🧪 Lab 1 – Task 3: Use `GROUP BY` to Identify Number of Items per Category

🎯 Objective:

Use SQL to count the number of products per category and segment in the `products` table.

🛠️ Steps:

1. Run the following query to group products by category:

   SQL

   ```
   SELECT category, COUNT(*) AS itemCount

   FROM `thelook_ecommerce.products`

   GROUP BY category;
   ```

2. To group by segment instead of category, run:

   SQL

   ```
   SELECT segment, COUNT(*) AS itemCount

   FROM `thelook_ecommerce.products`

   GROUP BY segment;
   ```

✅ Correct Answers:

Question 1: How would you modify the query to identify which segment each item has been assigned?

- ✅ **Replace `category` with `segment` in both the SELECT and GROUP BY clauses.**

Question 2: What is the total number of luxury items?

- ✅ **6476**

📘 Feedback:

✅ Excellent! You used `GROUP BY` correctly to aggregate data by category and segment. This helps identify product distribution and potential outliers in each group.

🧪 Lab 1 – Task 4: Filter Data Using `GROUP BY` and `HAVING`

🎯 Objective:

Filter out categories with fewer than 1000 items using the `HAVING` clause.

🛠️ Query to Run:

SQL

```
SELECT category, COUNT(*) AS itemCount
FROM `thelook_ecommerce.products`
GROUP BY category
HAVING itemCount > 1000;
```

📘 Feedback:

✅ Great job! You correctly used `HAVING` to filter aggregated results. This is useful for focusing on significant categories and ignoring noise from small groups.

🧪 Lab 1 – Task 5: Sample a BigQuery Table Using `TABLESAMPLE`

🎯 Objective:

Use `TABLESAMPLE` to retrieve a random subset of rows from the `products` table.

🛠️ Query to Run:

```
SQL
SELECT * FROM `thelook_ecommerce.products`
TABLESAMPLE SYSTEM (10 PERCENT);
```

📘 Feedback:

✅ Well done! Sampling is a powerful technique for quick data exploration and testing queries without scanning the entire dataset. Remember, results will vary each time due to randomness.

🧪 Lab 1 – Task 6: Explore the `order_items` Table

🎯 Objective:

Explore order data and identify the user with the highest total order value.

🛠️ Queries to Run:

1. Preview the table:

   ```
   SQL

   SELECT * FROM `thelook_ecommerce.order_items`

   LIMIT 10;
   ```

2. Count orders by status:

   ```
   SQL

   SELECT status, COUNT(*) AS total_orders

   FROM `thelook_ecommerce.order_items`

   GROUP BY status;
   ```

3. Find the top spender:

   SQL

   ```
   SELECT user_id,

   SUM(sale_price) AS total_amount

   FROM `thelook_ecommerce.order_items`

   GROUP BY user_id

   ORDER BY total_amount DESC

   LIMIT 1;
   ```

✅ Correct Answer:

Question: What is the user ID associated with the greatest total order value?

- ✅ **81321**

📘 Feedback:

✅ Excellent! You successfully explored the `order_items` table and used aggregation to identify key metrics. This is essential for understanding customer behavior and sales performance.

🎉 Conclusion

You've completed **Lab 1: Practice Transformation Methods** and applied key SQL techniques:

- ✅ **Limiting** rows to preview data.
- ✅ **Identifying duplicates** using `COUNT(DISTINCT name)`.

- ✅ **Aggregating** data with `GROUP BY`.
- ✅ **Filtering** with `HAVING`.
- ✅ **Sampling** with `TABLESAMPLE`.
- ✅ **Exploring** order data and identifying top customers.

You're now equipped to use **data limiting, sampling, and aggregation** to assess data quality and prepare for deeper analysis.
