Project p01: Sales Data Manager

Please use the following project naming convention and directory structure, where g## is your group number.

pyproj20251/

- . venv/
- p01_sales_g##/
    - p01_files
        - all_sales.csv
        - all_sales_copy.csv
        - imported_files.txt
        - sales_q1_2021_x.csv
        - sales_q2_2021_w.csv
        - sales_q3_2021_w.csv
        - sales_q4_2021_w.csv
    - p01sc01_control_structures
        - p01_m_sales_input.py
        - p01beg_m_sales_input.py
    - p01sc02_function_files
        - p01_m_sales.py
        - p01beg_m_sales.py
    - p01sc04_exception_libraries_3tier
        - p01_1da_sales.py
        - p01_2bl_salesmanager.py
        - p01_3ui_console.py
        - p01beg_1da_sales.py
        - p01beg_2bl_salesmanager.py
        - p01beg_3ui_console.py
    - p01sc06_OOPDBGUI3tier
        - p01_1da_sales.py
        - p01_1da_sales_db.py
        - p01_2bl_salesmanager.py
        - p01_3gui_tkinter.py
        - p01_3ui_console.py
        - p01beg_1da_sales.py
        - p01beg_1db_sales_db.py
        - p01beg_2bl_salesmanager.py
        - p01beg_3gui_tkinter.py
        - p01beg_3ui_console.py

The original content of the input files is shown below:

all_sales.csv

```
12493.0,2020-12-22,w
13761.0,2021-09-15,e
9710.0,2021-05-15,e
8934.0,2021-08-08,c
18340.0,2020-12-22,c
12345.0,2020-04-17,m
2929.0,2021-04-10,w
```

imported_files.txt

```
```

The original content of the tables of **sales_db.sqlite** is shown below:

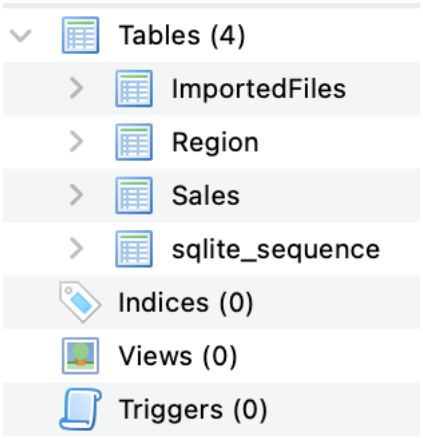


Table: ImportedFiles

	fileName
	Filter
1	sales_q1_2021_w.csv

Table: Region

	code	name
	Filter	Filter
1	w	West
2	m	Mountain
3	c	Central
4	e	East

Table: Sales

	ID	amount	salesDate	region
	...	Filter	Filter	Filter
1	1	23456.0	2021-12-22	w
2	2	12365.0	2021-09-09	e
3	3	23757.0	2020-11-11	e
4	4	12549.0	2020-12-12	m
5	5	39393.0	2021-02-02	w

Table: sqlite_sequence

	name	seq
	Filter	Fil...
1	Sales	6

You are asked to make the following enhancements or modifications to the original application.

- Add an ID in each sales data record
- Use OOP to re-write the original application that is written in functional programming
- Add a module to create a GUI using tkinter library for user to read records from and update records in the SQLite database. Also, add a module to handle operations on the sales records in the SQLite database.

In the first phrase of enhancements or modifications to the original application. You are asked to

- Add an ID in each sales data record
- Use OOP to re-write the original application that is written in functional programming

Assume that content of the input files is as same as the original content shown before.
The first execution of the application written in OOP:

SALES DATA IMPORTER

COMMAND MENU
view - View all sales
add1 - Add sales by typing sales, year, month, day, and region
add2 - Add sales by typing sales, date (YYYY-MM-DD), and region
import - Import sales from file
menu - Show menu
exit - Exit program

Please enter a command: **view**

```
     Date           Quarter        Region                   Amount
-------------------------------------------------------------------
1.   2020-12-22     4              West                  $12,493.00
2.   2021-09-15     3              East                  $13,761.00
3.   2021-05-15     2              East                   $9,710.00
4.   2021-08-08     3              Central                $8,934.00
5.   2020-12-22     4              Central               $18,340.00
6.   2020-04-17     2              Mountain              $12,345.00
7.   2021-04-10     2              West                   $2,929.00
-------------------------------------------------------------------
TOTAL                                                   $78,512.00
```

view_sales: DataFileAccess.SALES_ID['Sales']=8

Please enter a command: **add1**
Amount: **0**
Amount must be greater than zero.
Amount: **3245**
Year (2000-2999): 0
Year must be between 2000 and 2999.
Year (2000-2999): **3000**

Year must be between 2000 and 2999.
Year (2000-2999): **2021**
Month (1-12): **0**
Month must be between 1 and 12.
Month (1-12): **20**
Month must be between 1 and 12.
Month (1-12): **2**
Day (1-28): **0**
Day must be between 1 and 28.
Day (1-28): **40**
Day must be between 1 and 28.
Day (1-28): **14**
Region ('w', 'm', 'c', 'e'):**x**
Region must be one of the following: ('w', 'm', 'c', 'e').
Region ('w', 'm', 'c', 'e'):**c**
Sales for 2021-02-14 is added.

add_sales1: DataFileAccess.SALES_ID['Sales']=9

Please enter a command: **add2**
Amount: **4324**
Date (yyyy-mm-dd): **0021-08-14**
Year of the date must be between 2000 and 2999.
Date (yyyy-mm-dd): **202a**
202a is not in a valid date format.
Date (yyyy-mm-dd): **2021-8-14**
Region ('w', 'm', 'c', 'e'):**e**
Sales for 2021-08-14 is added.

add_sales2: DataFileAccess.SALES_ID['Sales']=10

Please enter a command: **view**

	Date	Quarter	Region	Amount
1.	2020-12-22	4	West	$12,493.00
2.	2021-09-15	3	East	$13,761.00
3.	2021-05-15	2	East	$9,710.00
4.	2021-08-08	3	Central	$8,934.00
5.	2020-12-22	4	Central	$18,340.00
6.	2020-04-17	2	Mountain	$12,345.00
7.	2021-04-10	2	West	$2,929.00
8.	2021-02-14	1	Central	$3,245.00
9.	2021-08-14	3	East	$4,324.00
TOTAL				$86,081.00

view_sales: DataFileAccess.SALES_ID['Sales']=10

Please enter a command: **sales**

Invalid command. Please try again.

COMMAND MENU
view - View all sales
add1 - Add sales by typing sales, year, month, day, and region
add2 - Add sales by typing sales, date (YYYY-MM-DD), and region
import - Import sales from file
menu - Show menu
exit - Exit program

Please enter a command: **import**
Enter name of file to import: **region1**
Filename 'region1' doesn't follow the expected format of 'sales_qn_yyyy_r.csv'.

Please enter a command: **import**
Enter name of file to import: **sales_q1_2021_x.csv**
Filename 'sales_q1_2021_x.csv' doesn't include one of the following region codes: ['w', 'm', 'c', 'e'].

Please enter a command: **import**
Enter name of file to import: **sales_q1_2021_w.csv**
<class 'FileNotFoundError'>. Fail to import sales from 'sales_q1_2021_w.csv'.

Please enter a command: **import**
Enter name of file to import: **sales_q2_2021_w.csv**
No sales to view.

Please enter a command: **import**
Enter name of file to import: **sales_q3_2021_w.csv**

	Date	Quarter	Region	Amount
1.	2021-07-15	3	West	$13,761.00
2.*	?	0	West	$9,710.00
3.*	2021-09-15	3	West	?
TOTAL				$23,471.00

view_sales: DataFileAccess.SALES_ID['Sales']=10
File 'sales_q3_2021_w.csv' contains bad data.
Please correct the data in the file and try again.

Please enter a command: **import**
Enter name of file to import: **sales_q4_2021_w.csv**

	Date	Quarter	Region	Amount
1.	2021-10-15	4	West	$13,761.00
2.	2021-11-15	4	West	$9,710.00
3.	2021-12-15	4	West	$8,934.00

```
TOTAL                                                        $32,405.00
```

view_sales: DataFileAccess.SALES_ID['Sales']=10
Imported sales added to list.

Please enter a command: **view**

```
     Date            Quarter        Region                    Amount
-------------------------------------------------------------------
1.   2020-12-22     4              West                   $12,493.00
2.   2021-09-15     3              East                   $13,761.00
3.   2021-05-15     2              East                    $9,710.00
4.   2021-08-08     3              Central                 $8,934.00
5.   2020-12-22     4              Central                $18,340.00
6.   2020-04-17     2              Mountain               $12,345.00
7.   2021-04-10     2              West                    $2,929.00
8.   2021-02-14     1              Central                 $3,245.00
9.   2021-08-14     3              East                    $4,324.00
10.  2021-10-15     4              West                   $13,761.00
11.  2021-11-15     4              West                    $9,710.00
12.  2021-12-15     4              West                    $8,934.00
-------------------------------------------------------------------
TOTAL                                                    $118,486.00
```

view_sales: DataFileAccess.SALES_ID['Sales']=13

Please enter a command: **menu**
COMMAND MENU
view   - View all sales
add1   - Add sales by typing sales, year, month, day, and region
add2   - Add sales by typing sales, date (YYYY-MM-DD), and region
import - Import sales from file
menu   - Show menu
exit   - Exit program

Please enter a command: **exit**
Saved sales records.
Bye!

The content of the output files after the first execution of enhanced application:

all_sales.csv

```
12493.0,2020-12-22,w
13761.0,2021-09-15,e
9710.0,2021-05-15,e
8934.0,2021-08-08,c
18340.0,2020-12-22,c
12345.0,2020-04-17,m
2929.0,2021-04-10,w
3245.0,2021-02-14,c
4324.0,2021-08-14,e
13761.0,2021-10-15,w
9710.0,2021-11-15,w
8934.0,2021-12-15,w
```

imported_files.txt

```
/Users/hyan/Documents/ws_GitHub/hyan-aum/hyan_aum_nonpubpy/PyM00_00CodePractices/p01_salesimporter/p01_files/sales_q4_2021_w.csv
```

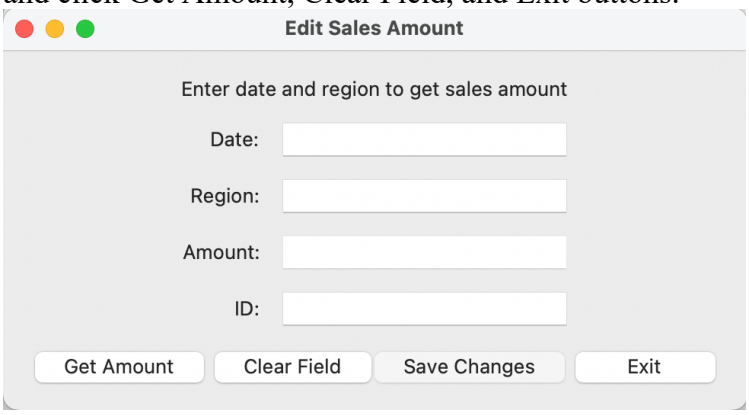
Note: make sure you copy the sales data from all_sales_copy.csv to all_sales.csv, and remove the content in the imported_files.txt for any future test.

In the second phrase of enhancements or modifications to the application. You are asked to

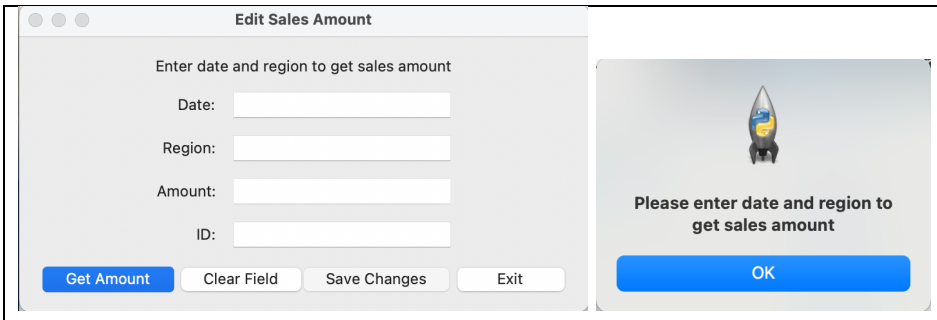
- Add a module to create a GUI for user to edit Sales Amount records in the SQLite database. Also, add a module to handle operations on the sales records in the SQLite database.

Assume that content of the database is as same as the original content shown before.
The execution of the application that uses GUI and SQLite DB:

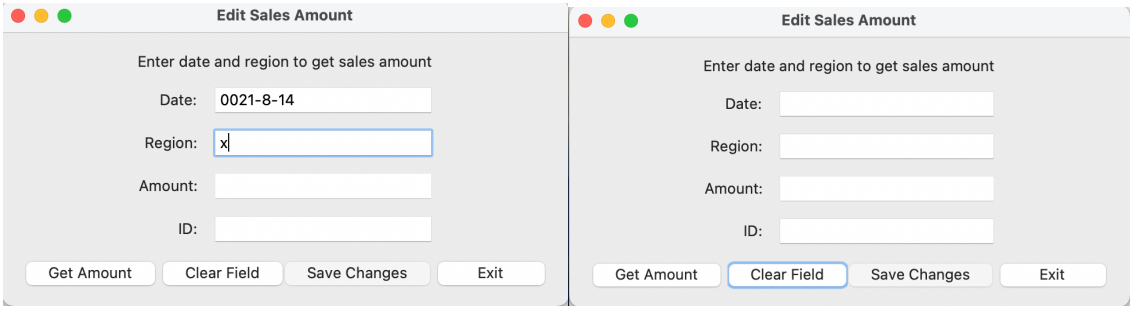
Initially, whenever the application starts, user may enter data in Date and Region text fields, and click Get Amount, Clear Field, and Exit buttons.



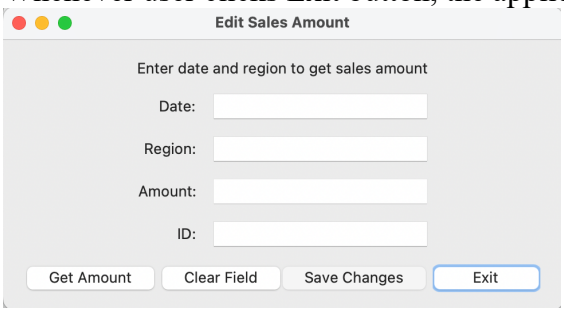
If user clicks Get Amount button without providing data and region information, an error message box will be shown.



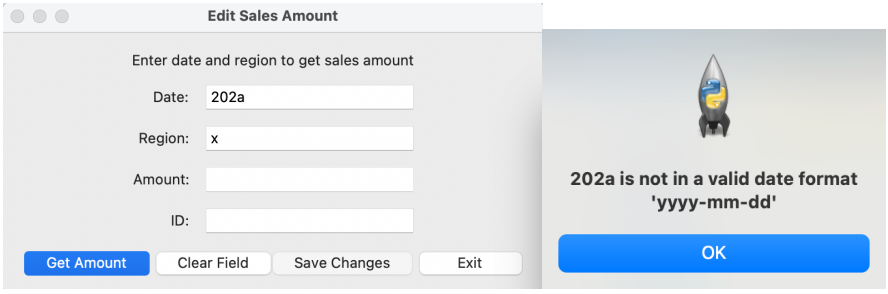
Whenever user clicks Clear Field button, the information user entered in Date and Region text fields will be removed.



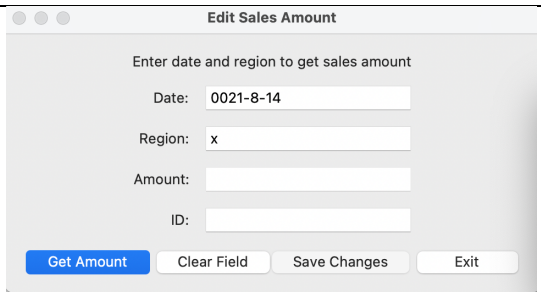
Whenever user clicks Exit button, the application terminates normally.



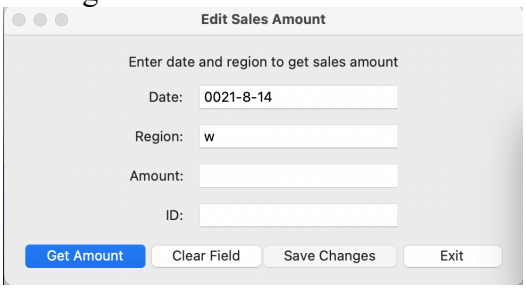
Note that the error checking required in the GUI application for the practice is much simpler than what we did in the console application. You are only asked to check the following cases.
Case1: valid date format ‘yyyy-mm-dd’. You don’t need to check if the year is between 2000 and 2999.

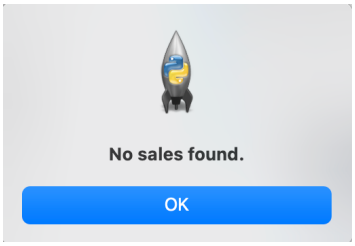


Case 2: valid region code.

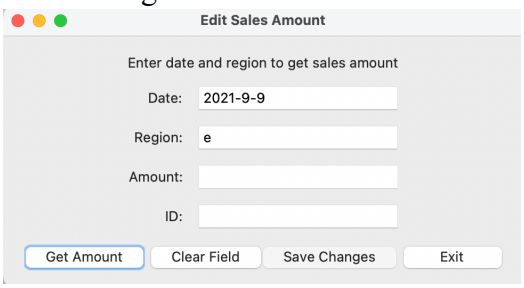


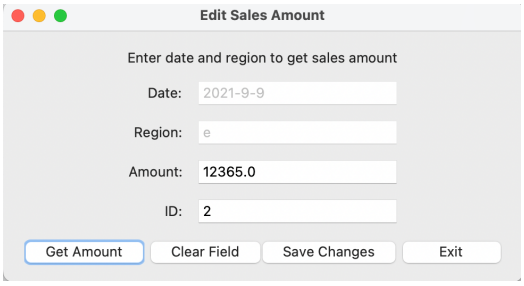
After the user provides both Date and Region information and clicks the Get Amount button, an error message will be shown if no sales record is found in the database for the given date and region information.



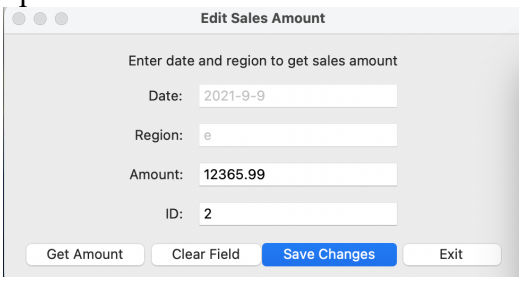


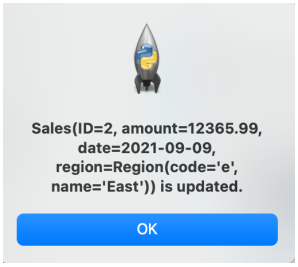
After the user provides both Date and Region information and clicks the Get Amount button, the corresponding Amount and ID will be shown if a sales record is found in the database for the given date and region information. Meanwhile, the Amount field becomes editable, and the Save Changes button becomes available/clickable.





After the user enter new Amount and click the Save Change button, the application will update the sales record in the database and display an information to confirm that the sales record is updated.





If you open the Sales table of the sales_db.sqlite database in the DB Browser for SQLite, you will see the corresponding sales record is updated.

Table: Sales

	ID	amount	salesDate	region
...	Filter	Filter	Filter	Filter
1	1	23456.0	2021-12-22	w
2	2	12366.99	2021-09-09	e
3	3	23767.0	2020-11-11	e
4	4	12549.0	2020-12-12	m
5	5	39393.0	2021-02-02	w