▼ Comments

Proposed grade: 100 / 100

Summary of tests

/ 1 test run/ 1 test p

4

Dashboard Assignments Completed Tasks Reports API Reference Study Material Feedback Calendar U - 4 V1 9 > / ?

```
program.sql D 🖂
1 /*
2 Query 1: Daily Net Balance Change Per Customer
3 Problem:
4 Write a SQL query to display the daily net transaction amount
 5 (deposits - withdrawals) for each customer.
 6 Only include customers who had at least one transaction on that day.
8 Expected Output Columns:
9 customer_id
10 transaction date (date only)
11 net_amount
12
13
14 -----
15 Database Name: fs
16 -----
17
18 TABLE: transactions
19
20 Field
                 Null
                                Default Extra
          Type
                         Key
          int
                 NO
                         PRI
                                NULL
21 id
22 customer id
                 int
                         YES
                                       NULL
23 amount decimal(10,2) YES
                                       NULL
                         varchar(10)
                                       YES
                                                      NULL
24 transaction_type
                                        YES
                                                      NULL
25 transaction date
                         timestamp
26 -----
27
28
29
30
31 Sample Output:
32 -----
33 customer_id transaction_date
                                        net_amount
34 101
          2023-10-01
                         5000.00
35 101
          2023-10-02
                         -2000.00
          2023-10-03
36 101
                         2500.00
          2023-10-04
                         7000.00
37 101
38 101
          2023-10-05
                         4000.00
          2023-10-06
                         6000.00
39 101
          2023-10-07
                         3000.00
40 101
41 101
          2023-10-08
                         -1500.00
          2023-10-09
                         8000.00
42 101
43 102
          2023-10-03
                         1000.00
44
45
46
47 */
48 use fs;
49 select customer id, date(transaction date) as transaction date,SUM(
50
51
           WHEN transaction_type='deposit' then amount
           when transaction_type='withdraw' then -amount
52
53
           else 0
54
           end
55
           ) as net amount
56 from transactions group by customer id, date(transaction date) order by customer id, date(transaction date)
```

Let 1, Col 1 SQL



U B 7 80 9 >- / ?

```
program.sql D 🖾
1 /*
 2 Query 2: Customers with At Least 3 Deposits Over Rs. 5000 in a Month
 3 Problem:
4 Write a SQL query to find customers who made at least 3 high-value deposits
 5 (amount > 5000) within the same calendar month.
 7 Your result should include:
 9 The customer_id
11 The year and month (e.g., '2023-10')
13 The total number of such high-value deposits in that month
14
15 Sort of result:
17 Each row in the output should represent a customer-month combination where
18 the condition was met.
19
20
21 -----
22 Database Name: fs
23 -----
24
25 TABLE: transactions
26
          Type
int
                                Default Extra
27 Field
                 Null
                         Key
28 id
                 NO
                         PRI
                               NULL
29 customer_id
                        YES
                 int
                                       NULL
30 amount decimal(10,2) YES
                                       NULL
                                                     NULL
31 transaction_type
                         varchar(10)
                                       YES
32 transaction date
                         timestamp
                                       YES
                                                     NULL
33 -----
34
35
36
37
38 Sample Output:
39 -----
40 customer_id month_label high_value_deposit_count
41 101 2023-10 3
42
43
44
45 */
```



```
U B 7 80 9 >- / ?
 program.sql D 🖾
 1 /*
 2 Query 3:
 4 Write a SQL query to compute the running total of deposit amounts for each
  5 customer in chronological order of transactions.
 7 Expected Output Columns:
 9 customer_id
 10 transaction date
 11 amount
 12 running_total
 13
 14 Hint: Use a window function like SUM(...) OVER (PARTITION BY ... ORDER BY ...).
 15
 16
 17 -----
 18 Database Name: fs
 19 -----
 20
 21 TABLE: transactions
 22
 23 Field
          Type Null
                                Default Extra
                         Key
          int
                         PRI
                 NO
 24 id
                                NULL
 25 customer_id int
                         YES
                                        NULL
 26 amount decimal(10,2) YES
                                        NULL
 27 transaction_type
                                                      NULL
                         varchar(10)
                                       YES
 28 transaction date
                                        YES
                                                      NULL
                         timestamp
 29 -----
 30
31
 32
 33
 34 Sample Output:
 35 -----
 36 customer_id transaction_date amount running_total
 37 101 2023-10-01 10:00:00 5000.00 5000.00
 38 101 2023-10-03 09:00:00 2500.00 7500.00
 39 101 2023-10-04 12:00:00 7000.00 14500.00
 40 101 2023-10-05 14:30:00 4000.00 18500.00
 41 101 2023-10-06 16:00:00 6000.00 24500.00
 42 101 2023-10-07 13:15:00 3000.00 27500.00
 43 101 2023-10-09 09:30:00 8000.00 35500.00
 44 102 2023-10-03 15:00:00 1000.00 1000.00
 45
 46
47
 48
 49 */
 50
51
```

U B 7 80 9 > 7 ?

```
program.sql D 🖂
1 /*
 2 Query 4:
4 Write a SQL query to find the students who have scored more than the average
 5 marks of each subject in that respective subject.
 7 Your output should include:
 9 student_id
10
11 student_name
12
13 This query should only return students who have outperformed the average mark
14 in every subject they appeared in.
15
16
17 -----
18 Database Name: fs
19 -----
20
21 TABLE: student_marks
22
23 Field Type Null Key Default Extra
24 student_id int YES NULL
25 student_name varchar(50) YES NULL
26 subject varchar(30) YES NULL
27 marks int YES NULL
28
29 -----
30
31
32
33
34 Sample Output:
35 -----
36 student_id student_name
          Bob
37 2
38 3
          Charlie
39 5
          Eva
40
41
42
43
44
45 */
46
47
```

```
U B 7 80 9 >- / ?
 program.sql D 🖾
 1 /*
 2 Query 5:
 4 Write a SQL query to compute the rank of each student per subject based on their marks.
 6 Your output should include:
 8 student id
 9 student_name
 10 subject
 11 marks
 12
13 subject_rank (1 for highest marks in that subject, 2 for second highest, etc.)
15 Use a window function like RANK() or DENSE_RANK() to assign ranks
 16
 17
 18
 19 -----
 20 Database Name: fs
 21 -----
 22
 23 TABLE: student_marks
 24
 25 Field Type Null Key Default Extra
 26 student_id int YES NULL
27 student_name varchar(50) YES NULL
28 subject varchar(30) YES NULL
 29 marks int YES NULL
 30
 31 -----
 32
33
 34
 35
 36 Sample Output:
 37 -----
 38 student_id student_name subject marks subject_rank
 39 3 Charlie Math 95 1
 40 5 Eva Math 92 2
 41 2 Bob Math 90 3
 42 1 Alice Math 80 4
 43 4 David Math 70 5
 44 5 Eva Science 89 1
 45 3 Charlie Science 88 2
 46 2 Bob Science 85 3
 47 1 Alice Science 75 4
 48 4 David Science 68 5
 49
 50
51
52
53
54
55 */
56
 57
```

U = 7 20 9 > 2 ?

```
program.sql D 🖾
 1 /*
 2 Query 6:
4 Write a SQL query to find students who have scored at least 85 marks in every
 5 subject they have appeared for.
 7 Your output should include:
 9 student_id
10 student name
12 These are consistent top performers - you should eliminate any student who has
13 even one subject below 85.
14
15
16
17
18
19 -----
20 Database Name: fs
21 -----
22
23 TABLE: student_marks
24
25 Field Type Null Key Default Extra
26 student_id int YES NULL
27 student_name varchar(50) YES NULL
28 subject varchar(30) YES NULL
29 marks int YES NULL
30
31 -----
32
33
34
35
36 Sample Output:
37 -----
38 student_id student_name
39 2 Bob
40 3 Charlie
41 5 Eva
42
43
44
45
46
47
48 */
49 use fs;
50 select * from student_marks
51 -- select student_id, student_name from student_marks
52 -- where
53
54
```

```
Dashboard Assignments Completed Tasks Reports API Reference Study Material Feedback Calendar
```

```
U B 7 80 9 >- / ?
 program.sql D 🖾
 1 /*
 2 Query 7:
 4 Write a SQL query to find the top-performing student (by total marks) in each 5 combination of class and section.
 7 Expected Output Columns:
 9 class
 10 section
 11 name
 12 total marks
14 This should return 1 student per class-section pair who has the highest total
15 across all subjects.
 16
17
 18
 19
20
 21
 22
 23 -----
 24 Database Name: fs
 25 -----
 26
 27 TABLE: students
 28
29 Field Type Null Key
                                Default Extra
 30 roll_no int YES
                                 NULL
31 name varchar(50)
                         YES
                                        NULL
 32 class varchar(10)
                         YES
                                        NULL
 33 section char(1) YES
                                 NULL
 34 subject varchar(30) YES
                                        NULL
 35 marks int YES
                                 NULL
 36
 37 -----
 38
 39
40
 41
 42 Sample Output:
 43 -----
 44 class section name total_marks
 45 10 A Eva 181
 46 10 B Charlie 183
 47
 48
49
50
51
52
53
54 */
55
56
```

```
U B 7 80 9 > / ?
 program.sql D 🖾
 1 /*
 2 Query 8:
 5 Write a SQL query to find the average marks scored in each subject for every
 8 The result should include:
10 The class
 11 The subject
 12 The average_marks (rounded to 2 decimal places)
14 Sort the output by average_marks in ascending order so that the
 15 lowest-performing class-subject combinations appear first.
 17
 18
 19
20
 21
22
 23
 24
 25 -----
 26 Database Name: fs
 27 -----
 28
 29 TABLE: students
 30
                                Default Extra
 31 Field Type Null Key
 32 roll_no int YES
                                NULL
 33 name varchar(50) YES
                                       NULL
 34 class varchar(10) YES
                                       NULL
 35 section char(1) YES
                                NULL
36 subject varchar(30) YES
                                       NULL
 37 marks int YES
                                NULL
 38
 39 -----
 40
 41
 42
 44 Sample Output:
 46 class subject average_marks
 47 10 Science 81.00
 48 10 Math 85.40
 49
 50
51
52
53
54
55
56
57 */
 58
59
```