SQL Assessment Tasks for Data Analytics Working Student

Dataset Overview

You will work with the following tables:

1. **orders**
   Contains individual order records, including the customer, date, product, and country.
   - Columns: `order_id`, `customer_id`, `order_date`, `product_number`, `country_iso`

2. **marketing_sources**
   Information about marketing channels associated with orders.
   - Columns: `country_iso`, `order_id`, `ordertime`, `utm_campaign`, `reporting_channel`

3. **product_locale**
   Contains localized product names.
   - Columns: `sku`, `product_name`, `locale`

4. **product_universal**
   Contains universal product information such as categories.
   - Columns: `sku`, `main_category`

All queries must be written using **BigQuery SQL syntax**.

Task 1: Customer Order Activity in Early Lifecycle

Objective: Understand early behavior of new customers.

Instructions:

1. For each customer, determine their **first order date**.
2. Calculate how many orders they placed within **10, 15, and 20 days** after their first order.
3. Bonus: Extend the analysis to show the breakdown by reporting_channel from the **marketing_sources** table.

Task 2: Product Popularity by Country

Objective: Identify popular and unpopular products and categories by geography.

Instructions:

1. For each country_iso, calculate:
    - Total number of orders and skus by main_category.
2. Use **RANK() or DENSE_RANK()** to:
    - Identify the **top 5** most ordered and **bottom 5** least ordered products per country.

Task 3: First Orders and New Customer Share by Day

Objective: Analyze the share of first-time buyers over time.

Instructions:

1. Use **ROW_NUMBER()** partitioned by customer_id to identify each customer's first order.
2. For each order_date, compute:
    - Total number of orders.
    - Number of orders that were the customer's **first ever**.
    - **Percentage** of total orders made by new customers.
3. Bonus: Add a breakdown by reporting_channel.

Tips

- Be sure to handle potential duplicates, nulls or nan values.
- Use SAFE_CAST or PARSE_DATE as needed for date operations in BigQuery.
- You may need to join multiple tables for full context in each task.