



Data Quality and Performance Analysis of Salesforce.com as a Customer Relationship Management Platform: Strategies for Improving Business Processes

Department of Software
Engineering

Term Project

by

Merve Emre

ORCID 0009-0007-2470-4976

Academic Advisor: Prof. Dr. Aytuğ Onan

August 2024

Data Quality and Performance Analysis of Salesforce.com as a Customer Relationship Management Platform: Strategies for Improving Business Processes

Abstract

This study aims to analyze the data quality and performance of Salesforce.com as a Customer Relationship Management (CRM) platform. The research examines the impact of data management challenges on the platform and focuses on strategies proposed to enhance the performance and data quality for Salesforce.com users. Analyses were conducted considering criteria such as data security, accuracy, integrity, and accessibility, with the goal of ensuring that businesses can achieve the highest efficiency from their CRM implementation.

Keywords: Salesforce.com, data quality, performance analysis, data management, data integrity, customer relationship management

Müşteri İlişkileri Yönetimi Platformu Olan Salesforce.com'un Veri Kalitesi ve Performans Analizi: İş Süreçlerini İyileştirmek İçin Stratejiler

ÖZ

Bu çalışma, Salesforce.com'un bir Müşteri İlişkileri Yönetimi (CRM) platformu olarak veri kalitesi ve performansını analiz etmeyi amaçlamaktadır. Araştırma, platform üzerindeki veri yönetimi sorunlarının etkilerini incelemekte ve Salesforce.com kullanıcılarının performansını ve veri kalitesini artırmak için önerilen stratejilere odaklanmaktadır. Analizler, veri güvenliği, doğruluk, bütünlük ve erişilebilirlik gibi kriterler göz önünde bulundurularak gerçekleştirilmiştir. İşletmelerin bu uygulamalardan en yüksek verimliliği sağlamaları amaçlanmıştır.

Anahtar Sözcükler: Salesforce.com, veri kalitesi, performans analizi, veri yönetimi, veri bütünlüğü, müşteri ilişkileri yönetimi

Table of Contents

Abstract	ii
Öz.....	iii
List of Figures	vi
List of Tables..	vii
List of Abbreviations.....	viii
1 Introduction.....	10
2 Problem Statements	15
2.1 Data Privacy and Security	15
2.2 Data Integrity and Quality	16
2.3 Data Migration	19
3 Definition of Terms	24
4 Strategies and Best Practices	29
4.1 Data Profiling and Cleansing.	29
4.1.1 Use Case of SOQL for Data Cleansing	31
4.1.2 Use Case of Apex for Data Cleansing	31
4.2 Data Quality Assessment	33
4.3 Data Validation	35
4.3.1 Validation Rules Use Case -1.....	36
4.3.2 Validation Rules Use Case -2.....	36
4.3.3 Validation Rues Use Case -3.....	37
4.4 Duplicate Management.....	37
4.4.1 Use Case of Duplicates and Matching Rules.....	37

4.4.2 Use Case of Apex for Duplicate Management.....	38
4.5 Data Import and Export.....	41
4.5.1 Data Import Process.....	43
5 Conclusion.....	46
References.....	47

List of Figures

Figure 1.1	The Visual Overview of Salesforce Architecture.....	11
Figure 1.2	The Layers in Salesforce Multitenancy Architecture.....	12
Figure 1.3	Sales Cloud User Interface Home Page in Salesforce.com.....	12
Figure 1.4	Salesforce Data Cloud User Interface Home Page.....	13
Figure 2.1	Impacts of Poor-Quality Data on Business Operations.....	16
Figure 2.2	Common data quality and integrity issues and their impact in Salesforce.com.....	18
Figure 2.3	A warning message before deleting ‘Address’ custom field of ‘Property’ custom object in Salesforce.com.....	19
Figure 2.4	A warning message before deleting ‘Property’ custom object in Salesforce.com.....	19
Figure 2.5	Data Mapping Visualizer in Salesforce Marketing Cloud Intelligence...22	
Figure 2.6	Data Mapping in Salesforce Data Cloud.....	22
Figure 2.7	Potential Effects of Large Data Volume in Salesforce Operations.....	23
Figure 4.1	Data Profiling and Cleansing Diagram.....	30
Figure 4.2	Definition and Examples of Data Quality Metrics in Salesforce.....	34
Figure 4.3	An Alert Notification of a Potential.....	37
Figure 4.4	Configuration of Duplicate Rules for Contact Object.....	38
Figure 4.5	Data Import Workflow.....	43

List of Tables

Table 4.1 Data Migration Methods.....42

Table 4.2 Common Field Mapping for Data Import.....44

List of Abbreviations

API	Application Programming Interface
App	Application
CRM	Customer Relationship Management
CSV	Comma Separated Values
IDE	Integrated Development Environment
LWC	Lightning Web Component
Org	Organization
SOQL	Salesforce Object Query Language
UI	User Interface
Xls	Excel Spreadsheet

1. Introduction

In today's highly competitive business environment, effective customer relationship management is crucial for the success and sustainability of any organization. The progression of information technology has simplified the processes of storing and sharing data within various departments of a corporation and with its customers. Salesforce.com was founded by Mark Benioff in 1999 and is a cloud service provider company based in the United States that specializes in CRM and offers application development and cloud services. The platform offers a comprehensive suite of tools designed to help businesses manage their customer interactions, streamline their processes and enhance their overall performance.

Modern companies cannot operate without customer relationship management solutions in order to boost efficiency, production, and income. Marketing automation, business analytics, customer care, and sales automation are just a handful of the numerous services provided by the widely used platform Salesforce (Mahlamäki et al., 2020). The CRM software aids business people in managing and organizing all of their operations while keeping an eye on their staff, resources, clients, and market needs (Ying-Yen, 2019). Salesforce drives efficiency, innovation, and growth, across sectors including healthcare, retail, and financial services. Nevertheless, Salesforce goes beyond just a CRM system; it revolutionizes the way organization's function, interact with customers, and make strategic choices (Brea, 2023).

Customer relationship management tools can unify customer and company data from many sources and even use Artificial Intelligence to help better manage relationships across the entire customer lifecycle, spanning departments like marketing, sales, digital commerce, and customer service interactions (Salesforce, 2024).

To understand the Salesforce terminology, you must first understand the multitenancy concept. Salesforce uses a multitenancy architecture, meaning that a

number of customers share IT resources such as database servers, application servers and infrastructure required to run applications, resulting in an environment that can be standardised and optimised for all efficiently and a lower cost for each 'tenant' as the resources are managed centrally (Gessner, 2024).

Salesforce Orgs work with relational databases. Hence, the data of an Org is represented by its records, which belong to objects. That is to say in Salesforce, objects essentially represent database tables that allow users to store data specific to the Org they belong to. Simply put, in Salesforce terminology, tables are known as objects. In Salesforce there are two main types of objects:

- Standard Objects; the objects already established within Salesforce, as for example: profiles, users, accounts, contacts, opportunities, products and many others.
- Custom Objects; the objects created by users to store information specific to their needs.

In addition, the same logic applies to fields, there are Standard Fields and Custom Fields. In this way, Standard and Custom Objects can be customized by populating them with new fields (Mouyal Amselem, 2020).

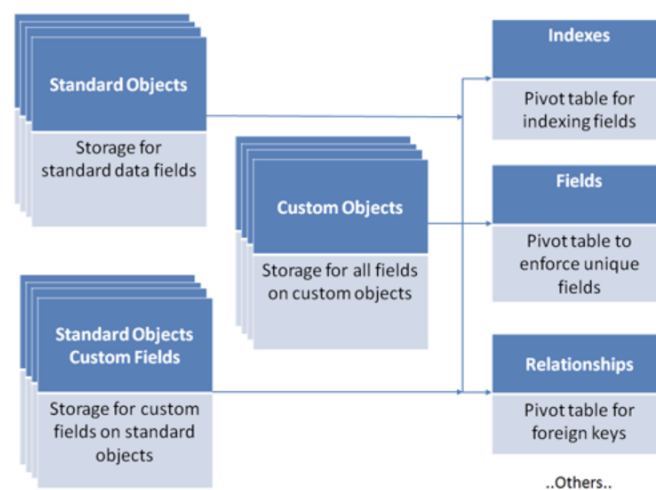


Figure 1.1: The Visual Overview of Salesforce Architecture (Salesforce, 2024)

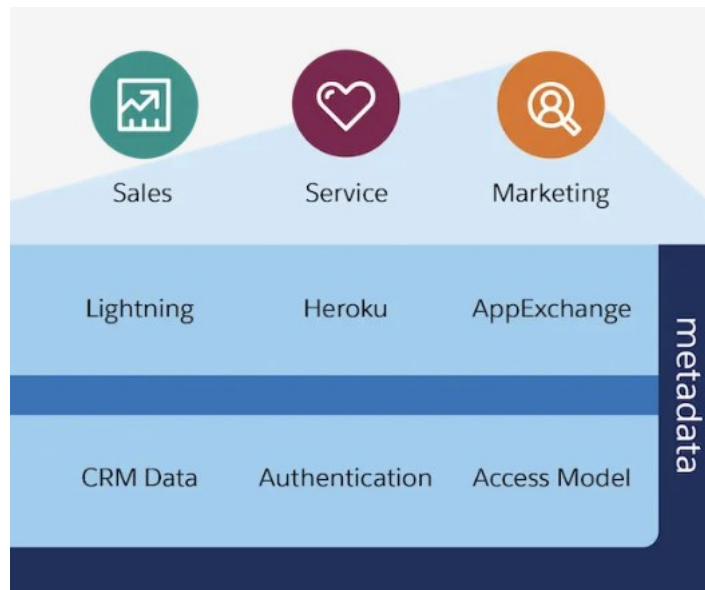


Figure 1.2: The Layers in Salesforce Multitenancy Architecture (Trailhead, 2024)

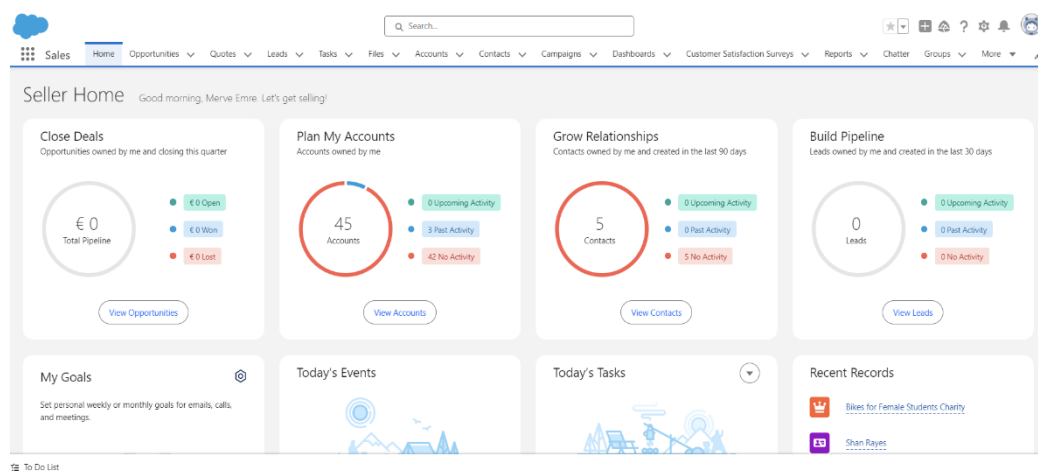


Figure 1.3: Sales Cloud User Interface Home Page in Salesforce.com

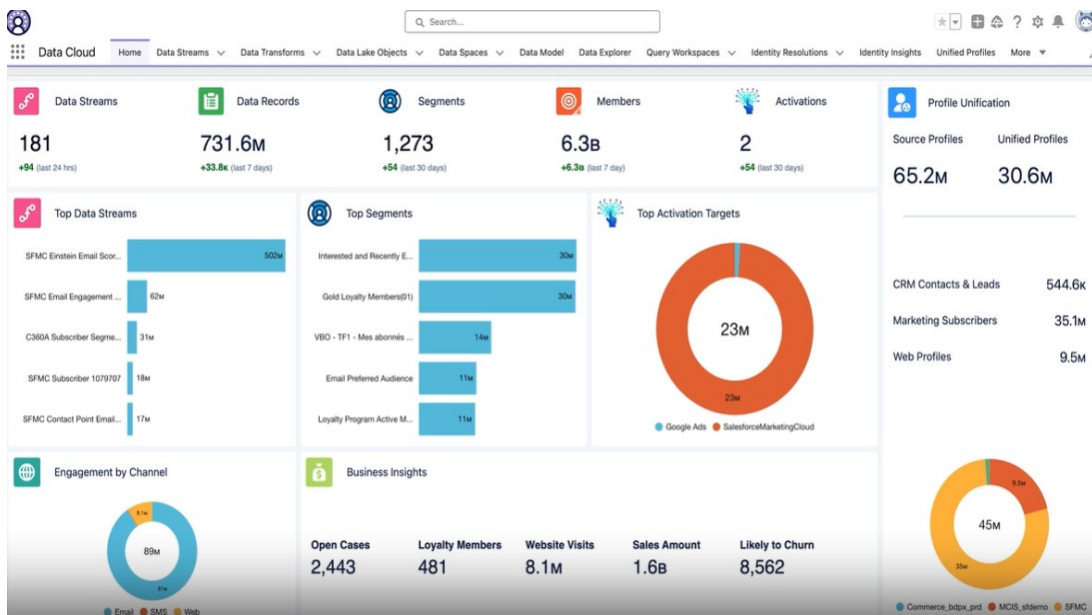


Figure 1.4: Salesforce Data Cloud User Interface Home Page (Salesforce, 2024)

Despite generally positive self-assessments by information technologies and analytics leaders, over 9 in 10 business leaders (94%) believe they should be getting even more value from their data – signaling room for improvement. To meet this need, analytics and IT leaders are focused on the fundamentals: Data quality, stronger security and AI-readiness (Salesforce, 2023).

The data quality is essential for deployment of increasingly integrated operational and analytical CRM systems, as well as for meaningful set of customers’ metrics maintenance. Industry analysts point the finger at bad data as one of the top three reasons why CRM projects fail. Because bad data leads to misleading, incomplete, and confusing information, it lowers adoption-another major reason why CRM projects fail (Boban et al., 2015). High-quality data helps in making accurate and timely decisions, while poor-quality data can lead to erroneous analyses and strategic mistakes. Salesforce.com offers various tools and mechanisms to ensure data accuracy, integrity, and accessibility.

This research aims to analyze the data quality, performance and security of Salesforce.com as a CRM Platform. By examining these critical aspects, the study

seeks to provide insights into how businesses can leverage Salesforce.com to improve their customer relationship management, ensure data security, and optimize their operations. The objectives of this project are threefold: first, to evaluate the current state of data security and quality in Salesforce.com and its impact on business processes; second, to assess the performance of Salesforce.com in handling large volumes of data and delivering timely, accurate information; and third, to propose strategies for improving data integrity, system performance, and security measures, thereby enabling businesses to achieve higher levels of efficiency, effectiveness and data protection in their CRM initiatives.

2. Problem Statements

2.1 Data Privacy and Security

Experian¹'s 2022 Global Data Management Research Report highlights the critical role of data agility in a rapidly evolving market. By surveying over 900 business leaders across North America, Europe and Australia, Experian sheds light on the impact of data security and quality in today's digital landscape. A majority of organisations are hiring data roles in the next six months to support a workforce development programme and enable their business's data management programme to mature. Data security protects the organisations as they accelerate digital efforts whether it is protecting consumer data online or on remote employee devices or just staying in compliance. Focusing on improving the customer experience - which should ultimately result in revenue growth - and data security measures can help organisations build resilience and secure their future (Experian, 2022).

Industry analysts point the finger at bad data as one of the top three reasons why CRM projects fail. Because bad data leads to misleading, incomplete, and confusing information, it lowers adoption – another major reason why CRM projects fail (Boban et al., 2015). Poor data quality not only disrupts business processes but also creates vulnerabilities in data privacy and security.

Recent incidents continue to shed light on vulnerabilities that organizations face. A notable case involves a networking products company whose support website inadvertently exposed sensitive customer information, as reported by KrebsOnSecurity². This incident underscores the critical importance of robust security measures in safeguarding customer data (Livneh, 2024). The support website that experiences a data exposure and was utilized by the networking products company was Salesforce. Although the issue was addressed promptly and users were assured of no compromise to identifiable personal customer data, questions arise regarding the adequacy of user permissions within Salesforce assets.

A data privacy security incident is any unauthorized use of personal data or customer data, whether accidental or intended. Consider these examples of common security incidents:

- A sales representative sends an email with customer data to the wrong customer.
- A manager prints a resume of a candidate, but on his way home, he leaves the document on the train.
- A former employee who has left the company still has access to your organization's systems and accesses customer records.

¹ Experian: a multinational data analytics and consumer credit reporting company headquartered in Ireland.

² Krebs on Security: a blog authored by Brian Krebs, provides in-depth security news and investigation.

- An intern opens an email attachment containing malware, which results in deleted or encrypted customer contact information.
- A shared drive is overly permissioned, granting too many people access to personal data.
- A customer’s credentials or secret keys are exposed on a public GitHub repository.
- An employee’s work device (for example, a laptop or smartphone) is lost or stolen.
- An employee accidentally discloses personal data in a response to an email that turns out to be part of a phishing attack (Trailhead, 2024).

2.2 Data Integrity and Quality

It is clear that data is more important than ever these days, but the quality of insights is what makes the data usable. To get the most of data insights and support organizations with their key strategies and beyond, data quality is a must. Yet almost three quarters say they have so much data in their organizations that it is difficult to prioritize where data management can add most value. The research report reveals that 85 of organizations indicate that poor-quality contact data for customers negatively impacts their operational processes and efficiency, thereby hindering their ability to be flexible and agile. Poor-quality data creates a ripple effect that exacerbates operational issues. Specifically, the study highlights that bad data leads to wasted resources and additional costs for 42% of the organizations, negatively affects the customer experience for 39% and damages the reliability and trust in analytics for 38%. (Experian, 2022).

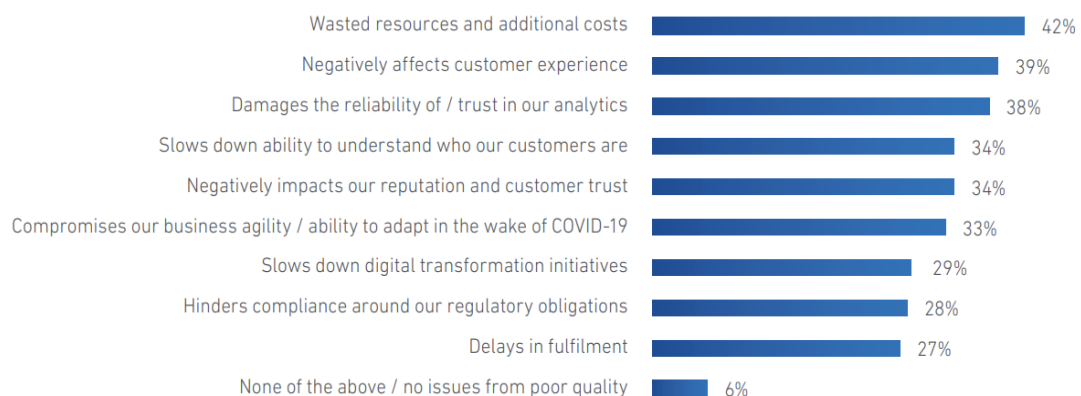


Figure 2.1: Impacts of Poor-Quality Data on Business Operations (Experian, 2022)

‘Bad data’ in the context of Salesforce refers to inaccurate, incomplete, outdated or inconsistent information stored within the Salesforce database. This can include incorrect contact details, duplicate records, incomplete entries, or irrelevant data. Dealing with bad data is a common challenge in CRM systems like Salesforce, as it can lead to inefficiencies, hinder decision-making, and affect the overall quality of customer interactions. Organizations often employ data cleansing and data quality

measures to identify and rectify bad data, ensuring that their Salesforce instance remains accurate and valuable. Common data-related errors in Salesforce:

- Duplicate Records; when the same data exists in the system multiple times, it can lead to confusion, wasted resources, and inaccurate reporting.
- Incomplete Entries; missing or incomplete data fields can hinder effective communication and decision-making. For example, a contact with missing phone numbers or email addresses.
- Inaccurate Information; incorrect data, such as wrong names, addresses, or contact details, can result in miscommunication and missed opportunities.
- Data Entry Mistakes; human errors during data entry can lead to typos, incorrect formatting, and other mistakes that impact data integrity.
- Outdated Data; stale or outdated information can lead to missed opportunities, incorrect analysis, and ineffective communication.
- Data Format Issues; data fields with inconsistent formats (e.g.; different date formats, and currency symbols) can cause confusion and hinder data analysis.
- Dependency Errors; when data is interconnected, changes in one area can affect related data, leading to errors if not managed properly.
- Data Integration Errors; when integrating data from different sources, inconsistencies can arise, leading to inaccurate or incomplete data (Forcetalks, 2023).

Accurate information and reports are the life blood of an effective sales force. Without it, management does not have the data to make good decisions, sales reps do not have the tools to turn leads into customers, and the company will find it difficult to reconcile CRM data with data in other systems. The results are lost opportunities and revenue, frustrated users and customers, and a lack of user adoption (Boban et al., 2015).

The figure below summarizes some data quality and integrity issues in Salesforce.com, detailing their definitions, reasons, effects and examples. These challenges can significantly impact data quality, consistency, and overall system performance. In the conclusion of your study, you can reference this table to emphasize the importance of implementing best practices and tools that address these issues. By understanding the reasons and results associated with each challenge, organizations can proactively prevent data errors, maintain data integrity, and ensure that their CRM systems support accurate and reliable business operations.

STATEMENT	DEFINITION	REASON	RESULT	EXAMPLE
Duplicate Records	Multiple entries that represent the same entity or customer, leading to redundant information	Manual data entry errors, importing data from different sources without deduplication, absence of validation rules	Confusion in customer interactions, inflating sales and marketing metrics, degrading the overall data quality	A customer named "John Doe" appears twice in the contact records; as "John Doe" and "J. Doe"
Unstandardized Data	The lack of consistent formats and conventions across data entries	Different data sources using varying formats, lack of standardized data entry protocols, insufficient guidelines	Reducing data quality, consistency, and reliability, complicating data analysis and reporting	Records are entered as "NL", "the Netherlands" and "Nederland" referring to the same country
Field Type Change	Changing the data type of an existing custom field can cause data loss in some situations	Incompatible conversions, inability to map directly to predefined values	Deleted list views based on the custom field, impacts on assignment and escalation rules	Changing field types from/to Date, Date/Time, Formula, Number, Percent, Currency, Checkbox and Picklist(Multi-Select)
Deleting a Custom Field	Data loss occurs when a custom field is deleted, leading to the removal of all data stored	Fields are deleted without exporting or backing up the data	Deleted information for dependent or controlling field, impacts on assignment and escalation rules	Deleting custom "Customer Support Status" picklist field of Case object which tracks the status as 'New, Working, Closed' values
Deleting a Custom Object	Data loss occurs when a custom object is deleted, leading to the removal of all data stored	Objects are deleted without exporting or backing up the data	Deleted data, workflow rules and list views actions, disabled reports, formula fields and validation rules	Deleting custom "Employee Performance Reviews" object which contains fields, validation rules

Figure 2.2: Common data quality and integrity issues and their impact in Salesforce.com

Before deleting any custom fields or objects, it is suggested to export or back up the data to prevent irreversible loss of valuable information. Encouraging all users within the organization to maintain standardized records can significantly improve data quality, making it easier to analyze and report on data across the system. Additionally, when changing field types, careful planning and testing should be conducted to avoid data loss. This can include mapping existing data to new field types or using data migration tools to ensure a seamless transition.

Changing the data type of an existing custom field can cause data loss in these situations:

- Changing to or from type Date or Date/Time
- Changing to Number from any other type
- Changing to Percent from any other type
- Changing to Currency from any other type
- Changing from Checkbox to any other type
- Changing from Picklist (Multi-Select) to any other type
- Changing to Picklist (Multi-Select) from any other type

If data is lost, any list view based on the custom field is deleted, and assignment and escalation rules can be affected. Formula fields are special read-only fields that cannot be converted to any other data type. Likewise, you cannot convert any other field type into a Formula field. (Salesforce, 2024).

The figure below shows the warning message that appears in Salesforce.com before deleting custom 'Address' field of 'Property' custom object. The warning highlights the data loss for any dependent or controlling fields and the changes in the result of Assignment and Escalation rules that rely on the field data.

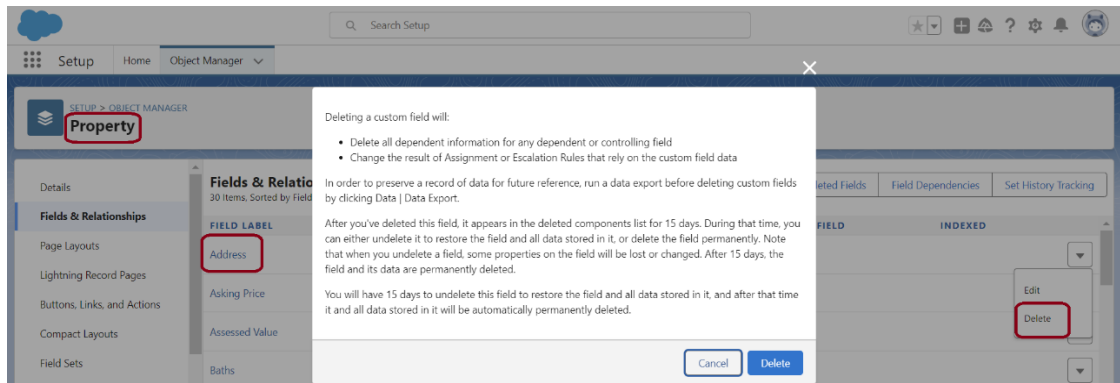


Figure 2.3: A warning message before deleting ‘Address’ custom field of ‘Property’ custom object in Salesforce.com

Another figure below shows the warning message displayed in Salesforce.com before deleting 'Property' custom object. It outlines the comprehensive consequences of deleting the object, including; removal of all data within the custom object's records, deletion of historical data associated with the object, deletion of custom tabs and list views associated with the object, removal of workflow rules and actions tied to the object, deletion of mappings to related entity interfaces and corresponding fields, concealment of the custom object definition and related definitions, hiding of the object's data records and related records, disabling of report types and custom reports where the object is primary, deactivation of custom formula fields, validation rules, and approval processes linked to the object.

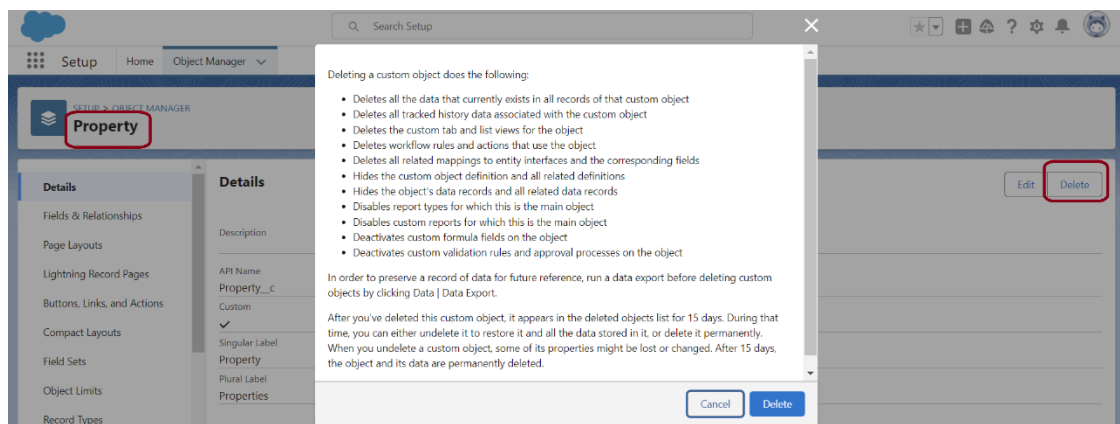


Figure 2.4: A warning message before deleting ‘Property’ custom object in Salesforce.com

2.3 Data Migration

Salesforce data migration is the process of transferring or moving data from one system or environment to another within the Salesforce platform. This may involve

moving data from an external system into Salesforce or migrating data from one Salesforce org to another. Data migration can be a complex process that requires careful planning and execution to ensure the data is transferred accurately and without errors (Sahni, 2023). The migration process can be time-consuming and challenging, but it's important to ensure that all customer data is transferred accurately and completely. The amount of time required for a successful Salesforce Data Migration depends on several factors including the size and complexity of the dataset being transferred, the source and destination systems used during transfer as well as any performance limitations on either end (Emorphis, 2024).

Salesforce data migration can be categorized into four types:

- **Database Migration:** moving data from an external database into Salesforce. This could be from a traditional SQL database, a NoSQL database or another CRM system. The complexity of the migration would depend on the schema differences between the source database and Salesforce.
- **Application Migration:** when you are transitioning from another CRM or business application to Salesforce. It not only involves moving data but also re-creating the application's functionality within Salesforce through configuration or customization.
- **Storage Migration:** this could refer to migrating attachments and files stored in Salesforce. As Salesforce has strict storage limits, organizations often migrate their Salesforce files to external storage systems or use Salesforce's own external file storage options.
- **Cloud Migration:** this refers to moving data from on-premises systems or other cloud platforms to Salesforce, which is a cloud-based platform. This could involve various cloud services like IaaS (Infrastructure as a Service), PaaS (Platform as a Service), or SaaS (Software as a Service) (Unnikrishnan, 2023).

In today's data-driven business environment, the ability to handle large volumes of data efficiently is critical for maintaining competitive advantage. Salesforce.com provides robust tools for data import, export, migration and conversion. However, managing extensive datasets poses significant challenges, including data integrity, performance bottlenecks and the risk of data loss. Effective data handling is essential for organizations to ensure seamless transitions during data migrations, accurate data imports and exports.

Organizations may need to perform a Salesforce data migration for several reasons, including transitioning from another CRM or data management system, merging multiple Salesforce instances into a single organization, consolidating data from various sources into a unified system, and cleaning up data to address quality issues such as duplicates and missing information. These migrations ensure that data is accurately and efficiently managed within Salesforce, enhancing overall operational effectiveness. The source system for Salesforce data migration can be any existing platform where data currently resides, such as on-premise databases, legacy CRM systems, spreadsheets, or other data sources. Conversely, data can also be migrated from Salesforce to other systems. The target system is typically Salesforce, where the data will be migrated to various instances, such as production environments, sandboxes or developer editions, depending on the migration's purpose. However, migrations from Salesforce to other platforms are also possible, depending on

organizational needs.

When planning a data migration, it's essential to understand the source system thoroughly. It helps identify potential challenges and risks and enables effective planning and execution of the migration process. Here are some factors that should be considered when understanding the source system for data migration:

- **Data Quality:** The quality of data in the source system can impact the success of the migration. It is essential to understand the data's accuracy, completeness, consistency, and relevance.
- **Data Volume:** The volume of data in the source system can impact the migration process. Understanding the data volume helps plan for data extraction, transformation, and loading. Based on volumes we can also decide which tool we should use for migration.
- **Data Complexity:** The complexity of data, including its structure, format, and relationships, can impact the migration process. Understanding data complexity can help determine the required data mapping and transformation.
- **Data Security:** Understanding data security requirements, such as data encryption, access control, and user roles, is essential to ensure data security during the migration.
- **Data Dependencies:** Understanding the dependencies among different data sets is crucial to ensure data integrity during the migration. It will help in related data migration.
- **Data Extraction:** The method used to extract data from the source system can impact the migration process. Understanding data extraction requirements, such as data access, extraction frequency, and performance, is essential.
- **Data Transformation:** Understanding the required data transformations, such as data mapping, data cleaning, and data formatting, is critical to ensure the migrated data meets the target system's requirements (Sahni, 2023).

Some of the broader terms that define the complexity of Salesforce Data Migration are the source systems state and configurations, existing data processes, data quality versus configuration of a destination (Salesforce) system, new data processes and usage at the time of the migration. e.g.; migrating the data into a system that already contains data is more complex than to a new Salesforce org. It becomes essential to have a strategy for migrating data to avoid the following consequences; adoption problems, customer relationship issues if wrong data gets exposed to customers, analytics problems as data are not recorded properly (Unnikrishnan, 2023).

Some of the significant problems in data migration include data mapping and transformation as well as data volume and scalability. Ensuring that data from legacy systems aligns with Salesforce's structure is essential for seamless migration. Without proper data mapping and transformation processes, inconsistencies and data integrity issues can arise. Additionally, large datasets can slow down migration processes and even cause timeouts. Handling significant volumes of data efficiently is critical to avoid performance bottlenecks and ensure a smooth migration process.

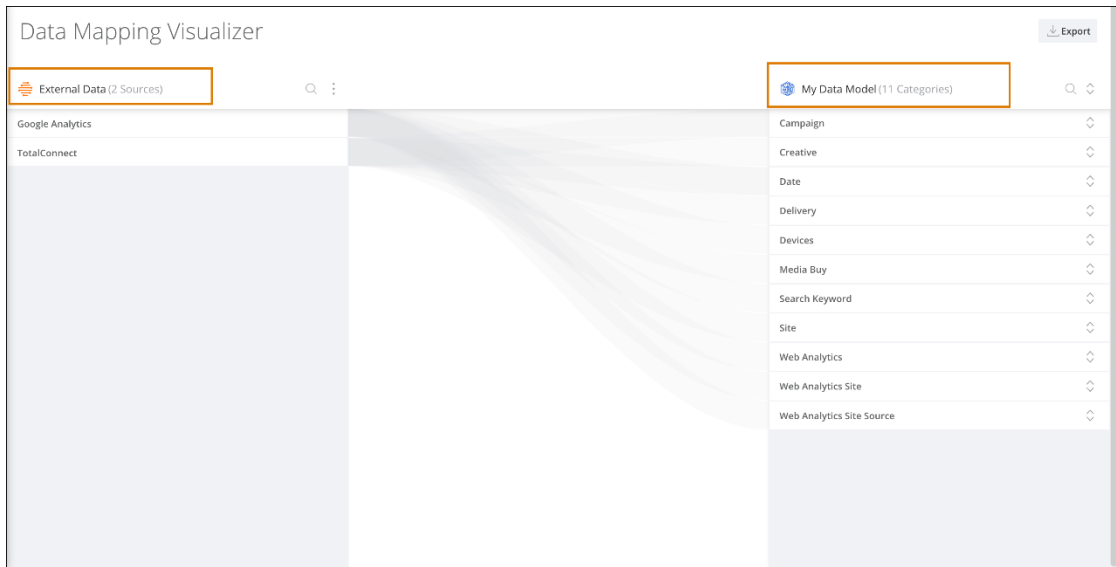


Figure 2.5: Data Mapping Visualizer in Salesforce Marketing Cloud Intelligence (Salesforce, 2024)

The figure above shows the Data Mapping Visualizer tool provides an interactive visual display that highlights how entities from the external data sources are mapped to the corresponding fields in the internal data model. Hovering over an entity in either column reveals the mapping connections. Without proper data mapping, organizations may face significant challenges in maintaining data integrity during migrations.

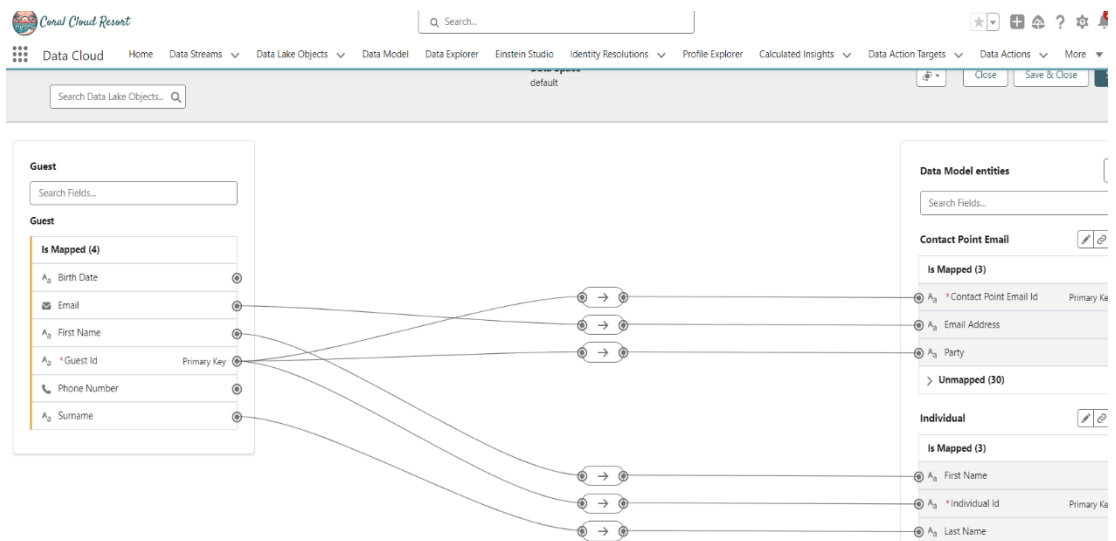


Figure 2.6: Data Mapping in Salesforce Data Cloud

The figure above shows mapping of contact-related fields from a source data stream to their corresponding target fields in Salesforce Data Cloud. It provides a visual representation of how contact information is organized and integrated within the system.

In the dynamic world of Salesforce, businesses encounter the complexities of managing extensive datasets. As organizations grow, navigating and optimizing performance in the face of Large Data Volumes becomes paramount. It signifies the

presence of a substantial amount of data within a Salesforce instance. This encompasses high record counts, intricate data relationships and extensive transaction histories, all of which can impact system performance and responsiveness (Joseph, 2024).

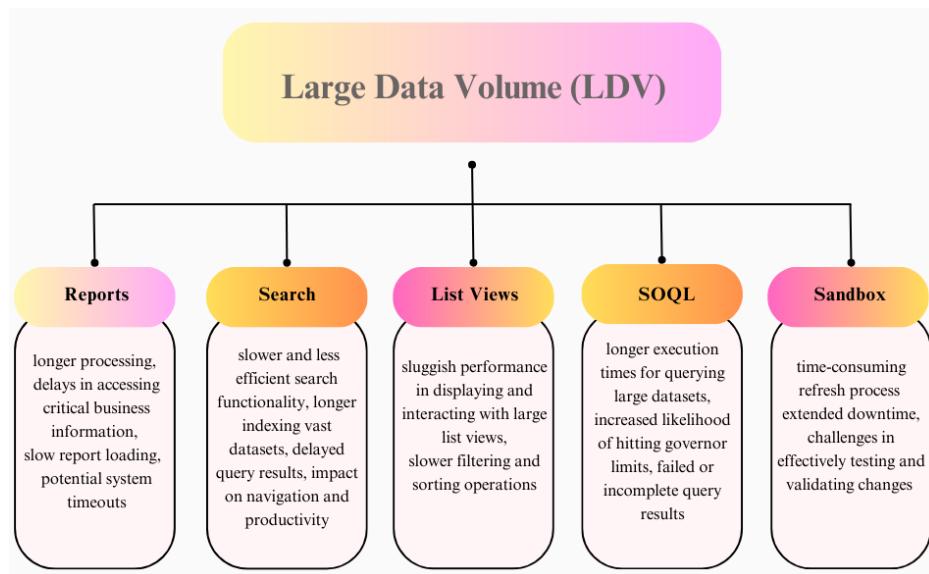


Figure 2.7: Potential Effects of Large Data Volume in Salesforce Operations

The figure above illustrates the potential impacts of Large Data Volume (LDV) on Salesforce operations, highlighting the challenges associated with managing extensive datasets. The challenges are categorized into four main areas: Reports, Search, List Views and SOQL.

If the Salesforce import tools are used, the data in most cases will need to be cleaned and formatted to match the fields in the library database. The data file needs to be in CSV format and the system requires information such as object choice and whether new records are being added or existing records are being edited. Fields then need to be accurately mapped to the database which requires careful attention. Batch loading data into the Salesforce database can save a lot of time when adding or editing a large number of records in the catalog. However, it can also be time-consuming when/if errors occur (McDonald, 2021).

3. Definition of Terms

Account: a standard object represents a company, organization, or other business entity that a user or organization interacts with.

Apex: an Object-Oriented Programming language that allows developers to execute flow and transaction control statements on Salesforce servers in conjunction with calls to the API.

API (Application Programming Interface): a set of protocols, tools, and definitions that allow different software applications to communicate with each other. Salesforce provides various APIs that developers can use to integrate external systems with Salesforce, access and manipulate data, and perform operations programmatically.

App: a collection of items that work together to serve a particular business need. Salesforce provides several standard apps such as Sales, Service, Marketing CRM Classic and Community and users can also create custom apps tailored to specific business processes.

AppExchange: an enterprise cloud marketplace offering a wide range of ready-to-install apps, solutions and consultants that extend Salesforce's capabilities across every industry and department. It allows businesses to enhance and customize their Salesforce environment with pre-built applications and services.

Assignment Rules: the process of assigning leads or cases to specific users or queues based on predefined criteria.

Cloud Computing: a technology that enables Internet-based services that let you sign up and log in through a browser. Salesforce delivers its service in the cloud. Other familiar cloud computing services include Google Apps and Amazon.com.

Cloud Service: application and computing resources, available over the Internet and managed by third-party providers. These services provide scalable and on-demand access to computing resources, including storage, servers, and applications, without the need for local infrastructure.

Contact: a standard object represents an individual person associated with an Account, typically a customer, client or stakeholder, with whom a user or organization has a relationship or communication.

CRM (Customer Relationship Management): a system for managing a company's interactions with current and potential customers. It helps businesses streamline processes, improve customer relationships and increase profitability by organizing, automating and synchronizing sales, marketing, customer service, and technical support.

Data Analysis: the process of inspecting, cleansing, transforming, and modeling data with the aim of discovering useful information, forming conclusions, and supporting decision-making. It involves various techniques to interpret data, identify patterns and extract actionable insights.

Data Cleansing: the process of detecting and rectifying errors, inconsistencies, and redundancies within the data to maintain its accuracy and reliability. This ensures that the customer data is up-to-date and trustworthy, enabling more effective business insights and strategic decisions.

Data Export Tools: allow users to extract data from Salesforce and save it in various formats such as CSV, Xls, Xlsx. These tools enable users to export Salesforce data in a format that can be used or analyzed elsewhere.

Data Import Tools: allow users to read data files from external sources and load the data into Salesforce in CSV format.

Data Integrity Features: built-in functionalities within Salesforce that ensure the accuracy, consistency, and reliability of data over its lifecycle. These features include validation rules, duplicate management and data cleansing.

Data Quality: the measure of the condition of data based on factors such as accuracy, completeness, reliability, and relevance. It involves maintaining high standards for data integrity, ensuring that data is clean, reliable, and actionable for effective decision-making and business processes. Salesforce provides various tools and features to help users improve and maintain data quality, such as validation rules, data cleansing utilities, duplicate management, and data governance policies.

Data Security: the measures and protocols used to protect sensitive information within the Salesforce Platform from unauthorized access, breaches and other security threats. This includes mechanisms as user authentication, role-based access control, encryption and regular security audits to ensure data integrity and compliance with industry standards.

Data Validation: the process of ensuring that a program operates on clean, correct and useful data. They are often used to prevent incorrect or inconsistent data entry. Formulas can be used within data validation rules to define complex conditions for data correctness and usefulness.

Database Management System: the underlying technology that manages the storage, retrieval, and organization of data within the Salesforce platform. It ensures data integrity, supports complex queries through SOQL and provides the necessary infrastructure for managing relational data within the Salesforce ecosystem.

Deployment: the process of moving metadata and configuration settings from one Salesforce environment to another. This can involve moving changes from a development sandbox to a production environment or from one sandbox to another. Deployment ensures that the customizations and developments made in a non-production environment are accurately and securely transferred to a live environment for end users.

Developer Console: a powerful tool that provides a user-friendly interface for developers to write, debug, and test Apex code, Visualforce pages, and Lightning components. It offers features like code editors, debug logs, execution logs, testing frameworks, and performance checkers, enabling developers to efficiently build and troubleshoot applications within the Salesforce Platform.

Developer Edition: a free version of Salesforce with limited functionality, designed for developers to build, test, and demonstrate applications. It provides access to most of the platform features, including Apex, Visualforce, and Lightning, without the need for a full Salesforce license.

Escalation Rules: dictate the automatic escalation of records to designated users or queues when predefined conditions are met, ensuring timely resolution of issues and adherence to Service Level Agreements (SLAs).

Field Type: the categorization of data fields based on the type of information they store. Some examples of field types include Text, Number, Date, Email, Phone, Picklist, Master-Detail and Lookup.

Fields: individual data points within an object, analogous to columns in a database table. Each field stores specific pieces of information about a record, such as name, email, phone number and more.

Flows: automated processes in Salesforce that guide users through a series of screens to collect and process data. They enable the creation of interactive workflows without requiring extensive coding, enhancing user experience and data accuracy.

Formulas: expressions used to calculate values dynamically based on fields, functions, and operators. They are used in various parts of Salesforce, such as custom fields, validation rules and workflow rules to automate calculations and logic.

Framework: a set of tools, technologies and API's that enable developers to design, develop, deploy and manage applications on the Salesforce Platform. It includes both declarative development tools, such as the Lightning App Builder and Process Builder, as well as programmatic development tools like Apex and Visualforce.

Large Data Volume: the storage and processing of an extensive amount of information which exceeds the typical or expected data load for an organization's typical usage and operations.

Legacy Systems: outdated or older software systems and databases that an organization has used historically to manage customer data and business processes.

Lightning App Builder: a tool within the Salesforce Platform that allows users to create custom applications and user interfaces. With its intuitive drag and drop interface, users can easily design and customize pages, layouts, and components to meet their specific business needs.

Lightning Web Component (LWC): a JavaScript Framework for building custom components in the Salesforce Lightning Experience. It offers a lightweight and standards-based approach to component development, enabling developers to create

responsive and efficient user interfaces.

Low Code Tools: tools that enable the rapid creation of applications with minimal hand-coding, using a visual development environment. Lightning App Builder and Flows are some of examples of low code tools.

Metadata: the configuration information and customizable components that define the behavior, appearance, and functionality of Salesforce orgs and applications. It includes data about objects, fields, layouts, workflows, processes, reports, dashboards and other elements that can be customized or extended within the platform.

MuleSoft: an integration platform that enables organizations to connect their applications, data and devices through APIs. It enhances Salesforce's capabilities by providing seamless integration and connectivity across various systems.

Multitenancy: the architecture of the Salesforce platform to host multiple organizations (tenants) on a shared infrastructure while keeping their data and configurations isolated and secure from each other.

Objects: a distinct type of data that serves as a container for storing related records. Each object consists of fields that define the specific attributes or properties of the data.

OOP (Object-Oriented Programming): a programming approach used in Salesforce to structure and manage code. OOP involves creating objects, which are instances of classes. Classes in Salesforce's Apex language define the structure and behavior of these objects, including their variables and methods.

Org: a single customer's data and application environment, which can be used by a single person or a group of people within the same organization. It contains all the data, configurations, customizations, and applications unique to that customer. Orgs are isolated from one another to ensure data security and integrity. There are different types of orgs, such as production orgs for live operations and sandboxes for development and testing.

Picklist Field Type: a data field that allows users to select a value from a predefined list of options. It enforces data consistency by limiting selections to predefined values.

Production: the live environment where users perform actual business operations using Salesforce applications.

Record: a single, uniquely identifiable instance of an object that contains specific data entries. Each record is composed of fields, which are individual pieces of data (akin to columns in a database table), and values corresponding to these fields.

Relationships: the connections between different objects in the database. Lookup relationships enable users to associate records across different objects without altering record ownership or access permissions, whereas in Master-Detail relationships, the detail object inherits security and sharing settings from the master object, and deletion of the master record automatically triggers the deletion of related detail records.

Reports: instrumental tools within Salesforce, allowing users to compile and visualize data from diverse angles, aiding in informed decision-making processes.

Salesforce: a cloud-based software company offering CRM solutions, founded by Marc Benioff in 1999.

Salesforce Application: a set of tasks or functions that are designed to perform specific duties within the Salesforce Platform.

Salesforce App Cloud: a collection of development tools that enable developers to create applications that run on the Salesforce Platform.

Salesforce Platform: a suite of tools and services that form the foundation for creating and deploying applications within the Salesforce ecosystem.

Sandbox: A testing environment in Salesforce that is separate from the production environment, used for development, testing and training without affecting live data.

Service Level Agreement (SLA): a contract between a service provider and a customer that defines the service to be provided and the level of performance to be expected.

SOQL (Salesforce Object Query Language): a specialized query language used to search and retrieve data stored in Salesforce objects. It allows users to construct queries to select fields, filter records and perform basic database operations within the Salesforce Platform.

Validation Rules: mechanisms that enforce specific criteria on data entries to ensure accuracy and consistency. They prevent users from saving records that do not meet defined conditions, providing error messages to guide corrections.

Workflows: automated processes that trigger a series of predefined actions based on certain criteria. These actions can include updating records, sending email alerts, creating tasks, or executing custom tasks, allowing users to automate routine tasks and streamline business processes within the Salesforce platform.

Xls: a proprietary file format developed by Microsoft for its Excel spreadsheet software. It is extensively used in various academic, business, and administrative settings for data storage, analysis and presentation.

4. Strategies and Best Practices

4.1 Data Profiling and Cleansing

Data profiling is all about understanding data. Companies should know where their data come from; spreadsheets, backend systems, or sticky notes all over reps' desks. An inventory of the data should be taken that includes the following information:

- The data sources should be listed as well as the names of the fields in which data are stored.
- Any potential problems with the data should be noted. Companies should ask questions such as: Do we have automated quality checks before new record can be saved? Are all fields mapped correctly? (Boban et al., 2015).

The Data Management Association³ defines data profiling as statistical analysis of data set contents to understand format, completeness, consistency, validity and structure of the data. For projects involving a significant amount of data, like Salesforce, they recommend a data profiling tool as the most efficient means of conducting this analysis (Orun, 2024).

Data profiling analyzes the fields, records and values of a Salesforce Object. It assesses the data volume, shape, and field usage trends. This analysis produces key statistical insights describing record volume, storage estimates, field usage and value frequency (Lloyd, 2023).

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. Having clean data will ultimately increase overall productivity and allow for the highest quality information in your decision-making. Benefits include:

- Ability to map the different functions and what your data is intended to do.
- Monitoring errors and better reporting to see where errors are coming from making it easier to fix incorrect or corrupt data for future applications (Tableau, 2024).

³ The Data Management Association: a non-profit professional association focused on advancing data management and data governance practices.

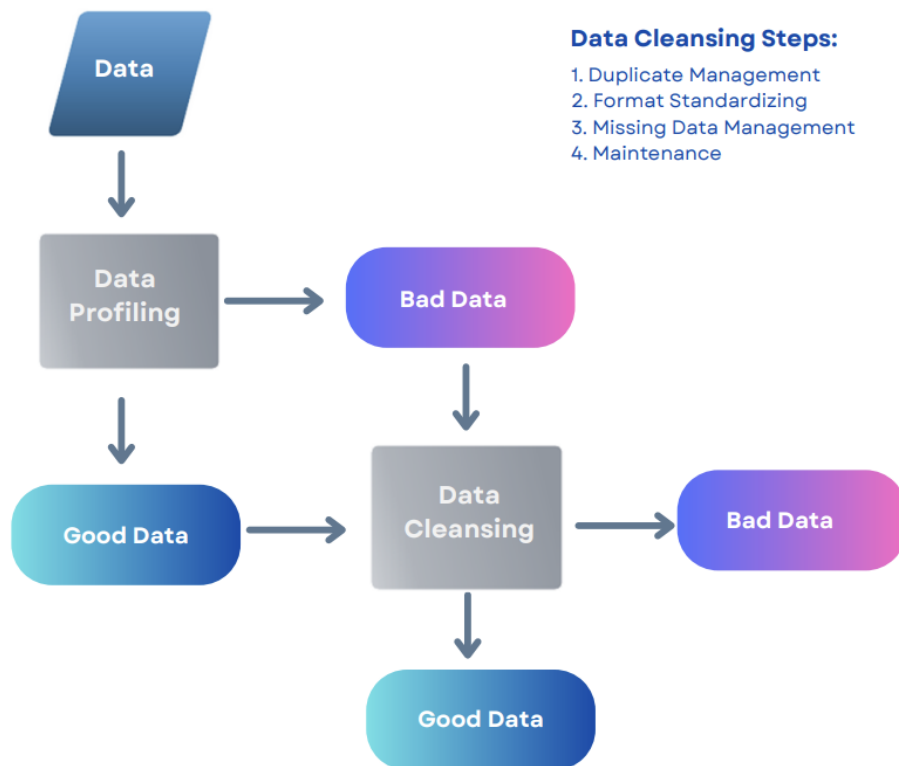


Figure 4.1: Data Profiling and Cleansing Diagram

The figure above illustrates the process of data profiling and cleansing within Salesforce. Initially, raw data is analyzed before undergoing the profiling stage. During profiling, data is categorized into good and bad segments. Both good and bad data are then subjected to a cleansing process, which can result in further categorization. There are four steps for data cleaning; duplicate management, format standardizing, missing data management and maintenance.

To enhance data profiling and cleaning in Salesforce, leveraging tools and products from the Salesforce AppExchange can be beneficial. It offers various specialized applications that provide robust features for identifying and merging duplicate records, validating data accuracy and standardizing data formats.

Salesforce's native capabilities such as Salesforce Object Query Language and Apex can enhance data profiling and cleaning efforts. SOQL allows administrators to query the Salesforce database efficiently, enabling detailed data analysis and identification of problematic records. Apex; Salesforce's programming language, offers robust functionality to automate data cleaning processes, such as merging duplicates, updating records, and ensuring data consistency. By incorporating SOQL and Apex into data management strategies, organizations can maintain high data quality, reduce manual effort and ensure the reliability of their Salesforce data.

4.1.1 Use Case of SOQL for Data Cleansing

Use Case Scenario: Organizations often face issues with incomplete, inconsistent or duplicate Account, Contact and Opportunity data within their Salesforce records. To identify and address these issues, Salesforce Object Query Language queries can be employed.

Example Queries for Data Cleansing with SOQL:

```
SELECT Id, Name FROM Account WHERE BillingCountry IS NULL OR BillingCountry = "
```

```
SELECT Name, COUNT(Id) FROM Contact GROUP BY Name HAVING COUNT(Id) > 1
```

```
SELECT Status, COUNT(Id) FROM Opportunity GROUP BY Status
```

```
SELECT Industry, COUNT(Id) FROM Account GROUP BY Industry
```

In these query steps, 'Id' and 'Name' fields from the Account object are filtered for records where the 'BillingCountry' field is either null or an empty string. This query helps identify accounts with missing or invalid billing country information. The next query aggregates Contact records by 'Name' field, counting the number of records for each unique name and filtering to only include names with more than one associated contact. This approach can be used to detect duplicate contact records. The third query groups Opportunity records by 'Status' field, providing an overview of the distribution of opportunities across different statuses. This analysis can reveal potential issues with inconsistent or incorrect status values. Lastly, the final query groups Account records by the 'Industry' field, counting the number of records for each industry. This information can be leveraged to identify accounts with missing or inaccurate industry data.

4.1.2 Use Case of Apex for Data Cleansing

Use Case Scenario: An organization is facing challenges with incomplete, inconsistent, or duplicate data in their Salesforce records for Accounts, Contacts and Opportunities. These challenges include issues such as missing or duplicate contact information, incomplete billing country details in accounts, inconsistent opportunity statuses and non-standard industry classifications. The following Apex classes are designed to address these data quality issues and perform essential data cleansing tasks.

Example Apex codes for data cleansing:

```

// Merging Duplicate Contacts
public class DuplicateCleaner {
    public static void mergeDuplicates(List<Contact> duplicates) {
        if (duplicates.size() > 1) {
            Contact master = duplicates.remove(0);
            Database.merge(master, duplicates);
        }
    }
}

// Updating Incomplete Account Records
public class IncompleteAccountUpdater {
    public static void updateIncompleteAccounts(List<Account> accounts) {
        for (Account acc : accounts) {
            if (acc.BillingCountry == null || acc.BillingCountry == "") {
                acc.BillingCountry = 'Unknown'; // Assign a default value
            }
        }
        update accounts;
    }
}

// Data Consistency Check and Correction
global class OpportunityStatusCheck implements Database.Batchable<SObject> {

    // The start method to define the query to retrieve all Opportunity records
    global Database.QueryLocator start(Database.BatchableContext bc) {
        // Query to retrieve all Opportunity records with Status__c field
        return Database.getQueryLocator([SELECT Id, Status__c FROM Opportunity]);
    }

    // The execute method to process each batch of records
    global void execute(Database.BatchableContext bc, List<SObject> scope) {
        List<Opportunity> opportunitiesToUpdate = new List<Opportunity>();

        // Iterate through the batch of opportunities
        for (SObject sObj : scope) {
            Opportunity opp = (Opportunity)sObj;

            // Check if the Status__c value is not one of the valid statuses
            if (!('Open'.equals(opp.Status__c) || 'Closed Won'.equals(opp.Status__c) ||
                'Closed Lost'.equals(opp.Status__c))) {
                opp.Status__c = 'Open'; // Correct to a default valid status
                opportunitiesToUpdate.add(opp);
            }
        }

        // Update records if there are any to update
        if (!opportunitiesToUpdate.isEmpty()) {
            try {
                update opportunitiesToUpdate;
            }
        }
    }
}

```



```

    } catch (DmlException e) {
        // Handle the exception
        System.debug('An error occurred during update: ' + e.getMessage());
    }
}

// The finish method called when the batch processing is complete
global void finish(Database.BatchableContext bc) {
    // Optional: Add logging or notifications
    System.debug('Batch job finished successfully.');
```

// Cleaning Industry Data

```

public class IndustryDataCleaner {
    public static void cleanIndustryData(List<Account> accounts) {
        for (Account acc : accounts) {
            if (!('Technology'.equals(acc.Industry) || 'Healthcare'.equals(acc.Industry) ||
'Finance'.equals(acc.Industry))) {
                acc.Industry = 'Other'; // Assign a default industry
            }
        }
        update accounts;
    }
}

```

The ‘DuplicateCleaner’ class merges duplicates into a single master record, reducing redundancy and ensuring data accuracy. Similarly, the ‘IncompleteAccountUpdater’ class addresses incomplete account information by updating records with missing billing country details, assigning a default value ‘Unknown’ to ensure completeness. The ‘OpportunityStatusCheck’ class is used to correct inconsistencies in the Status__c custom field on Opportunity records. The ‘IndustryDataCleaner’ class ensures that account records have accurate industry classifications by updating entries with non-standard values to a default industry ‘Other’.

4.2 Data Quality Assessment

After data cleaning, the next steps are data quality assessment and monitoring. The core of data quality assessment is to evaluate each dimension of the data. The current methods for evaluation fall into two categories: qualitative and quantitative methods. Qualitative evaluation is based on certain criteria and requirements and is performed by subject experts or professionals. The quantitative method is a formal, objective, and systematic process that utilizes numerical data to obtain information. The results of

quantitative evaluation are more intuitive and concrete due to their objectivity, generalizability and use of numbers (Shanmugam et al., 2023).

MEASURE	DEFINITION	EXAMPLE
ACCURACY	The extent in which data are able to represent reality	Contact e-mail addresses match the actual email addresses used by customers. Opportunity amounts reflect the true deal values.
INTEGRITY	The consistency of the structure of data and relationships among the entities and attributes	Customer information is consistent between Salesforce and integrated marketing automation tools.
CONSISTENCY	The uniformity of data across different datasets and systems	A lead's contact details remain consistent across Sales Cloud and Service Cloud.
COMPLETENESS	The extent to which all required data fields are populated and no critical information is missing	Each account record contains a valid address, industry, and contact information. Every opportunity has a stage, close date and amount specified.
VALIDITY	Data values fall within the defined ranges	Date fields contain valid dates in the correct format. Phone number fields contain only valid phone numbers with the correct format.
TIMELINESS	Data is up-to-date and reflects the most current information available	Lead status is updated promptly after each sales call. Customer support cases are logged and updated in real-time in Service Cloud.
ACCESSIBILITY	Data is accessible, comprehensible and usable	Sales reps can easily retrieve opportunity details from Salesforce mobile app; Marketing team can access campaign performance data quickly.

Figure 4.2: Definition and Examples of Data Quality Metrics in Salesforce

The figure above outlines key data quality metrics essential for assessing and ensuring the reliability and usability of data within an organization. Each metric is defined and exemplified to illustrate its application and importance in maintaining high data quality standards. Regular assessment of these data quality metrics helps organizations identify areas needing improvement and implement corrective measures promptly. By continuously monitoring and enhancing data quality, organizations can ensure that their Salesforce data remains a valuable and reliable asset.

Studies have confirmed data quality is a multi-dimensional concept. Companies must deal with both the subjective perceptions of the individuals involved with the data, and the objective measurements based on the data set in question. Subjective data quality assessments reflect the needs and experiences of stakeholders: the collectors, custodians, and consumers of data products. Objective assessments can be task-independent or task-dependent. Task-independent metrics reflect states of the data without the contextual knowledge of the application, and can be applied to any data set, regardless of the tasks at hand. Task-dependent metrics, which include the organization's business rules, company and government regulations, and constraints provided by the database administrator, are developed in specific application contexts. To use the subjective and objective metrics to improve organizational data quality requires three steps:

- Performing subjective and objective data quality assessments

- Comparing the results of the assessments, identifying discrepancies, and determining root causes of discrepancies
- Determining and taking necessary actions for improvement (Pipino et al., 2002).

4.3 Data Validation

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False”. Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value (Salesforce, 2024). This mechanism ensures data integrity by preventing incorrect or incomplete data from being entered into the system. By enforcing these rules, organizations can maintain high-quality data, which is crucial for accurate reporting, effective decision-making, and streamlined operations. Effective implementation of validation rules not only enhances data accuracy but also improves user trust in the system by providing immediate feedback on data entry errors.

Creating validation rules is a critical step in ensuring data integrity within a database system. Validation rules are logical expressions that evaluate the correctness of data entries based on predefined criteria. When defining these rules, it is essential to consider the specific requirements of the application to ensure that all data entered adheres to the expected format, range, and constraints. Properly defined validation rules help prevent data anomalies and maintain a high standard of data quality, which is crucial for reliable data analysis and decision-making processes. In Salesforce.com platform, validation rules consist of a formula or expression that evaluates data in one or more fields, returning a ‘boolean’ value. This evaluation ensures that only valid data is saved, with invalid data triggering an error message to prompt user correction before submission. System Administrators have the highest level of permissions in Salesforce and can create, modify and delete validation rules across all objects within the system.

Implementing validation rules in Salesforce.com ensures that data entered into the system adheres to the predefined standards and constraints. Once a Salesforce Administrator has defined the validation rules, the following process is executed whenever a user interacts with the system;

- The user creates a record or edit an existing record, the user clicks “Save”. All validation rules are verified.
- If all data is valid, the record is saved.
- If any data is invalid, the associated error message displays without saving the record.
- The user makes the necessary changes and clicks “Save” again.

4.3.1 Validation Rules Use Case – 1

A Salesforce Administrator is tasked with ensuring that the Close Date field on Opportunity object cannot be set by the users in the past. This request typically arises from the need to maintain accurate sales data and improve the overall integrity of the sales process. Allowing past close dates can lead to inconsistencies in data reporting and analysis. It may create confusion regarding the status of opportunities and affect decision-making. To achieve this task the Administrator creates and deploys a validation rule with an informative error message.

Formula for the validation rule:

```
CloseDate < TODAY()
```

Error message to guide users:

"The Close Date cannot be set in the past."

4.3.2 Validation Rules Use Case – 2

Close Date field on Opportunity object is wanted to be back-dated by only the Salesforce Administrator. A validation rule can help maintain data integrity while allowing administrators the flexibility to back-date opportunities in this scenario. To achieve this task the Administrator creates and deploys a validation rule with an informative error message.

Formula for the validation rule:

```
AND(  
  OR(  
    ISNEW(),  
    ISCHANGED(CloseDate)  
  ),  
  CloseDate < DATE(YEAR(TODAY()),  
MONTH(TODAY()), 1),  
  $Profile.Name <> "Custom: System Admin"  
)
```

Error message for the users:

"Close Date cannot be prior to current month."

4.3.3 Validation Rules Use Case – 3

A Salesforce Administrator is tasked with preventing users from saving an open case associated with a Contact that is no longer with the company. In this scenario there is a custom checkbox field on Contact object called ‘No Longer with Company’. To achieve this task the Administrator creates and deploys a validation rule with an informative error message.

Formula for the validation rule:

```
AND(Contact.Not_Longer_With_Company__c,  
      NOT(IsClosed))
```

Error message for the users:

“Unable to save this case. The related contact is no longer with the company. To continue, choose another contact.”

4.4 Duplicate Management

Duplicate management involves identifying and preventing duplicate records in Salesforce to maintain data integrity. It helps reduce data clutter, improve reporting accuracy, and enhance the user experience. A Salesforce Administrator can create and manage matching and duplicate rules, utilize built-in reports to identify existing duplicates and use third-party tools (such as data deduplication apps) and APIs to integrate with external data sources to validate records before data entry. The primary tools within Salesforce for managing duplicates are Matching Rules and Duplicate Rules, which work together to identify and handle duplicate records effectively.

4.4.1 Use Case of Duplicate and Matching Rules

Matching Rules define the criteria for what is considered a duplicate record by specifying which fields to compare and how to match them. Duplicate Rules, on the other hand, specify the actions that should be taken when a user attempts to create a record that matches the criteria defined by the Matching Rules. Each Duplicate Rule requires at least one Matching Rule to identify which existing records are potential duplicates.

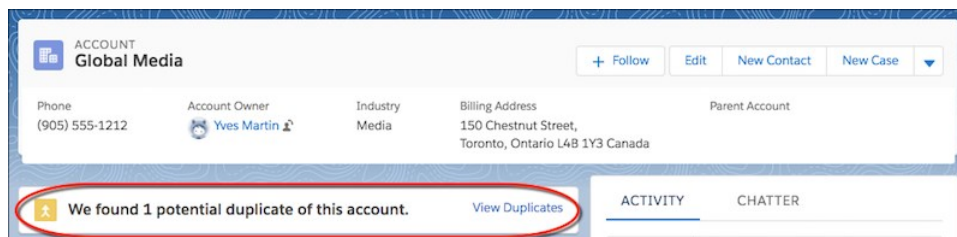


Figure 4.3: An Alert Notification of a Potential Duplicate (Salesforce, 2024)

The figure above illustrates a duplicate alert notification within a Salesforce account record for 'Global Media'. By viewing duplicates users are able to review and manage the duplicate records.

Use Case Scenario: A Salesforce Administrator is tasked with ensuring that the sales team maintains clean and accurate contact data within the platform. The team frequently imports leads from various sources, which occasionally results in duplicate contacts being created.

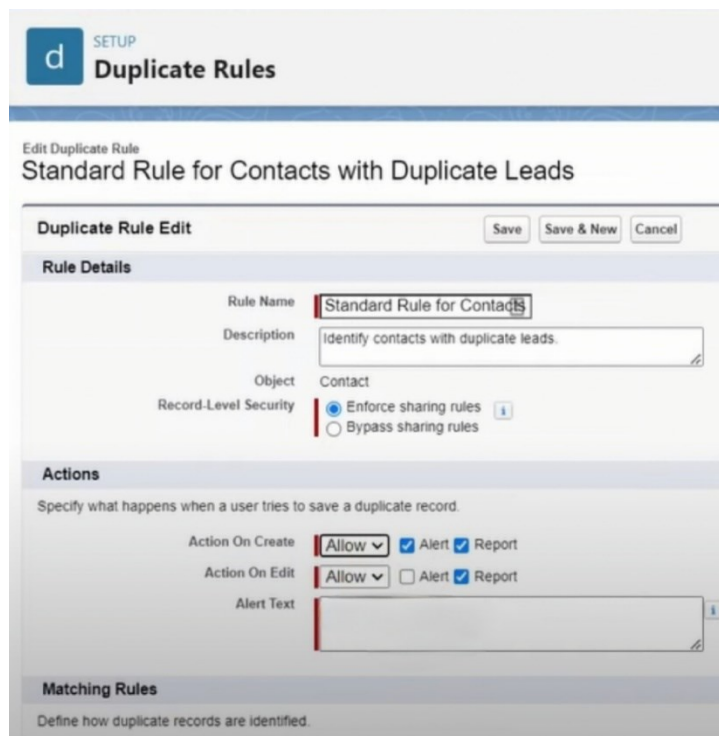


Figure 4.4: Configuration of Duplicate Rules for Contact Object

The figure above illustrates the configuration interface for setting up Duplicate Rules. By implementing this, the Salesforce Administrator ensures that the system automatically flags potential duplicates without blocking the workflow. This balance between control and flexibility allows the team to maintain accurate records while still operating efficiently. The example shows how administrators can define and manage duplicate detection and resolution processes by Matching Rules criteria and specifying what happens when a user tries to save a duplicate record.

4.4.2 Use Case of Apex for Duplicate Management

A company has been facing issues with duplicate customer records in its Salesforce platform. These duplicates arise primarily due to the integration of multiple lead generation sources, manual data entry errors and inconsistent data updates. The duplicates are particularly prevalent in the Contact and Account objects, causing

confusion in sales and support teams, inaccurate reporting, and difficulty in tracking customer interactions. A Salesforce Developer is tasked with implementing an Apex-based solution to automatically identify and merge these duplicates, ensuring that the data remains clean and accurate.

Example Apex classes for Duplicate Management:

```
// Batch class for managing duplicate Contacts
global class ContactDuplicateManager implements Database.Batchable<SObject> {

    // Start method to retrieve potential duplicate contacts based on email or phone
    number
    global Database.QueryLocator start(Database.BatchableContext bc) {
        String query = 'SELECT Id, Email, Phone, CreatedDate FROM Contact WHERE
        Email != null OR Phone != null ORDER BY CreatedDate ASC';
        return Database.getQueryLocator(query);
    }

    // Execute method to process each batch of contacts and merge duplicates
    global void execute(Database.BatchableContext bc, List<SObject> scope) {
        List<Contact> contacts = (List<Contact>) scope;

        // Find duplicates by email
        Map<String, List<Contact>> emailDuplicates = findDuplicates(contacts,
        'Email');
        // Find duplicates by phone
        Map<String, List<Contact>> phoneDuplicates = findDuplicates(contacts,
        'Phone');

        // Merge duplicates by email
        for (List<Contact> duplicates : emailDuplicates.values()) {
            if (duplicates.size() > 1) {
                mergeContacts(duplicates);
            }
        }

        // Merge duplicates by phone
        for (List<Contact> duplicates : phoneDuplicates.values()) {
            if (duplicates.size() > 1) {
                mergeContacts(duplicates);
            }
        }
    }

    // Helper method to find duplicate contacts based on Email or Phone
    private Map<String, List<Contact>> findDuplicates(List<Contact> contacts, String
    keyField) {
        Map<String, List<Contact>> duplicatesMap = new Map<String,
        List<Contact>>();

        for (Contact contact : contacts) {
```

```

String keyValue = "";

if (keyField == 'Email' && contact.Email != null) {
    keyValue = contact.Email;
} else if (keyField == 'Phone' && contact.Phone != null) {
    keyValue = contact.Phone;
}

if (keyValue != "") {
    if (!duplicatesMap.containsKey(keyValue)) {
        duplicatesMap.put(keyValue, new List<Contact>());
    }
    duplicatesMap.get(keyValue).add(contact);
}
}

return duplicatesMap;
}

// Helper method to merge a list of duplicate contacts
private void mergeContacts(List<Contact> duplicates) {
    Contact master = duplicates.remove(0); // Oldest record as the master
    try {
        Database.merge(master, duplicates);
    } catch (DmlException e) {
        System.debug('Merge failed: ' + e.getMessage());
    }
}

// Finish method to log completion of the batch job
global void finish(Database.BatchableContext bc) {
    System.debug('Contact duplicate management batch job finished.');
```

```

}

// Scheduler class to automate batch execution
global class ScheduleContactDuplicateManager implements Schedulable {
    global void execute(SchedulableContext sc) {
        ContactDuplicateManager batch = new ContactDuplicateManager();
        Database.executeBatch(batch);
    }
}

// To schedule the batch job to run daily at midnight, use the following line in the
Developer Console:
// System.schedule('Daily Contact Duplicate Manager', '0 0 0 * * ?', new
ScheduleContactDuplicateManager());
```


The solution involves a batch class that retrieves potential duplicates based on email and phone number, and merges them with the oldest record designated as the master. The process is automated through a scheduled job to ensure ongoing data cleanliness and accuracy, minimizing confusion and enhancing reporting and customer interaction tracking.

4.5 Data Import and Export

Data migration can be categorized into four types:

- **Database Migration:** Moving data from an external database into Salesforce. This could be from a traditional SQL database, a NoSQL database, or another CRM system. The complexity of the migration would depend on the schema differences between the source database and Salesforce.
- **Application Migration:** Transitioning from another CRM or business application to Salesforce. It not only involves moving data but also re-creating the application's functionality within Salesforce through configuration or customization.
- **Storage Migration:** Migrating attachments and files stored in Salesforce. As Salesforce has strict storage limits, organizations often migrate their Salesforce files to external storage systems or use Salesforce's own external file storage options.
- **Cloud Migration:** Moving data from on-premises systems or other cloud platforms to Salesforce, which is a cloud-based platform. This could involve various cloud services like IaaS (Infrastructure as a Service), PaaS (Platform as a Service), or SaaS (Software as a Service) (Unnikrishnan, 2023).

The table below provides a comparative overview of various data migration tools available in the Salesforce platform. It highlights key parameters such as supported editions, the number of records each tool can handle, import and export capabilities, whether the tool is internal or external to Salesforce and additional relevant details.

Table 4.1: Data Migration Methods (Salesforce, 2024)

Tool	Editions Supported	Number of Records	Import	Export	Internal or External to Salesforce	Additional Information
Basic Data Import	Enterprise, Performance	Up to 50,000	Yes	No	Internal	An in-browser tool that provides a simplified approach for users who only need to import contacts and leads
Data Import Wizard	All, except Personal and Database.com Editions	Up to 50,000	Yes	No	Internal	An in-browser wizard that imports the org's accounts, contacts, leads, solutions, campaign members and custom objects
Data Loader	Enterprise, Unlimited, Performance, Developer, Database.com Editions	Between 5,000 – 5,000,000	Yes	Yes	External	An application for the bulk import or export of data to insert, update, delete or export Salesforce records
dataloader.io	All	Varies by dataloader.io plan	Yes	Yes	External	A cloud-based data import tool powered by Mulesoft

The required format for data import is typically CSV (Comma-Separated Values). For data export, Salesforce primarily utilizes the CSV format as well, which can be processed by external systems or used for backup and reporting purposes. Additionally, exporting data in Excel (.xls, .xlsx) formats is supported, especially when utilizing tools such as Reports. This capability enhances users' ability to perform data analysis and reporting effectively. Images, documents and attachments can be included during exporting.

In the Salesforce platform, data can be integrated using point-and-click tools or by using code. The options and pathways chosen will depend on the level of expertise and skills available to accomplish the integration. Salesforce can bring together disparate systems to build a unified view by using its robust tools and powerful

Application Programming Interfaces (APIs), which is a software intermediary that allows two applications to talk to each other. The Salesforce Platform uses open APIs based on industry standards like Representational State Transfer (REST) and Simple Object Access Protocol (SOAP), so API integration is easy between Salesforce and external endpoints, such as apps or enterprise integration hubs (McDonald, 2021).

4.5.1 Data Import Process

The data import process is a critical component of data migration, particularly when dealing with large volumes of data in Salesforce. Proper planning and execution are essential to maintain data integrity and system performance. The following diagram outlines the key steps involved in this process.

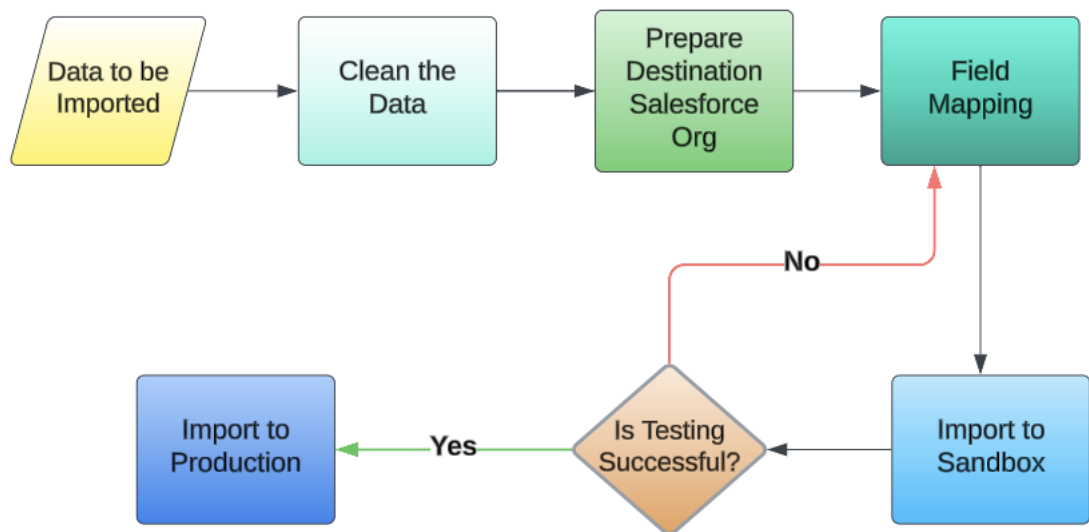


Figure 4.5: Data Import Workflow

Data to be Imported: Objects and specific records that will be migrated. It should be formatted and saved as a CSV file.

Clean the Data: Data cleansing involves removing duplicates, correcting errors, and standardizing formats.

Prepare Destination Salesforce Org: Setting up custom fields, objects and any necessary configurations to accommodate the incoming data. Ensuring that the destination environment is ready is essential for a smooth import process.

Field Mapping: The process of aligning the source data fields with the corresponding Salesforce fields. Proper field mapping is crucial for maintaining data integrity during the import.

Import to Sandbox: Importing data into the sandbox allows for validating the process and identifying any potential issues.

Decision Node: Is Testing Successful: After the sandbox import, the process evaluates whether the import was successful. If the testing is successful, the process can proceed to the next step. If not, it's necessary to address the issues encountered during the testing phase.

Import to Production: This step involves carefully executing the import to ensure that the live system is populated with accurate and clean data. It is recommended to use spreadsheet applications such as Excel to label the columns of the data to be imported before performing field mapping. This practice ensures clarity and accuracy when matching source fields with their corresponding Salesforce fields. The table below shows common fields in a data migration process, which are typically matched between a source system and Salesforce.

Table 4.2: Common Field Mapping for Data Import

Label for Import File	Salesforce Field
Birthdate	Contact: Birthdate
Contact Full Name or First Name & Last Name	Contact: First Name Contact: Last Name
E-mail Address	Contact: Email
Account Name	Account: Account Name Contact: Account
Industry	Account: Industry
Ownership	Account: Ownership
Billing City	Account: Billing City
Billing Country	Account: Billing Country
Website	Account: Website

In the table above, common field mappings are provided to demonstrate how data from an import file should be correctly aligned with corresponding Salesforce fields during a data migration. The 'Label for Import File' column represents typical headers or labels used in the source data file, while the 'Salesforce Field' column identifies the fields within Salesforce where this data should be imported. This mapping process is a crucial component of any data migration strategy. Proper field mapping ensures that each piece of data is accurately transferred to the right place within Salesforce, preserving the integrity and functionality of the system. If the correct fields are not accurately mapped to their corresponding objects, the resulting data inaccuracies could not only compromise the integrity of the information but also disrupt automated processes, hinder accurate reporting, and potentially jeopardize compliance with data governance standards that rely on precise and reliable data.

5. Conclusion

This study explores data management challenges within the Salesforce.com platform and analyzes strategies to overcome these issues, particularly focusing on data security, data integrity and data migration. Throughout the research, practical methods such as data profiling and cleansing, quality assessment, validation rules, and duplicate management were highlighted as essential practices for enhancing CRM performance.

The findings indicate that Salesforce.com users can achieve higher efficiency by adopting improved strategies for data security and quality. Emphasizing customer experience and data security can help organizations build resilience and better adapt to future digital transformations. This is aligned with insights from Experian's 2022 Global Data Management Research Report, which underscores the importance of data agility and security in today's rapidly evolving digital landscape. As highlighted in the report, many organizations are focusing on developing their data management programs to enhance security and customer experience, ultimately driving revenue growth and ensuring compliance with data protection regulations.

When managing data within Salesforce, users should pay particular attention to several best practices and tools that can significantly enhance data quality and security. Regular data profiling and cleansing routines are vital to maintaining data integrity. Implementing robust validation rules can prevent incorrect or incomplete data entries, while effective duplicate management strategies can streamline data processes and improve overall system efficiency.

Salesforce provides users with critical functionalities designed to simplify complex processes and enhance data integrity. Before deleting custom objects or fields, Salesforce's alert system notifies users of potential impacts, preventing accidental loss of valuable data. Validation rules enforce data entry standards, ensuring that users create accurate and complete records. These features significantly reduce the risk of data errors and enhance overall user experience by guiding users to follow best practices in data management.

In particular, SOQL and Apex can be leveraged to address various data management challenges and enhance overall platform performance. SOQL (Salesforce Object Query Language) is instrumental in data profiling and cleansing. It allows users to run targeted queries that identify inconsistencies, duplicates, or incomplete records within their CRM database. This capability helps maintain high data accuracy and relevance, which is crucial for making informed business decisions and improving operational efficiency.

Apex, Salesforce's proprietary programming language, plays a crucial role in automating complex data processes and enhancing performance. Developers can create custom triggers and classes to automate data validation, enforce business logic and manage data integration. Apex classes can automatically clean and standardize data upon entry, while custom validation rules ensure that only accurate and non-duplicate data is entered into the system. This not only enhances data integrity but also optimizes system performance by reducing manual errors and administrative overhead.

Data mapping and integration are also critical for ensuring that data flows accurately between Salesforce and other systems during migrations or ongoing operations. Proper data mapping aligns fields between systems, minimizing errors and improving the consistency and quality of integrated data. Integration tools facilitate seamless communication between Salesforce and external platforms, which helps maintain data consistency and enhances overall system performance.

To effectively manage large data volumes during data migration and integration within Salesforce, it is essential to follow a structured process, as depicted in the diagram. One of the key steps in this process is the use of a sandbox environment. By importing data first into a sandbox, organizations can thoroughly test and validate their data import procedures without impacting the production environment. This not only helps in identifying potential issues early but also ensures data integrity and accuracy before going live.

Additionally, the third-party AppExchange cloud marketplace provides a range of tools and solutions, including managed packages and consulting services, designed to enhance data management and improve Salesforce platform performance. Managed packages offer customizable, pre-built solutions tailored to specific needs, while consulting services deliver expert guidance for adapting Salesforce implementations to unique business requirements. By leveraging these resources, organizations can streamline data management processes, reduce manual efforts, and significantly boost the efficiency and effectiveness of their Salesforce implementations.

In conclusion, effective data management within Salesforce is pivotal for maintaining data quality, security and overall CRM performance. By adopting best practices in data profiling, cleansing, and validation, along with leveraging powerful tools like SOQL and Apex, organizations can significantly enhance their Salesforce implementations. As businesses continue to navigate the complexities of digital transformation, a strong focus on data management will be crucial for driving operational efficiency, ensuring compliance, and delivering exceptional customer experiences. The strategies and tools discussed in this study provide a comprehensive framework for addressing common data challenges, enabling organizations to unlock the full potential of their Salesforce platforms.

References

Boban, M., Ivkovic, M., Jevtic, V., & Milanov, D. (2015). The Data Quality in CRM Systems: Strategy and Privacy.

https://www.academia.edu/33011625/The_Data_Quality_in_CRM_Systems_Strategy_and_Privacy

Brea, E. (2023). A Framework for Mapping Actor Roles and Their Innovation Potential in Digital Ecosystems. *Technovation*, 125, Article 102783.

<https://doi.org/10.1016/j.technovation.2023.102783>

Emorphis. (2024). A Comprehensive Guide to Salesforce Data Migration. Retrieved June 25, 2024, from <https://blogs.emorphis.com/a-comprehensive-guide-to-salesforce-data-migration/>

Experian. (2022). Global Data Management Research: A Rapidly Transforming Market Relies on Agility in Data Management. Retrieved June 25, 2024, from <https://www.experian.co.uk/blogs/latest-thinking/wp-content/uploads/sites/13/2024/03/2022-global-research-report.pdf>

Forcetalks. (2023). Salesforce Data Cleansing Guide. Retrieved May 25, 2024, from <https://www.forcetalks.com/blog/salesforce-data-cleansing-guide/>

Gessner, M. (2023). Salesforce Instances vs Orgs vs Environments. Retrieved June 25, 2024, from <https://focusonforce.com/platform/salesforce-instances-vs-orgs-vs-environments/>

Joseph, R. (2024). Large Data Volumes (LDV) in Salesforce. Retrieved June 30, 2024, from <https://blogs.perficient.com/2024/01/30/large-data-volumes-ldv-in-salesforce/>

Livneh, H. (2024). Salesforce Data Security Challenges in Wake of the Recent Breach. Adaptive Shield. Retrieved June 9, 2024, from <https://www.adaptive-shield.com/blog/salesforce-data-security-challenges-in-wake-of-the-recent->

[breach](#)

Lloyd, A. (2023) Data Profiling Is Essential for Better Salesforce Data Quality.

Retrieved July 1, 2024 from <https://sfdq.co/what-is-data-profiling/#:~:text=Data%20profiling%20analyzes%20the%20fields,field%20usage%2C%20and%20value%20frequency>

Mahlamäki, T., Storbacka, K., Pylkkönen, S., & Ojala, M. (2020). Adoption of Digital Sales Force Automation Tools in Supply Chain: Customers' Acceptance of Sales Configurators. *Industrial Marketing Management*, 91, 162–173. <https://doi.org/10.1016/j.indmarman.2020.08.024>

McDonald, S. (2021). *Salesforce CRM: A Library Management Solution*. University of North Carolina at Chapel Hill. <https://doi.org/10.17615/r24y-as56>

Mouyal Amsalem, M. (2023). Development of a Data Migration Automation Tool from Salesforce to Salesforce. Polytechnic University of Catalunya, Barcelona School of Informatics.

<https://upcommons.upc.edu/handle/2117/335982>

Orun, M. (2024) Architect's Guide: Data Profiling to Assess and Monitor Data Reliability. Retrieved July 1, 2024, from

<https://www.salesforceben.com/architects-guide-data-profiling-to-assess-and-monitor-data-reliability/>

Pipino, L., Lee, Y., & Wang, R. (2002). Data Quality Assessment. *Communications of the ACM*, 45(4), 211-218 <https://doi.org/10.1145/505248.506010>

Sahni, D. (2023). Steps For Successful Salesforce Data Migration. Retrieved June 21, 2024, from <https://salesforcecodex.com/salesforce/steps-for-successful-salesforce-data-migration/>

Salesforce. (2024). Choosing a Method for Importing Data. Retrieved June 21, 2024, from

https://help.salesforce.com/s/articleView?id=sf.import_which_data_import_to_ol.htm&type=5

- Salesforce. (2024). Say Hello to Data Cloud. Retrieved June 14, 2024 from <https://www.salesforce.com/data/>
- Salesforce. (2024). Managing Validation Rules. Retrieved July 14, 2024, from https://help.salesforce.com/s/articleView?id=sf.fields_managing_field_validation.htm&type=5
- Salesforce. (2024). Multitenancy and Metadata Overview. Retrieved May 24, 2024, from https://developer.salesforce.com/docs/atlas.en-us.salesforce_large_data_volumes_bp.meta/salesforce_large_data_volumes_bp/ldv_deployments_concepts_multitenancy_and_metadata.htm
- Salesforce. (2024). Notes on Changing Custom Field Types. Retrieved June 10, 2024, from https://help.salesforce.com/s/articleView?id=sf.notes_on_changing_custom_field_types.htm&type=5
- Salesforce. (2024). Show Duplicate Records in Lightning Experience. Retrieved July 16, 2024, from https://help.salesforce.com/s/articleView?id=sf.duplicates_existing_setup.htm&type=5
- Salesforce. (2024). What is CRM (Customer Relationship Management)? Retrieved May 15, 2024, from <https://www.salesforce.com/eu/learning-centre/crm/what-is-crm/>
- Salesforce. (2023). State of Data Analytics. Retrieved May 15, 2024, from https://www.salesforce.com/content/dam/web/en_us/www/documents/research/state-of-data-analytics.pdf
- Salesforce. (2024). Use Managed Packages to Develop Your AppExchange Solution. Retrieved July 28, from https://developer.salesforce.com/docs/atlas.en-us.packagingGuide.meta/packagingGuide/managed_packaging_intro.htm
- Salesforce. (2024). Use the Data Mapping Visualizer. Retrieved June 24, 2024, from https://help.salesforce.com/s/articleView?id=sf.data_data_mapping_viz_use.htm

[tm&type=5](#)

Shanmugam, D. B., Dhilipan, J., Prabhu, T., Sivasankari, A. and Vignesh, A. (2023). The Management of Data Quality Assessment in Big Data Presents a Complex Challenge, Accompanied by Various Issues Related to Data Quality. <https://doi.org/10.9734/bpi/rhmcs/v8/18858D>

Tableau. (2024). Guide To Data Cleaning: Definition, Benefits, Components and How to Clean Your Data. Retrieved July 3, 2024, from <https://www.tableau.com/learn/articles/what-is-data-cleaning#tools-software>

Trailhead. (2024). Understand the Salesforce Architecture. Retrieved June 1, 2024, from https://trailhead.salesforce.com/content/learn/modules/starting_force_com/starting_understanding_arch

Trailhead. (2024). Report Data Privacy and Security Incidents. Retrieved May 30, 2024, from <https://trailhead.salesforce.com/content/learn/modules/data-privacy/report-data-privacy-security-incidents>

Unnikrishnan, A. (2023), Salesforce Data Migration 101 – Challenges & Best Practices. Retrieved June 25, 2024, from <https://www.saasguru.co/salesforce-data-migration-101/>

Ying-Yen, L. (2019). Knowledge Management Exploring the Fundamental Theory Effect of Corporate Knowledge Management for Strengthening Marketing Strategy Applications. International Journal of Organizational Innovation, 12(1), pp.220-241.