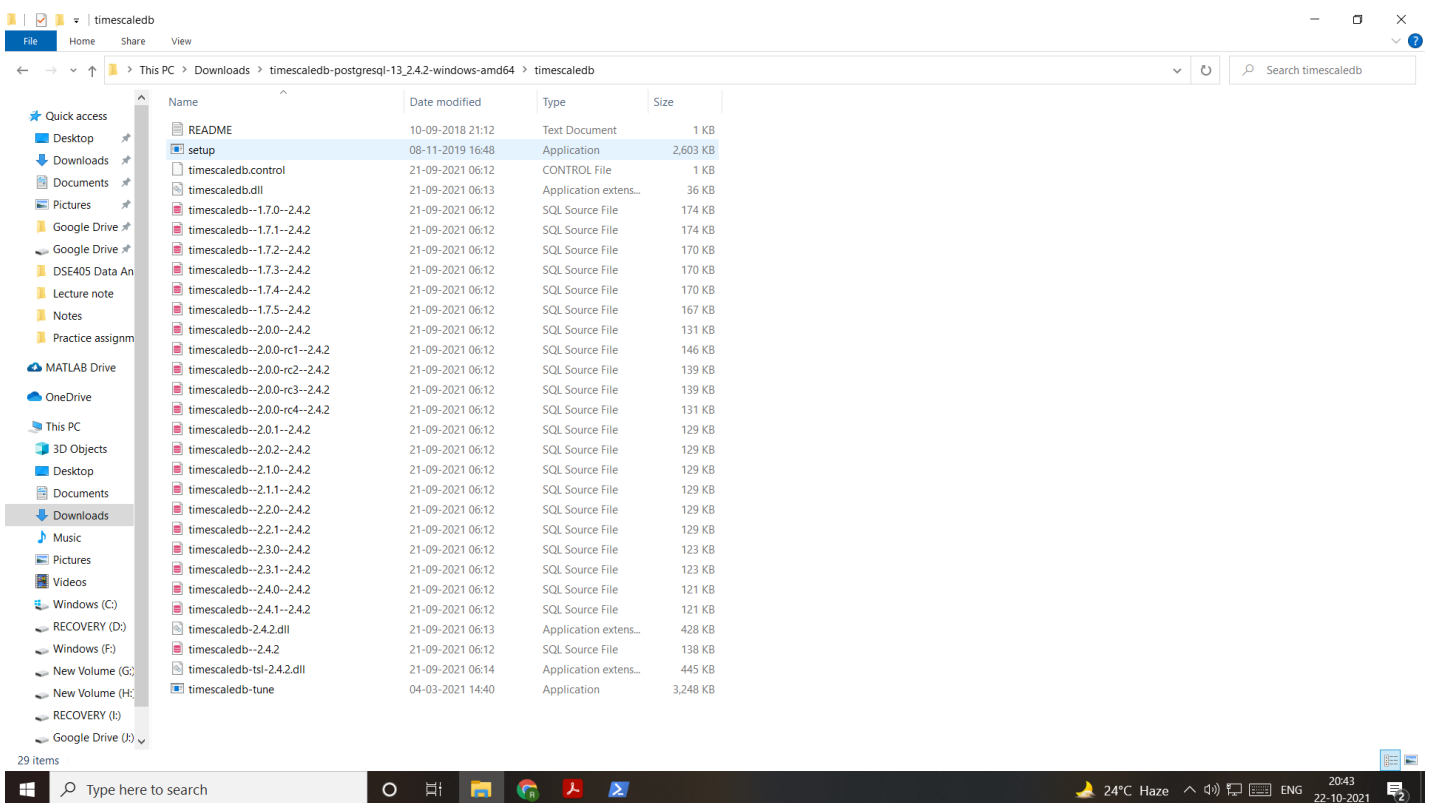
2. Stop PostgreSQL service for timescale installation.

```
net stop postgresql-x64-13
```



3. Downloaded and installed the [TimescaleDB setup](#) with the path of the postgresql configuration as:

C:\Program Files\PostgreSQL\13\data\postgresql.conf



```
C:\Users\RAJ PRITAM GUPTA\Downloads\timescaledb-postgresql-13_242-windows-amd64\timescaledb\setup.exe
2021/10/22 20:44:36 WELCOME to TimescaleDB Windows installer!
2021/10/22 20:44:37 timescaledb-tune is a program that modifies your postgresql.conf configuration to be optimized for your machine's resources.
Do you want to run timescaledb-tune.exe now? [(y)es / (n)o]: y
Please enter the path to your postgresql.conf:
C:\Program Files\PostgreSQL\13\data\postgresql.conf
== Using postgresql.conf at this path:
C:\Program Files\PostgreSQL\13\data\postgresql.conf
== Writing backup to:
C:\Users\RAJPRI-1\AppData\Local\Temp\timescaledb_tune.backup202110222045
SUCCESS: shared_preload_libraries is set correctly
-- Tune memory/parallelism/WAL and other settings? [(y)es/(n)o]: y
== Recommendations based on 3.92 GB of available memory and 4 CPUs for PostgreSQL 13
== Memory settings recommendations
SUCCESS: memory settings are already tuned
== Parallelism settings recommendations
SUCCESS: parallelism settings are already tuned
== WAL settings recommendations
SUCCESS: WAL settings are already tuned
== Miscellaneous settings recommendations
SUCCESS: miscellaneous settings are already tuned
== Saving changes to: C:\Program Files\PostgreSQL\13\data\postgresql.conf
2021/10/22 20:45:24 Installing TimescaleDB library files...
2021/10/22 20:45:24 Success!
2021/10/22 20:45:24 Installing TimescaleDB control file...
2021/10/22 20:45:24 Success!
2021/10/22 20:45:24 Installing TimescaleDB SQL files...
2021/10/22 20:45:24 Success!
TimescaleDB installation completed successfully.
Press ENTER/Return key to close...
```

4. Start the PostgreSQL service.

```
net start postgresql-x64-13
```

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> net stop postgresql-x64-13
The postgresql-x64-13 service is stopping.
The postgresql-x64-13 service was stopped successfully.

PS C:\WINDOWS\system32> net start postgresql-x64-13
The postgresql-x64-13 service is starting.
The postgresql-x64-13 service was started successfully.

PS C:\WINDOWS\system32>
```

5. Open the SQL shell and verify timescaleDB installation.

```
SELECT * FROM pg_available_extensions WHERE name='timescaledb';
```

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# SELECT * FROM pg_available_extensions WHERE name='timescaledb';
   name   | default_version | installed_version | comment
-----
timescaledb | 2.4.2          |                  | Enables scalable inserts and complex queries for time-series data
(1 row)

postgres=#
```

6. Enable the timescale installation.

CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# SELECT * FROM pg_available_extensions WHERE name='timescaledb';
   name   | default_version | installed_version | comment
-----
timescaledb | 2.4.2          |                  | Enables scalable inserts and complex queries for time-series data
(1 row)

postgres=# CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;
WARNING:
WELCOME TO

TimescaleDB
Running version 2.4.2

For more information on TimescaleDB, please visit the following links:
1. Getting started: https://docs.timescale.com/timescaledb/latest/getting-started
2. API reference documentation: https://docs.timescale.com/api/latest
3. How TimescaleDB is designed: https://docs.timescale.com/timescaledb/latest/overview/core-concepts

Note: TimescaleDB collects anonymous reports to better understand and assist our users.
For more information and how to disable, please see our docs https://docs.timescale.com/timescaledb/latest/how-to-guides/configuration/telemetry.

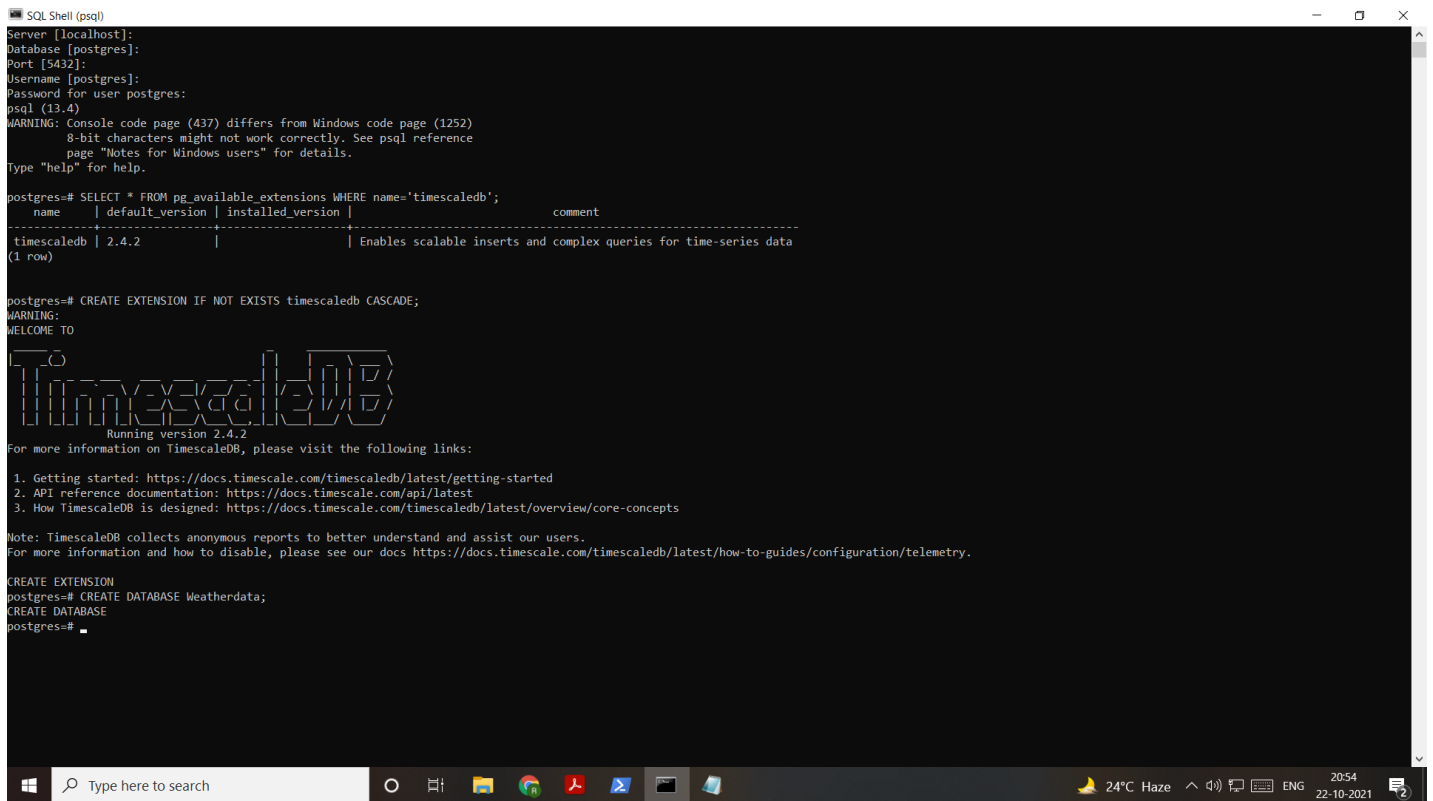
CREATE EXTENSION
postgres=#
```

Now, timescaleDB is installed successfully on the device. We can now proceed with data storage and database creation.

Part 1:

Create a database named “Weatherdata”. Code for this is:

```
CREATE DATABASE Weatherdata;
```



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

postgres=# SELECT * FROM pg_available_extensions WHERE name='timescaledb';
   name   | default_version | installed_version |      comment
-----|-----|-----|-----
timescaledb | 2.4.2           |                   | Enables scalable inserts and complex queries for time-series data
(1 row)

postgres=# CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;
WARNING:
WELCOME TO

TimescaleDB
Running version 2.4.2

For more information on TimescaleDB, please visit the following links:
1. Getting started: https://docs.timescale.com/timescaledb/latest/getting-started
2. API reference documentation: https://docs.timescale.com/api/latest
3. How TimescaleDB is designed: https://docs.timescale.com/timescaledb/latest/overview/core-concepts

Note: TimescaleDB collects anonymous reports to better understand and assist our users.
For more information and how to disable, please see our docs https://docs.timescale.com/timescaledb/latest/how-to-guides/configuration/telemetry.

CREATE EXTENSION
postgres=# CREATE DATABASE Weatherdata;
CREATE DATABASE
postgres=#
```

Part 2:

Within “Weatherdata” create a hypertable with the name “weather_table”. Code for same is:

1. Connect to the database:

```
\c weatherdata
```

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# SELECT * FROM pg_available_extensions WHERE name='timescaledb';
 name | default_version | installed_version | comment
-----+-----+-----+-----
timescaledb | 2.4.2 |  | Enables scalable inserts and complex queries for time-series data
(1 row)

postgres=# CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;
WARNING:
WELCOME TO

TimescaleDB
Running version 2.4.2
For more information on TimescaleDB, please visit the following links:
1. Getting started: https://docs.timescale.com/timescaledb/latest/getting-started
2. API reference documentation: https://docs.timescale.com/api/latest
3. How TimescaleDB is designed: https://docs.timescale.com/timescaledb/latest/overview/core-concepts

Note: TimescaleDB collects anonymous reports to better understand and assist our users.
For more information and how to disable, please see our docs https://docs.timescale.com/timescaledb/latest/how-to-guides/configuration/telemetry.

CREATE EXTENSION
postgres=# CREATE DATABASE Weatherdata;
CREATE DATABASE
postgres=# \c weatherdata
You are now connected to database "weatherdata" as user "postgres".
weatherdata=#
```

2. Extend the database with timescaleDB:

```
CREATE EXTENSION IF NOT EXISTS timescaledb;
```

```
SQL Shell (psql)
postgres=# CREATE DATABASE Weatherdata;
CREATE DATABASE
postgres=# \c weatherdata
You are now connected to database "weatherdata" as user "postgres".
weatherdata=# CREATE EXTENSION IF NOT EXISTS timescaledb;
WARNING:
WELCOME TO

TimescaleDB
Running version 2.4.2
For more information on TimescaleDB, please visit the following links:
1. Getting started: https://docs.timescale.com/timescaledb/latest/getting-started
2. API reference documentation: https://docs.timescale.com/api/latest
3. How TimescaleDB is designed: https://docs.timescale.com/timescaledb/latest/overview/core-concepts

Note: TimescaleDB collects anonymous reports to better understand and assist our users.
For more information and how to disable, please see our docs https://docs.timescale.com/timescaledb/latest/how-to-guides/configuration/telemetry.

CREATE EXTENSION
weatherdata=#
```

3. Create a table in the database:

```
CREATE TABLE weather_table (id INT NOT NULL, time TIMESTAMPTZ NOT NULL,
```

```
temperature_site1 DOUBLE PRECISION NULL, humidity_site1 DOUBLE PRECISION
NULL, temperature_site2 DOUBLE PRECISION NULL, humidity_site2 DOUBLE
PRECISION NULL);
```

```
SQL Shell (psql)
postgres=# CREATE DATABASE Weatherdata;
CREATE DATABASE
postgres=# \c weatherdata
You are now connected to database "weatherdata" as user "postgres".
weatherdata=# CREATE EXTENSION IF NOT EXISTS timescaledb;
WARNING:
WELCOME TO
TimescaleDB
Running version 2.4.2
For more information on TimescaleDB, please visit the following links:
1. Getting started: https://docs.timescale.com/timescaledb/latest/getting-started
2. API reference documentation: https://docs.timescale.com/api/latest
3. How TimescaleDB is designed: https://docs.timescale.com/timescaledb/latest/overview/core-concepts
Note: TimescaleDB collects anonymous reports to better understand and assist our users.
For more information and how to disable, please see our docs https://docs.timescale.com/timescaledb/latest/how-to-guides/configuration/telemetry.
CREATE EXTENSION
weatherdata=#
weatherdata=# CREATE TABLE weather_table (id INT NOT NULL, time TIMESTAMPTZ NOT NULL, temperature_site1 DOUBLE PRECISION NULL, humidity_site1 DOUBLE PRECISION NULL, temperature_site2 DOUBLE PRECISION NULL, humidity_site2 DOUBLE PRECISION NULL);
CREATE TABLE
weatherdata=#
```

4. Transform to hypertable:

```
SELECT create_hypertable('weather_table', 'time');
```

```
SQL Shell (psql)
postgres=# CREATE DATABASE Weatherdata;
CREATE DATABASE
postgres=# \c weatherdata
You are now connected to database "weatherdata" as user "postgres".
weatherdata=# CREATE EXTENSION IF NOT EXISTS timescaledb;
WARNING:
WELCOME TO
TimescaleDB
Running version 2.4.2
For more information on TimescaleDB, please visit the following links:
1. Getting started: https://docs.timescale.com/timescaledb/latest/getting-started
2. API reference documentation: https://docs.timescale.com/api/latest
3. How TimescaleDB is designed: https://docs.timescale.com/timescaledb/latest/overview/core-concepts
Note: TimescaleDB collects anonymous reports to better understand and assist our users.
For more information and how to disable, please see our docs https://docs.timescale.com/timescaledb/latest/how-to-guides/configuration/telemetry.
CREATE EXTENSION
weatherdata=#
weatherdata=# CREATE TABLE weather_table (id INT NOT NULL, time TIMESTAMPTZ NOT NULL, temperature_site1 DOUBLE PRECISION NULL, humidity_site1 DOUBLE PRECISION NULL, temperature_site2 DOUBLE PRECISION NULL, humidity_site2 DOUBLE PRECISION NULL);
CREATE TABLE
weatherdata=# SELECT create_hypertable('weather_table', 'time');
create_hypertable
-----
(1,public,weather_table,t)
(1 row)
weatherdata=#
```

Part 3:

1. Convert timestamp in format (dd/mm/yyyy HH:MM:SS). (Here using Python)

```
import pandas as pd

# read csv file
df = pd.read_csv('weather_data_2sites.csv')

# convert to datetime format
df['timestamp'] = pd.to_datetime(df.timestamp)

# get day first
df['timestamp'] = df['timestamp'].dt.strftime('%d/%m/%Y %H:%M:%S')

# rename id column
df.rename( columns={'Unnamed: 0':'id'}, inplace=True )

# save the modified CSV data
df.to_csv('weather_data_2sites.csv', index=False)
```

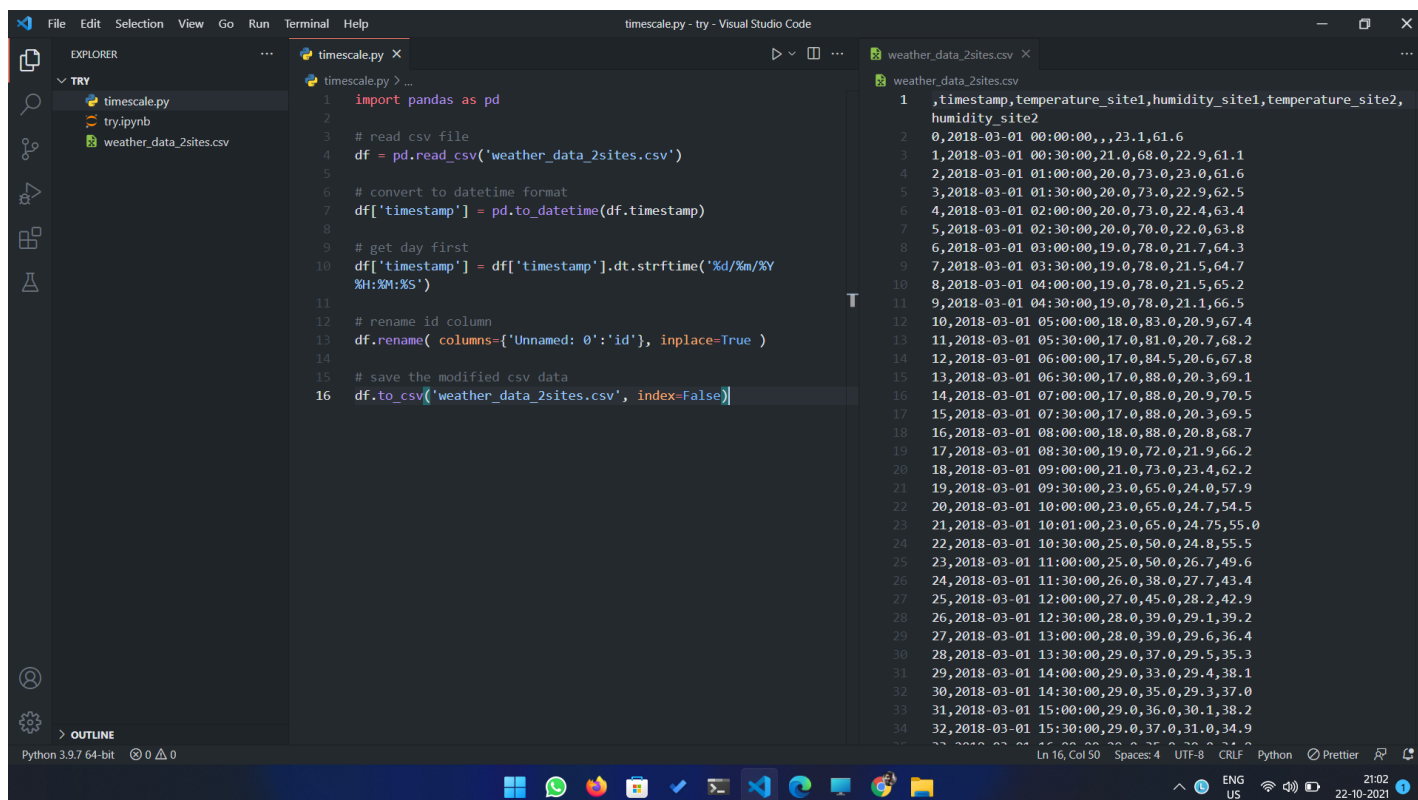


Fig.: Before running the script.

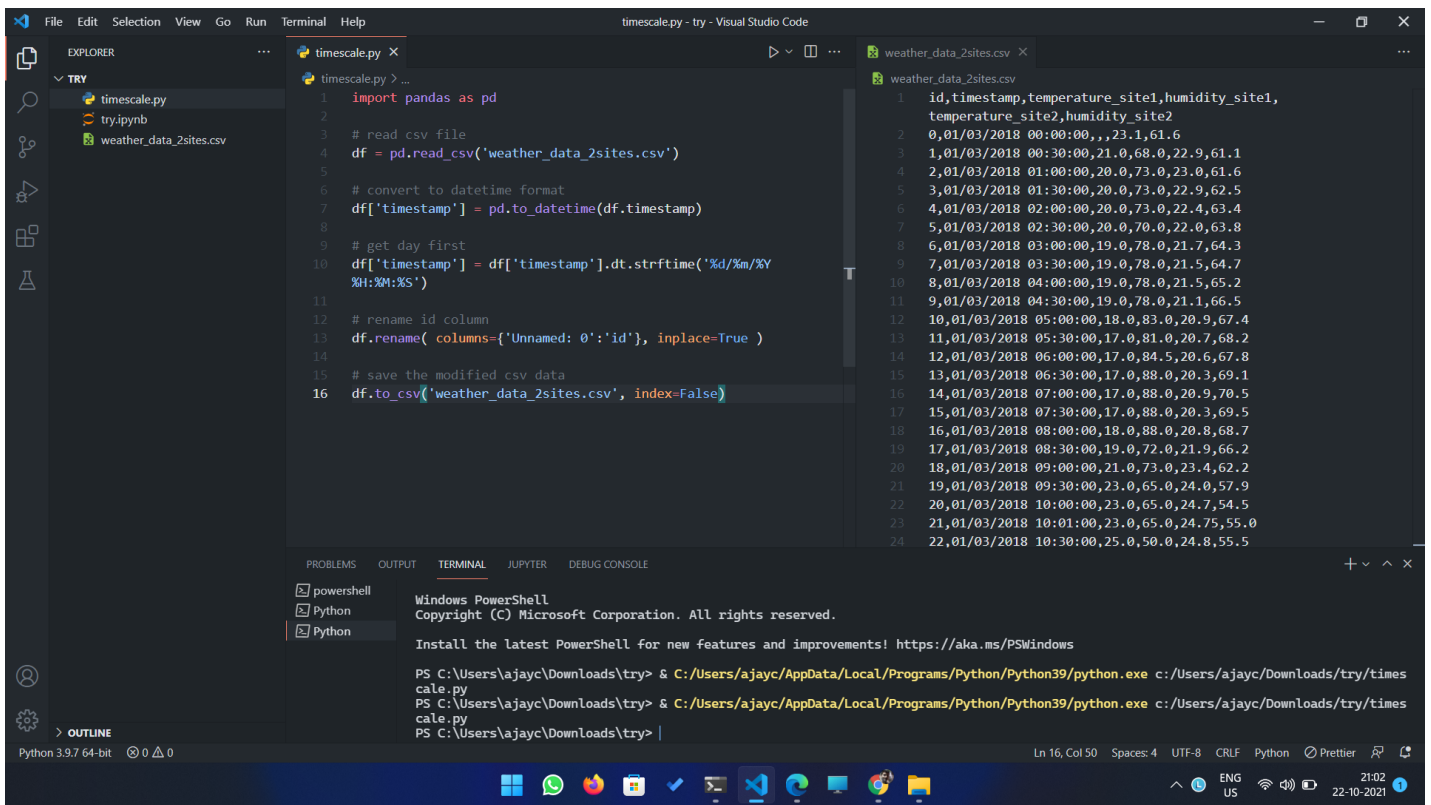


Fig.: After running the script

2. Insert data from CSV file to database:

```
INSERT INTO weather_table(id, time, temperature_site1, humidity_site1,
temperature_site2, humidity_site2) VALUES (0,'01/03/2018
00:00:00',NULL,NULL,23.1,61.6);
```

```
INSERT INTO weather_table(id, time, temperature_site1, humidity_site1,
temperature_site2, humidity_site2) VALUES (1,'01/03/2018
00:30:00',21.0,68.0,22.9,61.1);
```

.

.

.

```
INSERT INTO weather_table(id, time, temperature_site1, humidity_site1,
temperature_site2, humidity_site2) VALUES (8,'01/03/2018
04:00:00',19.0,78.0,21.5,65.2);
```

```
INSERT INTO weather_table(id, time, temperature_site1, humidity_site1,
temperature_site2, humidity_site2) VALUES (9,'01/03/2018
04:30:00',19.0,78.0,21.1,66.5);
```

```
SQL Shell (psql)
3. How TimescaleDB is designed: https://docs.timescale.com/timescaledb/latest/overview/core-concepts
Note: TimescaleDB collects anonymous reports to better understand and assist our users.
For more information and how to disable, please see our docs https://docs.timescale.com/timescaledb/latest/how-to-guides/configuration/telemetry.

CREATE EXTENSION
weatherdata=# CREATE TABLE weather_table (id INT NOT NULL, time TIMESTAMPTZ NOT NULL, temperature_site1 DOUBLE PRECISION NULL, humidity_site1 DOUBLE PRECISION NULL, temperature_site2 DOUBLE PRECISION NULL, humidity_site2 DOUBLE PRECISION NULL);
CREATE TABLE
weatherdata=# SELECT create_hypertable('weather_table', 'time');
create_hypertable
-----
(1,public,weather_table,t)
(1 row)

weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (0,'01/03/2018 00:00:00',NULL,NULL,23.1,61.6);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (1,'01/03/2018 00:30:00',21.0,68.0,22.9,61.1);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (2,'01/03/2018 01:00:00',20.0,73.0,23.0,61.6);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (3,'01/03/2018 01:30:00',20.0,73.0,22.9,62.5);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (4,'01/03/2018 02:00:00',20.0,73.0,22.4,63.4);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (5,'01/03/2018 02:30:00',20.0,70.0,22.0,63.8);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (6,'01/03/2018 03:00:00',19.0,78.0,21.7,64.3);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (7,'01/03/2018 03:30:00',19.0,78.0,21.5,64.7);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (8,'01/03/2018 04:00:00',19.0,78.0,21.5,65.2);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (9,'01/03/2018 04:30:00',19.0,78.0,21.1,66.5);
INSERT 0 1
weatherdata=# select * from weather_table order by time desc limit 5;
 id |          time          | temperature_site1 | humidity_site1 | temperature_site2 | humidity_site2
-----+-----+-----+-----+-----+-----
  9 | 2018-03-01 04:30:00+05:30 |          19       |          78    |          21.1     |          66.5
  8 | 2018-03-01 04:00:00+05:30 |          19       |          78    |          21.5     |          65.2
  7 | 2018-03-01 03:30:00+05:30 |          19       |          78    |          21.5     |          64.7
  6 | 2018-03-01 03:00:00+05:30 |          19       |          78    |          21.7     |          64.3
  5 | 2018-03-01 02:30:00+05:30 |          20       |          70    |          22       |          63.8
(5 rows)

weatherdata=#
```

Part 4:

1. Print first 5 rows from the table “weather_table”:
- SELECT * FROM weather_table ORDER BY time DESC LIMIT 5;
2. Print last 5 rows from the table “weather_table”:
- SELECT * FROM weather_table ORDER BY time ASC LIMIT 5 WHERE id<5;

```
SQL Shell (psql)
weatherdata=# SELECT create_hypertable('weather_table', 'time');
create_hypertable
-----
(1,public,weather_table,t)
(1 row)

weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (0,'01/03/2018 00:00:00',NULL,NULL,23.1,61.6);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (1,'01/03/2018 00:30:00',21.0,68.0,22.9,61.1);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (2,'01/03/2018 01:00:00',20.0,73.0,23.0,61.6);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (3,'01/03/2018 01:30:00',20.0,73.0,22.9,62.5);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (4,'01/03/2018 02:00:00',20.0,73.0,22.4,63.4);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (5,'01/03/2018 02:30:00',20.0,70.0,22.0,63.8);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (6,'01/03/2018 03:00:00',19.0,78.0,21.7,64.3);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (7,'01/03/2018 03:30:00',19.0,78.0,21.5,64.7);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (8,'01/03/2018 04:00:00',19.0,78.0,21.5,65.2);
INSERT 0 1
weatherdata=# INSERT INTO weather_table(id, time, temperature_site1, humidity_site1, temperature_site2, humidity_site2) VALUES (9,'01/03/2018 04:30:00',19.0,78.0,21.1,66.5);
INSERT 0 1
weatherdata=# select * from weather_table order by time desc limit 5;
 id |          time          | temperature_site1 | humidity_site1 | temperature_site2 | humidity_site2
-----+-----+-----+-----+-----+-----
  9 | 2018-03-01 04:30:00+05:30 |          19       |           78   |          21.1    |          66.5
  8 | 2018-03-01 04:00:00+05:30 |          19       |           78   |          21.5    |          65.2
  7 | 2018-03-01 03:30:00+05:30 |          19       |           78   |          21.5    |          64.7
  6 | 2018-03-01 03:00:00+05:30 |          19       |           78   |          21.7    |          64.3
  5 | 2018-03-01 02:30:00+05:30 |          20       |           70   |          22      |          63.8
(5 rows)

weatherdata=# select * from weather_table order by time asc limit 5;
 id |          time          | temperature_site1 | humidity_site1 | temperature_site2 | humidity_site2
-----+-----+-----+-----+-----+-----
  0 | 2018-03-01 00:00:00+05:30 |          21       |           68   |          23.1    |          61.6
  1 | 2018-03-01 00:30:00+05:30 |          20       |           73   |          22.9    |          61.1
  2 | 2018-03-01 01:00:00+05:30 |          20       |           73   |          23      |          61.6
  3 | 2018-03-01 01:30:00+05:30 |          20       |           73   |          22.9    |          62.5
  4 | 2018-03-01 02:00:00+05:30 |          20       |           73   |          22.4    |          63.4
(5 rows)

weatherdata=#
```