Hellenic Mediterranean University

DATA WAREHOUSING IN HIGHER EDUCATION

A CASE STUDY OF THE HELLENIC MEDITERRANEAN UNIVERSITY

MBA

MASTER OF BUSINESS ADMINISTRATION

OURANIA SMYRNAKI

THESIS SUPERVISOR: VASSILAKIS KOSTAS

Heraklion, Crete

2020

Data warehousing in higher education

A case study of the Hellenic Mediterranean University

Ourania Smyrnaki

Master of Business Administration

Hellenic Mediterranean University of Crete

Abstract

Business Information systems, such as ERP, CRM, MIS, etc., serve different needs in management, in customer relations, in the supply chain, in production monitoring etc. Data extracted from heterogeneous data sources can be exploited by Business Intelligence (BI) systems to support useful decision-making processes in business (industry). Advanced systems, such as ETL (Extract, Transform, Load) tools, are used for the collection of this data in Data Warehouses (DWs), thereby facilitating the processes of decision-making in Decision Support Systems (DSS).

The information systems of the Hellenic Mediterranean University support the daily operations of the Institution such as management, financial, education etc. and collect heterogeneous data that are often used by members of the academic community and for various purposes. However, for effective decision making, the integration of this data that exist in various data sources, is required. The Institution will benefit by drawing useful conclusions that will help in making the right decisions. This will become a useful information service for the Quality Assurance Unit and the educational leadership of the Institute.

In this thesis, a literature review was conducted regarding the application of BI technologies in education. BI systems are used in universities for supporting the transfer of knowledge in the society which is the principal aim of the universities. Integration of different data sources from the online learning system with the usage of BI techniques, is also conducted. An ETL tool, which enables the integration of data in a faster and cost-effective way, is used. For the manipulation of data, ETL processes were designed, implemented and executed. These ETL processes can be imported to other external applications and run independently. As a result, input data from heterogeneous sources were mapped to a common schema. Research indicates that the use of a BI technique can be applied to a larger scale of information systems of the

Hellenic Mediterranean University in order to integrate data effectively for supporting decision-making processes.

Αποθήκες Δεδομένων στην Ανώτατη Εκπαίδευση

Μελέτη περίπτωσης στο Ελληνικό Μεσογειακό Πανεπιστήμιο

Ουρανία Σμυρνάκη

Μεταπτυχιακή Εργασία

Ελληνικό Μεσογειακό Πανεπιστήμιο Κρήτης

Περίληψη

Τα πληροφοριακά συστήματα (ERP, CRM, MIS κ.ά.) των επιχειρήσεων εξυπηρετούν διάφορες ανάγκες όπως είναι η διοίκηση, η οικονομική διαχείριση, οι πελατειακές σχέσεις, η εφοδιαστική αλυσίδα, η παρακολούθηση της παραγωγής κλπ. Για την λήψη αποφάσεων συχνά απαιτείται ο συνδυασμός πολλών ετερογενών στοιχείων που υπάρχουν σε διάφορες βάσεις δεδομένων της επιχείρησης και μπορούν ν' αξιοποιηθούν από συστήματα επιχειρηματικής ευφυΐας (BI) για να εξαχθούν χρήσιμα συμπεράσματα. Για την συλλογή αυτών των στοιχείων χρησιμοποιούνται ειδικά συστήματα εξαγωγής, μετασχηματισμού και φόρτωσης των δεδομένων (Extract, Transform, Load -ETL) σε μια Αποθήκη Δεδομένων (Data Warehouse) διευκολύνοντας με αυτό τον τρόπο τις διαδικασίες ενός συστήματος λήψης αποφάσεων (Decision Support Systems).

Τα πληροφοριακά συστήματα του Ελληνικού Μεσογειακού Πανεπιστημίου εξυπηρετούν διάφορες ανάγκες (διοίκηση, οικονομική διαχείριση, εκπαίδευση), συλλέγουν στοιχεία από ποικίλες πηγές τα οποία συχνά είναι ετερογενή και αξιοποιούνται από διαφορετικά μέλη της ακαδημαϊκής κοινότητας και για διαφορετικούς σκοπούς. Για την λήψη αποφάσεων όμως απαιτείται ο συνδυασμός όλων των στοιχείων που υπάρχουν στις διάφορες βάσεις δεδομένων του Ιδρύματος για να εξαχθούν χρήσιμα συμπεράσματα που θα βοηθήσουν στη ορθή λήψη αποφάσεων. Αυτό θα αποτελέσει μια χρήσιμη υπηρεσία πληροφόρησης για την ηγεσία του Ιδρύματος και την Μονάδα Διασφάλισης Ποιότητας του Ιδρύματος.

Στην παρούσα διατριβή, διεξήχθη βιβλιογραφική ανασκόπηση σχετικά με την εφαρμογή των τεχνολογιών επιχειρηματικής ευφυΐας (BI) στην ανώτατη εκπαίδευση. Τα συστήματα BI μετουσιώνουν την παραγόμενη πληροφορία σε γνώση, διευκολύνοντας την διαδικασία λήψης αποφάσεων από τα διοικητικά στελέχη του Πανεπιστημίου. Για την εφαρμογή τεχνικών επιχειρηματικής ευφυΐας,

μελετήθηκε ο τρόπος ενσωμάτωσης δεδομένων που προέρχονται από βάσεις δεδομένων διαφόρων πληροφοριακών συστημάτων που λειτουργούν στο Πανεπιστήμιο. Συγκεκριμένα, χρησιμοποιήθηκε ETL εργαλείο που επιτρέπει την ενσωμάτωση δεδομένων με τον ταχύτερο και οικονομικότερο τρόπο. Ως συνέπεια, σχεδιάστηκαν και υλοποιήθηκαν διαδικασίες εξαγωγής, μετασχηματισμού και φόρτωσης δεδομένων (ETL Jobs), για την ενοποίηση των ετερογενών δεδομένων. Αυτές οι ETL διαδικασίες μπορούν να εισαχθούν σε άλλα εξωτερικά πληροφοριακά συστήματα και να εκτελούνται ανεξάρτητα. Ως αποτέλεσμα, τα δεδομένα εισόδου από ετερογενείς πηγές χαρτογραφήθηκαν σε ένα κοινό σχήμα. Η έρευνα έδειξε ότι η χρήση μιας τεχνικής BI μπορεί να εφαρμοστεί σε μια ευρύτερη κλίμακα πληροφοριακών συστημάτων του Ελληνικού Μεσογειακού Πανεπιστημίου για να βελτιωθεί η λήψη αποφάσεων από τα διοικητικά στελέχη του Πανεπιστημίου.

ΕΥΧΑΡΙΣΤΙΕΣ

Στην παρούσα διατριβή, θα ήθελα να ευχαριστήσω τον επιβλέποντα Καθηγητή μου κ. Κώστα Βασιλάκη για την καθοδήγηση, την αμέριστη συμπαράστασή του και την συνεχή αρωγή του κατά την διάρκεια εκπόνησης της μεταπτυχιακής μου εργασίας στο Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών «Οργάνωση & Διοίκηση για Μηχανικούς» (ΜΒΑ) του Ελληνικού Μεσογειακού Πανεπιστημίου Κρήτης.

Επίσης, ευχαριστώ τους Καθηγητές κ. Εμμανουήλ Δρακάκη και κ. Μάρκο Κουργιαντάκη για την προθυμία, τη διάθεση και τη θετική ανταπόκρισή τους να είναι μέλη της τριμελούς επιτροπής αξιολόγησης.

Ευχαριστώ θερμά την οικογένειά μου και το ευρύτερο φιλικό μου περιβάλλον για την εμπιστοσύνη, την ενθάρρυνση και την στήριξη τους κατά την διάρκεια των μεταπτυχιακών μου σπουδών.

Contents

List of Abbreviations

1NF	First Normal Form
2NF	Second Normal Form
3NF	Third Normal Form
BCNF	Boyce-Codd Normal Form
BI	Business Intelligence
CRM	Customer Relationship Management
CSV	Comma-Separated Values File
DB	Database
DM	Data Mart
DW	Data Warehouse
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load
GUI	Graphical User Interface
JAR	Java Archive
JSON	JavaScript Object Notation
NF	Normal Form
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
PDF	Portable Document Format
RDB	Relational Database
RDBMS	Relational Database Management System
RTF	Rich Text Format
SCM	Supply Chain Management
SQL	Structured Query Language
XML	Extensible Markup Language

List of Figures

Introduction

The main challenge for businesses, nowadays, is data analysis. Discovering and exploiting useful information can be achieved with the use of Business Intelligence (BI) techniques. Business Intelligence entails technologies, tools and processes used for transforming data into information and information into knowledge for optimizing business decisions (Eckerson, 2007). Data warehousing, data integration and reporting tools play a critical role in increasing a business' success.

A Data warehouse stores large volumes of data which are gathered from heterogeneous sources and integrated with the help of ETL. ETL is a three-step process (extracting-transforming-loading) that emphasizes on improving the quality of data. During an ETL process, data is extracted from multiple heterogeneous sources. It is transformed and modified according to the organization's needs. The transformation process entails data cleaning, integration and aggregation. Finally, the modified data is loaded into the DW (Vaisman & Zimányi, 2014; Kimball & Ross, 2013; Rahm & Do, 2000; Kimball & Ross, 2002).

Many higher educational institutes have implemented ETL processes and other Business Intelligence techniques. As a result, Business Intelligence has contributed to the assessment of learning in the universities.

The aim of this thesis is the proposal of a Data Warehouse solution for the Hellenic Mediterranean University. Business Intelligence technologies, such as ETL, were used for the integration of different data sources from the online learning system of the university. ETL processes were designed, implemented and executed. As a result, the research conducted proved that Business Intelligence techniques can be applied to a larger scale of information systems in the Hellenic Mediterranean University.

The thesis consists of five chapters.
In the first chapter, a reference is made to Data Warehouse design and definitions. Specifically, the architecture of a Data Warehouse and its components are analyzed, in detail. For better understanding the principles of it, a comprehensive introduction to relational databases concepts, is provided.
The literature review, regarding the application of Business Intelligence technologies in higher education, is discussed in the second chapter. Many higher educational institutes have proposed Data Warehouse solutions for improving decision-making and enabling effective strategic planning in universities.

The third chapter describes the importance of using ETL tools and how they make integration processes faster and cost-effective.

The fourth chapter contains the design and implementation of ETL processes used for the integration of data among heterogeneous data sources in the Hellenic Mediterranean University.

The conclusion and future work are discussed in the last chapter.

1. Data Warehouse

1.1 Introduction

Nowadays, most enterprises have integrated Informatics in their everyday activities. Technology has evolved in an integral and essential part of an organization's operation. Business data is processed digitally, and it's stored in OLTP (Online Transaction Processing) systems. OLTP systems support the daily operations of a business enterprise and store transactions such as sales, orders, reservations, customers' information, financing etc. Although OLTP systems play a vital role for the successful operations of an enterprise, these systems are commonly used for transactions and query processing and not for decision making. The data stored in OLTP systems is big and difficult to be analyzed. There is a need for systems that can analyze and aggregate data from different sources. A Data Warehouse emphasizes on storing large volumes of data and on improving decision making.

1.2 Definition

Bill Inmon who is considered the father of the data warehouse, defined a data warehouse (DW) as "a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions." (Inmon, 2002, p. 31). An explanation of the four features is presented below:

- **Subject oriented:** Data Warehouse is providing information for specific subjects of analysis, such as sales, suppliers, customers etc. (Singhal, 2007).
- **Integrated:** The data warehouse integrates data from multiple heterogeneous sources.
- **Nonvolatile**: Data entered in the data warehouse are not allowed to be modified or removed. Since data has its own lifecycle, it is purged from the data warehouse when needed (Malinowski & Zimányi, 2009).
- **Time-variant:** A data warehouse stores historical information over the last years or months. E.g. someone can retrieve Information related to sales evolution from the last three, six months or years (Vaisman & Zimányi, 2014).

Based on the definition of Ralph Kimball, "A data warehouse is a system that extracts, cleans, conforms, and delivers source data into a dimensional data store and then supports and implements querying and analysis for the purpose of decision making" (Kimball & Caserta, The Data Warehouse ETL Toolkit:Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data, 2004, p. 23). Data warehousing is a process that involves extraction of data from transactional and legacy database systems and transforms it into organized information for decision making.

1.3 Architecture of Data Warehouse

In Figure 1.1, a typical architecture of a DW is presented based on the research done on the topic (Guitart & Conesa, 2015; Vaisman & Zimányi, 2014; Aquila, Tria, Lefons, & Tangorra, 2008). The architecture proposed in Figure 1.1 is general. Depending on the needs of an organization, the architecture of a DW can vary.

As seen in Figure 1-1, the architecture consists of three tiers. The first tier involves the heterogeneous data sources which will be extracted, transformed and loaded into the DW. The DW tier contains the Data Warehouse, Data Marts and the Metadata. The third tier contains the tools needed for the analysis of data stored in the DW. These tools enable reporting, data mining and Online Analytical Processing (OLAP). In the next sections, a detailed description of the above components, is given.

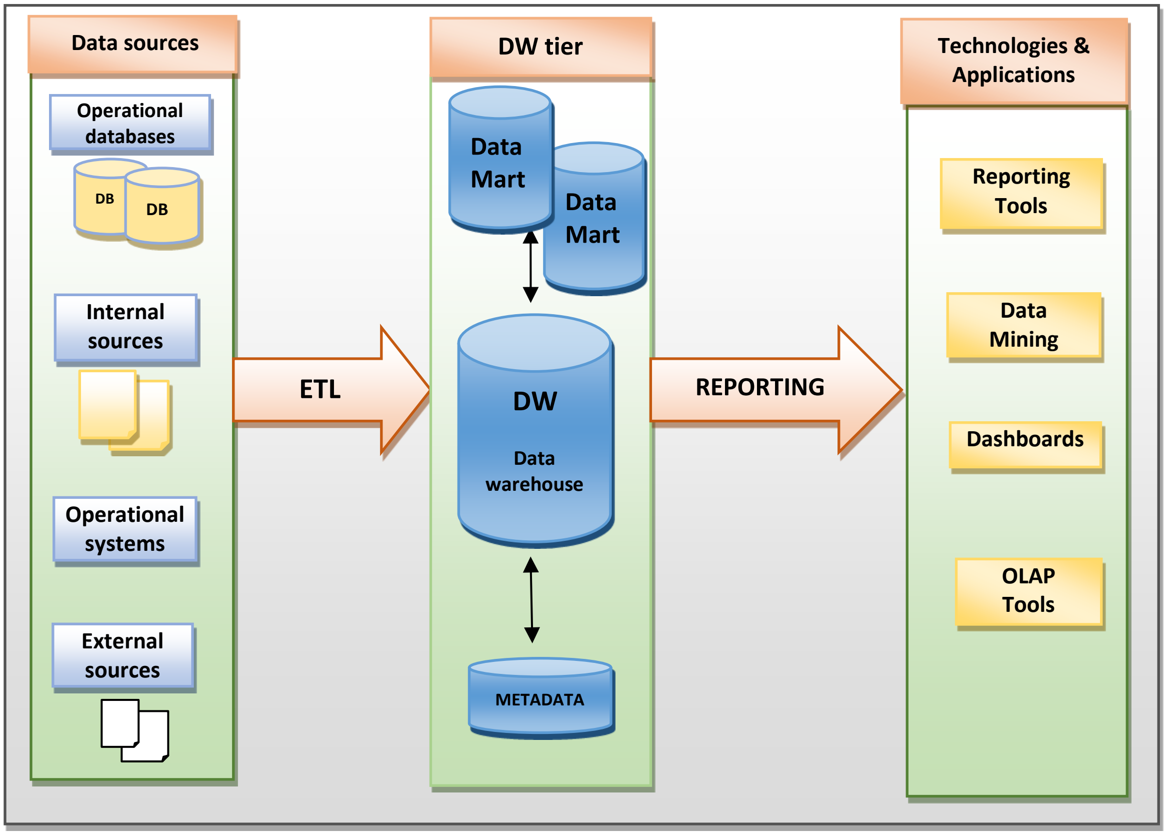


Figure 1-1 Data warehouse (DW) architecture

1.3.1 Data sources

As shown in figure 1.1, a DW collects data from multiple heterogeneous sources. These data sources can be internal or external sources, operational databases, and operational systems such as ERP, SCM or CRM (Malinowski & Zimányi, 2009)**.**

- ERP stands for enterprise resource planning and it "*is business process management software that allows an organization to use a system of integrated applications to manage the business and automate many back office functions related to technology, services and human resources"* (Beal).
- According to Martin Christopher, supply chain management (SCM) "*is the management of upstream and downstream relationships with suppliers and customers in order to deliver superior customer value at less cost to the supply chain as a whole*" (Christopher, 2016).
- CRM stands for customer relationship management. A CRM system manages business relationships and data related to them. It provides easy access to customer data within a company, such as service issues, contact information, sales opportunities and marketing campaigns (Salesforce, 2017).

1.3.2 ETL

ETL is referred as extraction-transformation-loading and it is a three-step process (Vaisman & Zimányi, 2014; Kimball & Ross, 2013; Rahm & Do, 2000; Kimball & Ross, 2002).

1) Extraction is the first step in the ETL process. Data is gathered from multiple heterogeneous data sources for further manipulation.

2) During the transformation phase, the format of the extracted source data is modified to the format of the DW. Transformation process includes data cleaning, integration and aggregation.

- Data cleaning detects and removes inconsistencies and errors from data (e.g. misspellings of data etc.). As a result, the quality of data is improved. Since a DW supports decision making, the correctness of the source data is essential to avoid a false conclusion (Rahm & Do, Data Cleaning: Problems and Current Approaches, 2000).
- Data integration refers to the integration of data from heterogeneous sources into a common DW target schema.
- In data aggregation process, the extracted source data is gathered and summarized depending on the granularity of the DW (Vaisman & Zimányi, 2014).

3) In the last step of the ETL process, transformed data is loaded into the DW.

Both extraction and loading are necessary, but they just gather and load data. On the other hand, the cleaning and the transformation of data play a critical role in the ETL process, since they modify data which are valuable and essential to the organization.

1.3.3 Data mart

As shown in Figure 1.1, the DW tier consists of a centralized DW, a set of data marts and the metadata. A data mart is a small local DW and it may originate from a DW or directly from the data sources. Data marts focus on specific departmental or functional areas within an enterprise, such as sales, marketing, financing etc. (Vaisman & Zimányi, 2014; Kimball & Ross, The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 2013; Kimball & Ross, The Data Warehouse Toolkit, 2002).

1.3.4 Metadata

The term metadata is defined as a set of data that describes and provides information about other data. The DW's metadata is created during the ETL process and it's stored in metadata repositories. The metadata has various forms depending on the needs of the DW's user groups, such as technical, business, administrative. The DW's metadata is divided into three categories: technical, business and process execution metadata. A brief description of the three categories is shown below:

- Technical metadata refers to the technical characteristics of data such as data types, length and lineage of data. Also, they provide information about the schema of the DW, the data source schemas and the ETL tasks.
- Business metadata describes the meaning of the data in business terms and stores business definitions of the data.
- Process execution metadata is created by the ETL procedure and contains information and statistics on the results of the ETL processes (Vaisman & Zimányi, 2014; Kimball & Ross, 2013; Malinowski & Zimányi, 2009; Kimball & Caserta, 2004;Kimball & Ross, 2002).

1.3.5 OLAP

OLAP stands for Online Analytical Processing. With OLAP, managers can perform complex analysis on the data stored in the DW. As already stated, a DW contains large amount of data from different heterogeneous sources that need to be analyzed. OLAP applications provide data analysis and support decision making queries. OLAP data is pre-aggregated and stored into cubes, instead of tables. "An OLAP cube is a snapshot of data at a specific point of time, for example at a specific day, week, month, or year" (Kravchenko, 2018). The multidimensional model enables the view of data from multiple perspectives and levels of detail. With OLAP operations, such as roll-up, drill-down, slice, dice, pivot etc., an end user can perform analysis over the aggregated data of a cube in different ways (Vaisman & Zimányi, 2014).

1.3.6 Data mining

Data mining is referred as "Knowledge Discovery in Databases (KDD)" and it is considered as the "evolution of Information Technology". Data mining extracts information and discovers patterns in large data sets for better decision making (Han, Kamber, & Pei, 2011; Fayyad, Piatetsky-shapiro, & Smyth, 1996).

1.3.7 Reporting tools

Reporting tools play an essential role in evaluating business strategies and improving decision-making. These tools analyze the extracted data and produce analytical reports and dashboards. These reports present information to managers and serve specific business needs.

1.4 Relational databases & Data warehouses

In this section, a set of schemas maintained by DWs, is presented: the star schema, the snowflake schema and the constellation schema. Before that, a comprehensive introduction to relational databases is provided for the reader to understand more easily the principles of designing a DW.

1.4.1 Relational model

A relational database (RDB) is based on the relational model. The relational model was introduced by E.F. Codd in 1970 (Wikipedia, 2019). It stores the data in a collection of tables, so as data is presented in a tabular form. The database tables are related to one another and form a collection of relations. Every table consists of rows and columns. Each row is called tuple and each column is called attribute.

- A primary key is a minimal set of attributes that uniquely identify a tuple in a relational table, and it shall not be null. For example, an 'employee' table has the following attributes: 'employee id, employee name, employee age, employee salary' etc. The primary key is the 'employee_id' which means that every employee has a unique ID (identifier).
- A foreign key is a set of attributes or fields in one table that is used to "refer" to a tuple in another table. Relationships between database tables can be established using foreign keys.
  For example, a table named 'employee_details' consists of the foreign key 'employee_id'. The foreign key 'employee_id' in table 'employee_details' refers to the primary key of 'employee' table (Wikipedia, 2019; Sumathi & Esakkirajan, 2007).

1.4.2 SQL

Structured Query Language (SQL) is used for managing and retrieving data from a relational database management system (RDBMS). SQL contains many types of statements which operate on relational tables. Its main purpose is to query, manipulate (insert/update/delete) and define data in a relational DB (Wikipedia, 2019; Vaisman & Zimányi, 2014).

1.4.3 Normalization & Denormalization

In a relational DB, data redundancy shall be eliminated. Data redundancy means that same data is stored more than once in the DB. It results in wastage of storage space and loss of data integrity (Sumathi & Esakkirajan, 2007). Normalization is the process of organizing the data in a relational DB and it removes redundancy. The normal forms that have been defined are: 1NF, 2NF, 3NF, Boyce-Codd Normal form (BCNF), 4NF, 5NF. Generally, "3NF is considered good enough" (Sumathi & Esakkirajan, 2007, p. 297). A disadvantage of normalization is the production of many normalized tables. As a result, the merging of data (table joins) requires more time and the queries become more complicated. Denormalization is the opposite of normalization and improves queries' performance.

Relational databases are used by organizations for their day-to-day operations. They contain data which comes from the daily transactions of an enterprise. Although, relational databases were widely used from late 1970s, they do not support decision making and analysis of the data. Nowadays, enterprises have different needs due to competitiveness and increasing market dynamics. By late 1990s, data warehousing was introduced. A DW contains ETL processing and OLAP tools for reports and data analysis. Managers can analyze data from different heterogeneous systems and take strategic decisions (Vaisman & Zimányi, 2014; Sumathi & Esakkirajan, 2007).

1.4.4 Data warehouse design

A Data warehouse design is based, at the logical level, on a multidimensional model which usually consists of relational tables that form specific structures named star schema, snowflake schema and constellation schema. Based on this relational representation, an OLAP server can build a data cube, that enables the view of data from multiple perspectives and levels of detail. This multidimensional model includes facts, measures, dimensions and hierarchies. In the next section, a set of schemas maintained by DWs, is presented (Vaisman & Zimányi, 2014; Malinowski & Zimányi, 2009).

1.4.4.1 Star schema

The star schema is considered one of the most popular schemas for the design and implementation of a DW (Huynh & Schiefer, 2001). It consists of a large, single, central table (called fact table) and multiple dimension tables. The data is organized into facts which are linked to dimensions. The schema is like a star, as shown in Figure 1.2.

The example in Figure 1.2 is adapted from book ‘Data Warehouse Systems-Design and Implementation’ by authors Vaisman A. & Zimányi E. , 2014, Berlin: Springer, p. 123.

- The **fact table** contains the bulk of data and it is located at the center of the star. A fact focuses on specific subjects of analysis such as sales, orders etc. and contains attributes, named measures (Singhal, 2007; Huynh & Schiefer, 2001). A measure is a numeric value.
    - In Figure 1.2, the fact table “Sales” is related to multiple dimension tables “Product”, Store”, “Promotion” and “Time”. For that reason, the key of the fact table “Sales” (key: ProductKey, StoreKey, PromotionKey, TimeKey) is a set of the foreign keys of the related dimensional tables. The fact table “Sales” contains the business measures: “Amount” and “Quantity” and it is also normalized (Vaisman & Zimányi, 2014).
- The **dimension tables** are organized around the fact table in a radial way. They contain several attributes which are used by OLAP queries. With dimensions someone can view the data from

multiple perspectives. For example, the time dimension is useful for the analysis of sales over different periods of time.

In the star schema, dimensions tables are denormalized in order to reduce the number of tables joins when queries are executed (Vaisman & Zimányi, 2014; Boukhalfa, et al., 2009; Singhal, 2007; Savonnet & Terrasse, 2001).

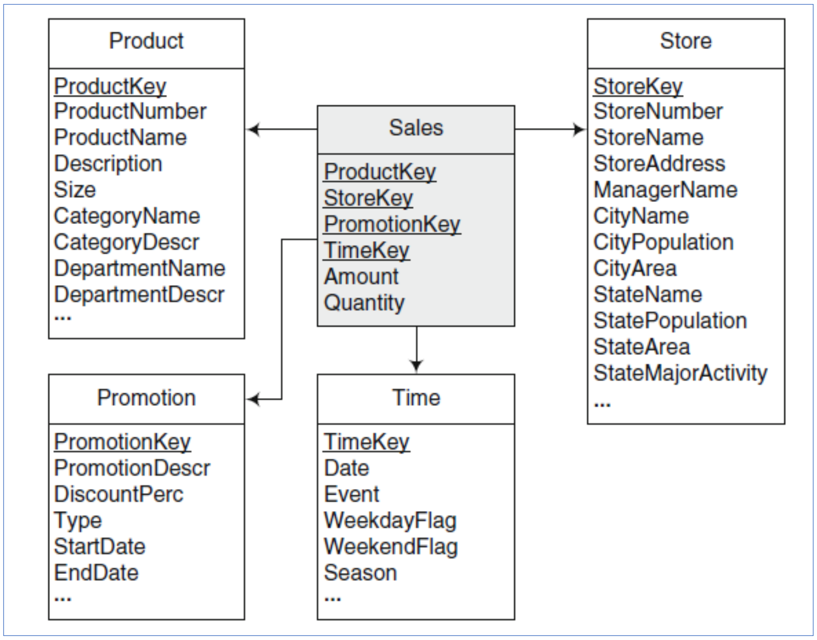


Figure 1-2 Example of a star schema
(Adapted from *'Data Warehouse Systems: Design and Implementation' by Vaisman A. & Zimányi E., 2014, Berlin:Springer, p. 123.)*

1.4.4.2 Snowflake schema

In the snowflake schema, the dimensions that surround the fact table are normalized and redundancy is reduced. In the example below, the fact table "Sales" remains as it is, but the dimension tables "Product" and "Store" are normalized.

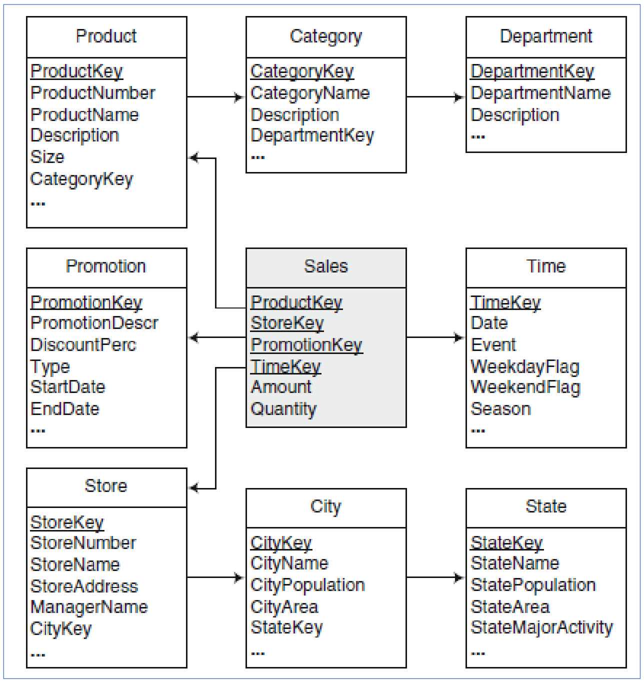


Figure 1-3 Example of a snowflake schema.
(Adapted from *'Data Warehouse Systems: Design and Implementation' by Vaisman A. & Zimányi E., 2014, Berlin:Springer, p. 124.)*

1.4.4.3 Constellation schema

A constellation schema consists of multiple fact tables which share dimension tables. As shown in Figure 1.4 the constellation schema contains two fact tables which are 'Sales' and 'Purchases'.

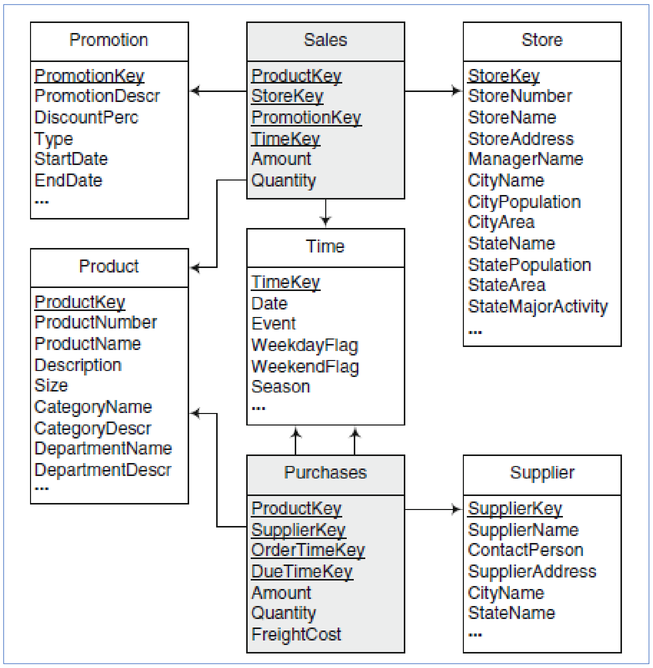


Figure 1-4 Example of constellation schema
(Adapted from *'Data Warehouse Systems: Design and Implementation' by Vaisman A. & Zimányi E., 2014, Berlin:Springer, p. 125.)*

2 Related Work

Business Intelligence (BI) technologies are involved in the process of decision-making in different areas due to the fact that they extract knowledge from stored data (Brandão, et al., A Benchmarking Analysis of Open-Source Business Intelligence Tools in Healthcare Environments, 2016). BI systems used in universities support the transfer of knowledge in the society which is the principal aim of the universities, unlike companies whose main purpose is making profit (Guitart & Conesa, 2015; Guitart & Conesa, 2016). In this section, a literature review is presented regarding the application of BI technologies in education.

Maia et. al (2018) proposed and designed a Web Intelligent System (WIS) in Higher Education (HE) for understanding the factors of student retention and success. The WIS gathers and analyses data from the "IoEduc" application. The "IoEduc" is an application used by both students and teachers for learning and teaching purposes. The proposed Web Intelligent System (WIS) consists of three stages: "Data source, Data preparation and data visualization" (Maia, Portela, & Santos, 2018, p. 178). At stage one, generated data from "IoEduc" application was used as input to the WIS. This input data was collected during one semester (2017/2018) of an academic course called "Web programming" at the University of Minho. At stage two, the input data was prepared for further analysis. The input data was "extracted, cleaned and transformed" (Maia, Portela, & Santos, 2018, p. 179). At stage three, reports and dashboards were created for the representation of the results.

Aquila, et al. (2008) designed a data warehouse for their university in order to provide information to different academic units and external agencies. The data of the designed Business Intelligence System comes from six different source databases. After the ETL processes, the data is cleaned, integrated and loaded into the data warehouse. The data warehouse consists of four data marts (didactics, finance, research and human resource). The OLAP layer of the BI system provides reports to the decision makers that come from national or internal agencies.

Guitart & Conesa (2016) proposed a Universal Analytical Information System (UAIS) that should be used globally by all the university staff and provide new functionalities not available in similar analytical systems. As stated by the authors, "*An UAIS is a set of processes and tools that collect and analyze the internal and external data of the university, converting them into meaningful and useful information that can be used to enable more effective and timely strategic decision-making about the activites of the* *university* (Guitart & Conesa, 2016, p. 176). The creation of the UAIS is a research goal and it is based on the structure of a BI system. The data imported in the UAIS shall come from a different set of data sources such as management data, educational data, performance data, navigational data, research data and external data. Then a set of ETL processes loads,extracts and transforms the data from the different data sources to the formulated data warehouse. Furthermore, information analysis can be performed by OLAP or data mining tecnhiques and a set of visualization tools is available for decision making. The UAIS system can be considered as a conceptual framework that contains subsets of other analytical systems.

Indrajani, et al. (2018) have built a data warehouse for analysing active student data. This data warehouse supports management decision making for the active students of XYZ university. Ralph Kimball's mehodology was used for the designation of the data warehouse and Pentaho Data Integration (PDI), an open source tool, was used for the ETL processing of the data from the different operational databases. Dashboards were designed for the visualisation of the data imported in the data warehouse.

Santoso & Yulia (2017) present the design and the implementation of a modern data warehouse for academic data with the use of Big Data technology. The significant growth of academic data in universities can be handled by combining Big data technology and data warehouse. Hadoop, an open source analytic tool for big data, was used for processing distributed and parallel academic big data and analyze large volumes of structured and unstructured data. Dashboards produced by the proposed system, are monitored by the academic staff consisting of top, middle and bottom management for supporting decision making. Further discussion is made on the characteristics of traditional and modern data warehouses.

Guitart & Conesa (2015) have proposed a Business Intelligence (BI) System that analyses data from Virtual Learning Environments (VLE). VLE gather data of all the agents that use the system, consisting of students and teachers, and improve learning and teaching within universities. The proposed BI system consists of ETL processes , a data warehouse and a set of dashboards. It supports decision making and provides dashboards for both teachers and academic managers of their university, as well. The dashboards display a group of academic indicators that are different between teachers and managers.

Duan, et al. (2013) presented an action research which was in progress back then. This research aimed to study the appliance of Business Intelligence (BI) on a student engagement tracking system (SES) in a UK higher education institution (HEI). The SES gathered data from different sources (e.g. library, seminar rooms, etc.) across various university's locations with the use of Radio Frequency Identification Devices (RFID). In this way, SES users could better re-engage students and prevent students' disengagement. Nevertheless, the SES system "served as merely an information source, rather than a decision support environment" (Duan, Cao, Ong, & Woolley, 2013). In this project, an action research was conducted on how BI solutions can be utilized with Big data for improving the current SES functionalities and supporting decision-making, as well.

Song, et al. (2016) have designed a Peer-Review Markup Language (PRML) in order to model data generated by different online peer assessment systems. Students are using these peer assessment systems. With PRML a data warehouse was built (based on the star schema) for storing the data from the different online peer assessment systems. ETL processes were involved for the loading of the data to the data warehouse with the use of Pentaho.

Bakar, et al. (2018) have integrated different data sources from UUM (Universiti Utara Malaysia) online learning systems with the usage of a data warehouse and business intelligence techniques. In this way, lecturers and UUM management can monitor and evaluate the usage of blended learning. By blended learning, they mean the combination of traditional learning methods with the use of new technology in Institutions of Higher Education (IHE).

In STMIK Pelita Nusantara university, a data warehouse was implemented based on the Kimball's nine step methodology for the accreditation of the university. BAN-PT is a national accreditation council of higher education in Indonesia. Accreditation assures the quality of an academic institution. For that reason, the university shall prepare and store documents for the accreditation process. The data warehouse analyses various data sources and exports charts and reports (Sinaga & Girsang, 2017).

Same in STMIK STIKOM Bali (an educational organization) which pertained to the designing of a data warehouse for the accreditation of the institution by BAN-PT. For that reason, a centralized data warehouse with integrated data from various data sources was needed for the executives to gain the information quickly and easier (Budiarta, Wijaya, & Partha, 2017).

Business Intelligence system "ICE/eduSTORE" is an open source web-based information system that supports higher education decision-making. It is produced by the German non-profit organization HIS Higher Education Information System Agency. Based on a data warehouse, "ICE/eduSTORE" imports data from different sources, such as administrative, statistical, college data etc. System's data is harmonized with an integrated key-system. As mentioned by the authors, the system "never forgets about previously imported data" (Muessig-Trapp & Skladovs, The Data Portal of the German Federal Ministry of Education and Research (BMBF) as part of the German Open Government Approach, 2013, p. 2). In this way, the system enables the generation of time series. Also, analytical tools and report generators are available by the system (Muessig-Trapp & Skladovs, The Data Portal of the German Federal Ministry of Education and Research (BMBF) as part of the German Open Government Approach, 2013; Muessig-Trapp & Quathamer, Business Intelligence in HISinOne [Conference Presentation], 2011).

A framework for the development of a Business Intelligence (BI) solution for universities is presented by Muntean et al. (2011). Since BI is important for the strategic planning of universities, a dimensional data model was proposed for the assessment of the distance learning in their university. Specifically, the dimensional data model was implemented, based on the constellation schema, for the assessment of Moodle (an open source e-learning platform and course management system). The imported research data was extracted from three data sources: Moodle database, Moodle logs and another database called SIMUR. For the e-learning assessment, dashboards were provided to different groups such as top management, faculty, teachers and distance (ID) learning department.

Di Tria, et al. (2015) present the architecture of an Academic Business Intelligence system. The vital part of the BI system is an academic data warehouse which consists of a set of data marts. The development of the academic data warehouse aimed at performing analysis on the aspects influencing the quality of a university which are the evaluation of Didactics and Research. Based on an ontological approach, a hybrid methodology was used, mostly automatic, for the integration of the different data sources.

3 ETL Tools

3.1 Introduction

An ETL tool is used for extracting data from various sources, transforming it to a specified format, and loading it into a target Data Warehouse or a repository. Any organization can use this type of software for handling Big Data.

Integration of advanced IT technology on every day activities has contributed to the production of vast amounts of data which can’t be managed efficiently. Big data come from different application sources and are categorized in types: IoT (Internet of Things), social media, multimedia and self-quantified (Yaqoob, et al., 2016; cloudmoyo).

- IoT data is defined as data, which is generated by equipment, mobile computing phones, GPS devices, sensors etc.
- Social data is generated by Facebook, Twitter, LinkedIn, Instagram etc.
- Multimedia data comes from images, texts, audios and videos. Even when the users navigate online, they generate this type of data.
- Wearable devices collect many types of data which relate to everyday activities and vital signs of humans. This kind of data constitute self-quantification data.

The amount of data collected is beyond human perception. Therefore, there is high demand for IT solutions which manage the large amount of data in an effective way such as the ETL tools.

The most used open source BI platforms are QlikView, JasperSoft, Palo, Pentaho, Tableau, SpagoBI, Vanilla, Actuate, and OpenI (Brandão, et al., A Benchmarking Analysis of Open-Source Business Intelligence Tools in Healthcare Environments, 2016; Lapa, Bernardino, & Figueiredo, 2014).

3.2 Benefits of ETL Tools

ETL tools consist of graphical interfaces (GUI) providing a visual flow of the steps involved in the ETL processes. The GUI presents the ETL components and the flow of data among them. The extraction, transformation and loading of data becomes an easy process. The ETL tools are user-friendly and easy to comprehend. The users can define the mappings between the source schemas (data sources) and the target schemas without writing complex code or queries. It is a low-code approach allowing users to use a GUI instead of writing traditional source code. As a result, the data integration is faster, the ETL processes speed up and the cost is reduced (SpringPeople, 2018).

An important feature of ETL tools is data cleansing which is considered one of the biggest issues in DWs. Data cleaning improves data quality by detecting and removing inconsistencies and errors. (Rahm & Do, Data Cleaning: Problems and Current Approaches, 2000). The set of cleaning data functions in ETL is richer in comparison to the functions available in SQL (SpringPeople, 2018).

Over time, the return-on-investment (ROI) improves. The time and cost are reduced, the querying is faster with less mistakes and the results generated are better (adverity).

A Business Intelligence solution saves money and resources. The IT personnel does not waste time preparing reports and the integration of heterogeneous data from different sources can be automated. Consequently, data can be easily accessed and analyzed, enabling effective strategic decision-making (Lebied, 2017).

The ETL tool used in the thesis is the “JasperSoft ETL, The Open Source Data Integration Platform” (https://community.jaspersoft.com/project/jaspersoft-etl). Specifically, Talend ETL is the built-in data integration tool used by JasperSoft. JasperSoft shares a partnership with Talend (Wise, 2012).

3.3 Introduction of TIBCO JasperSoft ETL Community

In this section a brief description of the tool and its components is presented. Additionally, an example of creating a simple ETL job is shown. TIBCO JasperSoft ETL Community (version 6.0.1) is a powerful and flexible tool which helps users to manipulate and control their data. A presentation of the open source tool is given below.

3.3.1 Features

JasperSoft ETL has a variety of features that enhance data integration between heterogeneous data sources. It consists of a drag-and-drop ETL Job designer (GUI) that enables the editing of ETL processes. With the use of specific ETL transformation components, JasperSoft ETL transforms data from a variety of sources (CSV, XML files, web services, databases etc.) and loads them into data marts and warehouses for further analysis and reporting. The tool generates Java or Perl code embedded in any system. As a result, an ETL process can be imported in an external Java application and can run independently (JasperSoft, Getting Started with Jaspersoft ETL. What is Jaspersoft ETL?).

JasperSoft ETL is part of the JasperSoft BI platform which provides dashboards, reports, data integration services, OLAP data analysis and other essential BI capabilities. The end user can manipulate data or create reports without the need of IT personnel. The BI platform has two license types: Community (free) and Commercial (additional costs). Except for JasperSoft ETL, JasperSoft BI platform consists of other components, as well (Vargas, Syed, Mohammad, & Halgamuge, 2016; Lapa, Bernardino, & Figueiredo, 2014).

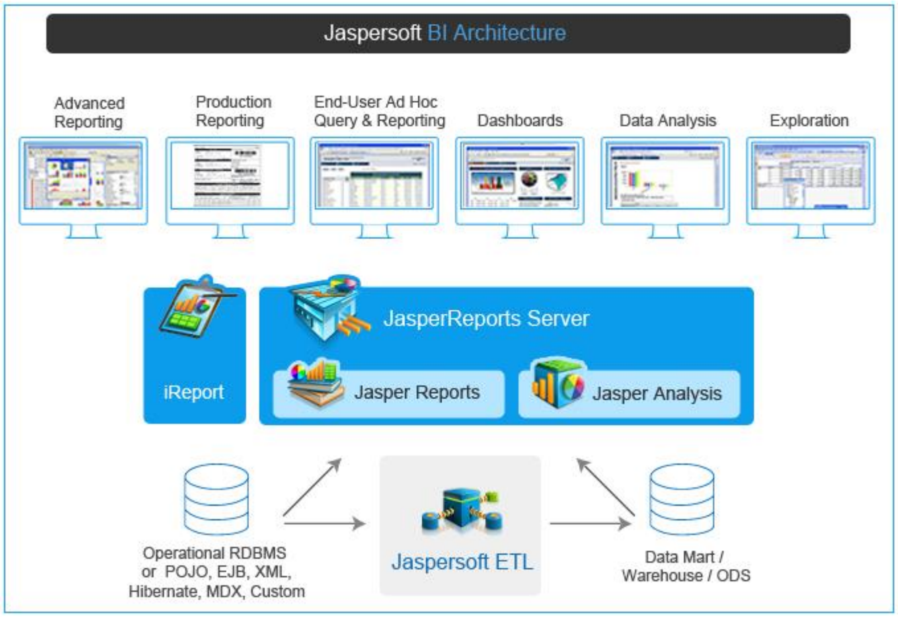

Jaspersoft BI Architecture
Advanced Reporting
Production Reporting
End-User Ad Hoc Query & Reporting
Dashboards
Data Analysis
Exploration
iReport
JasperReports Server
Jasper Reports
Jasper Analysis
Operational RDBMS or POJO, EJB, XML, Hibernate, MDX, Custom
Jaspersoft ETL
Data Mart / Warehouse / ODS

Figure 3-1 JasperSoft BI Architecture. Source: https://www.columnit.com/jaspersoft.html

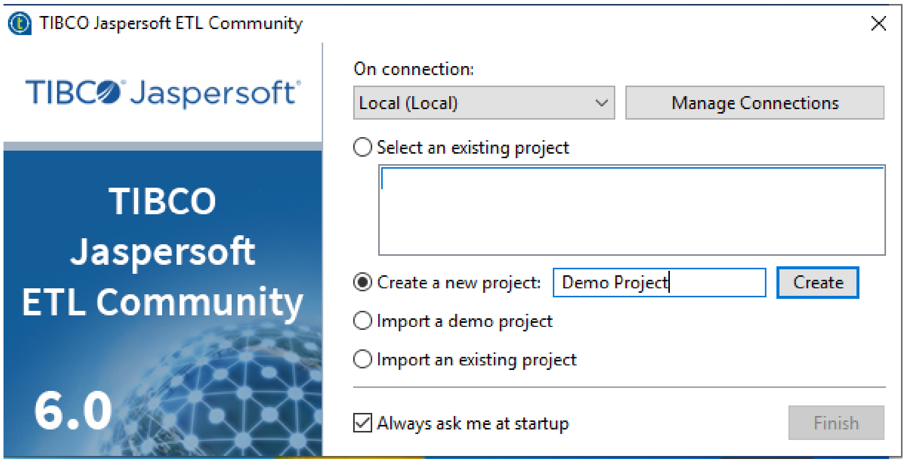
`JasperReports Server` is a reporting server which can be embedded into web or mobile apps. Furthermore, enterprises can use it as a central information hub for sharing, securing and transferring information to browsers, other devices or emails on a scheduled basis or on real-time (JasperSoft, JasperReports Server:Self-service Reporting and Analysis Server).

`JasperReports Library`, an open source reporting engine, manipulates different types of source data and creates documents which can be exported in several file formats such as PDF, Excel, HTML, OpenOffice and Word. This software is written in Java. It is available as standalone software and integrated in JasperReports Server software and in JasperSoft Studio (JasperSoft, JasperReports Library:Open Source Java Reporting Library; JasperSoft, The World's Most Popular Reporting Engine:Generate any report or visualization with lightning fast performance).

`JasperSoft Studio` is a report designer available as a standalone software or as an Eclipse plugin. It can access different types of data sources such as CSV, JDBC, NoSQL, JSON etc. and generate report documents in different file formats e.g. OpenOffice, RTF, PDF, PowerPoint, CSV, JSON, XML etc. (JasperSoft, Jaspersoft Studio:The Eclipse-based report development tool for JasperReports and JasperReports Server).

3.3.2 Graphical User Interface

In JasperSoft ETL Community tool, all the work is organized into Projects. After launching the ETL tool, the user shall create a new project first (Figure 3-2).



TIBCO Jaspersoft ETL Community

On connection: Local (Local)

○ Select an existing project

● Create a new project: Demo Project

○ Import a demo project

○ Import an existing project

☑ Always ask me at startup

Figure 3-2 Creation of new project in TIBCO JasperSoft ETL Community tool.

The GUI of TIBCO JasperSoft ETL Community composes of several views: (Talend, Discovering Talend Studio)

a. **Project Repository**. This view composes of project items such as Jobs, Code, Services, Metadata and Documentation. (Figures 3-3, 3-4)

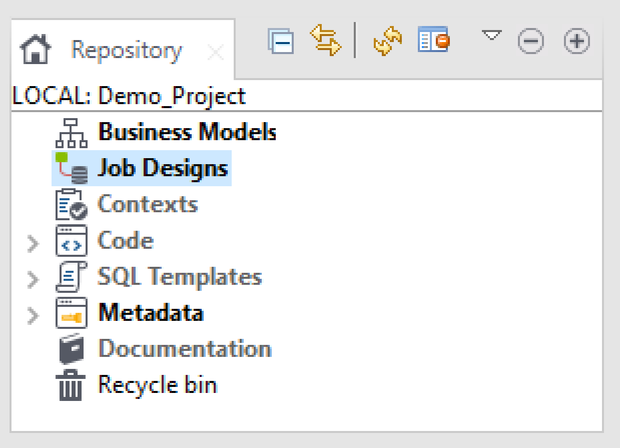


Figure 3-3 Project Repository View in JasperSoft ETL Community tool.

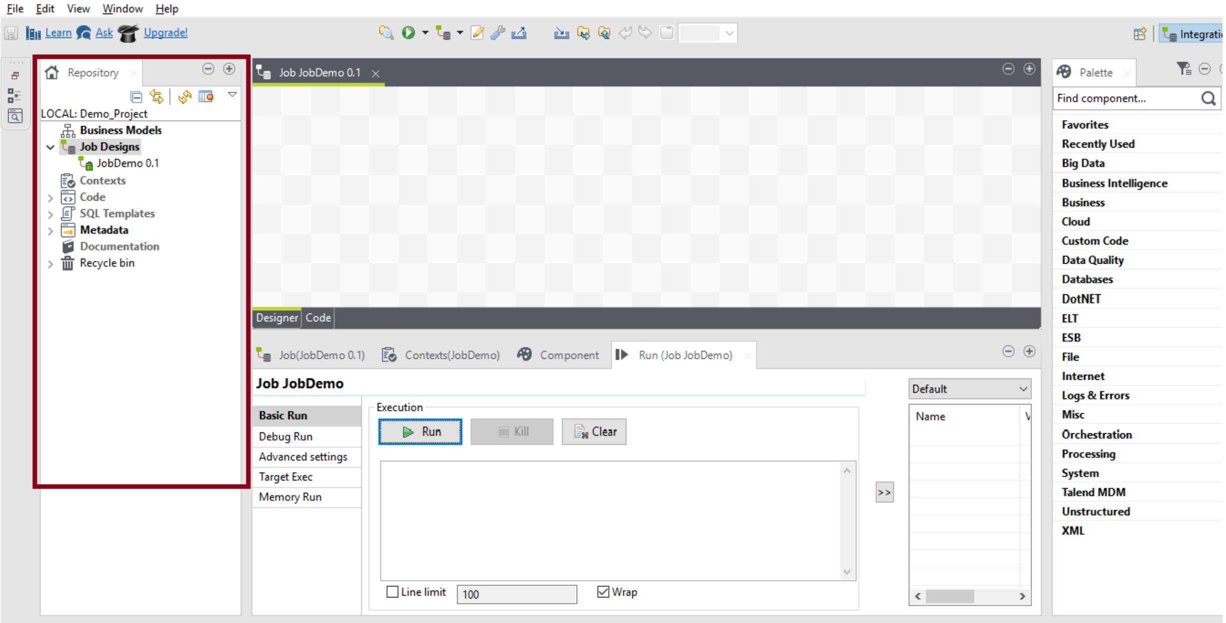


Figure 3-4 Project Repository View

b. The **Job Designer** is the main view of the TIBCO JasperSoft ETL Community and it's used for the creation of ETL jobs. (Figure 3-5) An ETL Job represents an ETL process which extracts, transforms and loads data from different heterogeneous sources. ETL Jobs will be presented in detail later in this thesis.

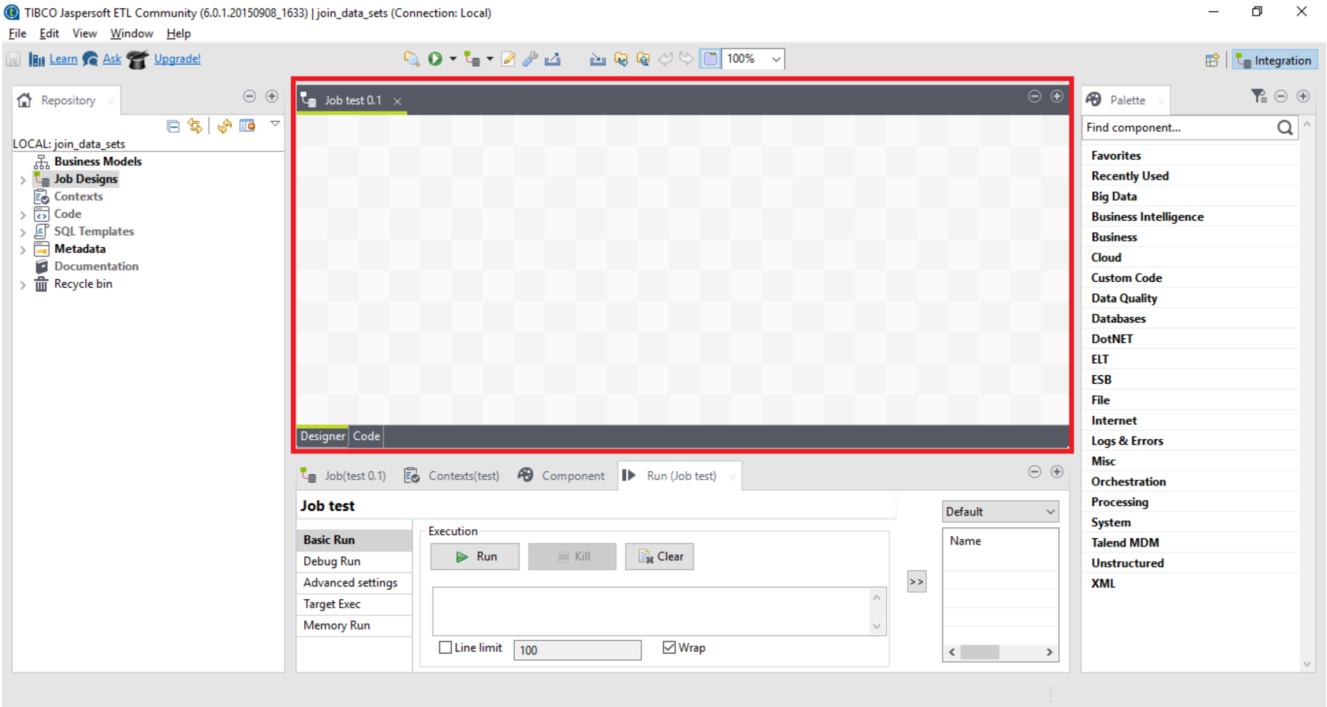


Figure 3-5 Job Designer View in JasperSoft ETL Community tool

c. The **Palette** contains all available components used for ETL jobs. (Figure 3-6)

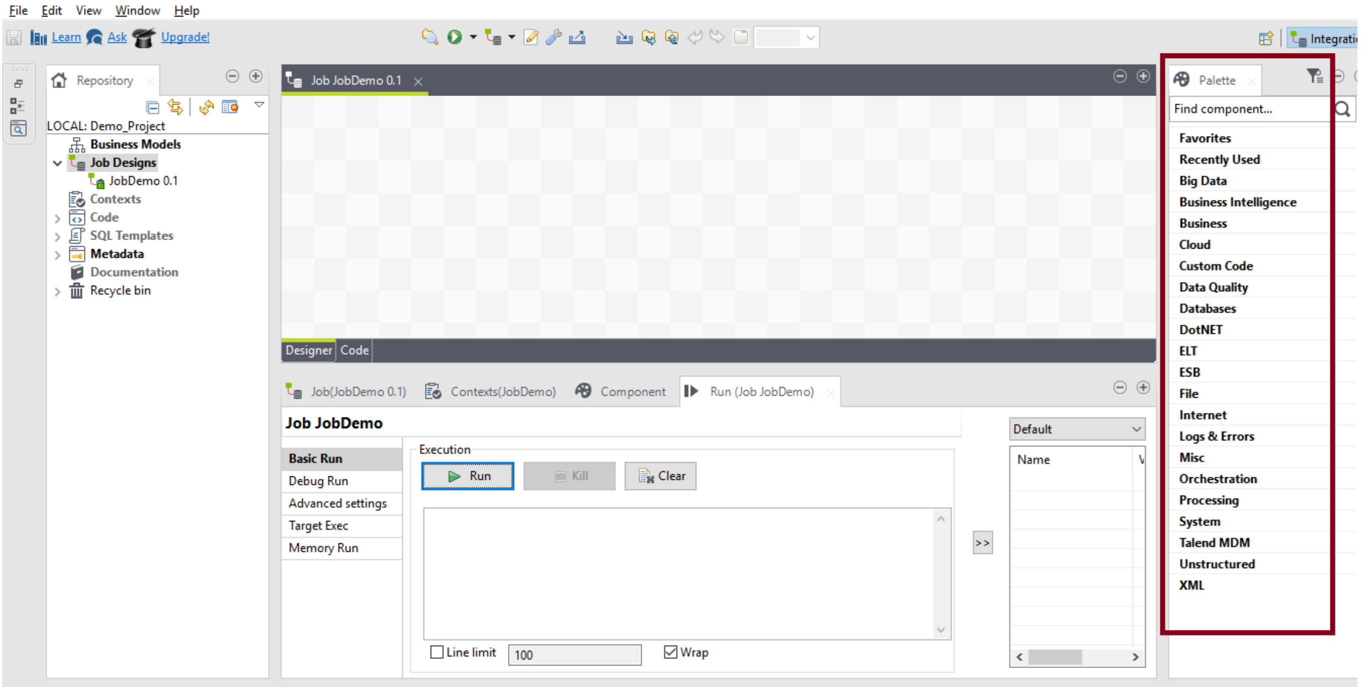
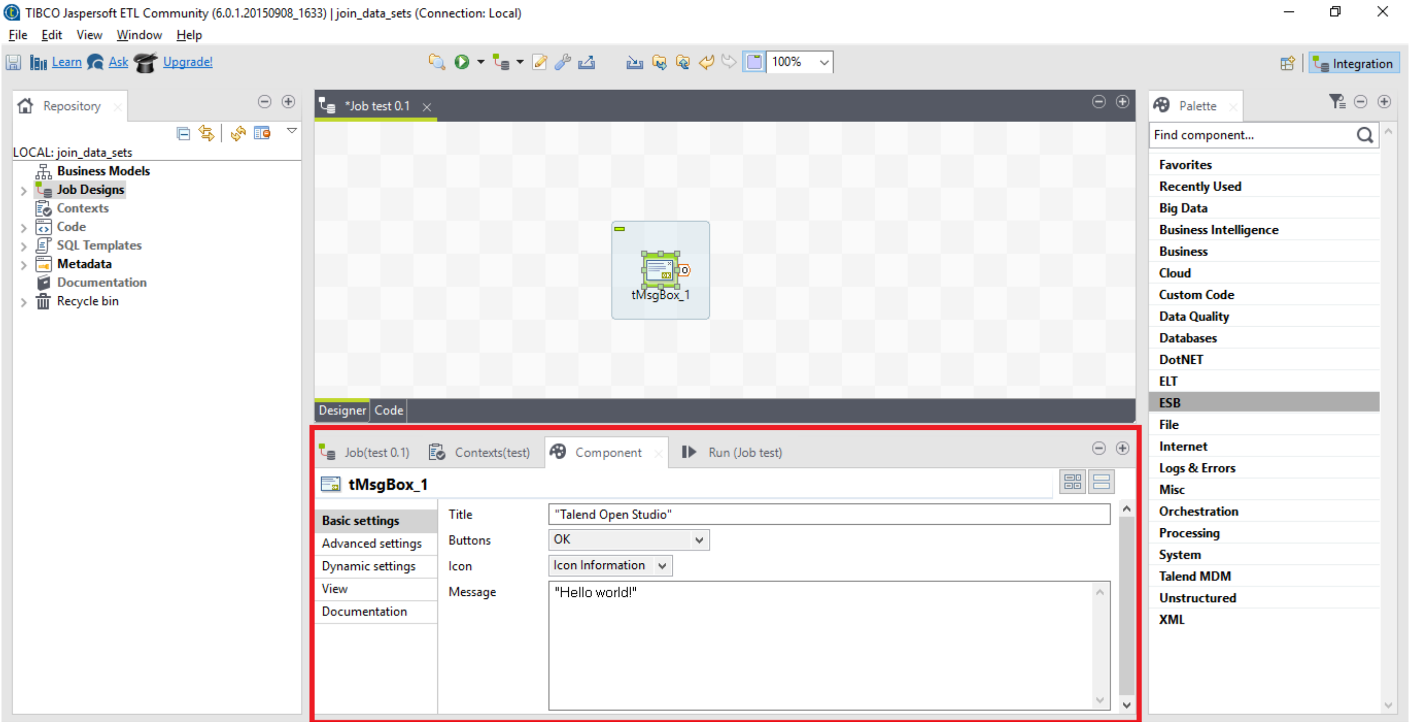


Figure 3-6 Palette View in JasperSoft ETL Community Tool

d. The **Component** view displays the components' configuration parameters. (Figure 3-7)



Title "Talend Open Studio"

Buttons OK

Icon Icon Information

Message "Hello world!"

Figure 3-7 Component's View in JasperSoft ETL Community tool

e. In the **Run View**, a job can be executed, and its results are displayed in the console window. (Figure 3-8)

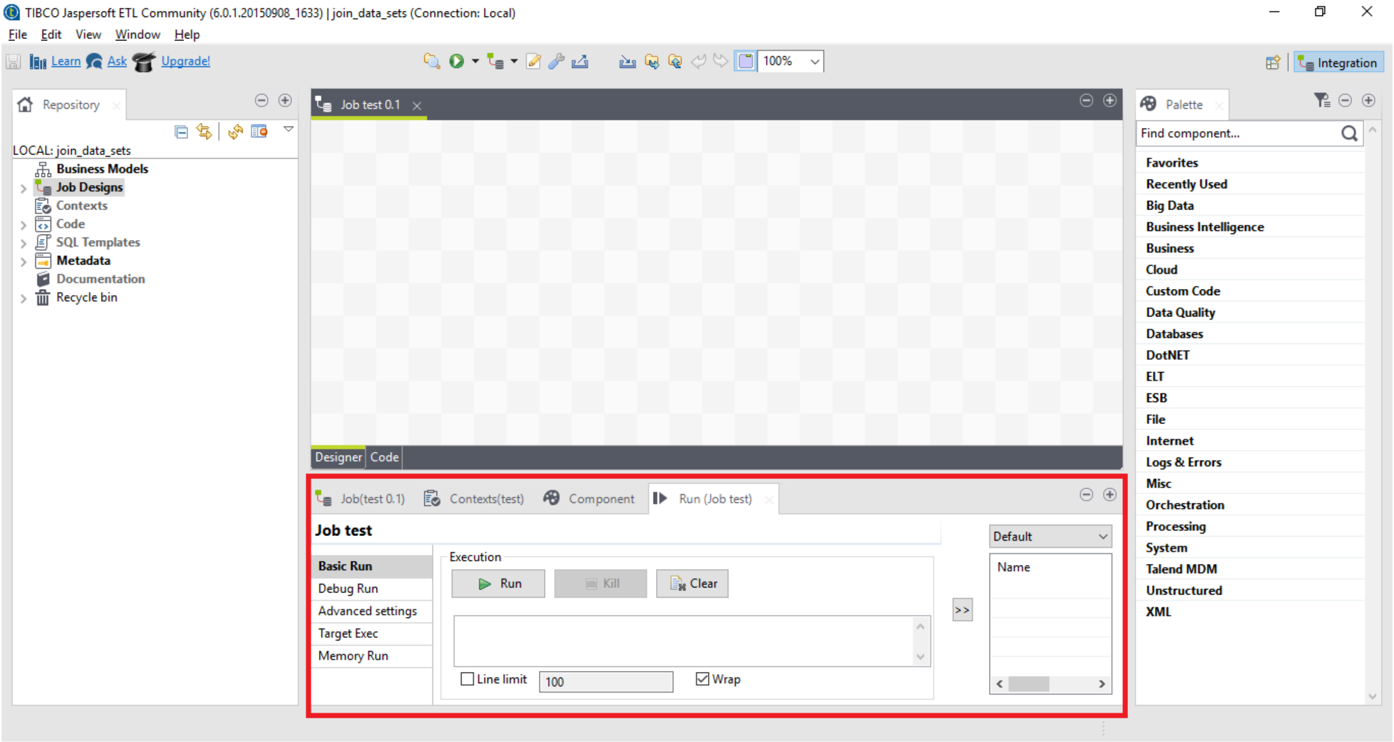


Figure 3-8 Run View

3.3.3 Design and implementation of an ETL Job in JasperSoft ETL Community

In the Project Repository, the user can right click on ‘Job Designs’ and choose ‘Create job’. (see Figure 3-9)

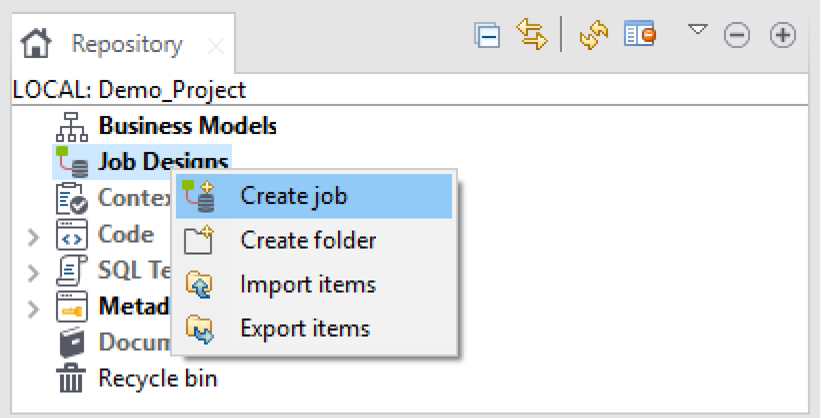
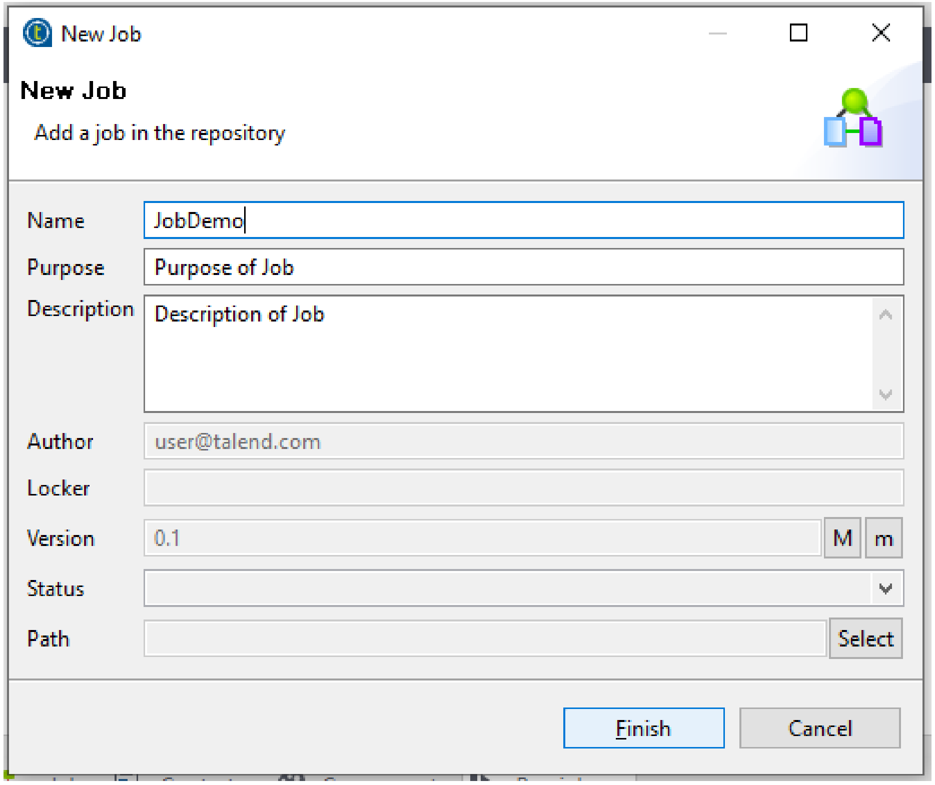


Figure 3-9 Creation of Job in JasperSoft ETL

A new panel opens. The user can type the ‘Name’, ‘Purpose’ and ‘Description’ of the ETL Job and clicks on ‘Finish’. (Figure 3-10)



New Job

New Job

Add a job in the repository

Name: JobDemo

Purpose: Purpose of Job

Description: Description of Job

Author: user@talend.com

Locker:

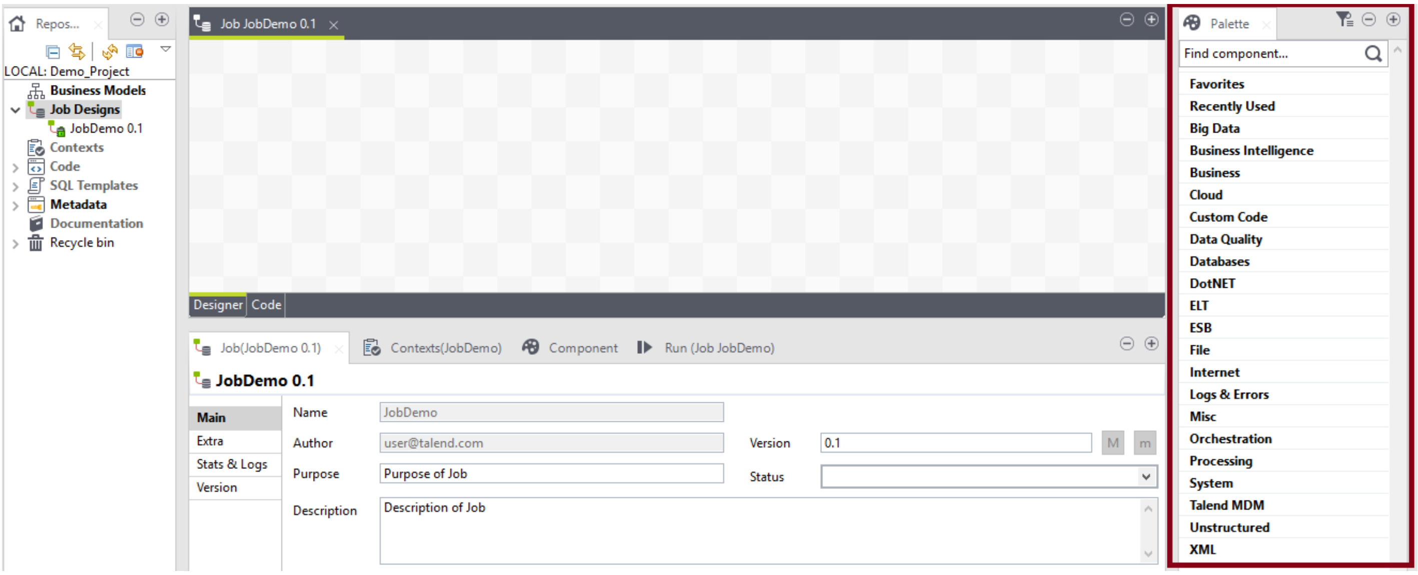
Version: 0.1

Status:

Path:

Figure 3-10 Defining the name, purpose and description of the new Job in JasperSoft ETL tool.

In the ‘Job Designer’ the new Job named ‘JobDemo’ is displayed which is empty. The user builds a Job with the use of components which are listed in the Palette View. Components can perform tasks such as aggregation, sorting, writing to different data sources and support other functionalities as well. (Figure 3-11)



JobDemo 0.1

Name: JobDemo

Author: user@talend.com

Purpose: Purpose of Job

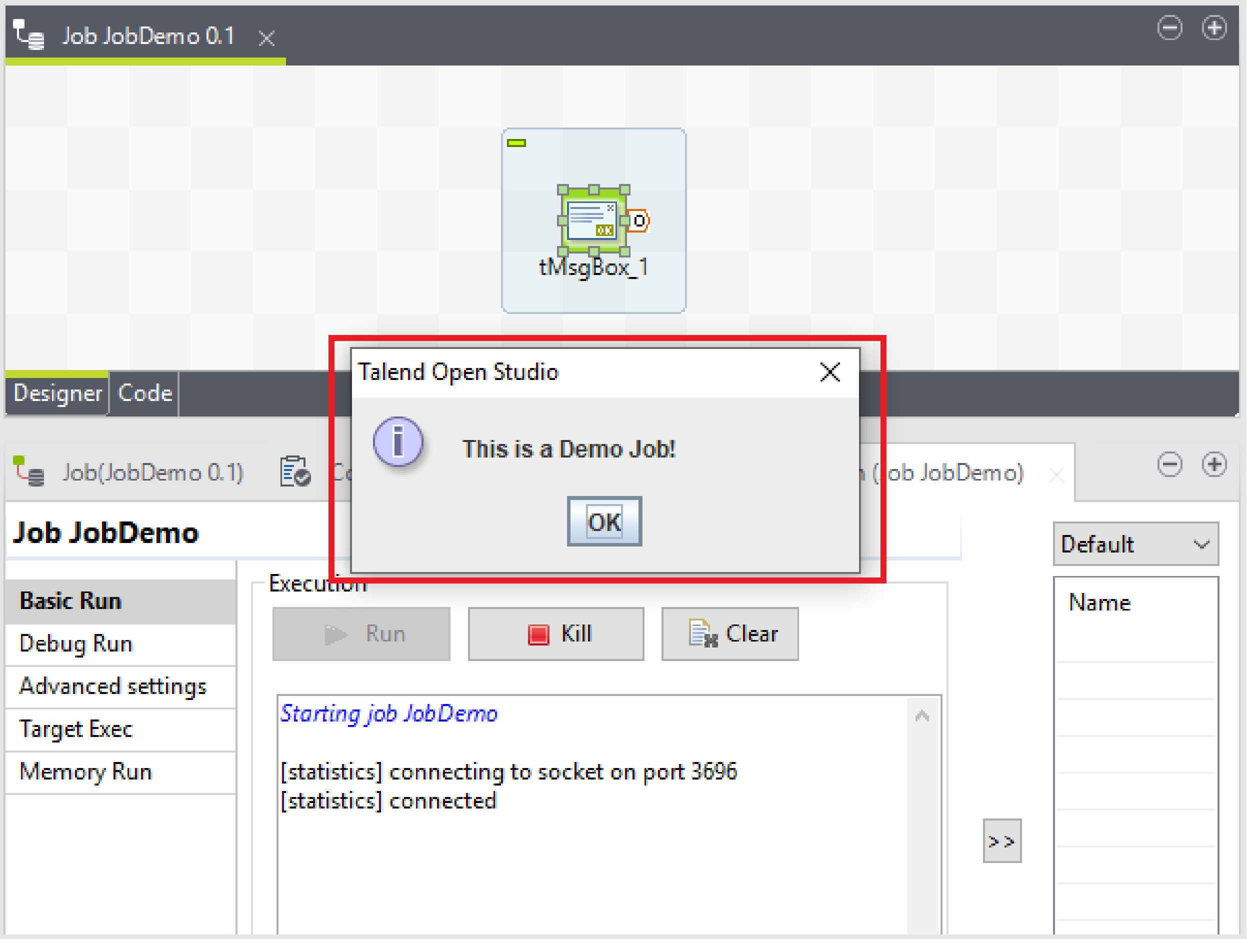
Description: Description of Job

Version: 0.1

Status:

Figure 3-11 Job Designer View

In the example shown below, a ‘tMsgBox’ component is chosen from the list of components. This component displays a message box. When the user triggers the execution of the Job, a message containing the text ‘This is a Demo Job is displayed’, is shown. (Figure 3-12)



Talend Open Studio

This is a Demo Job!

OK

```
Starting job JobDemo

[statistics] connecting to socket on port 3696
[statistics] connected
```

Figure 3-12 Running a Job that outputs a message.

4 Design and implementation

For the implementation and the execution of the ETL Jobs, data sources from the e-class platform of the Hellenic Mediterranean University (https://eclass.hmu.gr/) were used. The first data source is a csv file containing professors' data *(2960 records).* Similarly, the second data source is a csv file containing students' data *(165.428 records).*

4.1 Data Conversion

JasperSoft ETL tool enables the conversion of data to different formats, for example converting a CSV (UTF-8) file into an Excel file. For this Job, the following components have been used:

1. tFileInputDelimited
2. tFileOutputExcel

The 'tFileInputDelimited' component reads a file row by row and sends the output to the next Job component (Talend, tFileInputDelimited).
The 'tFileOutputExcel' component outputs data to a MS Excel file (Talend, tFileOutputExcel).

The steps for creating this job are shown below:

1. First, a Job named "Csv2Excel" shall be created.
2. In the Designer View, a 'tFileInputDelimited' shall be chosen from the Palette View and configured within the component View. The 'Component View' contains a list of configuration parameters: (See Figure 4-1)

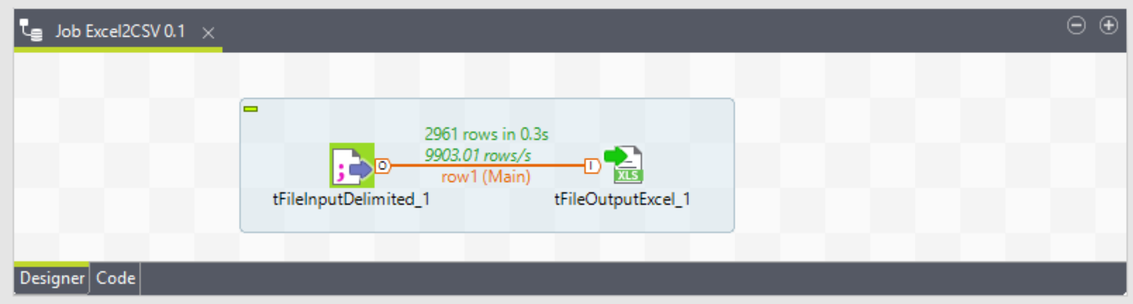
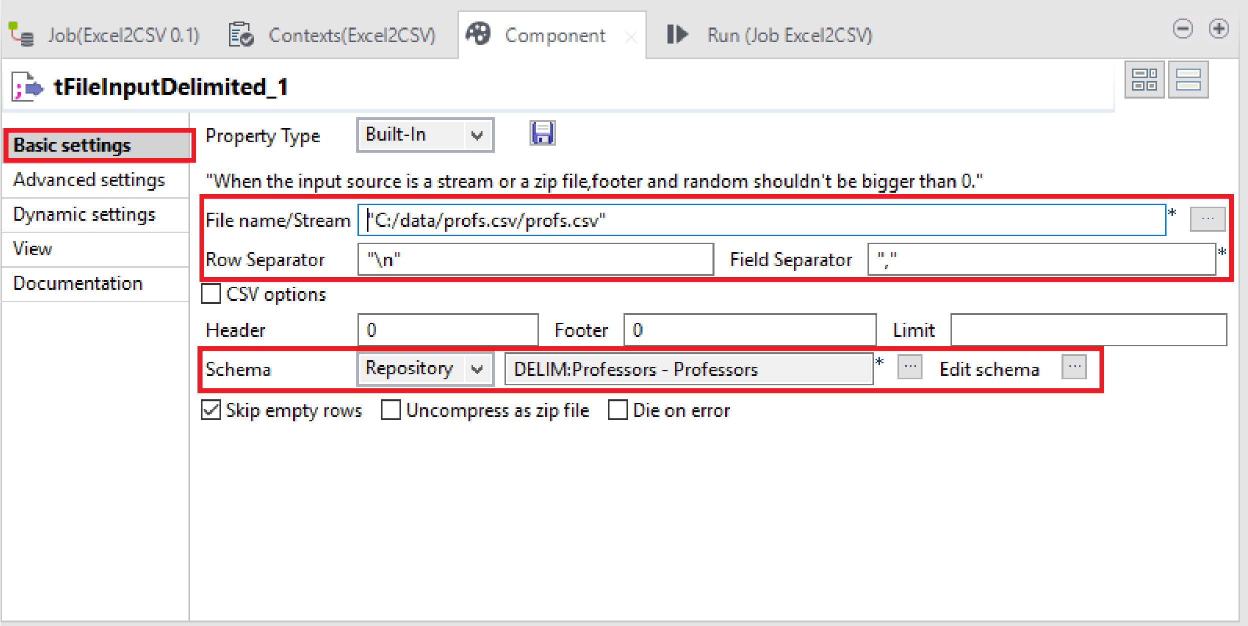


Figure 4-1 Designer View of Csv2Excel Job.

a. In the 'Basic settings' tab, the path of the input file (File name/Stream) shall be specified. (Figure 4-2)



Property Type: Built-In

"When the input source is a stream or a zip file,footer and random shouldn't be bigger than 0."

File name/Stream: "C:/data/profs.csv/profs.csv"

Row Separator: "\n"   Field Separator: ","

□ CSV options

Header: 0   Footer: 0   Limit:

Schema: Repository   DELIM:Professors - Professors   Edit schema

☑ Skip empty rows   □ Uncompress as zip file   □ Die on error

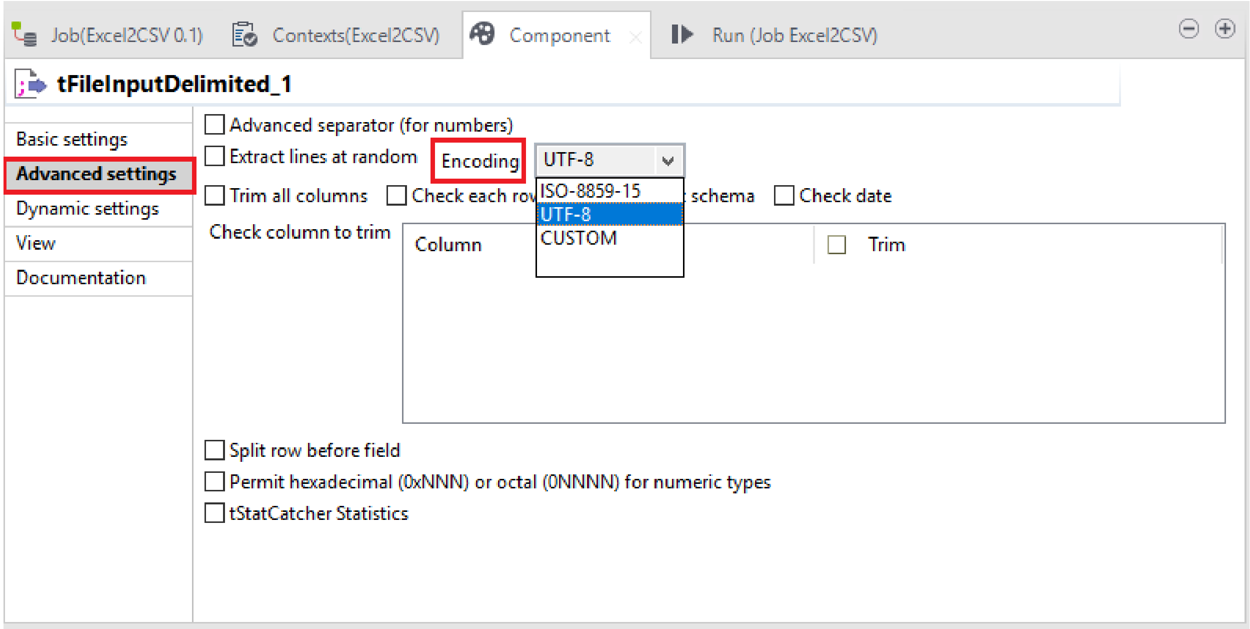
Figure 4-2 Basic settings tab of tFileInputDelimited component.

b. Also, the schema of the file shall be defined. The schema refers to the fields of the selected input file. Particularly the user can click on the 'Edit schema' option and the 'Schema Editor' opens. Within the 'Schema Editor', the user can edit the schema of the input file. (Figure 4-2)

c. The 'Row Separator' parameter is used for identifying the end of rows. e.g. "\n" stands for a new line. (Figure 4-2)

d. The 'Field Separator' parameter is used for separating the fields in the input data. e.g. "," or ":" can be used. (Figure 4-2)

e. In 'Advanced settings' tab the encoding can be defined, as well e.g. UTF-8. (Figure 4-3)



tFileInputDelimited_1

□ Advanced separator (for numbers)

□ Extract lines at random   Encoding UTF-8

ISO-8859-15
UTF-8
CUSTOM

□ Trim all columns   □ Check each row schema   □ Check date

Check column to trim

Column	□ Trim

□ Split row before field

□ Permit hexadecimal (0xNNN) or octal (0NNNN) for numeric types

□ tStatCatcher Statistics

Figure 4-3 Advanced settings tab of tFileInputDelimited Component

3. For the conversion of data to an excel file, a 'tFileOutputExcel' component is used that organizes the data in cells and displays all the rows of data in numbered rows and columns. (Figure 4-1)
4. Between the two components a 'Main' row connection is used for passing the data from one component to the other. (Figure 4-1)
5. After the execution of the Job, a new excel file is created containing all the records of the input schema.

4.2 Data Filtering and Replication

4.2.1 Filtering rows

The ETL tool can replicate data from any source schema. The tReplicate component replicates the incoming schema into two or more similar output flows (Talend, tReplicate). On the same input schema, different operations can be performed. An example is shown below. The following components have been used for the Job:

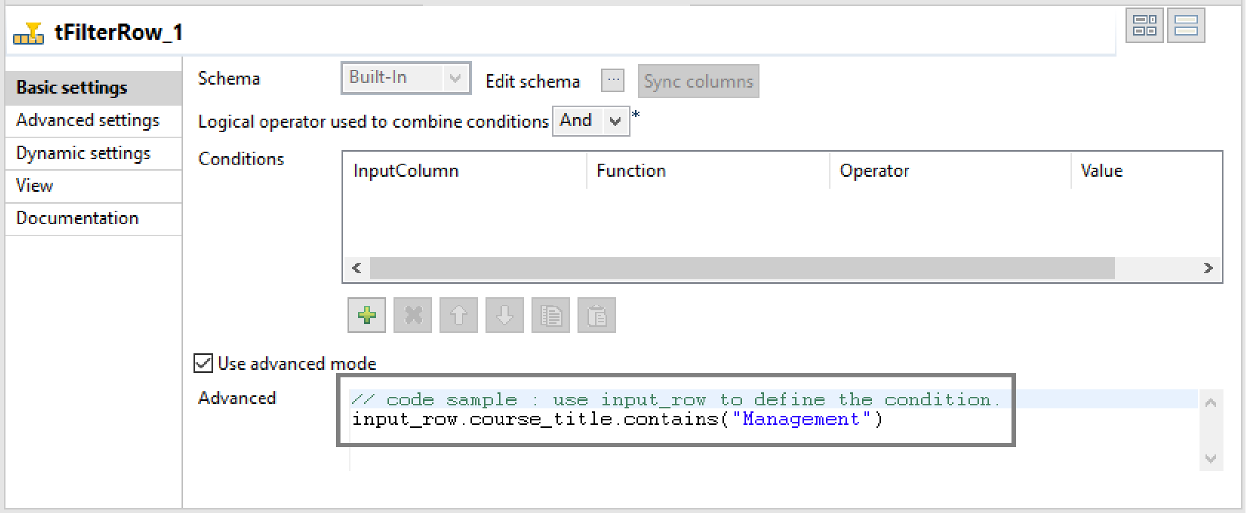
1. tFileInputDelimited
2. tReplicate

3. tFilterRow
4. tFileOutputDelimited

In this Job, the input schema contains the data from the CSV file of the 'Professors'. The 'tReplicate' component replicates the input flow into two similar output flows. Then the 'tFilterRow' component filters the output flows of the 'tReplicate' component by setting specific conditions on selected columns (Talend, tFilterRow).

In this scenario, the Job will create two output flows and specifically two CSV files. The first CSV file (Output1) will contain all the professors that teach courses which contain the word 'Management' in Greek. The second CSV file (Output2) will output only the professors that teach courses for the Interdepartmental Postgraduate Program in Business Administration. 'tReplicate' can be combined with other jobs as well.

In the tFilterRow Component View (Figure 4-4), the expression "*input_row.course_title.contains("Management")*" defines a condition on the 'course_title' column of the input data. This condition filters the rows that contain the word "Management".



Schema: Built-In

Logical operator used to combine conditions: And *

☑ Use advanced mode

Advanced:

```
// code sample : use input_row to define the condition.
input_row.course_title.contains("Management")
```

Figure 4-4 tFilterRow component View

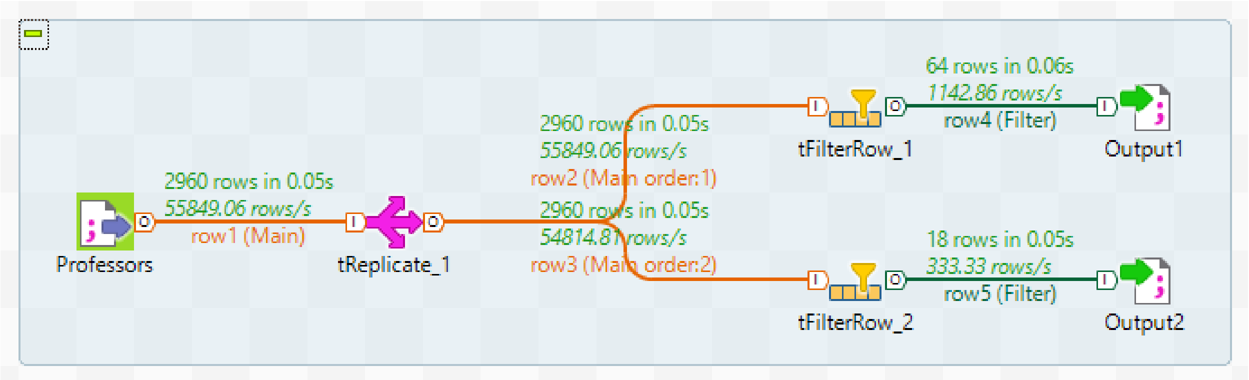


Figure 4-5 Results of ETL Job

After executing the job, the following results were outputted: (Figure 4-5)

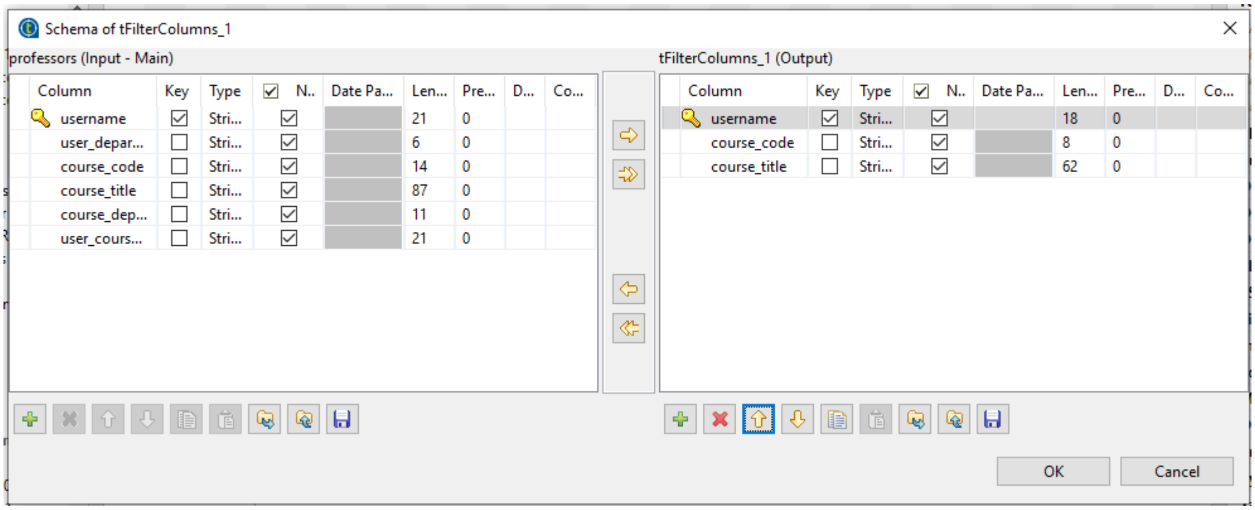
- For Output1, out of the 2960 rows, only 64 records contained the professors that teach courses which contain the word 'Management'.
- For Output2, out of 2960 rows, only 18 records contained the professors that teach courses for the Interdepartmental *Postgraduate Program* in Business Administration.

4.2.2 Filtering columns

Filtering columns defines which columns in the input schema shall be modified. For example, the order of the columns may change, new columns can be added in the input schema or unwanted columns may get removed (Talend, tFilterColumns). The scenario in Figure 4-7 describes a job which filters the input schema columns and removes several columns. The components used are:

1. tFileInputDelimited
2. tFilterColumns
3. tFileOutputDelimited

In the Component View of 'tFilterColumns' (Figure 4-6), the input schema (professors) can be edited and a new output schema can be defined.



Schema of tFilterColumns_1

professors (Input - Main)

Column	Key	Type	☑ N..	Date Pa...	Len...	Pre...	D...	Co...
username	☑	Stri...	☑		21	0		
user_depar...	☐	Stri...	☑		6	0		
course_code	☐	Stri...	☑		14	0		
course_title	☐	Stri...	☑		87	0		
course_dep...	☐	Stri...	☑		11	0		
user_cours...	☐	Stri...	☑		21	0		

tFilterColumns_1 (Output)

Column	Key	Type	☑ N..	Date Pa...	Len...	Pre...	D...	Co...
username	☑	Stri...	☑		18	0		
course_code	☐	Stri...	☑		8	0		
course_title	☐	Stri...	☑		62	0		

Figure 4-6 Editing the input schema

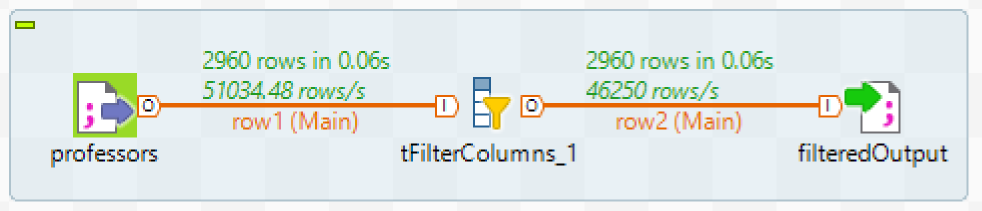


Figure 4-7 Results of ETL Job

After running the Job, the output schema contains the same number of rows as the input schema (2960 rows), but the number of columns is not the same. The output schema contains only three columns: 'username', 'course_code' and 'course_title' (Talend, Configuring the process of matching data).

4.2.3 Sorting rows

The 'tSortRow' component is used for sorting input data based on sort type (numerical/alphabetical) and sort order (ascending or descending) (Talend, tSortRow). The scenario in Figure 4-8 describes a job which sorts the professors' data based on criteria, such as sort by 'username', 'department' and 'course title'. The following components are used:

1. tFileInputDelimited
2. tReplicate
3. tSortRow (*The tSort component is used three times)*
    a. sort data by 'course_department'
    b. sort data by 'course_title'

    c. sort data by ‘username’

4. tFileOutputXML
5. tFileOutputJSON
6. tFileOutputExcel

After the execution of the job, three output files are generated.

- a. The first output is an Excel file which contains all the professors (2960 rows) ranked by the column ‘course_department’.
- b. The second output file is a JSON file ranking all professors (2960 rows) by the column ‘course_title’.
- c. The third output file is an XML file consisting of all the professors (2960 rows) sorted by the column ‘username’.

As mentioned above, all the output files contain the same number of rows (2960 rows). With data sorting, the data is arranged into some meaningful order for understanding and analyzing.

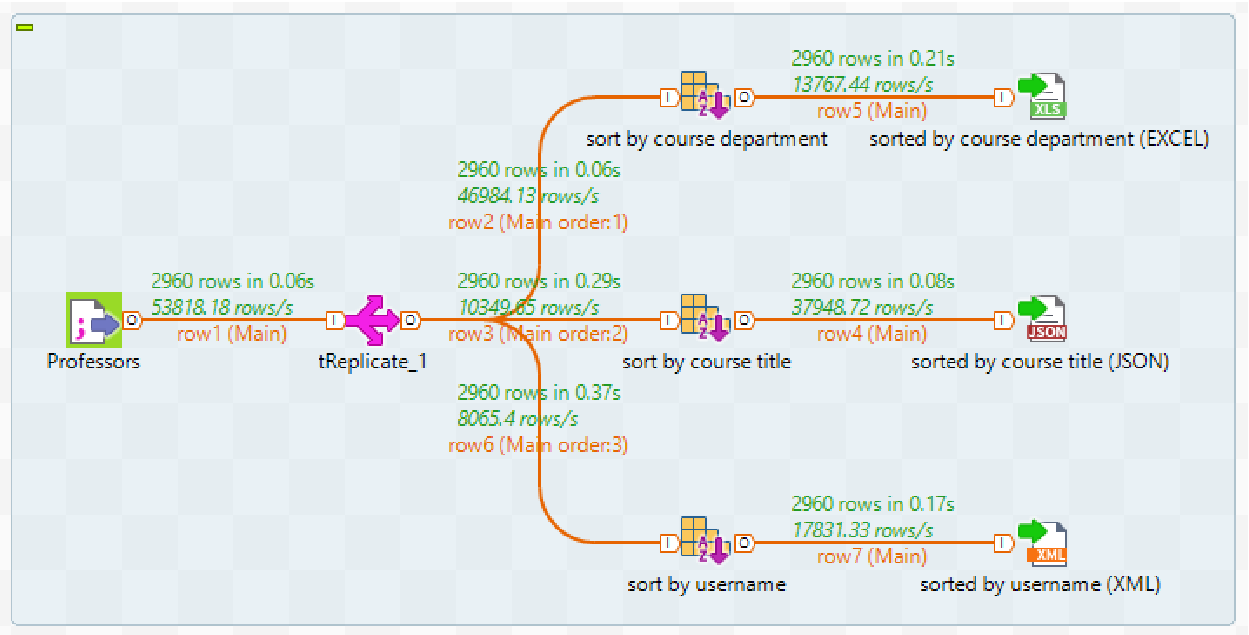


Figure 4-8 Sorting rows

4.2.4 Replacing data

The ‘tReplace’ component performs search and replace operations on columns of the input schema. In the following job the ‘tReplace’ component searches and replaces typos or strings in the csv input file (students.csv). Then, it performs a sorting operation on the students’ name before outputting the results to the new excel file output (StudentsOutput).

The components used in the Job are the following: (Figure 4-9)

1. tFileInputDelimited
2. tReplace
3. tSortRow
4. tFileOutputExcel

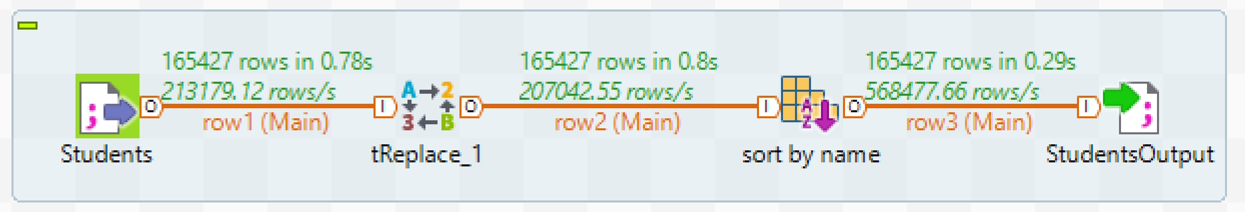
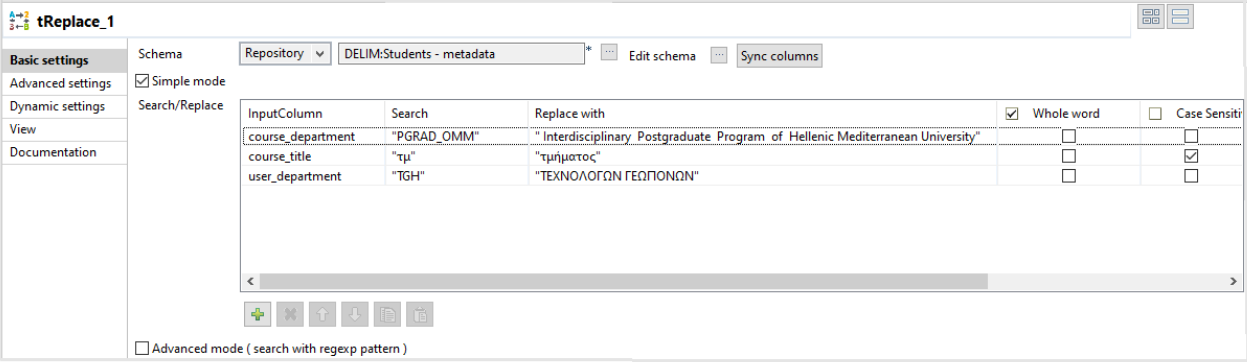


Figure 4-9 Results of ETL Job

In the Component View of ‘tReplace’ (Figure 4-10), the user can set the search and replace parameters. The ‘Search’ and ‘Replace with’ fields are essential for changing the values of input columns.

- The first input column selected for replacement is “course_department”. Particularly, in the ‘Search’ field “PGRAD_OMM” is typed and in the ‘Replace with’ field “Interdisciplinary *Postgraduate Program* of Hellenic Mediterranean University” is defined.
- The second input column selected for replacement is “course_title”. Typed “τμ” in ‘Search field’ and “τμήματος” in ‘Replace with’ field.
- The third input column is “user_department”. Typed “TGH” in ‘Search field’ and “ΤΕΧΝΟΛΟΓΩΝ ΓΕΩΠΟΝΩΝ” in ‘Replace with’ field.



tReplace_1

Schema: Repository | DELIM:Students - metadata | Edit schema | Sync columns

☑ Simple mode

Search/Replace

InputColumn	Search	Replace with	☑ Whole word	☐ Case Sensiti
course_department	"PGRAD_OMM"	" Interdisciplinary Postgraduate Program of Hellenic Mediterranean University"	☐	☐
course_title	"τμ"	"τμήματος"	☐	☑
user_department	"TGH"	"ΤΕΧΝΟΛΟΓΩΝ ΓΕΩΠΟΝΩΝ"	☐	☐

☐ Advanced mode (search with regexp pattern)

Figure 4-10 tReplace Component View

After the execution of the job, a new Excel file was created with the newly replaced values of the input columns.

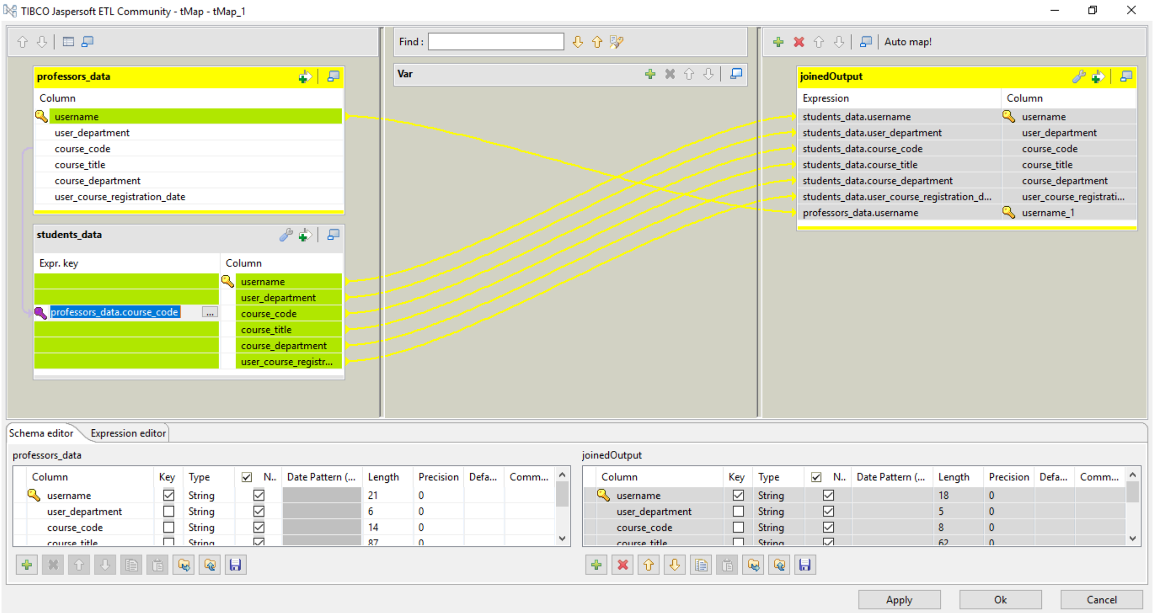
4.3 Joining data sources

The ETL tool, except for transforming, filtering or merging data, it can join data from multiple data sources. This section describes the process of joining two data sources from the e-class platform of the Hellenic Mediterranean University (https://eclass.hmu.gr/). The first data source is a csv file containing professors' data *(2960 records).* Similarly, the second data source is a csv file containing students' data *(165.428 records).*

4.3.1 Operations of 'tMap' Component

The 'tMap' component is one of the most used components in JasperSoft ETL tool. It is used for mapping input data to output data and for joining data from heterogeneous or homogeneous sources. The tMap component provides a list of parameters to configure the input and output flows. Within the graphical tool, 'Map Editor' (Figure 4-11), the manipulation of data is easier and more organized (Balkenende, 2018).

Additionally, 'tMap' can prefilter and transform the input data. It provides the Expression Builder editor for data transformation and for writing filter expressions (Talend, tMap Joins & Filtering, 2020; Talend, Using the expression editor, 2020).



TIBCO Jaspersoft ETL Community - tMap - tMap_1

professors_data

Column
username
user_department
course_code
course_title
course_department
user_course_registration_date

students_data

Expr. key	Column
	username
	user_department
professors_data.course_code	course_code
	course_title
	course_department
	user_course_registr...

joinedOutput

Expression	Column
students_data.username	username
students_data.user_department	user_department
students_data.course_code	course_code
students_data.course_title	course_title
students_data.course_department	course_department
students_data.user_course_registration_d...	user_course_registrati...
professors_data.username	username_1

professors_data

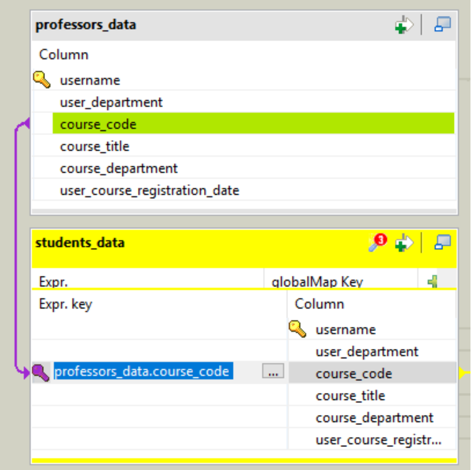
Column	Key	Type	☑ N..	Date Pattern (...	Length	Precision	Defa...	Comm...
username	☑	String	☑		21	0		
user_department	□	String	☑		6	0		
course_code	□	String	☑		14	0		
course_title	□	String	☑		87	0		

joinedOutput

Column	Key	Type	☑ N..	Date Pattern (...	Length	Precision	Defa...	Comm...
username	☑	String	☑		18	0		
user_department	□	String	☑		5	0		
course_code	□	String	☑		8	0		
course_title	□	String	☑		62	0		

Figure 4-11 Map Editor of tMap component

In the example used in the current section, the two inputs have been joined by the common column 'course_code'. (Figure 4-12). When joining different data sources, there is always one main input (e.g. Professors) which is considered the primary flow and multiple (one or more) lookup flows (e.g. Students) connected to the tMap component. The default Lookup Model is 'Load once'. A brief description of the different Lookup models is presented in the next sections.



professors_data

Column
username
user_department
course_code
course_title
course_department
user_course_registration_date

students_data

Expr.	globalMap Key

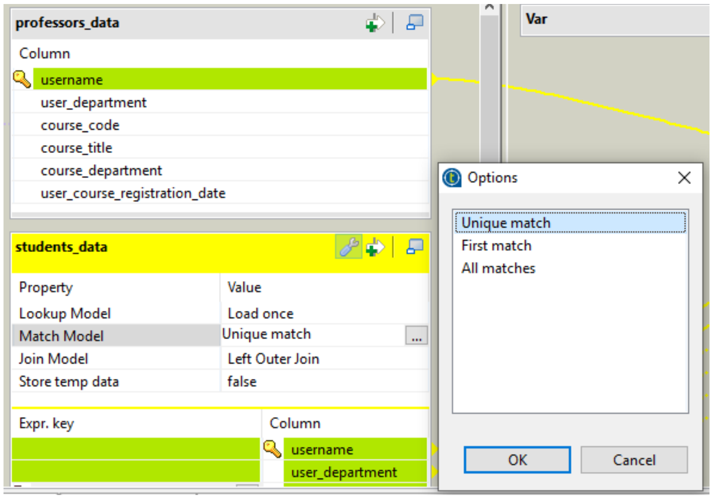
Expr. key	Column
	username
	user_department
professors_data.course_code	course_code
	course_title
	course_department
	user_course_registr...

Figure 4-12 Joining two data sources by the column 'course_code'

4.3.2 Match Model

The ‘tMap’ component provides three match models: **Unique Match, First Match and All Matches.** An ETL Job can implement different Match Models. (Figure 4-13) As a result different results are generated (Talend, The differences between Unique match, First match and All matches, 2019).

1. **Unique match:** The last matching record of the look up source is outputted. This is the default option when implementing a JOIN operation.
2. **First match:** The first matching record of the look up source is outputted.
3. **All matches:** All matching records of the look up source are shown.



professors_data

Column
username
user_department
course_code
course_title
course_department
user_course_registration_date

students_data

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Left Outer Join
Store temp data	false

Expr. key	Column
	username
	user_department

Figure 4-13 Configuring Match Models in Map Editor

4.3.3 Join Model

Join operations can be performed on the input data. The ‘tMap’ component provides two different types of join: ‘Left Outer Join’ and ‘Inner Join’ which can be configured within the ‘Map editor’ tool.

4.3.3.1 Left Outer Join

‘Left Outer Join’ is the default join option for joining data sources. The ETL job described in Figure 4-14 performs a left outer join between the main data source (professors) and the lookup data source (Students). The output produced contains all the students joined with professors based on the common column ’course_code’.

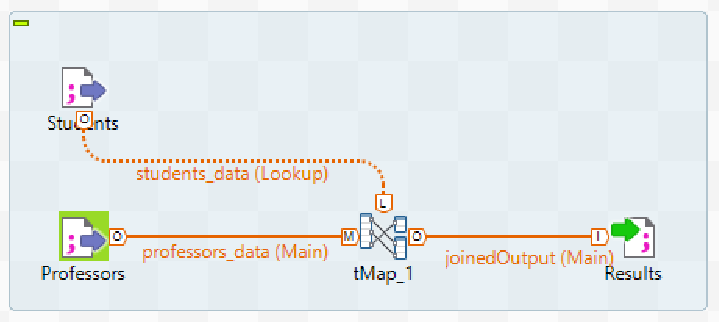
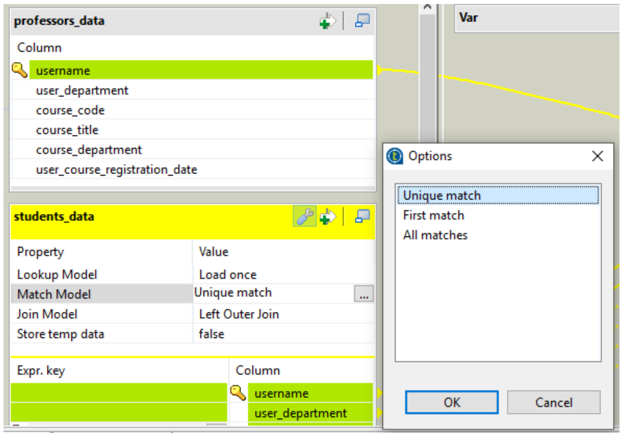


Figure 4-14 tMap Component

The above ETL job (Figure 4-14) has been implemented by using the three different match models. After executing the job, a set of different results was produced (Talend, Example Job implementing the different match models, 2019; Talend, Defining the match model for an explicit Join).



professors_data

Column
- username
- user_department
- course_code
- course_title
- course_department
- user_course_registration_date

students_data

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Left Outer Join
Store temp data	false

Expr. key	Column
	username
	user_department

Figure 4-15 Match Models in tMap Component 'Unique match', 'First match', 'All matches'

First run of ETL Job with the following configuration parameters:

1. **Join Model**: Left Outer Join
2. **Match Model**: Unique match
3. **Lookup Model**: Load once

After the execution of the Job, the output contained a set of 2960 rows. By choosing the option 'Unique match', the last matching record of the lookup source (Students) was outputted. (See Figure 4-16)

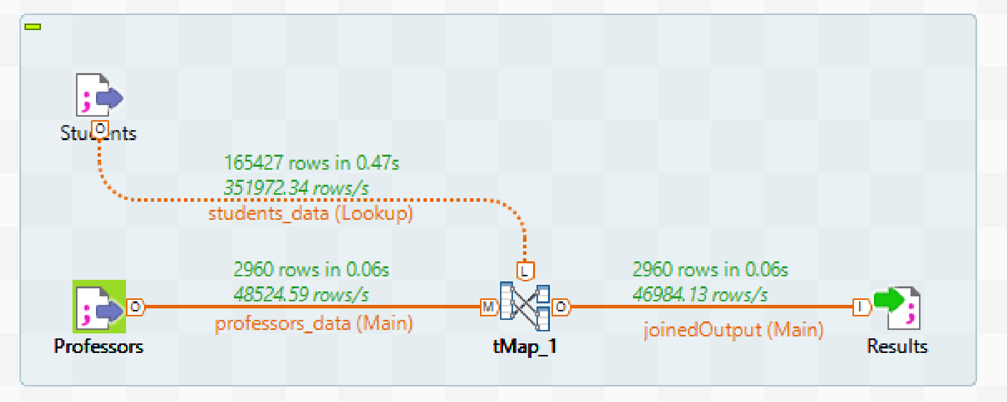


Figure 4-16 Joining data sources with 'Unique match' model

Second run of ETL Job with the following configuration parameters:

1. **Join Model:** Left Outer Join
2. <u>**Match Model:** First match</u>
3. **Lookup Model:** Load Once

After rerunning the ETL Job, the output contained the same number of rows: 2960 rows, but with different content. By setting the option to 'First match', the first matching record of the lookup source (Students) was produced. (See Figure 4-17)

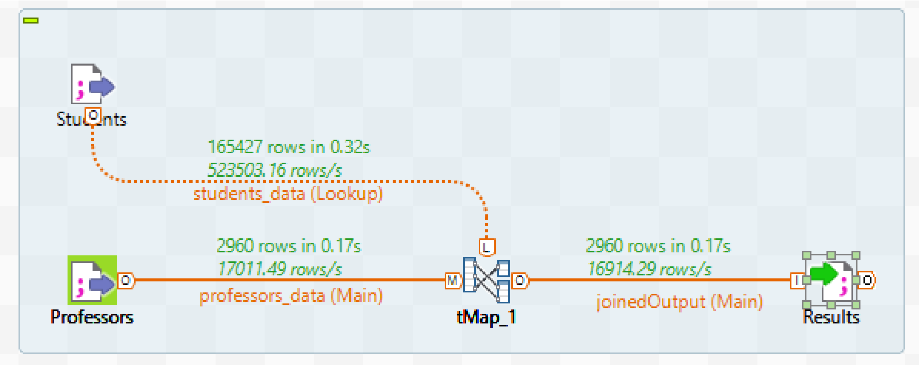


Figure 4-17 Joining data sources with 'First match' Model

Third run of ETL Job with the following configuration parameters: (Figure 4-18)

1. **Join Model:** Left Outer Join
2. <u>**Match Model:** All matches</u>
3. **Lookup Model:** Load Once

After the job executed, the output consisted of 292786 rows. With option 'All matches', all matching records of the look up source were shown.

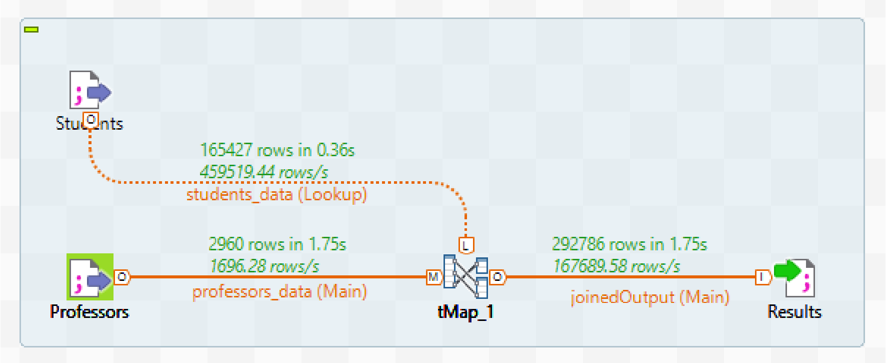


Figure 4-18 Joining data sources with All Matches Model

4.3.3.2 Inner Join

Inner join does not allow null values to be passed to the main output flow. Rejected rows are passed to a specific Inner Join Reject output. For the following ETL job (Figure 4-19), the option 'Inner Join' was used for joining the two data sources. Also, the 'All matches' option was used.

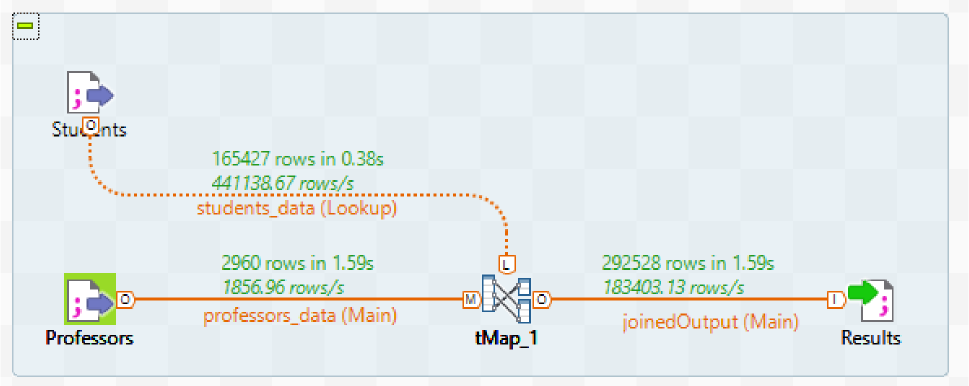


Figure 4-19 ETL Job

After running the job, the inner join produced matches for 292528 rows and as a result it rejected the other rows. Compared to 'Left Outer Join', the output returned by 'Inner Join' had 258 rows less.

With the creation of a second output 'Rejected Results' in the tMap component (Figure 4-20), it is possible to collect the inner join rejects by changing the 'Catch lookup inner join reject' property to 'true' (Figure 4-21).

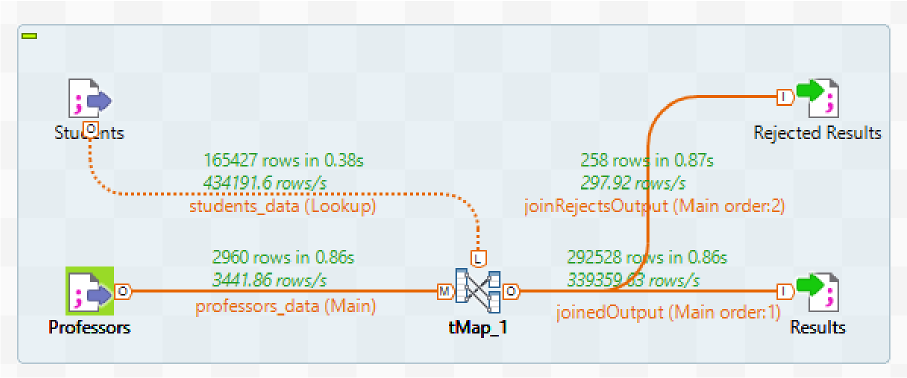
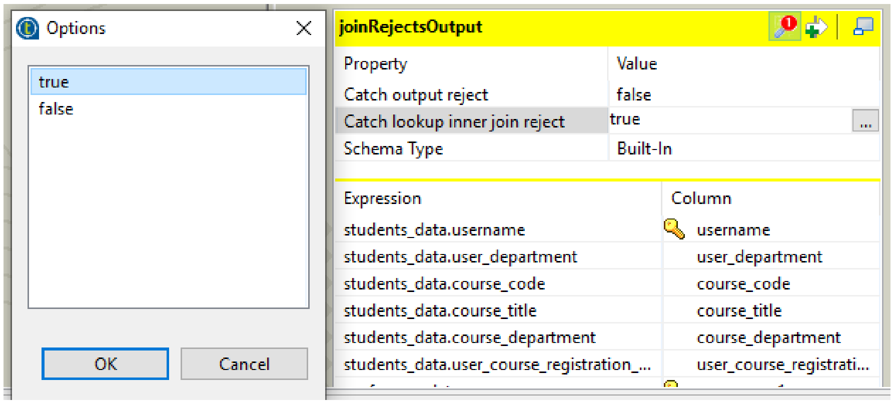


Figure 4-20 Inner Join with two outputs



joinRejectsOutput

Property	Value
Catch output reject	false
Catch lookup inner join reject	true
Schema Type	Built-In

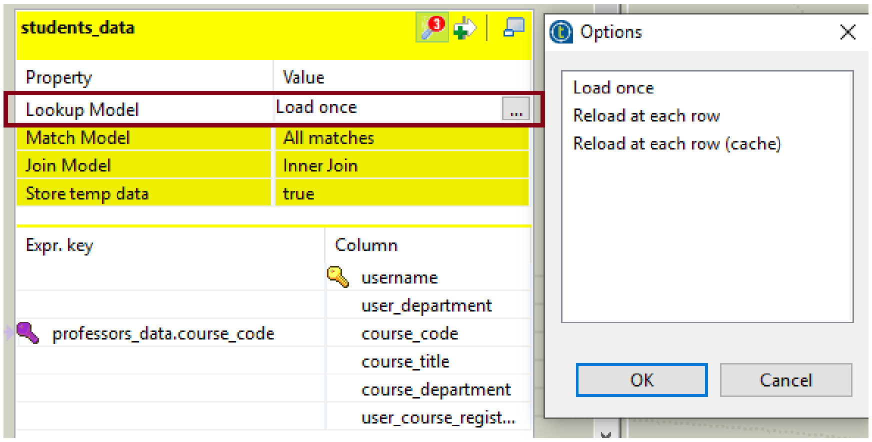
Expression	Column
students_data.username	username
students_data.user_department	user_department
students_data.course_code	course_code
students_data.course_title	course_title
students_data.course_department	course_department
students_data.user_course_registration_...	user_course_registrati...

Figure 4-21 Catch lookup inner join reject option

In the Job designer window, 258 rows are collected in the new ‘tFileOutputDelimited’ named ‘Rejected Results’ and 292528 rows are included in the ‘Results’ output. So, all the rows of data that were rejected by the inner join are stored in a new output.

4.3.4 Lookup Model

In the example presented, the two data sources have been joined by the common column ‘course_code’. The main input (Professors) connected to the ‘tMap’ component is considered the primary flow and the second input (Students) is considered the lookup flow. The default Lookup Model is ‘Load once’. A brief description of the different Lookup models is presented below.



students_data

Property	Value
Lookup Model	Load once
Match Model	All matches
Join Model	Inner Join
Store temp data	true

Expr. key	Column
	username
	user_department
professors_data.course_code	course_code
	course_title
	course_department
	user_course_regist...

Options

Load once
Reload at each row
Reload at each row (cache)

Figure 4-22 Lookup Models in tMap Component

Load once: All the records from the lookup flow are loaded only once in the memory or in a local file in case the option '*Store temp data'* has the true value. The student's data is considered as the look up flow and the professor's data as the main flow. The estimated execution time for this job is 2.49 seconds. (Figures 4-22, 4-23)

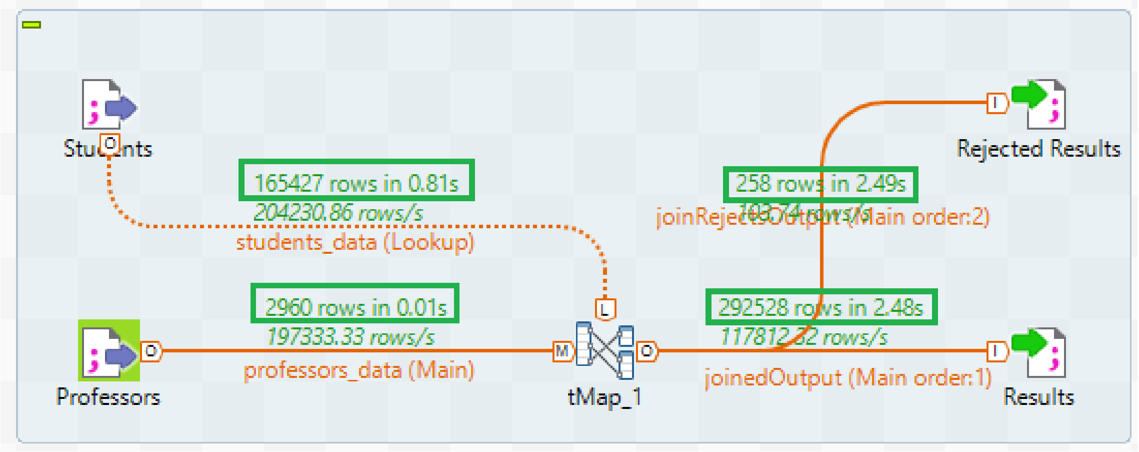
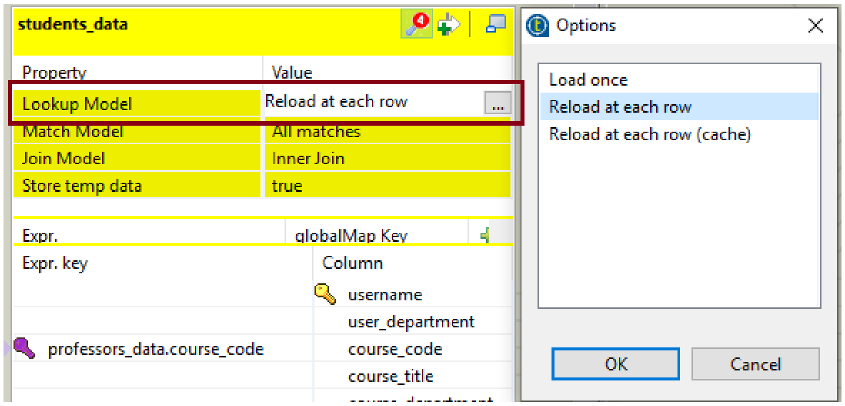


Figure 4-23 Results of ETL Job

Reload at each row: For every record of the main flow, all the records of the lookup flow will be reloaded. This type of Lookup model is used when the data in the lookup flow is updated in real-time and the latest look up data shall be loaded. One drawback is that the execution time increases. In Figure 4-25, the estimated job execution time of the mapping is 2186.87 seconds (36 minutes).



students_data

Property	Value
Lookup Model	Reload at each row
Match Model	All matches
Join Model	Inner Join
Store temp data	true

Expr.	globalMap Key

Expr. key	Column
	username
	user_department
professors_data.course_code	course_code
	course_title

Options

- Load once
- Reload at each row
- Reload at each row (cache)

Figure 4-24 Look up model

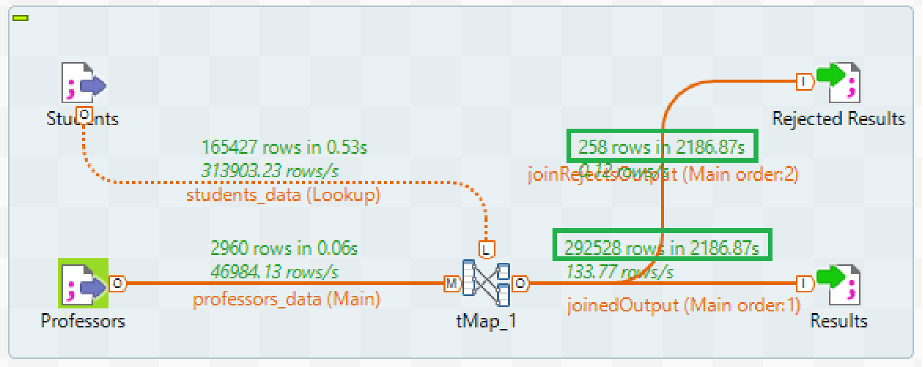


Figure 4-25 Results of ETL Job

Reload at each row (cache): It works as 'Reload at each row'. The lookup data is cached into memory. Only in the case of an update in the lookup flow, the data is loaded into the cache memory.

Execution time = 445 seconds (7.41 minutes) (Figures 4-26, 4-27)

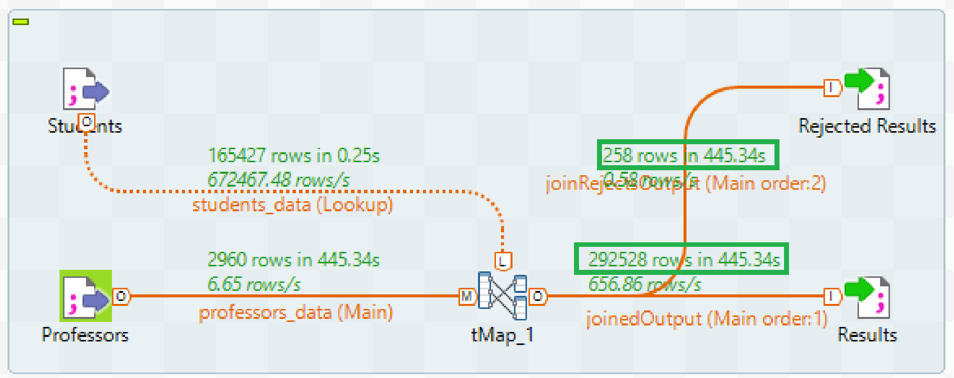
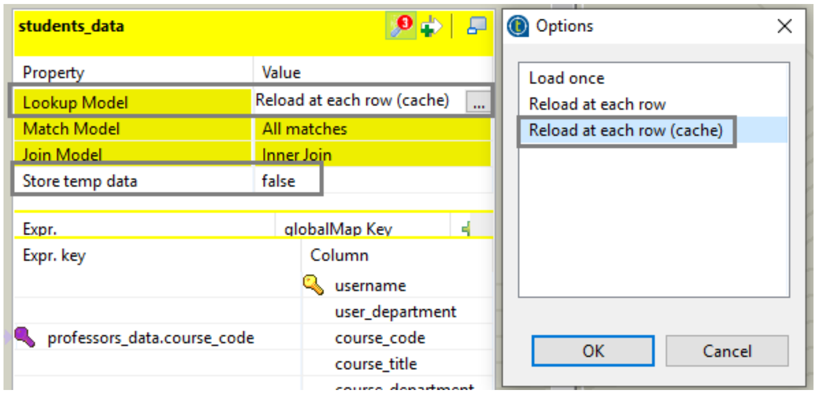


Figure 4-26 Results of ETL Job



students_data

Property	Value
Lookup Model	Reload at each row (cache)
Match Model	All matches
Join Model	Inner Join
Store temp data	false

Expr.	globalMap Key

Expr. key	Column
	username
	user_department
professors_data.course_code	course_code
	course_title

Options

- Load once
- Reload at each row
- Reload at each row (cache)

Figure 4-27 Look up Model (Reload at each row cache)

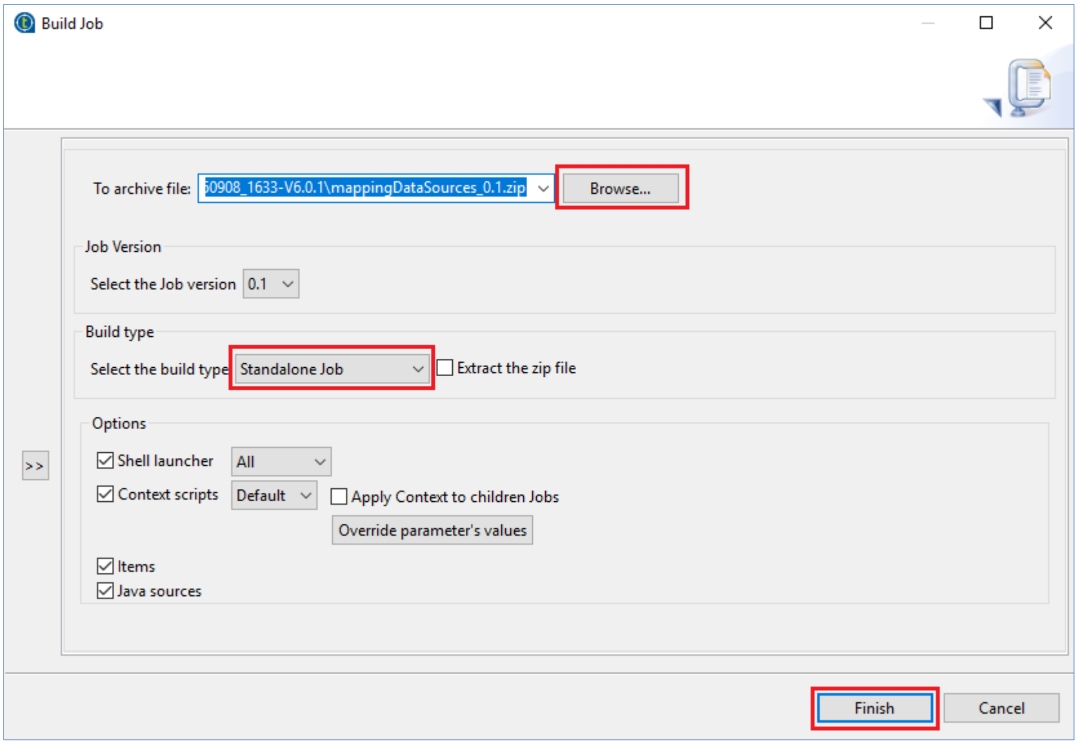
4.4 Call an ETL Job from a Java application

An ETL Job can be integrated into an external Java application. Eclipse is a Java Integrated Development Environment (IDE) that is used for developing applications in Java and in other programming languages, such as: C, C++, JavaScript etc. (Foundation; Contributors, 2020; Talend, Calling a Talend Job from an external Java application). An ETL Job can be integrated into Eclipse IDE. The free open source software 'Eclipse IDE for Enterprise Java Developers' (Version: 2019-09 R (4.13.0)) was used for the ETL Job integration.

Before importing the ETL Job into a Java project in Eclipse IDE, all the necessary files needed for the integration shall be exported from JasperSoft ETL tool. Therefore, the 'Build Job' feature in JasperSoft ETL tool enables the deployment and the independent execution of an ETL Job on any server by adding all the necessary files to an archive (Talend, How to build Jobs).

The steps for building an ETL Job are described in more details below.

1. After launching JasperSoft ETL tool and creating an ETL job, the new Job is listed in the 'Project Repository' View.
2. Right-click on it and select 'Build Job'. As a result, a new dialog opens (See figure 4-28)
   i. Specify a location for the zip archive file.
   ii. Select build type: 'Standalone Job'.
   iii. The checkbox `Shell launcher` shall be selected. By choosing option 'All' the .bat files (for Windows) and .sh files (for Unix) are exported, as well.
3. Click 'Finish'. An archive is created in the specified location.



Build Job

To archive file: 0908_1633-V6.0.1\mappingDataSources_0.1.zip

Job Version

Select the Job version: 0.1

Build type

Select the build type: Standalone Job

☐ Extract the zip file

Options

☑ Shell launcher: All

☑ Context scripts: Default

☐ Apply Context to children Jobs

☑ Items

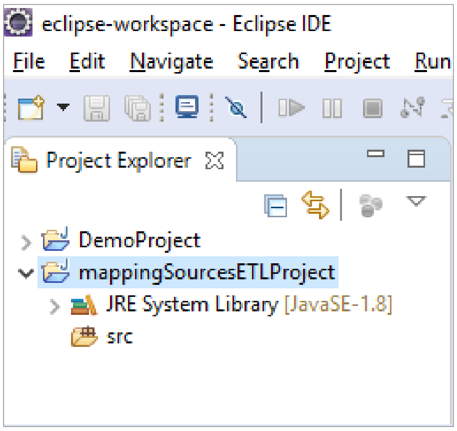
☑ Java sources

Figure 4-28 How to build an ETL Job in JasperSoft ETL.

Consequently, the produced archive includes all the JAR files that must be imported into Eclipse IDE. A JAR file stands for 'Java ARchive' and has a '.jar' file extension. It is a cross-platform package file format used for aggregating Java class files, resources (text, images, sound etc.) and metadata into one zip file (Wikipedia C. , JAR (file format), 2020; Oracle).

The steps for the integration process are described below.

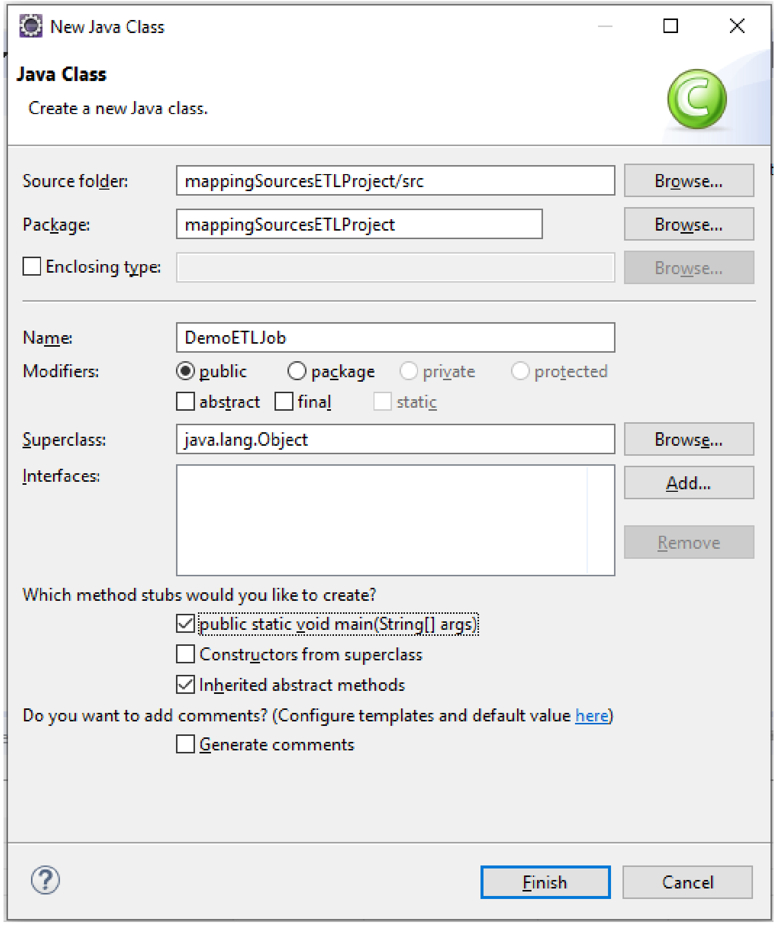
1. In Eclipse IDE, a Java project shall be created named 'mappedSourcesETLProject'. (Figure 4-29)



- DemoProject
- mappingSourcesETLProject
  - JRE System Library [JavaSE-1.8]
  - src

Figure 4-29 Creation of ‘mappingSourcesETLProject’ in Eclipse IDE.

2. Under the project, create a new Java class named: ‘DemoETLJob’ which will call the ETL Job. (Figure 4-30)



Source folder: mappingSourcesETLProject/src

Package: mappingSourcesETLProject

□ Enclosing type:

Name: DemoETLJob

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☑ public static void main(String[] args)

□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

□ Generate comments

Figure 4-30 Creation of new Java class named 'DemoETLJob' in Eclipse IDE

3. For the example described in this section, the integrated ETL Job performs a join between two different data sources, ‘Students’ and ‘Professors’. The jar files of the fore mentioned ETL Job shall be imported into the Java project.
   a. Right Click the root folder of the project ‘mappingSourcesETLProject’.
   b. Select ‘Build path’ > ‘Configure Build Path’. (Figure 4-31)
   c. A new window opens. Select ‘Java Build Path’.
   d. In the ‘Libraries’ tab, click on ‘Add External Jars’ in order to import all the dependent jars of the ETL job. (Figure 4-32)

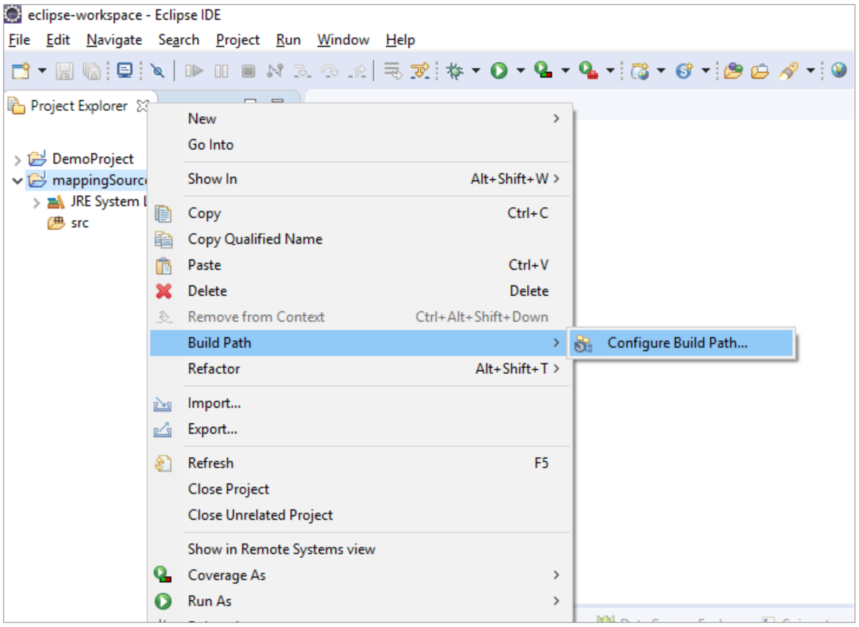
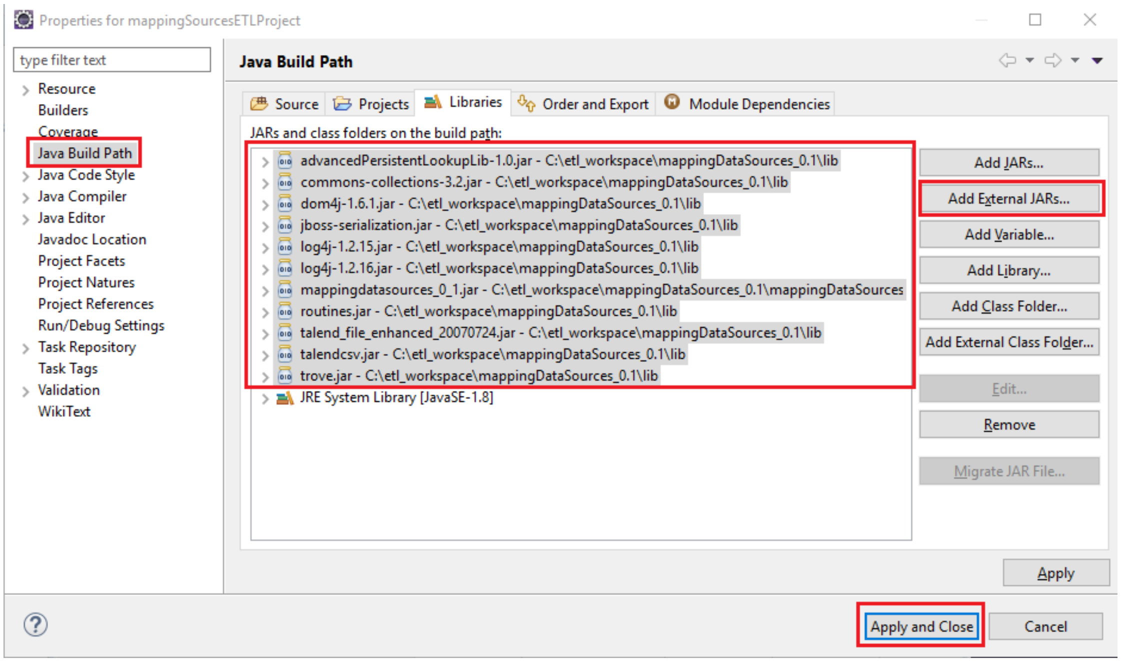


Figure 4-31 Configure Build Path of ‘mappingSourcseETLProject’ in Eclipse.



JARs and class folders on the build path:

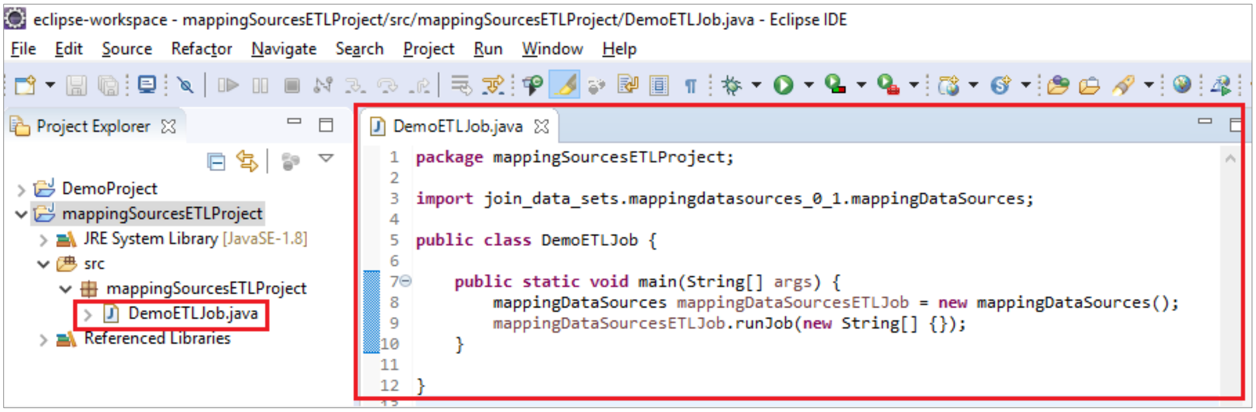
- advancedPersistentLookupLib-1.0.jar - C:\etl_workspace\mappingDataSources_0.1\lib
- commons-collections-3.2.jar - C:\etl_workspace\mappingDataSources_0.1\lib
- dom4j-1.6.1.jar - C:\etl_workspace\mappingDataSources_0.1\lib
- jboss-serialization.jar - C:\etl_workspace\mappingDataSources_0.1\lib
- log4j-1.2.15.jar - C:\etl_workspace\mappingDataSources_0.1\lib
- log4j-1.2.16.jar - C:\etl_workspace\mappingDataSources_0.1\lib
- mappingdatasources_0_1.jar - C:\etl_workspace\mappingDataSources_0.1\mappingDataSources
- routines.jar - C:\etl_workspace\mappingDataSources_0.1\lib
- talend_file_enhanced_20070724.jar - C:\etl_workspace\mappingDataSources_0.1\lib
- talendcsv.jar - C:\etl_workspace\mappingDataSources_0.1\lib
- trove.jar - C:\etl_workspace\mappingDataSources_0.1\lib
- JRE System Library [JavaSE-1.8]

Figure 4-32 Add External Jars in Eclipse Project

4. Edit the source code of the Java class 'DemoETLJob.java' and add the following code. This code calls the ETL Job. (Figure 4-33)

```
//Java Source Code
package mappingSourcesETLProject;
import join_data_sets.mappingdatasources_0_1.mappingDataSources;

public class DemoETLJob {
      public static void main (String[] args) {
            mappingDataSources mappingDataSourcesETLJob = new
mappingDataSources();
            mappingDataSourcesETLJob.runJob(new String[] {});
      }
}
//Java Source Code
```


eclipse-workspace - mappingSourcesETLProject/src/mappingSourcesETLProject/DemoETLJob.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
DemoProject
mappingSourcesETLProject
JRE System Library [JavaSE-1.8]
src
mappingSourcesETLProject
DemoETLJob.java
Referenced Libraries
DemoETLJob.java

```java
package mappingSourcesETLProject;

import join_data_sets.mappingdatasources_0_1.mappingDataSources;

public class DemoETLJob {

    public static void main(String[] args) {
        mappingDataSources mappingDataSourcesETLJob = new mappingDataSources();
        mappingDataSourcesETLJob.runJob(new String[] {});
    }

}
```

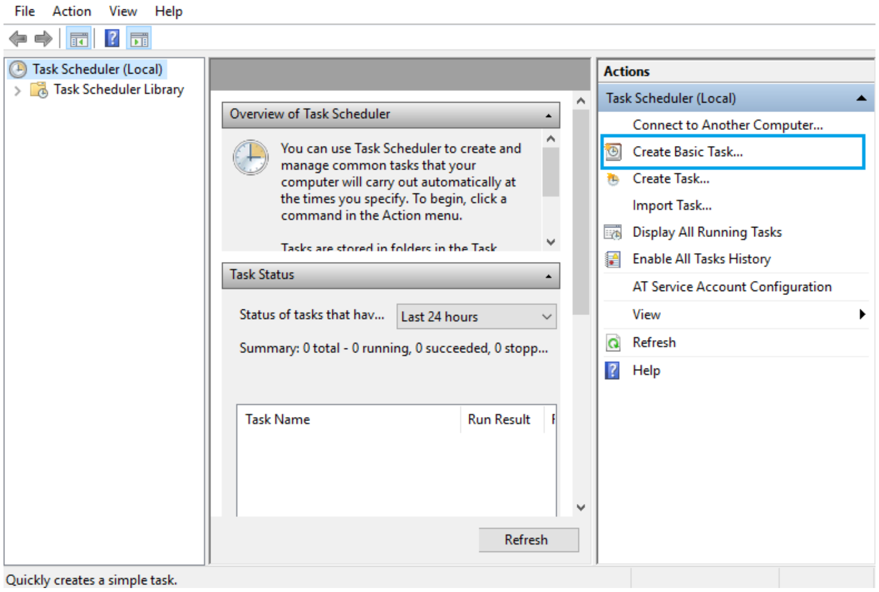
Figure 4-33 Calling an ETL Job from a Java Class in Eclipse IDE.

5. After the execution of the Java class, the ETL Job was successfully run. A new CSV file was produced which contained all the students joined with professors based on the common column ’course_code’.

4.5 Schedule ETL Job in Windows

In order to automatically run a job based on a schedule, an external scheduling tool can be used, such as ‘Task Scheduler’ of Windows. Microsoft Windows provides the ‘Task Scheduler’ tool which is used for scheduling programs or scripts (Wikipedia C. , Windows Task Scheduler, 2020). In order to schedule the automatic and independent execution of an ETL job a set of steps must be followed:

1. Launch the ‘Task Scheduler’ in Microsoft Windows. (Figure 4-34)



You can use Task Scheduler to create and manage common tasks that your computer will carry out automatically at the times you specify. To begin, click a command in the Action menu.

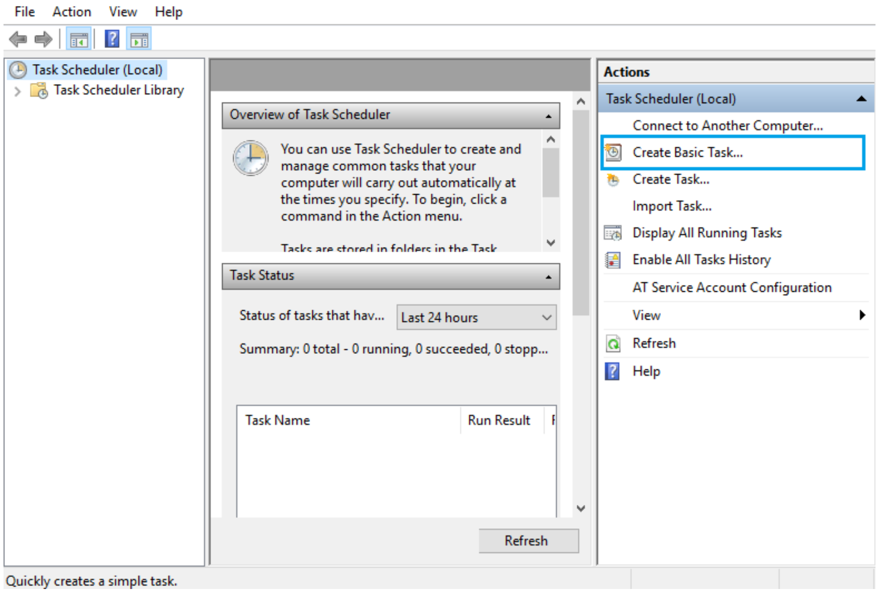
Tasks are stored in folders in the Task

Summary: 0 total - 0 running, 0 succeeded, 0 stopp...

Quickly creates a simple task.

Figure 4-34 Task Scheduler in Microsoft Windows.

2. From the Actions pane, select 'Create Basic Task' (Figures 4-35, 4-36)
    a. In the field 'Name', type the name of the task.
    b. In the field 'Description', type a brief description.



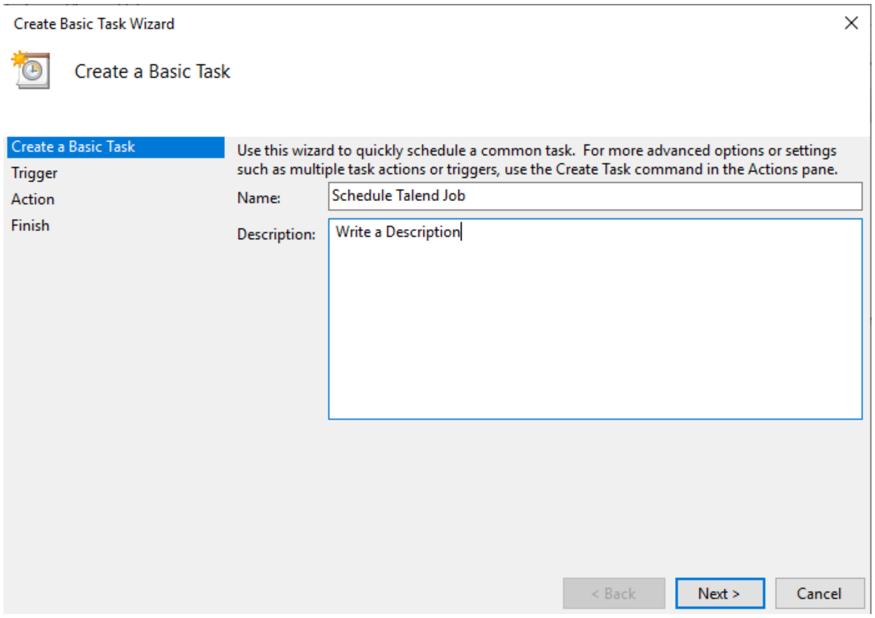
You can use Task Scheduler to create and manage common tasks that your computer will carry out automatically at the times you specify. To begin, click a command in the Action menu.

Tasks are stored in folders in the Task

Summary: 0 total - 0 running, 0 succeeded, 0 stopp...

Quickly creates a simple task.

Figure 4-35 How to create a Basic Task in Windows Task Scheduler.



Create Basic Task Wizard

Create a Basic Task

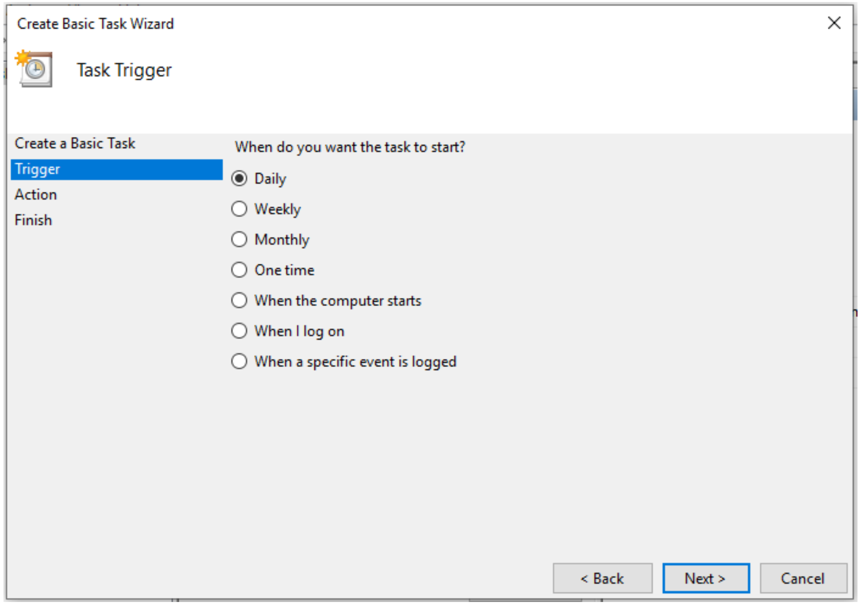
Use this wizard to quickly schedule a common task. For more advanced options or settings such as multiple task actions or triggers, use the Create Task command in the Actions pane.

Name: Schedule Talend Job

Description: Write a Description

Figure 4-36 Definition of Name and Description of a Basic Task in Windows Task Scheduler.

3. In the Trigger pane, select when the task will run: Daily, Weekly, monthly. (Figure 4-37)
   a. Based on the option chosen (e.g. Daily), a list of configuration parameters shall be defined, such as date and time of the execution. (Figure 4-38)



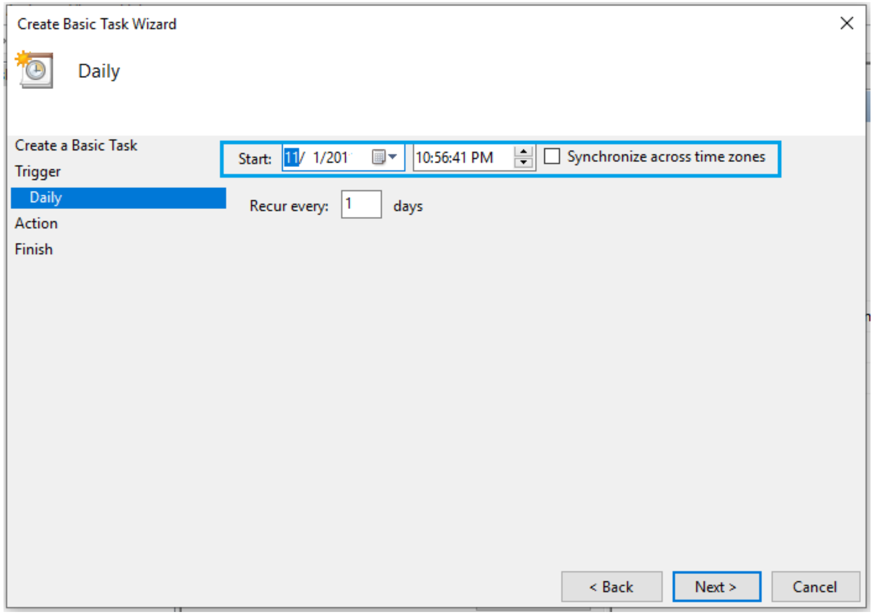
Create Basic Task Wizard

Task Trigger

When do you want the task to start?

● Daily
○ Weekly
○ Monthly
○ One time
○ When the computer starts
○ When I log on
○ When a specific event is logged

Figure 4-37 Define configuration parameters in Windows Task Scheduler.

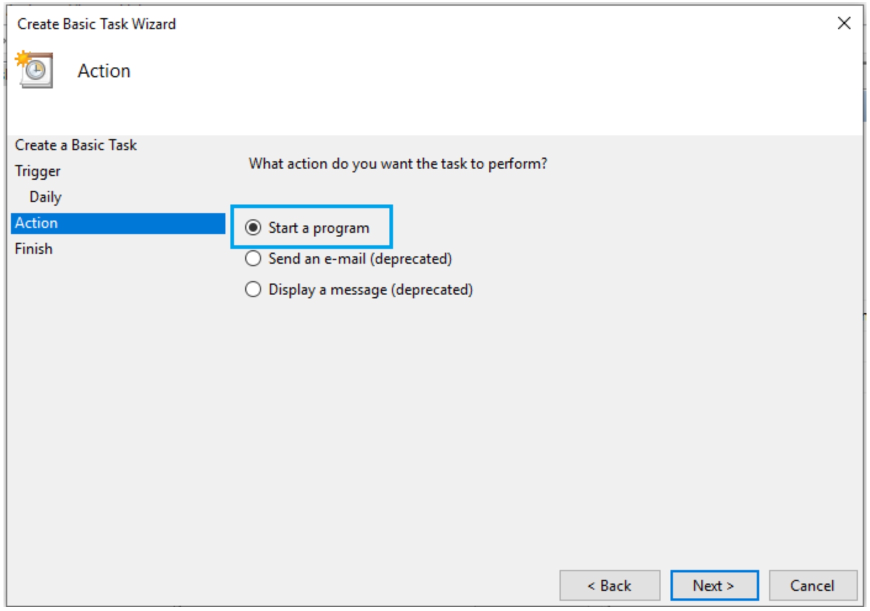


Start: 11/ 1/201 10:56:41 PM □ Synchronize across time zones

Recur every: 1 days

Figure 4-38 Task Triggering: setting starting date and time.

4. Choose an Action that the task will perform. In this example, `Start a Program` is selected. (Figure 4-39)



What action do you want the task to perform?

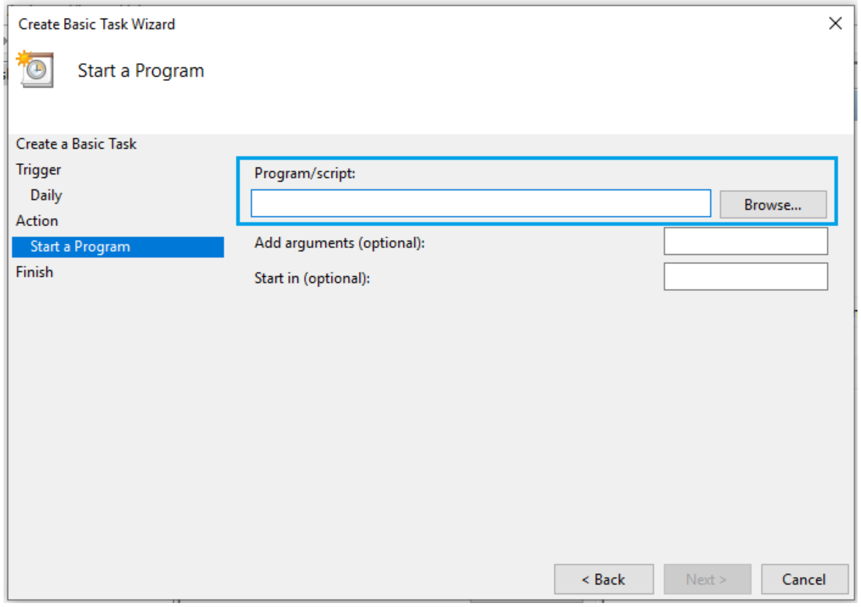
● Start a program

○ Send an e-mail (deprecated)

○ Display a message (deprecated)

Figure 4-39 Setting the action of task.

5. Click on ‘Browse’ in order to select the batch file of the ETL Job. (Figure 4-40)



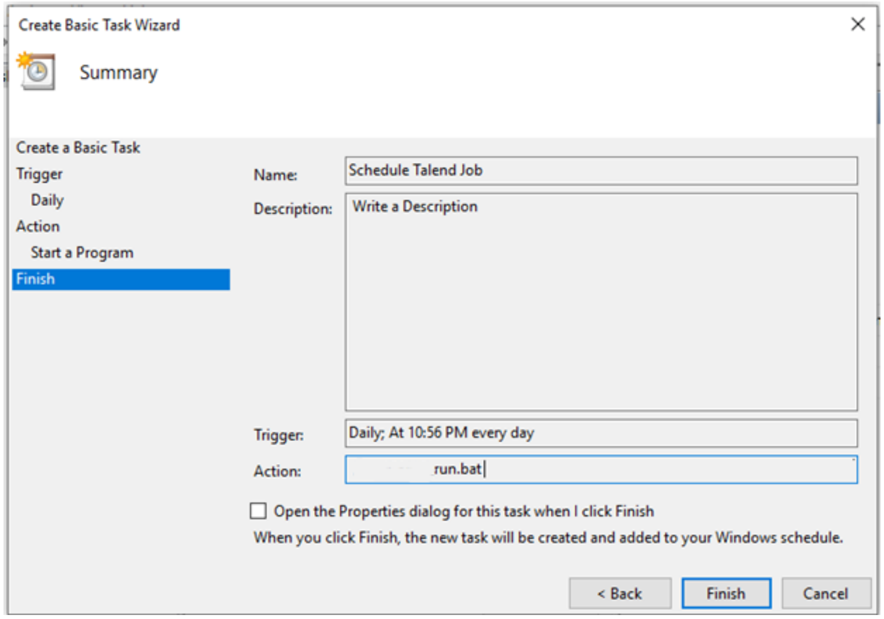
Program/script:

Add arguments (optional):

Start in (optional):

Figure 4-40 Setting the location of bat file.

6. Click on 'Finish'. The task is created. (Figure 4-41)



Name: Schedule Talend Job

Description: Write a Description

Trigger: Daily; At 10:56 PM every day

Action: run.bat

□ Open the Properties dialog for this task when I click Finish

When you click Finish, the new task will be created and added to your Windows schedule.

Figure 4-41 Summary of the task's settings and configurations.

As a result, the ETL job will run daily at a specific time.

5 Conclusions and Future Work

This thesis intends to propose a Data Warehouse (DW) solution for the Hellenic Mediterranean University. An introduction to Data Warehouse concepts is performed. In contrast to OLTP systems which support the daily transactions of a business enterprise, a DW emphasizes on storing large volumes of data and on improving decision making. Although, OLTP systems play an essential role for the successful operations of an enterprise, these systems are mainly used for transactions, query processing and not for decision making.

With the usage of BI technologies, an integration of different data sources from the online learning system of Hellenic Mediterranean University is performed. Specifically, the JasperSoft ETL Community tool is used. ETL tools extract, transform and load data from heterogeneous sources. They enable the integration of data in a faster and cost-effective way. Instead of writing traditional code with complex queries, users can define mappings between the source schemas (data sources) and the target schemas. Additionally, data quality is improved since ETL tools contain data cleansing features for removing inconsistencies and errors.

In this work, ETL processes were designed, implemented and executed. The ETL processes can be imported to other external applications and run independently. Research indicates that the use of a BI technique can be applied to a larger scale of information systems of the Hellenic Mediterranean University in order to integrate data effectively for supporting decision-making processes.

This thesis addresses the potential future work that must be done, concerning the designing of the ETL integration processes between the different information systems in the Hellenic Mediterranean University. For the accomplishment of that purpose, an open source ETL tool is proposed, that facilitates data collection, transformation and loading operations. This future work necessarily involves the participation of interested parties for the implementation of the Data Warehouse (DW). Interested parties in the academic community should focus on the structuring of the future steps which include the designing of the ETL procedures. The ETL procedures can be divided into three kinds of designing phases including (a) the extraction of data from the different data sources (b) data cleansing and transforming and (c) the loading of data into a newly created Data Warehouse. At the end, a centralized data repository containing information from all the heterogeneous information systems within the Hellenic Mediterranean University can be implemented. Furthermore, OLAP technology can be used for better analysis and reporting. The main goal is the development of a Business Intelligence System that will perform decision making processes for the managers of the academic community in the Hellenic Mediterranean University.

6 Bibliography

adverity. (n.d.). *THE DEFINITIVE GUIDE: ETL – EXTRACT TRANSFORM LOAD.* Retrieved 01 02, 2020, from https://www.adverity.com/etl-data-integration-analytics-quality/

Aquila, D. C., Tria, D. F., Lefons, E., & Tangorra, F. (2008). Business intelligence solution for university management. *10th WSEAS International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering (MMACTEE 2008)*, 318–324.

Bakar, M.S.A., Ta'a, A., Chit, S.C., & Soid, M.H.M. (2018). Data warehouse system for blended learning in institutions of higher education. *e-Academia Journal, 6*(2), 144-155.

Balkenende, M. (2018, 11 09). *4 Ways You Should be Using the Talend tMap Component.* Retrieved 07 07, 2019, from Talend: https://www.talend.com/blog/2018/11/09/4-ways-you-should-be-using-the-talend-tmap-component/

Beal, V. (n.d.). *Webopedia:ERP - enterprise resource planning.* Retrieved 02 01, 2020, from https://www.webopedia.com/TERM/E/ERP.html

Boukhalfa, K., Bellatreche, L., & Alimazighi, Z. (2009). HP&BJI: A Combined Selection of Data Partitioning and Join Indexes for Improving OLAP Performance. In S. Kozielski, & R. Wrembel (Eds.), *New Trends in Data Warehousing and Data Analysis* (pp. 210-233). US: Springer.

Brandão, A., Pereira, E., Esteves, M., Portela, F., Santos, M. F., Abelha, A., & Machado, J. (2016). A Benchmarking Analysis of Open-Source Business Intelligence Tools in Healthcare Environments. *Information, 7*(4), 57.

Budiarta, K., Wijaya, P. A., & Partha, C. G. I. (2017). Analysis and Design of Data Warehouse on Academic STMIK STIKOM Bali. *International Journal of Engineering and Emerging Technology, 2*(1), 35-39. Retrieved from https://simdos.unud.ac.id/uploads/file_penelitian_1_dir/87c5f9c3d84a999ab3b02cd98c1421be.pdf

Christopher, M. (2016). *Logistics & Supply Chain Management* (5th ed.). UK: Pearson.

cloudmoyo. (n.d.). *Sources of big data: Where does it come from?* Retrieved 01 10, 2020, from https://www.cloudmoyo.com/blog/data-architecture/what-is-big-data-and-where-it-comes-from/

Contributors, W. (2020, 26 01). *Eclipse (software).* Retrieved 01 30, 2020, from Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/w/index.php?title=Eclipse_(software)&oldid=937706251

Di Tria, F., Lefons, E., & Tangorra, F. (2015). Academic Data Warehouse Design Using a Hybrid Methodology. *Computer Science and Information Systems, 12*(1), 135-160.

Duan, Y., Cao, G., Ong, V. K., & Woolley, M. (2013). Big data in higher education: An action research on managing student engagement with business intelligence. Dubai: Second International Conference on Emerging Research Paradigm in Business and Social Science, Middlesex University.

Eckerson, W. W. (2007). *Predictive Analytics. Extending the Value of Your Data Warehousing Investment.* Retrieved from TDWI Best Practices Report: http://download.101com.com/pub/tdwi/Files/PA_Report_Q107_F.pdf

Fayyad, U., Piatetsky-shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 37-54.

Foundation, E. (n.d.). *Desktop IDEs.* Retrieved 06 01, 2019, from Eclipse IDE: https://www.eclipse.org/ide/

Guitart, I., & Conesa, J. (2015). Analytic Information Systems in the Context of Higher Education: Expectations, Reality and Trends. (pp. 294-300). Taipei: 2015 International Conference on Intelligent Networking and Collaborative Systems.

Guitart, I., & Conesa, J. (2016). Chapter 9 - Creating University Analytical Information Systems: A Grand Challenge for Information Systems Research. In S. Caballé, & R. Clarisó (Eds.), *Formative Assessment, Learning Data Analytics and Gamification* (pp. 167 - 186). Boston: Academic Press.

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers.

Huynh, T. N., & Schiefer, J. (2001). Prototyping Data Warehouse Systems. *Proceedings of the Third International Conference on Data Warehousing and Knowledge Discovery, 2114*, 195-207.

Indrajani, S., Yudha, P., Khotimah, N., & Vasthi, C. (2018). Building a Data Warehouse to Support Active Student Management: Analysis and Design. (pp. 460-465). Jakarta: 2018 International Conference on Information Management and Technology (ICIMTech).

Inmon, W. H. (2002). *Building the Data Warehouse* (3nd ed.). New York: John Wiley & Sons.

Jaspersoft ETL, T. O. (n.d.). *https://community.jaspersoft.com/project/jaspersoft-etl.* Retrieved from https://community.jaspersoft.com/project/jaspersoft-etl

JasperSoft. (n.d.). *Getting Started with Jaspersoft ETL. What is Jaspersoft ETL?* Retrieved 06 15, 2019, from https://community.jaspersoft.com/wiki/getting-started-jaspersoft-etl

JasperSoft. (n.d.). *JasperReports Library:Open Source Java Reporting Library.* Retrieved 12 12, 2019, from https://community.jaspersoft.com/project/jasperreports-library

JasperSoft. (n.d.). *JasperReports Server:Self-service Reporting and Analysis Server.* Retrieved 12 12, 2019, from https://community.jaspersoft.com/project/jasperreports-server

JasperSoft. (n.d.). *Jaspersoft Studio:The Eclipse-based report development tool for JasperReports and JasperReports Server.* Retrieved 12 02, 2019, from https://community.jaspersoft.com/project/jaspersoft-studio

JasperSoft. (n.d.). *The World's Most Popular Reporting Engine:Generate any report or visualization with lightning fast performance.* Retrieved 01 10, 2020, from https://www.jaspersoft.com/products/jasperreports-library

Kimball, R., & Caserta, J. (2004). *The Data Warehouse ETL Toolkit:Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data.* USA: John Wiley & Sons, Inc.

Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit* (2nd ed.). New York, USA: John Wiley & Sons, Inc.

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3nd ed.). Indianapolis: John Wiley & Sons, Inc.

Kravchenko, I. (2018, September 26). *Does Your Company Need a Business Intelligence Developer?* Retrieved February 01, 2019, from https://diceus.com/business-intelligence-developer/

Lapa, J., Bernardino, J., & Figueiredo, A. (2014). A comparative analysis of open source business intelligence platforms. *In Proceedings of the International Conference on Information Systems and Design of Communication (ISDOC '14)* (pp. 86-92). ACM.

Lebied, M. (2017, 06 29). *How Much ROI Does Business Intelligence Give You Anyways?* Retrieved 12 12, 2019, from https://www.datapine.com/blog/business-intelligence-roi/

Maia, A., Portela, F., & Santos, M. F. (2018). Web Intelligence in Higher Education: A Study on the Usage of Business Intelligence Techniques in Education. *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 176-181.

Malinowski, E., & Zimányi, E. (2009). *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications.* Springer-Verlag Berlin Heidelberg.

Muessig-Trapp, P., & Quathamer, D. (2011). Business Intelligence in HISinOne [Conference Presentation]. Hannover, Germany: EUNIS-BI Conference.

Muessig-Trapp, P., & Skladovs, V. (2013). The Data Portal of the German Federal Ministry of Education and Research (BMBF) as part of the German Open Government Approach. *EUNIS 2013 Congress Proceedings: 2013: ICT Role for Next Generation Universities, 1*.

Muntean, M., Bologa, A.-R., Bologa, R., & Florea, A. (2011). Business intelligence systems in support of university strategy. *Recent Researches in Educational Technologies*, 118-123.

Oracle. (n.d.). *JAR File Overview.* Retrieved 05 10, 2019, from Oracle: https://docs.oracle.com/javase/8/docs/technotes/guides/jar/jarGuide.html

ORACLE. (n.d.). *ORACLE*. Retrieved 05 01, 2018, from https://www.oracle.com/applications/erp/what-is-erp.html

Rahm, E., & Do, H. H. (2000). Data Cleaning: Problems and Current Approaches. *IEEE Data Eng. Bull., 23*(4), 3-13.

Salesforce. (2017, 08 16). *What is CRM?* Retrieved 11 3, 2018, from https://www.salesforce.com/blog/2013/01/what-is-crm-your-business-nerve-center.html

Santoso, L. W., & Yulia. (2017). Data Warehouse with Big Data Technology for Higher Education. *Procedia Computer Science, 124*, 93 - 99.

Savonnet, M., & Terrasse, M. N. (2001). Fragtique: Applying an OO Database Distribution Strategy to Data Warehouse. *Proceedings of the Third International Conference on Data Warehousing and Knowledge Discovery, 2114*, 339-348.

Sinaga, A. S., & Girsang, A. S. (2017). University Accreditation using Data Warehouse. *Journal of Physics: Conference Series, 801*(1), 012030.

Singhal, A. (2007). *Data Warehousing and Data Mining Techniques for Cyber Security.* Springer Science & Business Media.

Song, Y., Pramudianto, F., & Gehringer, E. F. (2016). A markup language for building a data warehouse for educational peer-assessment research. (pp. 1-5). In Frontiers in Education Conference (FIE).

SpringPeople. (2018, 12 13). *Data Warehousing Essentials: What Is ETL Tool? What Are Its Benefits?* Retrieved 12 12, 2019, from https://www.springpeople.com/blog/data-warehousing-essentials-what-is-etl-tool-what-are-its-benefits/

Sumathi, S., & Esakkirajan, S. (2007). *Fundamentals of Relational Database Management Systems.* Berlin, Heidelberg: Springer-Verlag.

Talend. (2019, 12 01). *Example Job implementing the different match models.* Retrieved from Talend Help Center: https://help.talend.com/reader/4sGi6uta35W98meBu6klVg/BLRK2T430L7DDJq3l47U2g

Talend. (2019, 12 02). *The differences between Unique match, First match and All matches.* Retrieved from Talend Help Center: https://help.talend.com/reader/4sGi6uta35W98meBu6klVg/~1Q8eE0O0Ru1ZGqty7lcJg

Talend. (2020, 01 07). *tMap Joins & Filtering.* Retrieved from Talend By Example: https://www.talendbyexample.com/talend-tmap-component-joins.html

Talend. (2020, 01 08). *Using the expression editor.* Retrieved from Talend Help Center: https://help.talend.com/reader/n2BYbtcWI4dtJfLCv3CrJw/dVvA1AafKB6~NBxrRtIM0A

Talend. (n.d.). *Calling a Talend Job from an external Java application.* Retrieved 05 02, 2019, from Talend Help Center: https://help.talend.com/reader/Lod~TDRaNw2L2VZKV9XgZw/V2Z1Ixh_f1XeLDp4fJW8cQ

Talend. (n.d.). *Configuring the process of matching data.* Retrieved 07 05, 2019, from Talend Help Center: https://help.talend.com/reader/8Byhmn0Igd39ieUOvmToDA/ZxnZNpN6WCu3gCO~ls~WEA

Talend. (n.d.). *Defining the match model for an explicit Join.* Retrieved 12 03, 2019, from Talend Help Center: https://help.talend.com/reader/St~M252yz9qZ1qf9VbrV4A/BfO5~U~SD4LGeoO6ZSw~zw

Talend. (n.d.). *Discovering Talend Studio.* Retrieved 05 02, 2019, from Talend Real-Time Open Source Integration Software: https://www.talend.com/resources/discovering-talend-studio/

Talend. (n.d.). *How to build Jobs.* Retrieved 05 03, 2019, from Talend Help Center: https://help.talend.com/reader/mhqCkTBnin7IXmJBUJoocQ/V5yP4nItw687WwHLVpxwFQ

Talend. (n.d.). *tFileInputDelimited.* Retrieved 07 02, 2019, from Talend Help Center: https://help.talend.com/reader/jomWd_GKqAmTZviwG_oxHQ/wZphtUXJPNp32p65z7XSgQ

Talend. (n.d.). *tFileOutputExcel.* Retrieved 07 03, 2019, from Talend Help Center: https://help.talend.com/reader/mjoDghHoMPI0yuyZ83a13Q/VVY~Y6vO2ZU76p78~VHl2Q

Talend. (n.d.). *tFilterColumns.* Retrieved 07 02, 2019, from Talend Help Center: https://help.talend.com/reader/hCrOzogIwKfuR3mPf~LydA/pybpRV~vjDqnFBzpULJGng

Talend. (n.d.). *tFilterRow.* Retrieved 07 02, 2019, from Talend Help Center: https://help.talend.com/reader/mjoDghHoMPI0yuyZ83a13Q/O_Vnk3B4SbKXtF5gTQl2Gg

Talend. (n.d.). *tReplicate.* Retrieved 06 02, 2019, from Talend Help Center: https://help.talend.com/reader/mjoDghHoMPI0yuyZ83a13Q/DqwuWmLFeq~jmKScHNDIYA

Talend. (n.d.). *tSortRow.* Retrieved 07 04, 2019, from Talend Help Center: https://help.talend.com/reader/mjoDghHoMPI0yuyZ83a13Q/HCkBMy3fiZ4zw5U2f9yxAg

Vaisman, A., & Zimányi, E. (2014). *Data Warehouse Systems : Design and Implementation.* Berlin: Springer.

Vargas, V., Syed, A., Mohammad, A., & Halgamuge, M. (2016). Pentaho and Jaspersoft : A Comparative Study of Business Intelligence Open Source Tools Processing Big Data to Evaluate Performances. *International Journal of Advanced Computer Science and Applications, 7*(10), 20-29.

Wikipedia, c. (2019, February 16). *Foreign key.* Retrieved March 14, 2019, from https://en.wikipedia.org/wiki/Foreign_key

Wikipedia, c. (2019, March 5). *Relational database.* Retrieved March 19, 2019, from https://en.wikipedia.org/wiki/Relational_database

Wikipedia, c. (2019, March 11). *SQL.* Retrieved March 14, 2019, from https://en.wikipedia.org/wiki/SQL

Wikipedia, C. (2020, 01 30). *JAR (file format).* Retrieved 01 30, 2020, from Wikipedia, The Free Encyclopedia.: https://en.wikipedia.org/wiki/JAR_(file_format)

Wikipedia, C. (2020, 01 06). *Windows Task Scheduler.* Retrieved 01 07, 2020, from Wikipedia: https://en.wikipedia.org/wiki/Windows_Task_Scheduler

Wise, L. (2012). *Using open source platforms for business intelligence: avoid pitfalls and maximize ROI.* Morgan Kauffman.

Yaqoob, I., Hashem, I.A.T., Gani, A., Mokhtar, S., Ahmed, E., Anuar, N.B., & Vasilakos, A.V. (2016). Big data: From beginning to future. *International Journal of Information Management, 36*(6), 1231-1247.