Data Migration

Svet Voloshin

Why migrate?

1. **Merging multiple Salesforce instances:** If a company has multiple Salesforce instances that are used by different business units with their own CRMs or Salesforce orgs, a data migration may be necessary to consolidate the data into a single instance.
2. **Org Splits**: A Salesforce org split refers to the process of dividing a single Salesforce instance (also known as an "org") into two separate instances for reasons of Compliance and data separation, Performance optimization, Departmental separation and other business reasons.
3. **Moving data from another CRM**: A company may want to move data from another CRM system into Salesforce, in order to take advantage of the platform's features and functionality.
4. **Archiving old data**: A company may want to move old, inactive data from their Salesforce instance to a separate archive, in order to improve performance and reduce storage costs.
5. **Data consolidation and clean-up**: A data migration can be used to consolidate and clean up data in Salesforce, such as by removing duplicates or standardizing data formats.
6. **Improving data accuracy and consistency**: A data migration can be used to improve the accuracy and consistency of data in Salesforce, by using automated tools to validate and correct data during the migration process.

7 Components of a Successful Data Migration

1. Build and maintain a data mapping workbook
2. Automate as much as possible
3. Stage your data in a relational database
4. Create an automated report of data quality issues
5. Implement lean loading for performance
6. Test frequently
7. Start early; iterate continuously

SFDC Data Model different then RDBMS - Normalization - Sub query (parent to child) - __r (child to parent - internal joins)

If Data Load is 24x7 or phased - No Down Time

Using Custom Setting at User , Profile level , Disable sharing rule, email, Trigger etc as it would impactne/ existing users
1. Send communication to existing user that they may hit performance
2. Have retry and queue mechanism for existing integration
3. Identify low usage time on basis of historical information , login history etc
4. Dark load data in advance and have Cut over plan in place to decide delta load, reuse same ETL script

Cut Over Plan

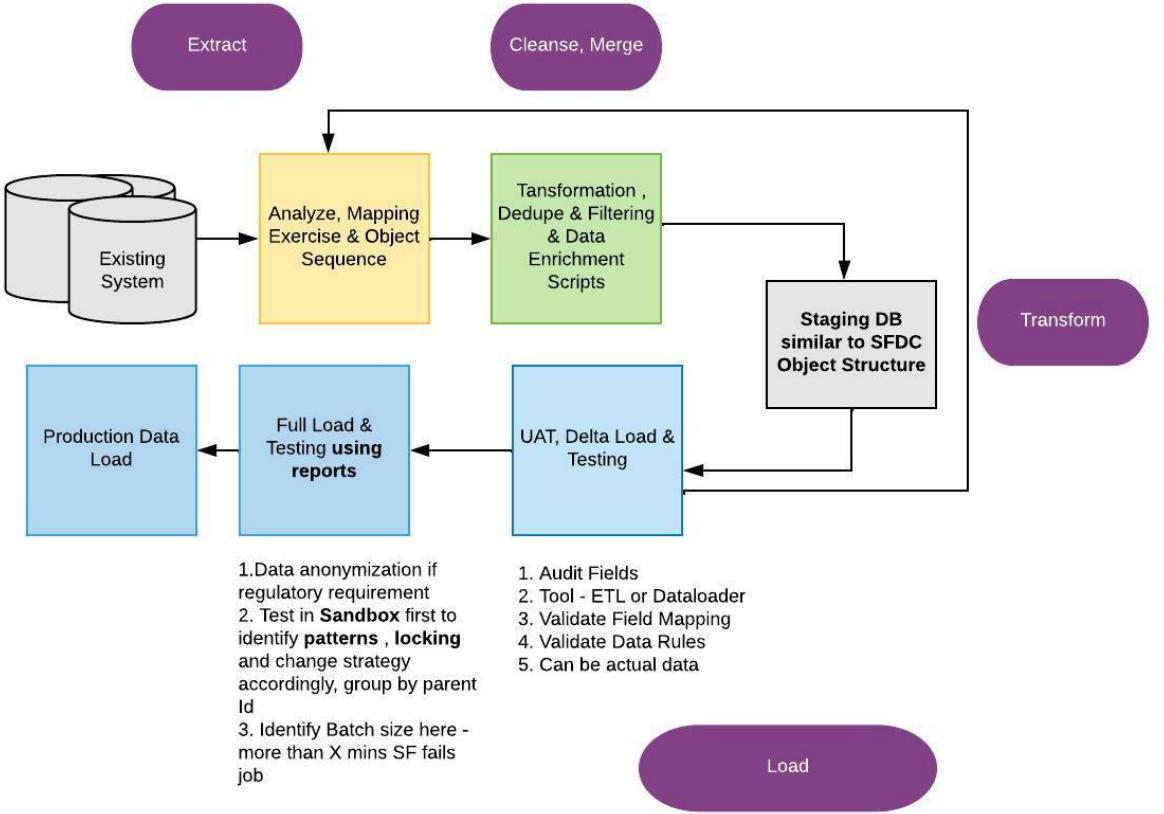
For Initial / one time load

1. **Defferred Sharing** & **Granular Locking**
2. Disable Outgoing emails
3. Load Users as Inactive
4. WF, Trigger, Validation Disable flags
5. Group records by **Parent** to avoid Locking errors
6. OWD R/W
7. **Bulk API** [Rest API] in parallel

Cut Over Plan

Delta Load

1. Load only Delta record
2. Possibilities of using standard API [SOAP] instead of Bulk API



General Steps

Here are the general steps for migrating data to Salesforce:

1. **Plan the migration:** Determine what data needs to be migrated, the sources of that data, and the target objects in Salesforce.
2. **Prepare the data:** Clean and format the data so that it can be easily loaded into Salesforce.
3. **Choose a migration tool:** Salesforce provides several data migration tools, including the Data Loader, Dataloader.io (premium) and the Salesforce API. Choose the tool that best suits your needs.
4. **Create a mapping:** Map the fields in the source data to the corresponding fields in the target objects in Salesforce.
5. **Validate the data:** Verify that the data is correct and complete before loading it into Salesforce.
6. **Load the data:** Use the migration tool to load the data into Salesforce.
7. **Monitor the migration:** Monitor the progress of the migration and check for any errors or issues.
8. **Test the data:** Verify that the data has been loaded correctly and that all relationships and dependencies have been maintained.
9. **Clean up:** Remove any duplicate records and resolve any errors that may have occurred during the migration.
10. **Go live:** Once the data has been successfully migrated and tested, you can go live with your new Salesforce implementation.

Migration Sequence

The sequence of objects during a Salesforce data migration depends on the specific requirements of your migration project and the relationships between the objects being migrated. However, here is a **general sequence** that can be followed:

1. **Reference objects:** Reference objects, such as picklist values or custom objects, should typically be created or updated first as they are often referenced by other objects.
2. **Parent objects:** Parent objects, such as accounts or opportunities, should be created or updated next as they are often the primary objects in a relationship.
3. **Child objects:** Child objects, such as contacts or opportunities, should be created or updated after the parent objects as they depend on the parent objects for their relationships.
4. **Detail objects:** Detail objects, such as quotes or line items, should be created or updated last as they are dependent on both the parent and child objects.
5. **Custom objects:** Custom objects, such as custom entities or custom fields, should be created or updated as needed to meet the specific requirements of your migration project.

It's important to note that the specific sequence of objects may vary depending on the size and complexity of your data, the migration tool being used, and the specific requirements of your migration project. The key is to identify the relationships between the objects and to create or update the objects in the order that maintains those relationships.

What is an ETL?

An ETL (Extract, Transform, Load) tool is used to move data from one or more source systems into a target system. The ETL process consists of three steps:

Extract: In this step, data is extracted from one or more source systems. This may involve reading data from databases, flat files, or cloud-based systems, among other sources. The extracted data is typically stored in a temporary location for processing.

Transform: In this step, the extracted data is transformed to match the structure and format of the target system. This may involve cleaning and standardizing the data, removing duplicates, and transforming data types.

Load: In this step, the transformed data is loaded into the target system. This may involve inserting data into a database, updating existing records, or creating new records, depending on the specific requirements of the target system.

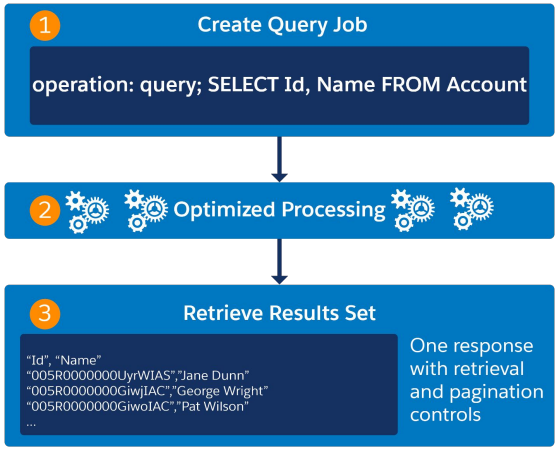
The ETL tool is responsible for automating the entire process of extracting data from source systems, transforming the data, and loading it into the target system. This reduces the time and resources required to move data, and helps to ensure that the data is accurate and up-to-date.

API

Bulk API: The Salesforce Bulk API is specifically designed for large-scale data migration and can handle millions of records at a time. It is optimized for high-volume data loads and is a great choice for data migration projects that require maximum performance.

Bulk API 2.0 allows for:

- Automatically performs **PK chunking**.
- Less client-side code writing.
- Easy-to-monitor job status.
- Automatic retry of failed records.
- Support for parallel processing.
- Fewer calls required to complete ingest or query workflows.
- Easier batch management.



Query Jobs

Bulk API 2.0 query jobs enable asynchronous processing of SOQL queries. Instead of manually configuring batches, Bulk API 2.0 query jobs **automatically determine the best way to divide your query job into smaller chunks**, helping to avoid failures or timeouts. **The API automatically handles retries.** If you receive a message that the API retried more than 15 times, apply a filter criteria and try again. When you get the results of a query job, the response body is always compressed.

With Bulk API, if you create a batch without PK chunking, then only one batch is created. If you create a batch with PK chunking enabled, batches are broken up based on the number of records in the chunk. This can range from 100,000 to 250,000 records.

PK Chunking

In the context of Salesforce, **PK Chunking** refers to the process of breaking up large data sets into smaller chunks for improved performance and efficiency when working with the data in the Salesforce platform. This is typically done when working with large amounts of data, such as when importing data from an external source or when working with large reports.

By breaking the data into smaller chunks, Salesforce can process the data more efficiently and avoid performance issues that can arise when working with very large data sets. The chunks of data can then be processed individually, allowing for better performance and a more streamlined experience when working with the data in Salesforce.

Third-Party Data Migration Tools

Jitterbit, Talend, and Informatica are all popular data integration and data migration tools that can be used to migrate data into Salesforce. Here is a comparison of these three tools:

- **Jitterbit:** Jitterbit is a cloud-based data integration platform that provides an easy-to-use interface for data migration projects. It offers drag-and-drop functionality for mapping data, making it a good choice for small to medium-sized migration projects. Jitterbit also supports real-time data integration and has pre-built connectors for several popular applications, including Salesforce.
- **Talend:** Talend is a powerful open-source data integration platform that provides a wide range of data management and migration capabilities. It supports both bulk and real-time data integration, making it a good choice for large-scale data migration projects. Talend also provides a comprehensive suite of data quality and data governance tools, making it a good choice for data-intensive projects.
- **Informatica:** Informatica is a leading data integration platform that provides a comprehensive suite of data management and migration capabilities. It supports both batch and real-time data integration, and provides a wide range of pre-built connectors for popular applications, including Salesforce. Informatica is a good choice for complex data migration projects, as it provides a rich set of data quality and data governance tools, as well as support for advanced data management scenarios.

Capstorm vs. Informatica

Functionality: Informatica provides a comprehensive suite of data integration and data migration capabilities, with a focus on data quality, data governance, and data security. Capstorm, on the other hand, provides a more streamlined set of data integration tools that are designed to be easy to use and flexible. **Capstorm does the "E" and the "L", but not the "T" (transform) - this makes it useful for Salesforce-to-Salesforce Org Migration.**

Scalability: Informatica's solutions are designed to handle large-scale data integration projects and can accommodate complex data management requirements. Capstorm's solutions, while flexible, may be more suited to small to medium-sized data integration projects.

Integration with Salesforce: Both Capstorm and Informatica provide solutions for integrating with Salesforce, but Informatica provides a more comprehensive set of tools for integrating with Salesforce, including pre-built connectors and a range of data management and data governance capabilities.

Cost: Informatica is typically more expensive than Capstorm, as it provides a more comprehensive suite of data integration tools and services.

Xappex XL-Connector

Xappex XL-Connector is a tool for integrating Microsoft Excel with Salesforce, allowing users to easily connect Microsoft Excel with Salesforce.

It enables users to manage their Salesforce data directly from within Microsoft Excel, without the need for complex coding or scripting.

The XL-Connector provides a simple and intuitive interface for mapping data between Salesforce and Excel, and supports a range of data operations including imports, exports, updates, and deletes.

ETL Databases

1. An ETL tool can be used with a wide range of databases, including:
2. **Relational databases:** This includes databases such as Oracle, SQL Server, MySQL, and PostgreSQL, among others.
3. **NoSQL databases:** This includes databases such as MongoDB, Cassandra, and Couchbase, among others.
4. **Cloud-based databases:** This includes databases such as Amazon Web Services (AWS) RDS, Google Cloud SQL, and Microsoft Azure SQL Database, among others.
5. **Mainframe databases:** This includes databases such as IBM DB2, among others.

ETL Staging Database

An ETL **staging database** is a **temporary database** used in the ETL process to store data as it is being extracted from source systems, transformed, and loaded into a target system. The staging database serves as a holding area for the data and provides a place for the ETL process to manipulate the data as needed.

The staging database typically has a structure that is different from the source and target systems, allowing for data to be transformed and reformatted as needed. The data stored in the staging database is used as the basis for the final data load into the target system, and is typically discarded after the load is complete.

The use of a staging database is an important step in the ETL process, as it allows for data to be transformed and validated before it is loaded into the target system. This can help to improve the accuracy of the data and reduce the risk of data loss or corruption during the data migration process.

In conclusion, an ETL staging database is a critical component of the ETL process, providing a temporary location for data to be transformed, validated, and loaded into a target system.

ETL Canonical Database

An ETL canonical database is a database used to store a single, standardized representation of data from **multiple sources**. The purpose of the canonical database is to provide a **common data model** for all data sources, which can be used as the basis for data integration and migration projects.

The canonical database is typically used in **conjunction with an ETL tool and a staging database to support data migration projects**. Data is extracted from source systems and loaded into the staging database, where it is transformed to match the structure of the canonical database. Once the data is transformed, it is loaded into the canonical database, where it can be used as a single source of truth for all data integration projects.

The use of a canonical database can help to simplify the data migration process by providing a consistent structure and format for data from multiple sources. This can help to reduce the time and effort required to integrate data from multiple sources and can improve the accuracy of the data by reducing the risk of data loss or corruption during the data migration process.

In conclusion, an ETL canonical database is a database used to store a standardized representation of data from multiple sources, providing a common data model for data integration and migration projects.

Large Data Volumes (LDV)

Handling large volumes of data is a common challenge in the ETL process. Here are some common techniques used to handle large data volumes in ETL:

1. **Parallel Processing:** ETL tools can divide large data volumes into smaller chunks and process them in parallel, which can significantly speed up the data processing time.
2. **Data Sampling:** ETL tools can use data sampling techniques to extract a subset of the data for processing, which can help to reduce the amount of time and resources required to process the data.
3. **Incremental Loading:** ETL tools can be configured to only extract and process data that has changed since the last load, reducing the amount of data that needs to be processed.
4. **Caching:** ETL tools can cache frequently used data in memory, reducing the amount of time required to access the data and speeding up the data processing time.
5. **Compression:** ETL tools can use data compression techniques to reduce the size of the data being processed, reducing the amount of time required to extract, transform, and load the data.
6. **Indexing:** ETL tools can use indexing techniques to improve the performance of data retrieval operations, reducing the amount of time required to extract and transform the data.

In conclusion, there are several techniques that can be used to handle large volumes of data in the ETL process, including parallel processing, data sampling, incremental loading, caching, compression, and indexing. The specific techniques used will depend on the specific requirements of the data migration project and the volume of data being processed.

Salesforce LDV Strategies

LDV strategies refer to the strategies for Large Data Volumes in Salesforce. These strategies help organizations manage and process large amounts of data effectively in Salesforce. Some common LDV strategies include:

1. Using the appropriate data model to optimize storage and processing.
2. Enabling data archiving and purging to reduce the amount of data stored in Salesforce.
3. Using Big Objects to store large amounts of data that do not need to be regularly queried.
4. Using external data storage solutions like a data warehouse, to store large amounts of data and integrate with Salesforce through ETL (Extract, Transform, Load) processes.
5. Optimizing custom objects, fields, and indexes to reduce query times and improve performance.
6. Using the bulk API or the Data Loader to efficiently insert or update large amounts of data in Salesforce.

These strategies can help organizations ensure efficient data processing and storage, while maintaining good performance in Salesforce.

Salesforce Data Archiving

Salesforce data archiving is the process of moving data that is no longer needed or used regularly from the production environment to a separate, archive environment. This helps to improve the performance and reliability of the production environment by reducing the amount of data stored in it.

Archiving data in Salesforce can be performed using the following methods:

1. **Custom Archiving Solutions:** Custom archiving solutions can be developed to meet the specific requirements of a data archiving project. These solutions can be developed using Apex or other programming languages and can be configured to move data from the production environment to the archive environment based on specific business rules.
2. **Third-party Tools:** There are many third-party tools available that can be used to archive data in Salesforce, including popular ETL tools, such as Informatica. These tools provide advanced features and capabilities, such as the ability to automate the data archiving process, and provide detailed reports.

When to archive data?

Organizations may choose to archive Salesforce data in order to reduce the amount of data stored in the production environment and improve the performance and reliability of their systems. Some common scenarios where data archiving may be appropriate include:

1. **Old or Inactive Records:** Archiving old or inactive records, such as old leads or cases, can help to reduce the amount of data stored in the production environment and improve performance.
2. **Large Data Volumes:** Archiving large data volumes, such as large numbers of attachments or large numbers of custom objects, can help to reduce the amount of data stored in the production environment and improve performance.
3. **Compliance Requirements:** Archiving data that is no longer needed or used regularly, such as old financial records or customer data, can help organizations meet their legal and regulatory obligations by preserving data that may be required for audits or legal proceedings.
4. **Data Retention Requirements:** Archiving data that is no longer needed or used regularly can help organizations meet their data retention requirements by preserving data that may be required for future reference.

In conclusion, organizations may choose to archive Salesforce data when they have old or inactive records, large data volumes, compliance requirements, or data retention requirements, in order to reduce the amount of data stored in the production environment and improve the performance and reliability of their systems.

ETL to Data Warehouse (DW) Offloading

ETL to data warehouse offloading is a process used to transfer data **from operational systems to a data warehouse for analysis and reporting**. The process involves extracting data from source systems, transforming the data to match the structure of the data warehouse, and loading the transformed data into the data warehouse.

The goal of ETL to data warehouse offloading is to create a separate repository of data that is optimized for analysis and reporting. The data warehouse is designed to handle large volumes of data and provide quick and efficient access to that data, making it ideal for business intelligence and data analytics applications.

The ETL process is performed on a regular basis, typically daily or weekly, to ensure that the data in the data warehouse is up-to-date. The frequency of the ETL process will depend on the specific requirements of the data warehouse and the volume of data being processed.

The data warehouse is populated with data from multiple source systems, including transactional systems, operational databases, and flat files. The data is transformed to match the structure of the data warehouse and is loaded into the warehouse as a single, integrated view of the data.

In conclusion, ETL to data warehouse offloading is a process used to transfer data from operational systems to a data warehouse for analysis and reporting. The goal of the process is to create a separate repository of data that is optimized for analysis and reporting and to provide a single, integrated view of data from multiple sources.

Data Warehouses

Some of the most popular and widely used data warehouses include:

1. **Amazon Redshift:** A fast and scalable data warehouse that is part of the Amazon Web Services (AWS) ecosystem.
2. **Google BigQuery:** A cloud-native data warehouse that provides fast, interactive SQL queries on massive datasets.
3. **Microsoft Azure Synapse Analytics** (formerly SQL Data Warehouse): A cloud-based data warehousing solution that combines the power of big data analytics with the familiarity of SQL.
4. **Snowflake:** A cloud-based data warehousing solution that provides fast and flexible data warehousing with separate compute and storage layers.
5. **Teradata:** A traditional data warehousing solution that provides high performance and scalability for large enterprises.

DW vs. Heroku

Data Warehouse and Heroku are two different technologies that serve different purposes.

Data Warehouse: A data warehouse is a central repository for storing structured, processed data that has been optimized for querying and analysis. Data warehouses are typically used for business intelligence and decision-making purposes, and are designed to support ad hoc querying and reporting.

Heroku: Heroku is a cloud platform as a service (PaaS) that provides a variety of services for building, running, and managing web applications. Heroku includes a **database as a service offering, Heroku Postgres**, which can be used for **storing and managing structured data**. However, Heroku Postgres **is not designed specifically for data warehousing** and may not provide the same level of performance, scalability, and reliability as a dedicated data warehouse solution.

In summary, data warehouses are **specifically designed for data warehousing and business intelligence**, while Heroku is a general-purpose cloud platform that includes a database service. The choice between a data warehouse and Heroku depends on the specific use case and the type and format of the data being stored.

DW vs. Data Lake

A data warehouse and a data lake are two distinct data storage and management systems that are used for different purposes.

Data Warehouse: A data warehouse is a central repository for storing structured, processed data that has been optimized for querying and analysis. Data warehouses are typically used for business intelligence and decision-making purposes, and are designed to support *ad hoc* querying and reporting.

Data Lake: A data lake is a central repository for storing **large amounts of raw, unstructured, and semi-structured data**. Data lakes are designed to store data in its **original format, without transforming or aggregating the data**. Data lakes are typically used for **big data and data science applications**, and are designed to support batch and real-time processing of large volumes of data.

In summary, data warehouses are optimized for querying and analysis of structured data, while data lakes are optimized for the storage and processing of large volumes of raw and unstructured data. The choice between a data warehouse and a data lake depends on the specific use case and the type and format of the data being stored.

Salesforce Big Objects

Salesforce Big Objects have several limitations that organizations need to consider before using them for data storage and management:

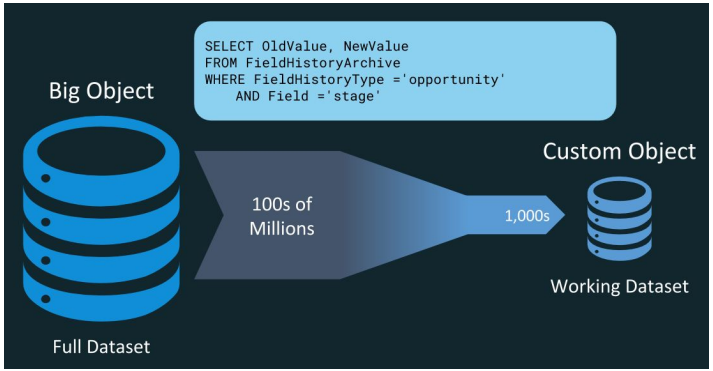
1. **Capacity**: Big Objects have a maximum capacity of **50 billion rows per object**, which may not be sufficient for organizations with extremely large data sets.
2. **Performance**: Big Objects can have slower query performance compared to other data storage options in Salesforce, especially when querying large amounts of data.
3. **Data Modeling**: The data modeling capabilities for Big Objects are limited, and it may not be possible to implement complex relationships or custom data structures.
4. **Data Management**: Big Objects do not support some of the data management features available in other parts of the Salesforce platform, such as record-level security or custom indexes.
5. **Data Access**: Big Objects can only be accessed through the Salesforce API, and there is **no user interface** for accessing or managing the data stored in Big Objects.
6. **Integration**: Integration with other parts of the Salesforce platform and with external systems can be complex, and may require **additional development work**.
7. **Cost**: Big Objects can be more expensive than other data storage options in Salesforce, especially for organizations with large data sets.
8. **API Support for Big Objects**: Bulk, Chatter and SOAP APIs. These APIs are the only APIs supported for big objects. The REST API, for example, isn't supported.
9. **Async SOQL:** a powerful tool for asynchronously processing massive volumes of data – read on for more details.

Async SOQL

Async SOQL is a method for running SOQL queries when you can't wait for immediate results. These queries are run in the background over Salesforce big object data. Async SOQL provides a convenient way to query large amounts of data stored in Salesforce.

Use Async SOQL when:

- You are querying against millions of records.
- You want to ensure that your query completes.
- You do not need to do aggregate queries or filtering outside of the index.



Async SOQL is scheduled for retirement in all Salesforce orgs as of Spring '23.

Data Virtualization

Data virtualization in Salesforce refers to the process of creating a unified, abstracted view of data that is stored in multiple disparate sources, including **external** Salesforce objects, databases, and other systems.

This technique can be used to provide a single point of access to data stored in other systems, simplifying the process of integrating data for business intelligence and decision-making purposes. Data virtualization in Salesforce can also help improve the performance of data warehousing by reducing the amount of data that needs to be physically stored in the Salesforce platform, and by improving the scalability and flexibility of the data warehousing environment.

Data virtualization in Salesforce can be achieved through the use of **Salesforce Connect**, a feature that allows data to be accessed and integrated from external sources into the Salesforce platform. Other data integration tools and technologies, such as ETL tools or data virtualization platforms, can also be used to implement data virtualization in Salesforce, depending on the specific requirements and needs of the organization.

It is important to note that data virtualization in Salesforce should be implemented with data governance and security in mind, as it involves accessing and integrating data from multiple sources. It is also important to consider the impact on performance and scalability, as virtualizing data can add additional complexity and overhead to the data integration process.

Data Cleansing with DemandTools

DemandTools is a third-party data management tool for Salesforce that provides a suite of data management and data cleansing features. Some of the key features of DemandTools include:

- Data deduplication: Automatically find and merge duplicate records to maintain a single, accurate view of your data.
- Data cleansing: Correct inaccuracies and inconsistencies in your data, such as incorrect values, misspellings, and formatting issues.
- Data enrichment: Enhance your data by adding missing information, such as addresses, phone numbers, and email addresses.
- Data migration: Migrate data between Salesforce instances, including migrations between sandbox and production environments.

DemandTools is designed to be an easy-to-use, cloud-based solution that can help organizations of all sizes manage their Salesforce data. With its powerful features and intuitive interface, it can help organizations improve the quality of their data, reduce the time and effort required to manage data, and support better decision making.

Data Quality Validation

Data quality validation for migration is an important step in the data migration process that ensures that the data being migrated is accurate, complete, and consistent. The following are common steps in data quality validation for migration:

1. **Data validation:** After the data has been cleaned and mapped, it can be validated to ensure that it meets the quality and accuracy requirements of the target system. This may involve verifying that the data meets specific data quality criteria, such as correct data formats or value ranges.
2. **Data reconciliation:** Finally, the migrated data can be reconciled with the source data to ensure that all of the data has been migrated successfully and that there are no discrepancies between the source and target systems.

Data quality validation for migration is a critical step that helps to ensure that the data being migrated is accurate and consistent, which is essential for making informed decisions and avoiding errors in the target system.

Data Transformation

Data transformation is the process of converting data from its source format into a format that can be stored and analyzed in a target system. It is an important step in the ETL process that helps to ensure that the data being stored in the target system is accurate, consistent, and ready for analysis. The following are the key aspects of data transformation:

1. **Data aggregation**: This involves combining multiple data sources into a single, cohesive view. This may involve combining data from different databases, data sources, or APIs.
2. **Data enrichment**: This involves adding additional information to the source data to make it more valuable for analysis. This may involve adding geographical information, demographic information, or other relevant data.
3. **Data reduction**: This involves reducing the size of the source data to make it easier to store and analyze. This may involve aggregating values, removing irrelevant data, or compressing the data.

Data transformation is a **complex and time-consuming process** that requires specialized tools and techniques to be executed effectively. ETL tools and data integration platforms provide a wide range of data transformation capabilities, including data mapping, data cleaning, data enrichment, and data reduction, to help organizations efficiently transform their data into a format that is ready for analysis.

Mulesoft Dataweave

MuleSoft Dataweave is a data transformation language that is used to transform data between various formats, such as JSON, XML, and CSV. It is integrated into the Mule runtime engine and is used to transform data within Mule applications. Dataweave provides a powerful, flexible, and user-friendly way to transform data, making it a popular choice for data integration and data migration projects.

Dataweave uses a simple and intuitive syntax, making it easy for developers to write and understand data transformation scripts. It supports a wide range of data types and provides functions for transforming, mapping, and filtering data. Dataweave also supports a variety of data sources, including databases, REST and SOAP web services, and files.

In addition to its data transformation capabilities, Dataweave also provides a number of built-in functions for working with dates, strings, numbers, and other data types, as well as support for complex data structures, such as arrays and objects. This makes it a versatile tool for working with data and transforming it into the desired format.

Data Enrichment

Salesforce data enrichment refers to the process of adding additional data to existing records in Salesforce to make the information more complete and accurate. This can be achieved through a variety of methods, including:

1. **Manually adding data**: This involves manually entering data into Salesforce using the user interface, such as adding information to a contact record or updating an account record.
2. **Importing data from external sources**: This involves importing data from external sources such as spreadsheets or databases, and merging it with existing data in Salesforce.
3. **Integrating with third-party systems**: This involves integrating Salesforce with third-party systems, such as marketing automation tools, to automatically add data to Salesforce records.
4. **Using APIs and custom code**: This involves using APIs and custom code to add data to Salesforce records programmatically.

Data enrichment is important because it helps to ensure that Salesforce data is complete and up-to-date, which in turn supports better decision-making, improves customer experiences, and enhances overall efficiency. By enriching Salesforce data, organizations can gain a more comprehensive view of their customers, leads, and prospects, and use this information to drive better business outcomes.

Delta Load

Delta load refers to the process of **only loading the changes or differences in data into a target system**, rather than loading all the data each time. This approach is often used in data migration or data integration scenarios where large amounts of data are being transferred between systems. Delta load helps to improve the efficiency of the data migration process by reducing the amount of data that needs to be transferred, processed, and stored.

Delta load is typically implemented by comparing the source data with the target data to identify changes and updates. Only the differences or new data are then loaded into the target system. This process can be repeated periodically to ensure that the target system remains up-to-date with the source data.

Delta load is particularly useful for scenarios where data is constantly changing or being updated, as it helps to ensure that the target system remains in sync with the source data without having to transfer all the data each time. It can also help to reduce the load on the source system, as well as the risk of data loss or corruption during the data migration process.

Data Migration Testing

Data migration testing is a critical step in the data migration process, especially when migrating to Salesforce. The purpose of data migration testing is to verify that the data has been transferred correctly, completely, and accurately from the source system to the target system (Salesforce). It helps to ensure that the data is consistent, clean, and usable after the migration.

Here are some common steps involved in data migration testing in Salesforce:

1. **Test Plan:** Create a test plan that outlines the scope, objectives, and approach of the data migration testing.
2. **Sample Data Selection**: Choose a representative sample of data to test, as it is not feasible to test all the data.
3. **Data Validation**: Validate the data to ensure that it has been transferred correctly, completely, and accurately. This may include comparing the source data with the target data, validating the data relationships, and verifying the data values.
4. **Data Integrity Testing**: Test the data integrity to ensure that it meets the data quality standards. This may include checking for duplicates, null values, and data consistency.
5. **User Acceptance Testing (UAT)**: Engage end-users to perform UAT to validate the data migration from the business perspective. This will help to identify any issues that may affect the business operations.
6. **Performance Testing**: Test the performance of the Salesforce system after the data migration to ensure that it can handle the increased volume of data.
7. **Post-Migration Clean-up**: Clean up any remaining data migration issues after the testing is complete.

It is important to perform thorough data migration testing to minimize the risk of data loss, corruption, or inconsistency during the migration. It will also help to ensure a successful and smooth transition to the target system (Salesforce).

Data Migration vs. Data Backup/Restore

Data migration and data backup are **two distinct processes** in information technology that **serve different purposes**.

Data migration refers to the process of **moving data from one location to another**, such as from one database, application, or system to another. The purpose of data migration is to move data from an older system to a newer system, or to consolidate data from multiple systems into a single system.

Data backup, on the other hand, is the process of **creating a copy of data for the purpose of preserving it and making it available in the event of data loss**. The purpose of data backup is to provide a way to restore data that has been lost due to a system failure, data corruption, or other disaster.

Backup & Restore for Data Migration?

Backup and restore can be used as a method for migrating data in some cases. By taking a backup of the data in one location and then restoring it in a new location, you can effectively migrate the data from one place to another.

However, this method is **not always suitable** for all types of data migration, and there are several factors that must be considered, such as the **format and structure of the data, compatibility with the target system, and the size of the data set**. Additionally, backup and restore methods can be time-consuming and may require specialized software or technical expertise to complete the process.

In some cases, it may be more appropriate to use a specialized data migration tool or an extract, transform, load (ETL) process, as these methods can offer more control and flexibility over the migration process, and are often better suited to handle large or complex data sets.

Data Migration Security

Here are a few measures that can be taken to ensure the security of data during migration:

1. **Encryption**: Data can be encrypted both **in transit and at rest** to protect it from unauthorized access.
2. **Access controls**: Access to the data during migration should be restricted to authorized personnel only.
3. **Data validation**: Data should be validated and checked for accuracy before and after migration to ensure that it has not been altered or corrupted during the process.
4. **Security documentation**: A comprehensive security plan should be developed that includes documenting all security measures in place and the roles and responsibilities of each person involved in the migration.
5. **Monitoring**: The migration process should be monitored regularly to detect any security incidents or anomalies.
6. **Data backup**: Regular backups of the data should be taken to ensure that the data can be restored in case of a security incident or data loss.
7. **Third-party tools**: Specialized third-party tools can be used to automate and manage the migration process, as well as to ensure data security and compliance with regulatory requirements.

Data Encryption (SSL & TLS)

Salesforce **data encryption in transit** refers to the protection of data transmitted between the client and the Salesforce servers. This encryption is used to secure the data from unauthorized access and tampering while it is being transmitted over the network. Salesforce implements TLS encryption for all client-server communications to ensure that data is transmitted securely.

SSL (Secure Sockets Layer) and TLS (Transport Layer Security) are protocols for secure communication over the internet.

- SSL was the first widely-used protocol for secure communication, but it has since been **replaced by TLS** due to security vulnerabilities in the protocol.
- **TLS is the successor to SSL** and is now the most widely-used protocol for secure communication on the internet. It provides encryption, authentication, and data integrity to protect data transmitted over the network.

Both SSL and TLS work by establishing a secure, encrypted connection between a client and a server, and both are commonly used to secure web communications, email, and other forms of network communication. However, TLS provides stronger security and is generally considered **more secure than SSL**.

ETL & Mutual TLS (MTLS)

When integrating Salesforce with an ETL tool, mutual TLS can be used to ensure secure communication between the two systems. The use of mutual TLS can help to ensure that the data being transferred between Salesforce and the ETL tool is encrypted and that the **identity of both parties is verified**.

When mutual TLS is enabled, the ETL tool and Salesforce will **both present digital certificates** to each other during the initial connection setup. These certificates are used to authenticate the identity of the two systems and to establish an encrypted connection for data transfer. This helps to ensure the confidentiality, integrity, and authenticity of the data being transferred.

Using mutual TLS for Salesforce and ETL integration can help to prevent unauthorized access to sensitive data, protect against data theft, and ensure regulatory compliance.

ETL Placement vis-a-vis the Firewall

The placement of the ETL (extract, transform, load) tool in relation to the firewall depends on the specific requirements and security policies of the organization. Generally, the ETL tool can be placed either in front of or behind the firewall.

Placing the ETL tool behind the firewall, within the network's secure boundary, may provide additional security for the organization's sensitive data. On the other hand, placing the ETL tool in front of the firewall may be more convenient for data integration with external systems.

ETL Scripts

In the context of salesforce data migration, ETL scripts refer to a **set of instructions that automate the extraction, transformation, and loading of data** from a source system to a target system (such as Salesforce).

These scripts are typically created using an ETL tool, and they handle the mapping of data from the source to the target, data cleaning and validation, and other activities that ensure that the data is properly loaded into the target system.

The specific steps involved in creating ETL scripts for salesforce data migration will vary depending on the tool being used, the complexity of the data, and other factors.

SOQL Queries

- SOQL (Salesforce Object Query Language) is a query language used to retrieve data from Salesforce objects.
- During data migration, SOQL can be used to extract data from a source system, such as a database, and then import it into Salesforce.
- SOQL provides a way to filter, sort, and aggregate data based on specified conditions, making it a valuable tool for data migration projects that require a high degree of control over the data being transferred.
- However, it is important to be mindful of the limitations of SOQL, such as query governor limits, and plan the data migration accordingly to ensure the process runs smoothly and efficiently.

Deferring Sharing Calculations

Defer Sharing Calculations is a feature in Salesforce that allows for the calculation of sharing rules for a set of records to be deferred until a later time. This feature can be useful for improving performance during data migration or bulk data operations, as the calculation of sharing rules can take a significant amount of time and slow down the overall process. By deferring the calculation, administrators can perform the data migration or bulk data operation more quickly, and then run the sharing calculation as a separate operation.

The defer sharing calculation feature isn't enabled by default. To enable it for your organization, contact Salesforce Customer Support.

Relaxing OWD

Relaxing OWD (Organization-Wide Defaults) in Salesforce refers to changing the default sharing settings for an object, such as Accounts, Contacts, Leads, etc., from "Private" to "Read" or "Read/Write." This is done to allow users in different roles and profiles to access and view the data stored in these objects. Relaxing OWD can be useful during a data migration as it can help facilitate data transfer and make sure the necessary data is accessible for migration. However, it is important to be cautious and consider the security implications of relaxing OWD before making any changes.

Group by Parent Id

In Salesforce data migrations, grouping records by parent ID is a common practice to ensure that the parent-child relationships between records are maintained. This involves organizing the data into hierarchical relationships, where a parent record has a unique ID and its child records reference the parent's ID as a foreign key.

The grouping helps to ensure that the parent-child relationships are preserved during the migration process, so that the data remains organized and accessible after the migration is complete.

To group records by parent ID in Salesforce data migrations, it's common to use tools such as an ETL tool or a Salesforce API to extract the data from the source system, organize it into parent-child relationships, and then load the data into the target system, preserving the relationships between records.

Parallel Sharing Rule Recalculation

In Salesforce, when a change is made to a record that affects its sharing rules, the platform recalculates the sharing rules to determine who can access the record.

When multiple records are updated, this recalculation process can become time-consuming, especially if the number of records and sharing rules is large.

To address this issue, Salesforce provides a feature called "**Parallel Sharing Rule Recalculation**," which allows you to **recalculate multiple records in parallel**, rather than serially.

This feature can significantly improve the performance of record sharing recalculations, making it possible to process larger numbers of records more quickly. To enable parallel sharing rule recalculation, you need to have a Salesforce administrator enable the feature in your organization's settings.

Locking Errors

Locking errors can also occur during a data migration to Salesforce if multiple users are attempting to update the same record at the same time. These errors can disrupt the migration process and result in data loss or data inconsistency.

To prevent locking errors during migration to Salesforce, you can take the following steps:

1. Schedule the migration during off-hours or low-usage times to minimize the risk of concurrent updates.
2. Temporarily disable record-level locking or set a longer lock timeout to reduce the chances of a lock conflict.
3. Use a dedicated migration user to perform the migration, so that the migration process is not disrupted by other users making updates to the records.
4. Monitor the migration process closely and resolve any locking errors that occur as soon as possible.
5. Consider using a third-party tool for data migration, as these tools often have built-in features for handling locking errors and ensuring data consistency during migration.

It's important to have a well-planned and tested data migration process to avoid any disruptions and ensure a smooth transition to Salesforce.

Granular Locking

Granular locking during data migration refers to a process where records in Salesforce are locked on a per-object or per-record level during a migration operation, to prevent conflicts and ensure data integrity.

This is especially important when multiple users are accessing the same data simultaneously, or when the data being migrated is critical to business operations.

By using granular locking, administrators can control the level of access to specific records and ensure that data is not lost or overwritten during a migration.

This helps to minimize downtime and reduce the risk of data corruption during the migration process.

External IDs

Salesforce External IDs are custom fields that are used to uniquely identify records in Salesforce. They are used as the basis for relationships between objects and can be used in data migration as a way to map data between systems.

In a data migration scenario, the external ID is used to **match data in the source system with data in the target system**, ensuring that the correct data is being moved and relationships are being maintained.

When migrating data to Salesforce, it is important to make sure that the External ID fields are set up correctly and populated with the correct data to ensure a successful migration.

Preventing Data Loss

Here are a few best practices for preventing data loss during Salesforce data migration:

1. **Create a backup of the data**: Before migrating the data, make sure to create a backup of the existing data in the source system. This will ensure that you have a copy of the original data in case of any data loss during migration.
2. **Clean and standardize data**: Clean and standardize the data before migrating it to Salesforce. This will help reduce the risk of data loss and ensure that the data is in the correct format for migration.
3. **Test in a sandbox environment**: Test the data migration process in a sandbox environment before migrating to production. This will help identify any potential issues with the data migration process and prevent data loss.
4. **Plan for data mapping and transformations**: Plan for data mapping and transformations in advance. This will ensure that the data is migrated accurately and reduce the risk of data loss.
5. **Automate the migration process**: Automating the data migration process will help ensure that the migration is completed accurately and efficiently.
6. **Validate data after migration**: After migrating the data, validate the data in Salesforce to ensure that it has been migrated correctly and that no data loss has occurred.
7. **Document the migration process**: Document the data migration process, including any data mapping and transformations, so that the process can be repeated if necessary.

By following these best practices, you can reduce the risk of data loss during Salesforce data migration and ensure that the migration is completed successfully.

Data Anonymization

Data anonymization is the process of masking or removing sensitive information from data sets, while still retaining enough information to perform analytical and reporting tasks. In Salesforce, data anonymization can be achieved by using tools such as Field Level Security, Record Types, Page Layouts, and validation rules, or through custom development.

ETL (Extract, Transform, Load) data masking in Salesforce involves obscuring sensitive information in a data set as it is being transferred from one system to another. This is done to protect privacy and security while still allowing the data to be used for testing, development, and analytical purposes.

In Salesforce, data masking during the ETL process can be achieved through the use of third-party tools such as Informatica Data Masking, Talend Data Masking, and MuleSoft Anonymizer. These tools allow you to automate the data masking process and control the level of detail in the masked data. They can also be integrated with Salesforce to ensure that sensitive data is masked before it is loaded into the system.

Custom development can also be used to build custom data masking solutions that can be integrated with the Salesforce ETL process. This approach allows for greater flexibility in masking data and can also be tailored to specific data protection requirements.

Regardless of the method used, data masking during the ETL process is an important step in ensuring that sensitive data is protected while still allowing it to be used for business purposes.

“Freezing” Users

During a data migration to Salesforce, freezing users can be a helpful strategy for ensuring data consistency and reducing the risk of data loss. When a user's account is frozen, they cannot log in or access any data within the Salesforce environment, reducing the chances of concurrent updates or other data inconsistencies during the migration process.

"Freezing" a user in Salesforce refers to deactivating a user account, which revokes their access to the platform. When a user's account is frozen, they cannot log in or access any data within the Salesforce environment.

Freezing a user in Salesforce is described in official documentation:

> Freeze or unfreeze multiple Users at once

Once a user's account is frozen, they will not be able to log in or access any data within Salesforce. This is a useful tool for administrators to manage user access, as it provides a way to temporarily revoke a user's access to the platform without having to delete their account and all of their associated data.

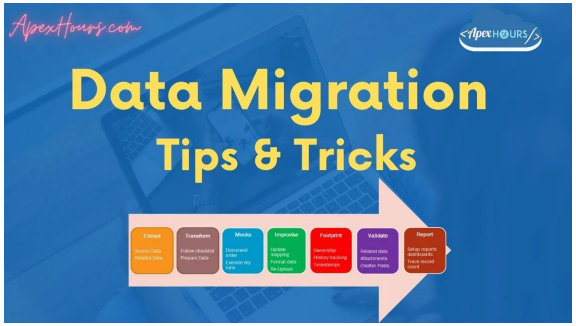
It's important to note that freezing a user in Salesforce only revokes their access to the platform, it does not delete their data or their user record. To reactivate a frozen user, simply change their user status back to "Active" and they will regain access to the platform.

Choosing the best ETL

The best ETL tool depends on the specific needs and requirements of your organization. There are many factors to consider when choosing an ETL tool, such as:

1. **Data sources**: Consider the types of data sources you need to extract data from, such as databases, web services, or files.
2. **Data volumes**: Consider the amount of data you need to process and whether the ETL tool can handle large volumes of data.
3. **Data complexity**: Consider the complexity of the data you need to extract, transform, and load, and whether the ETL tool can handle this complexity.
4. **Data formats**: Consider the formats of the data you need to process, such as CSV, XML, or JSON, and whether the ETL tool can handle these formats.
5. **Integration requirement**s: Consider your integration requirements, such as whether you need to integrate with other systems or whether you need to support real-time data integration.
6. **Scalability**: Consider whether the ETL tool can scale to meet the growing needs of your organization.
7. **Cost**: Consider the cost of the ETL tool and whether it fits within your budget.

Continued Learning



TRAILHEAD

[All things Data Migration Trailmix]

Sample Checklist

1	How many M-D fields you have?
	1.1 Is data populated on all the detail records?
	1.2 All Master records are available in Salesforce ?
2	How many lookup fields are present?
	2.1 Are there any lookup filters on fields?
	2.2 Is your data matching these filters?
3	How many Date columns are present?
	3.1 Are all date columns correctly formatted?
4	How many DateTime columns are present?
	4.1 Are all records formatted with correct Datetime format.
5	Do you have any columns which are null and needs to be uploded?
6	Are there any picklist columns?
	6.1 Are all these picklsit values present in Object?
	6.2 Are these Active?
	6.3 Is the Picklist restricted in Salesforce?
	6.4 Which record types should these new values be enabled for?
	6.5 Is this picklist using "Global Value" set?
	6.6 If Yes, which other objects are using the same set?
7	Have you verified data types from Source and Target system?
	7.1 Columns match the data type in both systems?
	7.2 If matching, do they have same length?
8	How many required fields you have?
	8.1 Is the field required on "Field Level"or on "Page Layout"?
	8.2 Was this field required in Source system as well?

Thank you!

Stay tuned in the Slack channel for the next topic and please feel free to suggest areas of interest.

Ways to get in touch...

Connect with me on LinkedIn
Email: svet@dc3me.com

Obvious People Slack Channel:
#salesforce-academy