



HANDS-ON ACTIVITIES

DATA ARCHITECTURE & MANAGEMENT

FEBRUARY, 2018



1. MIGRATION OF HISTORICAL DATA

Understand data movement best practices, considerations, and appropriate tools to use in a Large Data Volume (LDV) environment.

USE CASE

The Auto Division of Universal Containers is in the process of implementing their next-generation customer service capabilities using Salesforce Service Cloud. Their legacy CRM application is out of support and is unable to meet the new capabilities required by the customer service agents and managers. Due to compliance and regulatory reasons, one of the key requirements is to move 7 years of historical data from their legacy CRM application to the Salesforce platform. The legacy CRM application will be decommissioned once the data and application are migrated to Salesforce.

DETAILED REQUIREMENTS

Users, Accounts (Dealer and Individual), Contacts, and Cases are key data that are targeted for migration from the legacy CRM application. All closed cases for the last 7 years along with contacts are required to be available in Salesforce. VIN (Vehicle) data is required on all open cases, but not on closed cases.

Estimated Datasets to be Migrated

- 16 MM cases to be migrated
- 20 MM VINs
- 5 MM contacts
- 60K Dealers
- 100K Dealers' employees
- 100K Attachments
- 50K Knowledge Articles



For this build, consider an automotive customer service model using the following:

1. Standard Objects: Accounts, Contacts, Cases, Knowledge articles, FeedItems.
2. Custom Objects: VIN, Sub-Cases, Parts, Warranty.
3. Notes data from the legacy application will be loaded into the FeedItem Object.
4. Closed cases (historical cases) with partial related information.
5. Dealers to be mapped as Business Accounts.
6. Dealers' employees to be mapped as contacts to associated Business Accounts.
7. Contact Data to be cleansed prior to loading into Salesforce.

PREREQUISITE SETUP STEPS

1. Configure an ETL tool of choice, like Informatica, Jitterbit, or Talend, which has adaptors/connectors to on-premise databases, like Oracle, MySQL, SQLSERVER, etc., as well as pre-built connectors to Salesforce, i.e., Informatica Cloud.
2. Configure Data Loader for loading Attachments, Knowledge Articles, and reference data.
3. Create two on-premise databases: one for source data and the other for staging.
4. Document the Data Migration Solution Flow showing interaction between systems.
5. Design the SFDC data model.
6. Estimate data storage requirements for the staging database and Salesforce.
7. Generate test data to set up your source system database.



CONSIDERATIONS

1. Identify types of records to migrate (Accounts, Contacts, Cases, Attachments, etc.).
2. Determine the volume of records for each type.
3. Determine what data **must** be persisted in Salesforce and what data **can** be performed via a lookup to the on-premise application.
4. Is there reference data—countries, business locations or sites, counties, zip codes, SIC codes, etc.—which needs to be represented in the Salesforce.com data model?
5. Map fields from each system into the SFDC data model. Use a mapping template.
6. Mapping legacy application structures to the Salesforce data model is pretty time consuming, requiring a Subject Matter Expert who has in-depth knowledge of the business processes and workings of the legacy system.
7. Determine required data transformations: date formats, address and phone number formatting, picklist values, truncation of text, values, or codes to transform to Salesforce.com references, etc.
8. Determine what special restrictions apply to the client's data. For example, PII compliance? Is it permissible to load production data into a sandbox?
9. Establish how records of each entity type will be assigned to users in Salesforce.com. Owner Id is a very important field.
10. Establish the need for loading CreatedById and CreatedDate. You might need to open the case with Salesforce to enable it in all sandboxes and Production.
11. Identify the sequence of loading the required objects in Salesforce.



Solution Drivers and Benefits

Efficiency and quality are the guiding principles of our approach to data migration. This approach to data migration provides three main benefits:

- **Clean Data:** Data needs to be cleaned/de-duped outside Salesforce and before our initial loads.
- **Data Load Quality:** Data needs to be reviewed in a staged location before loading into Salesforce. Only necessary data should be loaded, instead of the whole set; bringing in extraneous data just dilutes quality.
- **Efficient Load Process:**
 - A staging area can be used to store Salesforce IDs to minimize queries during the load, which has a positive effect on the performance.
 - Salesforce target objects and functionality are analyzed to avoid activating triggers and other processing during the loads.

The key factor for successful data migration is collaboration between the technical and business teams. Both groups are responsible for data validation during test runs, as well as in Production.



SOLUTION

The end-to-end solution comprises the source system's databases, a staging database, and Salesforce. The staging database consists of two layers: the Transformation Layer and Target Layer.

- The Transformation layer is a set of intermediate database structures used for performing transformation and data quality rules. Only transformed and cleansed data will be loaded into the Target Layer.
- The Target Layer has tables structured identical to the Salesforce Objects—data types may differ depending on the database platform used.
- Data from the Target Layer will be loaded into Salesforce via Informatica cloud or any other ETL Cloud capable tool of choice.

SOLUTION DESCRIPTION

Initial Steps

1. Turn off triggers, workflow rules, validation rules, and sharing rules.
2. Code your ETL logic using the source-to-target mapping document.
3. Incorporate all of your triggers, workflow rules, and validation rules within your ETL logic.
4. Load reference data.
5. Define knowledge article types and categories in Salesforce:
 - FAQs and Operating Procedures as Article Types
 - Auto, Power Equipment as Categories
6. Eliminate migrating articles unused for a year and longer.
7. Migrate knowledge articles using Salesforce Data Loader.

Testing: Unit and Integration

1. Identify the appropriate load sequence. Consider relationships across all objects.
2. Run sample migration of a small subset of records from each legacy application; extract, transform, and load into SFDC.
3. Debug issues and update scripts as needed.
4. Run sample migration tests until they run clean with no errors.



Testing: Full Load and Performance Testing

1. Run full migration into sandbox. Full migration = extract, transform, and load all records.
2. Prepare reports on records in the source system or extracts, and the records loaded into Salesforce.com. Identify any missing data.
3. Fix issues and repeat until there are no errors.
4. Run full migration in a full sandbox environment.
5. Validate data from a technical and business perspective.

Restart Strategy

Create a job_run table having a job_id and batch_number for each associated Salesforce Object in the staging database. Load records in increments of 1M. Update the status of the Job ID associated with the object once successfully completed. This setup will allow for a restart at the point of failure (job level) and will avoid having to start the entire load from the beginning.

If the migration fails in loading any chunk of data, the issue needs to be identified and resolved. Also, the migration of the set of data that was being loaded needs to be restarted with Upsert turned on.



2. SCALABILITY

Describe the performance, scalability, and maintainability considerations for the data model.

USE CASE

Universal Containers is an online discount shopping portal where about 700,000 manufacturers and traders offer items for sale at discounted rates to consumers. Each trader can deal with one or more items. Currently the site supports 120 different categories with close to 100,000 items. Every day, about 4 million consumers visit the Universal Containers site and place close to 100,000 orders. Each order contains up to five items on average. The consumers can purchase items as a guest or as a registered user to receive credits and loyalty rewards for purchases. The site has a subscriber base of 2 million registered consumers and a total of 4.5 million consumers. The company fulfills orders from 1000 distribution centers located throughout the country. The traders can also fulfill orders directly from their own manufacturing/distribution centers. Major credit cards and PayPal are supported for payment. Each trader's bank account details are available with Universal Containers. Once the item is shipped to the consumer either by a Universal Container's distribution center or a trader, the funds are deposited directly to the trader's bank account after deducting Universal Containers' commission.

The company supports web, phone, and email channels for the support cases. 10,000 cases are created on average every day. There are 1,000 Support Reps and 100 Support Managers. Reps are assigned to managers based on geographic region. All Support Reps operate out of regional offices. Each region has two managers.



DETAILED REQUIREMENTS

Legacy systems for item master, trader database, and customer master are hosted on the Oracle platform. Billing order and inventory information is stored in an ERP system. Universal Containers uses Oracle Service Bus as middleware. Their main website runs on Scala and is deployed on Amazon.

1. The company wants to deflect as many support inquiries as possible using self-service for traders and consumers.
2. It is planning to use Salesforce Knowledge for answering common consumer questions.
3. Registered users should be able to view their purchase history from the last 3 years.
4. Users should be able to leave feedback for items which they have purchased. These comments are public comments, and up to 20 comments are displayed against each item on the website.
5. The Support reps would like to view order history for consumers who have created cases. They may occasionally look at comments as well as past returns for previous purchases by consumer.
6. The traders should be able to log into Universal Containers' site and view pending orders. They should also receive notification of a new order via email.
7. Traders should be able to log into the Universal Containers support site and get support related to their pending payment and inventory-related questions.
8. The management would like to use dashboards to track trends for various item categories. The dashboard should include data such as the number of items sold, the number of items returned, and a total value of items sold for every week for a reporting period.
9. Management wants to keep three years' worth of cases in the system.
10. The management is very much concerned about the performance of the application, and would like to receive recommendations around Salesforce application design to maximize performance.



PREREQUISITE SETUP STEPS

1. Requires a Knowledge Feature License for all Knowledge authors and contributors.
2. Requires a Community Cloud License for 2 million registered users and 700k traders.
3. Requires a Service Cloud License for 1000 Support reps and 100 Support Managers.

CONSIDERATIONS

1. What data should be stored in Salesforce? Use systems of record as much as possible and only bring reportable data into Salesforce.
2. How do you organize Knowledge Articles around Item Categories?
3. What is the method of access to Item and Customer master data?
4. What is the method of access to Purchase History?
5. What licenses would you use to support a high volume of 2 million registered customers and 700k traders?
6. How do you generate dashboards?
7. Data Archival Strategy: Archive older data to provide efficient access to current data.
8. How do you support high-volume page visits and bandwidth?
9. About 700k Traders and 2 million registered customers need to be able to access the site.



SOLUTION

For Consumers and Traders

- Model consumers using Person Accounts with a High-Volume Customer Community user license.
- Model traders using Business Accounts with trader users as contacts in those accounts. They will be using the High-Customer Community License as well as the requirement of purchase history, and cases can be solved using sharing sets.
- Integrate customer master data and item master data into Salesforce. Store key fields of master data in Salesforce. Use Oracle Service Bus to perform real-time integration to this.
- Save 3 years' of rolled-up order data in Salesforce.
- Do not store item-level data in Salesforce. Only store the rolled amount information at order level. Use Oracle Service Bus to do the data transformation.
- Store item categories as data categories in Salesforce and map it to match item hierarchy. Assign articles to corresponding data categories.
- Enable feed tracking on product/item so users can enter comments to the item and make it available to the public.
- Build a self-service community and provide Knowledge and a discussion search to create deflection and reduce the volume of cases.
- Create a data archiving solution using Apex Jobs or ETL solutions to:
 - Purge cases older than 3 years.
 - Purge item comments above 20.
 - Purge orders older than 3 years.
- Use sharing sets to efficiently share:
 - Account-level orders and cases to traders.
 - Users' purchase history and cases to consumers.
- Create two profiles to differentiate trader user from consumer user.
- The support site works with Universal Containers using shared single-sign-on to provide a seamless user experience.



For Support Reps and Managers

- Use Service Cloud for support reps and managers.
- Enable SSO for internal support users.
- Create region-based roles and queues to manage assignment rules.
- Use an external object to view items related to a particular order for support reps.
- Use SSO-based deep links to the Billing Order system built in Oracle to access previous purchases and returns.



3. LDV Considerations in Schema Design and Patterns to Access Data

Identify best practices to minimize the impact of LDV.

USE CASE

The insurance division of Universal Containers has close to 10 million subscribers. Each subscriber can hold one or more policies, the average being 2 policies per customer: one annuity and one term life. Each policy has one or more beneficiaries, the average being 4 beneficiaries per policy. Each annuity policy will have a monthly disbursement which typically starts at the age of 60 until the death of insured. For term life policy, there is disbursement after the death of the insured person.

Truelife wants to implement Service Cloud functionality. Customers will call for questions regarding their policies, beneficiary information, disbursement and so on. TrueLife executive management is interested in both operational as well as trend reporting for case data in Salesforce.

DETAILED REQUIREMENTS

1. The policy data in Salesforce is read-only.
2. The policy data is distributed in multiple legacy systems. The member master information holds contact data. The policy master holds policy and beneficiaries data. The Finance master holds disbursement data.
3. Enterprise uses Cast Iron as Middleware.
4. There is an existing web application which shows policy disbursement details.
5. Policy master is Odata compliant.
6. Members can request changes in beneficiaries, addresses, etc. The request needs to be recorded in Salesforce and communicated to the Policy Master, where actual changes will happen.

PREREQUISITE SETUP STEPS

1. Get a Salesforce Developer edition and set up case records.



CONSIDERATIONS

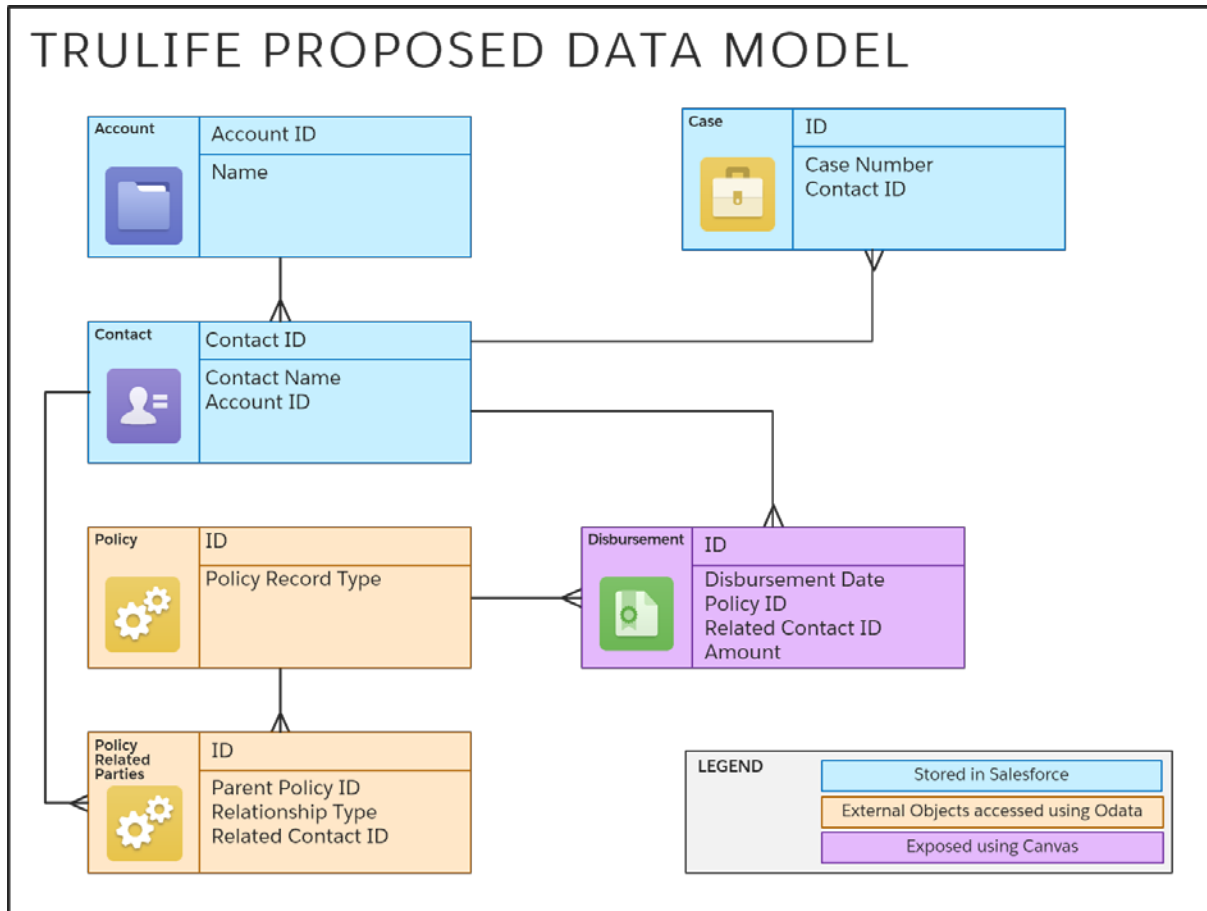
1. What data should be stored in Salesforce?
2. Show an entity relationship diagram showing which objects are stored in Salesforce and which are accessed virtually.
3. What mechanism accesses policy data?
4. What mechanism accesses beneficiary data?
5. What mechanism obtains disbursement data?
6. How do you complete a beneficiary change?
7. How do you complete an address change?
8. How do you generate trend reports and dashboards for cases?



SOLUTION

Policy and beneficiary data should be accessed using an external object in Salesforce.

Disbursement data should be shown using Canvas.



For address and beneficiary changes, either the external WSDL should be consumed in Salesforce and callouts should be used, or middleware should be used to get the data from Salesforce and consume an external WSDL.



SOLUTION DESCRIPTION

Initial Steps

1. Create a connected App in Salesforce.
2. Create an external Object in Salesforce.
3. Create an Apex class using WSDL to Apex.
4. Create an Apex class to consume REST API service in Salesforce.
5. Create a workflow to send outbound messages to middleware.
6. Build a client to access Salesforce data using REST API.
7. Create a reporting snapshot in Salesforce.

For points 1 to 6, you can practice build exercises once you complete training in the Integration domain.

Detailed Steps

The following section describes steps to be implemented to generate case trend reports in Salesforce. The exercise contains three main steps:

1. Create a new custom report that includes the fields you want to load as records into a target object.
2. Create a new custom object in which to store the records loaded from the source report.
3. Create fields on the target object that will receive the source report's results when the reporting snapshot runs.

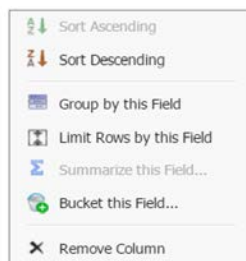


Step 1. Create new custom report "Cases by Product Created Today."

- Go to the Reports tab in Salesforce.
- Click the **New Report...** button.
- Click the + (plus icon) near the Customer Support Reports folder to show available report types for Cases.
- Select the **Cases** report type.
- Click the **Create** button.
- Salesforce displays a new Unsaved Report screen.
- Create a tabular report with following fields:
 - Case Owner
 - Subject
 - Product Category
 - Opened Date
 - Closed Date
 - Account Name

Case Owner	Subject	Product Category	Opened Date	Closed Date	Account Name
Admin User	Printer jam	Printers	10/26/2015		- LexCorp
Admin User	Printer jam	Printers	10/26/2015		- Colonial Movers
Admin User	Printer jam	Printers	10/26/2015		- Thatherton Fuels
Admin User	Desktop support	Desktops	10/26/2015		- Primatech
Admin User	Product Question	Laptops	10/26/2015		- Roberts Hotels & Resorts
Admin User	Product Question	Laptops	10/26/2015		- General Services Corporation
Admin User	Product Question	Laptops	10/26/2015		- Colaba Technologies
Admin User	Brand new A200 Desktop will not boot	Desktops	10/26/2015		- Enmore Installations

- Add grouping by product by clicking the **Product Category** column and selecting **Group by this Field**.



NOTE:



NOTE:

When a reporting snapshot runs, it can add up to 2,000 new records to the target object. If the source report generates more than 2,000 records, an error message is displayed for the additional records in the Row Failures related list. You can access the Row Failures related list via the Run History section of a reporting snapshot detail page.



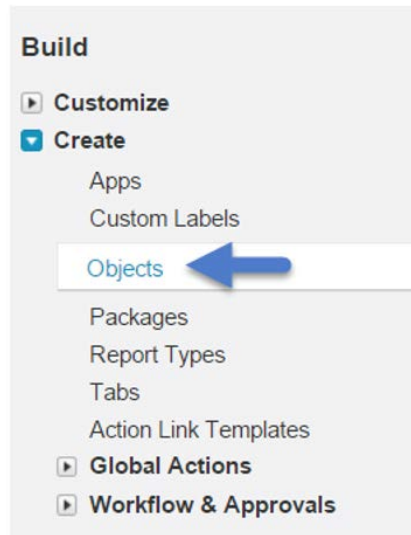
- Click **Save** to save the report with the name "Cases by Product."

A screenshot of the "Save Report" dialog box in Salesforce. The dialog has a title bar with "Save Report" and "Help for this Page" with a question mark icon. It contains four input fields: "Report Name" with the value "Cases by Product", "Report Unique Name" with the value "Cases_by_Product", "Report Description" which is empty, and "Report Folder" with a dropdown menu showing "My Personal Custom Reports". At the bottom, there are three buttons: "Save", "Save and Run Report", and "Cancel".

- Schedule the report to run daily.

Step 2. Create a new custom object to hold reporting snapshot data.

- From Setup, select **Create | Objects**.





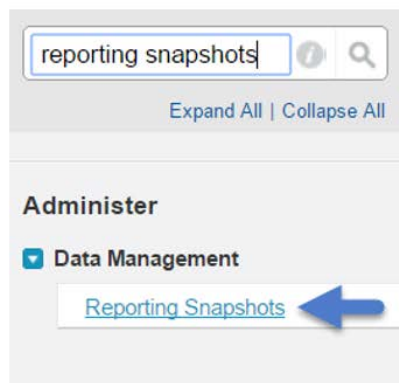
– Create a new custom object called Case_Snapshot with following fields:

Field Label	Field Type
Name	Auto Number
Case Created Date	Date/Time
Total Cases Created	Number
Product	Picklist

Action	Field Label	API Name	Data Type
Edit Del	Case Created Date	Case_Created_Date__c	Date/Time
Edit Del	Name	Name__c	Auto Number
Edit Del Replace	Product	Product__c	Picklist
Edit Del	Total Cases Created	Total_Cases_Created__c	Number(18, 0)

Step 3. Create a new custom reporting snapshot.

- From Setup, enter “Reporting Snapshots” in the Quick Find box, then select **Reporting Snapshots**.



- Click **New Reporting Snapshot**.
- Enter a name, a unique name, and a description for your reporting snapshot.



- Choose a user in the Running User field by clicking the **lookup** icon.

Edit Reporting Snapshot [Save] [Save & Edit Field Mappings] [Cancel]

Identification

Reporting Snapshot Name: Cases by Product

Reporting Snapshot Unique Name: Cases_by_Product

Description: [Empty text area]

Running User: Colin Daly [lookup icon]

Source Report: Cases by Product

Target Object: Case_Snapshot

- The user in the Running User field determines the source report's level of access to data. This bypasses all security settings, giving all users who can view the results of the source report in the target object access to data they might not be able to see otherwise.
- Only users with the “Modify All Data” permission can choose running users other than themselves.
- Select the **Cases by Product** report from the Source Report drop-down list.

Edit Reporting Snapshot [Save] [Save & Edit Field Mappings] [Cancel]

Enter information about this reporting snapshot.

Reporting Snapshot Name: --None--

Reporting Snapshot Unique Name: My Personal Custom Reports

Description: Cases by Product

Running User: [Empty text area]

A reporting snapshot runs a s [Empty text area]

Source Report: --None--

Target Object: --None--

a target object. Select a source report and target object.

- Click the **Save & Edit Field Mappings** button.
- Under Grouping Level, choose **Grouping 1: Product Category**.
- For Target Object, choose **Case_Snapshot** and click.



- Enter the following mapping:
 - Execution Time: map to case Created Date (Date/Time).
 - Product Category: map to Product (Picklist).
 - Record Count: map to Total Cases Created (Number(18, 0)).

Fields from Source Report Cases by Product	Map to	Fields in Target Object Case_Snapshot
Execution Time ▾	➔	Case Created Date (Date/Time)
--Load No Data-- ▾	➔	Case Name (Text(80))
(No fields with compatible data type)	➔	Object Access Level (Lookup(User Record Access))
(No fields with compatible data type)	➔	Owner (Lookup(User,Group))
Product Category ▾	➔	Product (Picklist)
Record Count [Record Count] ▾	➔	Total Cases Created (Number(18, 0))

- Click **Save**.

Step 4. Schedule a reporting snapshot.

- From Setup, type “Reporting Snapshots” in the Quick Find box, then select **Reporting Snapshots**.
- Select the **Cases by Product** snapshot which you created earlier.
- Scroll down to the Schedule Reporting Snapshot section and click **Edit**.
- Enter the following details:
 - For Email Reporting Snapshot, click the **To me** box.
 - For Frequency, pick **Daily**.
 - Input your desired Start and End dates.



Schedule Reporting Snapshot Cases by Product

Schedule Reporting Snapshot

Email Reporting Snapshot To me To others...

Scheduled Run

Frequency

Daily Weekly Monthly

Every weekday Every day

Start [12/1/2015]

End [12/1/2015]

Preferred Start Time [Find available options...](#)

Exact start time will depend on job queue activity.

- Click **Save**.

Step 5. Verify snapshot works as desired.

When the report runs as scheduled, it creates records in the custom object you defined in an earlier step.

Using this custom object to generate further trend reports makes it possible to run trend reports on large data volumes of cases. So, for trend reports, only custom object summary records need to be used instead of reporting going through a huge number of case records created during that time period.