Volume 3, Issue 5, May 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Review Paper on scope of ETL in retail domain

Satkaur [1]
Research scholar, S.K.I.E.T.
Kurukshetra, Haryana, India

Anuj Mehta[2]
Asst.Prof.,S.K.I.E.T.
Kurukshetra, Haryana, India.

Abstract: The software processes that facilitate the original loading and the periodic refreshment of the data warehousecontents are commonly known as Extraction-Transformation-Loading (ETL) processes. The intention ofthis survey is to present the research work in the field of ETL technology in a structured way. To this end,we organize the coverage of the field as follows:
(a) first, we cover the conceptual and logical modeling ofETL processes, along with some design methods
(b) we visit each stage of the E-T-L triplet, and examineproblems that fall within each of these stages
(c) we discuss problems that pertain to the entirety of an ETLprocess, and
(d) we review some research prototypes of academic origin.

Keywords: extraction,transformation and loading, datawarehouses, datamart, online analytical processing, online transaction protocol

1. Introduction of ETL:-

You need to load your data warehouse regularly so that it can serve its purpose of facilitating business analysis. To do this, data from one or more operational systems needs to be extracted and copied into the warehouse. The process of extracting data from source systems and bringing it into the data warehouse is commonly called ETL, which stands for extraction, transformation, and loading. The acronym ETL is perhaps too simplistic, because it omits the transportation phase and implies that each of the other phases of the process is distinct. We refer to the entire process, including data loading, as ETL. You should understand that ETL refers to a broad process, and not three well-defined steps.
The methodology and tasks of ETL have been well known for many years, and are not necessarily unique to data warehouse environments: a wide variety of proprietary applications and database systems are the IT backbone of any enterprise. Data has to be
shared between applications or systems, trying to integrate them, giving at least two applications the same picture of the world. This data sharing was mostly addressed by mechanisms similar to what we now call ETL.

1. ETL Tools

Designing and maintaining the ETL process is often considered one of the most difficult and resource-intensive portions of a data
warehouse project. Many data warehousing projects use ETL tools to manage this process. Oracle Warehouse Builder (OWB), for example, provides ETL capabilities and takes advantage of inherent database abilities. Other data warehouse builders create their own ETL tools and processes, either inside or outside the database.
Besides the support of extraction, transformation, and loading, there are some other tasks that are important for a successful ETL implementation as part of the daily operations of the data warehouse and its support for further enhancements. Besides the support for designing a data warehouse and the data flow, these tasks are typically addressed by ETL tools such as OWB.
Oracle9i is not an ETL tool and does not provide a complete solution for ETL. However, Oracle9i does provide a rich set of capabilities that can be used by both ETL tools and customized ETL solutions. Oracle9i offers techniques for transporting data between Oracle databases, for transforming large volumes of data, and for quickly loading new data into a data warehouse.

2. ETL process

During extraction, the desired data is identified and extracted from many different sources, including database systems and applications. Very often, it is not possible to identify the specific subset of interest, therefore more data than necessary has to be extracted, so the identification of the relevant data will be done at a later point in time. Depending on the source system's capabilities (for example, operating system resources), some transformations may take place during this extraction process. The size of the extracted data varies from hundreds of kilobytes up to gigabytes, depending on the source system and the business situation. The same is true for the time delta between two (logically) identical extractions: the time span may vary between days/hours and minutes to near real-time. Web server log files for example can easily become hundreds of megabytes in a very short period of time.
After extracting data, it has to be physically transported to the target system or an intermediate system for further processing. Depending on the chosen way of transportation, some transformations can be done during this process, too. For example, a SQL statement which directly accesses a remote target through a gateway can concatenate two columns

© 2013, IJARCSSE All Rights Reserved

as part of the SELECT statement.The emphasis in many of the examples in this section is scalability. Many long-time users of Oracle are experts in programming complex data transformation logic using PL/SQL. These chapters suggest alternatives for many such data manipulation operations, with a particular emphasis on implementations that take advantage of Oracle's new SQL functionality, especially for ETL and the parallel query infrastructure.

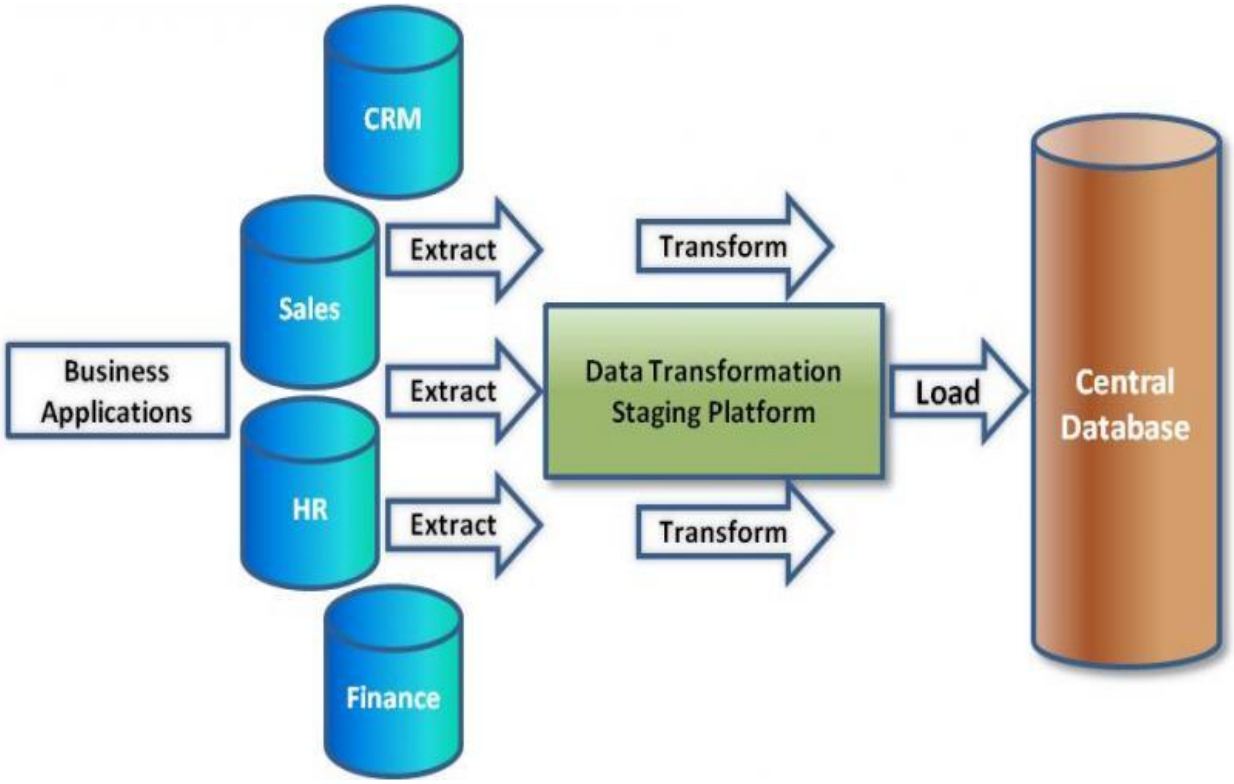

Fig. 1.1. ETL process

2. Need for ETL implementation

The successive loads and transformations must be scheduled and processed in a specific order. Depending on the success or failure of the operation or parts of it, the result must be tracked and subsequent, alternative processes might be started. The control of the progress as well as the definition of a business workflow of the operations is typically addressed by ETL tools such as OWB.

I. Extraction

Extraction is a process to extract the data from an unorganised form to an organised form and place the data from databases to datawarehouse

Extraction method

1. Logical Extraction Methods

3. Physical Extraction Methods

There are two kinds of logical extraction:

A. Full Extraction

B. Incremental Extraction

A. **Full Extraction:-** The data is extracted completely from the source system. Since this extraction reflects all the data currently available on the source system, there's no need to keep track of changes to the data source since the last successful extraction. The source data will be provided as-is and no additional logical information (for example, timestamps) is necessary on the source site. An example for a full extraction may be an export file of a distinct table or a remote SQL statement scanning the complete source table.

B. **Incremental Extraction:-** At a specific point in time, only the data that has changed since a well-defined event back in history will be extracted. This event may be the last time of extraction or a more complex business event like the last booking day of a fiscal period. To identify this delta change there must be a possibility to identify all the changed information since this specific time event.

4. Physical Extraction Methods

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

© 2013, IJARCSSE All Rights Reserved

C. **Online Extraction:-**
The data is extracted directly from the source system itself. The extraction process can connect directly to the source system to access the source tables themselves or to an intermediate system that stores the data in a preconfigured manner (for example, snapshot logs or change tables). Note that the intermediate system is not necessarily physically different from the source system.

D. **Offline Extraction:-**
The data is not extracted directly from the source system but is staged explicitly outside the original source system. The data already has an existing structure (for example, redo logs, archive logs or transportable tablespaces) or was created by an extraction routine.

5. Several techniques to implement the source system:-

A. Timestamps

The tables in some operational systems have timestamp columns. The timestamp specifies the time and date that a given row was last modified. If the tables in an operational system have columns containing timestamps, then the latest data can easily be identified using the timestamp columns. For example, the following query might be useful for extracting today's data from an orders table:

*SELECT * FROM orders WHERE TRUNC(CAST(order_date AS date),'dd') = TO_DATE(SYSDATE,'dd-mon-yyyy');*

If the timestamp information is not available in an operational source system, you will not always be able to modify the system to include timestamps. Such modification would require, first, modifying the operational system's tables to include a new timestamp column and then creating a trigger to update the timestamp column following every operation that modifies a given row.

B. Partitioning

Some source systems might use Oracle range partitioning, such that the source tables are partitioned along a date key, which allows for easy identification of new data. For example, if you are extracting from an orders table, and the orders table is partitioned by week, then it is easy to identify the current week's data.

C. Triggers

Triggers can be created in operational systems to keep track of recently updated records. They can then be used in conjunction with timestamp columns to identify the exact time and date when a given row was last modified. You do this by creating a trigger on each source table that requires change data capture. Following each DML statement that is executed on the source table, this trigger updates the timestamp column with the current time. Thus, the timestamp column provides the exact time and date when a given row was last modified.

6 . Extraction of data in two ways:

A. **Extraction Using Data Files:-** When the source system is an Oracle database, several alternatives are available for extracting data into files:
Extracting into Flat Files Using SQL*Plus

a) Extracting into Flat Files Using OCI or Pro*C Programs
b) Exporting into Oracle Export Files Using Oracle's Export Utility
c) Extracting into Flat Files Using SQL*Plus

The most basic technique for extracting data is to execute a SQL query in SQL*Plus and direct the output of the query to a file. For example, to extract a flat file, country_city.log, with the pipe sign as delimiter between column values, containing a list of the cities in the US in the tables countries and customers, the following SQL script could be run.

Extracting into Flat Files Using OCI or Pro*C Programs:- OCI programs (or other programs using Oracle call interfaces, such as Pro*C programs), can also be used to extract data. These techniques typically provide improved performance over the SQL*Plus approach, although they also require additional programming.

Like the SQL*Plus approach, an OCI program can extract the results of any SQL query. Furthermore, the parallelization techniques described for the SQL*Plus approach can be readily applied to OCI programs as well.

7. Loading and Transformation in Data Warehouses

Data transformations are often the most complex and, in terms of processing time, the most costly part of the ETL process. They can range from simple data conversions to extremely complex data scrubbing techniques. Many, if not all, data transformations can occur within an Oracle9i database, although transformations are often implemented outside of the database (for example, on flat files) as well.

Transformation Flow

From an architectural perspective, you can transform your data in two ways:-

(1) Pipelined Data Transformation
(2) Multistage Data Transformation

The data transformation logic for most data warehouses consists of multiple steps. For example, in transforming new records to be inserted into a sales table, there may be separate logical transformation steps to validate each dimension key.

Pipelined Data Transformation:-

© 2013, IJARCSSE All Rights Reserved

With the introduction of Oracle9i, Oracle's database capabilities have been significantly enhanced to address specifically some of the tasks in ETL environments. The ETL process flow can be changed dramatically and the database becomes an integral part of the ETL solution.

The new functionality renders some of the former necessary process steps obsolete whilst some others can be remodeled to enhance the data flow and the data transformation to become more scalable and non-interruptive. The task shifts from serial transform-then-load process (with most of the tasks done outside the database) or load-then-transform process, to an enhanced transform-while-loading.

Oracle9i offers a wide variety of new capabilities to address all the issues and tasks relevant in an ETL scenario. It is important to understand that the database offers toolkit functionality rather than trying to address a one-size-fits-all solution. The underlying database has to enable the most appropriate ETL process flow for a specific customer need, and not dictate or constrain it from a technical perspective. Figure 13-2 illustrates the new functionality, which is discussed throughout later sections.

Loading Mechanisms:-

You can use the following mechanisms for loading a warehouse:

SQL*Loader

External Tables

OCI and Direct-Path APIs

Export/Import

SQL*Loader

Before any data transformations can occur within the database, the raw data must become accessible for the database. One approach is to load it into the database. Chapter 12, "Transportation in Data Warehouses", discusses several techniques for transporting data to an Oracle data warehouse. Perhaps the most common technique for transporting data is by way of flat files.

SQL*Loader is used to move data from flat files into an Oracle data warehouse. During this data load, SQL*Loader can also be used to implement basic data transformations. When using direct-path SQL*Loader, basic data manipulation, such as datatype conversion and simple NULL handling, can be automatically resolved during the data load. Most data warehouses use direct-path loading for performance reasons.

8. Conclusion and Future Scope

Security to the data is one of the major challenges and area of concern in today‟s world. Current approaches for the modeling of ETL do not address the security issues in the ETL modeling. This research work proves and shows the improvement in data extraction speed by using flat files with security measures. The extraction time for smaller number of records such as 100 records does not show much difference in the extraction time for database file or flat file. The extraction speed for Database file consisting of 100 records takes 1ms, where as the same 100 records can be extracted in.99ms using flat files which is not much difference in the extraction speed When number of records is increased above 2000 records the difference in extraction time using flat file makes huge difference and makes sense in improving of the extraction process of ETLrather than using extraction by database file. The extraction time for 2000 records using flat files takes 16.37 mswhere as the extraction timefor the same takes 25ms using database file. Thus the improvement in the extraction process with respect to space and time domain is achieved successfully during this research work. Current approaches for the conceptual modeling of ETL do not address the security aspects in the conceptual modeling phase. The building process constitute of ETL i.e. Extraction, Transformation and Loading. This research proposes security and improvement in the first phase of extraction process of ETL. Protection to the file to be extracted from the source is implemented by applying pass code to the file. Once the file is extracted at the destination the same pass code is applied to open the file at the target machine. The user needs to remember the password which he applied before extraction, for him to open the file after extraction by applying the same pass code at the target machine [56]. Improvement during extraction process is achieved by converting the data base table into a flat file and extract. The flat file needs less storage space on the disk compared to data base table. The extraction using flat file is much faster than extraction of the database table. The extraction mechanism used is full extraction if the database is extracted for the first time and change data capture method, if the database is the modified once. We have observed that the data extraction using direct database file, requires more time compared to that of the flat file database. This research method of extraction does provide security to the flat file, thus the hacker cannot make use of the content of the file during the extraction process.

This work can be extended further for the later building process of the data warehouse for transformation and loading process of the ETL. Data for the largest merge extraction is not shown because it need high specification computer to able to sort or index the source data set with one million records.

References

[1] Ponas Vassiliadis, “A survey of Extract-transform-load technology”, International Journel of datawarehousing & mining, 5(3), 1-27, July September 2009 1.

[2] Vishal Gaur 1, Dr. S.S. Sarangdevot 2, Govind Singh Tanwar 3, Anand Sharma 4, “ Improve performance of Extract, transform and Load ” , Vishal Gaur et.al/ International Journal on Computer Science and Engineering Vol. 02, No.-03 ,2010 786-789.

© 2013, IJARCSSE All Rights Reserved

[3] Shaker H. Ali EI- Sappagh*a, Abdeltawab H. Ahmed Hendawi*b, Ali Hamed EI Bastawissy*b, " A Proposed Model For datawarehouse ETL Process", Journal of King Saud University-Computer and information Sciences (2011)23, 91-104.

[4] Radha Krishna author1 and Sreekanth Author 2, "An Object Oriented Modelling and Implementation of web based ETL process" , International Journel of computer science and network security , Vol. 10 No. 2, February 2010.

[5] Qin Hanlin, Jin Xianzhen, Zhang Xianrong, "Research on Extract,Transform and load in Land and Resources Star Schema Data Warehouses", Computational Intelligence and Design (ISC10), 2012 fifth International Symposium on (Volume 1), 28-29 Oct.2012,Pages 120-123.

[6] Nestor Rodruez, Kent Lawson, Eddie Molina, "Data Warehouses tool Evaluation. ETL Focused", Univesity of Texas-Pan American 1201 W. University Drive, Edinburg, TX (956) 665-UTPA.

[7] Fundulaki [1], Alex Averbuch [2], Eva Daskalaki [3], " Overview and analysis of Existing benchmark framework, LDBC Cooperative Project FP7-317548.

[8] Thomas Van Raalte, " Introduction to Oracle Retail data model Implementation and Operation Guide" , Release 11.3.2 E20363-03, January 2013.

[9] Alkis Simitsis 1, Panos Vasiliadis 2, "A Methodology for the Conceptual Modelling of ETL process", National Technical University of Athens, Dept. of Electrical and Computer Engineering , Computer Science Division , Iroon Polytechniou. 9, 15773, Athens, Greece asimi@ dbnet.ece.ntua.gr, University of Ioannina, Dept. of Computer Science, 45110,Ioannia, Greece. Pvassil@ cs.uoi.gu.

[10] Thomas Jorg[1], Stefan De Bloch, "Towards generating ETL process for incremental Loading" University of Kaiserslanterm, 67.653 Kaiserslam term, Germany.

[11] http:// dbs.uni-leipzig.de. Erhard Rahm*, Hang Hai Do, University of Leipzig, Germany, "Data Cleaning :Problem and Current Approaches".

[12] ALKIS SIMITSIS, " Modelling and Optimization of Extraction-Transformation, and loading (ETL) processes in Datawarehouses Environment" Athens, Ocober, 2004, Ph.d Electrical and computer Engineering N.T.U.A.

[13] Qin Hanlin; Jin Xianzhen; Zhang Xianrong, " Research on Extract, Tranform and Load (ETL) in land and Resources Star Schema data Warehouses" Computational Intelligence and design (ISCID), 2012 fifth International IEEE Symposium on (Volume :1), 28-29 oct, 2012, pages(120-123), ISBN-978-1-4673-2646-9.

[14] Huong Morris*{ Hui Liao, Sriram Padmanabhen, Sriram Srinivasan},{Phey Lau, Jing sham, Ryan Wisnesky}", "Bringing Business Objects into Extract-transform, Load (ETL) Technology" , IEEE International Conference on e-business engineering.

[15] M. Bouzeghoub, F. Fabret, M. Matulovic. "Modeling Data Warehouse Refreshment Process as a Workflow Application" In Proc. Intl. Workshop on Design and Managementof Data Warehouses (DMDW'99), Heidelberg, Germany, (1999).

[16] V. Borkar, K. Deshmuk, S. Sarawagi. "Automatically Extracting Structure from Free Text Addresses. Bulletin of the Technical Committee on Data Engineering", 23(4), (2000).

[17] G. Booch, I. Jacobson, J. Rumbaugh. "The Unified Modeling Language User Guide". Addison-Wesley Pub Co; ISBN: 0201571684; 1st edition, October 1998.

© 2013, IJARCSSE All Rights Reserved