Retail Data Quality Analysis – Case Study

Mikolaj Burzykowski – Data Specialist
Full repository: github.com/mikolajburzykowski/Retail_Data_Quality_Project

1. Project Overview

This case study presents an end-to-end retail data quality analysis project.
The goal was to identify data issues in a sample product dataset (e.g., missing attributes, inconsistent values, pricing anomalies) and propose improvements that would increase data reliability for reporting and business decision-making.

2. Dataset & Tools

Dataset

- Sample retail product and sales data (CSV files)
- Product hierarchy, categories, prices, dates and transactions

Tools & Technologies

- SQL – joins, aggregations, validation checks
- Excel / Power BI – exploratory analysis and visualization

Focus areas

- Data quality
- Consistency
- Readiness for reporting

3. Approach

1. **Understanding the business context** – assumed a typical retail scenario where incorrect product master data and prices lead to wrong reports and margin calculations.

2. **Loading and exploring the data** – imported the CSV files into a relational database, checked column types, unique keys and basic statistics.
3. **Data quality checks in SQL** – wrote queries to detect missing values, duplicates, inconsistent hierarchies and suspicious prices.
4. **Quantifying the issues** – measured how many records were affected by each type of problem.
5. **Proposing improvements** – prepared recommendations on validation rules and process changes that would prevent similar issues in a production environment.

4. Example SQL Checks

Below are examples of the SQL checks used in the project.

4.1 Error Rate Analysis

```
SELECT

    SUM(CASE WHEN AnyError = 'ERROR' THEN 1 ELSE 0 END) AS ErrorCount,

    COUNT(*) AS TotalRows,

    ROUND(100.0 * SUM(CASE WHEN AnyError = 'ERROR' THEN 1 ELSE 0 END) / COUNT(*),
2) AS ErrorRatePercent

FROM CleanedData;
```

This query calculates the total number of incorrect records and the percentage of errors in the dataset.
This metric allows quantifying the impact of bad data on reporting accuracy.

4.2 Frequent Error Categories

```
SELECT ErrorType, COUNT(*) AS Total

FROM checks

GROUP BY ErrorType

ORDER BY Total DESC;
```

Identifies dominant error categories across the dataset (e.g. missing values, invalid codes, mismatched IDs).
Helps prioritize areas requiring cleanup or rule enforcement.

4.3 Consolidated Error Report Table

```
CREATE TABLE ErrorReport AS

SELECT

c.PropertyID,

c.Country,

c.SourceSystem,

ck.ErrorType,

c.UpdatedDate

FROM CleanedData c

JOIN checks ck ON c.PropertyID = ck.PropertyID

WHERE c.AnyError = 'ERROR';
```

Produces a consolidated view of all problematic records.
Useful for communication with stakeholders, ticket creation and root-cause investigation.

4.4 Dataset Consistency Validation

```
DROP TABLE IF EXISTS MissingInChecks;

CREATE TABLE MissingInChecks AS

SELECT c.*

FROM CleanedData c

LEFT JOIN checks ck ON c.PropertyID = ck.PropertyID

WHERE ck.PropertyID IS NULL;
```

Detects records that exist in one dataset but not in the reference dataset.
This type of validation is crucial for identifying synchronization issues between systems.
MissingInChecks and MissingInCleanedData tables were created to validate consistency between the two datasets.

5. Summary & Business Impact

The analysis revealed key data quality issues that can affect reporting accuracy and system synchronization. Implementing validation rules and periodic monitoring would reduce correction effort and improve master data reliability.