Project: Data Warehouse with PostgreSQL

Table of Contents

Preparing the environment

This project uses Linux on Pop_OS!, with PostgreSQL as the database management system for the data warehouse.

Before running the data warehouse scripts, ensure that you have PostgreSQL installed and running on your system. You will also need to have access to the PostgreSQL user with sufficient privileges to create databases and tables. Currently, the scripts assume that the PostgreSQL user is `postgres`.

To install PostgreSQL:

```
sudo apt install postgresql
```

By default, the service will start automatically after installation. If it does not, you can start it manually using the following command:

```
sudo service postgresql start
```

Documentation

The project documentation is written in AsciiDoc. If you want to modify and rebuild the documentation, you will need Asciidoctor. The method I used to install asciidoctor was to install the latest Ruby gems:

```
sudo apt install ruby
sudo gem install asciidoctor asciidoctor-diagram asciidoctor-pdf
```

Building the documentation

To build the documentation, navigate to the `src/docsrc` directory and run:

```
docbuild.sh all
```

Diagram support

For diagrams, some additional packages are required.

```
sudo apt install graphviz plantuml
```

For mermaid, rather than use `mermaid-cli` from the package manager, I wanted the latest version of mermaid-cli via npm (after installing nvm for the latest version of nodejs).

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.5/install.sh | bash
source ~/.bashrc || source ~/.zshrc
nvm install --lts

npm install -g @mermaid-js/mermaid-cli
```

OCR image text extraction support

For OCR image text extraction, Tesseract and the Python Pillow and pytesseract libraries are required.

```
sudo apt install tesseract-ocr
pip install Pillow pytesseract
```

Preparing the specifications table

I decided to not make the extraction part of the pipeline and just prep the source beforehand. To

recreate the table preparation process:

1. Extract the text:

```
pushd src/docsrc/img-extract
python ocr-extract.py > ocr-output.txt
popd
```

2. ……*x*……
3. Okay, nevermind. ChatGPT did it all for me. I uploaded the image and gave it this prompt:

```
I will upload an image that is basically a fancy table.
Extract the text from the image and convert it to a table in AsciiDoc.
Let me know when you're ready.
```

But the OCR was a good exercise anyway.

Preparing the ETL mindmap

Let's try ChatGPT again.

Well done, ChatGPT! Here's the prompt and you can find the source image in this repository under `src/docsrc/img-extract/etl-map.png`.

```
New task: Now I'm going to upload an image of a fancy mindmap diagram.
I'd like you to extract the information from the image and recreate it
using mermaid syntax. Let me know when you're ready.
```

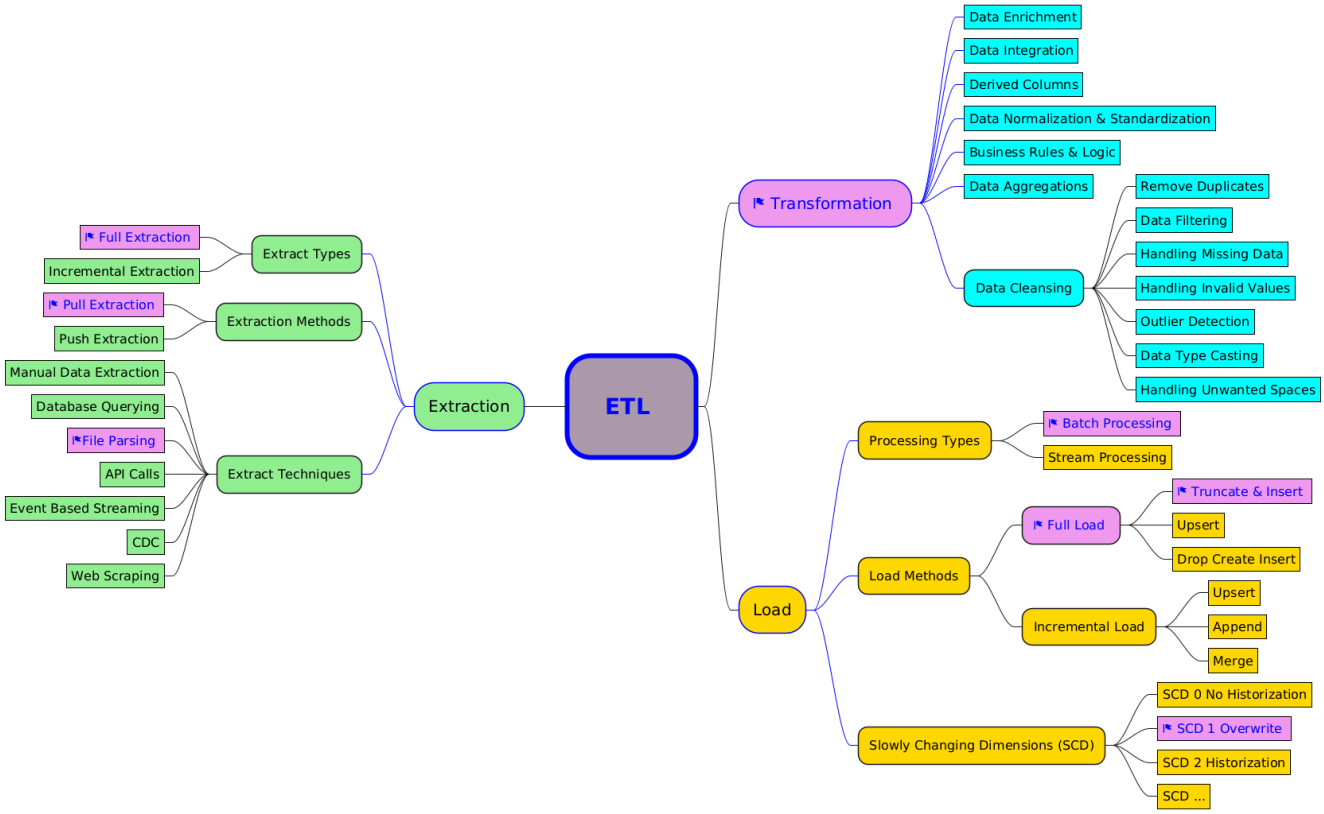
I'll clean it up with icons and shapes myself. Can't let AI rob me of all the fun!

(I later asked ChatGPT to convert to a Drawio file instead, but it had an error about IDs. After debugging with `xmllint` and fixing the IDs, I got a valid Drawio file, but it was only two node levels deep and didn't follow the style anyway. So, I'm sticking with mermaid for now.)

Data architecture

Before defining the data flow, it's essential to establish clear specifications for each layer in the data pipeline. This ensures that each layer serves its intended purpose and meets the requirements of the overall data strategy.

ETL task breakdown

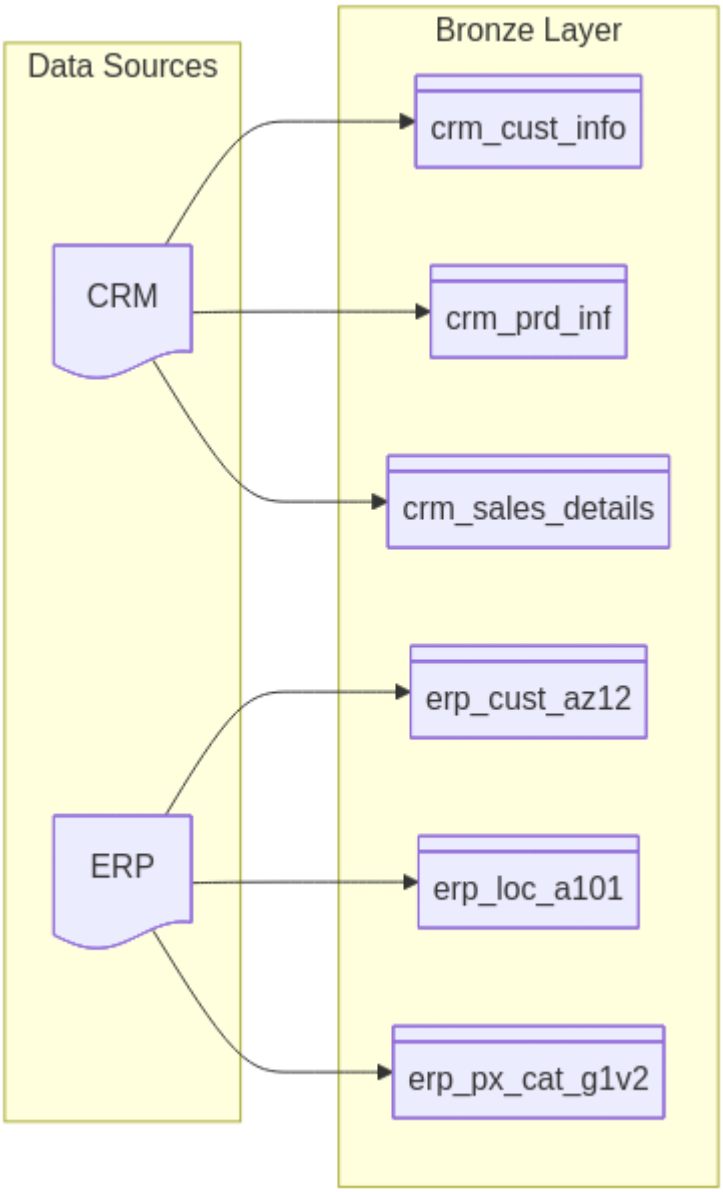


Specifications

Given the chosen tasks in the ETL process, we have the following specifications.

Category	Bronze Layer	Silver Layer	Gold Layer
Definition	Raw, unprocessed data as-is from sources	Clean & standardized data	Business-Ready data
Objective	Traceability & Debugging	(Intermediate Layer) Prepare Data for Analysis	Provide data to be consumed for reporting & Analytics
Object Type	Tables	Tables	Views
Load Method	Full Load (Truncate & Insert)	Full Load (Truncate & Insert)	None
Data Transformation	None (as-is)	Data Cleaning Data Standardization Data Normalization Derived Columns Data Enrichment	Data Integration Data Aggregation Business Logic & Rules
Data Modeling	None (as-is)	None (as-is)	Star Schema Aggregated Objects Flat Tables
Target Audience	Data Engineers	Data Analysts Data Engineers	Data Analysts Business Users

Data Flow



Bronze layer

The Bronze layer is the initial storage area for raw data ingested from various sources. This layer is designed to store data in its original format, preserving its integrity and providing a foundation for further processing and transformation.

Silver layer

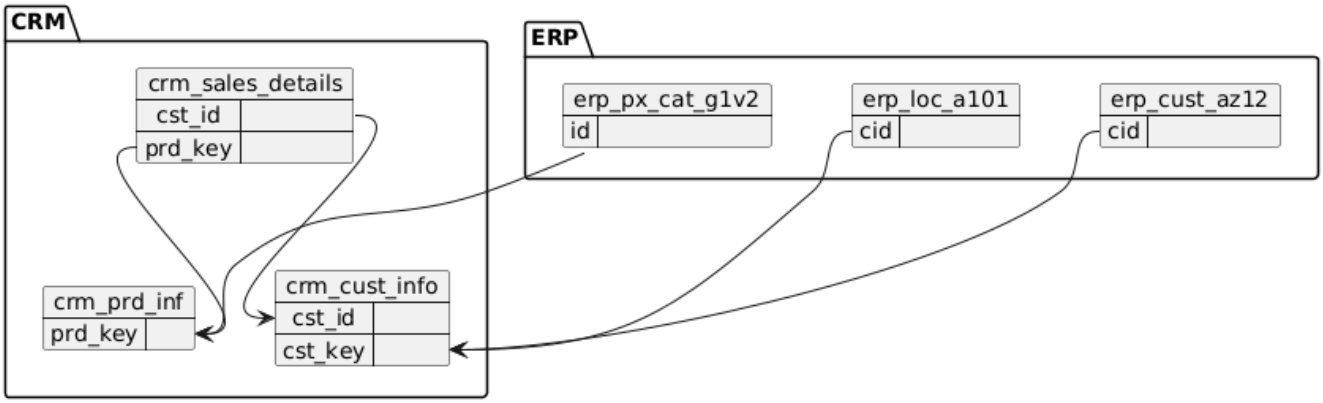
The Silver layer serves as the intermediate processing stage in the data pipeline. In this layer, raw data from the Bronze layer is cleaned, transformed, and enriched to enhance its quality and usability for analysis.

The key functions of the Silver layer include:

- Data Cleaning: Removing duplicates, handling missing values, and correcting inconsistencies in the data.
- Data Transformation: Converting data into a more structured format, applying business rules, and aggregating information as needed.
- Data Enrichment: Integrating additional data sources to provide more context and insights.

By implementing these processes in the Silver layer, we ensure that the data is reliable and ready for advanced analytics and reporting in the subsequent Gold layer.

Data integration analysis and sketch



Gold layer

The Gold layer represents the final stage in the data pipeline, where data is fully refined and optimized for business intelligence and analytics. In this layer, data from the Silver layer is aggregated, summarized, and structured to meet specific reporting and analytical needs.

The key functions of the Gold layer include:

- Data Aggregation: Summarizing data to provide high-level insights, such as totals, averages, and other key performance indicators (KPIs).
- Data Structuring: Organizing data into star or snowflake schemas to facilitate efficient querying and reporting.
- Data Optimization: Enhancing performance through indexing, partitioning, and other techniques to ensure fast access to critical information.

By implementing these processes in the Gold layer, we ensure that the data is not only accurate and reliable but also tailored to support strategic decision-making and advanced analytics across the organization.