Defend against known issues

In this reading, you'll learn about a defensive check applied to a data pipeline. **Defensive checks** help you prevent problems in your data pipeline. They are similar to performance checks but focus on other kinds of problems. The following scenario will provide an example of how you can implement different kinds of defensive checks on a data pipeline.

Scenario

Arsha, a Business Intelligence Analyst at a telecommunications company, built a data pipeline that merges data from six sources into a single database. While building her pipeline, she incorporated several defensive checks that ensured that the data was moved and transformed properly.

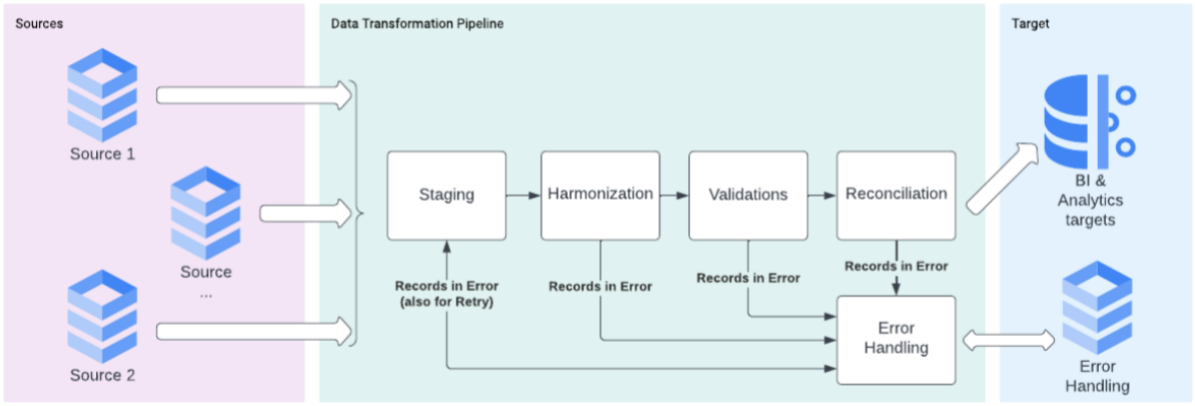
Her data pipeline used the following source systems:

1. Customer details
2. Mobile contracts
3. Internet and cable contracts
4. Device tracking and enablement
5. Billing
6. Accounting

All of these datasets had to be harmonized and merged into one target system for business intelligence analytics. This process required several layers of data harmonization, validation, reconciliation, and error handling.

Pipeline layers

Pipelines can have many different stages of processing. These stages, or **layers**, help ensure that the data is collected, aggregated, transformed, and staged in the most effective and efficient way. For example, it's important to make sure you have all the data you need in one place before you start cleaning it to ensure that you don't miss anything. There are usually four layers to this process: staging, harmonization, validation, and reconciliation. After these four layers, the data is brought into its target database and an error handling report summarizes each step of the process.



Staging layer

First, the original data is brought from the source systems and stored in the **staging layer**. In this layer, Arsha ran the following defensive checks:

- Compared the number of records received and stored
- Compared rows to identify if extra records were created or records were lost
- Checked important fields, such as amounts, dates, and IDs

Arsha moved the mismatched records to the error handling report. She included each unconverted source record, the date and time of its first processing, its last retry date and time, the layer where the error happened, and a message describing the error. By collecting these records, Arsha was able to find and fix the origin of the problems. She marked all of the records that moved to the next layer as “processed.”

Harmonization layer

The **harmonization layer** is where data normalization routines and record enrichment are performed. This ensures that data formatting is consistent across all the sources. To harmonize the data, Arsha ran the following defensive checks:

- Standardized the date format
- Standardized the currency
- Standardized uppercase and lowercase stylization
- Formatted IDs with leading zeros
- Split date values to store the year, month, and day in separate columns
- Applied conversion and priority rules from the source systems

When a record couldn’t be harmonized, she moved it to Error Handling. She marked all of the records that moved to the next layer as “processed.”

Validations layer

The **validations layer** is where business rules are validated. As a reminder, a **business rule** is a statement that creates a restriction on specific parts of a database. These rules are developed according to the way an organization uses data. Arsha ran the following defensive checks:

- Ensured that values in the “department” column were not null, since “department” is a crucial dimension
- Ensured that values in the “service type” column were within the authorized values to be processed
- Ensured that each billing record corresponded to a valid processed contract

Again, when a record couldn’t be harmonized, she moved it to error handling. She marked all the records that moved to the next layer as “processed.”

Reconciliation layer

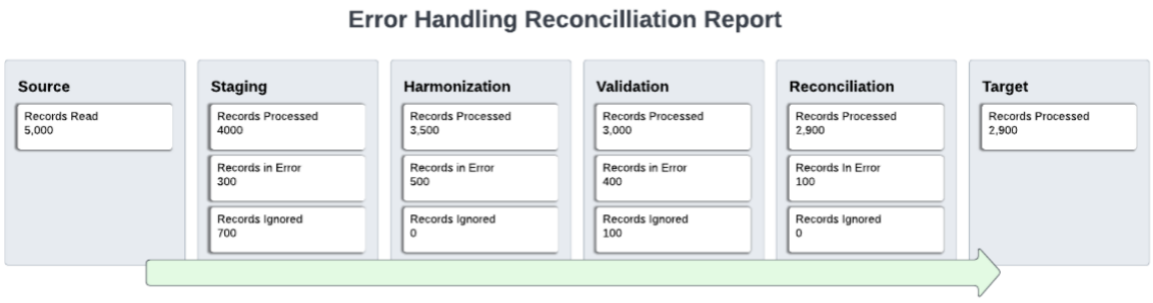
The **reconciliation layer** is where duplicate or illegitimate records are found. Here, Arsha ran defensive checks to find the following types of records:

- Slow-changing dimensions
- Historic records
- Aggregations

As with the previous layers, Arsha moved the records that didn't pass the reconciliation rules to Error Handling. After this round of defensive checks, she brought the processed records into the BI and Analytics database (OLAP).

Error handling reporting and analysis

After completing the pipeline and running the defensive checks, Arsha made an error handling report to summarize the process. The report listed the number of records from the source systems, as well as how many records were marked as errors or ignored in each layer. The end of the report listed the final number of processed records.



Key takeaways

Defensive checks are what ensure that a data pipeline properly handles its data. Defensive checks are an essential part of preserving data integrity. Once the staging, harmonization, validations, and reconciliation layers have been checked, the data brought into the target database is ready to be used in a visualization.