Data Curation Stage – The Silver Layer

The need for curating raw data

Data in the bronze layer is raw by nature in that it gets collected from several distinct and diverse data sources. Due to the diverse sources, it is natural for data to be delivered in unstandardized, invalid, inconsistent, non-uniform, duplicate, or insecure forms. In some other cases, raw data may have PII data in clear text, which should be properly masked before analytical consumption.

Unstandardized data

Data is collected using online transaction processing (OLTP) applications.

Problems:-

Web applications and mobile applications created in different countries, follow varying standards. During user input, some applications may accept first and last names as two separate fields, whereas other applications may have just one field for both.

In the US, it is common to follow the MM/ DD/YYYY format for dates, whereas many other countries prefer to use DD/MM/YYYY

order_date
05/12/2017
08/14/2020
02/26/2021
12/21/2019
07/12/2019
06/29/2019
06/12/2020

Store_orders data in the MM/DD/YYYY date format

order_date
18/05/2021
18/05/2021
18/05/2021

E-commerce transaction data in the DD/MM/YYYY date format

Before making this data available to end users, it needs to be properly standardized as per agreed-upon organizational policies.

Invalid data

Invalid data does not conform to constraints. These constraints can be of several types:

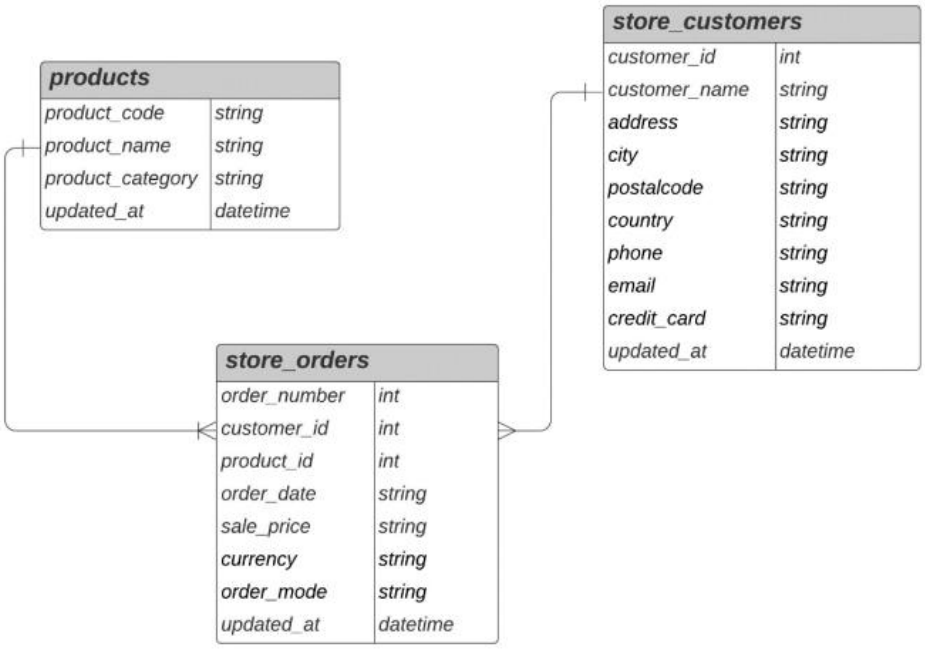
• Types: Columns should be of a specific data type, such as numeric, date, or floats.

• Ranges: The data in a column needs to fall within a range of values.

• Uniqueness: The data in the column should have unique values.

• Relational: The column should not be an orphan – the parent should mandatorily exist in the parent table.

Here is the entity-relationship diagram (ERD) for the Electroniz stores database



- Entity-relationship diagram of the stores database

Unfortunately, a careful study of the ERD will reveal that this dataset is short on many levels. Several problems need to be highlighted with this dataset:

• The products table has a product_code column that is the parent column for the product_id column in the store_orders table. As per best practices, the name of the columns in the parent and child tables should be the same.

• Notice that the products.product_code column is of the string data type, yet it is supposed to join with store_orders.product_id, which is of an integer type. The need for curating raw data

• After careful review of the data, it was discovered that the following two orders reference product_id rows that do not exist in the products table. In other words, they are orphan rows.

order_number	customer_id	product_id
1001	2840	20
1002	907	13

Fig: Orphan rows in store_order

Non-uniform data

Non-uniform data exists in varying forms across different datasets.

a couple of instances of non-uniform data in the bronze layer:

• In the Electroniz stores database, the country column exists in the store_customers table as follows:

country
United States
United Kingdom

Fig: Country data in store_customers

Whereas it exists like this in the e-commerce transactions:

country
USA
UK

Fig: Country data in e-commerce transactions

Before making this data available to end users, we should make it uniform across all data consumption layers.

Inconsistent data

Inconsistent data means having different values of the same data in two separate datasets.

As an example, a customer in the Electroniz stores database lives in a city named Crewe but whensame customer bought products on the e-commerce store claimed to live in a different city named Milton Keynes.

Before making this data available to end users, we should make it consistent across all data consumption layers.

Duplicate data

In the fast-changing world, data evolves all the time, from customers changing demographic data such as addresses and phone numbers to orders being bought and returned. Capturing and processing changed data is also referred to as change data capture (CDC) and has been an ongoing challenge in the data lake world

Insecure data

Lakehouse must abide by PCI regulations, so all PII data needs to be properly masked and encrypted.

Before making it available to end users, insecure data needs to be properly masked across all data consumption layers

The process of curating raw data

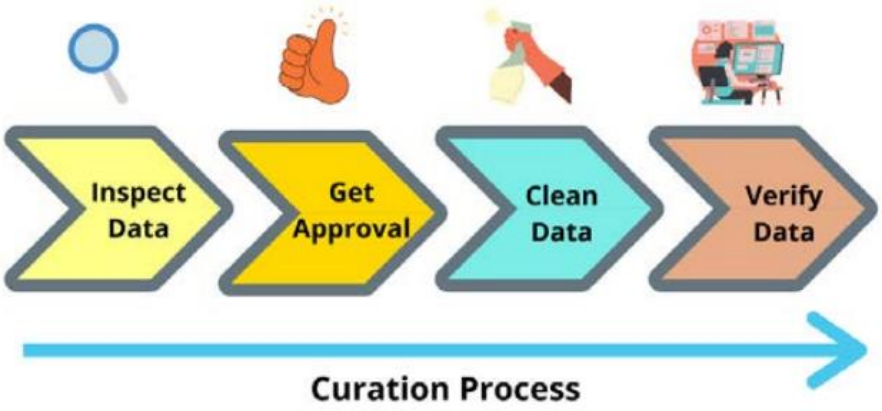


Fig: Data curation process

Inspecting data

It can start by visually inspecting data covering diverse data sources, although in many cases it may need to implement programming logic to discover data that is unstandardized, invalid, inconsistent, non-uniform, duplicate, or insecure.

Deliverable: A detailed report listing all the instances where data curation will be required, including a plan to fix each case.

Can include the pseudocode for the business logic that addresses the specific case.

Getting approval

The report created during the inspection phase should be formally submitted to the customer team for review and approval. This ensures that the customer is on board with the business logic that will be implemented to curate the data.

Deliverable: Formal approval by the customer team.

Cleaning data

Once the approval has been received, it is time to start the actual implementation of the business logic that implements data curation. a variety of frameworks, languages, and tools that can be used, although it is

better to use something that the customer is comfortable with. Generally, the two most frequently used ones are Apache Spark and SQL. Once the curation logic has been developed, it is integrated within the curation pipeline.

Deliverable: Logic that implements the curation.

Verifying data

The final step of data curation is to verify that the curation logic worked as intended. Typically, this activity should be performed by customer team members because they are closer to the data and therefore more likely to understand it better than the data engineer.

Deliverable: Formal approval by the Customer Team