Question - 1

SQL: Detecting Potential Payment Fraud in an Online Marketplace

The company needs a report identifying users who have failed transactions using different payment methods. Failed transactions have "Failed" in the *status* field.

The result should have the following columns: *user_id | failed_transactions | distinct_payment_methods*.

- *user_id* – User attempting multiple failed transactions.
- *failed_transactions* – Total number of failed transactions.
- *distinct_payment_methods* – Total number of unique payment methods used.

Note:

- Only users who have made more than 5 failed transactions in the entire dataset should be included in the report.
- Row order does not matter.

▼ Schema

transactions

Name	Type	Constraint	Description
transaction_id	INT	PRIMARY KEY	Unique identifier for a transaction
user_id	INT		User attempting the payment
payment_method	VARCHAR(255)		Payment method used
amount	DECIMAL(10,2)		Transaction amount
transaction_date	DATE		Date of the transaction
status	VARCHAR(255)		Status of transaction

▼ Sample Data Tables

transactions

transaction_id	user_id	payment_method	amount	transaction_date	status
101	202	Credit Card	200.43	2025-02-16	Completed
102	203	Netbanking	3233.10	2025-03-11	Failed
103	203	Netbanking	1195.35	2025-02-24	Failed
104	203	Debit Card	376.11	2025-03-10	Failed
105	203	Netbanking	112.01	2025-04-04	Failed
106	203	Credit Card	111.1	2025-09-12	Failed
107	203	Debit Card	2344.5	2025-10-03	Failed

Sample Output

```
user_id  failed_transactions  distinct_payment_methods
203        6                            3
```

Explanation

The user with *user_id* 203 attempted 6 distinct transactions, which have the status as "Failed", thus it is a potential case of fraud.

Question - 2
SQL: Average Response Time

A customer support team wants to analyze response times for resolving tickets to identify performance metrics and improve service quality. The goal is to generate a report calculating the average response time for successfully resolved customer support tickets. Resolved tickets have a value in the *resolved_at* field.

The result should have the following columns: *average_response_time*.

- *average_response_time* - The average time between *created_at* and *resolved_at*, calculated in hours and set to two decimal places, including trailing zeros if necessary (e.g., 5.00).

▾ Schema

support_tickets

Name	Type	Constraint	Description
id	INT	PRIMARY KEY	Unique identifier for each support ticket
customer_id	INT		Reference to the customer who created the ticket
created_at	VARCHAR(19)		Date and time when the ticket was created
resolved_at	VARCHAR(19)		Date and time when the ticket was resolved

▾ Sample Data Tables

support_tickets

id	customer_id	created_at	resolved_at
1	1	2023-12-21 05:42:00	2024-01-01 05:42:00
2	2	2023-07-08 14:22:00	NULL
3	3	2023-05-22 08:54:00	2023-06-17 08:54:00

Sample Output

```
average_response_time
444.00
```

Explanation

The sample output shows the average response time for resolved tickets to be 444.00, excluding Ticket *id* 2 since it has not been resolved.