Mini Project Report

on

Functional code development of existing data transformations for the retail data warehouse.

Submitted

BY

Ms. Ankita Ghule (Roll No: 04)

Ms.Annlip Gour (Roll No: 05)

Mr. Mayank Junankar (RollNo: 57)

Mr. S Akshansh (RollNo: 70)

Semester/Section: 7th/A

Under the Guidance of

Ms. Archana S. Pimpalkar

November 2022-23

Department of Computer Technology
YESHWANTRAO CHAVAN COLLEGE OF ENGINEERING, Nagpur

(An Autonomous Institution Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)

YESHWANTRAO CHAVAN COLLEGE OF ENGINEERING
NAGPUR
(An Autonomous Institution Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)

Department of Computer Technology

(2022-23)

Certificate

This is to certify that the Mini Project Report titled "Functional code development of existing data transformations for the retail data warehouse. " is submitted towards the partial fulfillment of the requirement of the Mini Project course in VII Semester, B.E.(Computer Technology).

Submitted by:

Ms. Ankita Ghule	**(RollNo: 04)**
Ms. Annlip Gour	**(RollNo: 05)**
Mr. Mayank Junankar	**(RollNo: 57)**
Mr. S Akshansh	**(RollNo: 70)**

is approved.

Project Guide

Ms. Archana S. Pimpalkar

Project Coordinator

Smita R. Kapse

Head, Department of Computer Technology

Dr. R.D.Wajgi

Date:___________
Place:_________

Certificate of collaboration (industry/research organization)

(To be printed on Industry letterhead)

Certificate of Completion

This is to certify that the following students of the final year Computer Technology Department, Yeshwantrao Chavan College of Engineering, Nagpur, have completed the Live/Industry/Joint research mini project titled "**Functional code development of existing data transformations for the retail data warehouse.**” under the guidance of (Ms. Archana S. Pimpalkar) and Co-guide (Mr. Koustubh Laghate) with industry name InCredo Technologies for the session 2022-23.

Name of student: Ankita Ghule	**Enrollment No: 19010345**
Name of student: Annlip Gour	**Enrollment No: 20030123**
Name of student: Mayank Junankar	**Enrollment No: 20030193**
Name of student: S Akshansh	**Enrollment No: 19010927**

Name and Signature of Industry Guide with Seal

ACKNOWLEDGEMENT

We would like to thank our guide Ms. Archana S. Pimpalkar and industry mentor Mr. Koustubh Laghate for thorough guidance in the project. We are extremely grateful and indebted to them for their expert, sincere, valuable guidance and encouragement which was of immense help to us.

We would like to express our sincere gratitude to **Dr. R.D.Wajgi**, Head, Department of Computer Technology, for her constant encouragement towards the successful completion of our work.

We wish to express our sincere thanks to Dr. U.P. Waghe, the Principal of our college, for providing us with all the necessary facilities and the infrastructure without which we would not have been able to complete our project successfully.

We would also like to thank our Project Coordinator Prof. S. R. Kapse for their continuous guidance owing to which the project could take shape.

We would like to thank the technical assistant, Mrs. B. H. Kulkarni, for providing the necessary technological support. Last, but not least, we would like to thank all the faculty members and non-teaching staff members who helped us despite their busy schedule

Abstract

The industry standard name for data extraction, transformation, and loading into the data warehouse is ETL (Extract Transform Load). Numerous ETL tools with graphical user interfaces and other built-in functions have been developed to make it easier to create and maintain ETL processes (parallelism, logging, transformation libraries, documentation generation, etc.). The drawback of such GUI ETL solutions is that they encourage developers to utilise mouse operations more frequently than writing programming code, which can seem inconvenient for certain developers, especially when dealing with numerous repeated, comparable jobs. In our project, we offer a different strategy, one that uses functional codes that is built on the Python scripting language and in which ETL activities are specified by writing Python code. The user can easily and effectively construct complicated ETL tasks with numerous sources and parallel tasks which implements a variety of common ETL transformations while utilising all of Python's flexibility. On a test case, we demonstrate how our code rivals the GUI technique and simplifies ETL development.

Table of Contents

Title **Page No.**

List of Figures

List of Tables

1.0 INTRODUCTION

Data Transformations are used for mapping that represents the operations that you want to perform on data. We will define the functionality of data transformation using the Python programming language. ETL (Extract Transform Load) process is the term for data extraction, transformation, and loading into the Data Warehouse (DW). Here, we’ll focus on transformations. To perform the development and maintenance of transformations, many tools have been developed with the basis of Graphical User Interfaces and various built-in functionalities (transformation libraries, documentation generation, etc.). The disadvantage of such GUI tools is that development is carried out using mouse operations and less by writing programming code, which feels unnatural for some developers, especially with many similar, repetitive tasks. Here, transformation tasks are defined by writing Python code. This implements various ETL transformations and allows the user to simply and efficiently define complex ETL tasks while leveraging the full flexibility of Python.

1.0 AIM & OBJECTIVES

Aim: Functional code development of existing data transformations for the retail data warehouse.

Objectives:

- Define transformation tasks by writing functional code.
- To simply and efficiently define complex ETL tasks with multiple sources while leveraging the full flexibility of functional codes.
- In place of GUI, developing a functional code to provide better understanding and customizability of tasks.
- There is a cost involved to use the existing ETL tools, but by using the functional code we can perform the required tasks at a very low cost or free of cost.

2.0 LITERATURE REVIEW

Reference No.	Title	Authors	Published in	Major Findings
1	ETLator - a scripting ETL framework	Miran Radonić; Igor Mekterović	2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)	The authors presented an alternative approach - an ETL framework "ETLator" based on Python scripting language where ETL tasks are defined by writing Python code. ETLator implements various typical ETL transformations and allows the user to simply and efficiently define complex ETL tasks with multiple sources and parallel tasks whilst leveraging the full flexibility of Python. ETLator also provides logging and can document ETL tasks by generating data flow images. In a test case, they show that ETLator simplifies ETL development and rivals the GUI approach.

2	pygrametl: A Powerful Programming Framework for Easy Creation and Testing of ETL Flows	Søren Kejser Jensen, Christian Thomsen, Torben Bach Pedersen & Ove Andersen	CHI '11: Proceedings of the SIGCHI Conference on Huma Factors in Computing Systems -- May 2011 Pages 3363–3372	They propose to develop ETL flows by writing code. To make the programming easy, they proposed the Python-based ETL framework pygrametl in 2009. They have extended pygrametl significantly since the original release, and in this paper, They present an up-to-date overview of the framework. pygrametl offers commonly used functionality for programmatic ETL development and enables the user to efficiently create effective ETL flows with the full power of programming. Each dimension is represented by a dimension object that manages the underlying table or tables in the case of a snowflake dimension. Thus, filling a slowly changing or snowflaked dimension only requires a single method call per row as pygrametl performs all of the required lookups, insertions, and assignment of surrogate keys.
3	Empirical Analysis of Programmable ETL Tools	Neepa Biswas, Anamitra Sarkar & Kartick Chandra Mondal	26 June 2019 Springer, Singapore	This paper focuses on an alternative ETL developmental approach taken by hand coding. In some contexts, it is appropriate to custom-develop an ETL code that can be cheaper, faster, and maintainable. Some well-known code-based open-source ETL tools (Pygrametl, Petl, Scriptella, R_etl) developed by the academic world have been studied in this article. Their architecture and implementation details are addressed here. This paper aims to present a comparative evaluation of these code-based ETL tools. Not to acclaim that code-based ETL is superior to the GUI-based approach. It depends on the particular requirement, data strategy, and infrastructure of any organization to choose the path between Code based and GUI-based approaches.

4	A Transformation System for Developing Recursive Programs	R. M. Burstall And John Darlington	Journal of the Assooat~on for Computing Machinery, Vol 24, No 1, January 1977, pp 44-67	A system of rules for transforming programs is described, with the programs in the form of recursion equations. An initially very simple, lucid, and hopefully, correct program is transformed into a more efficient one by altering the recursion structure Illustrative examples of program transformations are given, and a tentative implementation is described Alternative structures for programs are shown, and a possible initial phase for an automatic or semiautomatic program manipulation system is indicated They start with programs having extremely simple structures and only later introduce the complications which they usually take for granted even in high-level language programs. These complications arise by introducing useful interactions between what were originally separate parts of the program, benefiting from what might be called "economies of interaction." They proceed in quite an empirical manner, showing examples of various kinds.

5	Program Transformation Mechanics	Jonne van Wijngaarden Eelco Visser	May 2003 UU-CS-2003-048 Institute of Information and Computing Sciences Utrecht University	Transformation techniques are spreading from application in compilers to general use in generative programming and document processing. Since transformation requires operations such as pattern matching, generic structure traversal, and querying, which are not normally provided by general-purpose programming languages, many tools have been developed to provide higher-level support for the implementation of transformations. These tools come in many flavors each with their own merits and based on different paradigms, which makes comparison difficult. In this paper, They consider transformation from the point of view of mechanics and develop a classification of transformation mechanisms that provides a reference for comparing tools developed for different applications, using different implementations, and in different programming paradigms. To do so They distinguish three fundamental aspects of transformation mechanisms: scope, direction, and stages. They apply this classification in a discussion of design patterns for transformation, characterization of several typical transformations, and a systematic comparison of eleven representative transformation tools.

Table No. 1 Literature Reviews

3.0 PROPOSED METHODOLOGY

We propose to develop a similar retail data management system based on Python functional codes for performing the transformations. This will save the cost of the software for the retailer and provide more customizability and options for transformations to the managers.

The project will be based on SQL for storing data, and Python libraries like Pandas and Numpy for data transformations and analytics.

The system will offer various transformations like:

Transformation	Description
Source	Reads data from a source.
Target	Writes data to a target.
Aggregator	An active transformation that performs aggregate calculations on groups of data.
Joiner	An active transformation that joins data from two sources.
Convert	An active transformation that converts table format data to csv or json format.

Table no. 2 Data Transformations

4.0 RESULTS AND DISCUSSION

After working on the development of this project, we got a deeper understanding of Python libraries. We built a function to extract data from Excel(xlsx) files as data can be easily manipulated on the excel sheet, and it isn't a safe storage for a database of important sales records. After fetching the data, we combine different databases and excel sheets like product details and transaction details into a single normalized database and store it in SQL completely using Python. This completes our ETL process.

The same task would take thousands of lines of code in SQL, but Python makes it easy.

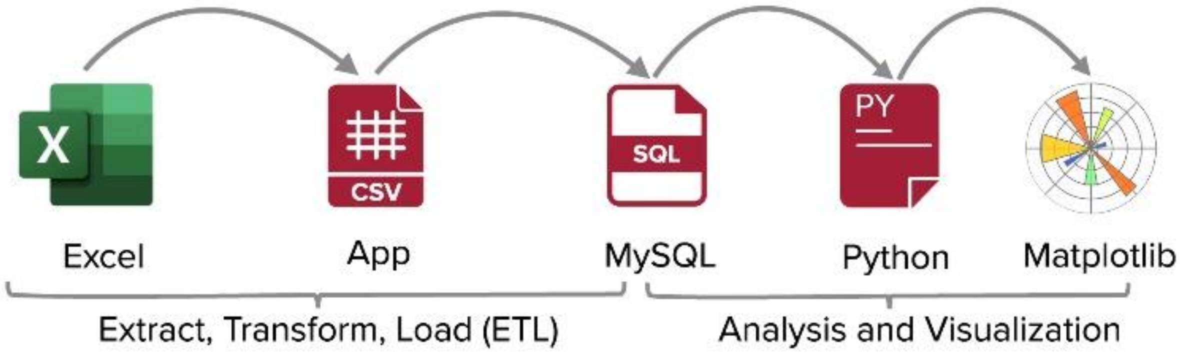

Fig No. 1 Flow of the system

Then as an extension to this project, we can use the database to visualize data patterns which will come in handy to analysts and managers of the company. This can be done with the help of Python libraries like matplotlib.

5.0 CONCLUSION AND FUTURE SCOPE

Conclusion:

We have created platform-independent Python functional codes, with support for various data sources like product and transaction records for retail data. It consists of writing short scripts in Python, with Python libraries at one's disposal, which results in expressive code, and overall an easier and faster experience of developing and maintaining projects.

Future Scope:

Future development of this will include built-in logging and documentation generation features (data flow charts included) which are traditional features found only in GUI-based ETL tools. Also, we plan to develop a rich data transformation and data quality library.

Other developers can use our project to build their projects too, as it is an open-source project.

And our future transformations to be performed include the following:

Transformation	Description
Cleanse	Use a cleanse asset to standardize the form and content of your data.
Data Masking	A passive transformation that masks sensitive data as realistic test data for nonproduction environments.
Deduplicate	Use a deduplicate asset to find instances of duplicate identities in a data set and optionally to consolidate the duplicates into a single record.
Expression	A passive transformation that performs calculations on individual rows of data.

Filter	An active transformation that filters data from the data flow.
Labeler	Use a labeler to identify the types of information in an input field and to assign labels for each type to the data.
Lookup	Looks up data and defines the lookup condition and the return values.
Normalizer	An active transformation that processes data with multiple-occurring fields and returns a row for each instance of the multiple-occurring data.
Rank	An active transformation that limits records to a top or bottom range.
Sequence Generator	A passive transformation that generates a sequence of values.
Sorter	A passive transformation that sorts data in ascending or descending order, according to a specified sort condition.
Union	An active transformation that merges data from multiple input groups into a single output group.
SQL	An active transformation that pushes the output to a single SQL table.

Table No. 3 Future Data Transformations

REFERENCES

[1] ETLator – a scripting ETL framework Miran Radonić*1, Igor Mekterović*2 * Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia

[2] “PEP 249 - Python Database API Specification v2.0”, https://www.python.org/dev/peps/pep-0249/, Accessed on 2/11/2022.

[3] Jensen, S.K., Thomsen, C., Pedersen, T.B., Andersen, O. (2021). pygrametl: A Powerful Programming Framework for Easy Creation and Testing of ETL Flows. In: Hameurlain, A., Tjoa, A.M. (eds) Transactions on Large-Scale Data- and Knowledge-Centered Systems XLVIII. Lecture Notes in Computer Science(), vol 12670. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-63519-3_3

[4] Biswas, N., Sarkar, A., Mondal, K.C. (2019). Empirical Analysis of Programmable ETL Tools. In: Mandal, J., Mukhopadhyay, S., Dutta, P., Dasgupta, K. (eds) Computational Intelligence, Communications, and Business Analytics. CICBA 2018. Communications in Computer and Information Science, vol 1031. Springer, Singapore. https://doi.org/10.1007/978-981-13-8581-0_22

[5]Beyer, M.A., Thoo, E., Selvage, M.Y., Zaidi, E.: Gartner magic quadrant for data integration tools (2020)

[6] www.datasciencecentral.com/profiles/blogs/10-open-source-ETL-tools.
[15/10/2022- 10:25 pm]

[7] J. Visser. Visitor combination and traversal control. In OOPSLA 2001 Conference Proceedings: Object-Oriented Programming Systems, Languages, and Applications, pages 270–282, 2001.

[8] K. Fisher and R. Gruber. Pads: a domain-specific language for processing ad hoc data. In ACM PLDI, pages 295–304, 2005.

[9] L. V. S. Lakshmanan, F. Sadri, and S. N. Subramanian. SchemaSQL: An extension to SQL for multi database interoperability. ACM Trans. Database Syst., 26(4):476–519, 200

[10] Transformation types [17/11/2022- 02:20pm]

[11] https://www.lucidchart.com/pages/examples/flowchart_software [20/10/2022- 08:56pm]

[12]PostgreSQL[21/10/2022- 03:15pm]

.

[13]Ali, S.M.F., Wrembel, R.: From conceptual design to performance optimization of ETL workflows: current state of research and open problems. VLDB J. (VLDBJ) 26(6), 777–801 (2017). https://doi.org/10.1007/s00778-017-0477-2

[14]Andersen, O., Thomsen, C., Torp, K.: SimpleETL: ETL processing by simple specifications. In: 20th International Workshop on Design, Optimization, Languages, and Analytical Processing of Big Data (DOLAP). CEUR-WS.org (2018)

[15]Beck, K.: Test Driven Development: By Example, pp. 194–195. Addison-Wesley Professional, Boston (2002)

[16]Chandra, P., Gupta, M.K.: Comprehensive survey on data warehousing research. Int. J. Inf. Technol. (IJIT) 10(2), 217–224 (2018). https://doi.org/10.1007/s41870- 017-0067-y

[17] https://www.informatica.com/resources/articles/what-is-etl.html [17/11/2022- 11:22pm]

[18]https://www.analyticsvidhya.com/blog/2020/03/understanding-transform-function-python[15/11/2022- 2:44pm]