Data Warehouse

A Data Warehouse is a large store of data which is constructed by many sources of data within an organization to develop analysis, decision making and enables management to take decisions.

The term DWH was first coined by Bill Inmon in 1990. Inmon's definition is nothing but the features of DWH.

In an organization which database they keep. They have operational database for their own use. An operational database is nothing but the database which is used for the daily transactions in an organization and acts as a source system for the DWH

Data Warehouse provides us basic generalized and combined view of data present in the database and required for your business. It is nothing but a database kept separate from the operational databases, so that the data is having a record. Hence no frequent updates are done on the DWH. Moreover we can delete, read, update or even insert into the operational databases but in DWH you can only read.

Definition of Data Warehouse

Inmon's definition of Data Warehouse: In 1993, the "father of data warehousing", Bill Inmon, gave this definition of a data warehouse as: A data warehouse is subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management's decision making process.

Data Warehouse Features

- **Subject-Oriented:** Since it provides the information about the subject not about the organizational operations, so it is Subject-Oriented.

- **Integrated:** Since DWH is comprised of different sources of data and then the data is integrated, so it is integrated.

- **Time Variant:** Data collected in DWH is noted by particular time and date, so it is Time – Variant.

- **Non Volatile:** Keeping the history of data is very important, so when new data is inserted then previous data is not deleted.

Functions of Data Warehouse Tools and Utilities

The following are the functions of data warehouse tools and utilities:

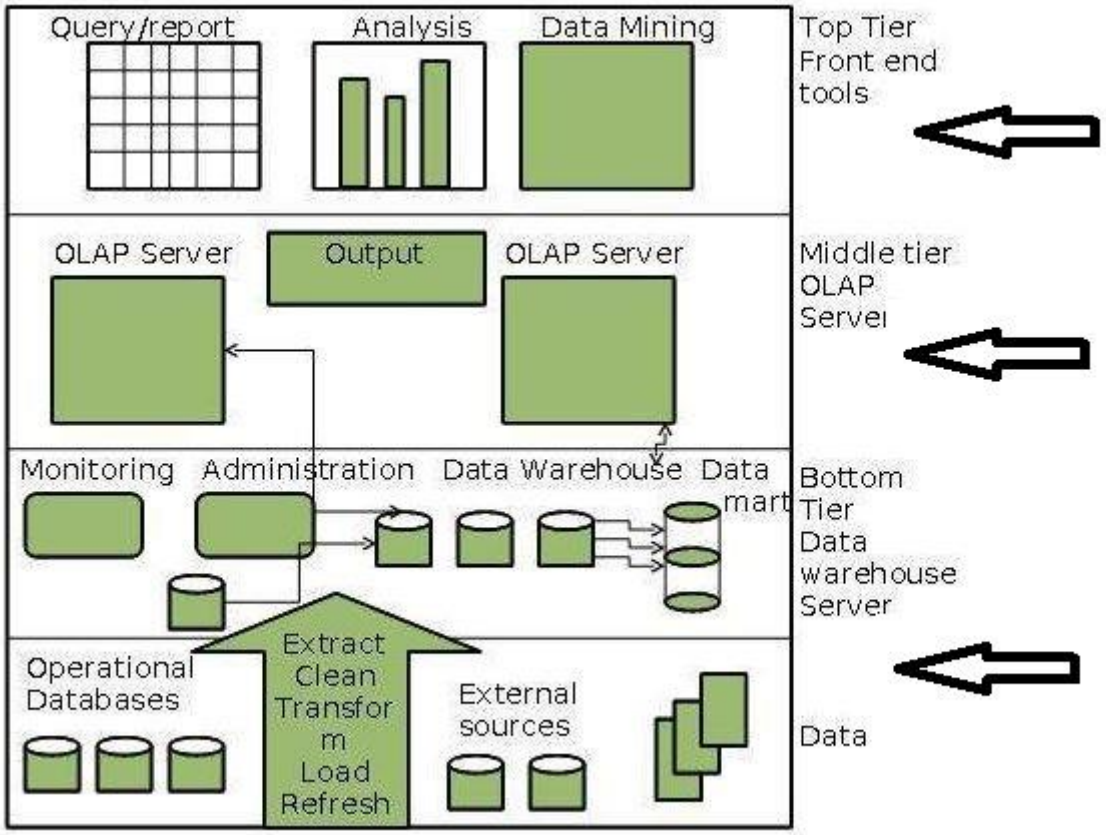
- **Data Extraction** - Involves gathering data from multiple heterogeneous sources.
- **Data Cleaning** - Involves finding and correcting the errors in data.
- **Data Transformation** - Involves converting the data from legacy format to warehouse format.
- **Data Loading** - Involves sorting, summarizing, consolidating, checking integrity, and building indices and partitions.
- **Refreshing** - Involves updating from data sources to warehouse.

Importance of Data Warehouse

- Data warehouse is an information system that contains historical and commutative data from single or multiple sources.
- A data warehouse is subject oriented as it offers information regarding subject instead of organization's ongoing operations.
- In Data Warehouse, integration means the establishment of a common unit of measure for all similar data from the different databases
- Data warehouse is also non-volatile means the previous data is not erased when new data is entered in it.
- A Data warehouse is Time-variant as the data in a DW has high shelf life.
- There are 5 main components of a Data warehouse. 1) Database 2) ETL Tools 3) Meta Data 4) Query Tools 5) Data marts
- These are four main categories of query tools 1. Query and reporting, tools 2. Application Development tools, 3. Data mining tools 4. OLAP tools
- The data sourcing, transformation, and migration tools are used for performing all the conversions and summarizations.

In the Data Warehouse Architecture, meta-data plays an important role as it specifies the source, usage, values, and features of data warehouse data.

Data Warehouse Architectures



There are mainly three types of Data warehouse Architectures: -

1. **Single-tier architecture**

The objective of a single layer is to minimize the amount of data stored. This goal is to remove data redundancy. This architecture is not frequently used in practice.

2. **Two-tier architecture**

Two-layer architecture separates physically available sources and data warehouse. This architecture is not expandable and also not supporting a large number of end-users. It also has connectivity problems because of network limitations.

3. **Three-tier architecture**

This is the most widely used architecture.

It consists of the Top, Middle and Bottom Tier.

Bottom Tier: The database of the Datawarehouse servers as the bottom tier. It is usually a relational database system. Data is cleansed, transformed, and loaded into this layer using back-end tools.

Middle Tier: Middle tier consists of OLAP server and it can be implemented in following ways:

- Relational OLAP: It maps the operations on multidimensional databases to standard relational operations.
- Multidimensional OLAP: This model basically focuses on operations on multidimensional databases.

Top Tier: This tier is client tier which deals directly with client, front end – client tier. The tier consists of reporting tools, query tools, analysis tools and data mining tools.

Types of Data Warehouse

Information processing, analytical processing, and data mining are the three types of data warehouse applications that are discussed below:

- **Information Processing** - A data warehouse allows to process the data stored in it. The data can be processed by means of querying, basic statistical analysis, reporting using crosstabs, tables, charts, or graphs.

- **Analytical Processing** - A data warehouse supports analytical processing of the information stored in it. The data can be analyzed by means of basic OLAP operations, including slice-and-dice, drill down, drill up, and pivoting.

- **Data Mining** - Data mining supports knowledge discovery by finding hidden patterns and associations, constructing analytical models, performing classification and prediction. These mining results can be presented using the visualization tools.

Difference between OLAP and OLTP

Sr.No.	Data Warehouse (OLAP)	Operational Database(OLTP)
1	It involves historical processing of information.	It involves day-to-day processing.
2	OLAP systems are used by knowledge workers such as executives, managers, and analysts.	OLTP systems are used by clerks, DBAs, or database professionals.
3	It is used to analyze the business.	It is used to run the business.
4	It focuses on Information out.	It focuses on Data in.
5	It is based on Star Schema, Snowflake Schema, and Fact Constellation Schema.	It is based on Entity Relationship Model.
6	It focuses on Information out.	It is application oriented.
7	It contains historical data.	It contains current data.
8	It provides summarized and consolidated data.	It provides primitive and highly detailed data.
9	It provides summarized and multidimensional view of data.	It provides detailed and flat relational view of data.
10	The number of users is in hundreds.	The number of users is in thousands.
11	The number of records accessed is in millions.	The number of records accessed is in tens.
12	The database size is from 100GB to 100 TB.	The database size is from 100 MB to 100 GB.
13	These are highly flexible.	It provides high performance.

Data Warehouse Usage

Data warehouses and data marts are used in a wide range of applications.

1. Business executives use the data in data warehouses and data marts to perform data analysis and make strategic decisions.
2. In many areas, data warehouses are used as an integral part for enterprise management.
3. The data warehouse is mainly used for generating reports and answering predefined queries.
4. It is used to analyze summarized and detailed data, where the results are presented in the form of reports and charts.
5. Later, the data warehouse is used for strategic purposes, performing multidimensional analysis and sophisticated operations.
6. Finally, the data warehouse may be employed for knowledge discovery and strategic decision making using data mining tools.

In this context, the tools for data warehousing can be categorized into *access and retrieval tools, database reporting tools, data analysis tools, and data mining tools.*

Data Warehouse Applications

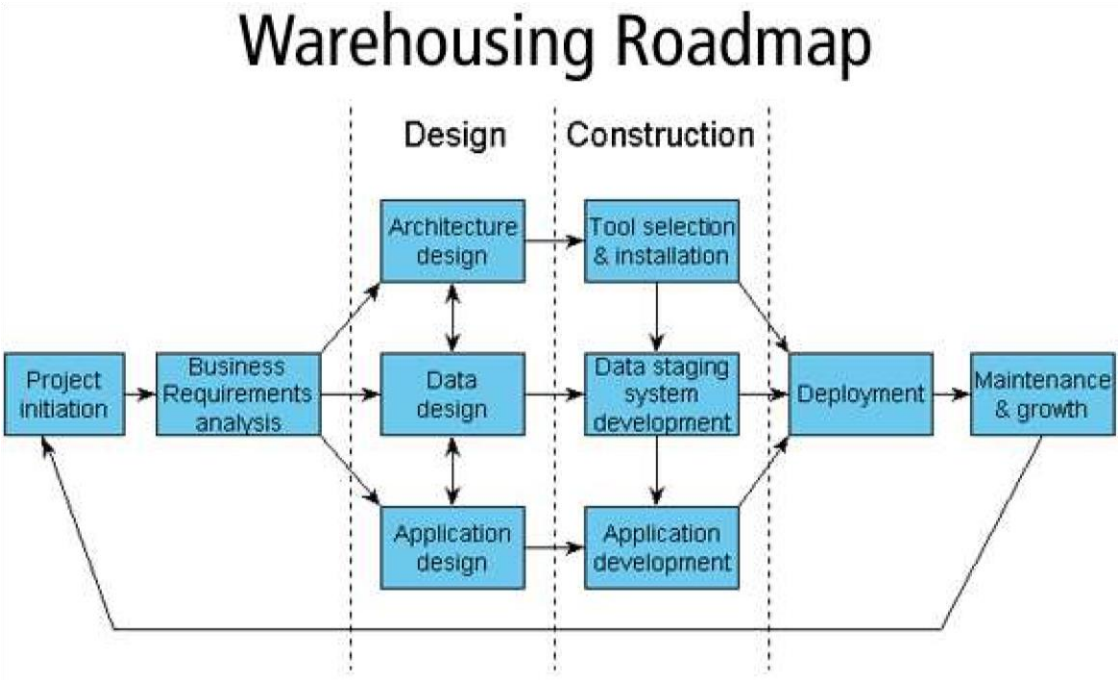
Data warehouses are widely used in the following fields −

- **Banking services:** Most banks also use warehouses to manage the resources available on deck in an effective manner.
- **Financial services:** Similar to the applications seen in banking, mainly revolve around evaluation and trends of customer expenses which aids in maximizing the profits earned by their clients.
- **Consumer goods:** They are used for prediction of consumer trends, inventory management, market and advertising research.
- **Retail sectors:** They use warehouses to track items, their advertising promotions, and the consumers buying trends.
- **Controlled manufacturing:** Data warehouses find themselves to be of use in the service sector for maintenance of financial records, revenue patterns, customer profiling, resource management, and human resources.

Planning Stages of Data warehousing

The key steps in developing a data warehouse can be summarized as follows:

1. Project initiation
2. Requirements analysis
3. Design (architecture, databases and applications)
4. Construction (selecting and installing tools, developing data feeds and building reports)
5. Deployment (release & training)
6. Maintenance



It is advisable to conduct a pilot exercise before embarking on a full-scale development effort. This will include most of the above steps, and provides an opportunity to:

- understand new concepts and processes, and identify potential problems;
- make more realistic plans and manage expectations;
- evaluate alternative tools;
- demonstrate benefits and gain management commitment.

Testing should be an integral part of construction, not a separate step in the development process.

1. Project initiation

No data warehousing project should commence without:

- a clear statement of business objectives and scope;
- a sound business case, including measurable benefits;
- an outline project plan, including estimated costs, timescales and resource requirements;
- high level executive backing, including a commitment to provide the necessary resources;

2. Requirements analysis

Establishing a broad view of the business' requirements should always be the first step. The understanding gained will guide everything that follows, and the details can be filled in for each phase in turn.

Collecting requirements typically involves 4 principal activities:

- Interviewing a number of potential users to find out what they do, the information they need and how they analyze it in order to make decisions. It is often helpful to analyze some of the reports they currently use.
- Interviewing information systems specialists to find out what data are available in potential source systems, and how they are organized.
- Analyzing the requirements to establish those that are feasible given available data.
- Running facilitated workshops that bring representative users and IT staff together to build consensus about what is needed, what is feasible and where to start.

3. Design

The goal of the design process is to define the warehouse components that will need to be built. The architecture, data and application designs are all interrelated, and are normally produced in parallel.

(i). Architecture design

The warehouse architecture describes all the hardware and software components that form the data warehousing environment and explains:

- how the components will work together;
- where they are located (geographically and on what platform);
- who uses them;
- who will build and maintain them.

The architecture needs to be considered at the outset, as this provides a framework for the selection of tools and the detailed design of individual components during the first and subsequent phases of development.

(ii).Data design

This step determines the structure of the primary data stores used in the warehouse environment, based on the outcome of the requirements analysis. It is best to produce a broad outline quickly, and then break the detailed design into phases, each of which usually progresses from logical to physical:

The logical design determines what data are stored in the main data warehouse and any associated functional data marts. There are a number of data modelling techniques that can be used to help.

Once the logical design is established, the next step is to define the physical characteristics of individual data stores (including aggregates) and any associated indexes required to optimize performance.

The data design is critical to further progress, in that it defines the target for the data feeds and provides the source data for all reporting and analysis applications.

(i). Application design

The application design describes the reports and analyses required by a particular group of users, and usually specifies:

- a number of template report layouts;
- how and when these reports will be delivered to users;
- the functional requirements for the user interface.

There may be one or more applications associated with each data mart or phase of development.

4. Construction

Warehouse components are usually developed iteratively and in parallel. That said, the most efficient sequence to begin construction is probably as follows:

a) **Tool selection & installation**

Selecting tools is best carried out as part of a pilot exercise, using a sample of real data. This allows the development team to assess how well competing tools handle problems specific to their organization, and to test system performance before committing to purchase.

The most important choices are the:

- ETL tool
- Database(s) for the warehouse (usually relational) and marts (often multi-dimensional)
- Reporting and analysis tools

Clearly these need to be compatible, and it is worth checking reference sites to make sure they work well together.

It pays to define standards and configure the development, testing and production environments as soon as tools are installed, rather than waiting until development is well underway. Most vendors are willing to provide assistance with these steps, and this is normally well worth the investment.

b) **Data staging system**

This comprises the physical warehouse database, data feeds and any associated data marts and aggregates. The following steps are typical:

- Create target tables in the central warehouse database;
- Request initial and regular extracts from source systems;
- Write procedures to transform extract data ready for loading (optionally creating interim tables in a data staging area);
- Write procedures to load initial data into the warehouse (using a bulk loader);
- Create and populate any data marts;
- Write procedure to load regular updates into the warehouse;
- Develop special procedures for a once-off bulk load of historic data;
- Write validation/exception handling procedures;
- Write archiving & backup procedures;
- Create a provisional set of aggregates;
- Automate all regular procedures;
- Document the whole process.

However thorough the design process, problems with the real data are bound to surface at this stage. Substantial time should be allowed to resolve any issues that arise, establish appropriate data cleansing procedures (preferably within the source systems environment) and to validate all data before they are released for live use.

c) **Application development**

This step can begin once a sample or initial extract has been loaded, but it is usually best to leave the bulk of application development until the underlying data mart (or part of the central warehouse) and associated meta-data (especially object names) are stable.

It is a good idea to involve users in the development of reports and analytic applications, preferably through prototyping, but at least by asking them to carry out acceptance testing. Most modern business intelligence tools do not require programming, so it is possible for non-IT staff to build some of their own reports as well.

5. Deployment

It is too often assumed that the first version of a data warehouse can be rolled out in a matter of weeks, simply by showing all the users how to use the new reporting tools.

In practice, training needs to cover not just the basic use of the tools, but also the data that have been made available, and, more significantly perhaps, the new business processes or different ways of working that are intended. This training usually works best if delivered on a one-to-one basis.

As well as training, planning for deployment needs to cover:

- Installing and configuring desktop PCs - any hardware upgrades or amendments to the 'standard build' need to be organized well in advance;
- Implementing appropriate security measures - to control access to applications and data;
- Setting up a support organization to deal with questions about the tools, the applications and the data. However thoroughly the data were checked and documented prior to publication, users are likely to spot anomalies requiring investigation and to need assistance interpreting the results they obtain from the warehouse and reconciling these with existing reports;
- Providing more advanced tool training later, when users are ready, and assisting potential power users to develop their first few reports.
- If the first users find errors and inconsistencies in the data, don't feel comfortable with the tool or can't be bothered to learn how to use it properly, or won't accept new procedures and responsibilities, all the time spent building the warehouse may ultimately be wasted. The following guidelines will help to reduce these risks:
    - Do not start deployment until the data are ready (available and validated) and the tools and update procedures have been tested;
    - Use a small, representative group to try out the finished system before rolling out, including users with a range of abilities and attitudes;
    - Do not grant system access to users until they have been trained.

6. Maintenance

A data warehouse is not like an OLTP system: development is never finished, but follows an iterative cycle (analyze – build – deploy). Also, once live, a warehousing environment requires substantial effort to keep running. Thus the development team should not anticipate handing over and moving on to other projects, but to spend half of their time on support and maintenance.

The most important activities are:

- Monitoring the realization of expected benefits;
- Providing ongoing support to users (see deployment);
- Training new staff;
- Assisting with the identification and cleansing of dirty data;
- Maintaining both feeds & meta-data as source systems change over time;

- Tuning the warehouse for maximum performance (this includes managing indexes and aggregates according to actual usage);
- Purging dormant data;
- Recording successes and using these to continuously market the warehouse.

In addition, mechanisms need to be established to manage growth, in particular the prioritization of requested enhancements, which often require the addition of further data sources.

Data Warehouse Design Approaches

Designing or Building of a Data Warehouse can be done following either one of the approaches. These approaches are notably known as:

* The Top-Down Approach

* The Bottom-Up Approach

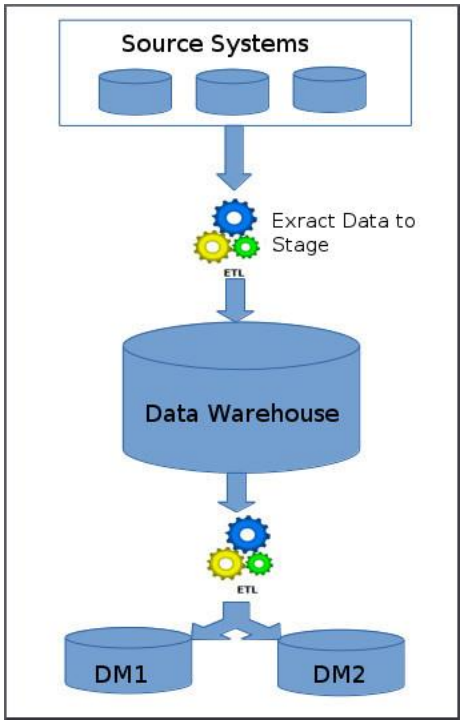
These approaches are defined by the two of the bearers of Data Warehousing namely Ralph Kimball and Bill Inmon.

The Top-Down Approach

This approach was proposed by Bill Inmon, as he stated "**Data warehouse is one part of the overall business intelligence ststem. An enterprise has one data warehouse and data marts source their information from the data warehouse. In the data warehouse, information is stored in 3rd normal form".**

In short Bill Inmon advocated a "dependent data mart structure".

The diagram shows how does the Top-Down model Works.



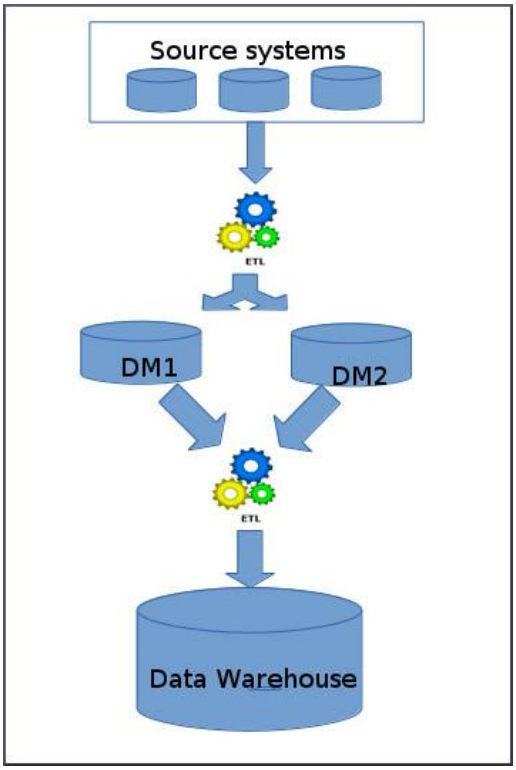
Here are the steps:

* The data is extracted from different/same data sources. This data is loaded into the staging areas and validated and consolidated for ensuring the level of accuracy and then pushed to the Enterprise Data Warehouse (EDW).

* Detailed data is regularly extracted from EDW and is temporarily hosted in staging area for aggregation, summarization and then extracted and loaded into Data Warehouse.

* Once the aggregation and summarization of data is completed the Data marts extract the data into data marts and apply fresh transformations on them. This is done so that the data which comes is in sync with the strutures defined for the data mart.

The Bottom-Up Approach



This approached was proposed by **Ralp Kimball,** stated as **"Data Warehouse is the conglomerate of all data marts within the enterprise. Information is always stored in the dimensional model."**

* Ralp kimball designed the data warehouse with the data marts connected to it with a bus structure.

* The bus structure as you can see above, contained all the common elements that are used by data marts such as conformed dimension, measures etc.

Basically, Kimball model reverses the Inmon model i.e. Data marts are directly loaded with the data from the source systems and then ETL process is used to load in to Data Warehouse.

Here are the steps:

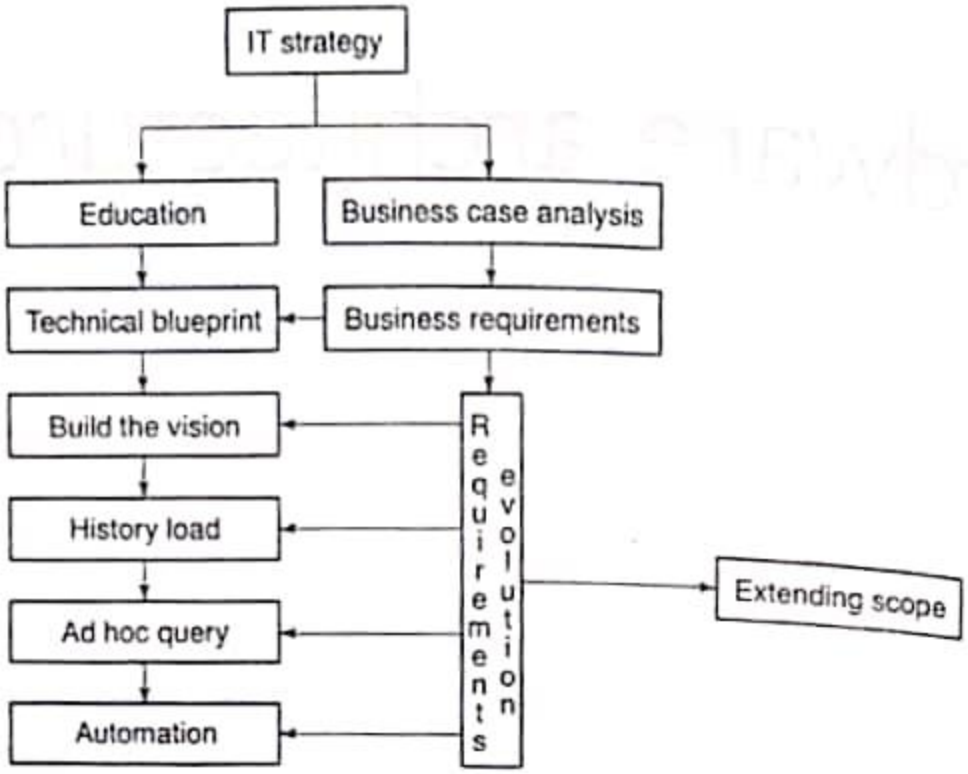
* The data flow in the bottom up approach starts from extraction of data from operational databases into the staging area where it is processed and loaded into the EDW.

* The data in EDW is refreshed or replaced by the fresh data being loaded. After EDW is refreshed the current data is once again extracted in staging area and transformations are applied to fit into the data mart structure. The data is the extracted from Data Mart to the staging area aggregated, summarized and so on loaded into EDW and then made available for the end user for analysis.

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 following diagram explains the stages in the delivery process:



Stages in the 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 credibility 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

Organizations 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:

Things to determine in this stage are following.

- 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 benefit.

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.

Ad hoc Query

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

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.

Requirements Evolution

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