Data Mining and Warehousing

Unit-1

Overview and Concepts

Need For data warehousing: The Need for Data Ware housing is as follows

Data Integration: Even if you are a small Credit Union, I bet your enterprise data flows through and lives in

A variety of in-house and external systems. You want to ask questions that represent those

Slices of key information (referred to as *Key Performance Indicators* or KPIs) such as -

What is the member profitability or member value attrition? Oh, by the way, you want to
Be able to analyze it across all products by location, time and channel. You realize that all
The required data is probably there but not *integrated* and organized in a way for you to
Get the answers easily.
Perhaps your IT staff has been providing the reports you need every time through a series
Of manual and automated steps of stripping or extracted the data from one source, sorting/ merging with data from other sources, manually scrubbing and enriching the data and then running reports against it.

You wonder there ought to be a better and reliable way of doing this.Data Warehouse serves not only as a repository for historical data but also as an excellent
Data integration platform. The data in the data warehouse is integrated, subject oriented,
Time-variant and non-volatile to enable you to get a 360° view of your organization.

Advanced Reporting & Analysis

The data warehouse is designed specifically to support querying, reporting and analysis
Tasks. The data model is flattened (denormalized) and structured by subject areas to make
It easier for users to get even complex summarized information with a relatively simple
Query and perform multi-dimensional analysis. This has two powerful benefits – multilevel
Trend analysis and end-user empowerment.
Multi-level trend analysis provides the ability to analyze key trends at every level across
Several different dimensions, e.g., Organization, Product, Location, Channel and Time,
And hierarchies within them. Most reporting, data analysis, and visualization tools take
Advantage of the underlying data model to provide powerful capabilities such as drilldown,
Roll-up, drill-across and various ways of slicing and dicing data.
The flattened data model makes it much easier for users to understand the data and write Queries rather than work with potentially several hundreds of tables and write long Queries with complex table joins and clauses.

Knowledge Discovery and Decision Support

Knowledge discovery and data mining (KDD) is the automatic extraction of non-obvious
Hidden knowledge from large volumes of data. For example, *Classification* models could
Be used to classify members into low, medium and high lifetime value. Instead of coming
Up with a one-size-fits-all product, the membership can be divided into different clusters
Based on member profile using *Clustering* models, and products could be customized for
Each cluster. *Affinity groupings* could be used to identify better product bundling Strategies.
These KDD applications use various statistical and data mining techniques and rely on
Subject oriented, summarized, cleansed and “de-noised” data which a well designed data
Warehouse can readily provide.
The data warehouse also enables an *Executive Information System (EIS)*. Executives
Typically could not be expected to sift through several different reports trying to get a Holistic picture of the organization’s performance and make decisions. They need the
KPIs delivered to them.
Some of these KPIs may require cross product or cross departmental analysis, which may
Be too manually intensive, if not impossible, to perform on raw data from operational systems. This is especially relevant to relationship marketing and profitability analysis the data in data

warehouse is already prepared and structured to support this kind of analysis.

Performance

Finally, the performance of transactional systems and query response time make the case
For a data warehouse. The transactional systems are meant to do just that – perform
Transactions efficiently – and hence, are designed to optimize frequent database reads and
Writes. The data warehouse, on the other hand, is designed to optimize frequent complex
Querying and analysis. Some of the ad-hoc queries and interactive analysis, which could
Be performed in few seconds to minutes on a data warehouse could take a heavy toll on
The transactional systems and literally drag their performance down.
Holding historical data in transactional systems for longer period of time could also
Interfere with their performance. Hence, the historical data needs to find its place in the Data warehouse.

<u>Trends in Data Ware Housing:</u>

Industries experience with data warehousing over the last decade has provided important lessons on what works in today's business intelligence (BI) solutions. It is not only these lessons, but also the emerging trends which are also shaping our industry directions in business solutions. As a result, our emerging reference architectures used

in building these enterprise data warehouse solutions are changing to meet business demands.
This evolving reference architecture used in building solutions will be overviewed, followed by the implications of these changes. It is these evolving reference architectures that are putting new demands on the databases that are used in warehousing. An important point is that although many of these concepts are not new, databases are being pushed in new ways which are requiring further technology invention.
With the emergence and evolution of the intranet, as well as more businesses exploiting semi-structured data, the more traditional business models are evolving with respect to such things as data accessibility, delivery, and concurrency. Technology such as XML and web services becomes more critical as databases integrate with web portals and BI tooling. Moreover, additional demands on more broad decision making within enterprises are causing heavy consolidation and non-traditional mixed workloads (heavily mixing OLTP and DSS) beyond what has been conventional in the past. Service level agreements, as well as normal operational characteristics are not the same (e.g., backups). Moreover, in many case consolidation is not an option and or desired.

In such latter cases, the business question still needs to be run. As a result, federation augmentation is also very real in enterprise systems. Query management in a federated environment is still a challenging task. A combination of consolidation and federation augmentation is being seen.
In addition to heavy consolidation and federation augmentation, both real-time (right-time) and active data warehousing systems

are being built. These systems present interesting challenges to traditional maintenance and extract/transformation/load operational procedures. Specifically, in large multi-terabyte systems which are 24x7x365. Queries in such systems that execute over aggregated data (including materialized views) need to be very close in time to a consolidated operational data store (ODS) in the same enterprise data warehouse. The maintenance challenges are pushing the technology. Finally, the closed loop processing in an enterprise-wide solution, allows warehouses to play an even more crucial role. Not only are operational systems creating events, so are data warehouses; they play a crucial active role in an enterprise. One such example of events produced in a warehouse is measures, which may be key business indicators (KPIs) used in business performance monitoring through portals.
In addition to this talk presenting emerging data warehousing reference architectures, trends and directions shaping these enterprise data warehousing installations will be overviewed. In doing so, some key implications to databases will be highlighted. In addition to the database itself, any warehouse solution consists of a solution stack. Implications on the whole stack will be touched upon, including such things as metadata and interoperability via standard interfaces such as XML

Architecture and Infrastructure:

Architectural Components:

Architecture

Operational database layer

The source data for the data warehouse — an organization's enterprise resource planning systems fall into this layer.

Data access layer

the interface between the operational and informational access layer — Tools to extract, transform, load data into the warehouse fall into this layer.

Metadata layer

The data dictionary — this is usually more detailed than an operational system data dictionary. There are dictionaries for the entire warehouse and sometimes dictionaries for the data that can be accessed by a particular reporting and analysis tool.

Informational access layer

The data accessed for reporting and analyzing and the tools for reporting and analyzing data — this is also called the data mart. Business intelligence tools fall into this layer. The Inman-Kimball differences about design methodology, discussed later in this article, have to do with this layer.

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.

Master data management has the aim of conforming data that could be considered "dimensions”.

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 “Immunities”, believe in Bill Inman’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 sale 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. “Make everything as simple as possible, but not simpler” (Albert Einstein). Dimensional structures are easy to understand for business users. This is because of the fact that the structured 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 joints. 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.

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).

<u>Infrastructure and Metadata:</u>

The primary rational for data warehousing is to provide businesses with analytics results from data mining, OLAP, Score carding and reporting. The cost of obtaining front-end analytics are lowered if there is consistent data quality all along the pipeline from data source to analytical reporting

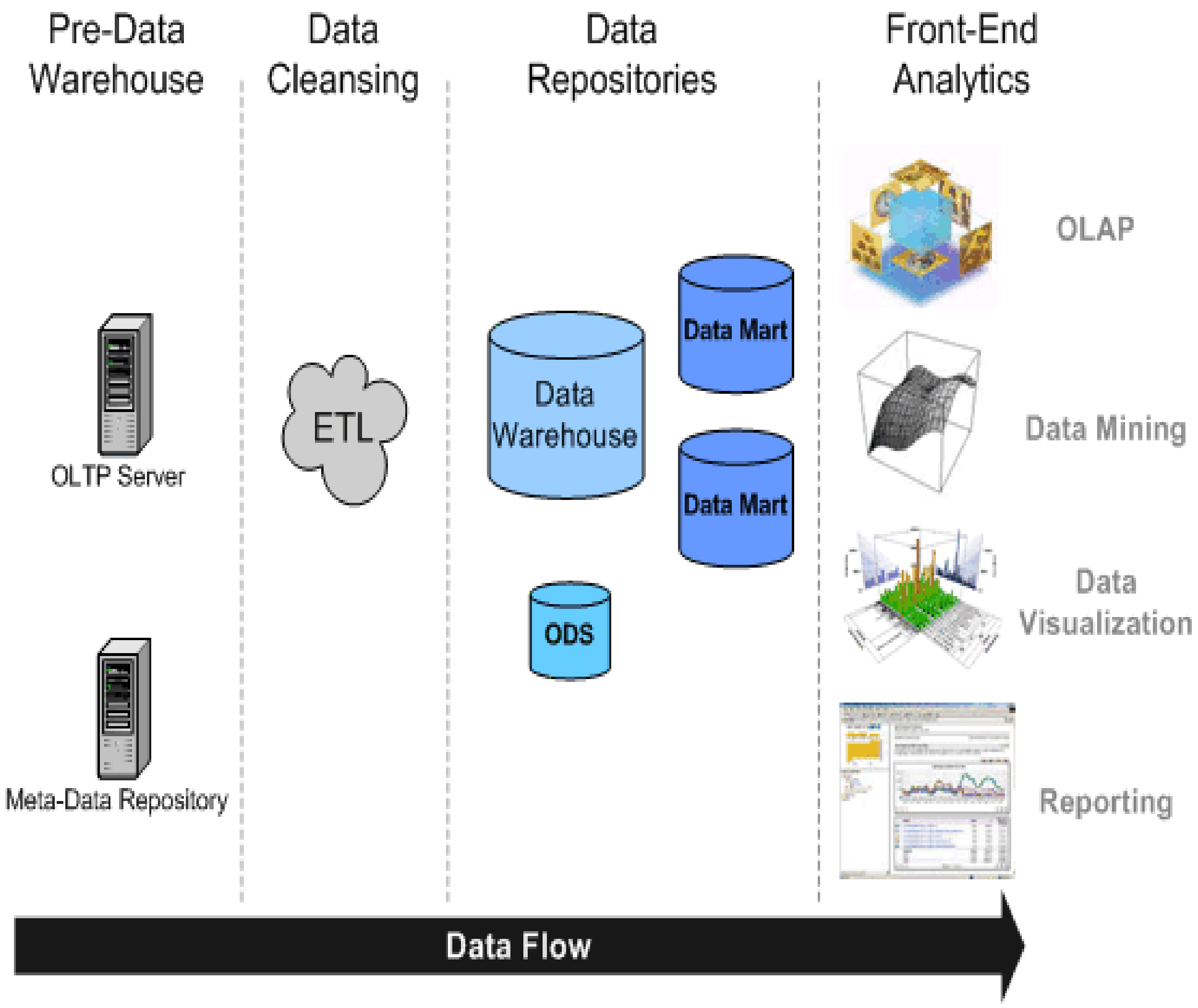


Figure 1. Overview of Data Warehousing Infrastructure

Metadata is about controlling the quality of data entering the data stream. Batch processes can be run to address data degradation or changes to data policy. Metadata policies are enhancing by using metadata repositories. One of the projects we recently worked on was with a major insurance company in North America. The company had amalgamated over the years with acquisitions and also had developed external back-end data integrations to banks and reinsurance partners.

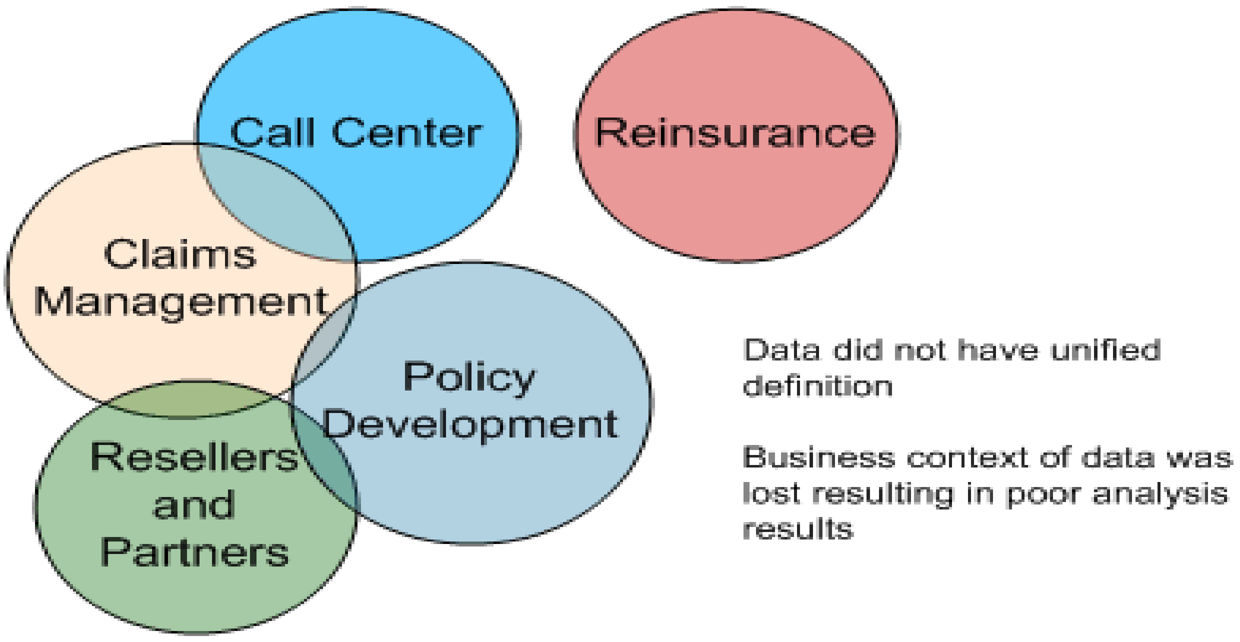

Call Center
Reinsurance
Claims Management
Policy Development
Resellers and Partners
Data did not have unified definition
Business context of data was lost resulting in poor analysis results

Figure 2. Disparate Data Definition Policies in Insurance Company

The client approached *DWreview* as they felt that they were not obtaining sufficient return-on-investments on their data warehouse. Prediction analysis, profit-loss ratio and OLAP reports were labor and time intensive to produce. The publicly listed insurance company was also in the process of implementing a financial Score carding application to monitor compliance with the Sarbanes-Oxley act.

In consultation with the company's IT managers, we analyzed the potential quid-pro-quos of different design changes.

The first step in the process of realignment the data warehousing policies was the examination of the metadata policies and deriving a unified view that can work for all stakeholders. As the company was embarking on a new Score carding initiative it became feasible to bring the departments together and propose a new enterprise-wide metadata policy.

Departments had created their own data marts for generating quick access to reports as they had felt the central data warehouse was not responsive to their needs. This also created a bottleneck as data was not always replicated between the repositories.

With the IT manager's approval and buy-in of departmental managers, a gradual phase in of a company-wide metadata initiative was introduced. Big bang approaches rarely work - and the consequences are extremely high for competitive industries such as insurance.

The metaphor we used for the project was the quote from Julius Cesar by Shakespeare given at the start of the article. We felt that this was a potentially disruptive move but if the challenges were met positively, the rewards would be just.

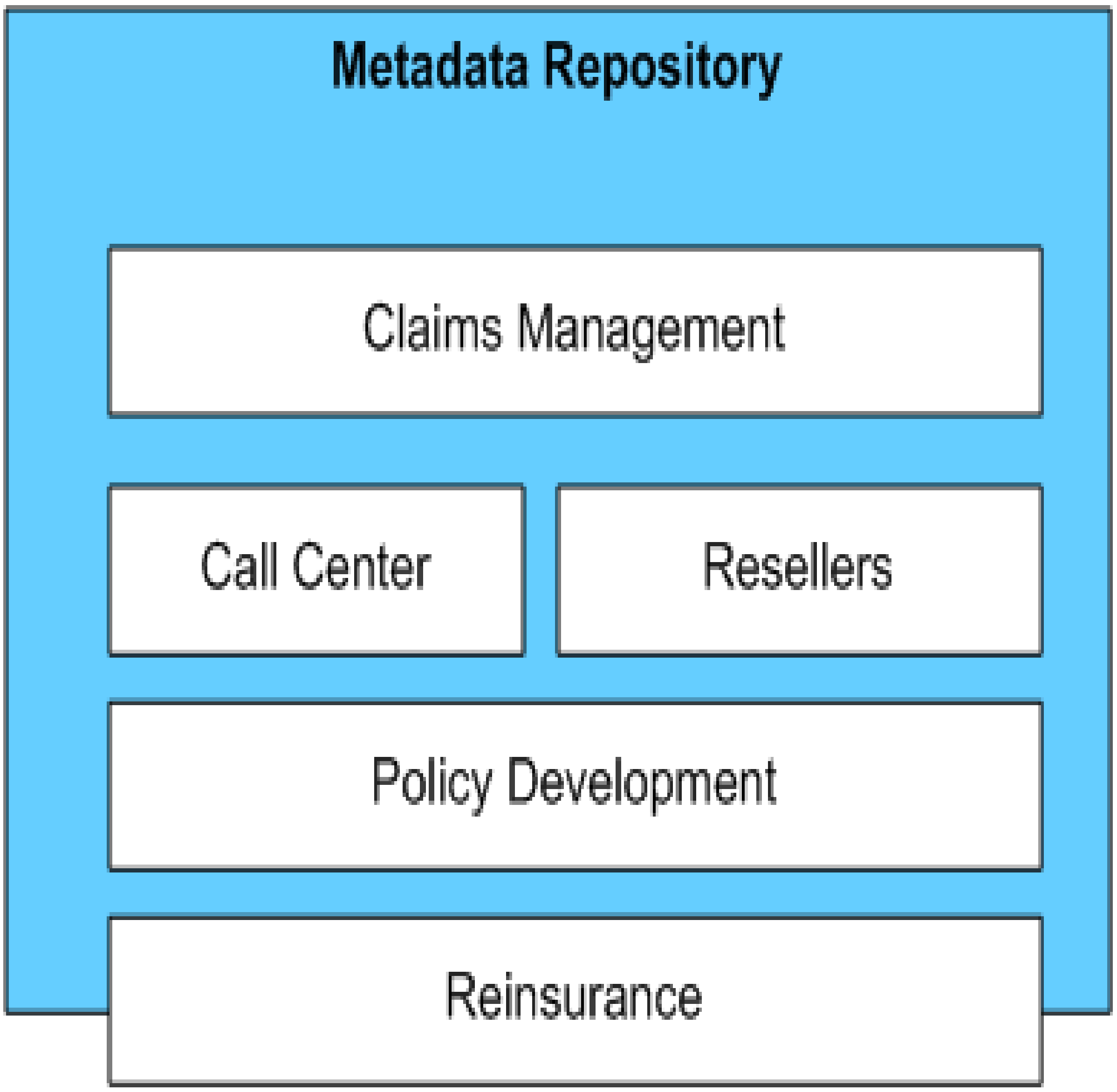


Figure 3. Company-wide Metadata Policy

Industry metadata standards exist in industry verticals such as insurance, banks, manufacturing. OMG's Common Warehouse Metadata Initiative (CWMI) is a vendor back proposal to enable easy interchange of metadata between data warehousing tools and metadata repositories in distributed heterogeneous environments.

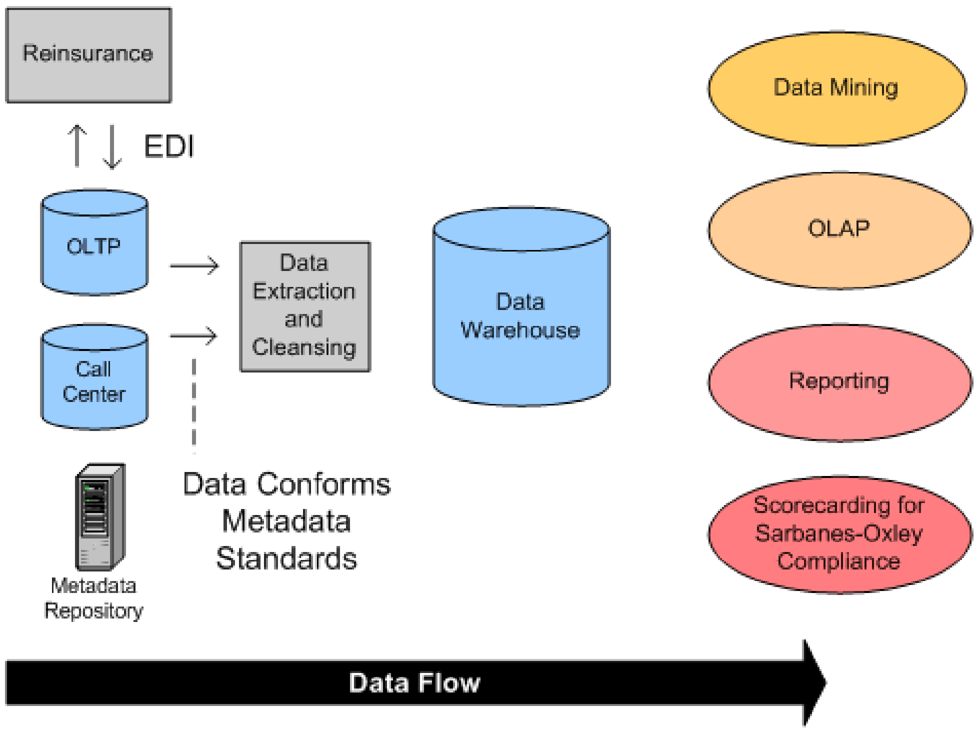

Reinsurance
EDI
OLTP
Call Center
Data Extraction and Cleansing
Data Warehouse
Metadata Repository
Data Conforms Metadata Standards
Data Mining
OLAP
Reporting
Scorecarding for Sarbanes-Oxley Compliance
Data Flow

Fig 4. Partial Schematic Overview of Data Flow after Company-wide Metadata Implementation

In the months since the implementation, the project has been moving along smoothly. There were training seminars given to keep staff abreast on the development and the responses were overwhelmingly positive.

The implementation of the Sarbanes-Oxley Score carding initiative was on time and relatively painless. Many of the challenges that would have been faced without a metadata policy were avoided.

With a unified data source and definition, the company is embarking further on the analysis journey. OLAP reporting is moving across stream with greater access to all employees. Data mining models are now more accurate as the model sets can be scored and trained on larger data sets.

Text mining is being used to evaluate claims examiners comments regarding insurance claims made by customers. The text mining tool was custom developed by *DWreview* for the client's unique requirements. Without metadata policies in place it would be next to impossible to perform coherent text mining. The metadata terminologies used in claims examination were developed in conjunction with insurance partners and brokers. Using the text mining application, the client can now monitor consistency in claims examination, identify trends for potential fraud analysis and provide feedback for insurance policy development.

Developing metadata policies for organizations falls into three project management spheres - generating project support, developing suitable guidelines and setting technical goals. For a successful metadata implementation strong executive backing and support must be obtained.

A tested method for gathering executive sponsorship is first setting departmental metadata standards and evaluating the difference in efficiency. As metadata is abstract in concept a visceral approach can be helpful. It will also help in gaining trust from departments that may be reluctant to hand over metadata policy.

Project Planning and Management:

Planning for the data warehouse: Data warehouse projects that are initiated in the absence of a project plan may suffer from lack of resources. They are usually late, exceed the budget, not of the desired quality and most importantly, they do not give the users what they except.

The main reason behind it is the lack of proper planning and application project management principles .The first question that must be asked is whether the company really needs a data warehouse. We must first assess expectations from the data warehouse and then decide which type of data warehouse is needed. We must have a clear view of where the data is going to come from, who will use the data warehouse, the manner in which the data ware house will be needed.

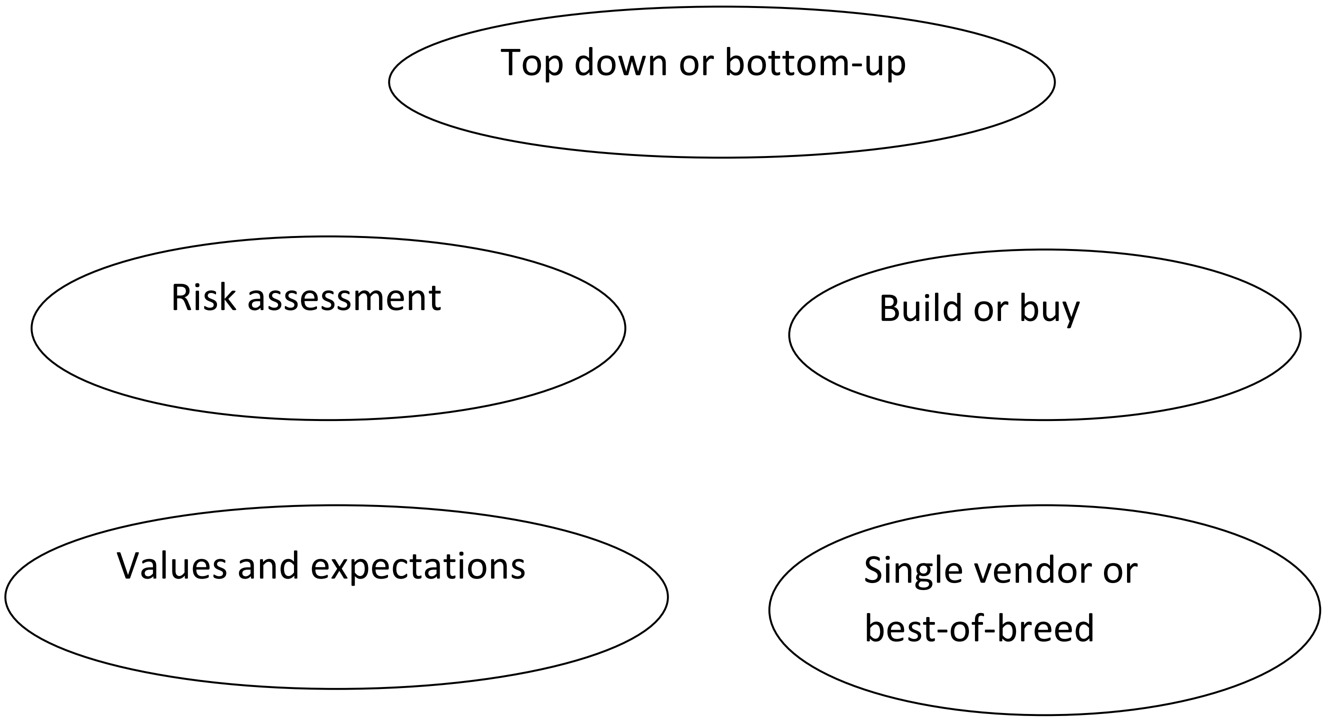


Fig 5.key issues to be considered while planning the data ware house

Values and expectations: It must be determined well in advance as to whether the data warehouse will help the executives and managers to make better decisions or will it be used to improve the bottom line of the business, if yes then by how much these are certain questions that need to be answered.

Assessment of the risk factor involved: The risk faced by the company in case the project fails is much more than simply the loss from the project costs. What are these risks and what losses are likely to be incurred if the project fails are certain issues that will be resolved by a proper mechanism of risk assessment.

Selection of an approach: The top-down approach implies starting at the enterprise –wide data warehouse, although it can be build iteratively. The data from the overall, large enterprise-wide data warehouse flows into departmental and subject data marts.

Build or Buy decision: This is a major issue. No organization can build the entire data warehouse from scratch by-in-house and how much work is to be done in-house and how much work needs to be outsourced. How many data marts must be built-in-house, how many data marts may be built by external organizations?

Single vendor or best-of-breed tools: There are different categories of vendors who offer multiple products that cater to the services and functions rendered by the data warehouse.Basically,any organizations has two main options-first, to use the products of a single vendor or to use the products from more than one vendor.

Project Management for data warehouse: Effective project management is a key to the success of a data warehouse project.

While project management is critical for the developing operational systems, it is even more critical for a data warehouse project, as there is little expertise available in this rapidly growing discipline. The data warehouse project manager is expected to embrace new tasks and deliverables, develop an entirely different strategy to work with the end users and work in an environment that is far less defined than traditional operational systems.

Data warehouse project management deals with project plans, scope agreements, resources, schedules, configuration and change management, contingency plans, and the application of project management tools and methodology.

The difference between a successful data warehouse and a disastrous data warehouse depends on the application of sound project management techniques were applied. It is wrong to believe that since data ware houses are different from operational systems, the project team may ignore the importance of project plans, schedules, and resources, risks, scope agreements, and management control.

Meta data .Meta data is data about data that describes the data warehouse. It is used for building, maintaining, managing and using the data warehouse. Meta data can be classified into:

- Technical meta data, which contains information about warehouse data for use by warehouse designers and administrators when carrying out warehouse development and management tasks.

- Business meta data, which contains information that gives users an easy-to-understand perspective of the information stored in the data warehouse.

Equally important, meta data provides interactive access to users to help understand content and find data. One of the issues dealing with meta data relates to the fact that many data extraction tool capabilities to gather meta data remain fairly immature. Therefore, there is often the need to create a meta data interface for users, which may involve some duplication of effort.

Meta data management is provided via a meta data repository and accompanying software. Meta data repository management software, which typically runs on a workstation, can be used to map the source data to the target database; generate code for data transformations; integrate and transform the data; and control moving data to the warehouse.

As user's interactions with the data warehouse increase, their approaches to reviewing the results of their requests for information can be expected to evolve from relatively simple manual analysis for trends and exceptions to agent-driven initiation of the analysis based on user-defined thresholds. The definition of these thresholds, configuration parameters for the software agents using them, and the information directory indicating where the appropriate sources for the information can be found are all stored in the Meta data repository as well.

How Is a Data Warehouse Really Different from a Normal Database?

An operational data store (ODS), which you may also call an Online Transaction Processing (OLTP) or transactional database has several key features:

- It contains detailed information. For example, it not only contains summary information such as the total amount of an order but also detailed information such as how much each item costs.
- It is designed to process transactions, meaning it's typically dealing with one piece of data at a time: one order, one product, one customer. It may be used to generate basic reports from this data, but it's structure is optimized to support rapid access to small chunks of data.
- The schema is rigid and unchanging.
- It contains up-to-date information, and is updated in real-time.
- The quality of input data is often very high, meaning applications and other elements ensure that correct data goes into the data warehouse.

A data warehouse, in contrast, often includes these features:

- Some data may be summarized, meaning detail is not available. You may be able to tell the total amount of a given order but not the cost of each product contained in the order.

- Its purpose is to drive analysis and decisions. Access is usually for large quantities of data, in order to see trends.
- The schema may be loosely structured and may change over time to support different analysis scenarios.
- The information is historical and may not be entirely up to date. The emphasis is on past data and trends, more so than immediate, real-time data.
- Data is often “cleaned,” meaning errant or “edge” data may be removed to make trends clearer. Data warehouses tend to focus on *sets* of data rather than on individual elements.

Companies with a data warehouse will always have one or more “normal” databases that feed the data warehouse.

Unit2
Data Design and Representation

Principles of dimensional modeling: Dimensional modeling gets its name from the business dimensions incorporate into the logical data model. It is a design technique to structure the business dimensions and the metrics.

This modeling technique is intuitive enough for analyzing the metrics along with the business dimensions. The dimension provides high performance for queries and analysis.

Need for a Dimensional Model: In data warehouse environment, a dimensional model is created so that it can provide the following pieces of information to the data warehouse developers.

- The process which includes the set of all subject areas that are actually the logical structures which have to be designed.
- The level of detail data.
- The facts to be included in the logical structure.
- The time the database should span. This is done by determining how much of archived data must be stored in the data warehouse.

Features of a Good Dimensional Model: The ideal dimension table must exhibit the following features:

- Best data access
- Query centric
- Optimized for query and analysis.

- Depict the way in which the fact table interacts with the dimension table.
- Allow equal interaction of every dimension table with the fact table.
- Enable the users to perform roll up and drill down operations along dimension hierarchies.

Strengths of Dimensional Modeling: The dimensional model offers certain features that the ER model lacks. A dimension model has a standard framework. End-user tools, like report writers and query tools, all use the dimension model to make the user interfaces simple and query processing more efficient.

It has predictable framework of the star schema in which every dimension is equal. All dimensions are symmetrical and have a direct relation with the fact table. Apart from the dimensions, the user interfaces, query strategies and the SQL queries generated against the dimensional model are all symmetrical.

A final strength of the dimensional model is the use of aggregates, which are nothing but summary records that are logically redundant with base data already in the data warehouse, but they are required to enhance query performance.

Data extraction: Data extraction is the act or process of retrieving (binary) data out of (usually unstructured or poorly structured) data sources for further data processing or data storage (data migration). The import into the intermediate extracting system is thus usually followed by data transformation and possibly the addition of metadata prior to export to another stage in the data workflow.[1]

Usually, the term data extraction is applied when (experimental) data is first imported into a computer from primary sources, like measuring or recording devices. Today's electronic devices will usually present a electrical connector (e.g. USB) through which 'raw data' can be streamed into a personal computer.

Typical unstructured data sources include web pages, emails, documents, PDFs, scanned text, mainframe reports, spool files etc. Extracting data from these unstructured sources has grown into a considerable technical challenge where as historically data extraction has had to deal with changes in physical hardware formats, the majority of current data extraction deals with extracting data from these unstructured data sources, and from different software formats. This growing process of data extraction from the web is referred to as Web scraping.

The act of adding structure to unstructured data takes a number of forms

- Using text pattern matching also known as Regular expression to identify small or large-scale structure e.g. records in a report and their associated data from headers and footers;
- Using a table-based approach to identify common sections within a limited domain e.g. in emailed resumes, identifying skills, previous work experience, qualifications etc using a standard set of commonly used headings (these would differ from language to language), e.g. Education might be found under Education/Qualification/Courses;
- Using text analytics to attempt to understand the text and link it to other information.

In this stage, the data flows from the data sources and pauses at the staging area as shown in fig. After transformation and integration, the data is made ready for loading into the data warehouse repository. For majority of the data warehouses, the primary data source consists of the enterprise's operational systems.

Many of the operational systems are still legacy systems, while some of the operational systems run on the client/server architecture and some have ERP data sources, so extracting data from such disparate systems is not a trivial issue.

Apart from extracting the data from these production systems, data from outside sources as well needs to be extracted and for that; temporary files are used to hold the data from these external sources.

Effective data extraction is a key to the success of data warehouses. Therefore, you need to consider the following issues and formulate a data extraction strategy for the success of the data warehouse.

- Identify the applications and systems from which the data will be extracted.
- For each identified data source, determine the method for data extraction, i.e. whether it will be done manually or by using the tools.
- For each data source, determine the extraction frequency, i.e. decide whether the data will be extracted daily, weekly or monthly from the source systems.
- Estimate the acceptable time window for the extraction process from each data source

- Determine the job sequencing feature that is, if the beginning of one extraction job has to wait until the previous one has completed or not

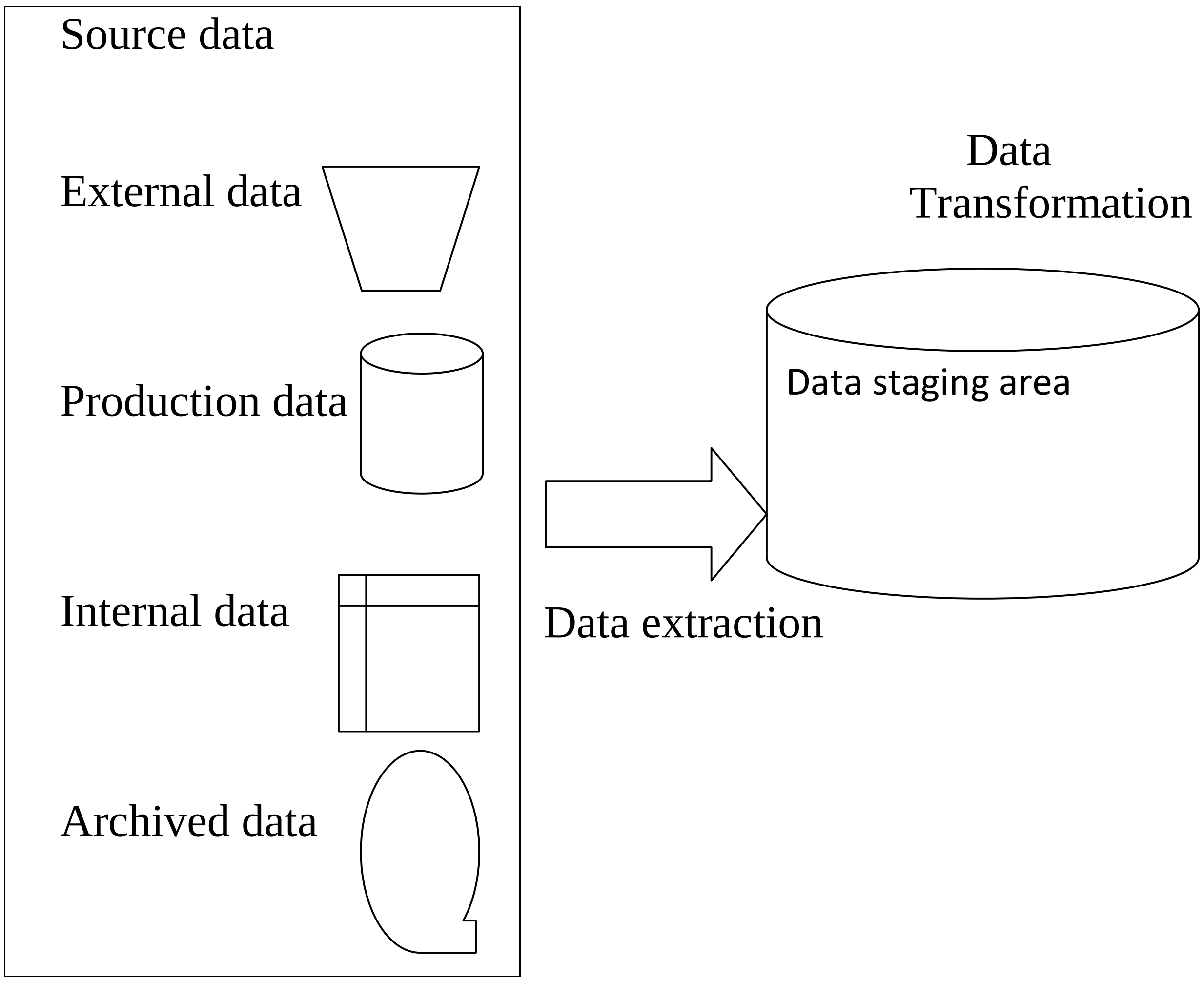


Fig2.1Technical architecture for data extraction

- Determine how the exceptional conditions will be handled like what will be done to handle the input records that could not be extracted accurately.

Data Transformation: The data extracted from the source system cannot be stored directly in the data ware house mainly because of two reasons. First, this is raw data that must be processed to be made usable in the data warehouse. The data in the operational systems is not usable for making strategic decisions. Second, because operational data in those systems may not be good enough for the data warehouse. So before putting this data in the data warehouse, the data needs to be enriched and its quality improved.

Before moving the extracted data into the data warehouse, various kinds of data transformation have to be performed. Since this data come from several dissimilar source systems, there is a need to transform the data according to a standardized format. Also, it must be ensured that the data does not violate any business rules.

Good quality data is of great importance in the data warehouse as this data forms the basics of a sound strategic decision. If the data quality is not good, then the effect of strategic decisions based on incorrect information can be devastating for the organization. Thus, improving the quality of data forms a major task within data transformation process.

1. Tasks Involved in Data Transformation: The different tasks involved in data transformation are as follows:

Format Revision: These revisions include changes to the data types and lengths of individual data fields. For example, in the source systems, the customer's income level may be identified by codes and ranges in which the fields may be text or numeric. Again the length of the customer's name field may vary from one source system to the other.

Decoding of fields: When the data comes from multiple source systems, the same data items may have been described by different field values. The most common example is the coding for gender, with one system using 0 and 1 for male and female, another using M and F, and the other using male, female. Data with cryptic codes must also be decoded before being moved in the data warehouse.

Splitting of fields: Earlier legacy systems stored names and addresses in large text fields. For example all the components of name-first name, middle name, and last name were all stored in one large field called 'Name'. Similarly, city, state, and zip were stored as a single field address. But the need today is to separate out or split these individual components of the name and address fields into separate fields.

Merging of Information: This type of data transformation is neither the opposite of the previous task nor it means bringing together the relevant information from different data sources.

Character set conversion: This type of data transformation is done to the textual data to convert its character set to an agreed standard character set to an agreed standard character set. Some of the legacy systems on the mainframes may have the source data in EBCDIC characters while in other source systems the data may be stored using the ASCII characters while in other source systems the data may be stored using the ASCII character set. So you need to convert the data from one character set to the other.

Conversion of Units: Many companies have global branches. So the sales amount may be represented in different currencies in different source systems. But before moving the data in the data warehouse, you need to convert the figures into a common unit of measurement.

Data and time conversion: The date and time values also need to be represented in a standard format. For example, the American and the British date formats may be standardized to an international format. The date of 1 October 2006 is written as 10/01/2006 in the US format and as 01/10/2006 in the British format. This data needs to be standardized and written in a common format.

Summarization: This type of transformation is done to derive summarized/aggregate data from the most granular data. The summarized data will then be loaded in the data warehouse instead of loading the most granular level of data.

Key Restructuring: While extracting data from the data sources, you have to form the primary keys for the fact tables and the dimensions tables. You cannot keep the primary keys of the source data tables as the primary keys for the fact and dimension tables because the primary keys of the source data have built-in meaning. In fig, the product code has a built-in meaning. If you use the product code as the primary key, then if you place the product in the new department ,then the code of the product will change and hence its primary key

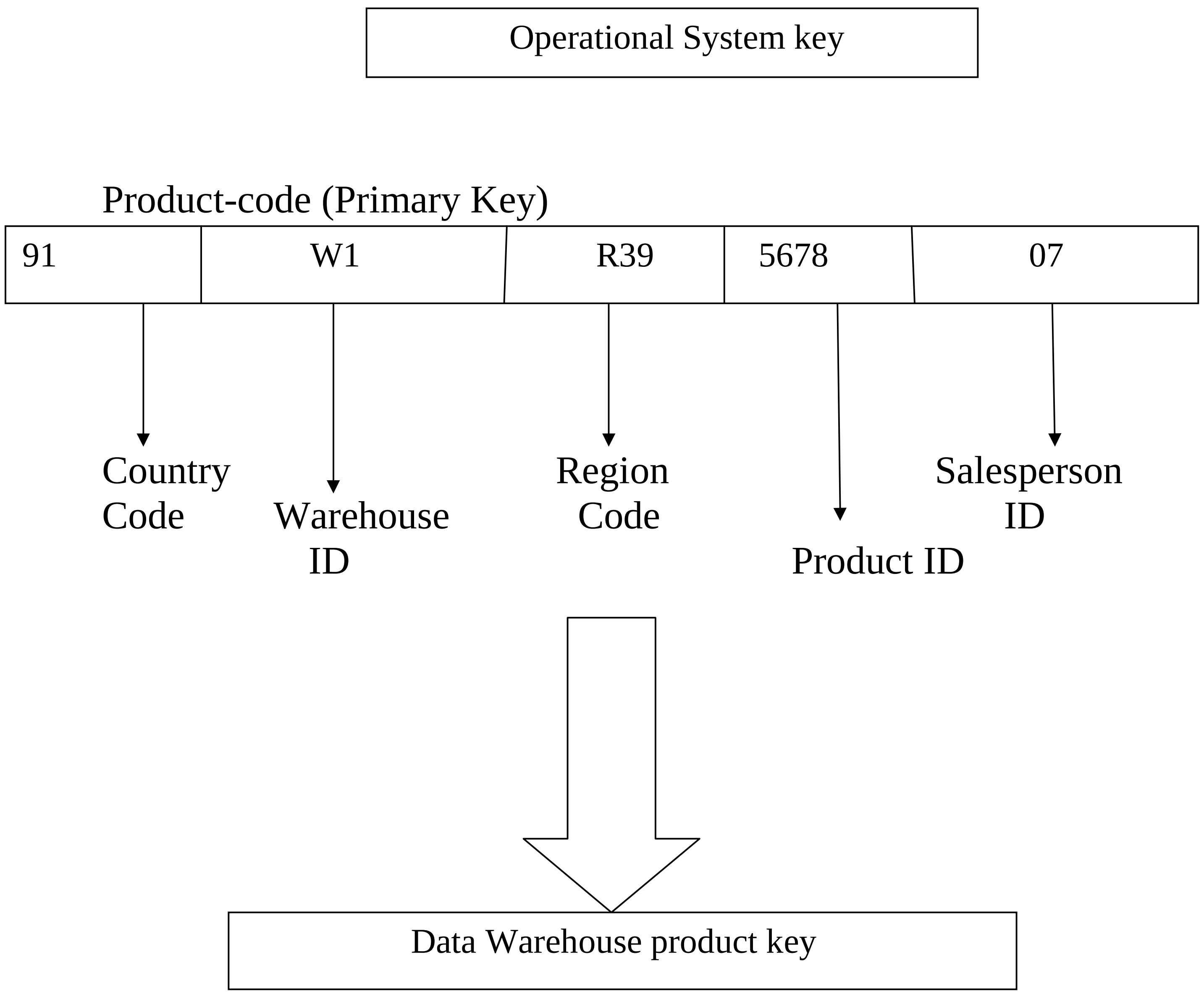


Fig 2.2 Key Structuring

De-duplication: In many companies, the records for the same customer may be stored in many files. When you extract data from the source systems, you have to pay special attention to find such duplicate records and remove the duplicates while storing the record in the data warehouse, ensuring that the information about the customers is stored only once forming a single record.

2. Role of Data Transformation Process:

The data transformation process takes the following course.

- Map the input data from the source systems to data for data warehouse repository
- Clean the data fill all the missing values by some default value.
- Remove duplicate the records so that they may be stored only once in the data warehouse.
- Perform splitting and merging of fields.
- Sort the records.
- De-normalize the extracted data according to the dimensional model of the data warehouse.
- Convert to appropriate data types.
- Perform aggregations and summarizations.
- Inspect the data for referential integrity.

Consolidation and integration of data from multiple source systems

Data Loading: In data loads, the data warehouse has to be offline for the duration of the loading process. So you need to find a time window when the loads may be scheduled without affecting the warehouse users. Therefore, it is better to divide the load process into smaller chunks and populate only a few tables at a time. This technique will render two main benefits – you can run smaller loads in parallel and keep some parts of the data warehouse online while loading the other parts. It is difficult to estimate the time that the loading process will take to complete, especially in the case of initial load and complete refresh.

After the loading process is over, you need to test the loads to verify the correctness of the loads. Procedures need to be provided to handle the data that could not be loaded as a part of the loading process. And also have a plan for quality assurance of the loaded records.

Let us now review the steps involved in completing the initial loading process.

- Drop any indexes built on the data warehouse tables. Index building during the loading process consumes a lot of time. Initial loading may involve large volumes of data containing millions of rows and anything slowing down the load process cannot be afforded.
- In some cases, the initial loading process may take several days to complete. If the initial loading process gets aborted midway, due to some failure in the system or because of any other reason, you will have to redo the entire process. Therefore, it is better to have proper checkpoints so that you can pick up from the latest checkpoint and continue form there.
- Load the dimension tables first and then the fact tables. This is done because the key of the fact table is nothing but a combination of the keys of the dimension tables.

Until the parent row is available; you cannot insert the child rows.

- Once the dimension tables and fact tables have been loaded, create the aggregate tables.
- After the loading of fact tables, dimension tables, and aggregate tables, now it is the time to create indexes on these tables.

Techniques of data loading: The different techniques of loading data are as follows:

Load: If the target table to be loaded already exists and contains some data records, then the load process will wipe out the existing data and store the data from the incoming file. This is illustrated in fig. However if the table is empty, the load process simply applies the data from the incoming file.

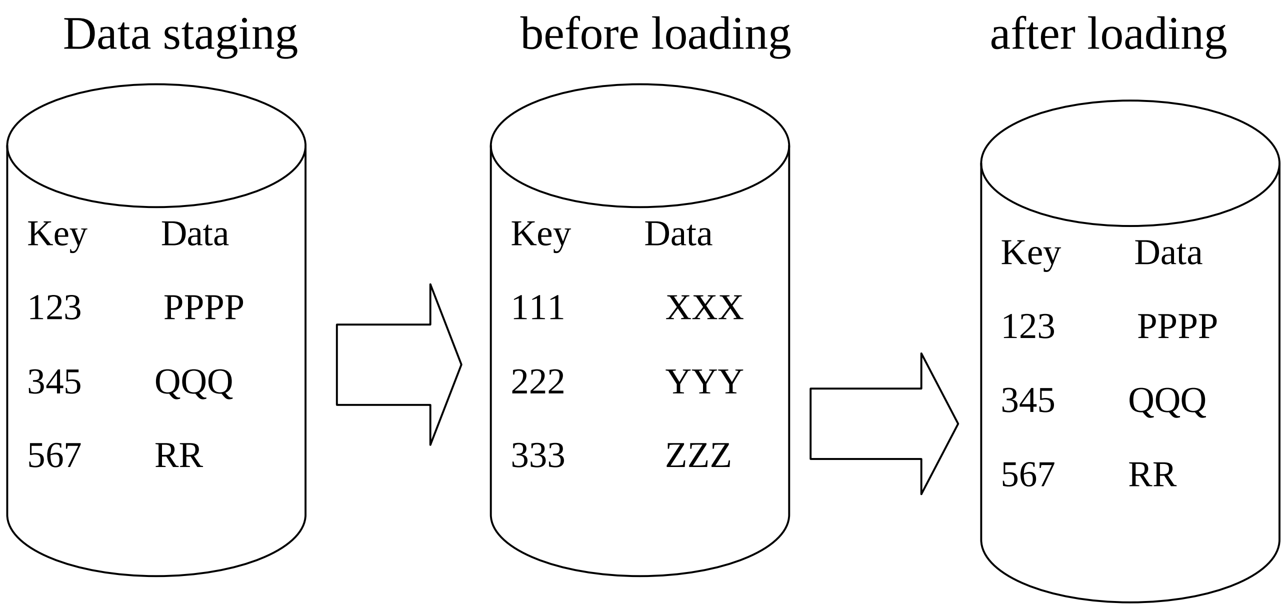


Fig Load mode of data loading

Append: This mode is an extension to the previous one. If data already exists in the table, the append process adds the incoming data thereby preserving the existing data in the target table. This is shown in fig. When the incoming record is a duplicate of an already existing record, then this situation may be handled in any one way. First, you may either allow the incoming data to be duplicate or may be rejected during the append process.

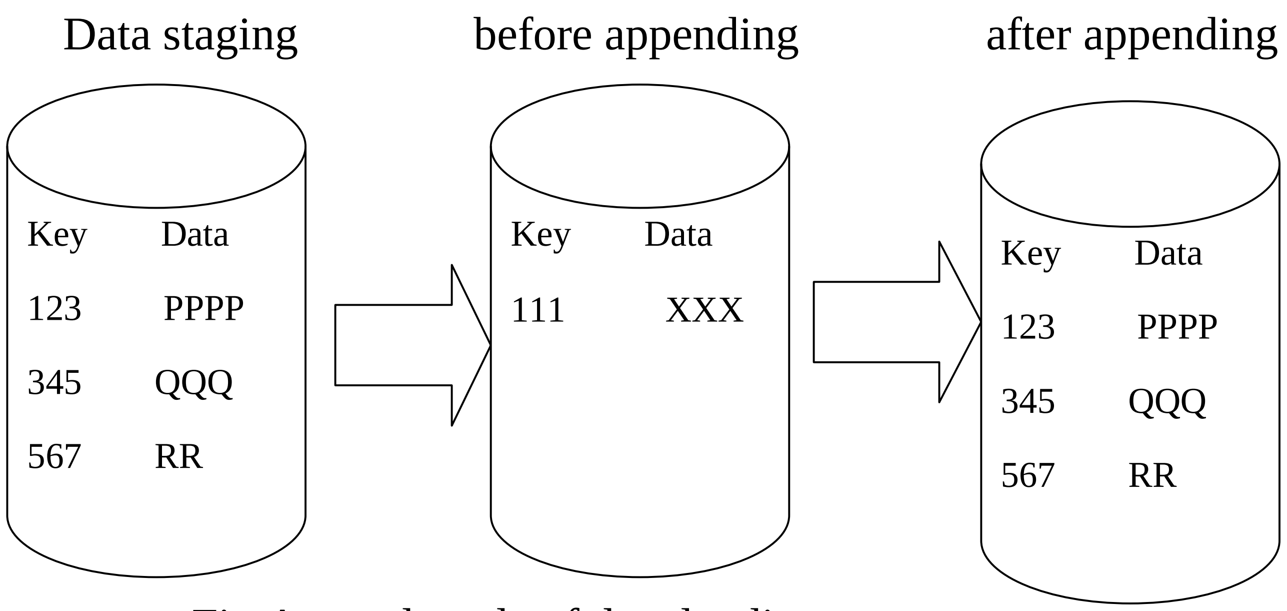


Fig Append mode of data loading

Destructive merge: In this mode, the incoming data is applied to the target data. If the primary key of the incoming record matches with the key of an existing record is updated. And in case the incoming record is a new record without a match with an existing record, the incoming record is added to the target table. Fig explains this process.

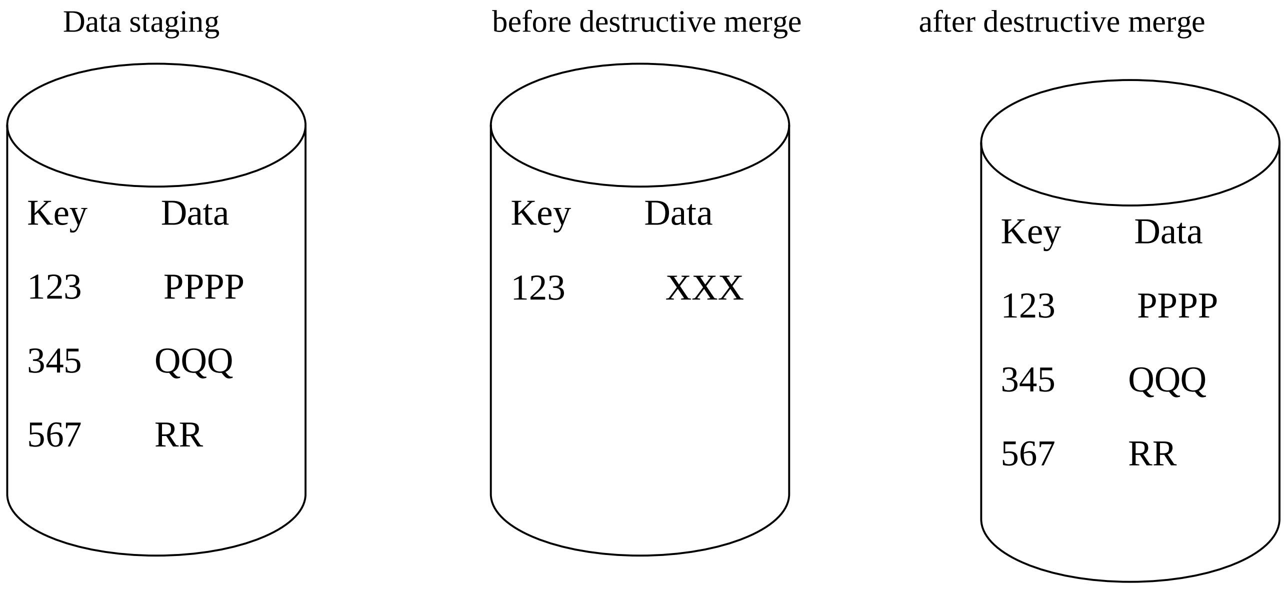


Fig Destructive merge

Constructive merge: In this mode, if the primary key of an incoming record matches with the key on an existing record, then leave the existing record, add the incoming record, and mark the added record as super ceding the old record

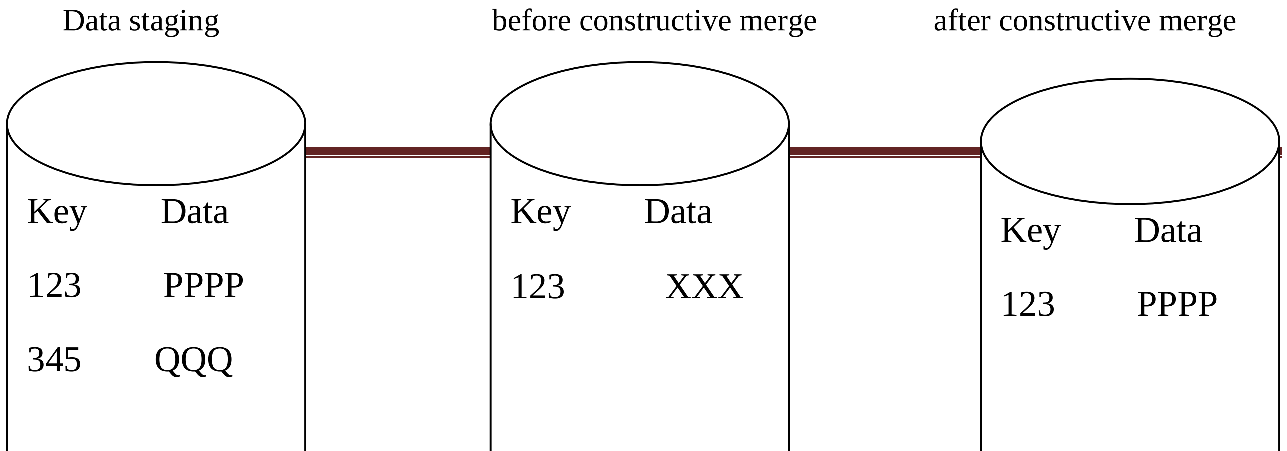
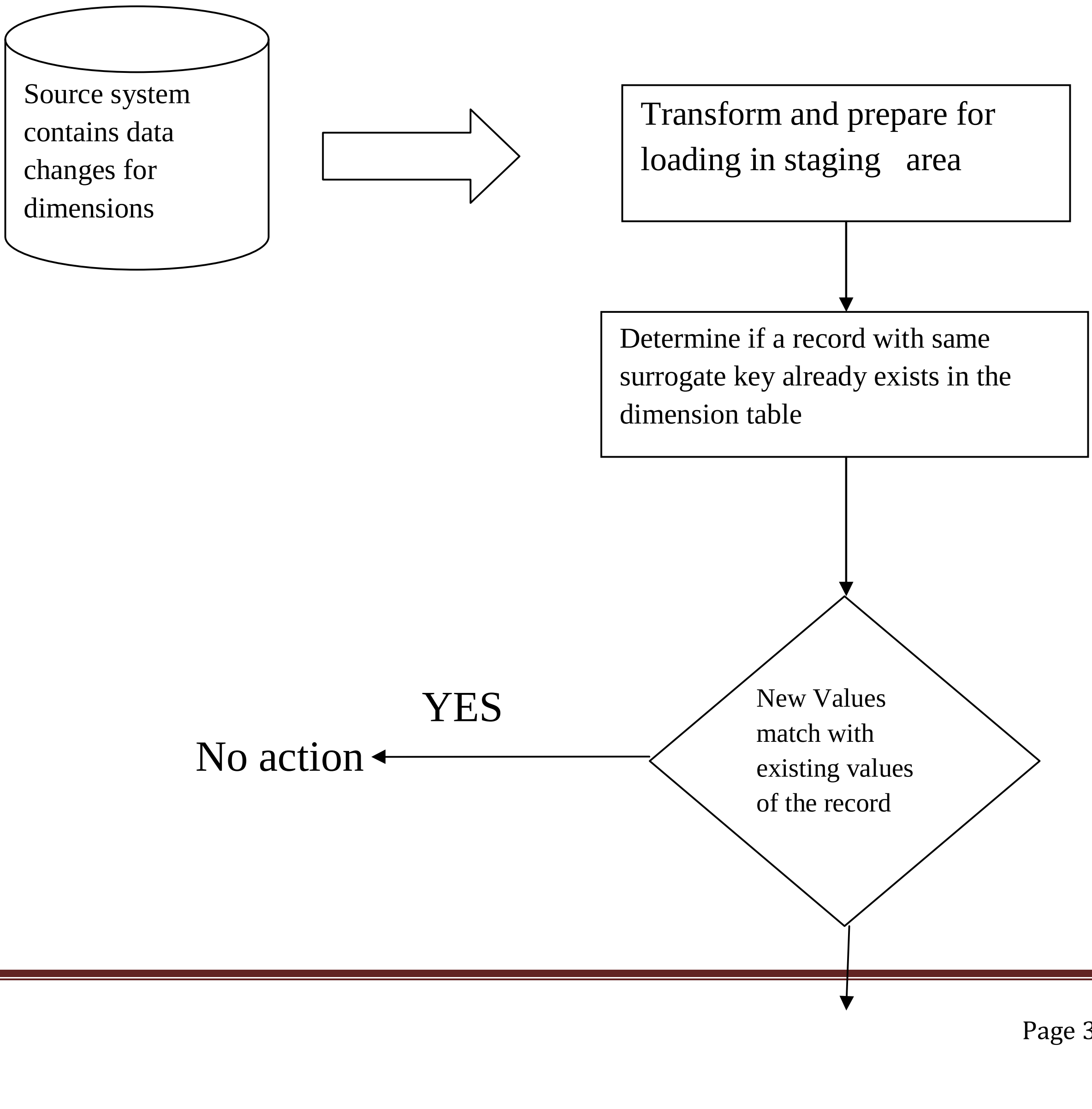


Fig Constructive merge

Loading the fact tables and the dimension tables: In a data warehouse, dimension tables contain attributes that are used to analyze basic measurements such as sales, costs, profits, etc. Customers, salesperson, time, region, product are some common examples of business dimensions. The procedure of maintenance of dimension tables, thereafter, applying the changes on an ongoing basis.

The keys of records in the source data can be applied to the dimension data warehouse. Therefore, before source data can be applied to the dimension tables, whether during initial loading or during updating, the production keys must be converted to system generated keys in the data warehouse. This key conversion must be done as a part of transformation process. The next issue is how to handle Type 1, Type2 and Type3 changes in the data warehouse.

Fig shows how we deal with this issue in the data warehouse.


Source system contains data changes for dimensions
Transform and prepare for loading in staging area
Determine if a record with same surrogate key already exists in the dimension table
New Values match with existing values of the record
YES
No action

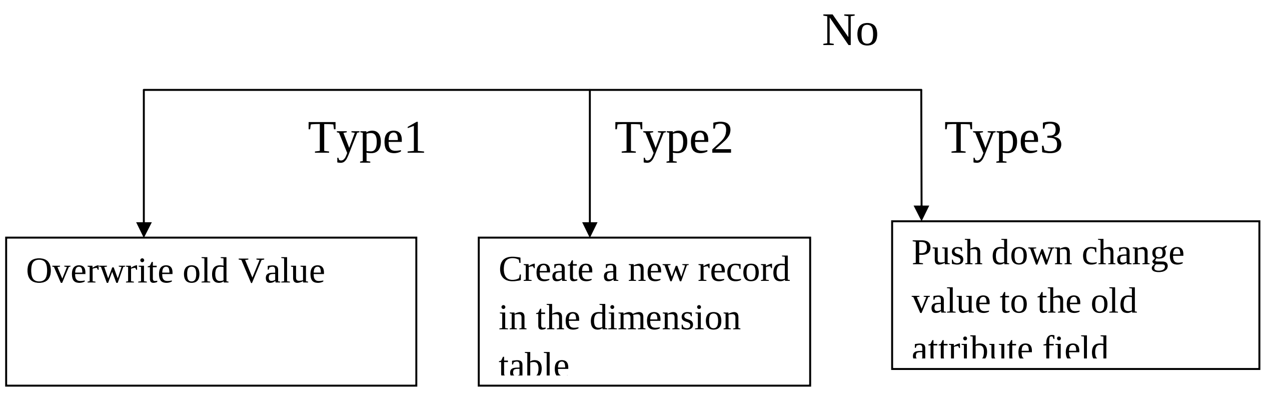

No
Type1
Type2
Type3
Overwrite old Value
Create a new record in the dimension table
Push down change value to the old attribute field

Fig Handling Type1, Type2, and Type3 changes

<u>Data Quality</u>: The moment the users realize that
The data is of unacceptable quality, they lose their confidence in the data warehouse. Hence all the efforts made by the users, sponsors, and members of the project team are lost. Once the users lose their trust in the warehouse, they will never be willing to come back to it for making strategic decisions.

Need for Data Quality: To understand the critically of data quality in a data warehouse system, let us first examine the benefits of data quality.

Boosts confidence in decision-making and enhance strategic decision making: If the data in the data warehouse is reliable and of high quality, then decisions based on the information will be sound and effective. No data warehouse can add value to a business until the data is clean and of a high quality.

Enables better customer service: Different types of customers need different types of services. Quality information helps the executives and managers to cater to the needs of different groups of users as per their aspirations and expectations.

Increases opportunity to add better value to the services: Good quality data in a data warehouse opens the doors to immense opportunities to cross-sell across product lines and departments. The marketing managers can form a list of buyers of one product and then use this information to determine all other products that every individual buyer may be interested in. So after getting this information, the marketing department can conduct well targeted campaigns.

Reduces costs and risks: The topmost risk behind lack of quality data in the data warehouse is that poor data would lead to poor decisions and can sometimes also lead to disastrous consequences. Other risks include wastage of time and resources and malfunctioning of the entire system. For example, if the data about customers is incorrect or incomplete in the data warehouse, sending mails for letting them know about promotional coupons will be futile.

Improves productivity: The primary goal of a data warehouse is to provide enterprise-wide information to the users for streamlining processes and operations. It is the accurate, timely, and business wide information which in turn leads to effective decision making thereby increasing the productivity of the business as a whole.

After analyzing the benefits of data quality, let us talk about the costs of poor data quality. Cleansing the data and improving the quality of data requires both money and effort, but these expenditures are justifiable in a data warehouse environment. The cost of not having good quality data can be viewed as below:

- Bad decisions.
- Lost business opportunities.

- Wastage of resources.
- Inconsistent data reports.
- Time and effort needed to correct the data

Categories of Errors: The categories of errors which affect the data quality can be divided into four categories:
(a) Incomplete errors, (b) Incorrect errors, ©Incomprehensibility errors and (d) Inconsistency errors

Incomplete Errors:

Missing Records: This means a record that should be in a source system is not present. It is not possible to spot this type of error unless there is another system or old reports to tally with.
Missing fields: These are fields that should be there but are absent.
Incorrect Errors:
Wrong codes: This generally takes place when an old transaction processing system has assigned a code that the new transaction processing system does not use. Now if the code is not valid, the system is going to fall into problems.
Incomprehensibility Errors: There are the types of conditions that make source data difficult to read.
Multiple fields within one field: This situation occurs when the source system has one field that in turn contains information for multiple fields in the data warehouse database. By far the most common occurrences of this problem is when the field name is broken into multiple fields like first name, middle name, and last name, e.g. “Tim E. Bur nard” , is kept in one field in the source system and it is necessary to parse this into three fields in the warehouse.

Inconsistency errors: The inconsistency errors form the biggest category of errors as a similar data from different systems can easily be inconsistent.
Inconsistent use of different codes: The most common example of this error is that one system uses M and F and another system that uses 0 or 1 to distinguish gender.
Inconsistent aggregating: Due to inconsistent business rules multiple sets of aggregated data will contain different results in different source systems.
Lack of referential integrity: This error occurs when the source systems have been built without this basic check.

PRASHANT SHUKLA 8TH SEM C.S.E MMCT

Unit-3
Information access and delivery

OLAP in data warehouse: Data warehouse users perform complex multidimensional analysis that involves enormous amount of calculations. The traditional report writers, query tools, spreadsheets, and language interfaces are not sufficient for this purpose. Obviously, the tools that are used in the operational databases do not match the set of tools and products that are specifically meant for serious analysis in a data warehouse environment. We need OLAP in the data warehouse. The basic virtues of OLAP are as follows:

- Enables the analysts, executives, and managers to gain useful insights in business.
- Enables the analysts to measure metrics along several dimensions.

- Allows data to be viewed from different perspectives.
- Supports multidimensional analysis.
- Enables the analysts to drill-down or roll-up within each dimension.
- Provides fast response, thereby encouraging speed-of-thought analysis.
- Improves the presentation of results through visual presentations using graphs and charts.
- Can be implemented on the web.

Uses and Benefits

After exploring the features of OLAP in sufficient detail, you must have already deduced the enormous benefits of OLAP. The ability to perform multidimensional analysis with complex queries sometimes also entails complex calculations. Let us summarize the benefits of OLAP systems:

- Increased productivity of business mangers, executives and analysts.
- Inherent flexibility of OLAP systems means that users may be self-sufficient in running their own analysis without IT assistance.
- Benefit for IT developers because using software specifically designed for the system development results in faster delivery of applications.
- Self-sufficiency of users, resulting in reduction in backlog.
- Faster delivery of applications following from the previous benefits.
- More efficient operations through reducing time on query executions and in network traffic

- Ability to model real-world challenges with business metrics and dimension.

OLAP models. Two types of OLAP model are there:

1. ROLAP (Relational Online Analytical Processing)
2. MOLAP (Model Online Analytical processing)

Implementation steps. These are the steps which implement the OLAP:

- Dimensional Modeling.
- Design and Building of the MDDB.
- Selection of the data to be moved into the OLAP system.
- Data acquisition or extraction for the OLAP system.
- Data loading into the OLAP server.
- Computation of data aggregation and derived data.
- Implementation of application on the desktop.
- Provision of user training.

Data Warehousing and the Web : A successful data warehouse is not solely dependent on surfacing data organized for decision support activities or controlling the processes by which the data warehouse is built. In fact, the success of the data warehouse is dependent on the ability of the end-user community to understand and gain access to the newly created Information. The decision support user must be comfortable with the information that they are basing their decisions on.

INTRODUCTION: A number of factors are contributing to an increase in users who have access to strategic corporate information most notably, data warehousing and web-based applications. Data is no longer restricted to a community of "power users". Web-based applications are opening the door to new communities of users. Once thought of as independent corporate initiatives, data warehousing and web browsers have come together to form an effective approach to surfacing information to an expanding community of users. By surfacing decision support data within a web-based application, users are able to easily view and act on information that can move their organization forward and maintain the competitive advantage necessary in today's market place.

THE ROLE OF METADATA

Organizations have had great success building warehouses with The SAS® System for a number of years. SAS has long been a provider of data access engines and a robust platform-independent transformation engine. The difficulty has come in trying to manage the warehousing process across a diverse environment. The solution lies with metadata. The warehouse metadata repository is a facility that acts as a central storage point for maintaining the information necessary to define and control the warehousing process.

SAS/Warehouse Administrator provides a framework for creating and managing the warehouse environment. The process is governed by a central metadata repository.

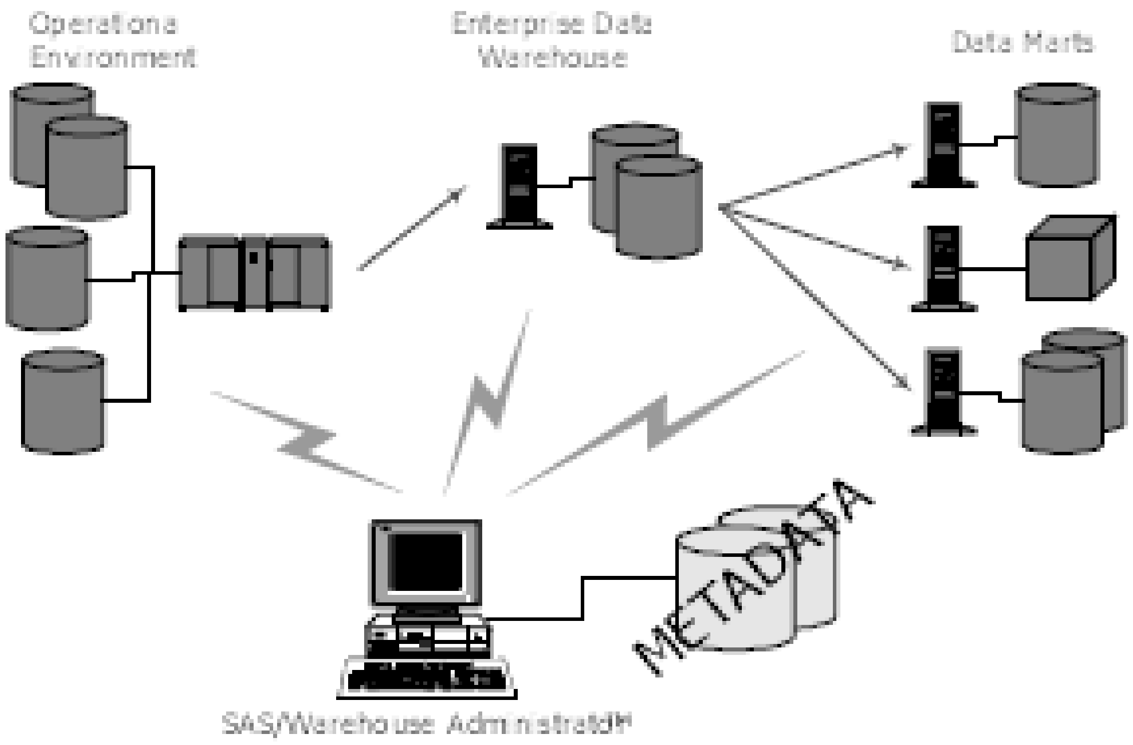


Figure 1 - Metadata, A Central Point of Control

Figure 1 shows the relationship between the warehousing Process, SAS/Warehouse Administrator, and metadata. SAS/Warehouse Administrator acts as an agent in the Enterprise to manager the flow of information from the Operational systems into the decision support Environment. Information on source data stores, Necessary transformations, data movement, and the target Warehouse is maintained in the central metadata Repository. SAS/Warehouse Administrator provides a Graphical front end to create and manage the metadata Layer. Information contained in the metadata includes:

· Source data locations

· Definition of source data
· Access methods for source data
· Business descriptions of source data
· Business rules associated with data transformation
· Location and timing of transformation processes
· Warehouse table definitions
· Location of warehouse information
· Access methods for warehouse information
· Business descriptions for warehouse objects

Warehouse Viewer
Warehouse Viewer is a sample application that is Available with SAS/Internet software that allows users to Browse the contents of a warehouse. With Warehouse Viewer, users can access the contents of a warehouse That is being managed by SAS/Warehouse Administrator. End users can use their web browsers to locate tables, Charts, graphs, and documentation associated with the Warehouse. Users are provided both technical and Business metadata.

Data warehouse Deployment: Deployment is the next phase after construction. The main concerns in the deployment phase relate to the users getting the training, support, and the hardware and tools they need to get into the warehouse.

To find our place in the whole life cycle of data warehouse deployment, let us summarize the functions and operations that have been completed up to this point. Here is the list of major activities:

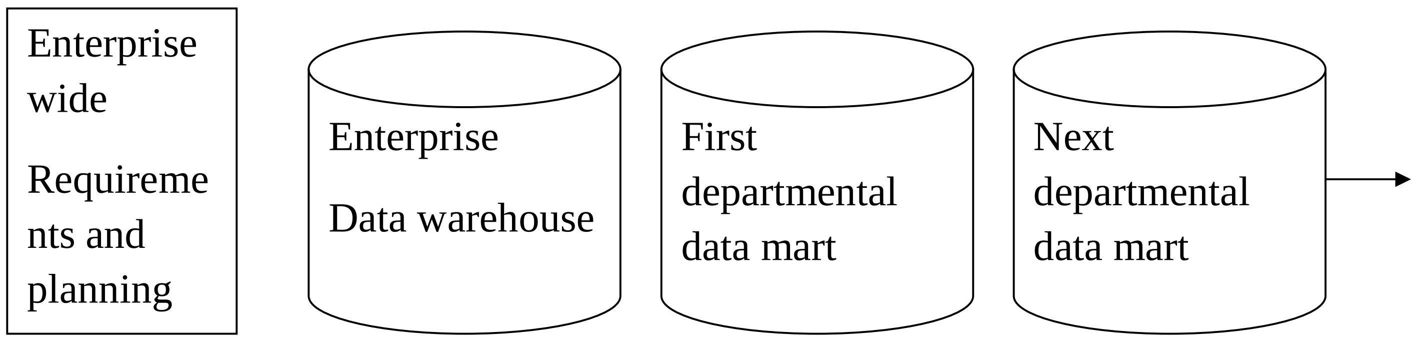
- The infrastructure is in place with the components fully tested.
- The validity of the architecture is already verified.

- The database is defined. Space allocation for the various tables is completed.
- The staging area is fully set up with file allocation.
- The extract, transformation, and all other staging area jobs are tested.
- The creation of the load images is tested in the development environment. Testing of initial loads and incremental loads is done.
- Query and reporting tools are tested in the development environment.
- The OLAP system is installed and tested.
- Web-enabling the data warehouse is completed.

Deploy in stages. Building **and** deploying a data warehouse is a major undertaking for any organization. This is a project that calls for many different types of skills. The data warehouse encompasses several different technologies. Dimensional modeling is a very different approach not previously used by designers in any operational systems.

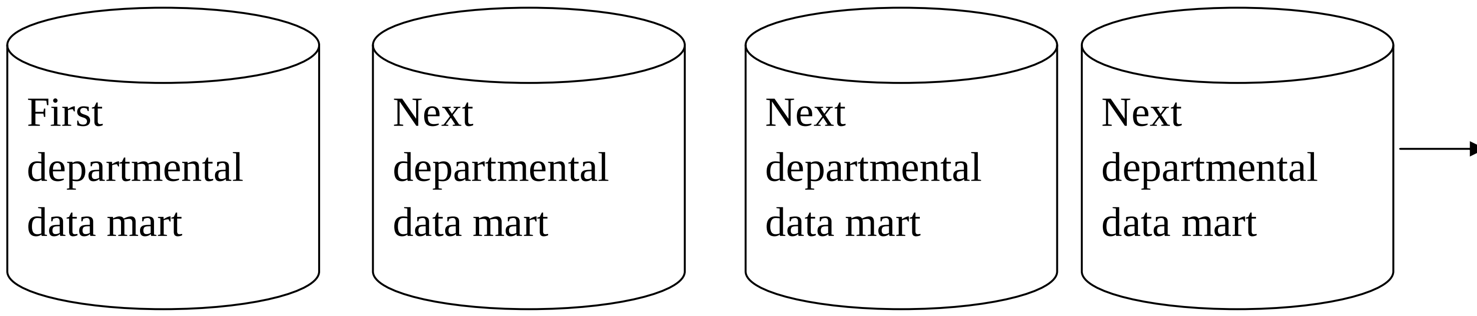
Under these circumstances, what is a reasonable method for deploying your data warehouse? Most decidedly, if you parcel the deployment into manageable parts, you can bring the data ware house in at a comfortable pace. Make the deployment happen in stages. Clearly chalk out the stages. Plan a schedule that is most effective from the point of view of the users as well as the project team.

You plan for the entire enterprise but you deploy the pieces in well-defined stages. Refer fig showing the staging of the deployment. Notice the suggested stages under the different approaches. Note how you build the overall enterprise wide data warehouse first in the top-down approach. Then the dependent data marts are deployed in a suitable sequence. The bottom-up approach is less structured and less refined. In the practical approach, you deploy one data mart at a time.


Enterprise wide
Requirements and planning
Enterprise Data warehouse
First departmental data mart
Next departmental data mart

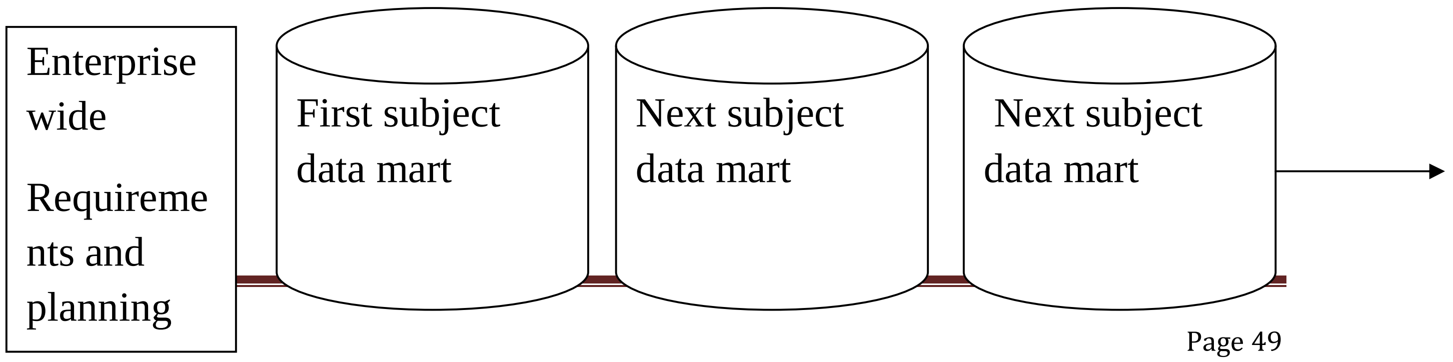
TOP-DOWN APPROACH

Deploy the overall enterprise data warehouse (E-R model) followed by the dependent data marts one by one.


First departmental data mart
Next departmental data mart
Next departmental data mart
Next departmental data mart

BOTTOM-UP APPROACH

Gather departmental requirements, plan, and deploy the independent data marts, one by one.



PRACTICAL APPROACH

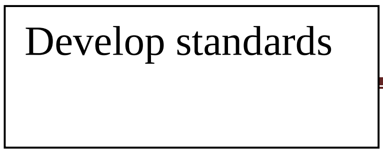
Deploy the subject data marts, one by one, wit fully conformed dimensions and facts, according to the preplanned sequence

Fig staged data warehouse deployment.

Security Policy. The security policies for the data warehouse are:

- The scope of the information covered by the policy.
- Physical security.
- Security at the workstation.
- Network and connections.
- Database access privileges.
- Security clearance for data loading.
- Metadata security.
- OLAP security.
- Web security.
- Resolution of security violations

Physical design process: Fig is a pictorial representation of the steps in the physical design process for a data warehouse. Note the steps indicated in the figure. In the following subsections, we will broadly describe the activities within these steps. You will understand how at the end of the process you arrive at the completed physical model.



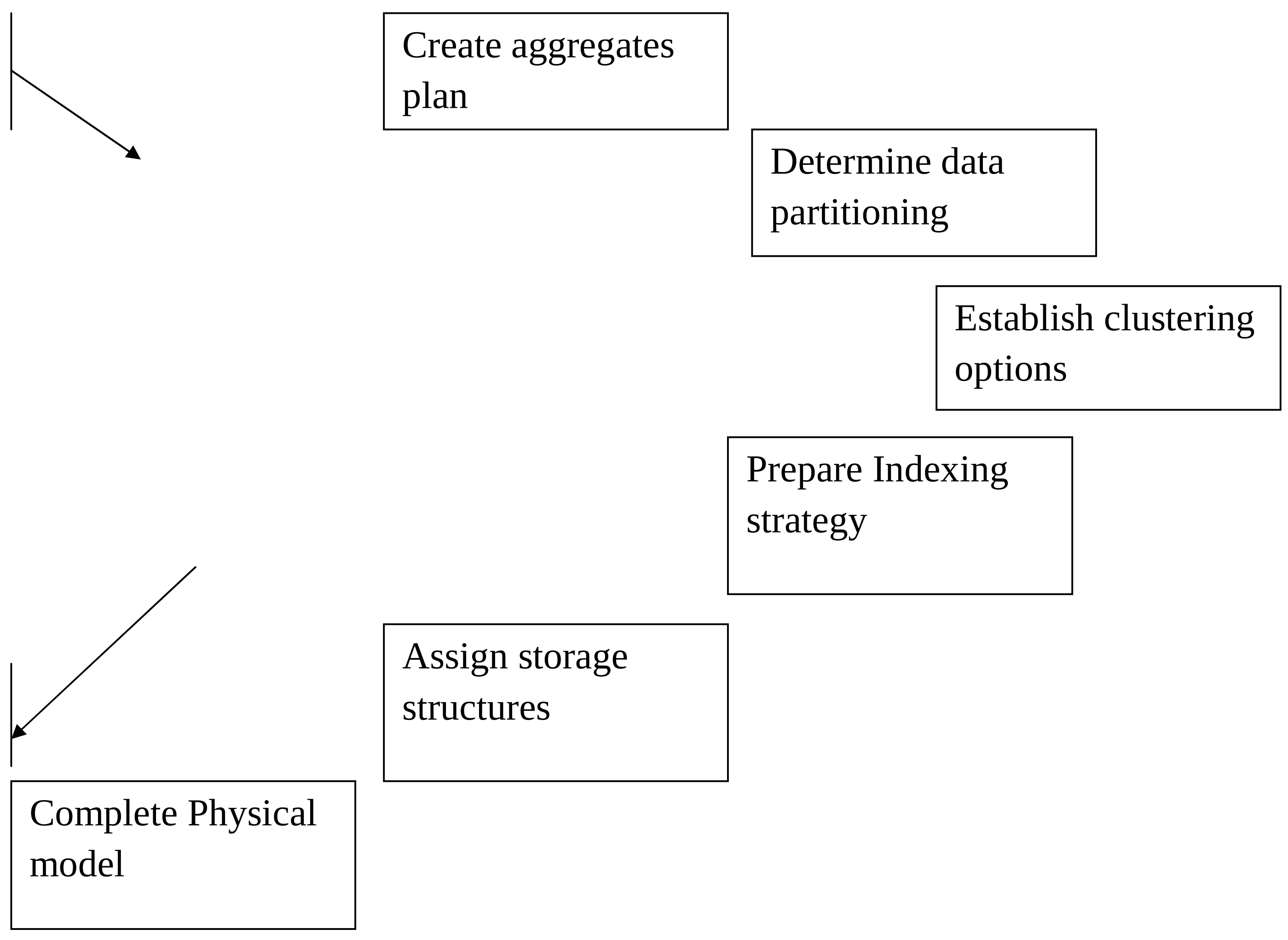

Create aggregates plan
Determine data partitioning
Establish clustering options
Prepare Indexing strategy
Assign storage structures
Complete Physical model

Fig The physical design process.

Develop Standards. In the data warehouse environment the scope of the standards expands to include additional areas. Standards ensure consistency across the various areas. If you have the same way of indicating names of the database objects, then you are leaving less room for ambiguity. Let us say the standards in your company require the name of an object to be a concatenation of multiple words separated by dashes and that the first world in the group indicates the business subject. With these standards, as soon as someone reads an object name, that person can know the business subject.

Standards take on greater importance in the data warehouse environment. This is because the usage of the object names is not confined to the IT department. The users will also be referring to the objects by names when they formulate and run their own queries.

Create aggregates plan. In this step, review the possibilities for building aggregate tables. You get clues from the requirements definition. Look at each dimension table and examine the hierarchical levels. Which of these levels are more important for aggregation? The plan must spell out the exact types of aggregates you must build for each level of summarization. It is possible that many of the aggregates will be present in the OLAP system. If OLAP instances are not for universal use by all users, then the necessary aggregates must be present in the main warehouse.

Determine the data portioning scheme. In this step, come up with a definite partitioning scheme. The scheme must include:

- The fact tables and the dimension tables selected for partitioning.
- The type of partitioning for each table-horizontal or vertical.
- The number of partitions for each table.
- The criteria for dividing each table (for example, by product groups)
- Description of how to make queries aware of partitions.

Establish clustering options. Establish the proper clustering options before completing the physical model. Examine the tables, table by table, and find pairs are related. This means that row from the related tables are usually accessed together for processing in many cases. Then make plans to store the related tables close together in the same file on the medium. For two related tables, you may want to store the records from both files interleaved. A record from one table is followed by all the related records in the other table while storing in the same file.

Prepare indexing strategy. This is a crucial step in the physical design. Unlike OLTP systems, the data warehouse is query-centric. As you know, indexing is perhaps the most effective mechanism for improving performance. A solid indexing strategy results in enormous benefits. The strategy must lay down the index plan for each table, indicating the columns selected for indexing. The sequence of the attributes in each table to determine which attribute qualifies for bit-mapped indexes.

Assign storage structures. In an OLTP system, all data resides in the operational database. When you assign the storage structures in an OLTP system, your effort is confined to the operational tables accessed by the user applications. In a data warehouse, you are not just concerned with the physical files for the data warehouse tables. Your storage assignment plan must include other types of storage such as the temporary data extract files, the staging area, and any storage needed for front-end applications. Let the plan include all types of storage structures in the various storage areas.

Complete physical model This final step reviews and confirms the completion of the prior activities and tasks. By the time you reach this step, you have the standards for naming the database objects. You have determined which aggregate tables are necessary and how you are going to partition the large tables. You have completed the indexing strategy and have planned for other performance options. You also know where to put the physical files.

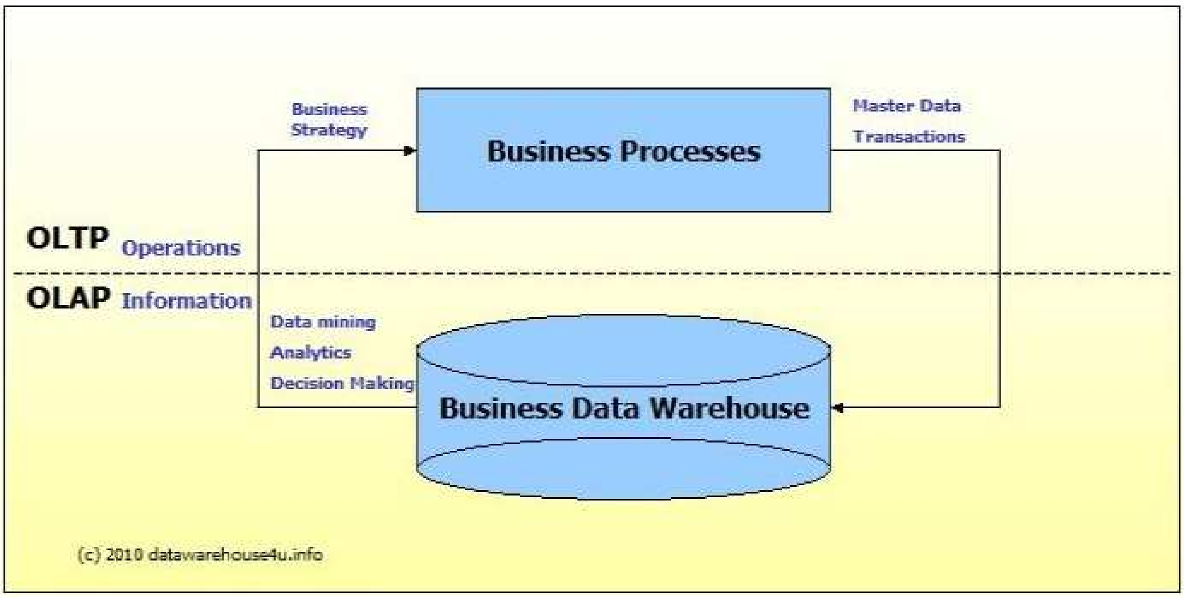
All the information from the prior steps enables you to complete the physical model. The result is the creation of the physical schema. You can code the data definition language statements (DDL) in the chosen RDBMS and create the physical structure in the data dictionary.

OLTP vs. OLAP

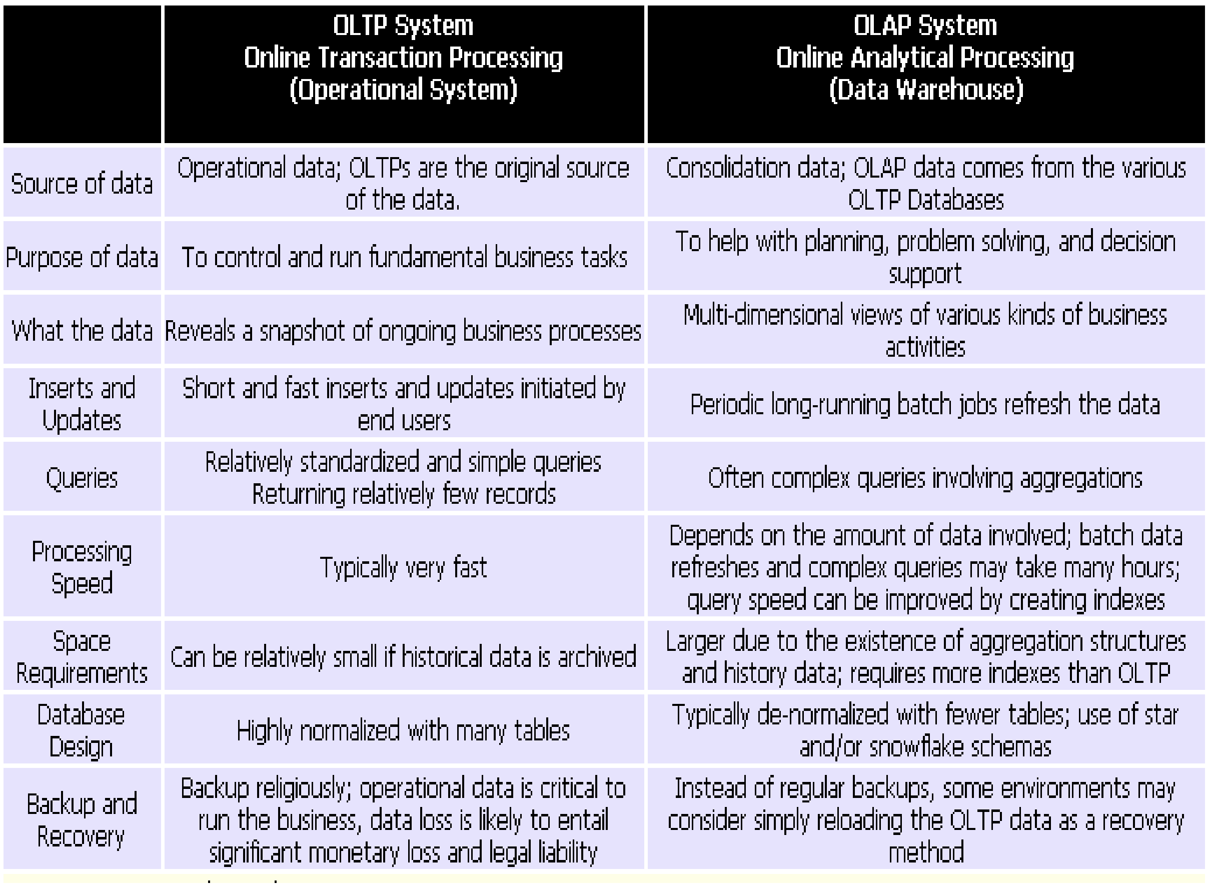
We can divide IT systems into transactional (OLTP) and analytical (OLAP). In general we can assume that OLTP systems provide source data to data warehouses, whereas OLAP systems help to analyze it.

OLTP (On-line Transaction Processing) is characterized by a large number of short on-line transactions (INSERT, UPDATE, and DELETE). The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second. In OLTP database there is detailed and current data, and schema used to store transactional databases is the entity model (usually 3NF).

OLAP (On-line Analytical Processing) is characterized by relatively low volume of transactions. Queries are often very complex and involve aggregations. For OLAP systems a response time is an effectiveness measure. OLAP applications are widely used by Data Mining techniques. In OLAP database there is aggregated, historical data, stored in multi-dimensional schemas (usually star schema).



The following table summarizes the major differences between OLTP and OLAP system design.



	OLTP System Online Transaction Processing (Operational System)	OLAP System Online Analytical Processing (Data Warehouse)
Source of data	Operational data; OLTPs are the original source of the data.	Consolidation data; OLAP data comes from the various OLTP Databases
Purpose of data	To control and run fundamental business tasks	To help with planning, problem solving, and decision support
What the data	Reveals a snapshot of ongoing business processes	Multi-dimensional views of various kinds of business activities
Inserts and Updates	Short and fast inserts and updates initiated by end users	Periodic long-running batch jobs refresh the data
Queries	Relatively standardized and simple queries Returning relatively few records	Often complex queries involving aggregations
Processing Speed	Typically very fast	Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes
Space Requirements	Can be relatively small if historical data is archived	Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP
Database Design	Highly normalized with many tables	Typically de-normalized with fewer tables; use of star and/or snowflake schemas
Backup and Recovery	Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability	Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method

Unit-4
Data Mining

Introduction: Data mining refers to the technique of searching useful and relevant information from the data warehouse. It is very clear that the technology has something to do with discovering knowledge. Data mining is used in applications, such as marketing, sales, credit analysis, and fraud detection. Data mining is somehow connected to data warehousing. Although a workable data warehouse is not a prerequisite for data mining, but it gives a practical boost to the data mining process.

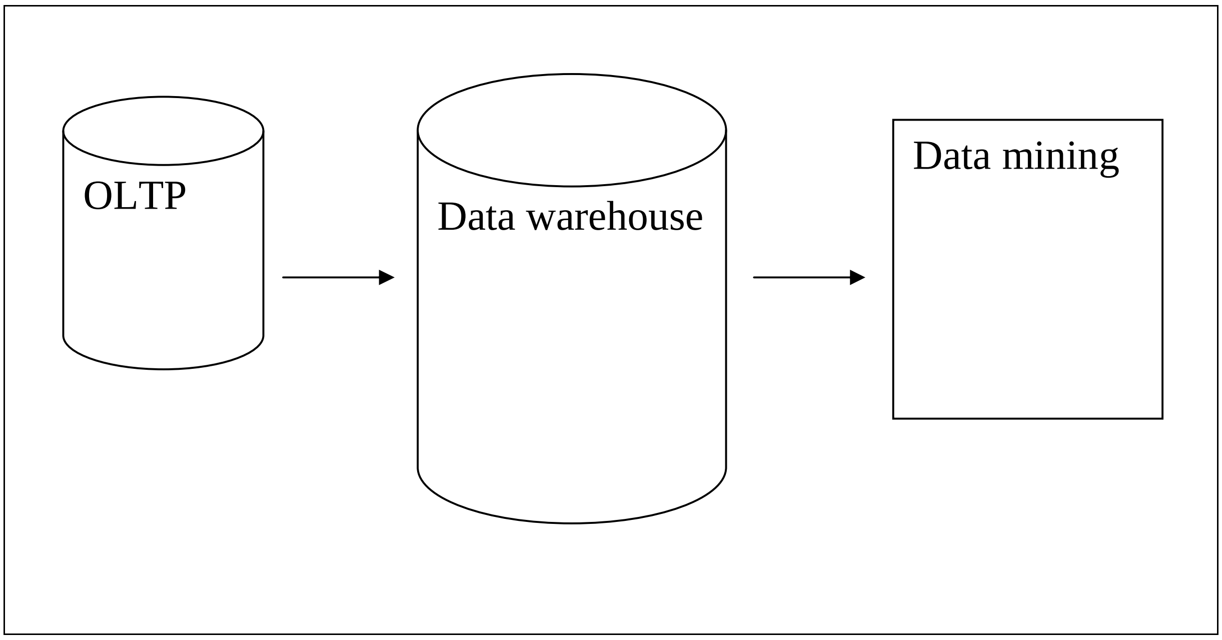


Fig Data flow

Fig shows the flow of data from operational systems into data warehousing systems.

Now let us study, “Why is data mining being put to use in more and more businesses? Here are some basic reasons:

- These days an organization produces more information in a week than most people can read in a lifetime. Thus, it is impossible for humans to study, decipher, and interpret all that data to find useful information.
- Recently, many data mining tools that support a wide range of applications have appeared in the market.
- Data mining needs substantial computing power. Parallel hardware, databases, and other powerful components are now available in the market at affordable prices.
- These days’ organizations are placing great emphasis on building sound customer relationships. Companies want to know how they can sell more to existing customers and determine which of their customers will prove to be of long –term value to them. So, data mining enables companies to find answers and discover patterns in their customer data.
- Finally, it is the fierce competitive considerations that weigh heavily on every company to get into data mining.

Knowledge Discovery Process: Data mining is often described as the process of the knowledge or information that you never knew existed in the data. Usually, this uncovered hidden knowledge is encapsulated as relationships or patterns in the data.

Relationships: Let us suppose you are at the super market. While you fetch the bread, you happen to see a pack of butter close by. Yes, you want that. You pause to look at the next five customers behind you. To your amazement, three of those customers also reach for the butter pack. Within a space of few minutes, bread and butter are bought together. Hence, data mining discovers relationships that exist between two or more different objects along with the time dimension.

Patterns: Pattern discovery is another outcome of data mining operations. Consider a credit card company that is always fussy about discovering the pattern of usage that usually warrants an increase in the credit limit or a card upgrade. They want to keep a check on their customers who must be lured with a card upgrade. The data mining algorithms are applied to mine the usage patterns of a number of card holders and discover the potential pattern of usage that will produce meaningful information. The major steps in Knowledge discovery (KDD) Process are as follows:

Step1 Define Business objectives: The first and foremost question to be answered is whether the need for a data mining solution really exists. Then define the objectives clearly like: Are you looking for marketing campaigns or detecting fraud in credit card usage? Are you looking for associations between products that sell together? In this step defined expectations. Finally, express how the final results will be presented to the users.

Step2 Prepare data: This step consists of data selection, processing, and data transformation. First of all, select the data to be extracted from the data warehouse. For this, use the business objectives to determine what data has to be selected. Then the appropriate metadata is selected that describes data about the selected data. The type of mining algorithm should have been selected at this time, as the mining algorithm has a bearing on data selection.

Step3 Perform data mining: The knowledge discovery engine applies the previously selected data mining algorithm to the prepared data. The output from this step is a set of relationships or patterns. This step and the next step of evaluation may be performed iteratively. After an initial evaluation, the data is adjusted to redo this step. However, the depth of this step depends upon the type of data mining application.

Step4 Evaluate results: With the help of knowledge discovery engine, we are actually trying to seek interesting patterns or relationships. These help in understanding the customers, products, markets, and the business. In the selected data, there may be a number of patterns or relationships and in this step all the resulting patterns have to be examined as each one of them may not be interesting for the business at all. So, you need to apply some filtering mechanism and select only the promising patterns to be presented and applied.

Step5 Present discoveries: Presentation of knowledge, discovered by applying data mining algorithm, may be in the form of visual navigation, charts, graphs, or free-form texts. Presentation also includes storing of interesting discoveries in the knowledge base for future use.

Step6 Incorporate usage of discoveries: The goal of application of data mining algorithm is to understand the business well, discern new patterns and relationships, and also turn this understanding into actions. Hence, in this step, the results of the knowledge discovery are assembled in the best way so that they can be exploited to improve the business.

Look at fig which shows the percentage of total effort in the KDD process.

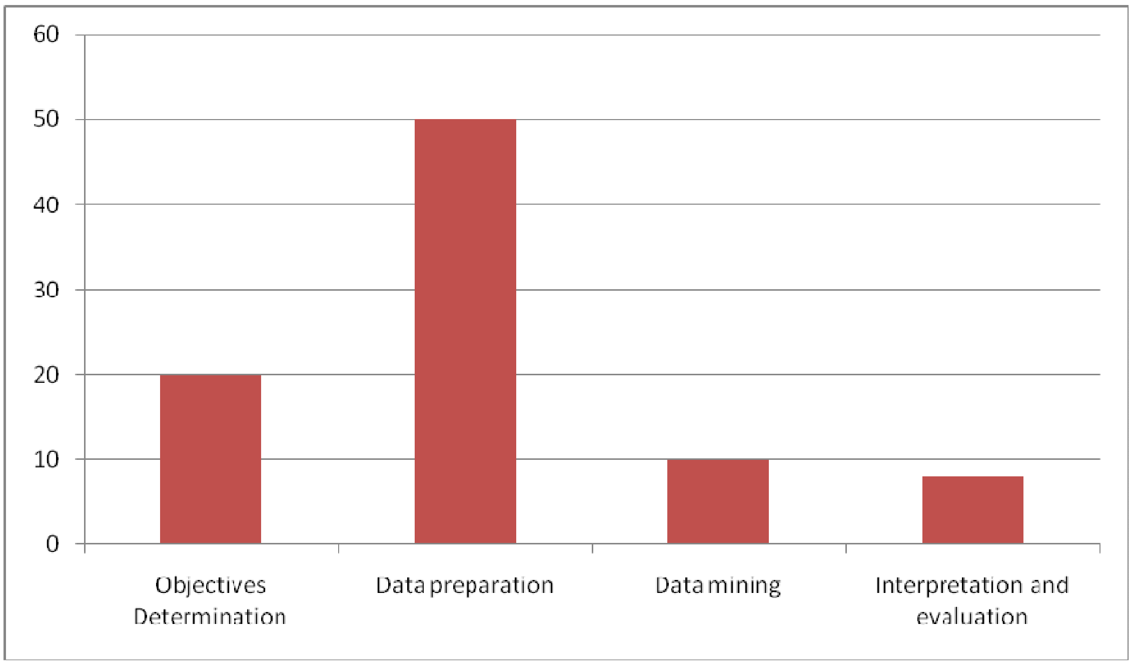


Fig Percentage of total effort in the KDD process

Data Mining Algorithm:

Classification: Classification is perhaps the most familiar and most popular data mining technique. Examples of classification applications include image and pattern recognition, medical diagnosis, loan approval, detecting faults in industry applications, and classifying financial market trends. Estimation and prediction may be viewed as types of classification. When someone estimates your age or guesses the number of marbles in a jar, these are actually classification problems. Prediction can be thought of as classifying an attribute value into one of a set of possible classes. It is often viewed as forecasting a continuous value, while classification forecasts a discrete value.

Algorithms:

Regression: Regression problems deal with estimation of an output value based input values. When used for classification, the input values are values from the database D and the output values represent the classes. Regression can be used to solve classification problems, but it can also be used for other applications such as forecasting. In actuality, regression takes a set of data and fits the data to a formula.

$$y=co+c1x1+\dots\dots\dots\dots+cn\ xn$$

By determining the regression coefficients Co, c1… Cn-1 the relationship between the output parameter, y, and the input parameters, x1… xn can be estimated.

Distance based algorithms: Each item that is mapped to the same class may be thought of as more similar to the other items in that class than it is to the items found in other classes. Therefore, similarity (or distance) measures may be used to identify the "alikeness" of different items in the database.

Algorithm

Input:

```
        C1, ------Cm // Centers for each class
         t           //  Input tuple to classify
```

Output:

```
        C          // Class to which t is assigned
```

Simple distance-based algorithm

```
dist=infinite;
For i: =1 to m do
    if dis (ci, t) <dist, then
    c=i;
    dist=dist (ci, t);
```

Decision Tree-Based Algorithms: The decision tree approach to classification is to divide the search space into rectangular regions. A tuple is classified based on the region into which it falls.

Algorithm

```
Input:
  D        // Training data
Output:
   T        // Decision tree
DTBuild algorithm:
             // Simplistic algorithm to illustrate naïve approach to
building DT
T=0;
Determine best splitting criterion;
T= Create root node node and label with splitting attribute;
T=Add arc to root node for each split predicate and label;
 For each arc do
    D=Database created by applying splitting predicate to D;
 If stopping point reached for this path, then
 T'=Create leaf node and label with appropriate class;
Else
   T'=DTBuild (D);
```

T=Add T' to arc;

Clustering: Clustering is the groups that are not predefined. Instead, the grouping is accomplished by finding similarities between data according to characteristics found in the actual data. The groups are called Clusters. Many definitions for clusters have been proposed:

- Set of like elements. Elements from different clusters are not alike.
- The distance between points in a cluster is less than the distance between a point in the cluster and any point outside it.

Hierarchical Algorithms: Hierarchical Algorithms actually creates sets of clusters. Example illustrates the concept. Hierarchical algorithms differ in how the sets are created. A tree data structure, called a dendrogram, can be used to illustrate the hierarchical clustering technique and the sets of different clusters. The root in a dendogram tree contains one cluster where all elements are together. The leaves in the dendogram each consist of a single element cluster. Internal nodes in the dendogram represent new clusters formed by merging the clusters that appear as its children in the tree. Each level in the clusters formed by merging the clusters that appear as its children in the tree. Each level in the tree is associated with the distance measure that was used to merge clusters. All clusters created at a particular level were combined because the children clusters had a distance between them less than the distance value associated with this level in the tree. A dendrogram for Example is seen in figure.

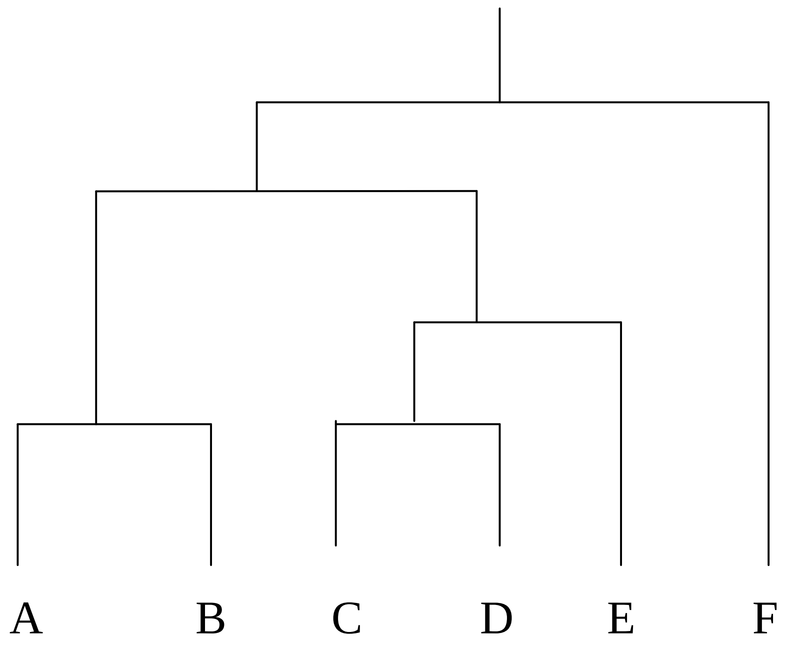


Fig Dendrogram for ex.

Example. Fig shows six elements, {A, B, C, D, E, F}, to be clustered. Parts (a) to (e) of the figure show five different set of clusters. In part (a) each cluster is viewed to consist of a single element. Part (b) illustrates four clusters. Here there are two sets of two-element clusters. These clusters are formed at this level because these two elements are closer to each other than any of the other elements. Part (c) shows a new cluster formed by adding a close element to one of the two element clusters. In part(d) the two-element and three-element clusters are merged to give a five-element cluster.

The space complexity for hierarchical algorithms is O (n2) because this is the space required for the adjacency matrix.

Genetic Algorithms: To determine how to perform clustering with genetic algorithms, we first must determine how to represent each cluster. One simple approach would be to use a bit-map representation of each possible cluster. So, given a database with four items, {A, B, C, D}, we would represent one solution to creating two clusters as 1001 and 0110. This represents the two clusters {A, D} and {B, C}.

Algorithm shows one possible iterative refinement technique for clustering that uses a genetic algorithm.

```
Algorithm
Input:
D= {t1, t2 …tn}        // Set of elements
K      //Number of desired clusters
Output:
K    //Set of clusters
GA clustering algorithm:
   Randomly create an initial solution;
Repeat
 Use crossover to create a new solution;
 Until terminate criteria is met;
```

Association rules: The purchasing of one product when another product is purchased represents an association rule. Association rules are frequently used by retail stores to assist in marketing, advertising, floor placement, and inventory control.

Basic Algorithm:

Apriori Algorithm: The Apriori algorithm is the most well known association rule algorithm and is used in most commercial products. It uses the following property, which we call the large item set property:

Any subset of a large item set must be large.

The basic idea of the Apriori algorithm is to generate candidate item sets of a particular size and then scan the database to count these to see if they are large. During scan i candidates of size i, Ci are counted. Only those candidates that are large are used to generate Ci+1. An item set is considered as a candidate only if all its subsets also are large. To generate candidates of size i+1, joins are made of large item sets found in the previous pass.

The Apriori-Gen algorithm is shown in algorithms which are as follows:

Input:

L_{i-1} // Large item sets of size i-1

Output:

C_i // Candidates of size i

Apriori-gen algorithm:

```
$C_i = \emptyset$;
 For each   I epsilons Li-1 do
   For each   J $\neq$ I epsilons Li-1 do
     If i-2 of the elements in I and J are equal then
          $C_k = C_k$ U {I U J};
```

Advanced Association Rule Technique:

Multiple-Level Association Rules: A variation of generalized rules is multiple-level association rules. With multiple-level rules, item sets may occur from any level in the hierarchy. Using a variation of the Apriori algorithm, the concept hierarchy is traversed in a top-down manner and large item sets are generated. When large item sets are found at level i, large item sets are generated for level i+1. Large k-item sets at one level in the concept hierarchy are used as candidates to generate large k-item sets for children at the next level.

Quantitative Association Rules: A quantitative association rule is one that involves categorical and quantitative data.

<u>Data Mining Techniques</u>: There are many different methods used to perform data mining tasks. These techniques not only require specific types of data structures, but also imply certain types of algorithmic approaches

Parametric models describe the relationship between input and output through the use of algebraic equations where some parameters are not specified. Thus for real world problems, these parametric models may not be useful.

Non parametric techniques are more appropriate for data mining applications. A non parametric model is one that is data-driven. No explicit equations are used to determine the model. This means that the modeling process adapts to the data at hand. Non parametric techniques include neural networks, decision trees, and genetic algorithms.

Decision Trees: A decision tree is a predictive modeling technique used in classification, clustering, and prediction tasks. Decision trees use a “divide and conquer” technique to split the problem search space into subsets. It is based on “Twenty Questions” game that children play, as illustrated by ex. Fig graphically shows the steps in the game. Nodes at the third level show question asked at the third level in the game. Leaf nodes represent a successful guess as to the object being predicted. This represents a correct prediction. Each question successively divides the search space much as a binary search does. Space is divided into two equal parts. Often young children tend to ask poor questions by being too specific, such as initially asking “Is it my Mother?" This is a poor approach because the search space is not divided into two equal parts.

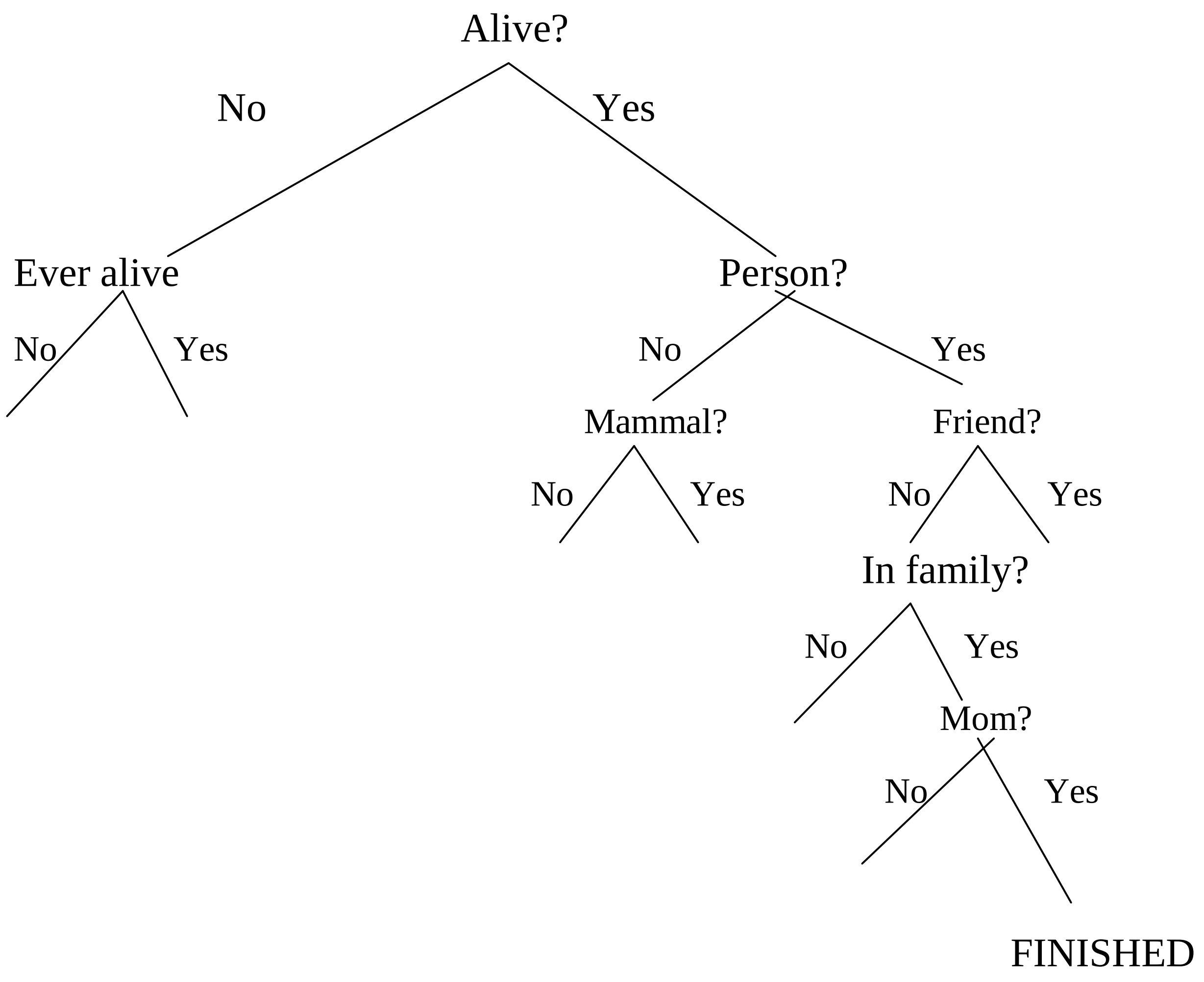


Fig .Decision tree

DEFINITION. A decision tree (DT) is a tree where the root and each internal node is labeled with a question. The arcs emanating from each node represent each possible answer to the associated question. Each leaf node represents a prediction of a solution to the problem under consideration.

Algorithm

```
Input:
T       // Decision tree
D     // Input database
Output:
M    // Model prediction
DTProc algorithm:
     // Simplistic algorithm to illustrate prediction technique using DT
  For each t E D do
    n=root node of T;
     While n not leaf node do
           Obtain answer to question on n applied to t;
           Identify arc from t, which contains correct answer;
           n=node at end of this arc;
Make prediction for t based on labeling of n;
```

Neural Networks: The first proposal to use an artificial neuron approached in 1943, but computer usage of neural networks did not actually begin until the 1980s. Neural networks (NN), often referred to as artificial neural networks (ANN) to distinguish them from biological neural networks, are modeled after the workings of the human brain.

The NN is actually an information processing system that consists of a graph representing the processing system as well as various algorithms that access that graph. As with the human brain, the NN consists of many connected processing elements.

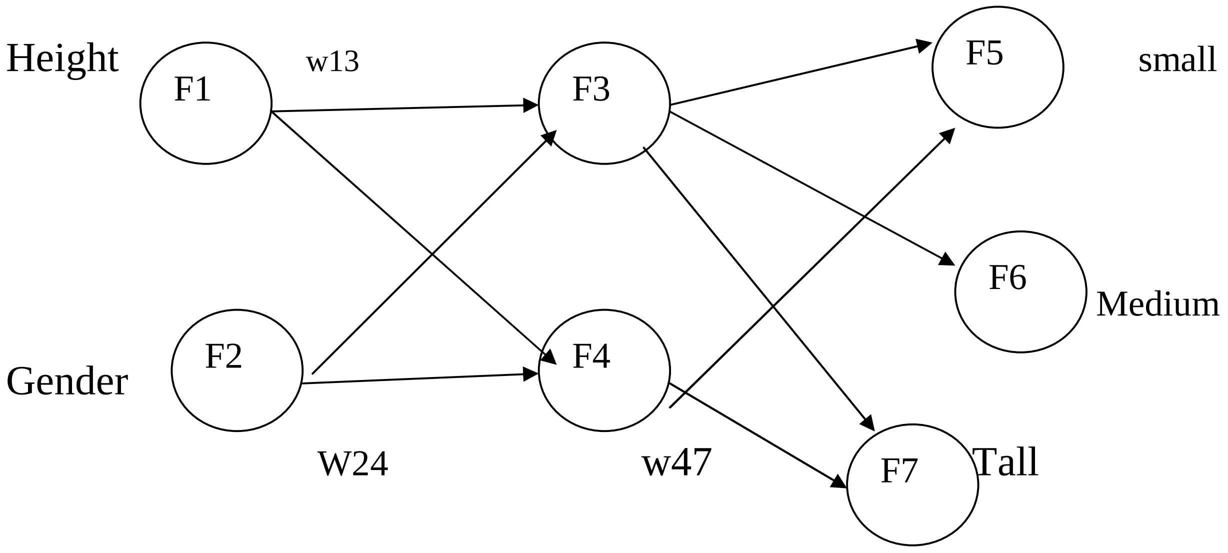


Fig. Neural network

DEFINITION. A neural network (NN) is a directed graph, F= (V, A) with vertices V= {1, 2… n} and arcs A={i , j} 1< i,j <n}, with the following restrictions:

1. V is partitioned into a set of input nodes, V_I, hidden nodes, V_H, and output nodes, V_O.
2. The vertices are also partitioned into layers {1,…..,k} with all input nodes in layer 1 and output nodes in layer k, All hidden nodes are in layers 2 to k-1 which are called the hidden layers.
3. Any arc (i, j) must have node i in layer h-1 and node j in layer h.
4. Arc (i, j) is labeled with a numeric value w_{ij}.
5. Node i is labeled with a function f_i.

Genetic Algorithms: Genetic algorithms are example of evolutionary computing methods and are optimization-type algorithms. Given a population of potential problem solutions (individuals), evolutionary computing expands this population with the new and potentially better solutions.

DEFINITION. A genetic algorithm (GA) is a computational model consisting of five parts:

1. Starting set of individuals, P.
2. Crossover technique.

3. Mutation algorithm.
4. Fitness function.
5. Algorithm that supplies the crossover and mutation techniques to P iteratively using the fitness function to determine the best individuals in P to keep. The algorithm replaces a predefined number of individuals from the population with each iteration and terminates when some threshold is met.

Algorithm:

```
Input:
P    //Initial population
Output:
P' //Improved population
Genetic algorithm:
    //Algorithm to illustrate genetic algorithm
  Repeat
   N= [P];
 P'=$\emptyset$;
  Repeat
    $I_1$, I2= Select (P);
    O1, O2=cross (I1, I2);
    O1=mutate (o1);
    O2=mutate (o2);
   P'=P'U {o1, o2};
Until [P'] =N;
P=P'
Until termination criteria satisfied;
```

Disadvantages:

- Genetic algorithms are difficult to understand and to explain to end users.
- The abstraction of the problem and method to represent individuals is quite different.

Unit-5

Web Mining

Syllabus:-

1.Web Content Mining.
2 Web Structure Mining.
3 Web Usage mining.
4.Advanced Topics:

Spatial mining, Temporal mining. Visualisation : Data generalization and summarization-based
characterization, Analytical characterization: analysis of attribute relevance, Mining class
comparisons: Discriminating between different classes, Mining descriptive statistical measures in
large databasesData Mining Primitives, Languages, and System Architectures: Data mining
primitives, Query language, Designing GUI based on a data mining query language, Architectures of
data mining systems Application and Trends in Data Mining: Applications, Systems products and
research prototypes, Additional themes in data mining, Trends in data mining

Web Content Mining: Web content mining implies the automatic search of information available online and involves web data content. The emphasis here is on the content of the web page.

The Web document usually contains several data types, such as text, image, audio, video, metadata, and hyperlinks where some of this data are semi-structured like HTML documents or structured like the data in the tables or database generated HTML pages. However, most of the data on the Web is unstructured text data. The unstructured characteristic of the web is unstructured text data. The unstructured characteristic of the web data forces the web content mining towards a more complicated approach.

Web content mining widely applies the techniques from other subjects like machine learning, statistical pattern recognition, and data mining for analyzing hypertext. Multimedia data mining is a subset of the web content mining that involves mining the high-level information and knowledge from large online multimedia sources. Multimedia data mining on the web has become the most demanding topic of research, as more and more researchers are working towards a unifying framework for representation and problem solving. Although learning from multimedia is a big challenge, this concept is still under research.

Web Structure Mining: Web structure mining is to generate a structural summary about the website and the web page. Web content mining mainly focuses on the structure of the inner-document, while Web structure mining tries to discover the link structure of the hyperlinks at the inter-document level. Based on the topology of the hyperlinks, Web structure mining categorizes the web pages and generates information about the similarity and relationship between different websites.

Web structure mining also involves discovering the structure of the web document itself and can be used to reveal the structure (schema) of web pages. Web structure mining is useful for navigation purposes and makes it possible to compare/integrate web page schemes. It also facilities the use of database techniques for accessing information in web pages by providing a reference schema.

Web structure mining reveals structural information that includes the information measuring the following:

1. Frequency of the local links in the web tuples in a web table.

2. Frequency of web tuples in a web table containing links those are within the same document.
3. Frequency of web tuples in a web table that contains links that are global and the links that span different websites.
4. Frequency of identical web tuples that appear in the web table or among the web tables.

If a web page is linked to another web page directly, or if certain to be the neighbors of that page, web structure mining will discover the relationships among web pages. This relationship may exist because of the presence of synonyms, having similar contents both the pages residing on the same web server.

Web structure mining also discovers the nature of the hierarchy or network of hyperlinks in the websites of a particular domain, thereby allowing query processing to be easier and more efficient.

Web usage mining: Web usage mining is a relatively new research area, and is gaining more and more attention in recent years. When users visit a website, the only information left behind by them is the path of the web pages that they have accessed. Web usage mining applies data mining techniques to discover user navigation patterns and tries to discover the useful information from the secondary data (the data taken from the web server access logs, proxy server logs, browser logs, user profiles, registration data, mouse clicks and scrolls) derived from the user interactions while surfing on the web.

Web mining lays emphasis on the techniques that could predict the behavior of every individual user who interacts with the Web, could make a comparison between expected and actual website usage, and make adjustments in the website to the interests of its users. Analyzing such data can help the organizations to value its customers by knowing more about them, using accurate cross-marketing strategies across products, and offering them effective promotional campaigns, etc.

Generally, the web content and website topology will be used as the information sources for preparing data for web usage mining. But even for effective web usage mining, data cleaning, and data transformation steps are performed before analysis.

The techniques for web usage mining are classified into two approaches.

1. The first approach maps the usage data of the web server into tables before data mining is performed. Data mining techniques such as clustering and classification could then be used to mine the usage data provided this data has been pre-processed.

The second approach makes use of the log data directly by utilizing special pre-processing techniques.

Web usage mining process involves the following steps:

(a) Identifying the problem, (b) collecting data, (c) pre-processing of data, and (d) application of pattern discovery and analysis techniques.

<u>Temporal Data Mining</u>: Temporal data mining succeeds over traditional data mining techniques, as it has the capability to infer casual and temporal relationships, and this is something that non-temporal data mining cannot do. However, data mining from temporal data is not temporal data mining, if the temporal component is either ignored or treated as a simple numerical attribute. Moreover, traditional data mining techniques cannot be applied to mine temporal rules from a database which does not have temporal components stored in it. Thus, the underlying database must be a temporal one.

Let us consider an association rule that states: “Any person who buys a computer also buys a printer.” When the temporal aspect is brought into consideration, this rule would be: “Any person who buys a computer also buys a printer after that.”

Let us take an example: “Data warehousing is being widely used now-a-days.” It shows another technique of mining from previously mined rules, to find trends in rule sets.

Temporal data mining tasks:

Temporal association: The association rule discovery can be extended to operate on temporal data and can be used to discover associations between temporal and non- temporal datasets.

Temporal classification: Temporal **classification** forms cluster of data items along temporal dimensions. For example, clusters of students who attend morning classes and those who prefer studying in evening classes can be formed similarly; clusters of students can be formed who opt for higher education.

Temporal characterization: For **temporal characterization,** we extend the concept of decision tree construction based on temporal attributes. For example, a rule could be "The first case of cholera is normally reported after floods that occur during the monsoon rains in the months of july-september".

Trend analysis: The analysis of time series data predicts trends in the data. Thus, trend analysis is used to find the relationships of changes in one or more static attributes, with respect to changes in the temporal **attributes**.

Sequence analysis: Events that occur at different instants of time may be related to each other by casual relationships. That is to say, an earlier event may appear to cause a later one. In order to discover such relationships, sequences of events may be analyzed to discover common patterns. This task discovers frequent events and also predicts certain events that are likely to occur.

Spatial Data mining: The importance of spatial data mining is growing with the increasing significance of large geo-spatial datasets, such as maps, repositories of remote-sensing images, and the decennial census. Spatial data mining is widely applied by the military forces to formulate the army's strategic, tactical, and operational plans. They use strategic mining to infer enemy tactics (e.g. flank attack), locate lost ammunition dumps and enemy sites.

- The difference between classical and spatial data mining is somewhat similar to the differences between classical and spatial statistics.
- While spatial data is embedded in a continuous space, the classical data is often discrete in nature.
- Spatial patterns are often local, whereas classical data mining techniques lays emphasis on global patterns.
- Classical statistical analysis assumes that data samples are independently generated, whereas in case of spatial data analysis, the assumption about the independence of samples is generally false because spatial data tends to be highly auto-correlated.

  For example, generally it is seen that people with similar characteristics, occupation, and background are clustered together in the same neighborhoods. In spatial statistics, we call this as spatial auto-correlation. If we ignore these spatial auto-correlations, while analyzing the data that has spatial characteristics, then we may produce inaccurate or inconsistent hypotheses and models.

Techniques in spatial mining:

Database primitives: A set of database primitives for spatial database mining that is sufficient to express most of the spatial data mining algorithms and which can be efficiently supported by a DBMS has already been developed. The use of these database primitives will enable the integration of spatial data mining with existing DBMS and these primitives will enable the integration of spatial data mining algorithms. The spatial data mining data base primitives are based on the concepts of neighborhood graphs and neighborhood paths.

Efficient DBMS support: Effective filters allow restricting the search. Neighborhood indices materialize certain neighborhood graphs to support **efficient** processing of the database primitives by a DBMS. These database primitives are implemented on top of the DBMS.

Algorithms for spatial data mining: New algorithms for spatial characterization and trend analysis are being developed. For spatial characterization, the underlying principle in determining the class membership of a database object is to consider the non- spatial attributes as well as the attributes of objects in its neighborhood.

Trends in Data Mining: The trends in data mining are as follows:

Applications exploration: The data mining **applications** available in the market till date helped businesses to gain competitive advantage. But as data mining is becoming popular, it is continuously being used for other applications as well. Besides, the exploration of data mining for businesses is continuously expanding, as E-commerce is becoming the mainstream element of the retail industry.

Scientific data mining methods: Data mining systems are expected to be able to handle huge amounts of data efficiently and if possible interactively. Since a large amount of data is continuously being collected, scalable algorithms for individual and integrated data mining functions has become essential.

Integration with data store systems: Database systems, data warehouse systems, and the World Wide Web have become the most widely used information processing systems. Thus, it is necessary that data mining can be smoothly integrated into such an information processing environment.

Standardization of data mining language: A data mining system that strictly adheres to a standard language or other standardization efforts will facilitate the systematic development of data mining solutions, improve interoperability among different data mining systems and function, and promote the use of data mining system in industry.

Data trends: The most crucial trend is the explosion of digital data over last few years. It has been observed that the amount of data has grown between 6 to 10 orders of magnitude. It is always better to use data mining techniques that can automate this data, and extract meaningful knowledge out of it rather than dumping it in archive files without any practical use.

Hardware trends: Application of data mining algorithms calls for statistically insensitive computations on large sets of data. The increasing memory and processing speed of computers have enabled the application of such algorithms on large sets of data.

Network trends: With the Internet connectivity, it has now become possible to correlate distributed datasets using current algorithms and techniques. Besides this, new protocols and algorithms are being developed that can facilitate distributed data mining using current and next generation networks.

Scientific computing trends: Data mining and the KDD process form a crucial part of linking theory, experiment, and simulation that result in large datasets.

Architecture of Data mining: Data mining is the process of discovering interesting and useful knowledge from large amounts of data warehouses. The architecture of a typical data mining system as given by jiawci Han and Michelin Kamber in their book Data mining concepts and Techniques (2001), consists of the following components as shown in fig.

Database or data warehouse or other information repository: This includes one or more **databases,** a data warehouse, or any other **information repository.** Data cleaning and data integration techniques have to be applied to the data before the data mining algorithm can be applied on it. However, in case the **Data** mining algorithms are applied directly on the: Data warehouses, then cleansing and integration functions may be skipped because a data **warehouse** already contains integrated and cleansed data.

Database or data warehouse server: The database or data warehouse server is used to fetch the relevant data, based on the user's **data mining request.**

Knowledge base: It contains the domain knowledge that is used to guide the search or used for evaluation of the interestingness of resulting patterns. Such knowledge can include concept hierarchies, user beliefs, and interestingness of resulting patterns. Such knowledge can include concept hierarchies, user beliefs, interestingness thresholds and the metadata.

Data mining engine: It consists of a set of functionalities for tasks such as a characterization, association, classification, cluster analysis, and evolution analysis.

Pattern Evaluation module: This component employs interestingness measures and other threshold values and interacts with the **data mining** engine so as to retrieve only the interesting results. For enhanced efficiency in the performance of **data mining** algorithms, the evaluation of pattern interestingness must be process so that the search can be confined to search only the interesting patterns discovered.

Graphical user interface (GUI): This module interacts with the users and the data mining system thereby allowing the user to use the data mining system either by specifying a query or a task like that of characterization, association, classification,

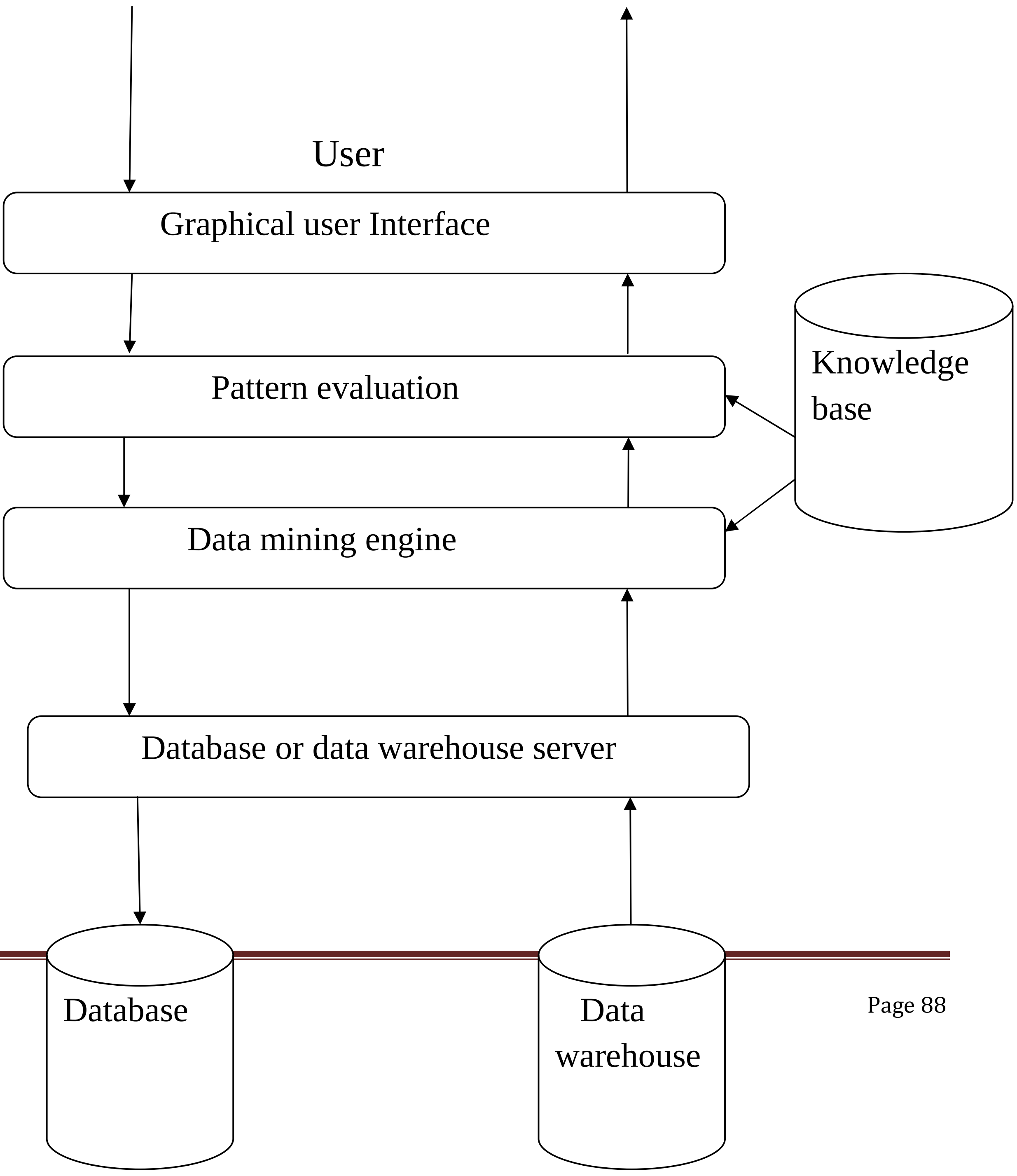


Fig Architecture of a Data mining system

Cluster analysis, and evolution analysis.

With GUI, users can also provide extra information to guide the search process. In addition, the GUI component enables the end-users to browse the database, evaluate the mined patterns, and visualize the discovered patterns in different formats.

A Data Mining Query Language (DMQL)

- Motivation
- A DMQL can provide the ability to support ad-hoc and interactive data mining
- By providing a standardized language like SQL
- Hope to achieve a similar effect like that SQL has on relational database
- Foundation for system development and evolution
- Facilitate information exchange, technology transfer, commercialization and wide acceptance

Syntax for DMQL

Syntax for
specification of
task
relevant data
the kind of knowledge to be mined
concept hierarchy specification
interestingness
measure
pattern presentation and visualization
Putting it all together —a DMQL query

Designing Graphical User Interfaces based on a data mining query language

- What tasks should be considered in the design GUIs based on a data mining query language?
- Data collection and data mining query composition
- Presentation of discovered patterns
- Hierarchy specification and manipulation
- Manipulation of data mining primitives
- Interactive multilevel mining
- Other miscellaneous information

Data Mining Primitives

Five primitives for specification of a data mining task

kind of knowledge to be
mined
background
knowledge
interestingness
measures
knowledge presentation and visualization techniques to be used for displaying the discovered patterns.

Visualization:

Different backgrounds/usages may require different forms of representation

- E.g. rules, tables, crosstabs, pie/bar chart etc.
- Concept hierarchy is also important
- Discovered knowledge might be more understandable when represented at high level of abstraction
- Interactive drill up/down, pivoting, slicing and dicing provide different perspective to data
- Different kinds of knowledge require different representation: association, classification, clustering, etc

Data Generalization and Summarization-based Characterization

Data generalization.

A process which abstracts a large set of task-relevant data in a database from a low conceptual levels to higher ones.

Attribute-Oriented Induction

Proposed in 1989 (KDD '89 workshop)

Not confined to categorical data nor particular measures.

How it is done?

Collect the task-relevant data(*initial relation*) using a relational database query

Perform generalization by attribute removal or attribute generalization.

Apply aggregation by merging identical, generalized tuples and accumulating their respective counts.

Interactive presentation with users.

Basic Principles of Attribute-Oriented Induction

Data focusing: task-relevant data, including dimensions, and the result is the *initial relation*.

Attribute-removal: remove attribute *A* if there is a large set of distinct values for *A* but (1) there is no generalization operator on *A*, or (2) *A*'s higher level concepts are expressed in terms of other attributes.

Attribute-generalization: If there is a large set of distinct values for *A*, and there exists a set of generalization operators on *A*, then select an operator and generalize *A*.

Attribute-threshold control: typical 2-8, specified/default.

Generalized relation threshold control: control the final relation/rule size.

Example

Describe general characteristics of graduate students in the Big-University database

use Big_University_DB

mine characteristics as “Science_Students”

in relevance to name, gender, major, birth_place, birth_date, residence, phone#, gpa

from student

where status in “graduate”

Corresponding SQL statement:

Select name, gender, major, birth_place, birth_date, residence, phone#, gpa

from student

where status in {“Msc”, “MBA”, “PhD” }

Mining Class Comparisons: Comparison: Comparing two or more classes.

Method: Partition the set of relevant data into the target class and the contrasting class(es)

Generalize both classes to the same high level concepts

Compare tuples with the same high level descriptions

Present for every tuple its description and two measures:

support - distribution within single class

comparison - distribution between classes

Highlight the tuples with strong discriminant features

Relevance Analysis:

Find attributes (features) which best distinguish different classes.

Example: Analytical comparison

Task

Compare graduate and undergraduate students using discriminant rule.

DMQL query

```
use Big_University_DB
mine comparison as "grad_vs_undergrad_students"
in relevance to name, gender, major, birth_place, birth_date, residence, phone#, gpa
for "graduate_students"
where status in "graduate"
versus "undergraduate_students"
where status in "undergraduate"
analyze count%
from student
```

Example: Analytical Characterization

Task

Mine general characteristics describing graduate students using analytical characterization

Given

attributes *name, gender, major, birth_place, birth_date, phone#,* and *gpa*

Gen(a) = concept hierarchies on a_i

U_i = attribute analytical thresholds for a_i

T_i = attribute generalization thresholds for a_i

R = attribute relevance threshold

1. Data collection

   target and contrasting classes

2. Attribute relevance analysis

   remove attributes *name, gender, major, phone#*

3. Synchronous generalization

controlled by user-specified dimension thresholds

prime target and contrasting class(es) relations/cuboids

Birth_country	**Age_range**	**Gpa**	**Count%**
Canada	20-25	Good	5.53%
Canada	25-30	Good	2.32%
Canada	Over_30	Very_good	5.86%
…	…	…	…
Other	Over_30	Excellent	4.68%

generalized relation for the target class: Graduate students

4. Drill down, roll up and other OLAP operations on target and contrasting classes to adjust levels of abstractions of resulting description

5. Presentation

as generalized relations, crosstabs, bar charts, pie charts, or rules

contrasting measures to reflect comparison between target and contrasting classes

e.g. count%