SQL Challenge: Orders by Weekday

Problem Statement

You are given two tables:

- `Items` containing item details including category.
- `Orders` containing order quantities and order dates.

Your task is to **generate a summary report** that displays the **total quantity of items ordered per category**, broken down **by each day of the week (Monday to Sunday)**.

Input Tables

Items

item_id	item_category
1	Electronics
2	Grocery
3	Furniture

Orders

order_id	item_id	order_date	quantity
101	1	2023-05-01	2
102	2	2023-05-02	3
...	...	...	...

Expected Output

CATEGORY	MONDAY	TUESDAY	...	SUNDAY
Electronics	10	5	...	2
Grocery	3	6	...	1
Furniture	0	0	...	4

This table summarizes item orders grouped by **item category and weekday**.

SQL Query

```
SELECT *
FROM (
    SELECT
        it.item_category AS CATEGORY,
        SUM(CASE WHEN DATENAME(dw, od.order_date) = 'Monday' THEN od.quantity ELSE 0
```

```
END) AS MONDAY,
        SUM(CASE WHEN DATENAME(dw, od.order_date) = 'Tuesday' THEN od.quantity ELSE 0
END) AS TUESDAY,
        SUM(CASE WHEN DATENAME(dw, od.order_date) = 'Wednesday' THEN od.quantity ELSE
0 END) AS WEDNESDAY,
        SUM(CASE WHEN DATENAME(dw, od.order_date) = 'Thursday' THEN od.quantity ELSE 0
END) AS THURSDAY,
        SUM(CASE WHEN DATENAME(dw, od.order_date) = 'Friday' THEN od.quantity ELSE 0
END) AS FRIDAY,
        SUM(CASE WHEN DATENAME(dw, od.order_date) = 'Saturday' THEN od.quantity ELSE 0
END) AS SATURDAY,
        SUM(CASE WHEN DATENAME(dw, od.order_date) = 'Sunday' THEN od.quantity ELSE 0
END) AS SUNDAY
    FROM Items AS it
    LEFT JOIN Orders AS od
        ON od.item_id = it.item_id
    GROUP BY it.item_category
) temp
ORDER BY CATEGORY ASC;
```

Understanding the Problem

We need to create a report that shows:

1. Total quantities of items ordered per category
2. Broken down by each day of the week (Monday through Sunday)
3. Using data from two tables: Items and Orders

Breaking Down the SQL Query

The query uses the following techniques:

1. JOIN to combine data from two tables
2. CASE statements to filter data by weekday
3. GROUP BY to organize data by category
4. Subquery (derived table) for clean formatting

Let's go through it step by step:

1. The Main Structure

```
SELECT *
FROM (
    -- Subquery here
) temp
ORDER BY CATEGORY ASC;
```

This outer query simply selects all columns from a subquery (temporary result set named "temp") and sorts the results alphabetically by category.

2. The Core Subquery

```
SELECT
    it.item_category AS CATEGORY,
    SUM(CASE WHEN DATENAME(dw, od.order_date) = 'Monday' THEN od.quantity ELSE 0 END)
AS MONDAY,
    SUM(CASE WHEN DATENAME(dw, od.order_date) = 'Tuesday' THEN od.quantity ELSE 0 END)
AS TUESDAY,
    -- Other days follow the same pattern
FROM Items AS it
LEFT JOIN Orders AS od
    ON od.item_id = it.item_id
GROUP BY it.item_category
```

This is where the main work happens.

3. Understanding the JOIN

```
FROM Items AS it
LEFT JOIN Orders AS od
    ON od.item_id = it.item_id
```

- `Items AS it`: We're using the Items table with alias "it"
- `LEFT JOIN Orders AS od`: We're joining with the Orders table (alias "od")
- `ON od.item_id = it.item_id`: The join condition matches items in both tables
- We use LEFT JOIN (not INNER JOIN) to ensure we include all item categories even if they have no orders

4. The CASE Expressions

```
SUM(CASE WHEN DATENAME(dw, od.order_date) = 'Monday' THEN od.quantity ELSE 0 END) AS
MONDAY
```

Let's break this down:

- `DATENAME(dw, od.order_date)`: Gets the weekday name from the order date
- `CASE WHEN ... THEN ... ELSE ... END`: Conditional expression
  - If the day is Monday, use the quantity value
  - If not, use 0
- `SUM()`: Adds up all the values that match our condition

This is repeated for each day of the week.

5. The GROUP BY Clause

```
GROUP BY it.item_category
```

This groups all our results by item category, so we get one row per category.

Explaining the Output

The result will be a table with:

- First column: Category name
- Seven columns (one for each day of the week)

- Each cell shows the total quantity of items ordered from that category on that day
- Categories are sorted alphabetically

For example, if there were 10 Electronics items ordered on Monday, the MONDAY column for the Electronics row would show 10.

Key Concepts for Beginners

1. **LEFT JOIN**: Ensures we include all categories, even if they have no orders
2. **CASE statements**: Allow us to conditionally count only quantities from a specific day
3. **DATENAME function**: Extracts the weekday name from a date
4. **SUM function**: Adds up all quantities that match our condition
5. **GROUP BY**: Combines results by category
6. **Subquery**: Creates a temporary result set that can be treated like a table

This approach is called "pivoting" - transforming rows (dates) into columns (days of the week) to create a more readable summary report.