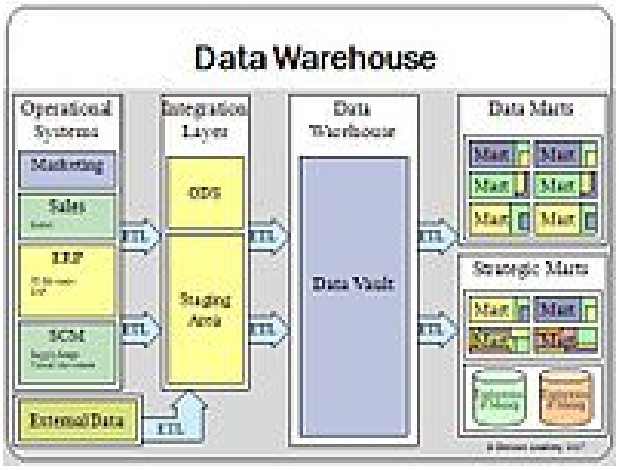
Data warehouse

From Wikipedia, the free encyclopedia
Various Comments inserted by Freymann, Attaway, & Austin (OHIP). Key passages underlined.



The figure to the left is obviously blurred, but doesn't really show anything different than what the Model Group of Dolce came up with.

The figure below is from a different source and is clearer and better represents these DW concepts as applied in the DPH.

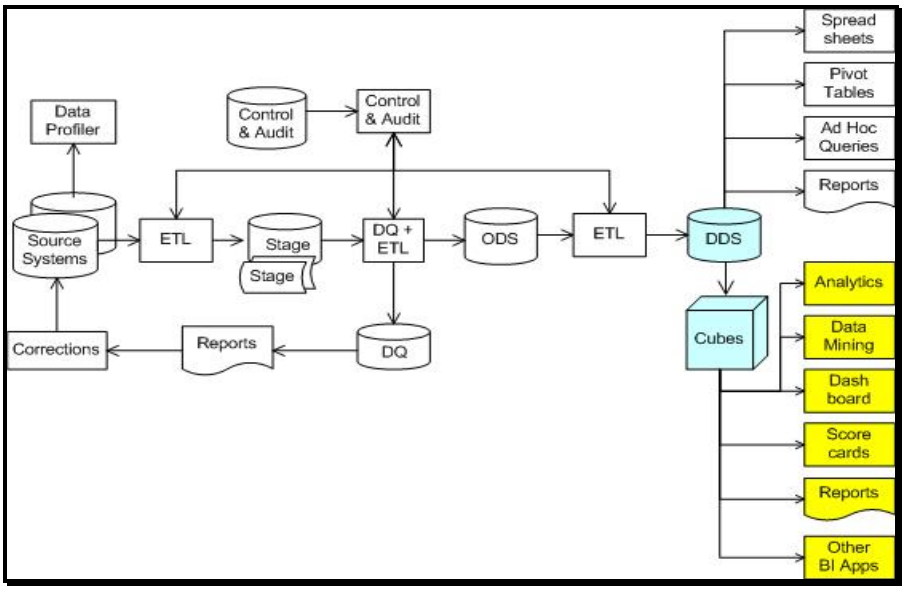


Figure 1

Figure 1 from
http://www.sqlservercentral.com/articles/design+and+theory/businessintelligenceordatawarehouse/2409/

Data Warehouse Overview

In computing, a **data warehouse** (**DW**) is a database used for reporting and analysis. The data stored in the warehouse is uploaded from the operational systems. The data may pass through an operational data store (ODS) for additional operations before it is used in the DW for reporting.

A data warehouse maintains its functions in three layers: staging, integration, and access. *Staging* is used to store raw data for use by developers. The *integration* layer is used to integrate data and to have a level of abstraction from users. The *access* layer is for getting data out for users.

One thing to mention about data warehouse is that they can be subdivided into data marts. With data marts it stores subsets of data from a warehouse, which focuses on a specific aspect of a company like sales or a marketing process.

This definition of the data warehouse focuses on data storage. The main source of the data is cleaned, transformed, catalogued and made available for use by managers and other business professionals for data mining, online analytical processing (OLAP), market research and decision support (Marakas & O'Brien 2009). However, the means to retrieve and analyze data, to extract, transform and load data, and to manage the data dictionary are also considered essential components of a data warehousing system. Many references to data warehousing use this broader context. Thus, an expanded definition for data warehousing includes business intelligence tools, tools to extract, transform and load (ETL) data into the repository, and tools to manage and retrieve metadata.

Benefits of a data warehouse

A data warehouse maintains a copy of information from the source transaction systems. This architectural complexity provides the opportunity to:

- Maintain data history, even if the source transaction systems do not.
- Integrate data from multiple source systems, enabling a central view across the enterprise. This benefit is always valuable, but particularly so when the organization has grown by merger.
- Improve data, by providing consistent codes and descriptions, flagging or even fixing bad data.
- Present the organization's information consistently.
- Provide a single common data model for all data of interest regardless of the data's source.
- Restructure the data so that it makes sense to the business users.
- Restructure the data so that it delivers excellent query performance, even for complex analytic queries, without impacting the operational systems.
- Add value to operational business applications, notably customer relationship management (CRM) systems.

Comment [grf1]: OHIP uses an ODS.

Comment [grf2]: In the layered view the staging layer would be the ODS structures. the integration layer would be the repository and cubes and the associated standards.

Comment [rma3]: Access layers are the **Repository and Oasis**. Access must also be abstract when done via the web because users cannot be expected to have the expertise needed to extract data properly

Comment [grf4]: GRITS could be one for example.

Comment [rma5]: Can be physical like a new database or logical like different schema to contain the different "marts"

Comment [grf6]: Right – OLAP – i.e. Cubes of data to drive OASIS

Comment [grf7]: Data Dictionary – the "Standard Data Properties" document – describes the data in the Repository (warehouse).

Comment [rma8]: This must be the only way an enterprise data warehouse can function due to the multivariate data sources involved

Comment [grf9]: OASIS web query, mapping, MINER (former Cross-tab), and Dashboard…

Comment [rma10]: Yes, this is why OHIP keeps the original fields

Comment [grf11]: Or, is a collection of various business operations – immunizations, WIC, vital records, etc…

Comment [grf12]: Data Standards! "Standard Data Properties" document.

Comment [grf13]: This point is critical. Doesn't matter what the source is. Already accomplished for VR, ER, Discharge…

Comment [grf14]: Done. Dimensionalized and cubed.

Comment [grf15]: - eg. programs, epi, in addition to crm using clinical data

History

The concept of data warehousing dates back to the late 1980s [1] when IBM researchers Barry Devlin and Paul Murphy developed the "business data warehouse". In essence, the data warehousing concept was intended to provide an architectural model for the flow of data from operational systems to decision support environments. The concept attempted to address the various problems associated with this flow, mainly the high costs associated with it. In the absence of a data warehousing architecture, an enormous amount of redundancy was required to support multiple decision support environments. In larger corporations it was typical for multiple decision support environments to operate independently. Though each environment served different users, they often required much of the same stored data. The process of gathering, cleaning and integrating data from various sources, usually from long-term existing operational systems (usually referred to as legacy systems), was typically in part replicated for each environment. Moreover, the operational systems were frequently reexamined as new decision support requirements emerged. Often new requirements necessitated gathering, cleaning and integrating new data from "data marts" that were tailored for ready access by users.

Comment [rma16]: This is the exact reason for Enterprise Data Standards that must be enforced, communicated and all of our work immersed so programs can easily integrate their operational data into the data warehouse. See the "Standard Data Properties" for an example on health.state.ga.us/phil

Key developments in early years of data warehousing were:

- 1960s — General Mills and Dartmouth College, in a joint research project, develop the terms *dimensions* and *facts*.[2]
- 1970s — ACNielsen and IRI provide dimensional data marts for retail sales.[2]
- 1970s — Bill Inmon begins to define and discuss the term: Data Warehouse
- 1975 — Sperry Univac Introduce MAPPER (MAintain, Prepare, and Produce Executive Reports) is a database management and reporting system that includes the world's first 4GL. It was the first platform specifically designed for building Information Centers (a forerunner of contemporary Enterprise Data Warehousing platforms)
- 1983 — Teradata introduces a database management system specifically designed for decision support.
- 1983 — Sperry Corporation Martyn Richard Jones defines the Sperry Information Center approach, which whilst not being a true DW in the Inmon sense, did contain many of the characteristics of DW structures and process as defined previously by Inmon, and later by Devlin. First used at the TSB England & Wales
- 1984 — Metaphor Computer Systems, founded by David Liddle and Don Massaro, releases Data Interpretation System (DIS). DIS was a hardware/software package and GUI for business users to create a database management and analytic system.
- 1988 — Barry Devlin and Paul Murphy publish the article *An architecture for a business and information systems* in *IBM Systems Journal* where they introduce the term "business data warehouse".
- 1990 — Red Brick Systems, founded by Ralph Kimball, introduces Red Brick Warehouse, a database management system specifically for data warehousing.
- 1991 — Prism Solutions, founded by Bill Inmon, introduces Prism Warehouse Manager, software for developing a data warehouse.
- 1992 — Bill Inmon publishes the book *Building the Data Warehouse*.[3]
- 1995 — The Data Warehousing Institute, a for-profit organization that promotes data warehousing, is founded.
- 1996 — Ralph Kimball publishes the book *The Data Warehouse Toolkit*.[4]
- 2000 — Daniel Linstedt releases the *Data Vault*, enabling real time auditable Data Warehouses warehouse.

Normalized versus Dimensional approach for storage of data

There are two leading approaches to storing data in a data warehouse — the dimensional approach and the normalized approach. The dimensional approach, whose supporters are referred to as “Kimballites”, believe in Ralph Kimball’s approach in which it is stated that the data warehouse should be modeled using a Dimensional Model/star schema. The normalized approach, also called the 3NF model, whose supporters are referred to as “Inmonites”, believe in Bill Inmon's approach in which it is stated that the data warehouse should be modeled using an E-R model/normalized model.

In a dimensional approach, transaction data are partitioned into either "facts", which are generally numeric transaction data, or "dimensions", which are the reference information that gives context to the facts. For example, a sales transaction can be broken up into facts such as the number of products ordered and the price paid for the products, and into dimensions such as order date, customer name, product number, order ship-to and bill-to locations, and salesperson responsible for receiving the order.

A key advantage of a dimensional approach is that the data warehouse is easier for the user to understand and to use. Also, the retrieval of data from the data warehouse tends to operate very quickly. Dimensional structures are easy to understand for business users, because the structure is divided into measurements/facts and context/dimensions. Facts are related to the organization’s business processes and operational system whereas the dimensions surrounding them contain context about the measurement (Kimball, Ralph 2008).

The main disadvantages of the dimensional approach are:

1. In order to maintain the integrity of facts and dimensions, loading the data warehouse with data from different operational systems is complicated, and
2. It is difficult to modify the data warehouse structure if the organization adopting the dimensional approach changes the way in which it does business.

In the normalized approach, the data in the data warehouse are stored following, to a degree, database normalization rules. Tables are grouped together by *subject areas* that reflect general data categories (e.g., data on customers, products, finance, etc.). The normalized structure divides data into entities, which creates several tables in a relational database. When applied in large enterprises the result is dozens of tables that are linked together by a web of joins. Furthermore, each of the created entities is converted into separate physical tables when the database is implemented (Kimball, Ralph 2008). The main advantage of this approach is that it is straightforward to add information into the database. A disadvantage of this approach is that, because of the number of tables involved, it can be difficult for users both to:

1. Join data from different sources into meaningful information and then
2. Access the information without a precise understanding of the sources of data and of the data structure of the data warehouse.

Comment [grf17]: Dimensions OHIP uses are for example Person Place and Time

Comment [rma18]: Due to the nature of Department data sources the data warehouse is a hybrid of these two concepts

Comment [grf19]: Again, at a minimum, the Age, Race, Sex of a person, When an event occurred, and Where.

Comment [rma20]: If data standards and programmatic program planning are done properly this is not hard to overcome.

Comment [grf21]: Therefore more suited to rapid transactions and frequent updates.

Comment [grf22]: Ah…. But this is precisely what we want to do!

Comment [rma23]: The Department data standards allow us to do this in a similar fashion to3NF structures without the loss of computing power

It should be noted that both normalized and dimensional models can be represented in entity-relationship diagrams as both contain jointed relational tables. The difference between the two models is the degree of normalization.

These approaches are not mutually exclusive, and there are other approaches. Dimensional approaches can involve normalizing data to a degree (Kimball, Ralph 2008).

Conforming information

Another important fact in designing a data warehouse is which data to conform and how to conform the data. For example, one operational system feeding data into the data warehouse may use "M" and "F" to denote sex of an employee while another operational system may use "Male" and "Female". Though this is a simple example, much of the work in implementing a data warehouse is devoted to making similar meaning data consistent when they are stored in the data warehouse. Typically, extract, transform, load tools are used in this work.

Comment [grf24]: A ton of the work. Work that needs to continue to be applied (before any code is written).

Top-down versus bottom-up design methodologies

Bottom-up design

Ralph Kimball, a well-known author on data warehousing,[5] is a proponent of an approach to data warehouse design which he describes as *bottom-up*.[6]

Comment [rma25]: This follows the concept for the Department in how data are structured. Row level data, Aggregated working data, High level analytic data

In the *bottom-up* approach data marts are first created to provide reporting and analytical capabilities for specific business processes. Though it is important to note that in Kimball methodology, the bottom-up process is the result of an initial business oriented Top-down analysis of the relevant business processes to be modelled.

Data marts contain, primarily, dimensions and facts. Facts can contain either atomic data and, if necessary, summarized data. The single data mart often models a specific business area such as "Sales" or "Production." These data marts can eventually be integrated to create a comprehensive data warehouse. The integration of data marts is managed through the implementation of what Kimball calls "a data warehouse bus architecture".[7] **The data warehouse bus architecture is primarily an implementation of "the bus", a collection of conformed dimensions and conformed facts, which are dimensions that are shared (in a specific way) between facts in two or more data marts.**

Comment [grf26]: Approach OHIP uses. At a minimum, person place and time.

The integration of the data marts in the data warehouse is centered on the conformed dimensions (residing in "the bus") that define the possible integration "points" between data marts. The actual integration of two or more data marts is then done by a process known as "Drill across". A drill-across works by grouping (summarizing) the data along the keys of the (shared) conformed dimensions of each fact participating in the "drill across" followed by a join on the keys of these grouped (summarized) facts.

Maintaining tight management over the data warehouse bus architecture is fundamental to maintaining the integrity of the data warehouse. The most important management task is making

sure dimensions among data marts are consistent. In Kimball's words, this means that the dimensions "conform".

Some consider it an advantage of the Kimball method, that the data warehouse ends up being "segmented" into a number of logically self contained (up to and including The Bus) and consistent data marts, rather than a big and often complex centralized model. Business value can be returned as quickly as the first data marts can be created, and the method gives itself well to an exploratory and iterative approach to building data warehouses. For example, the data warehousing effort might start in the "Sales" department, by building a Sales-data mart. Upon completion of the Sales-data mart, The business might then decide to expand the warehousing activities into the, say, "Production department" resulting in a Production data mart. **The requirement for the Sales data mart and the Production data mart to be integrable, is that they share the same "Bus", that will be, that the data warehousing team has made the effort to identify and implement the conformed dimensions in the bus, and that the individual data marts links that information from the bus.** Note that this **does not** require 100% awareness from the onset of the data warehousing effort, no master plan is required upfront. The Sales-data mart is good as it is (assuming that the bus is complete) and the production data mart **can be constructed virtually independent of the sales data mart (but not independent of the Bus).**

If integration via the bus is achieved, the data warehouse, through its two data marts, will not only be able to deliver the specific information that the individual data marts are designed to do, in this example either "Sales" or "Production" information, but can deliver integrated Sales-Production information, which, often, is of critical business value. An integration (possibly) achieved in a flexible and iterative fashion.

Top-down design

Bill Inmon, one of the first authors on the subject of data warehousing, has defined a data warehouse as a centralized repository for the entire enterprise.[7] Inmon is one of the leading proponents of **the *top-down* approach to data warehouse design, in which the data warehouse is designed using a normalized enterprise data model. "Atomic" data, that is, data at the lowest level of detail, are stored in the data warehouse.** Dimensional data marts containing data needed for specific business processes or specific departments are created from the data warehouse. In the Inmon vision the data warehouse is at the center of the "Corporate Information Factory" (CIF), which provides a logical framework for delivering business intelligence (BI) and business management capabilities.

Inmon states that the data warehouse is:

Subject-oriented
: The data in the data warehouse is organized so that all the data elements relating to the same real-world event or object are linked together.

Non-volatile
: Data in the data warehouse are never over-written or deleted — once committed, the data are static, read-only, and retained for future reporting.

Comment [rma27]: Each source cube is a Mart unto itself but when joined via the "Bus" architecture the power of conformed dimensionality reveals itself

Comment [grf28]: Via Dimensions.

Comment [grf29]: Another advantage in our complex and changing enterprise.

Comment [rma30]: These statements are the exact reason the Department has a hybrid approach to the data warehouse

Integrated
: The data warehouse contains data from most or all of an organization's operational systems and these data are made consistent.

Time-variant

The top-down design methodology generates highly consistent dimensional views of data across data marts since all data marts are loaded from the centralized repository. Top-down design has also proven to be robust against business changes. Generating new dimensional data marts against the data stored in the data warehouse is a relatively simple task. **The main disadvantage to the top-down methodology is that it represents a very large project with a very broad scope.** The up-front cost for implementing a data warehouse using the top-down methodology is significant, and the duration of time from the start of project to the point that end users experience initial benefits can be substantial. In addition, the top-down methodology can be inflexible and unresponsive to changing departmental needs during the implementation phases.[7]

Comment [grf31]: This probably was the thinking back in 1998, and partly why this approach failed.

Hybrid design

Data warehouse (DW) solutions often resemble hub and spoke architecture. Legacy systems feeding the DW/BI solution often include customer relationship management (CRM) and enterprise resource planning solutions (ERP), generating large amounts of data. To consolidate these various data models, and facilitate the extract transform load (ETL) process, DW solutions often make use of an operational data store (ODS). The information from the ODS is then parsed into the actual DW. To reduce data redundancy, larger systems will often store the data in a normalized way. Data marts for specific reports can then be built on top of the DW solution.

It is important to note that the DW database in a hybrid solution is kept on third normal form to eliminate data redundancy. A normal relational database however, is not efficient for business intelligence reports where dimensional modelling is prevalent. Small data marts can shop for data from the consolidated warehouse and use the filtered, specific data for the fact tables and dimensions required. The DW effectively provides a single source of information from which the data marts can read, creating a highly flexible solution from a BI point of view. The hybrid architecture allows a DW to be replaced with a master data management solution where operational, not static information could reside.

The Data Vault Modeling components follow hub and spoke architecture. This modeling style is a hybrid design, consisting of the best of breed practices from both 3rd normal form and star schema. The Data Vault model is not a true 3rd normal form, and breaks some of the rules that 3NF dictates be followed. It is however, a top-down architecture with a bottom up design. The Data Vault model is geared to be strictly a data warehouse. It is not geared to be end-user accessible, which when built, still requires the use of a data mart or star schema based release area for business purposes.

Data warehouses versus operational systems

Operational systems are optimized for preservation of data integrity and speed of recording of business transactions through use of database normalization and an entity-relationship model.

Operational system designers generally follow the Codd rules of database normalization in order to ensure data integrity. Codd defined five increasingly stringent rules of normalization. Fully normalized database designs (that is, those satisfying all five Codd rules) often result in information from a business transaction being stored in dozens to hundreds of tables. Relational databases are efficient at managing the relationships between these tables. The databases have very fast insert/update performance because only a small amount of data in those tables is affected each time a transaction is processed. Finally, in order to improve performance, older data are usually periodically purged from operational systems.

Comment [grf32]: Fine for Wal-mart that needs to know the second an item is at check-out and thus removed from the shelf.

Data warehouses are optimized for speed of data analysis. Frequently data in data warehouses are denormalised via a dimension-based model. Also, to speed data retrieval, data warehouse data are often stored multiple times—in their most granular form and in summarized forms called aggregates. Data warehouse data are gathered from the operational systems and held in the data warehouse even after the data has been purged from the operational systems.

Comment [grf33]: Yes – the OASIS web query returns are always ON-THE-FLY! Think about that…

Evolution in organization use

These terms refer to the level of sophistication of a data warehouse:

Offline operational data warehouse
: Data warehouses in this stage of evolution are updated on a regular time cycle (usually daily, weekly or monthly) from the operational systems and the data is stored in an integrated reporting-oriented data

Offline data warehouse
: Data warehouses at this stage are updated from data in the operational systems on a regular basis and the data warehouse data are stored in a data structure designed to facilitate reporting.

On time data warehouse
: Online Integrated Data Warehousing represent the real time Data warehouses stage data in the warehouse is updated for every transaction performed on the source data

Integrated data warehouse
: These data warehouses assemble data from different areas of business, so users can look up the information they need across other systems.[8]

Sample applications

Some of the applications data warehousing can be used for are:

- Decision support
- Trend analysis
- Financial forecasting
- Churn Prediction for Telecom subscribers, Credit Card users etc.
- Insurance fraud analysis
- Call record analysis
- Logistics and Inventory management[9]

References

1. ^ "The Story So Far". 2002-04-15. http://www.computerworld.com/databasetopics/data/story/0,10801,70102,00.html. Retrieved 2008-09-21.
2. ^ [a] [b] Kimball 2002, pg. 16
3. ^ Inmon, Bill (1992). *Building the Data Warehouse*. Wiley. ISBN 0471569607.
4. ^ Kimball, Ralph (1996). *The Data Warehouse Toolkit*. Wiley. ISBN 0471153370.
5. ^ Kimball 2002, pg. 310
6. ^ "The Bottom-Up Misnomer". 2003-09-17. http://www.intelligententerprise.com/030917/615warehouse1_1.jhtml. Retrieved 2008-11-05. [*dead link*]
7. ^ [a] [b] [c] Ericsson 2004, pp. 28-29
8. ^ "Data Warehouse". http://www.tech-faq.com/data-warehouse.html.
9. ^ Abdullah, Ahsan (2009). "Analysis of mealybug incidence on the cotton crop using ADSS-OLAP (Online Analytical Processing) tool , Volume 69, Issue 1". *Computers and Electronics in Agriculture* **69**: 59–72. doi:10.1016/j.compag.2009.07.003.

This article **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. *(February 2008)*

[edit] Further reading

- Davenport, Thomas H. and Harris, Jeanne G. *Competing on Analytics: The New Science of Winning* (2007) Harvard Business School Press. ISBN 978-1-4221-0332-6
- Ganczarski, Joe. *Data Warehouse Implementations: Critical Implementation Factors Study* (2009) VDM Verlag ISBN 3-639-18589-7 ISBN 978-3-639-18589-8
- Kimball, Ralph and Ross, Margy. *The Data Warehouse Toolkit* Second Edition (2002) John Wiley and Sons, Inc. ISBN 0-471-20024-7
- Linstedt, Graziano, Hultgren. *The Business of Data Vault Modeling* Second Edition (2010) Dan linstedt, ISBN 978-1-4357-1914-9