The Data warehouse Toolkit – Guide to Dimensional modelling

Data Capture and Data Analysis

Information is a very important asset and can be used either for operation record keeping or for analytical decision making. Some common attributes of an operational system are they are optimized to process transactional data very efficiently and that they usually don’t keep historical information. On the other hand, the Business intelligence systems are used to measure and evaluate performance. History is normally preserved.

What is the goal of a data warehouse?

To provide a scalable, efficient and easy way to access information

The way the data are structured should be understandable and intuitive to business users and not only to developers. In addition, the users should be able to use tools to access the data without large wait times – querying of the database should be fast

In addition, they data should be clean, consistent and credible – data quality is a very important aspect

There should be a single version of truth of the data

A data warehouse can help decision making – it’s a decision support system

But to be successful – the business users should actively use it and the management of a business or organization should perceive it as strategic.

Dimension modelling can provide data that are easy to understand and can improve query efficiency

ETL -The extract transform and Load system

Database systems, a practical approach

*Business intelligence (BI) is an umbrella term that refers to the processes for collecting and analyzing data, the technologies used in these processes, and the information obtained from these processes with the purpose of facilitating corporate decision making*

A single integrated view of the data is presented to the user

**Data Integration (the data warehouse etl tooklit)**

Significant Data Integration sometimes can happen before the data are imported in a data warehouse, for example when a business has an ERP. But even in those cases, it is likely that other systems and data sources exist outside of the ERP system

A data warehouse is not a project, it consists of many projects. Each data mart is a separate project with its own timeline - each data mart contains conformed dimensions so that each integrates into a single cohesive unit, the enterprise data warehouse. A better term is that a data warehouse is a process.

ETL

ETL stands for Extract, Transform and Load and it’s a term widely used in data warehousing. The ETL system is the backbone of a data warehouse as it is responsible for the extraction of the data from external sources, the cleaning of the data and the loading of the data into the data warehouse. The design and implementation of an ETL system is a complex project divided into many subtasks. There are many different methodologies, tools and technologies for ETL development and implementation. ETL design is a significant part of the Business Intelligence lifecycle (Moss and Atre Shaku, 2003). Before implementing any data flows, The ETL Team should take into consideration the business requirements and plan the ETL solution accordingly.

The Extract Step: The Extract Step of the ETL process should be planned based on the business needs. The business requirements set by end users define the data sources or specific entities and attributes of an operational system that need to be considered for integration into the data warehouse. Data Sources or other database objects that are not useful for analysis should not be considered.

The Transform Step: Part of the Transform step is also driven by the business needs. Specific Business rules are applied at this step. The data are cleaned, conformed and ready to be imported in the data warehouse.

The Load Step: The end users want to have easy access to information and they should be able to understand the underlying data model. The Business Intelligence Team should chose a data model that is simple, scalable and efficient and the ETL Team needs then to design the ETL processes to load the data efficiently in the data structure that has been implemented based on the chosen data model.

An additional business requirement that affects all of the steps above is the data latency requirement (Kimball Ralph, 2004). The frequency the data warehouse needs to be updated with fresh data is one of the most important aspects to consider by the design of an ETL solution.

To conclude, it is obvious that the ETL design and implementation is driven by the business requirements

The business requirements set by end users define the data sources or specific entities and attributes of an operational system that need to be considered for integration into the data warehouse. The end users also define the business rules that should be applied in the transformation stage.

(Kimball Ralph, 2004)

Capture changed data

Use audit fields / ensure that these fields are dependable and don’t have null values

The logic is to to compare the last modified date and

Time of each record to the maximum date and time that existed during the

Previous load and take all those that are greater.

Database Log Scraping or Sniffing /need to check this, it is used for real-time ETL. Drawback: The Log can be truncated and all transactions will be lost (p.107)

Avoid Time Extracts (take all rows that we inserted or modified getdate()-1) because if the ETL fails overnight then we may lose records

Process of Elimination (compare source tables and target table row by row to identify changes –not efficient)

Incremental load

Real Time Streaming ETL systems

From a technical architecture perspective, it has the potential to change the big-bang approach needed during the  
nightly batch ETL load windows to a continuous ETL-like flow throughout the day.

First generation of real/time ETL / the ODS, operational data store

Second generation: 2 fact tables, one real-time and one static p.427 – not sure what is the benefit of this

Point to Point vs Hub and Spoke solutions /Important! (Application integration)

**Micro batch ETL** / like conventional ETL but higher frequency

For micro batch, we have the following methods for identifying changed records:

* Timestamps/audit fields. These fields should have index to improve performance but the index increases operational overhead on inserts and updates
* ETL Log tables. A trigger is created in the OLTP database and populates a etl log table with the unique identifier of the rows that have changed or have updated. The etl process is joining then the etl log table with the source table and extracts the rows. The overhead on the OLTP system is reduced as the trigger driven Inserts are not heavy.
* (DBMS) log scrapers. Log scrappers can find the SQL statements of Inserts and Updates in the log files of the database and apply directly the changes to the target tables
* Network Sniffers

**Batch ETL** : simplest approach for delivering near real-time data warehousing reporting (one direction, no need to import back to source system)

**EIA: Enterprise Application Integration**

set of adapter and broker components

that move business transactions, in the form of messages, across the various

systems in the integration network

Publish&Subscribe Technology

**Capture, Transfrom, Flow**

The application layer of the transactional applications is bypassed. Instead, direct database-to-database exchanges are executed. Transactions, both new facts and dimension changes,  
can be moved directly from the operational systems to the data warehouse  
staging tables with low latency, typically a few second

**Enterprise Information Integration**

Real/time Reporting,transformations on the fly, no data warehouse

To read: https://tdwi.org/articles/2006/10/23/enterprise-information-integration-a-technology-for-providing-integrated-views.aspx

Paper-3:

Real Time Data Warehouses

1. Use the source system for reporting. Only if the source systems contains all the information we need and only if the performance of the OLTP database is not affected.
2. Extract
   1. Real Time Data replication with tools that mine the source database transaction logs and then micro batch or stream the transactions to the replication table. Drawback: No transformations.
   2. Message bus technology for transaction management. This is the publish/subscribe architecture. The data can be published to multiple consumer applications. Important: the data in the message bus should be published real-time.
   3. Micro batch architecture
   4. Mixed architecture E.g., use microbatching to transfer data to message bus.

Other comments: Don’t use staging/landing tables in real-time architecture. Redirect rejected rows to a file for post processing. Use stubbing in data loading, load both facts and dimensions the same time , if the transaction record is linked to new dimension records, map it with a dummy row in the dimension that will be created later.

Paper-4:

The data warehouse has become more operational and crucial to the business. Data has to be accurate, relevant, and complete. But most importantly, it has to be timely. Timely data ensures better-informed decisions. Check the diagram – data loading to DWH introduces the majority of data latency.

I can use the following for an introduction to evolution of ETL

The data warehouse evolution started 15 years ago when Bill Inmon defined it as "a subject-oriented,  
integrated, time-variant and non-volatile collection of data in support of management's decision making  
process." Businesses wanted to know “what happened”. During its early stages, data warehousing was all  
about reporting historic information aggregated from transaction processing systems with the goal of  
providing a unified, integrated view of business activities. Data was predominantly brought to the warehouse using custom scripts. Batch oriented, these scripts were executed monthly, weekly, and sometimes daily to  
update the warehouse

1st stage: Reporting –what happened

2nd stage: Analytical Why did it happen

3d stage: Predictive-What will happen

4th stage: Operational – what is happening now.

5th stage: Active – what do I want to happen

ETL methods:

Scripts -However, they pose many challenges such a drain on

Developer resource time and effort, in addition to administrative challenges such as manageability,

Documentation, and SLA compliance

ETL typical nightly maintenance windows- can be also combined with CDC but still the data are batched loaded to the data warehouse. The OLTP databases are growing and there is no time to take batch window .Micro batching, which is higher frequency of updating the data warehouse could be a solution but it it never achieves true real time.

EAI Enterprise Application Integration

Provide guaranteed data deliver and basic transformation

CDC-changed data captured

There are some CDC technologies that still operate in batch mode, with a pull approach. The ETL tool periodically request to receive a batch of all new changes. But there are also CDC technologies that push the data to the target database, so they offer a continuous streaming push approach

Golden Gate real time ETL is based on CDC technology that pushes the data to the target tables in real-time. Nice example of Montefiore Medical Center

Paper-7

SOA based real-time architecture

For the data capture, the Web Services are used. There is a module called data capture service

The data are in XML format and are exchanged through SOAP. So there is a Web Service Server for every marketing subsidiary and this data capture service which is also a Web Service is capturing the changed data from the other Web services!

Check more about web services and soap.

To expand further –EIA evolution from traditional messaging system to Apache Kafka, which supports also storage and transformations.

Most popular message brokers:

ActiveMQ, Apache Kafka, or RabbitMQ

Paper-22

EIA emerged in the mid-1990s – started as point to point connection but as the number of applications was growing, and the data transformation needs were more complicated , the Message-Oriented Middleware concept was introduced (MOM).

MOM provides asynchronous and loosely-coupled communications. It supports both queue and topic (publish/subscribe) model of messaging.

Pub-sub model : one to many, many to one , many to many communications.

Drawback – publishers and subscribers the still have to run on the MOM infrastructure to communicate.

Web services is a promising technology to achieve interoperation of heterogeneous environments but traditional WS uses a request response model and a synchronous communication. If the server has updated data, the client needs to send all the time new requests to get them. This wastes much network traffic and definitely increases service response time.

Solution to combine Web service technology with a pub/sub model that use push technology

Paper 20

Business demand today data as fresh as possible

Data warehouses are in use 25 hours a day

Latency of data integration is essential for the business value of the data warehouse

he capture of data from  
sources is either performed through incremental queries that filter based on a timestamp or  
flag, or through a Change Data Capture mechanism that detects any changes as it is  
happening. Architectures are further distinguished between pull and push operation, where a  
pull operation polls in fixed intervals for new data, while in a push operation data is loaded  
into the target once a change appears.

Check the table for a classification of etl techniques

Batch and Mini Batch, full or incremental load based on timestamps

Mini Batch / CDC –pull

Real Time /CDC –push

Oracle Golden Gate is using CDC as follows. It searched the transaction logs of the database and stage the changes in some files outside of the database and then to a staging database

There is a version of triggered CDC for medium loads

Then the changed records are moved to the target databases by using a publish subscribe system

I need to find out difference between SOA/ SOAP and kafka architecture

Paper -19

Service oriented architecture – evolution of a traditional BI architecture to support real-time, zero-latency delivery and closed loop architecture. All the components of the legacy system are replaced by service oriented components that communicate with open standard messaging protocols based on XML and SOAP

Paper – 9

A solution to real time –near real time, with a decrease of the batch window.

EAI vendors such as Tibco provide solutions for real-time data transport. For systems based on the latest Java technologies, Java Messaging Service (JMS) can be used to transmit each new data element from the source system to a lightweight listener application that in turn inserts the new data into the warehouse tables. For data that is received over the Internet, the data can be transmitted in XML via HTTP using the SOAP standard, and then loaded into the warehouse.

Drawback – the data warehouse cannot handle so frequent updates

Solution – stage the real time data other tables, similar to the structure of the tables in dwh, and then insert the data to the dwh in mini batches , check also paper 20 recommendation

Another solution is to have a dedicated server for the real time data, could be also an in-memory database separate from the data warehouse. As it is a dedicated environment, queries will be extremely fast.

If we choose a separate real time partion, we can then create a view and join it with the main table of the dwh that contains the historical data. If the real time partition/table is in-memory, querying will be fast..

Paper-12

Business value and action time (adapted from Hackathorn [6]) nice diagram to include

Complex logic with Grid computing, this paper is not focusing on real time ETL but there is a mention in continuous data streams.

Paper-11

Implementation of CDC working with triggers. For every insert or update the trigger is inserting to a log table the pk of the altered rows, together with some other information. The implementation of the trigger is introducing however significant overload of the database.

Paper-23

Log based CDC in oracle and it introduces a data queue and using an algorithm it gives priority to the most important data which are loaded first to the data warehouse

Paper 24

CDC, some tables in real-time, some other tables overnight, because the author claims that both ways have disadvantages: real-time add additional workload to the source systems , conventional ETL is vastness of data

Paper 25

One primary function of a data warehouse isto take the decision support system (DSS) load off the

operational systems. Data warehouses tend to be optimized for DSS queries while operational systems are usually optimized for production workloads such as OLTP. This paper discuss delta not only from the DBMS system but also from the other layers. From the databases the following methods are identified:

**Time stamps** (if these are included in the source system). Disadvantage-we capture only the last change prior to the exctraction, so this method is useful only when there is little change activity.

**Differential snapshot of the whole database** and then comparison (maybe in some systems only snapshots are allowed), like timestamps, it captures only the final state.

**Triggers -** If a source system permits **(**the triggered data are written to a table and then the table can be pulled to obtain the deltas).The advantage is that triggers capture all changes. The disadvantage is the significant -overhead. Especially when the triggers wrote data in another database (directly to the data warehouse for example) the overhead is ten to hundred times higher.

**Log extraction -**redo logs. No overhead, many replication software is based on logs. Disadvantage: Log formats change often and are different across different version of the same DBMS system. This means that exactly the same DBMS system must be the first recipient of the delta, or a third-party software is required to integrate logs that have different format. Another requirement is that the schema of the source and destination database should be the same.

**Op-Delta** log based method that captures deltas as operations that caused the changes, the sql statement itself is an Op-Delta. Inserts have the same overhead as generate equivalent number of sql statements, but the overhead of deletes and updates is significantly lower.

Paper 27

support day-to-day operations

How business conditions have changed?

-In the past, online systems would stop during the night or weekend to provide a window of time for running bulk ETL processes, today this is not acceptable.

-Business need timely information due to customer demand, competitive pressure and the need for improved decisions. Today decision makers need data that’s is updated a few times a day or near real time.

-Data volumes become larger and the ETL need more resources and time and the bulk extract windows are getting smaller.

-Cost reduction

Bulk ETL operations require more processing power, more memory and more band with, more administration and IT resources.

What CDC offers:

Makes integration efficient and real-time

CDC runs in the background and there is no degradation in the performance and service levels of the oltp systes

CDC provides current data by constantly identifying changes

CDC requires less resources in hardware and software

Key components of a CDC solution

-Change Capture Agents

Change capture agents are the software components that are responsible for the identification and capture of changes to the source operational data store

-Change Data Services

Filtering, life cycle management, auditing etc

-Change Delivery

Change delivery mechanisms are responsible for the reliable delivery of changed data to change consumers -typically an ETL program

Changes can be delivered including a push or pull model. A pull model means that the change

consumer asks for the changes on a periodic basis (as frequently as needed, typically every few minutes or hours),

preferably using a standard interface such as ODBC or JDBC. A push model means that the change consumer

listens and waits for changes, and those are delivered as soon as they are captured, typically using some

messaging middleware

Pull:

When should organizations use this approach? This batch-oriented approach is very easy to implement, as it is similar to traditional ETL processes and capitalizes on existing skill sets. Organizations should use this method when their latency requirements are measured in hours or minutes.

Push:

the change delivery

mechanism pushes the changes to the ETL program as soon as changes are captured. This is typically done using

a reliable transport such as an event-delivery mechanism or messaging middleware. This real-time approach is required when the BI

applications demand zero latency and the most up-to-date data

-Captured changes may need to be delivered to more than one consumer, such as

multiple ETL processes, data synchronization applications and business activity monitoring

Paper-29

Problems of near real time integration

. Repeated extraction and transformation of master

data for each data loading window is an overhead.

2. No plug-and-play suppor

master data -does not change often and transactional data which is updated regularly.

Enterprise Application Integration (EAI) [13][15]

also seems to integrate business applications near to

real-time basis. Two architectures, hub-spoke, and bus,

have been proposed to improve traditional ETL tools.

In the case of hub-spoke architecture scalability is

affected as the number of messages increase and also a

single point failure is introduced. In a bus architecture

the message transformation engine is distributed

among the applications, which improves the scalability

but makes it more complex to manage

Read again about the middleware

Paper-30

Web services etl, very generic , it doesn’t explain how these web services work

Paper 31

Change data capture can capture modified data in two modes:  
- Synchronous: Modified data is captured  
immediately. This is performed by using triggers on  
OLTP tables. This type of capturing mode effects  
OLTP as it is a part of transaction processing.  
- Asynchronous: Modified data is captured after SQL  
statement performs its operation. Then after  
committing the operation data is written into log  
files. These log files are read to capture modified  
data. It has no effect on OLTP.  
Some advantages of Change data Capture are:  
- Completeness: CDC can capture all effects of insert,  
update and delete operations. Also it can capture  
change in data value, if any, after modification.  
- Performance: Asynchronous change data capture  
has minimal impact on the source database.  
- Cost: Change data capture reduces overhead cost  
because it simplifies the extraction process as it  
extracts only change data from the database.  
- Makes ETL process efficient.  
- Data warehouse process and operational systems are  
completely decoupled from each other. This means  
any operational failure or recovery in either of the  
systems will not affect one another

There are various ways to implement CDC. Following are

some of the techniques to capture change data:

- Triggers: a trigger is a special kind of stored

procedure that is invoked whenever an attempt is

made to modify the data in the table against which

this trigger is written. Disadvantage of using

triggers for CDC is that it puts overhead burden on

the source database.

- Date & Timestamps: Each table in source database

contains a field named as „Last modified‟. It stores

the date and time when the data was lost modified.

CDC reads this last modified field and captures the

most recent modified data. Disadvantage of using

this CDC approach is that if program fails to write

into „Last Modified‟ field then some modified data

can be missed.

- Log – based CDC: Database maintains log files to

minimize the loss of data in case of an uncontrolled

shutdown. Speeds up the diagnostic process. Log

files contain detail of all the transactions which has

taken place on OLTP tables. Only log-based CDC is

one approach through which all changes can be

captured and updated into data warehouse and also

log-based technique has minimal impact on source

database.

Paper 33

Author claims that the best method for real time is the log based CDC

The key point is that database log files are designed to

reflect every possible change in the database.

Other approaches: Triggers also place a relatively

high overhead burden on the source database server

Database triggers – Can be deleted or removed, in addition they Triggers also place a relatively

high overhead burden on the source database server

Message Queues – a number of middleware

products such as IBM MQ Series also appear to have

the capacity to capture application, not database,

changes and report them to an ETL tool. An obvious

disadvantage is the cost of the license for these

products. More importantly, message queues are not

a completely reliable source of change information

since they only know about data changes that are

sent to them by the applications and not batch

routines or manual updates to the database.

Date and time stamps – many ERP applications

and other data sources maintain data fields within

each record that indicate when it was last changed.

An approach to CDC that has been tried a surprising

number of times reads through the data records and

looks for recent changes. The fatal flaw in this

approach is that it relies on the programs that change

data to unfailingly update this field. In addition, this

approach can lose track of deletions since the entire

record, including the time stamps, are gone

paper-37

demand for fresh data from the part of the users

In such a data warehouse setting, data are

extracted from the sources, transformed, cleaned, and eventually loaded to the warehouse. This set of activities takes place during a loading window, usually during the

night, to avoid overloading the source production systems with the extra workload

of this workflow.

Reasons to build an near real time data warehouse

-websites as a source, where the data can change fast

-increased competition, need for bigger sales, better monitoring of customers goals, precise monitoring of the stock market

-globalization of the economy e.g., organization’s branches may be spread in

places with totally different time-zones

From now on - Less data are moving from  
the source towards the data warehouse, more frequently, and at a faster rate

Data sources of ETL –OLTP systems, legacy systems, web sites, flat files

Problems of ETL

-Large volumes of data

-data quality, the data are not cleaned.

-evolution of source systems and data warehouse, higher level of maintenance

In practice, the ETL

process periodically refreshes the data warehouse during idle or low-load, periods of its operation; e.g., every night. Any failures of the process must also be

compensated within the specified time windows.

**Extraction phase**

Not easy because of two reasons:

the source must suffer minimum overhead during the extraction, since other administrative activities also take  
place during that period, and, (b) both for technical and political reasons, administrators are quite reluctant to accept major interventions to their system’s configuration

4 policies:

-import the whole data source in each ETL execution, not practical because of large volumes of data

-database triggers, not practical because it places overhead to the source and also it requires changes in the configurations.

-timestamps

-log files

**Transform phase**

**Load phase**

Bulk or insert statements row by row (not effective)

Still, it appears that

data warehouses have fallen victims of their success: users are no more satisfied

with data that are one day old and press for fresh data -if possible, with instant reporting.

A real-time data warehouse architecture should ensure a maximum freshness of data , minimal overhead at the source systems, a controlled environment where the administrator can respond quickly to unexpected events. Also it should scalable as new data sources will come and the user community will be more energetic.

**EAI**

Push technology , but very complicated an expensive for medium companies

whose functionality entails a set of adapter and broker

components that move business transactions - in the form of messages - across the

various systems in the integration network. An adapter creates and executes the

messages, while a broker routes messages, based on publications and subscription

rules.

See kimbal etl tool kit for the rest

Real time infrastructure

ETL processes, pipelining methods and partitioning methods.

Pipelining / each step is processed in a different processor (extract, transform , load)

Partitioning / the data source is split in partitions and each partition is being processed by a different processor simultaneously

Paper 28

The data warehouse facilitates complex data analyses without placing a burden on the operational source systems that run the day-to-day business

Today’s business users, however, demand for up-to-date data analyses to support timely decision making. A workable solution to this challenge is shortening

the data warehouse loading cycles. This approach is referred to as near realtime data warehousing or microbatch ETL

paper 32

The definition for each Data Quality Dimension is

as follows:

1. Completeness –The completeness of data

shows or ensures the extent to which the

expected attributes of data are provided and

all required information is available. It is

imperative to note that although the data

may not be available, it could still be

considered completed. This happens when

the data meets the expectations of the user.

Figure 3: Impediments to Information

Management Success

(Henschen, D., 2010)

Figure 4: Data Quality Dimensions

(Singh, R., Singh, K., 2010)

2. Consistency – In order for data to be

consistent, data across the enterprise should

be harmonious with each other with values

not having any conflicts across all data sets.

3. Validity – This refers to the correctness and

reasonableness of data.

4. Conformity – Data conformity means that

data values are consistent across specific

formats. Maintaining conformance to specific

formats is important.

5. Accuracy – Data is said to be accurate if the

data correctly reflects the real world object

or an event being described. For example,

incorrect spellings of product or person

names, addresses, and even untimely or not

current data can affect operational and

analytical applications.

6. Integrity – The integrity of data refers to the

trustworthiness of the data. If data is

missing important relationship linkages and

is unable to link related records together,

then it may actually introduce duplication

across all systems. (Singh, R., Singh, K.,

2010)

Additional factors concerning data quality include

confidentiality, availability, and security.

The quality of data and the effectiveness of a

data warehouse are directly dependent on the

efficiency of the ETL process. Hence, a quality

ETL process begets quality decision-making

power. Research has found that seventy percent

(70%) of the software implementation and

maintenance effort of data warehousing is spent

on the ETL system (Behrend, A., Jörg, T., 2010).

3 generations of ETL tools

The First-generation ETL tools were written in

the native code of the operating system platform

and would only execute on the native operating

system. The most commonly generated code

was COBOL code because the first generation

data was stored on mainframes. These tools

made the data integration process easy since

the native code performance was good but there

was a maintenance problem.

Second generation ETL tools have proprietary

ETL engines to execute the transformation

processes. Second generation tools have

simplified the job of developers because they

only need to know only one programming

language i.e. ETL programming. Data coming

from different heterogeneous sources should

pass through the ETL engine row by row and be

stored on the target system. This was a slow

process and this generation of ETL programs

suffered from a high performance overload.

Third Generation ETL tools have a distributed

architecture with the ability to generate native

SQL. This eliminates the hub server between the

source and the target systems. The distributed

architecture of third generation tools reduces the

network traffic to improve the performance,

distributes the load among database engines to

improve the scalability, and supports all types of

data sources. Third Generation ETL uses relational DBMS for

data transformations. In this generation the

transformation phase does processing of data

rather than row by row as in second generation

ETL tools. “In the ETL architecture, all database

engines can potentially participate in a

transformation—thus running each part of the

process where it is the most optimized. Any

RDBMS can be an engine, and it may make

sense to distribute the SQL code among different

sources and targets to achieve the best

performance. For example, a join between two

large tables may be done on the source” (De

Montcheuil, Y., 2005). RDBMS have power for

data integration; ETL tools are taking the

advantage of this feature of the RDBMS to

improve their performance.

Complex event processing concept