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.