Technical case

Data Engineer

Business context

Gross Merchandising Value (GMV) represents the total monetary value of completed transactions, considering only purchases whose payment has been successfully captured (i.e., *`release_date`* is populated) and not canceled.

You are provided with three event-based (CDC) tables *(documented at [link](#))*:

purchase *(events/cdc)* - Core transaction event

purchase (eventos)

transaction_datetime	transaction_date	purchase_id	buyer_id	prod_item_id	order_date	release_date	producer_id
2023-01-20 22:00:00	2023-01-20	55	15947	5	2023-01-20	2023-01-20	852852
2023-01-26 00:01:00	2023-01-26	56	369798	746520	2023-01-25	NULL	963963
2023-02-05 10:00:00	2023-02-05	55	160001	5	2023-01-20	2023-01-20	852852
2023-02-26 03:00:00	2023-02-26	69	160001	18	2023-02-26	2023-02-28	96967
2023-07-15 09:00:00	2023-07-15	55	160001	5	2023-01-20	2023-03-01	852852

product_item *(events/cdc)* - Line items and monetary amounts

product_item (eventos)

transaction_datetime	transaction_date	purchase_id	product_id	item_quantity	purchase_value
2023-01-20 22:02:00	2023-01-20	55	696969	10	50,00
2023-01-25 23:59:59	2023-01-25	56	808080	120	2400,00
2023-02-26 03:00:00	2023-02-26	69	373737	2	2000,00
2023-07-12 09:00:00	2023-07-12	55	696969	10	55,00

Purchase_extra_info *(events/cdc)* - Dimensional attributes such as subsidiary

transaction_datetime	transaction_date	purchase_id	subsidiary
2023-01-23 00:05:00	2023-01-23	55	nacional
2023-01-25 23:59:59	2023-01-25	56	internacional
2023-02-28 01:10:00	2023-02-28	69	nacional
2023-03-12 07:00:00	2023-03-12	69	internacional

These tables are ingested asynchronously into the data lake. Events may arrive late, out of order, or be re-sent for historical correction. For every logical purchase, there will always be one corresponding record in each table, but not necessarily on the same ingestion date or time.

Goal

We want you to design the ETL/data pipeline of an **immutable, historically consistent analytical table** that provides the **daily GMV by subsidiary**. Your goal is to **propose the end-to-end data modeling strategy**, including how raw CDC events are transformed into a stable analytical fact table.

Core Question

How would you design a **historical, append-only data model** that:

1. Correctly computes GMV using only released (paid) transactions
2. Handles late arriving and reprocessed events
3. Preserves past analytical results (immutability)
4. Supports “as of” queries (e.g., GMV of Jan/2023 as seen on Mar/31 vs today)
5. Allows easy access to:
    - Current valid records
    - Historical versions
    - Daily lineage and reconciliation
6. Is simple to query by non-expert SQL users (no joins needed)
7. Is partitioned by `transaction_date`
8. Is updated in D-1 batches
9. Guarantees that reprocessing does not rewrite historical truth

Expected Deliverables

1. **DDL of the final analytical table** - Explain the grain, partitioning, immutability strategy, how current vs historical data is identified, how late events are incorporated without mutating the past.
2. **ETL/ELT logic** - Provide a reproducible pipeline (preferably in Python, Spark, or Scala).
3. **Example output** - Sample rows of the final table (mocked).
4. **Analytical layer** - Provide a SQL query under the final dataset that pull the daily GMV by subsidiary
5. **Architecture explanation** - Describe the technical stack and design choices.

Evaluation Criteria

- Modeling depth
- Causal & temporal reasoning
- Analytical usability
- Engineering maturity
- Communication

Optional advanced topics (bonus)

You may **optionally** discuss:

- How this model would evolve to real-time (Lambda / Kappa / Streaming Lakehouse)
- How to expose this as a semantic metric layer.
- How to reconcile GMV with Finance (double entry, revenue recognition, etc.).
- How to support backdated corrections without rewriting partitions.

Sending your solution

Please, be aware of the deadline and send all your materials (i.e.documentation, codes, files, recording if wanted, etc.) to pollyanna.goncalves@teachable.com by Jan 18 up to 23:59.