

Sql Assignment

Q1 Given two tables created and populated as follows:

```
CREATE TABLE dbo.envelope(id int, user_id int);  
CREATE TABLE dbo.docs(idnum int, pageseq int, doctext  
varchar(100));
```

```
INSERT INTO dbo.envelope VALUES  
(1,1),  
(2,2),  
(3,3);
```

```
INSERT INTO dbo.docs(idnum,pageseq) VALUES  
(1,5),  
(2,6),  
(null,0);
```

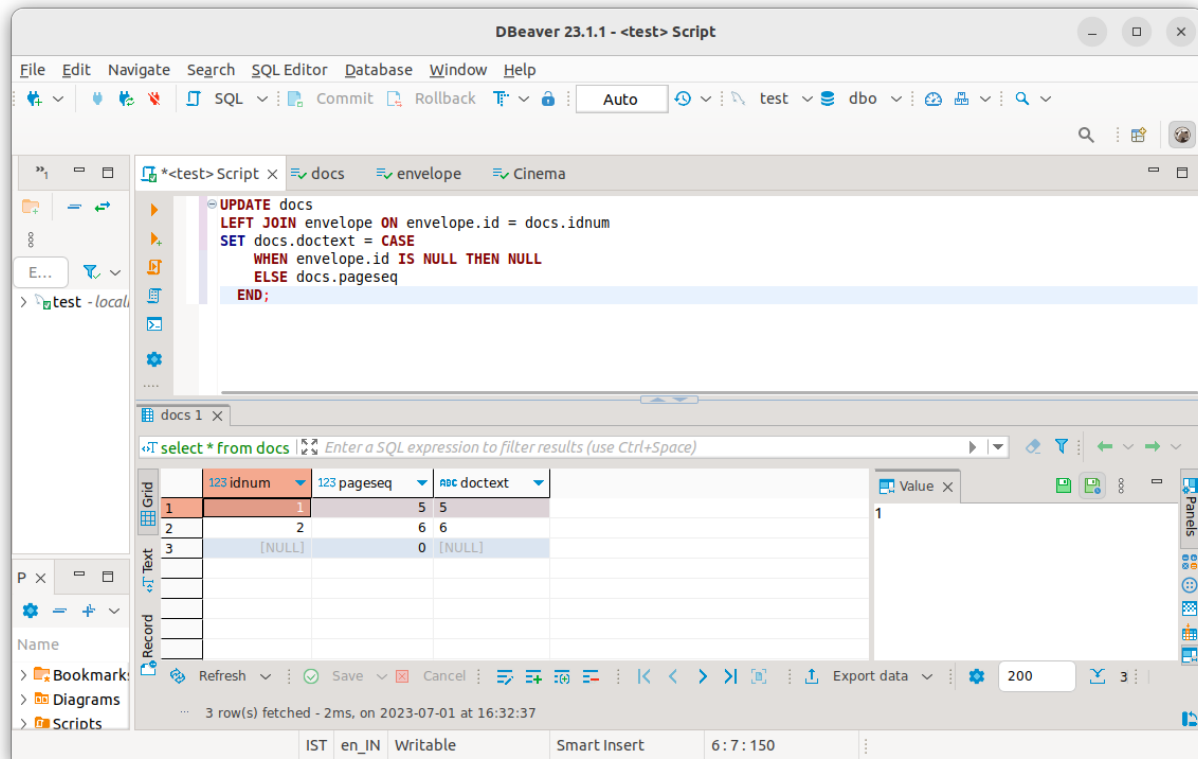
Write a sql query to get below output?

Expected Output:

idnum	pageseq	doctext
1	5	5
2	6	6
NULL	0	NULL

Ans 1: UPDATE docs
LEFT JOIN envelope ON [envelope.id](#) = docs.idnum
SET docs.doctext = CASE
WHEN [envelope.id](#) IS NULL THEN NULL
ELSE docs.pageseq
END;

UPDATE docs SET doctext=
CASE WHEN idnum IS NULL THEN NULL ELSE pageseq
END;



Q2. Table: Users

Column Name	Type
account_number	int
name	varchar

The account is the primary key for this table. Each row of this table contains the account number of each user in the bank. There will be no two users having the same name in the table.

Table: Transactions

Column Name	Type
trans_id	int
account_number	int
amount	int
transacted_on	date

trans_id is the primary key for this table. Each row of this table contains all changes made to all accounts. The amount is positive if the user received money and negative if they transferred money. All accounts start with a balance of 0.

Construct a SQL query to display the names and balances of people who have a balance greater than \$10,000. The balance of an account is equal to the sum of the amounts of all transactions involving that account. You can return the result table in any order.

Example:

Input: Users table:

Account_number	name
12300001	Ram
12300002	Tim
12300003	Shyam

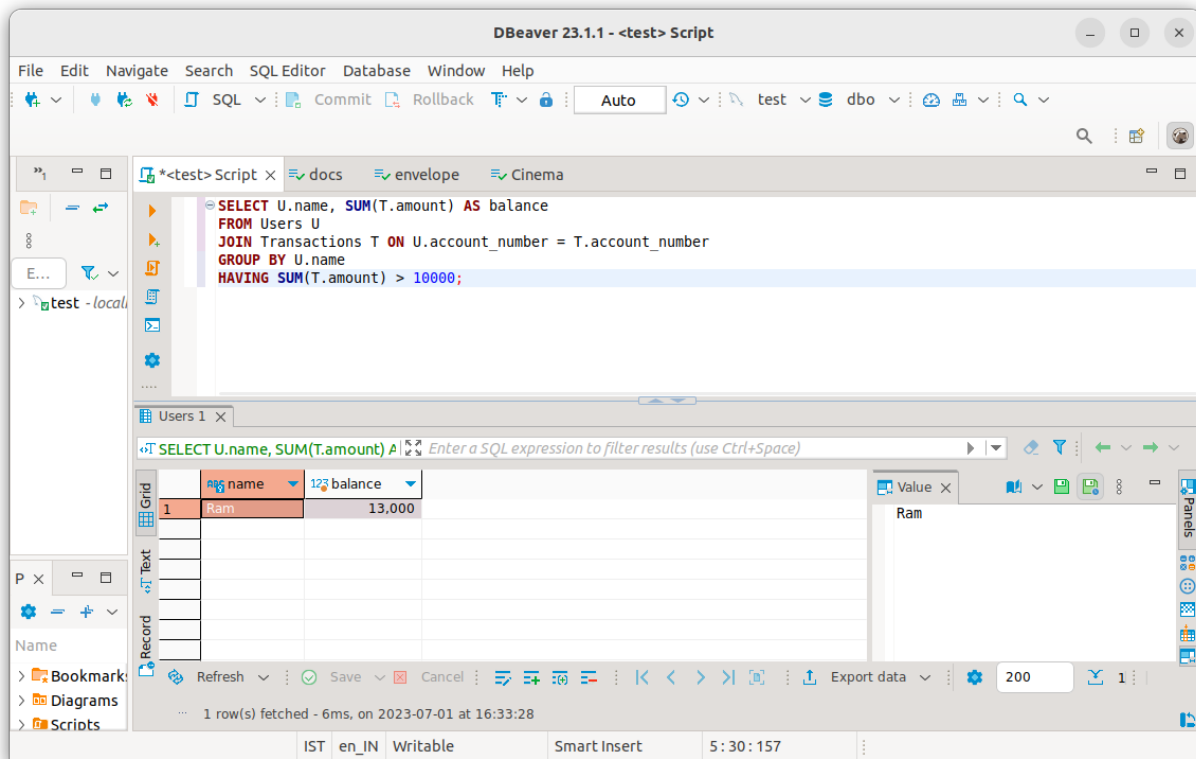
Transactions table:

trans_id	account_number	amount	transacted_on
1	12300001	8000	2022-03-01
2	12300001	8000	2022-03-01
3	12300001	-3000	2022-03-02
4	12300002	4000	2022-03-12
5	12300003	7000	2022-02-07
6	12300003	7000	2022-03-07
7	12300003	-4000	2022-03-11

Output:

name	balance
Ram	13000

Ans 2: SELECT U.name, SUM(T.amount) AS balance
 FROM Users U
 JOIN Transactions T ON U.account_number = T.account_number
 GROUP BY U.name
 HAVING SUM(T.amount) > 10000;



Q3. Consider the following table schema:

Table: Employee

Column Name	Type
Idhe	int
fname	varchar
lname	varchar

department	varchar
projectId	varchar
address	varchar
dateofbirth	varchar
gender	varchar

Table: Salary

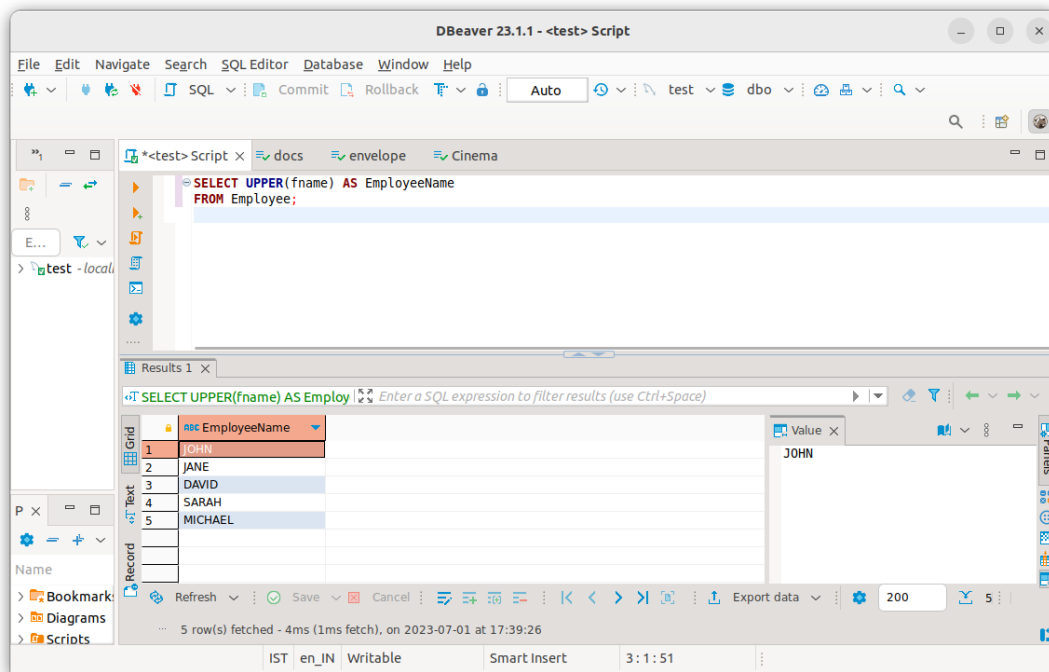
Column Name	Type
id	int
position	varchar
dateofjoining	varchar
salary	varchar

Now answer the following questions:

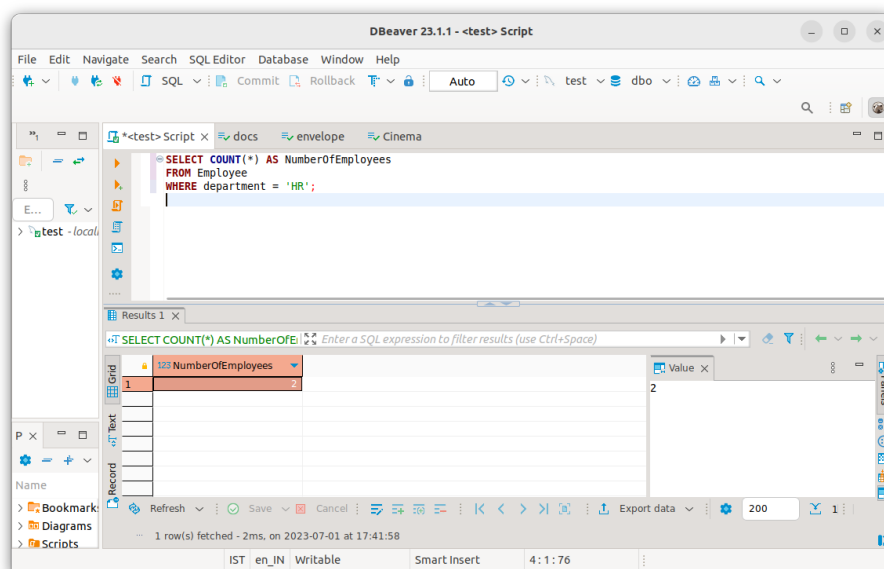
- 1) Construct an SQL query that retrieves the fname in upper case from the Employee table and uses the ALIAS name as the EmployeeName in the result.**
- 2) Construct an SQL query to find out how many people work in the "HR" department**
- 3) Construct an SQL query to retrieve the first four characters of the 'lname' column from the Employee table.**
- 4) Construct a new table with data and structure that are copied from the existing table 'Employee' by writing a query. The name of the new table should be 'SampleTable'.**
- 5) Construct an SQL query to find the names of employees whose first names start with "S".**
- 6) Construct an SQL query to count the number of employees grouped by gender**
- 7) Construct an SQL query to retrieve all employees who are also managers.**
- 8) Construct an SQL query to retrieve the employee count broken down by department and ordered by department count in ascending manner.**
- 9) Construct an SQL query to retrieve duplicate records from the Employee table.**

Ans 3:

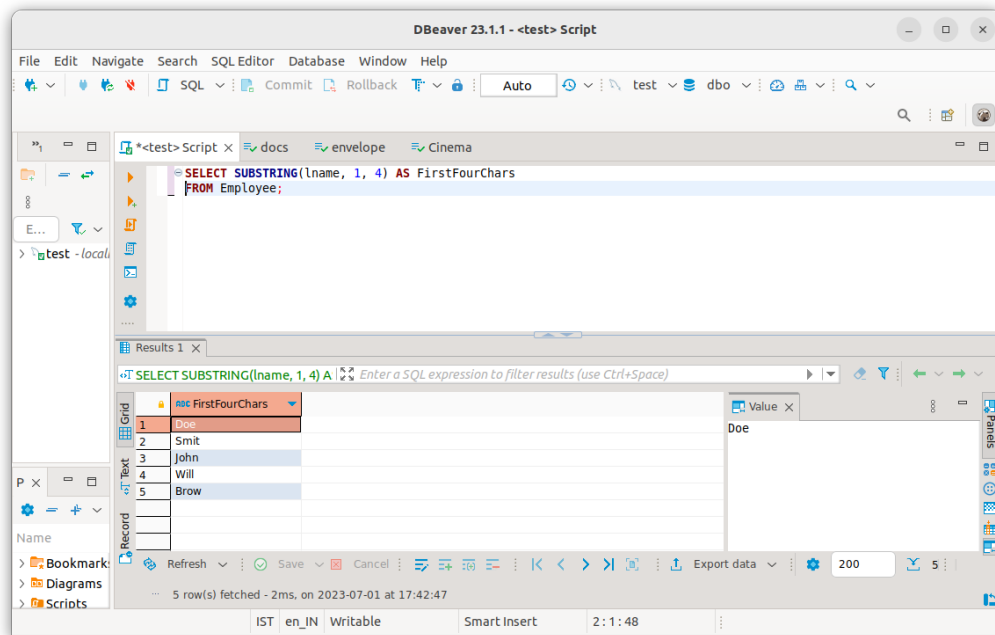
1. SELECT UPPER(fname) AS EmployeeName
FROM Employee;



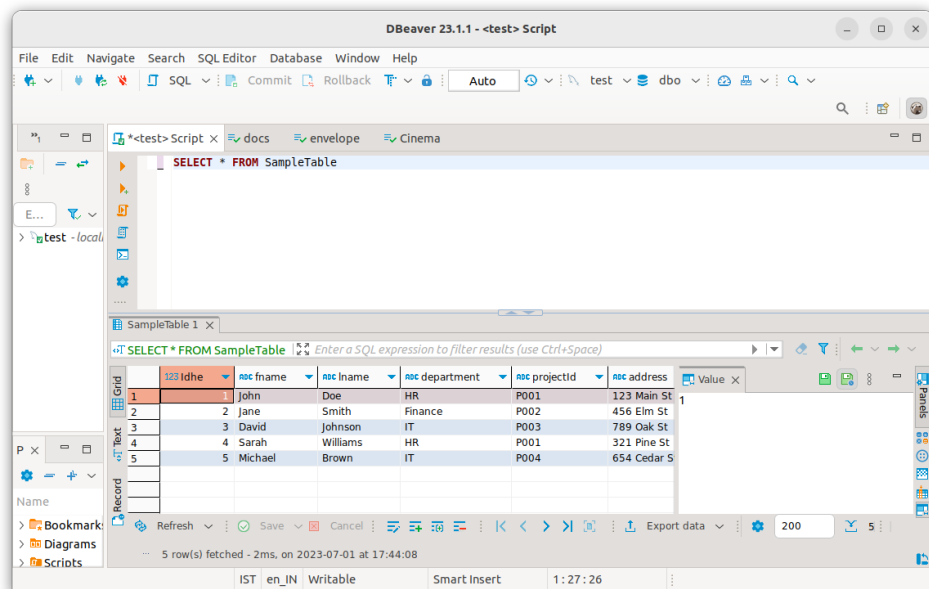
2. SELECT COUNT(*) AS NumberOfEmployees
FROM Employee
WHERE department = 'HR';



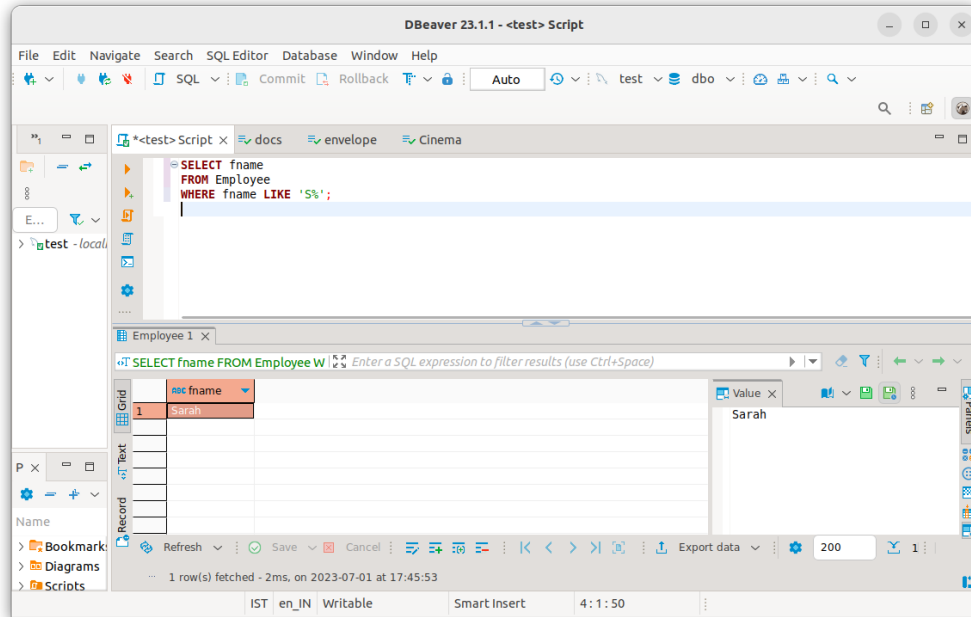
3. SELECT SUBSTRING(lname, 1, 4) AS FirstFourChars
FROM Employee;



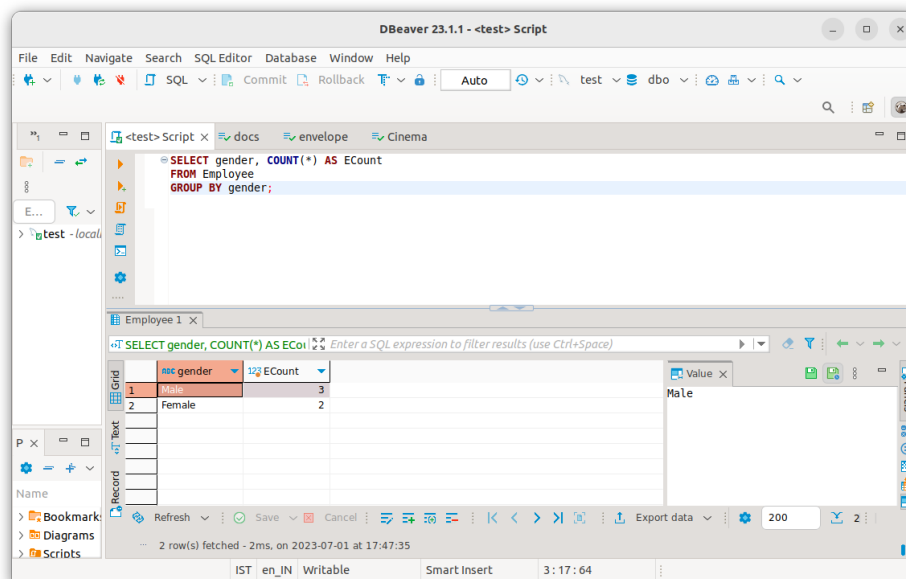
4. CREATE TABLE SampleTable AS
SELECT *
FROM Employee;



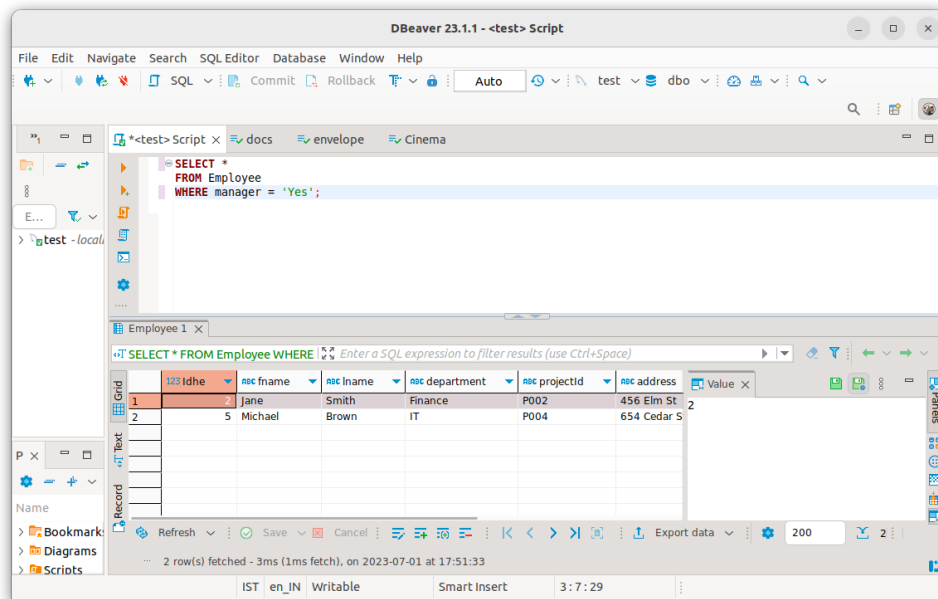
5. SELECT fname
FROM Employee
WHERE fname LIKE 'S%';



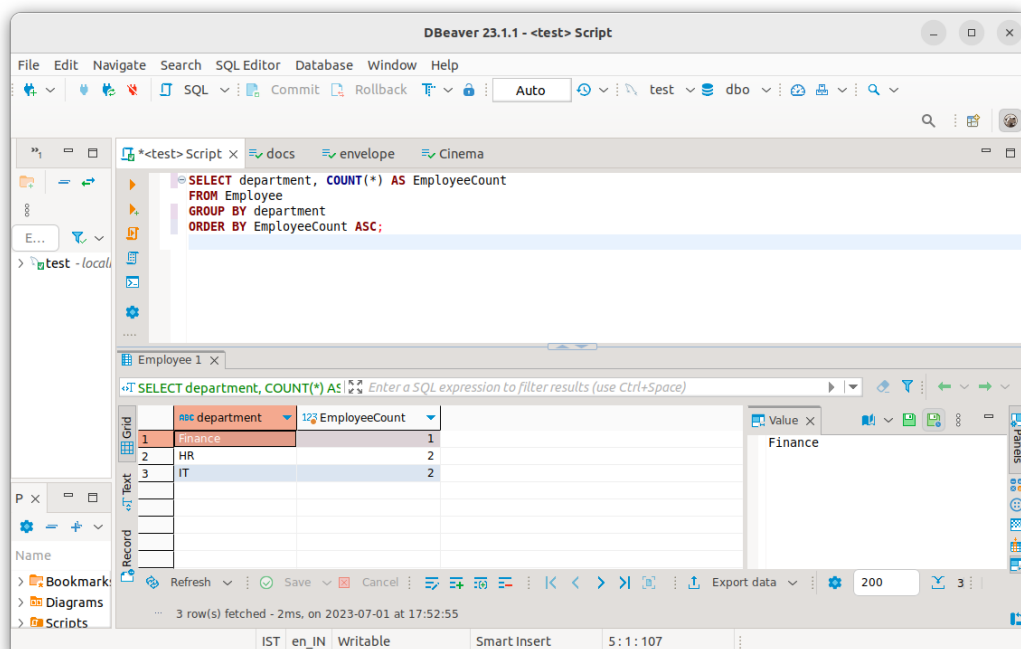
6. SELECT gender, COUNT(*) AS ECount
FROM Employee
GROUP BY gender;



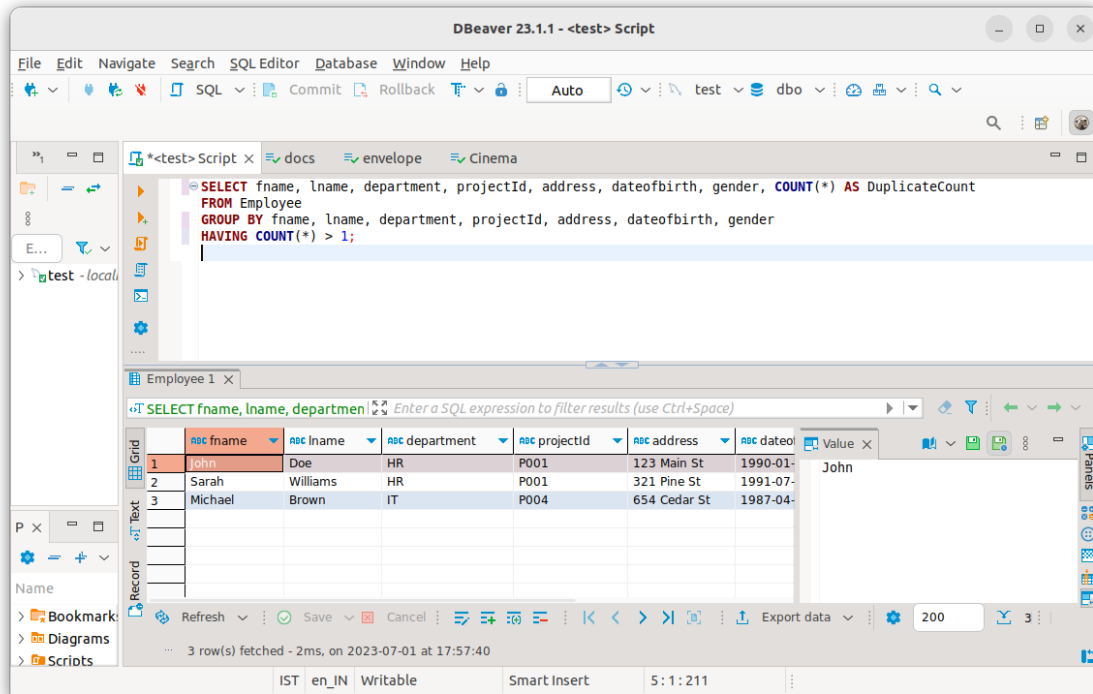
7. SELECT e.fname, s.position
FROM Employee e
LEFT JOIN Salary s
ON e.Idhe = s.id
WHERE s.position = 'Manager'



8. SELECT department, COUNT(*) AS EmployeeCount
FROM Employee
GROUP BY department
ORDER BY EmployeeCount ASC;



9. SELECT fname, lname, department, projectId, address, dateofbirth, gender, COUNT(*) AS DuplicateCount
- FROM Employee
- GROUP BY fname, lname, department, projectId, address, dateofbirth, gender
- HAVING COUNT(*) > 1;



Q4. Given the following schema:

Table: Cinema

Column Name	Type
id	int
movie	varchar
description	varchar
rating	float

The primary key for this table is id. Each row includes information about a movie's name, genre, and rating. rating is a float with two decimal digits in the range [0, 10].

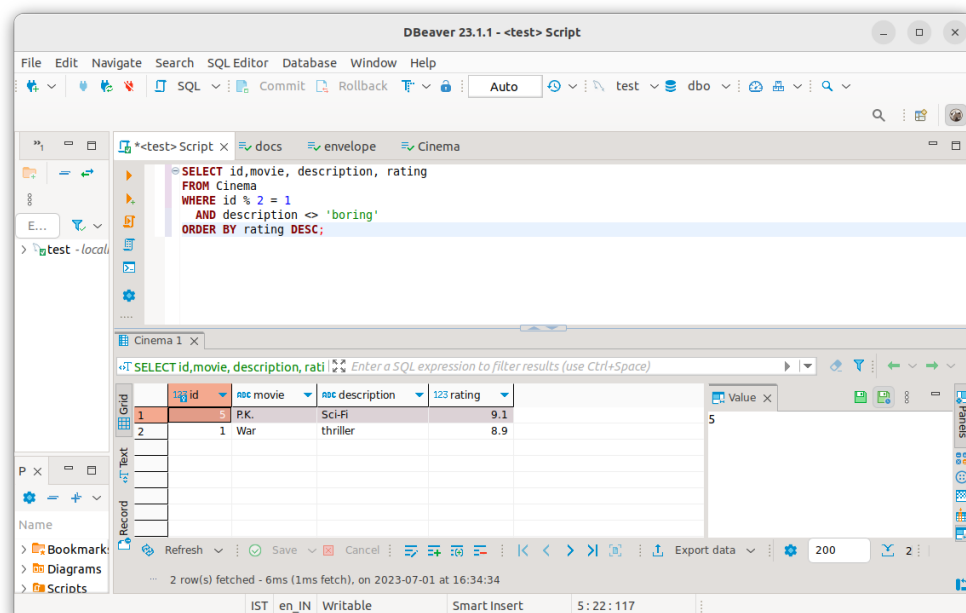
Write an SQL query to report the movies with an odd-numbered ID and a description that is not "boring". Return the result table ordered by rating in descending order.

Example:

Input: Cinema table:

id	movie	description	rating
1	War	thriller	8.9
2	Dhakkad	action	2.1
3	Gippi	boring	1.2
4	Dangal	wrestling	8.6
5	P.K.	Sci-Fi	9.1

Ans 4: SELECT id, movie, description, rating
FROM Cinema
WHERE id % 2 = 1
AND description <> 'boring'
ORDER BY rating DESC;



Python Assignment

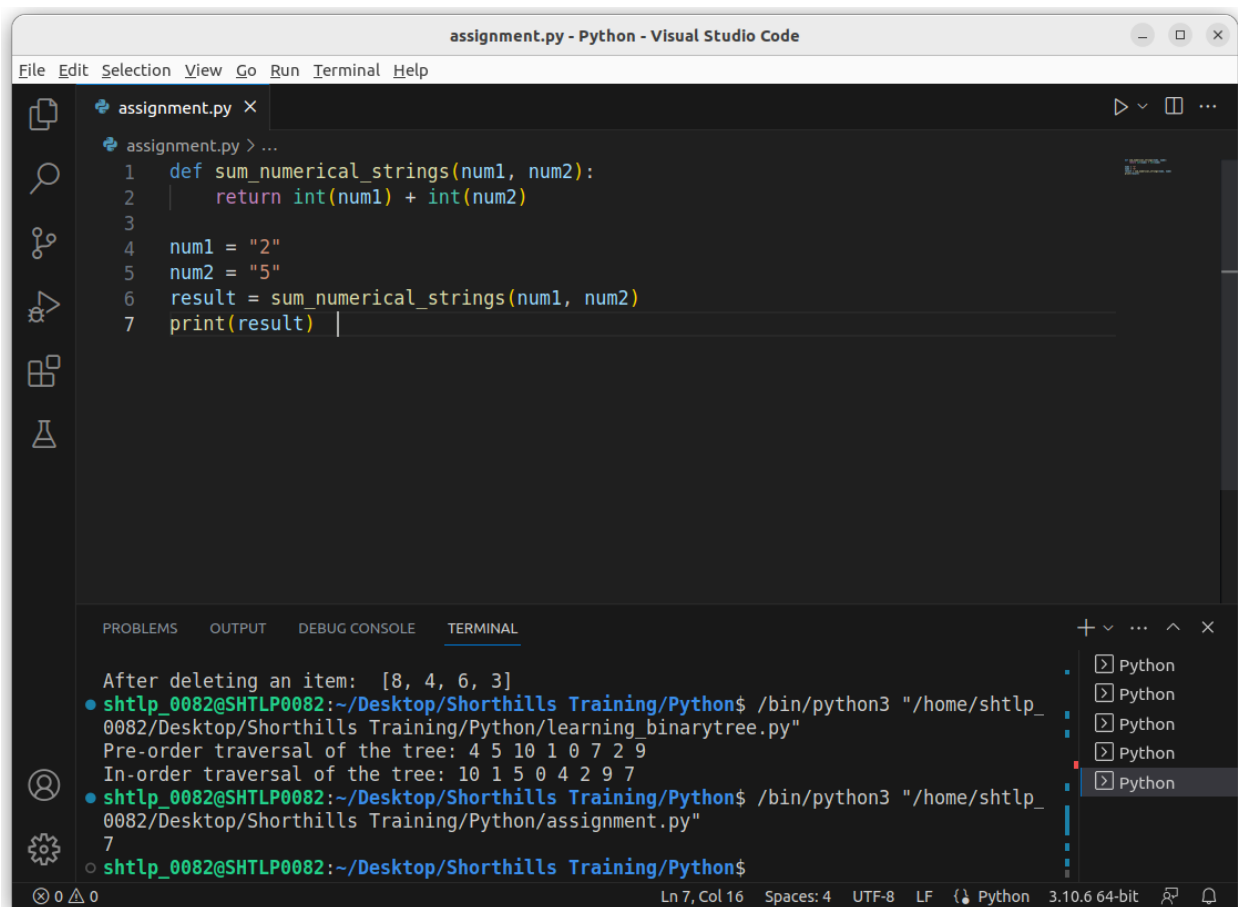
1) Implement a function in python which takes two numerical strings as arguments, returns the sum in integer.

Input: "2" and "5"

Output: 7

Ans 1: `def sum_numerical_strings(num1, num2):
 return int(num1) + int(num2)`

```
num1 = "2"  
num2 = "5"  
result = sum_numerical_strings(num1, num2)  
print(result)
```



The screenshot shows the Visual Studio Code interface with a file named `assignment.py` open. The code in the editor is as follows:

```
1 def sum_numerical_strings(num1, num2):  
2     return int(num1) + int(num2)  
3  
4 num1 = "2"  
5 num2 = "5"  
6 result = sum_numerical_strings(num1, num2)  
7 print(result)
```

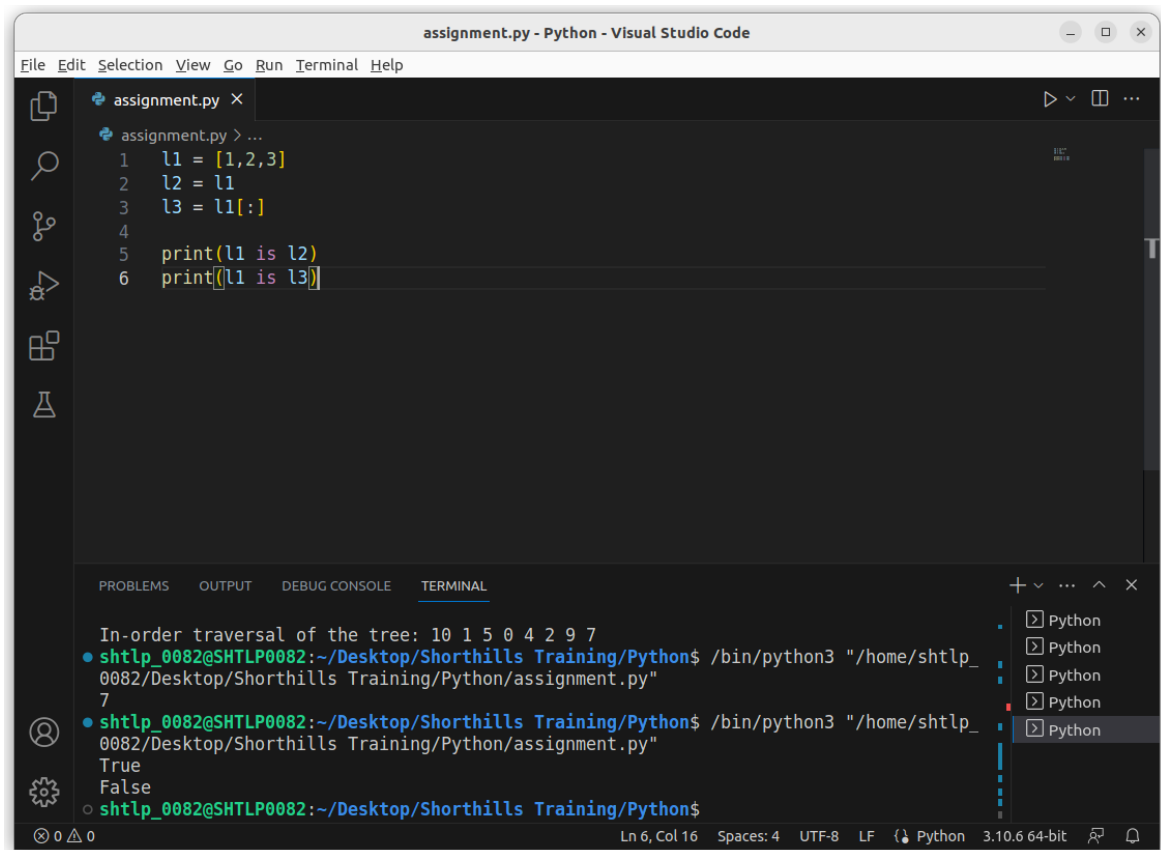
The bottom panel shows the TERMINAL output, which includes the execution of the script and the result:

```
After deleting an item: [8, 4, 6, 3]  
● shtlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/shtlp_0082/Desktop/Shorthills Training/Python/learning_binarytree.py"  
Pre-order traversal of the tree: 4 5 10 1 0 7 2 9  
In-order traversal of the tree: 10 1 5 0 4 2 9 7  
● shtlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/shtlp_0082/Desktop/Shorthills Training/Python/assignment.py"  
7  
○ shtlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$
```

The status bar at the bottom indicates the current file is `Ln 7, Col 16`, using `Spaces: 4`, `UTF-8` encoding, `LF` line endings, and the Python 3.10.6 64-bit interpreter.

2) What is the output of the following code snippet? Also explain why?

```
l1 = [1,2,3]
l2 = l1
l3 = l1[:]  
  
print(l1 is l2)  
print(l1 is l3)
```



The screenshot shows a Visual Studio Code window titled "assignment.py - Python - Visual Studio Code". The editor displays a Python script named "assignment.py" with the following code:

```
1 l1 = [1,2,3]
2 l2 = l1
3 l3 = l1[:]
4
5 print(l1 is l2)
6 print(l1 is l3)
```

The terminal at the bottom shows the output of running the script using Python 3.10.6. The output is:

```
In-order traversal of the tree: 10 1 5 0 4 2 9 7
shtlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/shtlp_0082/Desktop/Shorthills Training/Python/assignment.py"
True
False
shtlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$
```

Ans 2: In this case, l1 and l2 refer to the same list object, so the output is True. l1 and l3 refer to different list objects, even though they have the same elements. Hence, the output is False.

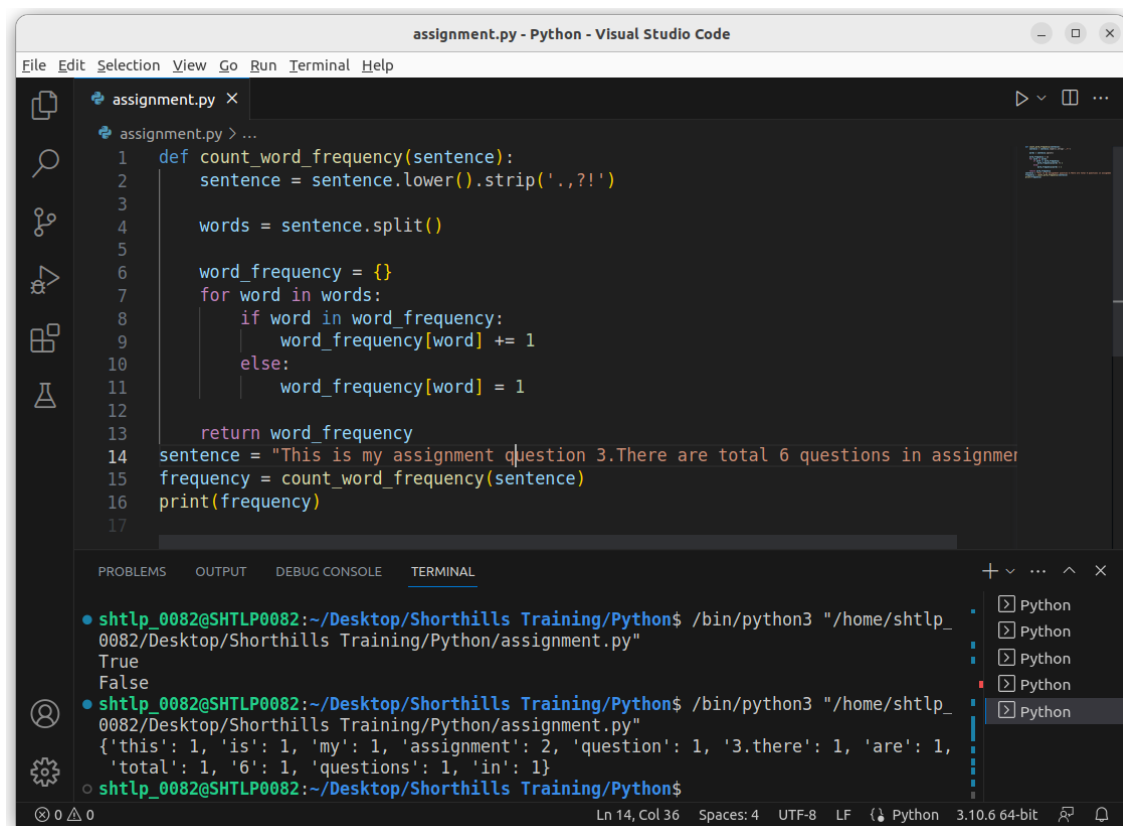
3) You are working on a text processing project and need to count the frequency of each word in a given sentence. Write a Python function called count_word_frequency that takes a sentence as input and returns a dictionary where the keys are the words and the values are the frequencies.

```
Ans 3: def count_word_frequency(sentence):  
sentence = sentence.lower().strip('.,?!')
```

```
words = sentence.split()
```

```
word_frequency = {}  
for word in words:  
if word in word_frequency:  
word_frequency[word] += 1  
else:  
word_frequency[word] = 1
```

```
return word_frequency  
sentence = "This is my assignment question 3. There are total 6 questions in  
assignment"  
frequency = count_word_frequency(sentence)  
print(frequency)
```



The screenshot shows a Visual Studio Code window titled "assignment.py - Python - Visual Studio Code". The editor displays a Python script with the following code:

```
1 def count_word_frequency(sentence):  
2     sentence = sentence.lower().strip('.,?!')  
3  
4     words = sentence.split()  
5  
6     word_frequency = {}  
7     for word in words:  
8         if word in word_frequency:  
9             word_frequency[word] += 1  
10        else:  
11            word_frequency[word] = 1  
12  
13    return word_frequency  
14 sentence = "This is my assignment question 3. There are total 6 questions in assignmer  
15 frequency = count_word_frequency(sentence)  
16 print(frequency)  
17
```

The terminal at the bottom shows the execution of the script using Python 3. The output is:

```
shltlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/shltlp_0082/Desktop/Shorthills Training/Python/assignment.py"  
True  
False  
shltlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/shltlp_0082/Desktop/Shorthills Training/Python/assignment.py"  
{'this': 1, 'is': 1, 'my': 1, 'assignment': 2, 'question': 1, '3.there': 1, 'are': 1, 'total': 1, '6': 1, 'questions': 1, 'in': 1}  
shltlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$
```

4) There is a class called MathUtils that contains various mathematical utility functions. One of the functions you want to

include is a utility function called `is_prime()` that checks whether a given number is prime or not. However, this function does not require any instance-specific data and does not use the `self` parameter. Add the function in the class `is_prime()` that accepts a single parameter `num` and returns `True` if the number is prime, and `False` otherwise.

Ans 4: class MathUtils:

```
@staticmethod
```

```
def is_prime(num):
```

```
    if num <= 1:
```

```
        return False
```

```
    for i in range(2, int(num ** 0.5) + 1):
```

```
        if num % i == 0:
```

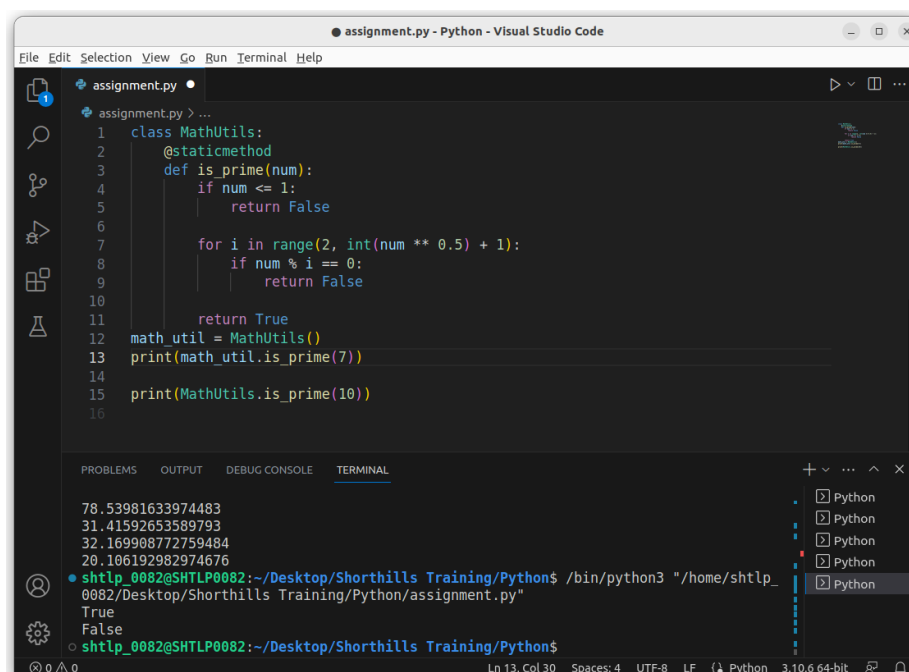
```
            return False
```

```
    return True
```

```
math_util = MathUtils()
```

```
print(math_util.is_prime(7))
```

```
print(MathUtils.is_prime(10))
```



The screenshot shows a Visual Studio Code window with a file named 'assignment.py'. The code in the editor is as follows:

```
1 class MathUtils:
2     @staticmethod
3     def is_prime(num):
4         if num <= 1:
5             return False
6
7         for i in range(2, int(num ** 0.5) + 1):
8             if num % i == 0:
9                 return False
10
11         return True
12 math_util = MathUtils()
13 print(math_util.is_prime(7))
14
15 print(MathUtils.is_prime(10))
16
```

The terminal at the bottom shows the output of the script:

```
78.53981633974483
31.41592653589793
32.169908772759484
20.106192982974676
shltlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/shltlp_0082/Desktop/Shorthills Training/Python/assignment.py"
True
False
shltlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$
```

5) Create a class called Circle that represents a circle. The class should have an attribute for the radius and methods to calculate the area and circumference of the circle. Write the class definition.

Ans 5: import math

class Circle:

```
def __init__(self, radius):  
    self.radius = radius
```

```
def calculate_area(self):  
    return math.pi * (self.radius ** 2)
```

```
def calculate_circumference(self):  
    return 2 * math.pi * self.radius
```

circle1 = Circle(5)

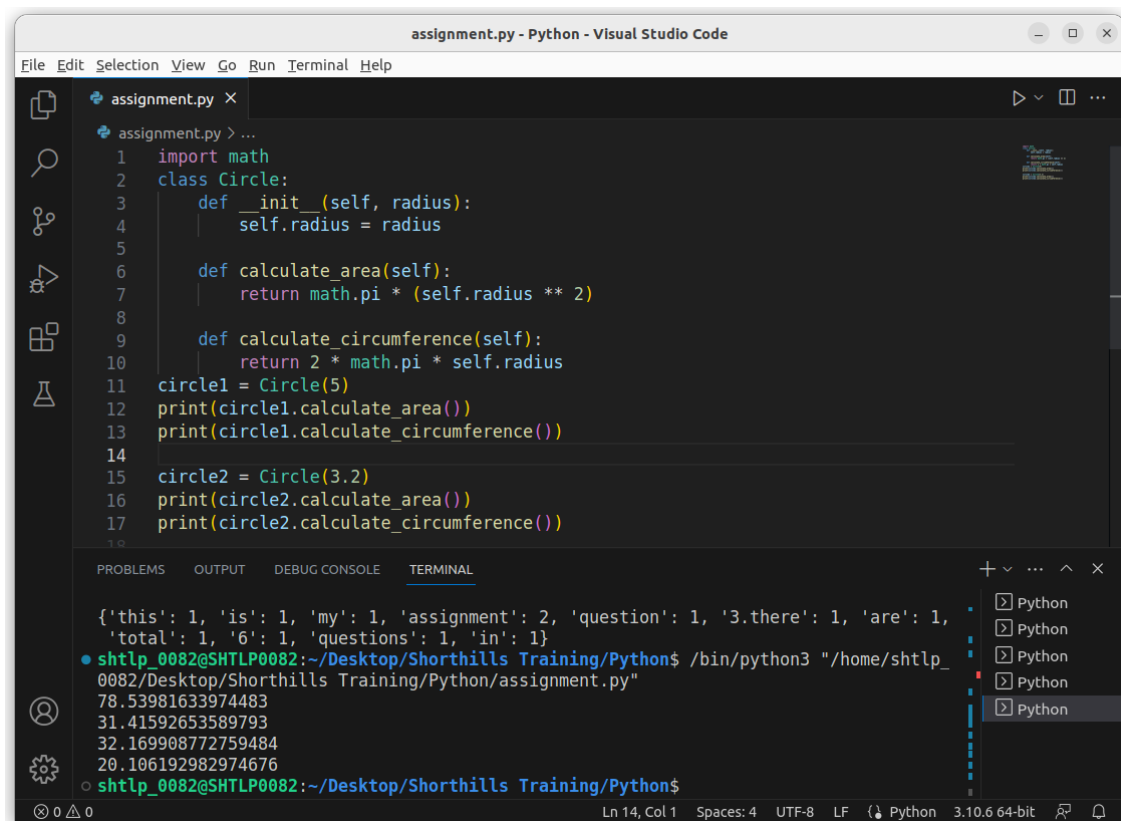
print(circle1.calculate_area())

print(circle1.calculate_circumference())

circle2 = Circle(3.2)

print(circle2.calculate_area())

print(circle2.calculate_circumference())



```
assignment.py - Python - Visual Studio Code  
File Edit Selection View Go Run Terminal Help  
assignment.py X  
assignment.py > ...  
1 import math  
2 class Circle:  
3     def __init__(self, radius):  
4         self.radius = radius  
5  
6     def calculate_area(self):  
7         return math.pi * (self.radius ** 2)  
8  
9     def calculate_circumference(self):  
10        return 2 * math.pi * self.radius  
11  
12 circle1 = Circle(5)  
13 print(circle1.calculate_area())  
14 print(circle1.calculate_circumference())  
15  
16 circle2 = Circle(3.2)  
17 print(circle2.calculate_area())  
18 print(circle2.calculate_circumference())  
19  
20  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
{'this': 1, 'is': 1, 'my': 1, 'assignment': 2, 'question': 1, '3.there': 1, 'are': 1,  
'total': 1, '6': 1, 'questions': 1, 'in': 1}  
• shtlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/shtlp_  
0082/Desktop/Shorthills Training/Python/assignment.py"  
78.53981633974483  
31.41592653589793  
32.169908772759484  
20.106192982974676  
• shtlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$  
Ln 14, Col 1 Spaces: 4 UTF-8 LF Python 3.10.6 64-bit
```


6) Given an integer list nums, return true if any value appears at least twice in the list, and return false if every element is distinct.

Input: nums = [1, 1, 4, 2, 3, 3, 3]

Output: true

Input: nums = [3, 4, 9, 7]

Output: false

Ans 6: def contains_duplicate(nums):

```
    num_set = set()
```

```
    for num in nums:
```

```
        if num in num_set:
```

```
            return True
```

```
        num_set.add(num)
```

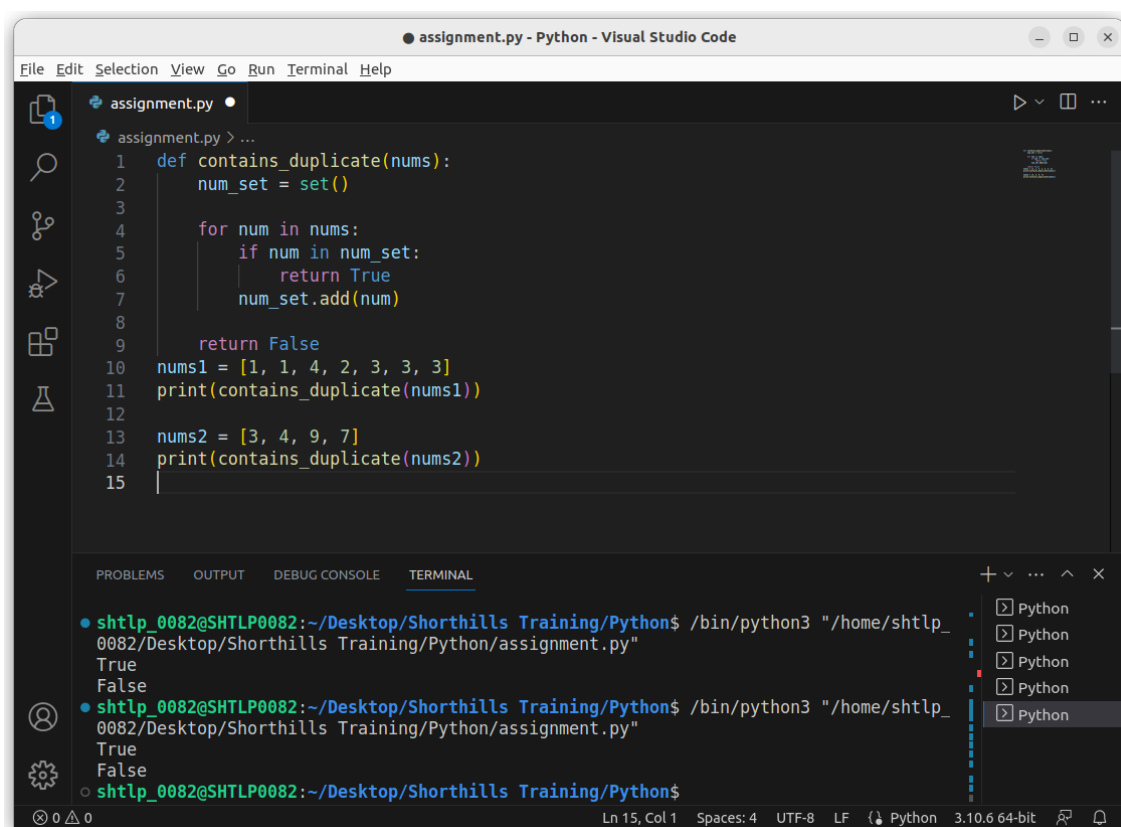
```
    return False
```

```
nums1 = [1, 1, 4, 2, 3, 3, 3]
```

```
print(contains_duplicate(nums1))
```

```
nums2 = [3, 4, 9, 7]
```

```
print(contains_duplicate(nums2))
```



The screenshot shows a Visual Studio Code window titled 'assignment.py - Python - Visual Studio Code'. The editor displays the following Python code:

```
1 def contains_duplicate(nums):
2     num_set = set()
3
4     for num in nums:
5         if num in num_set:
6             return True
7         num_set.add(num)
8
9     return False
10
11 nums1 = [1, 1, 4, 2, 3, 3, 3]
12 print(contains_duplicate(nums1))
13
14 nums2 = [3, 4, 9, 7]
15 print(contains_duplicate(nums2))
```

The bottom panel shows the 'TERMINAL' output, which displays the execution of the script twice. The first execution shows 'True' for the first list and 'False' for the second. The second execution shows 'True' for the first list and 'False' for the second.

```
shltlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/shltlp_0082/Desktop/Shorthills Training/Python/assignment.py"
True
False
shltlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/shltlp_0082/Desktop/Shorthills Training/Python/assignment.py"
True
False
shltlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$
```

7) Given two strings s and t, return true if t is an anagram of s, and false otherwise.

Input: s = "abc", t = "cba"

Output: true

Input: s = "hat", t = "cat"

Output: false

Ans 7: def is_anagram(s, t):

 return sorted(s) == sorted(t)

s1 = "abc"

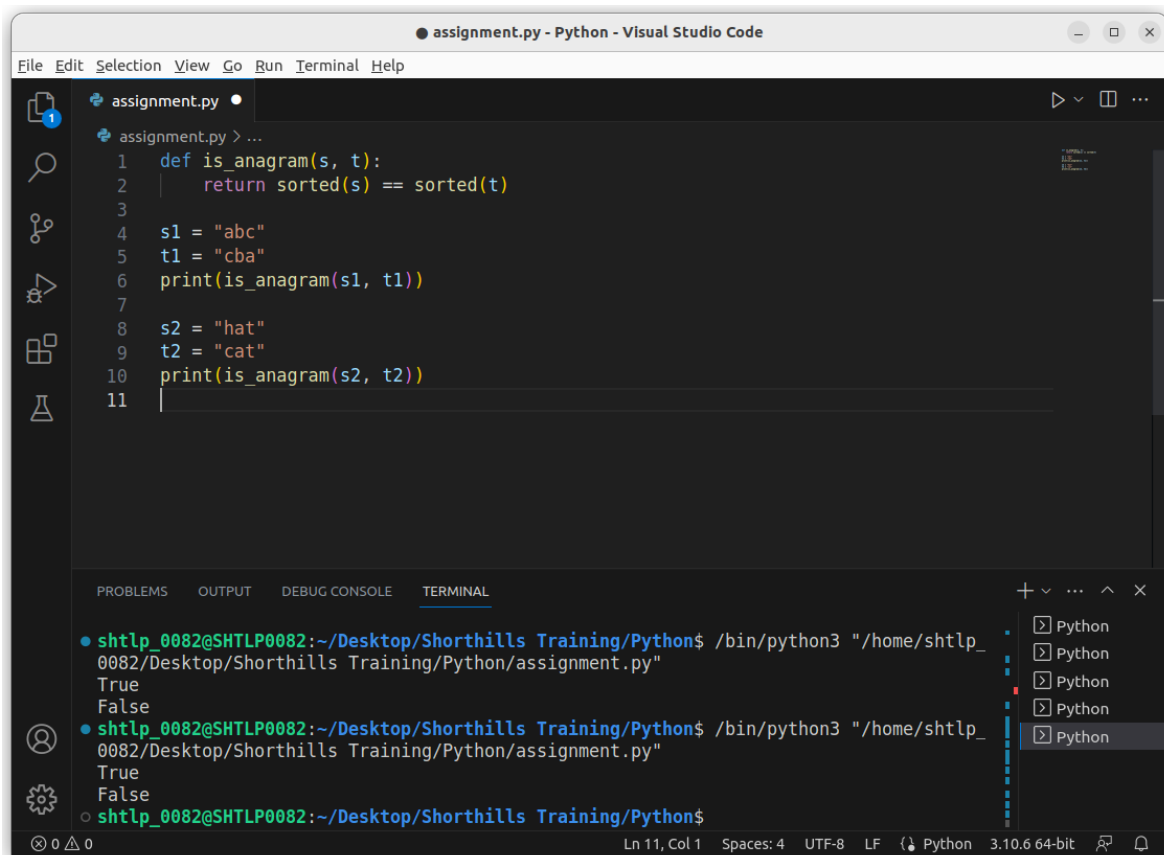
t1 = "cba"

print(is_anagram(s1, t1))

s2 = "hat"

t2 = "cat"

print(is_anagram(s2, t2))



The screenshot shows a Visual Studio Code window titled "assignment.py - Python - Visual Studio Code". The editor displays a Python script named "assignment.py" with the following code:

```
1 def is_anagram(s, t):
2     return sorted(s) == sorted(t)
3
4 s1 = "abc"
5 t1 = "cba"
6 print(is_anagram(s1, t1))
7
8 s2 = "hat"
9 t2 = "cat"
10 print(is_anagram(s2, t2))
11
```

The bottom panel shows the "TERMINAL" output, which displays the execution of the script using Python 3. The output is as follows:

```
sh_tlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/sh_tlp_0082/Desktop/Shorthills Training/Python/assignment.py"
True
False
sh_tlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/sh_tlp_0082/Desktop/Shorthills Training/Python/assignment.py"
True
False
sh_tlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$
```

The status bar at the bottom indicates the file is at "Ln 11, Col 1", uses "Spaces: 4", "UTF-8" encoding, "LF" line endings, and is a "Python 3.10.6 64-bit" file.

8) Write a Python one-liner to swap the values of two variables, a and b, without using a temporary variable.

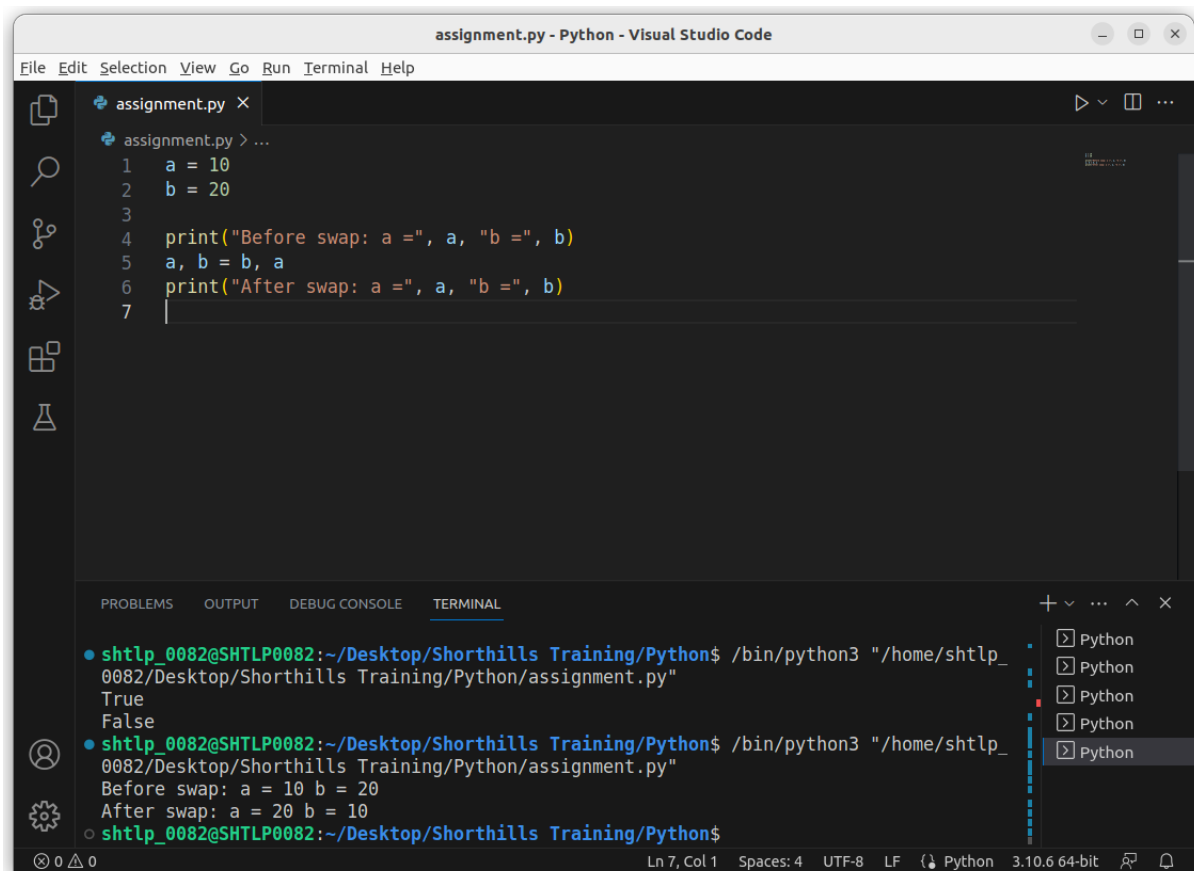
Ans 8: a = 10

b = 20

```
print("Before swap: a =", a, "b =", b)
```

```
a, b = b, a
```

```
print("After swap: a =", a, "b =", b)
```



The screenshot shows the Visual Studio Code interface with a file named `assignment.py` open. The code in the editor is as follows:

```
1 a = 10
2 b = 20
3
4 print("Before swap: a =", a, "b =", b)
5 a, b = b, a
6 print("After swap: a =", a, "b =", b)
7
```

The terminal at the bottom shows the execution of the script using `python3`. The output is:

```
sh_tlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/sh_tlp_0082/Desktop/Shorthills Training/Python/assignment.py"
True
False
sh_tlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$ /bin/python3 "/home/sh_tlp_0082/Desktop/Shorthills Training/Python/assignment.py"
Before swap: a = 10 b = 20
After swap: a = 20 b = 10
sh_tlp_0082@SHTLP0082:~/Desktop/Shorthills Training/Python$
```

The status bar at the bottom indicates the current line and column (Ln 7, Col 1), the number of spaces (4), the encoding (UTF-8), the line ending (LF), the Python version (3.10.6 64-bit), and the file type (Python).