恩門 BILLINMON.COM

What Is A Data Warehouse?

BY

W. H. Inmon

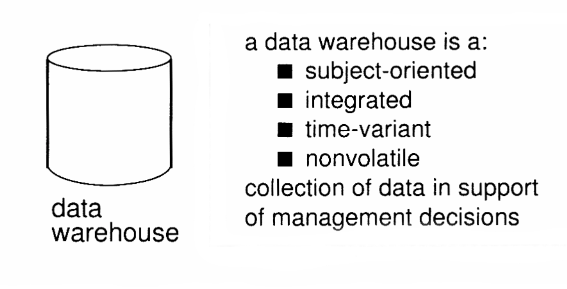
Data warehouse is the center of the architecture for information systems for the 1990's. Data warehouse supports informational processing by providing a solid platform of integrated, historical data from which to do analysis. Data warehouse provides the facility for integration in a world of unintegrated application systems. Data warehouse is achieved in an evolutionary, step at a time fashion. Data warehouse organizes and stores the data needed for informational, analytical processing over a long historical time perspective. There is indeed a world of promise in building and maintaining a data warehouse.

What then is a data warehouse?

A data warehouse is a:

- subject oriented,
- integrated,
- time variant,
- non volatile

collection of data in support of management's decision making process, as shown in Figure 1.



a data warehouse is a:

- subject-oriented
- integrated
- time-variant
- nonvolatile

collection of data in support of management decisions

Figure 1: What is a data warehouse?

The data entering the data warehouse comes from the operational environment in almost every case. The data warehouse is always a physically separate store of data transformed from the application data found in the operational environment.

This bookish definition of a data warehouse deserves a full explanation because there are some important issues and subtleties underlying the characteristics of a warehouse.

Subject Orientation

The first feature of the data warehouse is that it is oriented around the major subjects of the enterprise. The data driven, subject orientation is in contrast to the more classical process/functional orientation of applications, which most older operational systems are organized around. Figure 2 shows the contrast between the two types of orientations.

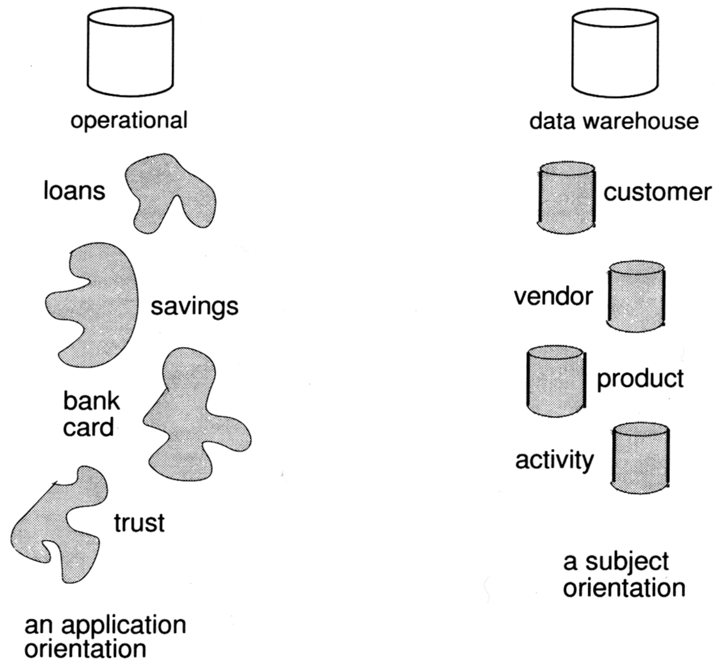


Figure 2: The data warehouse has a strong subject orientation.

The operational world is designed around applications and functions such as loans, savings, bankcard, and trust for a financial institution. The data warehouse world is organized around major subjects such as customer, vendor, product, and activity. The alignment around subject areas affects the design and implementation of the data found in the data warehouse. Most prominently, the major subject area influences the most important part of the key structure.

The application world is concerned both with data base design and process design. The data warehouse world focuses on data modeling and database design exclusively. Process design (in its classical form) is not part of the data warehouse environment.

The differences between process/function application orientation and subject orientation show up as a difference in the content of data at the detailed level as well. Data warehouse data does not include data that will not be used for DSS processing, while operational application oriented data contains data to satisfy immediate functional/processing requirements that may or may not have any use to the DSS analyst.

Another important way in which the application oriented operational data differs from data warehouse data is in the relationships of data. Operational data maintains an ongoing relationship between two or more tables based on a business rule that is in effect. Data warehouse data spans a spectrum of time and the relationships found in the data warehouse are many. Many business rules (and correspondingly, many data relationships) are represented in the data warehouse between two or more tables. (For a detailed explanation of how data relationships are handled in the data warehouse, refer to the Tech Topic on that subject.)

From no other perspective than that of the fundamental difference between a functional/process application orientation and a subject orientation, there is a major difference between operational systems and data and the data warehouse.

INTEGRATION

Easily the most important aspect of the data warehouse environment is that data found within the data warehouse is integrated. ALWAYS. WITH NO EXCEPTIONS. The very essence of the data warehouse environment is that data contained within the boundaries of the warehouse is integrated.

The integration shows up in many different ways - in consistent naming conventions, in consistent measurement of variables, in consistent encoding structures, in consistent physical attributes of data, and so forth.

Contrast the integration found within the data warehouse with the lack of integration found in the applications environment, and the differences are stark, as shown by Figure 3.

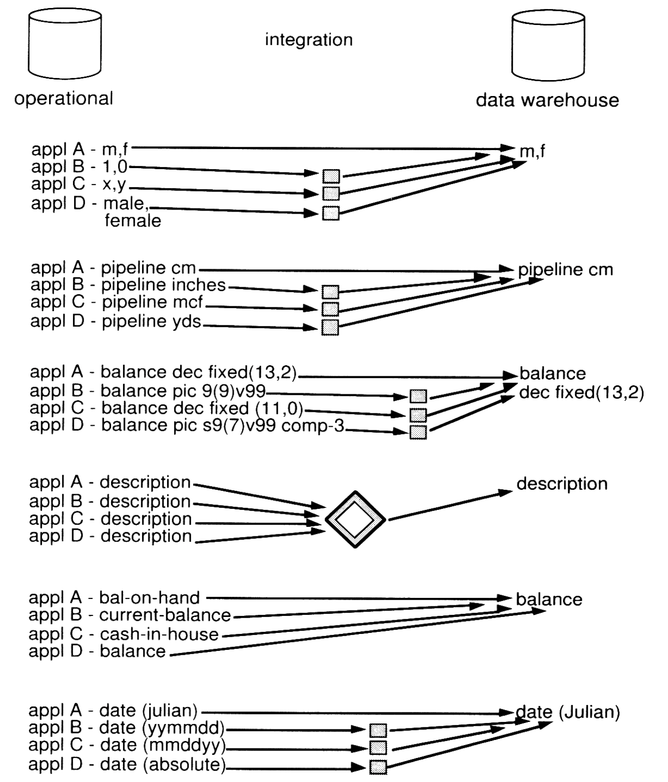


Figure 3: When data is moved to the data warehouse from the application-oriented operational environment, the data is integrated before entering the warehouse.

Over the years the different applications designers have made their own, many decisions as to how an application should be built. The style and the individualized design decisions of the application designer show up in a hundred ways: In differences in encoding, key structures, physical characteristics, naming conventions, and so forth. The collective ability of many application designers to create inconsistent applications is legendary.

Figure 3 shows some of the most important differences in the ways applications are designed.

Encoding:
Application designers have chosen to encode the field - gender - in different ways. One designer represents gender as an "m" and an "f". Another application designer represents gender as a "1" and a "0". Another application designer represents gender as an "x" and a "y". And yet another application designer represents gender as "male" and "female." It doesn't matter much how gender arrives in the data warehouse. "M" and "F" are probably as good as any representation. What matters is that whatever source gender comes from, that gender arrives in the data warehouse in a consistent integrated state. Therefore when gender is loaded into the data warehouse from an application where it has been represented in other than a "M" and "F" format, the data must be converted to the data warehouse format.

Measurement of Attributes:
Application designers have chosen to measure pipeline in a variety of ways over the years. One designer stores pipeline data in centimeters. Another application designer stores pipeline data in terms of inches. Another application designer stores pipeline data in million cubic feet per second. And another designer stores pipeline information in terms of yards. Whatever the source, when the pipeline information arrives in the data warehouse it needs to be measured the same way.

As shown in Figure 3 the issues of integration affect almost every aspect of design - the physical characteristics of data, the dilemma of having more than one source of data, the issue of inconsistent naming standards, inconsistent date formats, and so forth.

Whatever the design issue, the result is the same - the data needs to be stored in the data warehouse in a singular, globally acceptable fashion even when the underlying operational systems store the data differently.

When the DSS analyst looks at the data warehouse, the focus of the analyst should be on using the data that is in the warehouse, rather than on wondering about the credibility or consistency of the data.

TIME VARIANCY

All data in the data warehouse is accurate as of some moment in time. This basic characteristic of data in the warehouse is very different from data found in the operational environment. In the operational environment data is accurate as of the moment of access. In other words, in the operational environment when you access a unit of data, you expect that it will reflect accurate values as of the moment of access.

Because data in the data warehouse is accurate as of some moment in time (i.e., not "right now"), data found in the warehouse is said to be "time variant". Figure 4 shows the time variancy of data warehouse data.

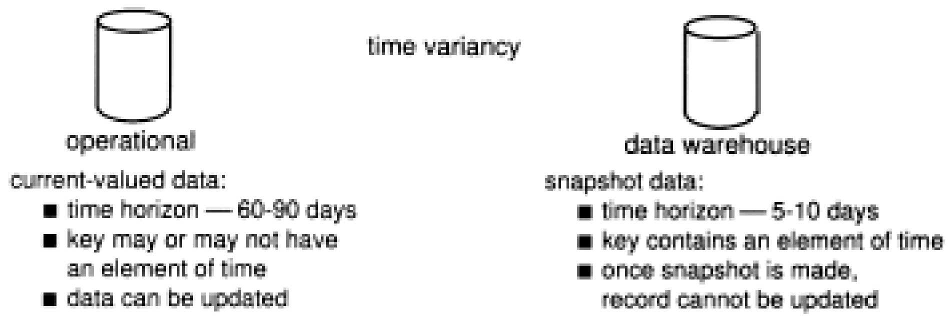

time variancy
operational
current-valued data:
■ time horizon — 60-90 days
■ key may or may not have an element of time
■ data can be updated
data warehouse
snapshot data:
■ time horizon — 5-10 days
■ key contains an element of time
■ once snapshot is made, record cannot be updated

Figure 4

The time variancy of data warehouse data shows up in several ways. The simplest way is that data warehouse data represents data over a long time horizon - from five to ten years. The time horizon represented for the operational environment is much shorter - from the current values of today up to sixty to ninety days. Applications that must perform well and must be available for transaction processing must carry the minimum amount of data if they are to have any degree of flexibility at all. Therefore operational applications have a short time horizon, as a matter of sound application design.

The second way that time variancy shows up in the data warehouse is in the key structure. Every key structure in the data warehouse contains - implicitly or explicitly - an element of time, such as day, week, month, etc. The element of time is almost always at the bottom of the concatenated key found in the data warehouse. On occasions, the element of time will exist implicitly, such as the case where an entire file is duplicated at the end of the month, or the quarter.

The third way that time variancy appears is that data warehouse data, once correctly recorded, cannot be updated. Data warehouse data is, for all practical purposes, a long series of snapshots. Of course if the snapshot of data has been taken incorrectly, then snapshots can be changed. But assuming that snapshots are made properly, they are not altered once made. In some cases it may be unethical or even illegal for the snapshots in the data warehouse to be altered. Operational data, being accurate as of the moment of access, can be updated as the need arises.

NON VOLATILE

The fourth defining characteristic of the data warehouse is that it is non volatile. Figure 5 illustrates this aspect of the data warehouse.

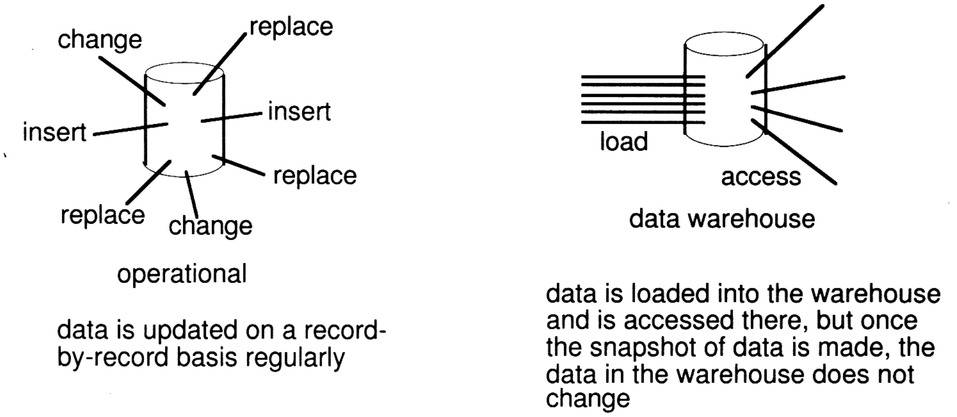

change
replace
insert
insert
replace
replace
change
operational
data is updated on a record-by-record basis regularly
load
access
data warehouse
data is loaded into the warehouse and is accessed there, but once the snapshot of data is made, the data in the warehouse does not change

Figure 5

Figure 5 shows that updates - inserts, deletes, and changes - are done regularly to the operational environment on a record by record basis. But the basic manipulation of data that occurs in the data warehouse is much simpler. There are only two kinds of operations that occur in the data warehouse - the initial loading of data, and the access of data. There is no update of data (in the general sense of update) in the data warehouse as a normal part of processing.

There are some very powerful consequences of this basic difference between operational processing and data warehouse processing. At the design level, the need to be cautious of the update anomaly is no factor in the data warehouse, since update of data is not done. This means that at the physical level of design, liberties can be taken to optimize the access of data, particularly in dealing with the issues of normalization and physical denormalization. Another consequence of the simplicity of data warehouse operation is in the underlying technology used to run the data warehouse environment. Having to support record by record update in an online mode (as is often the case with operational processing) requires the technology to have a very complex foundation underneath a facade of simplicity. The technology supporting backup and recovery, transaction and data integrity, and the detection and remedy of deadlock is quite complex and unnecessary for data warehouse processing.

The characteristics of a data warehouse - subject orientation of design, integration of data within the data warehouse, time variancy, and simplicity of data management - all lead to an environment that is VERY, VERY different from the classical operational environment.

The source of nearly all data warehouse data is the operational environment. It is a temptation to think that there is massive redundancy of data between the two environments. Indeed the first impression many people draw is that of great redundancy of data between the operational environment and the data warehouse environment. Such an understanding is superficial and demonstrates a lack of understanding as to what is occurring in the data warehouse. In fact there is a MINIMUM of redundancy of data between the operational environment and the data warehouse environment.

Consider the following:

- data is filtered as it passes from the operational environment to the data warehouse environment. Much data never passes out of the operational environment. Only that data that is needed for DSS processing finds its way into the data warehouse environment.

- the time horizon of data is very different from one environment to the next. Data in the operational environment is very fresh. Data in the warehouse is much older. From the perspective of time horizons alone, there is very little overlap between the operational and the data warehouse environments.

- the data warehouse contains summary data that is never found in the operational environment.

- data undergoes a fundamental transformation as it passes into the data warehouse. Figure 3 illustrates that most data is significantly altered upon being selected for and moving into the data warehouse. Said another way, most data is physically and radically altered as it moves into the warehouse. It is not the same data that resides in the operational environment from the standpoint of integration.

In light of these factors, data redundancy between the two environments is a rare occurrence, resulting in less than 1% redundancy between the two environments.

THE STRUCTURE OF THE WAREHOUSE

Data warehouses have a distinct structure. There are different levels of summarization and detail that demark the data warehouse. The structure of a data warehouse is shown by Figure 6.

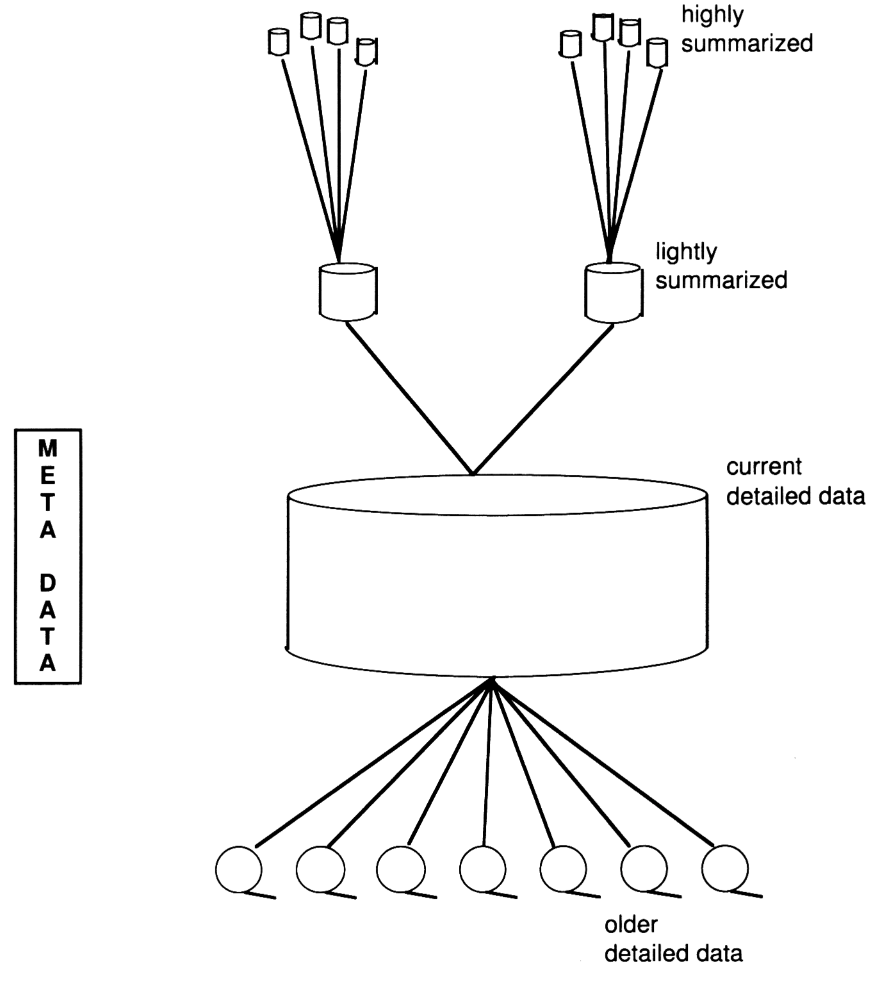


Figure 6: The structure of data inside the data warehouse.

Figure 6 shows that the different components of the data warehouse are:

- metadata,
- current detail data,
- old detail data,
- lightly summarized data, and
- highly summarized data.

Far and away the major concern is the current detail data. It is the major concern because:

- current detail data reflects the most recent happenings, which are always of great interest, and
- current detail data is voluminous because it is stored at the lowest level of granularity, and
- current detail data is almost always stored on disk storage, which is fast to access, but expensive and complex to manage.

Older detail data is data that is stored on some form of mass storage. It is infrequently accessed and is stored at a level of detail consistent with current detailed data. While not mandatory that it be stored on an alternate storage medium, because of the anticipated large volume of data coupled with the infrequent access of the data, the storage medium for older detailed data is usually not disk storage.

Lightly summarized data is data that is distilled from the low level of detail found at the current detailed level. This level of the data warehouse is almost always stored on disk storage. The design issues facing the data architect in building this level of the data warehouse are:

- what unit of time is the summarization done over, and
- what contents - attributes - will the lightly summarized data contain.

The next level of data found in the data warehouse is that of the highly summarized data. Highly summarized data is compact and easily accessible. Sometimes the highly summarized data is found in the data warehouse environment and in other cases the highly summarized data is found outside the immediate walls of the technology that houses the data warehouse. (in any case, the highly summarized data is part of the data warehouse regardless of where the data is physically housed.)

The final component of the data warehouse is that of metadata. In many ways metadata sits in a different dimension than other data warehouse data, because metadata contains no data directly taken from the operational environment. Metadata plays a special and very important role in the data warehouse. Metadata is used as:

- a directory to help the DSS analyst locate the contents of the data warehouse,
- a guide to the mapping of data as the data is transformed from the operational environment to the data warehouse environment,
- a guide to the algorithms used for summarization between the current detailed data and the lightly summarized data and the lightly summarized data and the highly summarized data, etc.

Metadata plays a much more important role in the data warehouse environment than it ever did in the classical operational environment.

In order to bring to life the different levels of data found in the data warehouse, consider the example shown in Figure 7.

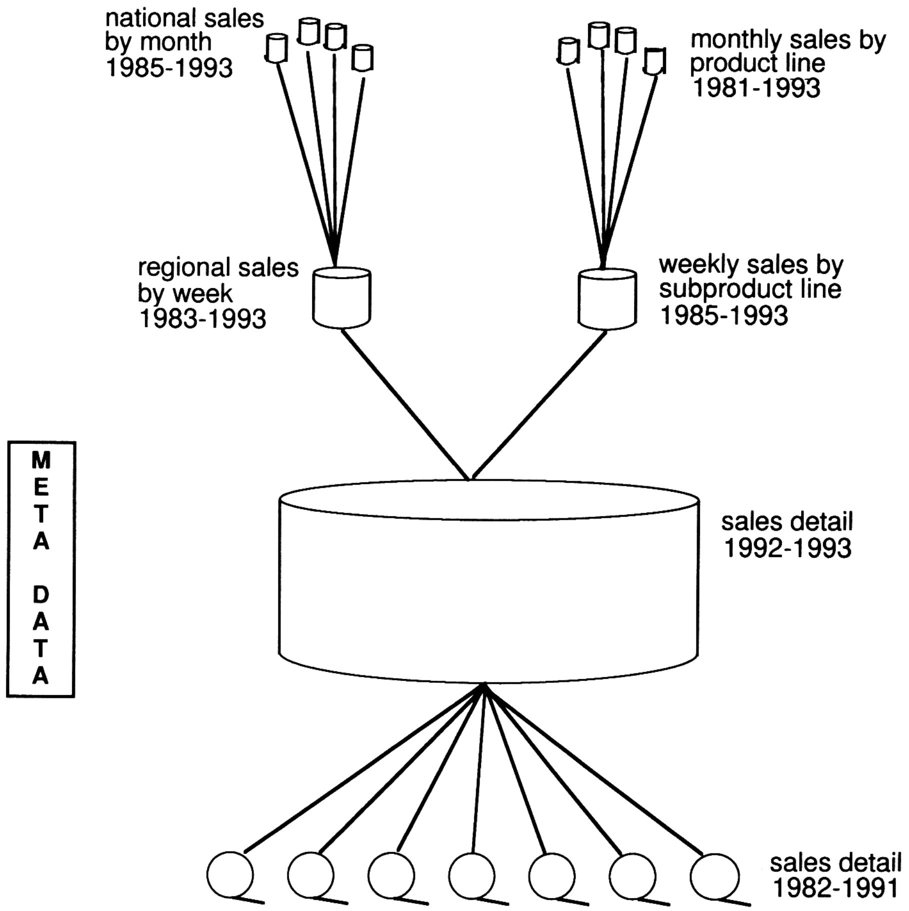


Figure 7: An example of the levels of summarization that might be found in the data warehouse.

In Figure 7 old sales detail is that detail about sales that is older than 1991. All sales detail from 1982 (or whenever the data architect is able to start collecting archival detail) is stored in the old detail level of data.

The current value detail contains data from 1991 to 1992 (assuming that 1992 is the current year.) As a rule sales detail does not find its way into the current level of detail until at least twenty-four hours have passed since the sales information became available to the operational environment. In other words, there was a time lag of at least twenty-four hours between the time the operational environment got news of the sale and the moment when the sales data was entered into the data warehouse.

The sales detail is summarized weekly by sub product line and by region to produce the lightly summarized stores of data.

The weekly sales detail is further summarized monthly along even broader lines to produce the highly summarized data.

Metadata contains (at the least!):

- the structure of the data,
- the algorithms used for summarization,
- the mapping from the operational to the data warehouse, and so forth.

Note that not every summarization ever done gets stored in the data warehouse. There will be many occasions where analysis will be done and one type or the other of summary will be produced. The only type of summarization that is permanently stored in the data warehouse is that data that is frequently used. In other words, if a DSS analyst produces a summarized result that has a very low probability of ever being used again, then that summarization is not stored in the data warehouse.

Old Detail Storage Medium

The symbol shown in Figure 7 for old detail storage medium is that of magnetic tape. Indeed magnetic tape may be used to store that type of data. In fact there are a wide variety of storage media that should be considered for storing old detail data. Figure 8 shows some of those media.

■ photo optical storage
■ RAID
■ micro fiche
■ magnetic tape
■ mass storage

Figure 8: The storage medium for the bulk portion of the data warehouse can be a wide variety of storage types.

Depending on the volume of data, the frequency of access, the cost of the media, and the type of access, it is entirely likely that other storage media will serve the needs at the old level of detail in the data warehouse.

Flow Of Data

There is a normal and predictable flow of data within the data warehouse. Figure 9 shows that flow.

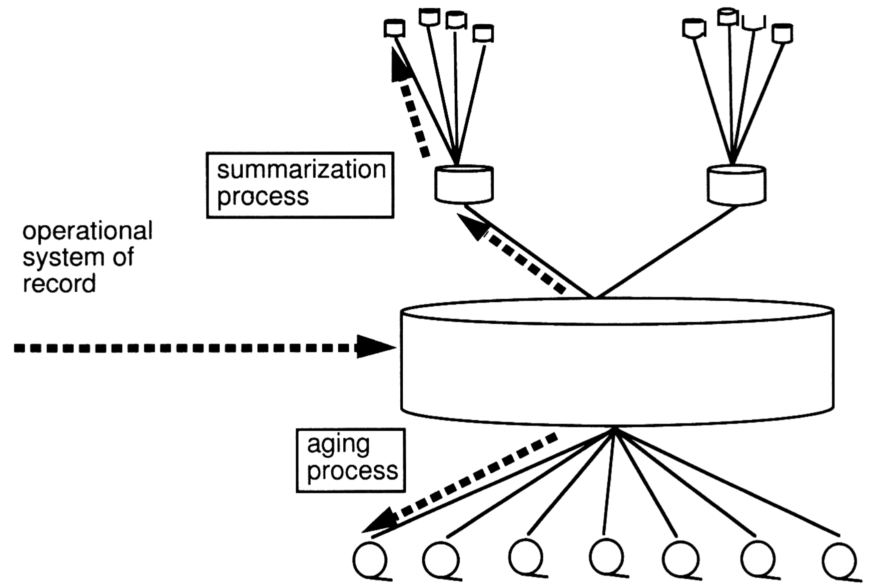


Figure 9: The flow of data inside the data warehouse.

Data enters the data warehouse from the operational environment. (NOTE: there are a few very interesting exceptions to this rule. However, NEARLY ALL data enters the data warehouse from the operational environment.) As data enters the data warehouse from the operational environment, it is transformed, as has been described earlier.

Upon entering the data warehouse, data goes into the current detail level of detail, as shown. It resides there and is used there until one of three events occurs:

- it is purged,
- it is summarized, and/or
- it is archived.

The aging process inside a data warehouse moves current detail data to old detail data, based on the age of data. The summarization process uses the detail of data to calculate the lightly summarized data and the highly summarized levels of data.

There are a few exceptions to the flow as shown (that will be discussed later.) However, in general, for the vast majority of data found inside a data warehouse, the flow of data is as depicted.

USING THE DATA WAREHOUSE

The different levels of data within the data warehouse receive different levels of usage, not surprisingly. As a rule, the higher the level of summarization, the more the data is used, as shown in Figure 10.

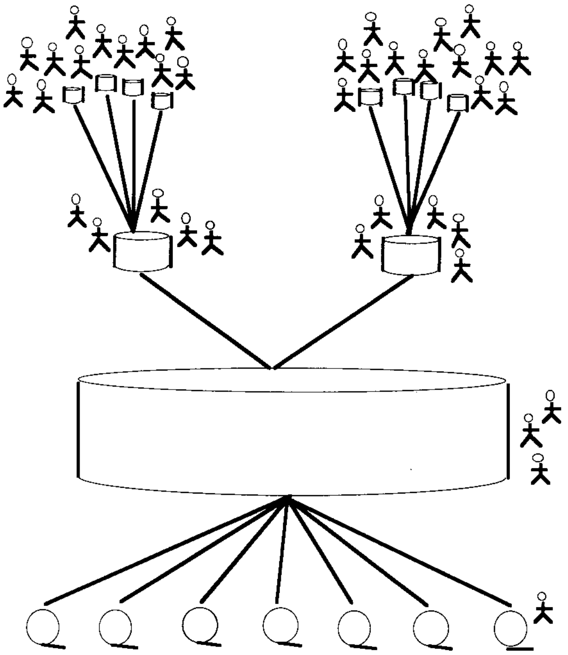


Figure 10: The higher the levels of summarization, the more the usage of the data.

Figure 10 shows that much usage occurs in the highly summarized data, while the old detail data is hardly ever used.

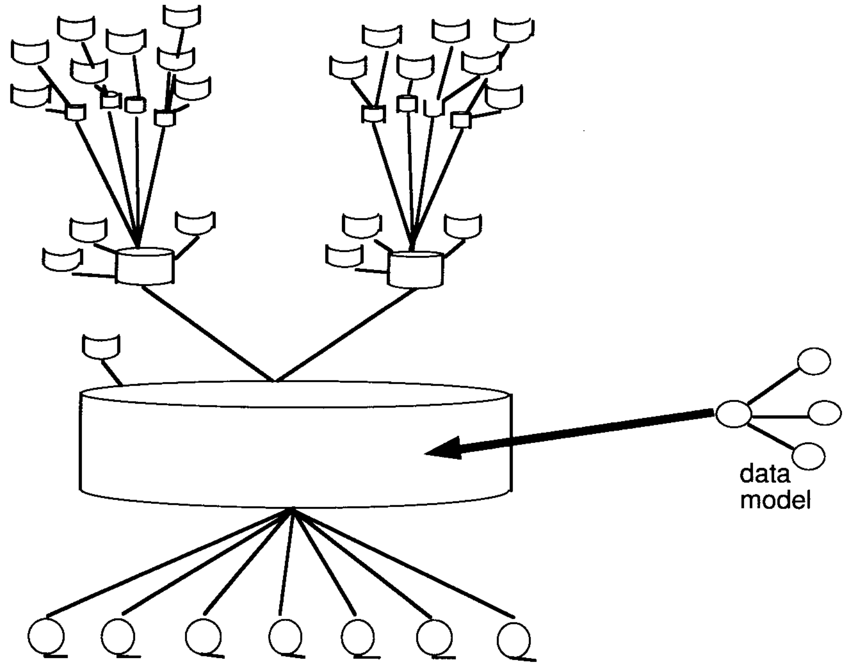
There is a good reason for moving the organization to the paradigm suggested in Figure 10 - resource utilization. The more summarized the data, the quicker and the more efficient it is to get to the data. If a shop finds that it is doing much processing at the detailed levels of the data warehouse, then a correspondingly large amount of machine resources are being consumed. It is in everyone's best interests to do processing at as high a level of summarization as possible.

For many shops, the DSS analyst in a pre data warehouse environment has used data at the detailed level. In many ways getting to detailed data is like a security blanket, even when other levels of summarization are available. One of the tasks of the data architect is to wean the DSS user from constantly using data at the lowest level of detail. There are two motivators at the disposal of the data architect:

- installing a chargeback system, where the end user pays for resources consumed, and
- pointing out that very good response time can be achieved when dealing with data at a high level of summarization, while poor response time results from dealing with data at a low level of detail.

Other Considerations

There are some other considerations of building and administering the data warehouse. Figure 11 shows some of those considerations.



- the more summarized the data is, the more indexes there are
- the data model applies to the current level of detail

Figure 11

The first consideration is that of indexes. Data at the higher levels of summarization can be freely indexed, while data at the lower levels of detail is so voluminous that it can be indexed sparingly. By the same token, data at the high levels of detail can be restructured relatively easily, while the volume of data at the lower levels is so great that data cannot be easily restructured.

Accordingly, the data model and formal design work done that lays the foundation for the data warehouse applies almost exclusively to the current level of detail. In other words, the data modeling activities do not apply to the levels of summarization, in almost every case.

Another structural consideration is that of the partitioning of data warehouse data. Figure 12 shows that current level detail is almost always partitioned.

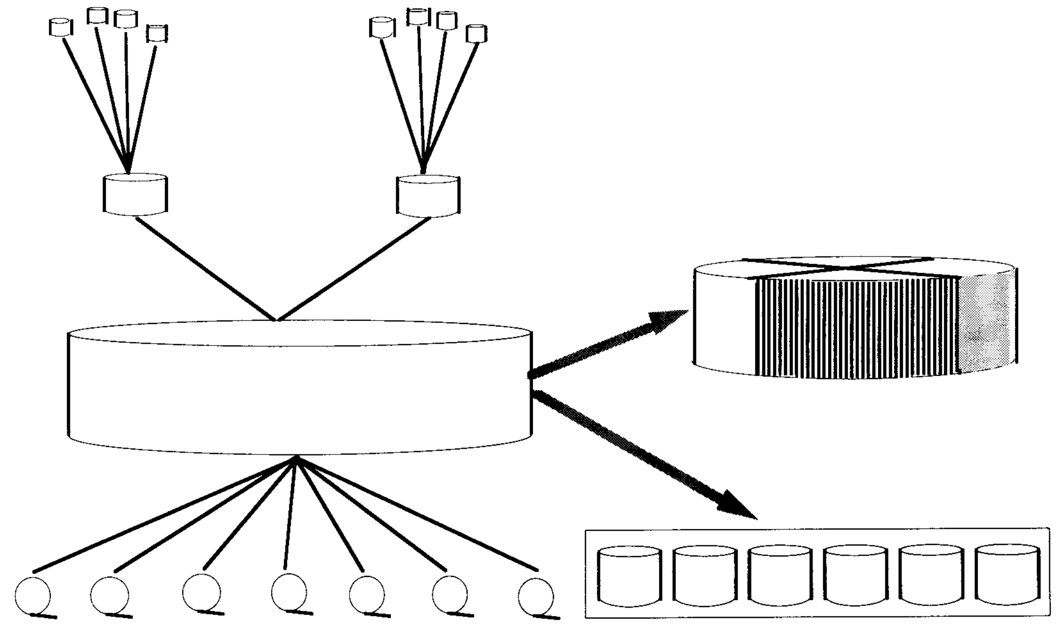


Figure 12: Current detailed data is almost always partitioned.

Figure 12 shows that partitioning can be done in two ways - at the dbms level and at the application level. In dbms partitioning, the dbms is aware of the partitions and manages them accordingly. In the case of application partitioning, only the application programmer is aware of the partitions, and responsibility for the management of the partitions is left up to the programmer.

Under dbms partitioning, much infrastructure work is done automatically. But there is a tremendous degree of inflexibility associated with the automatic management of the partitions. In the case of application partitioning of data warehouse data, much work falls to the programmer, but the end result is much flexibility in the management of data in the data warehouse.

AN EXAMPLE OF A DATA WAREHOUSE

Figure 13 shows a hypothetical example of a data warehouse structured for a manufacturing environment.

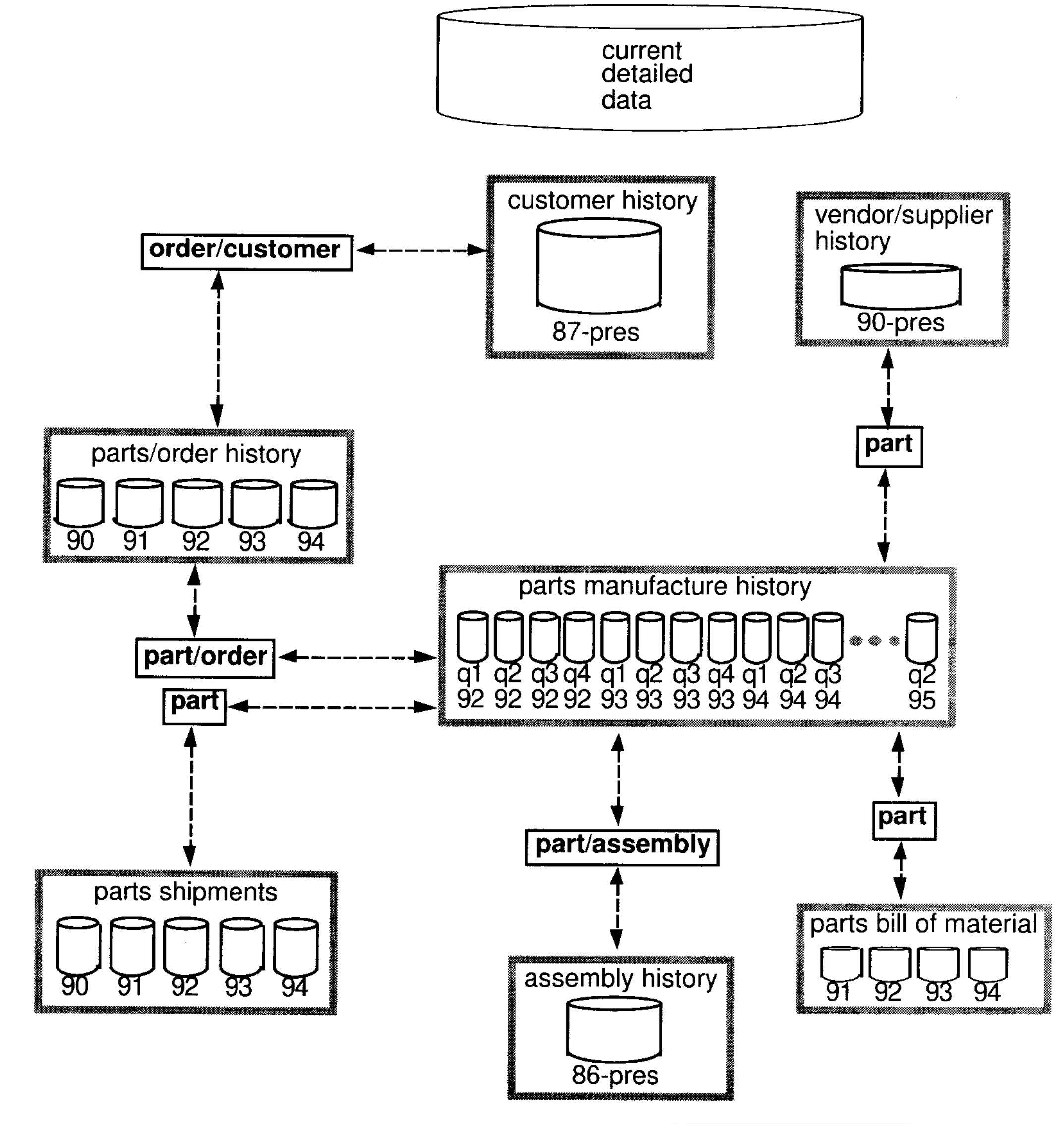


Figure 13: The internal structuring of data in the data warehouse.

Figure 13 shows only current detail data. The levels of summarization are not shown, nor is the old detail archive shown.

Figure 13 shows that there are tables of the same type divided over time. For example, for parts manufacture history, there are many physically separate table, each representing a different quarter. The structure of the data is consistent within the parts manufacture history table, even though there are physically many tables that logically comprise the history.

Note that for different types of tables there are different units of time physically dividing the units of data. Manufacturing history is divided by quarter, part/order history is divided by year, and customer history is a single file, not divided by time at all.

Also note that the different tables are linked by means of a common identifier - either parts, parts/orders, etc. (NOTE: the representation of a relationship in the data warehouse environment takes a very different form than relationships represented in other environments, such as the operational environment. Refer to the Tech Topic on data relationships in the data warehouse for an indepth explanation.)

Other Anomalies

While the data warehouse components work in the fashion described for almost all data, there are a few worthwhile exceptions that need to be discussed. One exception is that of public summary data. Public summary data is summary data that has been calculated outside the boundaries of the data warehouse but is used throughout the corporation. Public summary data is stored and managed in the data warehouse, even though its calculation is well outside the data warehouse. A classical example of public summary data is the quarterly filings made by every public company to the SEC. The accountants work to produce such numbers as quarterly revenue, quarterly expenses, quarterly profit, and so forth. The work done by the accountants is well outside the data warehouse. However, those benchmark numbers produced by the accountants are used widely within the corporation - by marketing, by sales, etc. Once the SEC filing is done, the data is stored in the data warehouse.

Another anomaly not addressed by the discussions in this monograph is that of external data.

Another exceptional type of data sometimes found in a data warehouse is that of permanent detail data. Permanent detail data results in the need of a corporation to store data at a detailed level permanently for ethical or legal reasons. If a corporation is exposing its workers to hazardous substances there is a need for permanent detail data. If a corporation produces a product that involves public safety, such as building airplane parts, there is a need for permanent detail data. If a corporation engages in hazardous contracts, there is a need for permanent detail data, and so forth.

The corporation simply cannot afford to let go of details because in future years, in the case of a lawsuit, a recall, a disputed building flaw, etc. the exposure of the company will be great. Therefore there is a unique type of data known as permanent detail data.

Permanent detail shares many of the same considerations as other data warehouse data, except that:

- the medium the data is stored on must be as safety proof as possible,
- the data must be able to be restored, and
- the data needs special treatment in the indexing of it, otherwise the data may not be accessible even though it has been safely stored.

SUMMARY

A data warehouse is a subject oriented, integrated, time variant, non volatile collection of data in support of management's decision needs. Each of the salient aspects of a data warehouse carries its own implications.

In addition there are four levels of data warehouse data:

- old detail,
- current detail,
- lightly summarized data, and
- highly summarized data.

Metadata is also an important part of the data warehouse environment. Each of the levels of detail carry their own considerations.