**ETL TOOL**

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING**

BY

**Shubhi Upadhyay**

**EN18CS302049**

Under the Guidance of

**Mrs. Varsha Sharda**



**Department of Computer Science & Engineering**

**Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**APRIL 2022**

**ETL TOOL**

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING**

BY

**Shubhi Upadhyay**

**EN18CS302049**

Under the Guidance of

**Mrs. Varsha Sharda**



**Department of Computer Science & Engineering**

**Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**APRIL 2022**

**Report Approval**

The project work **“ETL TOOL”** is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner

Mrs. Varsha Sharda

Asst. Professor

External Examiner

Name:

Designation

Affiliation

**Declaration**

I/We hereby declare that the project entitled **“ETL TOOL”** submittedin partial fulfillment for the award of the degree of Bachelor of Technology in Computer

Science & Engineering’ completed under the supervision of **Mrs. Varsha Sharda** **Asst. Professor,** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, I/we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

**Signature and name of the student with date**

**Certificate**

I, **Varsha Sharda** certify that the project entitled **“ETL TOOL”** submittedin partial fulfillment for the award of the degree of Bachelor of Technology by **Shubhi Upadhyay** istherecordcarried out by him/them under my/our guidance and that the work has not formed the basis of award of any other degree elsewhere.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mrs. Varsha Sharda

Computer Science & Engineering Department

Medi-Caps University, Indore

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<Name of External Guide (If any)>

<Name of the Department>

Name of the Organization

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Pramod S. Nair

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

**Acknowledgement**

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal,** who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Dilip K. Patnaik,** Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D K Panda,** Pro Vice Chancellor, **Dr. Suresh Jain,** DeanFaculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Pramod S. Nair** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my **External Guide**, Team Lead, Indus Valley Partners (India) Pvt. Ltd as well as to my Internal Guide, Mrs. Varsha Sharda**,** Assistant Professor, Department of Computer Science, MU, without whose continuous help and support, this project would ever have reached to the completion.

I would also like to thank to my team who extended their kind support and help towards the completion of this project.

It is their help and support, due to which we became able to complete the design and technical report. Without their support this report would not have been possible.

Without their support this report would not have been possible.

**Shubhi Upadhyay**

B.Tech. IV Year

Department of Computer Science & Engineering

Faculty of Engineering

Medi-Caps University, Indore

**Abstract**

ETL is one of the important processes required by **Business Intelligence**. Business Intelligence relies on the data stored in data warehouses from which many analyses and reports are generated which helps in building more effective strategies and leads to tactical, and operational insights and decision-making.

ETL refers to the Extract, Transform, and Load process. It is a kind of data integration step where data coming from different sources gets extracted and sent to data warehouses. The word ‘data’ is very crucial as most of the business is run around this data, data flow, data format, etc. Modern applications and working methodology require real-time data for processing purposes and in order to satisfy this purpose, there are various **ETL tools** available in the market.

**Enterprise Data Management (EDM)** is one such tool, given the sheer volume of today’s data flows, the world’s most sophisticated hedge funds and private equity firms turn to Indus Valley Partners to better manage an increasingly complex environment. Enterprise Data Management is an award-winning, best-in-class platform engineered to solve the challenges of modern data management. It’s part of our Master Data Management suite of solutions, which offers a comprehensive framework for managing buy-side data.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Chapter Headings** | **Page No.** |
|  |  |  |
|  | Report Approval | iii |
|  | Declaration | iv |
|  | Certificate | v |
|  | Offer Letter of the Project work-II/Internship | vi |
|  | Completion letter/certificate | vii |
|  | Acknowledgement | viii |
|  | Abstract | ix |
|  | Table of Contents | x |
|  | List of figures | xi |
| Chapter 1 | Introduction |  |
|  | 1.1 Introduction | 1 |
|  | 1.2 Literature Review | 2 |
|  | 1.3 Objectives | 5 |
|  | 1.4 Significance | 5 |
|  | 1.5 Source of Data | 6 |
| Chapter 2 | Infrastructure of Setup |  |
|  | 2.1 Experimental Set-up | 7 |
|  | 2.2 Procedures Adopted | 9 |
| Chapter 3 | ETL in Data Warehousing |  |
|  | 3.1 What is Data Warehousing | 11 |
|  | 3.2 ETL Process in Data Warehouses | 12 |
|  | 3.3 Staging Architecture | 12 |
|  | 3.4 Data Extraction | 13 |
|  | 3.4.1 Data Extraction Methods | 14 |
|  | 3.5 Data Transformation | 16 |
|  | 3.5.1 Transformation Types | 17 |
|  | 3.6 Data Loading | 20 |
| Chapter 4 | The Role of Data Pipelines in the EDW |  |
|  | 4.1 Data Pipeline Automation | 24 |
|  | 4.2 Optimizing Data Pipelines | 24 |
|  | Results & Discussions | 25 |
| Chapter 5 | Summary and Conclusions | 27 |
| Chapter 6 | Future scope | 28 |
|  | Appendix | 29 |
|  | Bibliography | 30 |
|  |  |  |

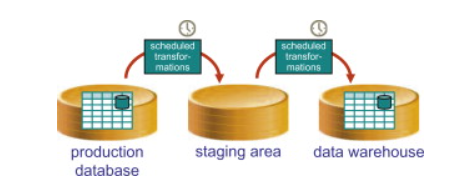
**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Figures List** | **Page No.** |
| Chapter 1 | Fig 1.1: Develop the staging area and the data warehouse and use scheduled transformation to load them. | 1 |
|  | Fig 1.2: Data Pipeline | 1 |
|  | Fig 1.3 Popular ETL Implementation Approaches | 2 |
|  | Fig 1.4: ETL Quality Attributes | 3 |
|  | Fig 1.5: Challenges in Developing ETL Solutions | 4 |
|  | Fig 1.6: Application Domains of ETL | 4 |
| Chapter 2 | Fig 2.1: Experimental Setup | 7 |
|  | Fig 2.2: Procedures Adopted | 9 |
| Chapter 3 | Fig 3.1: Data Warehouse Data Flow | 11 |
|  | Fig 3.2: ETL Process | 12 |
|  | Fig 3.3: Staging data in preparation for loading into an analytical environment. | 13 |
|  | Fig 3.4: Data Extraction | 13 |
|  | Fig 3.4.1: Offline Data Extraction | 14 |
|  | Fig 3.4.2: Preview of Data Extracted Offline | 15 |
|  | Fig 3.5.1: Multistage Data Transformation | 16 |
|  | Fig 3.5.2 Pipelined Data Transformation | 17 |
|  | Fig 3.5.3: Data Transformation | 18 |
|  | Fig 3.5.4: Data Transformation for ETL workflow | 19 |
|  | *Fig 3.5.5:* Data Cleansing for ETL workflow | 19 |
|  | *Fig 3.5.6:* Data Mapping for ETL workflow | 20 |
|  | Fig 3.6.1: Data Loading | 20 |
|  | Fig 3.6.2: Data Loading | 22 |
|  | Fig 3.6.3: Data Loading in SQL Server DB Table | 22 |
| Chapter 4 | Fig 4.1: Data Pipeline performing extraction and Loading | 23 |
|  |  |  |

**Chapter 1**

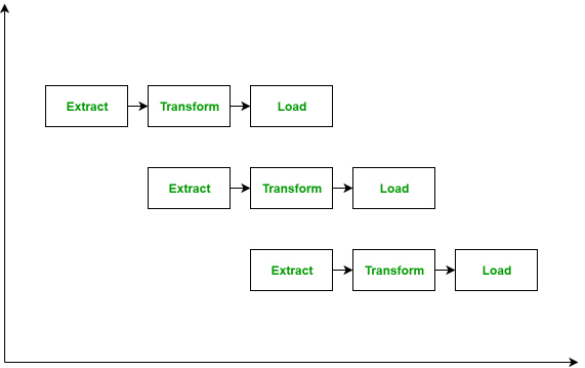
**Introduction**

**1.1 Introduction**

ETL is a process in Data Warehousing and it stands for Extract, Transform and Load. It is a process in which an ETL tool extracts the data from various data source systems, transforms it in the staging area, and then finally, loads it into the Data Warehouse system.

*Fig 1.1: Develop the staging area and the data warehouse and use scheduled transformation to load them.*

ETL process can also use the pipelining concept i.e., as soon as some data is extracted, it can be transformed and during that period some new data can be extracted. And while the transformed data is being loaded into the data warehouse, the already extracted data can be transformed.



*Fig 1.2: Data Pipeline*

It’s tempting to think a creating a Data warehouse is simply extracting data from multiple sources and loading into database of a Data warehouse. This is far from the truth and requires a complex ETL process. The ETL process requires active inputs from various stakeholders including developers, analysts, testers, top executives and is technically challenging.

In order to maintain its value as a tool for decision-makers, Data warehouse system needs to change with business changes. ETL is a recurring activity (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.

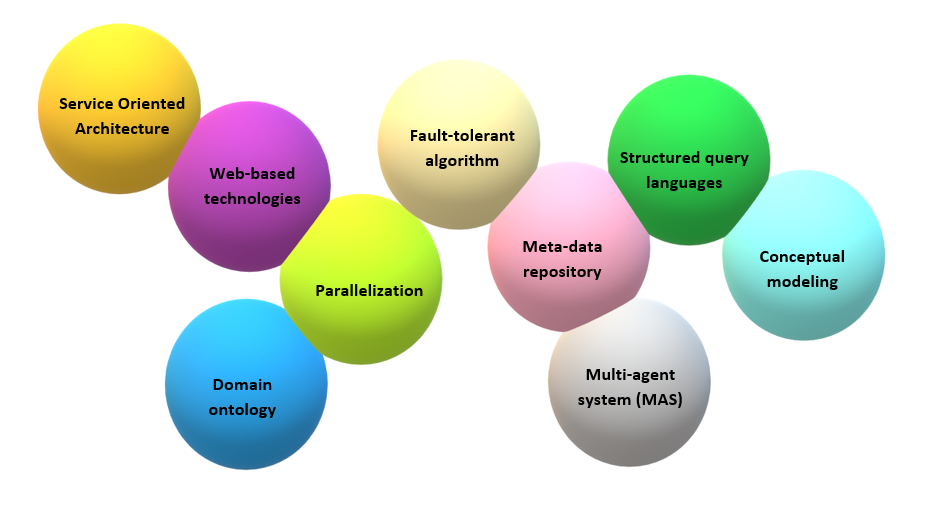
**1.2 Literature Review:**

Our aim in conducting this review is to provide a critical perspective on the current state of research in ETL, especially with regards to challenges, implementation approaches, quality attributes and the depth of coverage.

Review Questions: We identified 4 review questions with answers that closely align with this objective:

**RQ1**: What are the techniques or approaches used for implementing ETL workflows or processes?

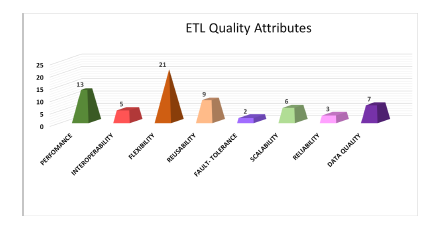
* Below figure given nine (9) popular implementation approaches. These include approaches that are implemented using: (i) Service oriented architecture (SOA) (ii) Web-based technologies, e.g., semantic web, these are reported (iii) Fault-tolerant algorithm (iv) Structured query languages (SQL) (v) Parallelization (parallel computing paradigm), e.g., Map Reduce (vi) Domain ontology (vii) multi-agent system (MAS) (viii) Conceptual modeling e.g., unified modeling language (UML) and business process modeling (BPMN) and finally (ix) Meta-data repository.



*Fig 1.3 Popular ETL Implementation Approaches*

**RQ2**: What are the quality attributes to be considered when selecting ETL solutions?

* Below figure shows the 8 quality attributes identified from primary studies. They include **Performance** which implies that the ETL process must be completed within a specified time window and also the functional mapping of data from source to target must be correct **Interoperability**, implies that ETL solutions should support the integration of data from heterogeneous operational data sources, irrespective of technological, syntactic, and semantic schemas. **Flexibility** refers to the ability of an ETL solution to accommodate changes in data integration requirements. **Reusability**, the ability of an ETL solution to be used in various application. **Fault-tolerance** i.e., the ability of ETL solution to function optimally in the presence of fault. **Scalability**, which is reported in the ability of ETL solution to be used for data of varying volumes and complexities. **Reliability** is the probability that ETL solution will not fail to perform its intended functions within the specified time. **Data quality**, refers to the degree to which data produced by an ETL solution is fit for use.



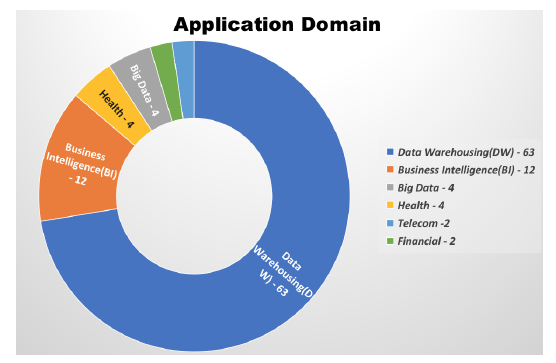
*Fig 1.4: ETL Quality Attributes*

**RQ3**: What are the prevailing challenges in developing ETL solutions?

* Below Figure shows a word cloud summarizing the challenges identified from the primary studies considered in our work. We identified 7 key challenges in developing ETL solutions. These include complexity, data heterogeneity, cost, time, maintenance, lack of automation and standardization issues.

*Fig 1.5: Challenges in Developing ETL Solutions*

**RQ4**: What is the current depth of coverage in ETL research with respect to the application domain?

* As shown in below Figure, there are 6 domains where ETL solutions are applied. These include data warehouse, business intelligence, big data, health, telecommunications and finance.

*Fig 1.6: Application Domains of ETL*

**1.3 Objectives**

ETL tools collect, read, and migrate large volumes of raw data from multiple data sources and across disparate platforms. They load that data into a single database, data store, or data warehouse for easy access.

ETL allows businesses to consolidate data from multiple databases and other sources into a single repository with data that has been properly formatted and qualified in preparation for analysis. This unified data repository allows for simplified access for analysis and additional processing. It also provides a single source of truth, ensuring that all enterprise data is consistent and up-to-date.

To gain a 360-degree view of data and drive successful business outcomes, organizations have always relied on ETL processes. However, with the advancements in technology, ETL has evolved from a hand-coded approach to an automated process that works with large datasets in minimal time.

**1.4 Significance**

The importance of ETL in an organization is in direct proportion to how much the organization relies on data warehousing. ETL tools collect, read, and migrate large volumes of raw data from multiple data sources and across disparate platforms. They load that data into a single database, data store, or data warehouse for easy access. They process the data to make it meaningful with operations like sorting, joining, reformatting, filtering, merging, and aggregation. Finally, they include graphical interfaces for faster, easier results than traditional methods of moving data through hand-coded data pipelines.

ETL tools break down data silos and make it easy for your data scientists to access and analyze data, and turn it into business intelligence. In short, ETL tools are the first essential step in the data warehousing process that eventually lets you make more informed decisions in less time.

A Web-based ETL works like a Web service to help you integrate your data. There are many different models of ETL tools in today’s BI market, from complex, specialized products to light, Web-based solutions that work easily with multiple data sources.

**Benefits of Web-based ETL Tools**

A Web-based ETL gives you these unique benefits:

* **Fully Web-based Data Integration** – With a Web-based ETL, you can not only seamlessly integrate your data, but also integrate the ETL with your other BI applications-regardless of vendor or brand. Use the ETL as a Web Service, launch ETL jobs from any standard-type processes and Web processes. Integrate the ETL into your business processes and workflows tied to triggers and alerts.
* **Unique Web Data Sources** – Do more with a diverse set of data: a Web-based ETL gives you easy connections out of the box with Web Services and other Web-oriented data sources (e.g., SalesForce.com, Google Docs, RSS and ATOM feeds). Today’s most cutting-edge, Web-based ETL tools connect with relational databases and flat-file data sources.
* **Elemental Development Methodology** – The same concepts you use to define logic in reports, templates, process files, etc., can be applied and even reused/shared in a Web-based ETL.
* **Optimization for BI and Reporting** – Look for a Web-based ETL that is designed to work with data geared towards reporting, analysis and visualization. In particular, there are Web-based ETL tools that are created and marketed by companies specializing in BI; apart from truly optimizing data for reporting and analysis, this type of ETL will integrate seamlessly with your other BI applications.

**1.5 Source of Data**

* Akkaoui, Z.E., Zimanyi, E., Mazon, J.: A model-driven framework for ETL process

development. Proceedings of the ACM (2011)

* Aqlan, F., Nwokeji, J.C.: Applying product manufacturing techniques to teach

data analytics in industrial engineering: A project based learning experience. In:

2018 IEEE Frontiers in Education Conference (FIE), pp. 1{7 (2018). DOI 10.1109/

FIE.2018.8658588

* Aqlan, F., Nwokeji, J.C., Shamsan, A.: Teaching an introductory data analytics

course using microsoft access and excel. In: 2020 IEEE Frontiers in Education

Conference (FIE), pp. 1{10 (2020). DOI 10.1109/FIE44824.2020.9274247

* Bansal, S.K.: Towards a semantic extract-transform-load (etl) framework for big

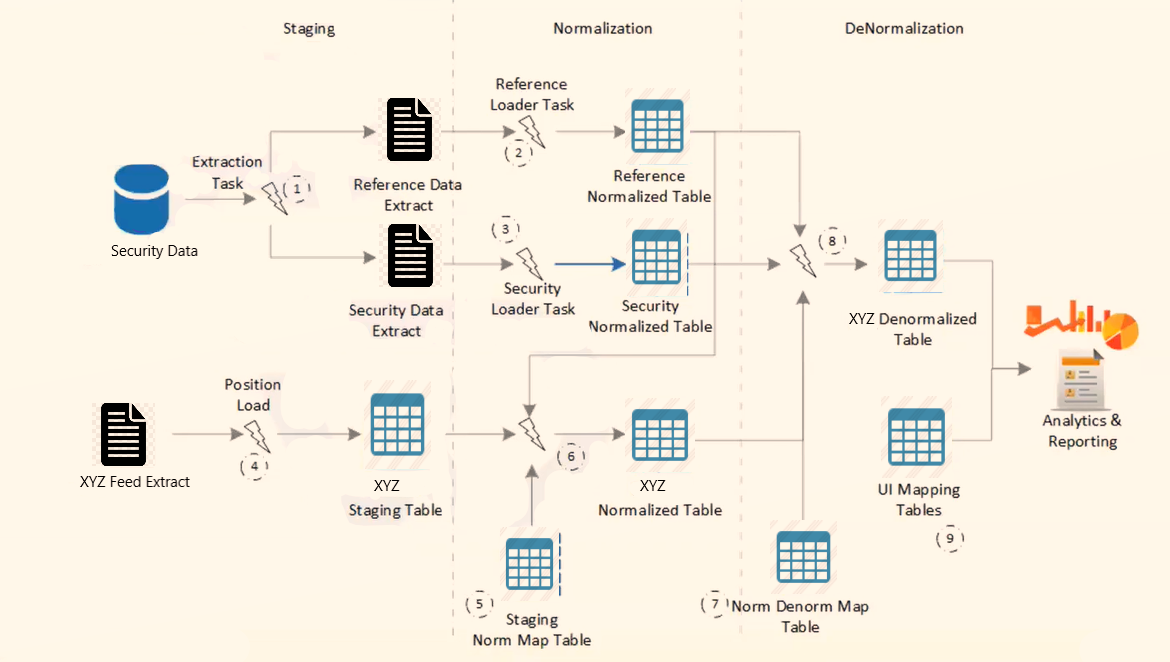
data integration. In: 2014 IEEE International Congress on Big Data, pp. 522{529

(2014). DOI 10.1109/BigData.Congress.2014.82

* <https://www.astera.com/type/blog/etl-what-it-means-and-why-is-it-important/>
* <https://www.matillion.com/resources/blog/the-importance-of-etl-tools-in-data-warehousing>

**Chapter – 2**

**Infrastructure of Setup**

**2.1 Experimental Setup**

*Fig 2.1: Experimental Setup*

The figure above represents a Data Warehouse ETL Flow comprising mainly of three phases:

* Staging
* Normalization
* Denormalization

**STAGING –**

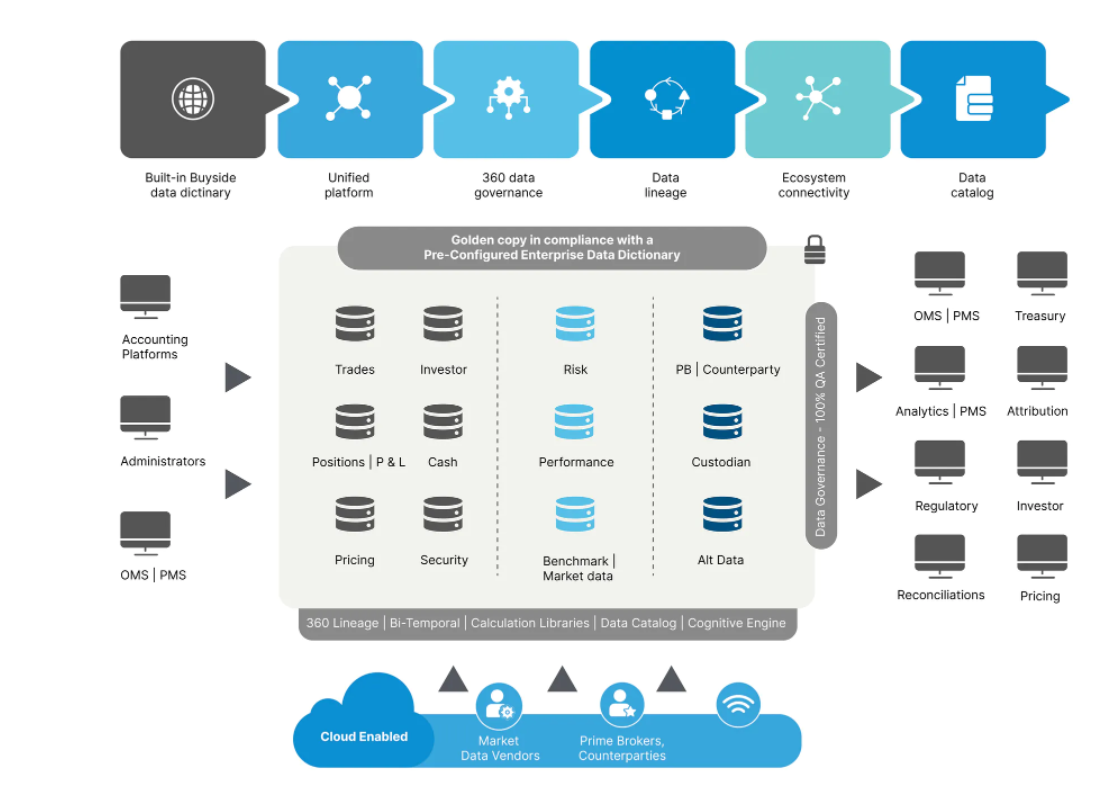
* Security and reference data is staged into different extracts.
* Any particular (ex: XYZ) feed is loaded into staging table.

**NORMALIZATION –**

* Security and reference normalized tables are made by performing respective loader task.
* With the help of Staging-Norm Map dictionary, Staging table of XYZ feed gets mapped with Sec-Ref norm tables into a XYZ normalized table.

**DENORMALIZATION –**

* Norm-Denorm Map Table is used to map XYZ Norm table and Sec-Ref norm tables to form a XYZ denormalized layer/view.
* The view is then further modified as per requirement and is then sent for analytics and reporting

**2.2 Procedures Adopted**

*Fig 2.2: Procedures Adopted*

* **Cloud Ready Solution**

Cloud-optimized scalable implementation allows firms to execute multiple loads in parallel and to support all major cloud services.

* **Data Governance**

Ensure data quality with comprehensive governance tools, including exception handling and workflow control.

* **Data Security**

Advanced user roles and privileges combined with SFTP supported tools ensure complete security.

**SFTP (SSH File Transfer Protocol)** is a secure file protocol that is used to access, manage, and transfer files over an encrypted SSH transport.

When compared with the traditional FTP protocol, SFTP offers all the functionality of FTP, but it is more secure and easier to configure.

Transferring Files with SFTP:

*Downloading File*

**sftp> get filename.zip**

**OUTPUT**

Fetching /home/remote\_username/filename.zip to filename.zip

/home/remote\_username/filename.zip 100% 24MB 1.8MB/s

*Uploading File*

**sftp> put filename.zip**

**OUTPUT**

Uploading filename.zip to /home/remote\_username/filename.zip

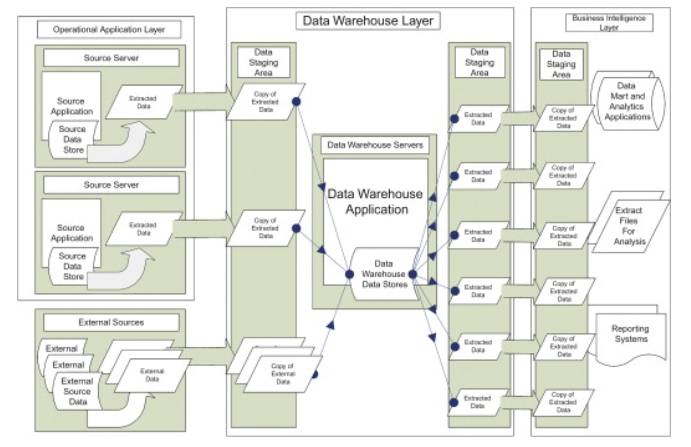
filename.zip 100% 12MB 1.7MB/s 00:06

**Chapter-3**

**ETL in Data Warehousing**

**3.1 What is Data Warehousing**

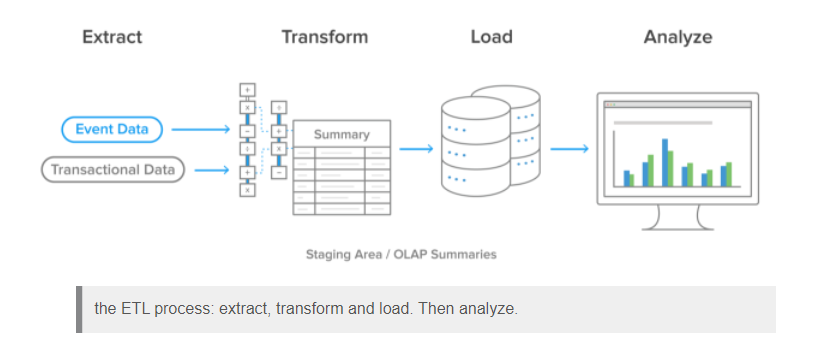
Data warehousing is the secure electronic storage of information by a business or other organization. The goal of data warehousing is to create a trove of historical data that can be retrieved and analyzed to provide useful insight into the organization's operations.

Data added to the warehouse do not change and cannot be altered. The warehouse is the source that is used to run analytics on past events, with a focus on changes over time. Warehoused data must be stored in a manner that is secure, reliable, easy to retrieve, and easy to manage.

*Fig 3.1: Data Warehouse Data Flow*

**3.2 ETL Process in Data Warehouses**

While the data warehouse acts as the storage place for all your data and BI tools serve as the mechanism that consumes the data to give you insights, **ETL** is the intermediary that pushes all of the data from your tech stack and customer tools into the data warehouse for analysis. The ETL phase is where businesses spend a good chunk of their time and energy when developing a warehouse solution.

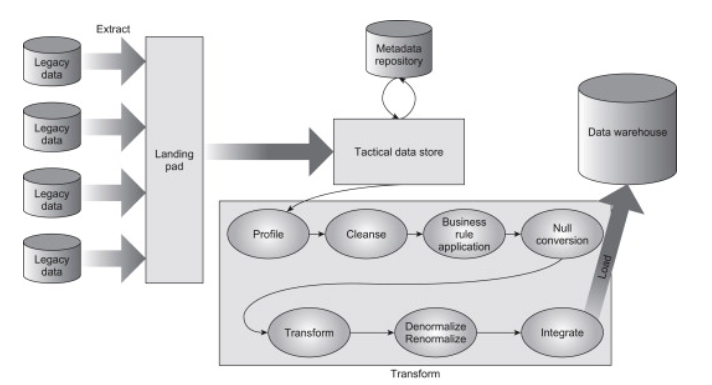
*Fig* *3.2: ETL Process*

The process of extracting data from source systems and bringing it into the data warehouse is commonly called ETL, which stands for extraction, transformation, and loading. ETL is the process by which data is extracted from data sources (that are not optimized for analytics), and moved to a central host (which is). Note that ETL refers to a broad process, and not three well-defined steps. The acronym ETL is perhaps too simplistic, because it omits the transportation phase and implies that each of the other phases of the process is distinct. Nevertheless, the entire process is known as ETL.

**3.3 Staging Architecture**

The first part of the ETL process is to assemble the infrastructure needed for aggregating the raw data sets and for the application of the transformation and the subsequent preparation of the data to be forwarded to the data warehouse. This is typically a combination of a hardware platform and appropriate management software that we refer to as the staging area.

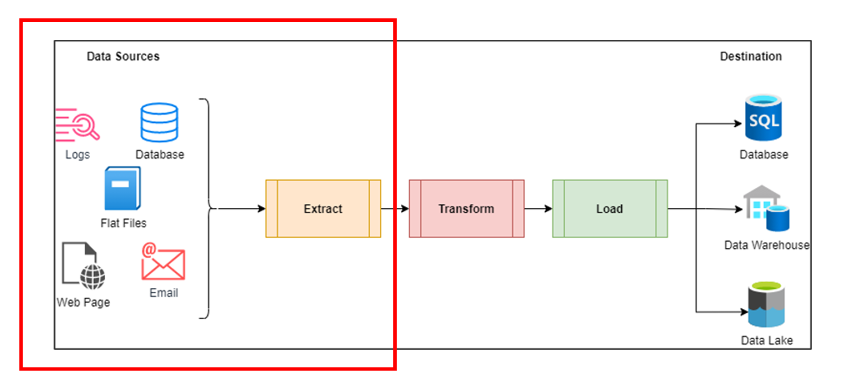
A staging area, or landing zone, is an intermediate storage area used for data processing during ETL process. The data staging area sits between the data source(s) and the data target(s), which are often data warehouses, data marts, or other data repositories.



*Fig 3.3: Staging data in preparation for loading into an analytical environment*.

**3.4 Data Extraction**

The first step of the ETL process is extraction. In this step, data from various source systems is extracted which can be in various formats like relational databases, No SQL, XML, and flat files into the **staging area**. It is important to extract the data from various source systems and store it into the staging area first and not directly into the data warehouse because the extracted data is in various formats and can be corrupted also. Hence loading it directly into the data warehouse may damage it and rollback will be much more difficult. Therefore, this is one of the most important steps of ETL process.

A logical data map is required to extract data from multiple data sources and load it into the Staging area. The extraction process can be full or partial extraction based on the business requirements.

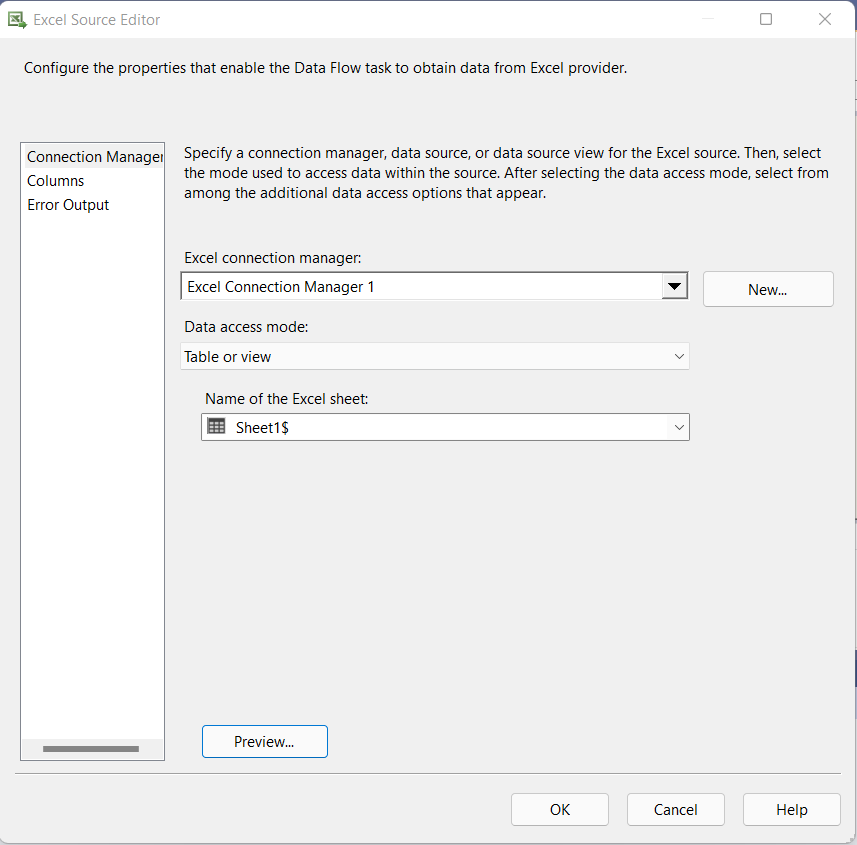
*Fig 3.4: Data Extraction*

**3.4.1 Data Extraction Methods**

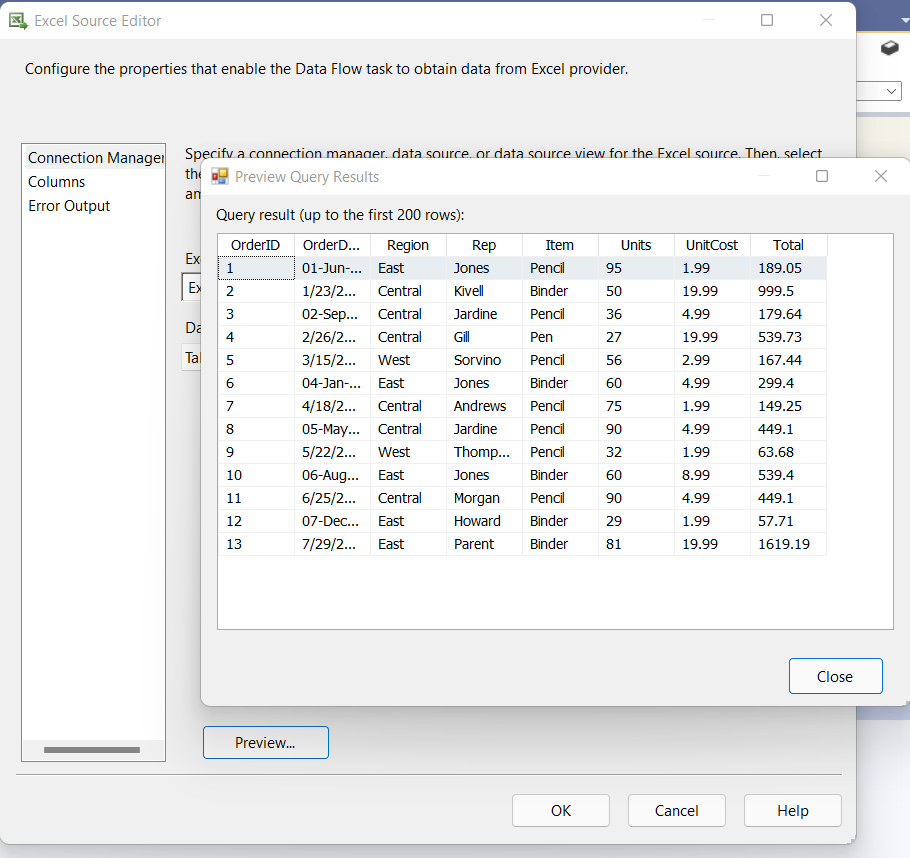
Depending on the chosen logical extraction method and the capabilities and restrictions on the source side, the extracted data can be physically extracted by two mechanisms. The data can either be extracted online from the source system or from an offline structure. Such an offline structure might already exist or it might be generated by an extraction routine.

There are the following methods of physical extraction:

* **Online Extraction –** The data is extracted directly from the source system itself.
* **Offline Extraction –** The data is not extracted directly from the source system but is staged explicitly outside the original source system. The data already has an existing structure, like flat files where data is in a defined, generic format.



*Fig 3.4.1: Offline Data Extraction*



*Fig 3.4.2: Preview of Data Extracted Offline*

Some validations are done during Extraction:

* Reconcile records with the source data
* Make sure that no spam/unwanted data loaded
* Data type check
* Remove all types of duplicate/fragmented data
* Check whether all the keys are in place or not

How data should be extracted may depend on the scale of the project, the number (and disparity) of data sources, and how far into the implementation the developers are. Extraction can be as simple as a collection of simple SQL queries, the use of adapters that connect to different originating sources, yet can be as complex as to require specially designed programs written in a proprietary programming language. There are tools available to help automate the process, although their quality (and corresponding price) varies widely.

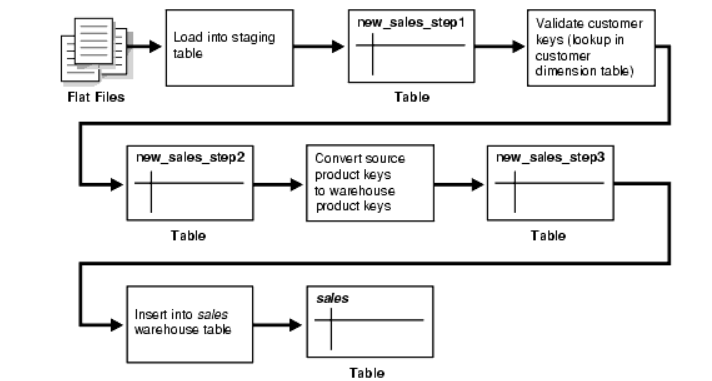
Automated extraction tools generally provide some kind of definition interface specifying the source of the data to be extracted and a destination for the extract, and they can work in one of two major ways, both of which involve code generation techniques. The first is to generate a program to be executed on the platform where the data is sourced to initiate a transfer of the data to the staging area. The other way is to generate an extraction program that can run on the staging platform that pulls the data from the source down to the staging area.

With the help of ETL tool you can simply do this process by configuring few basic blocks, wherein you can easily define the source, the structure in which you want your data to be staged and the destination where the staging data will reside.

**3.5 Data Transformation**

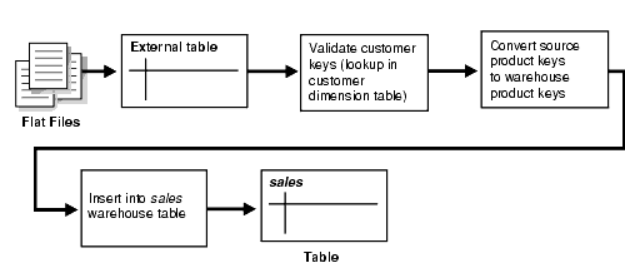
Transformation refers to the cleansing and aggregation that may need to happen to data to prepare it for analysis. Architecturally speaking, there are many ways to approach ETL transformation:

* **Multistage data transformation** – This is the classic extract, transform, load process. Extracted data is moved to a staging area where transformations occur prior to loading the data into the warehouse.



*Fig 3.5.1: Multistage Data Transformation*

* **In-warehouse data transformation** – In this approach, the process flow changes to something more like ELT. Data is extracted and loaded into the analytics warehouse, and transformations are done there.
* **Pipelined data transformation** – The new functionality renders some of the former necessary process steps obsolete while some others can be remodeled to enhance the data flow and the data transformation to become more scalable and non-interruptive. The task shifts from serial *transform-then-load* process or *load-then-transform process*, to an enhanced *transform-while-loading.*

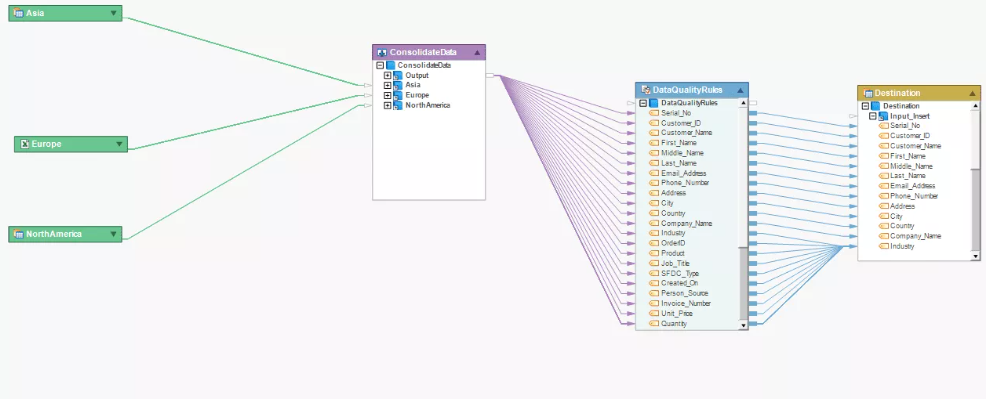


*Fig 3.5.2 Pipelined Data Transformation*

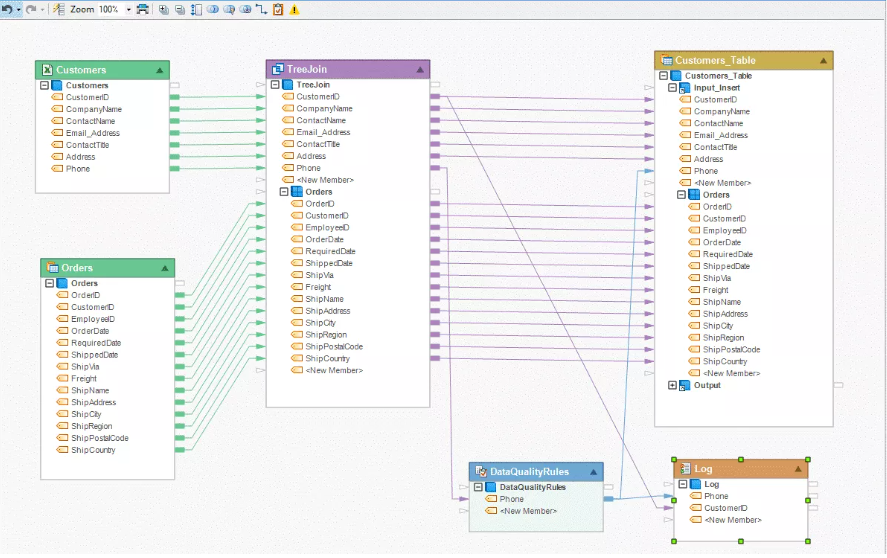
**3.5.1 Transformation Types**

* Data type conversion – This includes parsing strings representing integer and numeric values and transforming them into the proper representational form for the target machine, and converting physical value representations from one platform to another.
* Data cleansing: The data profiling rules along with directed actions that can be used to correct data that is known to be incorrect and where the corrections can be automated. This component also covers data-duplicate analysis and elimination and merge/purge. Mapping NULL to 0 or "Male" to "M" and "Female" to "F," date format consistency, etc.
* Integration: This includes exploiting the discovery of table and foreign keys for representing linkage between different tables, along with the generation of alternate (i.e., artificial) keys that are independent of any systemic business rules, mapping keys from one system to another, archiving data domains and codes that are mapped into those data domains, and maintaining the metadata (including full descriptions of code values and master key-lookup tables).
* Referential integrity checking: In relation to the foreign key relationships exposed through profiling or as documented through interaction with subject matter experts, this component checks that any referential integrity constraints are not violated and highlights any nonunique (supposed) key fields and any detected orphan foreign keys.
* Derivations: Any transformations based on business rules, new calculations, string manipulations, and such that need to be applied as the data moves from source to target are applied during the transformation stage. For example, a new “revenue” field might be constructed and populated as a function of “unit price” and “quantity sold.”
* Denormalization and renormalization: Frequently data that is in normalized form when it comes from the source system needs to be broken out into a denormalized form when dimensions are created in repository data tables. Conversely, data sourced from join extractions may be denormalized and may need to be renormalized before it is forwarded to the warehouse.
* Aggregation: Any aggregate information that is used for populating summaries or any cube dimensions can be performed at the staging area.
* Audit information: As a matter of reference for integrity checking, it is always useful to calculate some auditing information, such as row counts, table counts, column counts, and other tests, to make sure that what you have is what you wanted. In addition, some data augmentation can be done to attach provenance information, including source, time and date of extraction, and time and date of transformation.
* Filtering: Selecting only certain rows and/or columns
* Joining: Linking data from multiple sources

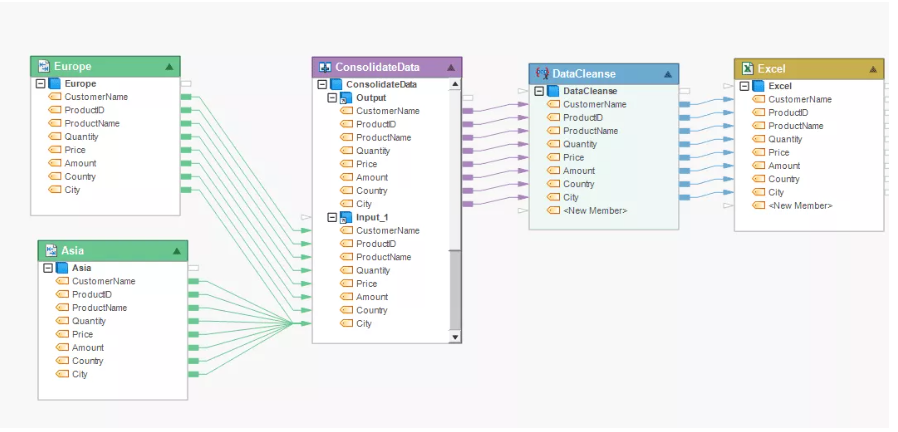
The screenshot below illustrates a data transformation use case, which combines source data from different sales regions and verifies it against a set of business rules.



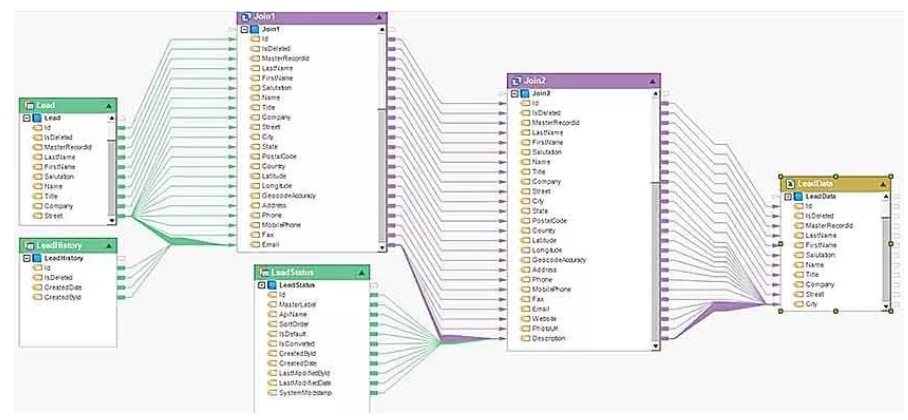
*Fig 3.5.3: Data Transformation*

The screenshot below illustrates a scenario in which data from two different sources is merged, verified against defined data quality rules, and loaded into the destination table.

*Fig 3.5.4:* Data Transformation for ETL workflow

****The screenshot below illustrates a scenario in which data is cleansed before being written into an Excel destination.

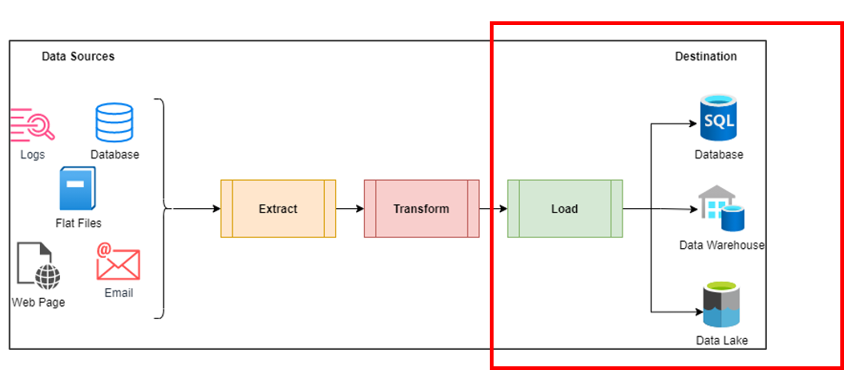
*Fig 3.5.5:* Data Cleansing for ETL workflow

The screenshot below shows an example of the join transformation from three distinct data sources in an ETL flow.

*Fig 3.5.6:* Data Mapping for ETL workflow

**3.6 Data Loading**

Loading data into the target Datawarehouse database is the last step of the ETL process. In a typical Data warehouse, huge volume of data needs to be loaded in a relatively short period (nights). Hence, load process should be optimized for performance.



*Fig 3.6.1: Data Loading*

In case of load failure, recover mechanisms should be configured to restart from the point of failure without data integrity loss. Data Warehouse admins need to monitor, resume, cancel loads as per prevailing server performance.

The loading component of ETL is centered on moving the transformed data into the data warehouse. The critical issues include the following:

* Target dependencies, such as where and on how many machines the repository lives, and the specifics of loading data into that platform.
* Refresh volume and frequency, such as whether the data warehouse is to be loaded on an incremental basis, whether data is forwarded to the repository as a result of triggered transaction events, or whether all the data is periodically loaded into the warehouse in the form of a full refresh.

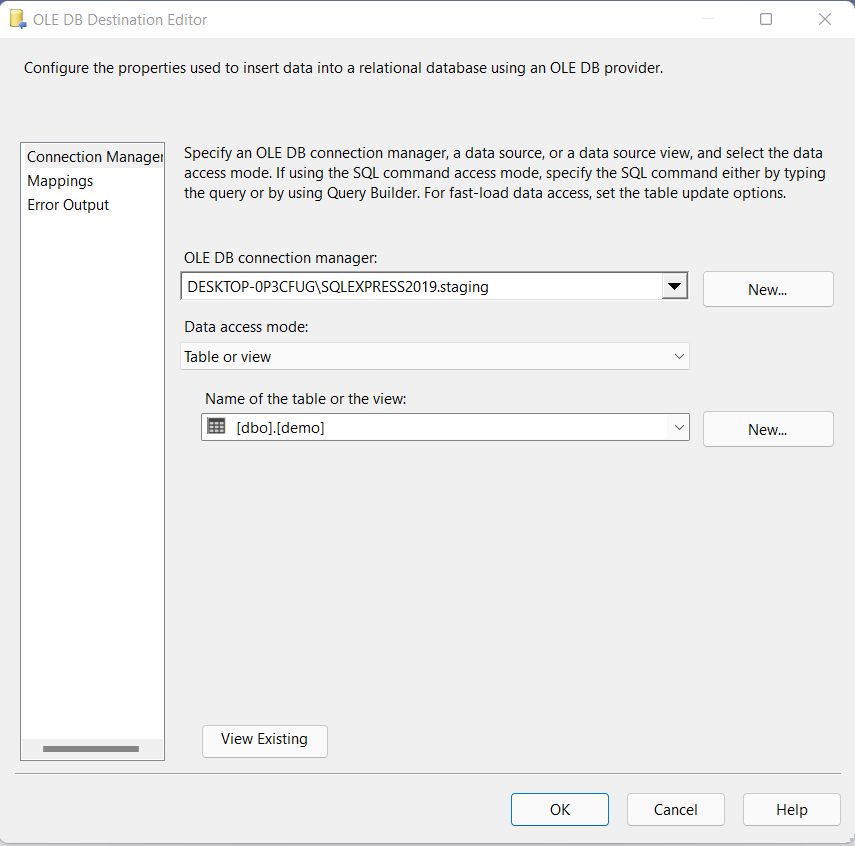
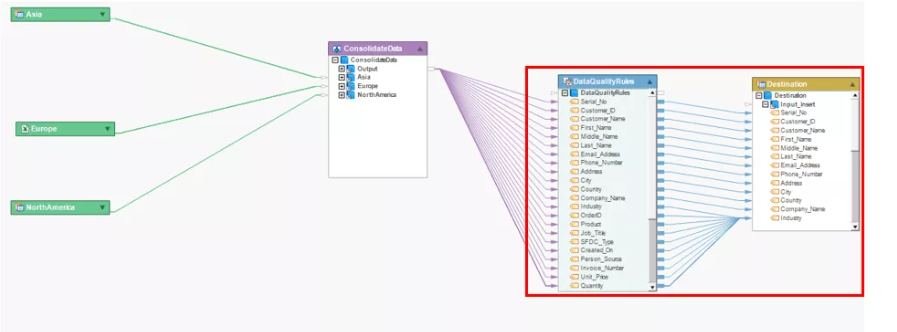
**Types of Loading:**

Initial Load — populating all the Data Warehouse tables.

Incremental Load — applying ongoing changes as when needed periodically.

Full Refresh — erasing the contents of one or more tables and reloading with fresh data.

Below screenshot is of the configuration screen for a destination block

*Fig 3.6.2: Data Loading*

*Fig 3.6.3: Data Loading in SQL Server DB Table*

The screenshot below shows how processed; high quality-data is being loaded into an SQL Server database table.

**Chapter – 4**

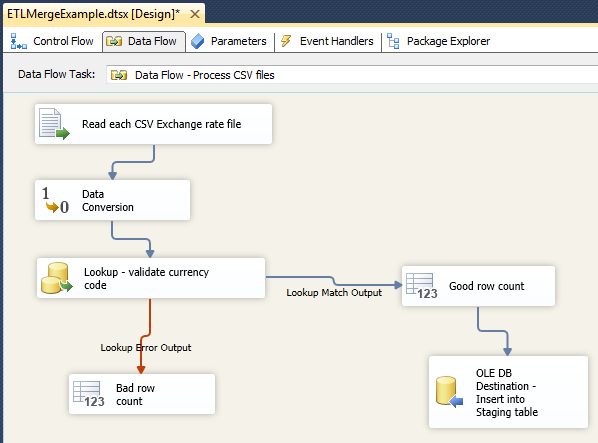
**The Role of Data Pipelines in the EDW**

A lot of effort goes into unlocking the true power of your data warehouse. Using a metadata-driven ETL approach, you can build low-latency data pipelines that are reliable and flexible.

A data warehouse is populated using data pipelines. They transport raw data from disparate sources to a centralized data warehouse for reporting and analytics. Along the way, the data is transformed and optimized.

However, the increase in volume, velocity, and variety has rendered the traditional approach to building data pipelines *—involving manual coding and reconfiguration*— ineffective and obsolete.

Automation is an integral part of building efficient data pipelines that can match the agility and speed of your business processes.

Below is the screenshot of a Data Pipeline which is extracting a csv file and performing data conversion which results into staging data. Then the staged data undergoes data validation by performing mapping against valid data. Further the data gets loaded into a SQL table.

*Fig 4.1: Data Pipeline performing extraction and Loading*

**4.1 Data Pipeline Automation**

You can seamlessly transport data from source to visualization through data pipeline automation. It is a modern approach to populating data warehouses that requires designing functional and efficient dataflows.

As we all know, timeliness is one of the crucial elements of high-quality business intelligence — and automated data pipelines help you make data available in the data warehouse as quickly as possible.

Leveraging the power of automated and scalable data pipelines, you can eliminate obsolete, trivial, or duplicated data, maximizing data accessibility and consistency to ensure high-quality analytics.

With a metadata-driven ETL process, you can seamlessly integrate new sources into your architecture and support iterative cycles to fast-track your BI reporting and analysis.

Also, you can follow the ELT approach, where the data is loaded directly to the warehouse, so you can leverage the compute capacity of the destination system to carry out transformations efficiently.

**4.2 Optimizing Data Pipelines**

An enterprise must focus on building automated data pipelines that can dynamically adapt to changing circumstances, for instance, adding and removing data sources or changing transformations.

Of course, moving entire databases when you need data for reporting or analysis can be highly inefficient.

The best practice is to load data incrementally using change data capture to populate your data warehouse. It helps eliminate redundancy and ensures maximum data accuracy.

Other essential capabilities needed to create automated data pipelines are incremental loading, job monitoring, and job scheduling.

* Incremental loading ensures you don’t have to copy all the data to your data warehouse every time there’s a change at the source table to ensure your data warehouse is always accurate and up-to-date.
* Job monitoring helps you understand any issues with your current system and allow you to make any necessary changes to optimize the process.
* Job scheduling allows you to process your data daily, weekly, monthly, or only when specific triggers or conditions are met to streamline the process.

Orchestrating and automating your data pipelines can eliminate manual work, introduce reproducibility, and maximize efficiency.

**Results & Discussions**

ETL as a concept is here to stay, we will always need to extract and transform data. What might change is where this concept manifests or is implemented, in traditional ETL tools or in newer analytical tools.

Having said that – the power and agility of analytical tools like Tableau, PowerBI, Looker, etc. depend on how easy it is to consume and interpret the underlying data. ETL tools are usually used to transform data from its raw complex (normalized) form to an easier-to-use (denormalized) form.

Technically it might be possible to do these transforms in any layer, ETL or presentation/visualization. The main trade-off is performance i.e., the more complex the queries that tools such as Tableau run, the slower they perform. In addition, you need to assess if the data changes so frequently that you need to run these transforms every time an end-user queries the data.

Example, if the end-users only look at daily data (not intra-day), then it probably makes sense to do all the heavy lifting in a nightly ETL batch and then let the analytical tools run simpler and faster queries, instead of doing the complex transforms every time an end-user looks at a report.

There are other considerations too, such as code re-usability, version control and governance which are relatively easier in an ETL layer controlled by IT than in the analytical tools controlled by users in business functions usually not trained in version control, etc.

Advances in data speed, infrastructure, and data processing have done more to shape the future of ETL than perhaps any other factor. After all, these advances laid the foundation for the shift toward cloud-storage and the arrival of big data. Consider the evolution of the internet itself: when the world wide web was invented in 1989 (this is an abbreviated history!), few people had access to the internet. In 1995, there were only 16M users worldwide, and dial-up internet speeds topped out at 56 kbps. The early 2000s brought the emergence of fiber optic networks and dramatic improvements in data transfer speeds. Today, there are over 4 billion internet users across the globe and the fastest average connection speed has grown to 28.6 mbps. While that might seem impressive enough, Google Fiber now boasts a connection speed of one gigabit per second.

Along with improved internet speeds and the explosion of the number of internet users, advances in programming and data architecture are also impacting the future of ETL. With the birth of Apache Hadoop in 2011, the average organization gained access to a fast, dependable framework for distributed computing. This allowed powerful processors—which previously sat mostly idle—to share in the work of processing large data jobs. The results were significant improvements in speed, capacity, and reliability. As the Hadoop framework grew, more and more companies reduced their dependence on expensive onsite servers in favor of distributed computing clusters or, as they are often collectively referred to, the cloud.

In 2013, Apache introduced Spark, a real-time big data analytics technology that could process tasks at up to 100 times the speed of Hadoop. This made near real-time ETL widely accessible and changed the way industry professionals approached data analytics and business intelligence.

Today, ETL processes are handling vast amounts of data at incredible speeds. ETL has evolved in other ways too; ETL can now scale in tandem with the ebb and flow of web traffic, and many cloud service providers charge only for the actual ETL processing time used. The result is ETL that is flexible, fast, and cost-effective.

**Chapter – 5**

**Summary & Conclusions**

Data extraction, transformation and loading (ETL) is a popular technique used in data integration i.e., gathering data from various operational sources into the data warehouse. Due to its vital role in data integration, ETL is receiving an increasing attention in research and practice, yet fewer studies have synthesized existing studies in ETL to define future research direction that is robust and sustainable.

Businesses today are more complex and almost exclusively digital. That means they need to incorporate the latest technology, such as BI tools and connections, to stay on top of business growth and customer needs. Using the top ETL solutions for 2022 is an excellent way to maximize capabilities and instantly access real-time data stored in data warehouses and then transfer that information directly to connected parties

Modern applications and working methodology require real-time data for processing purposes and in order to satisfy this purpose, there are various ETL tools available in the market.

Using such databases and ETL tools makes the data management task much easier and simultaneously improves data warehousing.

ETL platforms that are available in the market save money as well as time to a great extent. Some of them are commercial, licensed tools and few are open-source free tools.

There are many ETL tools on the market, representing the top performers for data management in 2022. Consider a leading data integration tool to help you manage your big data daily business and gain better insights for teams across several departments. There are options for those with more technical knowledge and capabilities and those who want a simple no-code solution. ETL is an easier way to move data with better security and features.

**Chapter – 6**

**Future Scope**

In the past, ETL processes were executed locally or on-site. In other words, ETL was managed in a facility in close proximity to the physical location where the data would ultimately be used or stored. Today, ETL processes are increasingly migrating away from centralized data centers and toward systems that run partially or completely in the cloud. The movement toward cloud-native storage and processing is itself the result of advances in technology, increased efforts to prevent data loss, faster internet speeds, and cybersecurity threats.

The trend toward cloud-native storage and processing isn’t the only factor transforming the ETL process. The proliferation of device connectivity, improvements in processes that collect and store information, and the Internet of Things (IoT) have all resulted in a big data boom. As our data set grows larger and the number of data sources continues to multiply, companies are increasingly dependent on data for maintaining their competitive advantage.

As a result of both of these factors, ETL must now be able to accommodate more data from more sources more quickly than ever before. In many cases, ETL must also be able to handle streaming data, which means it must process data as it is generated in real-time. ETL tools are also evolving in response to the kinds of data that are now available, so that companies are able to process data effectively and mine it for business intelligence and actionable insights, no matter where it comes from.

With so many changes taking place in the data landscape, it can be difficult to know which ETL tools make the most sense. The good news is that ETL tools have continued to evolve in order to meet changing business needs, and the right platform will provide the flexibility and adaptability to manage your data today and tomorrow. So, whether you rely on a data warehouse or a data lake, there is an effective ETL solution.

**Appendix**

Data Lakes and ETL

Before we tackle the effect of data lakes on ETL, it might be helpful to spend some time discussing data lakes in general. As with all things data-driven, the way data is collected and stored continues to change. In the past, companies have primarily relied on data warehouses for storing, reporting, and analyzing data. In more simple terms, data warehouses are systems which contain current and historical data that has been processed and standardized. The warehouse is the central location from which all data is retrieved.

In contrast, data lakes are repositories of data in a more fluid sense (pun inevitable). Data lakes store both raw and transformed data, from a variety of sources, in any virtually any format. More complex and adaptable than data warehouses, data lakes offer companies the capacity for storing data in any form for use at any time. For example, data lakes contain:

* Unstructured data — data that does not comply to any standardized format
* Semi-structured data — data stored in its own loose format, but tagged with identifiers that make it accessible in structured environments
* Structured data — data pre-organized to comply with an expected and explicitly defined layout.

So as far as ETL is concerned, what do the differences between data warehouses and data lakes mean? The original ETL processes were developed under the data warehouse model, in which data was structured and organized systematically. But some ETL technologies have adapted to the emergence of data lakes and are now called ELT. That’s right. The “extract, transform, load” approach has become the “extract, load, transform” approach. In other words, when it comes to data lakes, the process has to be changed up a bit. Instead of transforming data before it reaches its final destination, different types of data are collected from multiple sources and delivered to a location, so that they can then be transformed.

As more organizations move to the data lake storage solution, ETL is in some cases being eclipsed by its cousin ELT. But that doesn’t mean that ETL is going away. In fact, ETL continues to play a vital role in data migration and integration. Either process may be appropriate, depending on the company, the data, and the situation.

|  |  |
| --- | --- |
| **Material Type** | **Works Cited** |
| Research Paper | <https://www.researchgate.net/publication/326533253_Big_Data_ETL_Implementation_Approaches_A_Systematic_Literature_Review> |
| Research Paper | <https://www.researchgate.net/publication/353051990_A_Systematic_Literature_Review_on_Big_Data_Extraction_Transformation_and_Loading_ETL> |
| Article | <https://www.matillion.com/resources/blog/the-importance-of-etl-tools-in-data-warehousing> |
| Article | <https://www.snowflake.com/guides/what-etl> |
| Article | <https://www.investopedia.com/terms/d/data-warehousing.asp#:~:text=Data%20warehousing%20is%20the%20secure,insight%20into%20the%20organization's%20operations>. |
| Article | <https://www.guru99.com/etl-extract-load-process.html#3> |
| Article | <https://www.astera.com/type/blog/what-is-data-warehousing/> |
| Article | <https://www.geeksforgeeks.org/etl-process-in-data-warehouse/> |
| Article | <https://www.integrate.io/blog/etl-data-warehousing-explained-etl-tool-basics/#whatisetl> |
| Article | <https://docs.oracle.com/cd/B19306_01/server.102/b14223/ettover.htm> |
| Article | <https://www.stitchdata.com/etldatabase/etl-extract> |
| Article | <https://www.sciencedirect.com/topics/computer-science/data-staging-area> |
| Blog | <https://www.bi.wygroup.net/digital-transformation/what-is-the-future-of-etl-tools-considering-agility-of-new-analytical-tools/> |

**Bibliography**