
Malaviya National Institute of Technology, Jaipur
Submitted To: Dr Dinesh Gopalani
Submitted by: Yukti Khurana, 2017UCP1234
Subject: Advanced Topics in Databases (CST440)
LAB ASSIGNMENT- 4

Ques

Write a program which shall read a file containing a concurrent schedule involving n transactions and find and display Read-Write ratio for each transaction. Also find what data items are accessed commonly by two or more transactions and give the list of such data items along with the transaction ids accessing the data items.

APPROACH

I made this program in **Python Language**. Following approach was used to code this:

- Firstly, I created an input file named **“Assign4_input.txt”**, which contains the concurrent schedule to be used to evaluate the requirements given in the problem statement.
- Then I created 4 helper functions to read these concurrent schedules properly –
 1. **transaction_id(op)** – returns the transaction id of the operation of schedule
 2. **op_type(op)** – tells whether the operation being considered is read or write in nature
 3. **data_item(op)** – returns the data item that the transaction is working on in the current operation
 4. **display_schedule(t,d,schedule)** – this function takes the number of transactions, data items and concurrent schedule, respectively as input to display in a format easily understood by us
- Then I created the **“read_schedule(t,d,schedule)”** function to convert each schedule in the format of a list of transactions.

Example: for Schedule-1 of input file,

This function returns:

```
s = [['vacant', 'R1(a)', 'vacant', 'R1(a)', 'vacant', 'vacant', 'W1(a)', 'vacant', 'vacant', 'vacant', 'vacant'], ['vacant', 'vacant', 'R2(a)', 'vacant', 'W2(a)', 'vacant', 'vacant', 'W2(b)', 'vacant', 'vacant', 'vacant'], ['W3(a)', 'vacant', 'vacant', 'vacant', 'vacant', 'W3(a)', 'vacant', 'vacant', 'R3(a)', 'vacant', 'vacant'], ['vacant', 'vacant', 'vacant', 'vacant', 'vacant', 'vacant', 'vacant', 'vacant', 'vacant', 'vacant', 'R4(b)', 'W4(c)']]
```

- Two functions were required to be created for solving the above problem –
 1. **read_write_ratio(s)**
 2. **common_data(schedule)**

Note: s is the output of function – **read_schedule(t,d,schedule)**

- **Finding the read-write ratio of each transaction of the schedule**

ALGORITHM

start

Function read_write_ratio(s):

 Declare a dictionary in python– rwr

 For transaction in schedule:

 Initialize reads and writes as 0

```

    Declare "trans" as an empty string
    For operation in transaction:
        If operation is not vacant:
            id = get transaction id using helper function
            type = get operation type using helper function
            if type is "read":
                increment the reads counter
            if type is "write":
                increment the writes counter

        if writes are not 0:
            store the ratio-reads/writes in dictionary with transaction id as key
        otherwise:
            store None in the dictionary as ratio will be infinity
    return dictionary-rwr
rwr = read_write_ratio(s)
iterate the dictionary-rwr:
    print key:item pairs if value is not None
    else print "infinity"
end

```

- **Finding the data items are accessed commonly by two or more transactions and printing the corresponding transactions:**

ALGORITHM

start

Function common_data(schedule):

```

    Declare dictionary-data_dict
    For operation in schedule:
        d = get the data item in this operation using helper function
        ti = get transaction id of this operation using helper function
        if d is not in dictionary-data_dict:
            add the d as key and ti as the value
        else:
            add ti in the list of values of key=d
        sort the list containing transactions for symmetry
    return dictionary-data_dict
items = common_data(schedule)
iterate items dictionary:
    print key:value pairs in appropriate format

end

```

INPUT

"Assign4_input.txt" file contains the concurrent schedule inputs

The input format is as follows –

<no of transactions>, <no of data items>, <schedule operations of transactions>

Example: R1(a) represents "Read of data item 'a' by transaction-1"

```

Assign4.py × assign4_input.txt ×
1 4,3,W3(a) R1(a) R2(a) R1(a) W2(a) W3(a) W1(a) W2(b) R3(a) R4(b) W4(c)
2 3,2,W3(a) R1(a) R2(a) R1(b) W2(a) W3(b) W1(a) W2(b) R3(a)
3 3,4,R1(a) W2(d) W3(a) W2(a) R2(a) R3(b) W1(c)
4 3,3,R1(a) W1(a) R2(b) W2(b) R2(a) R3(c) W3(c) R1(c) W1(c) W2(a) R3(a) W3(a)
5 2,2,R1(X) R1(Y) R2(X) R2(Y) W2(Y) W1(X)
6 3,1,R1(Q) W2(Q) R3(Q) W1(Q)
7 2,1,R1(A) W1(A) R2(A) W2(A) R1(B) W1(B) R2(B) W2(B)
8 4,2,W1(P) R2(P) R2(Q) W3(Q) R4(Q) W2(Q)
9 |

```

CODE

```

"""
ATD Assignment-4
Concurrent Schedules

Author: Yukti Khurana
Date: 13-04-2021

"""

# helper functions to read the input file

# to return the transaction id of transactions in each schedule
def transaction_id(op):
    index = op.find("(")
    # transaction id can be find on index after the starting bracket
    t = int(op[1:index])
    return t

# to return the type of operation
def op_type(op):
    d = {'R':"read", 'W':"write"}
    t = str(op[0]).upper()
    # returns "read" if R is in op, otherwise "write" for W
    return d[t]

# to return the data item used by the operation
def data_item(op):
    index1 = op.find("(")
    index2 = op.find(")")
    # to extract the data item inside the bracket
    return str(op[index1+1:index2])

# to display the schedule in readable format
def display_schedule(t,d,schedule):
    print("\n##### SCHEDULE #####\n")
    for i in range(t):
        print("T"+str(i+1), end="\t\t\t\t")
        print()
        for i in schedule:
            t_no = transaction_id(i)
            d = data_item(i)
            s = "\t\t\t\t"*(t_no-1)
            op = op_type(i)+"("+d+") "
            print(s+op)

```

```

print("\n\n#####\n")

# to convert the schedule into the list of all transactions
def read_schedule(t,d,schedule):
    # make 2-d array of lists for different transactions
    s = [[] for _ in range(t)]
    # filling array for each transaction
    for i in range(len(schedule)):
        t = schedule[i]
        tid = transaction_id(t)
        for j in range(len(s)):
            if (tid == j+1):
                s[j].append(t)
            else:
                s[j].append("vacant")

    return s

# function to calculate the read-write ratio of each transaction of a
# schedule
def read_write_ratio(s):
    # dictionary to store ratio for each transaction in the schedule
    rwr = {}
    for ti in s:
        r, w = 0, 0
        trans = ""
        for ops in ti:
            if ops != 'vacant':
                trans = transaction_id(ops)
                optype = op_type(ops)
                if (optype == 'read'):
                    r+=1
                elif (optype == 'write'):
                    w+=1
        tr = "T"+str(trans)
        if (w!=0):
            rwr[tr] = (1.0*r/w)
        else:
            # if num of writes are zero, the ratio will become infinity
            rwr[tr] = (None)
    return rwr

# function to find the transactions using each data item of the schedule
def common_data(s):
    data_dict = {}
    for op in s:
        # getting the transaction id and data item name for each operation
        # in a transaction
        d = data_item(op)
        ti = transaction_id(op)
        # appending the Transaction id using the current data item in the
        # declared dictionary
        if d not in data_dict.keys():
            data_dict[d] = [ti]
        else:
            if ti not in data_dict[d]:
                data_dict[d].append(ti)
            data_dict[d].sort()
    return(data_dict)

```

```

# Reading the input
f = open("assign4_input.txt")
schedules = f.readlines()

# to count the total number of schedules
counter = 0

# iterating through all the schedules
for i in schedules:
    counter += 1
    print("\nSchedule - ", counter)
    # reading various elements of the input file
    t = str(i).strip().split(",")
    no_transactions = int(t[0])
    no_data = int(t[1])
    schedule = t[2].split(" ")
    slen = len(schedule)

    # to display the schedule
    display_schedule(no_transactions, no_data, schedule)

    # to get the list of transactions of schedule
    s = read_schedule(no_transactions, no_data, schedule)

    # dictionary containing read writes of each transaction
    rwr = read_write_ratio(s)

    # displaying the data items which have been accessed by 2 or more
    transactions
    # and the names of the transactions accessing them
    print("\nThe read-write ratio of all transactions of this schedule are:
")
    for i in rwr:
        if (rwr[i] is None):
            print(i, " : ", "infinity")
        else:
            print(i, " : ", rwr[i])

    data_item_info = common_data(schedule)
    print("\nData items that are accessed commonly by two or more
transactions are as follows : ")
    for i in data_item_info.keys():
        if len(data_item_info[i]) >= 2:
            print("Data item-", i, ":")
            for ti in data_item_info[i]:
                tstr = "T"+str(ti)
                print(tstr, end=",")
            print("\n")
    print("-----")
    print("-----")

# Yukti Khurana 2017UCP1234

```

OUTPUT

```
C:\Users\yukti\Desktop\atd-3\venv\Scripts\python.exe C:/Users/yukti/Desktop/atd-3/Assign4.py

Schedule - 1

##### SCHEDULE #####

T1          T2          T3          T4
read(a)
read(a)      read(a)
read(a)      write(a)
write(a)     write(a)
              read(a)
              read(b)
              write(c)

#####

The read-write ratio of all transactions of this schedule are:
T1 : 2.0
T2 : 0.5
T3 : 0.5
T4 : 1.0

The read-write ratio of all transactions of this schedule are:
T1 : 2.0
T2 : 0.5
T3 : 0.5
T4 : 1.0

Data items that are accessed commonly by two or more transactions are as follows :
Data item- a :
T1,T2,T3,

Data item- b :
T2,T4,
```

The read-write ratio of all transactions is being printed correctly.

E.g., for T2, no of reads = 1, no of writes=2, so reads/writes ratio = 0.5

As we can see in the above schedule- data items a and b are accessed by 2 or more transactions while c is not, therefore a and b and their corresponding transactions are printed

Schedule - 2

SCHEDULE

T1	T2	T3
		write(a)
read(a)		
	read(a)	
read(b)		
	write(a)	
		write(b)
write(a)		
	write(b)	
		read(a)

#####

The read-write ratio of all transactions of this schedule are:

T1 : 2.0

T2 : 0.5

T3 : 0.5

Data items that are accessed commonly by two or more transactions are as follows :

Data item- a :

T1,T2,T3,

Data item- b :

T1,T2,T3,

Schedule - 3

SCHEDULE

T1	T2	T3
read(a)		
	write(d)	
		write(a)
	write(a)	
	read(a)	
		read(b)
write(c)		

#####

The read-write ratio of all transactions of this schedule are:

T1 : 1.0

T2 : 0.5

T3 : 1.0

Data items that are accessed commonly by two or more transactions are as follows :

Data item- a :

T1,T2,T3,

Schedule - 4

Schedule - 4

SCHEDULE

T1	T2	T3
read(a)		
write(a)		
	read(b)	
	write(b)	
	read(a)	
		read(c)
		write(c)
read(c)		
write(c)		
	write(a)	
		read(a)
		write(a)

#####

The read-write ratio of all transactions of this schedule are:

T1 : 1.0
T2 : 1.0
T3 : 1.0

Data items that are accessed commonly by two or more transactions are as follows :

Data item- a :

T1,T2,T3,

Data item- c :

T1,T3,

Schedule - 5

SCHEDULE

T1	T2
read(X)	
read(Y)	
	read(X)
	read(Y)
	write(Y)
write(X)	

#####

The read-write ratio of all transactions of this schedule are:

T1 : 2.0
T2 : 2.0

Data items that are accessed commonly by two or more transactions are as follows :

Data item- X :

T1,T2,

Data item- Y :

T1,T2,

Schedule - 6

SCHEDULE

T1	T2	T3
read(Q)		
	write(Q)	
		read(Q)
write(Q)		

#####

The read-write ratio of all transactions of this schedule are:

T1 : 1.0
T2 : 0.0
T3 : infinity

Data items that are accessed commonly by two or more transactions are as follows :

Data item- Q :

T1,T2,T3,

Schedule - 7

SCHEDULE

T1	T2
read(A)	
write(A)	
	read(A)
	write(A)
read(B)	
write(B)	
	read(B)
	write(B)

#####

The read-write ratio of all transactions of this schedule are:

T1 : 1.0
T2 : 1.0

Data items that are accessed commonly by two or more transactions are as follows :

Data item- A :

T1,T2,

Data item- B :

T1,T2,

Schedule - 8

SCHEDULE

T1	T2	T3	T4
write(P)			
	read(P)		
	read(Q)		
		write(Q)	
			read(Q)
	write(Q)		

#####

The read-write ratio of all transactions of this schedule are: |

T1 : 0.0
T2 : 2.0
T3 : 0.0
T4 : infinity

Data items that are accessed commonly by two or more transactions are as follows :

Data item- P :

T1,T2,

Data item- Q :

T2,T3,T4,
