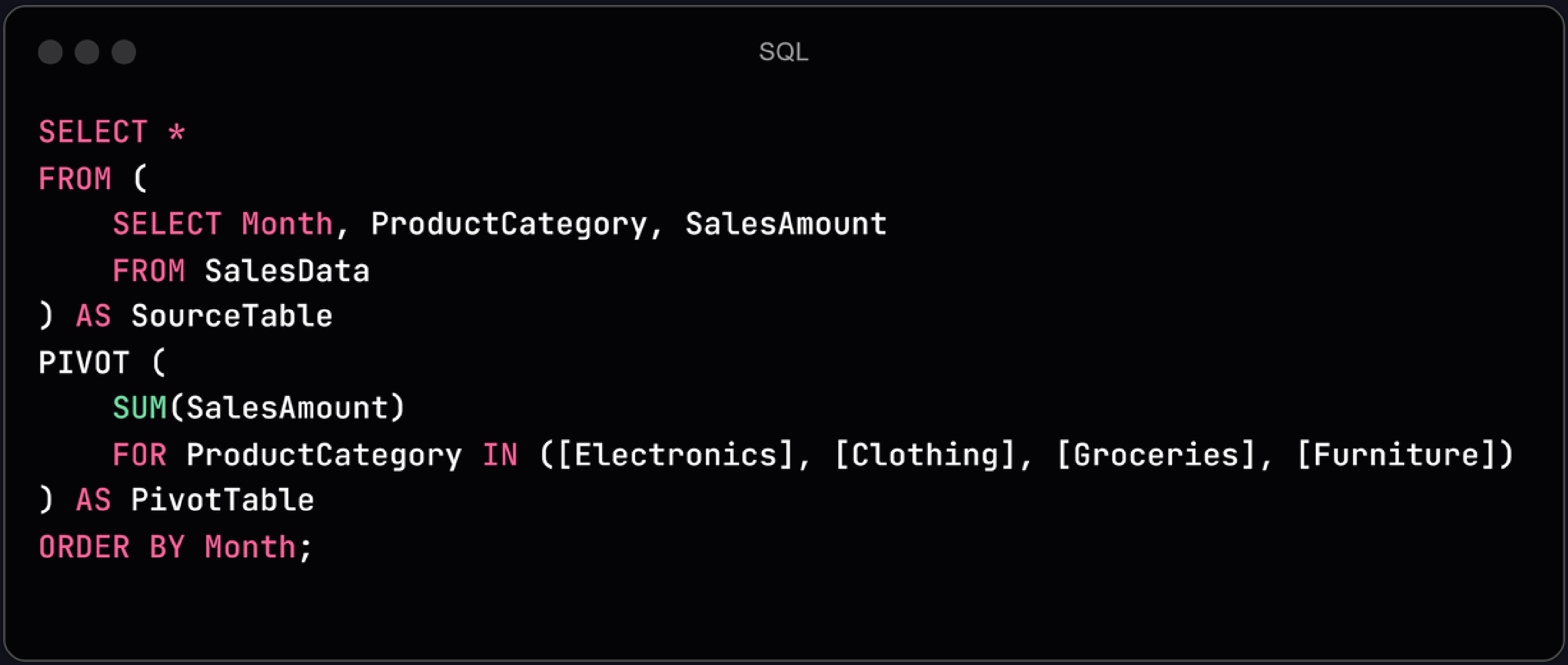
PIVOT()

Problem Statement: As a data analyst in a retail company, you want to analyze monthly sales data for different products across various categories. You need to transform this data into a format that allows for easy comparison of sales performance by product category each month. Using the PIVOT function will help summarize the data effectively.



```sql
SELECT *
FROM (
    SELECT Month, ProductCategory, SalesAmount
    FROM SalesData
) AS SourceTable
PIVOT (
    SUM(SalesAmount)
    FOR ProductCategory IN ([Electronics], [Clothing], [Groceries], [Furniture])
) AS PivotTable
ORDER BY Month;
```

PIVOT()

BREAKDOWN:

- **Inner SELECT Statement:** Retrieves the necessary columns (Month, ProductCategory, and SalesAmount) from the SalesData table. This provides the raw data needed for transformation.

- **Source Table Alias:** The inner query is given an alias (SourceTable) to be used as the input for the PIVOT operation.

- **PIVOT Function:** The PIVOT function is applied to the SalesAmount column. It aggregates the sales data using SUM(), which calculates total sales for each product category.

- **FOR Clause:** Specifies ProductCategory as the column to pivot on, transforming unique product categories into individual columns (e.g., Electronics, Clothing, etc.).

- **IN Clause:** Lists the specific product categories that will become columns in the resulting table. Ensure that these categories exist in the data.

- **ORDER BY Clause:** Sorts the final result set by Month, allowing for a clear chronological view of sales performance.