IDENTIFYING & ANALYZING SECURITY VULNERABILITIES WHILE INTEGRATION OF DATA FROM CLOUD TO SQL DBMS: AN INDUSTRIAL CASE STUDY

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Simranjit Kaur

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Program:
Computer Science

November 2020

Fargo, North Dakota

North Dakota State University
Graduate School

Title

IDENTIFYING & ANALYZING SECURITY VULNERABILITIES WHILE INTEGRATION OF DATA FROM CLOUD TO SQL DBMS: AN INDUSTRIAL CASE STUDY

By

Simranjit Kaur

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Gursimran Singh Walia
Chair

Dr. Oksana Myronovych

Dr. Limin Zhang

Approved:

11-20-2020
Date

Dr. Simone Ludwig
Department Chair

ABSTRACT

Data integration defines propulsion of data from numerous sources into a database system in a way that yields the information representing powerful analytics. Various industries are always working towards keeping their data confidential and in a manner that will let them mine it later & retrieve strong statistics out of it for procuring future profits. This research will discuss about using SaaS (Software as a service) and SQL (Structured Query Language) as a combined model for storing inspections data to achieve the above-mentioned goals that the companies are on the road to. Since SaaS and SQL comes into play, a major part of the research automatically pops up, i.e. security vulnerabilities. Hence, this study details about various security threats while using Cloud storage, SQL database and during the ETL (Extraction, Transformation & Loading) from former to latter, in addition to connecting these database systems to produce an overall secure system.

ACKNOWLEDGMENTS

Foremost, I would like to convey deep appreciation to my advisor, Dr. Gursimran Singh Walia. His persistent guidance and encouragement helped me a lot throughout this beautiful journey of my master's degree. I could not have imagined anyone but him as my advisor, who helped me making the most out of my co-op experience and utilizing all that I did wisely into this research.

I would like to express my warm gratitude towards my supervisors, Les Knudson and Craig Simmons. They appreciated me every single day I worked during my co-op. They assisted me throughout my research journey with all the possible resources and knowledge. I could never thank them enough.

I also want to thank my committee members, Dr. Oksana Myronovych and Dr. Limin Zhang for their valuable comments and feedback, it enhanced my research content and writing to a greater degree. Additionally, having Dr. Simone Ludwig as a department chair is the best thing that happened to me. I really appreciate her kindness.

Last but not the least, I am obliged to have my parents, my brother and all my friends by my side, every single second offering their perpetual support and love. This research ride could only have ended successfully with having them alongside.

DEDICATION

To my parents, my brother and my closest friends for their emboldening, unfathomable love & care and having confidence in me consistently.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SaaS	Software as a Service
SQL	Structured Query Language
NoSQL	not only Structured Query Language
CSP	Cloud Service Provider
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
BI	Business Intelligence
DB	Database
SLA	Service Level Agreement
ETL	Extraction, Transformation & Loading
DBMS	Database Management System
API	Application Programming Interface
DAX	Data Analysis Expressions
SDK	Software Development Kit
IT	Information Technology
PDF	Portable Document Format
JSON	JavaScript Object Notation
CSV	Comma-separated Values
NIST	National Institute of Standard and Technology
GC	Grid Computing
CC	Cloud Computing
UC	Utility Computing
AC	Automatic Computing
URL	Uniform Resource Locator

TTP .....Trusted Third Party

ACPS.....Advanced Cloud Protection System

VM .....Virtual Machine

BGP.....Border Gateway Protocol

AS .....Autonomous System

DoS .....Denial of Service

DDoS.....Distributed Denial of Service

SMP.....Service Management Point

SCP .....Service Control Point

XML.....eXtensible Markup Language

FRS .....Functional Requirement Specification

TVDc.....Trusted Virtual Datacenter

PALM .....Privileged Access Lifecycle Management

VNSS .....Network Security Sandbox for Virtual Computing

SQLIPA.....SQL Injection Protector for Authentication

FHE .....Fully Homomorphic Encryption

VPN.....Virtual Private Network

RD .....Remote Desktop

ML.....Machine Learning

1. INTRODUCTION AND PROBLEM STATEMENT

Data Protection has been a vital concern of all kinds of industries like automotive, computer, electronics, agriculture, etc., for decades since they capture and store tons of data related to facility inspections, safety checklists, hiring paperwork, production reports, financial documents and numerous other data every day. One of the biggest fears that these industries face are the security attacks on their private and confidential information which includes APT (Advanced Persistent Threats), password, insider, and malware attacks. Henceforth, they have been working towards securing their data by utilizing all different kinds of security layers, encryption levels, and a safe DBMS (Database Management System). A secure database/combination of database systems can solve the issue of safeguarding the data and defend important information against these threats.

The objective of this research is to provide a proof-of-concept based on the results from one candidate software company that utilized a combination of cloud and SQL storage to fulfil various requirements of safeguarding vulnerable data from a variety of SQL injections, hacker attacks, and creating useful visualizations and tables using business intelligence applications post loading and querying the data from our database storage systems.

Querying the data is another concept that needs attention. Just capturing the data and extracting it is only 50% of the project being done. The data collected will have all the responses required but the actual usage of those responses is when modified into valuable information. For example, gathering the parameters of production being done every day will only make sense when compared with what was captured the previous day and what will be captured in the future. To know the performance of the employees working in an organization, the data collected has to be studied thoroughly to look into the results obtained on a certain day the employee worked, time

spent to obtain the results, amount of output obtained and other extra productions done while he/she was working. This information, which is the mined data will be utilized towards creating visualizations, building the statistics to understand the benefits and losses suffered, to understand which employee is the best at his/her work, what modifications have to be done in the existing system of conducting inspections for efficient process flow, financial profits gained, etc.

Data can be stored in manually or digitally in industries and both have a proper database warehouse for respective data collection methods. As manual data gathering and tracking is extremely difficult to deal with, digitization and automation are prevalent and preferred in current times. We will talk about having to collect the digital information and using appropriate software tools for their storage, backtracking, future data mining out of it and fetching the best analytics. Implementation of using different software tools for data collection, integrity and integration is the part of this research. A tuple of what this research includes is as follows.

- Security issues encountered while using SaaS for conducting digital inspections in the industries and how to handle them.
- Security vulnerabilities of using SQL for storing private & confidential data in a tabular, structured manner to mine, backtrack data from it later.
- Security attacks faced while the data transfer and integrations process take place from Cloud to SQL.
- Why is it better to use both the storages partially for solving different purposes rather than having to use a single database system?
- How this methodology works and benefits than other research of using cloud and NoSQL (not only SQL), using private cloud or other database systems strategies?

- Steps towards extracting important information from the given data, how to do it in an efficient way using data mining and business intelligence.
- What is the future scope of improvising this suggested research?

Restating the problem statement, data safety in industries, is of utmost importance and is still not 100% achieved due to ever-growing attacks externally and internally. Industries need to make a choice of better database systems and improvise it time and then for an overall protection of their data. Answers to who, what, when, how and where does this problem affects is:

- the industries/companies
- their classified data
- all the time
- due to the external security issues, hacker attacks
- through internet, web attacks, third party attacks, within the facility, etc.

respectively.

2. RELATED WORK

Researches have been going on Cloud computing and the possible security attacks, challenges, and preventive measures for those security issues, how to maintain privacy and safety in cloud computing, security troubles due to SQL injection and how to overcome them with SQL encoding, how to prevent those SQL injections beforehand using attestation techniques and other researches like building a secure database as a service with full encryption, etc.

2.1. Cloud Computing

Cloud Computing is the technology that provides data services much like the internet which provides various network services. It offers an entire lodge of software and hardware with an unlimited storage space, that is where the name comes from, Cloud. It is based on distributed architectonics which gathers server resources to provide metered (pay per use) computing resources and services. It brings convenience and on demand access to networks, storage space, applications, servers with least efforts and CSP's (Cloud Service Provider) interactions (der & Prasad, 2015).

Cloud has a great potential in reducing the set-up cost for computing services, such as software and hardware requirements, which is why companies are increasingly drifting towards cloud solutions. It is storing and accessing the information over the internet without the management by the users, which makes it more attractive. Starting from the server, database, infrastructure, platform to use services, software, application, maintenance cost to almost everything is managed by the CSPs, in addition to being served from any part of the world and being accessed in some other part of the world (Bhadauria & Sanyal, 2012).

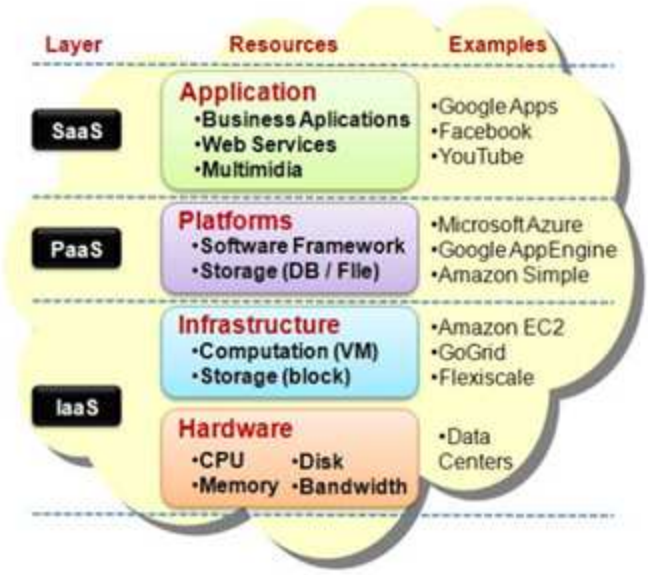

Figure 1. Cloud Architecture (Borges et al., 2012)

Combination of various technologies gave birth to the concept of Cloud. The features and characteristics of cloud realistically comes from the features of several other computing technologies as mentioned below (Springer Gabler Verlag, 2015).

- *Virtualization*: This is a technique when the resources of one physical machine are used by various other machines virtually. Basically, a single unit of hardware serves numerous operating systems with efficiency. Cloud computing uses virtualization for overhead balancing by using dynamic outfitting and dispersion of virtual machines. Since, virtualization helps filtering too, it is very helpful in Cloud owing to security reasons (Lombardi & Di Pietro, 2011).

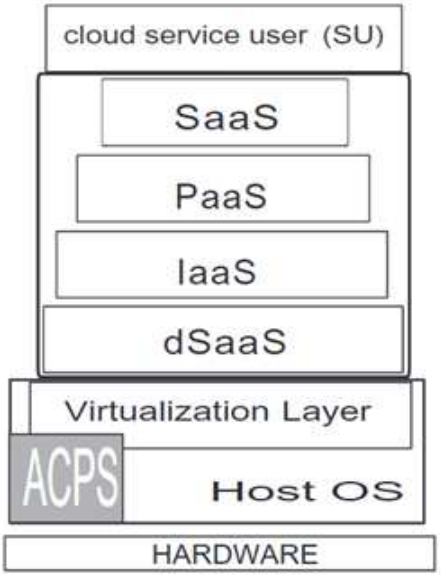

Figure 2. Cloud Layers and the Advanced Cloud Protection System (Lombardi & Di Pietro, 2011)

- *AC (Automatic Computing)*: This is a technique which automatically creates platforms in cloud environment by utilizing SLA (Service Level Agreements) and model driven approach. Automatic computing refers to providing on demand services and resources, automatic maintenance, installations, unlimited memory, flexibility to the users (Borges et al., 2012).
- *GC (Grid Computing)*: This is a technique in which various computers around the world work in parallel to reach a common end goal. Since the operating systems are scattered in numerous locations, that is why it forms a grid (Yu & Buyya, 2006). Some of the most important characteristics of Grid Computing are optimization and simplification, where the former defines spacious data centers and the latter defines user concern towards the application rather than the underlying hardware or

management. Both these features are very useful in Cloud and form the basis of cloud computing (Smirnov, 2005)

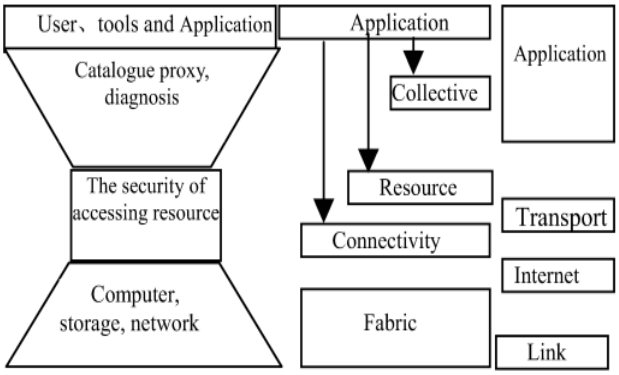

Figure 3. The Layered Grid Architecture and its Relationship to Internet Protocol Architecture (Board & Mitchell, n.d.)

- *UC (Utility Computing)*: This is a technique which works on pay per use kind of model is one of the features of CC (cloud computing). It provides computing resources and services to the users based on whenever they require them and costs them on their usage amount. Its characteristics are very similar to Grid computing and Cloud computing. It is the most compatible, convenience and cost effective approach which is used in cloud computing as well (Patel et al., 2011).

Therefore, an overview of how these different computing services are leveled to work as an overall computing architecture is displayed in the figure below.

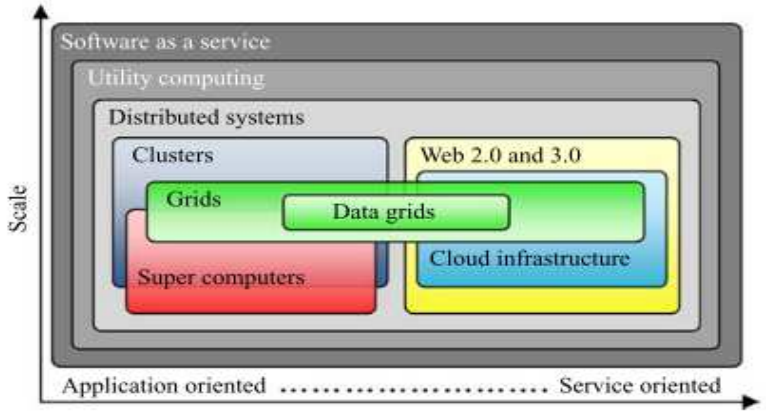

Figure 4. Different Levels of Computing Overview (Patel et al., 2011)

2.1.1. Cloud Service & Deployment Models

Cloud services are classified into three categories, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In SaaS service, the applications, data, operating system, storage, servers and networking, everything is managed and maintained by the cloud service providers and the users have nothing to do in terms of installations and management. PaaS requires management of applications and data, rest is provisioned and sustained by the CSPs. IaaS has a little more to be administered by the users like applications, data, middleware, operating system, and runtime, whereas servers, storage and networking is still fed by the cloud providers (Sun et al., 2014).

Deployment models are categorized into fours kinds namely Public, Private, Hybrid and Community Clouds. Public Cloud is fully hosted and managed by the CSPs, its infrastructure is beyond the company firewall. Users are charged for the resources they utilize; no authentication or access controls can be applied since multiple enterprises work on the same infrastructure provided. Private Cloud, on the other hand can be owned, leased, and maintained by the users themselves, it naturally becomes more secure due to restricted access and network. Hybrid Cloud as the name suggests, offers features of both public and private clouds. It becomes difficult to

manage, reason being complexity in creating and governing a well-constructed hybrid model as it involves two categories, users and CSPs working together. Community Cloud is a shared platform, rarely used model, works on agreement basis between various related business organizations; usually managed by the third party or the organizations involved (der & Prasad, 2015).

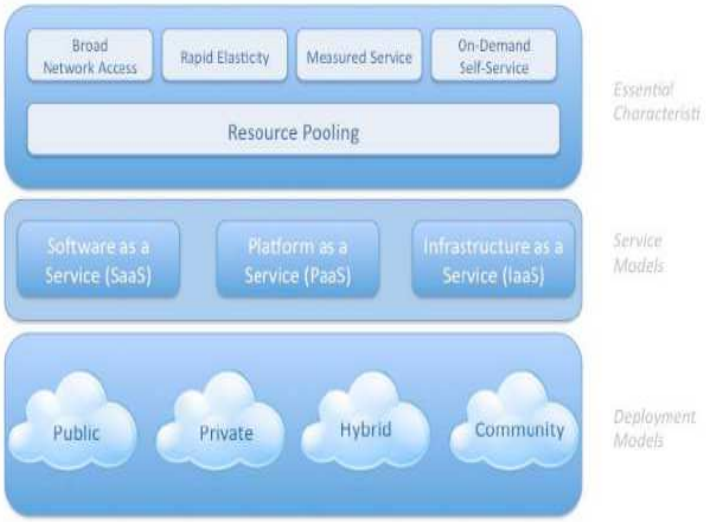

Figure 5. NIST Visual Model of Cloud Computing Definition (Al Jadaani et al., 2016)

2.1.2. Security Structure in Cloud

Various cloud providers install numerous security measures depending on its cloud offering and architecture. Cloud's architecture has two components, multi-layer security and URL (Uniform Resource Locator) filtering (Pushpalatha et al., 2018).

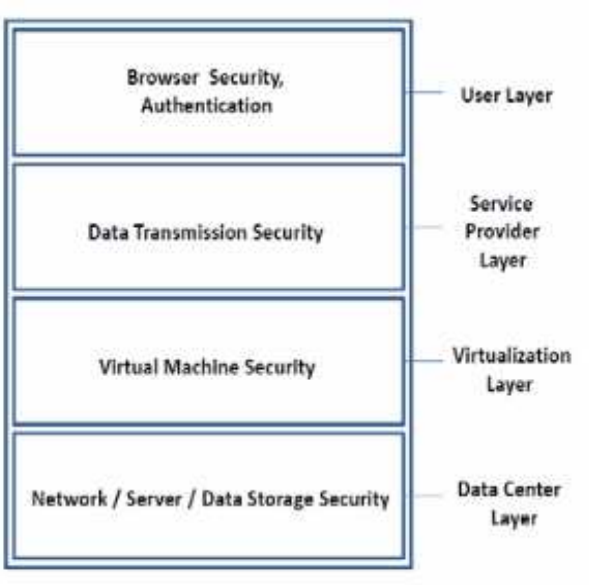

Figure 6. High Level Security Architecture of Cloud Computing (der & Prasad, 2015)

Table 1. Comparative Analysis for Strengths and Limitations of Some of the Existing Security Schemes (Pushpalatha et al., 2018)

Security Scheme	Suggested Approach	Strengths	Limitations
Data Storage Security	Uses homomorphic token with distributed verification of erase-coded data towards ensuring data storage security and locating the server being attacked.	Supports dynamic operations on data blocks such as update, delete and append without data corruption and loss. Efficient against data modification and server colluding attacks as well as against byzantine failures.	The security in case of dynamic data storage has been considered. However, the issues with fine-grained data error location is still remaining to be addressed.

Table 1. Comparative Analysis for Strengths and Limitations of Some of the Existing Security Schemes (Pushpalatha et al., 2018) (continued)

Security Scheme	Suggested Approach	Strengths	Limitations
User identity safety in cloud computing	Uses active bundles scheme, whereby predicates are compared over encrypted data and multiparty computing.	Does not need TTP (trusted third party) for the verification or approval of user identity. Thus, the user's identity is not disclosed. The TTP remains free and could be used for other purposes such as decryption.	Active bundle may not be executed at all the host of the requested service. It would leave the system vulnerable. The identity remains a secret and the users are not granted permission to his requests.
Trust model for interoperability and security in cross cloud	Separate domains for providers and users, each with a special trust agent. Different trust strategies for service providers and customers. Time and transaction factors are considered for trust assignment.	Helps the customers to avoid malicious suppliers. Helps the providers to avoid co-operating/serving malicious users.	Security in a very large-scale cross cloud environment. This scheme is able to handle only a limited number of security threats in a fairly small environment.

Table 1. Comparative Analysis for Strengths and Limitations of Some of the Existing Security Schemes (Pushpalatha et al., 2018) (continued)

Security Scheme	Suggested Approach	Strengths	Limitations
Virtualized defence and reputation-based trust management	Uses a hierarchy of DHT-based overlay networks, with specific tasks to be performed by each layer. Lowest layer deals with reputation aggregation and probing colluders. The highest layer deals with various attacks.	Extensive use of virtualization for security clouds	The proposed model is in its early developmental stage and needs further simulations to verify the performance.
Secure Virtualization	Idea of an ACPS (Advanced Cloud Protection System) to ensure the security of guest virtual machines and of distributed computing middleware is proposed. Behavior of cloud components can be monitored by logging and periodic checking of executable system files.	A virtualized network is prone to different types of security attacks that can be launched by a guest VM (Virtual Machine), an ACPS system monitors the guest VM without being noticed and hence any suspicious activity can be blocked and system's security system notified.	System performance gets marginally degraded and a small performance penalty is encountered. This acts as a limitation towards the acceptance of an ACPS system.

Table 1. Comparative Analysis for Strengths and Limitations of Some of the Existing Security Schemes (Pushpalatha et al., 2018) (continued)

Security Scheme	Suggested Approach	Strengths	Limitations
Safe, virtual network in cloud environment	Cloud providers have been suggested to obscure the internal structure of their services and placement policy in the cloud and also to focus on side-channel risks in order to reduce the chances of information leakage.	Ensures the identification of adversary or the attacking party and helping us find a far- off place for an attacking party from its target and hence ensuring a more secure environment for the other VMs.	If the adversary gets to know the location of the other VMs, it may try to attack them. This may harm the other VMs in between.
BGP (Border Gateway Protocol)	A pretty good BGP architecture has been suggested to check the cases where an Autonomous system may announce itself wrongly as the destination for all the data that is being transferred over that network.	Checks the ASs (autonomous systems) and performs anomaly detection with a response system to ensure that the data does not get routed to the wrong AS. It also gives us the flexibility to run the BGP protocol on some of the Ass towards protecting the entire network.	Vulnerable to Denial of Service (DoS) attacks. This approach only takes care of the routing control messages but does not verify the path that actual traffic follows.

2.1.3. Privacy Concern-Basis & Vulnerabilities

Privacy and security concerns are based on variety of considerations like business, legal, compliance, industrial, geographic etc. The concerns usually are addressed by questions such as, how can the user/organization be confirmed that their data is not accessible to the staff working in

CSP centers & being misused by them or other organizations running their data on the same platform? Secondly, how can the user/organization make sure whether the data has been permanently removed from the cloud once they end up the licensing? Cloud computing is cost-effective, the main reason being using more than one data centers to divide the data load, and sometimes a third data center with the sub cloud provider (Whitley et al., 2013). In that situation, what is the guarantee that data is still safeguarded and kept confidential?

Varied security vulnerabilities that arise considering these concerns are physical, network structure, system software, application, protocol vulnerabilities etc. Physical vulnerabilities refer to the damage to equipment which in this case is the data system centers, naturally or externally. Communication between SMP (Service Management Point) and SCP (Service Control Point) can be a possible attack in network structures since intrusions can come from the SMPs and infect the SCP. In system software, the vulnerability can arise from design and implementation of code in the software, mostly viruses and trojans. In application vulnerability, the attacker attacks the system through authentication mode since it is too week and tries to gain administrator access. Protocol vulnerability comes from using internet, even with an encrypted mode, sometimes interception and eavesdropping is not too difficult (Ragab, 2020).

2.1.4. Security Attacks

Several security issues that exist in cloud are flooding attack, cloud malware injections, browser security issues and XML (eXtensible Markup Language) signature element wrapping.

Attackers or hackers use different methods to intrude into the system. Sometimes even with web security on and encryptions, they try to upload trojans to the cloud systems. They use element wrapping when trying to attack the system using swapping the target element with the value they

want (Al Jadaani et al., 2016). A diversity of attacks exists like DoS, distributed DoS, google hacking, CAPTCHA breaking, Cookie poisoning, hidden field manipulations, dictionary attacks.

2.1.4.1. Security-Threats Analysis in SaaS

Since SaaS provides on demand services, for instance enterprise resource planning, customer relationship management, software configuration management, emails, etc. and responsibility & handling of every demand is under the control of CSPs, privacy and security concerns are raised, third party attacks and keeping the infrastructure secured consistently (Hashizume et al., 2013).

2.1.4.2. Security-Threats Analysis in IaaS

IaaS gives resources, servers, storages, and networks for the users to work in which are accessed through the internet. Since users have control, so they can handle the services allocated to them on their VMs. If they don't have a security hole in their VM monitors, and have better control over the threats through internet, then they are better than the other two models (Hashizume et al., 2013).

2.1.4.3. Security-Threats Analysis in PaaS

PaaS includes securing the platform including the runtime engine and thus face miscellaneous security challenges. For example, PaaS provides multiple applications that can be hosted in the cloud, which affects the software development lifecycle and the security. So, the applications must be upgraded in a timely manner. These upgradations can affect the security of data. Hence, the developers need to be educated data legal issues and how to deal with their security while maintaining the updates (Hashizume et al., 2013).

2.1.4.4. Security Issues in Public Cloud

Confidentiality, integrity form the basis of security and having public cloud, makes it difficult to achieve because there is no control of user son the CSP security practices. Moreover, public clouds work on shared infrastructure and thus multiple organizations share space with each other. During the data sharing, processing, or creating, chances of data leakage become very high. If the cloud providers use third party vendors to provide services, security risks raise if proper SLAs have not been defined and ensured (Bhadauria & Sanyal, 2012).

2.1.4.5. Security Issues in Private Cloud

Virtualization methodology is popular in private cloud. Users have full control over the system, so while working on the virtual machines, they do interact with other VMs too, but there is a possibility that they accidently interact with the VMs they are not supposed to. Hence proper encryption is still required even if using private cloud. Internet attacks are mostly reduced but sometimes attacks originating from within the organization are easily forgotten (Bhadauria & Sanyal, 2012).

2.1.5. Mitigation Strategies

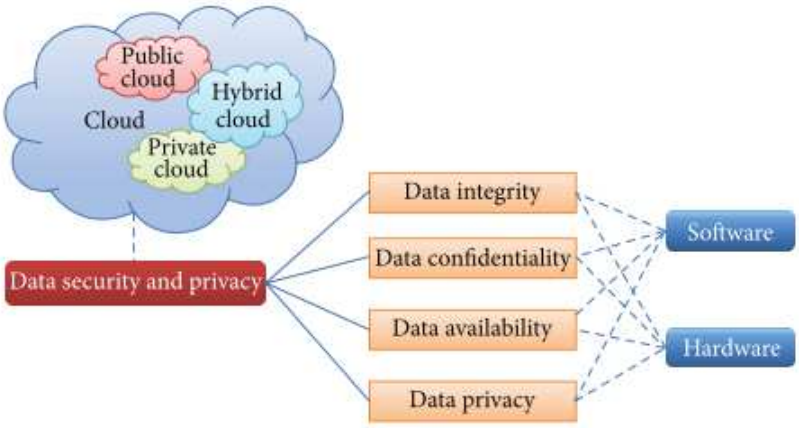

Figure 7. Organization of Data Security and Privacy in Cloud Computing (Sun et al., 2014)

Digital signatures, RAID (Redundant Array of Inexpensive Disks), monitoring mechanisms help maintain data integrity. Confidentiality of data can be done using homomorphic encryption, using distributed storage, encrypted search in the database, hybrid technique, data concealment & deletion confirmation. Data availability can be achieved with reliable storage agreement, and hard disks. Data privacy can be intruded by service abuse, i.e., the attackers increase the cost of the services and use the data. Fraudulent resource consumption can be reduced using identification and fraud detection techniques. Another way to maintain data privacy is averting the DoS attacks (Sun et al., 2014).

For the security regarding multimedia file sharing in cloud, there has been recent research completed which uses algorithms including six phases to ensure the secure transfer of media in cloud. The phases image encryption, cloud interface with authenticated image access, and finally image decryption. The pseudo codes are designed to give the image a random sequence number to encrypt it. Once encrypted, proper unique credentials are required to access that image during the cloud interface and then during the process of decryption, the image is converted back to original after reassembling its pieces (Chidambaram et al., 2019).

Table 2. Relationships between Threats, Vulnerabilities, and Countermeasures (Hashizume et al., 2013)

Threats	Vulnerabilities	Incidents	Countermeasures
Account or service hijacking	Insecure interfaces and APIs	An attacker can use the victim's account to get access to the target's resources	Identity and access management guidance Dynamic credential

Table 2. Relationships between Threats, Vulnerabilities, and Countermeasures (Hashizume et al., 2013) (continued)

Threats	Vulnerabilities	Incidents	Countermeasures
Data scavenging	Data related vulnerabilities	Data from hard drives that are shared by several customers cannot be completely removed	Specify destruction strategies on SLAs
Data leakage	Data related vulnerabilities Vulnerabilities in VMs, VM images, VM networks	Necessary steps to gain confidential information from the other VMs co-located in the same server as attacker are discussed above	FRS (Functional Requirement Specification) techniques Digital Signatures Encryption Homomorphic Encryption
Denial of service	Insecure interfaces and APIs Unlimited allocation of resources	An attacker can request more computational resources, so other users are not able to get additional capacity	Cloud providers can force policies to offer limited computational resources
Customer-data manipulation	Insecure interfaces and APIs	Some examples, such as SQL, command injection, cross-site scripting	Web application scanners
VM escape	Vulnerabilities in Hypervisors	A zero-day exploit in the hyper VM app that destroyed about 100,000 websites	HyperSafe TCCP TVDc (Trusted virtual datacenter)

Table 2. Relationships between Threats, Vulnerabilities, and Countermeasures (Hashizume et al., 2013) (continued)

Threats	Vulnerabilities	Incidents	Countermeasures
VM hopping	Vulnerabilities in VMs and Hypervisors	A study demonstrates security flaws in most VM monitors	
Malicious VM creation	Vulnerabilities in VM images	An attacker can create a VM image containing malware and publish it in public repository	Mirage
Insecure VM migration	Vulnerabilities in VMs	Attacks against migration functionality of the latest version of Xen and VMware virtualization products	PALM (Privileged access lifecycle management) TCCP VNSS (network security sandbox for virtual computing)
Sniffing/Spoofing virtual networks	Vulnerabilities in virtual networks	Sniffing and spoofing virtual networks	Virtual network framework based on Xen network modes: bridged & routed

2.2. SQL Database Systems

SQL is a text language used to interact with the relational database systems. The unit of execution in SQL is a query, which is a set of statements. When executed and run, it produces a result set for the defined query. The result is basically manipulations that one wants to be performed on the contents/data of the SQL table into a desired display.

Web applications use various database systems, like SQL to store, insert, modify, and retrieve their data. An important aspect to be considered here is the malicious attack that can be made on SQL too, usually called SQL injections (Anley, 2002).

2.2.1. SQL Inoculations

SQL inoculations are the queries or series of SQL statements that hackers try to inject by crafting inputs to database driven applications. There are various advanced SQL injection techniques using which hackers bypass the defenses against SQL attacks. There can lead to threats like system fingerprinting, DOS, and stealth of private data (Janot & Zavarsky, 2008).

2.2.1.1. Basic login injections

Attacker enters OR 1=1 -- (double dash) in login queriers. Therefore, an expression: *SELECT * FROM users WHERE login='' AND password=''* becomes *SELECT * FROM users WHERE login='' OR 1=1 – 'AND password=''*. The double dash leads to access to all users as it comments the password section and OR 1=1 makes the expression result to true even if is false for one of the conditions (Janot & Zavarsky, 2008).

2.2.1.2. Strings without Quotes

Developers protect the system escaping the single quotes and using replace function instead to avoid SQL insertions. For example, *function escape(input); input = replace(input, " ' ", " ' ' "); escape = input; end function.* But, if the hackers wish to insert, they can use 'char' function. For example, *insert into users values(666, char(0x63) + char(0x68) + char(0x72) + char(0x69) + char(0x73), 0xffff)* (Anley, 2002).

2.2.1.3. Length Limits

Limiting the length of the input data can help restricting SQL attacks but still there are some possibilities of other kind of insertions. For example, if the attacker uses *Username:*

';shutdown-- or if *drop table <table name>*, this will simply shut down the SQL server instance (Anley, 2002).

2.2.1.4. Audit Evasion

SQL server includes auditing interface, which can help watch what has been done by the attackers if enabled. But if the hacker is able to append the string, say *sp-password* , then even if the administrator can track something has happened, it will still result this: *--'sp-password' was found in the text of this event. – the text has been replaced with this comment for security reasons*, which means the attackers are able to hide the injections they have performed (Anley, 2002).

2.2.2. SQL Encryption Mechanism

SQL injections can be taken care of using some encryption techniques. Various methods have been proposed for avoiding and controlling SQL insertions like Analysis and monitoring for neutralizing SQL-Injection attacks, which checks the queries at runtime and identify hotspots and declare a query as valid or malicious at the runtime. Some techniques for SQL encryption are discussed below, which have been proposed in previous researches (Ali et al., 2009).

2.2.2.1. SQLIPA (SLQ Injection Protector for Authentication)

This technique uses hash values for both usernames and passwords. Hash values are first created and stored for both the parameters and the when the query is executed, then the hash values are calculated during the runtime and verified with the stored hash values to check if they are the same. The query for this looks like: *SELECT * FROM User_account WHERE Username_Hash_ Value ="Username_Hash_Value" AND Password_Hash_Value ="Password_Hash_Value" AND Username='' OR 1=1 – AND password ='Password'* (Ali et al., 2009). The architecture of SQLIPA has consists of three components as explained in the figure below.

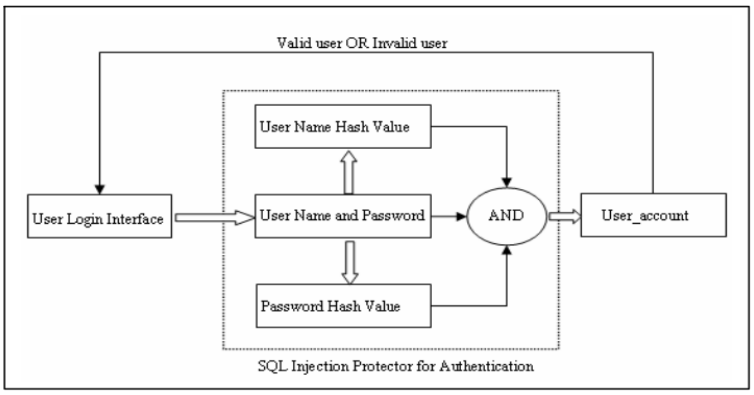

Figure 8. Architecture of SQL Injection Protector for Authentication (SQLIPA) (Ali et al., 2009)

2.2.2.2. FHE (Fully Homomorphic Encryption)

Fully Homomorphic Encryption consists of three algorithms, evaluate, encrypt, and decrypt. The concept of FHE says that it can evaluate any function and process any query. However, the working not very simple, it has various issues (Mani et al., 2013).

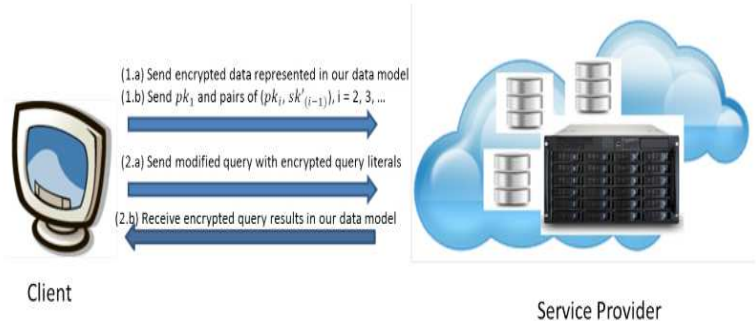

Figure 9. Proposed Architecture for Processing Encrypted Data (Mani et al., 2013)

This model consists of the following (Mani et al., 2013):

- programming language control structures can be used on the variables that the unencrypted literals are assigned to
- no programming language control structures can be used on the variables that the encrypted literals are assigned to

- function calls can be made with both encrypted and unencrypted literals as parameters by the service providers
- service providers can evaluate fixed, combinatorial circuit on encrypted values, and can obtain encryption on any literal using encryption keys.

3. PROPOSED WORK & METHODOLGY

This research study conducted relates to the needs and demands in the industry. Working real time in industries, make a lot of difference in comprehending how things work out. The concept began with modifying the existing reports/manual inspections like facility inspections, safety checklists, hiring paperwork, production reports, financial documents, railcar inspections, sample requests, into digital ones. Gradually, the concept started to turn up bigger with addition of database, security vulnerabilities while working with different databases. Digitization of reports into automated audits and inspections was making more sense when used for different purposes than just cutting down the paperwork. For instance, numerous other benefits like:

- data mining,
- data backtracking,
- powerful analytics,
- strong dashboards,
- interconnected tables between different areas in the facility,
- timesaving,
- getting rid of big data warehouses,
- more efficient and managed workplace,
- storing, defining, and managing the data,
- good investment returns,

coming along with just digitization gave this operational research much more meaning. A roll out plan was prepared following the ideology above.

3.1. Roll Out Plan Preparation

This operations research had four divisions, but every division included one common research part, i.e. identifying, and analyzing various security attacks and trying to reduce them by designing a secure model. The SaaS cloud security attacks (APT, password attacks), possible SQL system attacks (SQL injection, malware & insider attacks), threats while working with the report generating dashboards, like Power BI (Business Intelligence) application, are the vulnerabilities that will be discussed, analyzed and worked upon (countermeasures taken) in detail in this thesis.

The threats that can cause issues in the data collected in using Cloud application could be tackled by pushing pieces of important information safely through the API gateway using an authorization/API token into the encrypted SQL database. Therefore, using both the database management systems partially, will help build more protected area for the company's data reason being, data collected in pieces in two different DBMS will become information when combined.

Henceforth, data collection can be done more efficiently using applications that own Cloud storages owing to the benefits of unlimited storage, offline saving the data, backup features, and auto maintenance & management services, whereas data defining, storing, manipulating, backtracking, retrieving and most importantly shielding can be best performed using SQL queries and database management systems. Additionally, the reason for splitting the data as mentioned above is, just dumping the data by gathering it through inspections in Cloud would not make sense to a third party or attackers unless visualized with its second half in SQL that completes it with important statistics and analytics. Similarly, pieces of information in different structures will remain way more protected than all the data in the same data storage structure.

3.1.1. Inspection Conducting Software

First part of the research was identifying an appropriate software for performing inspections on every single report/form that must be filled out in the facility and jumping into the first testing phase of evaluating that software's usage. Evaluation of the software was necessary as there were certain requirements that the software must fulfil to make this process of digitization work efficiently. Some features that the product should have offered included:

- support multiple operating systems like iOS, Android, and Windows,
- reports/templates building ease,
- offline data collection, data integration,
- timestamps and geolocation,
- document management to some extent like pre-fill inspections,
- archiving,
- templates menu and search bar with different filters like bookmarked, completed, inspection date, region, site.

The chosen one for performing audits was iAuditor Inspection Software & Application (Preusler, 2020) owing to strong templates building feature, public libraries for accessing templates, APIs (Application Programming Interface), administrator controls, third-party integrations, annotations, dashboards which were the necessities since companies own numerous kinds of forms (templates) starting from food safety reports, production reports, hiring catalogs to the critical financial charters. Once this part of the rollout plan was released, then it was time to dive into the search pool of the best fit database systems for data manipulation.

3.1.2. Database Systems

There is a pool of database systems like MySQL, Apache, Drizzle, InterBase, Cloud (IaaS, PaaS, SaaS), Oracle, to make a choice from that best fits into this study's requirements. One DB (Database) offers some impressive features which may not exist in the other one.

But there is one way to resolve this conflict of choosing one over the other, that is choosing multiple models and using them together, with some sort of connectivity. This brought up the idea of using two very renowned DB systems, Cloud Storage and Microsoft SQL, a perfect combination of super secure model because of the following features:

- high performance programming in SQL
- high security in SQL using permissions on tables, procedures, and views
- robust transactions management in SQL
- scalability & flexibility in SQL
- open-source programming in SQL
- data manipulation in SQL
- triggers in SQL
- client server execution & remote database access in SQL
- broad network access in Cloud
- multi-tenancy & resource pooling in Cloud
- on-demand self-service in Cloud
- high security provided by Cloud
- automated and easy maintenance in Cloud
- pay as you go policy in Cloud

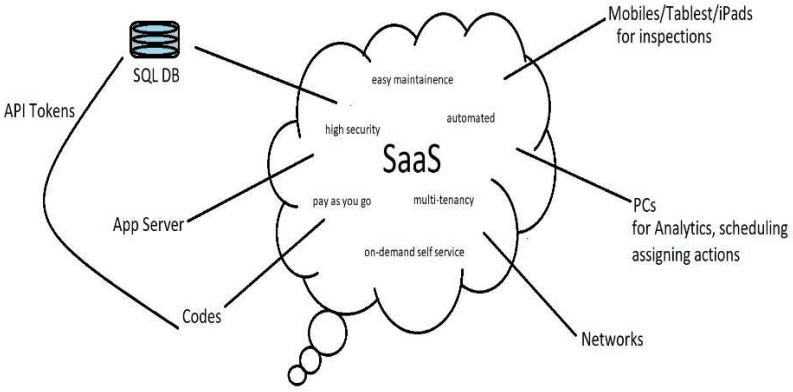

Figure 10. SaaS Services & SQL DB Connectivity

Since iAuditor is a SaaS product, so it has Cloud storage and its features. But, to build a trustworthy system for an industry's classified data, there must be an enterprise database system for backups as well as access to confidential data only for specified users. As opposed to researching, the best combination matching all these requirements and criteria stated in bullets above, was the connectivity between SaaS cloud and SQL enterprise database system. The methodology was generating API tokens and getting to push the data from iAuditor into enterprise SQL DBMS.

3.1.3. Analytics & Mining Tool

The last part of this research was then having to choose strong, dashboards, software tools for fetching powerful reports, which will benefit the industries statistics in future improvisations and gains.

Searching for tools like Power BI and Crystal Reports was conducted and Microsoft Power BI (Smith, 2018) surpassed other tools because of its distinctive features like:

- transform and model data

- share data, dashboards & reports with other Power BI users
- create workspaces (where colleagues collaborate to create reports & dashboards)
- flexibility to publish data across the organization without receipts being required to buy licenses individually
- scalability and high performance
- maintain BI assets on-premises with BI server
- provides real time information
- Cortana integration
- artificial intelligence like image recognition create ML models, etc.

It helped understanding how well an industry is doing in the recent years. It offered unending services like getting information about the data even from past 10 years. It enabled setting up strong DB tables connectivity between different forms like safety inspection, food safety inspection, production report, metal detection, magnet checks, etc. from different areas in the facility. Data could be mined anytime, backtracking of information about a particular product or commodity could be done anytime and anywhere. Querying the tables with pivot/unpivots using BI query editor, creating visualizations from the queried tables, transforming and modeling data using DAX (Data Analysis Expressions) was exactly what was expected from an Analytics tool.

4. ARCHITECTURE USED & OUTCOMES

This research was focused on two major tasks. First, identifying and utilizing a new combination of two very popular database management systems, SQL, and Cloud for managing confidential information, like audits conducted, production reports containing commodity information, metric tons produced, etc. Second, discovering newer security threats and possible vulnerabilities besides the already existing ones while working on the real time data. This study is accompanied by the evaluations, validations and results of the actual, synchronal, coeval inputs captured while performing this research.

4.1. Template Building

Creating templates for the forms on a SaaS platform can be quite difficult and cumbersome. iAuditor uses SaaS services, does not require extensive hardware, and typically uses subscription-based model. It provides and manages every single step from application, data, runtime, middleware, operating systems, virtualization, storage to networking, yet it lets the user customize the platform, dashboards, application services, themselves.

This study started with this very first phase of exercising template designing. The templates were designed keeping in mind that thousands of forms that were existing from past many years were required to be designed digitally in such a way that it reduces the inspection time to at least half of the time it took to complete the reports manually. The duration of audits conducted should cut down to the minimum.

Therefore, pre-filled choices, drop down menus, failed responses possibilities, mandatory fields, scoring audits, multiple choice options, attaching notes, adjoin action button, etc., these factors were reinforced within the templates. Some figures related to conduction of the above-

mentioned steps on my iAuditor account describe the functioning of SaaS infrastructure in customizing one's own SaaS workspace while working with the live concurrent data.

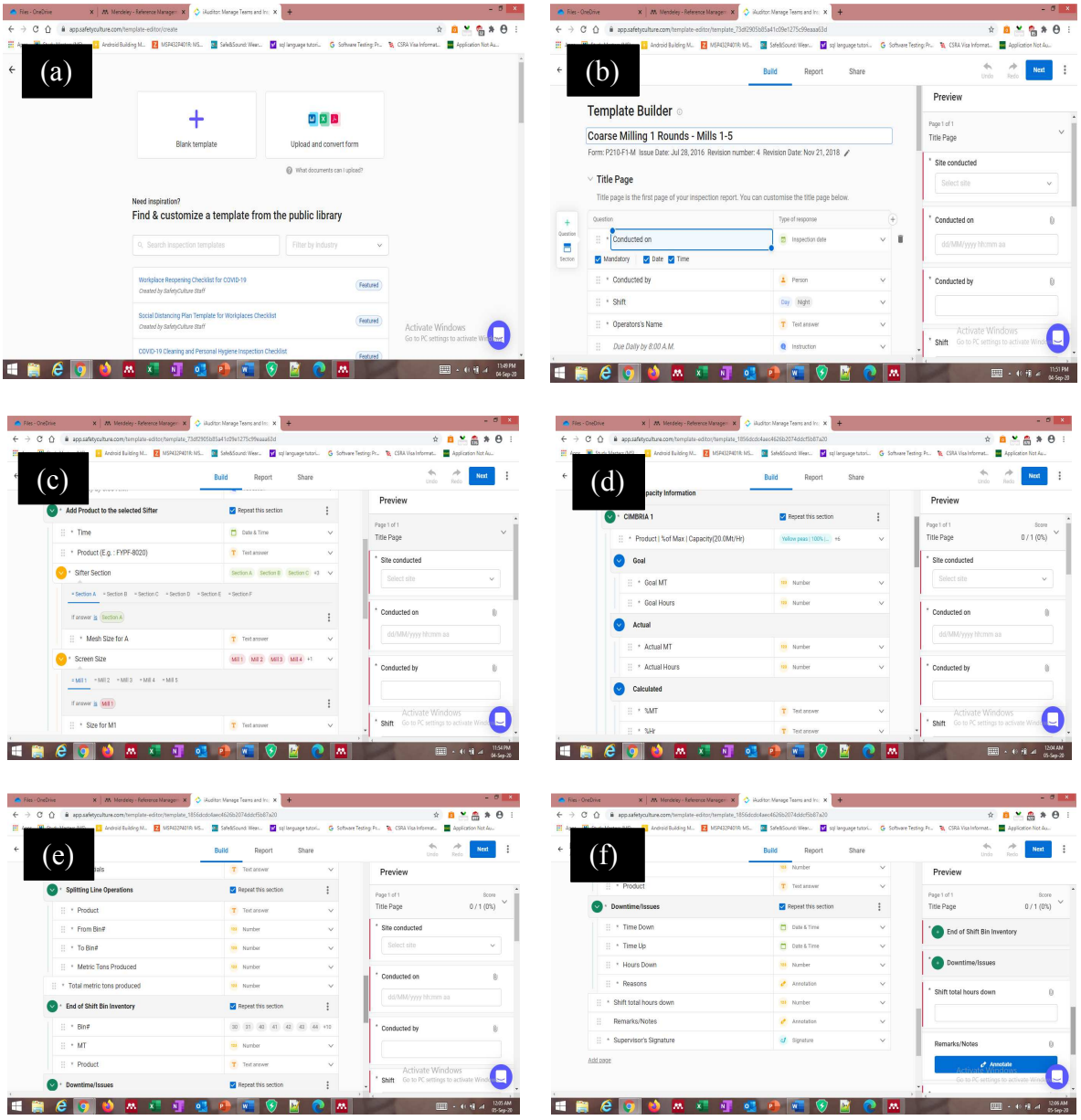

Figure 11. (a) Constructed New Templates, (b) Added Inspection Pages, (c) Appended Logics to the Questions, (d) Designed Templates Using Regular Sections, (e) Improvised Utilizing Repeat Sections, and (f) Affixed Befitting Responses to the Questions

4.1.1. Access Controlled Dashboards

iAuditor features access-controlled dashboards, which means the dashboard that appears in different user accounts display the tabs and objects inside them based upon the admin levels the accounts hold. Accounts for Operators, Supervisors, Area Leads, Plant Supervisor, Division Head were set up that held different levels to access to the final reports. This includes:

- admin access to edit templates, inspections, integrations, work on analytics, which can only be done by the Template Author and Template Owner, me
- access to manipulate audits/inspections, which can be controlled by Audit Owner and Audit Author, operators
- access to create schedules, archive inspections, and verify them, which can be controlled by the Audit Owner, account that have the audits shared with them, supervisors, and the area leads

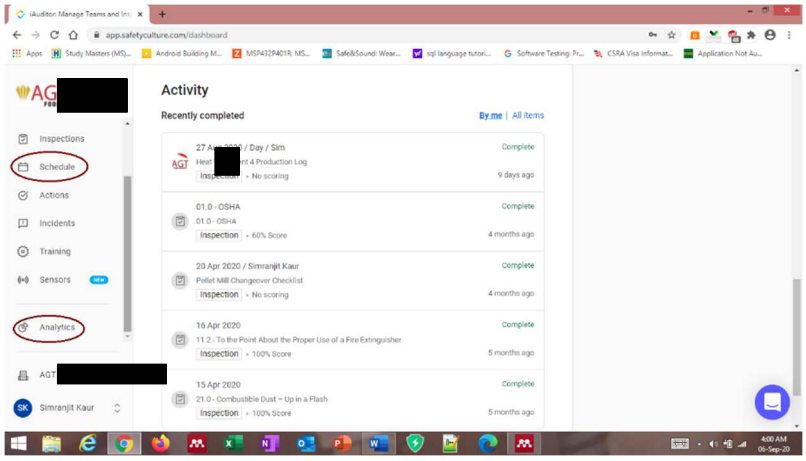

Figure 12. iAuditor Dashboard

4.1.1.1. Manage Schedules

Schedules could be created for inspections to be conducted at certain times of the day, repeating every week, biweekly, monthly, yearly and with other possible customizations including

the due date. This is a very useful feature that helps looking up the inspections scheduled for, manage schedule, check the status of inspections, search scheduled inspections by filtering them, pause or delete schedules as per the admin controls discussed previously, tracking the number of missed inspections, who missed them and when were they missed.

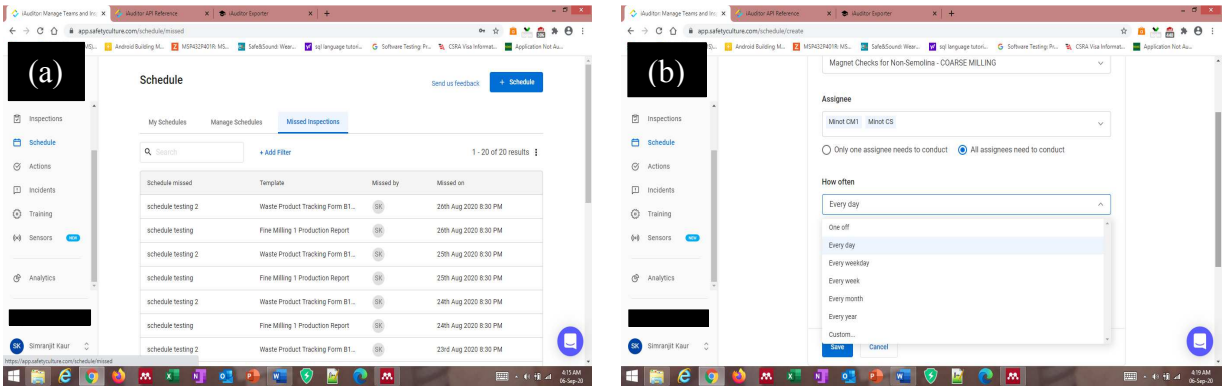

Figure 13. (a) Schedule Audits, (b) Customize Schedules

4.1.1.2. Analytics Tab

Analytics tab offers various outputs like, total number of inspections that have been conducted in an organization, number of failed inspections (which depends upon number of failed responses while performing audits), the time duration for every single audit along with the average time taken toc conduct inspections, helps identifying completed, incomplete and archived audits, number of unique people who conducted the inspections, display performance in the form of bar and line charts for certain week, month, etc., exporting inspections in so many formats. Different options for exporting the work done are Word, PDF (Portable Document Format), JSON (JavaScript Object Notation), CSV (Comma-separated Values), charts.

Timestamps is a very important and beneficial feature of iAuditor. The inspections let the users input date and time while conducting an audit if the template is designed to collect responses for those parameters. Even after the user inputs the fields, iAuditor is designed in such a way that

it still captures the actual time zone owing to the automatic timestamp feature. This helps perceiving the true performance of the workers and making modifications to the working system.

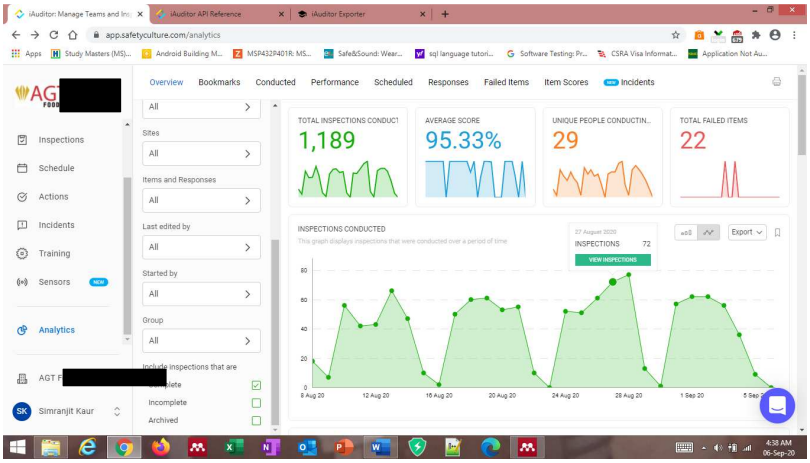

Figure 14. Manage Company's Work Using Analysis Tool

4.2. Data Integrations

Data Integration is a process of consolidating the data residing in various sources like Cloud and SQL storages in the current experimentation, and creating a single, unified visualization out of it. It is a significant procedure that commercial companies and scientific domains follow for their data. The answer to the question as to why data integration is important is because it provides data enrichment, management of business data, easy feeding of consistent & cleansed data into the data warehouses. There are various techniques of data integration like application-based, manual, middleware, physical. In the study, integrations followed were offered by iAuditor itself.

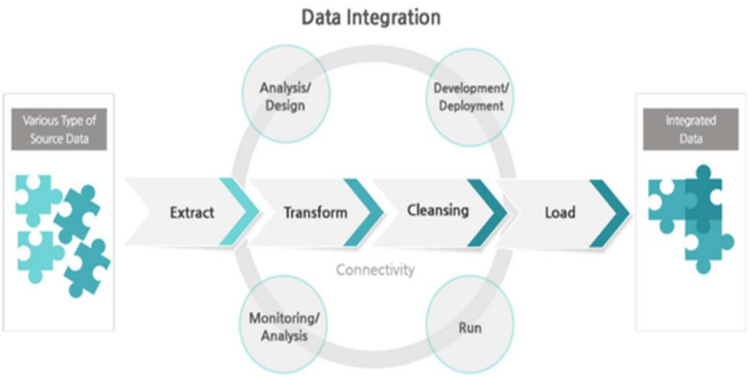

Figure 15. Integrations Process Flow ("Integrations Process Flow", n.d.)

4.2.1. Inspections Exportation

iAuditor offers various types of integrations. It includes automatic exportation of the inspections to SharePoint, Google Drive, SMS, OneDrive, Email, Dropbox, Evernote, online spreadsheets, like Google Sheets. The integrations can be done using different ways in this software.

4.2.1.1. Zapier-Based Integrations

This technique requires Zapier account to create Multi-step Zaps. It allows automatic inspections sharing with the users or groups on the site chosen for an inspection. It provides bulk transfer of CSV formats of the performed audits.

Zapier works a little setup at the start but after that automatically functions to export inspections. It follows three steps, namely creating zap, triggers, and actions on the inspections that need to be pulled out. During the setup, API token is required too. Since data manipulations and getting statistic visualizations are important factors required in this experimentation, hence API based integrations using iAuditor exporter tools were conducted.

4.2.1.2. API tokens-Based Integrations

This technique requires generating an API token, which is specific to each user account. It expires after 30 days of inactivity and one will have to regenerate a new one, with a maximum allowance of up to 10 tokens. Organization admin account for iAuditor owns all the inspections and so it was used to pull the data out instead of using APIs for every single user. After generating an API token, iAuditor exporter tool was used to transfer the inspection data into the enterprise SQL database.

The tool uses Python scripting to configure the files. Hence, Python was installed and the script along with the configuration files were downloaded. Different methods are available for installing the scripts like manually or using GitHub. The script is then run by running the downloaded config file on installed Python SDK (Software Development Kit). There are certain settings that can be done when using the exporter tool. Manual settings like, exporting incomplete, complete, and archived inspections as per the organization requirement. Merging the rows can be set up to true or false, depending upon the requirement. After running the script and setting up these conditions, there are two options to pull data into the SQL database, either export data in an existing table or if not, python creates one for you. The results were letting python create the database table itself which is a best practice too as per recommendations.

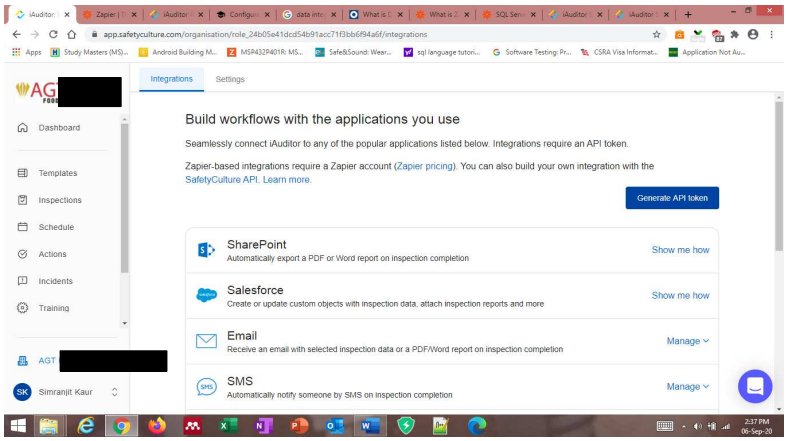

Figure 16. API Data Integrations

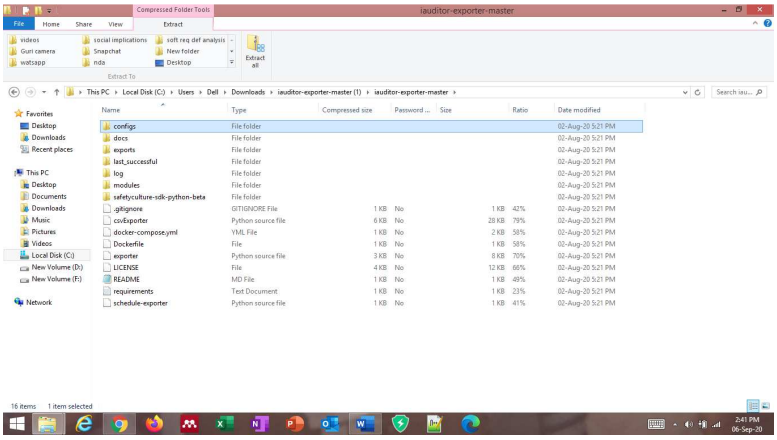

Figure 17. Python Script Containing Configurations, Exports, Log Files and Modules for Connecting iAuditor Cloud with SQL Database

4.3. SQL Querying

SQL tables started populating with auto refreshing of 15 minutes. The columns were looking like just the data dumped into it. It executed result with more than 14 million rows with inspection data in it, each entry per row. As real time data is always enormous, so querying the data becomes more complex. The data will only become informative if proper querying is performed on it. The table included column names related to the fields that were used while creating templates. Since it had many ID columns, like Audit ID, Template ID, Date, and many

other useful columns like the date the inspections were started, date completed, date last modified, audit duration, archived, labels and their responses, Item ID, template names, audit author, owner, and many more.

This giant table needs to have a primary key as a combination of two or three candidate keys, like Audit ID, Date, Item ID to be able to output unique columns without recurrences. The database was created on the enterprise SQL server. The access to that server was only provided to specific IT (Information technology) people, and only using corporate remote desktop.

4.3.1. Main Table, iauditor_data

This table outputted more than 14M rows with all the data that was collected when iAuditor was tested and put into use from the start till date. The data looked scattered as it simply exported and dumped everything into the table. For example, the labels are the questions set up while building forms in iAuditor. The table had responses with so many blank entries, as it included the headings, dynamic sections, lists, etc. So, the table needed cleansing, in terms of useful data representation.

The main table, iauditor_data contained the data from the all the facilities of the company. Therefore, for Minot specific data, another table was created with rows up to certain date to play with. The query testing for cleaning, organizing, and maintaining the information coming out of the data gathered was performed on that table, iauditor_minotdata.

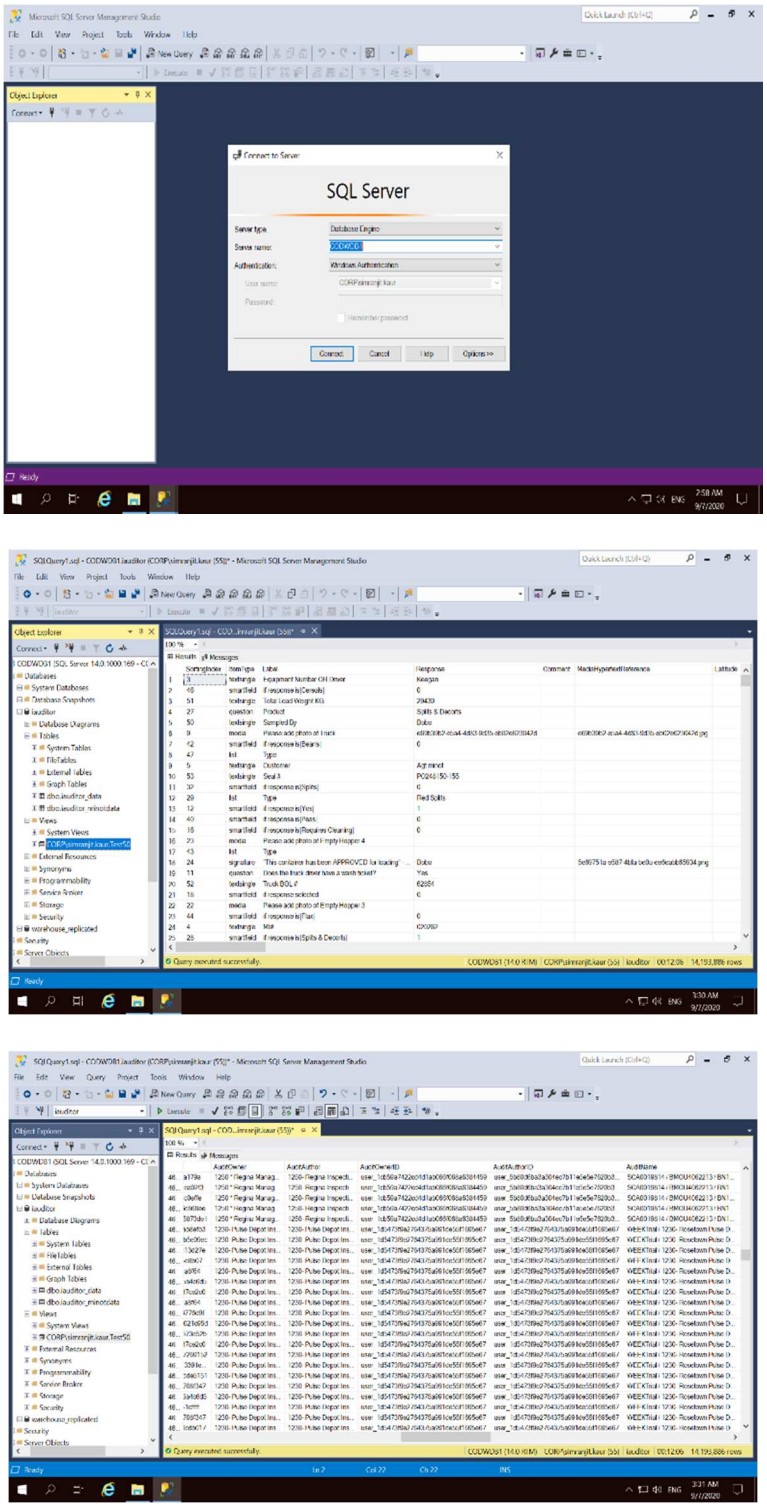

Figure 18. Screenshots of Connecting to SQL Server, Executing Queries to Pull Out the Data from Inspections Performed in the Organization

4.3.2. Specified Table, iauditor_minotdata

A separate table was created for working on small data, for the Labels and Responses to give useful results like understanding the performance of workers, progress made, failed responses, calculation of average audit durations over a period, etc. SQL query for pulling some data:

*SELECT*FROM iauditor_minotdata WHERE TemplateAuthor='Simranjit Kaur'*

order by DateStarted desc, SortingIndex

4.3.2.1. Creating Views

Creating Views will help cutting down some columns that are unnecessary while bringing out useful information from the given data. View query was written as follows:

CREATE VIEW [Test50] AS

SELECT

AuditID, ItemID, DatePK, SortingIndex,

ItemType, Label, Response, AuditDuration,

DateStarted, DateCompleted, DateModified,

TemplateName, AuditOwner, AuditName,

ParentID, RepeatingSectionParentID, Archived

FROM

iauditor_minotdata

WHERE

TemplateAuthor='Simranjit Kaur' AND CONTAINS (AuditName, 'Sendy')

The view gave out some results with a improvements against the original data.

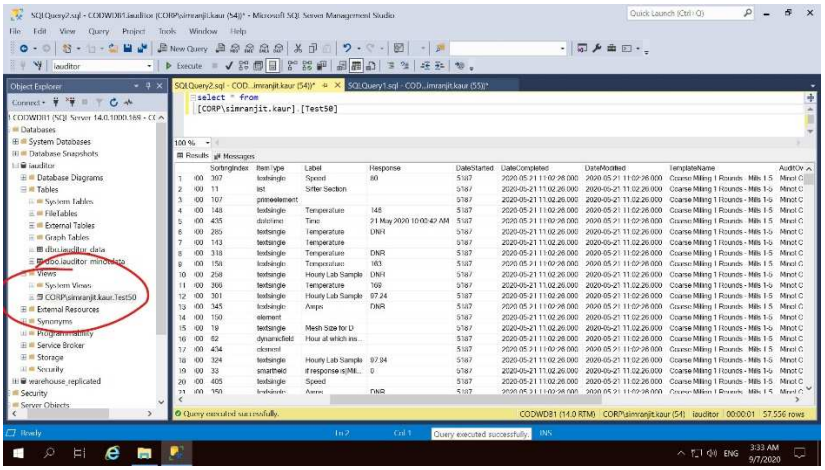

Figure 19. Snapshot of Results from Creating View

4.3.2.2. Pivoting/Unpivoting Query

Pivot query could help improvise the output to a greater degree. The was the why pivoting was given a try as the questions and responses to those questions, if transformed with each other would reduce the null values, display data with more accuracy, reduce the number of columns to rows which is how pivot works.

*SELECT*FROM [CORP\simranjit.kaur].[Test50]*

SELECT Label, Response1, Response2, Response3, AuditName, DateStarted

FROM

(

SELECT Label, Response, AuditName, DateStarted,

'Response' + CAST (ROW_NUMBER () over (Partition by Label order by Label) as varchar (10) as ColSqn

FROM [Test50]

) test2

Pivot

(

MAX (Response)

FOR ColSqn

IN (Response1, Response2, Response3)

) PIV

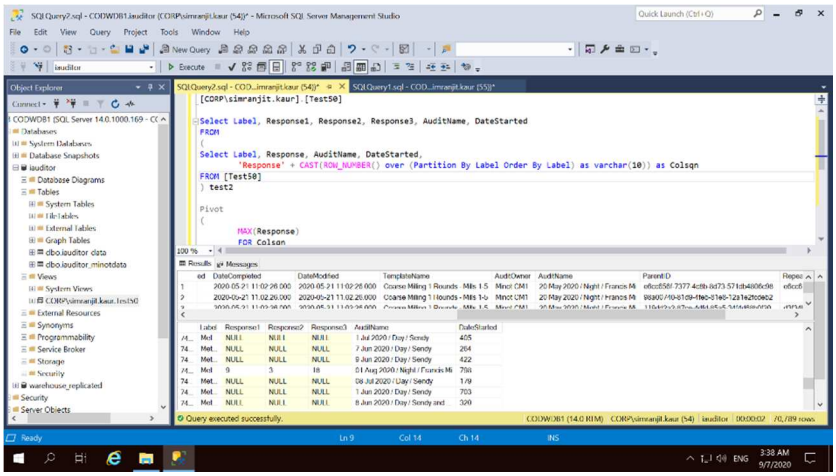

Figure 20. Output from Using Pivot

4.3.2.3. Dynamic SQL

Outcomes from pivot query led to rethink about using other queries that would deal with the number of rows and columns that are not defined. Pivot/unpivot rows and columns works best when the number of rows and columns are defined, and the data is quite simple. A little more research explained that using dynamic SQL queries can transpose undefined number of rows and columns. Sample of dynamic SQL query that can used to resolve the issues from results above.

DECLARE @cols AS NVARCHAR(MAX),

@query AS NVARCHAR(MAX)

Select @cols = STUFF (SELECT',' +QUOTENAME(ColumnName)

From yourtable

```
        Group by ColumnName, id
        Order by ID
FOR XML PATH (''), TYPE
).value('.','NVARCHAR(MAX)'),1,1,'')
Set @query=N'SELECT'+@cols+N' from
(
   Select value, ColumnName
   From yourtable
) x
Pivot
(
Max(value)
For ColumnName in ('+@cols+N')
) p'
Exec sp_executesql @query;
```

4.4. BI Query Editing

Power BI was used as it is the most popular business intelligence tool used by the industries to author in the kitchen of BI, function modeling using relationships and DAX, create powerful analytics, beautiful visualizations and finally be able to use online dashboards to share the visualizations with managers, division head, and various branches of the organization.

4.4.1. Kitchen of Power BI, Query Editor

Power Query Editor allows viewing your table by loading it from SQL database and transforming it to fit in the requirements of a useful table. It offers various settings to the data like

filter rows, unpivot columns, reorder columns, transpose, mathematical functions, run R and Python scripts, merge queries, advance editing, set up header rows, pivot columns, etc. After working on different requirements, close and apply the changes from the editor to the final table.

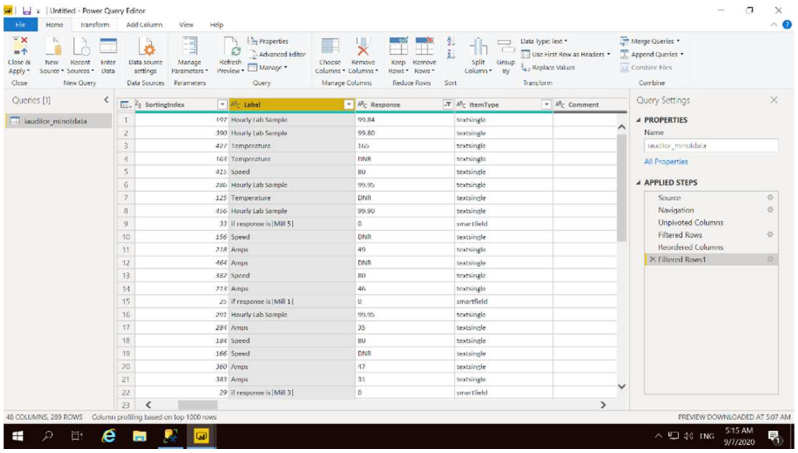

Figure 21. Power Editor Displaying Query Performed on the Right Side

The table shows null values removed, by working on one template at a time. But on pivoting the rows even after working on the single template at a time does not provide desired result and gives many nulls. The reason could be the restricted settings available for using pivot.

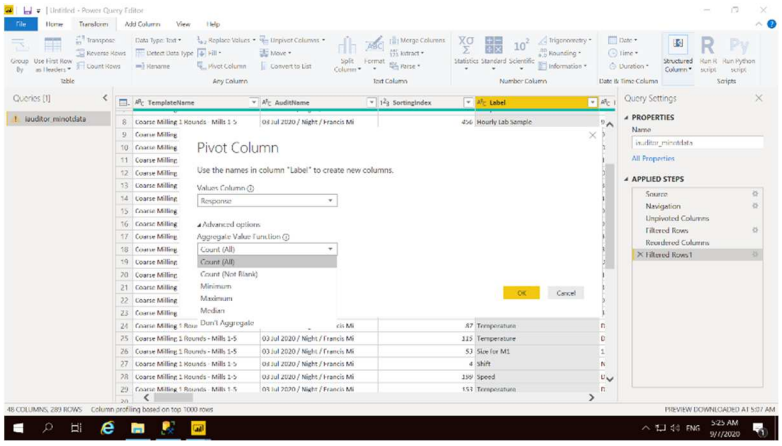

Figure 22. Limited Pivot Settings

4.4.2. Heart of Power BI, Relationships & DAX Model

The limited features available in editor can be improved using DAX expressions and relationship modeling, creating new tables, and trying to modify relations to change the table view. It helps utilizing different statistic functions, variance (), average (), sum (), maximum (), count (). For using arithmetic expressions, it was required have same data types of columns that will have DAX expressions applied on them. Relationships were created by designing new tables and segregating data in different tables into smaller tables to have better connectivity between different column items. Relationships can be managed by providing cardinalities between different tables like, one to one, many to one, one to many, many to many, etc.

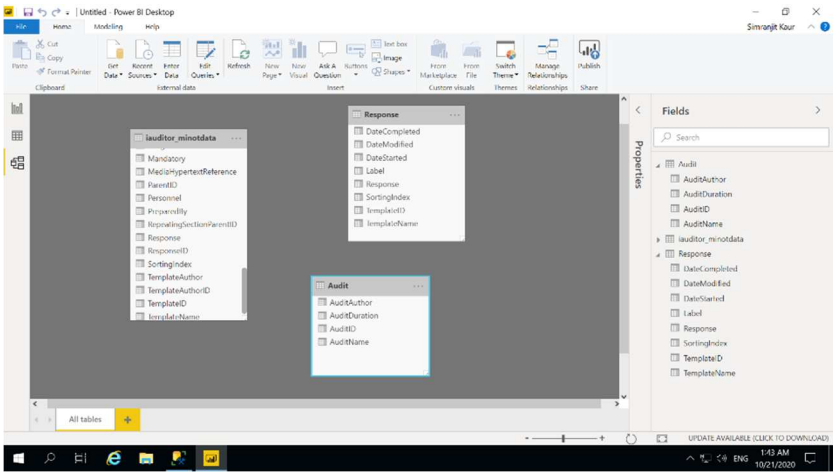

Figure 23. Relationships & Modelling

4.4.3. Visualizations

Visualizations act as a dashboard where the organization can have a quick look at the statistics of how it is performing, what changes are required to benefit and bring progress to the corporate, what all has been achieved, etc. It helps display insights that have been discovered from pulling the data. Visuals can be created in report view and can further be pinned to online dashboards.

Visualizations pane offer various different displays of data, like line chart, column chart, pie chart, format slicer, gauges, single-numbered cards, heat maps, scatter charts, bubble charts, and numerous other options for creating views for our data reports. A snapshot of visuals pane is as below.

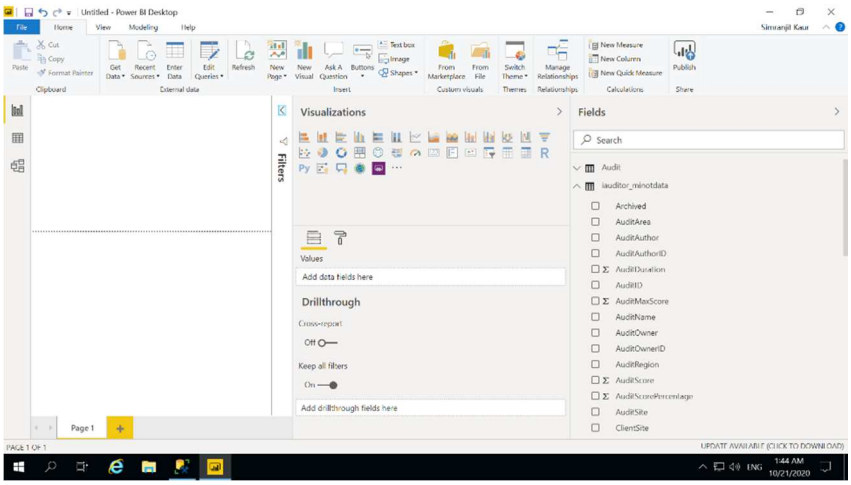

Figure 24. Visuals in Power BI

5. SECURITY VULNERABILITIES ANALYSIS & COUNTERMEASURES

Numerous security vulnerabilities are confronted when working with Cloud and SQL. This research focuses on studying various ongoing threats not having proper solutions to avoid them, what strategies were used to prevent them, varied attacks possible while transferring data from SaaS cloud to Microsoft SQL DB, how can we prevent SQL insertions, countermeasures on all the threats encountered.

5.1. Cloud Security Threats & Defenses

Despite all the benefits of cloud, many enterprises still hold back from adopting cloud computing, owing to the data security issues. Cloud infrastructure can be complex and lead to making errors and exposing the organization's confidential and private data. Additionally, data management in cloud computing is provided by a third party. Data storage, management, transferring in and out of the cloud is a major security consideration.

5.1.1. Insecure APIs & Solutions

API generation was required to integrate the data from iAuditor into enterprise SQL. APIs can be easily exposed to the attackers if not kept secured and protected. The APIs exposure can cause DoS and DDoS (Distributed Denial of Service) attacks. The important thing to keep in mind is proper management of APIs. For example, if in use then authenticated accurately, and if not should be disposed correctly.

The solutions to insecure API that was adapted in this study was only using an API of the admin account and the credentials not being distributed to other colleagues, operators, supervisors, developers. The developers were working on the database directly without caring about the API used. An admin account on iAuditor has access to all the data that can be captured using all other iAuditor accounts associated with the organization.

5.1.2. Data Loss & Remedies

One of the important aspects in using cloud that should be taken care of is the data being lost. Even with the accidental deletion of data, it can be lost permanently. It is sometimes possible to recover the data back, but only if requested within a limited period. The other reason for data loss is hacker's attack. Data loss is also caused by data alteration if the attacker deletes sensitive data, and if the hackers encrypts data with strong encryption keys to execute their malicious activities.

Data loss can be prevented by having geo-diversified backups, having offline backups, considering softwares that prevent data loss. Current study had one backup in SQL, and second back up in company's AllTrack and DAX alongside. AllTrack is a web-based system used for and by the internal stakeholders for producer/vendor's management (activities, samples, quotes), tracking commodity's delivery & departures. It is an ERP (Enterprise Resource Planning) fully developed by the organization to analyze the products & commodities. It is hosted on the servers at enterprise's data center. Dynamics, on the other hand focused on the financial aspect of the relationship with the producer/vendor. This backup system leaves almost no chances of data loss.

5.1.3. Compliance Violations & Control

Cloud offers the benefit of easily accessing data within an organization and it becomes difficult to track who is accessing the data and what can they do with it. This shared responsibility model can cause compliance violations.

In this research different account holders were used to prevent this threat. iAuditor is a SaaS service and has everything to be managed by the service providers. The organization's working expected to have different user accounts with different levels of access to the data. For example, the operators were given a separate account so that they do not have access to analytics

and statistics being captured, and only supervisor's account could access that. The supervisor account was given access to the web version to have a control over the statistics of how the work was being completed, in what average duration, with what outputs. The operators were given access to the application version of iAuditor so that they can only perform basic data gathering, data sharing, raising incidents and actions.

5.1.4. No Control over End-User Actions

The end users, like operators and supervisors in the current study, work with the cloud services and the company is not aware of how they are using the services, in which case enterprise's proprietary information becomes vulnerable to attacks and insider threats. For the insider attacks, there is no requirement of breaking through VPNs (Virtual Private Network) or firewalls to gain access to the data.

For preventing such threats, one iAuditor admin account was used to build all the templates and shared with the specified area for performing audits to keep track of how inspections are being performed and the services are being used, the employees were given proper training explaining them the vulnerabilities such as phishing and malware. Routine check on the audits and account usage was done to avoid any attacks.

5.1.5. Data Breaches

Data Breach is the leakage of private and safeguarded data of the organization from the cloud servers and storage on the internet without their permission. Cloud malware injection attacks are one of the most critical ones. Hackers use this method to insert malicious code to an end-user system. This can create serious security implications, like deadlock, Dos, DDoS, etc.

The methods that could be used to avoid data breaches is encrypting the data, using a secure cloud computing service, and taking benefits of the security features provided by the service. In

the current study, a secure cloud computing is being used. SafetyCulture, iAuditor premium account provides SaaS service with the best security. For example, encrypting the customer data, actively monitoring, and testing the IT environment for vulnerabilities, applying due diligence, etc.

5.1.6. System Vulnerabilities

System vulnerabilities includes:

- lack of logging mechanism
- not closing the connections properly
- absence of input validation on the user input.

Solutions to system vulnerabilities is better encryption which as the above point mentions is provided by the SaaS cloud service, we are using in the current study in SafetyCulture iAuditor.

5.2. SQL Threats

SLQ Intrusions are vital to handle for obtaining a secure system. Various insertions like login injections, string without quotes attacks, audit evasions, length limit attacks are possible while working with SQL. Attackers can insert set of SQL statements having -- (double dash), use OR 1=1 statement to enter the database by injecting these queries in the data inputs. They use code injection techniques to attack the data driven applications.

Advanced SQL insertions need to be controlled to protect the private data from being breached and lost. It is one of the major security vulnerabilities to be taken care of. Apart from the insertions, there are system vulnerabilities as well like, not closing the database connection properly, failed open error handling. In this research, these vulnerabilities have been considered and proper security was implemented to avoid and prevent hacker attacks.

5.3. Remedies Against SQL Inoculations

Previous studies have taught how Analysis and monitoring for neutralizing SQL-Injection attacks, SQLIPA and FHE can be used towards data protection and building a safe & secure database. But in the current study, certain security measures were taken to safeguard both stored data and its flow into the SQL. Several security layers were applied within the organization work environment to protect the data. The countermeasures used are discussed below.

5.3.1. RD (Remote Desktop)

This technique was used for all the developers to work on the same remote desktop, so that activities they perform like, login attempts, why was a connection not established, for what duration the connection was created, etc., can be traced. Every developer was given access to the SQL database server through a terminal that directs him/her to a remote computer. An RD gateway server for the organization established the connection. Moreover, if you have established a connection and after entering the remote session you are not performing any activity, then it automatically closes the connection and if one attempts to rebuild the connection, credentials are required again.

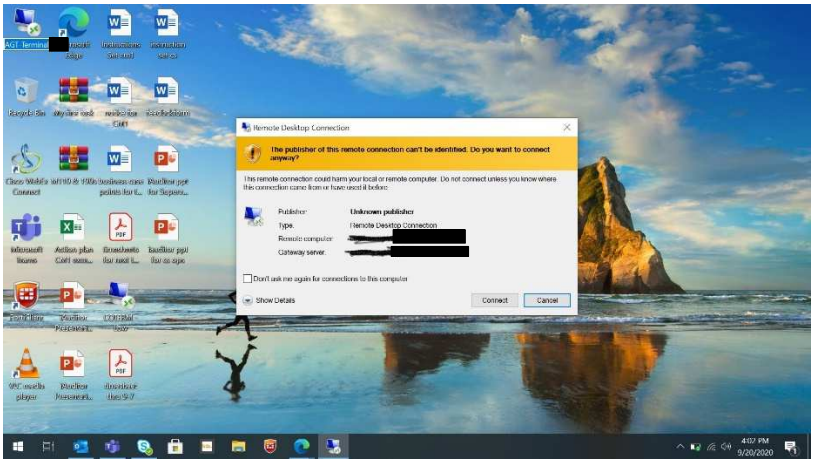

Figure 25. Establishing RD Connection through Terminal

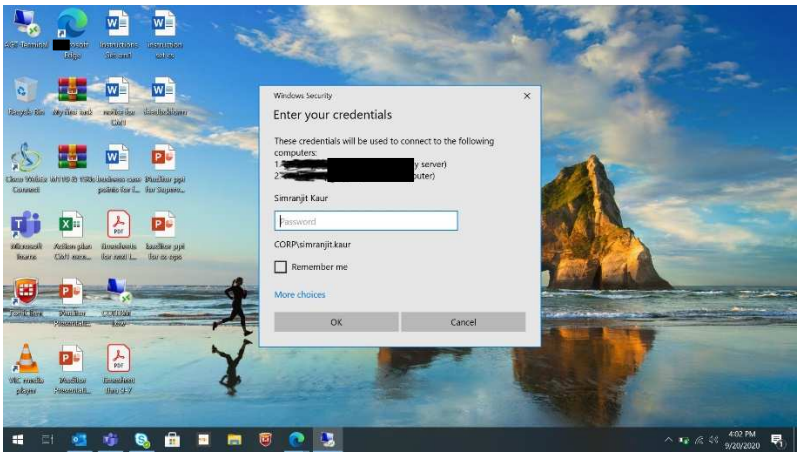

Figure 26. Authentication Required Prior to Connecting

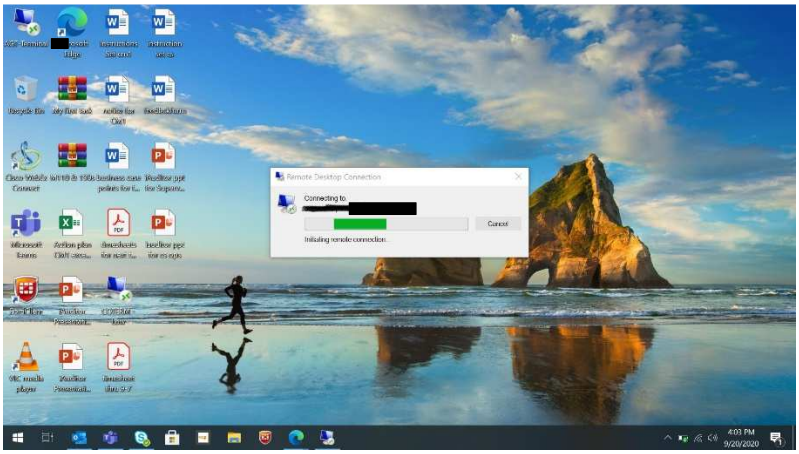

Figure 27. Connection Established Successfully

5.3.2. Using Corporate Network

Remote connection could only be built if entry to the remote session is made through the corporate network provided by the organization. The wired connection can initiate the process of connection to the gateway.

Additionally, no Wi-Fi connections, if even it is organization's Wi-Fi, no connection to the remote desktop can be made. This was another security layer that helped maintain the database security and prevent attacks.

5.3.3. SQL Server Credentials Access

Now after establishing the connection with the remote desktop, when a connection to SQL server is required, it could only be connected through windows authentication. This meant that no other developers other than the admin is given access to the SQL server credentials for upgrading the security. This step is well explained in Figure 18 as well.

5.3.4. FortiClient VPN

In the event of COVID-19, remote working became prevalent. But this does not mean that a compromise with data security can be made. So, the idea of VPN was implemented. FortiClient application helped a VPN connection with the gateway terminal. To connect to organization gateway, first a VPN connection was made and then the terminal connection was initiated. Now if the VPN is down, the server connection to the remote computer automatically shuts and reconnects back only after connecting through the VPN again. This created another layer of security in the work environment.

5.3.5. Pull-Data Approach, AllTrack & DAX

Another very useful approach that was used for ensuring security was the pull approach. In order to avoid exposure in case of iAuditor, SaaS cloud service, certain structures were used. The pull approach was preferred where the organization's systems request information from outside sources like iAuditor rather than letting them push data into enterprise system.

In case, push method is required, then small isolated systems were used for temporarily holding the data. The main database was not populated directly, rather staging database was used to hold the data when being pushed from outside sources. This helped preventing intruders to enter the main system. There are several similar push-pull systems built in the organization to secure the

data. This includes company's AllTrack and DAX, where the data is stored and secured from numerous penetrations.

DAX uses codes for the data being used in the facility. The forms being directly worked upon by the employees uses simple names to define the raw materials. But DAX uses codes for those products while dealing with the inventories, commodity trading, transferring data between warehouses, and hiding important formulae used to check on the metric tons produced out of the given raw material. Some of the forms built in iAuditor were also being developed in AllTrack, like the production forms, incoming trucks report, etc. AllTrack has better access levels like, ready for production, completed, approved, verified to work with, which can be very beneficial for crucial reports like production, incoming railcars summary, bin inventories. AllTrack has both development and testing and live environments to work in, thus making it a very efficient.

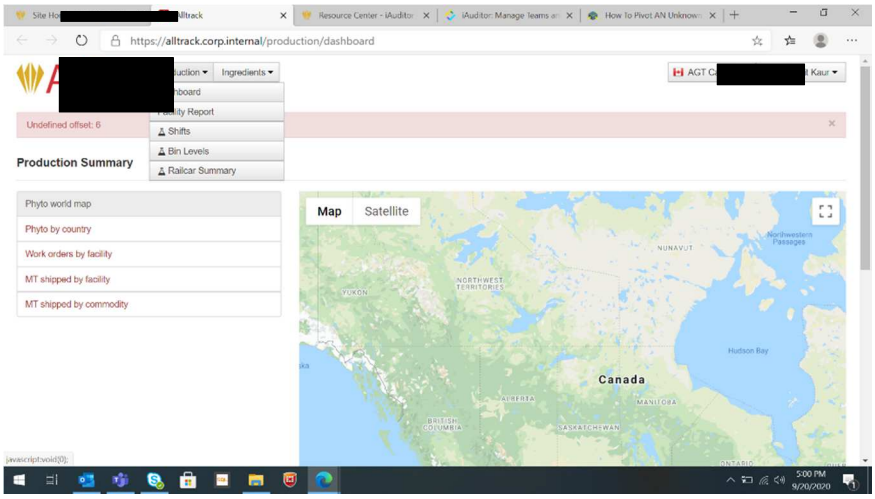

Figure 28. AllTrack Live Environment

6. CONCLUSION

This research came up with useful conclusions. The two main issues that were focused upon in this study were, combining SaaS cloud and SQL database to work together as a secure model for storing, defining, manipulating, retrieving, backtracking the data, and the security vulnerabilities experienced during this process flow. Upon working with the models, it was found out that the two storage systems can work very well together to capture and safeguard the private and confidential information of the enterprise systems. Eventually, the data can be more precisely handled using the organization AllTrack, shared drives, enterprise DAX models, Telogis, to make the workflow efficient, super secure and easy to deal with.

The security vulnerabilities that came up were successfully tackled using the multiple push-pull systems, security layering using corporate network, remote computers, FortiClient, AllTrack leveled accesses, limited server credentials access and Dax codes.

The focus of this research was accomplished, i.e., experimenting to verify if a hybrid combination of SQL and Cloud would work better than the existing combinations such as cloud & NoSQL. Furthermore, the existing security attacks and approaches to handle them were studied thoroughly and a new approach towards making the system even more secure was initiated, which actually produced very good results in terms of security to the sensitive data of the organization.

7. FUTURE SCOPE

As there is always a room for improvement, there are some future recommendations to this study as well. The integrations of data from cloud to SQL is currently very difficult to handle, reason being the tables generated have millions of rows with each row treated as unique even with the same outputs and so many rows coming up with null values. Query processing on such data is very difficult. The solutions could be either having iAuditor produce more efficient tables, or in-depth complex querying of the present data itself, dynamic SQL querying, etc.

This research study is never-ending, it is an ongoing experiment as newer security threats can appear in future, updates made to the existing systems can make the systems vulnerable to other non-existing threats at present. Cloud is itself growing every day, there can be possible discoveries which could help resolve the issues that will still occur after the multi-layered secure gateways been applied in the study. Similarly, can also lead to newer threats than the existing ones. Both the topics of this research are prone to changes in the future with the ever-growing attacks, discoveries of better database systems and new technologies coming into play.

REFERENCES

Al Jadaani, S., Al Maliki, M., Al Ghamdi, W., & Hemalatha, M. (2016). Security issues in cloud computing. *International Journal of Applied Engineering Research*, *11*(12), 7669–7671. https://doi.org/10.4018/978-1-4666-0879-5.ch707

Ali, S., Rauf, A., & Javed, H. (2009). SQLIPA: An authentication mechanism against SQL injection. *European Journal of Scientific Research*, *38*(4), 604–611.

Anley, C. (2002). Advanced SQL injection in SQL server applications. *NGSSoftware Insight Security Research*, 25. http://alsouza.googlecode.com/svn/trunk/Monografia/subsidios/sqlinjection/Advanced SQL Injection.pdf%5Cnhttps://sparrow.ece.cmu.edu/group/731-s11/readings/anley-sql-inj.pdf

Bhadauria, R., & Sanyal, S. (2012). Survey on Security Issues in Cloud Computing and Associated Mitigation Techniques. *International Journal of Computer Applications*, *47*(18), 47–66. https://doi.org/10.5120/7292-0578

Board, E., & Mitchell, J. C. (n.d.). *Lecture Notes in Computer Science*.

Borges, H. P., De Souza, J. N., Schulze, B., & Mury, A. R. (2012). Automatic generation of platforms in cloud computing. *Proceedings of the 2012 IEEE Network Operations and Management Symposium, NOMS 2012*, 1311–1318. https://doi.org/10.1109/NOMS.2012.6212068

Chidambaram, N., Raj, P., Thenmozhi, K., Rajagopalan, S., & Amirtharajan, R. (2019). A cloud compatible DNA coded security solution for multimedia file sharing & storage. *Multimedia Tools and Applications*, *78*(23), 33837–33863. https://doi.org/10.1007/s11042-019-08166-z

der, R. R., & Prasad, R. V. V. S. . (2015). Cloud Computing Research : Challenges and Security Issues. *International Journal of Computer Trends and Technology*, *30*(3), 157–161. https://doi.org/10.14445/22312803/ijctt-v30p128

Hashizume, K., Rosado, D. G., Fernández-Medina, E., & Fernandez, E. B. (2013). An analysis of security issues for cloud computing. *Journal of Internet Services and Applications*, *4*(1), 1–13. https://doi.org/10.1186/1869-0238-4-5

Janot, E., & Zavarsky, P. (2008). Preventing SQL Injections in Online Applications: Study, Recommendations and Java Solution Prototype Based on the SQL DOM. *Owasp*, 15. file://localhost/Users/akiezun/Documents/Papers/Janot/Janot2008Preventing SQL Injections in Online.pdf

Lombardi, F., & Di Pietro, R. (2011). Secure virtualization for cloud computing. *Journal of Network and Computer Applications*, *34*(4), 1113–1122. https://doi.org/10.1016/j.jnca.2010.06.008

Mani, M., Shah, K., & Gunda, M. (2013). *Enabling Secure Database as a Service using Fully Homomorphic Encryption: Challenges and Opportunities*. 1–13. http://arxiv.org/abs/1302.2654

Patel, A., Seyfi, A., Tew, Y., & Jaradat, A. (2011). *Comparative study and review of grid , cloud , utility computing and software as a service for use by libraries*. *3*, 25–32. https://doi.org/10.1108/07419051111145145

Pushpalatha, V., Sudeepa, K. B., & Mahendra, H. N. (2018). A survey on security issues in cloud computing. *International Journal of Engineering and Technology(UAE)*, *7*(3.34 Special Issue 34), 758–761.

Ragab, A. R. (2020). *Vulnerabilities of Intelligent Network System*. *7*, 756–761. https://doi.org/10.35940/ijitee.G5761.059720

Smirnov, M. (2005). Lecture Notes in Computer Science: Preface. In *Lecture Notes in Computer Science* (Vol. 3457).

Springer Gabler Verlag. (2015). Cloud Computing: State of the Art and Security Issues. *Gabler Wirtschaftslexikon*, *40*(2).

Sun, Y., Zhang, J., Xiong, Y., & Zhu, G. (2014). Data Security and Privacy in Cloud Computing. *International Journal of Distributed Sensor Networks*, *2014*. https://doi.org/10.1155/2014/190903

Whitley, E., Willcocks, L., & Venters, W. (2013). Privacy and Security in the Cloud: A Review of Guidance and Responses. *Journal of International Technology and Information Management*, *22*(3), 77.

Yu, J., & Buyya, R. (2006). *A Taxonomy of Workflow Management Systems for Grid Computing*. 171–200. https://doi.org/10.1007/s10723-005-9010-8

https://www.omnisci.com/technical-glossary/data-integration. Figure 15. Integrations Process Flow (“Integrations Process Flow”, n.d.)