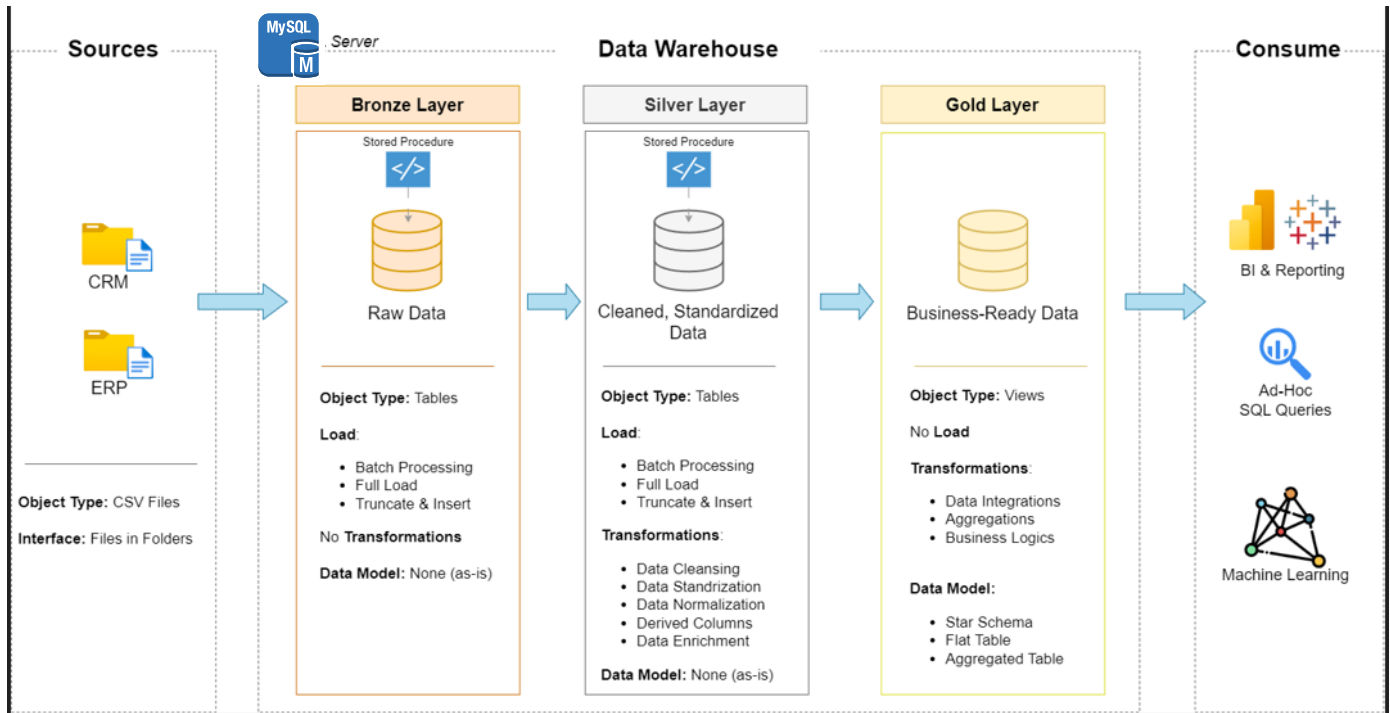
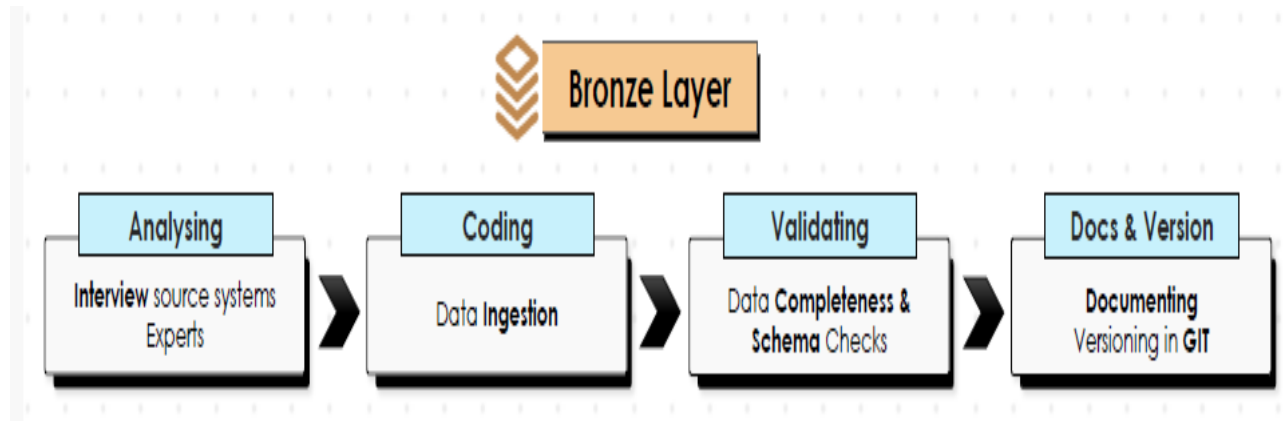


MEDALLION ARCHITECTURE

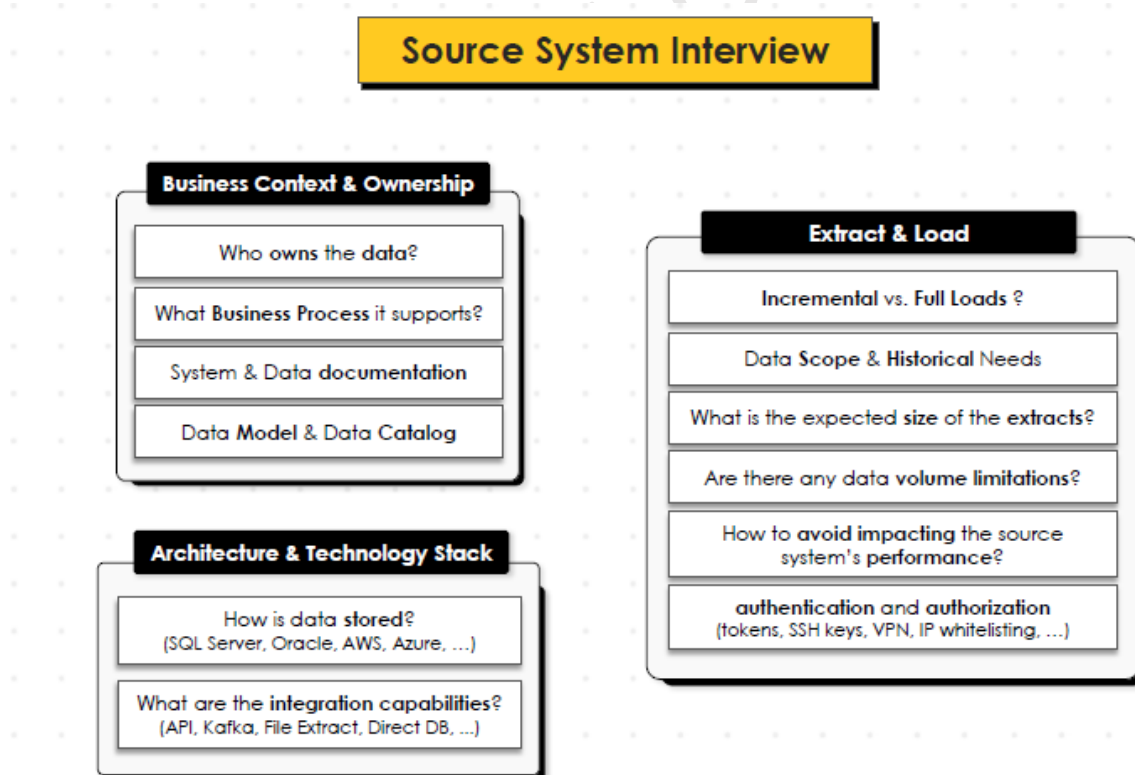


	Bronze Layer	Silver Layer	Gold Layer
Definition	Raw, unprocessed data as-is from sources	Clean & standardized data	Business-Ready data
Objective	Traceability & Debugging	(Intermediate Layer) Prepare Data for Analysis	Provide data to be consumed for reporting & Analytics
Object Type	Tables	Tables	Views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

Bronze Layer



Step 1 – Analysing:



Step 2: Coding

This involves creating DDL for the Bronze Layer and the Load method.

1. DDL For Bronze Layer DB:

Good practices while writing a DDL are as follows:

- Always **write comments** describing the purpose of the code/script and warnings to keep in mind before running, if any.
- Standard practice is to drop the table if it already exists, but if there is data in an already available table, we have to make sure that it is ok to drop it.
- One DDL script to create all tables at a go is a common practice for better performance.

Please refer to the DDL_Bronze.sql file available in the resources in sql session folder. (SQL Sessions folder is made available in the mom analysts hub recordings folder.

2. Load Method: Full Load – Truncate & Insert:

In MySQL, Bulk Insert is not supported, but we do have a concept called Load Data Infile.

This script enables us to load the raw data directly from the source files in seconds.

The script has 7 steps:

1. **Disable Strict Mode:**

- This is done to allow the script to accept blanks, enabling us to create an as-is copy of the source.
- SET @OLD_SQL_MODE = @@sql_mode;
SET SESSION sql_mode = REPLACE(@@sql_mode, 'STRICT_TRANS_TABLES', '');

2. **Disable Keys:**

- If a table has indexes (primary keys, unique, or secondary indexes), MySQL normally updates them **for every row inserted**.
- This slows down bulk insert operations.
- ALTER TABLE bronze_crm_cust_info DISABLE KEYS;

3. **Truncate table:**

- Ensures the table is empty before loading.
- Prevents duplicate data.
- Truncate is faster than DELETE because it resets the storage structure and auto-increment counters.
- TRUNCATE TABLE bronze_crm_cust_info;

4. Load Data from CSV:

- This is the **main bulk load**.
- FIELDS TERMINATED BY ',' → CSV separator.
- OPTIONALLY ENCLOSED BY '"' → handles quotes in text fields.
- LINES TERMINATED BY '\n' → newline per row.
- IGNORE 1 LINES → skips header row.
- Fastest method in MySQL to load large CSVs directly into tables.
- The file location is a default location that is allowed in all the MySQL servers. Please **place the data sets in the given location before running the script**. (Some systems allow location of choice, but some don't. Feel free to try it out.)
- ```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/cust_info.csv'
INTO TABLE bronze_crm_cust_info
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

#### 5. Enable Keys:

- Rebuilds the indexes after loading.
- Makes queries fast for future operations.
- **ALTER TABLE bronze\_crm\_cust\_info ENABLE KEYS;**

#### 6. Repeat for All Six tables:

- Same pattern for all tables ensures consistency.
- Optimized for performance (truncation + disable keys + bulk load).

#### 7. Enable Strict Mode:

- Returns SQL mode to original state.
- Ensures future queries respect strict mode.
- **SET SESSION sql\_mode = @OLD\_SQL\_MODE;**

Understand the logic and try to build it, and later verify with the file added in the resources.

## Step 3: Validating:

The next step is to check for data completeness and schema checks.

Check whether the data is loaded as-is. You can compare the no of rows in the CSV file and the DB Table.

Since this layer is all about having the raw, unstructured data as-is from the source, check if the bad data or null values, or blanks are also loaded into the DB table as it is from CSV files.

## Step 4: Documenting & Versioning in GIT hub:

Once the data is loaded and validation is over, upload the script files and data sets to your GitHub repository.

Mom Analysts hub