Hevo Exercise - Assessment II : Post-Load with Hevo Models

Prerequisites (Must have)

- Access to GitHub (public repo for deliverables)
- Basic database knowledge (PostgreSQL) and familiarity with Snowflake SQL commands.
- Ability to run a local PostgreSQL instance (Docker suggested)
- Basic networking knowledge to expose the local DB to Hevo
- Snowflake Trial account
- Hevo's Free Product Trial (via **Snowflake Partner Connect**)
- Ability to understand and build customized SQL queries, based on the requirements.

Assessment: Messy E-Commerce Orders (Snowflake Data Cleaning Challenge)

Background

You are working with raw data ingested from PostgreSQL into Snowflake. After the load, you observe duplicates, inconsistent formats, <null> values, inactive records, and orphaned references.

Your task is to clean and transform the data into a single reliable dataset for analytics.

Raw Tables schema

1. customers_raw

Table schema for **customers_raw**					
customer_id	**email**	**phone**	**country_code**	**updated_at**	**created_at**
101	John@example.com	111-222-3333	US	2025-07-01 10:15:00	2025-01-01 08:00:00
101	john.d@example.com	(111)2223333	usa	2025-07-03 14:25:00	2025-01-01 08:00:00
102	alice@example.com	<null>	UnitedStates	2025-07-01 09:10:00	<null>

Table schema for **customers_raw**					
customer_id	**email**	**phone**	**country_code**	**updated_at**	**created_at**
103	michael@abc.com	9998887777	<null>	2025-07-02 12:45:00	2025-03-01 10:00:00
104	bob@xyz.com		IND	2025-07-05 15:00:00	2025-03-10 09:30:00
104	bob@xyz.com		India	2025-07-06 18:00:00	2025-03-10 09:30:00
106	duplicate@email.com	1234567890	SINGAPORE	2025-07-01 08:00:00	2025-04-01 11:45:00
106	duplicate@email.com	123-456-7890	SG	2025-07-10 12:00:00	2025-04-01 11:45:00
108	<null>	<null>	<null>	<null>	<null>

2. **orders_raw**

Table schema for **orders_raw**					
order_id	**customer_id**	**product_id**	**amount**	**created_at**	**currency**
5001	101	P01	120.00	2025-07-10 09:00:00	USD
5002	102	P02	80.5	2025-07-10 09:05:00	usd
5003	103	<null>	200.00	2025-07-10 09:15:00	INR
5004	105	P99	<null>	2025-07-10 09:20:00	<null>
5002	102	P02	80.50	2025-07-10 09:05:00	USD
5005	106	P03	-50	2025-07-10 09:25:00	SGD
5006	107		300	2025-07-11 10:00:00	usd

Table schema for **orders_raw**					
order_id	**customer_id**	**product_id**	**amount**	**created_at**	**currency**
5007	108	P04	500	2025-07-11 10:15:00	EUR

3. products_raw

Table schema for **products_raw**			
product_id	**product_name**	**category**	**active_flag**
P01	keyboard	hardware	Y
P02	MOUSE	Hardware	Y
P03	Monitor	Hardware	N
P04	Premium Cable	Accessory	Y

4. country_dim

Table schema for **country_dim**	
country_name	**iso_code**
United States	US
India	IN
Singapore	SG
Unknown	<null>

Tasks

1. **Set up Postgres Database**
    - Install a PostgreSQL database locally
2. **Create tables & load data**
    - Create tables in PostgreSQL (customers_raw, orders_raw, products_raw, country_dim) using CREATE statements as defined in the schema above, and load data into them using INSERT queries.
3. **Set up the Hevo pipeline**:
    - Source = PostgreSQL
    - Destination = Snowflake
    - Ingestion mode must be **Logical Replication**
4. **Load data to Snowflake**:
    - Let Hevo push raw data into your Snowflake Trial account.

— Use Models from HERE —
Write SQL query/queries for the following tasks

5. **Deduplicate Customers**
    - Keep only the most recent record for each customer_id.
    - Standardize emails to lowercase.
    - Standardize phone numbers into a 10-digit format, or mark as "Unknown" if invalid or missing.
6. **Fix <null>s & Country Issues**
    - Standardize country_code values using country_dim.
    - Handle variations such as usa, UnitedStates, IND, SINGAPORE.
    - If created_at is <null>, replace with a default timestamp (1900-01-01).
7. **Clean Orders**
    - Remove exact duplicate rows.
    - Handle invalid amounts:

        - If amount is negative, replace with 0.
        - If amount is <null>, replace with a reasonable fallback (e.g., median of customer's transactions).
    - Standardize currency codes to uppercase.
    - Create a derived column amount_usd by converting all amounts into USD using predefined conversion rates.

8. **Clean Products**
    - Standardize product names (capitalize properly).
    - Standardize category names (In "Title Case").
    - If a product is inactive (active_flag = 'N'), mark it as "Discontinued Product".

9. **JOIN the resultants**
    - Produce a unified dataset joining customers, orders, and products.
    - If customer_id does not exist in cleaned customers, mark email as "Orphan Customer".
    - If product_id is missing or invalid, mark it as "Unknown Product".

10. **Edge Cases**
    - Customers with completely <null> records should be marked as "Invalid Customer".
    - Orders referencing non-existent customers or products should still appear in the final dataset, with appropriate placeholders.
    - Mixed currency handling should be consistent across all rows.

Deliverables (To be attached in the Google form):

- GitHub repo link
- Hevo **Account Team Name**, **Pipeline Number**, and **Model Number**
- Use **Loom** **screen recorder** to record and present all the tasks that were undertaken in a chronological order, with detailed explanations of all the complexities.

(All documentation, assumptions, and notes must live inside the README.md file in your repo.)

Note: Write SQL queries to clean and transform the above raw tables into a single final dataset ready for analytics. You can take all the help from Hevo Docs that you need to replicate data through a pipeline and transform it.

Important Notes

- Do not hardcode or publish any credentials, database info, or access keys in the repo. Use environment variables, config files outside source control, or Hevo's UI settings.

- Please follow the academic honor code while doing the exercise. Candidates will support this culture of academic honesty by neither giving nor accepting unpermitted academic aid in any work that serves as a component of grading or evaluation, including assignments, examinations, and research.

- Any violations of the above honor code will result in immediate disqualification.