Colorado Technical University

Colorado Technical University
Instructor: Dr. John Conklin
Unit 5 – Maintenance & SQL Scripts

AGENDA

- Performance Monitoring
- Database Tuning
- Back-up and Recovery
- Database Management Problems
- Extract, Transform, Load (ETL)

https://techcrunch.com/

PERFORMANCE MONITORING/MANAGEMENT

PERFORMANCE MANAGEMENT

- There is no consensus as to what performance management is.
- IT research firms observed that business intelligence software vendors, vendors that had data-mining query-and-reporting functionality rather than production the raw transactions data were integrating analytical information across multiple departments and systems.
- A better way to understand what performance management is about is to understand what problems it solves, such as:
  - Failure by executives to execute their well-formulated strategy.
  - Lack of trust among managers to achieve results in an increasing concern.
  - Change is constant.
  - Mistrust of the managerial accounting system and its flawed, inaccurate and/or incomplete product, channel and customer profitability reporting.
  - Poor customer value management
  - Dysfunctional supply chain management

(Cokins, G, 2009)

PERFORMANCE MANAGEMENT

- The result is that rather than just *monitoring* the dials on the performance dashboard, organizations *move* those dials. The purpose of performance management is not just managing but *improving* performance.
- One of the problem we are faced with business intelligence information is that it does not always complete the task of solving the problem or moving to the next step of creating and realizing value.

(Cokins, G, 2009)

PERFORMANCE

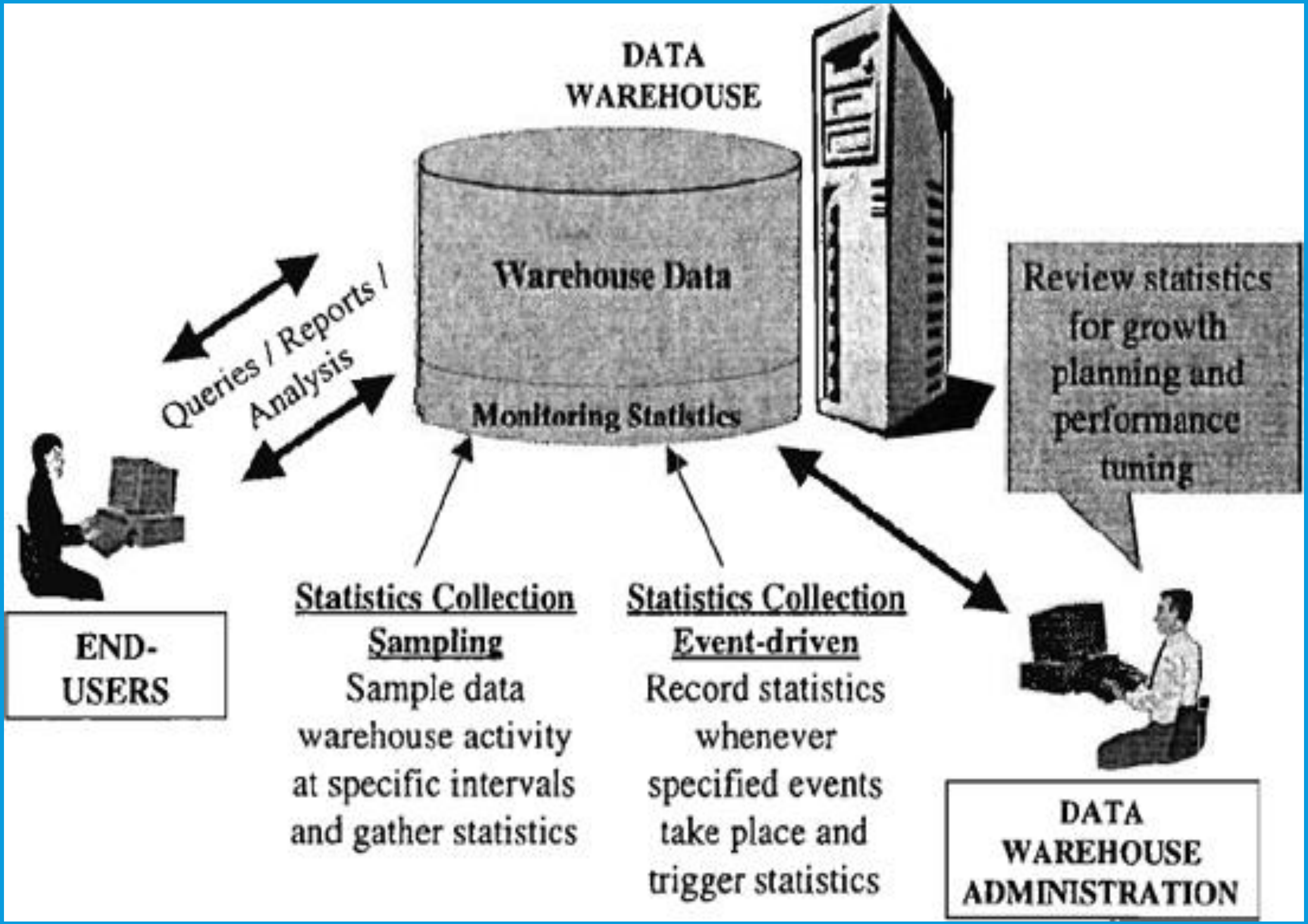
- One of the most important factors that face any database implementation project is database performance.
- Not all DBMS system have performance monitoring and fine-tuning tools embedded in their software, which makes it difficult to evaluate performance.
- Performance evaluation is also made difficult by fact that there are no standardized performance measure in place.
- Important factors in database performance also include:
  - System and database configuration parameters, including:
    - Data placement
    - Access path definitions
    - Use of indexes, and
    - Buffer size.

MONITORING

- Monitoring is comparable to what happens in an OLTP system, except there is one major difference. Monitoring an OLTP system dwindles in comparison to the monitoring done in a data warehouse environment.
- Desired results can not be obtained unless the monitoring of the data warehouse is done in a formalized manner.

- Collection of statistics
  - Indicators that provide information about data warehouse functions is called monitoring statistics.
  - Two common methods apply to this collection process:
    - ***Sampling Methods*** – measure specific aspects of system activity at regular intervals. This method has minimal impact on the system overhead.
    - ***Event-driven Methods*** – measures only when a specific event takes place. This method adds to the system overhead, but are more through then sampling methods.

MONITORING (CONT.)


DATA WAREHOUSE
Warehouse Data
Monitoring Statistics
Queries / Reports / Analysis
Review statistics for growth planning and performance tuning
END-USERS
Statistics Collection Sampling
Sample data warehouse activity at specific intervals and gather statistics
Statistics Collection Event-driven
Record statistics whenever specified events take place and trigger statistics
DATA WAREHOUSE ADMINISTRATION

(Ponniah, P, 2001)

MONITORING (CONT.)

- Collection of statistics (cont.)
  - The tools that come with the database server and host operating system are generally turned on to collect the monitoring statics.
  - Most of these tools gather the values for the indicators and then also interpret the results.
  - The following is a list of monitoring statics that are most useful:
    - Physical disk storage space utilization
    - Number of times the DBMS is looking for space in blocks or causes fragmentation
    - Memory buffer activity
    - Buffer cache usage
    - Memory management
    - Profile of the warehouse content
    - Size of each database table
    - Accesses to the fact tables
    - Number of completed queries by time slots during the day.
    - Count of valid users.
    - Query response times.
    - Number of active tables in the database.

MONITORING (CONT.)

- Using statistics for Growth Planning
  - The number of users and complexity of queries is going to increase as you deploy more versions of your data warehouse.
  - Monitoring statistics are going to prompt the following list of actions:
    - Allocate more disk space to existing database tables
    - Plan for new disk space for additional tables
    - Modify file block management parameters to minimize fragmentation
    - Create more summary tables to handle large number of queries looking for summary information
    - Reorganize the staging area files to handle more data volume
    - Add more memory buffers and enhance buffer management
    - Upgrade database servers
    - Offload report generation to another middle tier
    - Smooth out peak usage during the 24-hour cycle
    - Partition tables to run loads in parallel and to manage backups

MONITORING (CONT.)

- Using statistics for Fine-Tuning
  - You will find that a number of monitoring statics prove to be very useful over time for fine-tuning the data warehouse.
  - The following list of functions are ones that are normally improved based on information gathered from statistical monitoring:
    - Query performance
    - Query formulation
    - Incremental loads
    - Frequency of OLAP loads
    - OLAP system
    - Data warehouse content browsing
    - Report formatting
    - Report generation (Ponniah, P, 2001)

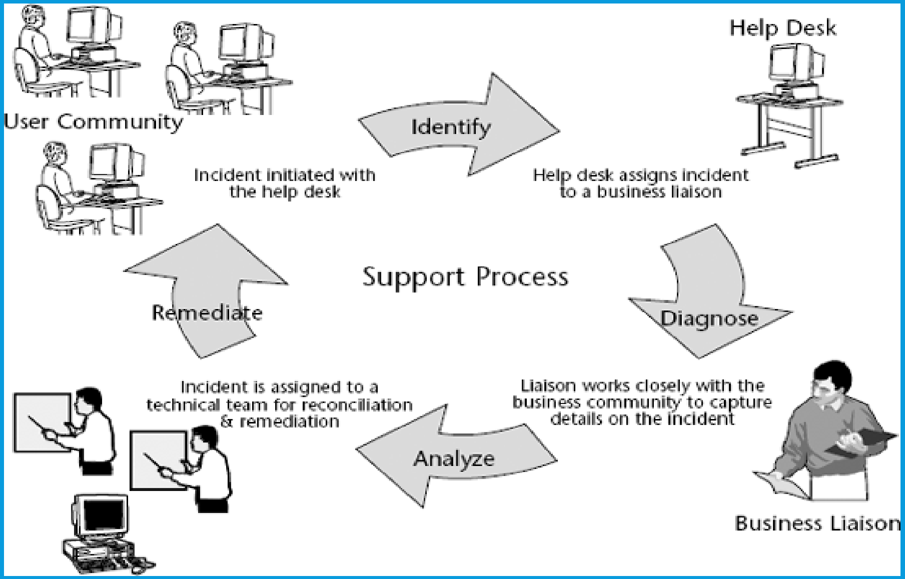
- Data warehouse performance monitoring and management are critical components to the overall health of your data warehouse. This effort allows the end users the ability to draw meaningful insights from this data.

UNDERSTANDING PERFORMANCE CHALLENGES

- One of the highest risk factors to the long-term success of a data warehouse initiative is performance monitoring.
- Performance challenges are very often expressed in complaints from the business-users when their response times fail to meet their expectations.
- Performance tuning and monitoring are often considered post-production support activities and not an integral to full life-cycle development activities.
- The risks associated with performance can only be reconciled with a proper approach spanning the solutions life cycle.

UNDERSTANDING PERFORMANCE CHALLENGES (CONT.)

- Who is responsible?
  - Most often it is the responsibility of the DBA or technical team to properly identify, analyze and remediate the impacts associated with any performance related issue.
  - A common mistake, when a perceived performance challenge is reported, is querying the database with the goal of proving that poor performance is not an infrastructure-related issue but is due to improper use of the application.
  - It is important to establish a full life-cycle support infrastructure that can properly identify, analyze, and route performance incidents to the proper team for reconciliation.


User Community
Help Desk
Identify
Incident initiated with the help desk
Help desk assigns incident to a business liaison
Support Process
Remediate
Diagnose
Incident is assigned to a technical team for reconciliation & remediation
Liaison works closely with the business community to capture details on the incident
Analyze
Business Liaison

(Rayman, J, et al., 2007)

UNDERSTANDING PERFORMANCE CHALLENGES (CONT.)

- Causes of Poor Performance
  - Poor performance can have a variety of causes such as the wrong combination of hardware, software, and storage for an increasing workload or a poor data model design that doesn't meet the changing requirements of the business.

(Rayman, J, et al., 2007)

EDUCATING THE BUSINESS

- New users of business intelligence solutions can have a detrimental impact on the overall performance of the solution and may not leverage it properly.
- A comprehensive training plan can help ensure that these users gain a full understanding and appreciation of the capabilities that the new solution can provide.
- Business Processes
  - Optimization of business processes and new interfaces can introduce user challenges that impact the overall performance and efficiency of the solution, especially during early deployment.
  - The project team should use change management techniques to ensure a seamless transition from existing business processes and solutions maps for enhanced processes.

(Rayman, J, et al., 2007)

EDUCATING THE BUSINESS (CONT.)

- Data
  - Data models can provide a comprehensive view of data, but are more difficult to understand and are best suited for developers and power users.
  - Many organizations address this challenge by constructing a centralized metadata repository that is used to catalog all corporate data.
  - The most common scenarios when users lack the necessary understanding of data include ad hoc queries using the wrong data structures, excessive summarization activities, and improperly joined data based on common naming or data characteristics.
- **Hardware**
  - One possible contributor to poor performance can be a poorly designed hardware infrastructure.
  - Many underlying components that determine hardware performance, including processing capacity, memory, and networking and storage systems.
  - It can lead to configuration complexity especially where the administrative staff has limited experience with multiple vendor offerings.

(Rayman, J, et al., 2007)

EDUCATING THE BUSINESS (CONT.)

- *Processing Capacity*
  - A common contributing factor to poor hardware performance is inadequate processing capacity.
  - Processing capacity symptoms are relatively easy to diagnose using operating system utilities that measure CPU utilization.
- *Memory*
  - Constrained memory can significantly impact performance in data warehousing solutions.
- *Network Infrastructure*
  - A lesser contributing factor impacting the performance of a data warehousing solution is the network infrastructure bandwidth.
  - Proper networking strategies should include filtering of traffic such that bulk transfers and file movements are isolated from active user communities.

(Rayman, J, et al., 2007)

EDUCATING THE BUSINESS (CONT.)

- *Storage Infrastructure*
  - One of the most challenging hardware components to design for optimal performance is the storage infrastructure.
  - Many business intelligence environments are I/O intensive, especially where large amount of data is retrieved, and require more channels and bandwidth to storage devices.
- *Memory*
  - Constrained memory can significantly impact performance in data warehousing solutions.
- *Network Infrastructure*
  - A lesser contributing factor impacting the performance of a data warehousing solution is the network infrastructure bandwidth.
  - Proper networking strategies should include filtering of traffic such that bulk transfers and file movements are isolated from active user communities.

(Rayman, J, et al., 2007)

EDUCATING THE BUSINESS (CONT.)

- **Software**
  - Software configuration also impacts performance.
  - To fully understand the overall impact of software configurations on a business intelligence solution, we segregate software into two distinct categories: database, and the business intelligence tools and applications.

  - *Database* –
    - The flexibility and robustness of the Oracle database is one reason why it is so popular for data warehousing solutions.
    - The most common challenges are related to:
      - Database configurations for business intelligence
      - Poor optimization execution plans
      - Application of features
      - Adjusting database parameters
      - Storage management
      - Sizing system global area (SGA) memory
      - Variability in response
      - More data yields slower performance

(Rayman, J, et al., 2007)

BUSINESS INTELLIGENCE TOOLS AND APPLICATION SOFTWARE

- The software is the window into the data and provides not only visualization capabilities, but also the ability to explore information without the need to understand or write complex SQL.
- All business intelligence tools and applications are not created equal.
- Some organizations, custom applications are developed that address specific business capabilities.
- These applications are often targeted toward a select group of users and often lack the flexibility, scalability, and performance necessary to support large user communities.
- Most organizations choose instead to purchase commercial business intelligence tools and applications that provide a wider range of functionality.
- Such tools sometimes generate complex, sub-optimal SQL that falls short of performance expectations. This is one of the most difficult challenges for administrators where the manual optimization of generated code is not possible without modifications to the underlying database design.

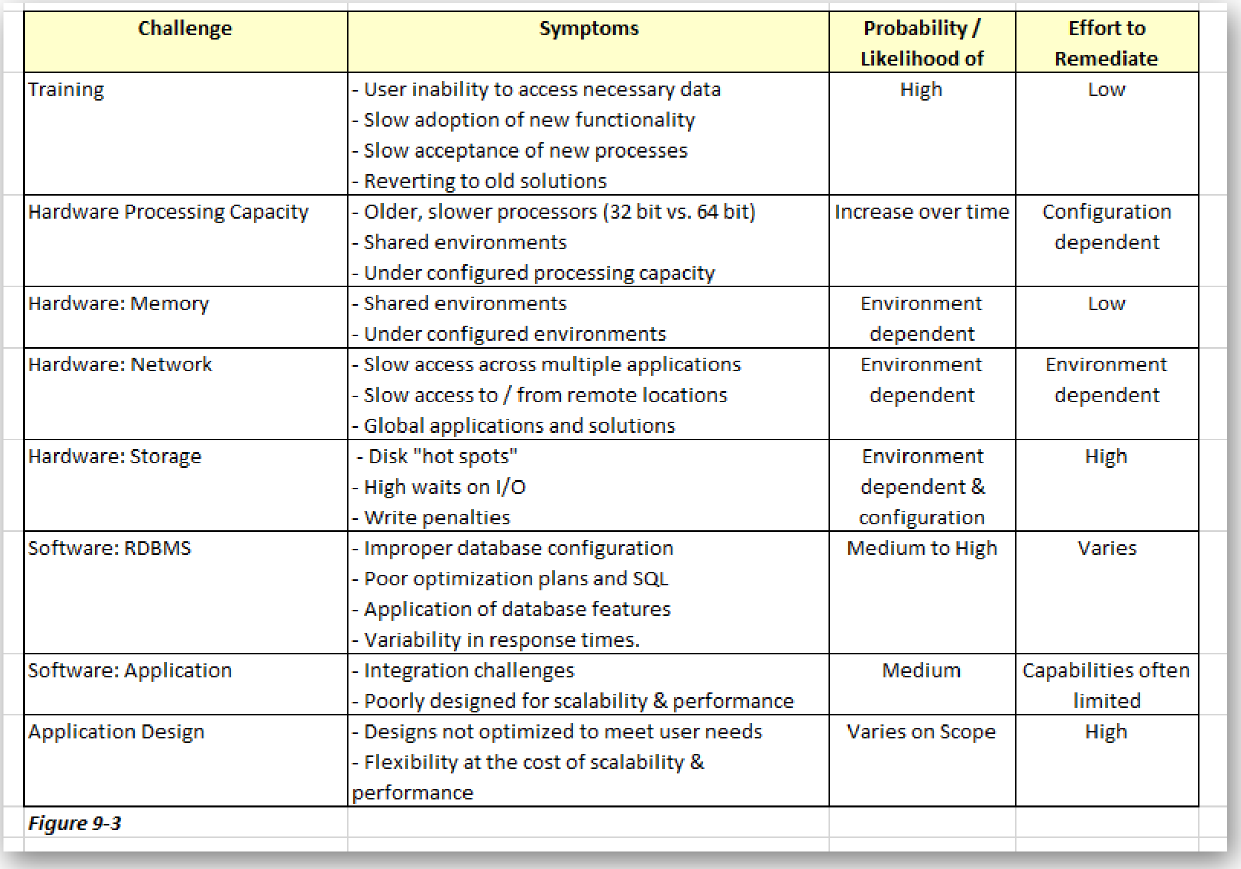
(Rayman, J, et al., 2007)

APPLICATION AND DATABASE DESIGN

- One of the most common causes of poor performance is poor application or database design.
- There are circumstances where the current design might not be aligned to meet business needs, causing a significant negative impact on performance.
- Poor designs can jeopardize a business intelligence solution by limiting solution adoption and lead to full abandonment and searching for alternative solutions.
- Figure 9-3 highlights the performance challenges discussed previously, the underlying symptoms, the likelihood of occurrence and the anticipated effort to remediate.

(Rayman, J, et al., 2007)

APPLICATION AND DATABASE DESIGN (CONT.)



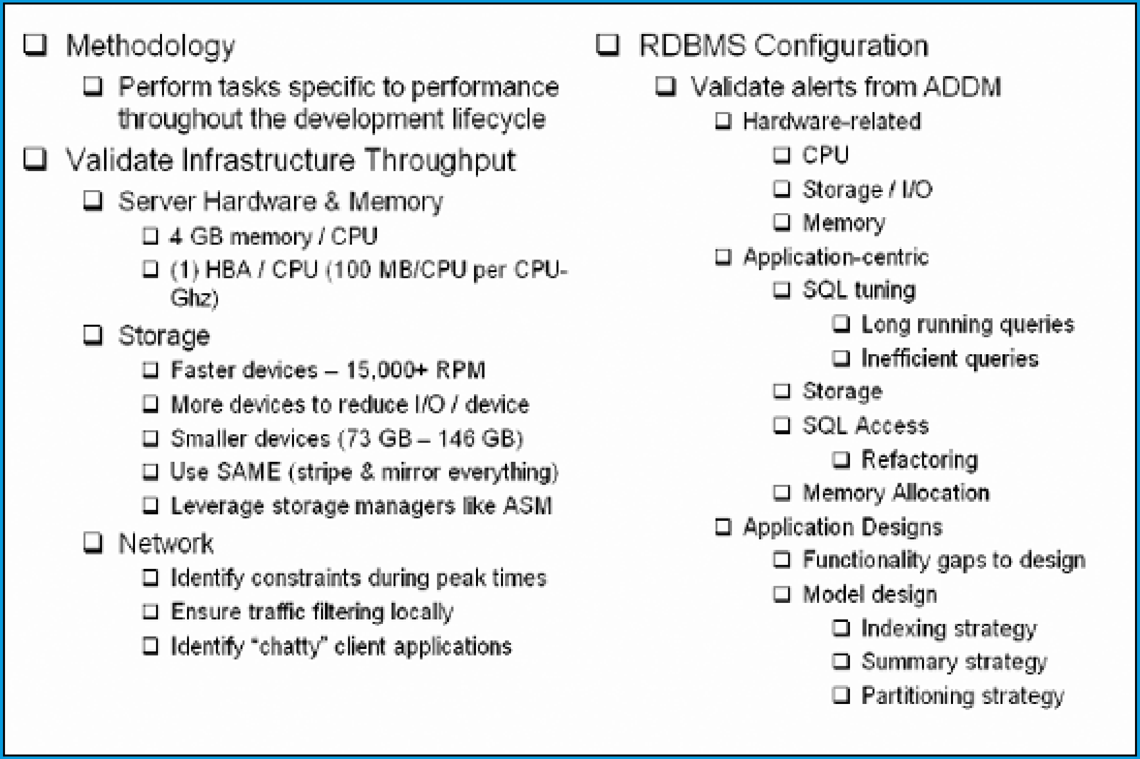
Challenge	Symptoms	Probability / Likelihood of	Effort to Remediate
Training	- User inability to access necessary data - Slow adoption of new functionality - Slow acceptance of new processes - Reverting to old solutions	High	Low
Hardware Processing Capacity	- Older, slower processors (32 bit vs. 64 bit) - Shared environments - Under configured processing capacity	Increase over time	Configuration dependent
Hardware: Memory	- Shared environments - Under configured environments	Environment dependent	Low
Hardware: Network	- Slow access across multiple applications - Slow access to / from remote locations - Global applications and solutions	Environment dependent	Environment dependent
Hardware: Storage	- Disk "hot spots" - High waits on I/O - Write penalties	Environment dependent & configuration	High
Software: RDBMS	- Improper database configuration - Poor optimization plans and SQL - Application of database features - Variability in response times.	Medium to High	Varies
Software: Application	- Integration challenges - Poorly designed for scalability & performance	Medium	Capabilities often limited
Application Design	- Designs not optimized to meet user needs - Flexibility at the cost of scalability & performance	Varies on Scope	High

Figure 9-3

(Rayman, J, et al., 2007)

SUCCESSFUL APPROACHES TO PERFORMANCE TUNING

- Selecting an approach for monitoring and tuning is critical to ensuring performance of your data warehouse meets anticipated service levels.
- To reduce the overall risk of poor performance, focus on solving performance challenges during the design and testing cycles to ensure they are not present in production.



- ❑ Methodology
  - ❑ Perform tasks specific to performance throughout the development lifecycle
- ❑ Validate Infrastructure Throughput
  - ❑ Server Hardware & Memory
    - ❑ 4 GB memory / CPU
    - ❑ (1) HBA / CPU (100 MB/CPU per CPU-Ghz)
  - ❑ Storage
    - ❑ Faster devices – 15,000+ RPM
    - ❑ More devices to reduce I/O / device
    - ❑ Smaller devices (73 GB – 146 GB)
    - ❑ Use SAME (stripe & mirror everything)
    - ❑ Leverage storage managers like ASM
  - ❑ Network
    - ❑ Identify constraints during peak times
    - ❑ Ensure traffic filtering locally
    - ❑ Identify "chatty" client applications
- ❑ RDBMS Configuration
  - ❑ Validate alerts from ADDM
    - ❑ Hardware-related
      - ❑ CPU
      - ❑ Storage / I/O
      - ❑ Memory
    - ❑ Application-centric
      - ❑ SQL tuning
        - ❑ Long running queries
        - ❑ Inefficient queries
      - ❑ Storage
      - ❑ SQL Access
        - ❑ Refactoring
      - ❑ Memory Allocation
    - ❑ Application Designs
      - ❑ Functionality gaps to design
      - ❑ Model design
        - ❑ Indexing strategy
        - ❑ Summary strategy
        - ❑ Partitioning strategy

Figure 9-4. Checklist – best practices approach

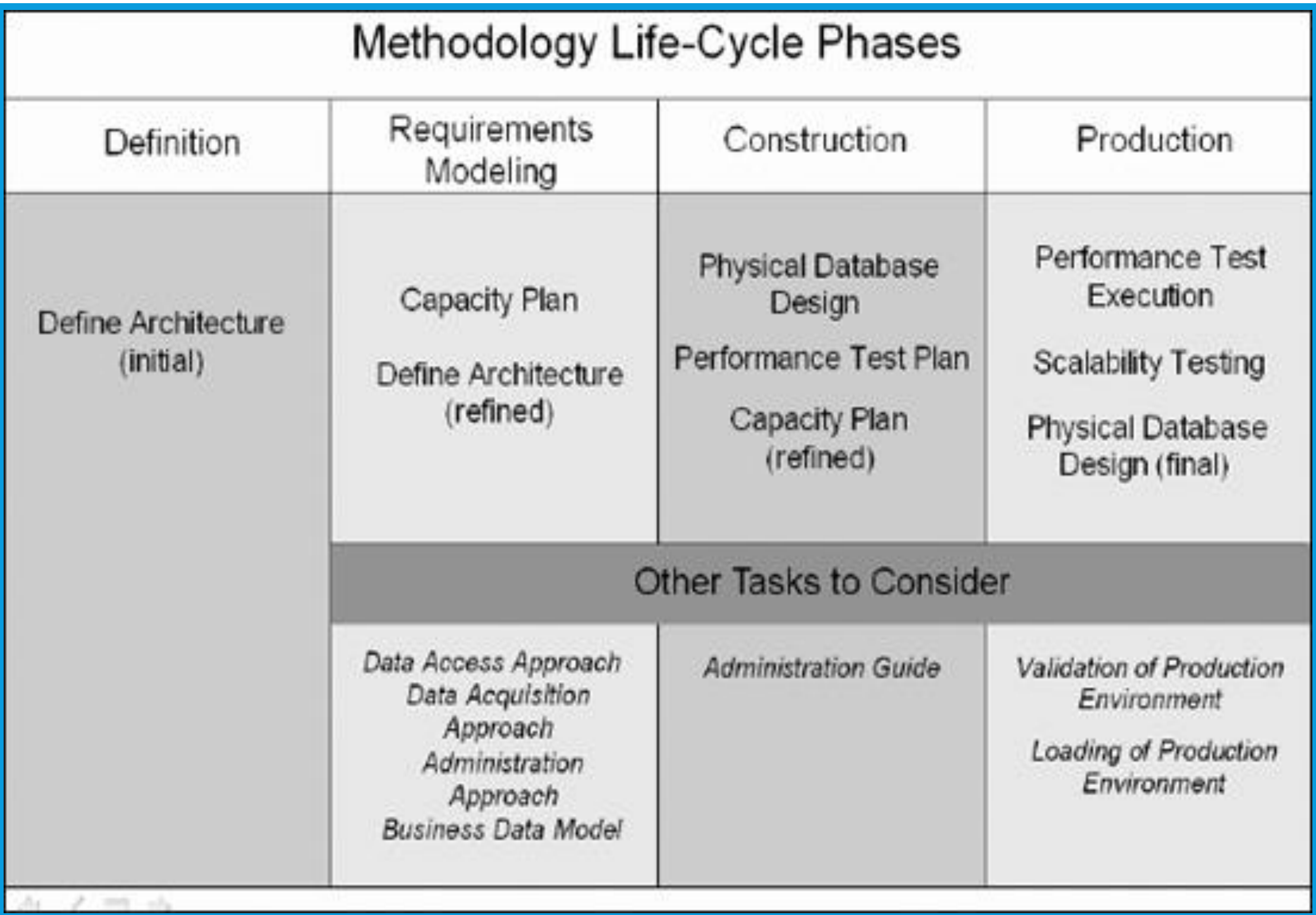
(Rayman, J, et al., 2007)

CRITICAL TASKS FOR PERFORMANCE TUNING LIFECYCLE

- The best approach to avoid performance challenges is to thoroughly understand business requirements and fundamental design principles for data warehousing.
- Many methodology tasks that have correlation to the performance delivered by your business intelligence project. In each phase of the methodology, the associated tasks are designed to ensure the highest levels of performance, capacity, and scale. These are illustrated in Figure 9-5.

(Rayman, J, et al., 2007)

CRITICAL TASKS FOR PERFORMANCE TUNING LIFECYCLE (CONT.)



Methodology Life-Cycle Phases			
Definition	Requirements Modeling	Construction	Production
Define Architecture (initial)	Capacity Plan Define Architecture (refined)	Physical Database Design Performance Test Plan Capacity Plan (refined)	Performance Test Execution Scalability Testing Physical Database Design (final)
	Other Tasks to Consider		
	Data Access Approach *Data Acquisition Approach* *Administration Approach* *Business Data Model*	*Administration Guide*	*Validation of Production Environment* *Loading of Production Environment*

Figure 9-5.

(Rayman, J, et al., 2007)

CRITICAL TASKS FOR PERFORMANCE TUNING LIFECYCLE (CONT.)

- Definition Phase
  - Commences solution discovery by reviewing business and system objectives while confirming existing documentation and prioritizing the high-level business requirements for the solution.
    - Task: Define Architecture - begins the formulation of the technical design based on initial business requirements.
- Requirements Modeling Phase
  - Begins solution transition from high-level requirements gathered during the definition phase to sufficient detail to ensure that business processes and requirements are met.
  - The following sections describe the Requirements Modeling phase tasks that provide the greatest impact.
    - Task: Capacity Plan - provides an initial pass at defining infrastructure capacities based on current requirements and anticipated future growth.
    - Task: Define Architecture – (Refined)- refined during the requirements modeling phase to capture additional details on the technical, data, and application architecture layers.
    - Task: Others to Consider - There are several other tasks that are addressed during the Requirements Modeling phase. These tasks include *Data Access Approach*, *Data Acquisition Approach*, *Administration Approach*, and *Business Data Model*.

(Rayman, J, et al., 2007)

CRITICAL TASKS FOR PERFORMANCE TUNING LIFECYCLE (CONT.)

- Construction Phase
  - Includes the final design and building of all modules that comprise the data warehouse initiative.
    - Task: Physical Database Design - provides the translation of the logical database design into physical design constructs. The testing spans both functional and infrastructure components to ensure that all modules work together successfully.
    - Task: Capacity Plan – (Refined) - refined during the construction phase to support current and near-term infrastructure capacity needs.
    - Task: Others to Consider - Another task to consider during the construction phase is inclusion of performance-related content in an *Administration Guide*. The *Administration Guide* contains the final set of processes and procedures for administering, managing, and monitoring the target environment.

(Rayman, J, et al., 2007) 27

CRITICAL TASKS FOR PERFORMANCE TUNING LIFECYCLE (CONT.)

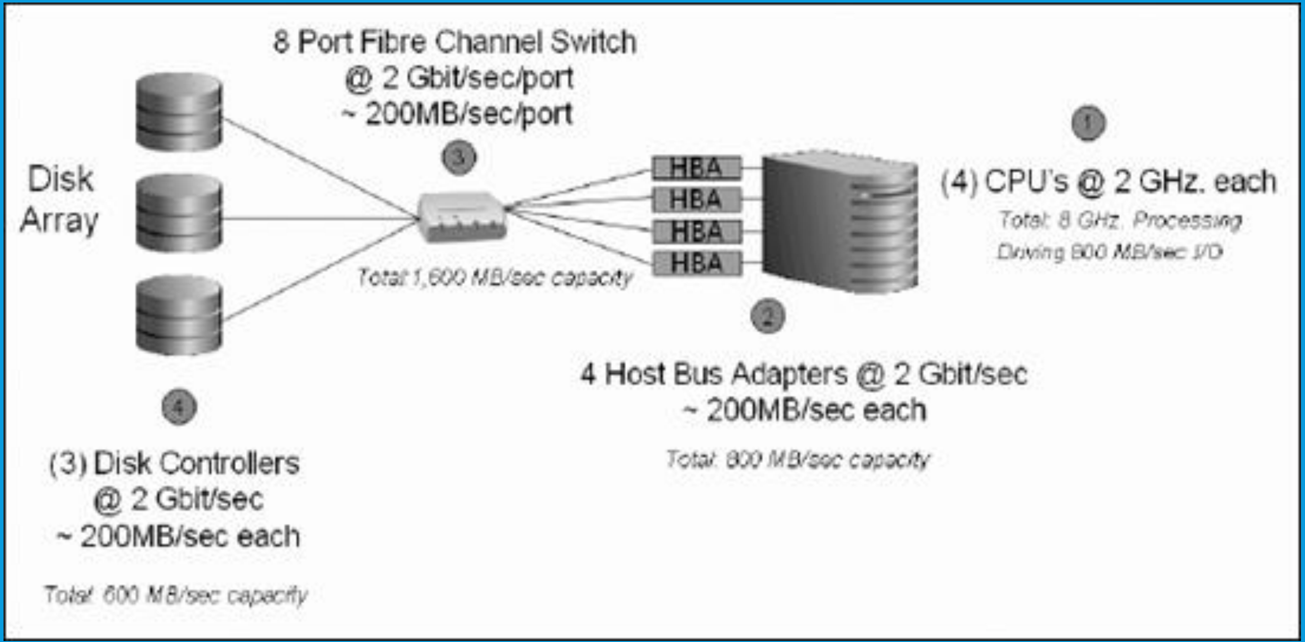
- Production Phase
  - The culmination of the project when the solution is installed and transitioned to production.
    - Task: Performance Test Execution - completed during the production phase and the results are assessed to determine if they meet the functionality and performance requirements.
    - Task: Scalability Testing - designed to execute specific use case tests under increased data volumes and resource loads to characterize the performance.
    - Task: Physical Database Design – (Final) - finalized during the production phase and deployed to the production environment for initial data loading.
    - Task: Others to Consider - The additional tasks during the production phase that impact performance include the *Validation of Production Environment*, results from the *Loading of the Production Environment* and review of *Pre-production Testing Results*.
- Hardware Configuration
  - Although some organizations attempt to extend the life of existing servers, many others find that replacing existing servers with faster and less expensive alternatives provide improvements needed to overcome performance challenges.

(Rayman, J, et al., 2007)

CRITICAL TASKS FOR PERFORMANCE TUNING LIFECYCLE (CONT.)

- Validate Theoretical Throughput
  - Most effective methods to confirm that an existing hardware configuration will scale appropriately is to ensure that the theoretical performance throughput ratios between processor speeds, memory, and I/O subsystems are adequate.

Figure 9-6 - Sample configuration



(Rayman, J, et al., 2007)

CRITICAL TASKS FOR PERFORMANCE TUNING LIFECYCLE (CONT.)

- Additional Infrastructure to Validate
  - There are other infrastructure components that should be validated to ensure that the throughout of the system is optimal, including network capacity, server memory, and the storage subsystem.
  - Network capacity can impact performance especially in environments that have undergone recent expansion or are part of aging infrastructures.
  - Server Memory - can become a limiting factor to server performance, especially where multiple solutions reside on the same server.
  - Storage Subsystem - can have a significant impact on performance if improperly sized and configured. When selecting storage devices to support a data warehouse initiative, it is important to select the devices with the fastest spindle speed. Another important factor in reducing I/O device contention is to ensure that data is spread across as many devices as possible.

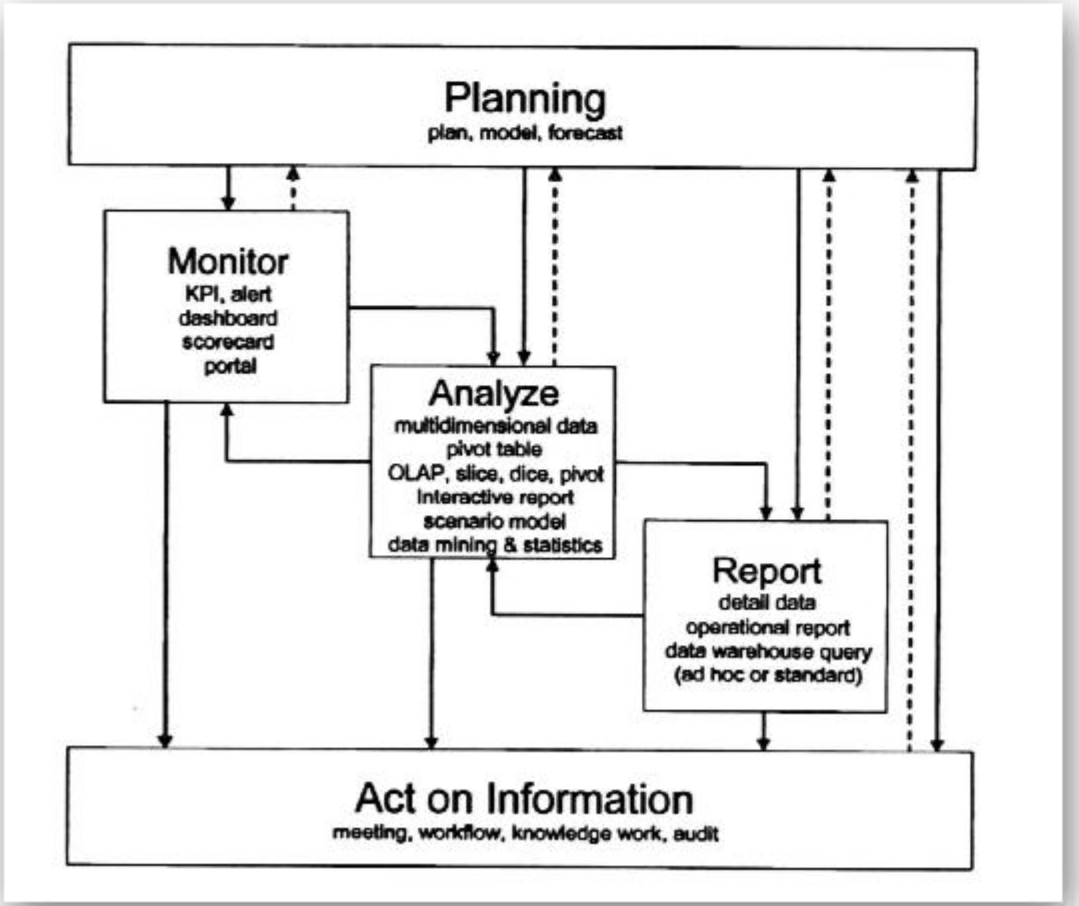
(Rayman, J, et al., 2007)

CORPORATE PERFORMANCE MANAGEMENT (CPM)

- CPM is an umbrella term used to describe the methodologies, metrics, processes and systems used to monitor and manage the business performance of any enterprise.
- Some of the primary expectations of CPM include:
  - *The use of metrics* – You cannot change what you cannot measure.
  - *The use of a balanced set of performance metrics* – a call for balancing short term and long-term objectives.
  - *The right-time delivery of actionable management information* – fundamental need for up-to-date management information.
  - *Horizontally integrated management* - aligning the value creating steps throughout the core enterprise processes with relentless focus on the customer.
  - *Vertically integrated management* – requires a clear articulation of the strategic objectives and hypotheses.
  - *Closed loop management* – objective is to be and remain in control of the execution.

(Viaene, S & Willems, J, 2007)

CORPORATE PERFORMANCE MANAGEMENT (CPM) (CONT.)



(Viaene, S & Willems, J, 2007)

https://www.thecrazyprogrammer.com/2014/12/database-tuning.html

DATABASE TUNING

GENERAL SUGGESTIONS

- Some general database tuning suggestions are:
  - Good database design
  - Disk I/O optimization – related directly to throughput and scalability. I/O reduces the number of disk accesses and might substantially increate throughput in a heavily loaded production environment.
  - Checkpointing – periodically flushes all dirty cache data to disk, which increases I/O activity and system resource usage for the duration of the checkpoint.
  - Disk and database over head can sometimes be dramatically reduced by batching multiple operations together and/or increasing the number of operations that run parallel.

GENERAL SUGGESTIONS – PERFORMANCE TUNING TIPS

- Some general tuning tips are:
  1. Database statistics – the most important resource to SQL optimizer is the statistics collected for different tables within the catalog.
  2. Create optimized indexes – it is important to have a right balance of index on tables. Besides the number of indexes, fields that are involved and their order is important.
     1. Composite Index: Indexes containing more than one field are called composite index. Such indexes should be created if you expect to run queries that will have multiple fields in the WHERE clause and all fields combined will give significantly less rows than the first field alone.
     2. Clustered index - A clustered index determines the physical order of data in a table - meaning the actual data is sorted according to the fields in the index. This is similar to a telephone directory, which arranges data by last name.
  3. Avoid functions on RHS of the operator
  4. Predetermine expected growth - One way to minimize this negative affect is to specify an appropriate value for fill factor when creating indexes.
  5. Specify optimizer hints in SELECT
  6. Use EXPLAIN
  7. Avoid foreign key constraints - Foreign keys constraints ensure data integrity at the cost of performance. Therefore, if performance is your primary goal you can push the data integrity rules to your application layer.

(Synametrics.com, n.d.)

GENERAL SUGGESTIONS – PERFORMANCE TUNING TIPS (CONT.)

- Some general tuning tips are:
  8. Two heads are better than one - When you use multiple physical disks, I/O operations speed up significantly since more heads fetch data in parallel.
  9. Select limited data - less data retrieved, the faster the query will run.
  10. Drop indexes before loading data - Consider dropping the indexes on a table before loading a large batch of data. This makes the insert statement run faster. Once the inserts are completed, you can recreate the index again.

- SQL performance tuning can be an incredibly difficult task, particularly when working with large-scale data where even the most minor change can have a dramatic (positive or negative) impact on performance.
- When working with large-scale data, even the most minor change can have a dramatic impact on performance.

(Synametrics.com, n.d.)

(toptal.com, n.d.)

SQL PERFORMANCE TUNING – FOR DEVELOPERS

- Indexes - indexing is an effective way to tune your SQL database that is often neglected during development. In basic terms, an index is a data structure that improves the speed of data retrieval operations on a database table by providing rapid random lookups and efficient access of ordered records.
  - This means that once you've created an index, you can select or sort your rows faster than before.
  - Basically, the goal is to index the major searching and ordering columns.
  - Further, DBAs often drop their SQL indexes before performing batch inserts of million-plus rows to speed up the insertion process. After the batch is inserted, they then recreate the indexes.
- Execution Plans in SQL Server - Execution Plan tool in SQL Server can be useful for creating indexes.

(toptal.com, n.d.)

SQL PERFORMANCE TUNING – FOR DEVELOPERS (CONT.)

- Avoid coding loops
- Avoid correlated SQL Subqueries - A correlated subquery is one which uses values from the parent query. This kind of SQL query tends to run row-by-row, once for each row returned by the outer query, and thus decreases SQL query performance.
  - A more efficient SQL performance tuning technique would be to refactor the correlated subquery as a join
- Select Sparingly - One of my favorite SQL optimization tips is to avoid SELECT *! Instead, you should individually include the specific columns that you need. Again, this sounds simple, but I see this error all over the place.
- The Wise Use of Temporary Tables (#Temp) - Temporary tables usually increase a query's complexity. If your code can be written in a simple, straightforward manner, I'd suggest avoiding temp table

(toptal.com, n.d.)

UNDERSTANDING DATABASE TUNING

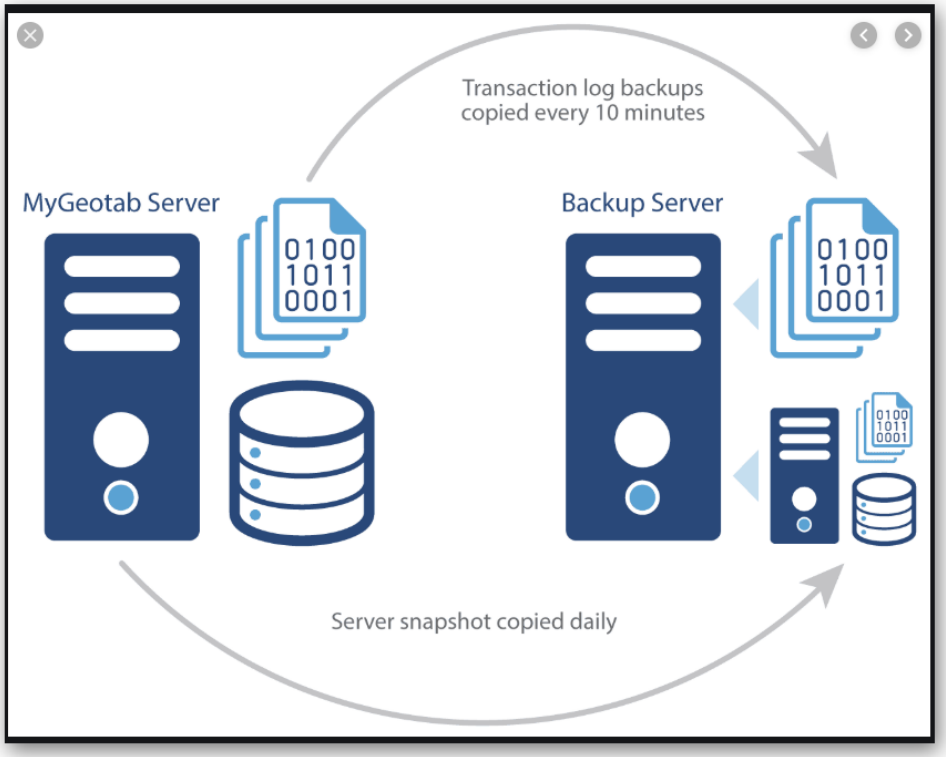
- The concept of database tuning involves many factors beyond the DBMS, but understanding what these factors are permits an assessment of its integration in database management courses.
- A performance bottleneck occurs when a database component is assigned an excessive amount of work at a given time, which it is incapable of satisfying adequately.
- We list some of these components in order to provide a better understanding of the tasks facing IT graduates during database tuning (Vaidyanatha, 2001; Niemiec and Lane, 2002; McGehee, 2003a,b; SQL-Server-Performance.com, 2003; Mohamed, 2002; and Quest Software, 2003).
  - ***Disk Input/output (I/O) Bottlenecks*** - i/o operations require read/write disk drive heads to physically move across the drive platters, potentially incurring a significant time penalty in the process.
  - ***Central Processing Unit (CPU) Bottlenecks*** - occur when too many resources compete for computer processing time at once

(Pons, A, 2003)

UNDERSTANDING DATABASE TUNING (CONT.)

- ***Random Access Memory (RAM) Bottlenecks*** - RAM, like the CPU, is a physical resource of the server itself, so any other processes running on the server will take away from the amount of RAM available to the database system.
- ***Network Bottleneck*** - is related to traffic problems on the network, but also to the way that the database server behaves within the network.
- ***Application Decomposition*** - Perhaps the greatest stress on the network is a poorly structured client/server application.
- ***SQL*** - SQL (Structured Query Language) is a declarative language, which hides many of the details concerning the manner that operations are performed.
- ***Denormalization*** - This is a technique to move from higher normal forms of database modeling to lower ones in order to speed up database access. Denormalization is applied during the process of deriving a physical data model from a logical form.
- ***Indexes*** - Indexes may provide the most difficult challenge, as their haphazard use can harm as much as help database performance. The purpose of an index is to enhance the performance of select statements against a table.

(Pons, A, 2003)


Transaction log backups copied every 10 minutes
MyGeotab Server
0100
1011
0001
Backup Server
0100
1011
0001
Server snapshot copied daily

https://www.geotab.com/blog/backup-recovery/

BACK-UP AND RECOVERY

BACKUP VS. ARCHIVE

- The ability to access and recover data is important for companies of all sizes. Data backup vs. archive are two ways you can accomplish this, but how do you know which your company should be using?
- What is a Backup?
  - A backup is a copy of data that can be used to restore the original in the event that your data is lost or damaged. If a company experiences data loss due to hardware failure, human error or natural disaster, a backup can be used to quickly restore that data.
- What is an Archive?
  - An archive is a collection of historical records that are kept for long-term retention and used for future reference. Typically, archives contain data that is not actively used.
- Basically, a backup is a copy of a set of data, while an archive holds original data that has been removed from its original location.

(Solarwindsmsp.com, n.d.)

WHY BACK UP THE DATA WAREHOUSE?

- Loss of data can produce serious consequences.
- A data warehouse houses huge amounts of data that has taken years to gather and accumulate.
- Before the data arrives at the data warehouse, you know that it has gone through an elaborate process of cleansing and transformation.
- Data in the warehouse represents an integrated, rich history of the enterprise.
- It is critical that you are able to recreate the data if and when any disaster happens to strike.
- Within a short time after deployment, the number of users increases rapidly. The complexity of the types of queries and analysis sessions intensifies.
- With a large number of users intimately dependent on the information from the warehouse, backing up the data content and ability to recover quickly from malfunctions reaches new heights of importance.

(Ponniah, P, 2001)

WHY BACK UP THE DATA WAREHOUSE?

- Backup Strategy - A sound backup strategy comprises several crucial factors. Here is a collection of useful tips on what to include in your backup strategy:
  - Determine what you need to back up.
  - The enormous size of the data warehouse stands out as a dominant factor.
  - Strive for a simple administrative setup.
  - Be able to separate the current from the historical data and have separate procedures for each segment.
  - Apart from full backups, also think of doing log file backups and differential backups.
  - Do not overlook backing up system databases.
  - Choosing the medium for backing up is critical.
  - The commercial RDBMSs adopt a "container" concept to hold individual files. A container is a larger storage area that holds many physical files.
  - Although the backup functions of the RDBMSs serve the OLTP systems, data warehouse backups need higher speeds.
  - Plan for periodic archiving of very old data from the data warehouse.

(Ponniah, P, 2001)

WHY BACK UP THE DATA WAREHOUSE?

- Setting up a Practical Schedule - you need to back up the data warehouse properly. But the enormous size is a serious factor in all decisions about backup and recovery. It takes an inordinately long time to back up the full data warehouse.
- Consider the following issues for making the decisions:
  - As you know, backups for OLTP systems usually run at night. But in the data warehouse environment, the night slots get allocated for the daily incremental loads. The backups will have to contend with the loads for system time.
  - If your user community is distributed in different time zones, finding a time slot becomes even more difficult.
  - Mission-critical OLTP systems need frequent backups. In forward recovery, if you do not have regular full backups and frequent log file backups, the users must reenter the portion of the data that cannot be recovered.
  - Setting up a practical backup schedule comes down to these questions. How much downtime can the users tolerate before the recovery process is completed? How much data are the users willing to lose in the worst-case scenario?

WHY BACK UP THE DATA WAREHOUSE?

- A practical backup schedule for your data warehouse certainly depends on the conditions and circumstances in your organization. Generally, a practical approach includes the following elements:
  - Division of the data warehouse into active and static data
  - Establishing different schedules for active and static data
  - Having more frequent periodic backups for active data in addition to less frequent backups for static data
  - Inclusion of differential backups and log file backups as part the backup scheme
  - Synchronization of the backups with the daily incremental loads
  - Saving of the incremental load files to be included as part of recovery if applicable

(Ponniah, P, 2001)

BEST PRACTICES

Comparison of Oracle and MS SQL Server

(Navid, A, et.al., 2012)

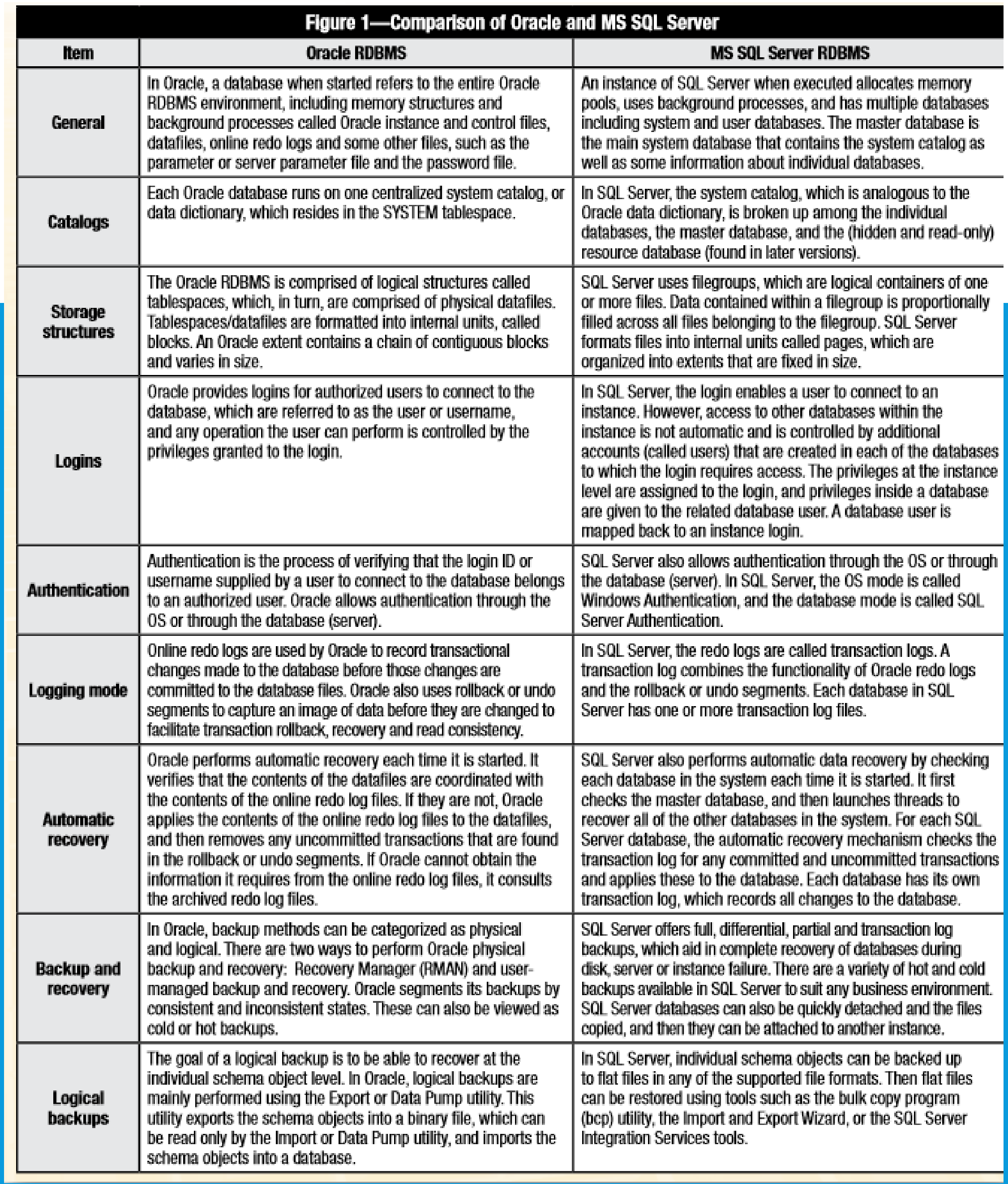


Figure 1—Comparison of Oracle and MS SQL Server

Item	Oracle RDBMS	MS SQL Server RDBMS
General	In Oracle, a database when started refers to the entire Oracle RDBMS environment, including memory structures and background processes called Oracle instance and control files, datafiles, online redo logs and some other files, such as the parameter or server parameter file and the password file.	An instance of SQL Server when executed allocates memory pools, uses background processes, and has multiple databases including system and user databases. The master database is the main system database that contains the system catalog as well as some information about individual databases.
Catalogs	Each Oracle database runs on one centralized system catalog, or data dictionary, which resides in the SYSTEM tablespace.	In SQL Server, the system catalog, which is analogous to the Oracle data dictionary, is broken up among the individual databases, the master database, and the (hidden and read-only) resource database (found in later versions).
Storage structures	The Oracle RDBMS is comprised of logical structures called tablespaces, which, in turn, are comprised of physical datafiles. Tablespaces/datafiles are formatted into internal units, called blocks. An Oracle extent contains a chain of contiguous blocks and varies in size.	SQL Server uses filegroups, which are logical containers of one or more files. Data contained within a filegroup is proportionally filled across all files belonging to the filegroup. SQL Server formats files into internal units called pages, which are organized into extents that are fixed in size.
Logins	Oracle provides logins for authorized users to connect to the database, which are referred to as the user or username, and any operation the user can perform is controlled by the privileges granted to the login.	In SQL Server, the login enables a user to connect to an instance. However, access to other databases within the instance is not automatic and is controlled by additional accounts (called users) that are created in each of the databases to which the login requires access. The privileges at the instance level are assigned to the login, and privileges inside a database are given to the related database user. A database user is mapped back to an instance login.
Authentication	Authentication is the process of verifying that the login ID or username supplied by a user to connect to the database belongs to an authorized user. Oracle allows authentication through the OS or through the database (server).	SQL Server also allows authentication through the OS or through the database (server). In SQL Server, the OS mode is called Windows Authentication, and the database mode is called SQL Server Authentication.
Logging mode	Online redo logs are used by Oracle to record transactional changes made to the database before those changes are committed to the database files. Oracle also uses rollback or undo segments to capture an image of data before they are changed to facilitate transaction rollback, recovery and read consistency.	In SQL Server, the redo logs are called transaction logs. A transaction log combines the functionality of Oracle redo logs and the rollback or undo segments. Each database in SQL Server has one or more transaction log files.
Automatic recovery	Oracle performs automatic recovery each time it is started. It verifies that the contents of the datafiles are coordinated with the contents of the online redo log files. If they are not, Oracle applies the contents of the online redo log files to the datafiles, and then removes any uncommitted transactions that are found in the rollback or undo segments. If Oracle cannot obtain the information it requires from the online redo log files, it consults the archived redo log files.	SQL Server also performs automatic data recovery by checking each database in the system each time it is started. It first checks the master database, and then launches threads to recover all of the other databases in the system. For each SQL Server database, the automatic recovery mechanism checks the transaction log for any committed and uncommitted transactions and applies these to the database. Each database has its own transaction log, which records all changes to the database.
Backup and recovery	In Oracle, backup methods can be categorized as physical and logical. There are two ways to perform Oracle physical backup and recovery: Recovery Manager (RMAN) and user-managed backup and recovery. Oracle segments its backups by consistent and inconsistent states. These can also be viewed as cold or hot backups.	SQL Server offers full, differential, partial and transaction log backups, which aid in complete recovery of databases during disk, server or instance failure. There are a variety of hot and cold backups available in SQL Server to suit any business environment. SQL Server databases can also be quickly detached and the files copied, and then they can be attached to another instance.
Logical backups	The goal of a logical backup is to be able to recover at the individual schema object level. In Oracle, logical backups are mainly performed using the Export or Data Pump utility. This utility exports the schema objects into a binary file, which can be read only by the Import or Data Pump utility, and imports the schema objects into a database.	In SQL Server, individual schema objects can be backed up to flat files in any of the supported file formats. Then flat files can be restored using tools such as the bulk copy program (bcp) utility, the Import and Export Wizard, or the SQL Server Integration Services tools.

BEST PRACTICES (CONT.)

- Comprehensive Backup plan - DBAs are responsible for making a comprehensive backup plan for databases for which they are accountable.
  - Decide what needs to be backed up.
    - OS Software
    - RDBMS Software
    - Application Software (where applicable)
  - Determine the appropriate backup type to use for your data.
    - Oracle databases:
      - Logical backups
      - Physical offline or cold backups
      - Physical online or hot backups
    - SQL Server databases:
      - Logical backups
      - Physical backups
  - Establish an appropriate backup schedule and window
  - Decide where to store backups

(Navid, A, et.al., 2012)

BEST PRACTICES (CONT.)

- Effective Backup Management - After making a solid backup plan and completing initial work, the DBA should properly manage backups, keeping the following points in mind:
  - Automating backups—For Oracle, either set backups through OEM or use an OS scheduling tool, and Spool output to a log file that can be reviewed for any errors. In SQL Server, use Maintenance Plans for scheduling backups.
  - Monitoring backups—Set up monitoring using appropriate tools so that the DBA gets an e-mail or alert through a pager or cell phone for any failed backups, which should be rerun as soon as possible.
  - Backup logs and catalogs—Review backup logs and backup catalog information periodically for any issues. Use RMAN reporting to show backup status. For Oracle, back up the RMAN catalog database by exporting all catalog schemas periodically as well as by doing an export backup of RMAN catalog schema at the end of each backup. For SQL Server, backup system databases, especially master and msdb.

(Navid, A, et.al., 2012)

BEST PRACTICES (CONT.)

- Database catalog maintenance—With Oracle databases, use “delete obsolete” to remove backups that are outside the organization’s retention policy. If obsolete backups are not deleted, the catalog will continue to grow and performance will become an issue. Cross-checking (cross-check backup) will check that the catalog/control file matches the physical backups.
- Validating backups—Validate and verify backups without doing actual restores.
- Setting up dependencies—When backing up to disk, archive these backups to tape as soon as backup to disk completes. Set up a process so that disk backups get transferred to tape without loss of time.

(Navid, A, et.al., 2012)

RECOVERY

- Have a clear recovery plan. List the various disaster scenarios and indicate how recovery will be done in each case.
- Test the recovery procedure carefully. Conduct regular recovery drills.
- Considering the conditions in your organization and the established recovery procedure, estimate an average downtime to be expected for recovery. Get a general agreement from the users about the downtime. Do not surprise the users when the first disaster strikes.
- In the case of each outage, determine how long it will take to recover. Keep the users properly and promptly informed.
- Generally, your backup strategy determines how recovery will be done. If you plan to include the possibility of recovering from the daily incremental load files, keep the backups of these files handy.
- If you have to go to the source systems to complete the recovery process, ensure that the sources will still be available.

https://www.ethany.com/

DATABASE MANAGEMENT PROBLEMS

COMMON DB MANAGEMENT CHALLENGES

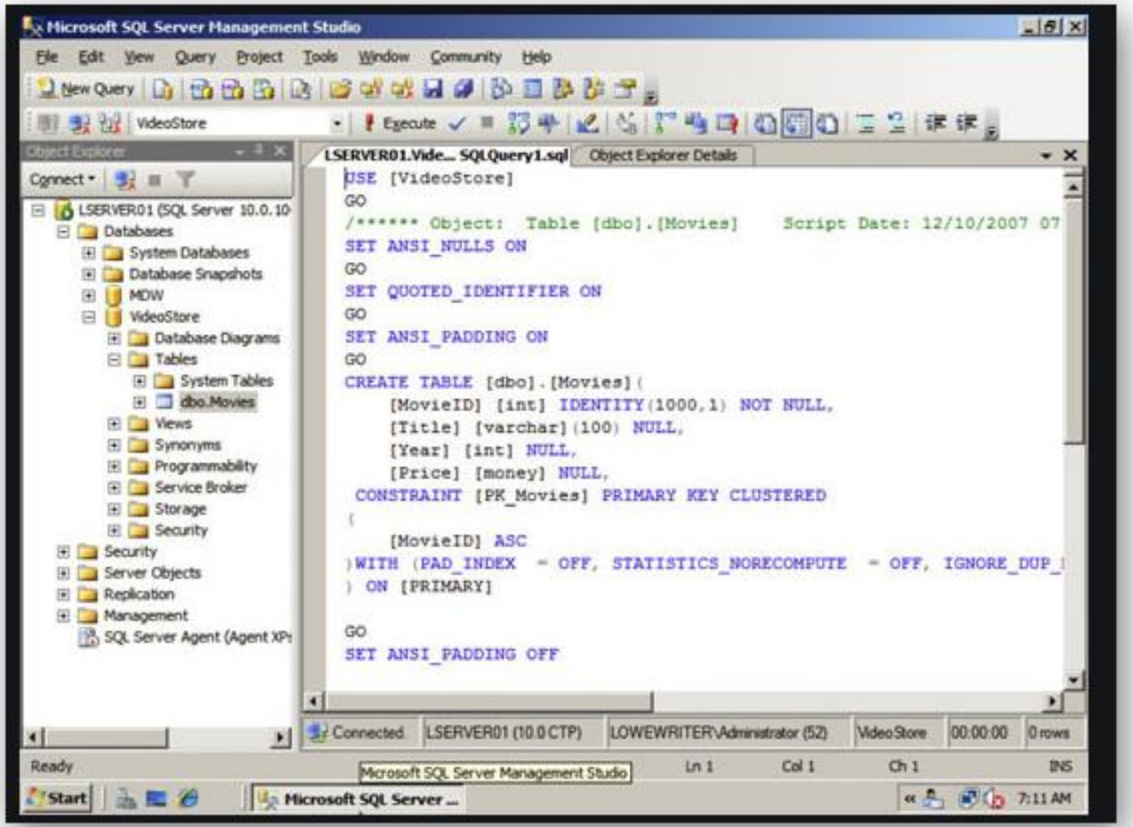
- In the spirit of making your database management as efficient as possible for your business, here are some of the most common issues and how to overcome them.
  - *Scalability*
  - *Cybersecurity* - No matter how great your security measures are, there is always room for improvement. Only allow access to employees who have an actual need to view the data. Always encrypt your information.
  - *Back up your database* - Cyber-attacks cost small businesses in our country between $84,000 and $148,000. And that's not even counting the loss of reputation or maybe even having to close down your business. Don't risk it. Always back up your data.
  - *Speed* - If every time you are trying to retrieve data, you get another wrinkle on your face, it's time to optimize your systems. Start by indexing properly and by not including too many joins on your SQL queries
  - Integration - If you provide omni-channel services, now you also have to integrate data from all of your many sources. You can do so with software specifically designed for this purpose.

(soaringeagle.biz, 2018)

CHALLENGES AND OPPORTUNITIES

- Most people in a business environment have heard the expression 'data is king', but that expression took on a whole new meaning in 2018.
- Here are the top five predictions from last year according to information-age.com:
  - IT will be forced to take responsibility for cloud data management and cut costs - 2017, 69% of organizations wrongfully believed data protection, data privacy and compliance were the responsibility of the cloud service provider, significantly exposing those organizations to a higher likelihood of a data breach and regulatory non-compliance.
  - The severity of data breaches will increase - 2016 saw 1,093 data breaches, a 40% increase from 2015. 2017 almost hit that mark by July.
  - One of the first companies to be fined under the General Data Protection Regulation (GDPR) will be outside of Europe - Veritas believes that one of the first companies to be fined under the GDPR will be outside of Europe.
  - Data management will get a major IQ boost from analytics - Businesses had a helping hand as advancements in analytics move the traditional archiving, backup and storage conversation far beyond 'add more capacity'.
  - Data will grow exponentially, but data storage will slow for the first time – In 2017, the annual data growth rate skyrocketed to 48.7%, filling valuable storage capacity at an incredible clip.

(information-age.com, *2018)*



```
USE [VideoStore]
GO
/****** Object:  Table [dbo].[Movies]    Script Date: 12/10/2007 07
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Movies](
    [MovieID] [int] IDENTITY(1000,1) NOT NULL,
    [Title] [varchar](100) NULL,
    [Year] [int] NULL,
    [Price] [money] NULL,
 CONSTRAINT [PK_Movies] PRIMARY KEY CLUSTERED
(
    [MovieID] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

https://www.dummies.com/

EXTRACT, TRANSFORM, LOAD (ETL)

WHAT IS ETL?

- This process is potentially one of the biggest tasks of building the data warehouse.
- It is very complex, time consuming, and consumes the majority of the data warehouse's implementation efforts, costs and resources.
- The general framework of this process is pictured below.

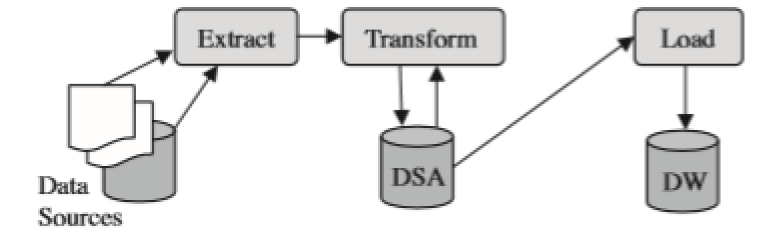


Figure 1 A general framework for ETL processes.

(El-Sappagh, et.al., 2011)

WHAT IS ETL? (CONT.)

- ETL Phases – many data warehouses also incorporate data from on-OLTP systems such as text files, legacy systems and spreadsheets. The ETL process is not a one-time event. The phases must be designed for ease of modification.
  - Extraction – This is the first step in the ETL scenario. This process consists of two phases, initial extraction and changed data extraction.
  - Transformation – The second step in any ETL scenario which tends to make some cleaning and conforming on the incoming data to gain accurate data which is correct, complete, consistent and unambiguous.
  - Loading – The final step in the ETL scenario which targets multidimensional structure for loading of the data.

MEASURES FOR ETL PROCESSES

- Due to their relevance, the quality of these processes should be formally assessed since the early stages of development, in order to avoid making bad decisions as a result of incorrect data.
- Designing ETL process is extremely complex, prone to failure and time consuming. It has been argued extensively in the literature that the ETL processes are costly and are one of the most important parts of developing a DW.
- In recent years several proposals have been defined for the conceptual modeling of ETL processes for DW.
- Munoz, Mazon & Trujillo (2009) describes a set of measures to evaluate the maintenance of the conceptual modeling of ETL processes, starting from the hypothesis that the low ease of maintenance of ETL models influence its global quality, and therefore, it may have a critical impact in the development of the DW

(Munoz, Mazon & Trujillo, 2009)

DESIGNING & DEVELOPING THE ETL SYSTEM

- So many challenges are buried in the data sources that developing the ETL application invariably takes more time than you expect.
- Before you begin the ETL system design for a dimensional model you should have completed the logical design, drafted your high-level architecture plan, and drafted the source-to-target mapping for all data elements.
- The ETL system design process is critical.
- Use a source code control system from the outset. Again, what you actuallyneedtododependsonthescopeofyourprojectandthetools available to you.

(Kimball, R., Ross, M., Thornthwaite, W., & Becker, B. (2008))

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop the ETL Plan**
  - ***Step 1 – Draw the High-Level Plan*** - Start the design process with a simple schematic of the pieces of the plan that you know: sources and targets. This one-pager is a good way to communicate some of the project's complexity to management.
  - ***Step 2 – Choose an ETL Tool*** - Most of the major database vendors offer an ETL tool, usually at additional licensing cost. There are also excellent ETL tools available from third party vendors. ETL tools serve a variety of functions. ETL tools read data from a range of sources, including flat files, ODBC, OLE DB, and native database drivers for most relational databases.
  - ***Step 3 – Develop Default Strategies*** - These activities include:
    - Extract from each major source system
    - Archive extracted data
    - Police data quality for dimensions and facts.
    - Manage changes to dimension attributes
    - Ensure the data warehouse and ETL system meet the requirements for system availability.
    - Design the data auditing subsystem.
    - Organize the ETL staging area.

(Kimball, R., Ross, M., Thornthwaite, W., & Becker, B. (2008))

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop the ETL Plan (cont.)**
  - ***Step 4 – Drill Down by Target Table –*** start drilling into the detailed transformations needed to populate each target table in the data warehouse. At the same time you're finalizing the source-to-target mappings, you must complete the data profiling.
    - Ensure Clean Hierarchies - It's particularly important to investigate whether hierarchical relationships in the dimension data are perfectly clean.
    - Develop Detailed Table Schematics - You should also be planning which tables to work on and in which order, as well as sequencing the transformations within each table.

(Kimball, R., Ross, M., Thornthwaite, W., & Becker, B. (2008))

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop the ETL Specification Document**
  - All the documents you have developed so far — the source-to-target mappings, data profiling reports, physical design decisions — should be rolled into the first sections of the ETL specification. Then document all the decisions we have discussed in this chapter, including:
    - Default strategy for extracting from each major source system.
    - Archival strategy.
    - Data quality tracking and metadata.
    - Default strategy for managing changes to dimension attributes.
    - System availability requirements and strategy.
    - Design of the data auditing subsystem.
    - Locations of staging areas.

(Kimball, R., Ross, M., Thornthwaite, W., & Becker, B. (2008))

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop a Sandbox Source System**
  - During the ETL development process, you'll need to investigate the source system data at great depth. If your source system is heavily loaded, and you don't already have some kind of reporting instance for operational queries, the DBAs may be willing to set up a static snapshot of the database for the ETL development team.
  - The simplest way to build the sandbox source is to restore the source databases from a consistent point-in-time backup. If your source data is very large, consider using your ETL tool to grab a random set of fact data.

(Kimball, R., Ross, M., Thornthwaite, W., & Becker, B. (2008))

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop One-Time Historic Load Processing**
  - We typically focus on developing the ETL process for the one-time load of historic data.
  - Occasionally you can make the same ETL code perform both the initial historic load and the ongoing incremental loads, but more often you build separate ETL processes for each.
  - In general, start building the ETL system with the simplest dimension table.
  -

(Kimball, R., Ross, M., Thornthwaite, W., & Becker, B. (2008))

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop the ETL Plan (cont.)**
  - ***Step 5 – Populate Dimension Tables with Historic Data –***
    - ***Populate Type 1 Dimension Tables*** - The easiest type of table to populate is a dimension table for which all attributes are managed as type 1 or update in place.
    - Dimension Transformations
      - ***Simple Data Transformations*** - The most common, and easiest, form of data transformation is data type conversion.
      - ***Combine from Separate Sources*** - Often dimensions are derived from several sources. Customer information may need to be merged from several lines of business and from outside sources.
      - ***Decode Production Codes*** – A common merging task in data preparation is looking up text equivalents for production codes.
      - ***Validate Many-to-One and One-to-One Relationships*** – Your most important dimensions will probably have one or more roll up paths, such as product to product model, subcategory and category.
      - ***Dimension Surrogate Key Assignment*** - Once you are confident that you have a version of your dimension table with one row for each true unique dimension value, you can assign the surrogate keys.
      - ***Load Type 2 Dimension Table History*** - Recall that each attribute in each dimension table is typically managed as type 1 (update in place) or type 2 (track history by adding new rows to the dimension table). Most dimension tables contain a mixture of type 1 and type 2 attributes.
      - ***Populate Date and Other Static Dimensions*** - Every data warehouse database should have a date dimension, usually at the granularity of one row for each day. The date dimension should span the history of the data, starting with the oldest fact transaction in the data warehouse.

(Kimball, R., Ross, M., Thornthwaite, W., & Becker, B. (2008))

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop the ETL Plan (cont.)**
  - ***Step 6 – Perform the Fact Table Historic Load***
    - ***Historic Fact Table Extracts*** - As you are identifying records that fall within the basic parameters of your extract, you need to make sure these records are useful for the data warehouse.
    - ***Audit Statistics*** - During the planning phase for the ETL system, you should have identified various measures of data quality. These are usually calculations, such as counts and sums, that you compare between the data warehouse and source systems to cross-check the integrity of the data.
    - ***Fact Table Transformations*** - In most projects, the fact data is relatively clean. The ETL system developer spends a lot of time improving the dimension table content, but the facts usually require fairly modest transformation.
    - ***Null Fact Table Values*** - All major database engines support a null value explicitly.
    - ***Improve Fact Table Content*** - all of the facts in the final fact table row must be expressed in the same grain. This means that there must be no facts representing totals for the year in a daily fact table or totals for some geography larger than the fact table grain.
    - ***Restructure Fact Data*** - Improving the content of the data isn't the only fact table transformation that takes place in the ETL system.
    - ***Pipeline the Dimension Surrogate Key Lookup*** - It is tremendously important that you maintain referential integrity (RI) between the fact table and dimension tables; you must never have a fact row that references a dimension member that doesn't exist

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop the ETL Plan (cont.)**
  - ***Step 6 – Perform the Fact Table Historic Load (cont.)***
    - ***Assign Audit Dimension Key*** – an enterprise class data warehouse should include an audit key on each fact row. The audit key points to an audit dimension that describes the characteristics of the load, including timings and measures of data quality.
    - ***Fact Table Loading*** - Your main concern when loading the fact table will be load performance. If you have followed our recommendations, you should have a perfectly clean set of data to load into the fact table.
    - ***Test, Test, and Test Again -*** You will no doubt feel like celebrating when the historic fact data finally finishes its loading process.
    - ***Develop Incremental ETL Processing*** - One of the biggest challenges with the incremental ETL process is identifying new, changed, and deleted rows. Hopefully your source systems can easily provide a stream with the rows that have been inserted or updated since the previous load.
  - ***Step 7 – Dimension Table Incremental Processing***
    - ***Dimension Table Extracts*** - In many cases, there is a customer master file or product master file that can serve as the single source for a dimension. In other cases, the raw data is a mixture of dimensional and fact data.
    - ***Identify New and Changed Dimension Rows*** - The DW/BI team may not be successful in pushing the responsibility for identifying new, updated, and deleted rows to the source system owners. In this case, the ETL process needs to perform an expensive comparison operation to identify new and changed rows.

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop the ETL Plan (cont.)**
  - ***Step 7 – Dimension Table Incremental Processing (cont.)***
    - ***Process Changes to Dimension Attributes*** - The ETL application must contain business rules to determine how to handle an attribute value that has changed from the value already stored in the data warehouse.
  - ***Step 8 – Fact Table Incremental Processing*** – Most data warehouse databases are too large to replace their central fact tables in a single load window. Instead, new and updated fact rows are processed incrementally.
    - ***Fact Table Extract and Data Quality Checkpoint*** – As soon as you extract the new and changed fact rows from the source system, write a copy of the untransformed data to the staging area. The staged data serves three purposes:
      - Archive for auditability
      - Provide a starting point after data quality verification.
      - Provide a starting point for restarting the process.
    - ***Fact Table Transformations and Surrogate Key Pipeline*** - The transformations for the incremental fact data are very closely related to the transformations that you built for the historic load. There are several methods for handling referential integrity violations:
      - Halt the load.
      - Throw away error rows.
      - Write error rows to a file or table for later analysis.
      - Fix error rows by creating a dummy row in the dimension, and returning its surrogate key to the pipeline.

(Kimball, R., Ross, M., Thornthwaite, W., & Becker, B. (2008))

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop the ETL Plan (cont.)**
  - **Fact Table Transformations and Surrogate Key Pipeline (cont.)**
    - Not recommended: "Fix" error rows by mapping them to a single unknown member in each dimension.
  - ***Late Arriving Facts and the Surrogate Key Pipeline*** - In most data warehouses, the incremental load process begins soon after midnight and processes all the transactions that occurred the previous day. However, there are scenarios where some facts arrive late.
  - ***Incremental Fact Table Load*** - The physical process of loading the incremental fact table rows is the same as we have previously discussed. As with the historic fact load, you should use your database's bulk load technology to perform the inserts into the fact table.
  - ***Speed Up the Load Cycle*** - Processing only changed increments is one way to speed up the data staging cycle.
    - More Frequent Loading - Although it is a huge leap to move from a monthly or weekly process to a nightly one, it is an effective way to shorten the load time/
    - Parallel Processing - One way to shorten the load time is to parallelize the ETL process. This can happen in two ways: multiple steps running in parallel and a single step running in parallel.
    - Parallel Structures - You can set up a three-way mirror or clustered configuration on two servers to maintain a continuous load data warehouse, with one server managing the loads and the second handling the queries.

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop the ETL Plan (cont.)**
  - ***Step 9 – Aggregate Table and OLAP Loads***
    - An aggregate table is logically easy to build. It's simply the results of a really big aggregate query, stored as a table. Not surprisingly, it's usually more difficult than that, but luckily, aggregate tables are often fairly easy to maintain.
  - ***Step 10 – ETL System Operation and Automation***.
    - ***Schedule Jobs*** - Scheduling jobs is usually straightforward. Your ETL tool should contain functionality to schedule a job to kickoff at a certain time.
    - ***Handle Predictable Exceptions and Errors Automatically*** - Although it's easy enough to launch jobs, it's a harder task to make sure they run to completion, gracefully handling data errors and exceptions. Elegant error handling is something that needs to be built into your ETL jobs from the outset.
    - ***Handle Unpredictable Errors Gracefully*** - Some errors are predictable, such as receiving an early arriving fact or a NULL value in a field that's supposed to be populated. For these errors, as we just described, you're generally able to design your ETL system to fix the data and continue processing. Other errors are unforeseen, and range from receiving data that's completely garbled to experiencing a power outage during processing.

(Kimball, R., Ross, M., Thornthwaite, W., & Becker, B. (2008))

DESIGNING & DEVELOPING THE ETL SYSTEM (CONT.)

- **Develop the ETL Plan (cont.)**
  - ***Step 10 – ETL System Operation and Automation**. (cont.)*
    - ***Maintain Database Objects*** - There are many database management tasks that fit smoothly into the ETL cycle. These tasks include database backups, periodic index maintenance, table partition management, and tablespace growth.
    - ***Develop and Test ETL Automation***- You should also take the time to evaluate data quality throughout the ETL process, and perhaps automatically halt the load if the data does not pass the quality benchmark.
  - ***Conclusion*** - Developing the ETL system is one of the greatest challenges of the DW/BI project. No matter how thorough your interviews and analyses are, you will uncover data quality problems when you start building the ETL system. Some of these data quality problems may be bad enough to force a redesign of the schema.

(Kimball, R., Ross, M., Thornthwaite, W., & Becker, B. (2008))

TIP TO IMPROVE ETL PERFORMANCE

- If you use an ETL tool that is able to execute SQL commands, the following tips may help you to implement fast ETL jobs or to improve the performance of long-running jobs.
  - ***1. Use Set-based Operations*** - The first tip should be obvious to every ETL developer: Set-based operations in SQL run usually faster than row-based executions of procedural languages. An INSERT/SELECT statement or a CREATE TABLE AS SELECT is a much better solution than a cursor loop in PL/SQL.
  - ***2. Avoid Nested Loops*** - A Nested Loops Join is a wonderful and efficient join method for small result sets in an OLTP application. If you want to select a few rows out of a big table and join these rows with another table, nested loops in combination with index scans are the fastest way to retrieve the required rows. But ETL jobs usually read many (or all) rows of an input table and join them with other tables. For this kind of queries, a Hash Join is definitely a better choice.

(danischnider.wordpress.com, 2017)

TIP TO IMPROVE ETL PERFORMANCE

- ***3. Drop Unnecessary Indexes***- Many Data Warehouses are "over-indexed". Unlike in OLTP systems, indexes are rarely used in a Data Warehouse. But for many database administrators and developers, creating indexes is still their favorite approach to "solve" any performance problem: "*The query runs slow, so let's create another index*". For ETL jobs, this usually doesn't help, it even increases the load times.
- ***4. Avoid Functions in WHERE Condition***- There can be several reasons for wrong estimations of the query optimizer. A common pitfall are expressions or function calls in WHERE conditions. If you use an SQL function like UPPER, SUBSTR, TO_CHAR, etc. in a filter condition, it is much harder for the optimizer to estimate the cardinality than if you use just unmodified columns.
- ***5. Take Care of OR in WHERE Condition*** - Although the optimizer is able to estimate the selectivity of multiple conditions combined with an OR, it is sometimes "confused" with complex WHERE conditions. In many cases, conditions with OR can be replaced by other expressions.

(danischnider.wordpress.com, 2017)

TIP TO IMPROVE ETL PERFORMANCE

- ***6. Reduce Data as Early as Possible***- The earlier the amount of data can be reduced; the less work has Oracle to do to read and join the relevant rows of each table. This is the most important rule for performance tuning in OLTP applications with selective queries.
- ***7. Use WITH to Split Complex Queries***- It allows to split a complex job into separate smaller parts that can be executed much faster. Instead of loading an intermediate table that is used as input for the next ETL job, these steps can be combined in one SQL statement, using a WITH clause.
- ***8. Run Statements in Parallel***- For loads of large data sets, it is highly recommended to run the jobs in parallel. Oracle is able to execute SQL statements with multiple parallel processes. Parallel execution is enabled for queries and DDL commands by default.
- ***9. Perform Direct-Path INSERT*** - Direct-Path INSERT is an efficient way to load large data sets into a target table. In contrast to Conventional INSERT statements, this method is much faster because new data is always appended in additional table blocks at the end of the table.
- ***10. Gather Statistics after Loading each Table*** - The last step of every ETL job should always gathering statistics on the target table. This is especially important for tables that are reloaded from scratch.

(danischnider.wordpress.com, 2017)

ETL TOOLS - DO YOU NEED THEM?

- ETL processes the heterogeneous data and makes it homogeneous which in turn makes it seamless for data scientists and data analysts to analyze the data and derive business intelligence from it.
- ETL tools contain graphical interfaces which speed up the process of mapping tables and columns between the source and target databases.
- ETL tools can collect, read and migrate data from multiple data structures and across different platforms, like a mainframe, server, etc. ETL technology can also identify “”**delta”” changes as they occur.**
- **ETL tools include ready to use o**perations like filtering, reformatting, sorting, joining, merging, and aggregation

(springpeople.com, 2018)

ETL TOOLS - DO YOU NEED THEM?

- Key advantages of ETL Tools:
  - ***Ease of Use*** - The tool itself specifies data sources and the rules for extracting and processing data, and then, it implements the process and loads that data.
  - ***Visual Flow*** - ETL tools are based on Graphical User Interface (GUI) and offer a visual flow of the system's logic.
  - ***Operational Resilience*** - ETL tools possess built-in error-handling functionality which helps data engineers to build on the features of an ETL tool to develop a resilient and well-instrumented ETL system.
  - ***Good for Complex Data Management Situations*** - ETL tools offer better utility for moving large volumes of data and transferring them in batches.
  - ***Advanced Data Profiling and Cleansing*** - ETL tools bring in a richer set of cleansing functions as compared to the ones available in SQL.
  - ***Enhanced Business Intelligence*** - ETL tools improve the access to data as it simplifies the process of extracting, transforming and loading.
  - ***High Return on Investment (ROI)*** - The use of ETL tools saves cost, thereby enabling businesses to generate higher revenue.
  - ***Performance*** - simplifies the process of building a high-quality data warehousing system.

(springpeople.com, 2018)

ETL TOOLS SURVEY

- The working of the ETL tools is based on ETL (Extract, transform, load) process.
  - ***A. ETL tools comparison criteria***
    - mode of connectivity or adapter support
    - mode of data transformation and delivery support
    - data modeling and metadata support
    - architecture design, development and data governance support
    - debugging facility and execution or runtime platform support
    - additional services and requirements for vendors
    - customers usability support
    - cost of hardware or software , installation, OS, Support
    - functionality, flexibility and performance support
    - infrastructure support

(Mali, N, Bojewar, S, 2015)

ETL TOOLS SURVEY (CONT.)

- ***B. Details of ETL Tools -*** Most popular used commercial and freeware (open sources) ETL Tools are listed below.
  - *IBM Infosphere DataStage*
  - *Informatica Power Center*
  - *Oracle Warehouse Builder (OWB)*
  - *Oracle Data Integrator (ODI)*
  - *SAS ETL Studio*
  - *Business Objects Data Integrator (BODI)*
  - *SQL Server Integration Services (SSIS)*

The table on the next slide details the advantages and disadvantages of each of these tools.

(Mali, N, Bojewar, S, 2015)

ETL TOOLS SURVEY (CONT.)

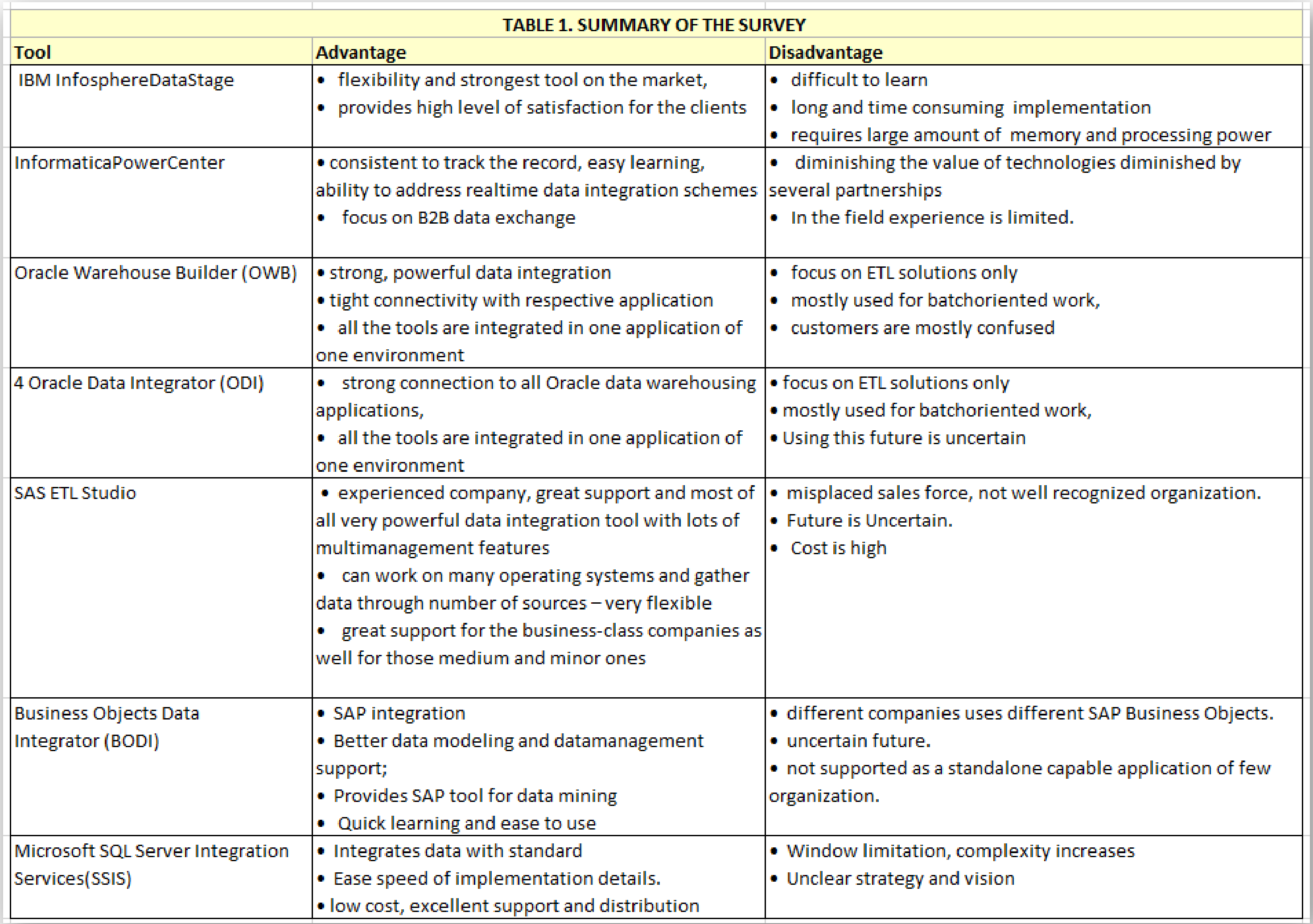


TABLE 1. SUMMARY OF THE SURVEY

Tool	Advantage	Disadvantage
IBM InfosphereDataStage	• flexibility and strongest tool on the market, • provides high level of satisfaction for the clients	• difficult to learn • long and time consuming implementation • requires large amount of memory and processing power
InformaticaPowerCenter	• consistent to track the record, easy learning, ability to address realtime data integration schemes • focus on B2B data exchange	• diminishing the value of technologies diminished by several partnerships • In the field experience is limited.
Oracle Warehouse Builder (OWB)	• strong, powerful data integration • tight connectivity with respective application • all the tools are integrated in one application of one environment	• focus on ETL solutions only • mostly used for batchoriented work, • customers are mostly confused
4 Oracle Data Integrator (ODI)	• strong connection to all Oracle data warehousing applications, • all the tools are integrated in one application of one environment	• focus on ETL solutions only • mostly used for batchoriented work, • Using this future is uncertain
SAS ETL Studio	• experienced company, great support and most of all very powerful data integration tool with lots of multimanagement features • can work on many operating systems and gather data through number of sources – very flexible • great support for the business-class companies as well for those medium and minor ones	• misplaced sales force, not well recognized organization. • Future is Uncertain. • Cost is high
Business Objects Data Integrator (BODI)	• SAP integration • Better data modeling and datamanagement support; • Provides SAP tool for data mining • Quick learning and ease to use	• different companies uses different SAP Business Objects. • uncertain future. • not supported as a standalone capable application of few organization.
Microsoft SQL Server Integration Services(SSIS)	• Integrates data with standard • Ease speed of implementation details. • low cost, excellent support and distribution	• Window limitation, complexity increases • Unclear strategy and vision

(Mali, N, Bojewar, S, 2015)

UNIT 5 – INDIVIDUAL PROJECT

Week 5: Part 1: Final Deliverable for Key Assignment

- Develop structured query language (SQL) statement scripts that will implement the data warehouse.
- This must be done in Word, and it must be at least 2–3 pages.
- Turn in the overall Key Assignment, incorporating all instructor feedback and the feedback from the students from Week 4, along with the section that is due this week.
- Name the document "yourname_CS683_IP5.doc."

Worked Example

- Please refer to the Worked Example below for an example of this assignment based on the Problem-Based Learning Scenario. The worked example is not intended to be a complete example of the assignment but will illustrate the basic concepts required for completion of the assignment and can be used as a general guideline for your own project. Your assignment submission should be more detailed and specific and should reflect your own approach to the assignment rather than just following the same outline provided in the worked example.
- Note that the worked example includes material from previous worked examples. The new material will be found under the Week 4 sections of the Table of Contents.
- The worked example is provided here to help with this assignment.

UNIT 5 – INDIVIDUAL PROJECT

Week 5: Part 2: Argumentation Conclusion

- Envision that you now have to present your solution to your client. You need to make a formal presentation to a group of stakeholders, and you need to be prepared to answer their questions. As a final step in the project, you will create a 5-slide PowerPoint presentation plus title and reference slides in which you provide the following:
- In the Notes pages of each slide, and based on the context of the information on that slide, provide a narrative of 2–3 paragraphs of your rationale for the data warehouse (DW) design document that you are recommending.
- Discuss the company's strengths, weaknesses, opportunities and threats (**SWOT** analysis). How will implementing the DW support the company's SWOT?
- On each slide, identify the evidence that you have to support the arguments.
- All sources that are used to support your argument and solution should be cited both in-text (on the slide or in Notes section) and on a References slide in APA format.
- Name the document "yourname_ CS683_IP5.pptx."
- **Please submit your assignment.**
- **For assistance with your assignment, please use your text, Web resources, and all course materials.**

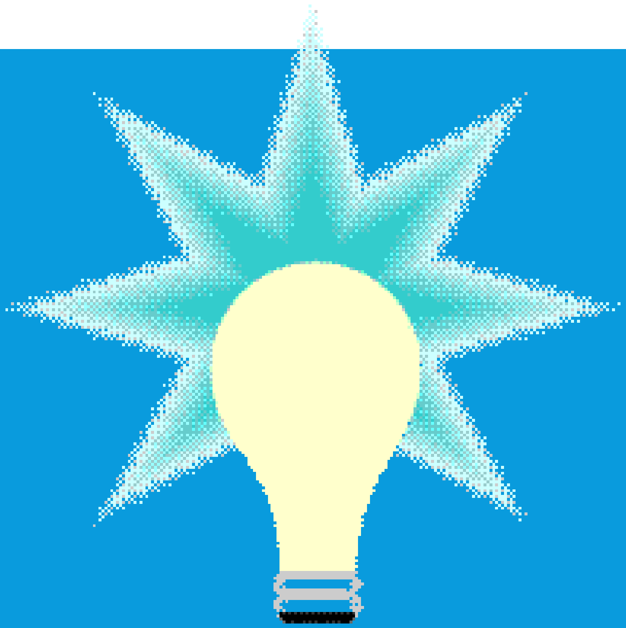
CONTACT INFORMATION

- My e-mail address- JConklin@coloradotech.edu
- Office Hours - Wednesday 6:00 P.M. – 7:00 P.M. CST

  Saturday 11:00 A.M. – 12:00 P.M. CST
- Live Chats - Thursday 7:00 P.M. – 8:00 P.M. CST

* Please note that only one live chat session per week is required for this course. However, optional live chat sessions may be held sporadically throughout the course.

QUESTIONS / COMMENTS



(Ideas/Think Web Graphics, 2019).

REFERENCES

Akhtar, A. N., J. Buchholtz, et al. (2012). "Database Backup and Recovery Best Practices." ISACA JOURNAL **1**.

Colorado Technical University. (2019). Instructor's guide for CS 683-1903B-01. Retrieved from Colorado Technical University Online, Virtual Campus, Course Overview: https://campus.ctuonline.edu

Connolly, T. and C. Begg (2015). Database systems; a practical approach to design, implementation, and management, 6th ed. Portland, Pearson Education

Cokins, G. (2009). *Performance management : Integrating strategy execution, methodologies, risk, and analytics*. Retrieved from https://ebookcentral-proquest-com.library.capella.edu

Dunham, J. (1999). A guide to large-database tuning. *UNIX Review's Performance Computing, 17*(5), 35-41. Retrieved from https://search.proquest.com/docview/237187882?accountid=36783

Golfarelli, M., Rizzi, Stefano (2009). Data Warehouse Design: Modern Principles and Methodologies McGraw Hill.

REFERENCES

Hiremath, D. S. and S. B. Kishor (2016). "Distributed Database Problem areas and Approaches." IOSR Journal of Computer Engineering

Ideas/Think Web Graphics. (2019). In *Desktop Publishing*. Retrieved from: http://desktoppub.about.com/od/freeclipart/l/blidea1.htm

Koch, R. (n.d.). "SQL Database Performance Tuning for Developers." Retrieved 09/09/2019, from https://www.toptal.com/sql-server/sql-database-tuning-for-developers.

Kimball, R., M. Ross, et al. (2008). Data Warehouse Lifecycle Toolkit. Hoboken, UNITED STATES, John Wiley & Sons, Incorporated.

Kraynak, J. (2017). Cloud Data Warehouse for Dummies. Hoboken, NJ, John Wiley & Sons, Inc.

Laberge, R. (2011). The Data Warehouse Mentor: Practical Data Warehouse and Business Intelligence Insights McGraw Hill.

Moran, B. (2007, 03). The 3 principles of database tuning. *SQL Server Magazine*, *9*, 5. Retrieved from https://search.proquest.com/docview/214854373?accountid=36783

REFERENCES

Ponniah, P. (2001). DATA WAREHOUSING FUNDAMENTALS: A Comprehensive Guide for IT Professionals, Wiley Interscience.

Pons, A. P. (2003). Database tuning and its role in information technology education. *Journal of Information Systems Education, 14*(4), 381-387. Retrieved from https://search.proquest.com/docview/200109300?accountid=36783

Rayman, J., R. Stackowiak, et al. (2007). Oracle Data Warehousing and Business Intelligence Solutions. Wiley, John Wiley & Sons.

Techtarget (n.d.). Chapter 12 - Backup and Recovery. Techtarget.com.

Viaene, S., & Willems, J. (2007). Corporate performance management: Beyond dashboards and Scorecards1. *Journal of Performance Management, 20*(1), 13. Retrieved from https://search.proquest.com/docview/214034904?accountid=36783

XTIVIA (2017). "Backup Strategies For Sql Server Data Warehouses." Retrieved 09/09/2019, from https://www.xtivia.com/backup-strategies-sql-server-data-warehouses/.