VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018

Project Phase-1 Synopsis
Report on

"E-commerce Data Order ETL pipeline" *Submitted in the partial fulfillment of the requirements for the award of*

BACHELOR OF ENGINEERING DEGREE
In
COMPUTER SCIENCE & ENGINEERING (ARTIFICIAL INTELLIGENCE &MACHINE LEARNING)

Submitted by

Name : Dhanush Gowda	**USN:4AD23CI011**
Name: S Dheepanjali	**USN:4AD23CI044**
Name: Thashwini H R	**USN:4AD23CI055**
Name:Thejashwini M	**USN:4AD23CI056**

Under the guidance of

Apoorva S M

Assistant Professor
Department of CSE(ARTIFICIAL INTELLIGENCE &MACHINE LEARNING)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (ARTIFICIAL INTELLIGENCE &MACHINE LEARNING)

ATME College of Engineering

ATME College of Engineering,

13th Kilometer, Mysore-Kanakapura-Bangalore Road
Mysore-570028

ABSTRACT

E-commerce platforms generate massive volumes of order data from multiple sources such as websites, mobile apps, payment gateways, and inventory systems. Managing and analyzing this data efficiently remains a challenge due to its volume, variety, and velocity. This project proposes an automated, scalable ETL (Extract, Transform, Load) pipeline to ingest, clean, transform, and load e-commerce order data from diverse sources into a central data warehouse. Using open-source orchestration tools like Apache Airflow, combined with Python-based data processing and a robust database system like PostgreSQL with time-series extensions, the system aims to provide clean, validated, and consistent order data for downstream analytics and machine learning applications. The solution supports batch and near real-time processing, ensuring timely availability of data for business insights, inventory forecasting, and customer analytics. The project focuses on modular design, scalability, and efficient automation to demonstrate industry-relevant data engineering skills.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 OVERVIEW

The exponential growth of e-commerce has created unprecedented data generation across multiple touchpoints including web platforms, mobile applications, payment gateways, inventory systems, and customer service channels. Modern online retailers process millions of transactions daily, generating structured and semi-structured data that requires systematic extraction, transformation, and loading for business intelligence and decision-making purposes.

Traditional data processing approaches struggle with the complexity, scale, and real-time demands of contemporary e-commerce environments. Manual data handling creates bottlenecks, introduces errors, and limits analytical capabilities. This project addresses these challenges by developing an automated ETL pipeline specifically designed for e-commerce order data processing using Apache Airflow, Python-based transformation modules, and PostgreSQL with TimescaleDB extensions.

1.2 Existing system

Current e-commerce data workflows rely on manual ETL scripts and legacy batch systems that run overnight or on-demand. Custom Python/SQL scripts necessitate manual execution and monitoring, causing human errors and processing bottlenecks during peak sales. Tools like SSIS or Talend offer limited API integration and real-time capabilities, while data remains fragmented across disparate platforms without unified schemas. Validation, error handling, and pipeline monitoring require significant human intervention, increasing operational overhead. These systems struggle to scale for flash sales or high traffic, resulting in delayed reporting and missed opportunities. Automated, end-to-end orchestration with robust monitoring is needed to overcome these constraints.

1.3 Drawbacks

- High latency and inefficient data processing during sales peaks
- Lack of automation leads to manual errors
- Difficulty integrating data from various sources (APIs, databases, files)
- Limited capability for near real-time insights and predictions

1.4 PROPOSED SYSTEM

The proposed solution automates the ETL pipeline to ingest order data from multiple sources such as e-commerce APIs, payment systems, and inventory databases. Using Python, Apache Airflow for orchestration, and PostgreSQL for storage, the system cleans, validates, and processes data before loading it into a centralized warehouse, enabling near real-time analytics and ML use cases.

1.5 WORKING

1. **Data Ingestion**

   - Connect to e-commerce APIs (Shopify, WooCommerce), payment gateways (Stripe, PayPal), and inventory databases.
   - Fetch order, customer, and product data via secure RESTful calls or CDC streams.
   - Implement retry logic and API rate–limit handling for robust extraction.

2. **Data Storage**

   - Store raw JSON/CSV payloads in a staging area on cloud storage (S3/GCS).
   - Archive raw data for audit and lineage tracking.

3. **Data Transformation**

   - Use Python modules to clean and standardize fields: remove duplicates, impute missing values, normalize date/time and currency formats.
   - Enrich data by joining customer and product metadata, calculating derived metrics (order value, customer lifetime value).
   - Apply business rules for fraud flagging and data validation checks.

4. **Data Loading**

- Load transformed records into a PostgreSQL data warehouse with TimescaleDB partitions for time-series performance.
- Maintain dimensional schema (facts and dimensions) optimized for analytics.

5. **Orchestration & Automation**

    a. Use Apache Airflow to schedule and manage the ETL workflow: data extraction, transformation, and loading tasks.
    b. Implement monitoring, logging, and alerting to detect failures or performance degradation.

6. **Output & Integration**

    a. Expose processed data via REST APIs and JDBC connectors.
    b. Feed downstream BI dashboards (Tableau, Power BI) and ML pipelines for forecasting and analytics.

Chapter 2

LITERATURE SURVEY

2.1 Survey Papers

1. The paper titled "**A Review on Real-Time Data Pipelines for E-Commerce Transactional Data Analytics**" by Mahesha K, Sagar BR, Tejonidhi M, Yashwanth N, and Ambika V (EPRA IJRD, August 2025) offers a comprehensive overview of technologies and methodologies for building scalable, low-latency data pipelines in e-commerce. It explores tools like Apache Kafka, Spark, Flink, and AutoML, emphasizing their roles in ingestion, stream processing, and analytics. The review highlights how real-time systems enhance fraud detection, dynamic pricing, and personalized recommendations. However, limitations persist in areas such as model interpretability, data privacy compliance (e.g., GDPR), and resource-intensive AutoML integration. The paper also notes challenges in handling data spikes during peak events and the need for ethical AI frameworks. Future directions include edge computing, federated learning, and quantum-based processing to improve responsiveness and trust.

2.The paper by Samyukta Rongala and Godavari Modalavalasa, titled "**Automating Extract, Transform, Load (ETL) Pipelines using Machine Learning Triggered Workflow Optimization**" (IJISAE, 2024), presents a robust machine learning-driven framework to automate ETL processes. It leverages anomaly detection, probabilistic imputation, and reinforcement learning to enhance data quality, reduce manual intervention, and improve scalability. The proposed system achieves a 95% anomaly detection rate, reduces data loss to 1%, and improves schema mapping accuracy to 98%, resulting in a 36.49% reduction in overall ETL time. However, limitations include potential vendor lock-in due to reliance on proprietary serverless platforms, challenges in schema matching across heterogeneous data sources, and limited generalizability of certain ML models to dynamic real-time data environments. While the framework demonstrates strong performance on large datasets, further research is needed to ensure adaptability across diverse domains and evolving data architectures.

3.The paper titled "**Developing an ETL Pipeline for Data Analysis**" by A. S. Prajwal Babu and Prof. Suma B (IJCATR, 2022) presents a practical framework for building scalable ETL pipelines using Node.js, Amazon S3, Athena, and PostgreSQL. Designed to simplify data extraction, transformation, and loading for analytics, the framework enables automation through cron scheduling and supports dashboard integration for real-time insights. It emphasizes modularity, ease of use, and minimal programming requirements, making it accessible to non-experts. However, the system faces limitations such as lack of automated retries for failed jobs, absence of cross-job dependencies, non-parsable logs that hinder debugging, and excessive Slack notifications that dilute alert effectiveness. These challenges highlight the need for enhanced monitoring, retry mechanisms, and better log management. Future improvements include Kubernetes integration, support for multi-language tasks, and cost tracking for operations.

4.The paper titled "**Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers**" by Harald Foidl, Valentina Golendukhina, Rudolf Ramler, and Michael Felderer (2024) presents a comprehensive taxonomy of 41 influencing factors (IFs) affecting the quality of data pipelines, grouped into five themes: data, development & deployment, infrastructure, life cycle management, and processing. Through a multivocal literature review and expert interviews, the authors identify key challenges such as incorrect data types, compatibility issues, and developer struggles with data ingestion and integration. Empirical studies on GitHub and Stack Overflow further reveal that most data-related issues occur during the cleaning and ingestion stages, while compatibility emerges as a distinct problem area. However, the study has limitations: it primarily focuses on tabular data, potentially biasing the taxonomy; it excludes managerial and business-related factors; and it does not explore interdependencies among the identified IFs. These constraints suggest that further research is needed to generalize findings across data types and domains.

5. The paper "**DOD-ETL: Distributed On-Demand ETL for Near Real-Time Business Intelligence**" by Gustavo V. Machado, Ítalo Cunha, Adriano C. M. Pereira, and Leonardo B. Oliveira introduces a modular, technology-independent framework that addresses the latency and scalability limitations of traditional batch ETL processes. DOD-ETL employs log-based change data capture (CDC) via its Change Tracker component, which continuously monitors source database transaction logs and emits fine-grained change events. These events are routed through partitioned message queues—enabling parallelism—and processed by the Stream Processor, which performs in-memory transformations, buffering for consistent joins, and loading into the target data warehouse. Key innovations include on-demand data stream pipelines, in-memory master data caching, and a unified programming model compatible with frameworks like Spark and Beam. Deployed in a large steelworks facility, DOD-ETL reduced ETL processing times from hours to under a minute and demonstrated up to tenfold performance improvements over conventional stream processing frameworks while maintaining resilience to node failures. However, its performance can degrade with complex data models (e.g., ISA-95), and users must manually configure transformation logic with Spark operators, which may limit accessibility. Additionally, cache reinitialization during failover and dependencies on specific tools (Kafka, Spark, H2) introduce integration challenges and overhead.

6. The article "**Spearheading Big Data Solutions: Optimizing Data Pipelines For Enhanced Efficiency And Performance**" by Kiran Polimetla and Farah Jenny (Educational Administration: Theory and Practice, Vol. 30, No. 6, 2024, pp. 4106–4116), the authors explore advanced strategies for enhancing big data pipeline performance, particularly in educational and scientific contexts. Kiran Polimetla focuses on harmonizing data workflows and leveraging tools like Google BigQuery and Configo to streamline processing of petascale datasets, while also proposing optimizations such as pod caching and parallelism policies. Farah Jenny emphasizes scalable infrastructure through microservices, real-time analytics, and robust ETL design, advocating for performance tuning and data quality checks. However, Polimetla's work is limited by its reliance on high-performance cloud platforms and a lack of comparative benchmarks for Configo, while Jenny's architectural proposals introduce orchestration complexity and require significant engineering effort for customization. Together, their contributions offer a forward-looking blueprint for big data efficiency, though practical implementation may vary across domains and resource availability.

7. In the paper "**Modelling Data Pipelines**" presented at the 2020 Euromicro Conference on Software Engineering and Advanced Applications, authors Aiswarya Raj, Jan Bosch, Helena Holmström Olsson, and Tian J. Wang propose a conceptual model for fault-tolerant, automated, and traceable data pipelines, based on case studies from the telecommunications, manufacturing, and automotive sectors. Raj and Bosch focus on the structural and operational aspects of pipeline modeling, emphasizing the need for standardized components like nodes and connectors to streamline data flow and enable monitoring, fault detection, and mitigation. Olsson contributes insights on agile methodology and organizational challenges, while Wang brings industry-specific validation from Ericsson. The model is praised for its potential to unify data practices across teams and domains, but limitations include its abstract nature, lack of detailed implementation strategies, and insufficient attention to reinforcement learning workflows. Additionally, while the model addresses many data management challenges, it cannot fully resolve issues like low-quality data generation or storage constraints.

8. In the article "**Incorporating Deep Learning Model Development With an End-to-End Data Pipeline**" published in IEEE Access (2024), Kaichong Zhang presents a comprehensive framework that integrates data engineering and deep learning workflows to support scalable, real-time model development. Using a case study from Sprocket Central Pty Ltd, Zhang outlines a four-part pipeline: database management (via SQL Server), business intelligence (using Power BI), model training (with PyTorch), and incremental learning for recommender systems. The framework emphasizes efficient ETL processes, star schema design for optimized querying, and robust model evaluation using metrics like ROC AUC and F1 score. However, limitations include the relatively small dataset used (20,000 transactions), which may not fully reflect challenges in large-scale deployments. Additionally, while the pipeline supports hybrid setups, its reliance on specific tools (e.g., SQL Server, Power BI) may constrain portability across platforms. The paper also notes that while incremental learning is implemented, catastrophic forgetting and memory optimization remain open challenges.

Chapter 3

PROBLEM STATEMENT

The project aims to develop a automated ETL data pipeline that effectively collects, cleans, and integrates order data from various e-commerce sources to provide timely, reliable, and clean data for analytics. The pipeline will handle batch and near real-time data processing, ensuring high data quality, scalability, and ease of integration with downstream analytics and business intelligence platforms.

Key features of the proposed solution:

- Automated, Modular ETL Pipeline: Seamlessly extracts, transforms, and loads e-commerce order data from multiple sources (APIs, payment gateways, inventory systems) with minimal manual intervention.
- Real-Time and Batch Processing Support: Handles both near real-time data streams and scheduled batch jobs, enabling timely and flexible data availability.
- Robust Data Cleaning & Validation: Performs deduplication, missing value imputation, type standardization, and business-rule enforcement to ensure high data quality.
- Advanced Orchestration: Utilizes Apache Airflow for scheduling, dependency management, automatic retries, and workflow monitoring.
- Scalable Storage: Loads processed data into a PostgreSQL warehouse with TimescaleDB extensions for efficient time-series querying and analytics.
- Seamless Integration: Exposes curated data through REST APIs, JDBC connectors, and is compatible with BI dashboards (Tableau, Power BI) and machine learning pipelines.
- Monitoring & Alerting: Implements logging, health checks, and automatic alerts to promptly detect failures or performance issues.
- Extensibility: Modular architecture allows easy adaptation for new data sources, business logic, and analytics requirements as the business evolves.

These features collectively deliver a reliable, scalable, and future-ready e-commerce data backbone for analytics and intelligent decision-making

Chapter 4

OBJECTIVES

- Automate data ingestion from diverse e-commerce sources (APIs, CSV files, databases) with built-in retry logic and rate-limit handling.

- Perform robust data cleaning and validation, including duplicate removal, missing-value imputation, normalization, and enforcement of business rules.

- Transform and unify data into a consistent schema, computing derived metrics (e.g., order value, customer lifetime value) for downstream analytics.

- Orchestrate ETL workflows using Apache Airflow, ensuring scheduled, monitored, and fault-tolerant execution.

- Store and expose processed data in a PostgreSQL warehouse with TimescaleDB partitions, providing REST/JDBC access for BI dashboards and ML pipelines

Chapter 5

METHODOLOGY

Data Collection:

Connect to diverse e-commerce sources—such as APIs (Shopify, WooCommerce, Stripe) or batch CSV/Excel files—to gather order, product, and customer data needed for business analytics.

Data Cleaning:

Apply rigorous cleaning processes by removing duplicate records, correcting inconsistencies, and handling missing values to ensure data integrity and accuracy.

Data Transformation:

Standardize data formats (e.g., timestamps, currencies), map values to unified schemas, and compute derived fields such as order totals or average cart size, preparing data for analysis.

Data Loading:

Insert the cleaned and transformed data into a PostgreSQL database, designing schemas that support efficient querying and reporting.

Pipeline Automation:

Utilize Apache Airflow to schedule, orchestrate, and automate all ETL tasks, ensuring seamless, repeatable workflows with monitoring and error handling.

Testing & Validation:

Perform systematic data validation, including integrity and consistency checks, and execute load testing to verify pipeline performance and stability under realistic workloads.

Visualization:

Build insightful reports and dashboards—using Tableau, Power BI, or Python plotting libraries—to present key metrics and trends for business stakeholders.

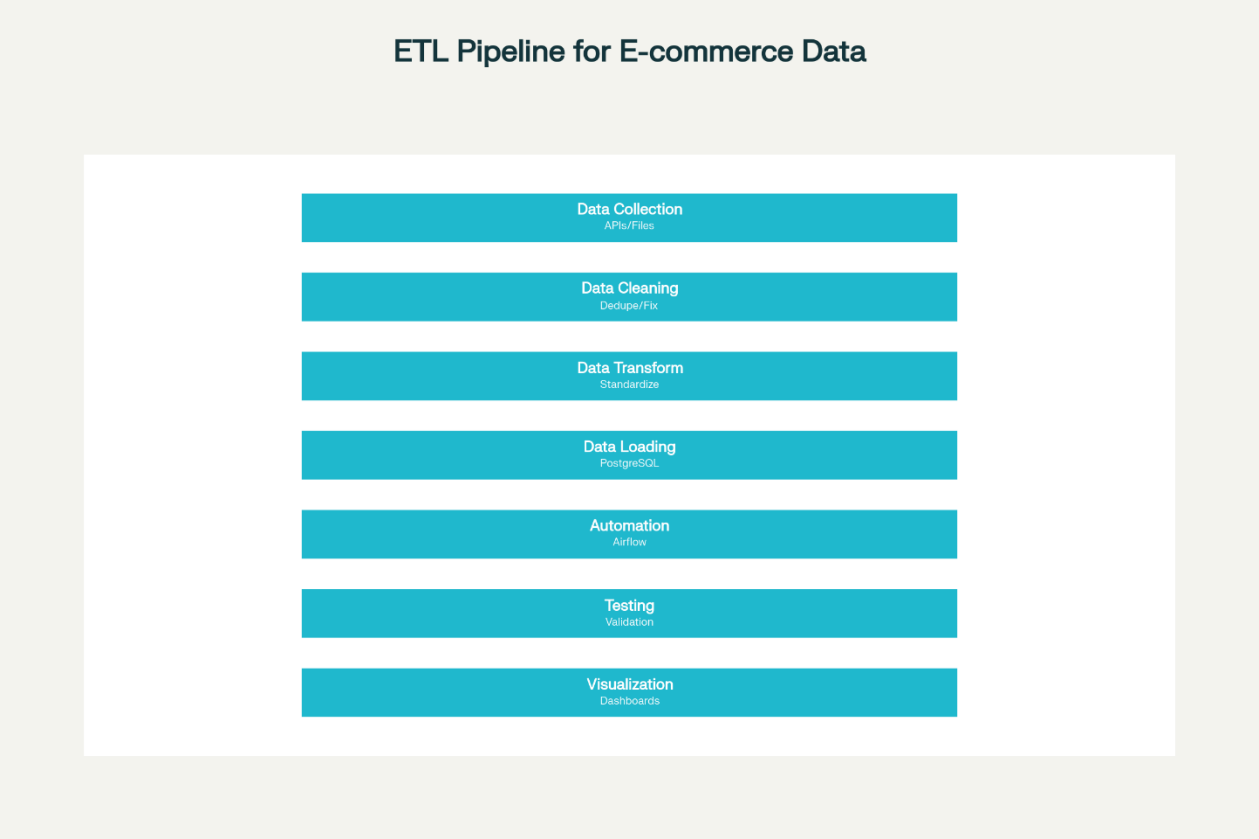

Fig: Flow Chart of ETL Pipeline for E-Commerce Data

Chapter 6

SYSTEM REQUIREMENTS

6.1 Software Requirement

Category	Tools/Libraries
Programming	Python 3.8+, SQL
ETL Framework	Apache Airflow
Database	PostgreSQL with TimescaleDB extension
Visualization	Power BI, Tableau, or Python (Matplotlib)
Version Control	Git/GitHub
Deployment	Render or Docker

6.2 Hardware Requirements

Component	Specification/Details
Development PC	Intel i5 or AMD Ryzen 5, 8 GB RAM
RAM	Minimum 8 GB (16 GB recommended)
Storage	256 GB SSD (500 GB recommended)
Cloud VM	2 vCPUs, 8 GB RAM, 50 GB storage

Chapter 7

CONCLUSION

The project delivers a robust, scalable ETL pipeline tailored for e-commerce order data, automating the ingestion, transformation, and loading of large volumes of data. The architecture supports modular data processing and orchestration that can be easily extended to support real-time analytics. This solution enhances data quality, reduces manual errors, and provides a foundation for advanced analytics and business intelligence. The project aligns with industry standards and equips the team with practical data engineering expertise, positioning it for further expansion at enterprise scale.

REFERENCES

1) Raj, A., Bosch, J., Holmström Olsson, H., & Wang, T. J. (2020). “Modelling Data Pipelines”. 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA).IEEE. https://doi.org/10.1109/SEAA51224.2020.00014

2) Author :"Kaichong Zhang" “Incorporating Deep Learning Model Development With an End-to-End Data Pipeline” published in IEEE Access (2024), * https://doi.org/10.1109/ACCESS.2024.3456113

3) Authors : "Aiswarya Raj" , "Jan Bosch" ,"Helena Holmstrom Olsson and Tian J. - “Modelling Data Pipelines” > https://doi.org/10.1109/SEAA51224.2020.00014

4) “Spearheading Big Data Solutions: Optimizing Data Pipelines For Enhanced Efficiency And Performance” by *Kiran Polimetla* and *Farah Jenny* (Educational Administration: Theory and Practice

5) Zarate, G., Lopez Osa, M. J., Torre-Bastida, A. I., Iturraspe, U., Arjona, J., Navarro, B., & Gimeno, A. (2024). Evolution of Extract-Transform-Load (ETL) processes towards data product pipelines. In Proceedings of the 4th Eclipse Security, AI, Architecture and Modelling Conference on Data Space (eSAAM 2024), Mainz, Germany. ACM. https://doi.org/10.1145/3685651.3686662

6) Shojaee Rad, Z., & Ghobaei-Arani, M. (2024). Data pipeline approaches in serverless computing: a taxonomy, review, and research trends. Journal of Big Data, 11(82). https://doi.org/10.1186/s40537-024-00939-0

7) Machado, G. V., Cunha, Í., Pereira, A. C. M., & Oliveira, L. B. (2019). DOD-ETL: Distributed On-Demand ETL for Near Real-Time Business Intelligence. arXiv preprint arXiv:1907.06723. https://arxiv.org/abs/1907.06723

8) Foidl, H., Golendukhina, V., Ramler, R., & Felderer, M. (2024). Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers. The Journal of Systems and Software, 207, 111855. https://doi.org/10.1016/j.jss.2023.111855

9) Prajwal Babu, A. S., & Suma B. (2022). *International Journal of Computer Applications Technology and Research, Vol. 11, Issue 08, pp. 315–319. DOI: [10.7753/IJCATR1108.1004](https://doi.org/10.7753/IJCATR1108.1004)

10) This paper is authored by Anurag Awasthi from IIT Kanpur, India, and Aniket Vaidya from Carnegie Mellon University, USA. The title of the paper is "ETL Pipeline Integration for Machine Learning-Based Product Classification: A Comprehensive Guide."

11) Authored by Gustavo V. Machado, Ítalo Cunha, Adriano C. M. Pereira, and Leonardo B. Oliveira, and is titled “DOD-ETL: Distributed On-Demand ETL for Near Real-Time Business Intelligence.”

12) Mahesha K et al. (2025). EPRA International Journal of Research and Development, Vol. 10, Issue 8, DOI: [10.36713/epra23838] https://doi.org/10.36713/epra23838

Signature of Guide
Guide Name: Ms Apoorva S M
Designation: Assistant Professor

Signature of Coordinator
Name of Coordinator: Dr UmaMahesh R N
Designation: Associate Professor