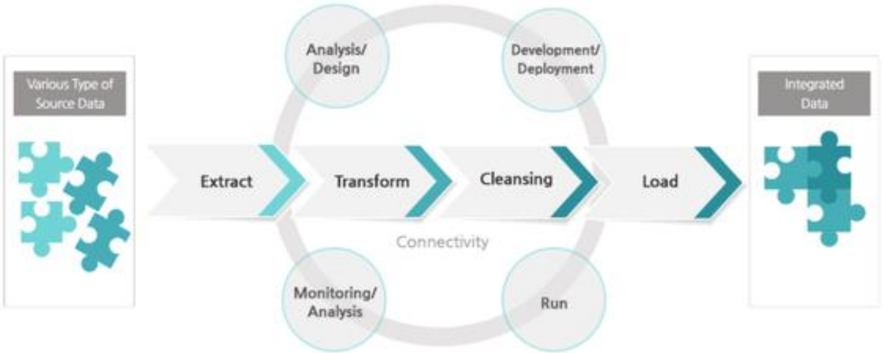
P4 – Data Integration

Integrating disparate systems is one of the most important things that needs to be done in a nontrivial application. This aspect of system-building is often overlooked by vendors who claim that their product can satisfy all your needs. In this final practice, we will examine the issues around integrating multiple different data systems, potentially with different data models and optimized for different access patterns, into one coherent application architecture.



Data Integration pipeline

The goal of data integration is to make sure that data ends up in the right form in all the right places. Doing so requires consuming inputs, transforming, joining, filtering, aggregating, training models, evaluating, and eventually writing the appropriate outputs. Batch and stream processors are the tools for achieving this goal. The outputs of batch and stream processes are derived datasets such as search indexes, materialized views, recommendations to show to users, aggregate metrics, and so on.

The range of different things you might want to do with data is dizzyingly wide. What one person considers to be an obscure and pointless feature may well be a central requirement for someone else. The need for data integration often only becomes apparent if you zoom out and consider the dataflows across an entire organization.

Derived Data in the Data Model

A system of record, also known as **source of truth**, holds the authoritative version of your data. Each fact is represented exactly once (the representation is typically normalized). Data in a **derived system** is the result of taking some existing data from another system and transforming or processing it in some way. Technically speaking, derived data is redundant, in the sense that it duplicates existing information. However, it is often essential for getting good performance on read queries.

Always make a clear distinction between systems of record and derived data in your architecture, it's a very helpful distinction to make, because it clarifies the dataflow through your system: it makes explicit which parts of the system have which inputs and which outputs, and how they depend on each other. The distinction between system of record and derived data system depends not on the tool, but on how you use them in your application.

Hands-on Data Integration

Our goal is to build an organized and structured Data Integration pipeline that will result in a final report. The report will be delivered in the format of a .pdf document generated from a jupyter notebook. It will concatenate code and text explaining the use case and the tasks executed to complete the pipeline in detail. It is highly recommended to structure the final report with this index.

1) Introduction to the problem or situation of analysis

   What is the intention of this analysis?

2) Introduction to the source of truth

   List the data sources (minimum 2 different sources) that you explored and justify why you decided to integrate them to solve the problem. Describe their schema and how they relate to each other. Evaluate their Data Quality: design two rules and specify which DQ dimensions they would improve.

3) List of database systems or file systems used to store the datasets for the analysis

   List the advantages of using the selected storage system during the implementation of the report. Justify how would you design the storage layer in a Production environment and how would your systems integrate. An architecture diagram might be useful.

4) Explanation of the derived data structures you expect to generate

   Include your thoughts and intentions. Explain why you decided to organize the information in that way.

5) Design of the strategy of transformations to perform in the Analytics environment (Databricks) to generate the new data structures

   This is how you are going to make the transformations, the logical transformation over each data field. Derivate, calculated, or aggregated information.

6) Implementation of the transformations with Spark

   Indicate the enrichment we obtain with the transformations. Explain why the data is more valuable this way in the output data structures.

7) Validation of the output data structures

   Evaluate Data Quality in the final data structures. Which DQ dimensions are better now that at the beginning of the ETL? Implement the two rules design in point 2 and profile the data in source and target to justify the answers.

Most of the previous sections are generic actions that are usually executed in a Data Integration pipeline. Depending on the use case, case some might be more difficult than others. Feel free to enrich them showcasing those you consider most valuable in your pipeline.

Hands on Data Analytics

In addition to the Data Integration report, we will attach a presentation deck containing the analysis. This deck will be used to communicate the story of the analysis to the audience in class. Please, consider the following:

a) The message must be clear and simple. The story must show the audience the goal of the analysis in the first place.
b) Share the insights backed with the generated information. Justify the strategy in your investigation. Remember that the important stuff here is the insight, not how you got it.
c) Finish the story with three to five conclusions and a call to action.