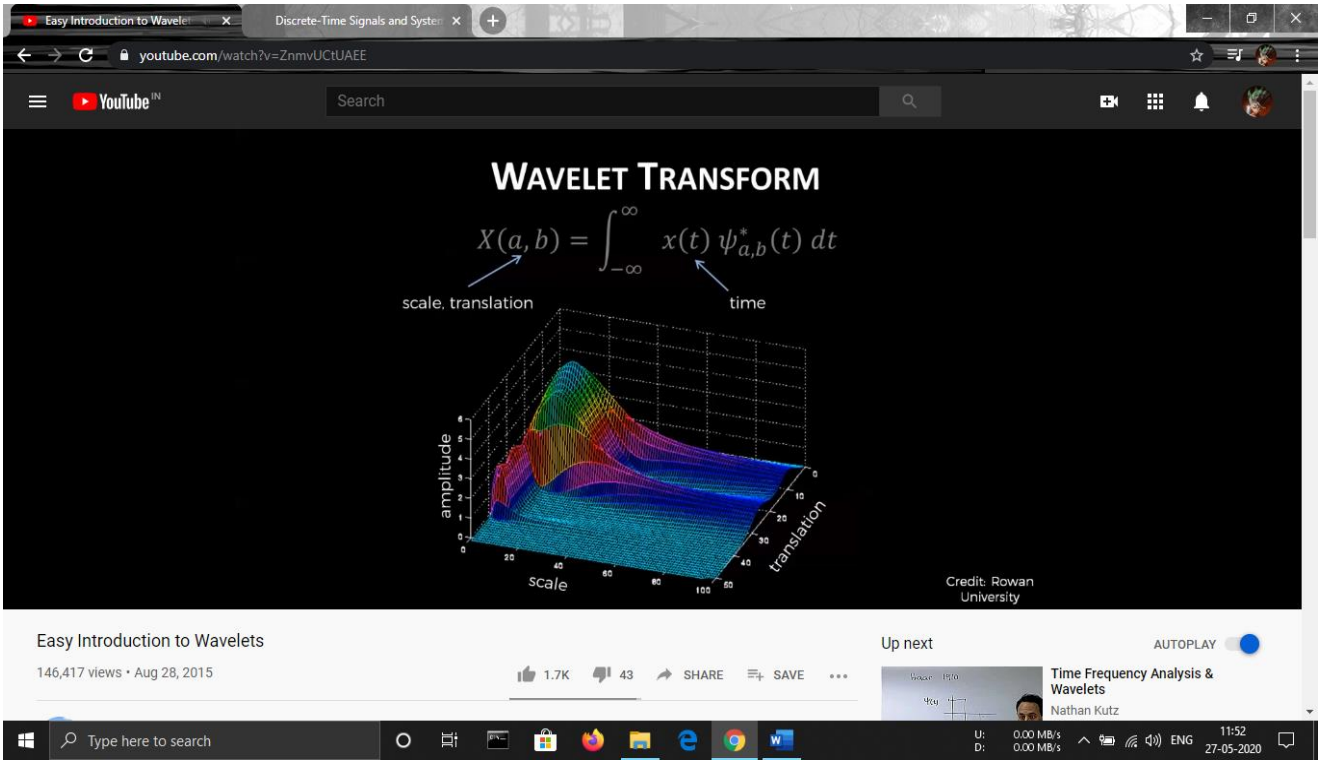
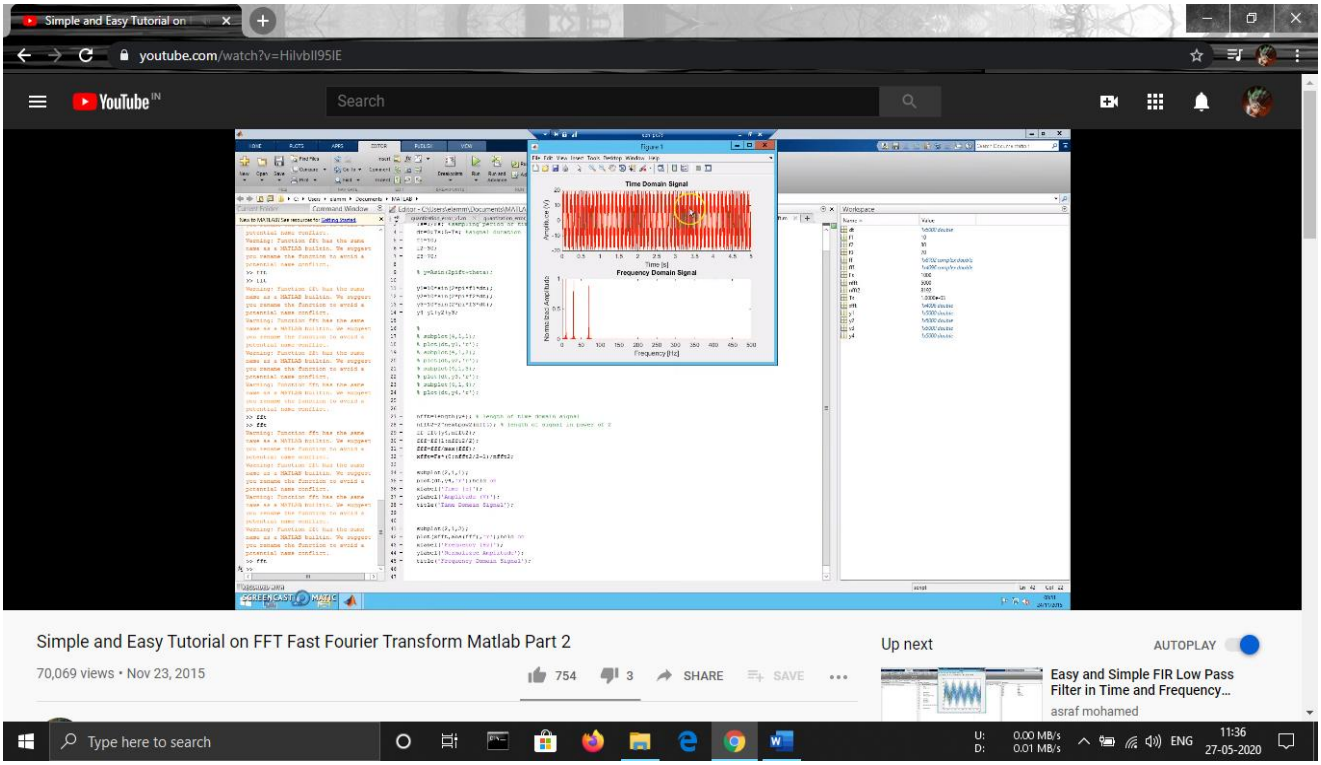
DAILY ASSESSMENT REPORT

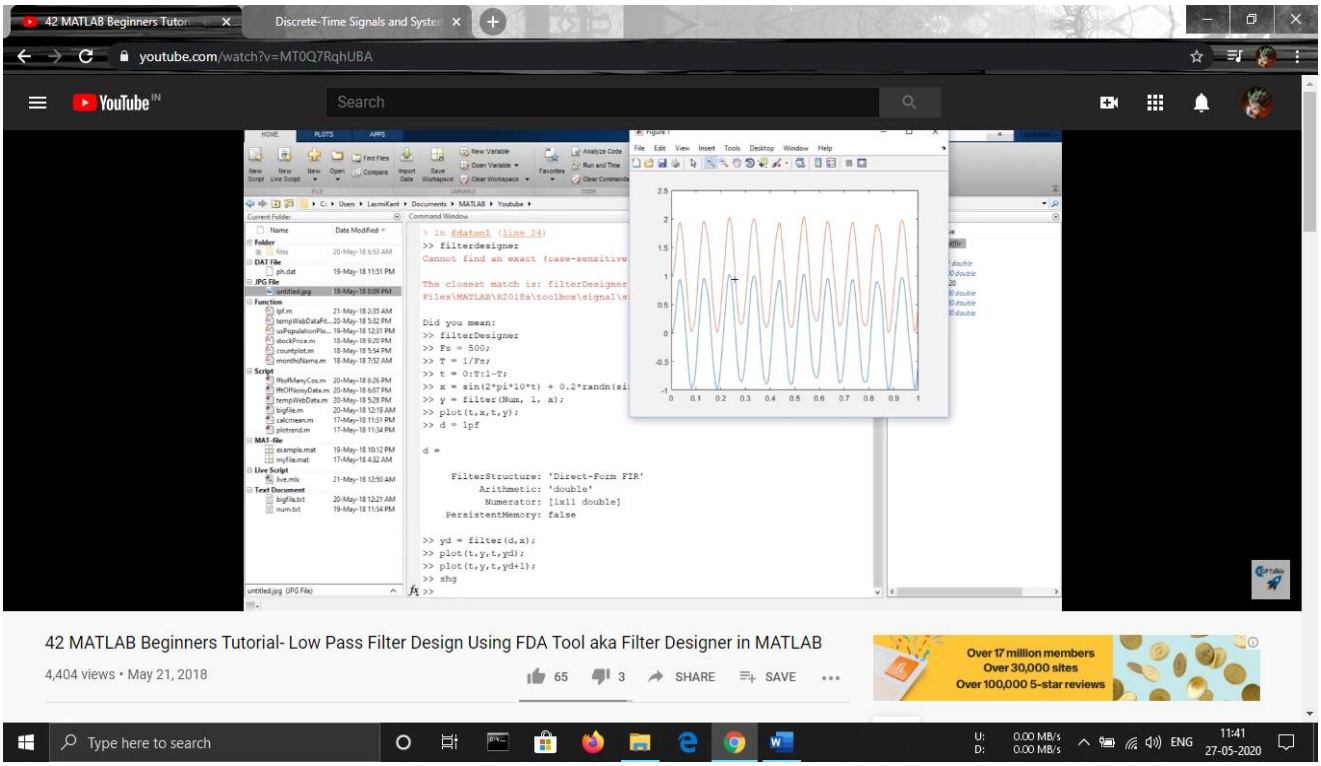
Date:	**27/05/2020**	**Name:**	**Abhishek M Shastry K**
Subject:	**Digital Signal Processing**	**USN:**	**4AL17EC002**
Topic:	**1] FFT, Fast Fourier Transform MATLAB** **2] FIR and IIR Filters** **3] Study and analysis FIR and IIR using FDA tool in MATLAB** **4] Introduction to WT** **5] CWT & DWT** **6] Implementation of signal Filtering signal using WT in MATLAB** **7] Short-time Fourier Transform and the Spectrogram** **8] Welch's method and windowing** **9] ECG Signal Analysis Using MATLAB**	**Semester & Section:**	**6th 'A'**
Github Repository:	**AbhishekShastry-Courses**		

FORENOON SESSION DETAILS

Image of session




Simple and Easy Tutorial on
youtube.com/watch?v=HiIvbII95IE
YouTube IN
Search
Time Domain Signal
Frequency Domain Signal
Simple and Easy Tutorial on FFT Fast Fourier Transform Matlab Part 2
70,069 views • Nov 23, 2015
754
3
SHARE
SAVE
Up next
AUTOPLAY
Easy and Simple FIR Low Pass Filter in Time and Frequency...
asraf mohamed
Type here to search
U: 0.00 MB/s
D: 0.01 MB/s
ENG
11:36
27-05-2020


42 MATLAB Beginners Tutori
Discrete-Time Signals and Syster
youtube.com/watch?v=MT0Q7RqhUBA
YouTube IN
Search
>> filterdesigner
>> Fs = 500;
>> T = 1/Fs;
>> t = 0:T:1-T;
>> x = sin(2*pi*10*t) + 0.2*randn(si
>> y = filter(Num, 1, x);
>> plot(t,x,t,y);
>> d = lpf
d =
FilterStructure: 'Direct-Form FIR'
Arithmetic: 'double'
Numerator: [1x11 double]
PersistentMemory: false
>> yd = filter(d,x);
>> plot(t,y,t,yd);
>> plot(t,y,t,yd+1);
>> shg
42 MATLAB Beginners Tutorial- Low Pass Filter Design Using FDA Tool aka Filter Designer in MATLAB
4,404 views • May 21, 2018
65
3
SHARE
SAVE
Over 17 million members
Over 30,000 sites
Over 100,000 5-star reviews
Type here to search
U: 0.00 MB/s
D: 0.00 MB/s
ENG
11:41
27-05-2020

Report

Fast Fourier Transform (FFT)

A **fast Fourier transform** (**FFT**) is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT).

Basically, there are two types of FFT algorithm.

- Decimation in time algorithm (DIT-FFT).

The radix-2 decimation-in-time algorithm rearranges the discrete Fourier transform (DFT) equation into two parts: a sum over the even-numbered discrete-time indices n=[0,2,4,…,N−2] and a sum over the odd-numbered indices n=[1,3,5,…,N−1] as in Equation:

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) e^{-\left(i\frac{2\pi nk}{N}\right)} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-\left(i\frac{2\pi\times(2n)k}{N}\right)} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-\left(i\frac{2\pi(2n+1)k}{N}\right)} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)} + e^{-\left(i\frac{2\pi k}{N}\right)} \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-\left(i\frac{2\pi nk}{\frac{N}{2}}\right)} \\ &= \mathrm{DFT}_{\frac{N}{2}}\left[[x(0), x(2), \ldots, x(N-2)]\right] + W_N^k \, \mathrm{DFT}_{\frac{N}{2}}\left[[x(1), x(3), \ldots, x(N-1)]\right] \end{aligned}$$

- Decimation in frequency algorithm (DIF-FFT).

The radix-2 decimation-in-frequency algorithm rearranges the discrete Fourier transform (DFT) equation into two parts: computation of the even-numbered discrete-frequency indices X(k) for k=[0,2,4,…,N−2] (or X(2r)) and computation of the odd numbered indices k=[1,3,5,…,N−1] (or X(2r+1)).

$$\begin{aligned} X(2r) &= \sum_{n=0}^{N-1} x(n) W_N^{2rn} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{2rn} + \sum_{n=0}^{\frac{N}{2}-1} x\left(n+\tfrac{N}{2}\right) W_N^{2r\left(n+\frac{N}{2}\right)} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{2rn} + \sum_{n=0}^{\frac{N}{2}-1} x\left(n+\tfrac{N}{2}\right) W_N^{2rn} 1 \\ &= \sum_{n=0}^{\frac{N}{2}-1} \left(x(n) + x\left(n+\tfrac{N}{2}\right)\right) W_{\frac{N}{2}}^{rn} \\ &= \mathrm{DFT}_{\frac{N}{2}}\left[x(n) + x\left(n+\tfrac{N}{2}\right)\right] \\ X(2r+1) &= \sum_{n=0}^{N-1} x(n) W_N^{(2r+1)n} \\ &= \sum_{n=0}^{\frac{N}{2}-1} \left(x(n) + W_N^{\frac{N}{2}} x\left(n+\tfrac{N}{2}\right)\right) W_N^{(2r+1)n} \\ &= \sum_{n=0}^{\frac{N}{2}-1} \left(\left(x(n) - x\left(n+\tfrac{N}{2}\right)\right) W_N^n\right) W_{\frac{N}{2}}^{rn} \\ &= \mathrm{DFT}_{\frac{N}{2}}\left[\left(x(n) - x\left(n+\tfrac{N}{2}\right)\right) W_N^n\right] \end{aligned}$$

FIR and IIR Filters

- In signal processing, a finite impulse response (FIR) filter is a filter whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time.
- An infinite impulse response (IIR) filter is a digital filter that depends linearly on a finite number of input samples and a finite number of previous filter outputs.
- The crucial difference between FIR and IIR filter is that the FIR filter provides an impulse response of finite period. As against IIR is a type of filter that generates impulse response of infinite duration for a dynamic system.

DWT and CWT

- The Discrete Wavelet Transform (DWT), simply put, is an operation that receives a signal as an input (a vector of data) and decomposes it in its frequential components.
- the Continuous Wavelet Transform (CWT) is a formal (i.e., non-numerical) tool that provides an overcomplete representation of a signal by letting the translation and scale parameter of the wavelets vary continuously.

Welch's method and windowing

Welch's method [296] (also called the periodogram method) for estimating power spectra is carried out by dividing the time signal into successive blocks, forming the periodogram for each block, and averaging.

Denote the m^{th} windowed, zero-padded frame from the signal x by,

$$x_m(n) \stackrel{\Delta}{=} w(n)x(n+mR), \quad n = 0, 1, \ldots, M-1, \; m = 0, 1, \ldots, K-1,$$

where R is defined as the window hop size, and let K denote the number of available frames. Then the periodogram of the m^{th} block is given by,

$$P_{x_m,M}(\omega_k) = \frac{1}{M} \left| \mathrm{FFT}_{N,k}(x_m) \right|^2 \stackrel{\Delta}{=} \frac{1}{M} \left| \sum_{n=0}^{N-1} x_m(n) e^{-j2\pi nk/N} \right|^2$$

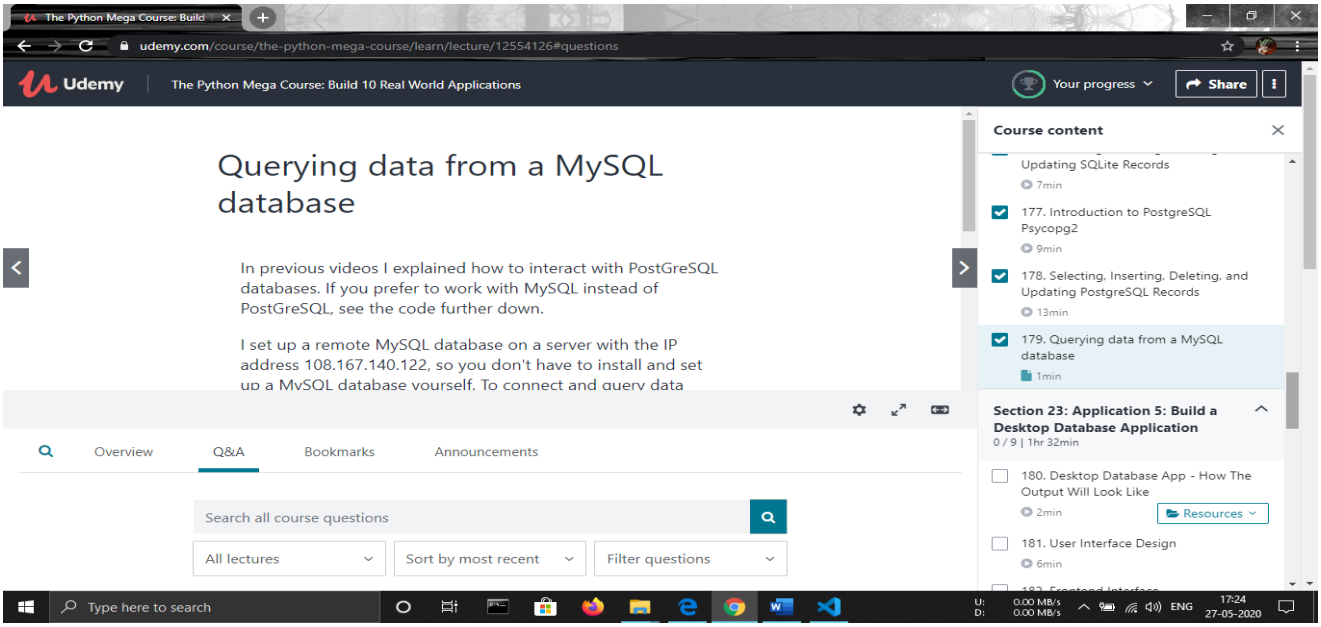
as before, and the Welch estimate of the power spectral density is given by

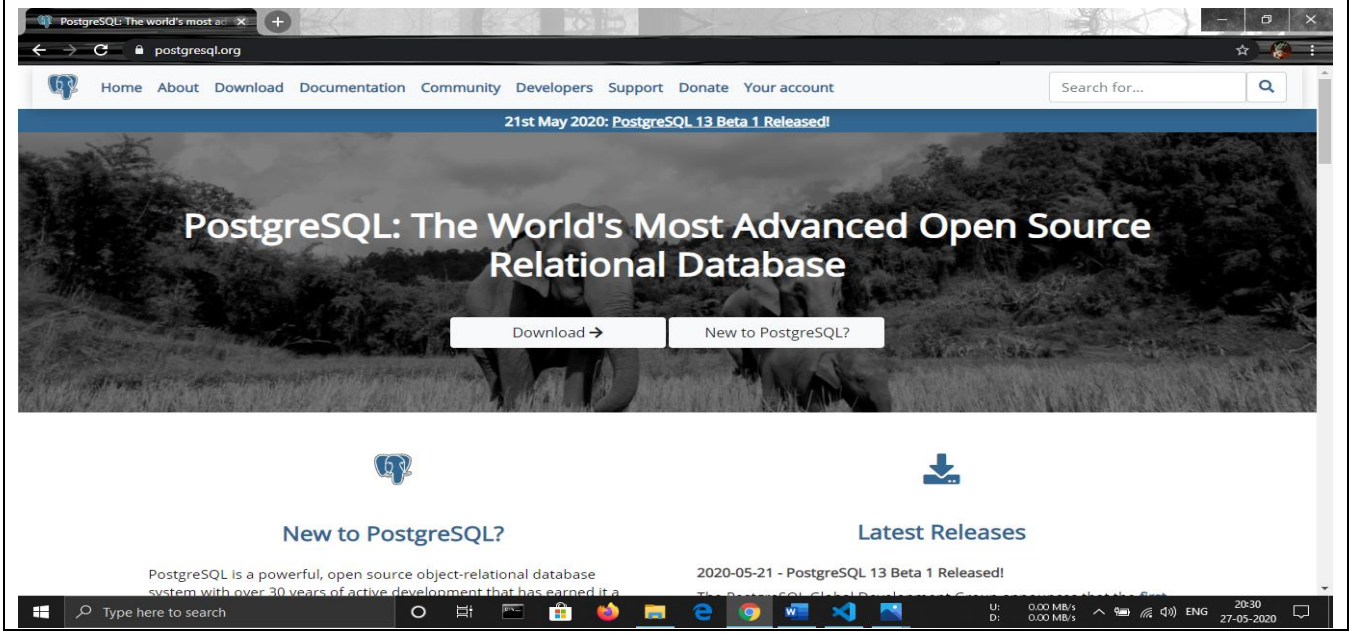
$$\hat{S}_x^W(\omega_k) \stackrel{\Delta}{=} \frac{1}{K} \sum_{m=0}^{K-1} P_{x_m,M}(\omega_k).$$

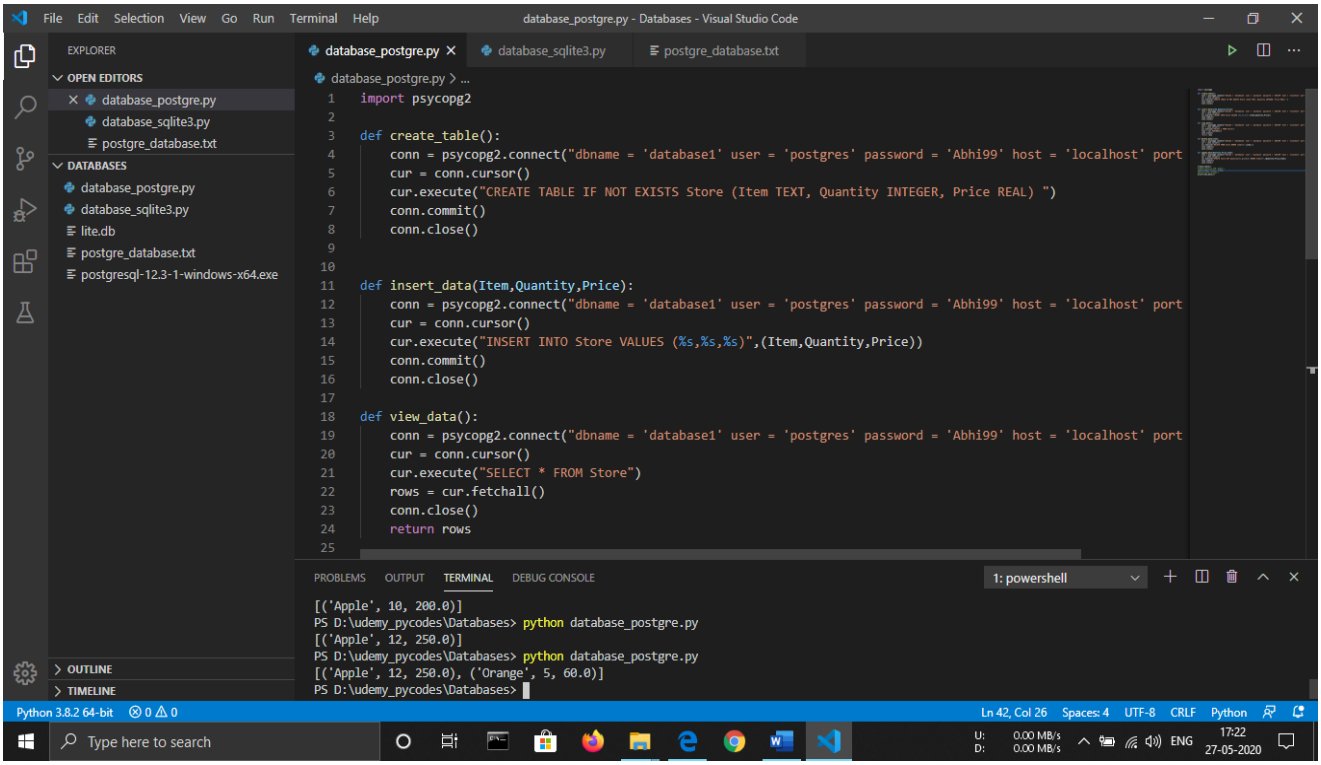
Date:	27/05/2020	Name:	Abhishek M Shastry K
Course:	The Python Mega Course: Build 10 Real World Applications	USN:	4AL17EC002
Topic:	1] Interacting with Databases	Semester & Section:	6th 'A'
Github Repository:	AbhishekShastry-Courses		

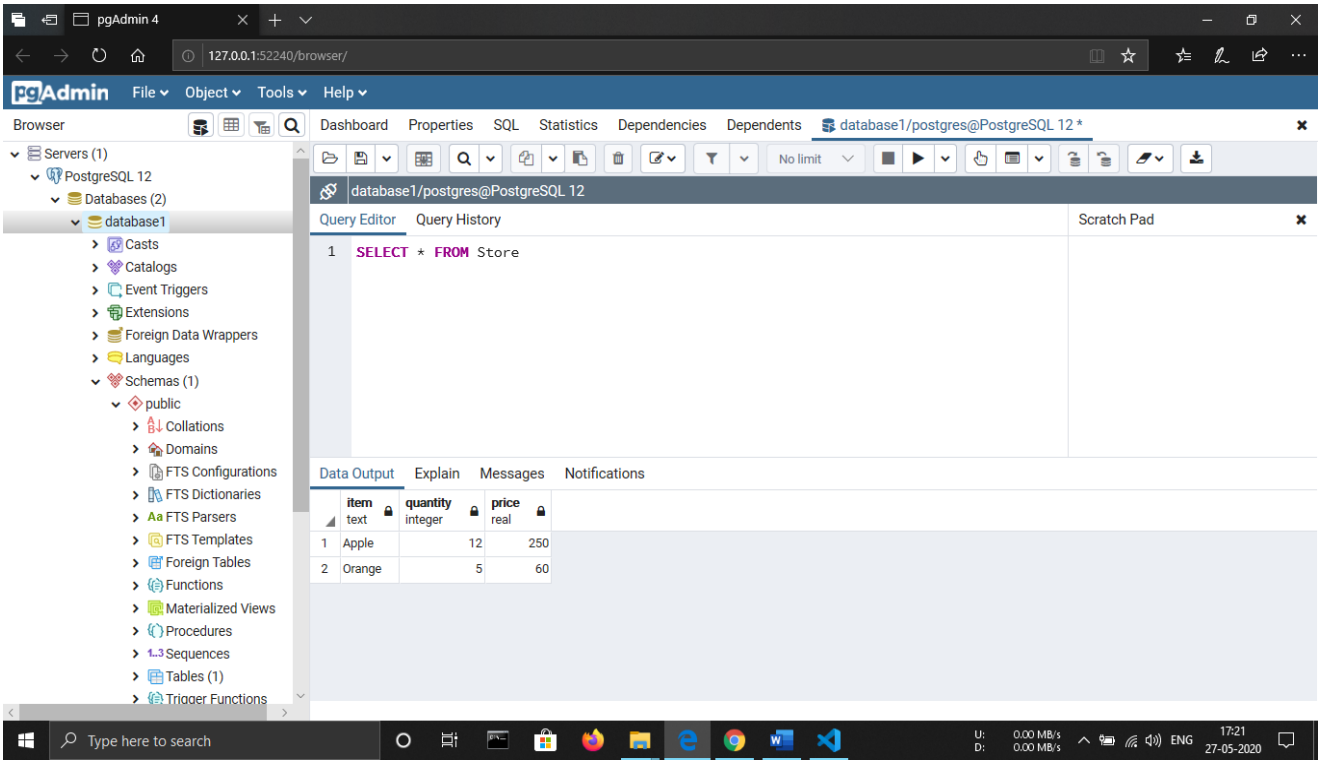
AFTERNOON SESSION DETAILS

Image of session






File Edit Selection View Go Run Terminal Help
database_postgre.py - Databases - Visual Studio Code
EXPLORER
OPEN EDITORS
database_postgre.py
database_sqlite3.py
postgre_database.txt
DATABASES
database_postgre.py
database_sqlite3.py
lite.db
postgre_database.txt
postgresql-12.3-1-windows-x64.exe
OUTLINE
TIMELINE
database_postgre.py
database_sqlite3.py
postgre_database.txt
database_postgre.py > ...
1 import psycopg2
2
3 def create_table():
4 conn = psycopg2.connect("dbname = 'database1' user = 'postgres' password = 'Abhi99' host = 'localhost' port
5 cur = conn.cursor()
6 cur.execute("CREATE TABLE IF NOT EXISTS Store (Item TEXT, Quantity INTEGER, Price REAL) ")
7 conn.commit()
8 conn.close()
9
10
11 def insert_data(Item,Quantity,Price):
12 conn = psycopg2.connect("dbname = 'database1' user = 'postgres' password = 'Abhi99' host = 'localhost' port
13 cur = conn.cursor()
14 cur.execute("INSERT INTO Store VALUES (%s,%s,%s)",(Item,Quantity,Price))
15 conn.commit()
16 conn.close()
17
18 def view_data():
19 conn = psycopg2.connect("dbname = 'database1' user = 'postgres' password = 'Abhi99' host = 'localhost' port
20 cur = conn.cursor()
21 cur.execute("SELECT * FROM Store")
22 rows = cur.fetchall()
23 conn.close()
24 return rows
25
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: powershell
[('Apple', 10, 200.0)]
PS D:\udemy_pycodes\Databases> python database_postgre.py
[('Apple', 12, 250.0)]
PS D:\udemy_pycodes\Databases> python database_postgre.py
[('Apple', 12, 250.0), ('Orange', 5, 60.0)]
PS D:\udemy_pycodes\Databases>
Python 3.8.2 64-bit
Ln 42, Col 26 Spaces: 4 UTF-8 CRLF Python
Type here to search
ENG 17:22 27-05-2020


pgAdmin 4
127.0.0.1:52240/browser/
pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents database1/postgres@PostgreSQL 12 *
Servers (1)
PostgreSQL 12
Databases (2)
database1
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (1)
Trigger Functions
No limit
database1/postgres@PostgreSQL 12
Query Editor Query History
Scratch Pad
1 SELECT * FROM Store
Data Output Explain Messages Notifications
item text
quantity integer
price real
1 Apple 12 250
2 Orange 5 60
Type here to search
ENG 17:21 27-05-2020

Report

Interacting with Databases

- A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.
- Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.
- **SQLite** is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.
- The **PostgreSQL** can be integrated with Python using **psycopg2** module. sycopg2 is a PostgreSQL database adapter for the Python programming language. psycopg2 was written with the aim of being very small and fast, and stable as a rock.
- Standard process of interacting with database consists of five steps. They are:
    - ✓ Connect to database.
    - ✓ Create a cursor object.
    - ✓ Write an SQL query.
    - ✓ Commit changes.
    - ✓ Close database connection.
- To create a new table in SQLite and PostgreSQL, CREATE TABLE statement is used.
- **Cursor** is a Temporary Memory or Temporary Work Station. It is Allocated by Database Server at the Time of Performing DML operations on Table by User. Cursors are used to store Database Tables.
- **Selecting**, **Inserting**, **Updating** and **Deleting** SQLite records and PostgreSQL records can be done using SELECT, INSERT, UPDATE AND DELETE SQL commands respectively.