*Week 2: Essential Data Retrieval & Filtering (Focus on Expense Tracker Data)*

Part 1: Retrieving Data with SELECT (30 minutes)

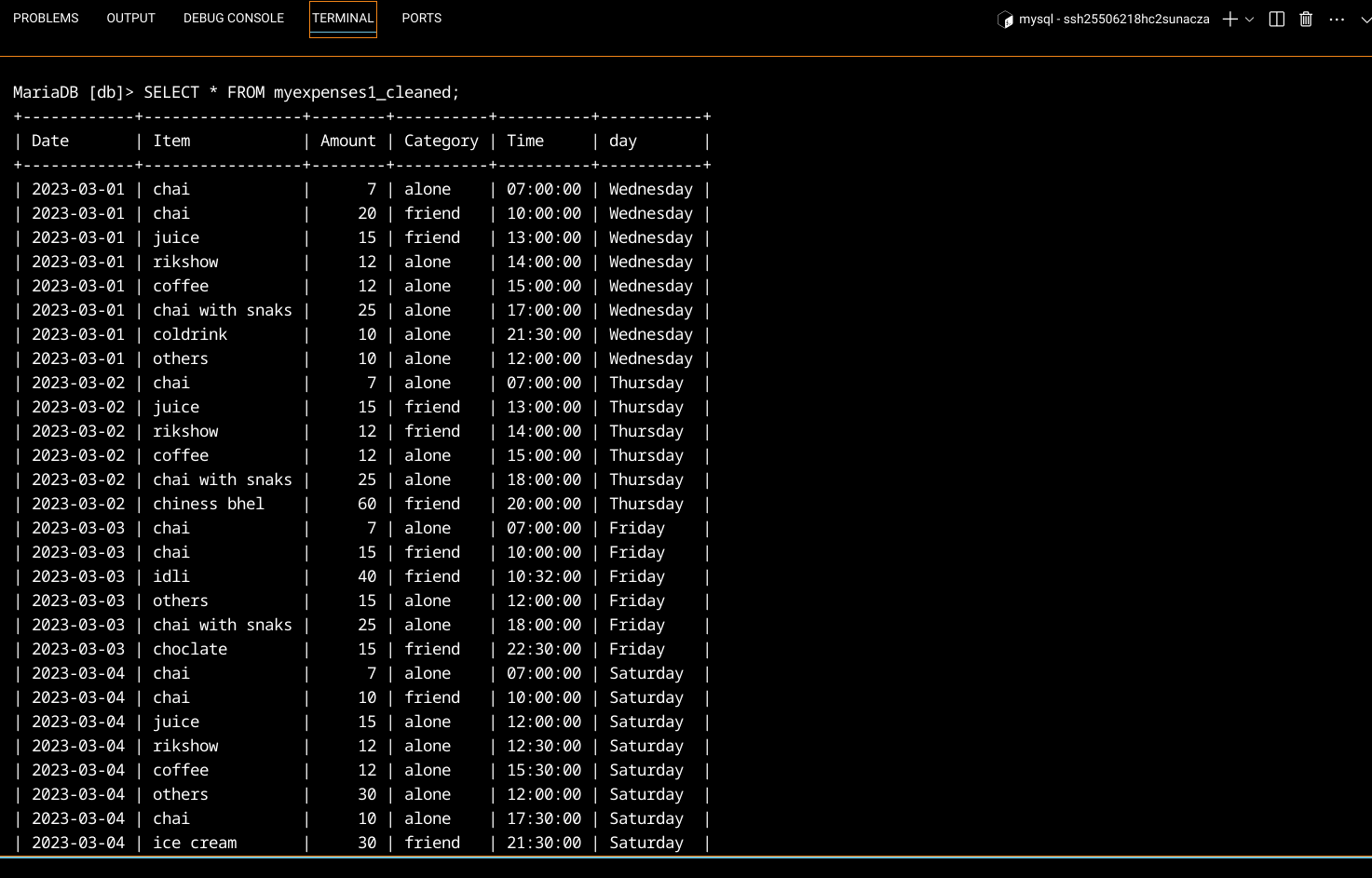
Based on the Expense Tracker table you designed in Week 1, which likely includes columns like "expense\_id," "amount," "date," and "category," complete the following tasks:

1.1 Retrieving All Expenses:

Write a SQL query to retrieve all data points (columns) from the "Expenses" table.

Answer:

MariaDB [db]> SELECT \* FROM myexpenses1\_cleaned;



1.2 Specific Columns:

Modify your query to select only specific columns relevant to your analysis.For example, you might choose "date," "category," and "amount" to analyze spending patterns by category and date.

Answer:

MariaDB [db]> SELECT Date, Category, Amount FROM myexpenses1\_cleaned;

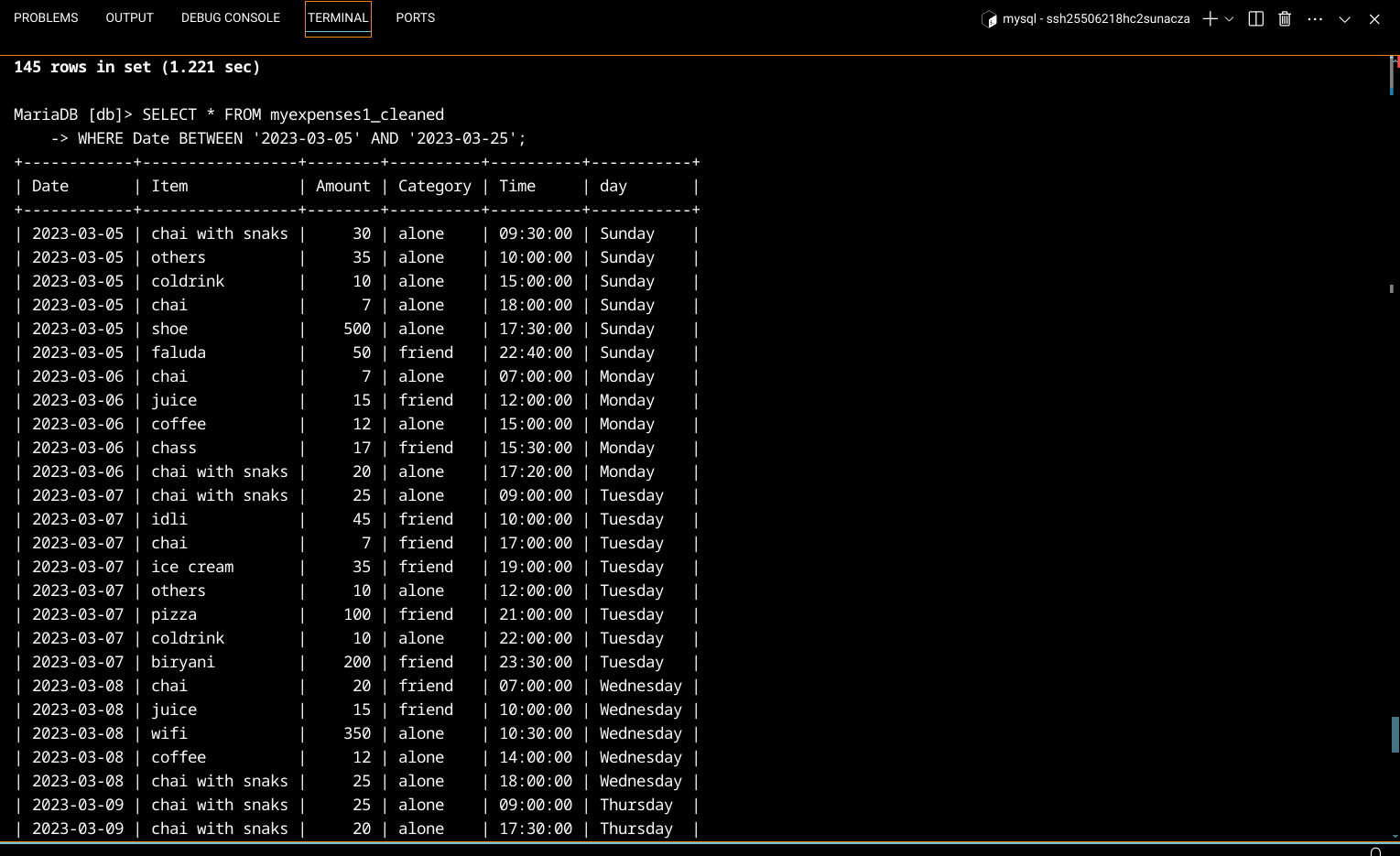


1.3 Filtering by Date Range: Write a query to retrieve expenses charged between a specific date range (e.g., January 1, 2024, to February 15, 2024). Remember to use the appropriate data type for the "date" column when specifying the date range in your query.

Answer :

MariaDB [db]> SELECT \* FROM myexpenses1\_cleaned

-> WHERE Date BETWEEN '2023-03-05' AND '2023-03-25';



Part 2: Filtering with WHERE Clause (45 minutes)

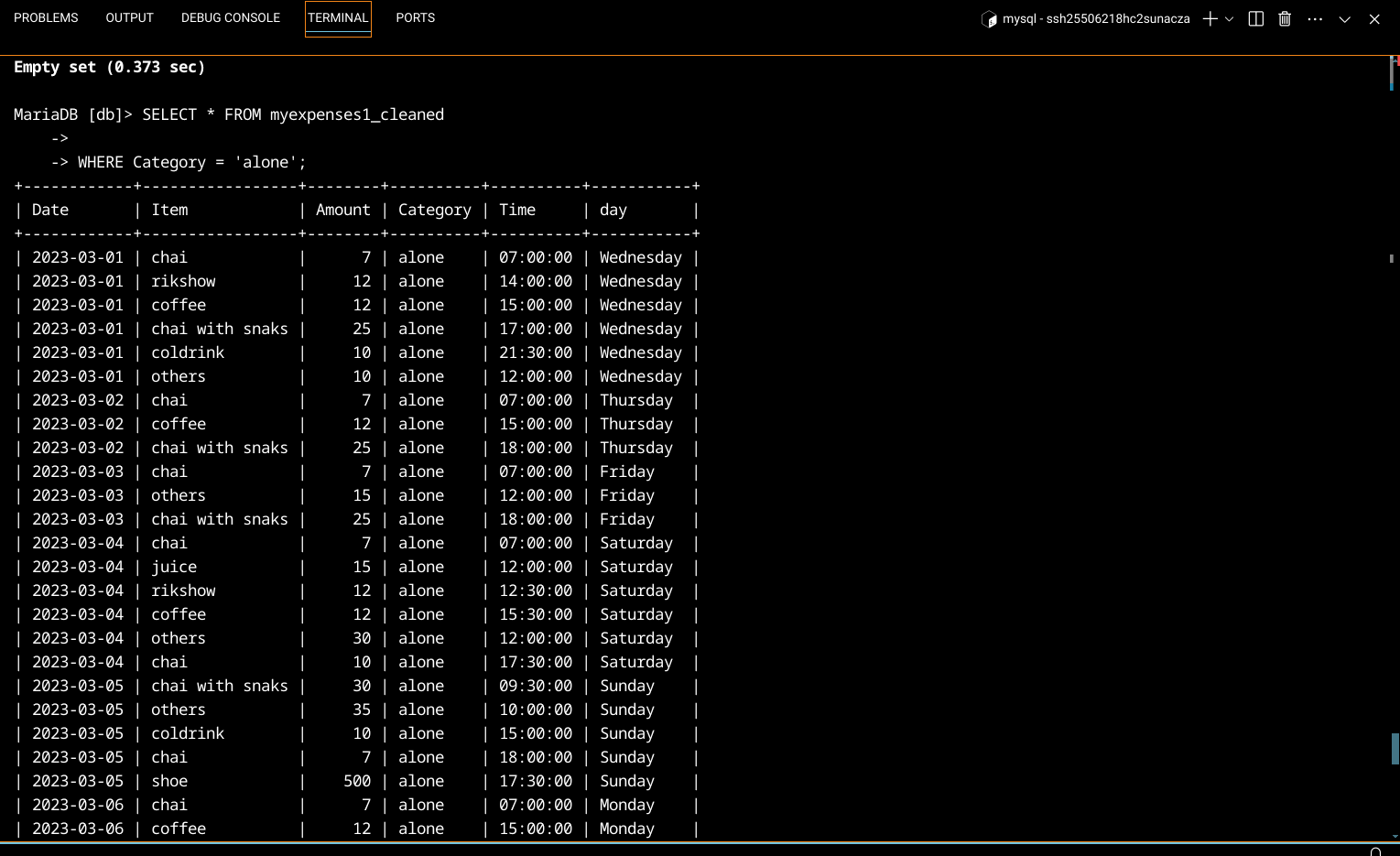
2.1 Filtering by Category: Write a query to find all expenses belonging to a specific category (e.g., "Entertainment").

Answer:

MariaDB [db]> SELECT \* FROM myexpenses1\_cleaned

->

-> WHERE Category = 'alone';

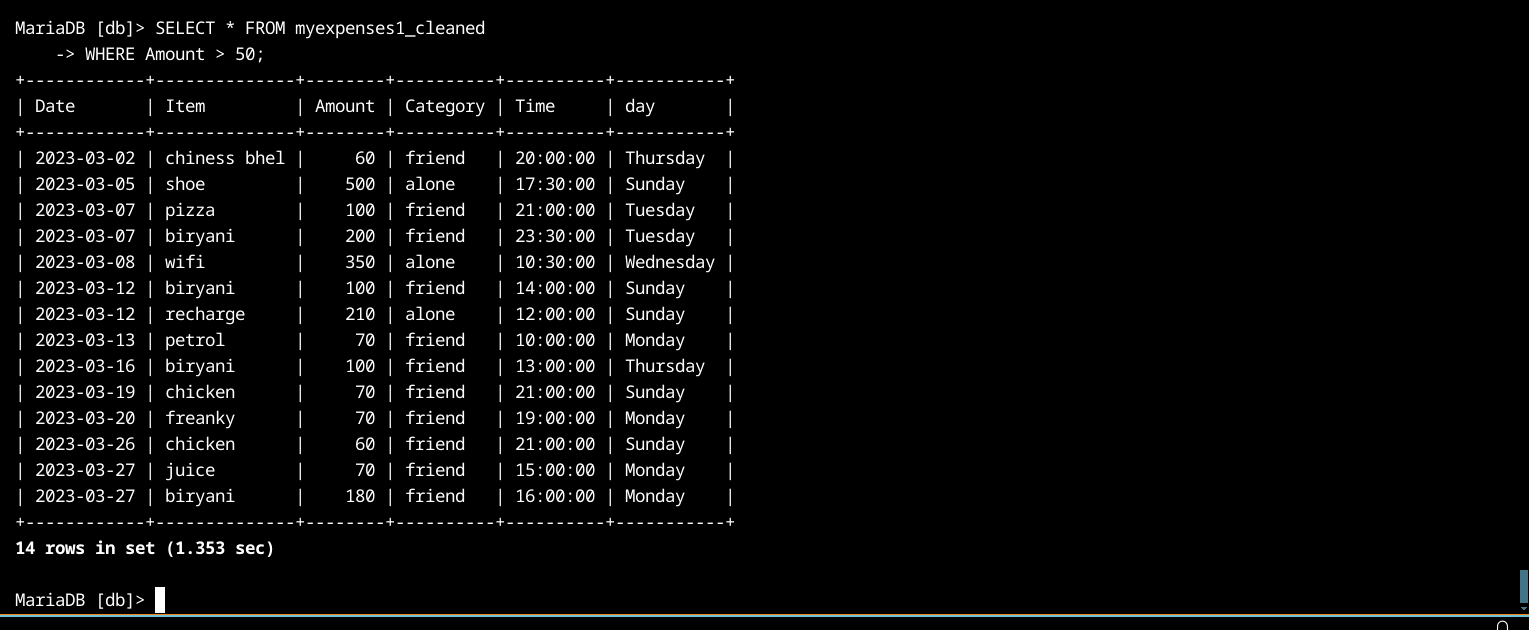


2.2 Filtering with Comparison Operators: Find expenses with an amount greater than a certain value (e.g., $50).

Answer:

MariaDB [db]> SELECT \* FROM myexpenses1\_cleaned

-> WHERE Amount > 50;



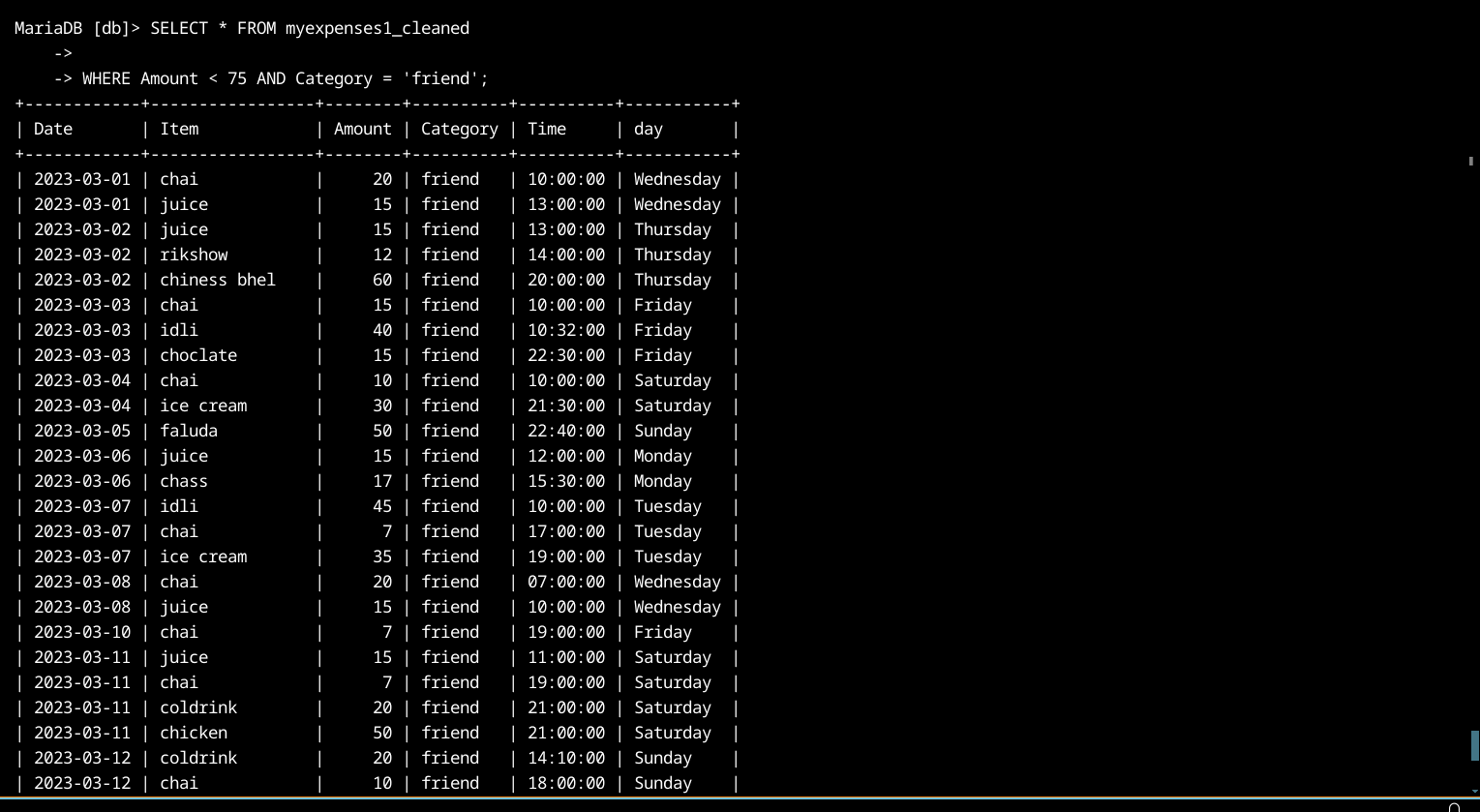
2.3 Combining Filters (AND):  
Refine your query to find expenses that meet multiple criteria. For example, you might search for expenses greater than $75 AND belonging to the "Food" category.

Answer:

MariaDB [db]> SELECT \* FROM myexpenses1\_cleaned

->

-> WHERE Amount < 75 AND Category = 'friend';



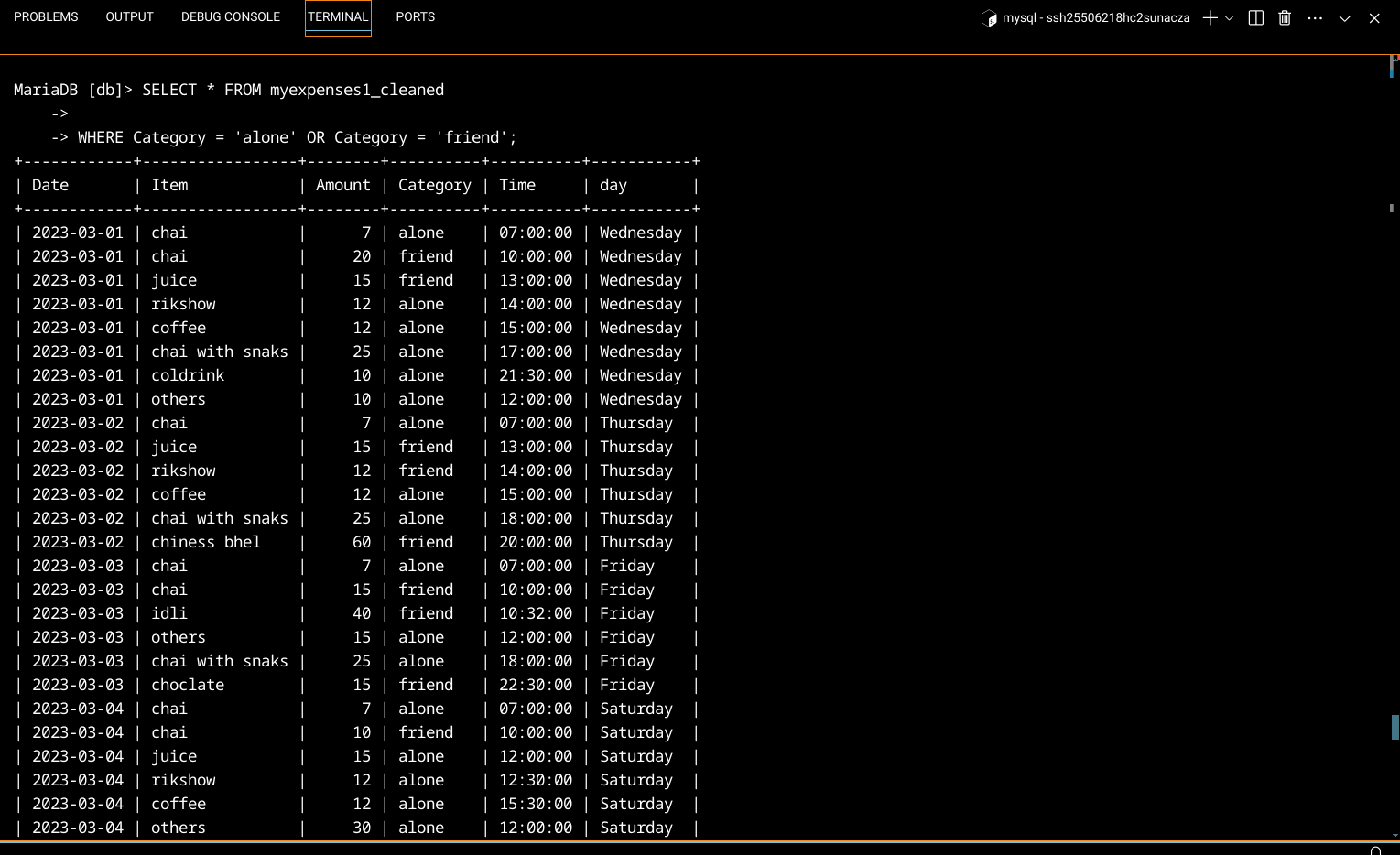
2.4 Combining Filters (OR): Modify your query to find expenses belonging to one category or another (e.g., "Transportation" OR "Groceries").

Answer:

MariaDB [db]> SELECT \* FROM myexpenses1\_cleaned

->

-> WHERE Category = 'alone' OR Category = 'friend';



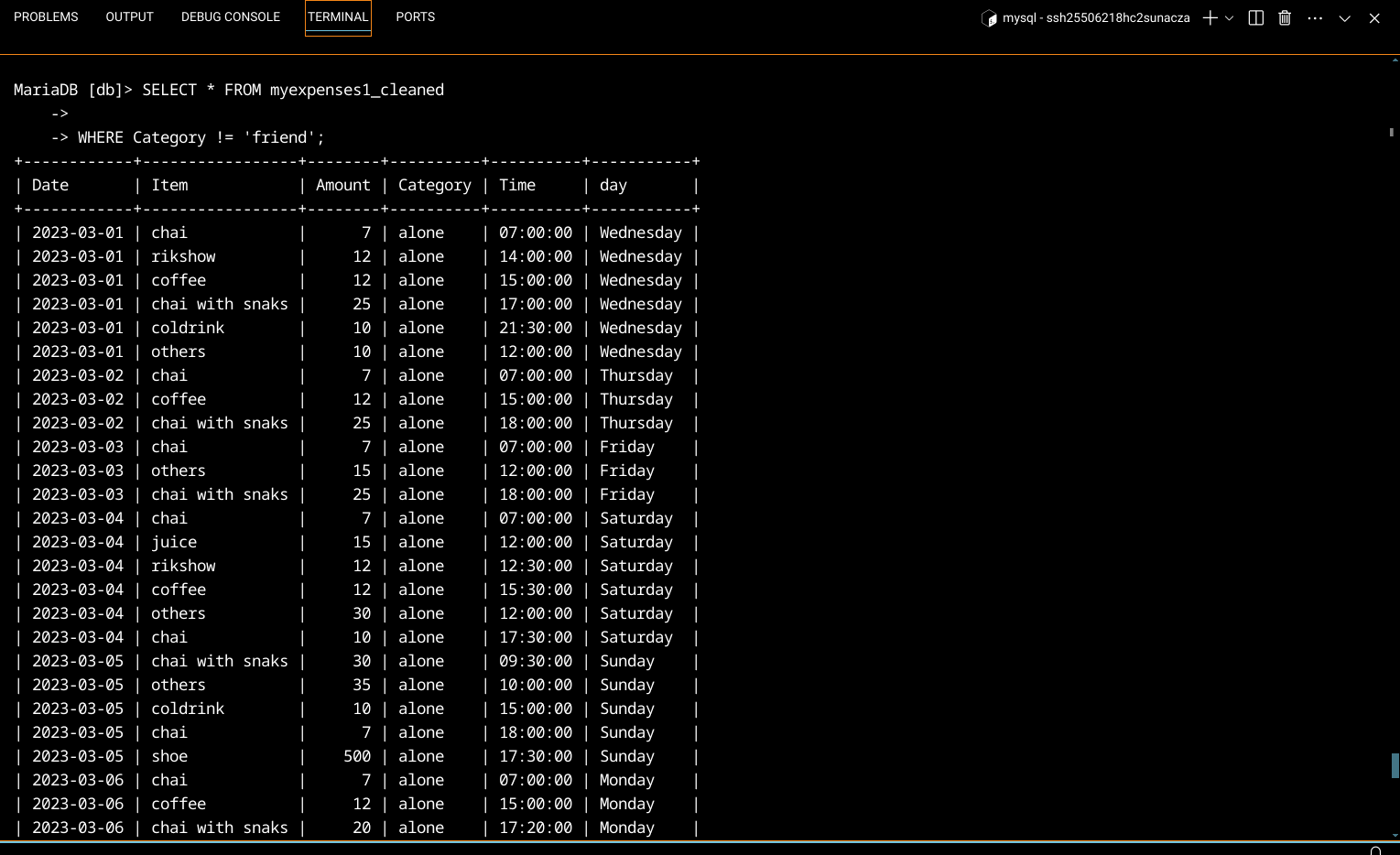
2.5 Filtering with NOT: Write a query to display expenses that are NOT related to a specific category (e.g., "Rent").

Answer:

MariaDB [db]> SELECT \* FROM myexpenses1\_cleaned

->

-> WHERE Category != 'friend';



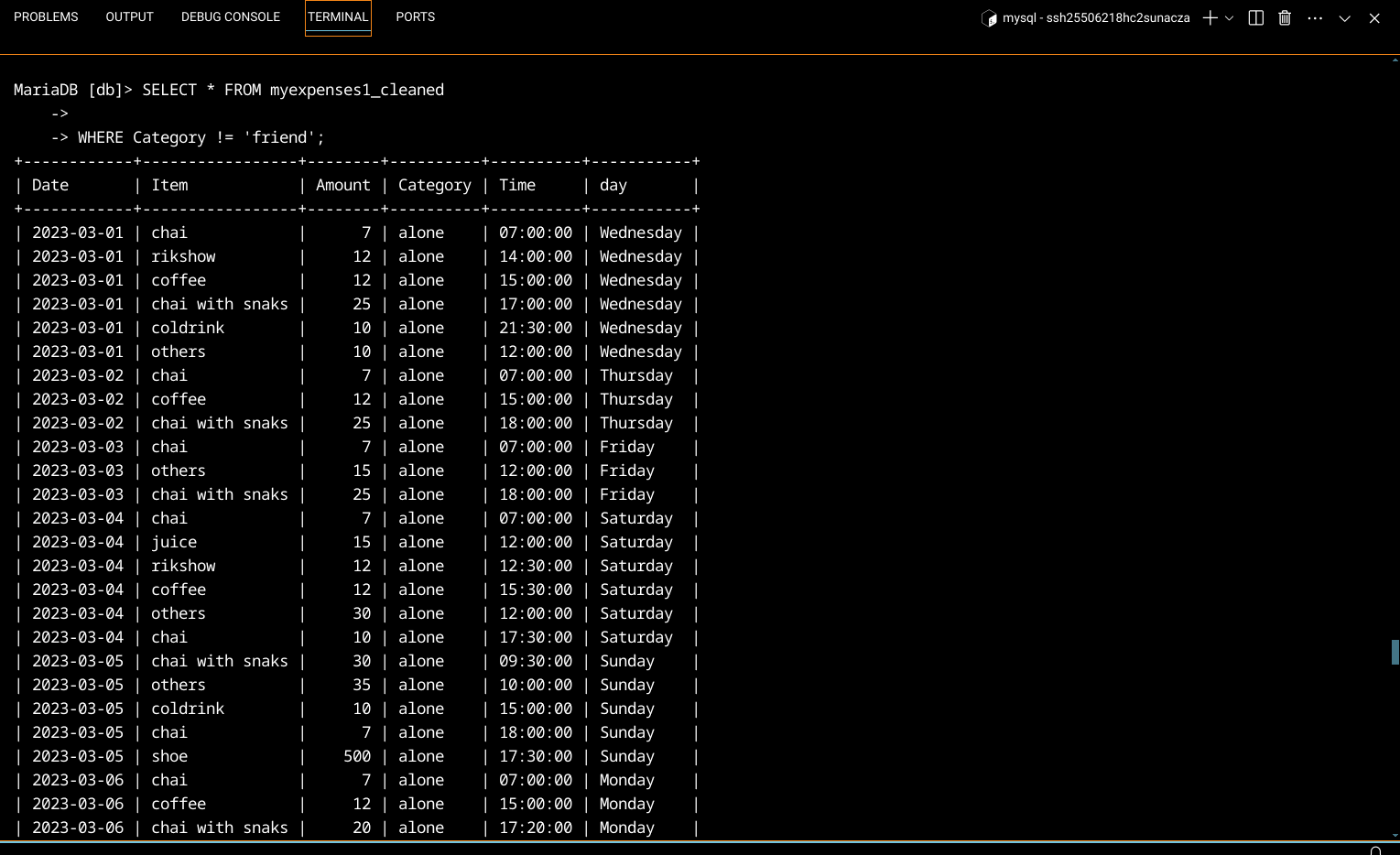
Part 3: Sorting Retrieved Data (45 minutes)

3.1 Sorting by Amount: Write a query to display all expenses sorted by amount in a specific order (e.g., descending order for highest to lowest spending).

Answer:

MariaDB [db]> SELECT \* FROM myexpenses1\_cleaned

-> ORDER BY Amount DESC;



3.2 Sorting by Date and Category:  
Modify your query to sort expenses based on multiple columns. For example, you might sort first by date (descending order) and then by category (ascending order) to see recent spending trends by category.

Answer:

MariaDB [db]> SELECT \* FROM myexpenses1\_cleaned

-> ORDER BY Date DESC, Category ASC;

