ETL Processing

Contents

Abstract

As the data warehouse is a dwelling IT device, sources and targets would possibly change. Those changes ought to be maintained and tracked through the lifespan of the machine without overwriting or deleting the vintage information. We need to load data warehouse often so that it is able to serve its motive of facilitating enterprise analysis and hold updated. The procedure of extracting facts from source structures and bringing it into the facts warehouse is commonly known as ETL, which stands for extraction, transformation, and loading. The intention of this survey is to provide the research work inside the discipline of ETL technology in a structured way.

Introduction

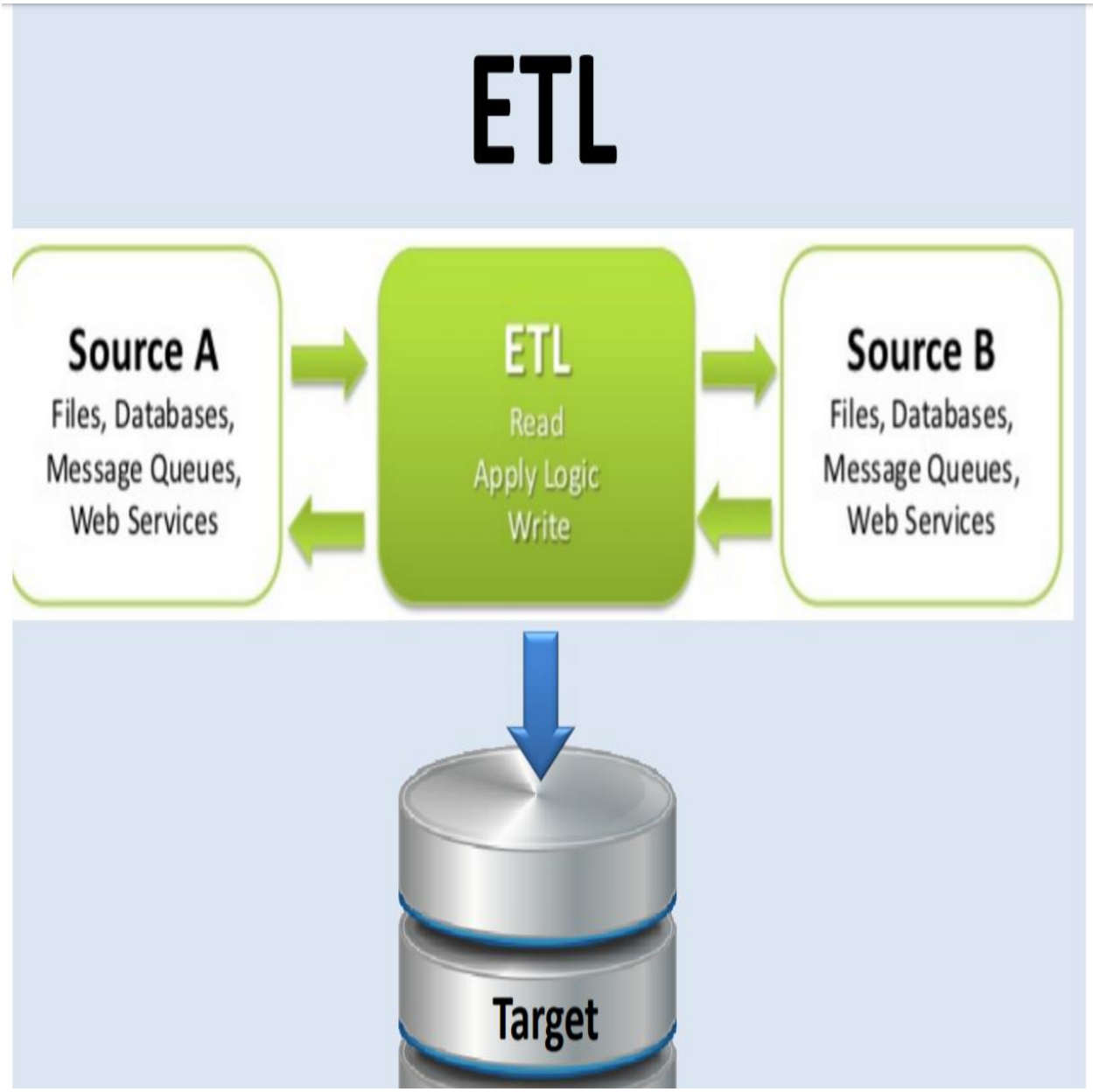
ETL is answerable for the extraction of data, their cleaning, conforming and loading into the target. ETL is a Critical layer in DW setting. It is widely identified that building ETL processes is pricey concerning time, cash and effort. In this, first of all we overview commercial ETL tools and prototypes coming from educational world. After that we evaluation designing works in ETL subject and modelling ETL renovation issues. We review works in reference to optimization and incremental ETL, then in the end demanding situations and research opportunities around ETL strategies.

In computing, extract, transform and load (ETL) refers to a process in database usage and especially in data warehousing that involves:

• Extracting data from outside sources

• Transforming it to fit operational needs (which can include quality levels)

• Loading it into the end target (database, more specifically, operational data store, data mart or data warehouse)



Extract

The first part of an ETL process involves extracting the data from the source systems. In many cases this is the most challenging aspect of ETL, as extracting data correctly will set the stage for how subsequent processes will go. ETL Architecture Pattern Most data warehousing projects consolidate data from different source systems. Each separate system may also use a different data organization/format. Common data source formats are relational databases and flat files, but may include non-relational database structures such as Information Management System (IMS) or other data structures such as Virtual Storage Access Method (VSAM) or Indexed Sequential Access Method (ISAM), or even fetching from outside sources such as through web spidering or screen-scraping. The streaming of the extracted data source and load on-the-fly to the destination database is another way of performing ETL when no

intermediate data storage is required. In general, the goal of the extraction phase is to convert the data into a single format which is appropriate for transformation processing. An intrinsic part of the extraction involves the parsing of extracted data, resulting in a check if the data meets an expected pattern or structure. If not, the data may be rejected entirely or in part.

Transform

The transform level applies to a series of guidelines or features to the extracted information from the supply to derive the records for loading into the cease goal. Some information sources will require very little or even no manipulation of records. In other cases, one or extra of the subsequent transformation types can be required to fulfill the commercial enterprise and technical needs of the target database:

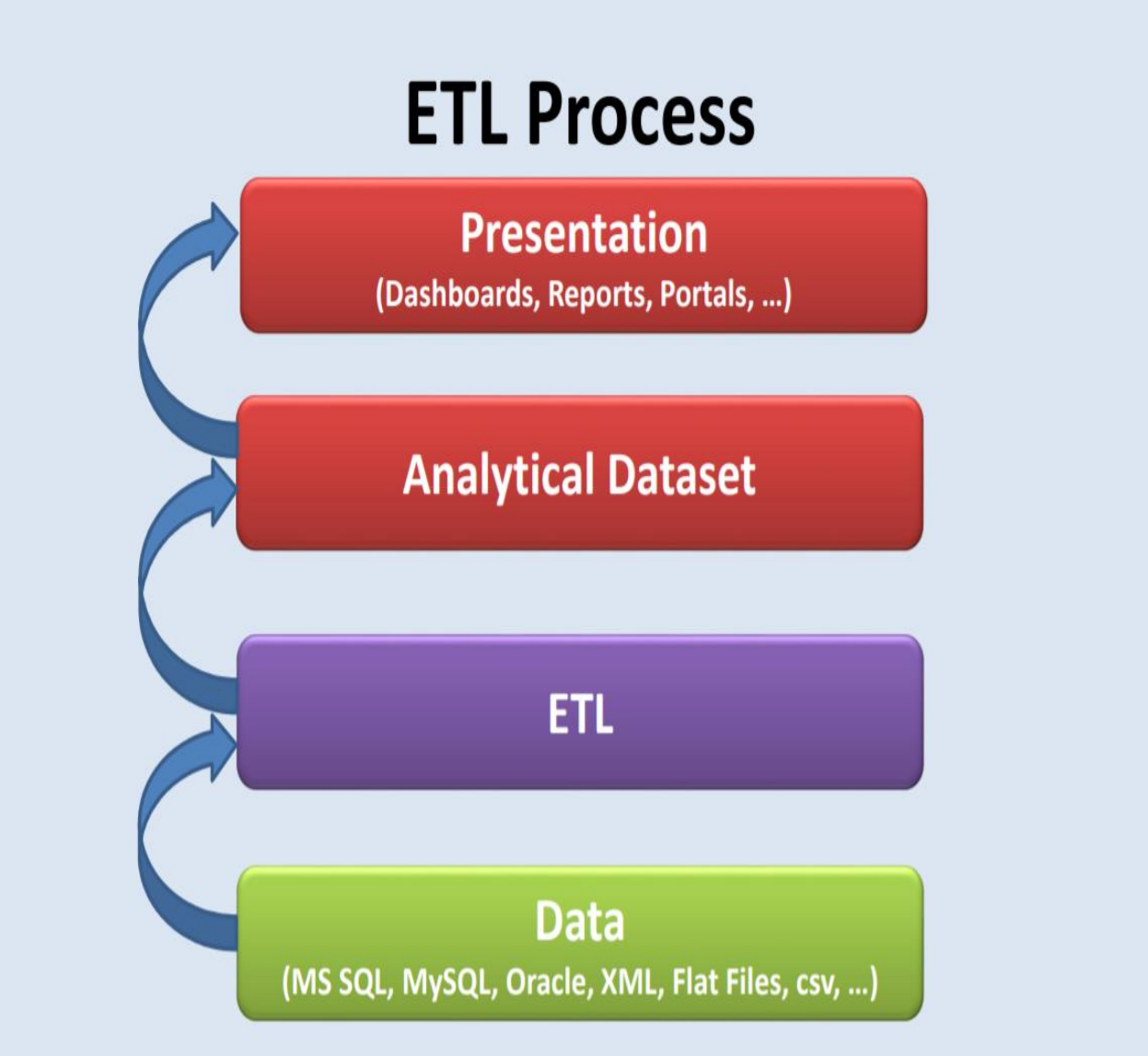
- Selecting handiest sure columns to load (or selecting null columns now not to load). For example, if the supply information has 3 columns (also called attributes), as an instance roll_no, age, and salary, then the extraction can also take most effective roll_no and salary. Similarly, the extraction mechanism may additionally ignore all those facts in which salary isn't always present (salary = null).
- Translating coded values (e.G., if the source machine stores 1 for male and a couple of for female, however the warehouse stores M for male and F for female)
- Encoding free-form values (e.G., mapping "Male" to "1") Deriving a new calculated value (e.G., sale_amount = qty * unit_price)
- Sorting
- Joining facts from multiple sources (e.G., lookup, merge) and deduplicating the facts
- Aggregation (for example, rollup — summarizing more than one rows of records — total sales for every store, and for every region, etc.)
- Generating surrogate-key values
- Transposing or pivoting (turning multiple columns into a couple of rows or vice versa)
- Splitting a column into more than one columns (e.G., setting a comma-separated list unique as a string in a single column as man or woman values in one of a kind columns)
- Disaggregation of repeating columns right into a separate detail table (e.G., moving a chain of addresses in one record into single addresses in a hard and fast of records in a linked address table)
- Lookup and validate the relevant records from tables or referential files for slowly changing dimensions.
- Applying any form of easy or complex records validation. If validation fails, it could result in a full, partial or no rejection of the information, and hence none, a few or all the statistics is handed over to the next step, relying at the rule design and exception handling. Many of the above transformations can also result in exceptions, as an instance, while a code translation parses an unknown code in the extracted facts.

Load

The load phase loads the data into the end target, usually the data warehouse (DW). Depending on the requirements of the organization, this process varies widely. Some data warehouses may overwrite existing information with cumulative information, frequently updating extract data is done on daily, weekly or monthly basis. Other DW (or even other parts of the same DW) may add new data in a historicized form, for example, hourly. To understand this, consider a DW that is required to maintain sales records of the last year. Then, the DW will overwrite any data that is older than a year with newer data. However, the entry of data for any one year window will be made in a historicized manner. The timing and scope to replace or append are strategic design choices dependent on the time available and the business needs. More complex systems can maintain a history and audit trail of all changes to the data loaded in the DW. As the load phase interacts with a database, the constraints defined in the database schema — as well as in triggers activated upon data load — apply (for example, uniqueness, referential integrity, mandatory fields), which also contribute to the overall data quality performance of the ETL process.

- For example, a financial institution might have information on a customer in several departments and each department might have that customer's information listed in a different way. The membership department might list the customer by name, whereas the accounting department might list the customer by number. ETL can bundle all this data and consolidate it into a uniform presentation, such as for storing in a database or data warehouse. • Another way that companies use ETL is to move information to another application permanently. For instance, the new application might use another database vendor and most likely a very different database schema. ETL can be used to transform the data into a format suitable for the new application to use.

- An example of this would be an Expense and Cost Recovery System (ECRS) such as used by accountancies, consultancies and lawyers. The data usually ends up in the time and billing system, although some businesses may also utilize the raw data for employee productivity reports to Human Resources (personnel dept.) or equipment usage reports to Facilities Management.



Real-life ETL cycle The typical real-life ETL cycle consists of the following execution steps:

1. Cycle initiation
2. Build reference data
3. Extract (from sources)
4. Validate
5. Transform (clean, apply business rules, check for data integrity, create aggregates or disaggregates)
6. Stage (load into staging tables, if used)

7. Audit reports (for example, on compliance with business rules. Also, in case of failure, helps to diagnose/repair) 8. Publish (to target tables)
8. Archive
9. Clean up

Challenges

ETL approaches can involve large complexity, and big operational problems can occur with improperly designed ETL systems.

The variety of statistics values or statistics quality in an operational device can also exceed the expectations of designers at the time validation and transformation rules are specified. Data profiling of a source during facts analysis can identify the records situations that will want to be managed by using transform regulations specifications. This will lead to an modification of validation policies explicitly and implicitly implemented inside the ETL technique.

Data warehouses are commonly assembled from a selection of data resources with exceptional formats and purposes. As such, ETL is a key technique to carry all the information collectively in a standard, homogeneous environment.

Design analysts need to set up the scalability of an ETL device across the lifetime of its usage. This includes expertise the volumes of facts that will must be processed within service degree agreements. The time available to extract from supply systems may change, which may additionally mean the identical quantity of statistics might also ought to be processed in less time. Some ETL systems should scale to procedure terabytes of information to update records warehouses with tens of terabytes of records. Increasing volumes of facts may also require designs that could scale from day by day batch to multiple-day microbatch to integration with message queues or real-time change-facts capture for non-stop transformation and update.

Performance

ETL companies benchmark their report-structures at multiple TB (terabytes) in step with hour (or ~1 GB per second) using effective servers with multiple CPUs, multiple hard drives, more than one gigabit-network connections, and lots of memory. The fastest ETL report is currently held by using Syncsort,[1] Vertica and HP at 5.4TB in underneath an hour which is greater than twice as rapid as the earlier report held by means of Microsoft and Unisys.

In actual life, the slowest part of an ETL system usually takes place in the database load phase. Databases may additionally perform slowly because they have to attend to concurrency, integrity maintenance, and indices. Thus, for better performance, it can make experience to employ:

- Direct Path Extract approach or bulk unload each time is possible (rather than querying the database) to reduce the load on supply gadget while getting excessive speed extract

- most of the transformation processing outdoor of the database
- bulk load operations every time possible. Still, even using bulk operations, database access is commonly the bottleneck inside the ETL manner. Some not unusual strategies used to increase performance are:
- Partition tables (and indices). Try to keep partitions similar in size (watch for null values that can skew the partitioning).
- Do all validation inside the ETL layer before the load. Disable integrity checking (disable constraint ...) in the goal database tables in the course of the load.
- Disable triggers (disable trigger ...) within the target database tables all through the load. Simulate their effect as a separate step.
- Generate IDs within the ETL layer (not inside the database).
- Drop the indices (on a table or partition) before the load - and recreate them after the load (SQL: drop index ...; create index ...).
- Use parallel bulk load whilst possible — works nicely when the table is partitioned or there aren't any indices. Note: try to do parallel masses into the equal desk (partition) generally reasons locks — if not on the information rows, then on indices.
- If a demand exists to do insertions, updates, or deletions, find out which rows must be processed in which way within the ETL layer, and then manner these 3 operations in the database separately. You often can do bulk oad for inserts, however updates and deletes normally go through an API (the use of SQL).

Whether to do sure operations in the database or out of doors can also contain a trade-off. For example, removing duplicates using wonderful can be gradual inside the database; thus, it makes feel to do it outside. On the other side, if the usage of wonderful will extensively (x100) decrease the quantity of rows to be extracted, then it makes sense todo away with duplications as early as possible inside the database earlier than unloading records.

A common source of troubles in ETL is a big wide variety of dependencies amongst ETL jobs. For example, job "B" cannot start while job "A" is not finished. You can normally achieve better overall performance by way of visualizing all processes on a graph, and attempting to lessen the graph making most use of parallelism, and making "chains" of consecutive processing as short as possible. Again, partitioning of huge tables and in their indices can surely help.

Another not unusual issue happens while the facts is unfold between several databases, and processing is executed in those databases sequentially. Sometimes database replication can be involved as a way of copying statistics between databases - and this could appreciably sluggish down the whole technique. The commonplace solution is to reduce the processing graph to most effective 3 layers:

- Sources
- Central ETL layer

- Targets

This permits processing to take maximum benefit of parallel processing. For example, if you need to load information into two databases, you could run the hundreds in parallel (instead of loading into 1st - after which replicating into the 2nd). Of course, every so often processing ought to take region sequentially. For example, you typically need to get dimensional (reference) data earlier than you can get and validate the rows for main "fact" tables.

Parallel processing

A recent development in ETL software program is the implementation of parallel processing. This has enabled a range of techniques to improve overall overall performance of ETL strategies when dealing with huge volumes of records.

ETL applications implement three main types of parallelism:

- Data: By splitting a unmarried sequential file into smaller facts documents to offer parallel access.
- Pipeline: Allowing the simultaneous going for walks of several additives on the equal information stream. For example: searching up a value on file 1 at the equal time as adding two fields on report 2.
- Component: The simultaneous going for walks of a couple of tactics on different information streams in the equal job, for example, sorting one input document at the same time as disposing of duplicates on another document.

All 3 types of parallelism commonly operate combined in a unmarried job. An additional issue comes with making sure that the information being uploaded is extraordinarily consistent. Because a couple of supply databases may have different update cycles (some can be up to date every few minutes, at the same time as others may additionally take days or weeks), an ETL machine can be required to maintain returned certain records until all sources are synchronized. Likewise, wherein a warehouse may additionally should be reconciled to the contents in a source machine or with the popular ledger, organizing synchronization and reconciliation points turns into necessary.

Data warehousing procedures normally subdivide a big ETL method into smaller pieces jogging sequentially or in parallel. To preserve tune of records flows, it makes feel to tag every information row with "row_id", and tag each piece of the technique with "run_id". In case of a failure, having these IDs will help to roll back and rerun the failed piece.

Best practice also calls for "checkpoints", which are states when certain phases of the process are completed. Once at a checkpoint, it is a superb idea to write everything to disk, clean out some temporary files, log the state, and so on.

Virtual ETL

As of 2010 statistics virtualization had started to improve ETL processing. The utility of facts virtualization to ETL allowed fixing the most not unusual ETL duties of data migration and application integration for more than one dispersed data sources. So-called Virtual ETL operates with the abstracted

illustration of the items or entities amassed from the kind of relational, semi-based and unstructured facts sources. ETL gear can leverage object-oriented modeling and work with entities' representations persistently saved in a centrally placed hub-and-spoke architecture. Such a collection that includes representations of the entities or gadgets collected from the information sources for ETL processing is known as a metadata repository and it could live in memory or be made continual. By the usage of a persistent metadata repository, ETL gear can transition from one-time projects to continual middleware, performing information harmonization and statistics profiling continually and in near-real time.

ETL and DataWarehouse

Enterprises as agencies spend money on DW projects in order to enhance their interest and for measuring their performance. It pursuits to enhance decision technique by using supplying particular access to several resources. In this we've got two kinds. The two

famous kinds are databases and flat files. Finally let notice that sources are self-sustaining or semi self-sustaining. It is the integration layer in DW environment . ETL gear pull facts from several assets (databases tables, flat files, ERP, internet, and so on), apply complex transformation to them. ETL is a vital component in DW environment. Indeed, it's miles

widely identified that building ETL processes, at some point of DW project, are high priced regarding time and money.A. Data Warehouse layersSources: They embody all kinds of statistics assets. They are records provider. The famous kinds are databases and flat files. Finally let note that sources are independent or semi self sustaining.

ETL: It is the mixing layer in DW environment. ETL gear pull information from numerous resources apply complex transformation to them. Finally inside the give up, facts are loaded into the target that is information warehouse save in DW environment.

Data Warehouse: is a crucial repository to save information produced by using ETL layer. DW is a DB includes fact tables and size tables. Together these tables are blended in a particular schema that can be superstar schema or snowflake schema.

Reporting and Analysis: Collected information are served to quit-customers in several formats. For instance records is formatted into reports, histograms.

Extraction: The hassle facts from a set of assets which can be local or distant. Logically, facts sources come from operational applications, but there's an option to use outside statistics assets for enrichment. External records supply means records coming from external entities. Thus at some point of extraction step, ETL attempts to get right of entry to available sources, pull out the relevant data, and reformat such information in a distinct format.

Transformation: This step is the most arduous one wherein ETL provides value. This step is associated with words: clean and conform. In one hand, cleansing information aims to restore erroneous statistics and to deliver clean facts for end users (decisions makers). Dealing with missing records, rejecting bad information are examples of data cleansing operations. In other hand, conforming records objectives to

make statistics correct, in compatibility with other master records. Checking commercial enterprise rules, checking keys and lookup of referential statistics are instance of conforming operations.

Loading: This step conversely to previous step, has the trouble of storing facts to a set of targets. During this step, ETL loads records into goals that are reality tables and size in DW context.

Commercial ETL Tools We have two sorts of ETL tools. On the only hand, there's subfamily of payable ETL Data Stage and Informatica. On the opposite hand, the second subfamily of business ETL comes with no charge.

Informatica: Informatica is extensively used ETL device for extracting the source records and loading it into the target after applying the Required Transformation .ETL builders map the extracted records from source structures and cargo it to target structures after making use of the specified transformations .

Data Stage: Its basic detail for information manipulation is referred to as "stage." Thus, for this device an ETL system is a combination of "ranges." Thus we talk about transformation degrees and degrees for extracting and loading information called connectors since release which might be interconnected via links.

SSIS:S SIS imposes ranges of duties combination. The first level is referred to as "Flow Control" and the second level controlled by using the first, is referred to as "Data flow." Indeed, the first stage is dedicated to put together the execution environment (deletion, control, shifting files, etc....) and materials obligations for this purpose. The 2nd stage (facts flow) that's a particular challenge of the primary level plays classical ETL mission. The Data-Flow task offers numerous responsibilities for statistics extraction, transformation and loading.

SIRIUS:It develops an technique metadata oriented that permits the modelling and execution of ETL processes. It is based on SIRIUS Meta model issue that represents metadata describing the essential operators or functions for enforcing ETL processes. In other words, SIRIUS gives capabilities to explain the assets, goals description and the mapping among those parts.

ARKTOS: A RKTOS is every other framework that makes a specialty of the modelling and execution of ETL processes. Indeed, ARKTOS affords primitives to capture ETL obligations often used. More exactly, to explain a certain ETL system, this framework offers three ways that are GUI and languages XADL (XML variant) and SADL (SQL like language).

DWPP: DWPP is a set of modules designed to solve the typical troubles that occur in any ETL project. DWPP isn't a device but it's miles a platform. Exactly, it's miles C functions library shared underneath UNIX operating system for the implementation of ETL processes. Consequently, DWPP affords a fixed of useful features for information manipulation.

MODELLING AND DESIGN OF ETL ETL are areas with high added value labelled costly and risky. In addition, software engineering requires that any project is doomed to switch to maintenance mode. For these reasons, it is essential to overcome the ETL modelling phase with elegance in order to produce simple models and understandable. This method is spread over four steps: 1. Identification of sources 2. Distinction between candidates' sources and active sources. 3. Attributes mapping. 4. Annotation of diagram (conceptual model) with execution constraints. A. Meta-data models based on ETL Design the

designer needs to: 1. Analyse the structure and sources. 2. Describe mapping rules between sources and targets. The based on meta-model, provides a graphical notation to meet this necessitate.

ETL PROCESS MAINTENANCE: When changes happen, analyzing the impact of change is mandatory to avoid errors and mitigate the risk of breaking existent treatments. As a consequence, without a helpful tool and an effective approach for change management, the cost of maintenance task will be high. Particularly for ETL processes, previously judged expensive and costly [14], [15]. Using ETL terminology, above previous research efforts focus on the target unlike the proposal of which focuses onchanges in the sources. In these proposal dealing with change management in ETL are interesting and offer a solution to detect changes impact on ETL processes. However, change incorporation is not addressed.

Data warehousing and Application

Information assets are immensely valuable to any enterprise, and because of this, these assets must be properly stored and readily accessible when they are needed. However, the availability of too much data makes the extraction of the most important information difficult, if not impossible. View results from any Google search, and you'll see that the data = information equation is not always correct—that is, too much data is simply too much. Data warehousing is a phenomenon that grew from the huge amount of electronic data stored in recent years and from the urgent need to use that data to accomplish goals that go beyond the routine tasks linked to daily processing. In a typical scenario, a large corporation has many branches, and senior managers need to quantify and evaluate how each branch contributes to the global business performance. The corporate database stores detailed data on the tasks performed by branches. To meet the managers' needs, tailor-made queries can be issued to retrieve the required data. In order for this process to work, database administrators must first formulate the desired query (typically an aggregate SQL query) after closely studying database catalogs. Then the query is processed. This can take a few hours because of the huge amount of data, the query complexity, and the concurrent effects of other regular workload queries on data. Finally, a report is generated and passed to senior managers in the form of a spreadsheet. Many years ago, database designers realized that such an approach is hardly feasible, because it is very demanding in terms of time and resources, and it does not always achieve the desired results. Moreover, a mix of analytical queries with transactional routine queries inevitably slows down the system, and this does not meet the needs of users of either type of query. Today's advanced data warehousing processes separate online analytical processing (OLAP) from online transactional processing (OLTP) by creating a new information repository that integrates basic data from various sources, properly arranges data formats, and then makes data available for analysis and evaluation aimed at planning and decision-making processes (Lechtenbörger, 2001).

Let's review some fields of application for which data warehouse technologies are successfully used: •

- Trade Sales and claims analyses, shipment and inventory control, customer care and public relations
- Craftsmanship Production cost control, supplier and order support
- Financial services Risk analysis and credit cards, fraud detection

- Transport industry Vehicle management
- Telecommunication services Call flow analysis and customer profile analysis

Health care service Patient admission and discharge analysis and bookkeeping in accounts departments The field of application of data warehouse systems is not only restricted to enterprises, but it also ranges from epidemiology to demography, from natural science to education. A property that is common to all fields is the need for storage and query tools to retrieve information summaries easily and quickly from the huge amount of data stored in databases or made available by the Internet. This kind of information allows us to study business phenomena, learn about meaningful correlations, and gain useful knowledge to support decision-making processes.

Data Staging and ETL Now let's closely study some basic features of the different architecture layers. We will start with the data staging layer. The data staging layer hosts the ETL processes that extract, integrate, and clean data from operational sources to feed the data warehouse layer. In a three-layer architecture, ETL processes actually feed the reconciled data layer—a single, detailed, comprehensive, top-quality data source—that in its turn feeds the data warehouse. For this reason, the ETL process operations as a whole are often defined as reconciliation. These are also the most complex and technically challenging among all the data warehouse process phases. ETL takes place once when a data warehouse is populated for the first time, then it occurs every time the data warehouse is regularly updated. Figure 1-8 shows that ETL consists of four separate phases: extraction (or capture), cleansing (or cleaning or scrubbing), transformation, and loading. In the following sections, we offer brief descriptions of these phases.

The scientific literature shows that the boundaries between cleansing and transforming are often blurred from the terminological viewpoint. For this reason, a specific operation is not always clearly assigned to one of these phases. This is obviously a formal problem, but not a substantial one. We will adopt the approach used by Hoffer and others (2005) to make our explanations as clear as possible. Their approach states that cleansing is essentially aimed at rectifying data values, and transformation more specifically manages data formats. Chapter 10 discusses all the details of the data-staging design phase. Chapter 3 deals with an early data warehouse design phase: integration. This phase is necessary if there are heterogeneous sources to define a schema for the reconciled data layer, and to specifically transform operational data in the data-staging phase.

CONCLUSION

ETL is identified with two tags: complexity and cost. Due its importance, this paper focused on ETL, the backstage of DW, and presents the research efforts and opportunities in connection with these processes. It is widely familiar that building ETL processes is expensive concerning time, money and effort. It consumes up to 70% of resources. Therefore, in current survey, firstly we give a review on open source and commercial ETL tools, along with some ETL prototypes coming from academic world. Namely, SIRIUS, ARKTOS. Before conclusion, we have given an picture of performance issue along review of some works dealing with this issue, particularly, ETL optimization and incremental ETL. Finally, this surveys ends with presentation of main challenges and research opportunities around ETL processes.

References

1. W. Inmon D. Strauss and G.Neushloss, “DW 2.0 The Architecture for the next generation of data warehousing”, Morgan Kaufman, 2007.
2. A. Simitisis, P. Vassiliadis, S.Skiadopoulos and T.Sellis, “DataWarehouse Refreshment”, Data Warehouses and OLAP: Concepts, Architectures and Solutions, IRM Press, 2007, pp 111-134.
3. R. Kimball and J. Caserta. “The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data”, Wiley Publishing, Inc, 2004.
4. A. Kabiri, F. Wadjinny and D. Chiadmi, "Towards a Framework for Conceptual Modelling of ETL Processes ", Proceedings of The first international conference on Innovative Computing Technology (INCT 2011), Communications in Computer and Information Science Volume 241, pp 146-160.
5. P.Vassiliadis and A.Simitsis, “EXTRACTION, TRANSFORMATION, AND LOADING“, http://www.cs.uoi.gr/~pvassil/publications/2009_DB_encyclopedia/Extract-Transform-Load.pdf
6. J. Adzic, V. Fiore and L. Sisto, “Extraction, Transformation, and Loading Processes”, Data Warehouses and OLAP: Concepts, Architectures and Solutions, IRM Press, 2007, pp 88-110. [7] W. Eckerson and C. White, “Evaluating ETL and Data Integration Platforms”, TDWI REPORT SERIES, 101communications LLC, 2003.
7. IBM InfoSphereDataStage, http://www-01.ibm.com/software/data/infosphere/datastage/