

CS614 Final Term Notes.2

Data Quality Management Process:

1. Establish Data Quality Management Environment Given the existing barriers to communication, establishing the data quality environment involves participation of both functional users and information system administrators.

2. Scope Data Quality Projects & Develop Implementation Plan

Defines the scope of the project and defines the level of analysis that will be the most beneficial for the project under question. Draft an initial plan that addresses the following elements.

- Task Summary: Project goals, scope, and potential benefits
- Task Description: Describe data quality analysis tasks
- Project Approach: Summarize tasks and tools used to provide a baseline of existing data quality
- Schedule: Identify task start, completion dates, and project milestones
- Resources: Identify resources required to complete the data quality assessment.

Include costs connected with tools acquisition, labor hours (by labor category), training, travel, and other direct and indirect costs

3. Implement Data Quality Projects (Define, Measure, Analyze, Improve):

A data quality analysis project consists of four activities. The data quality project manager performs these activities with input from the functional users of the data, system developers, and database administrators of the legacy and target database systems.

- Define: Identify functional user data quality requirements and establish data quality metrics.
- Measure: Measure conformance to current business rules and develop exception reports.
- Analyze: Verify, validate, and assess poor data quality causes. Define improvement opportunities.
- Improve: Select/prioritize data quality improvement opportunities.

4. Evaluate Data Quality Management Methods

The last step in the Data Quality Management process is to evaluate and assess progress made in implementing data quality initiatives and/or projects.

The House of Quality:

The House of Quality and is one of many techniques of Quality Function Deployment, which can briefly be defined as “a system for translating customer requirements into appropriate company requirements”.

How to improve Data Quality?

The four categories of Data Quality Improvement

- Process: Improve the functional processes used to create, manage, access, and use data.
- System: Software, hardware, and telecommunication changes can improve data quality.
- Policy & Procedure: Resolve conflicts in existing policies and procedures and institutionalize behaviors that promote good data quality.
- Data Design: Improve the overall data design and use data standards. Adding primary key constraints, indexes, unique key constraints, triggers, stored functions and procedures, controlling administration of user privileges, enforcing security features, and referential integrity constraints can improve database design.

	Management understanding	Quality organ. status	Problem handling	Cost of quality % of sales	Company attitude
Stage-1 Uncertainty	No comprehension of quality.	Quality Dept. part of manufacturing or engineering.	Fire fighting approach.	Reported unknown, actual high.	No organized activity.
Stage-2 Awakening	Recognize quality management may be of value.	Quality Dept. still part of manufacturing or engineering.	Short term solutions, no long term approach.	Reported as low, actually high.	All talk no real action.
Stage-3 Enlightenment	Become supportive and helpful.	Quality Dept. reports to top management.	Problems faced and solved orderly.	Reported as medium actually on the higher side.	Identifying and resolving problems
Stage-4 Wisdom	Understand absolutes of quality management.	Senior quality manager position.	Identified at an early stage.	Reported as about medium actually about medium.	Defect prevention is a routine.
Stage-5 Certainty	Quality management essential part of company policy.	Quality manager on board of directors.	Identified and resolved at an early stage.	Reported as low, actually is low.	Know why there are problems.

The quality management maturity grid:

Stage 1: Uncertainty

Stage-1 is denial. People don’t think data quality is an issue. It is the responsibility of the people collecting the data. It is their problem. There is no data quality inspection; nobody actually cares about the data quality in the processes.

Stage 2: Awakening

People think that data quality management does have a value. Management talks about it,

For More Visit

www.VUAnswer.com

but does nothing. This is like all talk and no action stage.

Stage 3: Enlightenment

A light goes on, may be there is a problem. The quality department starts reporting instead of hiding the problems in the closet. You actually start to measure data quality and are reasonably close.

Stage 4: Wisdom

Value of quality is seen as something that is needful and meaningful. Data quality starts coming on the performance evaluation of employees i.e. what are they doing for data quality. So rather than fixing problems AFTER they have occurred, you start taking steps BEFORE the problems occur

Stage 5: Certainty

The approach is “I know about the quality of my data” and/or “I am measuring it for real”. Quality improvement is NOT something that is done on ad-hoc basis, but is part of everything. People know why they are doing it, and why they do not have data quality problems.

Misconceptions on Data Quality

1. You Can Fix Data

Fixing implies that there was something wrong with the original data, and you can fix it once and be done with it. In reality, the problem may have been not with the data itself, but rather in the way it was used. When you manage data you manage data quality. It's an ongoing process. Data cleansing is not the answer to data quality issues.

2. Data Quality is an IT Problem

Data quality is a company problem that costs a business in many ways. Although IT can help address the problem of data quality, the business has to own the data and the business processes that create or use it. The business has to define the metrics for data quality - its completeness, consistency, relevancy and timeliness. **For More Visit**

3. The Problem is in the Data Sources or Data Entry

www.VUAnswer.com

Data entry or operational systems are often blamed for data quality problems. Although incorrectly entered or missing data is a problem, it is far from the only data quality problem.

4. The Data Warehouse will provide a Single Version of the Truth

If everyone uses the data warehouse's data exclusively and it meets your data quality metrics then it is the single version of the truth.

5.It came from the legacy system so must be correct

Another misconception is, “It came from the production transaction processing environment, so it must be correct.” However the reality is, the elements required for decision support are often quite different than those required for transaction processing.

Parallel execution is sometimes called parallelism. Simply expressed, parallelism is the idea of breaking down a task so that, instead of one process doing all of the work in a query, many processes do part of the work at the same time. Parallel execution dramatically reduces response time for data-intensive operations on large databases typically associated with Decision Support Systems (DSS) and data warehouses. You can also implement parallel execution on certain types of online transaction processing (OLTP) and hybrid systems.

When to parallelize?

Useful for operations that access significant amounts of data.

Useful for operations that can be implemented independent of each other “Divide-&-Conquer”

Parallel execution improves processing for:

- Large table scans and joins
- Creation of large indexes
- Partitioned index scans
- Bulk inserts, updates, and deletes
- Aggregations and copying

For More Visit

www.VUAnswer.com

Parallelism can be exploited, if there is...

- Symmetric multi-processors (SMP), clusters, or Massively Parallel (MPP) systems AND
- Sufficient I/O bandwidth AND
- Underutilized or intermittently used CPUs (for example, systems where CPU usage is typically less than 30%) AND
- Sufficient memory to support additional memory-intensive processes such

as sorts, hashing, and I/O buffers

Word of caution/Limitations

Parallelism can reduce system performance on over-utilized systems or systems with small I/O bandwidth.

Speed-Up

More resources means proportionally less time for given amount of data.

Scale-Up

If resources increased in proportion to increase in data size, time is constant

Amdahl's law

In computer architecture, Amdahl's law is a formula which gives the theoretical speedup in latency of the execution of a task at fixed workload that can be expected of a system whose resources are improved. The **important** point is to always remember **Amdahl's law**. This states that the overall performance improvement gained by optimizing a single part of a system is limited by the fraction of time that the improved part is actually used. ... Don't guess at where the performance hot spots are

$$S \leq \frac{1}{f + (1 - f)/N}$$

f is the fraction of the problem that must be computed sequentially

N is the number of processors

Parallelization OLTP Vs. DSS

There is a big difference.

DSS

Parallelization of a SINGLE query

OLTP

Parallelization of MULTIPLE queries

Or Batch updates in parallel

Brief Intro to Parallel Processing

- Parallel Hardware Architectures
- Symmetric Multi Processing (SMP)
- Distributed Memory or Massively Parallel Processing (MPP)

- Non-uniform Memory Access (NUMA)
 - Parallel Software Architectures
 - Shared Memory
 - Shared Disk
 - Shared Nothing
 - Types of parallelism
 - Data Parallelism
- Spatial Parallelism

For More Visit

www.VUAnswer.com

NUMA

In NUMA systems, some memory can be accessed more quickly than other parts, and thus called as Non-Uniform Memory Access. This term is generally used to describe a shared-memory computer containing a hierarchy of memories, with different access times for each level in the hierarchy

SMP (Symmetric Multiprocessing) is a computer architecture that provides fast performance by making multiple CPUs available to complete individual processes simultaneously (multiprocessing). Unlike asymmetrical processing, any idle processor can be assigned any task, and additional CPUs can be added to improve performance and handle increased work load.

What does Virtual Shared Memory (VSM) mean?

Virtual shared memory (VSM) is a technique through which multiple processors within a distributed computing architecture are provided with an abstract shared memory.

Advantages:

A benefit of the shared disk approach is it provides a high level of fault tolerance with all data remaining accessible even if there is only one surviving node.

Disadvantages:

Maintaining locking consistency over all nodes can become a problem in large clusters.

So I can have multiple database instances each with it's own database buffer cache all accessing the same set of disk blocks. This is a shared everything disk architecture. Now if multiple database instances are accessing the same tables and same blocks, then some locking mechanism will be required to maintain database buffer cash coherency.

A **shared-nothing architecture** (SN) is a distributed computing **architecture** in which each update request is satisfied by a single node (processor/memory/storage unit). The intent is to eliminate contention among nodes. Nodes do not **share** (independently access) memory or storage

Advantages

This works fine in environments where the data ownership by nodes changes relatively

infrequently. The typical reasons for changes in ownership are either database reorganizations or node failures.

There is no overhead of maintaining data locking across the cluster

Disadvantages

The data availability depends on the status of the nodes. Should all but one system fail, then only a small subset of the data is available.

Data partitioning is a way of dividing your tables etc. across multiple servers according to some set of rules. However, this requires a good understanding of the application and its data access patterns (which may change).

Table 1. Summary of the Capabilities of Shared-disk versus Shared-nothing Clustering

Shared-Disk	Shared-Nothing
Quick adaptability to changing workloads	Can exploit simpler, cheaper hardware
High availability	Works well in a high-volume, read-write environment
Dynamic load balancing	Fixed load balancing based upon the partitioning scheme
Performs best in a heavy read environment	Almost unlimited scalability
Data need not be partitioned	Data is partitioned across the cluster
Messaging overhead limits total number of nodes	Depends on partitioning, data shipping can kill scalability

Shared Nothing RDBMS & Partitioning

Shared nothing RDBMS architecture requires a static partitioning of each table in the database.

How do you perform the partitioning?

- Hash partitioning
- Key range partitioning.
- List partitioning.
- Round-Robin
- Combinations (Range-Hash & Range-List)

Range partitioning maps data to partitions based on ranges of partition key values that you establish for each partition.

Hash partitioning

maps data to partitions based on a hashing algorithm that database product applies to a partitioning key identified by the DBA.

List partitioning enables you to explicitly control how rows map to partitions. You do this

by specifying a list of discrete values for the partitioning column in the description for each partition.

Round robin is just like distributing a deck of cards, such that each player gets almost the same number of cards. Hence it is “fair”.

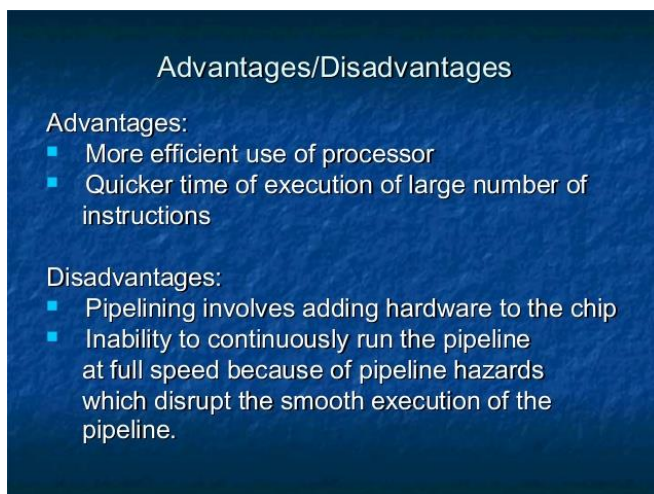
Data Parallelism: Concept

- Parallel execution of a single data manipulation task across multiple partitions of data.
- Partitions static or dynamic
- Tasks executed almost-independently across partitions.
- “Query coordinator” must coordinate between the independently executing processes.

Data Parallelism: Ensuring Speed-UP

To get a speed-up of N with N partitions, it must be ensured that:

- There are enough computing resources.
- Query-coordinator is very fast as compared to query servers.
- Work done in each partition almost same to avoid performance bottlenecks.
- Same number of records in each partition would not suffice.
- Need to have uniform distribution of records w.r.t. filter criterion across partitions.



Advantages/Disadvantages

Advantages:

- More efficient use of processor
- Quicker time of execution of large number of instructions

Disadvantages:

- Pipelining involves adding hardware to the chip
- Inability to continuously run the pipeline at full speed because of pipeline hazards which disrupt the smooth execution of the pipeline.

Partitioning & Queries

- Full Table Scan
- Point Queries:

- Range Queries
- Round Robin
- Hash Partitioning
- Range Partitioning

Handling Skew in Range-Partitioning

- Sort
- Construct the partition vector
- Duplicate entries or imbalances

Barriers to Linear Speedup & Scale-up

- Amdahl' Law
- Startup
- Interference
- Skew

Why do we need indexing in database?

Indexes are used to quickly locate data without having to search every row in a **database** table every time a **database** table is accessed. **Indexes** can be created using one or more columns of a **database** table, providing the basis for both rapid random lookups and efficient access of ordered records.

Indexing is a data structure technique to efficiently retrieve records from the **database** files based on some attributes on which the **indexing** has been done. **Indexing in database** systems is similar to what we see in books. ... Clustering Index – Clustering index is defined on an ordered data file.

Conventional indexes

- Basic Types:
 - Sparse
 - Dense
 - Multi-level (or B-Tree)
- Primary Index vs. Secondary Indexes

Indexing is defined based on its indexing attributes. Indexing can be of the following types –

- **Primary Index** – Primary index is defined on an ordered data file. The data file is ordered on a **key field**. The key field is generally the primary key of the relation.
- **Secondary Index** – Secondary index may be generated from a field which is a candidate key and has a unique value in every record, or a non-key with duplicate values.

- **Clustering Index** – Clustering index is defined on an ordered data file. The data file is ordered on a non-key field.

Ordered Indexing is of two types –

- Dense Index
- Sparse Index

Dense Index

In dense index, there is an index record for every search key value in the database. This makes searching faster but requires more space to store index records itself. Index records contain search key value and a pointer to the actual record on the disk.

Sparse Index

In sparse index, index records are not created for every search key. An index record here contains a search key and an actual pointer to the data on the disk. To search a record, we first proceed by index record and reach at the actual location of the data. If the data we are looking for is not where we directly reach by following the index, then the system starts sequential search until the desired data is found.

Multilevel Index

Index records comprise search-key values and data pointers. Multilevel index is stored on the disk along with the actual database files. As the size of the database grows, so does the size of the indices. There is an immense need to keep the index records in the main memory so as to speed up the search operations. If single-level index is used, then a large size index cannot be kept in memory which leads to multiple disk accesses.

B⁺ Tree

A B⁺ tree is a balanced binary search tree that follows a multi-level index format. The leaf nodes of a B⁺ tree denote actual data pointers. B⁺ tree ensures that all leaf nodes remain at the same height, thus balanced. Additionally, the leaf nodes are linked using a link list; therefore, a B⁺ tree can support random access as well as sequential access.

For More Visit

www.VUAnswer.com

Advantages

- In indexed sequential access file, sequential file and random file access is possible.
- It accesses the records very fast if the index table is properly organized.
- The records can be inserted in the middle of the file.
- It provides quick access for sequential and direct processing.
- It reduces the degree of the sequential search.

Disadvantages

- Indexed sequential access file requires unique keys and periodic reorganization.
- Indexed sequential access file takes longer time to search the index for the data access or retrieval.
- It requires more storage space.
- It is expensive because it requires special software.
- It is less efficient in the use of storage space as compared to other file organizations.

Dense Index: Every key in the data file is represented in the index file

Pro:

A dense index, if fits in the memory, costs only one disk I/O access to locate a record given a key

Con:

A dense index, if too big and doesn't fit into the memory, will be expensive when used to find a record given its key

Sparse Index: Adv & Dis Adv

- Store first value in each block in the sequential file and a pointer to the block.
- Uses even less space than dense index, but the block has to be searched, even for unsuccessful searches.
- Time (I/Os) logarithmic in the number of blocks used by the index.

B-tree Indexing: Limitations

If a table is large and there are fewer unique values.

Capitalization is not programmatically enforced (meaning case-sensitivity does matter and "FLASHMAN" is different from "Flashman").

Outcome varies with inter-character spaces.

A noun spelled differently will result in different results.

Insertion can be very expensive.

B-tree vs. Hash Indexes

Indexing (using B-trees) good for range searches, e.g.:

```
SELECT * FROM R WHERE A > 5
```

Hashing good for match based searches, e.g.:

```
SELECT * FROM R WHERE A = 5
```

Primary Indexing: Criterion

- Primary index selection criteria:
 - o Common join and retrieval key.
 - o Can be Unique UPI or Non-unique NUPI.
 - o Limits on NUPI.
 - o Only one primary index per table (for hash-based file system).

Special Index Structures

- Inverted index
- Bit map index
- Cluster index
- Join indexes

In computer science, an **inverted index** (also referred to as a postings file or **inverted file**) is a **database index** storing a mapping from content, such as words or numbers, to its locations in a table, or in a document or a set of documents (named in contrast to a forward **index**, which maps from documents to content).

A **bitmap** index is a special kind of **database** index that uses **bitmaps**. **Bitmap indexes** have traditionally been considered to work well for low-cardinality columns, which have a modest number of distinct values, either absolutely, or relative to the number of records that contain the data.

Bitmap Index: Adv.

- Very low storage space.
- Reduction in I/O, just using index.
- Counts & Joins
- Low level bit operations.

Bitmapped indexes can provide very impressive performance speedups; execution times of certain queries may improve by several orders of magnitude. A significant advantage of bitmapped indexes is that multiple bitmapped indexes can be

used to evaluate the conditions on a single table.

Bitmap Index: Dis. Adv.

- Locking of many rows
- Low cardinality
- Keyword parsing
- Difficult to maintain - need reorganization when relation sizes change (new bitmaps)

Row locking: A potential drawback of bitmaps involves locking. Because a page in a bitmap contains references to so many rows, changes to a single row inhibit concurrent access for all other referenced rows in the index on that page.

Low cardinality: Bitmap indexes create tables that contain a cell for each row times each

possible value (the product of the number of rows times the number of unique values).

Therefore, a bitmap is practical only for low- cardinality columns that divide the data into a small number of categories, such as "M/F", "T/F", or "Y/N" values.

Keyword parsing: Bitmap indexes can parse multiple values in a column into separate keywords. For example, the title "Marry had a little lamb" could be retrieved by entering the word "Marry" or "lamb" or a combination. Although this keyword parsing and lookup capability is extremely useful, textual fields tend to contain high-cardinality data (a large number of values) and therefore are not a good choice for bitmap indexes.

The big advantage of a cluster index is that all the rows with the same cluster index value will be placed into adjacent locations in a small number of data blocks.

The **SQL** Joins clause is used to combine records from two or more tables in a database. A JOIN is a means for combining fields from two tables by using values common to each.

- Nested loop join
- Sort Merge Join
- Hash based join
- Etc.

1. **Nested Loop Join:** In **nested loop join** algorithm, for each tuple in outer relation we have to compare it with all the tuples in the inner relation then only the next tuple of outer relation is considered. All pairs of tuples which satisfies the condition are added in the result of the **join**.

A nested

loop join involves the following steps:

1. The optimizer determines the major table (i.e. Table_A) and designates it as the outer table. Table_A is accessed once. If the outer table has no useful indexes, a full table scan is performed. If an index can reduce I/O costs, the index is used to locate the rows.
 2. The other table is designated as the inner table or Table_B. Table_B is accessed once for each qualifying row (or tuple) in Table_A.
 3. For every row in the outer table, DBMS accesses all the rows in the inner table.
- The outer loop is for every row in outer table and the inner loop is for every row in the inner table.

For More Visit

www.VUAnswer.com

Nested-Loop Join: Variants

1. Naive nested-loop join
2. Index nested-loop join
3. Temporary index nested-loop join

The **sort-merge join** (also known as **merge join**) is a **join** algorithm and is used in the implementation of a relational database management system. The basic problem of a **join** algorithm is to find, for each distinct value of the **join** attribute, the set of tuples in each relation which display that value.

Sort-Merge Join: Note

Very fast.

Sorting can be expensive.

Presorted data can be obtained from existing B-tree.

The **Hash Join** algorithm is used to perform the natural **join** or equi **join** operations. The concept behind the **Hash join** algorithm is to partition the tuples of each given relation into sets. The partition is done on the basis of the same **hash** value on the **join** attributes. The **hash** function provides the **hash** value.

The optimizer uses a hash join to join two tables if they are joined using an equijoin and if either of the following conditions are true:

- A large amount of data needs to be joined.
- A large portion of the table needs to be joined.

Cost of Hash-Join

- In partitioning phase, read + write both operations requires $2(M+N)$ I/Os.
- In matching phase, read both requires $M+N$ I/Os.

Data mining is an interdisciplinary subfield of computer science and statistics with an overall goal to extract information (with intelligent methods) from a **data** set and transform the information into a comprehensible structure for further use.

What is Data Mining?: Formal

- Knowledge Discovery in Databases (KDD).
- Data mining digs out valuable non-trivial information from large multidimensional apparently unrelated data bases (sets).
- It's the integration of business knowledge, people, information, algorithms, statistics and computing technology.
- Finding useful hidden patterns and relationships in data.

In the given definition, the 5 key words are;

For More Visit

www.VUAnswer.com

Nontrivial

By nontrivial, we mean that some search or inference is involved; that is, it is not a straightforward computation of predefined quantities like computing the average value of a set of numbers.

Value

The term *value* refers to the importance of discovered hidden patterns to the user in terms of its usability, validity, benefit and understandability. Data mining is a way to intelligently probing large databases to find exactly where the *value* resides.

Multidimensional

By multidimensional we mean a database designed as a multidimensional hypercube with one axis per dimension.

Unrelated

Humans often lack the ability to comprehend and manage the immense amount of available and unrelated data.

Business Knowledge

The domain business processes must be known apriority before applying defaming techniques.

People

Human involvement in the data mining process is crucial in sense that value of patterns is well known to the user. Since data mining focuses on “*unknown unkowns*”, people factor plays a key role in directing data mining probe in a direction that ultimately ends in

something that is previously unknown, novel, and above all of value.

Data mining consists of algorithms for extracting useful patterns from huge data. Their goal is to make prediction or/and give description. Prediction involves using some variables to predict unknown values (e.g. future values) of other variables while description focuses on finding interpretable patterns describing the data.

Statistics

Data Mining uses statistical algorithms to discover patterns and regularities (or “knowledge”) in data.

Computing Technology

Data mining is an inter disciplinary approach having knowledge from different fields such as databases, statistics, high performance computing, machine learning, visualization and mathematics to automatically extract concepts, and to determine interrelations and

patterns of interest from large databases.

Motivation: Why Data Mining?

- Holy Grail - Informed Decision Making
- Lots of Data are Being Collected
 - Business - Transactions, Web logs, GPS-track, ...
 - Science - Remote sensing, Micro-array gene expression data, ...
- Challenges:
 - Volume (data) >> number of human analysts
 - Some automation needed
- Limitations of Relational Database
 - Can not predict future! (questions about items not in the database!)
 - Ex. Predict tomorrow's weather or credit-worthiness of a new customer
 - Can not compute transitive closure and more complex questions
 - Ex. What are natural groups of customers?
 - Ex. Which subsets of items are bought together?
- Data Mining may help!
 - Provide better and customized insights for business
 - Help scientists for hypothesis generation

For More Visit

www.VUAnswer.com

Claude Shannon's info. theory

Claude Shannon's theory states that as the volume increases the information content decreases and vice versa.

Data mining VS Data warehousing

Data warehouse	Data mining
Process of storing data in order in given dataset	Process of finding pattern in given dataset.
Data warehousing is the process of extracting and storing data to allow easier reporting.	Data mining is the use of pattern recognition logic to identify trends within a sample data set and extrapolate this information against the larger data pool
The tools in data warehousing are designed to extract data and store it in a method designed to provide enhanced system performance	A typical use of data mining is to create targeted marketing programs, identify financial fraud,
Helps in identifying the certain data in a collection of data	Helps in figuring out a certain pattern of a data or a cluster of data

- Data Mining (Knowledge-driven exploration)
 - Query formulation problem.
 - Visualize and understand of a large data set.
 - Data growth rate too high to be handled manually.
- Data Warehouses (Data-driven exploration):
 - Querying summaries of transactions, etc. Decision support
- Traditional Database (Transactions):
 - Querying data in well-defined processes. Reliable storage

Now lets discuss something about what is included in DM and what is not. First we will discuss what DM is.

Decision Trees (DT): Decision trees consist of dividing a given data set into groups based on some criteria or rule. The final structure looks like an inverted tree, hence the technique called DT.

Clustering: It is one of the most important Dm techniques; we will discuss it in detail in coming lectures. As a brief for understanding it involves the grouping of data items without taking any human parametric input.

Genetic Algorithms: These are based on the principle survival of the fittest. In these techniques, a model is formed to solve problems having multiple options and many values.

DM application areas or techniques

CLASSIFICATION

□ Classification consists of examining the properties of a newly presented observation and assigning it to a predefined class.

ESTIMATION

As opposed to discrete outcome of classification i.e. YES or NO, deals with continuous valued outcomes

PREDICTION

Same as classification or estimation except records are classified according to some predicted future behavior or estimated value.

MARKET BASKET ANALYSIS

Determining which things go together, e.g. items in a shopping cart at a super market.

CLUSTERING

Task of segmenting a heterogeneous population into a number of more homogenous sub-groups or clusters.

DESCRIPTION

Describe what is going on in a complicated database so as to increase our understanding.

The metrics we use for comparison of DM techniques are;

Accuracy: Accuracy is the measure of correctness of your model e.g. in classification we have two data sets, training and test sets. A classification model is built based on the data properties and relationships in training data.

Speed: In previous lectures we discussed the term “Need for Speed”. Yes speed is a crucial aspect of Dm techniques. Speed refers to the time complexity. If a technique has $O(n)$ and another has $O(n \log n)$ time complexities then which is better? Yes $O(n)$ is better.

Robustness: It is the ability of the technique to work accurately even in conditions of noisy or dirty data. Missing data is a reality and presence of noise also true. So a technique is better if it can run smoothly even in stress conditions i.e. with noisy and missing data.

Scalability: As we mentioned in our initial lectures that the main motivation for data warehousing is to deal huge amounts of data. So scaling is very important, which is the

ability of the method to work efficiently even when the data size is huge.

Interpretability: It refers to the level of understanding and insight that is provided by the method. As we discussed in clustering one of the complex and difficult tasks is the cluster analysis.

There are two main types of unsupervised clustering.

1. One-way Clustering-means that when you clustered a data matrix, you used all the attributes. In this technique a similarity matrix is constructed, and then clustering is performed on rows. A cluster also exists in the data matrix for each corresponding cluster in the similarity matrix.

2. Two-way Clustering/Biclustering-here rows and columns are simultaneously clustered. No any sort of similarity or dissimilarity matrix is constructed. Biclustering gives a local view of your data set while one-way clustering gives a global view. It is possible that you first take global view of your data by performing one-way clustering and if any cluster of interest is found then you perform two-way clustering to get more details. Thus both the methods complement each other.

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. ... A **cluster** refers to a collection of data points aggregated together because of certain similarities. You'll **define** a target number **k**, which refers to the number of centroids you need in the dataset.

Weakness

- Applicable only when *mean* is defined, then what about categorical data?
- Need to specify *k*, the *number* of clusters, in advance
- Unable to handle noisy data and *outliers*

For More Visit

Not suitable to discover clusters with non-convex shapes

www.VUAnswer.com

At the end of the lecture there are some comments about k-means:

1. K-means is a fairly fast technique and normally when terminates , then clusters formed are fairly good.
2. It can only work for data sets where there is the concept of mean (the answer to the question posed in a few slides back). If data is non numeric such as likes dislikes, gender, eyes color etc. then how to calculate means. So this is the first problem with the technique.
3. Another problem or limitation of the technique is that you have to specify the

number of cluster in advance.

4. The third limitation is that it is not a robust technique as it not works well in presence of noise.

Current data

warehouse development methods can fall within three basic groups: data-driven, goal-driven and user-driven.

Implementation strategies

- Top down approach
- Bottom Up approach

Development methodologies

- Waterfall model
- Spiral model
- RAD Model
- Structured Methodology
- Data Driven
- Goal Driven
- User Driven

Implementation Strategies

Top Down & Bottom Up approach: A Top Down approach is generally useful for projects where the technology is mature and well understood, as well as where the business problems that must be solved are clear and well understood. A Bottom Up approach is useful, on the other hand, in making technology assessments and is a good technique for organizations that are not leading edge technology implementers. This approach is used when the business objectives that are to be met by the data warehouse are unclear, or when the current or proposed business process will be affected by the data warehouse.

Development Methodologies T

The waterfall model is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks. The approach is typical for certain areas of engineering design.

Spiral Model: The model is a sequence of waterfall models which corresponds to a risk oriented iterative enhancement, and recognizes that requirements are not always available and clear when the system is first implemented. Since designing and building a data warehouse is an iterative process, the spiral method is one of the development methodologies of choice.

RAD: Rapid Application Development (RAD) is an iterative model consisting of stages like scope, analyze, design, construct, test, implement, and review. It is much better suited to the development of a data warehouse because of its iterative nature and fast iterations.

User requirements are sometimes difficult to establish because business analysts are too

1. Assemble a small, very bright team of database programmers, hardware

technicians, designers, quality assurance technicians, documentation and decision support specialists, and a single manager.

2. Define and involve a small "focus group" consisting of users (both novice and experienced) and managers (both line and upper). These are the people who will provide the feedback necessary to drive the prototyping cycle. Listen to them carefully.
3. Generate a user's manual and user interface first. These will prove to be amazing in terms of user feedback and requirements specification.
4. Use tools specifically designed for rapid prototyping. Stay away from C, C++, COBOL, SQL, etc. Instead use the visual development tools included with the database.
5. Remember a prototype is NOT the final application.

Structured Development: When a project has more than 10 people involved or when multiple companies are performing the development, a more structured development management approach is required.

Data-Driven Methodologies: Bill Inmon, the founder of data warehousing argues that data warehouse environments are data driven, in comparison to classical systems, which have a requirement driven development lifecycle. According to Inmon, requirements are the last thing to be considered in the decision support development lifecycle. Goal-Driven Methodologies: In order to derive the initial data warehouse structure,

Böhnlein and Ulbrich-vom Ende have presented a four-stage approach based on the SOM (Semantic Object Model) process modeling technique. The first stage determines goals and services the company provides to its customers. In the second step, the business process is analyzed by applying the SOM interaction schema that highlights the customers and their transactions with the process under study. In third step, sequences of transactions are transformed into sequences of existing dependencies that refer to information systems. The last step identifies measures and dimensions, by enforcing (information request) transactions, from existing dependencies.

User-Driven Methodologies: Westerman describes an approach that was developed at Wal-Mart and has its main focus on implementing business strategy. The methodology assumes that the company goal is the same for everyone and the entire company will therefore be pursuing the same direction.

WHERE DO YOU START?

The majority of successful data warehouses have started with a clear understanding of a business problem and the user requirements for information analysis. It is strongly recommended that the team assembled to create a data warehouse be comprised of IT professionals and business users.

What specific Problems the DWH will solve?

Write down all the problems. The problems should be precise, clearly stated and testable i.e. success criteria is known or can easily be specified. Make sure to get user and management feedback by publicizing these written problems.

The cyclic model consists of 5 major steps described as follows

1. Design: It involves the development of robust star-schema-based dimensional data models from both available data and user requirements. It is thought that the best data warehousing practitioners even work with available organizational data and incompletely expressed user requirements. Key activities in the phase typically include end-user interview cycles, source system cataloguing, definition of key performance indicators and other critical business definitions, and logical and physical schema design tasks which feed the next phase of the model directly.
2. Prototype: In this step a working model of a data warehouse or data mart design, suitable for actual use, is deployed for a select group of end users. The prototyping purpose shifts, as the design team moves design-prototype-design sub-cycle. Primary objective is to constrain and /or reframe end-user requirements by showing them precisely what they had asked for in the previous iteration. As difference between stated needs and actual needs narrows down over iterations the prototyping shifts towards gaining commitment to the project at hand from opinion leaders in the end-user communities to the design, and soliciting their assistance in gaining similar commitment.
3. Deploy: The step includes traditional IT system deployment activities like formalization of user authenticated prototype for actual production use, document development, and training etc. Deployment involves two separate deployments (i) prototype deployment into a production –test environment (ii) Stress- and performance tested production configuration deployment into an actual production environment. The phase also contains the most important and often neglected component of documentation.

Lack of documentation may stall system operations as management people can not manage what they don't know. Also, it may ultimately be used for educating the end users, prior to roll out.

4. Operation: The phase includes data warehouse/mart daily maintenance and management activities. The operations are performed to maintain data delivery services and access tools, and manage ETL processes that keep the data warehouse/mart current with respect to the authoritative source system.

5. Enhancement: The step involves modifications of physical technological components, operations and management processes (ETL etc.) and logical schema diagrams in response to changing business requirements. In situations of discontinuous changes, enhancement moves back into the fundamental design phase.

Business Dimensional Lifecycle: The Road Map Ralph Kimball's Approach

Implementing a data warehouse requires tightly integrated activities. As we discussed earlier, there are different DWH implementation strategies, we will be following Kimball's Approach. Kimball is considered as an authority in the DWH field, and his goal driven approach is a result of decades of practical experience. This presentation is an overview of a data warehouse project lifecycle, based on this approach, from inception through ongoing maintenance, identifying best practices at each step, as well as potential vulnerabilities. It is believed that everyone on the project team, including the business analyst, architect, database designer, data stager, and analytic application developer,

needs a high-level understanding of the complete lifecycle of a data warehouse. The business dimensional lifecycle framework, as shown in Figure 33.1, is depicted as a road map, that is extremely useful if we're about to embark on the unfamiliar journey of data warehousing. The Kimball's iterative data warehouse development approach drew on decades of experience to develop the business dimensional lifecycle. The name was because it reinforced several of key tenets for successful data warehousing. First and foremost, data warehouse projects must focus on the needs of the business. Second, the data presented to the business users must be dimensional. Finally while data warehousing is an ongoing process, each implementation project should have a finite cycle with a specific beginning and end. Ongoing project management serves as a foundation to keep the remainder of the lifecycle on track.

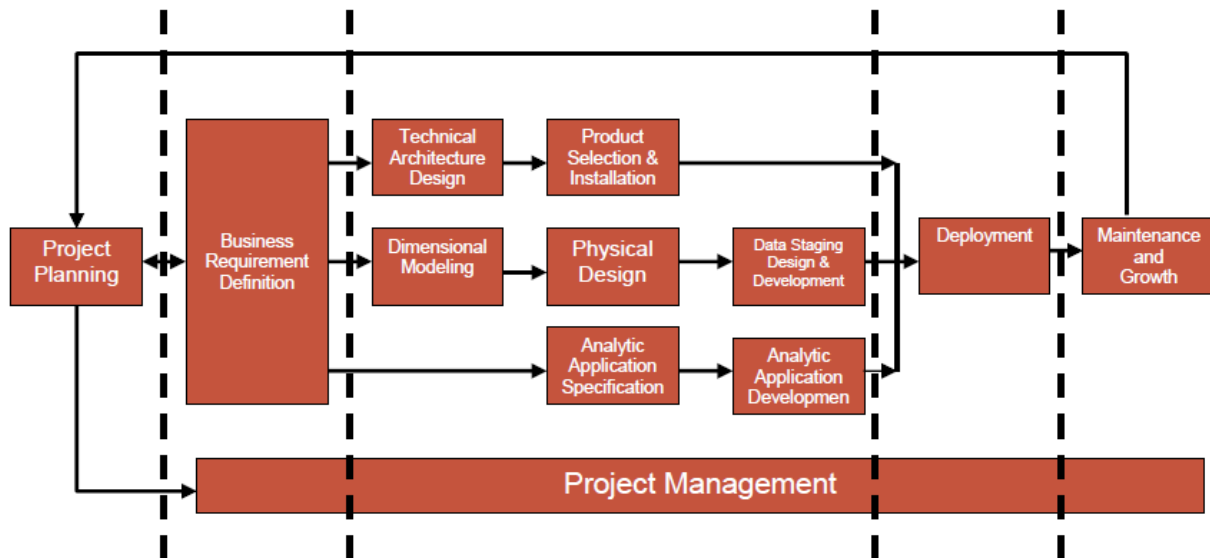


Figure -33.1: Business Dimensional Lifecycle (Kimball's Approach)

DWH Lifecycle: Key steps

1. Project Planning
2. Business Requirements Definition
3. Parallel Tracks
 - 3.1 Lifecycle Technology Track
 - 3.1.1 Technical Architecture
 - 3.1.2 Product Selection
 - 3.2 Lifecycle Data Track
 - 3.2.1 Dimensional Modeling
 - 3.2.2 Physical Design
 - 3.2.3 Data Staging design and development
 - 3.3 Lifecycle Analytic Applications Track
 - 3.3.1 Analytic application specification
 - 3.3.2 Analytic application development
4. Deployment
5. Maintenance

Top 10-Common Mistakes to Avoid

- Mistake 1: Not interacting directly with the end users.

- ❑ Mistake 2: Promising an ambitious data mart as the first deliverable.
- ❑ Mistake 3: Never freezing the requirements i.e. being an accommodating person.
- ❑ Mistake 4: Working without senior executives in loop, waiting to include them after a significant success.
- ❑ Mistake 5: Doing a very comprehensive and detailed first analysis to do the DWH right the very first time.
- ❑ Mistake 6: Assuming the business users will develop their own “killer application” on their own.
- ❑ Mistake 7: Training users on the detailed features of the tool using dummy data and consider it a success.
- ❑ Mistake 8: Isolating the IT support people from the end or business users.
- ❑ Mistake 9: After DWH is finished, holding a planning and communications meeting with end users.
- ❑ Mistake 10: Shying away from operational source systems people, assuming they are too busy.

For More Visit

Top 7-Key Steps for a smooth DWH implementation... www.VUAnswer.com

- ❑ Step-1: Assigning a full-time project manager, or doing it yourself full-time.
- ❑ Step-2: Consider handing-off project management.
- ❑ Step-3: During user interview don't go after answers, let the answers come to you.
- ❑ Step-4: Assigning responsibilities to oversee and ensure continuity.
- ❑ Step-5: Accept the “fact” that DWH will require many iterations before it is ready.
- ❑ Step-6: Assign significant resources for ETL.
- ❑ Step-7: Be a diplomat NOT a technologist.

Web Warehousing: An introduction

Internet population stands at around a billion users.

Exponential growth rate of web and size. Indexable web is more than 11.5 billion pages (Jan 2005).

Adoption rate of intranet & extranet warehouse having similar growth rate

Web enabled versions of tools are available, but adoption differ.

Media is diverse and other than only data i.e. text, image, audio etc.

39.2 Web searching

Web is large, actually very large.

To make it useful must be able to find the page(s) of interest/relevance.

How can the search be successful?

Three major types of searches, as follows:

1. Keyword-based search
2. Querying deep Web sources
3. Random surfing

Drawbacks of traditional web searches

1. Limited to keyword based matching.
2. Can not distinguish between the contexts in which a link is used.
3. Coupling of files has to be done manually.

Why web warehousing-Reason no. 1?

- Web data is unstructured and dynamic, keyword search is insufficient.
- To increase usage of web must make it more comprehensible.
- Data Mining is required for understanding the web.
- Data mining used to rank and find high quality pages, thus making most of search time.

Why web warehousing-Reason no. 2?

Web log contains wealth of information, as it is a key touch point.

Every customer interaction is recorded.

Data Warehousing (CS614)

© Copyright Virtual University of Pakistan 352

Success of email or other marketing campaign can be measured by integrating with other operational systems.

Common measurements are:

- Number of visitors
- Number of sessions

- Most requested pages
- Robot activity

Why web warehousing-Reason no. 3?

- Shift from distribution platform to a general communication platform.
- New uses from e-government to e-commerce and new forms of art etc. between different levels of society.
- Thus web is worthy to be archived to be used for several other projects.
- Such as snapshots of preserving time.

Figure 39.3 illustrates the steps during a client server interaction on the WWW. Each of the activity or action has been shown with a sequence . Lets briefly look at these sequence of actions.

Action 1

User tries to access the site using its URL.

Action 2

The server returns the requested page, websitepage.html. Once the document is entirely retrieved, the visitor's browser scans for references to other Web documents that it must fulfill before its work is completed. In order to speed up the response time, most browsers will execute these consequential actions in parallel, typically with up to 4 or more HTTP requests being serviced concurrently.

Action 3

The visitor's browser finds a reference to a logo image that is located at Website. Com. The browser issues a second request to the server, and the server responds by returning the specified image.

Action 4

The browser continues to the next reference for another image from Banner-ad.com. The browser makes this request, and the server at Banner-ad.com interprets a request for the image in a special way. Rather than immediately sending back an image, the banner-ad server first issues a cookie request to the visitor's browser requesting the contents of any cookie that might have been placed previously in the visitor's PC by Banner-ad.com.

There are two options based on the response of the cookie request;

Option I: Cookie Request Fulfilled: The banner-ad Web site retrieves this cookie and uses the contents as a key to determine which banner ad the visitor should receive. This decision is based on the visitor's interests or on previous ads. Once the banner-ad server makes a determination of the optimal ad, it returns the selected image to the visitor. The banner-ad server then logs which ad it has placed along with the date and the clickstream data from the visitor's request.

Option II: No Cookie Found: If the banner-ad server had not found its own cookie, it would have sent a new persistent cookie to the visitor's browser for future reference, sent a random banner ad, and started a history in its database of interactions with the visitor's browser.

Referrer: The HTTP request from the visitor's browser to the banner-ad server carried with it a key piece of information known as the referrer. The referrer is the URL of the agent responsible for placing the link on the page. In the example the referrer is Website.com/websitepage.html. Because Banner-ad.com now knows who the referrer was, it can credit Website.com for having placed an advertisement on a browser window.

Action 5

In the original HTML document, websitepage.html had a hidden field that contained a request to retrieve a specific document from Profiler.com. When this request reached the profiler server, Profiler.com immediately tried to find its cookie in the visitor's browser. This cookie would contain a user ID placed previously by the profiler that is used to identify the visitor and serves as a key to personal information contained in the profiler's database.

Action 6

The profiler might either return its profile data to the visitor's browser to be sent back to the initial Web site or send a real-time notification to the referrer, Website.com, via an alternative path alerting Website.com that the visitor is currently logged onto Website.com and viewing a specific page. This information also could be returned to the HTML document to be returned to the referrer as part of a query string the next time an HTTP request is sent to Website.com.

Low level web traffic Analysis

For More Visit

www.VUAnswer.com

- Is anyone coming?
- If people are coming, identify which pages they are viewing.
- Once you rank your content, you can tailor it to satisfy your visitors.
- Detailed analysis helps increase traffic, such which sites are referring visitors.

High level web traffic analysis

- Combining your Web site traffic data with other data sources to find which banner ad campaigns generated the most revenue vs. just the most visitors.
- Help calculate ROI on specific online marketing campaigns.
- Help get detailed information about your online customers and prospects.

Where does traffic info. come from?

1. Log files.
2. Cookies.
3. Network traffic.
4. Page tagging.
5. ISP (Internet Service Provider)
6. Others

To track traffic on a web site

http://www.alex.com/data/details/traffic_details?q=&url=http://www.domain.com

The principal sources of web traffic are as follows:

1. Log files.
2. Cookies.
3. Network traffic.
4. Page tagging.
5. ISP

Web log file formats

Format of web log dependent on many factors, such as:

- Web server
- Application
- Configuration options

Our example proxy log data file contained following fields

For More Visit

www.VUAnswer.com

i. Timestamp (date in Table 39.1)

ii. Elapsed Time

This is the time that transaction busied the cache. This time is given in milliseconds. For the request where there was a cache-miss this time is minimal, where the request engaged the cache, this time is considerable.

iii. Client Address (host in Table 39.1)

iv. Log Tag

This field tells the result of the cache operation.

v. HTTP Code (status in Table 39.1)

vi. Size (bytes in Table 39.1)

vii. Request Method (request in Table 39.1)

This is the method which client used to initiate the request and be dealt with the proper treatment on the server side.

viii. URL

This is the URL which was request by the client. There can be many variations in the representation, start and termination of the URL.

ix. User Ident (ident in Table 39.1)

This field is used to identify the requesting user on the network.

x. Hierarchy Data

This field provides the hierarchy data of the request from the same client in the current request.

xi. Content Type

This field contains the type of data which was requested. The values in this field are the standard MIME types which describe the data contents.

Clickstream

Clickstream is every page event recorded by each of the company's Web servers

Web-intensive businesses

Although most exciting, at the same time it can be the most difficult and most frustrating.

Not JUST another data source.

Issues of Clickstream Data

Clickstream data has many issues.

1. Identifying the Visitor Origin
2. Identifying the Session
3. Identifying the Visitor
4. Proxy Servers
5. Browser Caches

Identifying the Session

- Web-centric data warehouse applications require every visitor session (visit) to have its own unique identity
- The basic protocol for the World Wide Web, HTTP, is stateless so session identity must be established in some other way.
- There are several ways to do this
- Using Time-contiguous Log Entries
- Using Transient Cookies
- Using HTTP's secure sockets layer (SSL)
- Using session ID Ping-pong
- Using Persistent Cookies

The basic protocol for the World Wide Web, HTTP, is stateless-that is, it lacks the concept of a session. There are no intrinsic login or logout actions built into the HTTP, so session identity must be established in some other way. There are several ways to do this

1. Using Time-contiguous Log Entries
2. Using Transient Cookies
3. Using HTTP's secure sockets layer (SSL)
4. Using session ID Ping-pong
5. Using Persistent Cookies

1- Using Time-contiguous Log Entries

- A session can be consolidated by collecting time-contiguous log entries from the same host (Internet Protocol, or IP, address).
- Limitations

- The method breaks down for visitors from large ISPs
- Different IP addresses
- Browsers that are behind some firewalls.

Proxy servers can introduce three problems, as illustrated in Figure in next slide

- i. A proxy may deliver outdated content. Although Web pages can include tags that tell proxy servers whether or not the content may be cached and when content expires, these tags often are omitted by Webmasters or ignored by proxy servers.
- ii. Proxies may satisfy a content request without properly notifying the originating server that the request has been served by the proxy. When a proxy handles a request, convention dictates that it should forward a message that indicates that a proxy response has been made to the intended server, but this is not reliable. As a consequence, the Web warehouse may miss key events that are otherwise required to make sense of the events that comprise a browser/Web site session.
- iii. If the visitor has come through a proxy, the Web site will not know who made the page request unless a cookie is present.

Forward Proxy

The type of proxy we are referring to in this discussion is called a forward proxy. It is outside of our control because it belongs to a networking company or an ISP.

Reverse Proxy

Another type of proxy server, called a reverse proxy, can be placed in front of our enterprise's Web servers to help them offload requests for frequently accessed content. This kind of proxy is entirely within our control and usually presents no impediment to Web warehouse data collection. It should be able to supply the same kind of log information as that produced by a Web server.

Browser caches

- Most browsers store a copy of recently retrieved objects in a local object cache in the PC's file system.
- A visitor may return to a page already in his or her local browser cache
- We can never be certain that we have a full map of the visitor's actions.
- We can attempt to force the browser to always obtain objects from a server rather than from cache
- A similar uncertainty can be introduced when a visitor opens multiple browser windows to the same Web site

A web warehouse can be implemented with some slight modifications in the development lifecycle, such as:

- Teaming with new partners.
- Beware of slow string functions.
- Large data so min loading.

The data warehouse manager

faces the following challenges in the clickstream data mart development life cycle:

- Unlike most data mart implementations, in which primarily DBAs and data administrators own the knowledge of the source data, the data warehouse team will need to work closely with webmasters and programmers. These groups may be the only resources who can supply the crucial information necessary for finding the key elements the business requires.

For More Visit

www.VUAnswer.com

- The types of substring and instr functions required to process the Web logs during the ETL process are terribly inefficient in most DBMSs and ETL tools. Because of the daunting size, you must optimize the clickstream ETL process for performance. Inefficient ETL processing will not reach completion in the allocated load window.
- The vast amount of data makes it necessary to limit the revisiting or reloading of logs to an absolute minimum. Therefore, the ETL process must discover, transform, and load new pages, page types, Web servers, and customers during the first pass of the Web log. Additionally, the process must also handle new robots, exclusions, and altered business rules gracefully.