Basic data validation

Data validation refers to **the process of ensuring the accuracy and quality of data**. It is implemented by building several checks into a system or report to ensure the logical consistency of input and stored data. In automated systems, data is entered with minimal or no human supervision.

While data validation is a critical step in any data workflow, it's often skipped over. It may seem as if data validation is a step that slows down your pace of work, however, it is essential because it will help you create the best results possible. These days data validation can be a much quicker process than you might've thought. With data integration platforms that can incorporate and automate validation processes, validation can be treated as an essential ingredient to your workflow rather than an additional step.

Why Validate?

Validating the accuracy, clarity, and details of data is necessary to mitigate any project defects. Without validating data, you run the risk of basing decisions on data with imperfections that are not accurately representative of the situation at hand.

While verifying data inputs and values is important, it is also necessary to validate the data model itself. If the data model is not structured or built correctly, you will run into issues when trying to use data files in various applications and software.

Both the structure and content of data files will dictate what exactly you can do with data. Using validation rules to cleanse data before use helps to mitigate "garbage in = garbage out" scenarios. Ensuring the integrity of data helps to ensure the legitimacy of your conclusions.

Types of Data Validation

Validation Rules for Consistency

The most straightforward (and arguably the most essential) rules used in data validation are rules that ensure data integrity. You're probably familiar with these types of practices. Spell check? Data validation. Minimum password length? Data validation.

Every organization will have its own unique rules for how data should be stored and maintained. Setting basic data validation rules will help your company uphold organized standards that will effectively make working with data more efficient. Some other common examples of data validation rules that help maintain integrity and clarity include:

- Data type (ex. integer, float, string)
- Range (ex. A number between 35-40)
- Uniqueness (ex. Postal code)
- Consistent expressions (ex. Using one of St., Str, Street)
- No null values

Format Standards

Validating the structure of data is just as important as validating the data itself. Doing so will ensure that you are using the appropriate data model for the formats that are compatible with the applications you would like to use data in.

File formats and their standards are maintained by non-profit organizations, government departments, industry advisory panels, and private companies. With their assistance, they help to continuously develop, document, and define file structures that hold data.

When validating data, the standards and structure of the data model that the dataset is stored in should be well understood. Failing to do so may result in files that are incompatible with applications and other datasets with which you may want to integrate that data.

How to Perform Data Validation

Validation by Scripts

Depending on your fluency in coding languages, writing a script may be an option for validating data. You can compare data values and structure against your defined rules to verify that all the necessary information is within the required quality parameters. Depending on the complexity and size of the data set you are validating, this method of data validation can be quite time-consuming.

Validation by Programs

Many software programs can be used to perform data validation for you. This method of validation is very straightforward since these programs have been developed to understand your rules and the file structures you are working with. The ideal tool is one that lets you build validation into every step of your workflow, without requiring an in-depth understanding of the underlying format.