Lecture 24: Need for Speed: Parallelism

Learning Goals

- When to parallelize?
- Understand Scalability
- Speed-Up & Amdahl's Law
- Parallelization OLTP Vs. DSS

Data warehouses often contain large tables and require techniques both for managing these large tables and for providing good query performance across these large tables.

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.

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. An example of this is when four processes handle four different quarters in a year instead of one process handling all four quarters by itself. The improvement in performance can be quite high. In this case, data corresponding to each quarter will be a partition, a smaller and more manageable unit of an index or table.

24.1 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:

Past

Size ▪ Large table scans and joins
Size ▪ Creation of large indexes
D&C ▪ Partitioned index scans
Size ▪ Bulk inserts, updates, and deletes
D&C ▪ Aggregations and copying

Every operation can not be parallelized, there are some preconditions and one of them being that the operations to be parallelized can be implemented independent of each other. This means that there will be no interference between the operations while they are being parallelized. So what do we gain out of parallelization; many things which can be divided into two such as with reference to size and with reference to divide and conquer. Note that divide and conquer means that we should be able to divide the problem and then solve it and then compile the results i.e. conquer. For example in case of scanning a large table every row has to be checked, in such a case this can be done in parallel thus reducing the overall time. There can be and are many examples too.

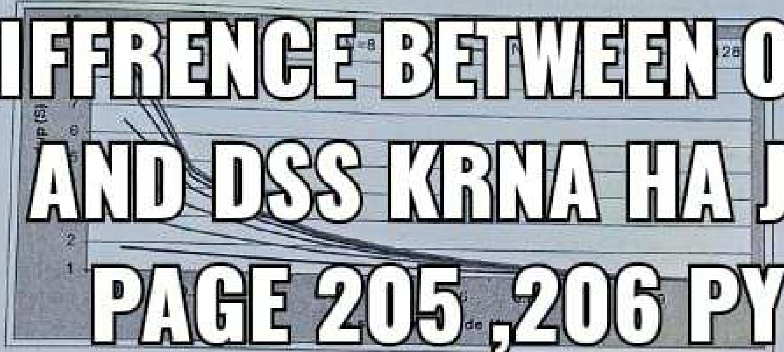
As *f* approaches 0, S approaches N

Example-1: f = 5% and N = 100 then S = 16.8 }
Example-2: f = 10% and N = 200 then S = 9.56 } Not 1:1 Ratio

The processing for parallel tasks can be spread across multiple processors. So, if 90% of our processing can be parallelized and 10% must be serial we can speed up the process by a factor of 3.08 when we use four independent processors for the parallel portion. This example also assumes 0 overhead and "perfect" parallelism. Thus, a database query that would run for 10 minutes when processed serially would, in this example, run in 2.63 minutes (10/3.08) when the parallel tasks were executed on four independent processors.

As you can see, if we increase the overhead for parallel processing or decrease the amount of parallelism available to the processors, the time it takes to complete the query will increase according to the formula above.

Amdahl's Law: Limits of parallelization



DIFFRENCE BETWEEN OLTP AND DSS KRNA HA JO PAGE 205 ,206 PY MOJOUD HA

For less than 80% paral[illegible]lly

At 90% parallelism, 128 processors give performance of less than 10 processors.

Figure-24.4: Amdahl's Law

As we can see in the graphical representation of Amdahl's Law as shown in Figure24.4, the realized benefit of additional processors is significantly diminishes as the amount of sequential processing increases. In this graph, the vertical axis is the system speed-up. As the overall percentage of sequential processing increases (with a corresponding decrease in parallel processing) the relative effectiveness (utilization) of additional processors decreases. At some point, the cost of an additional processor actually exceeds the incremental benefit.

24.3 Parallelization OLTP Vs. DSS

Past

There is a big difference.

DSS

ONLY SPEED-UP AMDAHL'S LAW KA FORMULA YAD KRNA HA

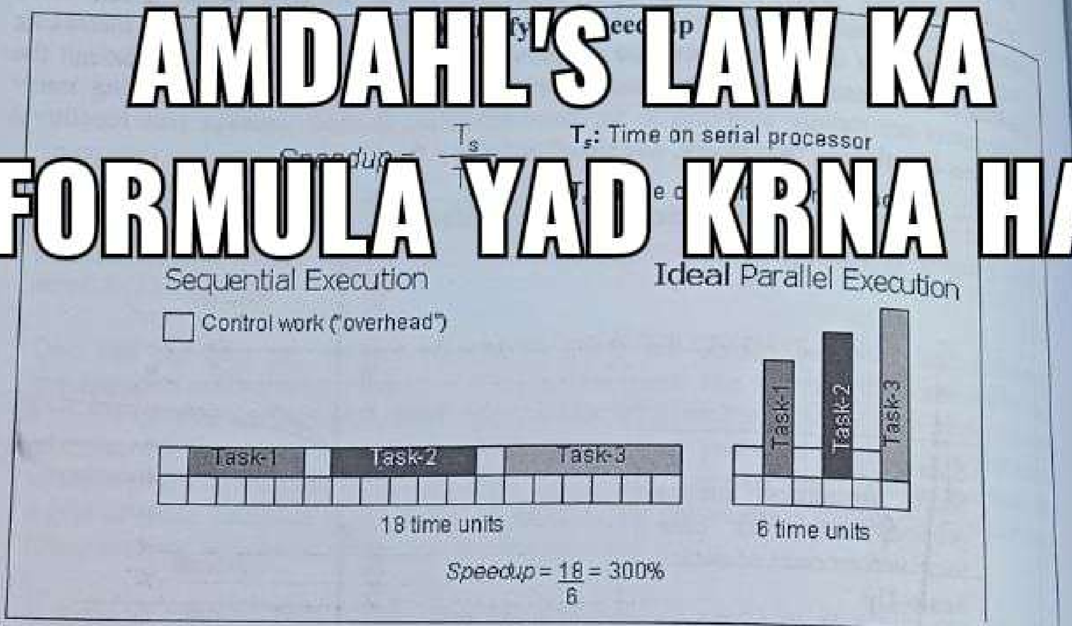


Figure-24.3: Quantifying Speed-up

Data dependencies between different phases of computation introduce synchronization requirements that force sequential execution. Moreover, there is a wide range of capabilities available in commercially implemented software in regard to the level of granularity at which parallelism can be exploited.

As shown in figures 24.2 and 24.3, , the goal of ideal parallel execution is to completely parallelize those parts of a computation that are not constrained by data dependencies. The smaller the portion of the program that must be executed sequentially (s), the greater the scalability of the computation.

24.2 Speed-Up & Amdahl's Law

Reveals maximum expected speedup from parallel algorithms given the proportion of task that must be computed sequentially. It gives the speedup S as

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

Past

f is the fraction of the problem that must be computed sequentially
N is the number of processors

Parallelization of a SINGLE query

OLTP

Parallelization of MULTIPLE queries
Or Batch updates in parallel

During business hours, most OLTP systems should probably not use parallel execution. During off-hours, however, parallel execution can effectively process high-volume batch operations. For example, a bank can use parallelized batch programs to perform the millions of updates required to apply interest to accounts.

Asima

The most common example of using parallel execution is for DSS. Complex queries, such as those involving joins or searches of very large tables, are often best run in parallel.

24.4 Brief Intro to Parallel Processing

Past

- Parallel Hardware Architectures
  - Symmetric Multi Processing (SMP)
  - Distributed Memory or Massively Parallel Processing (MPP)
  - Non-uniform Memory Access (NUMA)

- Parallel Software Architectures
  - Shared Memory
  - Shard Disk
  - Shared Nothing

  Shared everything

- Types of parallelism
  - Data Parallelism

Spatial Parallelism

24.5 NUMA

Paper

Usually on an SMP system, all memory beyond the caches costs an equal amount to reach for each CPU. 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. The distinguishing feature is that the time required to access memory locations is not uniform i.e. access times to different locations can be different.

Symmetrical Multi Processing (SMP)

shared disk approach, transactions running on any instance can directly read or modify any part of the database. Such systems require the use of inter-node communication to synchronize update activities performed from multiple nodes. When two or more nodes contend for the same data block, the node that has a lock on the data has to act on the data and release the lock, before the other nodes can access the same data block.

Shared Disk RDBMS Architecture :

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. Because if a data block is in the buffer cache of P1 and the same data block is in the buffer cash of P2 then there is a problem. So there is something called distributed lock management that has to be implemented to maintain coherency between the databases buffer cashes across these different database instances.

And that leads to a lot of performance issues in shared everything databases because every time when lock management is performed, it becomes serial processing. There are two approaches to solving this problem i.e. hardware mechanism and a software mechanism. In the hardware mechanism, a coupling facility is used. The coupling facility manages all the locks to control coherency in the database buffer cash. Another vendor took a different approach; because it sells a more portable database that runs on any platform, therefore, it couldn't rely on special hardware. Therefore, there is a software lock management system called the distributed lock manager, which is used to mange across different database instances. In most cases both techniques must guarantee that there is never incoherency of data blocks across database instances.

Shared Nothing RDBMS Architecture

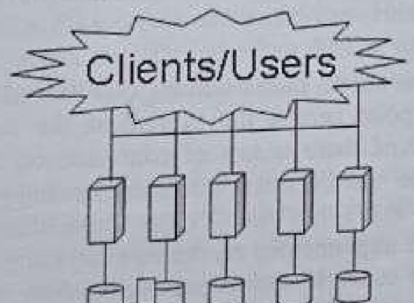


Figure-24.8: Shared Nothing RDBMS Architecture

In case of shared nothing architecture as shown in Figure 24.8, there is no lock contention and therefore any time you have locking problem then you also have

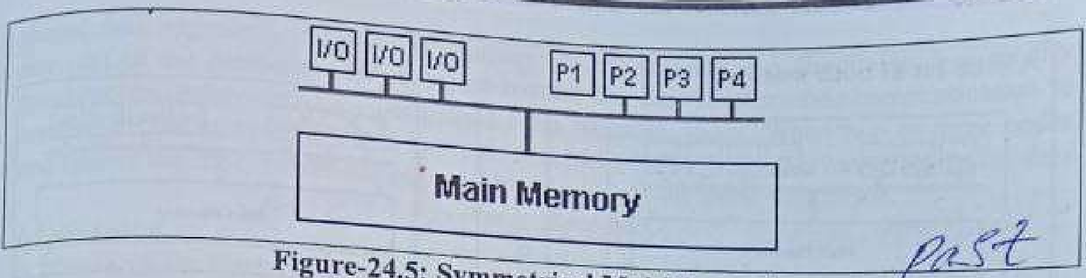


Figure-24.5: Symmetrical Multi Processing

Past

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. A variety of specialized operating systems and hardware arrangements are available to support SMP. Specific applications can benefit from SMP if the code allows multithreading.

SMP uses a single operating system and shares common memory and disk input/output resources. Both UNIX and Windows NT support SMP.

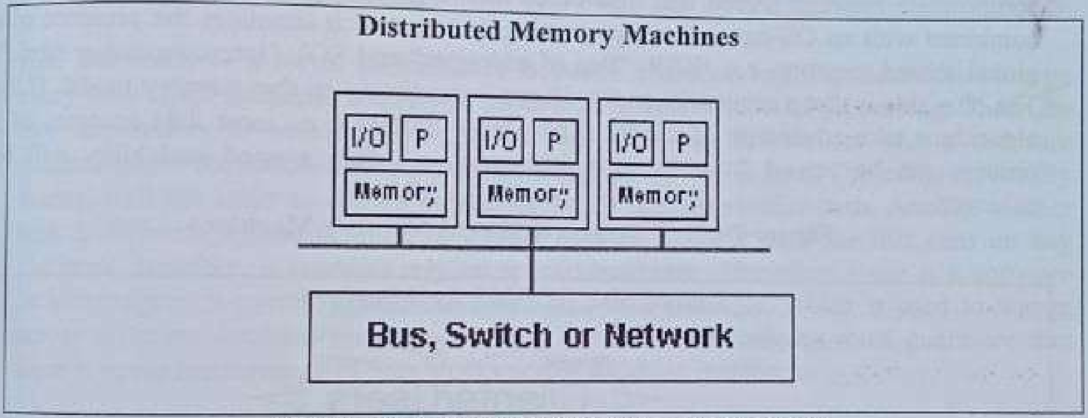


Figure-24.6: Distributed Memory Machines

Special-purpose multiprocessing hardware comes in two flavors i.e. shared memory and distributed memory machines. In a shared-memory machine, all processors have access to a common main memory. In a distributed-memory machine, each processor has its own main memory, and the processors are connected through a sophisticated interconnection network. A collection of networked PCs is also a kind of distributed-memory parallel machine.

Communication between processors is an important prerequisite for all but the most trivial parallel processing tasks (thus bandwidth can become a bottleneck). In a shared-memory machine, a processor can simply write a value into a particular memory location, and all other processors can read this value. In a distributed-memory machine, exchanging values of variables involves explicit communication over the network, thus need for a high speed interconnection network.

Distributed Shared Memory Machines

serialization issue. The idea is that each database table partition in the database instances e.g. The customer table and Order table exist on all the database instances. So the parallelism is really already built in. There is never any confusion and there is never any locking problem. If we join two tables with the same partitioning column, and the partitioning was performed using hash partitioning, then that is a local join and is very efficient.

A request will be made to the "owning" database instance to send the desired columns (projection) from qualifying rows of the source table when data is required by one database instance that is partitioned to a different database instance. In the function shipping approach, the column and row filtering is performed locally by the "owning" database instance so that the amount of information communicated to requesting database instance is only what is required. This is different than in shared disk database architectures where full data blocks (no filtering) are shipped to the requesting database instance. Shared Nothing RDBMS Architecture:

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).

24.6 Shared disk Vs. Shared Nothing RDBMS

- Important note: Do not confuse RDBMS architecture with hardware architecture.
- Shared nothing databases can run on shared everything (SMP or NUMA) hardware.

Shared disk databases can run on shared nothing (MPP) hardware.
Now a very important point here is not to confuse the software architecture with the hardware architecture. And there is lots of confusion on that point. People think that shared nothing database architectures can only be implemented on shared nothing hardware architectures, that's not true. People think that shared everything database architectures can only be implemented on shared everything hardware architecture, which is not true either. So for example shared nothing database like Teradata can work on an SMP machine, that's not a problem. Because the software is shared nothing that does not mean that the hardware has to be shared nothing. SMP is symmetric multi processing, shared memory, shared bus structure, shared I/O system and so on, it is not a problem. In fact Informix is a shared nothing database architecture which was originally implemented on a shared everything hardware architecture, which is an SMP machine.

Amdhal's Calculation:

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

- Skew

As I said in the first lecture on parallelism, intuitively we feel that as the number of processors increase, the speedup should also increase. Thus theoretically there should be a linear speedup. However, in reality this is not the case. The biggest hurled is the Amdahl's law which we have discussed in detail. The other problem is the startup cost, it takes a while for the system to get started and that time when amortized over the entire processing time results in a less than a linear speedup. Then is the interference among different processes or the dependencies among the processes or some operations within the problem itself such as empting of the pipeline, this result in degradation of performance.

Finally the skew in the data: Parallelization is based on the premise that there is a full utilization of the processors and all of them are bust most or all of the time. However, if there is a skew in the partitioning of data i.e. a non-uniform distribution, then some of the processors will be working while other will be idle. And the processor that takes the most time (which has the most data too) will become the bottleneck.

Parallel Sorting:

(i) Scaning, Partitioning in Parallel.

(ii) After Partioning data is available perform local Sorting (iii) Problem: skew "hot spot" (iv) Solution: Sample the data at Start to determine Partionting Points.

Pipeling speed —

$$S = \frac{NT}{T+(N-1)\frac{T}{M}}$$

So shared disk databases some times called shared everything databases are also run on shared nothing hardware. Oracle is a shared everything database architecture and the original implementation of the parallel query feature was written on machine called the N-Cube machine. N-Cube machine is an MPP machine that is a shared nothing hardware architecture but that has a shared everything database. In order to do that, a special layer of software called the VSD (Virtual shared disk) is used. So when an I/O request is made, in a shared everything database environment like ORACLE, every instance of the database can see every data block. If it is a shared nothing environment how do I see other data blocks? With a basically an I/O device driver which looks at the I/O request and if it is local, it says ok access it locally, if it is remote, it ships the I/O request to another Oracle instance it does the I/O for me and then it ships the data back.

24.7 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)

Past

Range partitioning maps data to partitions based on ranges of partition key values that you establish for each partition. It is the most common type of partitioning and is often used with dates. For example, you might want to partition sales data into monthly partitions.

Most shared nothing RDBMS products use a hashing function to define the static partitions because this technique will yield an even distribution of data as long as the hashing key is relatively well distributed for the table to be partitioned. Hash partitioning maps data to partitions based on a hashing algorithm that database product applies to a partitioning key identified by the DBA. The hashing algorithm evenly distributes rows among partitions, giving partitions approximately the same size. Hash partitioning is the ideal method for distributing data evenly across devices. Hash partitioning is a good and easy-to-use alternative to range partitioning when data is not historical and there is no obvious column or column list where logical range partition pruning can be advantageous.

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. This is different from range partitioning, where a range of values is associated with a partition and with hash partitioning, where you have no control of the row-to-partition mapping. The advantage of list partitioning is that you can group and organize unordered and unrelated sets of data in a natural way.

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".

implications? But these implications should never affect the answer that I will get. Should only affect how long it takes to get the answer.

Think about an index as an analogy with a library. Think of the library as a collection of books, and there is huge room full of books. If you are looking for a particular book, unless the books are sorted by the title of the book, the one way you can look for that book is that you start out with the left hand wall and then you start looking at each and every book until you find the book you are interested in or you don't. So on average if there are *n* books in the library how long dose it take? It will take *O (n)* or *n*/2 (on average). So it is an *O (n)* algorithm for accessing the data.

Now consider using the card catalog. The card catalog is organized in many different ways. By author, by topic by title, by what ever you want. Each of these have a different index. I have index on author, I have index on title and I have index on topic etc. so it takes me a little bit of extra time to go to the catalog first, but the catalog is sorted, so I can very quickly find the author that I am looking for because it is sorted and then I look at the card and know exactly which shelf to find the book. The catalog has some numbering system, which acts as a pointer, and points to shelf and the row on the shelf where the book is located. So I have to do a little bit of extra work to go to the catalog but then it takes me directly to the book I am interested in, that is what Indexing is. Indexing is just a card catalog to get access to the data i.e. efficient access to data. There is no actual information that I really want to get from the card catalog about the book. There is no data it is just an efficient way of accessing the book that I want. So the index in a database is just like that.

Past

Indexing Goal

Look at as few blocks as possible to find the matching record(s)

The point of using an index is to increase the speed and efficiency of searches of the database. Without some sort of index, a user's query must sequentially scan the database, finding the records matching the parameters in the WHERE clause.

26.3 Conventional indexes

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

Past

and so forth. Another table might identify products with a unique product ID and fields identifying price, brand, inventory level, etc.

Now, imagine a simple query: how many customers are there in Karachi? The simplest way to do this is called a table scan: merely read every row sequentially and count the number of times you find a CITY field containing KHI. In a large database, of course, this will take a very long time. In this case, a simple index will greatly speed up data access by indexing the customer table by the city field. Of course, this means that a Database Administrator (DBA) has to create the index and keep it up to date.

Now, consider a more complex query: how many customers in Karachi made calls during April ? Now, the database must perform what is called a table join: it must look at both the CALLS table, qualifying the CALL DATE field in terms of a date range, and the Customer table, looking at the CITY field. Table joins are also very slow. In this case, a DBA might tune the database by creating a summary table containing customer calls by month and, again, the DBA would have to keep it up to date.

An even more complex query would be to ask for the records of customers in Karachi that what are the average call charges per customer in Karachi in April? This is a multidimensional query that requires building a multi-part key for another index or one might call for an aggregated result that is it may require all of the above operations plus the mathematical averaging of the $AMOUNT fields from all the qualifying Calls. Again, a DBA can create and update a summary table to handle this kind of query.

Need For Indexing: I/O Bottleneck (short) — Past

(Throwing more hardware at the problem doesn't really help, either. Expensive and multi-processing servers can certainly accelerate the CPU-intensive parts of the process, but the bottom line of database access is disk access, so the process is I/O bound and I/O doesn't scale as fast as CPU power. You can get around this by putting the entire database into main memory, but the cost of RAM for a multi-gigabyte database is likely to be higher than the server itself! Therefore we index.)

Although DBAs can overcome any given set of query problems by tuning, creating indexes, summary tables, and multiple data marts, or forbidding certain kinds of queries, they must know in advance what queries users want to make and would be useful, which requires domain-specific knowledge they often don't have. While 80% of database queries are repetitive and can be optimized, 80% of the ROI from database information comes from the 20% of queries that are not repetitive. The result is a loss of business or competitive advantage because of the inability to access the data in corporate databases in a timely fashion.

26.2 Indexing Concept

Indexing is purely a physical database concept and has nothing to do with the logical model. In fact an index should be completely invisible to someone who is doing the programming on the database. You should never access the index directly, it is the optimizer that should chose to use the index when it is appropriate to do so. The reality is of course is that the programmer generally will know what are the performance

In this case, normally only one key per data block is kept. A sparse index uses less space at the expense of somewhat more time to find a record given its key.
What happens when record 35 is inserted?

Sparse Index: Adv & Dis Adv Paper

- 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.

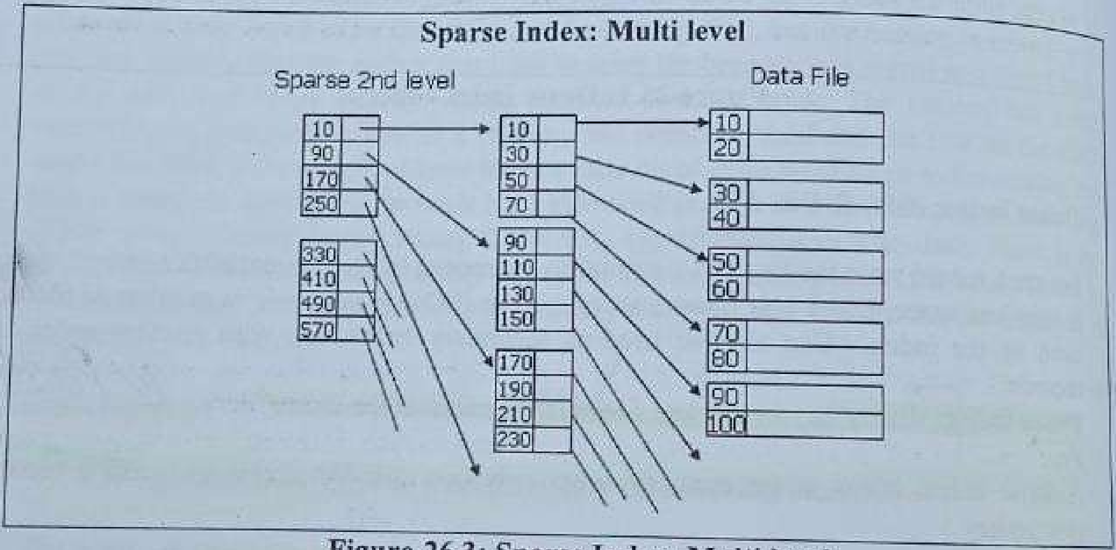

Sparse Index: Multi level

Figure-26.3: Sparse Index: Multi level

26.4 B-tree Indexing

Paper

- Can be seen as a general form of multi-level indexes.
- Generalize usual (binary) search trees (BST).
- Allow efficient and fast exploration at the expense of using slightly more space.
- Popular variant: B+-tree
- Support more efficiently queries like:

  SELECT * FROM R WHERE a = 11
  SELECT * FROM R WHERE 0<= b and b<42

B-tree indexes are the most common index type used in typical OLTP applications and provide excellent levels of functionality and performance. Used in both OLTP and data warehouse applications, they speed access to table data when users execute queries with varying criteria, such as equality conditions and range conditions. B-tree indexes improve the performance of queries that select a small percentage of rows from a table. As a

Dense Index: Concept

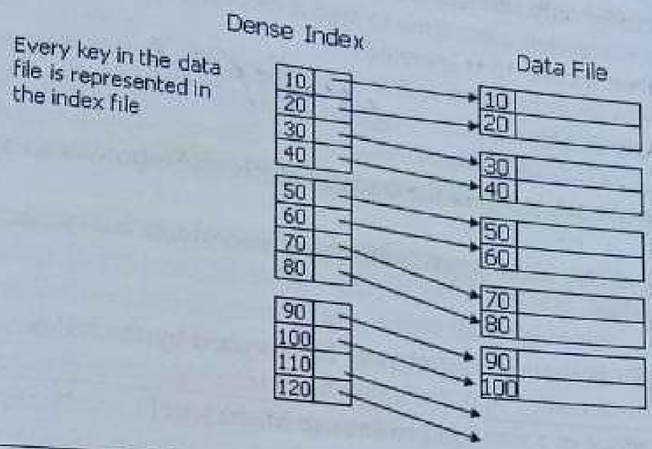


Figure-26.1: Dense index concept

Dense Index: Adv. & Dis. Adv.

For each record store the key and a pointer to the record in the sequential file. Why? It uses less space, hence less time to search. Time (I/Os) logarithmic in number of blocks used by the index. Can also be used as secondary index, i.e. with another order of records.

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

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 expense when used to find a record given its key

Sparse Index: Concept

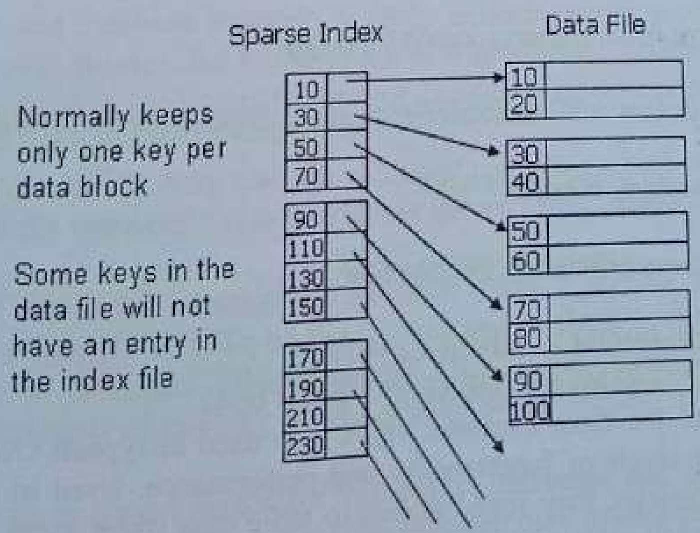

Sparse Index
Data File
Normally keeps only one key per data block
Some keys in the data file will not have an entry in the index file
10
30
50
70
90
110
130
150
170
190
210
230
10
20
30
40
50
60
70
80
90
100

Figure-26.2: Sparse index concept

- "Which students from Lahore are enrolled in 'CS'?"
  Perform a bit-wise AND of two bitmaps: answer – s1 and s9
- "How many students are enrolled in 'CS'?"
  Count 1's in the degree bitmap vector

Answer is 4

Bitmaps are not good for high- cardinality textual columns that have numerous values, such as names or descriptions, because a new column is created in the index for every possible value. With high-cardinality data and a large number of rows, each bitmap index becomes enormous and takes a long time to process during indexing and retrievals.

Bitmap Index: Adv.

write the Bit-map index Advantages:- (Past) important

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

An obvious advantage of this technique is the potential for dramatic reductions in storage overhead. Consider a table with a million rows and four distinct values with column header of 4 bytes resulting in 4 MB. A bitmap indicating which of these rows are for these values requires about 500KB.

More importantly, the reduction in the size of index "entries" means that the index can sometimes be processed with no I/O and, more often, with substantially less I/O than would otherwise be required. In addition, many index-only queries (queries whose responses are derivable through index scans without searching the database) can benefit considerably.

Database retrievals using a bitmap index can be more flexible and powerful than a B-tree in that a bitmap can quickly obtain a count by inspecting only the index, without retrieving the actual data. Bitmap indexing can also use multiple columns in combination for a given retrieval.

Finally, you can use low-level Boolean logic operations at the bit level to perform predicate evaluation at increased machine speeds. Of course, the combination of these factors can result in better query performance.

Bitmap Index: Performance Guidance

Bitmapped indexes can provide very impressive performance speedups; execution times of certain queries may improve by several orders of magnitude. The queries that benefit the most from bitmapped indexes have the following characteristics:

- The WHERE-clause contains multiple tests on low-cardinality columns

Lecture 27: Need for Speed: Special Indexing Techniques

Learning Goals

- Index Structures
- Performance
- Bitmap Indexing
- Clustered Indexing

Without indexes, the DBMS may be forced to conduct a full table scan (reading every row in the table) to locate the desired data, which can be a lengthy and inefficient process. However, creating indexes requires careful consideration. Although indexes can be quite useful for speeding data retrieval, they can slow performance of database writes. This slowdown occurs because a change to an indexed column actually requires two database writes-one to reflect a change in the table and one to reflect a corresponding change in the index. Thus, if the activities associated with a table are primarily write-intensive, it is important to make judicious use of indexes on the relevant tables. Indexes also require a certain amount of disk space, which must be considered when allocating resources to the database.

(short)

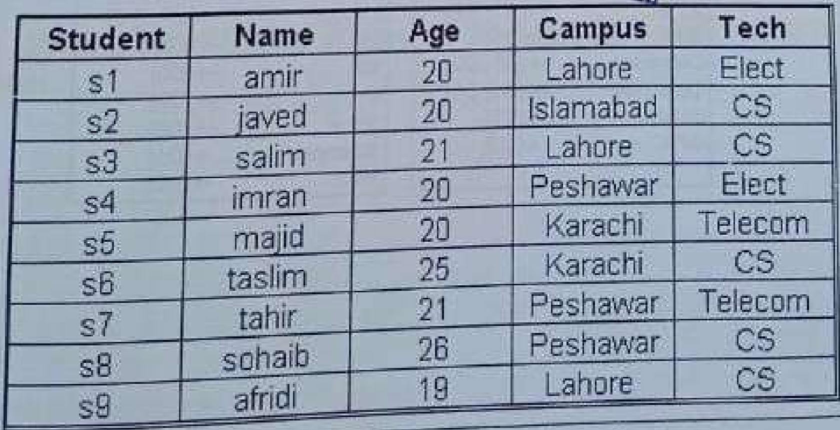
Before looking at your indexing options, we must first discuss the two ways to access data: non-keyed access and keyed access. Non-keyed access uses no index. Each record of the database is accessed sequentially, beginning with the first record, then second, third and so on. This access is good when you wish to access a large portion of the database (greater than 85%). Keyed access provides direct addressing of records. A unique number or character(s) is used to locate and access records. In this case, when specified records are required (say, record 120, 130, 200 and 500), indexing is much more efficient than reading all the records in between.

Special Index Structures

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

JICB

Past



Student	Name	Age	Campus	Tech
s1	amir	20	Lahore	Elect
s2	javed	20	Islamabad	CS
s3	salim	21	Lahore	CS
s4	imran	20	Peshawar	Elect
s5	majid	20	Karachi	Telecom
s6	taslim	25	Karachi	CS
s7	tahir	21	Peshawar	Telecom
s8	sohaib	26	Peshawar	CS
s9	afridi	19	Lahore	CS

Table-27.1: Special Index Structures

- The individual tests on these low-cardinality columns select a large number of rows
- The bitmapped indexes have been created on some or all of these low-cardinality columns
- The tables being queried contain many rows

A significant advantage of bitmapped indexes is that multiple bitmapped indexes can be used to evaluate the conditions on a single table. Thus, bitmapped indexes are very appropriate for complex ad-hoc queries that contain lengthy WHERE-clauses. Performance, storage requirements, and maintainability should be considered when evaluating an indexing scheme.

Bitmap Index: Dis. Adv. ✓ Past

- 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.

Long

27.3 Cluster Index: Concept

- A Cluster index defines the sort order on the base table. I
- Ordering may be strictly enforced (guaranteed) or opportunistically maintained II
- At most one cluster index defined per table. III
- Cluster index may include one or multiple columns. IV
- Reduced I/O. ✓

Qualifying blocks for *Table_B* $QB(B) = 100$

Join cost A&B = 500 + 50×700 = 35,500 I/Os
Join cost B&A = 700 + 100×500 = 50,700 I/Os

i.e. an increase in I/O of about 43%.

For example, if qualifying blocks for *Table_A* QB(A) = 50 and qualifying blocks for *Table_B* QB(B) = 100 and size of Table_A is 500 blocks and size of Table_B is 700 blocks then Join cost A&B = 500 + 50×700 = 35,500 I/Os and using the other order i.e. *Table_B* outer table and *Table_A* as inner table, the join cost B&A = 700 + 100×500 = 50,700 I/Os i.e. an increase in I/O of about 43%.

Nested-Loop Join: Variants

✓ Past

Short + MCQs

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

Working of Query optimizer

There are many variants of the traditional nested-loop join. The simplest case is when an entire table is scanned; this is called a naive nested-loop join. If there is an index, and that index is exploited, then it is called an index nested-loop join. If the index is built as part of the query plan and subsequently dropped, it is called as a temporary index nested-loop join. All these variants are considered by the query optimizer before selecting the most appropriate join algorithm/technique.

28.4 Sort-Merge Join

Joined tables to be sorted as per WHERE clause of the join predicate.

Query optimizer scans for (cluster) index, if exists performs join.

In the absence of index, tables are sorted on the columns as per WHERE clause.

If multiple equalities in WHERE clause, some merge columns used.

The Sort-Merge join requires that both tables to be joined are sorted on those columns that are identified by the equality in the WHERE clause of the join predicate. Subsequently the tables are merged based on the join columns. The query optimizer typically scans an index on the columns which are part of the join, if one exists on the proper set of columns, fine, else the tables are sorted on the columns to be joined, resulting in what is called a cluster index. However, in rare cases, there may be multiple equalities in the WHERE clause, in such a case, the merge columns are taken from only some of the given equality clauses.

Bitmap index : Adv
(i) very low storage space
(ii) Reduction I/O

(iii) Counts & Joins
(iv) low level Bit operations

Lecture 28: Join Techniques

Disadvantage:
(i) lock of many rows
(ii) Keyword Parsing
(iii) Difficult to parsing.

Leaning Goals

- Join Techniques
- Nested loop join
- Sort Merge Join
- Hash based join

Background

- Used to retrieve data from multiple tables.
- Joins used frequently, hence lot of work on improving or optimizing them.
- Simplest join that works in most cases is nested-loop join but results in quadratic time complexity.
- Tables identified by FROM clause and condition by WHERE clause.
- Will cover different types of joins.

Join commands are statements that retrieve data from multiple tables. A join is identified by multiple tables in the FROM clause, and the relationship between the tables to be joined is established through the existence of a join condition in the WHERE clause. Because joins are so frequently used in relational queries and because joins are so expensive, lot of effort has gone into developing efficient join algorithms and techniques. The simplest i.e. nested-loop join is applicable in all cases, but results in quadratic performance. Several fast join algorithms have been developed and extensively used; these can be categorized as sort-merge, hash-based, and index-based algorithms. In this lecture we will be covering the following join algorithms/techniques:

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

Past

28.1 About Nested-Loop Join

Nested loop Join and its variants—? (Past)

Typically used in OLTP environment.

Limited application for DSS and VLDB

In DSS environment we deal with VLDB and large sets of data.

Traditionally Nested-Loop join has been and is used in OLTP environments, but for many reasons, such a join mechanism is not suitable for VLDB and DSS environments. Nested loop joins are useful when small subsets of data are joined and if the join condition is an efficient way of accessing the inner table. Despite these restrictions/limitations, we will

Fig-28.2 shows the process of merging two sorted tables with IDs shown. Conceptually the merging is similar to the merging you must have studies in Merge_Sort in your Algorithm course.

Sort-Merge Join: Note features

Very fast.

Sorting can be expensive.

Presorted data can be obtained from existing B-tree.

Sort-Merge join itself is very fast, but it can be an expensive choice if sort operations are required frequently i.e. the contents of the table's change often resulting in deterioration of the sort order. However, it may so happen that even if the data volume is large the desired data can be obtained presorted from existing B-tree. For such a case sort-merge join is often the fastest available join algorithm.

28.5 Hash-Based Join: Working

Suitable for the VLDB environment. in Past Paper

The choice which table first gets hashed plays a pivotal role in the overall performance of the join operation, this decided by the optimizer.

The joined rows are identified by collisions i.e. collisions are "good" in case of hash join.

Hash joins are suitable for the VLDB environment as they are useful for joining large data sets or tables. The choice about which table first gets hashed plays a pivotal role in the overall performance of the join operation, and left to the optimizer. The optimizer decides by using the smaller of the two tables (say) *Table_A* or data sources to build a hash table in the main memory on the join key used in the WHERE clause. It then scans the larger table (say) *Table_B* and probes the hashed table to find the joined rows. The joined rows are identified by collisions i.e. collisions are "good" in case of hash join.

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.

This method is best used when the smaller table fits in the available main memory. The cost is then limited to a single read pass over the data for the two tables. Else the "smaller" table has to be partitioned which results in unnecessary delays and degradation of performance due to undesirable I/Os.

every problem is a oppertunity

جہاں چیزیں زیادہ ہوں وہاں data mining کی جاتی ہے مثال کے طور پر کوئلے کی کانوں سے کوئلہ نکالنا اور volume جب زیادہ ہو تو تب ہی data mining کرتے ہیں جب ہم data mining کی بات کرتے ہیں اس کا مطلب یہ ہوتا ہے کہ بہت زیادہ ڈیٹا میں سے وہ چیزوں کو دریافت کرنا Data mining کہلاتا ہے

Lecture 29: A Brief Introduction to Data mining (DM)

Learning Goals

- Definition
- What is Data Mining?
- Why Data Mining?
- How Data Mining is different?
- Data Mining Vs. Statistics

(I) مثال: جیسے کہ ہمیں اپنے نام کے بارے میں پتہ ہے اور ایڈریس کے بارے میں کہ ہمیں پتہ ہے

In our one of previous lectures, we discussed "putting the pieces together". One of the things in those pieces was data mining. We mine data when we need to discover something out of a lot. It is a broad discipline, dedicated courses being offered solely. However, we will go through a brief introduction of the field so that we get ample knowledge about data mining concepts, applications and techniques.

29.1 What is Data Mining?: Informal

(I) کچھ چیزیں ایسی ہیں کہ جن کے بارے میں ہمیں پتا ہے

"There are things that we know that we know…

(II) ایسی چیزیں جن کے بارے میں آپ کو پتہ ہے کہ آپ کو ان کے بارے میں نہیں پتہ

there are things that we know that we don't know…

(III) کچھ ایسی بھی چیزیں ہیں کہ جن کے بارے میں آپ کو کبھی بھی نہیں پتہ کہ آپ کو ان کے بارے میں نہیں پتہ

there are things that we don't know we don't know."

Donald Rumsfield

US Secretary of Defence

(II) مثال: مثال کے طور پر آپ کے بارے میں کوئی کیا سوچ رہا ہے آپ کو نہیں پتہ ہے مثال کے طور پر آپ نے لاٹری کا ٹکٹ خریدا ہوا ہے آپ کو نہیں پتہ کہ آپ کا انعام نکلے گا کہ نہیں

Let's start data mining with a interesting statement. Why interesting because the statement covers the overall concept of DM and is given by a non-technical person who neither is a scientist nor a data mining guru. The statement, given by Donald Rumsfeld, Defense Secretary of the USA in an interview, is as under.

As we know, there are known knowns. There are things we know that we know like you know your names, your parent's names. We also know there are known unknowns. That is to say, we know that there are some things we do not know like what one is thinking about you, what you will eat after six days, what will be result of a lottery and so on. But there are also unknown unknowns, the ones we don't know that we don't know. Are they beneficial if you know? Or it is harmful no to know them?

There are also unknown knowns, things we'd like to know, but don't know, but know someone who can doctor them and pass them off as known knowns. To associate Rumsfeld's above quotation with data mining, we identify four core phrases as

1. Known knowns
2. Known unknowns
3. Unknown unknowns

In Partitioning phase read + write both operation requirs 2(M+N) I/O

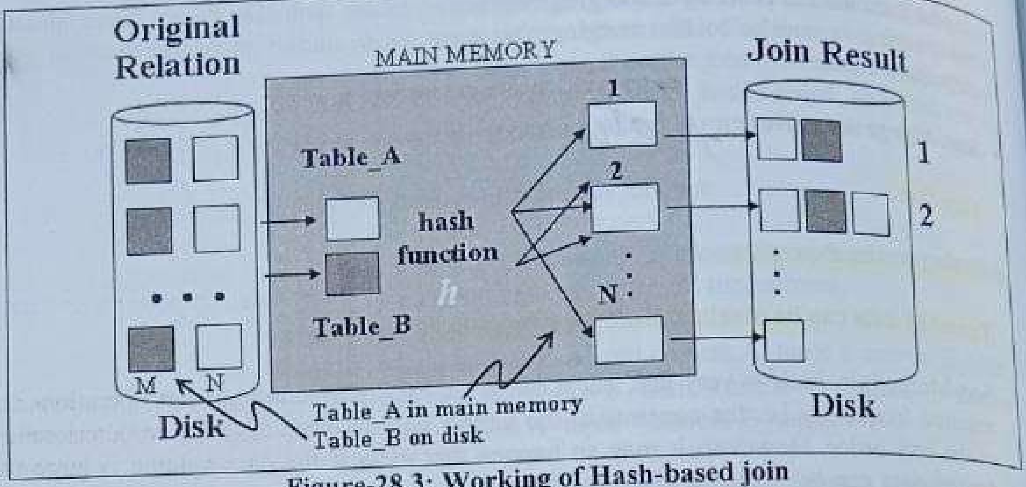


Figure-28.3: Working of Hash-based join

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.

Hash-Based Join: Large "small" Table

- Smaller of the two tables may grow too big to fit into the main memory.
- Optimizer performs partitioning, but is not simple.
- Multi-step approach followed, each step has a build phase and probe phase.
- Both tables entirely consumed and partitioned via hashing.
- Hashing guarantees any two joining records will fall in same pair of partitions.
- Task reduced to multiple, but smaller, instances of the same tasks.

It may so happen that the smaller of the two tables grows too big to fit into the main memory, then the optimizer breaks it up by partitioning, such that a partition can fit in the main memory. However, it is not that simple because the qualifying rows of both the tables have to fall in the corresponding partition pairs that are hashed (build) and probed. Thus in such a case the hash join proceeds in several steps. Each step has a build phase and probe phase. Initially, the two tables are entirely consumed and partitioned (using a hash function on the hash keys) into multiple partitions. The number of such partitions is sometimes called the partitioning fan-out. Using the hash function on the hash keys (based on the predicates in the WHERE clause) guarantees that any two joining records must be in the same pair of partitions. Therefore, the task of joining two large tables gets reduced to multiple, but smaller, instances of the same tasks. The hash join is then applied to each pair of partitions.

4. Unknown knowns

The items 1 3, and 4 deal with "*Knowns*". Data mining has relevance to the third point in red. It is an art of digging out what exactly we don't know that we must know in our business. The methodology is to first convert "*unknown unkowns*" into "*known unknowns*" and then finally to "*known knowns*".

29.2 What is Data Mining?: Slightly Informal

ڈیٹا ویئر ہاؤس سے آپ یہ پوچھتے ہیں کہ مجھے ایک ایسی

Tell me something that I should know.

چیز بتاؤ جو کہ مجھے نہیں پتہ یعنی کہ مجھے وہ بتاؤ جو

When you don't know what you should be knowing, how do you write SQL?

You cant!! مجھے پتہ ہونا چاہیے

Now a slightly technical view of DM. Not that much technical but you may easily understand. Tell me something that I should know i.e. you ask your DWH, data repository that tell me something that I don't know, or I should know. Since we don't know what we actually don't know and what we must know to know, we can't write SQL's for getting answers like we do in OLTP systems. Data mining is an exploratory approach, where browsing through data using data mining techniques may reveal something that might be of interest to the user as information that was 'unknown previously. Hence, in data mining we don't know the results.

29.3 What is Data Mining?: Formal غیر معمولی

- 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.

Before looking into the technical or formal view of DM, consider the quote that you might have heard in your childhood i.e. *finding a needle in the haystack*. It is a tough job to find a needle in a big box full of hay. Dm is finding in the hay stack (huge data) the needle (knowledge). You don't have idea about where the needle can be found or even you don't know whether the needle is there in the haystack or not.

Historically, the notion of finding useful patterns in data has been given a variety of names, including data mining, knowledge extraction, information discovery, information harvesting, data archaeology, and data pattern processing. The term data mining has mostly been used by statisticians, data analysts, and the management information systems (MIS) communities. It has also gained popularity in the database field. KDD refers to the overall process of discovering useful knowledge from data, and data mining refers to a particular step in this process. Data mining is the application of specific algorithms for extracting patterns from data. The additional steps in the KDD process, such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining, is essential to ensure that useful knowledge is derived from the data. Blind application of data-mining methods (rightly

1. developing algorithms and systems to mine large, massive and high dimensional data sets;
2. developing algorithms and systems to mine new types of data (images, music, videos);
3. developing algorithms, protocols, and other infrastructure to mine distributed data; and
4. improving the ease of use of data mining systems;
5. developing appropriate privacy and security techniques for data mining.

Data mining evolved as a mechanism to cater the limitations of OLTP systems to deal massive data sets with high dimensionality, new data types, multiple heterogeneous data resources etc. The conventional systems couldn't keep pace with the ever changing and increasing data sets. Data mining algorithms are built to deal high dimensionality data, new data types (images, video etc.) , complex associations among data items , distributed data sources and associated issues (security etc.)

29.6 Data Mining is HOT!

- **10 Hottest Jobs of year 2025**

Time Magazine, 22 May, 2000

- **10 emerging areas of technology**
  MIT's Magazine of Technology Review, Jan/Feb, 2001

The TIME Magazine May 2000 issue has given a list of the ten hottest jobs of year 2025. Data miners and knowledge engineers were at 5th and 6th position respectively. The proposed course/Curriculum will cover Data Mining. Hence Data mining is a hot field having wide market opportunities.

Similarly, MIT's Technology Review has identified 10 emerging areas of technology that will soon have a profound impact on the economy and how we live and work. Among the list of emerging technologies that will change the world, Data mining is at the 3rd place.

Thus in view of the above facts, *data miners* have a long career in national as well as international market as major companies both private and government are quickly adopting the technology and many have already adopted.

Start

29.7 How Data Mining is different?

Long in past

- **Knowledge Discovery**

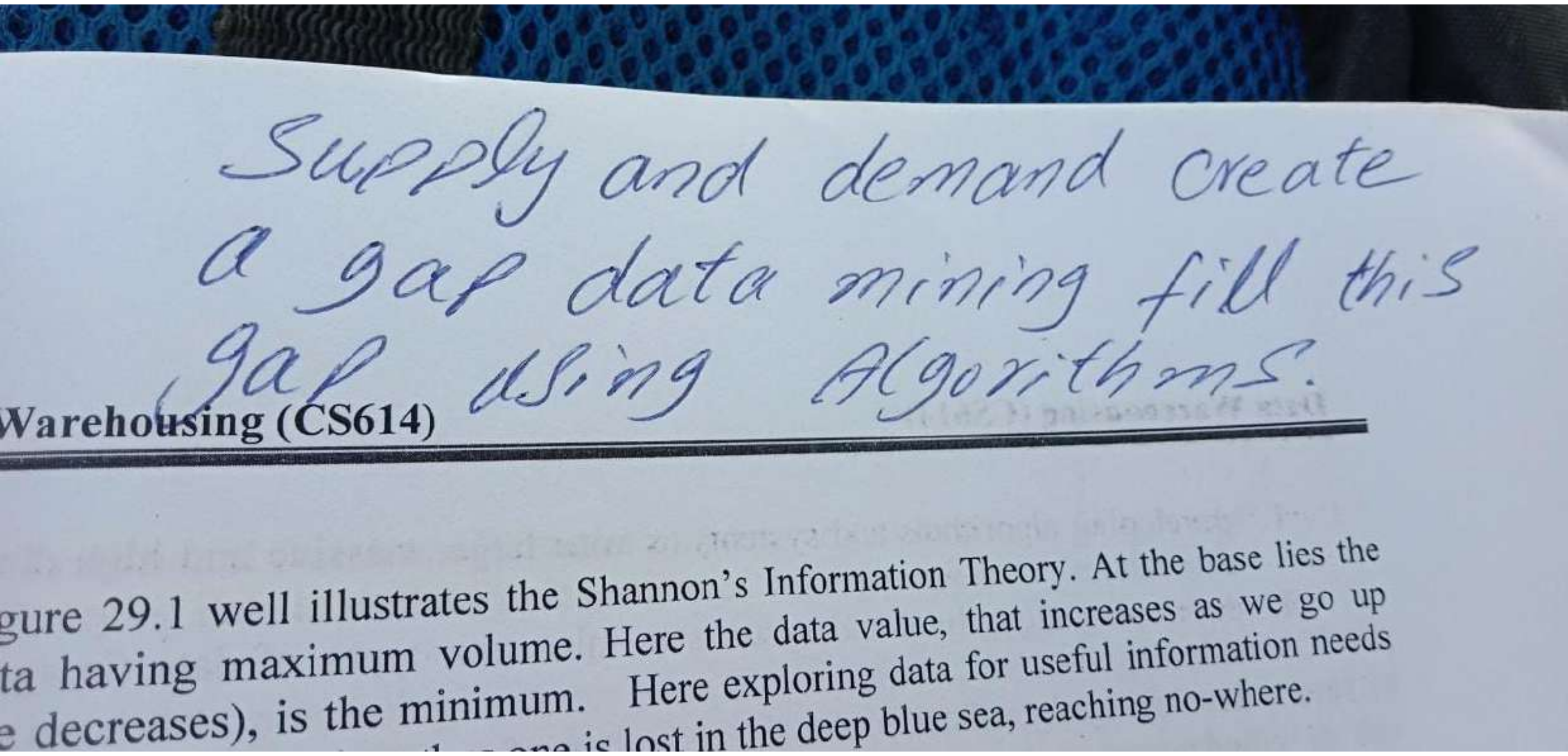
-- ~~Overall~~ process of discovering useful knowledge

- **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***



Supply and demand create a gap data mining fill this gap using Algorithms.

gure 29.1 well illustrates the Shannon's Information Theory. At the base lies the ta having maximum volume. Here the data value, that increases as we go up e decreases), is the minimum. Here exploring data for useful information needs

is lost in the deep blue sea, reaching no-where.

- **Traditional Database** (Transactions):
  -- Querying data in well-defined processes. ***Reliable storage***

end

Conventional data processing systems or online transaction processing systems (OLTP) lie at the bottom level. These systems have well defined queries and no any sort of knowledge discovery is performed. OLTP systems are meant to support day to day transactions and do not support decision making in any way. We can better understand with the analogy that when you travel from your home to university you may follow a same route very often. While on the way you will see same trees, same signals and same buildings every day provided you follow the same route. It is not possible that each and every day you see different buildings, trees than the previous day. Similar is the case for OLTP systems where you have well defined queries by running which you know what sort of results you can get. Nothing new or no discoveries are here.

Data Mining provides a global macroscopic view or aerial view of your data. You can easily see what you could not see at microscopic level. But before applying mining algorithms data must be brought in a form so that the knowledge exploration from huge, heterogeneous and multi source data can efficiently and effectively be performed. Thus DWH is the process of bringing input data in a form that can readily be used by data mining techniques to find hidden patterns. Both terns KDD and DM are sometimes used to refer to the same thing but KDD refers to the overall process from data extraction from legacy source systems, data preprocessing, DWH building, data mining and finally the output generation. So KDD is a mega process having sub processes like DWH and DM being its constituent parts.

How Data Mining is different...

Past

Data Mining Vs. Statistics

- Formal (statistical) inference is assumption driven i.e. a hypothesis is formed and validated against the data.
- Data mining is discovery driven i.e. patterns and hypothesis are automatically extracted from data.
- Said another way, data mining is knowledge driven, while statistics is human driven.

Although both of the two are for data analysis and none is good or bad, some of the difference between statistics and Data mining are;

Statistic is assumption driven. A hypothesis is formed using the historical data and is then validated against current known data. If true the hypothesis becomes a model else the process is repeated with different parameters. DM, on the other hand, is discovery driven i.e. based on the data hypothesis is automatically extracted from the data. The purpose is to find patterns which are implicit and hidden in the data sea otherwise. Thus data mining is knowledge driven while statistics is human driven.

developers. Now where to place a specific news item on the web site? What should be the hierarchical position of the news item, what should be the news chapter, category? Either it should be in the sports or weather section and so on. What is the problem in doing all this? The problem is that it's not a matter of placing a single news item. The site as already mentioned contains a number of content developers and also many categories. If sorting is performed humanly, then it is time consuming. That is why classification techniques can scan and process the document to decide its category or class. How and what sort of processing will be discussed in the next lecture. It is not possible and there are flaws in assigning category to any news document just based on the keyword. Frequent occurrence of the word keyword cricket in a document doesn't necessary means that the document be placed in the sports category. The document may be actually political in nature.

30.2 ESTIMATION

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

Example:

Building a model and assigning a value from 0 to 1 to each member of the set.

Then classifying the members into categories based on a threshold value.

As the threshold changes the class changes.

Next category of problems that can be solved with DM is using estimation. In classification we did binary assignment i.e. data items are assigned to either of the two categories or classes, this or that. The assignment value was integer in nature, and to be absolute was not essential. However in case of estimation a model/mechanism is formed then data analysis is performed in that model which was actually formed from data itself. The difference is that the model is formed from the relationships in the data and then data categorized in that model. Unlike classification, categorization here is not absolute but there is a real number that is between 0 and 1. This number tells the probability of a record/tuple/item etc. to belong to a particular group or category or class. So this is a more flexible approach than classification. Now the question arises how a real number between 0 and 1 reveals the probability of belonging to a class? Why not an item falls in two groups or more at the same time? The answer is that categorization is performed by setting the threshold values. It is predefined that if the value crosses this then in this class else in another class and so on. Note that if thresholds are reset which is possible (as nothing is constant except change so changes can be made in the threshold) then the category or class boundaries change resulting in the movement of records and tuples in other groups accordingly.

30.3 PREDICTION

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

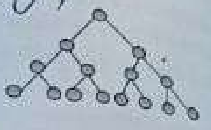
⑩ At every node we decide split or not reco

Data Mining is...

① Tables splits into records and records splits into again multiple records

- Decision Trees

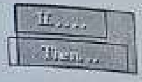
اس ٹری سے رولز بن جاتے ہیں جو ہمیں پہلے پتہ نہیں ہوتے Training سیٹ سے یہ ٹری بناتے ہیں



- Neural Networks

- Rule Induction ⟹

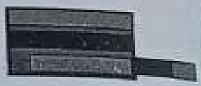
اس میں ہمیں پہلے سے ہی کلسٹر کا پتہ ہوتا ہے اور ان کلسٹر کی خصوصیات کا بھی مثلاً نئے آنے والے گاہک اچھے ہونگے یا نہیں



- Clustering →

اس میں ہمیں پہلے سے کلاسز کا پتہ نہیں ہوتا اور نہ ان کلاسز کی خصوصیات کا پتہ ہوتا ہے

- Genetic Algorithms



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. Suppose a table having a number of records. The tree construction process will group most related records or tuples in the same group. Decision at each node is taken based on some rule, if this then this else this. Rules are not known a priori and are digged out of the training data set.

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 involes the grouping of data items without taking any human parametric input. We don't know the number of clusters and their properties a priori. Two main types are one way clustering and two way clustering. One way clustering is when only data records (rows) are used. Two way clustering is when all the rows and columns are being used for clustering purpose.

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. Briefly, these techniques are used to select the optimal solution out of a number of possible solutions. However, are not much robust as can not perform well in the presence of noise.

It could be the first step to the market segmentation effort.

What else data mining can do? We can do clustering with DM. Clustering is the technique of reshuffling, relocating exiting segments in given data which is mostly heterogeneous so that the new segments have more homogeneous data items. This can be very easily understood by a simple example. Suppose some items have been segmented on the basis of color in the given data. Suppose the items are fruits, then the green segment may contain all green fruits like apple, grapes etc. thus a heterogeneous mixture of items. Clustering segregates such items and brings all apples in one segment or cluster although it may contain apples of different colors red, green, yellow etc. thus a more homogeneous cluster than the previous cluster.

Clustering is a difficult task, why? In case of classification we already know the number of classes, either good or bad or yes or no or any number of classes. We also have the knowledge of classes properties so its easy to segment data into known classes. However, in case of clustering we don't know the number of clusters a priori. Once clusters are found in the data business intelligence, domain knowledge is needed to analyze the found clusters. Clustering can be the first step towards market segmentation i.e. we can use countermining to know the possible clusters in the data. Once clusters found and analyzed classification can be applied thus gaining more accuracy than any standalone technique. Thus clustering is at higher level than classification not only because of its complexity but also because it leads to classification.

Examples of Clustering Applications

Past

- Marketing: Discovering distinct groups in customer databases, such as customers who make lot of long-distance calls and don't have a job. Who are they? Students. Marketers use this knowledge to develop targeted marketing programs.
- Insurance: Identifying groups of crop insurance policy holders with a high average claim rate. Farmers crash crops, when it is "profitable".
- Land use: Identification of areas of similar land use in a GIS database.
- Seismic studies: Identifying probable areas for oil/gas exploration based on seismic data.

We discussed that what clustering is and how it works. Now to know the real spirit of it, lets look at some of the real world examples to show the blessings of clustering;

1. **Knowing or discovering about your market segment:** Suppose a telecom company whose data when clustered revealed that there is a group or cluster of people or customers whose long distance calls are greater in number. Is this a discovery that such a group exists? Nope not really. The real discovery is analyzing the cluster, the real fun part. Why these people are in a cluster? Is important to know. Analysis of the cluster reveals that all the people in the group are unemployed! How come it is possible that unemployed people are making expensive far distance calls? The excitement lead to further analysis which ultimately revealed that the people in the

Discovering Association Rules

- Given a set of records, each containing set of items
  - Produce dependency rules that predict occurrences of an item based on others
- Applications:
  - Marketing, sales promotion and shelf management
  - Inventory management

TID	Items
1	Bread, Cola, Milk
2	Juice, Bread
3	Juice, Cola, Diaper, Milk
4	Juice, Bread, Diaper, Milk
5	Cola, Diaper, Milk

Rules:
{Milk} → {Cola}
{Diaper, Milk} → {Juice}

Table -30.1: Discovering association rules

Discovering Association Rules is another name given to market basket analysis. Here rules are formed from the dependencies among data items which can be used to predict the occurrence of an item based on others e.g. suppose hardware shop where whenever a customer buys color tins it is more likely that he /she will buy painting brushes too. So based on the occurrence or event of paint purchase, we can predict the occurrence of item paint brush. What is the benefit of knowing all this? We have already discussed this. Now look at the Table 30.1, here two columns TIC (transaction ID) and other is the list of items. This is not the view of a real database, as a single column can not contain multiple entries like items column here. This is an example table just to show rule formation process. Looking at the table we come to know that whenever milk is purchased, cola is also purchased. Similarly, whenever diaper and milk are purchased juice is also purchased. So, the two association rules are obtained from the sample data in Table 30.1. Now a question arises which of the two rules strongly implies? This can not be answered depending on a lot of factors. However, we can tell what has been discovered here? What is the unknown unknown? The discovery is that the sale of juice with diapers and milk is non trivial. This can never be guessed because no obvious association is found among the items.

اس میں نہ ہمیں گروپس کا پتہ ہوتا ہے نہ ان کی خصوصیات کا پتہ ہے ۔

30.5 CLUSTERING

مختلف

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

ایک جیسی

Unlike classification, it does NOT depend on predefined classes.

It is up to you to determine what meaning, if any, to attached to resulting clusters.

Robustness: this is the ability of the method to make correct predictions/groupings given noisy data or data with missing values

We discussed different data mining techniques. Now the question, which technique is good and which bad? Or say like which is the best technique for a given problem. Thus we need to specify evaluation criteria like data metrics as we did in the data quality lecture. The metrics we use for comparison of DM techniques are;

short

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. Once built the model is tested for accuracy in terms of % correct results as the classification of the test data is already known. So we specify the correctness or confidence level of the technique in terms % accuracy.

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. This is the computational time but user or business decision maker is interested in the absolute clock time. He has nothing to do with complexities. What he is interested in is, knowing how fast he gets the answers. So just comparing on the basis of complexities is not sufficient. We must look at the overall process and interdependencies among tasks which ultimately result in the answer or information generation.

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. The techniques can be compared on the basis of their interpretational ability e.g. there might be some methods which give additional functionalities to provide meaning to the discovered information like color coding, plots and curve fittings etc.

Past Question: what is diffence between data matrix and similarity/dissimilarity matrix in terms of rows and columns which one is symmetric ?

Ans :

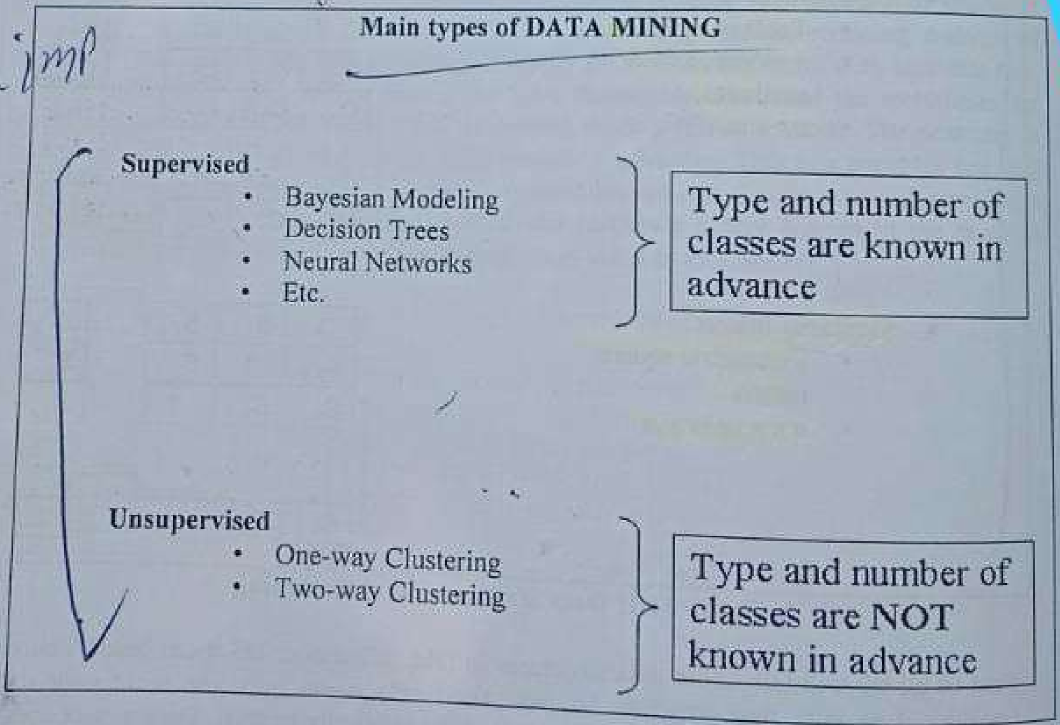
or size of that table normally? The size of records (rows) is much greater than the number of columns. The attributes may be 10, 15 or 25 but the number of [illegible] far exceeds the number of columns. [illegible] customer table may have [illegible] attributes but the total records may be in millions. As I said previously that the mobile users in Pakistan are about 10 million. If a company even has 1/3 of the customers then 3.3 *lakh* customer records in the customer table. Thus greater number of rows than columns and there will be indices *i* and *j* in the table and you can pick the particular contents of a cell by considering the intersection of the two indices.

The second matrix in the Figure 31.1 is called the similarity matrix. Lets talk about its brief background. Similarity or dissimilarity matrix is the measure the similarity or dissimilarity obtained by pair wise comparison of rows. First of all you measure the similarity of the row1 in data matrix with itself that will be 1. So 1 is placed at index 1, 1 of the similarity matrix. Then you compare row 1 with row 2 and the measure or similarity value goes at index 1, 2 of the similarity matrix and son. In this way the similarity matrix is filled. It should be noted that the similarity between row1 and row2 will be same as between row 2 and 1. Obviously, the similarity matrix will then be a square matrix, symmetric and all values along the diagonal will be same (here 1). So if your data matrix has *n* rows and *m* columns then your similarity matrix will have *n* rows and *n* columns. What will be the time complexity of computing similarity/dissimilarity matrix? It will be O (n^2) (m), where m accounts for the vector or header size of the data. Now how to measure or quantify the similarity or dissimilarity? Different techniques available like Pearson correlation and Euclidean distance etc. but in this [illegible] we have used Pearson correlation which you might have studied in your statistics course.

dissimilarity

31.2 Main types of DATA MINING

V.v.imp



Main types of DATA MINING

Supervised
- Bayesian Modeling
- Decision Trees
- Neural Networks
- Etc.

Type and number of classes are known in advance

Unsupervised
- One-way Clustering
- Two-way Clustering

Type and number of classes are NOT known in advance

Supervised learning: اس کا مطلب ہے کہ جب
Data mining کی جا رہی ہے تو کچھ ایسی معلومات بھی
دستیاب ہے جو آپ کو mining میں مدد دے سکتی ہے

Lecture 31: Supervised Vs. Unsupervised Learning

مثلاً آپ کو پہلے سے ہی پتہ ہو کہ mining
Perform کرتے ہوئے Dataset میں
کتنی کلاسز اور گروپس ہونی چاہیے
گروپس اور کلاسز کی Properties کیا ہے۔

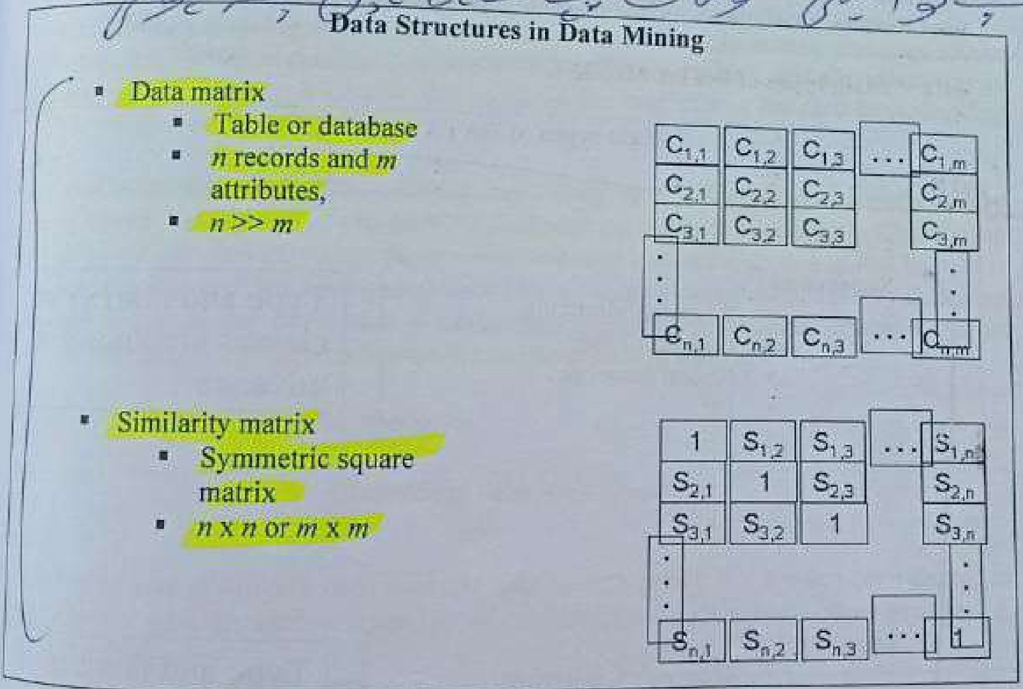
Learning Goals

- Main types of DATA MINING
- How Clustering works?
- Data Mining Agriculture data
- How Classification work?
- Understanding the K-Means Clustering

In the previous lecture we discussed briefly DM concepts. Now we look with some greater details, two main DM methods supervised and unsupervised learning. Supervised learning is when you are performing DM the supporting information that helps in the DM process is also available. What could be that information? You may know your data that how many groups or classes your data set contains. What are the properties of these classes or clusters? When we will talk about unsupervised learning you will not have such known or a priori knowledge. In other words you can not give such factors as input to the DM technique which can facilitate your DM process. So wherever the user gives some input that is supervised else that is unsupervised learning.

Unsupervised learning: مطلب کہ Data mining کرتے
ہوئے آپ کو ایسی معلومات پہلے سے ہی نہیں پتہ ہوں گی

31.1 Data Structure in Data Mining



Data Structures in Data Mining

- Data matrix
  - Table or database
  - *n* records and *m* attributes,
  - *n* >> *m*

$C_{1,1}$	$C_{1,2}$	$C_{1,3}$	. . .	$C_{1,m}$
$C_{2,1}$	$C_{2,2}$	$C_{2,3}$		$C_{2,m}$
$C_{3,1}$	$C_{3,2}$	$C_{3,3}$		$C_{3,m}$
⋮				⋮
$C_{n,1}$	$C_{n,2}$	$C_{n,3}$	. . .	$C_{n,m}$

- Similarity matrix
  - Symmetric square matrix
  - *n* x *n* or *m* x *m*

1	$S_{1,2}$	$S_{1,3}$	. . .	$S_{1,n}$
$S_{2,1}$	1	$S_{2,3}$		$S_{2,n}$
$S_{3,1}$	$S_{3,2}$	1		$S_{3,n}$
⋮				⋮
$S_{n,1}$	$S_{n,2}$	$S_{n,3}$	. . .	1

Figure 31.1: Data Structures in data mining

First of all we will talk about data structures in DM. What does DS mean here? You can consider it as pure data structure but we specifically mean how you store your data. Figure 31.1 shows two metrics data matrix and the similarity matrix. Data matrix means the table or database used as the input to the DM algorithm. What will be the dimensions technique. Figure 32.9(B) is the clustered output and when cluster detection will be performed on B, three clusters will successfully be detected. If A is taken as input to the cluster detection algorithm instead of B, you may end with nothing. There is a misconception among new data miners that cluster detection is a simple task. They think that using K-means is everything. This is not always the case, k-means can not work always. Can it work for matrices A and B? Wait till last slide for the answer.

31.7 The K-Means Clustering

- Given *k*, the *k-means* algorithm is implemented in 4 steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  - Assign each object to the cluster with the nearest seed point.
  - Go back to Step 2, stop when no more new assignment.

Before mentioning the strengths and weaknesses of the k-means, lets first discuss it working. It is implemented in four steps.

Step 1: In the first step you assign *k* clusters in your data set. Thus it's a supervised technique as you must know the number of classes and their properties a priori.

Step 2: The second step is to compute the seed points or centroids of your defined clusters i.e. which value is a most representative value of all the points in a cluster. For the sake of your convenience, we are talking about 2- D space, otherwise k-means can work for multidimensional data sets as well. The centroid can be the mean of these points, hence called k-means.

Step 3: In this step you take the distance of each point from the cluster centroids or means. On the basis of a predefined threshold value, it is decide that which point belongs to which cluster.

Step 4: You may repeat the above steps i.e. you find the means of newly formed clusters then find the distances of all points from those means and clusters are reconfigured. The process is normally repeated until some changes occur in clusters and mostly you get better results.

Long

Now we will discuss the two main types Dm techniques as briefed in the beginning. First we will discuss supervised learning which includes Bayesian classification, decision trees, neural networks etc. Lets discuss Bayesian classification or modeling very briefly. Suppose you have a data set and when you process that data set, say when you do profiling of your data you come to know about the probability of occurrence of different items in that data set. On the basis of that probability, you form a hypothesis. Next you find the probability of occurrence of an item in the available data set on that hypothesis. Similarly, how this can be used in decision trees? To understand suppose there is insurance company and is interested in knowing about the risk factors. If a person is of age between 20 and 25, he is unmarried and his job is unstable then there is a risk in offering insurance or credit card to such a person. This is because if married one may drive carefully even thinking of his children than otherwise. Thus when the tree was formed the classes, low risk and high risk were already known. The attributes and the properties of the classes were also known. This is called supervised learning.

Now unsupervised learning where you don't know the number of clusters and obviously no idea about their attributes too. In other words you are not guiding in any way the DM process for performing the DM, no guidance and no input. Interestingly, some people say their techniques are unsupervised but still give some input although indirectly. So a pure unsupervised algorithm is the one which don't have any human involvement or interference in any way. However, if some information regarding the data is needed, the algorithm itself can automatically analyze and get the data attributes. There are two main types of unsupervised clustering.

Long in Part

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.

31.3 Clustering: Min-Max Distance

Clustering: Min-Max Distance

Finding groups of objects such that the objects in a group are similar (or related) to one another and dissimilar from (or unrelated to) the objects in other groups e.g. using K-Means.

The *K-Means* Clustering: Example

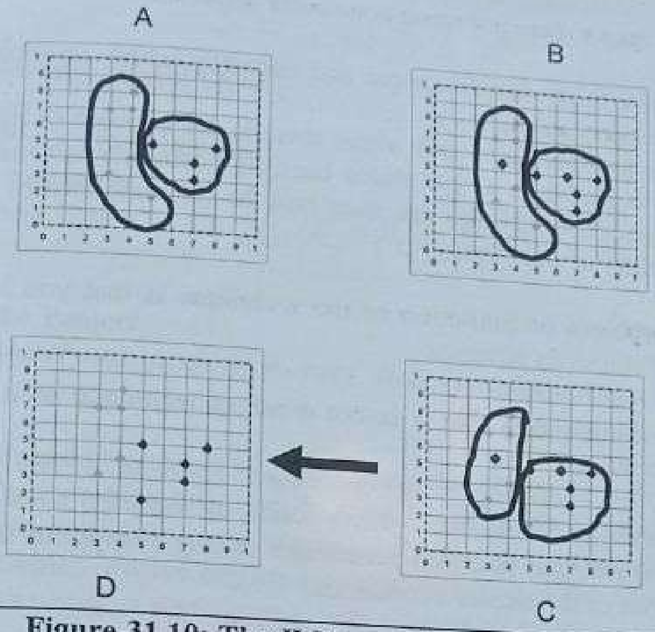


Figure-31.10: The *K-Means* Clustering Example

Consider the example in the figure 32.10 for better understanding k-means working. Figure A shows two sets of color points and the two colors represent two different classes. The polygons drawn around points in different clusters signify the cluster boundaries. Now at Figure B a red point has come in each of the two clusters. This is the centroid or mean of the value in that cluster. The next step is to measure the distances of all the points from each of these centoirds. So those distances which are above some threshold will go in a cluster for each mean point or centroid. Now look at the figure C, here on the basis of distances measured in the previous step new cluster boundaries have been made. In figure D the boundaries have been removed and we see that a point has been removed from one of the clusters and added to the other. As we will repeat the process, the result will get more and finer.

The K-Means Clustering: Comment Long in Part

- Strength
  - *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- 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*

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.

32.3 Development Methodologies

A *Development Methodology* describes the expected evolution and management of the engineering system.

MCQ

Waterfall Model: The model is a linear sequence of activities like requirements definition, system design, detailed design, integration and testing, and finally operations and maintenance. The model is used when the system requirements and objectives are known and clearly specified. While one can use the traditional waterfall approach to developing a data warehouse, there are several drawbacks. First and foremost, the project is likely to occur over an extended period of time, during which the users may not have had an opportunity to review what will be delivered. Second, in today's demanding competitive environment there is a need to produce results in a much shorter timeframe.

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 close to the existing infra-structure to easily envision the larger empowerment that data warehousing can offer. Development and delivery of early prototypes will drive future requirements as business users are given direct access to information and the ability to manipulate it. Management of expectations requires that the content of the data warehouse be clearly communicated for each iteration.

There are 5 keys to a successful rapid prototyping methodology: ✓ Paper

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.

DWH: This about knowledge Discovery. Data warehouse is a part of Decision Support System.

Lecture 32: DWH Lifecycle: Methodologies

Learning Goals

- Data warehouse project Layout
- Information System.
- Understand the Project Development
- Implementation strategies
- DWH Development Cycle

Explore: تحقیق و تفتیش شدہ، معلوم ہوتا

32.1 Layout the Project

A data warehouse project is more like scientific research than anything in traditional IS!

The normal Information System (IS) approach emphasizes on knowing what the expected results are before committing to action. In scientific research, the results are unknown up front, and emphasis is placed on developing a rigorous, step-by-step process to uncover the truth. The activities involve regular interactions between the scientist and the subject and also among the project participants. It is advised to adopt an exploratory, hands-on process involving cross-disciplinary participation.

Building a data warehouse is a very challenging job because unlike software engineering it is quite a young discipline, and therefore, does not yet has well-established strategies and techniques for the development process. Majority of projects fail due to the complexity of the development process. To date there is no common strategy for the development of data warehouses; they are more of an art than science. 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

Short (Part)

W. S R S D: G U.

32.2 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

What skills are required?

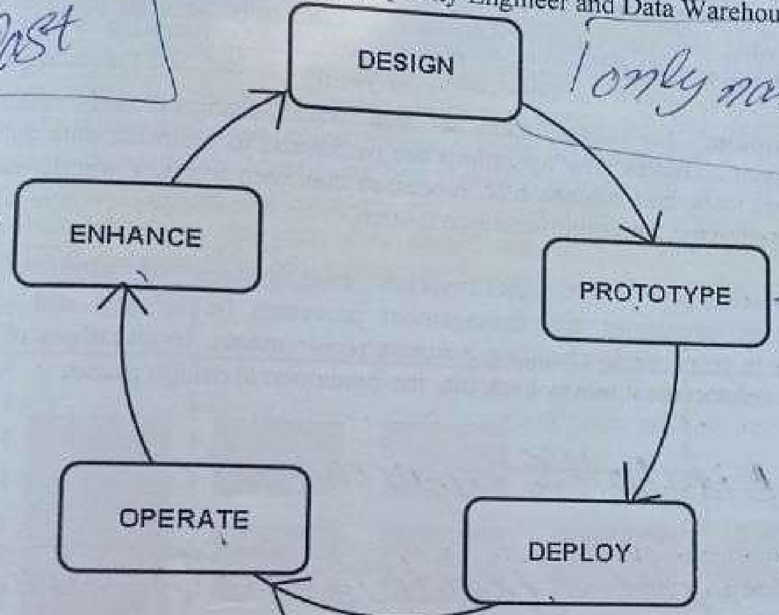
The level of complexity involved in successfully designing and implementing a data warehouse must not be underestimated. Time must be spent to acquire and develop additional skills for data warehousing developers and users. Some options are:

- Invest in just-in-time training (provided by data warehousing tool vendors)
- Use pilot projects as seeds for new technology training
- Develop reward systems that encourage experimentation
- Use outside system integrators and individual consultants

As additional motivation for data warehousing team members, a new class of job titles is being created. Companies are beginning to use dedicated titles such as: Data Warehouse Steward, Data Warehouse Architect, Data Quality Engineer and Data Warehouse Auditor.

Past

only names



بڑھانا

نمونہ یا مثال

کھولنا

Figure-32.1: DWH Development Cycle

Although specific vocabularies vary from organization to organization, the data warehousing industry is in agreement of the fundamental data warehouse lifecycle model as shown in Figure 32.1. The cyclic model consists of 5 major steps described as follows

MCQ

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

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. It servers a means of making the user more expressive about requirements and developing in them a clear understanding and vision of the system. Prototypes are meant to be copied into production models. Once the prototypes are successful, begin the development processing using development tools, such as C, C++, Java, SQL, etc.

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. Note that rapid prototyping can be a subset of the structured development approach. This approach applies a more disciplined approach to the data warehouse development. Documentation requirements are larger, quality control is critical, and the number of reviews increases. While some parts may seem like overkill at the time, they can save a project from problems, especially late in the development cycle.

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. Requirements are understood AFTER the data warehouse has been populated with data and results of queries have been analyzed by the end users. Thus the data warehouse development strategy is based on the analysis of the corporate data model and relevant transactions. This is an extreme approach ignoring the needs of data warehouse users a priori. Consequently company goals and user requirements are not reflected at all in the first cycle, and are integrated in the second cycle.

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. This approach is suitable only well when business processes are designed throughout the company and are combined with business goals.

Kimball also proposes a four-step approach where he starts to choose a business process, takes the grain of the process, and chooses dimensions and facts. He defines a business process as a major operational process in the organization that is supported by some kind of legacy system (or systems). We will discuss this in great detail in lectures 33-34.

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

TDA → Tracks / Parallel Tracks = TDA
PBDM

specific beginning and end. Ongoing project management serves as a foundation to keep the remainder of the lifecycle on track.

33.2 DWH Lifecycle: Key steps

MCQs

Long in Past

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

Lifecycle Key Steps
Lifecycle begins with project planning during which we assess the organization's readiness for a data warehouse initiative, establish the preliminary scope and justification, obtain resources, and launch the project.

The second major task focuses on business requirements definition. The two-way arrow between project planning and business requirements definition (as shown in Figure 33.1) shows the much interplay between these two activities. Data warehouse designers must understand the needs of the business and translate them into design considerations. Business users and their requirements have an impact on almost every design and implementation decision made during the course of a warehouse project. In road map, this is reflected by the three parallel tracks that follow.

The top track deals with technology. Technical architecture design establishes the overall framework to support the integration of multiple technologies. Using the capabilities identified in the architecture design as a shopping list, we then evaluate and select specific products.

The middle track emanating from business requirements definition focuses on data. We begin by translating the requirements into a dimensional model which is then transformed into a physical structure. Physical design activities focus on, performance tuning strategies, such as aggregation, indexing, and partitioning. Last but not least, data staging Extract-Transform-Load (ETL) processes are designed and developed.

The final set of tasks spawned by the business requirements definition is the design and development of analytic applications. The data warehouse project isn't done when we [illegible] آپ

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.

Kimball's Approach:

(i) Kimball's iterative Datawarehouse Development approach develop the bussiness Dimensional lifecycle.

Parallel Tracks = TDA
PBDM

⇒ Key steps

AFS JT

FAST

deliver data. Analytic applications, in the form of parameter-driven templates and analyses, will satisfy a large percentage of the analytic needs of business users.

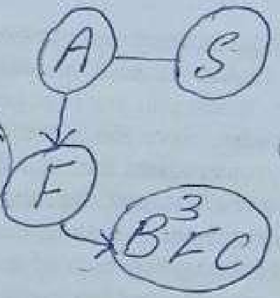
Note:

1. Equally sized boxes (as shown in Figure 33.1) don't represent equally sized efforts, there is a vast difference in the time and effort required for each major activity
2. Data warehousing is an ongoing process, each implementation project should have a cycle with a specific beginning and an end.

33.3 DWH Lifecycle- Step 1: Project Planning

only names

- Assessing Readiness
  - Factors
    - Business sponsor
    - Business motivation
    - Feasibility
    - Business/IT relationship
    - Culture
- Scoping



FASTJ

The DWH lifecycle begins with the project planning phase. It consists of multiple activities that must be performed before proceeding ahead in the lifecycle. Let's discuss the planning phase in detail;

Readiness and risk assessment: Before proceeding ahead with significant data warehouse expenditures, it is prudent to assess the organization's readiness to proceed. Five factors have been identified as leading indicators of data warehouse success; any shortfalls represent risks or vulnerabilities. Brief description in rank order of importance follows.

Business Sponsor: It is the most critical factor for successful data warehousing. Business sponsors should have a clear vision for the potential impact of a data warehouse on the organization. They should be passionate and personally convinced of the project's value while realistic at the same time. Optimally, the business sponsor has a track record of success with other internal initiatives. He or she should be a politically astute leader who can convince his or her peers to support the warehouse.

Business motivation: The second readiness factor is having a strong, compelling business motivation for building a data warehouse. This factor often goes hand in hand with sponsorship. A data warehouse project can't merely deliver a nice-to-have capability; it needs to solve critical business problems in order to garner the resources required for a successful launch and healthy lifespan.

Feasibility: There are several aspects of feasibility, such as technical or resource feasibility, but data feasibility is the most crucial. Are we collecting real data in real operational source systems to support the business requirements? Data feasibility is a

(i) Road map Ralph Kimballs Approach (ii) goal driven approach (iii) result of decades of practical experience (iv) it is believed that everyone on the project team needs a high-level understanding of the complete lifecycle of a Data warehouse.

Lecture 33: DWH Implementation: Goal Driven Approach

Learning Goals

- Business Dimensional Lifecycle
- The Road Map Ralph Kimball's Approach
- DWH Lifecycle

(Lecture based on "The data warehousing toolkit by Ralph Kimball and Margy Ross, 2nd Edition)

33.1 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 a 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.

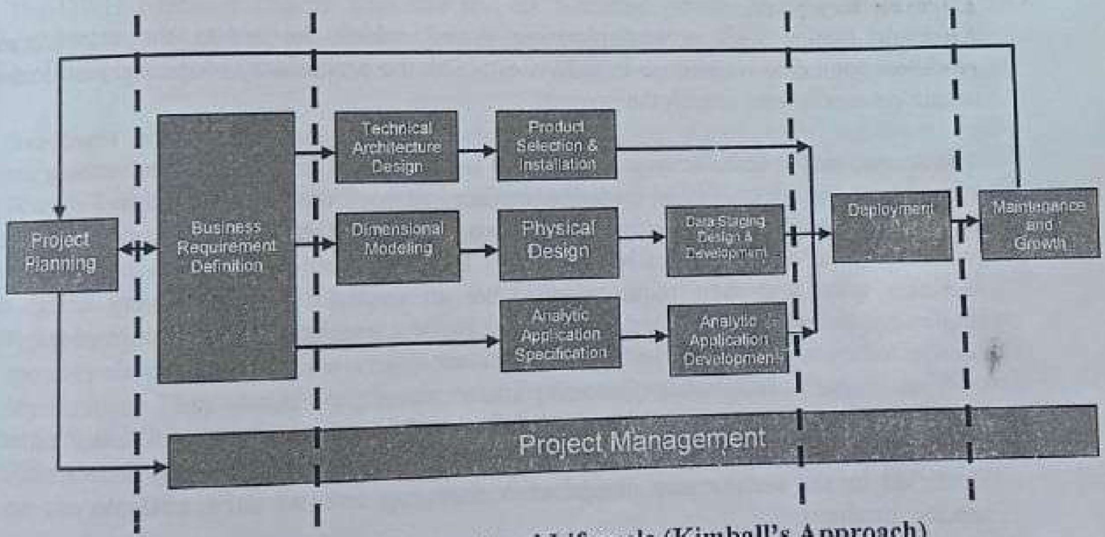

Project Planning
Business Requirement Definition
Technical Architecture Design
Product Selection & Installation
Dimensional Modeling
Physical Design
Data Staging Design & Development
Analytic Application Specification
Analytic Application Development
Deployment
Maintenance and Growth
Project Management

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

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

- Prioritization
  - Review & Prioritize findings
  - Quadrant Technique

Cong

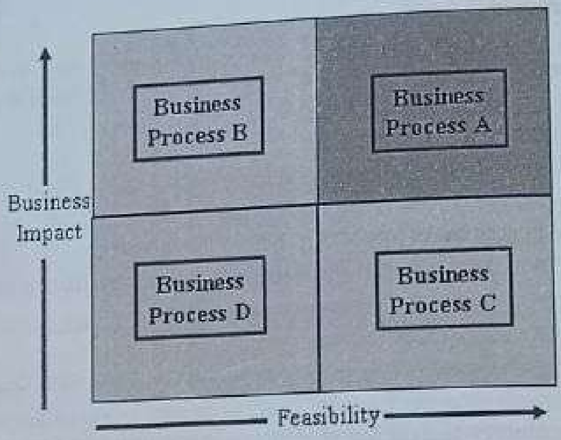


Figure -33.2: The quadrant method

So the process having higher feasibility and impact is given higher priority over the process having lower feasibility and impact. In example of Figure 33.2, process A has highest priority while the process D has lowest priority.

major concern because there is no short-term fix if we're not already collecting reasonably clean source data at the right granularity.

Business/IT relationship: The fourth factor focuses on the relationship between the business and IT organizations. In your company, does the IT organization understand and respect the business? Conversely, does the business understand and respect the IT organization? The inability to honestly answer yes to these questions doesn't mean that you can't proceed. Rather, the data warehouse initiative can be an opportunity to mend the fence between these organizations, assuming that both deliver.

Culture: The final aspect of readiness is the **current analytic culture** within your company. Do business analysts make decisions based on facts and figures, or are their decisions based on intuition, anecdotal evidence, expert judgment or gut feeling?

Scoping: Requires the joint input of both the IT organization and business management. The scope should be both meaningful in terms of its value to the organization and manageable. When you are first getting started, you should focus on data from a single business process. Save the more challenging, cross-process projects for a later phase. Sometimes project teams feel that the delivery schedule is cast in concrete before project planning is even initiated. The prioritization process can be used to convince IT and business management that adjustments are required. Finally, remember to avoid the *law of too* when scoping-too firm of a commitment to too brief of a timeline involving too many source systems and too many users in too many locations with too diverse analytic requirements.

DWH Lifecycle- Step 1: Project Planning

- Justification (cost vs. benefit)
- Team development
- Project Plan
  - Identifying all tasks.
  - User acceptance, milestones and deliverables.
  - Making and following a communication matrix.
  - Avoiding scope creep.
  - Partnership with business user.
- Keys to project planning & Management
  - Business sponsor
  - Scope
  - Best team
  - Excellent project manager

Justification requires an estimation of the benefits and costs associated with a data warehouse. The anticipated benefits grossly outweigh the costs. IT usually is responsible for deriving the expenses. You need to determine approximate costs for the requisite hardware and software. Data warehouses tend to expand rapidly, so be sure the estimates allow some room for short-term growth.

Also, lessons learned from prior information delivery projects, as well as the organization's willingness to accommodate operational change on behalf of the warehouse, can be uncovered such as identifying updated transactions in the source system.

Short in Past Paper

3. **Document Architecture Requirements** of Kimball's model:

Imp

Once the business requirements definition process is leveraged and supplemental IT interviews conducted, the findings need to be documented. A simplistic MATRIX can be used for this purpose. The rows of the matrix list each business requirement that has an impact on the architecture, while matrix columns contain the list of architectural implications.

As an example supposes that a business is spread globally and there is a need to deliver global sales performance data on a nightly basis. The technical implications might include 24/7 worldwide availability, data mirroring for loads, robust metadata for support global access, adequate network bandwidth, and sufficient staging horsepower to handle the complex integration of operational data and so on.

DWH Lifecycle- Step 3.1: Technology Track

4. Develop a high-level Arch. Model DAM

- Several days of heavy thinking in conference room.
- Grouping of requirements (data staging, data access, meta)
- High level refinement (up-front, nuts-bolts hidden) of major systems.

5. Design and Specify the Subsystems DSS

- For each subsystem (data staging), detailed list of capabilities.
- Do research (internet, peers etc.) more graphic models generated.
- Also consider security, physical infrastructure, and configuration.
- Sometimes infrastructure i.e. HW and SW pre-determined.
- Determine Architecture Implementation Phases.
- For more than 1TB DWH, revisit infrastructure.

After the architecture requirements have been documented, models are formulated to support the identified needs the architecture task force often sequesters itself in a conference room for several days of heavy thinking. The team groups the architecture requirements into major components, such as data staging, data access, metadata, and infrastructure. From there the team drafts and refines the high-level architectural model. This drawing is similar to the front elevation page on housing blueprints. It illustrates what the warehouse architecture will look like from the street, but it is dangerously simplistic because significant details are embedded in the pages that follow.

It is time now to do a detailed design of the subsystems. For each component, such as data staging services, the task force will document a laundry list of requisite capabilities. The more specific, the better, because what's important to your data warehouse is not necessarily critical to mine. This effort often requires preliminary research to better understand the market. Fortunately, there is no shortage of information and resources available on the Internet, as well as from networking with peers. The subsystem specification results in additional detailed graphic models. In addition to documenting the

34.2 DWH Lifecycle- Step 3.1: Technology Track

8-Step Process

1. Establish an Architecture Task Force (2-3 people) EAT.

2. Collect Architecture-Related Requirements CAR
   - Business needs => HW not other way round
   - Architectural implications of business needs
   - Timing, performance and availability needs
   - Talk to IT people for current standards, directions and boundaries.
   - Lessons learned.

3. Document Architecture Requirements DAR
   - Make a matrix (row business process & column architectural implication)
   - Global sales coverage => 24/7 availability, data mirroring, adequate network bandwidth etc.

8 Step Process

Data warehouse teams approach the technical architecture design process from opposite ends of the spectrum. Some teams are so focused on data warehouse delivery that the architecture feels like a distraction and impediment to progress and eventually, these teams often end up rebuilding. At the other extreme, some teams want to invest two years designing the architecture while forgetting that the primary purpose of a data warehouse is to solve business problems, not address any plausible (and not so plausible) technical challenge. Neither end of the architecture spectrum is healthy; the most appropriate response lies somewhere in the middle. Kimball suggests an eight-step process for building technical architecture. All steps will be discussed in detail one by one.

1. Establish an Architecture Task Force

It is most useful to have a small task force of two to three people focus on architecture design. Typically these are technical architect, the data staging designer and analytic application developer. This group needs to establish its charter and deliverables time line. It also needs to educate the rest of the team (and perhaps others in the IT organization) about the importance of architecture.

2. Collect Architecture-Related Requirements:

Defining the technical architecture is not the first box in the lifecycle diagram, as shown in Figure 34.1. This implies that the architecture is created to support high value business needs; it's not meant to be an excuse to purchase the latest, greatest products.

The key input into the design process should come from the business requirements definition findings with a slightly different filter to drive the architecture design. The focus is to uncover the architectural implications associated with the business's critical needs e.g. like any timing, availability, and performance needs.

In addition to leveraging the business requirements definition process, additional interviews within the IT organization are also conducted. These are purely technology-focused sessions to understand current standards, planned technical directions, and nonnegotiable boundaries.

capabilities of the primary subsystems, we also must consider our security requirements, as well as the physical infrastructure and configuration needs. Often, we can leverage enterprise-level resources to assist with the security strategy. In some cases the infrastructure choices, such as the server hardware and database software, are predetermined. At this point in time we must have got an idea of what should be the implementation steps/phases that will be used for the DWH implementation. However, if building a large data warehouse, over 1 TB in size, we should revisit these infrastructure platform decisions to ensure that they can scale as required. Size, scalability, performance, and flexibility are also key factors to consider when determining the role of OLAP cubes in the overall technical architecture.

DWH Lifecycle- *Step 3.1: Technology Track*

6. Determine Architectural implementation phases DAI,
   - Can't implement everything simultaneously.
   - Some are negotiable mandatory, others nice-to-haves, later.
   - Business requirements set the priority.
   - Priorities assigned by looking at all the requirements.

7. Document the Technical Architecture DTA
   - Document phases decided.
   - Material for those not present in the conference room.
   - Adequate details for skilled professional (carpenter in kitchen)

8. Review and finalize the technical architecture RFTa
   - Educate organization, manage expectations.
   - Communicate to varying level of details to different levels of team
   - Subsequently, put to use immediately for product selection

Like the Kitchen's analogy, we likely can't implement all aspects of the technical architecture at once. Some are nonnegotiable mandatory capabilities, whereas others are nice-to-haves that can be deferred until a later date. Again, we refer back to the business requirements to establish architecture priorities. Business requirements drive the architecture and not the other way round. We must provide sufficient elements of the architecture to support the end-to-end requirements of the initial project iteration. It would be ineffective to focus solely on data staging services while ignoring the capabilities required for metadata and access services.

We need to document the technical architecture, including the planned implementation phases, for those who were not sequestered in the conference room. The technical architecture plan document should include adequate detail so that skilled professionals can proceed with construction of the framework, much like carpenters frame a house based on the blueprint. Eventually the architecture building process (Technology Track) comes to an end.

With a draft plan in hand, the architecture task force is back to educating the organization and managing expectations. The architecture plan should be communicated, at varying levels of detail, to the project team, IT colleagues, business sponsors, and business leads. Following the review, documentation should be updated and put to use immediately in the product selection process.

Data warehouse

Narrow options to a short list and perform detailed evaluations. Despite the plethora of products available in the market, usually only a small number of vendors can meet both our functionality and technical requirements. By comparing preliminary scores from the evaluation matrix, we should focus on a narrow list of vendors about whom we are serious and disqualify the rest. Once we're dealing with a limited number of vendors, we can begin the detailed evaluations. Business representatives should be involved in this process if we're evaluating data access tools. As evaluators, we should drive the process rather than allow the vendors to do the driving. We share relevant information from the architecture plan so that the sessions focus on our needs rather than on product bells and whistles. Be sure to talk with vendor references, both those provided formally and those elicited from your informal network. If possible, the references should represent similarly sized installations.

DWH Lifecycle- *Step 3.1: Technology Track*

3.1.2 Product selection and Installation

- Conduct prototype, if necessary — cp
  - If one clear winner bubbles up, it is good.
  - Winner due to experience, relationship, commitment
  - Prototype with no more than two products
  - Demonstrate using a limited, yet realistic application using flat text file.
- Keep the competition "hot"
  - Even if single winner, keep at least two in
  - Use virtual competition to bargain with the winner

Short in Post

- Select product, install on trial, and negotiate
  - Make private not public commitment.
  - Don't let the vendor you are completely sold.
  - During *trial period*, put to real use.
  - Near the end of trial, negotiate.

Conduct prototype, if necessary: After performing the detailed evaluations, sometimes a clear winner bubbles to the top, often based on the team's prior experience or relationships. In other cases, the leader emerges due to existing corporate commitments. In either case, when a sole candidate emerges as the winner, we can bypass the prototype step. If no vendor is the apparent winner, we conduct a prototype with no more than two products.

Keep the competition "hot": Even if a single winner is left, it is a good piece of advice that always keep at least two. What if you keep one? The sole vendor may take benefit of the situation that he is the only player and create a situation favorable for him. He might get an upper hand in the bargaining process, and mold things according to his facility and benefit. To avoid such a situation enlist a competitor too, even if a single vendor is the winner. This will create a competitive environment which may ultimately turn into your favor.

cs

cs cs

cs cs c cs cs cs cs

paper only names

DWH Lifecycle- *Step 3.1: Technology Track*

3.1.2 Product selection and Installation

MUNP

- UCP— Understand corporate purchasing process
- Pem— Product evaluation matrix
  - Not too vague/generic
  - Not too specific

UCP CP

Pem MR

- MR— Market research (own ugly son)
  - Understand players and offerings
  - Internet, colleagues, exhibitions etc.
  - RFP is an option, but time consuming and beauty contest

NOP

- NOP— Narrow options, perform detailed evaluations
  - Few vendors can meet tech.& functional requirements
  - Involve business reps.
  - You drive the process, not the vendors.
  - Centered around needs, not bells-and-whistles.
  - Talk to references of similar size installations.

Understand the corporate purchasing process: The first step before selecting new products is to understand the internal hardware and software purchase approval processes, whether we like them or not. Perhaps expenditures need to be approved by the capital appropriations committee. Or you may be asked to provide a bank guarantee against the funds released to buy hardware.

Develop a product evaluation matrix: Using the architecture plan as a starting point, we develop a spreadsheet-based evaluation matrix that identifies the evaluation criteria, along with weighting factors to indicate importance. The more specific the criteria, the better. If the criteria are too vague or generic, every vendor will say it can satisfy our needs. On the other hand, if the criterion is too specific, everyone will shout favoritism.

Conduct market research: We must be informed buyers when selecting products, which mean more extensive market research to better understand the players and their offerings. We must not place the ball in vendor's court because he will never bring forth limitations of his tool. Its like once a *Badsha Salamat* asked his *Wazir* to bring the most beautiful child of his Kingdom. *Wazir* returned thrice with the same boy who was ugly. *Badshah* warned his *Wazir* of severe consequences and gave him yet another chance to search. *Wazir* returned with the same boy again. *Badshah* was astonished and angry (at the same time) and asked his *Wazir* why he was bringing the same ugly boy again and again although he was not up to the standards? *Wazir* replied that he had walked around all over the town, but couldn't find anyone as beautiful as that boy, who was his son. Thus we must not rely on vendors and must make self efforts to gain as much insight into the tools as possible. For this purpose, we can use potential research sources including the Internet, industry publications, colleagues, conferences, vendors, exhibitions and analysts (although be aware that analyst opinions may not be as objective as we're lead to believe).

be critical to this prioritization process. While 15 applications may not sound like much, the number of specific analyses that can be created from a single template merely by changing variables will surprise you.

Before we start designing the initial applications, it's helpful to establish standards for the applications, such as common pull-down menus and consistent output look and feel. Using the standards, we specify each application template, capturing sufficient information about the layout, input variables, calculations, and breaks so that both the application developer and business representatives share a common understanding.

During the application specification activity, we also must give consideration to the organization of the applications. We need to identify structured navigational paths to access the applications, reflecting the way users think about their business. Leveraging the Web and customizable information portals are the dominant strategies for disseminating application access.

DWH Lifecycle- *Step 3.3: Analytic Applications Track*

- **3.3.2 Analytic applications development**
  - Standards: naming, coding, libraries etc.
  - Coding begins AFTER DB design complete, data access tools installed, subset of historical data loaded.
  - Tools: Product specific high performance tricks, invest in tool-specific education.
  - Benefits: Quality problems will be found with tool usage => staging.
  - Actual performance and time gauged.

2MCQS in this Paragraph

When we move into the development phase for the analytic applications, we again need to focus on standards. Standards for naming conventions, calculations, libraries, and coding should be established to minimize future rework. The application development activity can begin once the database design is complete, the data access tools and metadata are installed, and a subset of historical data has been loaded. The application template specifications should be revisited to account for the inevitable changes to the data model since the specifications were completed.

MCQ

Each tool on the market has product-specific tricks that can cause it to literally walk on its head with eyes closed. Therefore, rather than trying to learn the techniques via trial and error, you should invest in appropriate tool-specific education or supplemental resources for the development team.

While the applications are being developed, several ancillary benefits result. Application developers, armed with a robust data access tool, quickly will find needling problems in the data haystack despite the quality assurance performed by the staging application. This is one reason why we prefer to get started on the application development activity prior to the supposed completion of staging. Of course, we need to allow time in the schedule to address any flaws identified by the analytic applications. The developers also will be the first to realistically test query response times. Now is the time to begin reviewing our

Select product, install on trial, and negotiate: It is time to select a product. Rather than immediately signing on the dotted line, preserve your negotiating power by making a private, not public, commitment to a single vendor. Embark on a *trial period* where you have the opportunity to put the product to real use in your environment. As the trial draws to a close, you have the opportunity to negotiate a purchase that's beneficial to all parties involved.

34.3 DWH Lifecycle- Step 3.3: Analytic Applications Track

- **Overview**
  - Design and develop applications for analysis.
  - It is really the "fun part".
  - Technology used to help the business.
  - Strengthen relationship between IT and business user.
  - The DWH "face" to the business user.
  - Querying NOT completely ad-hoc.
  - Parameter driven querying satisfy large % of needs.
  - Develop consist analytic frame-work instead of shades of Excel macros.

The final set of parallel activities following the business requirements definition in Figure 34.1 is the analytic application track, where we design and develop the applications that address a portion of the users' analytic requirements. As a well-respected application developer once told, "Remember, this is the fun part!" We're finally using the investment in technology and data to help users make better decisions. The applications provide a key mechanism for strengthening the relationship between the project team and the business community. They serve to present the data warehouse's face to its business users, and they bring the business needs back into the team of application developers.

While some may feel that the data warehouse should be a completely ad hoc query environment, delivering parameter-driven analytic applications will satisfy a large percentage of the business community's needs. There's no sense making every user start from scratch. Constructing a set of analytic applications establishes a consistent analytic framework for the organization rather than allowing each Excel macro to tell a slightly different story. Analytic applications also serve to encapsulate the analytic expertise of the organization, providing a jump-start for the less analytically inclined.

DWH Lifecycle- ***Step 3.3: Analytic Applications Track***

- **3.3.1 Analytic applications specification**
  - Starter set of 10-15 applications.
  - Prioritize and narrow to critical capabilities.
  - Single template use to get 15 applications.
  - Set standards: Menu, O/P, look feel.
  - From standard: Template, layout, I/P variables, calculations.
  - Common understanding between business & IT users.

Long 'Em
Past

Following the business requirements definition, we need to review the findings and collected sample reports to identify a starter set of approximately 10 to 15 analytic applications. We want to narrow our initial focus to the most critical capabilities so that we can manage expectations and ensure on-time delivery. Business community input will

- The data design is finished before participants have experimented with the tools and live data. As we have discussed at length in lecture no. 33, involve the business users from the very beginning, get user requirement definition, record it and follow it.

35.2 Eleven Possible Pitfalls

Past

- **1. Weak business sponsor:** Getting stuck by office politics, need CXO on your side.
- **2. Not having multiple servers:** Penny wise pound Foolish. (i) Both going down (ii) Performance degradation
- **3. Modeling without domain expert:**
- **4. Not enough time for ETL:** Giving too little time or over complicating by including everything conceivable. Users will forgive:
  - Less Formatting, slow system, few features, few reports BUT NOT incorrect results

11-Possible pitfalls in DWH Life Cycle & Development

Many early data warehousing projects failed, having fallen into one or more of the traps we will discuss. These pitfalls are still difficult to avoid, unless those steering the project are able to understand and anticipate the associated risks.

1. Weak business sponsor

This phase often turns out to be the trickiest phase of the data warehousing implementation and is also the Part-II(a) of your semester project. Because data warehousing by definition includes data from multiple sources spanning many different departments within the enterprise. Therefore, there are often political battles that center on the willingness of information sharing. Even though a successful data warehouse benefits the enterprise, there are occasions where departments may not feel the same way. As a result of unwillingness of certain groups to release data or to participate in the data warehousing requirements definition, the data warehouse effort either never gets off the ground, or could not get started in the right direction defined originally. When this happens, it would be ideal to have a strong business sponsor. If the sponsor is at the CXO level (X: Information, Knowledge, Financial etc), he/she can often exert enough influence to make sure everyone cooperates.

2. Not having multiple servers

This is a classical example of penny wise and pound foolish. To save capital, often data warehousing teams will decide to use only a single database and a single server for the different environments i.e. development and production. Environment separation is achieved by either a directory structure or setting up distinct instances of the database. This is awkward for the following reasons:

- Sometimes it is possible that the server needs to be rebooted for the development environment. Having a separate development environment will prevent the production environment from being effected by this.

Data ware house given a macro view of data

غلطیاں خطرے

Lecture 35: DWH Life Cycle: Pitfalls, Mistakes, Tips

Learning Goals

- Possible pitfalls in DWH Life Cycle & Development
- Common Data warehouse mistakes to avoid
- Key Steps for a smooth DWH implementation
- Conclusions

In this lecture we will discusses the problems, troubles and mistakes that commonly occur while building a data warehouse. We will discuss things to do, and also things not to do. We will discuss ways to avoid common mistakes that may halt data warehousing process.

35.1 Five Signs of trouble

1. Project proceeded for two months and nobody has touched the data.
2. End users are not involved hands-on from day one throughout the program.
3. IT team members doing data design (modelers and DBAs) have never used the access tools.
4. Summary tables defined before raw atomic data is acquired and base tables have been built.
5. Data design finished before participants have experimented with tools and live data.

Signs of trouble

First of all we will discuss "5 signs of trouble". Any of these signs if present, will serve as a key indicator that the data warehousing project is under threat. The following situations indicate a project in trouble:

- The project has proceeded for two months and nobody has even touched the data. Before even embarking on the project, the team should have had a thorough understanding and look and feel of the data. As I always tell my students, "know your data intimately".

- The future consumers are not involved hands-on from day one throughout the program. Working in isolation will result in systems that no one will accept, and consequently none is going to use, resulting in negative marketing for you and your company. Thus avoid this at all costs.

- The team members doing data design (modelers and DBAs) have never used the access tools. You need experienced campaigners not "green apples" or raw hands. You need people who know the job, not those who will learn on the job, and in the process sink the ship.

- The summary tables are defined before the raw atomic data is acquired and base tables have been built. A converse process has been followed, which again is a recipe for disaster.

- There may be interference while having different database environments on a single server. For example, having multiple long queries running on the development server could affect the performance on the production server, as both are same.

3. Modeling without domain expert

It is essential to have a subject-matter expert as part of the data modeling team. This person can be an outside consultant or can be someone in-house with extensive industry experience. Without this person, it becomes difficult to get a definitive answer on many of the questions, and the entire project gets dragged out, as the end users may not always be available.

4. Not enough time for ETL

This is common everywhere, ETL getting the least time, remember data is always dirtier than you think. There is a tendency to give this particular phase of DWH too little time and other resources. This can prove suicidal to the project, as the end users will usually tolerate less formatting, longer time to run reports, less functionality (slicing and dicing), or fewer delivered reports; one thing that they will never ever tolerate is wrong information.

A second common problem is that some people unnecessarily make the ETL process complicated. In ETL design, the primary goal should be to optimize load speed without sacrificing on quality. This is, however, sometimes not followed. There are cases when the design goal is to cover all possible future uses and possible scenarios, some of which may be practical, while others just plain impractical. When this happens, ETL performance suffers, and often so does the performance of the entire data warehousing system.

11-Possible Pitfalls (continued)

- **5. Low priority for OLAP Cube Construction**: Giving it the lowest priority.
- **6. Fixation with technology:** End users impressed by timely information NOT advanced infra-structure.
- **7. Wrong test bench:** Required performance NOT on fast production level machines.
- 8. QA people NOT DWH literate: Ensure QA people are educated about DWH.

5. Low priority for OLAP Cube Construction

Make sure your OLAP cube-building or pre-calculation process is optimized and given the right priority. It is common for the data warehouse to be on the bottom of the nightly batch loads, and after the loading the DWH, usually there isn't much time left for the OLAP cube to be refreshed. As a result, it is worthwhile to experiment with the OLAP cube generation paths to ensure optimal performance.

6. Fixation with technology

- There may be interference while having different database environments on a single server. For example, having multiple long queries running on the development server could affect the performance on the production server, as both are same.

3. Modeling without domain expert

It is essential to have a subject-matter expert as part of the data modeling team. This person can be an outside consultant or can be someone in-house with extensive industry experience. Without this person, it becomes difficult to get a definitive answer on many of the questions, and the entire project gets dragged out, as the end users may not always be available.

4. Not enough time for ETL

This is common everywhere, ETL getting the least time, remember data is always dirtier than you think. There is a tendency to give this particular phase of DWH too little time and other resources. This can prove suicidal to the project, as the end users will usually tolerate less formatting, longer time to run reports, less functionality (slicing and dicing), or fewer delivered reports; one thing that they will never ever tolerate is wrong information.

A second common problem is that some people unnecessarily make the ETL process complicated. In ETL design, the primary goal should be to optimize load speed without sacrificing on quality. This is, however, sometimes not followed. There are cases when the design goal is to cover all possible future uses and possible scenarios, some of which may be practical, while others just plain impractical. When this happens, ETL performance suffers, and often so does the performance of the entire data warehousing system.

11-Possible Pitfalls (continued)

- 5. **Low priority for OLAP Cube Construction**: Giving it the lowest priority.
- 6. **Fixation with technology:** End users impressed by timely information NOT advanced infra-structure.
- 7. **Wrong test bench:** Required performance NOT on fast production level machines.
- 8. QA people NOT DWH literate: Ensure QA people are educated about DWH.

5. Low priority for OLAP Cube Construction

Make sure your OLAP cube-building or pre-calculation process is optimized and given the right priority. It is common for the data warehouse to be on the bottom of the nightly batch loads, and after the loading the DWH, usually there isn't much time left for the OLAP cube to be refreshed. As a result, it is worthwhile to experiment with the OLAP cube generation paths to ensure optimal performance.

6. Fixation with technology

impediment to progress and eventually, these teams often end up rebuilding. At the other extreme, some teams want to invest two years designing the architecture while forgetting that the primary purpose of a data warehouse is to solve business problems, not address any plausible (and not so plausible) technical challenge. Neither end of the architecture spectrum is healthy; the most appropriate response lies somewhere in the middle.

35.3 Top 10-Common Mistakes to Avoid

Post

- **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.*

10-Common Data Warehouse mistakes to avoid

So far you have been told what to do, however now we'll balance those recommendations with a list of what not to do. When building and managing a data warehouse, the common mistakes to avoid are listed. These mistakes are described as a series of negative caricatures. The goal is for you to learn from these as George Santayana said, "Those who cannot remember the past are condemned to repeat it." Let's all agree not to repeat any of these mistakes. Each of the mistakes will be discussed one by one.

Mistake 1: *Not interacting directly with the end users;* your job is to be the publisher of the right data. To achieve your job objectives, you must listen to the business users, who are always right. Nothing substitutes for direct interaction with the users. Develop the ability to listen.

Mistake 2: *Promising an ambitious data mart as the first deliverable;* these kinds of data marts are 'consolidated, second-level marts with serious dependencies on multiple sources of data. Customer profitability requires all the sources of revenue and all the sources of cost, as well as an allocation scheme to map costs onto the revenue! For the first deliverable, focus instead on a single source of data, and do the more ambitious data marts later.

Mistake 3: *Never freezing the requirements i.e. being an accommodating person;* You need to think like a software developer and manage three very visible stages of developing each data mart: (1) the business requirements gathering stage, where every suggestion is considered seriously, (2) the implementation stage, where changes can be accommodated~ but must be negotiated and generally will cause the schedule to slip, and (3) the rollout stage, where project features are frozen. In the second and third stages, you must avoid insidious scope creep (and stop being such an accommodating person).

Just remember that the end users do not care how complex or how technologically advanced your front end (or for that matter back-end) infrastructure is. All they care is that they should receive their information in a timely manner and in the way they specified.

7. Wrong test bench

Make sure the development environment is very similar to the production environment as much as possible - Performance enhancements seen on less powerful machines sometimes do not happen on the larger, production-level machines.

8. QA people NOT DWH literate

As mentioned above, usually the QA team members know little about data warehousing, and some of them may even resent the need to have to learn another tool or tools. Make sure the QA team members get enough education about data warehousing so that they can complete the testing themselves.

11-Possible Pitfalls (continued)

- 9. **Uneducated user:** Take care and address end-user education needs. Intuition does not work.
- 10. **Improper documentation:** Complete documentation before developers leave.
- 11. **Doing incremental enhancements:** Definite no-no. Dev => QA =>Production

9. Uneducated user

Take care and address the user education needs. There is nothing more frustrating to spend several months to develop and QA the data warehousing system, only to have little usage because the users are not properly trained and educated. Regardless of how intuitive or easy the interface may be, it is always a good idea to send the users to at least a one-day training course to let them understand what they can achieve by properly using the data warehouse.

10. Improper documentation

Usually by this time most, if not all, of the developers will have left the project, so it is essential that proper documentation is left for those who are handling production maintenance. There is nothing more frustrating than staring at something another person did, yet unable to figure it out due to the lack of proper documentation.

Another pitfall is that the maintenance phase is usually boring. So, if there is another phase of the data warehouse planned, start on that as soon as possible.

11. Doing incremental enhancements

Because a lot of times the changes are simple to make, it is very tempting to just go ahead and make the change in production. This is a definite no-no. Many unexpected problems will pop up if this is done. It is very strongly recommend that the typical cycle of **Development → QA → Production** be followed, regardless of how simple the change may seem.

Mistake 4: *Working without senior executives in loop, waiting to include them after a significant success;* the senior executives must support the data warehouse effort from the very beginning. If they don't, your organization likely will not be able to use the data warehouse effectively. Get their support prior to launching the project.

Mistake 5: *Doing a very comprehensive and detailed first analysis to do the DWH right the very first time;* Very few organizations and human beings can develop the perfect comprehensive plan for a data warehouse upfront. Not only are the data assets of an organization too vast and complex to describe completely, but also the urgent business drivers will change significantly over the life of the data warehouse. Start with lightweight data warehouse bus architecture of conformed dimensions and conformed facts, and then build your data warehouse iteratively. You will keep altering and building it forever.

Top 10-Common Mistakes to Avoid (Continued...)

- **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.*

Mistake 6: *Assuming the business users will develop their own "killer application" on their own;* Business users are not application developers. They will embrace the data warehouse only if a set of prebuilt analytic applications is beckoning them.

Mistake 7: *Training users on the detailed features of the tool using dummy data and consider it a success;* Delay training until your first data mart is ready to go live on real data. Keep the first training session short, and focus only on the simple uses of the access tool. Allocate more time to the data content and analytic applications rather than to the tool. Plan on a permanent series of beginning training classes and follow-up training classes as well. Take credit for the user acceptance milestone when your users are still using the data warehouse six months after they have been trained.

Mistake 8: *Isolating the IT support people from the end or business users;* Data warehouse support people should be physically located in the business departments, and while on assignment, they should spend all their working hours devoted to the business content of the departments they serve. Such a relationship engenders trust and credibility with the business users.

Mistake 9: *After DWH is finished, holding a planning and communications meeting with end users;* Newsletters, training sessions, and ongoing personal support of the business community should be to gather items for the first rollout of the data warehouse.

large number of transactions. So what is large is this circular logic or a trick question? Neither. Large customers mean tens of thousands of customers and similarly tens and thousands of transactions per week. Note that what we call large on our country, may be small for the developed world. Once you have identified such a company, submit a report, we call this report_2. The report should have four reasons why you have selected a particular company or organization. What could be those four reasons?, this is a good question, the four reasons are (i) the number of customers (ii) the number of transactions (iii) typical early adopter (from next slide) and (iv) any other reason. Once you have submitted the report, you can not just by default move on to Part-II(b) of the project. The company selected must be approved by the instructor before you can proceed ahead.

Large and Typical Early Adopters

1. Financial service/insurance.
2. Telecommunications.
3. Transportation.
4. Government.
5. Educational.

Past

Here you would be asking the question, what is meant by an early adopter? Will discuss this at the end of the lecture, but if you can't wait, please check fig-36.1. For a developing country like ours, we are in the phase of talking about typical early adopters of DWH. However, in the developed world; this stage is no longer there for many large companies, as DWH are now in the mainstream. The types of organizations and business listed are typically those with large number of customers and large number of transactions.

Example DWH Target Organizations

- Financial service/insurance.
  - Union Bank
  - State Bank of Pakistan
- Telecommunications.
  - UFone
  - PTCL
  - PAKNET
- Transportation.
  - PIA
- Government.
  - NADRA

Cong in past &
V.V. Important Question.

For example, as per www.paksearch.com Muslim Commercial Bank has 900+ branches all over Pakistan. With an average of 500 customers per branch, the total number of customers is in the order of half a million. It would not be surprising if the weekly ATM transactions all over Pakistan run into millions. Such banks are potential candidates for a data warehouse. Same is true for telecommunication companies. As per recent government figures, there are 10+ million mobile phone users in Pakistan, and as per www.fdi.com the number of mobile phone users of Mobilink is 3.7 million. Again, it would not be surprising to have literally millions of mobile phone calls made/received per day. So these businesses fall under the category which you should be looking at to select and study as part of your semester project.

to be devising ETL procedures to seek out and retrieve information like this forevermore. You do yourself and the project a great service by establishing a method of doing this right the first time. Have your development people put in the extra time to explore old data thoroughly, characterize "dirty" data issues realistically, and to design and implement robust extraction and transformation procedures exhaustively. The ETL portion of a data warehouse can consume as much as 80 percent of your total project resources! Make sure you spend wisely.

7. Be a diplomat NOT a technologist Short ⇒ Past)

The biggest problem you will face during a warehouse implementation will be people, not the technology or the development. You're going to have senior management complaining about completion dates and unclear objectives. You're going to have development people protesting that everything takes too long ~~and why can't they do it the old way?~~ You're going to have users ~~with outrageously unrealistic expectations~~, who are used to systems that require mouse-clicking but not much intellectual investment on their part. And you're going to grow exhausted, separating out Needs from Wants at all levels. Commit from the outset to work very hard at communicating the realities, encouraging investment, and cultivating the development of new skills in your team and your users (and even your bosses).

Most of all, keep smiling. When all is said and done, you'll have a resource in place that will do magic, and your grief will be long past. Eventually, your smile will be effortless and real.

35.5 Conclusions

✓ Paper

for MCQS

- DWH is not simple.
- DWH is very expensive.
- DWH is not ONLY about technology.
- DWH designers must be capable of working across the organization.
- DWH team requires a combination of many experiences and expertise.

Conclusions

By now you have realized that a building a data warehouse is not an easy task. DWH are very expensive to build, with the average cost of a system valued at around US$ 2 million. Hence, the right people, methodology and experience is critical. The dependence on technology is only a small part in realizing the true business value buried within the mountain of data collected and stored within organizations business systems and operational databases. Data warehouses touch the organization at all levels, and the people that design and build the data warehouse must be capable of working across the organization at all levels as well, thus communication skills of the people are of utmost importance. Thus the key requirements are industry and product experience of a diverse team, coupled with a business focus and proven methodology. This will make the difference between just a functional system and true success story.

نوٹ: کپاس کی فصل میں دو طرح کے کیڑے ہوتے ہیں

(i) Sucking pest → (رس چوسنے والے کیڑے)

یہ کیڑے تب رس چوستے ہیں جب کپاس کے پودے بالکل چھوٹے ہوتے ہیں مثلاً سفید مکھی (white fly) اور Jassid وغیرہ۔

(ii) Boll worms → (کترنے والے کیڑے)

ان کو ٹینڈا کترنے والے کیڑے بھی کہتے ہیں مثلاً گلابی سنڈی (Pink boll-worm) اور لشکری کیڑے (Army boll-worm) وغیرہ۔

Lecture 37: Case Study: Agri-Data Warehouse

Learning Goals

- Impact of Agriculture in Pakistan
- Major Players in Agriculture
- Economic Threshold Level ETL_A
- The Need of IT in Agriculture
- Dimensional Model of Agri-DWH

Data being recorded for decades by several organization, mostly never digitized and never used for decision making. Under-utilization.

Data is horizontally wide i.e. 100+ attributes and vertically deep i.e. tens of thousands of rows.

Huge potential for long-term and short-term decision making.

Decision making not data driven, but based on "expert" judgment, sometimes with tragic results.

Every year different government departments are tasked to monitor dynamic agricultural situations all around Punjab- the breadbasket of Pakistan. As a result, thousands of digital and non digital data files are generated from hundreds of pest-scouting and yield surveys, metrological data recordings and other such undertakings. The data collected, due to its multivariate nature and disparate origins, is hard to integrate and thus does not provide a complete picture. Thus the lack of data integration (and standardization) contributes to an under-utilization of valuable and expensive historical data, and inevitably results in a limited capability to provide decision support and analysis.

In this case study, the implementation of a Pilot Agriculture Data Warehouse (Agri-DWH) is discussed. Such a data warehouse can support decision making using Data Mining and Online Analytical Processing (OLAP). Based on literature review, no such work was found to have been undertaken in the agriculture sector of Pakistan and elsewhere. Data warehouses are quite popular in telecommunications, travel industry, government etc. but an application in agriculture extension is a novel idea. The strength of this novel idea is demonstrated through a pilot implementation and discussion of interesting findings using real data.

37.1 Background

Impact of Agriculture in Pakistan

Pakistan is one of the five major cotton-growing countries in the world. Almost 70% of world cotton is produced in China (Mainland), India, Pakistan, USA and Uzbekistan. As textile exports comprise more than 60% of Pakistan's total exports, the success or failure of cotton crop has a direct bearing on the economy. Cotton production is the inherent comparative advantage of the textile sector of Pakistan; with total textile industry exports amounting to US$ 7 billion and 68% share in export earnings.

semester: is the semester, some possible semesters are F05 i.e. Fall 2005 or SP06 i.e. Spring 2006 or SPL05 i.e. special semester 2005 or SM06 i.e. summer 2006.

campus: The name of the VU campus where you are registered or taking the course along with the name of the city or town. Write full name of the campus, do not use underscore i.e. _ or dashes or space in the campus name.

rollno: Since it is a group project, so roll numbers of students in the group separated by a comma.

CS614: This will be at the end of every file name i.e. the course code.

Don't try to copy-paste or use someone else's work with your name, there are smart tools to catch this, once you are caught, this can result into zero credit.

Why would companies entertain you? Cong in past

- You are students, and whom you meet were also once students.
- You can do an assessment of the company for DWH potential at no cost.
- Since you are only interested in your project, so your analysis will be neutral.
- Your report can form a basis for a professional detailed assessment at a later stage.
- If a DWH already exists, you can do an independent audit.

If you present your case well you are likely to be entertained i.e. welcome by the organization. The first and the foremost reason to get help is, whom you are talking to was once a student too as you are currently, so there is a common bond. Since you have studied well DWH (hopefully), therefore, you can do a requirement assessment (in the form of lifecycle study) of the company at no cost; the company has nothing to lose. Your only interest is completion of your project and grade, so you are going to be very objective and very neutral; hence the company has still nothing to lose. After you have done the lifecycle study, the same can be used as a seed or input by a professional organization for an in depth study, thus saving in money to the company for which you have done the work. Again the company has nothing to lose. It may so happen that the company you contact already has a data warehouse in place, in such a case doing the lifecycle development study can be used as a internal audit of the DWH implementation. Hence in short, if the company allows and helps you with the study, it will be a win-win scenario for both the parties.

Major players in agriculture are insects.

نوٹ: ایسا کیڑا جو فصل کو کھاتا ہے اسے pest کہتے ہیں

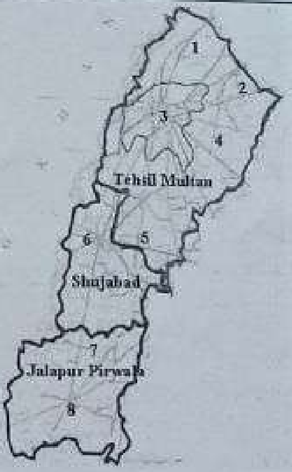
Area under Study

MCQS Past Paper

- Punjab is the bread-basket of Pakistan, and administratively divided into eight divisions.
- Multan division is the cotton hub of Punjab, which consists of district Multan.
- District Multan has three tehsils, which are further divided into central points or Markaz.
- This study is about 10 tehsils of Multan using data for years 2001 and 2002.

Punjab is the breadbasket of Pakistan, and is administratively divided into eight divisions, including Multan division. Multan division is further divided into six districts, including district Multan. District Multan has three *Tehsils*. Within each *Tehsil* are central points or *Markaz*. This work is centered around district Mutlan. The area under study is shown in Figure-37.1.

Figure-37.1(a): Map of Pakistan (www.cia.gov)



Key	Markaz
1	Bosan
2	Qadirpurran
3	Multan
4	Makhdum Rashid
5	Mumtazabad
6	Shujabad
7	Hafizwala
8	Jalalpur Pirwala
9	Qasba Marral

Figure-37.1(b): Area under study District Multan

37.2 Major Players

1 MCQS in Past Paper

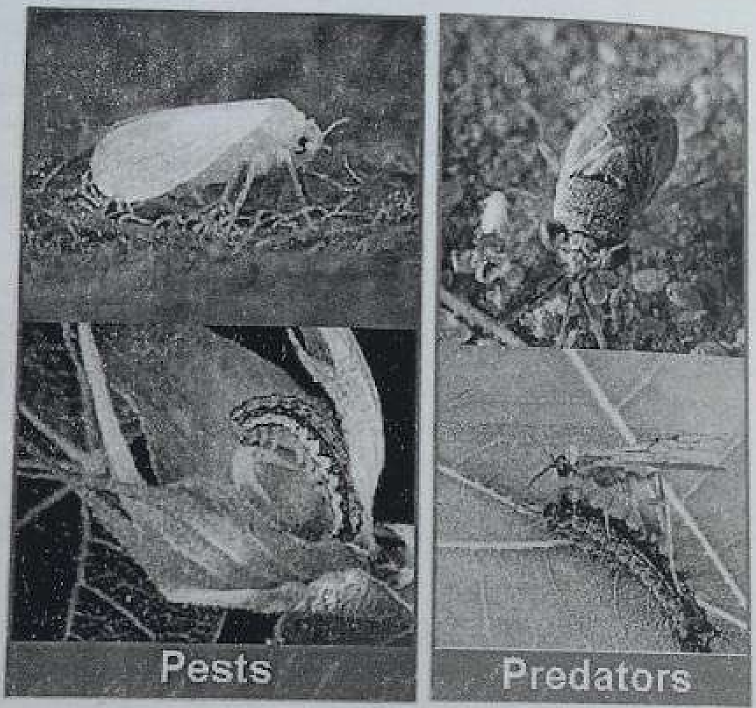
- A pest is an insect that eats the crop. There are two types of cotton pests:
  - Sucking pests (White Fly, Jassid, Thrips etc.)
  - Boll Worms (spotted boll-worm, pink boll-worm)
- A predator is an insect that eats the pest.
  - Such as Lady bug Beetle, Spiders, Ants etc.
- Cotton is also effected by virus

نوٹ: ایسا کیڑا جو Pest کو کھاتا ہے اسے Predator کہتے ہیں

- Cotton Leaf Curl Virus (CLCV) → پتہ مروڑ وائرس

In the context of this case study, a pest is an insect that eats the crop. Since cotton crop is being considered, so some of the pests considered are Jassid, Thrips, SBW (Spotted Ball Worm) etc. A predator is an insect that eats the pests. Some of the cotton pest predators are ladybug beetles, spiders, ants, Assassin Bug etc. A sample of cotton virus, pest and predator are shown in Figure-37.2. Other than pests, the cotton crop is also effected by viruses, the predominant one being CLCV (Cotton Leaf Curl Virus). In this case study, **field** would mean the cultivated land, with certain area and ownership.

Pests & Predators



Source: United States Department of Agriculture (USDA)

Figure 37.2: Cotton pests and predators

37.3 Economic Threshold Level ETL_A

- The pest population beyond which it is cost effective to use pesticide.
- Pesticide is a poison which is used to kill pests.
- Note that eradicating pests is NOT feasible, controlling pest population is feasible.

ETL_A: Economic Threshold Level in agriculture extension is that pest population beyond which the benefit of spraying outweighs its cost. It is highly infeasible and expensive to eradicate all pests, therefore, pest control measure are employed, when pest populations cross a certain threshold. This threshold varies from pest to pest, and from

* every problem is a opportunity

crop to crop. Figure-37.3 shows the ETL_A by a dotted line, and undesired pest populations by "humps" above the said line.

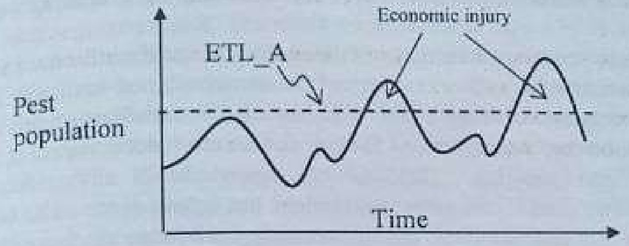


Figure-37.3: Agriculture Economic Threshold Level (ETL_A) and time.

The Need

- Extensive losses to cotton crop due to pest attacks in 1983 resulted in the establishment of Directorate of Pest Warning in 1984.
- Since 1984 scouts from the Directorate have been sampling fields and recording data and advising farmers.
- During 2003-04 season, Boll Worm attack on the cotton crop resulted in a loss of nearly 0.5 M bales.
- Weather not the only factor, but points to a multitude of factors, requiring efficient and effective data analysis, for better decision making. Short in past

Pest scouting is a systematic field sampling process that provide field specific information on pest pressure and crop injury. The pest scouting data is being constantly recorded by the Directorate of Pest Warning and Quality Control of Pesticides (DPWQCP), Punjab since 1984. However, despite pest scouting, yield losses have been occurring. The most recent being the Boll Worm attack on the cotton crop during 2003-04, resulting in a loss of nearly 0.5 million bales. This loss can not be attributed to weather alone, but points to a multitude of factors, requiring efficient and effective data analysis, for better decision making.

37.4 The need: IT in Agriculture

- The volume of pest scouting data accumulated todate is enormous both horizontally and vertically.
- A typical pest scouting sheet consists of 35 variables or attributes.
- موسمی کیفیت
  Metrological data consists of 50+ variables.
- Coarse estimate of pest scouting data recorded for the cotton crop alone stands at 5+ million records, and growing.

Issues: (i) Pest Scouting sheets are larger than flat-bed A4 size Scanners

Lecture 38: Case Study: Agri-Data Warehouse

Learning Goals

- Data Acquisition and Cleansing
- Data Transform, Transport & Populate
- Resolving the Issues
- Deployment & System Management

حصول / وصول

38.1 Step 6: Data Acquisition & Cleansing

Trained scouts from DPWQCP periodically visit randomly selected points and manually note 35 attributes, with some given in Table 2. These hand-written sheets are subsequently filed. For the last 10 years, the data collected was recorded by typing the hand-filled pest scouting sheets. Copy of a hand filled pest scouting sheet is shown in Figure-38.1(a).

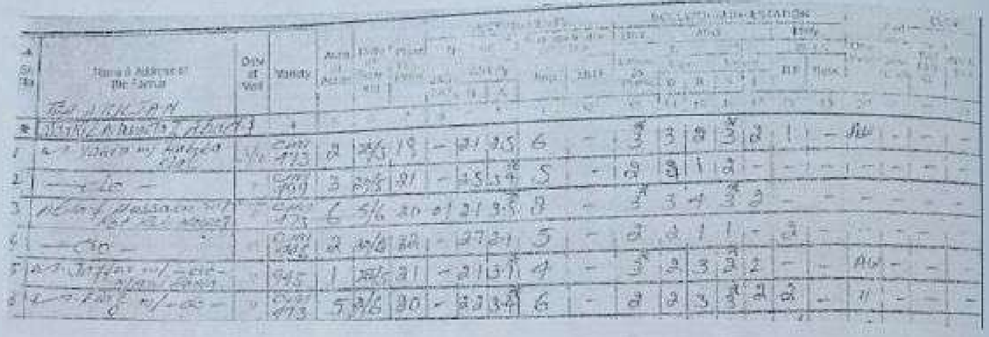
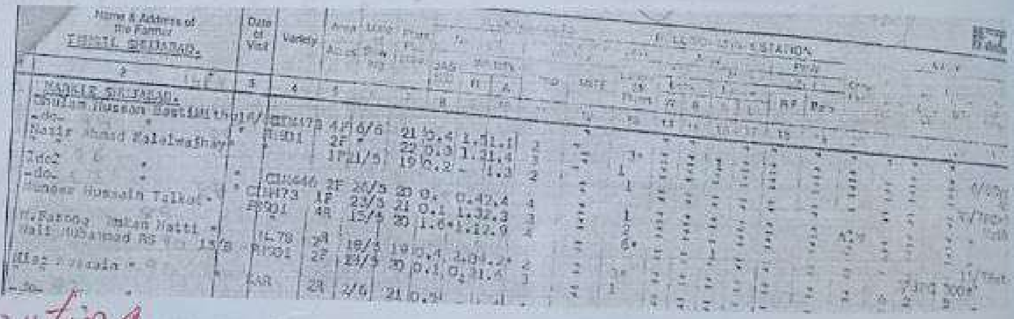


Figure-38.1(a): Hand filled Pest Scouting sheet



Pest Scouting issues:

Figure-38.1(b): Typed Pest Scouting sheet

The * in Figure-38.1 corresponds to pest hot spot or flare-up or ETL_A.

Step-6: Issues Long in Past

1

- The pest scouting sheets are larger than A4 size (8.5" x 11"), hence the right end was cropped when scanned on a flat-bed A4 size scanner.

extensive inventment
بڑے پیمانے پر سرمایہ کاری کرنا

- Tasking the human brain alone, for synthesis of information from this data is not only impractical but dangerous too.

- Need a Data Warehouse, OLAP tools and Data Mining to analyze the data.

The volume of pest scouting data that has been accumulated until now by DPWQCP is enormous both horizontally (scores of factors or attributes) and vertically i.e. number of records. A typical pest scouting sheet consists of 35 variables or attributes. Coarse estimate of pest scouting data recorded for the cotton crop alone stands at more than 7.5 million records, and growing. Tasking the human brain alone, for synthesis of information from this data is not only impractical but is unjust too. The objective of the work discussed in this case study is, complimenting knowledge discovery in this massive data set, using proven information management tools and techniques, so as to support decision making.

*Agriculture is backbone of economy.

Agro-Informatics

- *"I.T. sector is at the heart of the economic revival of Pakistan"* President of Pakistan, Launching of VU, Mar. 23, 2003.

- Agriculture is the backbone of our economy, upto 70% of the population is dependent on it.

- IT is an enabler, and has the potential to benefit everyone when applied in Agriculture.

- IT + Agriculture: A win-win scenario.

full-blown
مکمل تیار

IT touches a fraction of our population, but everyone has to eat and cloth, this agriculture affects everyone. Using IT in agriculture i.e. Agro-Informatics has the potential to benefit everyone, and can bring about an economic revolution in Pakistan, thus everyone will be a winner. To know more about Agro-Informatics, visit **www.nu.edu.pk/cairindex.asp**

How to go about? why pilot stratgy is seommeded for construction of DWH?

(Long)

- Discussed several DWH implementation methodologies in lectures 32-35.
- Will adopt a pilot project approach, because:
  - 1 A full-blown DWH requires extensive investment.
  - 2 Show users the value of DSS.
  - 3 Establish blue print for full-blown system.
  - 4 Identify problem areas.
  - 5 Reveal true data demographics.
  - 6 Pilot projects are supposed to work with limited data.

A pilot project strategy is highly recommended in data warehouse construction, as a full blown data warehouse construction requires significant capital investment, effort and resources. Therefore, the same must be attempted only after a thorough analysis, and a valid proof of concept. A small scale project in this regard serves many purposes such as (i) show users the value of DSS information, (ii) establish blue print processes for later what issues may occur during data acquisition and cleansing in agriculture case study ?

Ans:

- II The right part of the scouting sheet is also the most troublesome, because of pesticide names for a single record typed on multiple lines i.e. for multiple farmers.
- III As a first step, OCR (Optical Character Reader) based image to text transformation of the pest scouting sheets was attempted. But it did not work even for relatively clean sheets with very high scanning resolutions.
- IV Subsequently DEO's (Data Entry Operators) were employed to digitize the scouting sheets by typing.

The pest scouting sheets are larger than A4 size (8.5" x 11"), hence the right end was cropped when scanned on a flat-bed A4 size scanner. The right part of the scouting sheet is also the most troublesome, because of pesticide names for a single record typed on multiple lines i.e. for multiple farmers.

As a first step, OCR (Optical Character Reader) based image to text transformation of the pest scouting sheets was attempted. But it did not work even for relatively clean sheets with very high scanning resolutions, such as 600 dpi. Subsequently DEO's (Data Entry Operators) were employed to digitize the scouting sheets by typing. To reduce spelling errors in pesticide names and addresses, drop down menu or combo boxes with standard and correct names were created and used.

38.2 Step-6: Why the issues?

- Major issues of data cleansing had arisen due to data processing and handling at four levels by different groups of people
  1. Hand recordings by the scouts at the field level.
  2. Typing hand recordings into data sheets at the DPWQCP office.
  3. Photocopying of the typed sheets by DPWQCP personnel.
  4. Data entry or digitization by hired data entry operators.

Data cleansing and standardization is probably the largest part in an ETL exercise. For Agri-DWH major issues of data cleansing had arisen due to data processing and handling at four levels by different groups of people i.e. (i) Hand recordings by the scouts at the field level (ii) typing hand recordings into data sheets at the DPWQCP office (iii) photocopying of the scouting sheets by DPWQCP personnel and finally (iv) data entry or digitization by hired data entry operators.

After achieving acceptable level of data quality, the data was loaded into Teradata data warehouse; subsequently each column was probed using SQL for erroneous entries. Some of the errors found were correct data in wrong columns, nonstandard or invalid variety names etc. There were some intrinsic errors, such as variety type "999" or spray_date "12:00:00 AM" inserted by the system against missing values. Variations found in pesticide names and cotton variety names were removed by comparing them with standard names.

Step 7: Data Transform, Transport & Populate

Among the different types of transformations performed in the implementation, only the more complex i.e. multiple M:1 transformations for field individualization will be discussed in this section.

The media delivered by a Web warehouse include not only data, but text, graphics, image, sound, video, and other forms as well.

39.1 Reasons for web warehousing

SAA

1. Searching the web (web mining).
2. Analyzing web traffic.
3. Archiving the web.

The three reasons for warehousing web data are as listed in the slide. First, web warehousing can be used to mine the huge web content for searching information of interest. Its like searching the golden needle from the haystack. Second reason of Web warehousing is to analyze the huge web traffic. This can be of interest to Web Site owners, for e-commerce, for e-advertisement and so on. Last but not least reason of Web warehousing is to archive the huge web content because of its dynamic nature. As we proceed we will discuss all theses concepts in further detail.

39.2 Web searching

Web is large, actually very large.

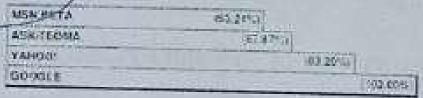
To make it useful must be able to find the page(s) of interest/relevance.

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

The success of google



The Web—an immense and dynamic collection of pages that includes countless hyperlinks and huge volumes of access and usage information—provides a rich and unprecedented data mining source. How can a search identify that portion of the Web that is truly relevant to one user's interests? How can a search find high-quality Web pages on a specified topic?

Application

Currently, users can choose from three major approaches when accessing information stored on the Web:

Long m Park

(i) Keyword-based search or *topic-directory* browsing with search engines such as Google or Yahoo, which use keyword indices ~~or manually built directories~~ to find documents with specified keywords or topics;

(ii) Querying deep Web sources—where information, such as amazon.com's book data and realtor.com's real-estate data, hides behind searchable database query forms—that, unlike the surface Web, cannot be accessed through static URL links; and
(iii)Random surfing that follows Web linkage pointers.

Motivation for Transformation

- Trivial queries give wrong results.
- Static and dynamic attributes
- Static attributes recorded repeatedly

F75

Table 2 gives details of the main attributes recorded at each point. Static attributes are those attributes that are recorded on each visit by the scouts, usually does not changes.

T.Past important Table: P4

Long m Part

Static Attributes		Dynamic Attributes	
1	Farmer Name	1	Date of Visit
2	Farmer Address	2	Pest Population
3	Field Acreage	3	CLCV
4	Variety(ies) Sown	4	Predator Population
5	Sowing date فصل لگانے کی	5	Pesticide Spray Dates
6	Sowing method	6	Pesticide(s) Used

Table-38.1: Cotton pest scouting attributes recorded by DPWQCP surveyors

The data recorded consists of two parts i.e. static and dynamic (Table-38.1). On each visit, the static, as well as the dynamic data is recorded by the scouts, thus resulting in static values getting recorded repeatedly. Since no mechanism is used to uniquely identify each and every farmer, therefore, trivial queries, such as total area scouted, distribution of varieties sown etc. gives wrong results. For example, while aggregating area, the area of the farmer with multiple visits during the season is counted multiple times, giving incorrect results, same is true for varieties sown. Therefore, to do any reasonable analysis after data cleansing, the most important step of data transformation being individualization of the cultivated fields, not farmers. The reason being, a farmer usually has multiple fields, but a field is associated or owned by a single farmer.

Step-7: Resolving the issue

- **Solution:** Individualization of cultivated fields.
  - Technique similar to BSN used to fix names.
  - Unique ID assigned to farmers.
  - BSN used again, and unique ID assigned to fields.
- **Results:**

	Before	After
Area (acers): 2001	23,293	14,187
Area (acers): 2002	26,088	13,693
Farmers	2,696	1,567

- **Limitation:** Field individualization not perfect. Some cases of farmers with same geography, sowing date, same variety and same area. Such cases were dropped.

- Viewing which pages, using which path and how long a view.
- Which visitors spent the most money…
- Thus a lot to discover.

First, you can determine who is visiting your Web site. Minimally, you can determine what company the person is from (the host computer that they are using to surf the Web—ford.com would be the Ford Motor Company, for example). Additionally, if a visitor filled out an online form during a visit to your Web site, you can link the form data with his or her Web site traffic data and identify each visitor by name, address, and phone number (and any other data that your online forms gather).

You can also learn where your visitors are coming from. For example, did they find your site by using a search engine such as Google or did they click on a link at another site? If they did use a search engine, which keywords did they use to locate your site?

Furthermore, you can identify which pages your Web site visitors are viewing, what paths they are taking within your site, and how long they are spending on each page and on the site. You can also determine when they are visiting your site and how often they return.

At the highest level, you can determine which of your Web site visitors spent the most money purchasing your products and services and what the most common paths and referring pages were for these visitors.

As you can see, you can discover a great deal about your Web site visitors—and we only touched upon a few introductory topics.

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.alexa.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

Others

We will not discuss all of them.

The success of these techniques, especially with the more recent page ranking in Google and other search engines shows the Web's great promise to become the ultimate information system.

39.3 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.

Data warehousing concepts are being applied over the Web today. Traditionally, simple search engines have been used to retrieve information from the Web. These serve the basic purpose of data recovery, but have several drawbacks. Most of these engines are based on keyword searches limited to string matching only. That narrows down our retrieval options. Also we have links, at times several levels of them in a particular context. But simple search engines do not do much justice to obtaining information present in these links. They provide direct information recovery, but not enough indirect link information. Also if we have files related to certain subjects and need to couple these, the coupling has to be done manually. Web search engines do not provide mechanisms to incorporate the above features. These and other reasons have led to further research in the area of Web knowledge discovery and have opened the window to the world of Web Warehousing..

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.

The Web with billions of Web pages provides a fertile ground for data mining. However, searching, comprehending, and using the semi structured information stored on the Web poses a significant challenge because this data is more sophisticated and dynamic than the information that commercial database systems store.

To supplement keyword-based indexing, which forms the cornerstone for Web search engines; researchers have applied data mining to Web-page ranking. In this context, data mining helps Web search engines find high-quality Web pages and enhances Web clickstream analysis. For the Web to reach its full potential, however, we must improve its services, make it more comprehensible, and increase its usability. As researchers continue to develop data mining techniques, we believe this technology will play an increasingly important role in meeting the challenges of developing the intelligent Web.

Why web warehousing-Reason no. 2?

Web log contains wealth of information, as it is a key touch point.
Every customer interaction is recorded.

gestures → اشارے

- Web-intensive businesses
- Although most exciting, at the same time it can be the most difficult and most frustrating.
- Not JUST another data source.

پھیلاؤ، اشتہار Short in Past امانت داری

Web-intensive businesses have access to a new kind of data, in some cases literally consisting of the gestures of every Web site visitor. This is called as the *clickstream.* In its most elemental form, the clickstream is every page event recorded by the web server. The clickstream contains a number of new dimensions such as page, session, and referrer-that were previously unknown in conventional DWH environment.

The clickstream is a stream of data, easily being the largest text and number data set we have ever considered for a data warehouse. Although the clickstream is the most exciting new development in data warehousing, at the same time it can be the most difficult and most frustrating to handle and process.

The clickstream is not just another data source that is extracted, cleaned, and dumped into the data warehouse. It is an evolving collection of data sources having more than a dozen Web server log file formats for capturing clickstream data. These formats have optional data components that, if used, can be very helpful in identifying visitors, sessions, and the true meaning of behavior.

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

→ Past

Unlike data from OLTP system, where there were nice user identifications such as unique IDs that were the primary keys, in the context of a web log, this is one of the most issues i.e. identification of the visitor, so is where the visitor actually came from. In OLTP system there was a clean session beginning and session ending, but web is session less. It is very difficult and challenging to identify the session of a visitor, and the list goes on.

Clickstream data contains many ambiguities. Identifying visitor origins, visitor sessions, and visitor identities is something of an interpretive art. Browser caches and proxy servers make these identifications even more challenging.

Identifying the Visitor Origin

- There is no easy way to determine from a log whether or not (your) site is set as a browser's home page of the visitor.
- The site may be reached as a result of a click through----- a *deliberate click on a text or graphic link from another site.*

Web log file formats

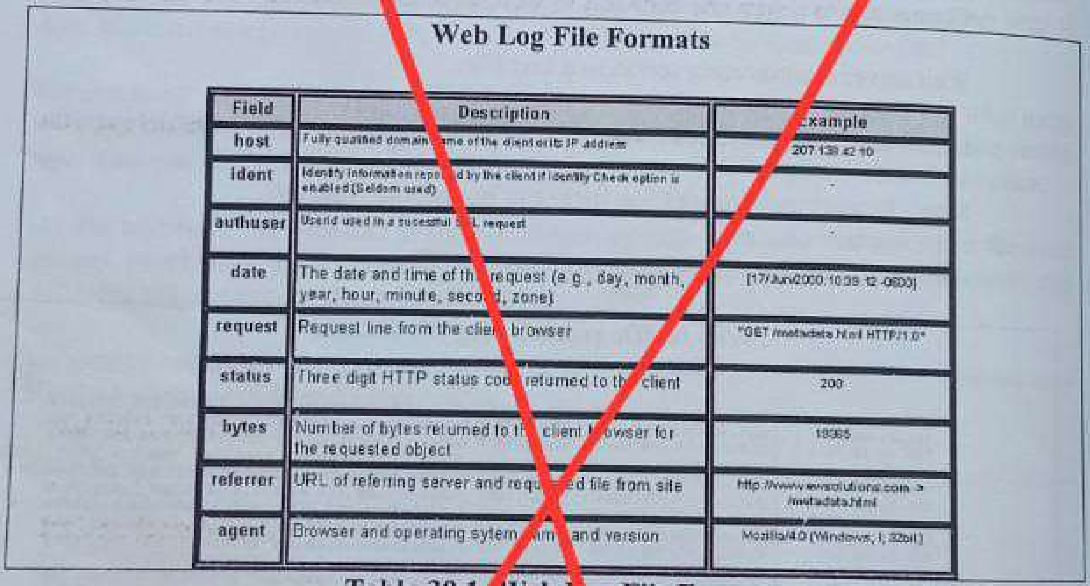
WAC

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

- Web server
- Application
- Configuration options

Several servers support ~~CLF ECLF~~ format.

Web log file formats vary depending on the Web server application and configuration options selected during installation. Most Web server applications (including those from Apache, Microsoft and Netscape) support Common Log file Format (CLF, sometimes pronounced "clog") or Extended Common Log file Format (ECLF). CLF and ECLF formats share the same initial seven fields, but ECLF adds referrer and agent elements.



Web Log File Formats

Field	Description	Example
host	Fully qualified domain name of the client or its IP address	207.138.42.10
ident	Identify information reported by the client if Identify Check option is enabled(Seldom used)	-
authuser	Userid used in a sucessful SSL request	-
date	The date and time of the request (e.g., day, month, year, hour, minute, second, zone)	[17/Jun2000:10:39:12 -0600]
request	Request line from the client browser	"GET /metadata.html HTTP/1.0"
status	Three digit HTTP status code returned to the client	200
bytes	Number of bytes returned to the client browser for the requested object	18365
referrer	URL of referring server and requested file from site	http://www.evsolutions.com > /metadata.html
agent	Browser and operating sytem name and version	Mozilla/4.0 (Windows; I; 32bit)

Table-39.1: Web Log File Formats

Our example proxy log data file contained following fields

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)

1- Using Time-contiguous Log Entries

Long imp Part

- A session can be consolidated by collecting time-contiguous log entries from the same host (Internet Protocol, or IP, address).
- Limitations
- (i) The method breaks down for visitors from large ISPs
- (ii) Different IP addresses
- (iii) Browsers that are behind some firewalls.

In many cases, the individual hits comprising a session can be consolidated by collating time-contiguous log entries from the same host (Internet Protocol, or IP, address). If the log contains a number of entries with the same host ID in a short period of time (for example, one hour), one can reasonably assume that the entries are for the same session.

Limitations

- This method breaks down for visitors from large ISPs because different visitors may reuse dynamically assigned IP addresses over a brief time period.
- Different IP addresses may be used within the same session for the same visitor.
- This approach also presents problems when dealing with browsers that are behind some firewalls.

Notwithstanding these problems, many commercial log analysis products use this method of session tracking, which requires no cookies or special Web server features.

2- Using Transient Cookies

Long Imp

- Let the Web browser place a session-level cookie into the visitor's Web browser.
- Cookie value can serve as a temporary session ID
- **Limitations**
  - You can't tell when the visitor returns to the site at a later time in a new session.

Another, much more satisfactory method is to let the Web browser place a session-level cookie into the visitor's Web browser. This cookie will last as long as the browser is open and, in general, won't be available in subsequent browser sessions. The cookie value can serve as a temporary session ID not only to the browser but also to any application that requests the session cookie from the browser. This request must come from the same Web server (actually, the same domain) that placed the cookie in the first place. Using a transient cookie value as a temporary session ID for both the clickstream and application logging allows a straightforward approach to associating the data from both these sources during post session log processing.

Limitation

Using a transient cookie has the disadvantage that you can't tell when the visitor returns to the site at a later time in a new session with a new transient cookie.

Let's start with the origin of the visitor.

There is no easy way to determine from a log whether or not your site is set as a browser's home page. This is pretty unlikely unless one is the Webmaster for a portal site or an intranet home page, but many sites have buttons that, when clicked, prompt the visitor to set his or her URL as the browser's home page. Unfortunately, a visitor may be directed to the site from a search at a portal such as Yahoo! or Alta Vista. Such referrals can come either from the portal's index or table of contents, for which placement fee might have been paid, or from a keyword or content search.

The site may be reached as a result of a *click through*---a deliberate click on a text or graphic link from another site. This may be a paid-for referral as via a banner ad or a free referral from an individual or cooperating site. In the case of click-throughs, the referring site almost always will be identifiable in the Web site's referrer log data. Capturing this crucial clickstream data is important to verify the efficacy of marketing programs. It also provides crucial data for auditing invoices one may receive from click-through advertising charges

V.V.V. Important Past Paper

Identifying the Session

Past

- 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, stateless 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

TCL
TC
SSL

Transient cookies
Persistent cookies
Session ID Ping-Pong

~~Most~~ web-centric data warehouse applications will require every visitor session (visit) to have its own unique identity tag similar to a grocery store point-of-sale ticket ID or *session ID*. The rows of every individual visitor action in a session, whether derived from the clickstream or from an application interaction, must contain this tag. However, it must be kept in mind that the operational application generates this session ID, not the Web server.

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

5- Using Persistent Cookies

مستقل

- Establish a persistent cookie in the visitor's PC
- **Limitations**
  - No absolute guarantee that even a persistent cookie will survive.
- Certain groups of Web sites can agree to store a common ID tag

The Web site may establish a persistent cookie in the visitor's PC that is not deleted by the browser when the session ends.

Limitations

- It's possible that the visitor will have his or her browser set to refuse cookies or may clean out his or her cookie file manually, so there is no absolute guarantee that even a persistent cookie will survive.
- Although any given cookie can be read only by the Web site that caused it to be created, certain groups of Web sites can agree to store a common ID tag that would let these sites combine their separate notions of a visitor session into a super session

Identifying the Visitor

- Identifying a specific visitor who logs onto our site presents some of the most challenging problems
- Web visitors wish to be anonymous.
- If we request a visitor's identity, he or she is likely to lie about it.
- We can't be sure which family member is visiting our site.
- We can't assume that an individual is always at the same computer.

Identifying a specific visitor who logs onto our site presents some of the most challenging problems facing a site designer, Webmaster, or manager of data warehousing for the following reasons:

Web visitors wish to be anonymous. Unlike a bank ATM machine with dedicated access, a user accessing a web site via Internet may have no reason to trust, the Internet, or their PC with personal identification or credit card information.

If we request a visitor's identity, he or she is likely to lie about it. It is believed that when asked their name on an Internet form, men will enter a pseudonym 50 percent of the time and women will use a pseudonym 80 percent of the time.

We can't be sure which family member is visiting our site. If we obtain an identity by association, for instance, from a persistent cookie left during a previous visit, the

Long Question

3- Using HTTP's secure sockets layer (SSL)

Long im Part

- Offers an opportunity to track a visitor session
- **Limitations**
  - To track the session, the entire information exchange needs to be in high overhead SSL
  - Each host server must have its own unique security certificate.
- Visitors are put-off by pop-up certificate boxes.

This offers an opportunity to track a visitor session because it may include a login action by the visitor and the exchange of encryption keys.

Limitations

- The downside to using this method is that to track the session, the entire information exchange needs to be in high overhead SSL, and the visitor may be put off by security advisories that can pop up when certain browsers are used.
- Each host server must have its own unique security certificate.

4- Using session ID Ping-pong

- Maintain visitor state by placing a session ID in a hidden field of each page returned to the visitor.
- Session ID can be returned to the Web server as a query string appended to a subsequent URL.
- **Limitations**
  - Requires a great deal of control over the Web site's page-generation methods
- Approach breaks down if multiple vendors are supplying content in a single session

If page generation is dynamic, you can try to maintain visitor state by placing a session ID in a hidden field of each page returned to the visitor. This session ID can be returned to the Web server as a query string appended to a subsequent URL.

Limitation

- This method of session tracking requires a great deal of control over the Web site's page-generation methods to ensure that the thread of session ID is not broken. If the visitor clicks on links that don't support this method a single session will appear to be multiple sessions.
- This approach also breaks down if multiple vendors are supplying content in a single session.

Proxey Servers: Problems Illustration

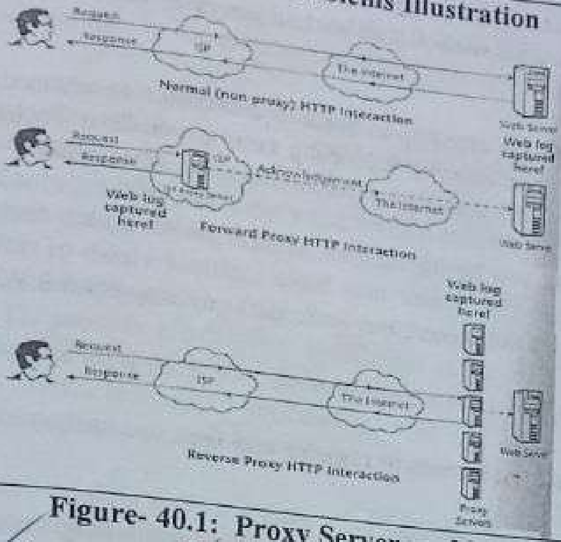


Figure- 40.1: Proxy Server problems

(Short)

Forward Proxy ✓ V.V.V.V. important Past

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.

(Short)

Reverse Proxy ✓ V.V.V.V. Important Past

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. رکاوٹ

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

Browser caches also introduce uncertainties in our attempts to track all the events that occur during a visitor session. Most browsers store a copy of recently retrieved objects such as HTML pages and images in a local object cache in the PC's file system. If the visitor returns to a page already in his or her local browser cache (for example, by identification is only for the computer, not for the specific visitor. Any family member or company employee may have been using that particular computer at that moment in time

We can't assume that an individual is always at the same computer. Server provided cookies identify a computer, not an individual. If someone accesses the same Web site from an office computer, a home PC, and a laptop computer, a different Web site cookie is probably put into each machine.

40.3 Proxy servers → past Past

- An HTTP request is not always served from the server specified in a URL.
- Many ISPs make use of proxy servers to reduce Internet traffic.
- Proxy servers can introduce three problems:
    - May deliver outdated content.
    - May satisfy a content request without properly notifying the originating server that the request has been served by the proxy.
- Web site will not know who made the page request unless a cookie is present.

When a browser makes an HTTP request, that request is not always served from the server specified in a URL. Many ISPs make use of proxy servers to reduce Internet traffic. Proxy servers are used to cache frequently requested content at a location between its intended source and an end visitor. An HTTP request may not even leave the visitor's Pc. It may be satisfied from the browser's local cache of recently accessed objects

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 though a proxy, the Web site will not know who made the page request unless a cookie is present.

DTS is to collect data from different Sources and then Extract, Transform and consolidate data into single or multiple Destinations.

Lab Lecture-1: Data Transfer Service (DTS)

Slide 1

Virtual University of Pakistan

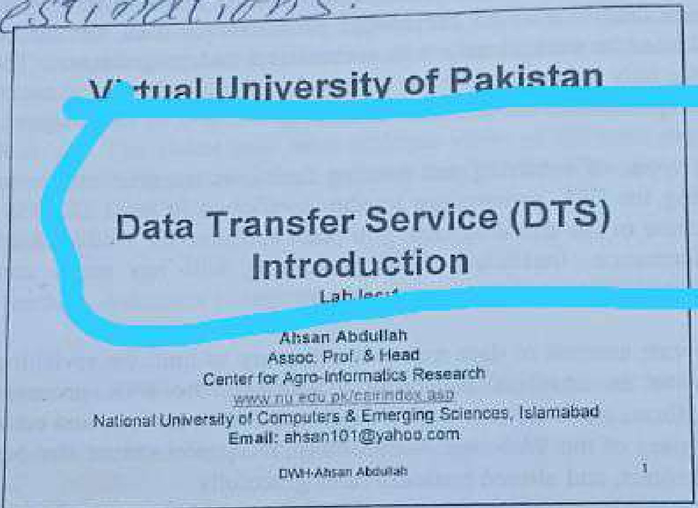
Data Transfer Service (DTS)

Introduction

DTS overview: The function of DTS is to collect data from different Sources and then Extract, Transform and consolidate data into Single or multiple Destinations.

Lab Lecture-1: Data Transfer Service (DTS)

Slide 1



Virtual University of Pakistan

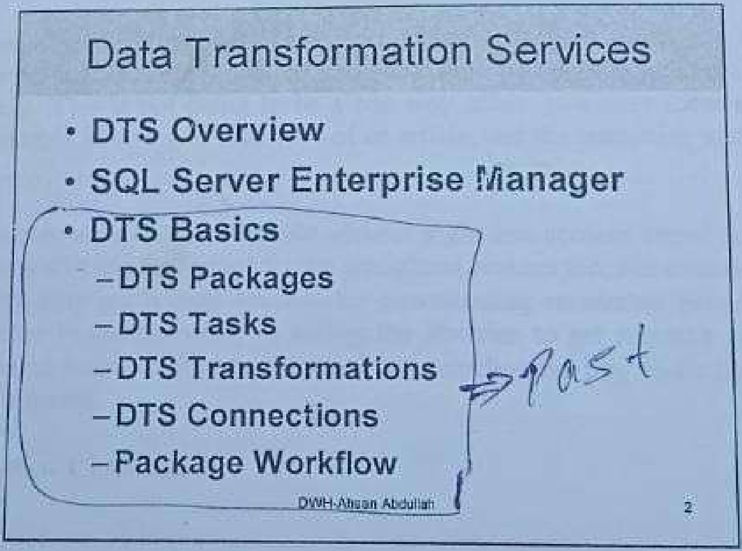
Data Transfer Service (DTS) Introduction

Lab lect

Ahsan Abdullah
Assoc. Prof. & Head
Center for Agro-Informatics Research
www.nu.edu.pk/cairindex.asp
National University of Computers & Emerging Sciences, Islamabad
Email: ahsan101@yahoo.com

DWH-Ahsan Abdullah 1

Slide 2



Data Transformation Services

- DTS Overview
- SQL Server Enterprise Manager
- DTS Basics
  - DTS Packages
  - DTS Tasks
  - DTS Transformations ⇒ past
  - DTS Connections
  - Package Workflow

DWH-Ahsan Abdullah 2

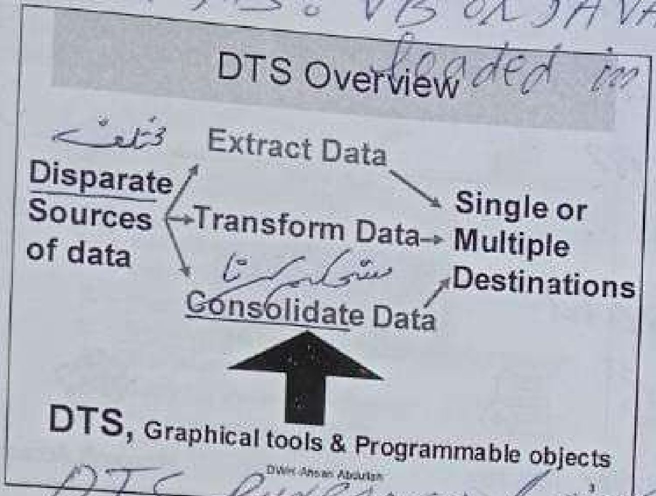
Microsoft® SQL Server™ 2000 Data Transformation Services (DTS) is a set of graphical tools and programmable objects that allow you extract, transform, and consolidate data from disparate sources into single or multiple destinations. SQL Server Enterprise Manager provides an easy access to the tools of DTS.

The purpose of this lecture is to get an understanding of DTS basics, which is necessary to learn the use of DTS tools. These DTS basics describe the capabilities of DTS and summarize the business problems it addresses.

Past short. Question: which script language is used to perform complex transformation in DTS Package

Ans: VB or JAVA script that is loaded in DTS Package.

Slide 3



DTS Overview

مختلف

Extract Data

Disparate Sources of data

Transform Data

Single or Multiple Destinations

مستحکم کرنا

Consolidate Data

DTS, Graphical tools & Programmable objects

DWH-Ahsan Abdullah

Past

DTS Purposes Comg also short:

Many organizations need to centralize data to improve corporate decision-making. However, their data may be stored in a variety of formats and in different locations. Data Transformation Services (DTS) address this vital business need by providing a set of tools that let you extract, transform, and consolidate data from disparate sources into single or multiple destinations supported by DTS connectivity.

DTS allows us to connect through any data source or destination that is supported by OLE DB. This wide range of connectivity that is provided by DTS allows us to extract data from wide range of legacy systems. Heterogeneous source systems store data with their local formats and conventions. While consolidating data from variety of sources we need to transform names, addresses, dates etc into a standard format. For example consider a student record management system of a university having four campuses. A campus say 'A' follows convention to store city codes "LHR" for Lahore. An other campus say 'B' stores names of cities "Lahore", campus 'C' stores city names in block letters 'LAHORE', and the last campus 'D' store city names as 'lahore'. When the data from all the four campuses is combined as it is and query is run "How many students belong to 'Lahore'?" We get the answer only from campus B because no other convention for Lahore matches to the one in query.

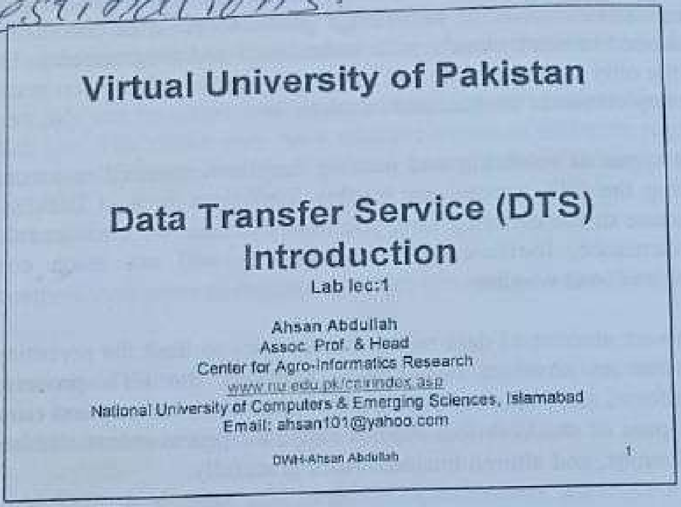
To combine data from heterogeneous sources with the purpose of some useful analysis requires transformation of data. Transformation brings data in some standard format.

Microsoft SQL Server provides graphical tools to build DTS packages. These tools provide good support for transformations. Complex transformations are achieved through VB Script or Java Script that is loaded in DTS package. Package can also be programmed by using DTS object model instead of using graphical tools but DTS programming is rather complicated.

DTS overview: The function of DTS is to collect data from different sources and then Extract, Transform and consolidate data into single or multiple Destinations.

Lab Lecture-1: Data Transfer Service (DTS)

Slide 1



Virtual University of Pakistan

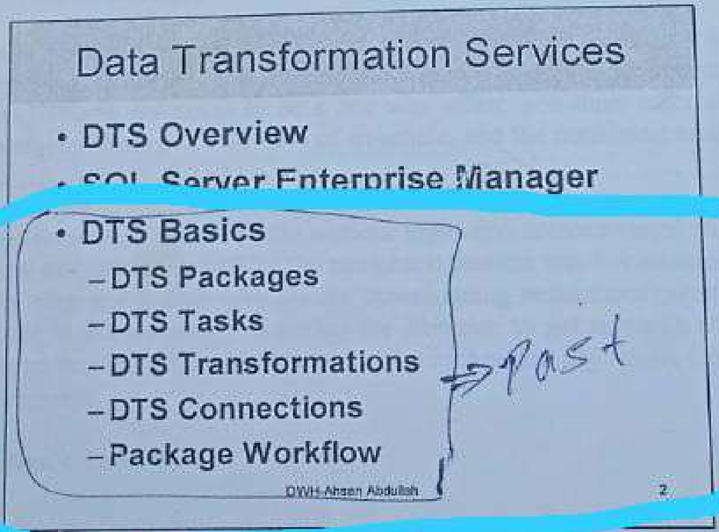
Data Transfer Service (DTS) Introduction

Lab lec:1

Ahsan Abdullah
Assoc. Prof. & Head
Center for Agro-Informatics Research
www.nu.edu.pk/cairindex.asp
National University of Computers & Emerging Sciences, Islamabad
Email: ahsan101@yahoo.com

DWH-Ahsan Abdullah 1

Slide 2



Data Transformation Services

- DTS Overview
- SQL Server Enterprise Manager
- DTS Basics
  - DTS Packages
  - DTS Tasks
  - DTS Transformations
  - DTS Connections
  - Package Workflow

DWH-Ahsan Abdullah 2

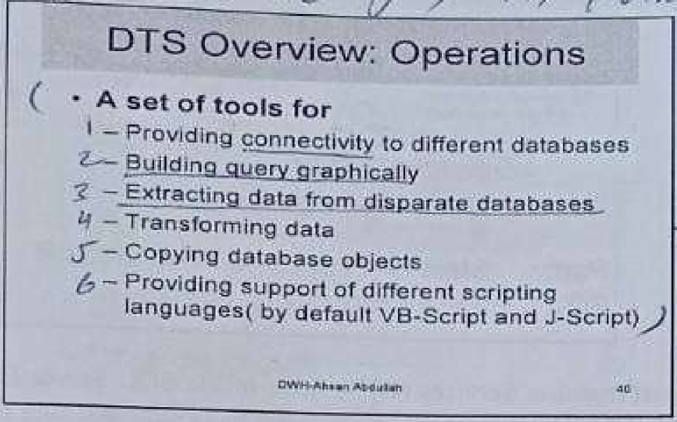
→ Past

Microsoft® SQL Server™ 2000 Data Transformation Services (DTS) is a set of graphical tools and programmable objects that allow you extract, transform, and consolidate data from disparate sources into single or multiple destinations. SQL Server Enterprise Manager provides an easy access to the tools of DTS.

The purpose of this lecture is to get an understanding of DTS basics, which is necessary to learn the use of DTS tools. These DTS basics describe the capabilities of DTS and summarize the business problems it addresses.

all the three locations. So while consolidating data names are transformed to standard spellings of names. Similarly Date formats are different in both source systems and it is standardized in destination system (middle one).

Slide 6 (Short + long) in Past Papers



DTS Overview: Operations

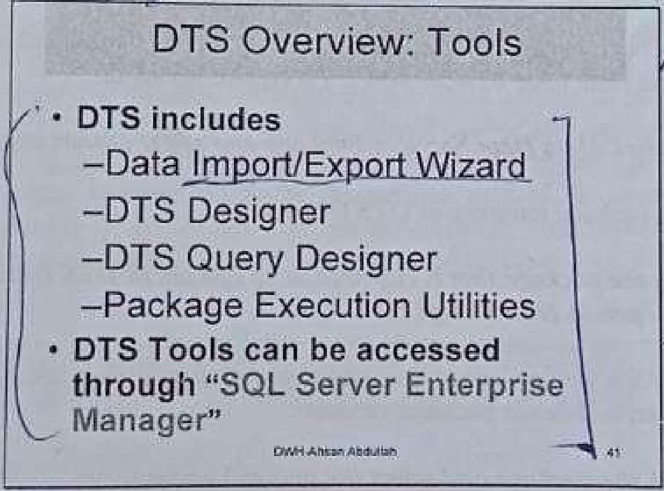
- A set of tools for
  - 1 – Providing connectivity to different databases
  - 2 – Building query graphically
  - 3 – Extracting data from disparate databases
  - 4 – Transforming data
  - 5 – Copying database objects
  - 6 – Providing support of different scripting languages(by default VB-Script and J-Script)

DWH-Ahsan Abdullah 40

Imp

DTS contains a set of tools that provides a very easy approach to build a package and execute it. Writing or building a package through programming is a complex task but DTS tools like DTS Designer and Import/Export Wizard do this entire complex task for user just through a single click of button. Not only package building but query building has also very sophisticated support in DTS tools.

Slide 7 DTS Tools Provided For programmer:



DTS Overview: Tools

- DTS includes
  - –Data Import/Export Wizard
  - –DTS Designer
  - –DTS Query Designer
  - –Package Execution Utilities
- DTS Tools can be accessed through "SQL Server Enterprise Manager"

DWH-Ahsan Abdullah 41

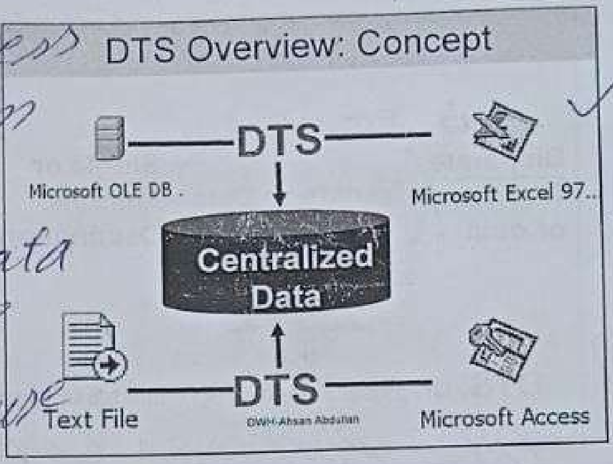
Imp

Package execution utilities are used to run or execute a package, no matter package is designed through the tools provided by DTS or any external tool like Visual Basic. All these tools can be accessed through the SQL Server Enterprise Manager.

Open the node Data Transfer Services in SQL Server Enterprise Manager. Choose the option in which any finished package is saved. Right click the package and get option to execute it.

DTS gets data from different Sources like microsoft OLE DB and text file and microsoft Excel and microsoft access and Transform data into a centralized data which becomes a datawarehouse

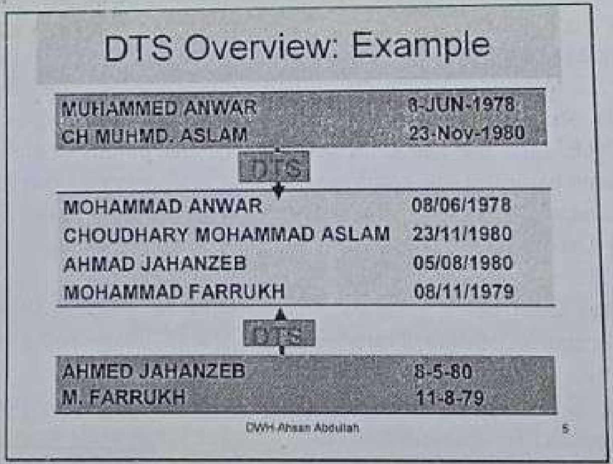
Slide 4


DTS Overview: Concept

✓ Imp

The slide shows the heterogeneous sources of data. Position of DTS while consolidating the data into a single source is also clear from the slide. In legacy systems we may come across the text files as a source of data. Microsoft Access is a database management system, maintains data in tables, and columns validate the input to the system. We often find legal values are stored in these sort of data management systems. But when we deal with text files no validation mechanism for input is there. Therefore we may come across illegal and rubbish values in text files. This makes the process of transformation further complicated.

Slide 5


DTS Overview: Example

MUHAMMED ANWAR	8-JUN-1978
CH MUHMD. ASLAM	23-Nov-1980

DTS

MOHAMMAD ANWAR	08/06/1978
CHOUDHARY MOHAMMAD ASLAM	23/11/1980
AHMAD JAHANZEB	05/08/1980
MOHAMMAD FARRUKH	08/11/1979

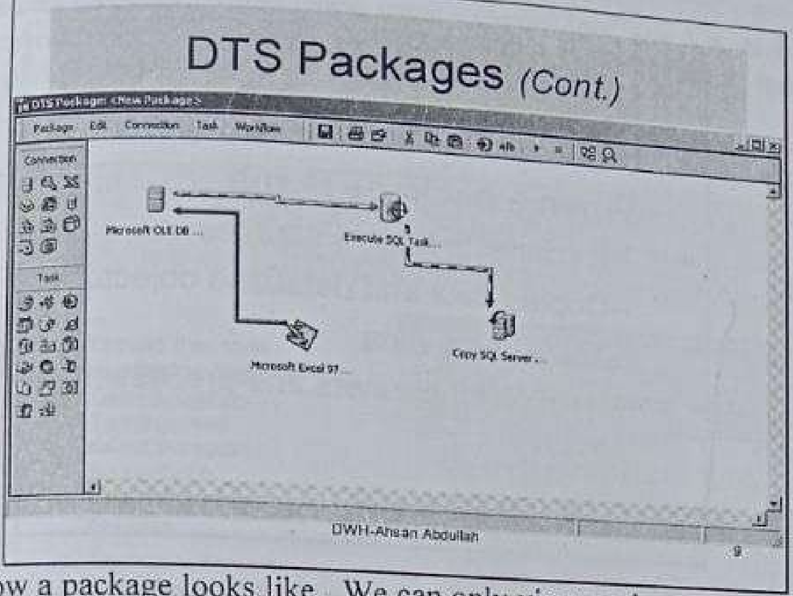
DTS

AHMED JAHANZEB	8-5-80
M. FARRUKH	11-8-79

DWH-Ahsan Abdullah 5

In the this slide we may see three data management systems. Data is extracted from two systems, top and bottom, and is loaded into the standardized system shown in the middle. We may see two transformations over here. First one is name transformation and the other one is date transformation. In the database management system shown at the top we have two names Muhammed Anwer and Choudhary Mohammed Aslam. Whereas in the system shown at the bottom we have two different names Ahmed Jahanzeb and Muhammed Farrukh. Out of four names three names contain Muhammed but with different spellings. Computer can not identify that the word 'Muhammed' is intended at

Slide 11



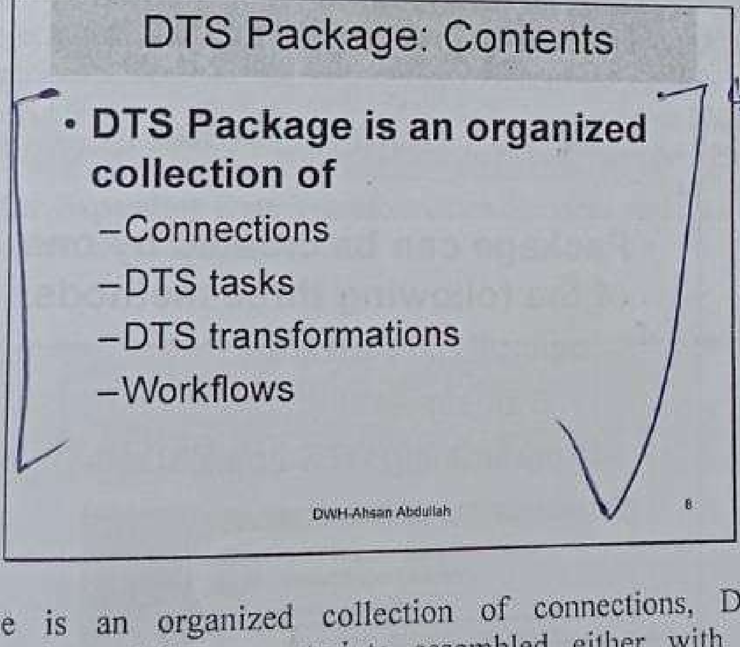
DTS Packages *(Cont.)*

DWH-Ahsan Abdullah

9

Slide shows how a package looks like. We can only view package as a form of graphical objects as shown in the slide. Here two connections are established. "Microsoft OLEDB Driver" and "Microsoft Excel 97" are connections. Black link between two connections is transformation task. "Execute SQL" and "Copy SQL Server" both are tasks. Green and blue links are workflows. Green link shows ***'On the Success of'*** *i.e.* on the success of Connection establishment execute task execute SQL. Blue link shows ***'On the Failure of'*** on the failure of the previous task execute another task Copy SQL Server objects.

Slide 12



DTS Package: Contents

- **DTS Package is an organized collection of**
  - –Connections
  - –DTS tasks
  - –DTS transformations
  - –Workflows

DWH-Ahsan Abdullah

8

Past

Imp

A DTS package is an organized collection of connections, DTS tasks, DTS transformations, and workflow constraints assembled either with a DTS tool or programmatically and saved to Microsoft® SQL Server™, SQL Server 2000 Meta Data Services, a structured storage file, or a Microsoft Visual Basic® file.

Each package contains one or more steps that are executed sequentially or in parallel when the package is run. When executed, the package connects to the correct data sources, copies data and database objects, transforms data, and notifies other users or processes of events.

Before learning to use DTS some basic concepts like DTS packages, DTS tasks, transformations and workflows are important to understand.

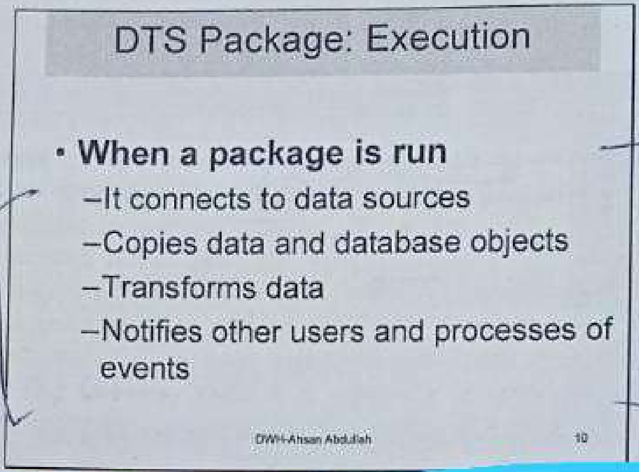
When we want to use computers to perform some particular task through programming, what we do? We write a program in some programming language. Program is a sequence of logical statements that collectively achieve the purpose of the programmer. This analogy is useful in understanding the concept of package and tasks in DTS. DTS package is exactly like a computer program. Like a computer program DTS package is also prepared to achieve some goal. Computer program contains set of instructions whereas DTS package contains set of tasks. Tasks are logically related to each other. When a computer program is run, some instructions are executed in sequence and some in parallel. Likewise when a DTS package is run some tasks are performed in sequence and some in parallel. The intended goal of a computer program is achieved when all instructions are successfully executed. Similarly the intended goal of a package is achieved when all tasks are successfully accomplished.

DTS task is a unit of work in a package. Tasks can be establishment of connection to source and destination databases, extraction of data from the source, transformation of data, loading of data to the destination, generation of error messages and emails etc.

In real world systems when we talk about heterogeneous sources of data there arise a lot of complicated issues. Heterogeneous systems contain data with different storage conventions, different storage formats, different technologies, and different designs etc. Power of DTS lies in extracting the data from these heterogeneous sources, transforming to some standard format and convention, and finally load data to some different system with totally different parameters like technology, design etc. Microsoft SQL Server provides user-friendly tools to develop DTS Packages. Through graphical editor/ designer or wizards we can put together set of tasks in a package. Order or sequence in which the tasks are required to be performed can be set through conditions like "On success of task A task B should be performed otherwise task C should be performed." This order or sequence of execution is called Workflow of a package.

In this lecture we will see these concepts in detail and in subsequent lectures we will develop packages and practically get into the use of DTS functionalities.

Slide 13



DTS Package: Execution

- **When a package is run**
  - –It connects to data sources
  - –Copies data and database objects
  - –Transforms data
  - –Notifies other users and processes of events

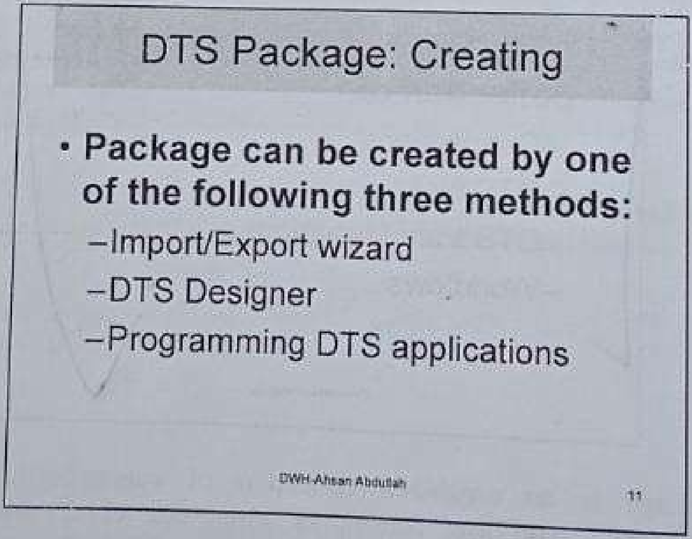
DWH-Ahsan Abdullah 10

When we run a Data Transformation Services (DTS) package, all of its connections, tasks, transformations, and scripting code are executed in the sequence described by the package workflow.

We can execute a package from:

- Within a DTS tool.
- SQL Server Enterprise Manager.
- Package execution utilities.

Slidel 4



DTS Package: Creating

- **Package can be created by one of the following three methods:**
  - –Import/Export wizard
  - –DTS Designer
  - –Programming DTS applications

DWH-Ahsan Abdullah 11

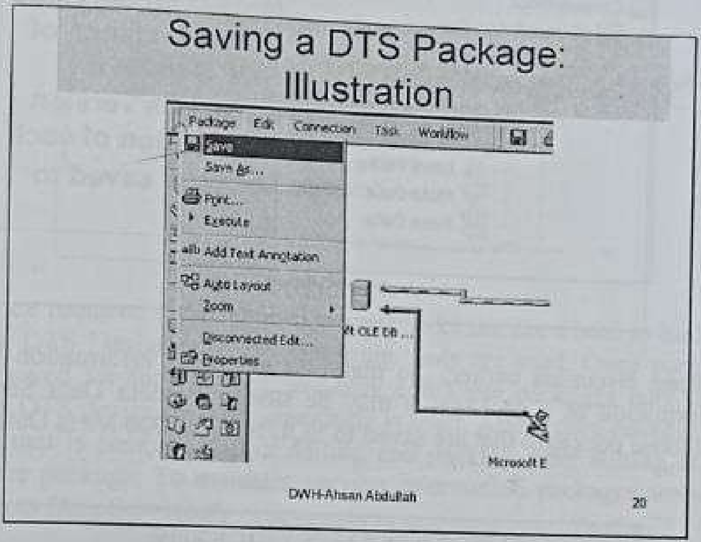
Microsoft SQL Server provides a good support for the tools that are helpful in building a package. Import/Export Wizard and DTS Designer both are the graphical methods of building a package. Both tools provide support to run the package also. Building a package means putting all the tasks that are supposed to be performed in a particular package together and setting their order of execution or defining workflow. Whereas when we actually run a package all the tasks are actually performed.

Server Meta Data services:

(short) → in past papers

(Server Meta Data Services. The advantage which we get when we store our package to SQL Server 2000 Meta Data Services is that we may maintain meta data information of the databases involved in the packages and we may keep version information of each package. Furthermore package can be stored in a structured file and Microsoft visual basic file.

Slide 22



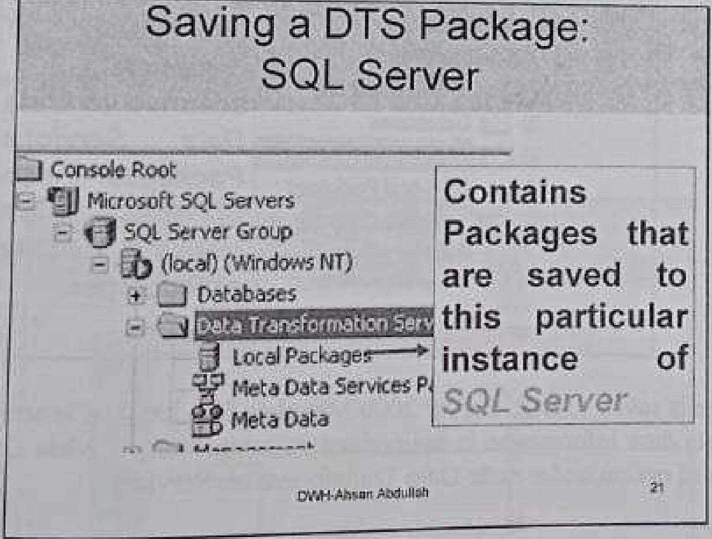
Saving a DTS Package: Illustration

DWH-Ahsan Abdullah

20

This slide illustrate the package saving process.

Slide 23



Saving a DTS Package: SQL Server

Contains Packages that are saved to this particular instance of SQL Server

DWH-Ahsan Abdullah

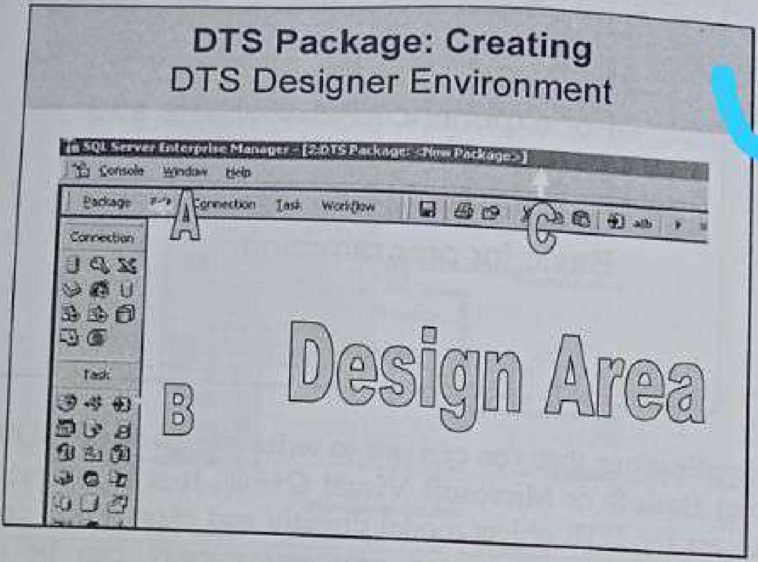
21

Data Transformation Services node of SQL Server Enterprise Manager contains three options to locate the package saved earlier. The first option is local Packages. These are the packages that are saved to this particular instance of SQL Server. Microsoft SQL Server may have multiple instances on each machine or over a Network. Local packages are those that are saved to this particular instance of SQL Server.

a. After Expanding Data Transformation Services node select Action > New Package

b. After Expanding Data Transformation Services node select ⁂ on toolba
to access DTS Designer

Slide 19



DTS Package: Creating
DTS Designer Environment

The slide shows environment of DTS Designer. In designer we can see four windows

A. Connection toolbar
B. Task toolbar
C. General toolbar
D. Design Area

Imp

A. **Connection toolbar**

Connection toolbar shows all available connections in the form of icons or symbols. All OLE DB supported connections are available. To establish a new connection just click the correct icon and drag to design area. Then set properties to your connection. In case of any difficulty in identifying the connection icon, click on Connection on Menu bar just above the connection toolbar.

B. **Task Toolbar**

Tasks toolbar shows icons for all tasks that are supported by DTS. For example ⇒ is used to set transformation task. This also works as drag and drop. DTS Designer is very friendly to use as it guides user about what to do after picking a certain option. For new users who do not recognize the tasks through icons, in the top menu bar 'Task' is available.

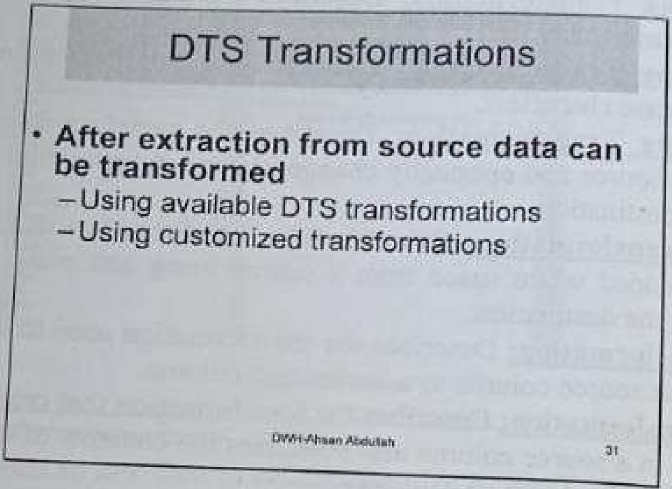
C. **General Toolbar**

This toolbar provides general functionality like saving a package, executing a package. ▶ is used to execute a package

D. **Design Area**

g a set of Transact-SQL statements or Microsoft ActiveX® scripts against a data source. The Execute SQL and ActiveX Script tasks allow you to write your own SQL statements and scripting code and execute them as a step in a package workflow.

Slide 33



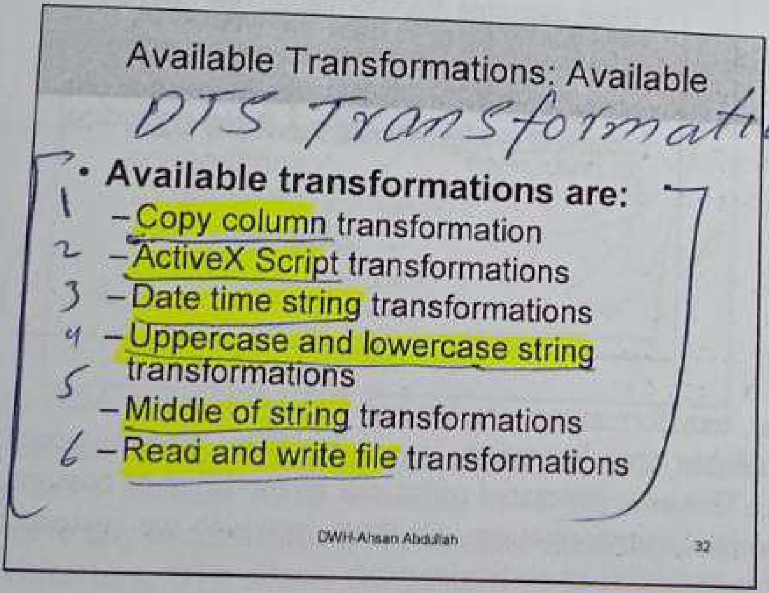
DTS Transformations

- After extraction from source data can be transformed
  - Using available DTS transformations
  - Using customized transformations

DWH-Ahsan Abdullah 31

While transferring data from source to destination that may be a single source of truth, data may require to be transformed. Power of DTS tools lies in the support of data transformations. Some transformations are already available with DTS tools and customized transformations can be performed through VB Script or Java Script.

Slide 34



Available Transformations: Available

DTS Transformations

- Available transformations are:
  1 – Copy column transformation
  2 – ActiveX Script transformations
  3 – Date time string transformations
  4 – Uppercase and lowercase string transformations
  5 – Middle of string transformations
  6 – Read and write file transformations

DWH-Ahsan Abdullah 32

The slide shows the list of transformations that are already available with DTS tools i.e. DTS Designer and DTS import/export wizard. Wizard has a support of two transformations out of six shown over here:

- Copy column transformation
- Active-X script transformation

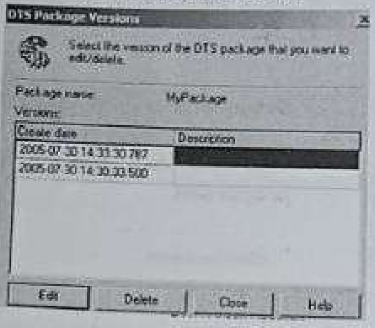
The rest four are accessed through DTS designer and scripts.

Copy Column Transformation: Describes the transformation used to copy source data to the destination.

Slide 30

DTS Package Operations: Versioning

- Right click any saved package to view its version information



DTS Package Versions

Select the version of the DTS package that you want to edit/delete.

Package name: MyPackage

Versions:

Create date	Description
2005-07-30 14:33:30.787	
2005-07-30 14:30:30.500	

28

If we want to get version information of a package we can see it by right clicking the package and selecting version information. First column contains creation date and the other column contains the description about changes if it is saved with the [illegible]

Slide 31

DTS Tasks

- DTS Package contains one or more tasks
- Task defines single work item
  - Establishing connections
  - Importing and exporting data
  - Transforming data
  - Copying database objects
  - etc

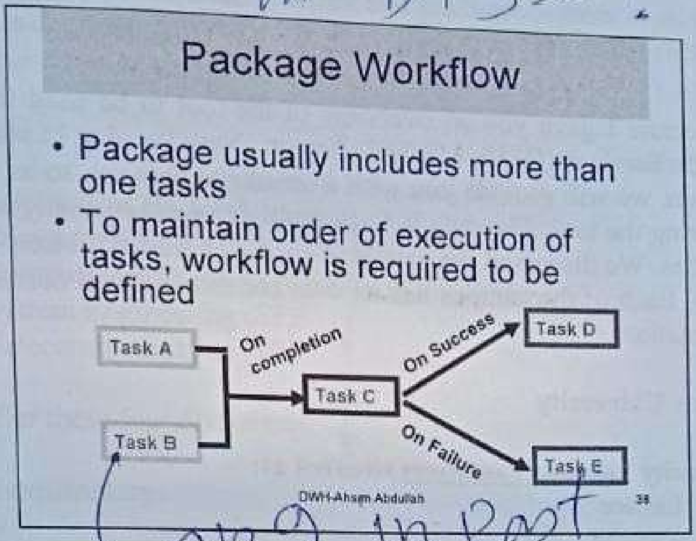
→ Past

DWH-Ahsan Abdullah 29

DTS Packages contain a sequence of tasks. When a package is executed these tasks are performed in sequence or in parallel. These tasks are the single work item in a package. Tasks can be establishing connections, extraction of data from sources, transformations applied on data, loading data to destination, generation of automated email messages to administrator in case of some problem during the package execution..

Past Paper Question:
Briefly explain types of constraints that we use in DTS...!

Slide 40



Package Workflow

- Package usually includes more than one tasks
- To maintain order of execution of tasks, workflow is required to be defined

(MCQ in Past

Ans:

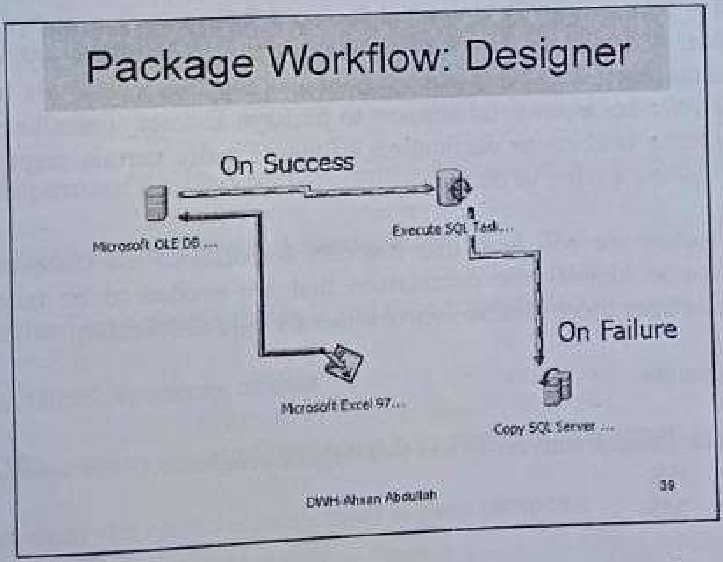
Precedence constraints sequentially link tasks in a package. In DTS, you can use three types of precedence constraints, which can be accessed either through DTS Designer or programmatically:

Unconditional: If you want Task 2 to wait until Task 1 completes, regardless of the outcome, link Task 1 to Task 2 with an *unconditional precedence* constraint.

On Success: If you want Task 2 to wait until Task 1 has successfully completed, link Task 1 to Task 2 with an *On Success precedence* constraint.

On Failure: If you want Task 2 to begin execution only if Task 1 fails to execute successfully, link Task 1 to Task 2 with an *On Failure precedence* constraint. If you want to run an alternative branch of the workflow when an error is encountered, use this constraint.

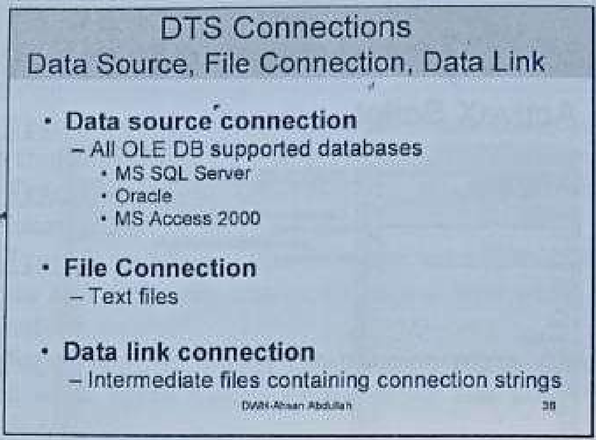
Slide 41



Package Workflow: Designer

This slide shows the process of making workflows using the designer. It provides a graphical interface making the workflow management very easy

Slide 38



DTS Connections

Data Source, File Connection, Data Link

- **Data source connection**
  - All OLE DB supported databases
    - MS SQL Server
    - Oracle
    - MS Access 2000
- **File Connection**
  - Text files
- **Data link connection**
  - Intermediate files containing connection strings

DWH-Ahsan Abdullah 38

(past)

DTS allows the following varieties of connections:

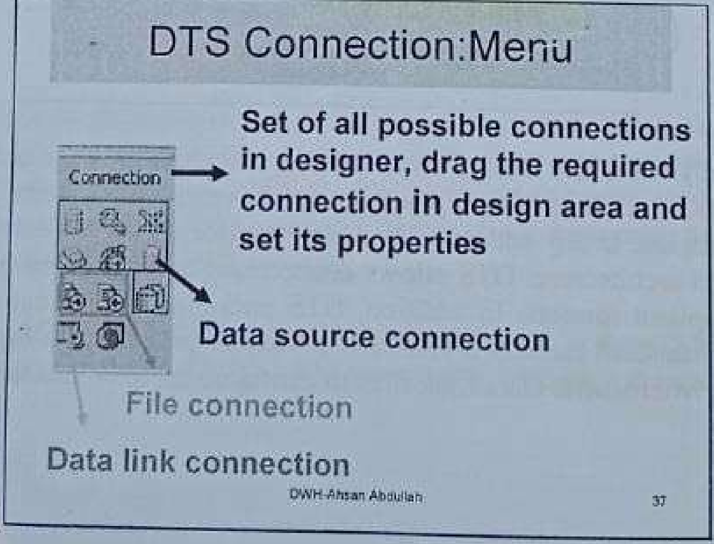
A data source connection. These are connections to: standard databases such as Microsoft SQL Server™ 2000, Microsoft Access 2000, Oracle, dBase, Paradox; OLE DB connections to ODBC data sources; Microsoft Excel 2000 spreadsheet data; HTML sources; and other OLE DB providers.

A file connection. DTS provides additional support for text files. When specifying a text file connection, you specify the format of the file. For example:

- Whether a text file is in delimited or fixed field format.
  Whether the text file is in a Unicode or an ANSI format.
- The row delimiter and column delimiter if the text file is in fixed field format.
- The text qualifier.
  Whether the first row contains column names.

A data link connection. These are connections in which an intermediate file outside of SQL Server stores the connection string.

Slide 39



DTS Connection:Menu

Set of all possible connections in designer, drag the required connection in design area and set its properties

Data source connection

File connection

Data link connection

DWH-Ahsan Abdullah 37

Example: Student Record System: Issues

- Figure out the issues related with each source system
- Issues include
  - Standards and formats of stored data
  - Number of tables in different source systems
  - Type of columns, their number and ordering in different tables to be combined

Here we need to figure out the issues in source systems. As source data is distributed over different campuses therefore the issues like difference in date formats, conventions of storing gender (M/F,0/1,1/0), etc are obvious. Microsoft SQL Server has a good support to resolve these issues.

Extracting University Data

1. Lets start our practical by loading data for the university
2. We have data from four different campuses
3. Initially we will develop four different databases, one for each campus, and load corresponding data
4. Then we will transform and standardized each database
5. Finally we will combine all the four databases to single source of truth (DWH)
6. At each step we will run queries to collect demographics

For loading data for the university, it is required to load the data for four campuses, separately and as it is, into the MS-SQL Server. Once all data is loaded to SQL Server then the tasks of transformation and standardization would be started. First we will transform the database of each of the campuses individually. Then we will standardize the databases of four campuses separately. Finally, the data from four different campuses will be put together. (Past) ⇒ What is task performed by import and export data wizard?

Extracting Data Using Wizards

Ans:

- *Import and Export Data Wizard* provides the easiest method of loading data.
- The wizard creates package which is a collection of tasks
- Tasks can be as follows:
  - 1 Establish connection through source / destination systems
  - 2 Creates similar table in SQL Server
  - 3 Extracts data from text files
  - 4 Apply very limited basic transformations if required
  - 5 Loads data into SQL Server table

After addressing the issues we decide to select a suitable tool in SQL Server to resolve these issues. At this stage we are not performing transformations rather we are just copying data from source to destination. For this purpose the easiest method is the use of wizard. Wizard would create package for us including all required tasks as:

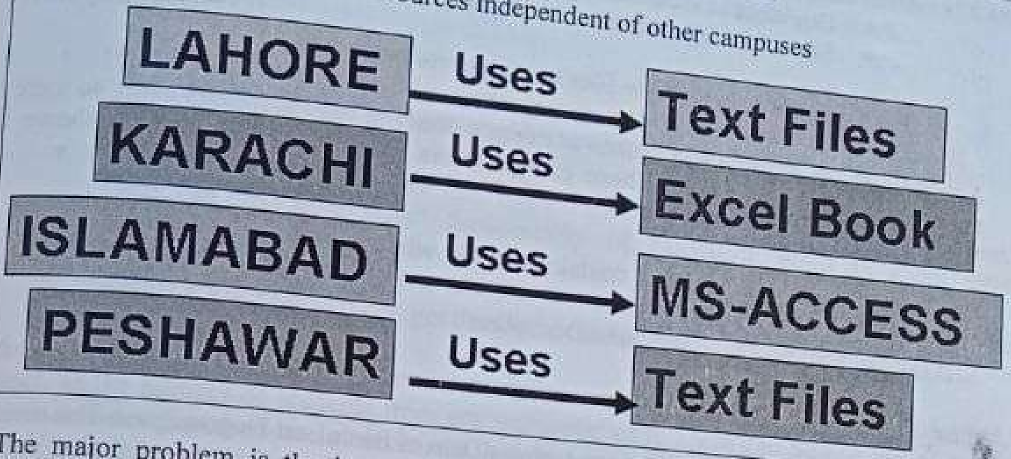
Important

Past Question: Explain the non standardized attributes of 1/0 and 0/1 stored in attribute library do what problem?

Now by looking at each of the campus data individually, we found following problems that need to be considered and solved properly before extracting the data and ultimately loading it into the central repository.

Problem-1: Non-Standard data sources

- Each campus uses data sources independent of other campuses



The major problem is the inconsistent data sources at different campuses. The slide summarizes the data sources at four campuses. We can see that Lahore and Peshawar campuses are using text files while Islamabad and Karachi campuses are using MS Access and MS Excel respectively.

Problem-2: Non-standard attributes

	Identificaion	Gender (M/F)	Degree
Lahore	SID	Gender (0/1) 4th Col	BS/MS
Karachi	St_ID	M/F 5th Col	Separate books
Islamabad	Roll Num	Gender (1/0) 9th Col	Discipline
Peshawar	Reg#		

The second problem is non standardized attributes across campuses. While looking at the header of data from different campuses we came to know the following problems regarding attributes and is summarized in the table in the slide.

Ans: ✓

Each of the campuses uses different attribute name for the identification or primary keys e.g. Lahore uses *SID* while Peshawar uses *Reg#* and so on.

✓

Different conventions for representing Gender across the campuses e.g. Lahore campus uses 0/1 while Islamabad uses 1/0 for representing male and female respectively.

✓

Similarly, there are different conventions for representing degree attribute across different campuses.

- Establishes connection through source / destination systems
- Creates similar table in SQL Server
- Extracts data from text files
- Applies very limited basic transformations, if required
- Loads data into SQL Server table

Extracting Data for Lahore Campus

- **First of all load data for the Lahore campus**
  1. Connect to source Text files
  2. Connect to Destination SQL Server
  3. Create new database 'Lahore_Campus'
  4. Create two tables Student & Registration
  5. Load data from the text files containing student information into Student table
  6. Load data from the text files containing registration records into Registration table
- **Import/Export Wizard is sufficient to perform the above mentioned task easily**

Loading data for Lahore campus includes following tasks:

1. Connect to source Text files

Since there are many text files for Lahore campus, we need to load those text files separately. First of all, select the file that is to be loaded first.

2. Connect to Destination SQL Server

In this case our source system is a text file. For transformation and standardization we will load all data as it is from source file to the SQL server and then through powerful tools of SQL Server, we will perform these intended task.

3. Create new database 'Lahore_Campus'

To load data for four campuses we will develop four separate databases. So, to load data for Lahore campus we will create a new data base named 'Lahore_Campus'.

4. Create two tables Student & Registration

All files containing student information will be loaded in one table Student and all other files containing registration information will be loaded in other table Registration. After this step we will have two populated tables only.

5. Load data from the text files containing student information into Student table.

6. Load data from the text files containing registration records into Registration table

long in past paper

Seven steps to Extract Data using SQL and DTS wizard:

Ans:

3. Choose a Database
    - Specification of file format incase of Text files
4. Specify the Destination
5. Choose Destination Database
    - Selection of existing database or creation of a new database
6. Select a table
    - Selection of existing table or creation of a new table
7. Finalizing and Scheduling the package

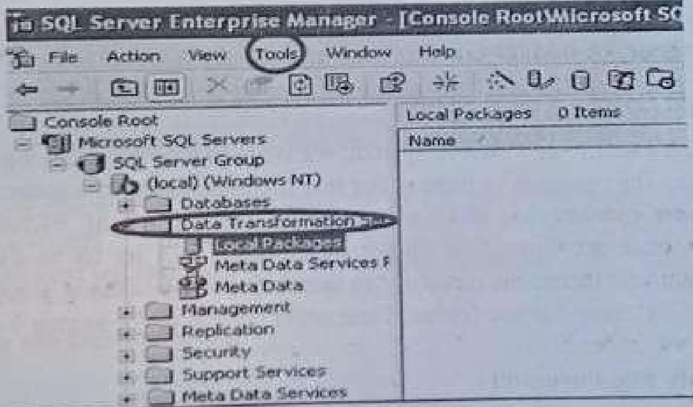
The slide states seven simple steps to create a package for data loading through wizard. Lets discuss each of the steps in detail.

Step1: Launch the wizard(1)

- **Two methods to launch the wizard**
    - Start > Programs > Microsoft SQL Server > Import & Export Data
    - Start > Programs > Microsoft SQL Server > Enterprise Manager
        1) On console root drop Data Transformation Service node
        2) Tools > Data Transformation Service > Import/Export data

These are two different methods to launch the wizard. We can use either.

Step1: Launch the wizard(2)



The slide shows the main screen of SQL Server enterprise manager. In the left pane we have Console root. We can see Data Transformation services highlighted. Expand the node mentioning Data Transformation Services and then press Tools in the menu bar. This will lead you to launch the wizard to load data.

Cong

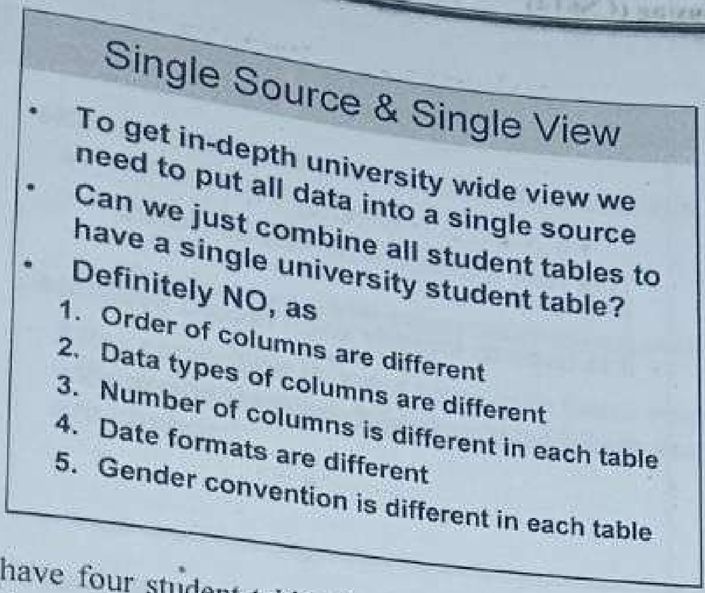
6. **Load data from the text files containing registration records into Registration table**

Import/Export Wizard is sufficient to perform all above mentioned tasks easily. So we will use the wizard as it can provide us good functionality in this scenario.

Seven Steps to Extract Data Using Wizard using SQL and DTS wizard

1. Launch the Wizard
2. Choose a Data Source

Slide 3



Single Source & Single View

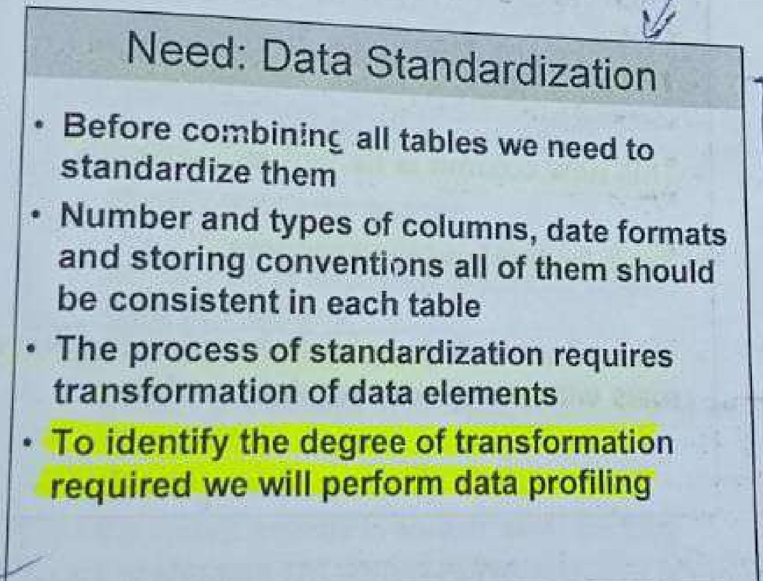
- To get in-depth university wide view we need to put all data into a single source
- Can we just combine all student tables to have a single university student table?
- Definitely NO, as
  1. Order of columns are different
  2. Data types of columns are different
  3. Number of columns is different in each table
  4. Date formats are different
  5. Gender convention is different in each table

At this time we have four student tables in different SQL databases. To consolidate all tables to get single source of truth we can not just glue all tables because of following facts:

Order of columns are different
Data types of columns are different
Number of columns is different in each table
Date formats are different
Gender convention is different in each table

Past

Slide 4

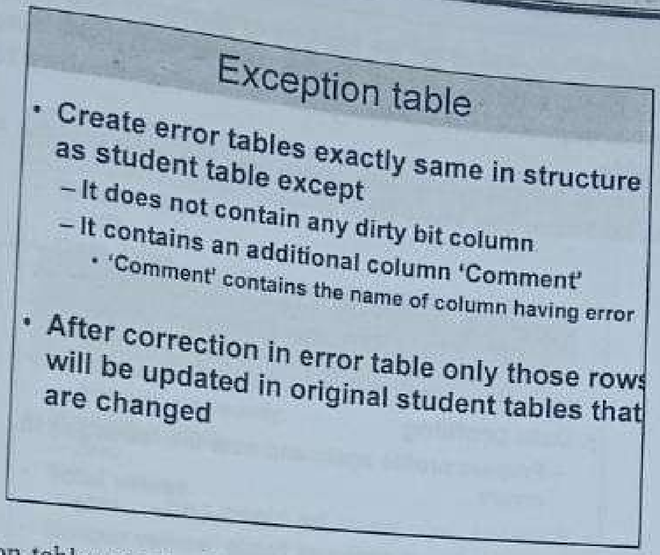


Need: Data Standardization

- Before combining all tables we need to standardize them
- Number and types of columns, date formats and storing conventions all of them should be consistent in each table
- The process of standardization requires transformation of data elements
- To identify the degree of transformation required we will perform data profiling

To remove the factors that are not letting us putting all data together we need to standardize all tables. Standardization process involves the consistency of number and types of columns, date formats, and storing conventions across all campuses and more. To standardize we need to transform data elements. To identify the degree of transformation required we need to perform data profiling.

Slide 7

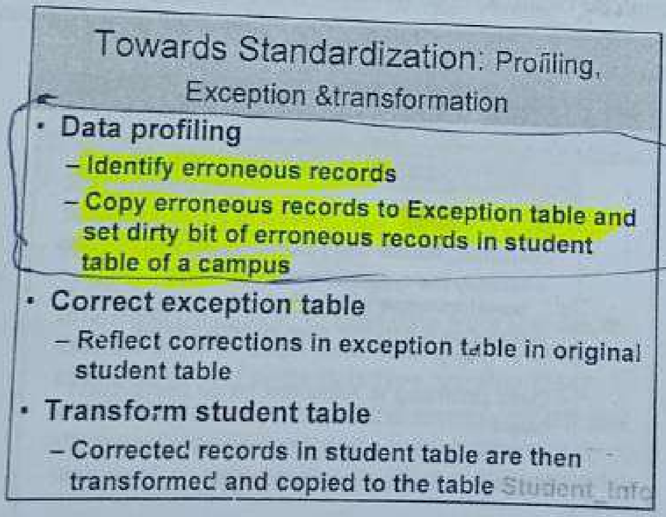


Exception table

- Create error tables exactly same in structure as student table except
  - It does not contain any dirty bit column
  - It contains an additional column 'Comment'
    - 'Comment' contains the name of column having error
- After correction in error table only those rows will be updated in original student tables that are changed

Error or exception table contains the copy of records that have corrupted values for any column in original student table. Error table is the copy of original student table except instead of dirty bit we will have comments column in the error table. In comments column we store the name of columns in which we encountered errors for this particular row. For example consider a record 'R', it has missing gender and incorrect date of birth. For the record R we will have comment **[Gender], [Date of Birth].** These comments will be used while correction of error tables. After correction of error table we will update corresponding rows in the original student table.

Slide 8



Towards Standardization: Profiling, Exception &transformation

- Data profiling
  - Identify erroneous records
  - Copy erroneous records to Exception table and set dirty bit of erroneous records in student table of a campus
- Correct exception table
  - Reflect corrections in exception table in original student table
- Transform student table
  - Corrected records in student table are then transformed and copied to the table Student_Info

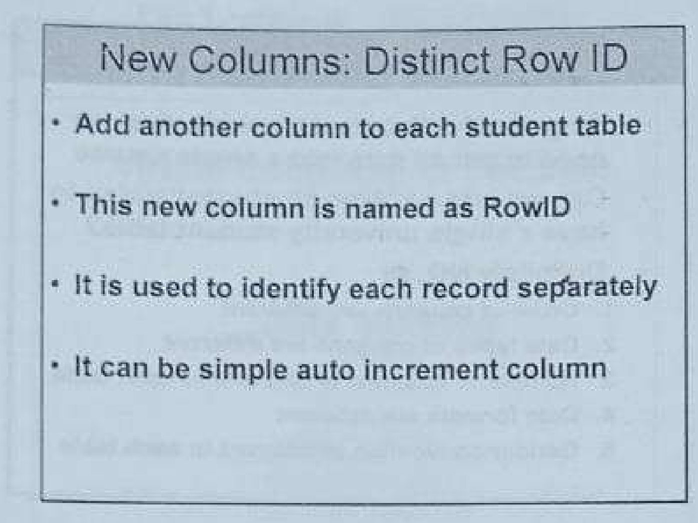
Past

Data profiling is a process which involves gathering of information about column through execution of certain queries with intention to identify erroneous records. In this process we identify the following:

- Total number of values in a column
- Number of distinct values in a column
- Domain of a column
- Values out of domain of a column
- Validation of business rules

Past

Slide 5

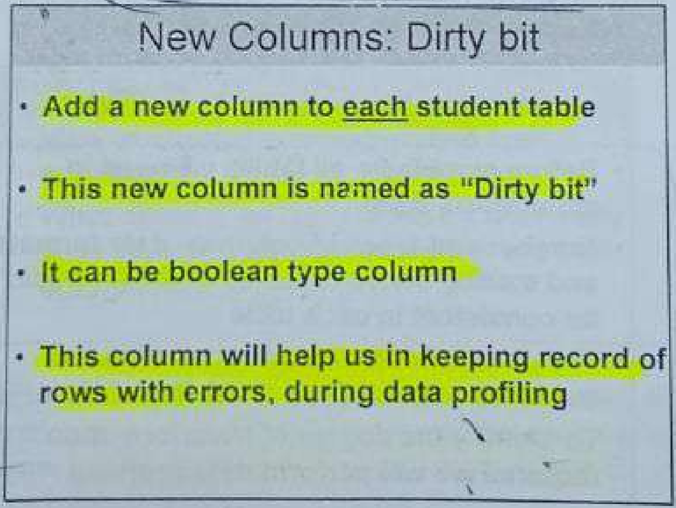


New Columns: Distinct Row ID

- Add another column to each student table
- This new column is named as RowID
- It is used to identify each record separately
- It can be simple auto increment column

By this time, due to lack of primary key, records can not be identified uniquely. At this stage we need an attribute that can identify each record uniquely. We need such an identifying attribute at this stage because a lot of records in this stage will be put to error or exception table due to error in any of the columns. In the error table they will be corrected and later on after correction the changes will be updated in original student table. For this update we need a column like row id to join error table with student table.

Slide 6 ✓ (short in part)



New Columns: Dirty bit

- Add a new column to each student table
- This new column is named as "Dirty bit"
- It can be boolean type column
- This column will help us in keeping record of rows with errors, during data profiling

To keep record of the rows that have been inserted into error tables due to certain errors we need an additional column in student table that will serve as dirty bit. Dirty bit of those records is set to true that are inserted in the error table.