International Journal of Research in Social Sciences
Vol. 9 Issue 4, April 2019,
ISSN: 2249-2496 Impact Factor: 7.081
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

AN EXPLORATORY STUDY ON BUSINESS DATA INTEGRITY FOR EFFECTIVE BUSINESS; A TECHNO BUSINESS LEADERSHIP PERSPECTIVE

ProfDr.C.Karthikeyan,[*]

AsstProfKrishna[**]

Ms Anna Benjamin[**]

Abstract: Data integrity is the maintenance of, and the assurance of the accuracy and consistency of, data over its entire life-cycle, and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data. The term *data integrity* is broad in scope and may have widely different meanings depending on the specific context – even under the same general umbrella of computing. Data integrity is the opposite of data corruption, which is a form of data loss. The overall intent of any data integrity technique is the same: ensure data is recorded exactly as intended (such as a database correctly rejecting mutually exclusive possibilities,) and upon later retrieval, ensure the data is the same as it was when it was originally recorded. In short, data integrity aims to prevent unintentional changes to information. Data integrity is not to be confused with data security, the discipline of protecting data from unauthorized parties. Any unintended changes to data as the result of a

[*] Director and Professor, Management Studies, T.John College, Bangalore, Affiliated to Bangalore University, Bangalore, Karnataka, and Accredited by NAAC "A", and Approved by AICTE, New Delhi

[**] Asst Professor, Management Studies, T.John College, Bangalore, Affiliated to Bangalore University,Bangalore, Karnataka, and Accredited by NAAC "A", and Approved by AICTE, New Delhi

[**] Asst Prof, T.John Institute of Management Science, Bangalore, Affiliated to Bangalore University,Approved by AICTE, New Delhi.

storage, retrieval or processing operation, including malicious intent, unexpected hardware failure, and human error, is failure of data integrity. If the changes are the result of unauthorized access, it may also be a failure of data security. Depending on the data involved this could manifest itself as benign as a single pixel in an image appearing a different color than was originally recorded, to the loss of vacation pictures or a business-critical database, to even catastrophic loss of human life in a life-critical system. An example of a data-integrity mechanism is the parent-and-child relationship of related records. If a parent record owns one or more related child records all of the referential integrity processes are handled by the database itself, which automatically ensures the accuracy and integrity of the data so that no child record can exist without a parent (also called being orphaned) and that no parent loses their child records. It also ensures that no parent record can be deleted while the parent record owns any child records. All of this is handled at the database level and does not require coding integrity checks into each applications.

Keywords: Data; Integrity; Storage; Warehouse; Protecting data; Integrity Concerns, Data Security.

Objectives:

(i) To learn about a broad overview of some of the different types and concerns of data integrity.
(ii) To understand the overall intent of any Data Integrity and Kinds of Data Integrity
(iii) To evaluate the process of data loss and how to have countermeasures

Methodology: Literature review from previous studies

Secondary Data; Secondary Data from reliable sources and sites

Review of Related Literature;

Alexander and Tate (1999) described six evaluation criteria - authority, accuracy, objectivity, currency, coverage/intended audience, and interaction/transaction features for web data.

Katerattanakul and Siau (1999) developed four categories for the information quality of an individual website and a questionnaire to test the importance of each of these newly developed information quality categories and how web users determine the information quality of individual sites.

Gauch (2000) proposed six quality metrics, including currency, availability, information-to-noise ratio, authority, popularity, and cohesiveness, to investigate.

Shanks and Corbitt (1999) studied data quality and set up an emiotic-based framework for data quality with 4 levels and a total of 11 quality dimensions.

Knight and Burn (2005) summarized the most common dimensions and the frequency with which they are included in the different data quality/information quality frameworks. Then they presented the IQIP (Identify, Quantify, Implement, and Perfect) model as an approach to managing the choice and implementation of quality related algorithms of an internet crawling search engine.

U.S. National Institute of Statistical Sciences (NISS) (2001), the principles of data quality are: 1. data are a product, with customers, to whom they have both cost and value; 2. as a product, data have quality, resulting from the process by which data are generated, data quality depends on multiple factors, including (at least) the purpose for which the data are used, the user, the time, etc.

S. Chakravarthy and D. Lomet, introduced, some changed are needed for Help counter, some indexes become obsolete and others need to be created, some aggregates are no longer referenced and others need to be evaluated, and the limits on parallel processing must be assessed and adjusted to fit the current demand. These and other tuning tasks should be carried out periodically Help and support to keep data warehouse performance smooth and constant. Data warehousing is becoming an increasingly important technology for information integration and data analysis

Fred R. McFadden in their paper entitled "Data Warehouse for EIS: Some Issues and Impacts" at 29th Annual Hawaii International Conference on System Sciences - 1996.This paper proposes a research study and includes a data model that relates user satisfaction and development

characteristic to organizational factors, the warehouse infrastructure, and management support. Many research opportunities exist in data warehouse and in **EIS and DSS**.

Hurley and Harris describe by survey in KPMG management consulting and the Nolan Norton institute. The main goal of this survey was to complete a logical understanding regarding data warehousing enterprise. The conclusion from the survey was that data warehouse technology heavily raises financial and business returns in the adopters. They found the project team ability, Technical infrastructure, Technical architecture, Good vendor capability, clear objectives 54 issues for successful data warehousing enterprise.

Joshi and Curtis in the paper entitled "Issues in building a successful data warehouse" in the executive's journal, 1999, explore some key factor that any corporate should think about before planning to acclimatize data warehouse technology. Their most factors were Data warehouse development, architecture, User Access issues and Data issues. Based on reviewing the related research papers, they developed important factors that the organization must consider to have a successful planning of a data warehouse project

D. Theodoratos, M. Bouzeghoub in their paper entitled "Data Currency Quality Factors in Data Warehouse Design" at International Workshop on Design and Management of Data Warehouses (DMDW'99). Heidelberg, Germany, described a DW system architecture that supports the performance quality goal and satisfies the data consistency and completeness quality goals. In this framework it has addressed the problem of checking whether a view selection satisfies the given constraints.

According to HAC's algorithm which is based on data set and in this it is show the result lead to a hierarchical tree structure. The main thing in which, I observe that as the hight of tree expands, best technique is binary search. A growing cell structure algorithm is initial run on the data set. Then the divide of the growing cell structure result lead to a hierarchical tree structure. Similar to the HAC, the hierarchy of Tree growing cell structure is a binary tree.

Dopazo et al. applied a self-organizing tree algorithm for clustering gene expression patterns. The alternative approach is to employ nonhierarchical clustering, such as Tamayo et al. used a self-organizing map to analyze expression patterns of 6,000 human genes.

Ben-Dor and Yakhini proposed a graph-theoretic algorithm for the extraction of high probability gene structures from gene expression data.

Smet et al. proposed a heuristic two-step adaptive quality based algorithm, and Yeung, K. Y. et al. have proposed a clustering algorithm based on probability models. All of these algorithms tried to find groups of genes that exhibit highly similar forms of gene expression.

According to the study by Watson the most common factors for the failure of a DW success include weak sponsorship and management support, insufficient funding, inadequate user involvement, and organizational politics. 61 There are several strategies to improve query response time in the data warehouse context: B+tree indexes are the most common supported structures in RDBMS, but it is a well-known fact that B+tree structures have inconveniences when the cardinality of the attribute is small. Another class of index structures, the bitmap indexes, try to overcome the problem by using a bit structure to indicate the rows containing specific values of the indexed attribute.

Sudesh M. Duggal, Inna Pylyayeva in their research paper entitled "Data Warehouse – Strategic Advantage" at IACIS 2001 explain the Data warehouse functions as a conduit between the users and corporate data resources creating a flexible staging area for decision support applications. It provides a specialized decision support database that manages the flow of information from exiting 65 corporate database and external sources to end user to support strategic decision-making. To gain maximum value, an organization that relies on information should consider the development of a warehouse in conjunction with other decision support tools, such as Online Analytical Processing Systems, data mining, and Executive Information Systems. The warehouse should serve as the foundation for the vital process of delivering key information to analysts and decision-makers, which are the key users of the information.

Wixom et. all ,presented an explanation of why some organizations realize more exceptional benefits than others after data warehouse installation. The authors started by giving a basic background about a data warehouse. Then they went through the obtainable benefits gained from data warehouse installation in general by the adopters. Three case studies of data warehousing initiatives, a large manufacturing company, an internal revenue service and a financial services company, were discussed within the context of the suggested framework. The results from the case studies highlighted the benefits achieved by the three organizations. The researchers noticed that some of them considered more significant payoffs than the other adopters. The researchers built an argument about the main issues behind the success in the three cases. The argument led to the following issues: Business need, Champion, Top management support, user involvement, training matters, Technical issues (adequate tools), Accurate definition of the project's objectives, growth and upgradeability, Organizational politics, skilful team.

Mukherjee and D'Souza in their paper entitled "Think phased implementation for successful data warehousing, information systems management", in 2003 presented a framework which might help the data ware house people to visualize how strategies can be included in each stage of data warehouse implementation process.

Neil Warner in their paper entitle "Information Superiority through Data Warehousing" says Data warehousing has evolved through repeated attempts on the part of different researchers and organizations to give their organizations flexible, effective and efficient means of getting at the sets of data that have come to represent one of the organizations most critical and valuable property. Data warehousing is a tools that has grown out of the integration of a number of dissimilar tools and experiences over the last two decades.

Jeff Theobald in "Strategies for Testing Data Warehouse Applications" publishes at Information Management Magazine, June 2007. This article provides practical recommendations for testing extract transform and load (ETL) applications based on years of experience testing data warehouses in the financial services and consumer retailing areas.

Mark I Hwang; Hongjiang Xu in their paper "A Structural Model of Data Warehousing Success" in 2008 this paper contribute to the understanding 71 of data ware house success by showing the interrelationship among a set of variable. He state operational factor, technical factor, schedule factor, economic factor, system quality, information quality could vary as the time frame or the environment change.

Vivekanand Gopalkrishnan in research paper entitled "On Scheduling Data Loading and View Maintenance in Soft Real-time Data Warehouses" at 15th International Conference on Management of Data COMAD 2009, Mysore, India, December 9-12, 2009 expressed that an efficient technique aimed at updating data and performing view maintenance for real-time data warehouses while still enforcing these two timing requirements for the OLAP transactions.

Victor González Castro in their research thesis entitled "The Use of Alternative Data Models in Data Warehousing Environments" at Heriot-Watt University, Edinburgh, United Kingdom in may 2009 describe the Data Warehouses are increasing their data volume at an accelerated rate; high disk space consumption; slow query response time and complex database administration are common problems in these environments. The lack of a proper data model and an adequate architecture specifically Help and support targeted towards these environments are the root causes of these problems. Inefficient management of stored data includes duplicate values at column level and poor management of data scarcity which derives from a low data density, and affects the final size of Data Warehouses.

Kimball, R in book entitled "The Data Warehousing Toolkit say that If the internal success factor is weak, then the likelihood is that there is an inefficient data model, not enough summarized data or the users are not well trained. If the external success factor is weak, then more resources should be put into addressing more of the needs. The success factor is therefore more than just an isolated statistic that can be reported to management, but is in itself a tool that can be used to make the data warehouse more effective for the enterprise.

According to Lauren, the absence or the lack of the following criteria is the common causes of data warehouse failures: pre- launch clear objectives or metrics, insider presence on data warehouse projects team, user demand for sophisticated data analysis, and initial involvement of

business managers. In addition, other problems can arise: many major systems projects underway simultaneously, the CEO sets budget and deadlines before project team is on the board, and source data availability unconfirmed at the outset. Another potential Pitfall involves escalating user demand and unrealistic expectations.

According to Haisten one of the major causes of failures is that the database product was driving the project, not being driven by it. Others are the lack or the absence of close and effective link between business process analysis and data normalization, multi- national views available with the context, serious normalization, architecture for an integrated repository of brows able metadata, study of scale and capacity issues, and coordination with legacy application portfolio management.

According to Gupta, errors that data warehouse can contain involve data of four categories: incomplete, incorrect, incomprehensible, and inconsistent. Incomplete errors consist of missing records or missing fields. Incorrect data has wrong codes, calculations, aggregations, information entered into the system, pairing of codes, or duplicate records.

According to Jorge Bernardino the distributed data warehousing affords several advantages to businesses, including suppliers and distributors. First and foremost, it is a cost-effective solution. By initially investing in a local Windows NT server, a department could build a small individual data store that could later be connected to additional ones, forming a "distributed" data warehouse. Instead of undertaking the painstaking task of integrating them all into one central store, the two or more could be virtually combined using network hardware and software.

Gerth Brodal in their research entitled "Cache Oblivious Search Trees via Binary Trees of Small Height" it is apparent that the effects of the memory hierarchy in today's computers play a dominant role for the running time of tree search algorithms, already for sizes of trees well within main memory. It also appears that in the area of search trees, the nice theoretical properties of cache obliviousness seem to carry over into practice: in our experiments, the van layout was competitive with cache aware structures, was better than structures not optimized for

memory access for all but the smallest n, and behaved robustly over several levels of the memory hierarchy.

Elena Demidova use query disambiguation techniques to process keyword queries automatically extracted from documents. he suggest a system that enables context-aware auto completion for SQL by taking into account previous query work-loads which are, in turn, represented as workload DAG.

Azefack present an automatic, dynamic index selection method for data warehouses that is based on incremental frequent item set mining from a given query workload . The main advantage of this approach is that it helps update the set of selected indexes when workload evolves instead of recreating it from scratch. Preliminary experimental results illustrate the efficiency of this approach, both in terms of performance enhancement and overhead.

Eisen et al. applied a variant of the hierarchical average-linkage clustering algorithm to identify groups of co-regulated genome-wide expression patterns. Loewenstein et al. applied agglomerative clustering to protein sequences that automatically builds a comprehensive evolutionary driven hierarchy of proteins from sequence alone.

Frey et al. applied an affinity propagation clustering technique to detect putative exons comprising genes from mouse chromosomes. While they claim lower computational cost in comparison to other algorithms, they do not include 80 the cost of pairwise similarity computations. Since this is the most expensive stage in large scale problems, the claimed advantage is exaggerated. Our focus is on reducing the computational complexity arising from this cost.

Frey et al. applied an affinity propagation clustering technique to detect putative exons comprising genes from mouse chromosomes. While they claim lower computational cost in comparison to other algorithms, they do not include the cost of pairwise similarity computations. Since this is the most expensive stage in large scale problems, the claimed advantage is exaggerated. Our focus is on reducing the computational complexity arising from this cost .

El-Sonbaty et al. proposed an on-line hierarchical clustering algorithm based on the single-linkage method that finds at each step the nearest k patterns with the arrival of a new pattern and updates the nearest seen k patterns so far. Finally the patterns and the nearest k patterns are sorted to construct the hierarchical dendrogram. While they claim their method is sequential, at the arrival of each data item they compute similarity to all the data seen previously.Thus there is little computational saving in their method, and it is equivalent to re-training a new model at the arrival of new data. Contrast that with a truly sequential algorithm, where it is the model that is adapted, similar in fashion to the Kalman filter.

According **to Bassam Farran, Amirthalingam Ramanan, and Mahesan Niranjan** in this paper they present an algorithm for on-line hierarchical clustering. The approach depends on the construction of an initial clustering stage using a random subset of the data. This establishes a scale structure of the input space. Subsequent data can be processed sequentially and the tree adapted constructively. They have shown that on small bioinformatics problems such an approach does not degrade the quality of the clusters obtained while saving 81 computational cost.

Dr.N.Rajalingam analyzes the performance of agglomerative and divisive algorithm for various data types. From this work it is found that the divisive algorithm works as twice as fast as that of agglomerative algorithm. It is also found that the time needed for string data type is high when compared to the other. The next observation is, in the case of binary field, the time needed to execute a two combined binary field is slightly larger or less equal to the time needed for single binary field. It is also found that the running time get increased on an average of 6 times when the number of records get bled.

Widyantoro et al. present the agglomerative incremental hierarchical clustering (IHC) algorithm that also utilizes a restructuring process while preserving homogeneity of the clusters and monotonicity of the cluster hierarchy. 83 New points are added in a bottom-up fashion to the clustering hierarchy, which is maintained using a restructuring process performed only on the

regions affected by the addition of new points. The restructuring process repairs a cluster whose homogeneity has been degraded by eliminating lower and higher dense regions.

Charikar et al. introduce new deterministic and randomized incremental clustering algorithms while trying to minimize the maximum diameters of the clusters. The diameter of a cluster is its maximum distance among its points and is used in the restructuring process of the clusters. When a new point arrives, it is either assigned to one of the current clusters or it initializes its own cluster while two existing clusters are combined into one. As indicated in the introduction, the clustering of a data stream is to some degree related to the incremental clustering of dynamically changing databases, although data streams impose different requirements on the mining process and the clustering algorithms. The purpose of both methodologies is to provide the user with "up-todate" clusters very quickly from dynamic data sets. However, when mining a dynamically changing database, the clustering algorithm has access to all points in the data base and not necessarily only the most recently inserted points, and the algorithm is not restricted to a sequential access to these new points. The clustering algorithm has slightly different requirements when mining data streams as explained next.

D. Barbara outlines the main requirements for clustering data 84 streams. These requirements consist of 1) compact representation of the points that can be maintained in main memory even as lots of new points arrive, 2) fast incremental processing of new data points, and 3) clear and fast identification of outliers. The cluster assignment of new points should use a function that does not depend on comparison to past points and yet performs well. G

anti et al. also examine mining of data streams. A block evolution model is introduced where a data set is updated periodically through insertions and deletions. In this model, the data set consists of conceptually infinite sequence of data blocks D1, D2. that arrive at times 1, 2, ... where each block has a set of records. Some applications require mining all of the data encountered thus far (unrestricted window scenario), while others require mining only the most recent part (restricted window scenario) and updating the models accordingly.

Aggarwal et al. use the data summarization method BIRCH in the context of mining data streams. BIRCH compresses a dataset into so-called clustering features (CFs). A clustering

feature CF = (n, LS, SS), where LS is the linear sum of the n points compressed by the CF and SS is their square sum. The scheme presented by

Aggarwal et al. for clustering data streams combines online micro clustering with offline macro clustering . During the micro clustering phase, a temporal version of the clustering features of BIRCH and pyramidal time frames are used to store on disk micro clusters from different time snapshots in a pyramidal pattern. Once the user specifies the window for mining the macro clusters, the micro clusters for that window are extracted using the additively property of the clustering features and the macro clusters are uncovered using a modified k-means algorithm that regards the micro clusters as points.

Zobel and Moffat provided a set of standard similarity measures that include combining and weighting functions. In this research, M-trees are not used for indexing spatial data objects, but they are used for clustering data objects which are then reconcile. Like B-trees and R-trees, M-trees grow in a bottom-up fashion and are balance. The M-trees differ from B-trees in the fact 86 that a node in M-trees is split when it is 100% full whereas a node in B-trees is split when its capacity reaches a certain threshold (usually when it is approximately 70%full).

Tasawar et al., proposed a hierarchical cluster based preprocessing methodology for Web Usage Mining. In Web Usage Mining (WUM), web session clustering plays a important function to categorize web users according to the user click history and similarity measure. Web session clustering according to Swarm assists in several manner for the purpose of managing the web resources efficiently like web personalization, schema modification, website alteration and web server performance. The author presents a framework for web session clustering in the preprocessing level of web usage mining. The framework will envelop the data preprocessing phase to practice the web log data and change the categorical web log data into numerical data. A session vector is determined, so that suitable similarity and swarm optimization could be used to cluster the web log data. The hierarchical cluster based technique will improve the conventional web session method for more structured information about the user sessions.

Yaxiu et al., put forth web usage mining based on fuzzy clustering. The World Wide Web has turn out to be the default knowledge resource for several fields of endeavor, organizations

require to recognize their customers' behavior, preferences, and future requirements, but when users browsing the Web site, several factors influence their interesting, and various factor has several degree of influence, the more factors consider, the more precisely can mirror the user's interest. This paper provides the effort to cluster similar Web user, by involving two factors that the page-click number and Web browsing time that are stored in the Web log, and the various degree of influence of the 87 two factors. The method suggested in this paper can help Web site organizations to recommend Web pages, enhance Web structure, so that can draw more customers, and improves customers' loyalty.

Jianxi et al, Data mining deals with the methods of nontrivial extraction of hidden, previously unknown, and potentially helpful data from very huge quantity of data. Web mining can be defined as the use of data mining methods to Web data. Web usage mining (WUM) is an significant kind in Web mining. Web usage mining is an essential and fast developing field of Web mining where many research has been performed previously. The author enhanced the fuzzy clustering technique to identify groups that share common B+tree indexes are the most common supported structures in RDBMS, but it is a well-known fact that tree structures have inconveniences when the cardinality of the attribute is small. Another class of index structures, the bitmap indexes, try to overcome the problem by using a bit structure to indicate the rows containing specific values of the indexed attribute by O'Neil in 1997.

According to Jurgens the performance of index structures depends 88 on many different parameters such as the number of stored rows, the cardinality of the data space, block size of the system, bandwidth of disks and latency time, only to mention some. The use of materialized views (or summary tables) is probably the most effective way to speedup specific queries in a data warehouse environment. Materialized views pre-compute and store (materialize) aggregates from the base data.

According to Chaudury in 1997 The data is grouped using categories from the dimensions tables, which corresponds to the subjects of interest (dimensions) of the organization. Storing all possible aggregates poses storage space problems and increases maintenance cost, since all stored aggregates need to be refreshed as updates are being made to the source databases. Many

algorithms have been proposed for selecting a representative subset of the possible views for materialization.

Ezeife et all said that corresponding to the most usual query patterns. But the main problems associated with materialized views are the difficulty to know in advance the expected set of queries, the problems of updating materialized views to reflect changes made to base relations and the large amount of space required to store the materialized views. It is worth noting that the techniques mentioned above (indexes and materialized views) are general techniques that can (and should) be used in the data warehouse approach proposed in the present paper. In fact, in a distributed environment each individual machine must also be tuned and optimized for performance.

According to DeWitt a large body of work exists in applying parallel processing techniques to relational database systems with the 89 purpose of accelerating query processing. The basic idea behind parallel databases is to carry out evaluation steps in parallel whenever possible, in order to improve performance. The parallelism is used to improve performance through parallel implementation of various operations such as loading data, building indexes and evaluating queries. One of the first works to propose a parallel physical design for the data warehouse.

Datta in 1998 in their work he suggest a vertical partitioning of the star schema including algorithms but without quantifying potential gains. Partitioning a large data set across several disks is another way to exploit the I/O bandwidth of the disks by reading and writing them in a parallel fashion. User queries have long been adopted for fragmenting a database in the relational, object-oriented and deductive database models.

According to Ezeife the set of user queries of a database is indicative of how often the database is accessed and of the portion of the database that is accessed to answer the queries. There are several ways to horizontally partition a relation. Typically, it can assign records to processors in a round-robin fashion (round-robin partitioning), It can use hashing (hash partitioning), or it can assign records to processors by ranges of values (range partitioning). Most of today's OLAP tools require data warehouses with a centralized structure where a single database contains all the

data. However, the centralized data warehouse is very expensive because of great setup costs and lack of structural flexibility in data storage.

Albrecht proposed the performance of many distributed queries is normally poor, mainly due to load balance problems. Furthermore, each individual data mart is primarily designed and tuned to answer the queries related to its own subject area and the response to global queries is dependent on global system tuning and network speed.

Introduction: Data integrity is a fundamental component of information security. In its broadest use, "data integrity" refers to the accuracy and consistency of data stored in a database, data warehouse, data mart or other construct. The term – Data Integrity - can be used to describe a state, a process or a function – and is often used as a proxy for "data quality". Data with "integrity" is said to have a complete or whole structure. Data values are standardized according to a data model and/or data type. All characteristics of the data must be correct – including business rules, relations, dates, definitions and lineage – for data to be complete. Data integrity is imposed within a database when it is designed and is authenticated through the ongoing use of error checking and validation routines. As a simple example, to maintain data integrity numeric columns/cells should not accept alphabetic data. As a process, data integrity verifies that data has remained unaltered in transit from creation to reception. As a state or condition, Data Integrity is a measure of the validity and fidelity of a data object. As a function related to security, a data integrity service maintains information exactly as it was inputted, and is auditable to affirm its reliability. Data undergoes any number of operations in support of decision-making, such as capture, storage, retrieval, update and transfer. Data integrity can also be a performance measure during these operations based on the detected error rate. Data must be kept free from corruption, modification or unauthorized disclosure to drive any number of mission-critical business processes with accuracy. Inaccuracies can occur either accidentally (e.g .through programming errors), or maliciously (e.g. through breaches or hacks). Database security professionals employ any number of practices to assure data integrity, including:Data encryption, which locks data by cipher,data backup, which stores a copy of data in an alternate location, access controls, including assignment of read/write privileges, Input validation, to prevent incorrect data entry,Data validation, to certify uncorrupted transmission. Software developers must also be

concerned with data integrity. They can define integrity constraints to enforce business rules on data when entered into an application. Business rules specify conditions and relationships that must always be true, or must always be false. When a data integrity constraint is applied to a database table, all data in the table must conform to the corresponding rule.

Figure: 1; Concept of Data Integrity: Source;Wikipedia.Org

Data integrity contains guidelines for data retention, specifying or guaranteeing the length of time data can be retained in a particular database. It specifies what can be done with data values when their validity or usefulness expires. In order to achieve data integrity, these rules are consistently and routinely applied to all data entering the system, and any relaxation of enforcement could cause errors in the data. Implementing checks on the data as close as possible to the source of input (such as human data entry), causes less erroneous data to enter the system. Strict enforcement of data integrity rules causes the error rates to be lower, resulting in time saved troubleshooting and tracing erroneous data and the errors it causes algorithms. Data integrity also includes rules defining the relations a piece of data can have, to other pieces of data, such as a *Customer* record being allowed to link to purchased *Products*, but not to unrelated data such as *Corporate Assets*. Data integrity often includes checks and correction for invalid data, based on a fixed schema or a predefined set of rules. An example being textual data entered where a date-time value is required. Rules for data derivation are also applicable, specifying how a data value is derived based on algorithm, contributors and conditions. It also specifies the conditions on how the data value could be re-derived.

Objective (i) To understand the overall intent of any Data Integrity and Kinds of Data Integrity

Kinds of Data Integrity: **Integrity types**: **Physical integrity**:Physical integrity deals with challenges associated with correctly storing and fetching the data itself. Challenges with physical integrity may include electromechanical faults, design flaws, material fatigue, corrosion, power outages, natural disasters, acts of war and terrorism, and other special environmental hazards such as ionizing radiation, extreme temperatures, pressures and g-forces. Ensuring physical integrity includes methods such as redundant hardware, an uninterruptible power supply, certain types of RAID arrays,radiation hardened chips, error-correcting memory, use of a clustered file system, using file systems that employ block level checksums such as ZFS, storage arrays that compute parity calculations such as exclusive or or use a cryptographic hash function and even having a watchdog timer on critical subsystems. Physical integrity often makes extensive use of error detecting algorithms known as error-correcting codes. Human-induced data integrity errors are often detected through the use of simpler checks and algorithms, such as the Damm algorithm or Luhn algorithm. These are used to maintain data integrity after manual transcription from one computer system to another by a human intermediary (e.g. credit card or bank routing numbers). Computer-induced transcription errors can be detected through hash functions. In production systems these techniques are used together to ensure various degrees of data integrity. For example, a computer file system may be configured on a fault-tolerant RAID array, but might not provide block-level checksums to detect and prevent silent data corruption. As another example, a database management system might be compliant with the ACID properties, but the RAID controller or hard disk drive's internal write cache might not be.

Logical integrity; This type of integrity is concerned with the correctness or rationality of a piece of data, given a particular context. This includes topics such as referential integrity and entity integrity in a relational database or correctly ignoring impossible sensor data in robotic systems. These concerns involve ensuring that the data "makes sense" given its environment. Challenges include software bugs, design flaws, and human errors. Common methods of ensuring logical integrity include things such as a check constraints,foreign key constraints, program assertions, and other run-time sanity checks. Both physical and logical integrity often share many common challenges such as human errors and design flaws, and both must appropriately deal with concurrent requests to record and retrieve data, the latter of which is entirely a subject on its own.

Objective (ii); To learn about a broad overview of some of the different types and concerns of data integrity.

Types of integrity constraints; Data integrity is normally enforced in a database system by a series of integrity constraints or rules. Three types of integrity constraints are an inherent part of the relational data model: entity integrity, referential integrity and domain integrity: **Entity integrity concerns** the concept of a primary key. Entity integrity is an integrity rule which states that every table must have a primary key and that the column or columns chosen to be the primary key should be unique and not null. **Referential integrity concerns** the concept of a foreign key. The referential integrity rule states that any foreign-key value can only be in one of two states. The usual state of affairs is that the foreign-key value refers to a primary key value of some table in the database. Occasionally, and this will depend on the rules of the data owner, a foreign-key value can be null. In this case we are explicitly saying that either there is no relationship between the objects represented in the database or that this relationship is unknown. **Domain integrity specifies** that all columns in a relational database must be declared upon a defined domain. The primary unit of data in the relational data model is the data item. Such data items are said to be non-decomposable or atomic. A domain is a set of values of the same type. Domains are therefore pools of values from which actual values appearing in the columns of a table are drawn. **User-defined integrity** refers to a set of rules specified by a user, which do not belong to the entity, domain and referential integrity categories. If a database supports these features, it is the responsibility of the database to ensure data integrity as well as the consistency model for the data storage and retrieval. If a database does not support these features it is the responsibility of the applications to ensure data integrity while the database supports the consistency model for the data storage and retrieval. Having a single, well-controlled, and well-defined data-integrity system increases, stability (one centralized system performs all data integrity operations), performance (all data integrity operations are performed in the same tier as the consistency model), re-usability (all applications benefit from a single centralized data integrity system), maintainability (one centralized system for all data integrity administration). Modern databases support these features (see Comparison of relational database management systems), and it has become the de facto responsibility of the database to ensure data integrity.

Companies, and indeed many database systems, offer products and services to migrate legacy systems to modern databases.

File systems; Various research results show that neither widespread filesystems (including UFS, Ext, XFS, JFS and NTFS) nor hardware RAID solutions provide sufficient protection against data integrity problems. Some filesystems (including Btrfs and ZFS) provide internal data and metadata checksumming, what is used for detecting silent data corruption and improving data integrity. If a corruption is detected that way and internal RAID mechanisms provided by those filesystems are also used, such filesystems can additionally reconstruct corrupted data in a transparent way. This approach allows improved data integrity protection covering the entire data paths, which is usually known as end-to-end data protection.

Data storage; Apart from data in databases, standards exist to address the integrity of data on storage devices. While it's easy to see why data has become so important to modern businesses, the hazards it presents must also be accounted for. With new risks and pitfalls appearing daily as technology evolves, it's no surprise that security is usually at the top of the priority list. In truth, though, security is only one part of the equation. For any organisation to properly harness the full power of data, it must also see accuracy and integrity in similar lights. Once these have been guaranteed, it's left to the way in which the data is applied to determine results. This is where the real-time concept comes in; information has a shelf-life, and it's shortening rapidly.

Accuracy; Put simply, data is used to provide insight. Businesses, when armed with this, are able to improve the everyday decisions they make. This isn't just for management, either – it applies from the ground up. However, data is rarely useful in its raw state; it must be processed and presented in a way that works on the appropriate levels so that it can be applied properly. The latest analytics tools make this part much easier, but there is still a journey that information must follow before it's usable. If data accuracy levels are low at the start of this process, the insight will be lacking and the decisions it influences are likely to be poor as a result. This is why organisations must realise that quality is more important that quantity; too many focus only on

gathering as much information as possible without thinking about whether it's correct and how it can be used. Add to this the question of whether it can be trusted and you have the issue of integrity to consider as well.

Objective: (iii) To evaluate the process of data loss and how to have countermeasures

Integrity; There's no getting away from the fact that data comes from everywhere these days. Just as a few examples, we have mobile devices, loyalty cards, customer relationship management (CRM) systems, social media sites, GPS location data and complex market research tools. The source pool is still growing too; the ongoing development of concepts like the Internet of Things (IoT) mean that machines are also becoming an integral part of the data deluge. Alongside the more traditional computerised devices, businesses will soon be mining information from seemingly inanimate objects (think fridges, tables and cars). With all of this in mind, data governance must be a priority. Not only will this information be arriving from all directions, it will exist in various formats: everything from numbers and formulas to individual words and pieces of text. Traditionally, just as many tools would be used to deal with it as well. Some staff would be relying on their own spreadsheets and word documents while other, more data-competent team members put their faith in advanced data visualisation tools. This kind of disparity causes its own set of unique problems – and can even make data useless. Unless every data-using employee is singing from the same hymn sheet, auditability will always be a chore at best. With different platforms being depended on, and information coming from lots of different sources, the reliability of information cannot be reliably determined; and when this is the case, it should not be used to influence decisions. At the very least, decision makers need to know about the data heritage and the (lack of) confidence they can place in this information. The solution to this is governance. Information has to be available to everyone, and while the latest wave of data discovery tools offer the right kind of accessibility, the need for central control is somewhat ignored. An organisation's use of data can't be limited to the IT department, but in order to ensure integrity on the whole, it's IT staff that should be in control.

Real-time integration; The time it takes experts to generate actionable insight from raw data has always been a stumbling block, especially when the older BI model is adopted. Data is undoubtedly the business' single greatest asset, but with every other organisation looking to

harness its power, the usefulness decreases as time goes on. It's no longer enough to embrace information. Now, it must be used quickly – or even instantly. The periods of time in which businesses have to make their decisions – both major and minor – are shortening. When sales staff are out talking to clients, for example, they must be able to draw on the facts and figures that are up to date and that support their cases there and then; instead of having to wait for days, or even longer, until it's ready. It's this instant access that enables a firm to get ahead of its competitors, or keep up with them at the very least. Responsiveness is also an important factor for firms to consider. To be truly competitive in the respective industries, companies must be in a position to respond to what's going on around them – whether this involves the actions of a competitor, the behaviour of clients or a major world event. It's this kind of versatility that separates a good business from a truly innovative one.

The danger of inaccurate data; Companies that utilise sophisticated data-led insights can't simply rest on their laurels and marvel at a job well done, however. Inaccuracies in the data can quickly escalate from a minor niggle into something that compromises all the hard work and effort previously invested. Analysts (among them Joel Curry of Experian QAS UK and Ireland) have noted that accurate data can drive efficiency, profitability and growth. Inaccurate data, on the other hand, can cause real detriment to a business – and its bottom line. This can be from as simple and seemingly innocuous circumstances as one person using data discovery whilst a second focuses instead on Excel or standard reports. It could even simply be under-briefed employees using slightly disparate terminology which throws their data out of sync. As innocent and accidental as this all sounds, the impact could be huge. Curry went on to cite a survey which showed that one pound in every six that is spent from departmental budget is wasted. The biggest consideration here is that these losses are wholly avoidable, by ensuring that data accuracy always remains a key consideration.

Data corruption; Data corruption refers to errors in computer data that occur during writing, reading, storage, transmission, or processing, which introduce unintended changes to the original data. Computer, transmission, and storage systems use a number of measures to provide end-to-enddata integrity, or lack of errors.In general, when data corruption occurs a file containing that data will produce unexpected results when accessed by the system or the related application.

Results could range from a minor loss of data to a system crash. For example, if a document file is corrupted, when a person tries to open that file with a document editor they may get an error message, thus the file might not be opened or might open with some of the data corrupted (or in some cases, completely corrupted, leaving the document unintelligble). The image to the right is a corrupted image file in which most of the information has been lost.Some types of malware may intentionally corrupt files as part of their payloads, usually by overwriting them with inoperative or garbage code, while non-malicious viruses may also unintentionally corrupt files when it accesses them. If a virus or trojan with this payload method manages to alter files critical to the running of the computer's operating system software or physical hardware, the entire system may be rendered unusable.Some programs can give a suggestion to repair the file automatically (after the error), and some programs cannot repair it. It depends on the level of corruption, and the built-in functionality of the application to handle the error. There are various causes of the corruption.There are two types of data corruption associated with computer systems: undetected and detected. Undetected data corruption, also known as *silent data corruption*, results in the most dangerous errors as there is no indication that the data is incorrect. Detected data corruption may be permanent with the loss of data, or may be temporary when some part of the system is able to detect and correct the error; there is no data corruption in the latter case.Data corruption can occur at any level in a system, from the host to the storage medium. Modern systems attempt to detect corruption at many layers and then recover or correct the corruption; this is almost always successful but very rarely the information arriving in the systems memory is corrupted and can cause unpredictable results.Data corruption during transmission has a variety of causes. Interruption of data transmission causes information loss. Environmental conditions can interfere with data transmission, especially when dealing with wireless transmission methods. Heavy clouds can block satellite transmissions. Wireless networks are susceptible to interference from devices such as microwave ovens.Hardware and software failure are the two main causes for data loss. Background radiation, head crashes, and aging or wear of the storage device fall into the former category, while software failure typically occurs due to bugs in the code. Cosmic rays cause most soft errors in DRAM.

Silent; Some errors go unnoticed, without being detected by the disk firmware or the host operating system; these errors are known as *silent data corruption*.There are many error sources

beyond the disk storage subsystem itself. For instance, cables might be slightly loose, the power supply might be unreliable, external vibrations such as a loud sound, the network might introduce undetected corruption, cosmic radiation and many other causes of soft memory errors, etc.

Countermeasures; When data corruption behaves as a Poisson process, where each bit of data has an independently low probability of being changed, data corruption can generally be detected by the use of checksums, and can often be corrected by the use of error correcting codes. If an uncorrectable data corruption is detected, procedures such as automatic retransmission or restoration from backups can be applied. Certain levels of RAID disk arrays have the ability to store and evaluate parity bits for data across a set of hard disks and can reconstruct corrupted data upon the failure of a single or multiple disks, depending on the level of RAID implemented. Some CPU architectures employ various transparent checks to detect and mitigate data corruption in CPU caches, CPU buffers and instruction pipelines; an example is *Intel Instruction Replay* technology, which is available on Intel Itanium processors. Many errors are detected and corrected by the hard disk drives using the ECC/CRC codes which are stored on disk for each sector. If the disk drive detects multiple read errors on a sector it may make a copy of the failing sector on another part of the disk, by remapping the failed sector of the disk to a spare sector without the involvement of the operating system (though this may be delayed until the next write to the sector). This "silent correction" can be monitored using S.M.A.R.T. and tools available for most operating systems to automatically check the disk drive for impending failures by watching for deteriorating SMART parameters.Some file systems, such as Btrfs, HAMMER, ReFS, and ZFS, use internal data and metadata checksumming to detect silent data corruption. In addition, if a corruption is detected and the file system uses integrated RAID mechanisms that provide data redundancy, such file systems can also reconstruct corrupted data in a transparent way. This approach allows improved data integrity protection covering the entire data paths, which is usually known as *end-to-end data protection*, compared with other data integrity approaches that do not span different layers in the storage stack and allow data corruption to occur while the data passes boundaries between the different layers. *Data scrubbing* is another method to reduce the likelihood of data corruption, as disk errors are caught and recovered from before multiple errors accumulate and overwhelm the number of parity bits. Instead of parity

being checked on each read, the parity is checked during a regular scan of the disk, often done as a low priority background process. Note that the "data scrubbing" operation activates a parity check. If a user simply runs a normal program that reads data from the disk, then the parity would not be checked unless parity-check-on-read was both supported and enabled on the disk subsystem.If appropriate mechanisms are employed to detect and remedy data corruption, data integrity can be maintained. This is particularly important in commercial applications (e.g.banking), where an undetected error could either corrupt a database index or change data to drastically affect an account balance, and in the use of encrypted or compresseddata, where a small error can make an extensive dataset unusable.

Data corruption; Data corruption refers to errors in computer data that occur during writing, reading, storage, transmission, or processing, which introduce unintended changes to the original data. Computer, transmission, and storage systems use a number of measures to provide end-to-enddata integrity, or lack of errors.In general, when data corruption occurs a file containing that data will produce unexpected results when accessed by the system or the related application. Results could range from a minor loss of data to a system crash. For example, if a document file is corrupted, when a person tries to open that file with a document editor they may get an error message, thus the file might not be opened or might open with some of the data corrupted (or in some cases, completely corrupted, leaving the document unintelligble). The image to the right is a corrupted image file in which most of the information has been lost.Some types of malware may intentionally corrupt files as part of their payloads, usually by overwriting them with inoperative or garbage code, while non-malicious viruses may also unintentionally corrupt files when it accesses them. If a virus or trojan with this payload method manages to alter files critical to the running of the computer's operating system software or physical hardware, the entire system may be rendered unusable.Some programs can give a suggestion to repair the file automatically (after the error), and some programs cannot repair it. It depends on the level of corruption, and the built-in functionality of the application to handle the error. There are various causes of the corruption. There are two types of data corruption associated with computer systems: undetected and detected. Undetected data corruption, also known as *silent data corruption*, results in the most dangerous errors as there is no indication that the data is incorrect. Detected data corruption may be permanent with the loss of data, or may be temporary when

some part of the system is able to detect and correct the error; there is no data corruption in the latter case.Data corruption can occur at any level in a system, from the host to the storage medium. Modern systems attempt to detect corruption at many layers and then recover or correct the corruption; this is almost always successful but very rarely the information arriving in the systems memory is corrupted and can cause unpredictable results. Data corruption during transmission has a variety of causes. Interruption of data transmission causes information loss. Environmental conditions can interfere with data transmission, especially when dealing with wireless transmission methods. Heavy clouds can block satellite transmissions. Wireless networks are susceptible to interference from devices such as microwave ovens.Hardware and software failure are the two main causes for data loss. Background radiation, head crashes, and aging or wear of the storage device fall into the former category, while software failure typically occurs due to bugs in the code. Cosmic rays cause most soft errors in DRAM.

Data loss; Data loss is an error condition in information systems in which information is destroyed by failures or neglect in storage, transmission, or processing. Information systems implement backup and disaster recovery equipment and processes to prevent data loss or restore lost data.*Data loss* is distinguished from *data unavailability*, which may arise from a network outage. Although the two have substantially similar consequences for users, data unavailability is temporary, while data loss may be permanent. Data loss is also distinct from data breach, an incident where data falls into the wrong hands, although the term data loss has been used in those incidents.

Types of data loss; Intentional Action; Intentional deletion of a file or program, Unintentional Action, Accidental deletion of a file or program, Misplacement of CDs or Memory sticks, Administration errors, Inability to read unknown file format.**Failure** Power failure, resulting in data in volatile memory not being saved to permanent memory. Hardware failure, such as a head crash in a hard disk. A software crash or freeze, resulting in data not being saved. Software bugs or poor usability, such as not confirming a file delete command. Business failure (vendor bankruptcy), where data is stored with a software vendor using Software-as-a-service and SaaS data escrow has not been provisioned. Data corruption, such as file system corruption or database corruption. **Disaster;** Natural disaster, earthquake, flood, tornado, etc.,

Fire.**Crime;** Theft, hacking, SQL injection, sabotage, etc.A malicious act, such as a worm, virus, hacker or theft of physical media. Studies show hardware failure and human error are the two most common causes of data loss, accounting for roughly three quarters of all incidents. Another cause of data loss is a natural disaster, which is a greater risk dependant on where the hardware is located. While the probability of data loss due to natural disaster is small, the only way to prepare for such an event is to store backup data in a separate physical location. As such, the best backup plans always include at least one copy being stored off-site. **Cost of data loss;** The cost of a data loss event is directly related to the value of the data and the length of time that it is unavailable yet needed. For an enterprise in particular, the definition of cost extends beyond the financial and can also include time. Consider: The cost of continuing without the data. The cost of recreating the data. **Prevention;** The frequency of data loss and the impact can be greatly mitigated by taking proper precautions, those of which necessary can vary depending on the type of data loss. For example, multiple power circuits with battery backup and a generator only protect against power failures, though using an Uninterruptable Power Supply can protect your drive against sudden power spikes. Similarly, using a journaling file system and RAID storage only protect against certain types of software and hardware failure. For hard disk drives, which are a physical storage medium, ensuring minimal vibration and movement will help protect against damaging the components internally, as can maintaining a suitable drive temperature. Regular data backups are an important asset to have when trying to recover after a data loss event, but they do not prevent user errors or system failures. As such, a data backup plan needs to be established and run in unison with a disaster recovery plan in order to lower risk. **Data recovery;** Data recovery is often performed by specialized commercial services that have developed often proprietary methods to recover data from physically damaged media. Service costs at data recovery labs are usually dependent on type of damage and type of storage medium, as well as the required security or cleanroom procedures.File system corruption can frequently be repaired by the user or the system administrator. For example, a deleted file is typically not immediately overwritten on disk, but more often simply has its entry deleted from the file system index. In such a case, the deletion can be easily reversed.Successful recovery from data loss generally requires implementation of an effective backup strategy. Without an implemented backup strategy, recovery requires reinstallation of programs and regeneration of data. Even with an effective backup strategy, restoring a system to the precise state it was in

prior to the *Data Loss Event* is extremely difficult. Some level of compromise between granularity of recoverability and cost is necessary. Furthermore, a *Data Loss Event* may not be immediately apparent. An effective backup strategy must also consider the cost of maintaining the ability to recover lost data for long periods of time.A highly effective backup system would have duplicate copies of every file and program that were immediately accessible whenever a*Data Loss Event* was noticed. However, in most situations, there is an inverse correlation between the value of a unit of data and the length of time it takes to notice the loss of that data. Taking this into consideration, many backup strategies decrease the granularity of restorability as the time increases since the potential *Data Loss Event*. By this logic, recovery from recent *Data Loss Events* is easier and more complete than recovery from *Data Loss Events* that happened further in the past.Recovery is also related to the type of *Data Loss Event.* Recovering a single lost file is substantially different from recovering an entire system that was destroyed in a disaster. An effective backup regimen has some proportionality between the magnitude of *Data Loss* and the magnitude of effort required to recover. For example, it should be far easier to restore the single lost file than to recover the entire system. **Initial steps upon data loss;** If data loss occurs, a successful recovery must ensure that the deleted data is not over-written. For this reason — one should avoid all write operations to the affected storage device. This includes not starting the system to which the affected device is connected. This is because many operating systems create temporary files in order to boot, and these may overwrite areas of lost data — rendering it unrecoverable. Viewing web pages has the same effect — potentially overwriting lost files with the temporary html and image files created when viewing a web page. File operations such as copying, editing, or deleting should also be avoided.Upon realizing data loss has occurred, it is often best to shut down the computer and remove the drive in question from the unit. Re-attach this drive to a secondary computer with a write blocker device and then attempt to recover lost data. If possible, create an image of the drive in order to establish a secondary copy of the data. This can then be tested on, with recovery attempted, abolishing the risk of harming the source data.

Data security; **Data security** means protecting data, such as those in a database, from destructive forces and from the unwanted actions of unauthorized users. Technologies; **Disk encryption;** Disk encryption refers to encryption technology that encrypts data on a hard disk

drive. Disk encryption typically takes form in either software (see disk encryption software) or hardware (see disk encryption hardware). Disk encryption is often referred to as on-the-fly encryption (OTFE) or transparent encryption. **Software versus hardware-based mechanisms for protecting data;** Software-based security solutions encrypt the data to protect it from theft. However, a malicious program or a hacker could corrupt the data in order to make it unrecoverable, making the system unusable. Hardware-based security solutions can prevent read and write access to data and hence offer very strong protection against tampering and unauthorized access.Hardware based security or assisted computer security offers an alternative to software-only computer security. Security tokens such as those using PKCS#11 may be more secure due to the physical access required in order to be compromised. Access is enabled only when the token is connected and correct PIN is entered (see two-factor authentication). However, dongles can be used by anyone who can gain physical access to it. Newer technologies in hardware-based security solves this problem offering full proof security for data.Working of hardware-based security: A hardware device allows a user to log in, log out and set different privilege levels by doing manual actions. The device uses biometric technology to prevent malicious users from logging in, logging out, and changing privilege levels. The current state of a user of the device is read by controllers in peripheral devices such as hard disks. Illegal access by a malicious user or a malicious program is interrupted based on the current state of a user by hard disk and DVD controllers making illegal access to data impossible. Hardware-based access control is more secure than protection provided by the operating systems as operating systems are vulnerable to malicious attacks by viruses and hackers. The data on hard disks can be corrupted after a malicious access is obtained. With hardware-based protection, software cannot manipulate the user privilege levels. It is impossible for a hacker or a malicious program to gain access to secure data protected by hardware or perform unauthorized privileged operations. This assumption is broken only if the hardware itself is malicious or contains a backdoor.[2] The hardware protects the operating system image and file system privileges from being tampered. Therefore, a completely secure system can be created using a combination of hardware-based security and secure system administration policies. **Backups;** Backups are used to ensure data which is lost can be recovered from another source. It is considered essential to keep a backup of any data in most industries and the process is recommended for any files of importance to a user.**Data masking;** Data masking of structured data is the process of obscuring (masking)

specific data within a database table or cell to ensure that data security is maintained and sensitive information is not exposed to unauthorized personnel. This may include masking the data from users (for example so banking customer representatives can only see the last 4 digits of a customers national identity number), developers (who need real production data to test new software releases but should not be able to see sensitive financial data), outsourcing vendors, etc. **Data erasure;** Data erasure is a method of software-based overwriting that completely destroys all electronic data residing on a hard drive or other digital media to ensure that no sensitive data is leaked when an asset is retired or reused.

Findings and Conclusion;

International laws and standards; International laws; Data Protection Act is used to ensure that personal data is accessible to those whom it concerns, and provides redress to individuals if there are inaccuracies. This is particularly important to ensure individuals are treated fairly, for example for credit checking purposes. The Data Protection Act states that only individuals and companies with legitimate and lawful reasons can process personal information and cannot be shared.

International standards; The international standards ISO/IEC 27001:2013 and ISO/IEC 27002:2013 covers data security under the topic of information security, and one of its cardinal principles is that all stored information, i.e. data, should be owned so that it is clear whose responsibility it is to protect and control access to that data.The Trusted Computing Group is an organization that helps standardize computing security technologies.The Payment Card Industry Data Security Standard is a proprietary international information security standard for organizations that handle cardholder information for the majordebit, credit, prepaid, e-purse, ATM and POS cards. **Safety-critical system;** A **safety-critical system** or **life-critical system** is a system whose failure or malfunction may result in one (or more) of the following outcomes: death or serious injury to people. loss or severe damage to equipment/property, environmental harm. A **safety-related system** (or sometimes **safety-involved system**) comprises everything (hardware, software, and human aspects) needed to perform one or more safety functions, in which failure would cause a significant increase in the safety risk for the people and/or environment involved. Safety-related systems are those that do not have full responsibility

for controlling hazards such as loss of life, severe injury or severe environmental damage. The malfunction of a safety-involved system would only be that hazardous in conjunction with the failure of other systems or human error. Some safety organizations provide guidance on safety-related systems, for example the Health and Safety Executive (HSE) in the United Kingdom. Risks of this sort are usually managed with the methods and tools of safety engineering. A safety-critical system is designed to lose less than one life per billion (10^9) hours of operation. Typical design methods include probabilistic risk assessment, a method that combines failure mode and effects analysis (FMEA) with fault tree analysis. Safety-critical systems are increasingly computer-based.

Reliability regimes; Fail-operational systems continue to operate when their control systems fail. Examples of these include elevators, the gas thermostats in most home furnaces, andpassively safe nuclear reactors. Fail-operational mode is sometimes unsafe. Nuclear weapons launch-on-loss-of-communications was rejected as a control system for the U.S. nuclear forces because it is fail-operational: a loss of communications would cause launch, so this mode of operation was considered too risky. This is contrasted with theFail-deadly behavior of the Perimeter system built during the Soviet era. Fail-safe systems become safe when they cannot operate. Many medical systems fall into this category. For example, an infusion pump can fail, and as long as it alerts the nurse and ceases pumping, it will not threaten the loss of life because its safety interval is long enough to permit a human response. In a similar vein, an industrial or domestic burner controller can fail, but must fail in a safe mode (i.e. turn combustion off when they detect faults). Famously, nuclear weapon systems that launch-on-command are fail-safe, because if the communications systems fail, launch cannot be commanded. Railway signaling is designed to be fail-safe.Fail-secure systems maintain maximum security when they can not operate. For example, while fail-safe electronic doors unlock during power failures, fail-secure ones will lock, keeping an area secure.Fail-Passive systems continue to operate in the event of a system failure. An example includes an aircraft autopilot. In the event of a failure, the aircraft would remain in a controllable state and allow the pilot to take over and complete the journey and perform a safe landing.Fault-tolerant systems avoid service failure when faults are introduced to the system. An example may include control systems for ordinary nuclear reactors. The normal method to tolerate faults is to have

several computers continually test the parts of a system, and switch on hot spares for failing subsystems. As long as faulty subsystems are replaced or repaired at normal maintenance intervals, these systems are considered safe. Interestingly, the computers, power supplies and control terminals used by human beings must all be duplicated in these systems in some fashion.

Software engineering for safety-critical systems; Software engineering for safety-critical systems is particularly difficult. There are three aspects which can be applied to aid the engineering software for life-critical systems. First is process engineering and management. Secondly, selecting the appropriate tools and environment for the system. This allows the system developer to effectively test the system by emulation and observe its effectiveness. Thirdly, address any legal and regulatory requirements, such as FAA requirements for aviation. By setting a standard for which a system is required to be developed under, it forces the designers to stick to the requirements. The avionics industry has succeeded in producing standard methods for producing life-critical avionics software. Similar standards exist for automotive (ISO 26262), Medical (IEC 62304) and nuclear (IEC 61513) industries. The standard approach is to carefully code, inspect, document, test, verify and analyze the system. Another approach is to certify a production system, a compiler, and then generate the system's code from specifications. Another approach uses formal methods to generate proofs that the code meets requirements. All of these approaches improve the software quality in safety-critical systems by testing or eliminating manual steps in the development process, because people make mistakes, and these mistakes are the most common cause of potential life-threatening errors.

References:

1. "Safety-critical system". encyclopedia.com. Retrieved 15 April 2017.
2. "FAQ – Edition 2.0: E) Key concepts". IEC 61508 – Functional Safety. International Electrotechnical Commission. Retrieved 23 October 2016.
3. "Part 1: Key guidance". Managing competence for safety-related systems (PDF). UK: Health and Safety Executive. 2007. Retrieved 23 October 2016.
4. Bowen, Jonathan P. (April 2000). "The Ethics of Safety-Critical Systems". Communications of the ACM. **43** (4). pp. 91–97. doi:10.1145/332051.332078.
5. "Inside the Apocalyptic Soviet Doomsday Machine". WIRED.

6. Bowen, Jonathan P.; Stavridou, Victoria (July 1993). "Safety-critical systems, formal methods and standards". Software Engineering Journal. **8** (4). IEE/BCS. pp. 189–209.doi:10.1049/sej.1993.0025.

7. "Medical Device Safety System Design: A Systematic Approach". mddionline.com.

8. "Safety of Nuclear Reactors". world-nuclear.org.

9. "Safety-Critical Systems in Rail Transportation" (PDF). Rtos.com. Retrieved 2016-10-23.

10. "Safety-Critical Automotive Systems". sae.org.

11. Leanna Rierson. Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance. ISBN 978-1-4398-1368-3.

12. "NASA Procedures and Guidelines : NPG: 8705.2" (PDF). Dept.aoe.vt.edu. June 19, 2003. Retrieved 2016-10-23.

13. Boritz, J. "IS Practitioners' Views on Core Concepts of Information Integrity".International Journal of Accounting Information Systems. Elsevier. Archived from the original on 5 October 2011. Retrieved 12 August 2011.

14. Vijayan Prabhakaran (2006). "IRON FILE SYSTEMS" (PDF). Doctor of Philosophy in Computer Sciences. University of Wisconsin-Madison. Retrieved 9 June 2012.

15. "Parity Lost and Parity Regained".

16. "An Analysis of Data Corruption in the Storage Stack" (PDF).

17. "Impact of Disk Corruption on Open-Source DBMS" (PDF).

18. "Baarf.com". Baarf.com. Retrieved November 4, 2011.

19. Bierman, Margaret; Grimmer, Lenz (August 2012). "How I Use the Advanced Capabilities of Btrfs". Retrieved 2014-01-02.

20. Yupu Zhang; Abhishek Rajimwale; Andrea C. Arpaci-Dusseau; Remzi H. Arpaci-Dusseau."End-to-end Data Integrity for File Systems: A ZFS Case Study" (PDF). Computer Sciences Department, University of Wisconsin. Retrieved 2014-01-02.

21. Smith, Hubbert (2012). Data Center Storage: Cost-Effective Strategies, Implementation, and Management. CRC Press. ISBN 9781466507814. Retrieved 2012-11-15. T10-DIF (data integrity field), also known as T10-PI (protection information model), is a data-integrity extension to the existing information packets that spans servers and storage systems and disk drives.

22. Scientific American (2008-07-21). "Solar Storms: Fast Facts". Nature Publishing Group. Retrieved 2009-12-08.

23. Eric Lowe (16 November 2005). "ZFS saves the day(-ta)!" (Blog). Oracle – Core Dumps of a Kernel Hacker's Brain – Eric Lowe's Blog. Oracle. Retrieved 9 June 2012.

24. bcantrill (31 December 2008). "Shouting in the Datacenter" (Video file). YouTube. Google. Retrieved 9 June 2012.

25. jforonda (31 January 2007). "Faulty FC port meets ZFS" (Blog). Blogger – Outside the Box. Google. Retrieved 9 June 2012.

26. "Are Disks the Dominant Contributor for Storage Failures? A Comprehensive Study of Storage Subsystem Failure Characteristics" (PDF). USENIX. Retrieved 2014-01-18.

27. Bernd Panzer-Steindel (8 April 2007). "Draft 1.3". Data integrity. CERN. Retrieved9 June 2012.

28. "Observations on Errors, Corrections, & Trust of Dependent Systems".

29. "Silent data corruption in disk arrays: A solution" (PDF). NEC. 2009. Retrieved2013-10-24.

30. "A Conversation with Jeff Bonwick and Bill Moore". Association for Computing Machinery. November 15, 2007. Retrieved December 6, 2010.

31. David S. H. Rosenthal (October 1, 2010). "Keeping Bits Safe: How Hard Can It Be?".ACM Queue. Retrieved 2014-01-02.

32. Lakshmi N. Bairavasundaram; Garth R. Goodson; Shankar Pasupathy; Jiri Schindler (June 2007). "An Analysis of Latent Sector Errors in Disk Drives". Proceedings of the International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS'07). San Diego, California, United States: ACM: 289–300.doi:10.1145/1254882.1254917. Retrieved 9 June 2012.

33. David Fiala; Frank Mueller; Christian Engelmann; Rolf Riesen; Kurt Ferreira; Ron Brightwell (November 2012). "Detection and Correction of Silent Data Corruption for Large-Scale High-Performance Computing" (PDF). fiala.me. IEEE. Retrieved 2015-01-26.

34. Steve Bostian (2012). "Rachet Up Reliability for Mission-Critical Applications: Intel Instruction Replay Technology" (PDF). Intel. Retrieved 2016-01-27.

35. "Read Error Severities and Error Management Logic". Retrieved 4 April 2012.

36. Margaret Bierman; Lenz Grimmer (August 2012). "How I Use the Advanced Capabilities of Btrfs". Oracle Corporation. Retrieved 2014-01-02.

37. Yupu Zhang; Abhishek Rajimwale; Andrea C. Arpaci-Dusseau; Remzi H. Arpaci-Dusseau (2010-02-04). "End-to-end Data Integrity for File Systems: A ZFS Case Study" (PDF). Computer Sciences Department, University of Wisconsin. Retrieved 2014-08-12.

38. Scientific American (2008-07-21). "Solar Storms: Fast Facts". Nature Publishing Group. Retrieved 2009-12-08.

39. Eric Lowe (16 November 2005). "ZFS saves the day(-ta)!" (Blog). Oracle – Core Dumps of a Kernel Hacker's Brain – Eric Lowe's Blog. Oracle. Retrieved 9 June 2012.

40. bcantrill (31 December 2008). "Shouting in the Datacenter" (Video file). YouTube. Google. Retrieved 9 June 2012.

41. jforonda (31 January 2007). "Faulty FC port meets ZFS" (Blog). Blogger – Outside the Box. Google. Retrieved 9 June 2012.

42. "Are Disks the Dominant Contributor for Storage Failures? A Comprehensive Study of Storage Subsystem Failure Characteristics" (PDF). USENIX. Retrieved 2014-01-18.

43. Bernd Panzer-Steindel (8 April 2007). "Draft 1.3". Data integrity. CERN. Retrieved9 June 2012.

44. "Observations on Errors, Corrections, & Trust of Dependent Systems".

45. "Silent data corruption in disk arrays: A solution" (PDF). NEC. 2009. Retrieved2013-10-24.

46. "A Conversation with Jeff Bonwick and Bill Moore". Association for Computing Machinery. November 15, 2007. Retrieved December 6, 2010.

47. David S. H. Rosenthal (October 1, 2010). "Keeping Bits Safe: How Hard Can It Be?".ACM Queue. Retrieved 2014-01-02.

48. Lakshmi N. Bairavasundaram; Garth R. Goodson; Shankar Pasupathy; Jiri Schindler (June 2007). "An Analysis of Latent Sector Errors in Disk Drives". Proceedings of the International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS'07). San Diego, California, United States: ACM: 289–300.doi:10.1145/1254882.1254917. Retrieved 9 June 2012.

49. David Fiala; Frank Mueller; Christian Engelmann; Rolf Riesen; Kurt Ferreira; Ron Brightwell (November 2012). "Detection and Correction of Silent Data Corruption for Large-Scale High-Performance Computing" (PDF). fiala.me. IEEE. Retrieved 2015-01-26.

50. Steve Bostian (2012). "Rachet Up Reliability for Mission-Critical Applications: Intel Instruction Replay Technology" (PDF). Intel. Retrieved 2016-01-27.

51. "Read Error Severities and Error Management Logic". Retrieved 4 April 2012.

52. Margaret Bierman; Lenz Grimmer (August 2012). "How I Use the Advanced Capabilities of Btrfs". Oracle Corporation. Retrieved 2014-01-02.

53. Yupu Zhang; Abhishek Rajimwale; Andrea C. Arpaci-Dusseau; Remzi H. Arpaci-Dusseau (2010-02-04). "End-to-end Data Integrity for File Systems: A ZFS Case Study" (PDF). Computer Sciences Department, University of Wisconsin. Retrieved 2014-08-12.

54. "Data Spill Management Guide". asd.gov.au. December 24, 2014. Retrieved January 23, 2015. A data spill is sometimes referred to as unintentional information disclosure or a data leak.

55. The cost of lost data - Graziadio Business Report

56. Leopando, Jonathan (2 April 2013). "World Backup Day: The 3-2-1 Rule". TrendLabs Security Intelligence Blog. Trend Micro. Retrieved 29 April 2015.

57. Connor, Chris (2 November 2013). "Data Loss Prevention: 10 Tips to Prevent Hard Drive Failure". Data Storage Digest. Retrieved 29 April 2015.

58. Summers, G. (2004). Data and databases. In: Koehne, H Developing Databases with Access: Nelson Australia Pty Limited. p4-5.

59. Waksman, Adam; Sethumadhavan, Simha (2011), "Silencing Hardware Backdoors" (PDF), Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California

60. "Data Masking Definition". Retrieved 1 March 2016.

61. "data masking". Retrieved 29 July 2016.

62. "data protection act". Retrieved 29 July 2016.

63. Peter Fleischer, Jane Horvath, Shuman Ghosemajumder (2008). "Celebrating data privacy". Google Blog. Retrieved 12 August 2011.

64. "PCI DSS Definition". Retrieved 1 March 2016.