Assignment 4- Databases, Exceptions and Software testing

Jasmine Bajracharya[u3312590]

Question 1

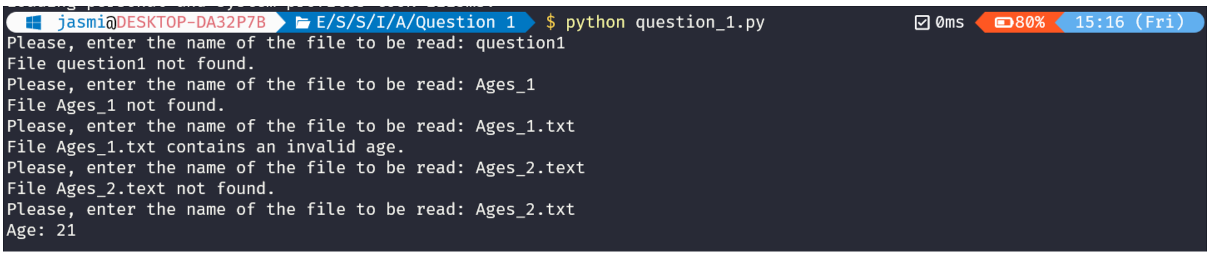
Solution:

The code for solution 1 is given as :

```
def main():
    while True:
        file_name = input('Please, enter the name of the file to be read: ')
        try:
            with open(file_name, 'r') as file:
                content = file.read().strip()
                age = int(content)
                print(f'Age: {age}')
            break
        except FileNotFoundError:
            print(f'File {file_name} not found.')
        except ValueError:
            print(f'File {file_name} contains an invalid age.')
        except Exception as e:
            print(f'An unexpected error occurred: {e}')
if __name__ == '__main__':
    main()
```

The solution returns and invalid age message when the age is not in number and returns the age only when the age is a number:

The output from solution 1 is obtained as follows:



```
jasmi@DESKTOP-DA32P7B  E/S/S/I/A/Question 1  $ python question_1.py          ☑ 0ms  80%  15:16 (Fri)
Please, enter the name of the file to be read: question1
File question1 not found.
Please, enter the name of the file to be read: Ages_1
File Ages_1 not found.
Please, enter the name of the file to be read: Ages_1.txt
File Ages_1.txt contains an invalid age.
Please, enter the name of the file to be read: Ages_2.text
File Ages_2.text not found.
Please, enter the name of the file to be read: Ages_2.txt
Age: 21
```

Question 2

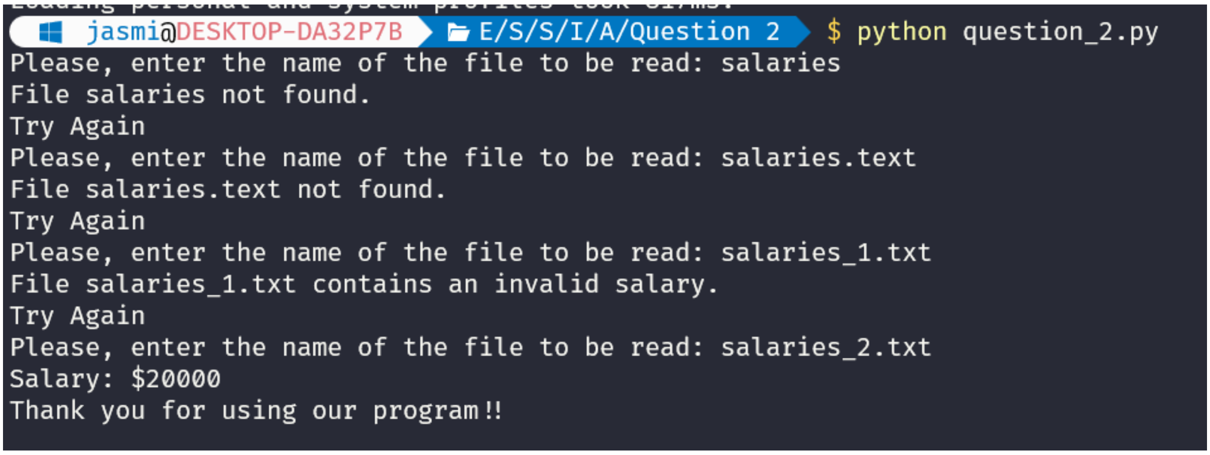
Solution:

The code for question 2 is as:

```
def main():

    while True:

        file_name = input('Please, enter the name of the file to be read: ')


        try:

            with open(file_name, 'r') as file:

                content = file.read().strip()

                age = int(content)

                print(f'Salary: ${age}')

                print('Thank you for using our program!!')


            break

        except FileNotFoundError:

            print(f'File {file_name} not found.')

            print('Try Again')

        except ValueError:

            print(f'File {file_name} contains an invalid salary.')

            print('Try Again')

        except Exception as e:

            print(f'An unexpected error occurred: {e}')

            print('Try Again')
```

```
if __name__ == '__main__':

    main()
```

The output from the program is obtained as:



```
 jasmi@DESKTOP-DA32P7B  E/S/S/I/A/Question 2  $ python question_2.py
Please, enter the name of the file to be read: salaries
File salaries not found.
Try Again
Please, enter the name of the file to be read: salaries.text
File salaries.text not found.
Try Again
Please, enter the name of the file to be read: salaries_1.txt
File salaries_1.txt contains an invalid salary.
Try Again
Please, enter the name of the file to be read: salaries_2.txt
Salary: $20000
Thank you for using our program!!
```

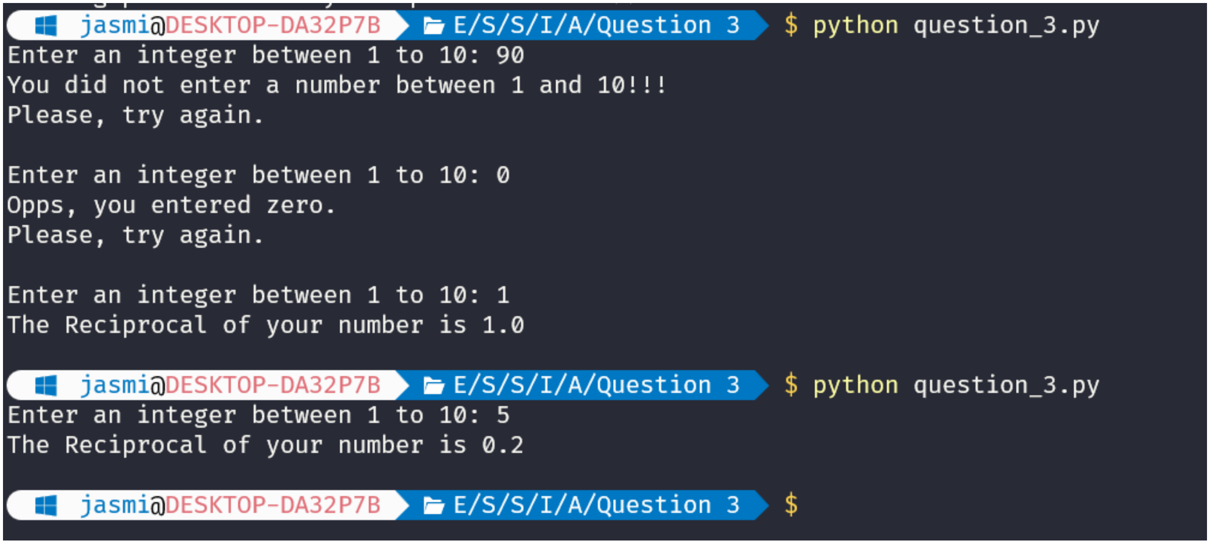
Question 3

Solution:

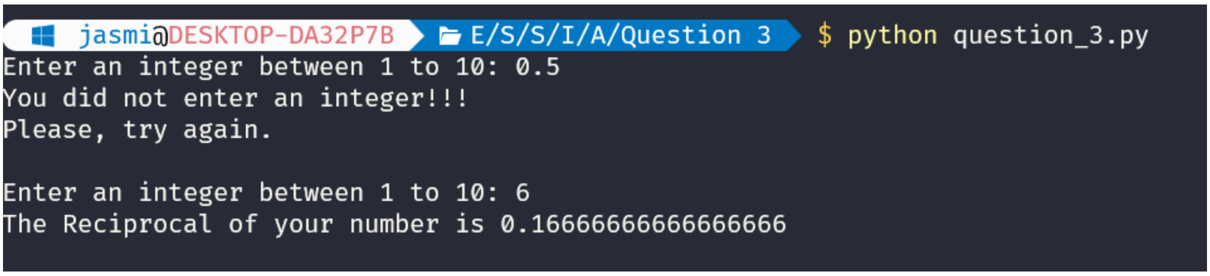
Assert is used in solution 3 to prevent the use of if..else conditions. Assert checks if the value is True or not and AssertionError raises an exception when the value is False.

```
def main():

    while True:

        try:

            value = int(input("Enter an integer between 1 to 10: "))


            # Used assert to prevent the use of if..else statement as using ValueError
requires using if..else statement.


            assert value != 0, "Opps, you entered zero."

            assert 1 <= value <= 10, "You did not enter a number between 1 and 10!!!"


            print(f"The Reciprocal of your number is {1 / value}")
            break
        except AssertionError as e:
            print(e)
            print('Please, try again.\n')
        except ValueError:
            print('You did not enter an integer!!!')
            print('Please, try again.\n')
        except Exception as e:
            print(f'An unexpected error occurred: {e}')
            print('Please, try again.\n')


if __name__ == '__main__':
    main()
```

The output obtained is:



```
 jasmi@DESKTOP-DA32P7B  E/S/S/I/A/Question 3  $ python question_3.py
Enter an integer between 1 to 10: 90
You did not enter a number between 1 and 10!!!
Please, try again.

Enter an integer between 1 to 10: 0
Opps, you entered zero.
Please, try again.

Enter an integer between 1 to 10: 1
The Reciprocal of your number is 1.0

 jasmi@DESKTOP-DA32P7B  E/S/S/I/A/Question 3  $ python question_3.py
Enter an integer between 1 to 10: 5
The Reciprocal of your number is 0.2

 jasmi@DESKTOP-DA32P7B  E/S/S/I/A/Question 3  $
```

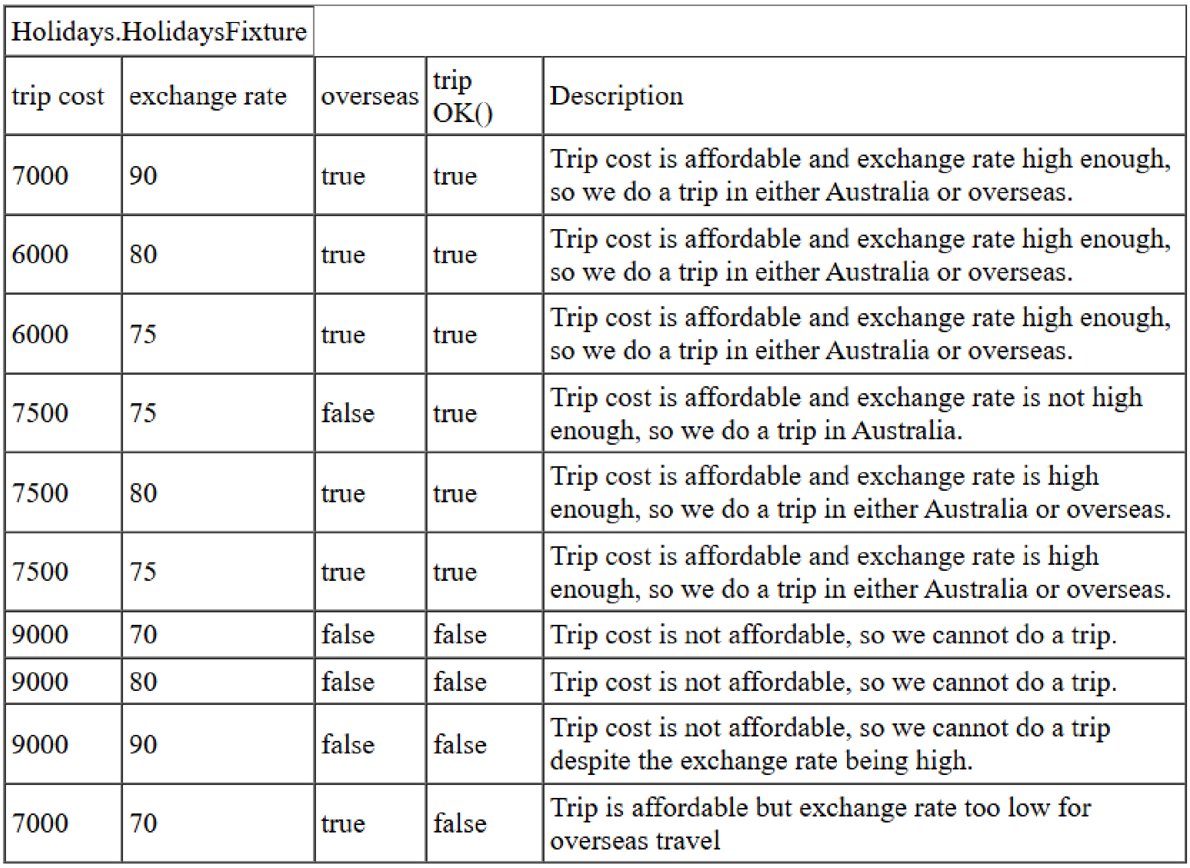


```
 jasmi@DESKTOP-DA32P7B  E/S/S/I/A/Question 3  $ python question_3.py
Enter an integer between 1 to 10: 0.5
You did not enter an integer!!!
Please, try again.

Enter an integer between 1 to 10: 6
The Reciprocal of your number is 0.16666666666666666
```

Question 4

Solution:

The following are the test cases for the conditions mentioned in the file Holiday.html

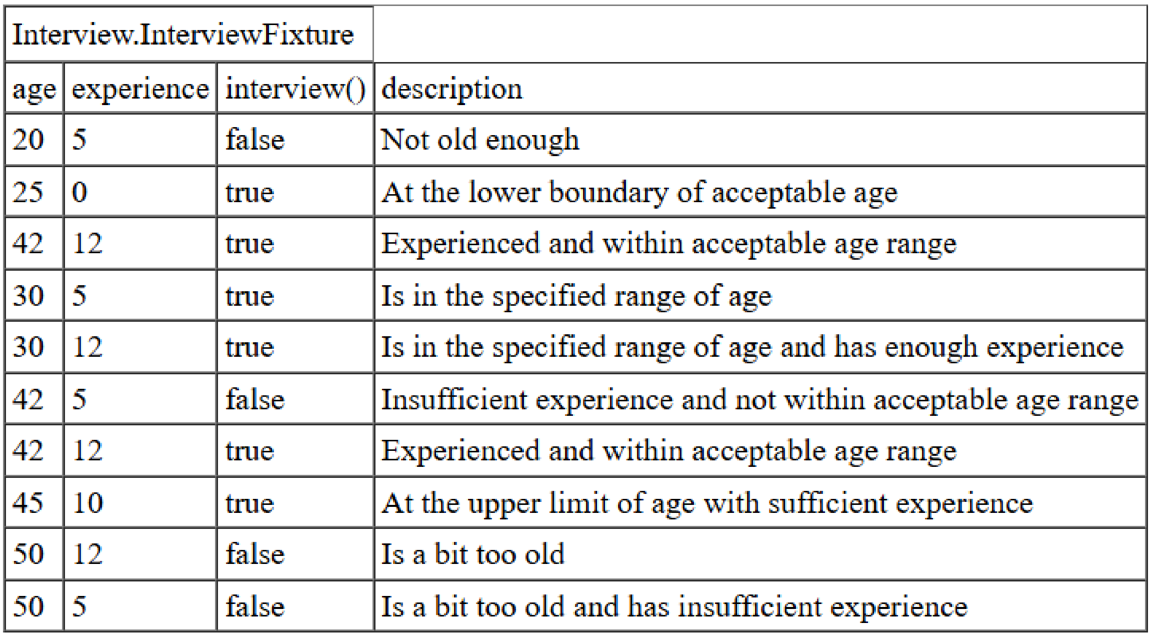


Holidays.HolidaysFixture				
trip cost	exchange rate	overseas	trip OK()	Description
7000	90	true	true	Trip cost is affordable and exchange rate high enough, so we do a trip in either Australia or overseas.
6000	80	true	true	Trip cost is affordable and exchange rate high enough, so we do a trip in either Australia or overseas.
6000	75	true	true	Trip cost is affordable and exchange rate high enough, so we do a trip in either Australia or overseas.
7500	75	false	true	Trip cost is affordable and exchange rate is not high enough, so we do a trip in Australia.
7500	80	true	true	Trip cost is affordable and exchange rate is high enough, so we do a trip in either Australia or overseas.
7500	75	true	true	Trip cost is affordable and exchange rate is high enough, so we do a trip in either Australia or overseas.
9000	70	false	false	Trip cost is not affordable, so we cannot do a trip.
9000	80	false	false	Trip cost is not affordable, so we cannot do a trip.
9000	90	false	false	Trip cost is not affordable, so we cannot do a trip despite the exchange rate being high.
7000	70	true	false	Trip is affordable but exchange rate too low for overseas travel

Question 5

Solution:

The following are the test cases for the conditions mentioned in the file interview.html

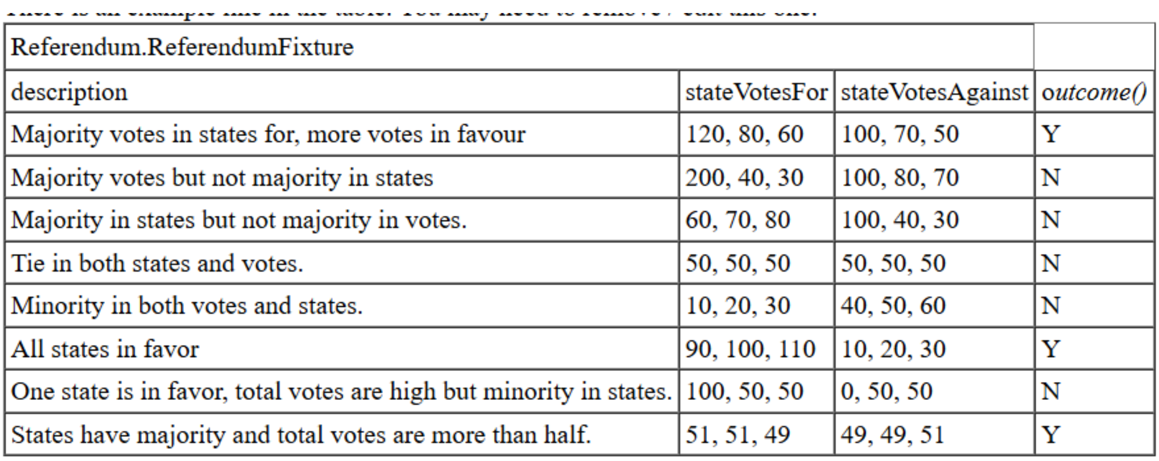


Interview.InterviewFixture			
age	experience	interview()	description
20	5	false	Not old enough
25	0	true	At the lower boundary of acceptable age
42	12	true	Experienced and within acceptable age range
30	5	true	Is in the specified range of age
30	12	true	Is in the specified range of age and has enough experience
42	5	false	Insufficient experience and not within acceptable age range
42	12	true	Experienced and within acceptable age range
45	10	true	At the upper limit of age with sufficient experience
50	12	false	Is a bit too old
50	5	false	Is a bit too old and has insufficient experience

Question 6

Solution:

The following are the test cases for the conditions mentioned in the file referendum.html



There is an example line in the table. You may need to remove / edit this one.

Referendum.ReferendumFixture			
description	stateVotesFor	stateVotesAgainst	*outcome()*
Majority votes in states for, more votes in favour	120, 80, 60	100, 70, 50	Y
Majority votes but not majority in states	200, 40, 30	100, 80, 70	N
Majority in states but not majority in votes.	60, 70, 80	100, 40, 30	N
Tie in both states and votes.	50, 50, 50	50, 50, 50	N
Minority in both votes and states.	10, 20, 30	40, 50, 60	N
All states in favor	90, 100, 110	10, 20, 30	Y
One state is in favor, total votes are high but minority in states.	100, 50, 50	0, 50, 50	N
States have majority and total votes are more than half.	51, 51, 49	49, 49, 51	Y

Question 7

Solution 7a

a) Find the total number of Managers and the sum of their salaries.

```
SELECT COUNT(*) AS manager_count, SUM(salary) AS total_salary

FROM Staff

WHERE oPosition = 'Manager';
```

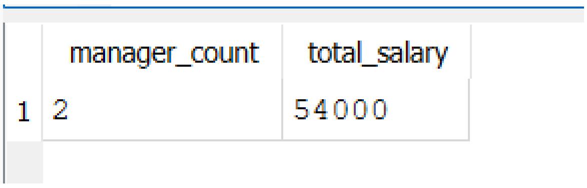
Primary Keys: none

Foreign Keys: none

Degree of Resulting View: 2

Cardinality of the Resulting View: 1

Content Overview based on the attributes required:



	manager_count	total_salary
1	2	54000

Solution 7b

b) Find the minimum, maximum, and average staff salary.

```
SELECT MIN(salary) as minimum_salary, MAX(salary) as maximum_salary, AVG(salary)
as average_salary

FROM Staff;
```

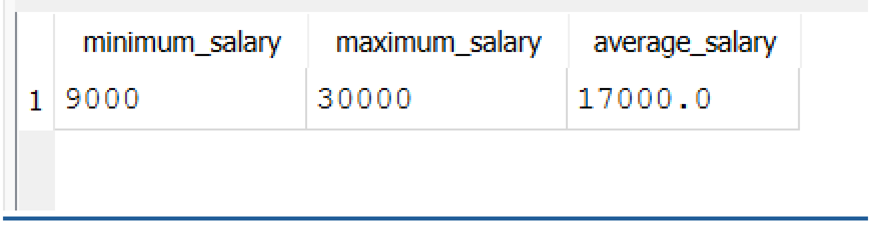
Primary Keys: none

Foreign Keys: none

Degree of Resulting View: 3

Cardinality of the Resulting View: 1

Content Overview based on the attributes required:



	minimum_salary	maximum_salary	average_salary
1	9000	30000	17000.0

Solution 7c

c) For each branch office with more than one member of staff, find the number of staff working in each branch and the sum of their salaries.

```
SELECT Branch.branchNo, COUNT(*) AS staff_count, SUM(Staff.salary) AS total_salary

FROM Staff

JOIN Branch ON Branch.branchNo = Staff.branchNo

GROUP BY Branch.branchNo

HAVING COUNT(*) > 1;
```

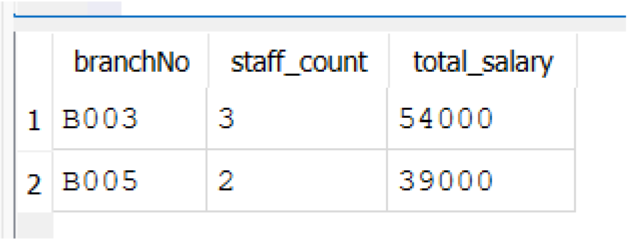
Primary Keys: Branch table- branchNo (branch.branchNo)

Foreign Keys: staff.branchNo references branch.branchNo

Degree of Resulting View: 3

Cardinality of the Resulting View: 2

Content Overview based on the attributes required:



	branchNo	staff_count	total_salary
1	B003	3	54000
2	B005	2	39000

Solution 7d

d) Construct a list of all cities where there is either a branch office or a property.

```
SELECT city

FROM Branch

UNION

SELECT city

FROM PropertyForRent;
```

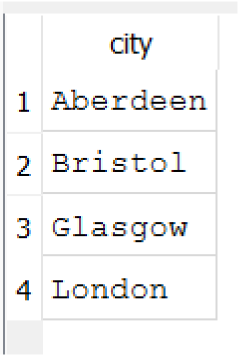
Primary Keys: none

Foreign Keys: none

Degree of Resulting View: 1

Cardinality of the Resulting View: 4

Content Overview based on the attributes required:



	city
1	Aberdeen
2	Bristol
3	Glasgow
4	London

Solution 7e

e) Construct a list of all cities where there is both a branch office or a property.

SELECT city

FROM Branch

INTERSECT

SELECT city

FROM PropertyForRent;

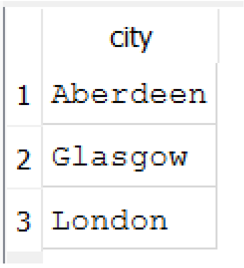
Primary Keys: none

Foreign Keys: none

Degree of Resulting View: 1

Cardinality of the Resulting View: 3

Content Overview based on the attributes required:



	city
1	Aberdeen
2	Glasgow
3	London

Solution 7f

f) Find the total number of Assistants, and the sum and average of their salaries.

```
SELECT COUNT(*) AS number_of_assistants, SUM(salary) AS total_salary, AVG(salary)
AS average_salry

FROM Staff

WHERE oPosition = 'Assistant';
```

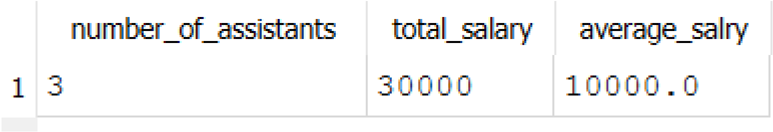
Primary Keys: none

Foreign Keys: none

Degree of Resulting View: 3

Cardinality of the Resulting View: 1

Content Overview based on the attributes required:



	number_of_assistants	total_salary	average_salry
1	3	30000	10000.0

Solution 7g

g) For each branch office, list the staff numbers and names of staff who manage properties alongside the properties they manage.

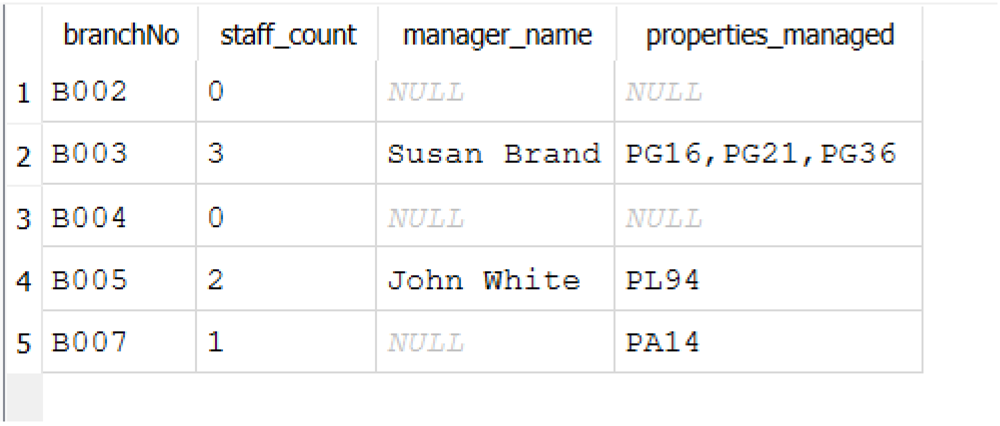
```
SELECT

   Branch.branchNo,

   COALESCE(COUNT(DISTINCT Staff.staffNo), 0) AS staff_count,

   MAX(CASE WHEN Staff.oPosition = 'Manager' THEN Staff.fName || ' ' || Staff.lName
END) AS manager_name,

   GROUP_CONCAT(DISTINCT PropertyForRent.propertyNo) AS properties_managed

FROM Branch

LEFT JOIN Staff ON Branch.branchNo = Staff.branchNo

LEFT JOIN PropertyForRent

   ON Staff.staffNo = PropertyForRent.staffNo

   AND PropertyForRent.branchNo = Branch.branchNo

GROUP BY Branch.branchNo;
```

Primary Keys: branch.branchNo, staff.staffNo

Foreign Keys: staff.branchNo references (branch.branchNo) , propertyForRent.staffNo (references staff.staffNo), propertyForRent.branchNo (references branch.branchNo)

Degree of Resulting View: 4

Cardinality of the Resulting View: 5



	branchNo	staff_count	manager_name	properties_managed
1	B002	0	NULL	NULL
2	B003	3	Susan Brand	PG16,PG21,PG36
3	B004	0	NULL	NULL
4	B005	2	John White	PL94
5	B007	1	NULL	PA14

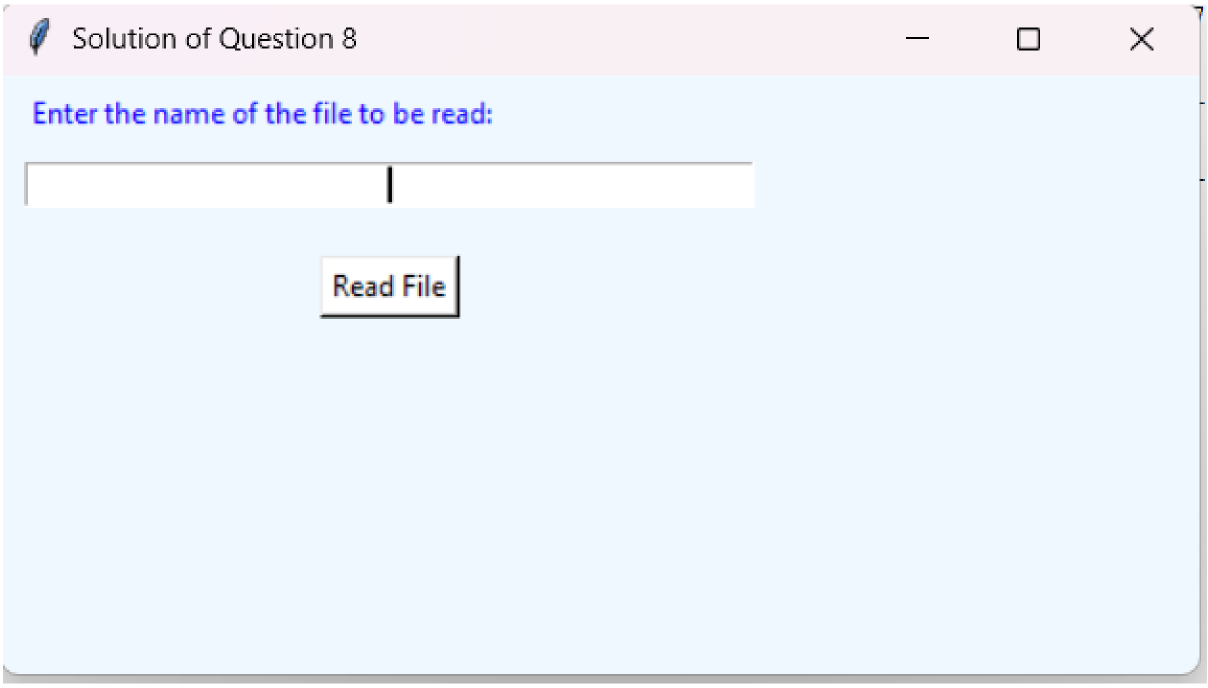
Question 8

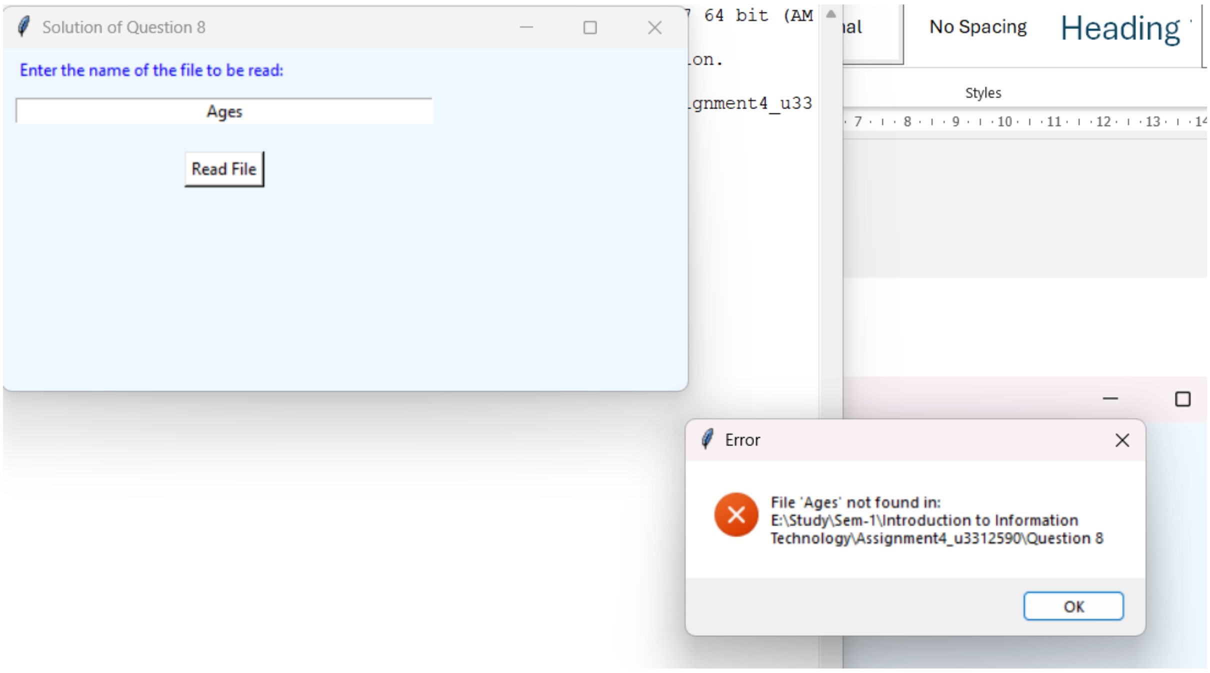
Solution

```
import tkinter as tk

from tkinter import messagebox

import os


def read_file():

    file_name = e1.get().strip()  # Get the filename from the entry box

    base_dir = os.getcwd()        # Current working directory

    full_path = os.path.join(base_dir, file_name)


    try:

        with open(full_path, 'r') as file:

            content = file.read().strip()

            age = int(content)

            messagebox.showinfo("Success", f"Age: {age}")
    except FileNotFoundError:
        messagebox.showerror("Error", f"File '{file_name}' not found in:\n{base_dir}")
    except ValueError:
        messagebox.showerror("Error", f"File '{file_name}' contains an invalid age.")
    except Exception as e:
        messagebox.showerror("Error", f"An unexpected error occurred:\n{e}")


def main():
    global e1  # to make e1 accessible inside read_file()
    root = tk.Tk()
    root.title("Solution of Question 8")
    root.geometry("500x250")
    root.configure(bg="#f0f8ff") #setting the background color to light blue



    label = tk.Label(root, text="Enter the name of the file to be read:", bg="#f0f8ff" , fg= "Blue")
    label.grid(row=1, column=0, padx=10, pady=5, sticky="w")

    e1 = tk.Entry(root, width=50, justify="center") #using the entry box to take input form the user for the file to be read
    e1.grid(row=2, column=0, padx=10, pady=5)

    read_btn = tk.Button(root, text="Read File", command=read_file, bg="white", fg="black")
    read_btn.grid(row=3, column=0, pady=15)

    root.mainloop()

if __name__ == '__main__':
```

```
main()
```

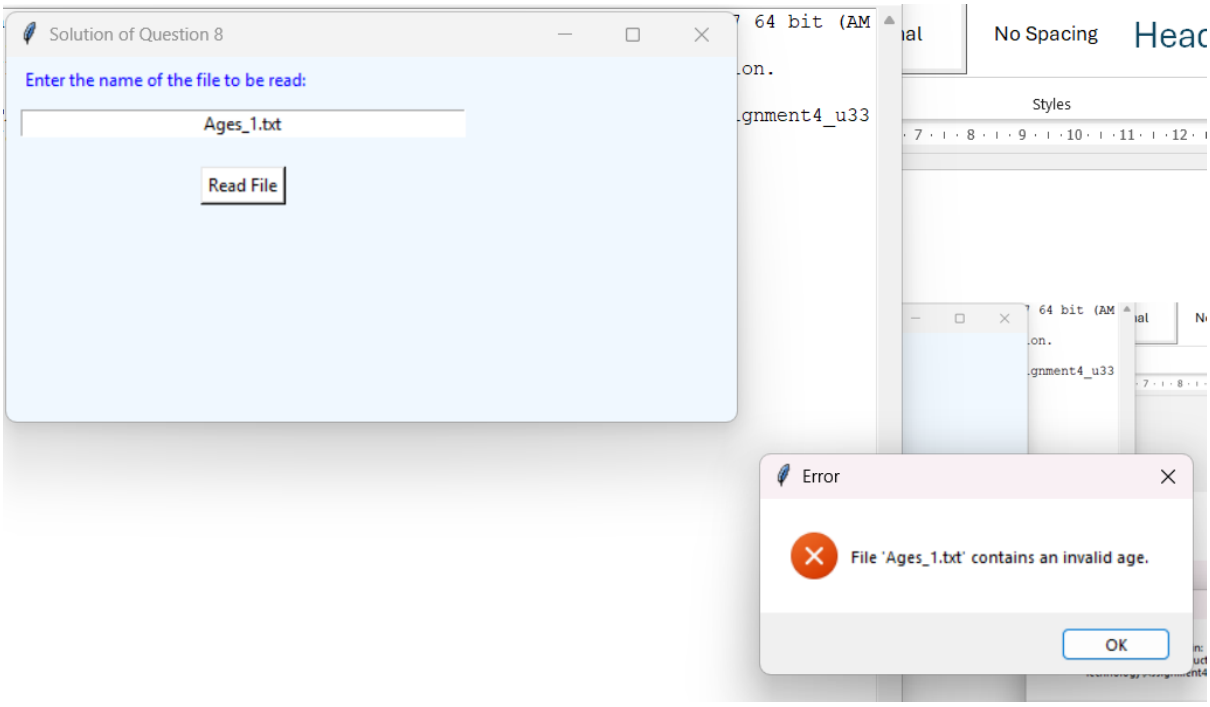
The output is obtained as:





Ages

File 'Ages' not found in:
E:\Study\Sem-1\Introduction to Information Technology\Assignment4_u3312590\Question 8



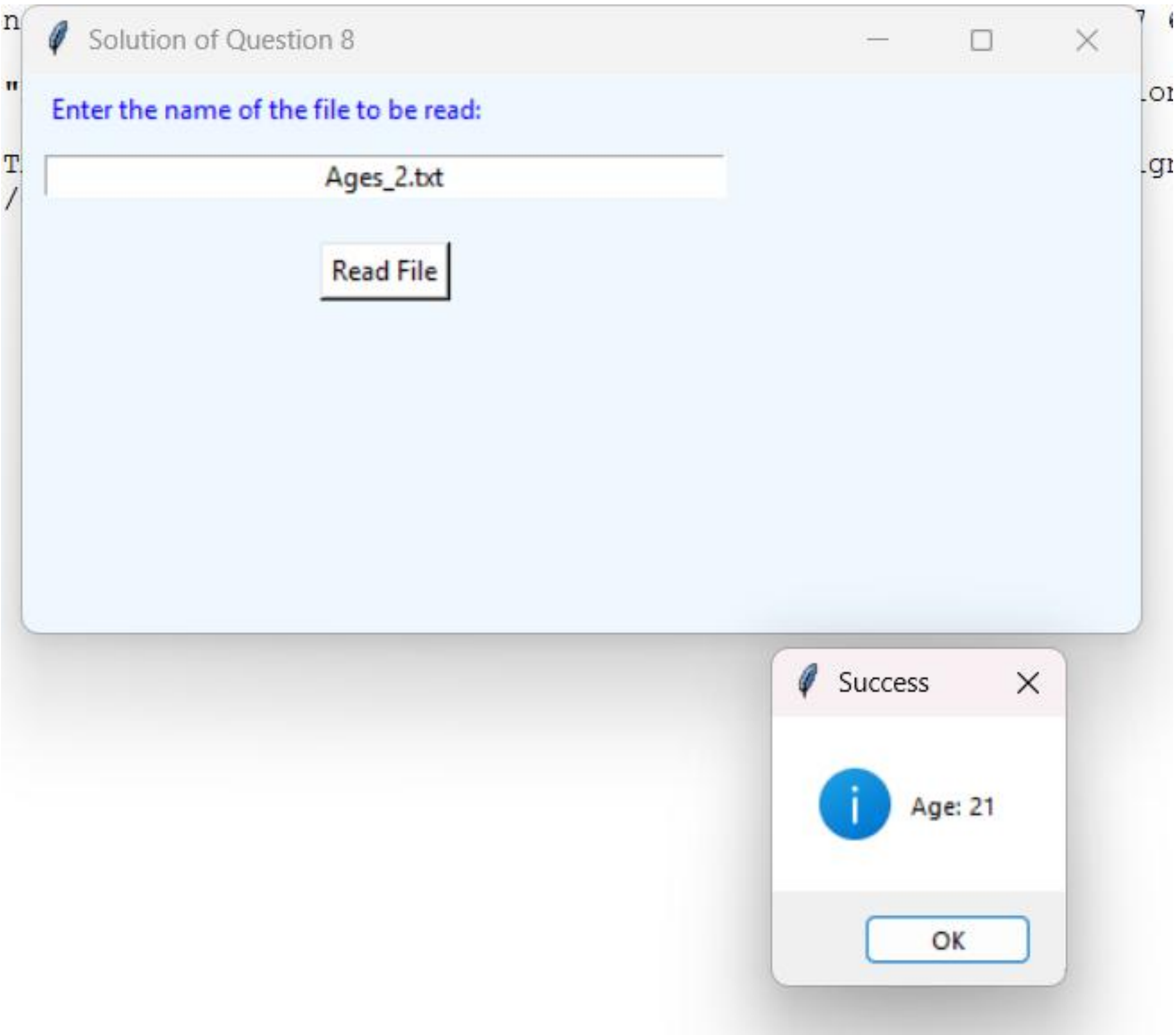
Solution of Question 8

Enter the name of the file to be read:

Ages_1.txt

Error

File 'Ages_1.txt' contains an invalid age.



Solution of Question 8

Enter the name of the file to be read:

Ages_2.txt

Success

Age: 21