Data Architecture and Management Designer Certification Study Guide

Prepared by: Tahsin Zulkarnine, tahsinz@gmail.com

Data Import

Salesforce offers two main methods for importing data.

Use the Data Import Wizard When:

- You need to load less than 50,000 records.
- The objects you need to import are supported by the wizard.
    - supports standard Salesforce objects, including accounts, contacts, leads, solutions, and person accounts and any custom object.
- You don't need the import process to be automated.

Use Data Loader When:

- You need to load 50,000 to five million records. If you need to load more than 5 million records, we recommend you work with a Salesforce partner or visit the AppExchange for a suitable partner product.
- You need to load into an object that is not supported by the Data Import Wizard.
- You want to schedule regular data loads, such as nightly imports.

Data Loader uses the SOAP API to process records. For faster processing, you can configure it to use the Bulk API instead.
The Bulk API is optimized to load a large number of records simultaneously. It is faster than theSOAP API due to parallel processing and fewer network round-trips.

This information can help you integrate your imported data into Salesforce.

- New Values for Picklists and Multi-Select Picklists—If you import a picklist value that doesn't match an existing picklist value:
    - For an unrestricted picklist, the Data Import Wizard uses the value that's in the import file.
    - For a restricted picklist, the Data Import Wizard uses the picklist's default value.
- Multi-Select Picklists—To import multiple values into a multi-select picklist, separate the values by a semicolon in your import file.
- Checkboxes—To import data into a checkbox field, use 1 for checked values and 0 for unchecked values.
- Default Values—For picklist, multi-select picklist, and checkbox fields, if you do not map the field in the import wizard, the default value for the field, if any, is automatically inserted into the new or updated record.
- Date/Time Fields—Ensure that the format of any date/time fields you are importing matches how they display in Salesforce per your locale setting.
- Formula Fields—Formula fields cannot accept imported data because they are read-only.
- Field Validation Rules—Salesforce runs validation rules on records before they are imported. Records that fail validation aren't imported. Consider deactivating the appropriate validation rules before running an import if they affect the records you are importing.

Data Export

Salesforce offers two main methods for exporting data.

- Data Export Wizard
    - . It allows you to export data manually once every 7 days (for weekly export) or 29 days (for monthly export).
    - You can also export data automatically at weekly or monthly intervals.

    - In Professional Edition and Developer Edition, you can generate backup files only every 29 days, or automatically at monthly intervals only.
- Data Loader

General Guidelines for Data Loads

- Always test your data loads in a sandbox organization first. Note that the processing times may be different in a production organization.
- Six guidelines to follow:
    a. Use Parallel Mode Whenever Possible
    b. Organize Batches to Minimize Lock Contention
    c. Be Aware of Operations that Increase Lock Contention
    d. Minimize Number of Fields, Workflow Rules, Triggers
    e. Optimize Batch Size
    f. Minimize Number of Batches in the Asynchronous Queue

Use Parallel Mode Whenever Possible

- enables faster loading of data
- can cause lock contention on records
- Don't process data in serial mode unless you know this would otherwise result in lock timeouts and you can't reorganize your batches to avoid the locks.

Organize Batches to Minimize Lock Contention

- If you organize AccountTeamMember records by AccountId so that all records referencing the same account are in a single batch, you minimize the risk of lock contention by multiple batches.
- If there are problems acquiring locks for more than 100 records in a batch, the Bulk API places the remainder of the batch back in the queue for later processing.
- If the Bulk API continues to encounter problems processing a batch, it's placed back in the queue and reprocessed up to 10 times before the batch is permanently marked as failed. E
    i. ven if the batch failed, some records could have completed successfully

Be Aware of Operations that Increase Lock Contention

- The following operations are likely to cause lock contention and necessitate using serial mode:
    i. Creating new users
    ii. Updating ownership for records with private sharing
    iii. Updating user roles
    iv. Updating territory hierarchies

Minimize Number of Fields, Workflow Rules, Triggers

- Foreign key, lookup relationship, and roll-up summary fields are more likely to increase processing time.

Optimize Batch Size

- The best course of action is to submit batches that process in less than 10 minutes.
- tart with 5000 records and adjust the batch size based on processing time.

i. If it takes more than five minutes to process a batch, it may be beneficial to reduce the batch size.
ii. If it takes a few seconds, the batch size should be increased.
iii. If you get a timeout error when processing a batch, split your batch into smaller batches, and try again.

Minimize Number of Batches in the Asynchronous Queue

- If more than 2,000 unprocessed requests from a single organization are in the queue, any additional requests from the same organization will be delayed while the queue handles requests from other organizations.
- Minimize the number of batches submitted at one time to ensure that your batches are not delayed in the queue.

Data Loading and Configuration Strategy

Data Loading Considerations

These setup options, which enable you to defer non-critical processes and speed LDV loading:

1. Org wide defaults
    - Making org-wide defaults to private initiates sharing calculate.
    - If you load with a Public Read/Write sharing model, you can defer this processing until after cutover.
2. Complex object relationship:
    - The more lookups you have defined on an object, the more checks the system has to perform during data loading.
3. Sharing rules
    - Any sharing rule either owner based or criteria based will initiate the sharing calculation.
4. Workflow rules, validation rules, and triggers.
    - Cleaner data but they can also slow down processing if they're enabled during massive data loads.

Bulk API vs. SOAP API Data Loading

SOAP API

- optimized for real-time client applications that update a few records at a time.
- becomes unwieldy and time-consuming with large data sets.

Requires developers and administrators to implement complex processes to upload data in bite-sized chunks, monitor results, and retry failed records.

Bulk API

- based on REST principles and was developed specifically to simplify and optimize the process of loading or deleting large data sets.
- Features:
    - super-fast processing speeds,
    - reduced client-side programmatic language,
    - easy-to-monitor job status,
    - automatic retry of failed records,
    - support for parallel processing,
    - minimal roundout trips to Force.com,
    - minimal API calls,
    - limited dropped connections, and
    - easy-to-tune batch size

How Bulk API Works

- upload records using Bulk API, those records are streamed to Force.com to create a new job.
- it's stored in temporary storage and then sliced up into user-defined batches (10,000 records max)

- Even as your data is still being sent to the server, the Force.com platform submits the batches for processing.
- If a job times out, the Bulk API automatically puts it back in the queue and re-tries it for you.
- Each batch is processed independently, and once the batch finishes (successful or not), the job is updated with the results.

Increase Speed by Suspending Events

There are three key phases to validate accurate data and the right relationships established while turning off events (validation, workflow and triggers):

1. Analyzing and Preparing Data
    - Find use cases which can meet requirements by data cleaning operations before and after
    - Validate them using query or lookup, roll-up summary fields.
2. Disabling Events for Loading
    - Custom setting to enable/disable trigger operation.
3. Post-Processing
    - When you've finished loading your data, it's time to complete the data enrichment and configuration tasks you've deferred until this point:

        i. Add lookup relationships between objects, roll-up summary fields to parent records, and other data relationships between records using Batch Apex or Bulk API.
        ii. Enhance records in Salesforce with foreign keys or other data to facilitate integration with your other systems using Batch Apex or Bulk API.
        iii. Reset the fields on the custom settings you created for triggers, so they'll fire appropriately on record creation and updates.
        iv. Turn validation, workflow, and assignment rules back on so they'll trigger the appropriate actions as users enter and edit records.

Lightning Platform API Basics

Types of API:

- REST
  - ease of integration and development
  - an excellent choice of technology for use with mobile applications and web projects
- SOAP
- BULK
- Streaming

Note the four data APIs that we talked about already. We'll be diving into each of them next.

API Name	Protocol	Data Format	Communication
REST API	REST	JSON, XML	Synchronous
SOAP API	SOAP (WSDL)	XML	Synchronous
Chatter REST API	REST	JSON, XML	Synchronous (photos are processed asynchronously)
User Interface API	REST	JSON	Synchronous
Analytics REST API	REST	JSON, XML	Synchronous
Bulk API	REST	CSV, JSON, XML	Asynchronous
Metadata API	SOAP (WSDL)	XML	Asynchronous
Streaming API	Bayeux	JSON	Asynchronous (stream of data)
Apex REST API	REST	JSON, XML, Custom	Synchronous
Apex SOAP API	SOAP (WSDL)	XML	Synchronous
Tooling API	REST or SOAP (WSDL)	JSON, XML, Custom	Synchronous

API Access and Authentication

- Make sure that you have the "API Enabled" permission
- All API calls, except for the SOAP API login() call, require authentication
- You can either use one of the supported OAuth flows or authenticate with a session ID retrieved from the SOAP API login() call.

API Limit

There are two types of API limits:

1. Concurrent limits cap the number of long-running calls (20 seconds or longer) that are running at one time.
    - For a sandbox org, it's 25 long-running calls.
2. Total limits cap the number of calls made within a rolling 24-hour period.
    - Total limits vary by org edition, license type, and expansion packs that you purchase.

When to Use What API

- To build UI for creating, reading, updating, and deleting records, including building UI for list views, actions, and dependent picklists, use **User Interface API.**
- Use **Chatter REST API** to display Chatter feeds, users, groups, and followers, especially in mobile applications.
- You can access Analytics assets—such as datasets, lenses, and dashboards—programmatically using the **Analytics REST API.**
- Use **Metadata API** to retrieve, deploy, create, update, or delete customizations for your org.
- Use **Streaming API** to receive near-real-time streams of data that are based on changes in Salesforce records or custom payloads.
    - Clients can also subscribe to some types of events with Apex triggers or declaratively with Process Builder and Flow Builder.
    - Type of streaming : PushTopic Event, Change Data Capture Event, Platform Event, Generic Event
- Use **Apex REST API** when you want to expose your Apex classes and methods so that external applications can access your code through REST architecture.
- Use **Apex SOAP API** when you want to expose Apex methods as SOAP web service APIs so that external applications can access your code through SOAP.
- Use **Tooling API** to integrate Salesforce metadata with other systems
    - use Tooling API to manage and deploy working copies of Apex classes and triggers and Visualforce pages and components

Use REST API

- A REST request consists of four components: a resource URI, an HTTP method, request headers, and a request body.
    - standard HTTP methods (HEAD, GET, POST, PATCH, DELETE)
- Core Endpoints:
    - Describe - /services/data/vXX.0/sobjects/account/describe
    - Create - /services/data/vXX.0/sobjects/account
    - Query - /services/data/vXX.0/query/?q=SELECT+Name+From+Account+WHERE+ShippingCity='San+Francisco'

Use SOAP API

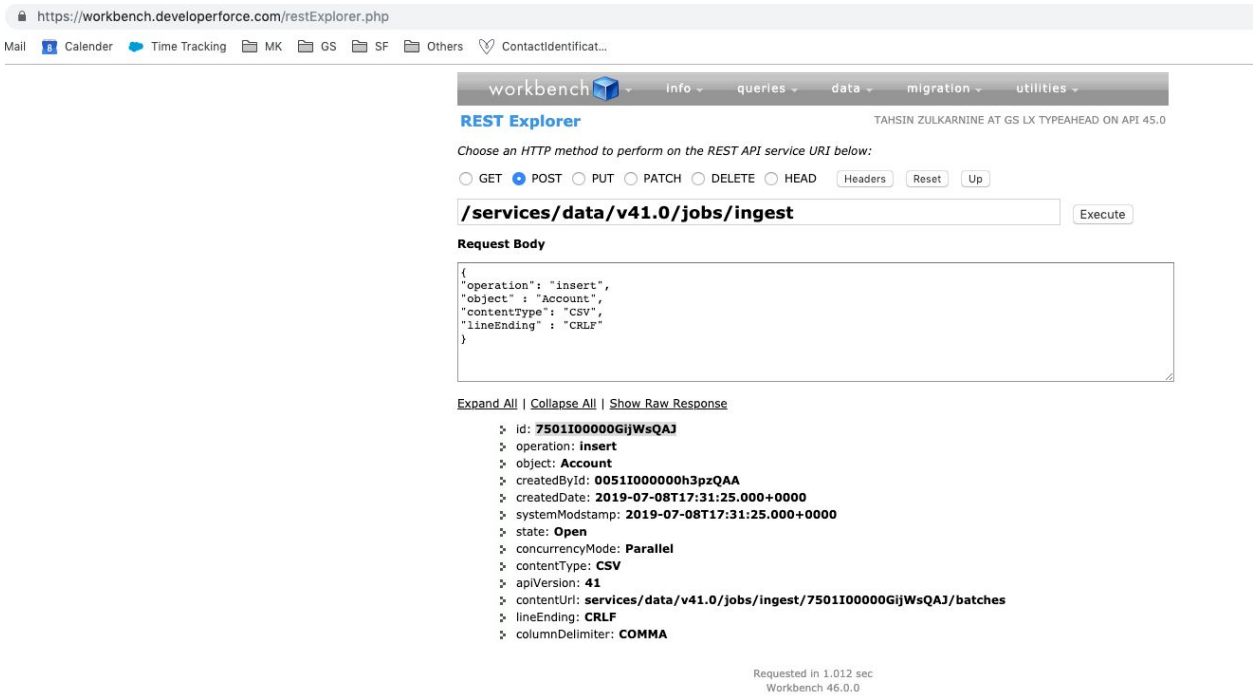
- Web Services Description Language (WSDL) file is basically your map to understanding how to use the API. It contains the bindings, protocols, and objects to make API calls.

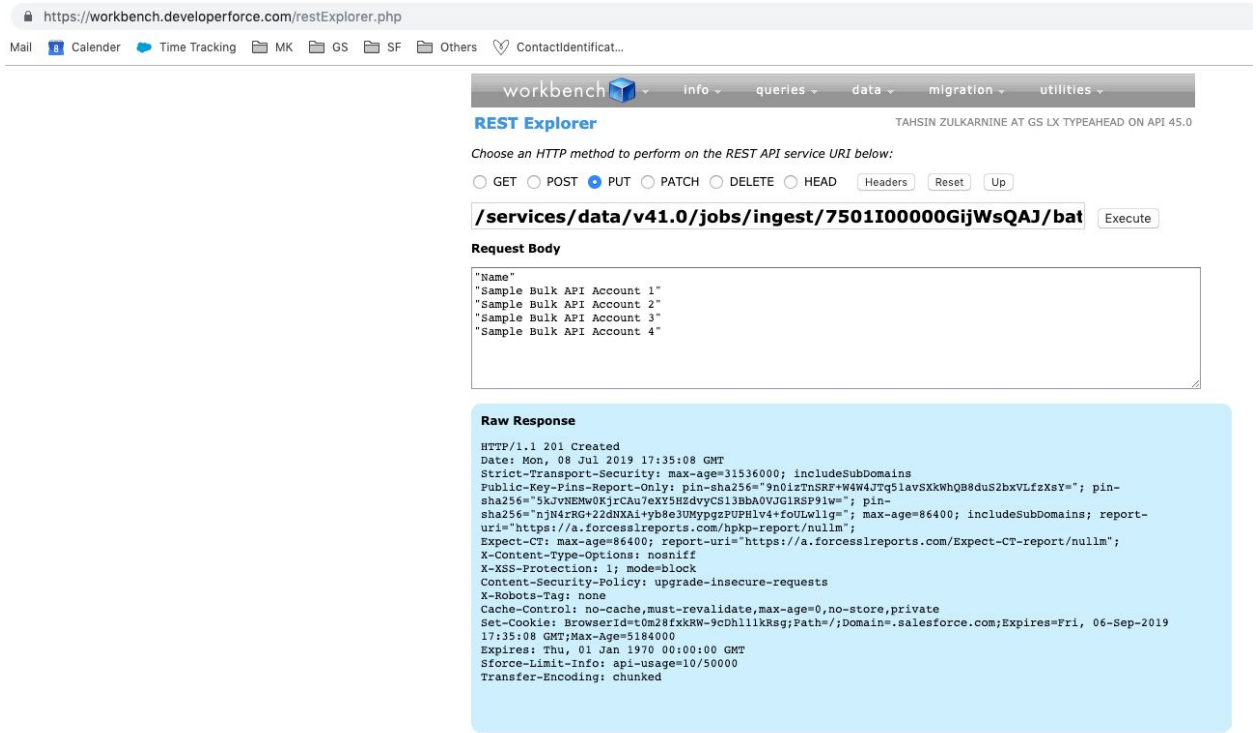
Enterprise and Partner WSDLs

- The enterprise WSDL is optimized for a single Salesforce org.
    - It's strongly typed, and it reflects your org's specific configuration, meaning that two enterprise WSDL files generated from two different orgs contain different information.
- The partner WSDL is optimized for use with many Salesforce orgs.
    - It's loosely typed, and it doesn't change based on an org's specific configuration.

Use Bulk API

- Bulk API is based on REST principles and is optimized for working with large sets of data.





Use Streaming API

PushTopic

- A PushTopic is an sObject that contains the criteria of events you want to listen to, such as data changes for a particular object.
- PushTopic queries support all custom objects and some of the popular standard objects, such as Account, Contact, and Opportunity.

```
PushTopic pushTopic = new PushTopic();
pushTopic.Name = 'AccountUpdates';
pushTopic.Query = 'SELECT Id, Name, Phone FROM Account WHERE BillingCity=\'San Francisco\'';
pushTopic.ApiVersion = 37.0;
insert pushTopic;

pushTopic.NotifyForOperationCreate = true;
pushTopic.NotifyForOperationUpdate = true;
pushTopic.NotifyForOperationUndelete = true;
pushTopic.NotifyForOperationDelete = true;
```

Platform Events

- Define platform events, i.e Order_Event__e
- The channel name is based on the event's API name and the format is /event/Event_Name. For example, /event/Order_Event__e

Publishing Platform Events

You can publish platform events using these declarative or programmatic tools on the Lightning Platform.

1. Process Builder using the Create a Record action
2. Flow using a Create Records element
3. Apex EventBus.publish() method
4. REST API sobjects resource
5. SOAP API create() call

Custom Notifications with Generic Streaming

To use generic streaming, you need:

- A streaming channel that defines the channel
- One or more clients subscribed to the channel
- The Streaming Channel Push resource to monitor and invoke events on the channel
- The format of the channel name for generic streaming is /u/ChannelName

```
StreamingChannel ch = new StreamingChannel();
ch.Name = '/u/Broadcast';
insert ch;
```

Alternatively, you can opt to have Salesforce create the streaming channel dynamically for you if it doesn't exist.

- To enable dynamic streaming channels in your org, from Setup, enter User Interface in the Quick Find box, then select User Interface.
- On the User Interface page, select the Enable Dynamic Streaming Channel Creation option.

To generate events, make a POST request to the following REST resource. Replace XX.0 with the API version and Streaming Channel ID with the ID of your channel.

/services/data/vXX.0/sobjects/StreamingChannel/Streaming Channel ID/push

Retrieve Past Notifications Using Durable Streaming

- The events are stored for 24 hours, and you can retrieve them at any time during that window.
- Each event message is assigned an opaque ID contained in the ReplayId field.
    - The ReplayId field value, which is populated by the system when the event is delivered to subscribers, refers to the position of the event in the event stream.

Data Model Design and Issues

Personal Account

Contacts to Multiple Account

Territory Management

- From sharing and visibility document.

DESIGNING RECORD ACCESS FOR ENTERPRISE SCALE

- From sharing and visibility document

Select State and Country from Picklists

- The states and countries in the picklists are based on ISO-3166 standard values, making them compatible with other applications.
- State and country picklists are available in the shipping, billing, mailing, and "other" address fields in the account, campaign members, contact, contract, lead, order, person accounts, quotes, and service contracts standard objects
- State and country picklists include 239 countries by default
    - They also include the states and provinces of the United States, Canada, Australia, Brazil, China, India, Ireland, Italy, and Mexico.
- State and country picklists that contain more than 1,000 states or countries can cause degraded performance
- State and country picklists do not work with:
    - Salesforce to Salesforce
    - Connect Offline
    - Change sets
- Picklist labels, not code values, are displayed in reports on state and country fields.

Implementing State and Country Picklists

Here's how to transition from text-based state and country fields to state and country picklists.

1. Configure the state and country values you want to use in your org.
2. Scan your org's data and customizations.
    a. Convert data and update customizations, such as list views, reports, and workflow rules, so that they continue to work with the new field type.
3. Convert existing data.
    a. For example, map U.S., USA, and United States to US.
4. Turn on the picklists for your users.
5. Optionally, rescan and fix customizations or records that have been created or edited since your first scan

Force.com Object Bloat

- ending up with objects with large numbers of fields, “just in case”.
- start getting complaints about slow custom-built pages, list views, and reports.

Skew

Data skew happens when more than 10,000 child records are associated with the same parent record within an org.

Record Locking

Here’s another scenario: You’re updating a large number of contacts under the same account in multiple threads. For each update, the system locks both the contact being changed and its parent account to maintain integrity in the database. Even though each lock is held for a very short time, because all the updates are trying to lock the same account, there’s a high risk an update will fail because a previous one is still holding the lock on the account.

Sharing issues

like changing the owner of an account, you may need to examine every one of the account’s child records and adjust their sharing, as well. That may include recalculating the role hierarchy and sharing rules. And if we’re talking about hundreds of thousands of child records, that can eat up tons of time.

Ownership Skew

When a large number of records with the same object type are owned by a single user, this imbalance causes ownership skew.

- This can cause performance issues due to sharing calculations required to manage the visibility of those records.
- In some cases, an ownership skew simply can’t be avoided. In these cases, it’s best to ensure the skewed owner doesn’t have a role.
- That way, you take the user and their records away from the role hierarchy and its associated sharing rules.

Lookup Skew

Lookup skew happens when a very large number of records are associated with a single record in the lookup object (the object you’re searching against).

Lookup fields in Salesforce are essentially foreign key relationships between objects. Every time a record is inserted or updated, Salesforce must lock the target records that are selected for each lookup field. This ensures that when the data is committed to the database, its integrity is maintained

when you add custom code and LDV simultaneously in an automated process, you might encounter lock exceptions that cause failures when you try to insert or update records.

it’s best to search based on patterns that will cause problems. You should evaluate objects with a large number of records and heavy concurrent insert and update activity.

Using External Objects

Another strategy for LDV is using external objects—which means there’s no need to bring data into Salesforce.

. External objects are best used when you have a large amount of data that you can’t or don’t want to store in your Salesforce org, and you only need to use a small amount of that data at any one time.

Relationship	Allowed Child Objects	Allowed Parent Objects	Parent Field for Matching Records
Lookup	Standard Custom External	Standard Custom	The 18-character Salesforce record ID
External Lookup	Standard Custom External	External	The External ID standard field
Indirect Lookup	External	Standard Custom	You select a custom field with the External ID and unique attributes

External lookup relationship

Use an external lookup relationship when the parent is an external object. An external lookup relationship links a child standard, custom, or external object to a parent external object. The values of the standard External ID field on the parent external object are matched against the values of the external lookup relationship field. For a child external object, the values of the external lookup relationship field come from the specified External Column Name.

Indirect lookup relationship

Use an indirect lookup relationship when the external data doesn't include Salesforce record IDs. An indirect lookup relationship links a child external object to a parent standard or custom object. When you create an indirect lookup relationship field on an external object, you specify the parent object field and the child object field to match against each other, selecting a custom unique external ID field on the parent object to match against the child's indirect lookup relationship field, whose values are determined by the specified External Column Name.

Query and Search

Indexing

- For data to be searched, it must first be indexed.
- Force.com automatically indexes most text fields so your users can build cross-object searches and quickly find records that contain strings of interest.
- Indexed searches are performed by first searching the indexes for appropriate records, then narrowing down the results based on access permissions, search limits, and other filters.
- This creates a result set, which typically contains the most relevant results.
- After the result set reaches a predetermined size, the remaining records are discarded.
- The result set is then used to query the records from the database to retrieve the fields that a user sees. And when large volumes of data are added or changed, this whole process could take a long time.

Cheat Sheet: List of Indexed Fields

Primary keys
- Id
- Name
- OwnerId

Foreign keys
- Lookup
- Master-detail
- CreatedById
- LastModifiedById

Audit dates
- CreatedDate
- LastActivityDate
- SystemModstamp

Custom fields
- Unique
- External ID

The exceptions are:
- non-deterministic formula fields,
- multi-select picklists,
- currency fields in a multi-currency org,
- long text area and rich text area fields, and
- binary fields (type blob, file or encrypted text).

Index Tables

- the platform creates an index table that contains a copy of the data, along with information about the data type
- the index tables do not include records that are null (records with empty values)

Two-Column Custom Indexes

- When a combination of two fields is a common filter in the query string, two-column indexes typically help you sort and display records.
- Two-column indexes are subject to the same restrictions as single-column indexes, with one exception.
    - Two-column indexes can have nulls in the second column, whereas single-column indexes can't unless Salesforce Customer Support explicitly enabled the option to include null

Query Building

query building is a key for designing an org that can handle LDV. It's important to design selective list views, reports, and SOQL queries, and to understand query optimization.

SOSL

- SOSL is Force.com's full-text search language. SOSL can tokenize multiple terms within a field, and can build a search index off of this.
- for each Apex transaction, the governor limit for SOSL queries is 2,000; for SOQL queries it's 50,000. So if you need to retrieve more than 2,000 records, SOQL is the better choice.

Force.com query optimizer

The Force.com query optimizer maintains a table of statistics about the distribution of data in each index.
It uses this table to perform pre-queries to determine whether using the index can speed up the query.
It works on the queries that are automatically generated to handle reports, list views, and both SOQL queries and the other queries that piggyback on them.

The query optimizer uses similar considerations to determine whether to use indexes when the WHERE clause contains AND, OR, or LIKE.

- For AND, the query optimizer uses the indexes unless one of them returns more than 20% of the object's records or 666,666 total records.
- For OR, the query optimizer uses the indexes unless they all return more than 10% of the object's records or 333,333 total records.
    - Note: All fields in the OR clause must be indexed for any index to be used.
- For LIKE, the query optimizer does not use its internal statistics table. Instead, it samples up to 100,000 records of actual data to decide whether to use the custom index.

- If the filter doesn't have an index, it won't be considered for optimization.
- If the filter has an index, determine how many records it would return:
    - For a standard index, the threshold is 30 percent of the first million targeted records and 15 percent of all records after that first million.

        - In addition, the selectivity threshold for a standard index maxes out at 1 million total targeted records, which you could reach only if you had more than 5.6 million total records.
    - For a custom index, the selectivity threshold is 10 percent of the first million targeted records and 5 percent all records after that first million.
        - In addition, the selectivity threshold for a custom index maxes out at 333,333 targeted records, which you could reach only if you had more than 5.6 million records.
- If the filter exceeds the threshold,it won't be considered for optimization.
- If the filter doesn't exceed the threshold, this filter IS selective, and the query optimizer will consider it for optimization.

Query Plan Tool

- Set 'Enable Query Plan' to TRUE in Developer Console

Reasons to use the Query Plan Tool

1. It will provide you with insight on the different plans and should you have some of the filters indexed, provide the cost of using the index compared to a full table scan.

The Query Plan Tool will show a list of available plans that our Query Optimizer can use for the query provided and will be arranged by cost ascending.

Composition

Each Plan will contain information

1. Cardinality
    a. The estimated number of records that the leading operation type would return.
2. Fields
    a. The indexed field(s) used by the Query Optimizer. I
    b. f the leading operation type is Index, the fields value is Index. Otherwise, the fields value is null.
3. Leading Operation Type
    a. The primary operation type that Salesforce will use to optimize the query.
        i. Index - The query will use an index on the queried object.
        ii. Sharing - The query will use an index based on the sharing rules associated with the user who is executing the query. If there are sharing rules that limit which records that user can access, Salesforce can use those rules to optimize the query.
        iii. TableScan - The query will scan all records for the queried object.
        iv. Other - The query will use optimizations internal to Salesforce.

4. Cost
    a. The cost of the query compared to the Force.com Query Optimizer's selectivity threshold. Values above 1 mean that the query won't be selective.
5. sObject Cardinality
    a. The approximate record count for the queried object.
6. sObject Type
    a. The name of the queried

Unsupported operations

- Custom index will never be used when comparisons are being done with an operator like "NOT EQUAL TO"
- Custom index will never be used when comparisons are being done with a null value like "Name = ''"
- Leading '%' wildcards are inefficient operators that also make filter conditions non-selective
- When using an OR comparison, all filters must be indexed and under the 10% threshold.
    - If you have a non-indexed field or one is above 10%, the plan will not be displayed.

Batch Apex

- You can query and process up to 50 million records using Batch Apex.
- Batch Apex doesn't work in all use cases
    - for example, if you have a synchronous use like as a Visualforce page that needs to query more than 50,000 records

Bulk Queries

- A bulk query can retrieve up to 15 GB of data, divided into fifteen 1 GB files.
- Bulk API query supports both query and queryAll operations.
    - The queryAll operation returns records that have been deleted because of a merge or delete. T
    - he queryAll operation also returns information about archived Task and Event records.
- When adding a batch to a bulk query job, the Content-Type in the header for the request must be either text/csv, application/xml, or application/json, depending on the content type specified when the job was created.
    - The actual SOQL statement supplied for the batch is in plain text format.

How Bulk Queries Are Processed

- If the query doesn’t execute within the standard **two-minute timeo**ut limit, the job fails and a QUERY_TIMEOUT error is returned.
    - If this happens, rewrite a simpler query and resubmit the batch.
- If the query succeeds, Salesforce attempts to retrieve the results.
    - If the results exceed the **1 GB file size** limit or t**ake longer than 10 minutes to retriev**e, the completed results are cached and another attempt is made. **After 15 attempts**, the job fails and the error message **Retried more than fifteen times** is returned.
        - If this happens, consider using the **PK Chunking** header to split the query results into smaller chunks. (More on PK chunking in a subsequent unit.)
- If the attempts succeed, the results are returned and stored for **seven days**.

Using Skinny Tables

- A skinny table is a custom table in the Force.com platform that contains a subset of fields from a standard or custom base Salesforce object.
- By having narrower rows and less data to scan than the base Salesforce object, skinny tables allow Force.com to return more rows per database fetch, increasing throughput when reading from a large object
- skinny tables don’t include soft-deleted rows, which often reduces the table volume.
- Custom indexes on the base table are also replicated, and they usually perform better because of the reduced table joins that happen in the underlying database queries

- To enable skinny tables, contact Salesforce Customer Support.
- The Force.com platform automatically synchronizes the rows between the base object and the skinny table, so the data is always kept current.
- The Force.com platform determines at query runtime when it would make sense to use skinny tables, so you don't have to modify your reports or develop any Apex code or API calls.

Consideration for skinny tables

1. Skinny tables are skinny. To ensure optimal performance, they contain only the minimum set of fields required to fulfill specific business use cases. I
2. Skinny tables don't get copied over to sandbox organizations.
3. Skinny tables are custom tables in the underlying Force.com database. They don't have the dynamic metadata flexibility you find in the base object.

Data Extraction, Archival and Backup

Soft vs. Hard Deletion

Recycle bin

- Salesforce uses a Recycle Bin for data that users delete.
- The data stays in the Recycle Bin for 15 days, or until the Recycle Bin grows to a specific size.
- emptied using the UI, the API, or Apex.

Soft Delete

- Instead of removing the data, it's flagged as deleted and visible through the Recycle Bin.
- still affects database performance because it's still living in the org, and deleted records have to be excluded from any queries.

Hard Delete

- bypass the Recycle Bin and immediately become available for deletion.
- Note that the hard delete option is disabled by default and must be enabled by an administrator.
- sing Bulk API's hard delete function is a recommended strategy for deleting large data volumes to free up space sooner and keep extraneous material from affecting performance.

Chunking Data

- When extracting data with Bulk API, queries are split into 100,000 record chunks by default which can be made smaller.
- Larger chunk sizes use up fewer Bulk API batches but may not perform as well.
- At extremely high volumes—hundreds of millions of records—defining these chunks by filtering on field values may not be practical
    - The number of rows that are returned may be higher than the selectivity threshold of Salesforce's query optimizer.
    - The result could be a full table scan and slow performance, or even failure.
    - Then you need to employ a different strategy.

Using PK Chunking

- PK stands for Primary Key—the object's record ID—which is always indexed.
- PK chunking splits bulk queries on very large tables into chunks based on the record IDs of the queried records.
- Enable PK chunking when querying tables with more than 10 million records or when a bulk query consistently times out.
- It's supported for Account, Campaign, CampaignMember, Case, Contact, Lead, LoginHistory, Opportunity, Task, and User, as well as all custom objects.
- To enable the feature, specify the header Sforce-Enable-PKChunking on the job request for your Bulk API query.
    - Sforce-Enable-PKChunking: chunkSize=50000.
- Each chunk is processed as a separate batch that counts toward your daily batch limit, and its results must be downloaded separately.

- Status
    - When a query is successfully chunked, the original batch's status shows as NOT_PROCESSED.
    - If the chunking fails, the original batch's status shows as FAILED, but any chunked batches that were successfully queued during the chunking attempt are processed as normal

Truncation

- try truncating those custom objects if you want custom objects to be deleted immediately
- Truncating a custom object erases all records currently sitting in the custom object's Recycle Bin;
- the custom object's history; and related events, tasks, notes, and attachments for each deleted record.
- Setup -> User Interface -> Custom Object -> Truncate

You cant truncate

- that are referenced by another object through a lookup field, or that are on the master side of a master-detail relationship,
- are referenced in a reporting snapshot, have a custom index or an external ID, or
- have activated skinny tables.
- when your org has reached its limit on allowed custom objects.

Optimizing Data Extraction

- We recommend both architecting data extractions to retrieve only the delta (i.e., updated, newly inserted, and deleted) rows and updating those rows in the external system.
- Chunking the Data into Smaller Sets
    - can be easily implemented by following these steps:
        - Create or use an existing auto-number field.
        - Create a formula field that converts the auto-number field text value into a numeric value—you cannot use an index with comparison operators such as "<=" (less than or equal to) or ">" (greater than) for the text-based auto-number field.
        - Place a custom index on the formula field by contacting salesforce.com Customer Support.
        - Run pre-queries to determine the boundaries.
        - Add a range filter to your extraction (i.e., a WHERE clause in your SOQL) to limit the number of targeted rows so that they are below the selectivity threshold.
- Increasing Data Throughput
    - Run your requests in parallel.
    - Have an administrator with access to all the data perform the extraction.
        - This practice can help you minimize sharing calculation overhead.
    - Always use the Bulk API or batch Apex when dealing with large data volumes.
    - If you need an extra boost to increase throughput, consider using a skinny table

Backup

Backup Overview

Why would you backup Salesforce data (including metadata)?

- Recover from data corruption (unintended user error or malicious activity)

- Prepare for a data migration rollback
- Archive data to reduce volumes
- Replicate data to a data warehouse/BI
- Take snapshots of development versions

What to backup and how?

- data and metadata.
- our APIs to backup data and metadata:
    - REST API
    - SOAP API
    - Bulk API
    - Metadata API

Backup Type	Comments	Pros	Cons
Full	Contains all data	Contains all information that may be required	Can represent a large volume, takes more time to retrieve a subset of data and be reactive to handle a data incident
Incremental	Backup differences since last full backup (daily incremental backup, weekly, monthly, etc.), the data replication API is particularly adapted to this type of backup	Efficient for retrieving a change that took place on a specific date, smaller files that are easier to handle	May lack related information and take longer to rebuild a complete picture (merge full backup and last incremental backups)
Partial	Backup a subset of data (for example, closed cases only)	Efficient for retrieving a record from a subset of data, ideal approach for archiving purposes (i.e. records older than 5 years)	May lack related information

backup plan might look something like:

- Weekly full backup
- Daily incremental backup
- Monthly partial backup (Cases closed more than 3 years ago) for archive and delete

API Options

Requirements	Recommended API	Explanation

You need to preserve governor limits regarding the number of API calls	Bulk	Bulk API does not consume API calls but consumes Bulk API calls, which are less restrictive
You need to preserve the governor limit regarding "number of batches per rolling 24 hour period"	REST or SOAP	REST or SOAP are not subject to Bulk-specific governor limits, however they have their own limits
You need to backup an object containing a large volume of records (i.e. more than 2M records) or do a backup that raises performance challenges	Bulk	Bulk API will generally be faster than the other APIs, but the REST or SOAP APIs might sometimes get better results depending on several factors: query batch size, current asynchronous load on the instance, degree of parallelization
You need to backup an object that is not yet supported by the Bulk API (i.e. CaseStatus, OpportunityStage, AcceptedEventRelation, etc)	REST or SOAP	Bulk API does not yet support all queryable objects in Spring `15 release
You need to backup an object that contains a lot of XML-like information (example: EmailMessage)	REST or Bulk	While this is not directly caused by the Salesforce SOAP API, we have seen some XML parsers encountering difficulties when processing the HTTP response (mix of XML-based protocol and XML data)
You need to backup metadata	Metadata	The Metadata API is by far the most exhaustive API to retrieve metadata, however a large part of the metadata is also available in the REST, SOAP and Tooling APIs
You need to back up files (Attachment, ContentVersion, Document, FeedItem, StaticResource, etc.)	REST or SOAP	The Bulk API does not yet support Base64 fields in Spring `15 release

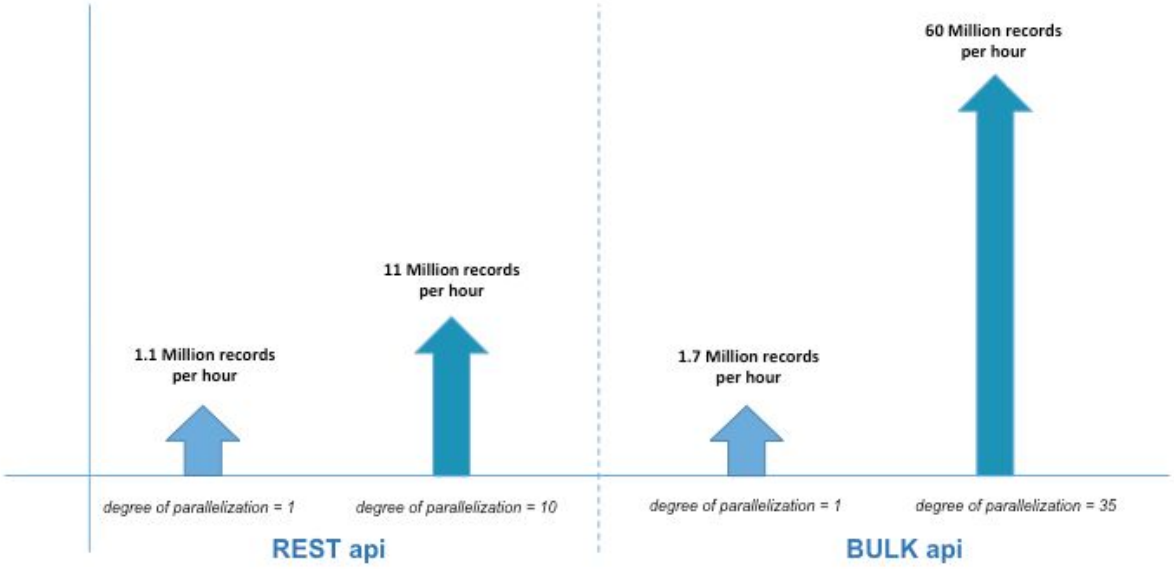
Performance

Depends on:

- Number of records (lines)
- Number of fields (columns)
- Type of fields (i.e. rich text fields are slower to backup than checkbox fields)
- Salesforce API selected
- Your network capacity

- Degree of parallelization
- Another important part of backup performance is the capacity to retry an operation to maximize its probability of success.

Figure 2 : average API speeds based on the backups of 12 Billion records; source benchmark odaseva.com 2015



Vertical optimization	• Backup time is, among other parameters, proportional to the number of records you are retrieving. • Splitting vertically your query into several queries to reduce the number of records included per query can bypass several incidents like timeouts or other limits • Using PK Chunking is one very efficient way of splitting a query vertically. • Partial backup is also a type of vertical optimization.
Horizontal optimization	• Backup time is, among other parameters, proportional to the number and types of columns you are retrieving. • Text areas (and other large fields) are fields that can really slow down a backup to the point of timeout. • Splitting horizontally your query into several queries, to reduce the number of fields included per query, will avoid several issues like timeouts or query size limit (Removing fields from the query (i.e. calculated fields) is also a type of horizontal optimization.

Restore

Why?

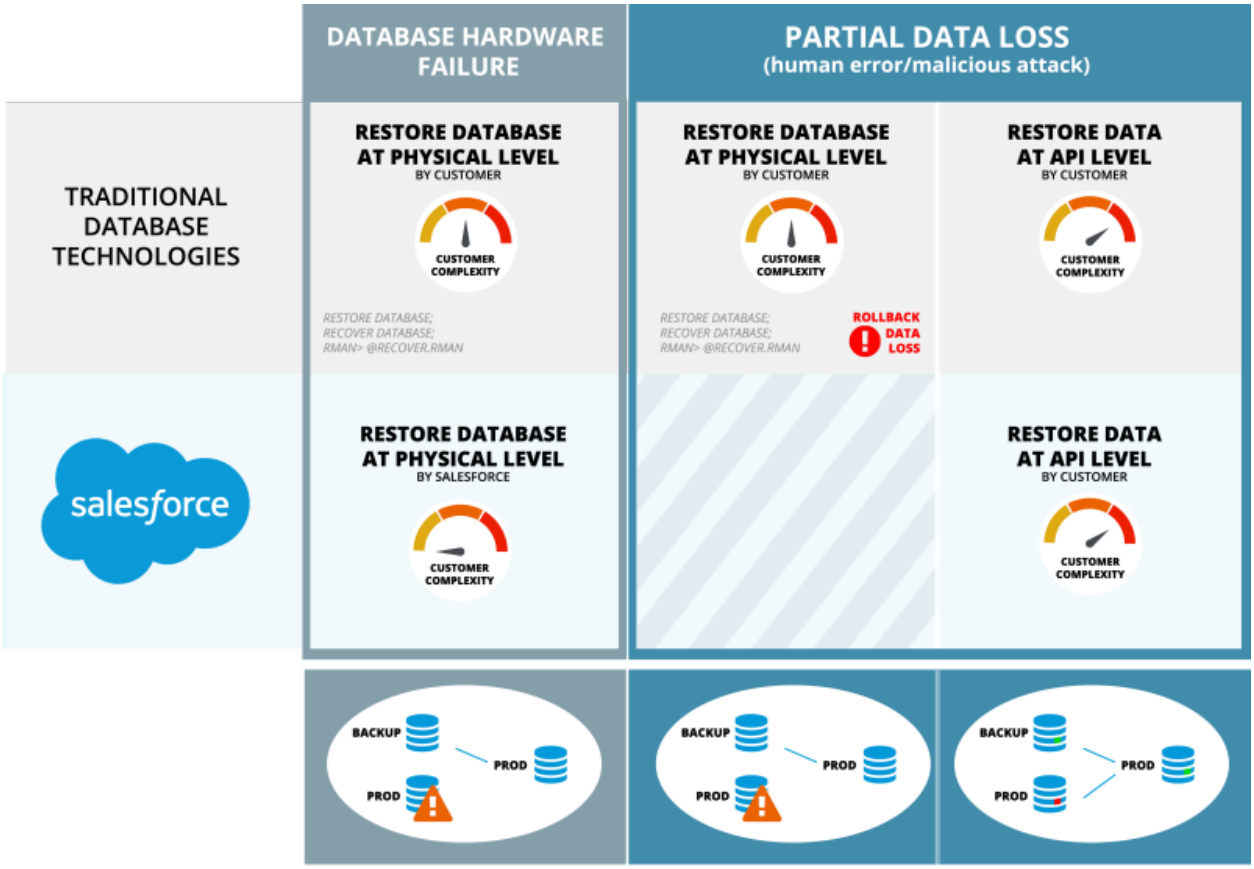


Fig: Data Recovery: Salesforce v traditional database - source odaseva.com

Additiona Security and disaster recovery plan (DRP):

- Security
  - Recover a subset of data that has been corrupted or deleted in production
- Release Management
  - Initialize a training environment with a specific set of data (business unit specific, country specific, other)
  - Automate transport of data/metadata across Salesforce environments
  - Initialize a sandbox with a set of master/technical data after refresh
  - Prepare or verify a training environment for a go live rollback strategy (data and metadata)
- Merger and Acquisition
  - Copy an org or split an org, often needed as part of a merger and acquisition process

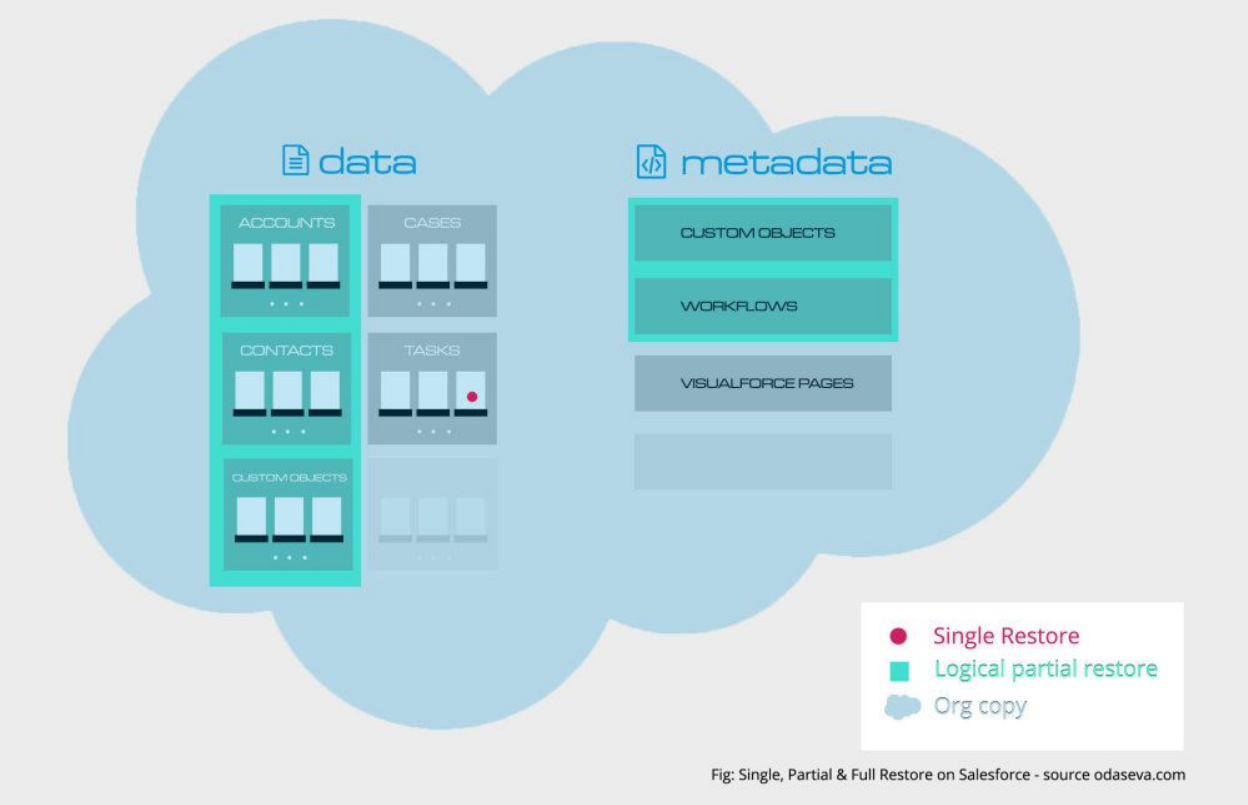


Fig: Single, Partial & Full Restore on Salesforce - source odaseva.com

Single restore

Use case: Recover a record (example: an event) that has been deleted or corrupted

Pros and cons: Simple and best RTO but might lose data

Duration from incident detection to incident closure: a few minutes

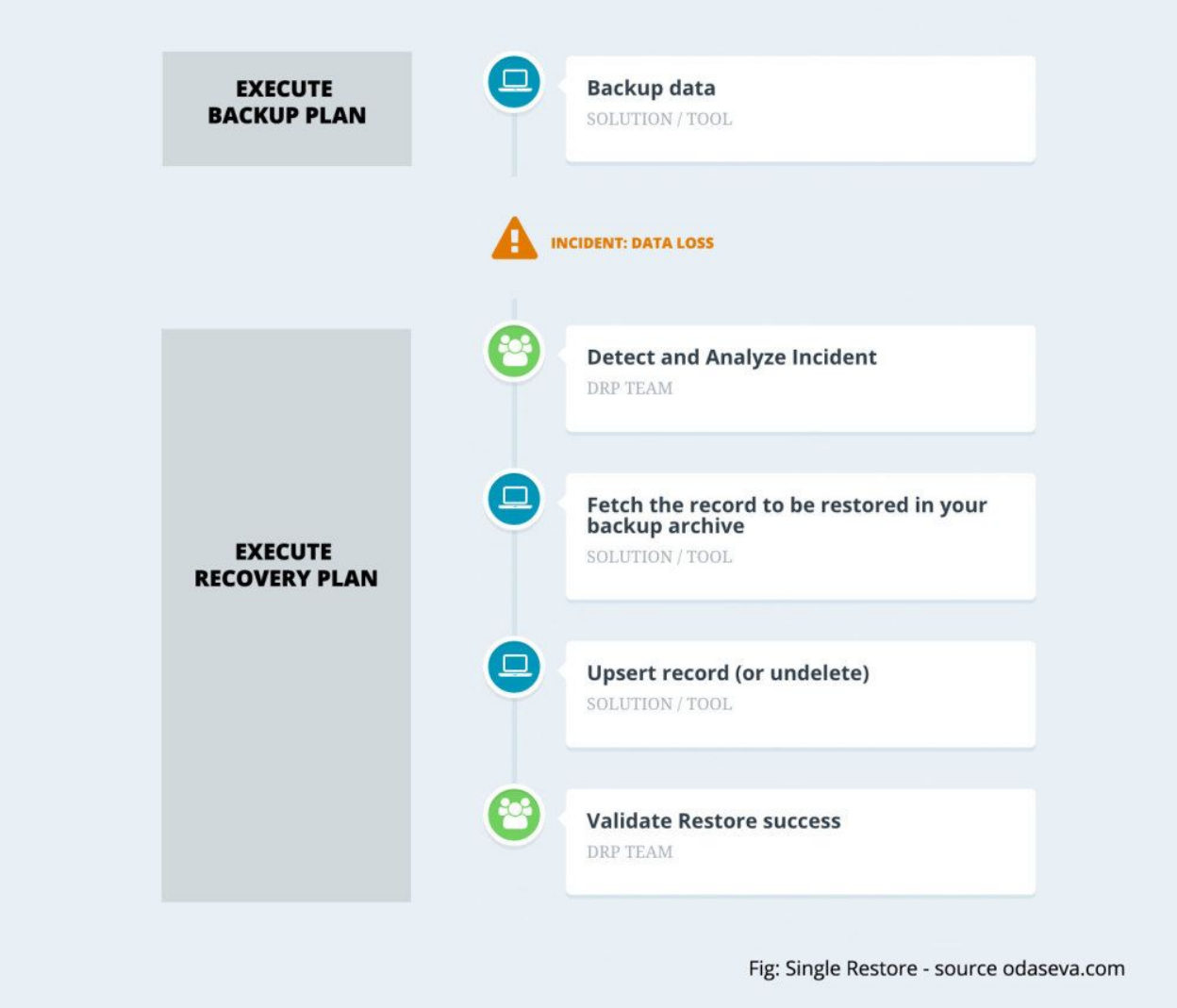


Fig: Single Restore - source odaseva.com

partial restore

Use case: Recover a record (example: an Account) that has been deleted alongside its child records (Contacts, Opportunities, etc)

Pros and cons: Does not lose data that was created post backup. Best RPO but requires preparation and the right tools.

Duration from incident detection to incident closure: a few hours

Fig: Logical Partial Restore - source odaseva.com

Org copy

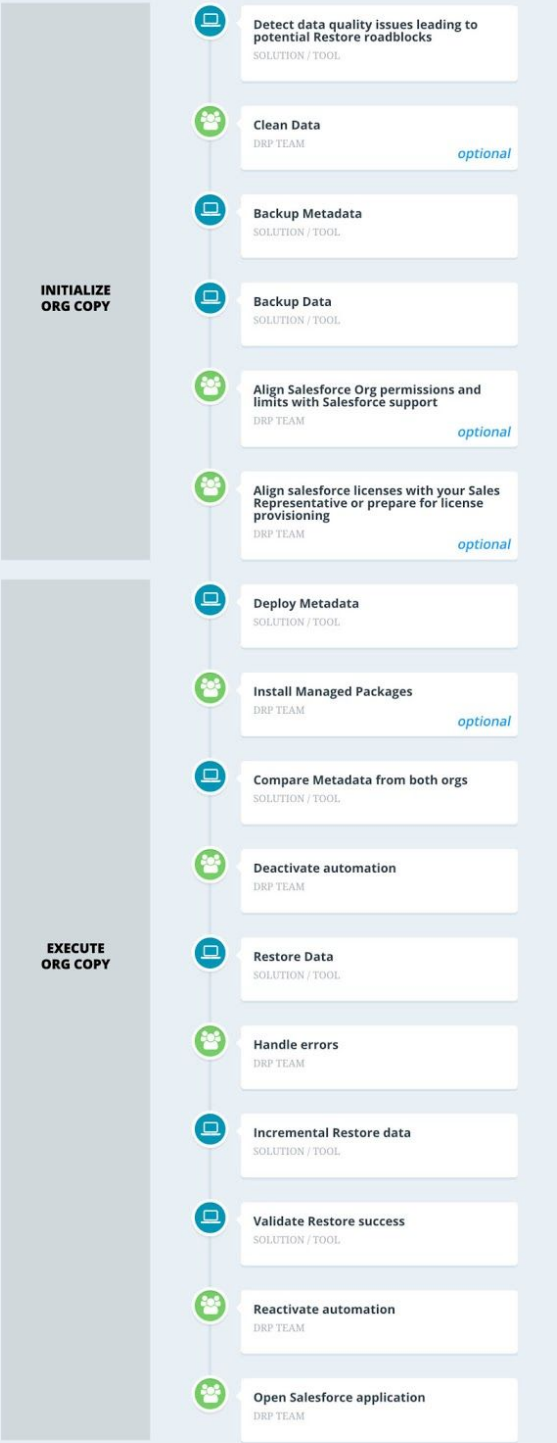
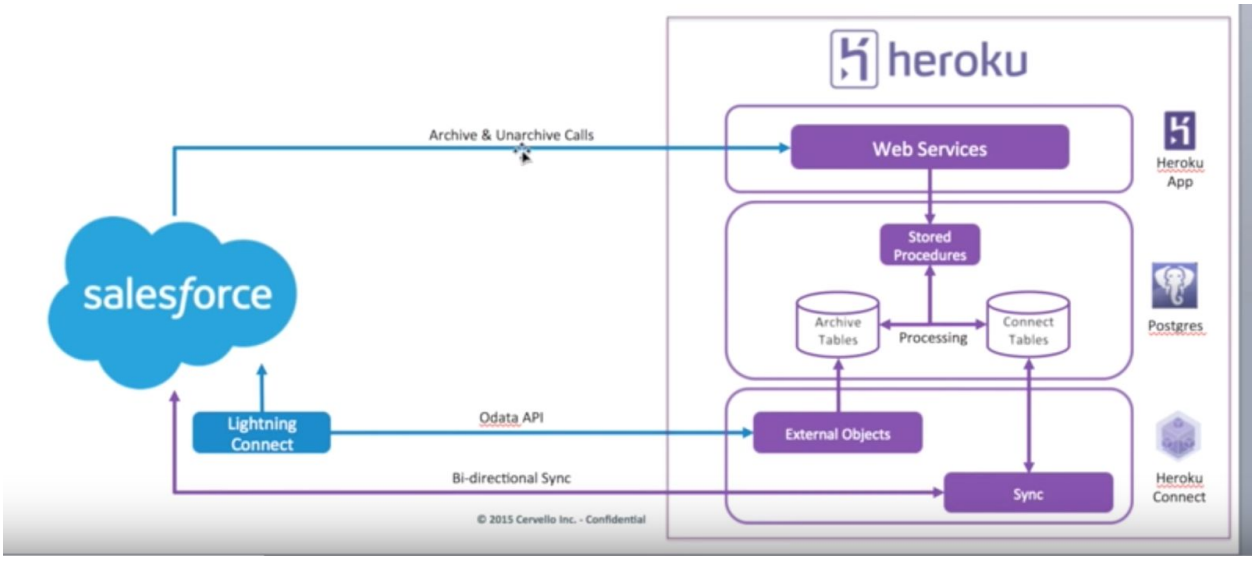


Fig: Org Copy - source odaseva.com

Archive using Heroku



Improving Data Quality

Bad Data

- Missing Records
- Duplicate Records
- No Data Standards
    - For example, California is listed as: CA, Calif, Cali, and, your favorite, “Surfin’, USA.”
- Incomplete Records
- Stale Data

- bad data is consistently linked with:
    - Lost revenue
    - Missing or inaccurate insights
    - Wasted time and resources
    - Inefficiency
    - Slow info retrieval
    - Poor customer service
    - Reputational damage

- It turns out that good data lets your company:
    - Prospect and target new customers
    - Identify cross-sell and upsell opportunities
    - Gain account insights
    - Increase efficiency
    - Retrieve the right info fast
    - Build trust with customers
    - Increase adoption by reps
    - Plan and align territories better
    - Score and route leads faster

Assess the Quality of Data

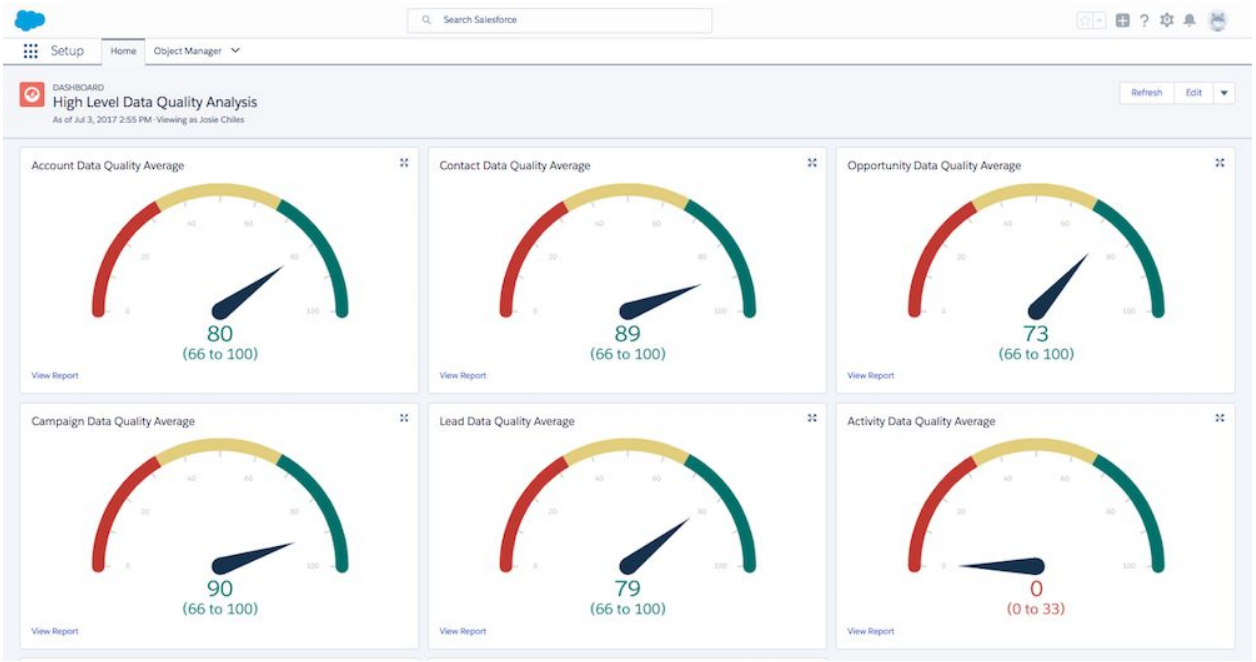
Determine How Your Business Uses Data

- What are your business objectives?
- What customer data is required to support those objectives?
- How are you using that customer data?
- Where is your customer data stored?

Review Data Quality Reports and Dashboards

Data Quality Dimension	How to Assess It
Age	Run a report on the Last Modified Date of records. What percentage of records have been updated recently?
Completeness	List the fields required for each business use. Then run a report that shows the percentage of blanks for these fields. You can also use a data quality app from AppExchange.
Accuracy	Install a data quality app from AppExchange. It can match your records against a trusted source and tell you how your data can be improved.
Consistency	Run a report to show the values used for date, currency, state, country, and region fields. How many variations are used for a single value?
Duplication	Use the Duplicate Management features in Salesforce and install a duplicate detection app from AppExchange.
Usage	Review the available tools and resources your business uses. Are you optimizing data use?

Install and Run the Data Quality Analysis Dashboards App



Improve Data Quality

Develop a Data Management Plan

Data Standard	Description	Example
Naming	Set naming conventions for records. Always include suffixes (Inc., Corp.)? Abbreviations?	Make sure that company names are never abbreviated except when the abbreviation is the standard name.
Formatting	Figure out how dates and money are represented.	Use dd/mm/yyyy for all date formats.
Workflow	Determine processes for record creation, reviewing, updating, and archiving. Determine all the stages a record goes through during its life cycle.	Route service requests associated with California companies to reps in California.
Quality	Set appropriate standards for data quality, including the ability to measure or score records. Put a value on age, completeness, usage, accuracy, consistency, and duplicates, along with any other quality or value metrics specific to your business.	Active leads are updated at least once per month.
Roles and Ownership	Determine who owns records, who's accountable for changes to data, and who's notified when there are changes to data.	Assign leads associated with California companies to sales reps in California.
Security and Permissions	Determine the appropriate levels of privacy for data. Make sure to comply with regulatory, legal, and contractual obligations.	Make sure that only regional team members can view confidential information on their leads.
Monitoring	Outline a process for ensuring quality control of data. Determine the frequency, scope, owners, and checks, including ways for updating data, preventing duplicates, merging records, adding records, and archiving records. Determine metrics that can be easily monitored in a dashboard.	Review leads without industry information on the first day of each month.

Implement Your Data Management Plan in Salesforce

1. Required Fields
2. Validation Rules
3. Workflow Rules
4. Page Layouts
5. Dashboards
6. Data Enrichment Tools
7. Duplicate Management
8. Custom Field Types

Duplicate Management

Duplicate Management helps you and your sales teams quickly and easily manage duplicates for:

- Business accounts
- Contacts
- Leads
- Person accounts
- Records created from custom objects

Manage Duplicates One at a Time

- Stop Users from Creating Duplicate Records
  When sales reps are in the process of creating or editing a record, alert them that the data they're entering duplicates an existing record. Or block sales reps from creating duplicate records altogether, whether or not they have access to the existing record. Activate duplicate rules and customize the settings.
- Show Duplicate Records in Lightning Experience
  If a record duplicates other records, you can alert sales reps when they open the record in Lightning Experience. Activate duplicate rules and add alerts to page layouts.
- Show Duplicate Records in Salesforce Classic
  If a record duplicates other records, you can alert sales reps when they open the record in Salesforce Classic by activating duplicate rules.

Manage Duplicates Globally

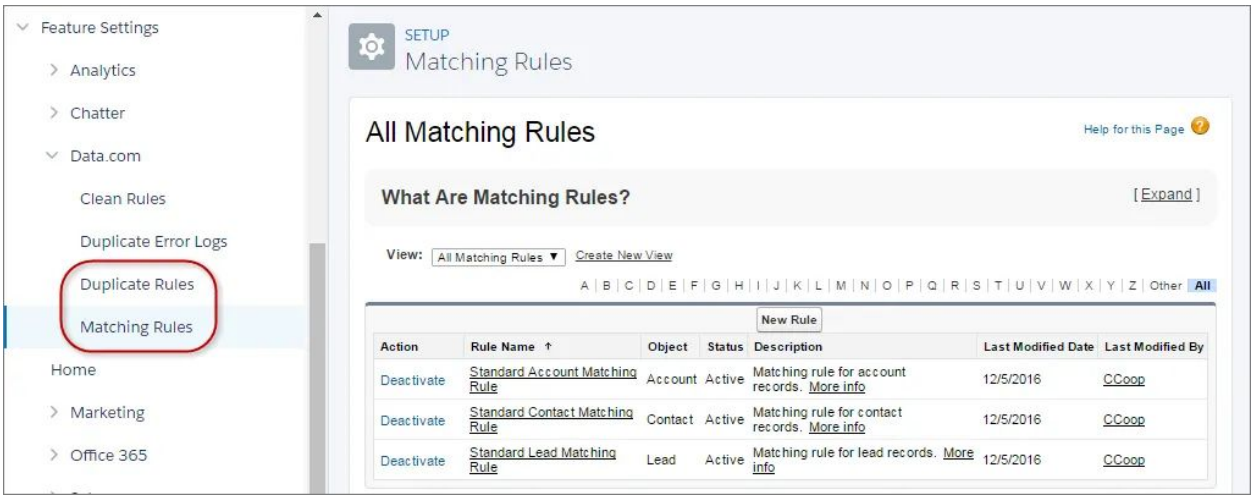
- Find Duplicates Across Your Org Using Duplicate Jobs in Lightning Experience
  Use duplicate jobs with standard or custom matching rules to scan your Salesforce business or person accounts, contacts, or leads for duplicates.
- Create Reports on Duplicate Records
  Use duplicate record reports to fine-tune your duplicate and matching rules and share the results of duplicate jobs.
- Manage Duplicates Using Duplicate Record Sets
  A duplicate record set is a list of items identified as duplicates. It's created when a duplicate rule or job runs. Let your Lightning Experience users merge duplicates by granting them access to duplicate record sets.
- View Error Logs for Duplicate Rules and Matching Rules
  Troubleshoot system errors that prevent duplicate rules or matching rules from running. Error logs are deleted after 90 days.

Rules for Identifying Duplicates and How to Handle Them

What It Is	What It Defines
Matching rule	The matching criteria to identify duplicate records. Salesforce comes with three standard matching rules: one for business accounts; one for contacts and leads, and another for person accounts.

Duplicate rule	When Salesforce engages matching rules and determines actions to take as it encounters duplicates. Depending on how you configure Duplicate Management, sales reps see an alert that they're about to create a duplicate. Or your reps are blocked from creating the duplicate altogether. If your company started using Salesforce in Spring '15 or later, we give you standard duplicate rules for business accounts, contacts, leads, and person accounts. If your company started using Salesforce in Winter '15 or earlier, like Maria, you create the rules on your own, which is easy.

Matching Rules



If you established your instance of Salesforce for Winter '15 or earlier, like Maria, you activate the standard rules you want to use.

Matching methods

- Exact
- Fuzzy

Matching Algorithms

Contact and Lead Field	Matching Algorithms	Special Handling	Example
First Name	• Exact • Initials • Jaro-Winkler Distance • Metaphone 3 • Name Variant	If the record contains a value for both First Name and Last Name fields, those values are transposed to consider possible data entry mistakes.	The first name is Luis and the last name is Antonio. The matching rule evaluates the first name as Antonio and the

			last name as Luis.
Title	● Acronym ● Exact ● Kullback-Liebler Distance	Considers acronyms and full titles.	The title is VP. The matching rule considers VP and Vice President.
Mailing Street	● Edit Distance ● Exact	Addresses are broken into sections and compared with those sections. Each section has its own matching method and match score. The section scores are weighted to determine one score for the field. This process works best with North American data. ● Street Name (Edit Distance, 50% of field's match score) ● Street Number (Exact, 20% of field's match score) ● Street Suffix (Exact, 15% of field's match score) ● Suite Number (Exact, 15% of field's match score)	Duplicate Management compares these two addresses. ● 123 Market St, Ste 100 ● 123 Market Dr, Ste 300 Only the street number and street name match. The field has a match score of 70 out of a possible 80. This comparison isn't a match.

Standard matching or custom rules can be selected for an object in duplicate rules to identify which rules should be used for finding the duplicates.

Duplicate Rules

- Gives warning that duplicate is present and can merge them
- Have option to block the creation

Reports

- Duplicate Record Items, so that newly created duplicate account records appear on the report. Then, she saves the report type.

Techniques for Optimizing Performance

Keeping Your Org Lean

- Only store what's needed.
    - Carefully study business requirements and design your application so that it's functional, and balances data storage and usability.
- Know your expected data growth rate.
    - Regardless of your organization's data volumes, follow coding and architecting best practices for projecting data volume growth.
    - Doing this allows you to test against base volume performance before going live, uncover potential risks, and even clarify what you need in your data-archiving and -purging strategy.
- Define criteria for data archiving and purging.

Using Mashups

- maintain large data sets in a different application, and then make that application available to Salesforce as needed.
- Salesforce supports the following mashup designs.
    i. External Website: The Salesforce UI displays an external website, and passes information and requests to it.
    ii. Callouts:
- Advantages:
    i. Data is never stale.
    ii. No proprietary method needs to be developed to integrate the two systems.
- Disadvantages:
    i. Accessing data takes more time.
    ii. Functionality is reduced.
        - For example, reporting and workflow do not work on the external data.

Defer Sharing Calculation

- defer sharing calculation permission to suspend and resume sharing calculations, and to manage two processes: group membership calculation and sharing rule calculation.
- let the recalculation process run overnight between business days or over a weekend.

Using SOQL and SOSL

- Use SOQL when:
    i. You know in which objects or fields the data resides.
    ii. You want to:
        - Retrieve data from a single object or from multiple objects that are related to one another
        - Count the number of records that meet specified criteria
        - Sort results as part of the query
        - Retrieve data from number, date, or checkbox fields
- Use SOSL when:

        i. You don't know in which object or field the data resides, and you want to find it in the most efficient way possible.
        ii. You want to:
            - Retrieve multiple objects and fields efficiently, and the objects might or might not be related to one another
            - Retrieve data for a particular division in an organization using the divisions feature, and you want to find it in the most efficient way possible
    - SOSL is generally faster than SOQL if the search expression uses a CONTAINS term
    - searching for a specific distinct term that you know exists within a field, you might find SOSL is faster than SOQL for these searches due to indexing
    - Executing a query with a WHERE filter that has null values for picklists or foreign key fields doesn't use the index, and should be avoided.
    - Using a large number of fields in query or search leads to a large number of permutations, which can be difficult to tune.

Deleting Data

- Covered in the data archival section above.

Search Architecture

- Salesforce provides search capabilities in many areas of the application, including:
    i. The sidebar
    ii. Advanced and global searches
    iii. Find boxes and lookup fields
    iv. Suggested Solutions and Knowledge Base
    v. Web-to-Lead and Web-to-Case
    vi. Duplicate lead processing
    vii. Salesforce Object Search Language (SOSL) for Apex and the API
- After a searchable object's record is created or updated, it could take about 15 minutes or more for the updated text to become searchable.
- Rest covered in the search section.
- To ensure good query performance as data volumes increase in an organization:
    i. take advantage of standard indexed fields whenever possible in your query designs
    ii. ask Salesforce Support to help you create custom indexes to speed up specific queries

Avoid Data Skew for Peak Performance

- Avoid assigning a single user as owner of a large amount of data whenever possible.
- we recommend that you avoid associating more than about 10k child records to a single parent record.
- You can create a larger number of accounts and modify your integration code or use a trigger on the child object to distribute the child records across this collection of "parking" accounts.
    - This will help to avoid the locking and performance issues you might experience with a highly skewed account.
- To avoid potentially long-running sharing recalculations caused by concentrating record ownership:
    - Design your ownership strategy from the beginning so that users own the data they create, then use the role hierarchy and sharing rules to provide access to others.

    - When you must have a single owner for a large amount of data, place that user in their own role at the top of the hierarchy, use sharing to provide access for other users, and don't move that user to a new role.

Managing Task Locks for Data Loads

- Ordering Task Loads by Account
    - A task record can reference an account in several different ways via the What or Who fields
    - If more than 10K under an account, you'll want to take that set of tasks and load them in a separate serial job using a controlled feed load
- Using Controlled Feed Loads
    - make sure to allow enough time between scheduled jobs so the jobs don't overlap
    - A job overlap might result in another scenario where tasks associated with the same account result in lock contention.

Field History Tracking

Field history data is retained for up to 18 months through your org, and up to 24 months via the API.

You can track the field history of custom objects and the following standard objects.

- Accounts
- Articles
- Assets
- Campaigns
- Cases
- Contacts
- Contracts
- Contract line items
- Entitlements
- Leads
- Opportunities
- Orders
- Order Products
- Products
- Service Contracts
- Solutions

➢ Since the Spring '15 release, increasing the entity field history retention period beyond the standard 18 to 24 months requires the purchase of the Field Audit Trail add-on
➢ Use Data Loader or the queryAll() API to retrieve field history that is from 18 to 24 months old.
➢ Changes to fields with more than 255 characters are tracked as edited, and their old and new values are not recorded.
➢ Tracked field values are not automatically translated; they display in the language in which they were made.
➢ Changes to custom field labels that have been translated via the Translation Workbench are shown in the locale of the user viewing the History related list

- Changes to date fields, number fields, and standard fields are shown in the locale of the user viewing the History related list.
- If a trigger causes a change on an object the current user doesn’t have permission to edit, that change is not tracked. Field history honors the permissions of the current user.
- In Lightning, you can see gaps in numerical order in the Created Date and ID fields. All tracked changes still are committed and recorded to your audit log.
    - However, the exact time that those changes occur in the database can vary widely and aren't guaranteed to occur within the same millisecond.
- Changes to time fields aren’t tracked in the field history related list.
- Field history tracking for accounts applies to both business and person accounts, so the 20-field maximum includes both types of accounts.

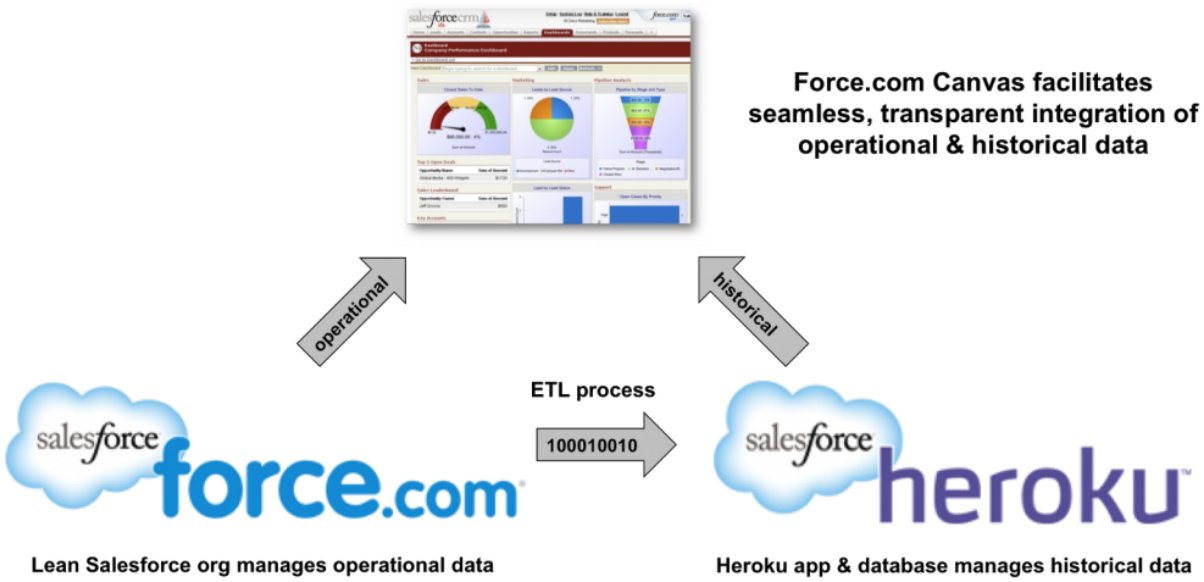
Field Audit Trail

- Field Audit Trail lets you define a policy to retain archived field history data up to 10 years from the time the data was archived.
- Use Salesforce Metadata API to define a retention policy for your field history for fields that have field history tracking enabled. Then use REST API, SOAP API, and Tooling API to work with your archived data.
- For information about enabling Field Audit Trail, contact your Salesforce representative.
- Field history is copied from the History related list into the FieldHistoryArchive big object.
- With Field Audit Trail, you can track up to 60 fields per object.
    - Without it, you can track only 20 fields per object.
- With Field Audit Trail, you retain archived field history data up to 10 years from the time the data was archived.
    - Without it, you retain archived data for only 18 months.
- You can include field history retention policies in managed and unmanaged packages.
- The following fields can't be tracked.
    - Formula, roll-up summary, or auto-number fields
    - Created By and Last Modified By
    - Expected Revenue field on opportunities
    - Master Solution Title or the Master Solution Details fields on solutions
    - Long text fields
    - Multi-select fields
- You can set field history retention policies on these objects.
    - Accounts, including Person Accounts
    - Assets
    - Campaigns
    - Cases
    - Contacts
    - Contracts
    - Contract Line Items
    - Entitlements
    - Individuals
    - Leads
    - Opportunities
    - Orders
    - Order Products
    - Price Books
    - Products
    - Service Appointments
    - Service Contracts
    - Solutions
    - Work Orders
    - Work Order Line Items
    - Custom objects with field history tracking enabled
- If platform encryption is enabled on the org, then AsyncSOQL on FieldHistoryArchive is not supported.

Best Practices for Large Data Volumes on Salesforce

Problem	Best Practices
Slow report on a large object	• Document your org's indexed fields. • Learn index selectivity rules. • Build reports that use indexes.
Slow bulk data load	• Cleanse and transform data pre-load. • Disable triggers, validations, and workflow rules pre-load. • Use the Bulk API.

Distributed Apps and Big Data



1. To maintain lightning fast access to operational data in the Sales Cloud, you can create an app on Heroku that manages historical data.
2. You keep your Sales Cloud org lean, are in compliance with your record keeping requirements, and can optimize the Heroku app and database to deliver the analytics you need on those historical records.
3. On top of it all, you glue together the UIs of your two apps using Force.com Canvas.
    a. This technology lets you embed the UI of a remote app within a Force.com app or customized Salesforce app, such that users don't even realize they are getting data from outside.

Salesforce Anti-Pattern

Pattern

- Customizing the Salesforce Schema
- Loading Large Amounts of Data
- Creating and Maintaining Local Database Backups
- Users and Feature Governance
- Business Process Configuration

Sidebar: Anti-Pattern

Data Loads Fail

- Some of the initial loads go very quickly–for example, the load of the parent Account object.
- However, subsequent loads of some related objects are plagued with lock contention errors causing batch retries, leading to painfully slow loads that often result in load failures.

➔ could easily work around this by pre-sorting the child records by parent Id in her CSV file to lessen the chance of parent record lock contention among parallel load batches

➔ By deferring the org's sharing calculations until after her data load, she could significantly increase both the load and sharing calculation performance.

Full Database Backups are Slow

➔ should be doing nightly incremental data backups–only backing up the data that is new or updated since the previous incremental backup.

➔ When doing this, she should use queries that filter records using SystemModstamp (a standard field in all objects that has an index) rather than LastModifiedDate field (not indexed).

Users Complain About Slow Reports

➔ Take the time to consider a great alternative–creating a library of public, controlled, and optimized reports that meet the requirements of XYZ's sales reps.

➔ Fewer reports to tune and maintain, plus high user satisfaction.

Formula Fields Unknowingly Slow Reports

➔ Salesforce performs many joins in the underlying query to run the report.

➔ Formula fields are a great tool when used in the proper context while understanding the scalability and response time implications for their usage.

- ◆ However, without proper understanding of how they function, formula fields can sometimes come back to bite you.

➔ using a trigger to populate denormalized related fields that would facilitate blazing report/query performance without runtime joins.

Sharing Recalculations Clobber Nightly Backups

- ➔ Sharing recalculations and Bulk API jobs both draw from the org's available pool of asynchronous processing threads.
- ➔ This competition causes both jobs to go slower than if they were scheduled to happen apart from one another.
- ➔ Also, if the full backup was changed to a much faster incremental backup, the time requirement for the backup (and thread usage) would be much smaller, allowing more time for the sharing recalculation.

Sales Reps Develop Productivity (and Eyesight) Issues

- Each Report refresh forces a sales rep to scan the entire list of Leads again and again to pick targets that are new or changed since the most recent refresh.

- ➔ Salesforce has many easy-to-implement workflow solutions such as workflow rules for this type of process.

Data Governance

ASSESS THE STATE OF YOUR DATA

1. Who is using customer data?
2. What are the business needs for data?
3. Which data is used the most?
4. How is the data being used?

DEVELOP YOUR GOVERNANCE PLAN

1. Data Definitions
    a. Develop and apply standards for naming and organizing data
2. Quality Standard
    a. Set appropriate standards for data quality, including the ability to measure or score records.
3. Roles & Ownership
    a. Responsible owns the data
    b. Accountable must sign off on or approve changes
    c. Consulted can provide information and support
    d. Informed needs to be notified, but not consulted
4. Security & Permissions
    a. Assign consistent and appropriate levels of privacy, confidentiality, and verified access to comply with regulatory, legal, and contractual obligations
5. Quality Control Process
    a. Outline the structure of a quality control process, including frequency, scope, owners, and checks. Include methods for cleansing, de-duping, blocking duplicates, merging, and adding to records
    b. Create a policy for data retention and archival, supported by metrics that can be tracked using simple dashboards.

IMPLEMENT YOUR GOVERNANCE PLAN

Make data entry and management easy.

1. Internally sourced:
    - Proprietary financial data
    - Legally sensitive data, like social security or payment numbers
    - Data unique to your organization, such as internal product SKUs
2. Externally sourced:
    - Business card data
    - Company profiles and locations
    - Industry classification details
    - Corporate hierarchy mapping

Keep data lean and effective with regular housekeeping.

Prepared by: Tahsin Zulkarnine, tahsinz@gmail.com

1. Standardize your data
2. Eliminate duplicates.
3. Compare your data against a referential data source.
4. Create rules and schedule automatic cleanings.
5. Track, report, and learn.

Potential Challenges with Traditional Data Governance Strategies

1. Data governance doesn't fit into the overall IT governance effort.
2. Data governance efforts are ignored.
3. Data governance is too onerous.
4. Data governors are too slow to respond
5. Data governors aren't perceived as providing value

Agile/Lean Data Governance Practices

1. Valued Corporate Assets.
2. Scenario-Driven Development.
3. Include data professionals as active participants on development teams.
4. Educate developers.
5. Adapt the Process.
6. Align Team Structure With Architecture.
7. Align HR Policies With IT Values.
8. Align Stakeholder Policies and IT Values.
9. Business-Driven Project Pipeline.
10. Embedded Compliance.
11. Flexible Architectures.
12. Pragmatic Governance Body.
13. Promote Self-Organizing Teams.
14. Risk-Based Milestones.

Data Governance Success Factors

1. Recognize that IT governance is the real goal.
2. The governance effort must be owned.
3. Have clear, quantifiable goals.
4. Measure and honestly report the results.
5. Less is more.
6. Educate the people affected.

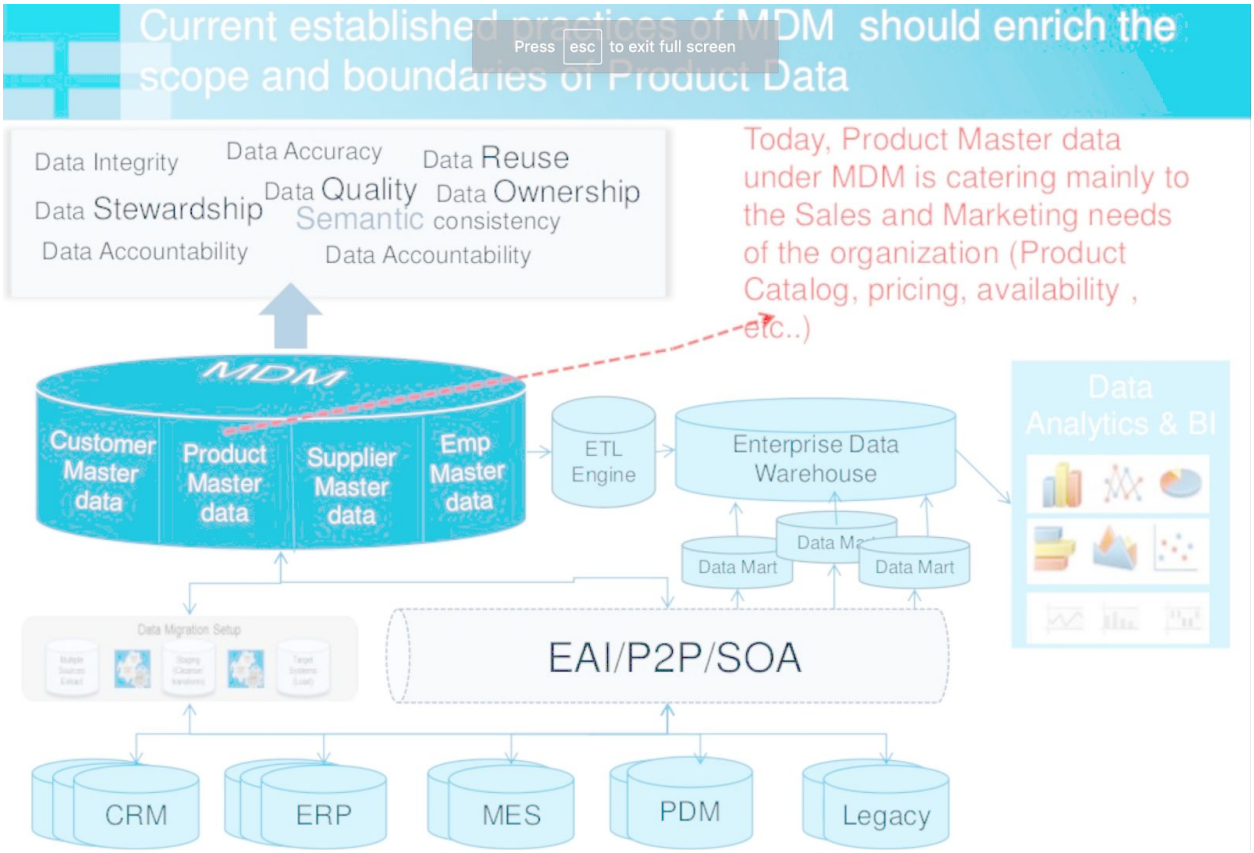
Master Data Management(MDM)

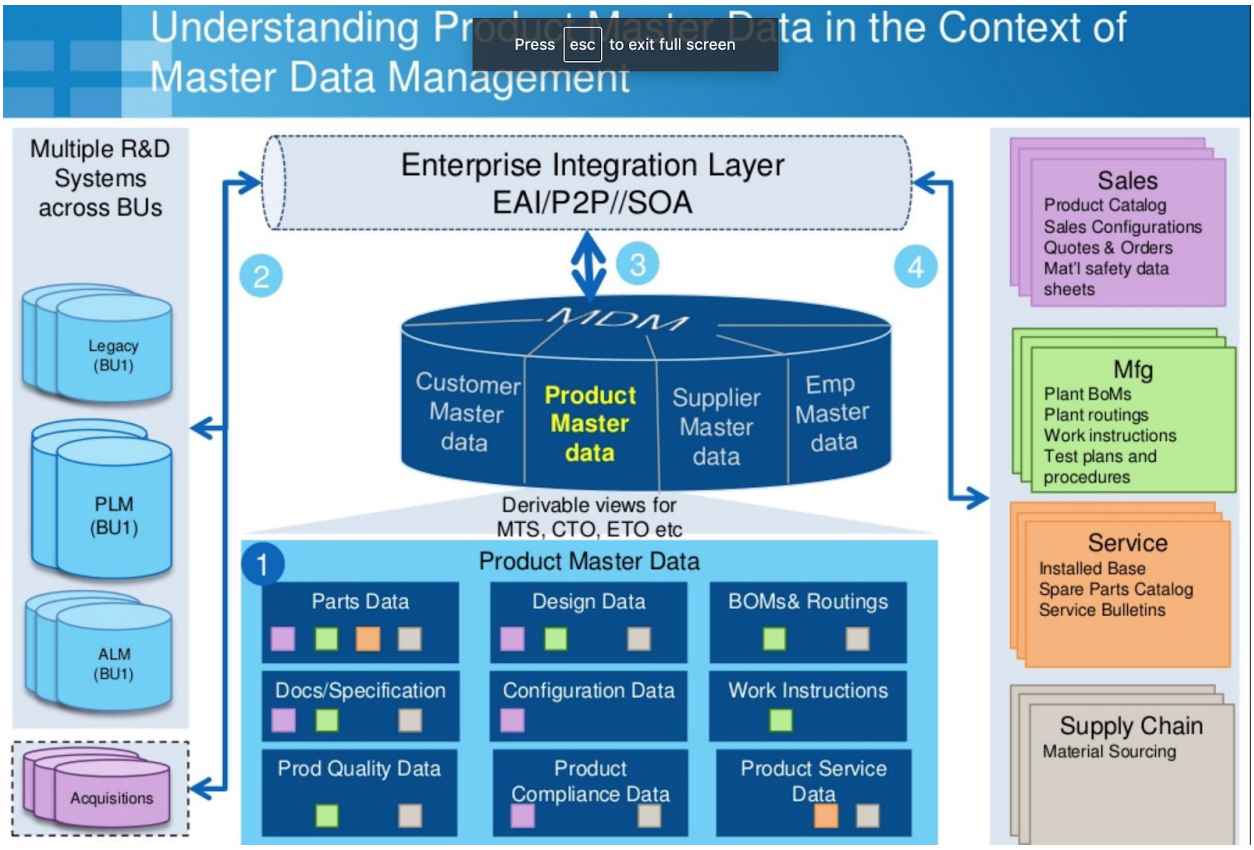
REFERENCE DATA MANAGEMENT

- used to categorize other data within applications and data bases.
    - We usually refer to this data as look-up, code table or domain values
    - Some of the best examples are – state codes, country codes, gender codes, marital status type codes etc.
- Errors in the reference data can have a major business impact
- Quality issues in reference data have ripple effect and can cause major issues in the downstream applications.

Key features of Reference Data Management Systems

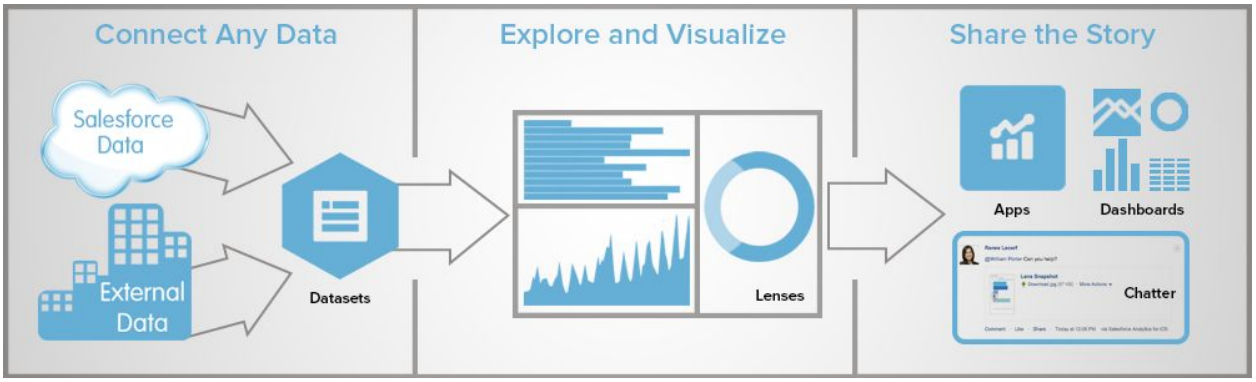
1. Robust interface controlled by role based security to support collaborated authoring of reference data.
2. Ability to manage and map relationships between different reference data sets which exist in an enterprise.
3. Versioning and auditing capability
4. Hierarchy management
5. Provision reference data via web services to support SOA framework
6. Ability to publish reference data
7. Efficient load/extract functionality
8. Good error tolerant search capability

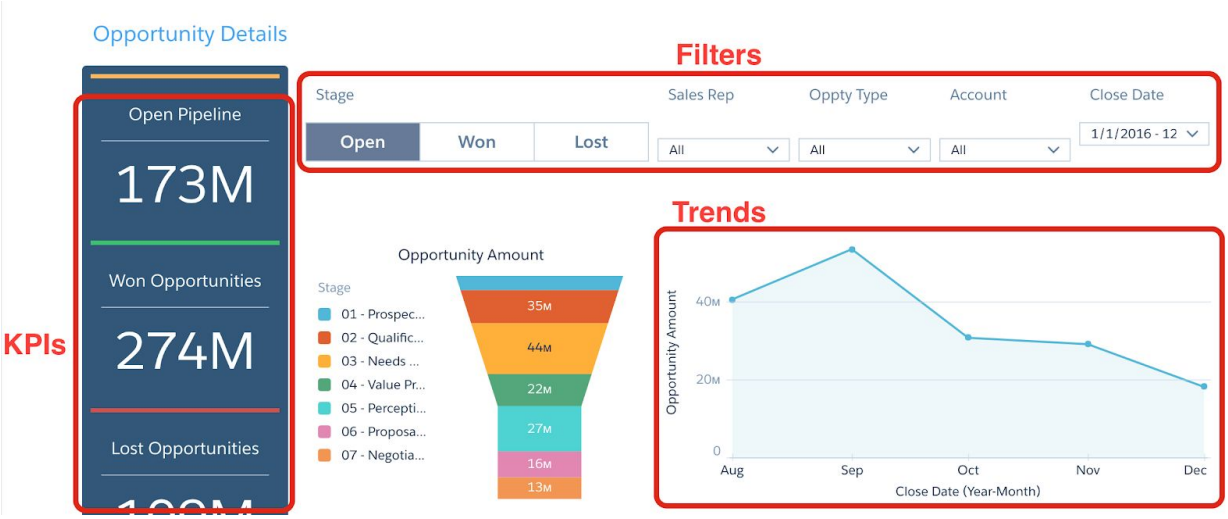

Current established practices of MDM should enrich the scope and boundaries of Product Data
Press esc to exit full screen
Data Integrity
Data Accuracy
Data Reuse
Data Stewardship
Data Quality
Data Ownership
Semantic consistency
Data Accountability
Data Accountability
Today, Product Master data under MDM is catering mainly to the Sales and Marketing needs of the organization (Product Catalog, pricing, availability , etc..)
MDM
Customer Master data
Product Master data
Supplier Master data
Emp Master data
ETL Engine
Enterprise Data Warehouse
Data Analytics & BI
Data Mart
Data Mart
Data Mart
Data Migration Setup
EAI/P2P/SOA
CRM
ERP
MES
PDM
Legacy


Understanding Product Master Data in the Context of Master Data Management
Press esc to exit full screen
Multiple R&D Systems across BUs
Legacy (BU1)
PLM (BU1)
ALM (BU1)
Acquisitions
Enterprise Integration Layer
EAI/P2P//SOA
2
3
4
MDM
Customer Master data
Product Master data
Supplier Master data
Emp Master data
Derivable views for MTS, CTO, ETO etc
1
Product Master Data
Parts Data
Design Data
BOMs& Routings
Docs/Specification
Configuration Data
Work Instructions
Prod Quality Data
Product Compliance Data
Product Service Data
Sales
Product Catalog
Sales Configurations
Quotes & Orders
Mat'l safety data sheets
Mfg
Plant BoMs
Plant routings
Work instructions
Test plans and procedures
Service
Installed Base
Spare Parts Catalog
Service Bulletins
Supply Chain
Material Sourcing

Analytics

- Go to developer.salesforce.com/promotions/orgs/wave-de.
- Fill out the form using an active email address. Your username must also look like an email address and be unique, but it doesn't need to be a valid email account. For example, your username can be yourname@waverocks.de, or you can put in your company name.
- After you fill out the form, click Sign me up. A confirmation message appears.





Analytics Notifications

1. Select Set Notification to open the notifications panel.
2. In the Notifications panel, select Equals or is less than from the upper drop-down menu.
3. Set the notification to run at 5:00 AM on Every Weekday.
4. Click Save and Run.

Put the Analytics Home Page to Work for You

- When the panel is in edit mode, you can access additional options.
- Track on Analytics home page lets you set whether the notification appears on your home page.
- Active lets you control whether Analytics runs your query and sends notifications.

Follow That Dashboard

1. Click  to open the annotations panel.
2. Click 

.

Other operations

1. Download Data from a Dashboard Widget
    a. Only get the data for the widget query
2. Take the Conversation to Chatter
    a. In the Share dialog, click the Post to Feed tab.
3. Share Dashboards with Others in Your Org
4. Use Annotations to Highlight Results
5. Present a Dashboard in Full-Screen Mode

Prepared by: Tahsin Zulkarnine, tahsinz@gmail.com