A data warehouse architecture proposal and ETL analysis: a case study of an Albanian banking system

Abstract

The increasing phenomenon of information overload is a direct result of the ongoing trend to reduce the cost of data distribution while the development of data processing platforms is not fast enough. Therefore, sending packets of data is not a big issue, but processing an increasing amount of data is a big challenge. Data Warehouse is a concept born in the late '80s and has been improved decade after decade with the growth of technology. Despite many application areas, the data warehouse concept is still new and unknown to Albanian businesses and companies with a large amount of data to store and analyze. This paper aims to provide an in-depth analysis of the concept of a data warehouse, the different architectures applied, and the Extract, Transform Load process. The ETL process as a concept consists of extracting the data from a source, transforming them into different formats or dimensions, and then loading them somewhere to be checked for analysis or other purposes. In this paper, we can find different definitions of the data warehouse, where it can be implemented, the different architectures according to the needs of the users, and how the ETL process works.

Keywords: Data; data warehouse; extract transform load (ETL) process; management information systems.

1. INTRODUCTION

Data collection and circulation have become a central component of increasingly more sectors of modern capitalism and economics [1]. A Data Warehouse is a place where historical data are stored and copied from older systems to store and analyze them [2]. By this, everyone who uses the system can do queries to get the information needed for the company and process the data for decision-making. “The value of better understanding is better decision-making” [2,3].

By integrating different data sources, the data warehouse provides a complete and accurate view of our organization's operational data. This data is aggregated into a set of strategic indicators or measurements that can be analyzed by the axis or dimension of interest. Therefore, data warehousing allows us to create business reports quickly and efficiently from operational data sources [4]. Currently, most data warehouses are implemented in a single database instance, but it is becoming more and more common to find larger and more complex data stores. Implemented in multiple logically linked databases as part of a data warehouse environment [5]. A data warehouse can also contain multiple front-end applications, depending on the needs of the user community [6]. The use of CPU and GPU also remains an important issue as expressed in [7] “Even when the data size is large, it is sometimes better to use the CPU-GPU heterogeneous query plan rather than only the GPU plan since some relational operations prefer the CPU over the GPU.”

An important process, being introduced in this paper is called Extract-Transform-Load (ETL) and extracts raw data from various sources, transforms them into a format that supports the analysis to be performed, and then loads it into the data warehouse. ETL procedure is an important step during data warehousing (DW) testing. This is almost the most complicated stage as it directly affects the data quality and retrieval speed [8-10].

1.1. Literature Review

The use of appropriate methodologies is very important for academic research as it plays an important role in providing the necessary assessments and suggestions.

The concept of the Data Warehouse was first mentioned by Bill Inmon, an American computer scientist who is known by everyone as the Father of Data Warehousing. It started in the 1970s as a need to improve the existing tools such as DSS (Decision Support System) since the database models were not always being optimized for in-depth analysis or complex reporting of the data. According to Bill Inmon “A Data Warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management’s decision-making process” [11]. Its characteristics of it can be mentioned as Integration, Subject-Oriented, Time-Variant, and Non-Volatile. Vincent Rainardi relates “A data warehouse (DW) is a system that retrieves and consolidates data periodically from the source systems into a dimensional or normalized data store. It usually keeps years of history and is queried for business intelligence or other analytical activities. It is typically updated in batches, not every time a transaction happens in the source system” [12].

An important process that can be mentioned in the data warehouse environment is the ETL process [12]. According to Ralph Kimball, The Extract-Transform-Load (ETL) system is the foundation of the data warehouse. A properly designed ETL system extracts data from the source systems, enforces data quality and consistency standards, conforms data so that separate sources can be used together, and finally delivers data in a presentation-ready format so that application developers can build applications and end users can make decisions [13]. ETL processes are the backbone component of a data warehouse since they supply the data warehouse with the necessary integrated and reconciled data from heterogeneous and distributed data sources. However, the ETL process development, and particularly its design phase, is still perceived as a time-consuming task. This is mainly because ETL processes are typically designed by considering a specific technology from the very beginning of the development process [4]. But as stated in [14] “nowadays, many graphical user interfaces (GUI)-based solutions are available to facilitate the ETL processes.” Different opportunities, challenges, and proposed models for the ETL process can also be found in [15] and [9].

1.1.1. Data Warehouse Architecture Properties Analysis

With decades of development, DW and its architecture not only focus on data analysis and decision-making but also meet new demands with unprecedented rapid updates in science and technology [16]. The following architectural properties are essential for a data warehouse system [17] separation, scalability, extensibility, and administrability. Different DW architectures can also be mentioned, such as Structure-Oriented, an architecture where the most common idea in any DW project is that the data is available in one or more data sources and is useful information to help decision-makers make decisions based on past behavior of the system. It is flexible and powerful and may be suitable for use by small and medium-sized DW developers who do not have a standardized or comprehensive DW testing framework [18]. When designing a general-purpose system, it is difficult to know in advance which data structures will be used to optimize their language and architecture. This does not apply to task-oriented architectures, architectures tailored to the execution of a particular well-defined problem [19]. Single-layer architectures are rarely used in practice. The goal is to minimize the amount of data stored. To achieve this goal, we need to eliminate data redundancy. This means that the data warehouse is implemented as a multidimensional view of operational data created by middleware or intermediate processing layers [20]. While this architecture can meet, integration and data accuracy requirements, it cannot log more data than the source. For these reasons, virtual approaches to data warehousing are only successful when the need for analysis is particularly limited and the amount of data to analyze is huge [21,22].

In the Two-Layer Architecture, the isolation requirements play a fundamental role in defining the general architecture of a data warehousing system. Usually referred to as a two-layer architecture to emphasize the separation of physically available sources and data warehouses, it consists of four consecutive layers of data flow [23]. A two-phase commit divides the normal commit process into two parts. First, distributed servers communicate with each other until all servers have expressed their willingness to commit a share of the transaction. Then everyone commits and announces the rest of the successes or failures [24,25]. The Three-Layer Architecture has the third layer an aligned data layer or operational data store. This layer embodies operational data acquired after source data integration and cleansing. As a result, this data is integrated, consistent, accurate, up-to-date, and detailed. The main advantage of the aligned data layer is that it creates a reference data model that is common to the entire enterprise. At the same time, it separates the source data extraction and integration issues [26] from the data warehouse filling issues. Matching layers may be used directly to better perform operational tasks. However, the tuned data increases the redundancy of the production source data.

Data Warehousing applications are widely used in banking as being one of the most important fields. The huge amount of data that this sector has for the customers, employees, products, etc. By this, the bankers can manage all their available resources more effectively to better analyze the data to facilitate better decision-making. Many banks in Albania use the concept of data warehousing to manage all the required tasks, and one of them is Tirana Bank where we had the opportunity to learn more about data warehouse, how it works, why is it important, and how it analyzes the data, create reports and help all other departments of the bank to make the job easier and safe for every task. Same as in the banking sector, data warehousing helps the financial industry to analyze every financial activity of the company analyze customer expenses, analyze profits, and costs of the company, and analyze the risk, fraud, and many other aspects of a company. Data warehousing in the education sector helps the school or the faculty to store historical data of the students, such as their data, grades, and other information that needs to be stored and analyzed for different purposes. Another important use of a data warehouse is in the Healthcare sector. Every hospital, pharmacy, and clinic has a huge amount of data on their patients to be stored and processed. Another interesting use of the ETL process is in the widening field of B0lockchain as explained in [27].

1.1.2. Extraction,Transformation,andLoadingProcessing

ETL, as explained earlier in this paper stands for Extract, Transform, and Load and represents the process of retrieving and transforming data from the source system and putting it into the data warehouse [12]. A well-designed ETL system pulls data from source systems, enforces data quality and consistency standards, conforms data so that different sources may be combined, and finally distributes data in a presentation-ready manner so that application developers and end-users can make choices [13]. Data warehouse tables are updated increasingly often as the need for real-time data warehouse results develops, and the time frame for conducting ETL processes is shortening (in minutes or seconds) [28]. Data warehousing and ETL maintenance work is driven not just by efficiency, but also by concerns for data freshness. The ETL's significance is here from the definition of its capability and the improvement effort.

ETL is chargeable for fetching the records from the heterogeneous assets' structures into the DW so every failure inside the ETL capability ends in loading wrong records in DW, which in flip ends in offering managers wrong records that main to faulty decisions [29]. At the highest level, the ETL team's job is to construct the data warehouse's back room. The ETL system must deliver data to end-user tools in the most efficient way possible. In the cleaning and conforming processes, provide value to data. There are many ETL tools on the market, but small businesses typically use hand-coded ETL routines to extract and integrate data [30]. The data lineage should be safeguarded and documented. The back room in practically every data warehouse must support four critical steps [13] such as data extraction from primary sources; data cleansing and quality assurance; establishing uniformity among the sources, the labels and metrics in the data must be aligned and provide data in a physical format that query tools, report writers, and dashboards may access.

The first step of ETL is data extraction, in which the data from the source system is first accessed and further processed to process and extract the required values. The whole process should be done with minimal resources. Having said that, the best extraction strategy is one that keeps the source system's performance and response time unchanged. The cleanup is carried out according to data standardization rules, by placing unique identifiers, e.g., phone numbers, conversion of emails to standard forms, and validation of address fields [11]. The extraction process analyzes the required data and retrieves it from one or more different sources, such as database systems and applications. The size of the extracted data can always vary and can range from hundreds of kilobytes to gigabytes. This depends entirely on the source system and the business situation.

The main goal of the extraction process is to pull all the required data from the source system using as few resources as possible. The extraction process should be designed so that it does not adversely affect the source system in terms of performance, response time, or locks of any kind. Data might originate from transactional applications, such as Salesforce's CRM data SAP's ERP data, or

Internet of Things (IoT) sensors that collect readings from a manufacturing line or factory floor activity, for example. Extraction often entails merging data from these numerous sources into a single data collection and then verifying the data by flagging or removing any erroneous data. Relational databases, XML, JSON, and other formats may be used to extract data [31]. There are two extraction methods: The logical extraction method and the physical extraction method. There are three types of logical extraction methods (Update Notification, Incremental Extract, and Full Extraction) and two types of physical extraction methods (Online Extraction and Offline Extraction).

From the perspective of who moves the data from the source system, the ETL methods can be categorized into four approaches [12]. The ETL process extracts data by polling the source system's database regularly. This is the most common approach. ETL connects to the database on the source system, queries the data, and retrieves the data. The Source system database triggers transfer data changes. A database trigger is a collection of SQL statements that are executed each time a table is inserted, updated, or deleted. We can use triggers to save the changed rows in a separate table. Scheduled processes in the source system export data regularly. This is like the first approach, but the program that queries the database is not an external ETL program. Instead, it is an internal exporter that runs on the source system server. The log reader reads the database log file to identify data changes. The database log file contains a record of transactions made to that database. A log reader is a program that understands the format of the data in a log file. Read the log file, retrieve the data, and save the data to another location.

By transformation, the data is converted into a usable format that can be easily stored in the data warehouse system. The conversion process involves applying calculations, DML (data manipulation language) operations, joins, constraints, primary keys, and foreign keys to the data. Some data does not require a transformation that can be pushed directly into the data warehouse. This data is called direct move or pass-through data. The data transformation process is incompatible with data correction, data cleansing, erroneous duplicate data removal, incomplete data formation and data error correction, data integrity, and before loading into the data warehouse system. It also includes the format of the data [32].

Independent of the presence of a reconciled data layer, establishing a mapping between the source data layer and the data warehouse layer is generally made difficult by the presence of many different, heterogeneous sources especially when dealing with big data, a complex integration phase is required when designing our data warehouse [33,34]. In the study of [25] it is "shown that, despite the many solutions that have been proposed in the literature, the issue of data integration in a big data environment still arises." The following points must be rectified in this phase: Loose texts may hide valuable information. Different formats can be used for individual data. For example, a date can be saved as a string or as three integers. The following points are the main transformation processes aimed at populating the reconciled data layer; conversion and normalization that operate on both storage formats and units of measure to make data uniform; matching that associates equivalent fields in different sources and selection that reduces the number of source fields and records. When populating a data warehouse, normalization is replaced by de-normalization because data warehouse data are typically de-normalized, and you need aggregation, to sum up, data properly [21].

Loading data into the desired multidimensional structure is the last ETL phase. The retrieved and converted data is then stored in dimensional structures that end-users and application systems may access. Both loading dimension tables and loading fact tables are included in the loading process. It's critical to make sure the load is completed correctly and with as few resources as feasible. The main aim is a Load process database. To help the load process go smoothly, we deactivate all restrictions and indexes before starting the load and re-enable them immediately once the load is finished. The ETL procedure should maintain referential integrity to ensure consistency [24]. Loading may be performed in two ways: Refresh, where the Data warehouse is rewritten. This approach that older statistics is replaced. Refresh is usually utilized in a mixture with static extraction to start with populating a statistics warehouse. The other is the Update, where only the adjustments carried out to supply statistics are brought to the statistics warehouse. The update is usually performed without deleting or enhancing preexisting statistics. This method is utilized in a mixture with incremental extraction to replace statistics warehouses regularly [21].

1.1.3. ETLtestingandstandardization

Testing ETL (extract, transform, load) techniques is an important and vital phase of data warehousing (DW) testing. This is almost the most complex phase as it directly affects the data quality [35]. Automated testing is a valuable tool for improving the quality of DW systems, but because the manual testing process is time and inaccurate, automated testing requires less time and cost for data quality (DQ). The testing procedure might be difficult due to the large amount of frequently involved data. ETL testing differs from traditional software testing because it focuses on data rather than code. Because each source's data might have a different data type, testing must be able to accept heterogeneous data types. Furthermore, because the data in the source and destination systems are frequently in different formats, it is harder to match them with each other, group them, and do all the comparisons needed to load the data for different purposes such as analyzing or creating views, building reports, etc [36].

Some problems might happen during the ETL Testing such as data may be missing during the ETL process; the database may contain incorrect, insufficient, or repeated data; it is very difficult to do an ETL test on the target system when the data storage system contains real data, so the data size may be too large; it is robust for developing and designing test cases when the data set is very large and complicated; ETL testers are unaware of the requirements for outputting consumer schemas and business data; ETL tests involve several complex SQL concepts for data validation in the target system; a destination assignment information. Testers usually have no idea of the source and when developing and testing a process, attention is paid to the maximum delay [11].

Implementation of standardization for large ETL projects is essential to establish standards at an early stage. Without standards, developers write inconsistent ETL jobs and incur exorbitant maintenance overhead on existing code. The ETL team needs to standardize the development method to provide a consistent and maintainable code environment. The following list contains areas of the ETL process that require the most standardization [13]. The criteria to consider are generating surrogate keys, looking up keys, and applying default values.

1.2. Purpose of study

This paper aims to bring a detailed analysis of a data warehouse, different architectures applied, specific characteristics that differentiate data warehouse from other methods applied in the market, and areas where we can implement it. Also in the paper, we are going to be explaining ETL processing as a procedure to use to be more effective and efficient in analyzing the data and for other purposes. After creating a data warehouse in MySQL Microsoft, which will contain two main databases, one that will serve as a source and the other as a destination, we will make the connection with Visual Studio 2019. Through the SSIS (SQL server integration services), package, we will perform the ETL process.

With this implementation, we managed to get an effective answer to the problem that we are trying to solve and suggest many companies, banks, financial institutions, etc., implement a data warehouse and develop ETL processing to have more productivity and increase its values.

2. METHODS AND MATERIALS

The methodology used in this study is deductive research as we have analyzed scientific studies by other authors. The literature used in this work is secondary literature. Secondary sources include books, documents, research papers, scientific research, and many publications by various authors. The scientific method also consists of applied research of a case study by creating a data warehouse and developing an ETL process in the banking system of a second-level bank in Albania.

3. RESULTS

Implementing the Data Warehouse and ETL processing we choose MySQL Microsoft. First, we create the Database SOURCE_DB (Fig 1) that contains the following tables: The SOURCE_DB will contain all the data saved for the customers that are added each time a new customer comes to the bank and gets one of the products or services that the bank offers. This database will be used for the ETL processing that will extract the data from the source database, will transform them into the needed format, and will be loaded in another database that is created called TARGET_DB. After creating the Database SOURCE_DB with all the tables explained above, we build the TARGET_DB Database (Fig 2) which will contain a table that will be used by a user (which may be, for example, the Department of Cards in the Bank). Until now we have built two databases, one of them will serve as a SOURCE for the ETL processing and the second one will serve as a destination of the procedure.

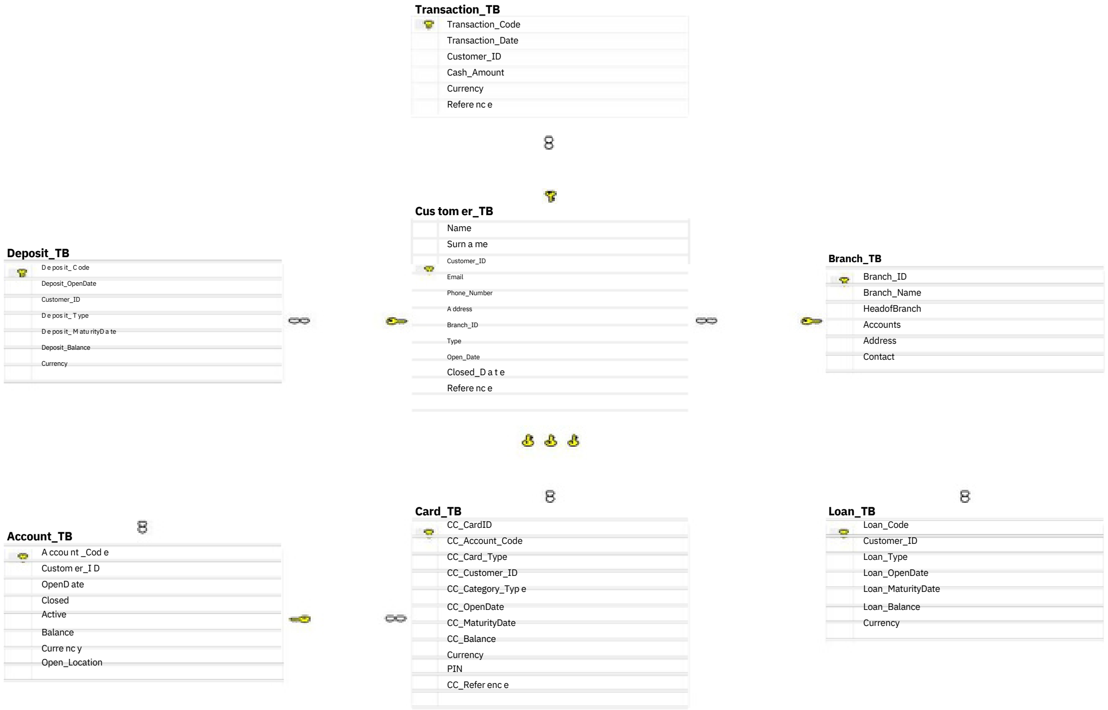
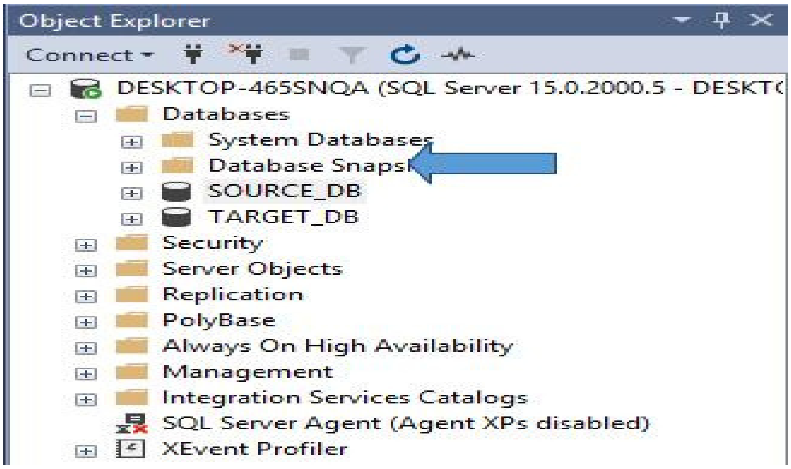


Fig 1. The diagram from the database created SOURCE_DB
Source: Authors



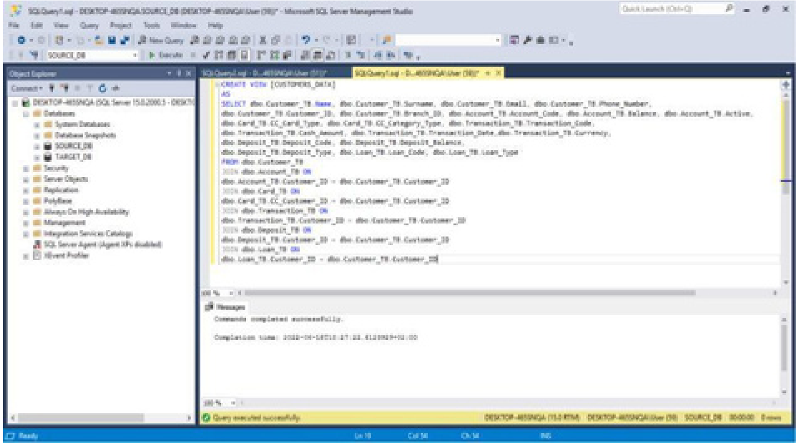
Object Explorer

Connect

DESKTOP-465SNQA (SQL Server 15.0.2000.5 - DESKT(
- Databases
  - System Databases
  - Database Snapsh
  - SOURCE_DB
  - TARGET_DB
- Security
- Server Objects
- Replication
- PolyBase
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)
- XEvent Profiler

Fig 2. Source and Destination Databases
Source: Authors

As mentioned above ETL stands for Extract, Transform, Load. The first step is to extract data from the SOURCE_DB and for this, we create a view (Fig 3) that contains most of the data that are stored in the database. To create the view, we must write a SQL code with the specific requirements, and it is stored in the SOURCE_DB.



```
CREATE VIEW [CUSTOMERS_DATA]
AS
SELECT dbo.Customer_TB.Name, dbo.Customer_TB.Surname, dbo.Customer_TB.Email, dbo.Customer_TB.Phone_Number,
dbo.Customer_TB.Customer_ID, dbo.Customer_TB.Branch_ID, dbo.Account_TB.Account_Code, dbo.Account_TB.Balance, dbo.Account_TB.Active,
dbo.Card_TB.CC_Card_Type, dbo.Card_TB.CC_Category_Type, dbo.Transaction_TB.Transaction_Code,
dbo.Transaction_TB.Cash_Amount, dbo.Transaction_TB.Transaction_Date,dbo.Transaction_TB.Currency,
dbo.Deposit_TB.Deposit_Code, dbo.Deposit_TB.Deposit_Balance,
dbo.Deposit_TB.Deposit_Type, dbo.Loan_TB.Loan_Code, dbo.Loan_TB.Loan_Type
FROM dbo.Customer_TB
JOIN dbo.Account_TB ON
dbo.Account_TB.Customer_ID = dbo.Customer_TB.Customer_ID
JOIN dbo.Card_TB ON
dbo.Card_TB.CC_Customer_ID = dbo.Customer_TB.Customer_ID
JOIN dbo.Transaction_TB ON
dbo.Transaction_TB.Customer_ID = dbo.Customer_TB.Customer_ID
JOIN dbo.Deposit_TB ON
dbo.Deposit_TB.Customer_ID = dbo.Customer_TB.Customer_ID
JOIN dbo.Loan_TB ON
dbo.Loan_TB.Customer_ID = dbo.Customer_TB.Customer_ID
```

```
Commands completed successfully.

Completion time: 2022-06-13T10:17:22.4120919+02:00
```

Fig 3. Creating the View
Source: Authors

In the TARGET_DB the table created is ETL_Transactions which contains all the columns that are needed to be accessed by a specific user. As we see the table is empty and needs to be filled with data. Now the ETL processing starts. To perform the ETL process we use Visual Studio Microsoft 2019. After downloading the needed packages or templates such as SSIS let's start the process. A Data Flow Task is created, and it's called an ETL Transaction (Fig 4). This will tell from where the data comes, where the destination is, and what will happen with the data. The source from where we will get the data is an OLE DB Source. We configure from where the data will come and connect the server with MySQL Microsoft. Since the Extracting step of the ETL is finished, the second thing to do is to transform the data in the needed format, with all the requirements that the user has. The Department of Cards wants every month a view of the customers with the specific columns: Name, Surname, Email, Branch_ID, Customer_ID, Transaction_Code, Transaction_Date, Cash _Amount, and Currency for analysis. The third and last step of the ETL process is loading the data to a specific destination we want. Above we created a specific table that will serve as a destination for data flow. All the transforming data will be loaded into the ETL Transactions table (Fig 5). Mapping of the tables is very important because the connection between them must be as accurate as possible so that there is not any error between the columns when the procedure is executed (Fig 6). Since all the configurations are made, we click the Start button to execute all the processes. As we see all the data is extracted from the source, transformed into the needed format, and loaded to the destination we wanted. The process was SUCCESSFUL (Fig 7). We go to the TARGET_DB and check for the results. All the data is transformed and loaded in the table, and it is ready to be used for the needed purposes such as analyzing, etc (Fig 8).

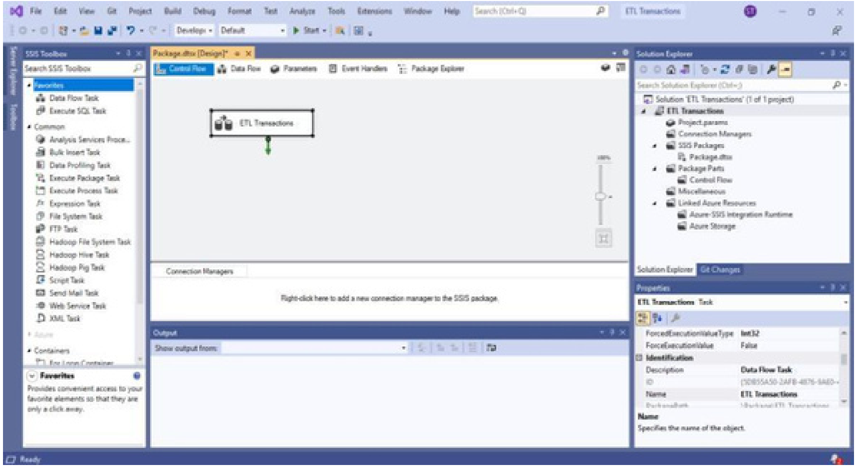
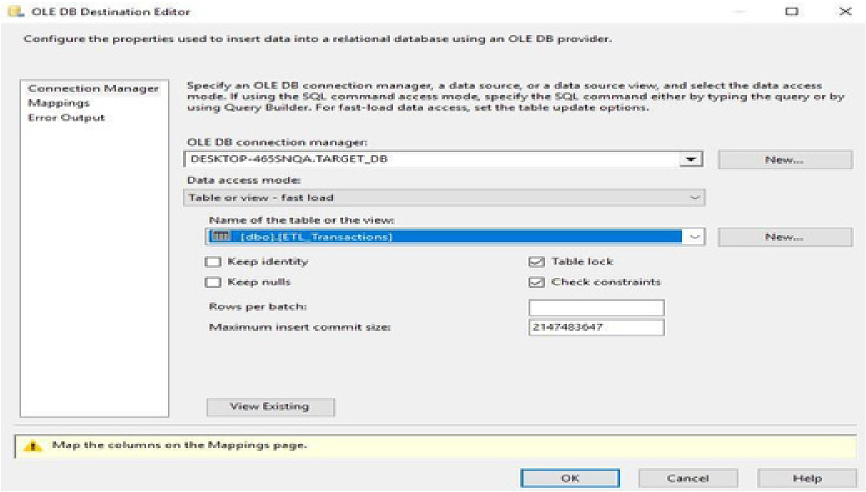


Fig 4. Data Flow Task
Source: Authors



OLE DB Destination Editor

Configure the properties used to insert data into a relational database using an OLE DB provider.

Connection Manager
Mappings
Error Output

Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.

OLE DB connection manager:
DESKTOP-465SNQA.TARGET_DB

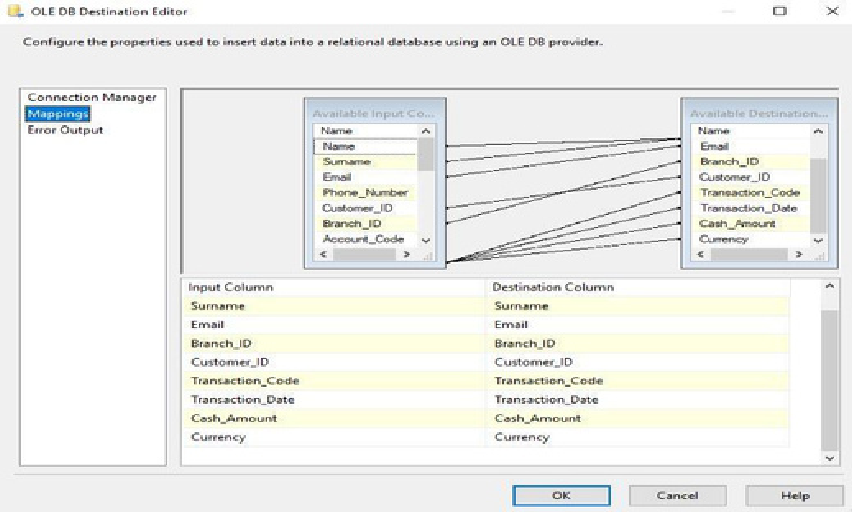
Data access mode:
Table or view - fast load

Name of the table or the view:
[dbo].[ETL_Transactions]

□ Keep identity
☑ Table lock
□ Keep nulls
☑ Check constraints
Rows per batch:
Maximum insert commit size: 2147483647

Map the columns on the Mappings page.

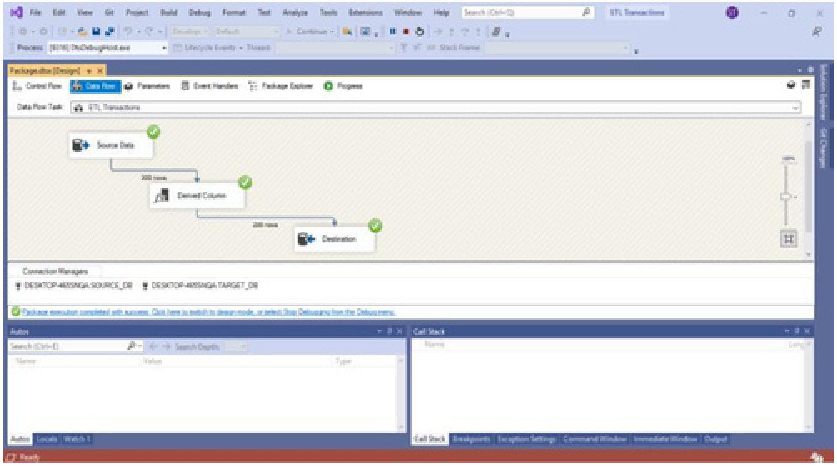
Fig 5. Configuring the server with the TARGET_DB
Source: Authors



Configure the properties used to insert data into a relational database using an OLE DB provider.

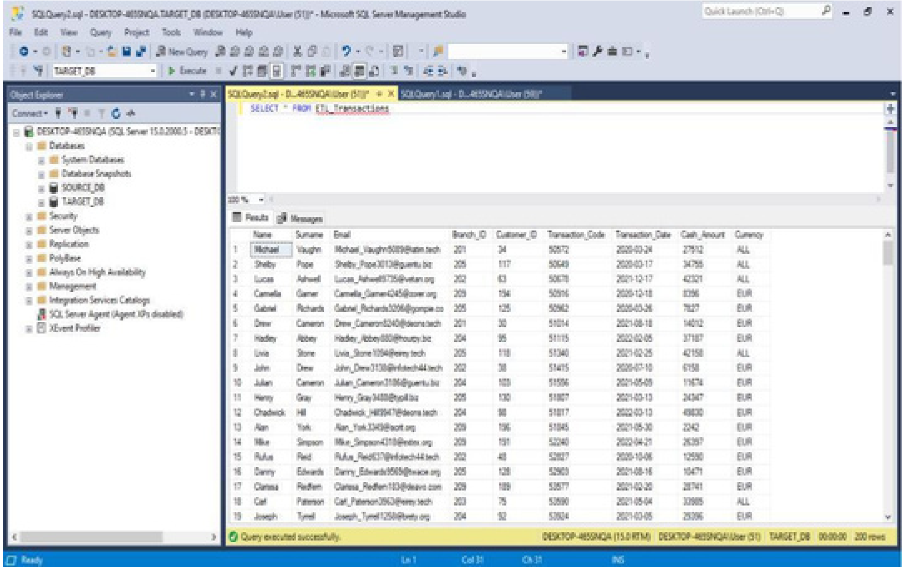
Input Column	Destination Column
Surname	Surname
Email	Email
Branch_ID	Branch_ID
Customer_ID	Customer_ID
Transaction_Code	Transaction_Code
Transaction_Date	Transaction_Date
Cash_Amount	Cash_Amount
Currency	Currency

Fig 6. Mapping of the Source and Destination Tables
Source: Authors



Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

Fig 7. Executing the ETL process
Source: Authors


SQLQuery2.sql - DESKTOP-4655NQA.TARGET_DB (DESKTOP-4655NQA\User (51))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
TARGET_DB
Execute
Object Explorer
Connect
DESKTOP-4655NQA (SQL Server 15.0.2000.5 - DESKTO
Databases
System Databases
Database Snapshots
SOURCE_DB
TARGET_DB
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
XEvent Profiler
SQLQuery2.sql - D...4655NQA\User (51))*
SQLQuery1.sql - D...4655NQA\User (59))*
Results
Messages
Ready
Ln 1
Col 31
Ch 31
INS

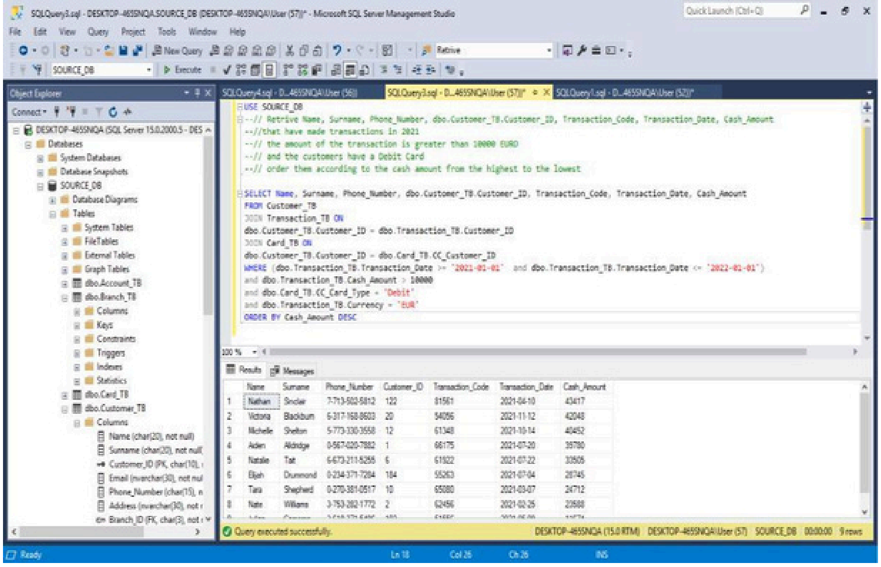
```
SELECT * FROM ETL_Transactions
```

	Name	Surname	Email	Branch_ID	Customer_ID	Transaction_Code	Transaction_Date	Cash_Amount	Currency
1	Michael	Vaughn	Michael_Vaughn5089@latin.tech	201	34	50572	2020-03-24	27512	ALL
2	Shelby	Pope	Shelby_Pope3013@quertu.biz	205	117	50649	2020-03-17	34798	ALL
3	Lucas	Ashwell	Lucas_Ashwell5735@vetan.org	202	63	50678	2021-12-17	42321	ALL
4	Camella	Garner	Camella_Garner4245@zover.org	209	194	50916	2020-12-18	8356	EUR
5	Gabriel	Richards	Gabriel_Richards3206@gompie.co	205	125	50962	2020-03-26	7827	EUR
6	Drew	Cameron	Drew_Cameron8240@deons.tech	201	30	51014	2021-08-18	14012	EUR
7	Hadley	Abbey	Hadley_Abbey000@housey.biz	204	95	51115	2022-02-05	37187	EUR
8	Livia	Stone	Livia_Stone1094@famy.tech	205	118	51340	2021-02-25	42158	ALL
9	John	Drew	John_Drew3130@infotech44.tech	202	38	51415	2020-07-10	6158	EUR
10	Julian	Cameron	Julian_Cameron3106@guertu.biz	204	103	51556	2021-05-09	11674	EUR
11	Henry	Gray	Henry_Gray3488@typ8.biz	205	130	51807	2021-03-13	24347	EUR
12	Chadwick	Hill	Chadwick_Hill9947@deons.tech	204	98	51817	2022-03-13	49830	EUR
13	Alan	York	Alan_York3349@acort.org	209	196	51845	2021-05-30	2242	EUR
14	Mike	Simpson	Mike_Simpson4318@extex.org	209	191	52240	2022-04-21	26357	EUR
15	Rufus	Reid	Rufus_Reid637@infotech44.tech	202	48	52827	2020-10-06	12550	EUR
16	Danny	Edwards	Danny_Edwards8585@twace.org	205	128	52903	2021-08-16	10471	EUR
17	Clarissa	Redfern	Clarissa_Redfern183@deavo.com	209	189	52877	2021-02-20	28741	EUR
18	Carl	Paterson	Carl_Paterson3563@famy.tech	203	75	52890	2021-05-04	32985	ALL
19	Joseph	Tyrrell	Joseph_Tyrrell1258@brety.org	204	52	52924	2021-03-05	29396	EUR

```
Query executed successfully.    DESKTOP-4655NQA (15.0 RTM)   DESKTOP-4655NQA\User (51)   TARGET_DB   00:00:00   200 rows
```

Fig 8. Checking the results
Source: Authors

Not every requirement from different departments of the bank is done periodically. In other words, there are some reports that the Head of the Department of Budgeting wants it only once. For this, we don't have to create an ETL process, but we can solve it by writing a query directly to the data warehouse (Fig 9). To make it more understandable a requirement comes from an employee who wants some data on customers who apply some conditions that he wants, to analyze what is going wrong or how to increase the profits and the value of the bank in the market. The step-by-step implementation and ETL processing can be followed in the tables of the Annexes area.


SQLQuery3.sql - DESKTOP-4655NQA.SOURCE_DB (DESKTOP-4655NQA\User (57))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Retrive
SOURCE_DB
Execute
Object Explorer
Connect
DESKTOP-4655NQA (SQL Server 15.0.2000.5 - DES
Databases
System Databases
Database Snapshots
SOURCE_DB
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Account_TB
dbo.Branch_TB
Columns
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Card_TB
dbo.Customer_TB
Columns
Name (char(20), not null)
Surname (char(20), not null)
Customer_ID (PK, char(10), ...
Email (nvarchar(30), not null)
Phone_Number (char(15), n...
Address (nvarchar(30), not ...
Branch_ID (FK, char(3), not ...
SQLQuery4.sql - D...4655NQA\User (56))
SQLQuery3.sql - D...4655NQA\User (57))*
SQLQuery1.sql - D...4655NQA\User (52))*
Results
Messages
Ready
Ln 18
Col 26
Ch 26
INS

```
USE SOURCE_DB
--// Retrive Name, Surname, Phone_Number, dbo.Customer_TB.Customer_ID, Transaction_Code, Transaction_Date, Cash_Amount
--//that have made transactions in 2021
--// the amount of the transaction is greater than 10000 EURO
--// and the customers have a Debit Card
--// order them according to the cash amount from the highest to the lowest

SELECT Name, Surname, Phone_Number, dbo.Customer_TB.Customer_ID, Transaction_Code, Transaction_Date, Cash_Amount
FROM Customer_TB
JOIN Transaction_TB ON
dbo.Customer_TB.Customer_ID = dbo.Transaction_TB.Customer_ID
JOIN Card_TB ON
dbo.Customer_TB.Customer_ID = dbo.Card_TB.CC_Customer_ID
WHERE (dbo.Transaction_TB.Transaction_Date >= '2021-01-01'  and dbo.Transaction_TB.Transaction_Date <= '2022-01-01')
and dbo.Transaction_TB.Cash_Amount > 10000
and dbo.Card_TB.CC_Card_Type = 'Debit'
and dbo.Transaction_TB.Currency = 'EUR'
ORDER BY Cash_Amount DESC
```

	Name	Surname	Phone_Number	Customer_ID	Transaction_Code	Transaction_Date	Cash_Amount
1	Nathan	Sinclair	7-713-502-5812	122	81561	2021-04-10	43417
2	Victoria	Blackburn	6-317-168-8603	20	54056	2021-11-12	42048
3	Michelle	Shelton	5-773-330-3558	12	61348	2021-10-14	40452
4	Aiden	Aldridge	0-567-020-7882	1	66175	2021-07-20	35780
5	Natalie	Tait	6-673-211-5258	6	61922	2021-07-22	33505
6	Elijah	Drummond	0-234-371-7204	184	55263	2021-07-04	28745
7	Tara	Shepherd	0-270-381-0517	10	65080	2021-03-07	24712
8	Nate	Williams	3-753-282-1772	2	62456	2021-02-25	23588

```
Query executed successfully.    DESKTOP-4655NQA (15.0 RTM)   DESKTOP-4655NQA\User (57)   SOURCE_DB   00:00:00   9 rows
```

Fig 9. Query Results
Source: Authors

4. CONCLUSION

The development of the data warehouse concept decade after decade has been growing and becoming more sophisticated due to the new technologies that are constantly being created. In this scientific paper, we analyzed the data warehouse as a necessary system to be used more widely in the Albanian banking market due to all the functions and advantages it brings.

This paper aims to introduce, mainly banking businesses or large organizations in more detail, the implementation of a data warehouse, its features, and which type of architecture to choose based on each company's needs. Also, it analyzed the concept of ETL as a procedure to be used more in a data warehouse because it is considered the backbone component of a data warehouse. It provides the data warehouse with the required integrated and tuned data from heterogeneous distributed data sources. This is primarily because ETL processes are typically designed with specific technologies in mind from the beginning of the development process. During this research, it is recommended who should implement this system, why they should use it, and the advantages and disadvantages it brings to the company. We would like to recommend the process we chose to implement a data warehouse in the banking system, because of an experience with one of the biggest banks here in Albania.