ETL Process Report

1. Introduction

The **ETL (Extract, Transform, Load)** process is a critical stage in data warehousing.
In this project, ETL was implemented to **generate, clean, transform, and load** synthetic retail sales data into a **SQLite-based Data Warehouse** for analytical purposes.

The ETL pipeline ensures that the data is:

- **Accurate** – Errors and invalid records are removed.
- **Consistent** – Data is formatted and typed correctly.
- **Analysis-ready** – Structured for OLAP and reporting.

2. ETL Process Overview

The ETL process is broken down into three main phases:

A. Extract

- **Objective:** Gather the raw data for processing.
- **Implementation:**
    - **Synthetic data generation** using the Faker library and NumPy.
    - 1000 sales records simulated for 100 unique customers.
    - Attributes include:
      CustomerID, Country, Product, Category, Quantity, UnitPrice, InvoiceDate.
    - Data saved to **retail_sales.csv** before transformation for backup.

1	CustomerID	Country	Product	Category	Quantity	UnitPrice	InvoiceDate
2	CUST052	Canada	Product_15	Home	8	61	2025-07-24 16:38:42.898746
3	CUST083	Japan	Product_11	Home	24	100	2023-10-20 16:38:42.898746
4	CUST003	Australia	Product_2	Toys	44	30	2024-07-24 16:38:42.898746
5	CUST064	France	Product_1	Toys	22	89	2024-01-31 16:38:42.898746

B. Transform

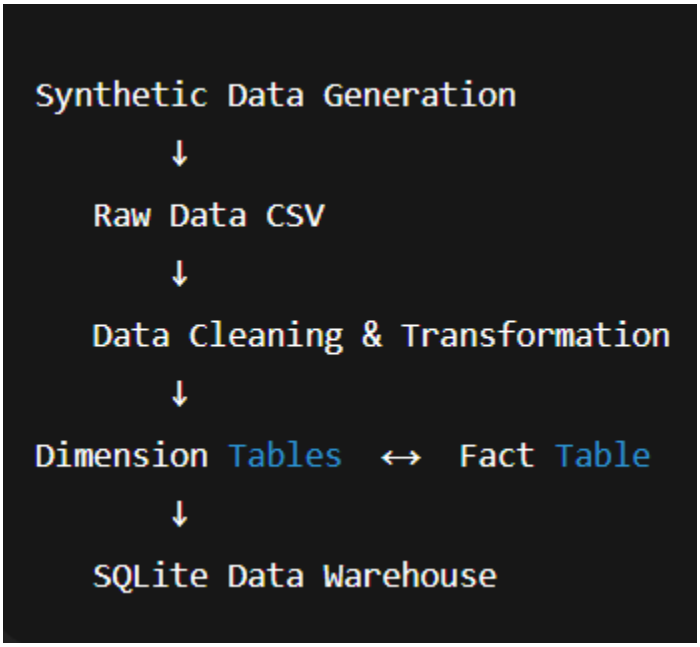
- **Objective:** Clean and prepare data for the warehouse.
- **Transformation Steps:**

- **Missing Value Removal** – Dropped any incomplete rows.
- **Data Type Enforcement** – Ensured `InvoiceDate` is a `datetime`.
- **Outlier Removal** – Filtered out rows where `Quantity <= 0` or `UnitPrice <= 0`.
- **New Feature Creation** – Computed `TotalSales = Quantity × UnitPrice`.
- **Time Filtering** – Selected only transactions from the last year
- **Dimension Table Creation:**
  - **Customer Dimension:** Aggregated purchases and orders per customer.
  - **Time Dimension:** Derived `Year`, `Quarter`, `Month`, and `Day` from `InvoiceDate`.

C. Load

- **Objective: Store the transformed data in the warehouse.**
- **Implementation:**
  - **Target: SQLite database (retail_dw.db).**
  - **Loaded:**
    - **CustomerDim – Customer attributes and purchase metrics.**
    - **TimeDim – Time hierarchy for OLAP analysis.**
    - **SalesFact – Transaction facts linked to dimensions.**

3. ETL Workflow Diagram



4. Key Features in the Code

1. **Logging:**
    - **Tracks each ETL stage:**

```
logging.info("Starting ETL process...")
logging.info(f"Generated {len(df)} rows.")
logging.info("ETL process completed successfully.")
```

2. **Reproducibility:**
    - **Random seeds (Faker.seed(42), np.random.seed(42)) ensure repeatable results.**
3. **Scalability:**
    - **Code can easily handle larger datasets by adjusting num_rows.**

5. Benefits of the ETL Process

- Synthetic data allows for safe testing without real customer information.
- Automated process – Entire ETL can be run with a single command.
- Warehouse-ready data – Clean, structured, and linked for OLAP.

6. Conclusion

This ETL pipeline successfully bridges the gap between raw synthetic data and structured warehouse-ready datasets.
It prepares a **Customer Dimension**, **Time Dimension**, and **Sales Fact Table** that integrate seamlessly into OLAP analysis for trend discovery and decision-making.