① DATA WAREHOUSING - DELIVERY PROCESS

A data warehouse is never static; it evolves as the business expands. As the business evolves, its requirements keep changing and therefore a data warehouse must be designed to ride with these changes. Hence a data warehouse system needs to be flexible

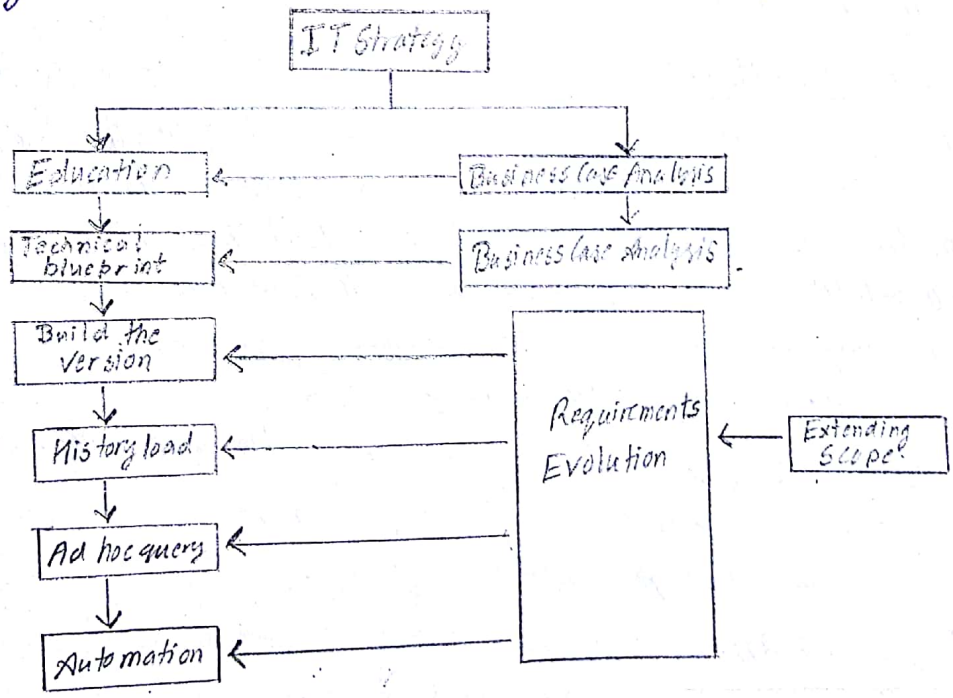
Ideally there should be a delivery process to deliver a data warehouse. However data warehouse projects normally suffer from various issues that make it difficult to complete tasks and deliverables in the strict and ordered fashion demanded by the waterfall method. Most of the times, the requirements are not understood completely. The architectures, designs, and build components can be completed only after gathering and studying all the requirements.

DELIVERY METHOD

The delivery method is a variant of the joint application development approach adopted for the delivery of a data warehouse. We have staged the data warehouse delivery process to minimize risks. The approach that we will discuss here does not reduce the overall delivery time-scales but ensures the business benefits are delivered incrementally through the development process.

★ The delivery process is broken into phases to reduce the project and delivery risk.

The following diagram explains the stages in the delivery process -



IT STRATEGY

Data warehouse are strategic investments that require a business process to generate benefits. IT Strategy is required to procure and retain funding for the project.

BUSINESS CASE

The objective of business case is to estimate business benefits that should be derived from using a data warehouse. These benefits may not be quantifiable but the projected benefits need to be clearly stated. If a data warehouse does not have a clear business case, then the business tends to suffer from credebility problems at some stage during the delivery process. Therefore in data warehouse projects, we need to understand the business case for investment.

EDUCATION AND PROTOTYPING

Organisations experiment with the concept of data analysis and educate themselves on the value of having a data warehouse before settling for a solution. This is addressed by prototyping. It helps in understanding the feasibility and benefits of a data warehouse. The prototyping activity on a small scale can promote educational process as long as-

- The prototype addresses a defined technical objective.
- The prototype can be thrown away after the feasibility concept has been shown.
- The activity addresses a small subset of eventual data content of the data warehouse.
- The activity timescale is non critical.

The following points are to be kept in mind to produce an early release and deliver business benefits-

- Identify the architecture that is capable of evolving.
- Focus on business requirements and technical blueprint phases.
- Limit the scope of the first build phase to the minimum that delivers business benefits.
- Understand the short-term and medium-term requirements of the data warehouse.

BUSINESS REQUIREMENTS

To provide quality deliverables, we should make sure the overall requirements are understood. If we understand the business requirements for both short-term and medium-term, then we can design a solution to fulfil short-term requirements. The short term solution can then be grown to a full solution.

The following aspects are determined in this stage-

- The business rule to be applied on data.
- The logical model for information within the data warehouse.
- The query profiles for the immediate requirement
- The source systems that provide this data.

TECHNICAL BLUEPRINT

This phase need to deliver an overall architecture satisfying the long term requirements. This phase also deliver the components that must be implemented in a short term to derive any business benefit.

The blueprint need to identify the followings:

- The overall system architecture.
- The data retention policy.
- The backup and recovery strategy
- The server and data mart architecture
- The capacity plan for hardware and infrastructure.
- The components of database design.

BUILDING THE VERSION

In this stage, the first production deliverable is produced. This production deliverable is the smallest component of a data warehouse. This smallest component adds business benefits.

HISTORY LOAD

This is the phase where the remainder of the required history is loaded into the data warehouse. In this phase, we do not add new entities, but additional physical tables would probably be created to store increased data volumes.

Let us take an example. Suppose the build version phase has delivered a retail sales analysis data warehouse with 2 months' worth of history. This information will allow the user to analyse only the recent trends and address the short-term issues. The user in this case cannot identify annual seasonal trends. To help him do so, last 2 years' sales history could be loaded from the archive. Now the 40 GB data is extended to 400 GB.

* The backup and recovery procedures may become complex, therefore it is recommended to perform this activity within a separate phase.

AD HOC QUERY

In this phase, we can figure an ad hoc query tool that is used to operate a data warehouse. These tools can generate the database query.

* It is recommended not to use these access tools when the database is being substantially modified.

AUTOMATION

In this phase, operational management processes are fully automated. These would include -

- Transforming the data into a form, suitable for analysis.
- Monitoring query profiles and determining appropriate aggregations to maintain system performance.
- Extracting and loading data from different source systems.

- Generating aggregations from predefined definitions within the data warehouse.
- Backing up, restoring and archiving the data.

EXTENDING SCOPE

In this phase, the data warehouse is extended to address a new set of business requirements. The scope can be extended in two ways:

- By loading additional data into the data warehouse
- By introducing new data marts using the existing information.

* This phase should be performed separately, since it involves substantial efforts and complexity.

REQUIREMENTS EVOLUTION

From the perspective of delivery process, the requirements are always changable. They are not static. The delivery process must support this and allow these changes to be reflected within the system.

This issue is addressed by designing the data warehouse around the use of data within business processes, as opposed to the data requirements of existing queries.

The architecture is designed to change and grow to match the business needs. the process operates as a pseudo-application development process, where the new requirements are continually fed into the development activities and the partial deliverables are produced. These partial deliverables are fed back to the users and then reworked ensuring that the overall system is continually updated to meet the business needs.

② OLAP Types

OLAP systems have been traditionally categorised using the following taxonomy.

1. MOLAP - Multidimensional 2. ROLAP - Relational 3. HOLAP - Hybrid

Multidimensional - MOLAP

- MOLAP (multi-dimentional online analytical processing) is the classic form of OLAP and is sometimes referred to as just OLAP.
- MOLAP stores this data in optimized multidimensional array storage, rather than in a relational database.
- Some MOLAP tools require the pre-computation and storage of derived data, such as consolidations - the operation known as processing.
- Such MOLAP tools generally utilize a pre-calculated data set referred to as a data cube.
- The data cube contains all the possible answers to a given range of questions.
- As a result, they have a very fast response to queries.
- On the other hand, updating can take a long time depending on the degree of pre computation.
- Pre-computation can also lead to, what is known as data explosion.
- The multidimentional data model is an integral part of On-line Analytical Processing or OLAP.
- Because OLAP is on-line, it must provide answers quickly; analysts pose iterative queries during interactive sessions, not in batch jobs that run over night.
- And because OLAP is also analytic, the queries are complex.
- The multidimentional data model is designed to solve complex queries in real time.
- The multidimentional data model is composed of logical cubes, measures, dimentions, hierarchies, levels and attributes.
- The simplicity of the model is inherent because it defines objects that represent real-world business entities.
- Analysts know which business measures they are interested in examining, which dimention and attributes make the data meaningful, and how the dimentions of their business are organised into levels and hierarchies.

- Other MOLAP tools, particularly those that implement the functional database model do not pre-compute derived data but make all calculations on demand, other than those that were previously requested and stored in cache.

Advantages of MOLAP

- Excellent performance: MOLAP cubes are built for fast data retrival, and are optional for slicing and dicing operations due to optimized storage, multidimensional indexing and caching.
- Smaller on-disk size of data compared to data stored in relational database due to compression techniques.
- Effective data extraction achieved through the pre-structuring of aggregated data i.e all calculations have been pre-generated when the cube is created. Hence, complex calculations are not only doable, but then they return quickly.
- Automated computation of higher level aggregates of the data.
- It is very compact for low dimention data sets.
- Array models provide natural indexing.

Disadvantages of MOLAP

- This is usually remedied by doing only incremental processing i.e processing only the data which have changed (usually new data) instead of reprocessing the entire data set.
- Limited in the amount of data it can handle: Because all calculations are performed when the cube is built, it is not possible to include a large amount of data in the cube itself. This is not to say that the data in the cube cannot be derived from a large amount of data. Indeed, this is possible. But in this case, only summary-level information will be included in the cube itself.
- Some MOLAP methodologies introduce data redundancy.
- Requires additional investment: Cube technologies are often proprietary and do not already exists in the organization. Therefore, to adopt MOLAP technology, chances are additional investments in human and capital resources are needed.

Relational-ROLAP

- ROLAP works directly with relational databases.
- The base data and the dimension tables are stored as relational tables and new tables are created to hold the aggregated information.
- It depends on a specialized schema design. The methodology relies on manipulating the data stored in the relational database to give the appearance of traditional OLAP's slicing and dicing functionality.
- In essence, each action of slicing and dicing is equivalent to adding a "WHERE" clause in the SQL statement.
- ROLAP tools do not use pre-calculated data cubes but instead pose the query to the standard relational database and its tables in order to bring back the data required to answer the question
- ROLAP tools feature the ability to ask any question because the methodology does not limit to the contents of a cube.
- ROLAP also has the ability to drill down to the lowest level of detail in the database.

Advantage of ROLAP

- Can handle large amounts of data: The data size limitation of ROLAP technology is the limitation on data size of the underlying relational database. In other words, ROLAP itself places no limitation on data amount.
- Can leverage functionalities inherent in the relational database: Often, relational database already comes with a host of functionalities. ROLAP technologies, since they sit on top of the relational database, can therefore leverage these functionalities.

Disadvantage of ROLAP

- Performance can be slow: Because each ROLAP report is essentially a SQL query (or multiple SQL queries) in the relational database, the query time can be long if the underlying data size is large.
- Limited by SQL functionalities: Because ROLAP technology mainly relies on generating SQL statements to query the relational database,

and SQL statements do not fit all needs (for example, it is difficult to perform complex calculations using SQL).

- ROLAP technologies are therefore traditionally limited by what SQL can do.
- ROLAP vendors have mitigated this risk by building into the tool out-of-the-box complex functions as well as the ability to allow users to define their own functions.

Hybrid - HOLAP

- HOLAP technologies attempt to combine the advantage of MOLAP and ROLAP
- For summary-type information, HOLAP leverages cube technology for faster performance.
- When detail information in needed, HOLAP can drill through from the cube into the underlying relational data.
- There is no clear agreement across the industry as to what constitutes "Hybrid OLAP" except that a database will divide data between relational and specialized storage.
- Sometimes HOLAP database will use relational tables to hold the larger quantities of detailed data, and use specialized storage for at least some aspects of the smaller quantities of more-aggregate or less-detailed data.
- HOLAP addresses the shortcomings of MOLAP and ROLAP by combining the capabilities of both approaches.
- HOLAP tools can utilize both pre-calculated cubes and relational data sources.

Comparison

Each type has certain benefits, although there is disagreement about the specifics of the benefits between providers.

- Some MOLAP implementations are prone to database explosion, a phenomenon causing vast amount of storage space to be used

by MOLAP database when certain common conditions are met: high number of dimentions, pre-calculated results and sparse multidimentional data.

- MOLAP generally delivers better performance due to specialized indexing and storage optimization. MOLAP also needs less storage space compared to ROLAP because the specialized storage typically includes compression techniques.

- ROLAP is more scalable. However, large volume pre-processing is difficult to implement efficiently so it is frequently skipped. ROLAP query performance can therefore suffer tremendously.

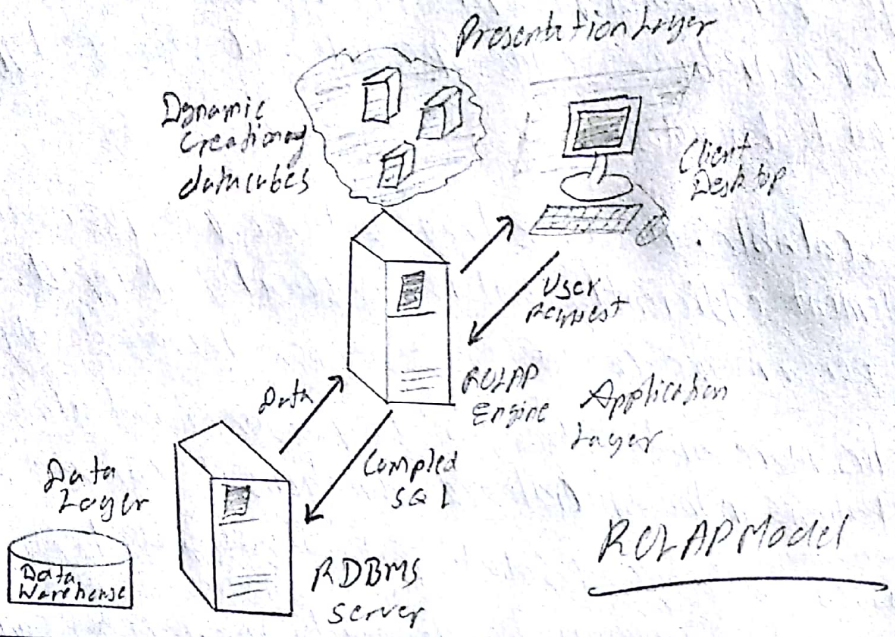
- Since ROLAP relies more on the database to perform calculations, it has more limitations in the specialized functions it can use.

- MOLAP encompasses a range of solutions that attempt to mix the best of ROLAP and MOLAP. It can generally pre-process swiftly, scale well, and offer good function support.

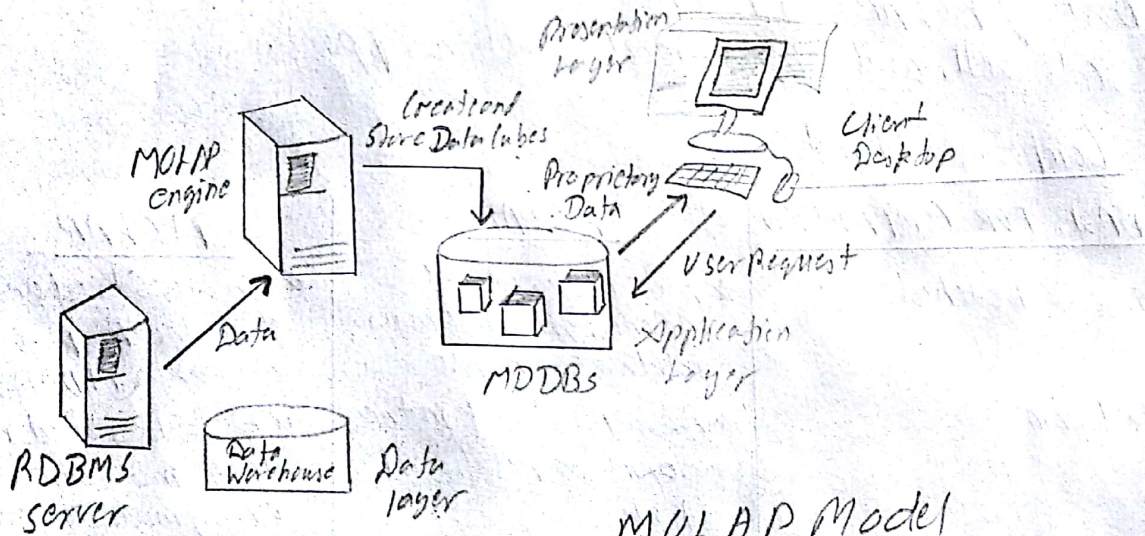
Comparison Chart

BASIS FOR COMPARISON	ROLAP	MOLAP
✱ Storage & Fetched	✱ Data is stored and fetched from the main data warehouse	✱ Data is stored and fetched from the proprietory database MDDBs.
✱ Data Form	✱ Data is stored in the form of relational tables	✱ Data is stored in the large multidimentional array mode as data cubes.
✱ Data Volumes	✱ Large data volumes	✱ Limited summaries data is kept in MDDBs.
✱ Technology	✱ Vs Complex SQL queries to fetch data from the main warehouse	✱ MOLAP engine creates a partitulated and prefabricated data cubes for multidimen-tional data views. · Sparse matrix is used.
✱ View	✱ ROLAP creates a multidim-entional view of data dynamically	✱ MOLAP already stores the static multidimentional view of data in MDDBs.
✱ Access	✱ Slow access	✱ Faster access

	Data Storage	Aggregations storage	Query performance	Latency
MOLAP	Cube	Cube	High	High
NOLAP	Relational database	Cube	Medium	Low (none)
ROLAP	Relational database	Relational database	Low	Low (none)

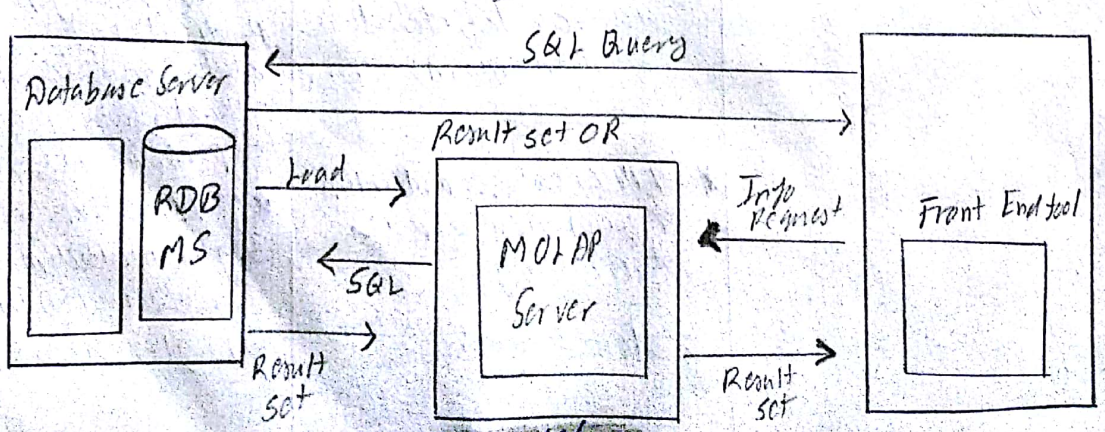


ROLAP Model



MOLAP Model

[HOLAP architecture]



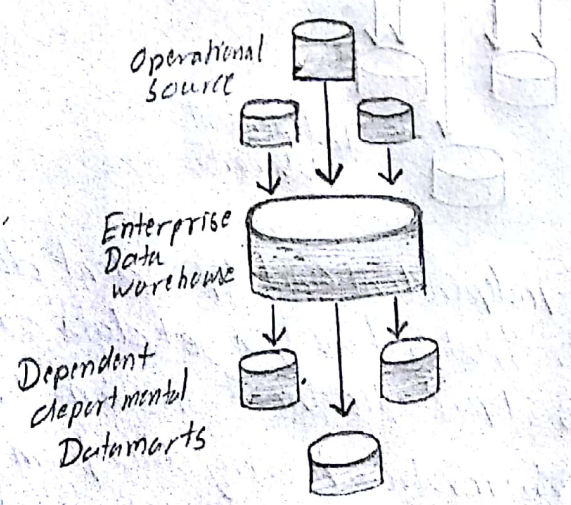
③ Types of Data Mart

1. Dependent: Dependent data marts are created by drawing data directly from operational, external or both sources.
2. Independent: Independent data mart is created without use of a Central data warehouse.

DEPENDENT DATA MART

A dependent dat mart allows sourcing organization's data from a single Data Warehouse. It offers the benefit of centralization. If you need to develop one or more physical data marts, then you need to configure them as dependent data marts.

Dependent data marts can be built in two different ways. Either where a user can access both the data mart and data warehouse, depending on need, or where access is limited only to the data mart. The second approach is not optimal as it produces sometimes referred to as a data junkyard. In the data junkyard, all data begins with a common source, but they are scrapped, and mostly junked.

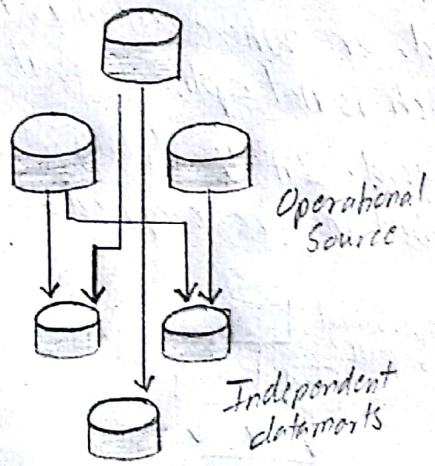


Independent Data Mart

An independent data mart is created without the use of central Data Warehouse. This kind of Data Mart is an ideal option for smaller groups within an organization.

An independent data mart has neither a relationship with the enterprise data warehouse nor with any other data mart. In independent data mart, the data is input separately, and its analysis are also performed autonomously.

Implementation of independent data marts is antithetical to the motivation for building a data warehouse. First of all, you need a consistent, centralised store of enterprise data which can be analysed by multiple users with different interests who want widely varying information.



→ The main difference between independent and dependent data marts is how you populate the data mart, that is, how you get data out of the sources and into the data mart. This step, called the Extraction-Transformation-Transportation (ETT) process, involves moving data from operational systems, filtering it, and loading it into data mart.

→ With dependent data marts, this process is somewhat simplified because formatted and summarized (clean) data has already been loaded into the central data warehouse. The ETT process for dependent data mart is mostly a process of identifying the right subset of data relevant to the chosen data mart subject and moving a copy of it, perhaps in a summarised form.

→ With independent data marts, however, you must deal with all aspects of the ETT process, much as you do with a central data warehouse. The number of sources are likely to be fewer and the amount of data associated with the data mart is less than the warehouse, given you focus on a single subject.

→ The motivations behind the creation of these two types of data marts are also typically different. Dependent data marts are usually built to achieve improved performance and availability, better control, and lower telecommunication costs resulting from local access of data relevant to a specific department. The creation of independent data marts is often driven by the need to have a solution within a shorter time.

Advantages and Disadvantages of a Data Mart

Advantages

→ Data marts contain a subset of organization-wide data. This data is valuable to a specific group of people in an organization.

→ It is cost-effective alternatives to a data warehouse, which can take high costs to build.

→ Data Mart allows faster access of Data.

→ Data Mart is easy to use as it is specifically designed for the needs of its users. Thus a data mart can accelerate business processes.

→ Data Mart needs less implementation time compare to Data Warehouse systems. It is faster to implement Data Mart as you only need to concentrate the only subset of the data.

→ It contains historical data which enables the analyst to determine data trends.

Disadvantages

- Many a times enterprises create too many disparate and unrelated data marts without much benefit. It can become a huge hurdle to maintain.
- Data Mart cannot provide company-wide data analysis as their data set is limited.

(4) How is Data Mart different from a Data Warehouse?

A data warehouse, in contrast to a data mart, deals with multiple subject areas and is typically implemented by a controlling central organisational unit such as the Corporate Information Technology (IT) group. Often it is called a central or enterprise data warehouse. Typically a data warehouse assembles data from multiple source systems.

A data mart is simple form of a data warehouse that is focused on a single subject (or functional area), such as Sales or Finance or Marketing. Data marts are often built and controlled by a single department within an organisation. Given their single-subject focus, data marts usually draw data from only a few sources. The sources could be internal operational systems, a central warehouse, or external data.

Nothing in these basic definitions limits the size of a data mart or the complexity of the decision-support data that it contains. Nevertheless, data marts are typically smaller and less complex than data warehouses; hence, they are typically easier to build and maintain. The following table summarises the basic differences between a data warehouse and a data mart:

	DATA WAREHOUSE	DATA MART
Scope	Corporate	Line-of-Business (LoB)
Subjects	Multiple	Single subject
Data Sources	Many	Few
Size (typical)	100 GB-TB+	< 100 GB
Implementation Time	Months to years	Months