Data-Analyst Assignment

Objective

The goal of this test is to evaluate your analytical skills and SQL coding skills. The following steps will guide you in submitting your solution.

1. Load data into your DB. For simplicity in evaluation we request you to use either SQLite DB, MySQL DB, or PostgreSQL DB.
2. List 4 insights you can derive from this data. Your insights should include:
   - A description of the insight.
   - The SQL query used to obtain data for your insight.
   - Result set of your SQL query.
3. See an example insight below for hints.
4. Use the result template file `results.xlsx` to share your insights. If you do not have access to Excel, upload your results to Google sheets and share a link to the sheet.
5. Evaluation
   - The query will be evaluated for precision and quality.
   - The insight will be evaluated on reasoning, uniqueness, and depth of enquiry.

Background

This folder contains seller on-boarding data from S-Mart, an e-commerce marketplace for businesses.

To become a seller on S-Mart, a business user has to first signup on a landing page. On successful signup the business user is called a lead and the details of the signup are stored in the leads table. Leads are pursued by sales representatives for on-boarding. On successful on-boarding, a lead becomes a seller and can start selling on the e-commerce platform.

Loading Data

The steps for loading data depends on your choice of DB

SQLite DB

- The DB file `sqlite3_data_analyst.db` is shared
- Execute the following command in SQLite shell:
  - `.open /path_to_folder/sqlite3_data_analyst.db`

MySQL DB

Data my be imported into MySQL either by using the provided `mysql_create_data.sql` script or by importing the raw csv data.

1. Use one of the below commands to import data using the `mysql_create_data.sql` script:
   - Using terminal: `mysql -h hostname -u user database < path/to/mysql_create_data.sql`

- If using MySQL GUI tool, then import the file `mysql_create_data.sql` and execute it
- Use the command `source path/to/mysql_create_data.sql` in MySQL shell

2. Use `leads_dataset.csv` and `sellers_dataset.csv` files to load the data yourself into the DB.

PostgreSQL

Use `leads_dataset.csv` and `sellers_dataset.csv` files to load the data yourself into the DB.

Schema

After successfully loading the data, you will find two tables in your database: leads and sellers. Their schema is tabulated below for your reference.

Leads

Column	Description
lead_id	unique ID for a lead
signup_date	date of signing up
landing_page_id	unique ID for a landing page
mkt_origin	marketing origin of the lead

Sellers

Column	Description
seller_id	unique ID for a seller
lead_id	lead ID of the seller
sales_rep_id	unique ID of the sales rep that led to onboarding
seller_onboarded_at	onboarding timestamp
business_segment	category of business
lead_type	type of lead
lead_behaviour_profile	behavioral segment
business_type	manufacturer, reseller, or other

Example Insight

Description

Of the top twenty landing pages by number of sign-ups, we find that the top 5 pages have a have an average signup conversion percentage of 20.9%. This drops to about 15.2% for the next 5 pages.

Queries

1. Average signup conversion top 5 pages by conversion %, where the pages are from the top 20 pages by number of sign-ups

```
SELECT AVG(seller_conversion_pct)
FROM
    (SELECT landing_page_id,
            seller_conversion_pct
     FROM
         (SELECT landing_page_id,
                 COUNT(seller_id) AS seller_count,
                 COUNT(seller_id) * 100 / COUNT(l.lead_id) AS seller_convers
          FROM leads AS l
          LEFT JOIN sellers AS s ON l.lead_id = s.lead_id
          GROUP BY 1
          ORDER BY 2 DESC
          LIMIT 20) AS x
     ORDER BY 2 DESC
     LIMIT 5) AS y
```

2. Average signup conversion top 6 - 10 pages by conversion %, where the pages are from the top 20 pages by number of sign-ups

```
SELECT AVG(seller_conversion_pct)
FROM
    (SELECT landing_page_id,
            seller_conversion_pct
     FROM
         (SELECT landing_page_id,
                 COUNT(seller_id) AS seller_count,
                 COUNT(seller_id) * 100 / COUNT(l.lead_id) AS seller_convers
          FROM leads AS l
          LEFT JOIN sellers AS s ON l.lead_id = s.lead_id
          GROUP BY 1
          ORDER BY 2 DESC
          LIMIT 20) AS x
     ORDER BY 2 DESC
     LIMIT 5
     OFFSET 5) AS y
```

Results

1. 20.91836
2. 15.2166

Useful Links

- Download SQLite Note that SQLite does not require installation.
- Download MySQL community server 8.0.17