



Boomi

# Developer 1

Section 1

Version 7.1

# Copyrights and Trademarks

This guide contains proprietary information protected by copyright and/or other legal grounds. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Boomi, Inc. ("Dell Boomi").

The information in this document is provided in connection with Dell Boomi products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Dell Boomi products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, DELL BOOMI (TOGETHER WITH DELL INC. AND ITS DIRECT AND INDIRECT SUBSIDIARIES) ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL DELL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ANY OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Dell Boomi makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Dell Boomi does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Boomi, Inc.  
Attn: Legal Dept.  
[legalnotices@dell.com](mailto:legalnotices@dell.com)

With a copy to:

Boomi, Inc., Legal Department, 1400 Liberty Ridge Drive, Chesterbrook, PA 19087

## Trademarks

Copyright © 2017 Boomi, Inc. All rights reserved. Dell, the Dell logo, Dell Boomi, Boomi, AtomSphere, Atom, and AtomSphere Integration Cloud are trademarks of Dell Inc. and/or its subsidiaries in the United States and/or other countries. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products.



# Table of Contents

---

Copyrights and Trademarks.....	2
Section 1 - SaaS Training .....	4
Exercise 1: Set up folders .....	5
Exercise 2: Download process endpoints from the Process Library .....	6
Exercise 3: View the Salesforce read Connector.....	7
Exercise 4: Create a Salesforce query Operation.....	8
Exercise 5: Set the Salesforce runtime parameter.....	12
Exercise 6: Test the process .....	14
Exercise 7: View the database write Connector.....	16
Exercise 8: Create a database write Operation.....	17
Exercise 9: Create a database write Profile .....	18
Class Activity 1: Create a Salesforce to database insert Map .....	21
Exercise 10: Map the source and destination fields .....	22
Exercise 11: Set default mapping values.....	24
Class Activity 2: Populate the CREATE_DATE field .....	25
Additional Challenge 1: Update the Salesforce query operation with contact information .....	26
Additional Challenge 2: Map the contact information .....	28
Exercise 12: Create a user-defined map function .....	29
Exercise 13: Test the process .....	34
Exercise 14: Add a Decision shape to query the database .....	35
Exercise 15: Test the decision lookup .....	39
Exercise 16: Create a database read process .....	40
Exercise 17: Set a database runtime parameter .....	46
Exercise 18: Test the process .....	47
Exercise 19: Create a Salesforce update action .....	48
Exercise 20: Create a Salesforce update map.....	50
Exercise 21: Create a mail operation.....	57
Exercise 22: Configure a message shape .....	59
Exercise 23: Test the mail connectivity.....	62
Additional Challenge 3: Format output to email .....	64
Exercise 24: Add a SQL command .....	65



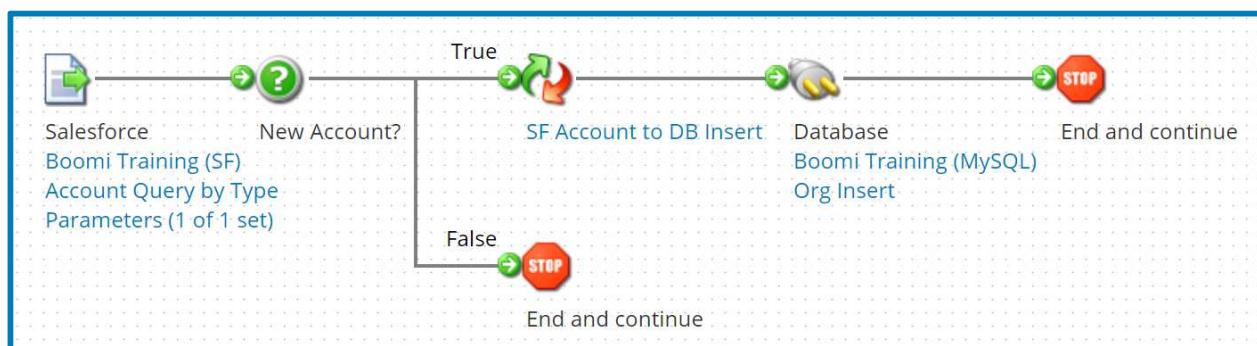
## Section 1 - SaaS Training

Welcome to Developer 1 - SaaS Training. For our cloud integration training, we will use Salesforce, a popular CRM cloud-based application, to create a new process to house the workflow for the Salesforce to Database integration.

In this session, we will build two separate processes that work together, covering:

- SaaS Integration Overview
- Salesforce Connector Review
- Database Connector Review
- Advanced Logic Concepts
- System Lookups and Best Practices
- Messaging
- Event Synchronization

Our first integration scenario calls for the company's sales team to enter prospect information into Salesforce.com. Operations wants to query Salesforce accounts, to send responses to their Organization Tracking system for lead management but prevent duplicate records from being created. Management, who monitors these accounts, wants to be notified daily of newly won customers.

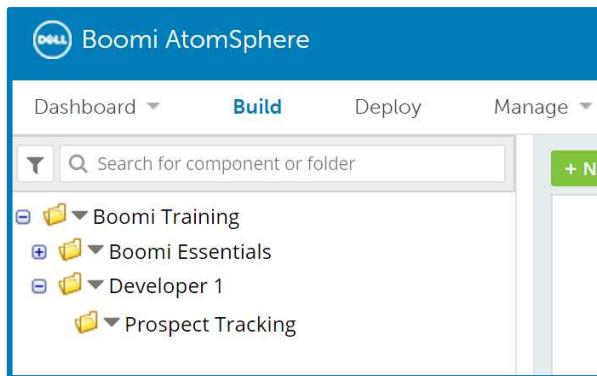
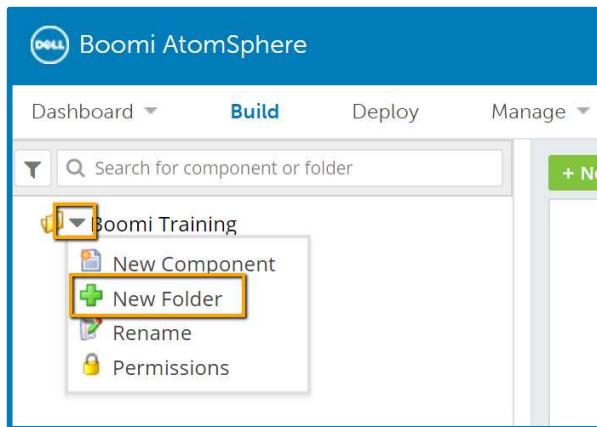


## Exercise 1: Set up folders

When developing a new integration project, organize your setup on the Build Tab of the Component Explorer by setting up folders to organize processes and components. This enables you to configure and store a unique Process containing the workflow and processing rules for your business scenario. We begin class by creating a new folder to store the processes we will use for our exercises.

### Create a folder and subfolder within your account

1. Click on the blue dropdown arrow next to the **Boomi Training** (root) folder.
2. Choose **New Folder** and name it **Developer 1**.
3. In the Developer 1 folder create a new folder and name it **Prospect Tracking**.



## Exercise 2: Download process endpoints from the Process Library

1. Open the Process Library by clicking on **Browse Process Library** at the bottom of the Component Explorer.

**Browse Process Library**

2. The Boomi Process Library lets you browse, view, and install published processes into your account. We will use the Process Library throughout this class to pull in processes for our activities. In the right column search bar enter the class ID ("Dev1") and press Enter.

The screenshot shows the Boomi Process Library interface. At the top right, there is a search bar containing "Dev1". Below it, a filter section titled "Filter by Publisher" includes checkboxes for "Dell Boomi" and "Education Services". The main content area displays a "Search Results" table with one row. The row contains the title "Prospect Tracking", a timestamp "Published on 03 Jun 2017 17:12:41 by Education Services", and two buttons: "View" and "Install".

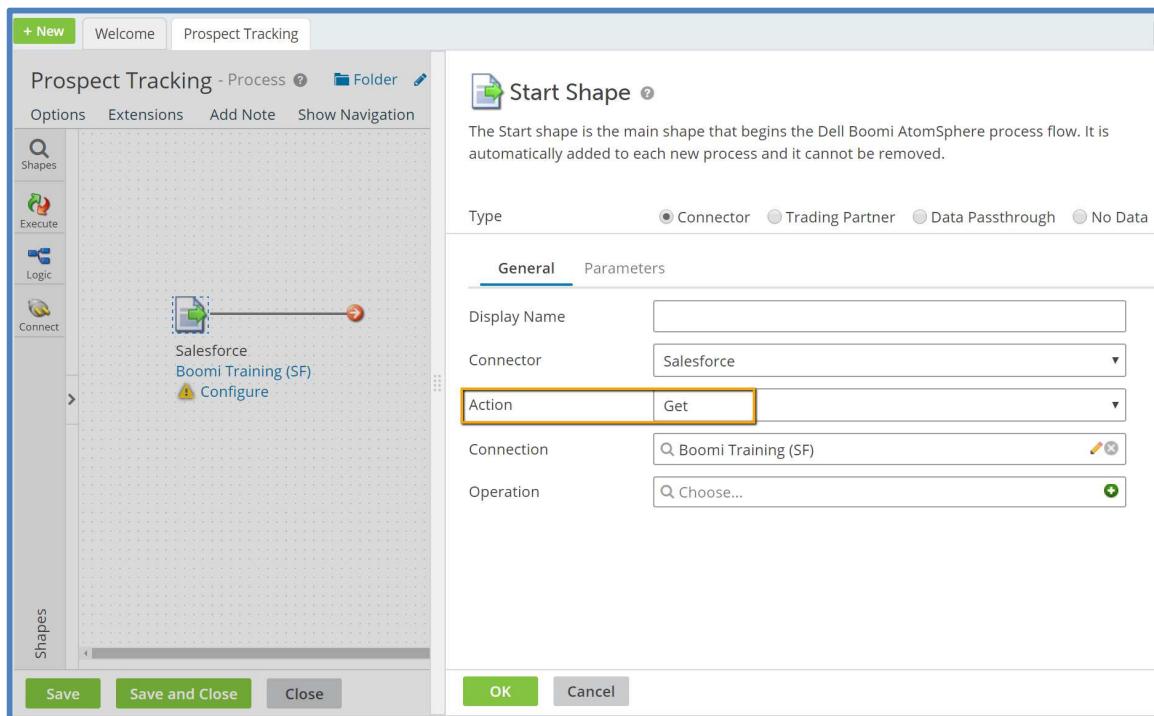
3. Click **Install** next to **Prospect Tracking**.
4. Click **Choose...** next to Select Installation Location and navigate to the Prospect Tracking folder.
5. Click **Install** in the lower right corner.
6. The process is now installed in your account. Click **View Process**.



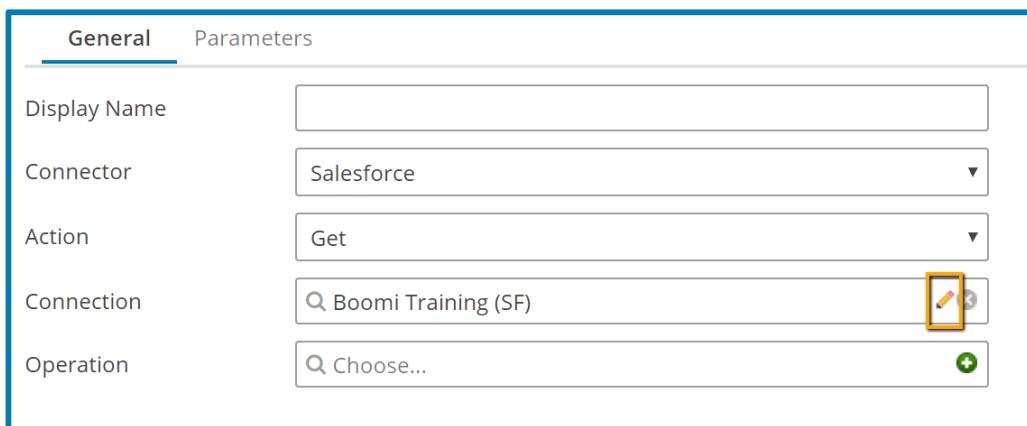
## Exercise 3: View the Salesforce read Connector

To retrieve account data, the Salesforce Connector is defined as a Read (Get) action in the Start shape. This defines a Connector containing the information to make a request to a specified Salesforce account.

1. Click on the **Start shape (Salesforce Connector)** to view the General settings. Notice the only Action available is Get, since this is a Start shape.



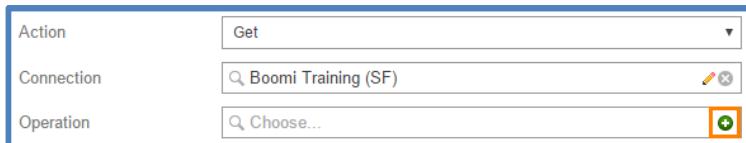
2. In the Connection field, **Boomi Training (SF)** has already been loaded. To view the Connection settings, click the **Edit** icon to the right of the Connection name.



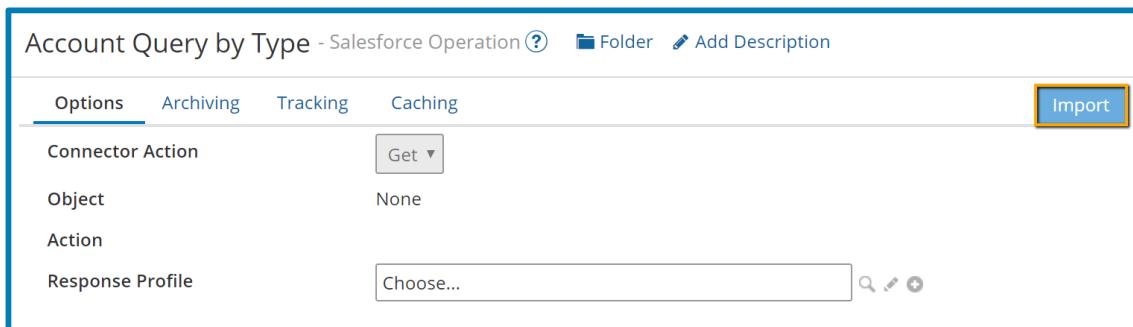
## Exercise 4: Create a Salesforce query Operation

When paired with the Connection component, the Salesforce Operation identifies the object data requested and determines how to limit the XML records returned to the process. In this exercise, queried Account and Contact information is returned based on a record's unique type.

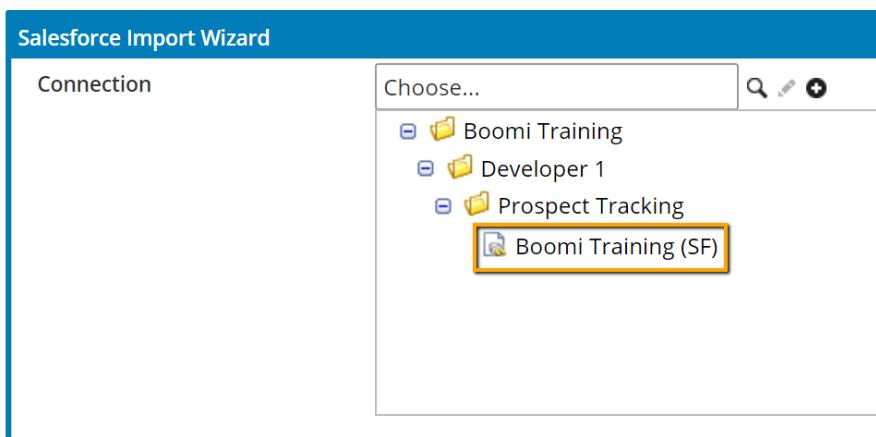
1. In the Start shape window next to Operation, click the **Create (+)** icon to open a new component tab.



2. In the Name window, enter **Account Query by Type**.
3. Click the **Import** button.



4. Click the **Browse** icon (magnifier) to open search. Choose the Salesforce Connection component in your Prospect Tracking folder.



5. Click **Next**.

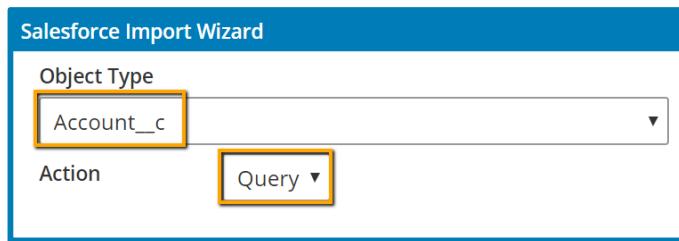
The Salesforce connection is established.

## Exercise 4: Create a Salesforce query Operation

6. Configure the following from the drop-down menus:

Object Type = **Account\_\_c**

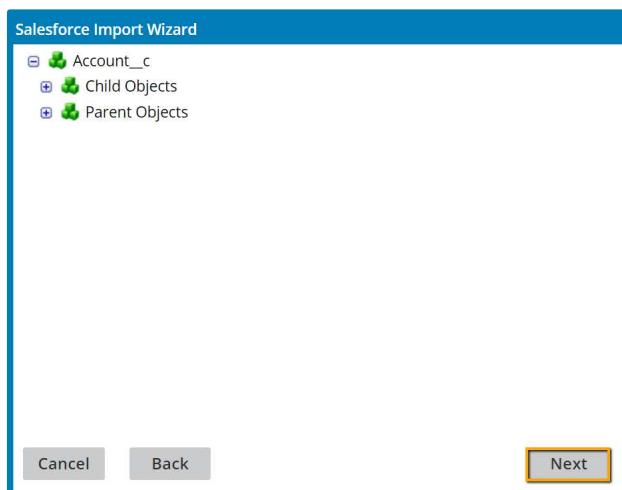
Action = **Query**



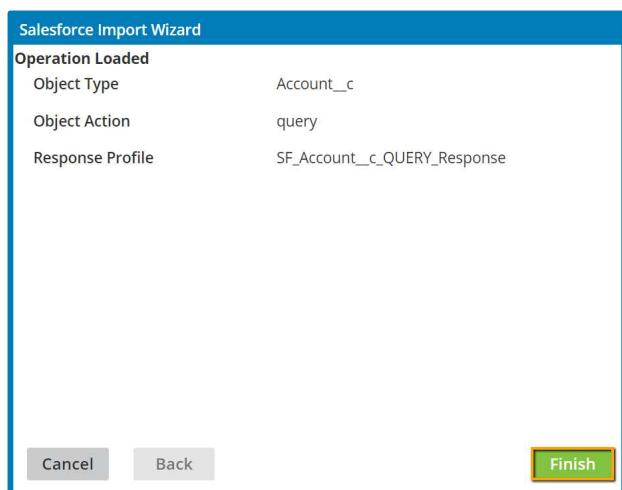
7. Click **Next**.

✓ *The Salesforce Import Wizard builds the object tree.*

8. Accept the defaults (do not select any child or parent objects). Click **Next**.

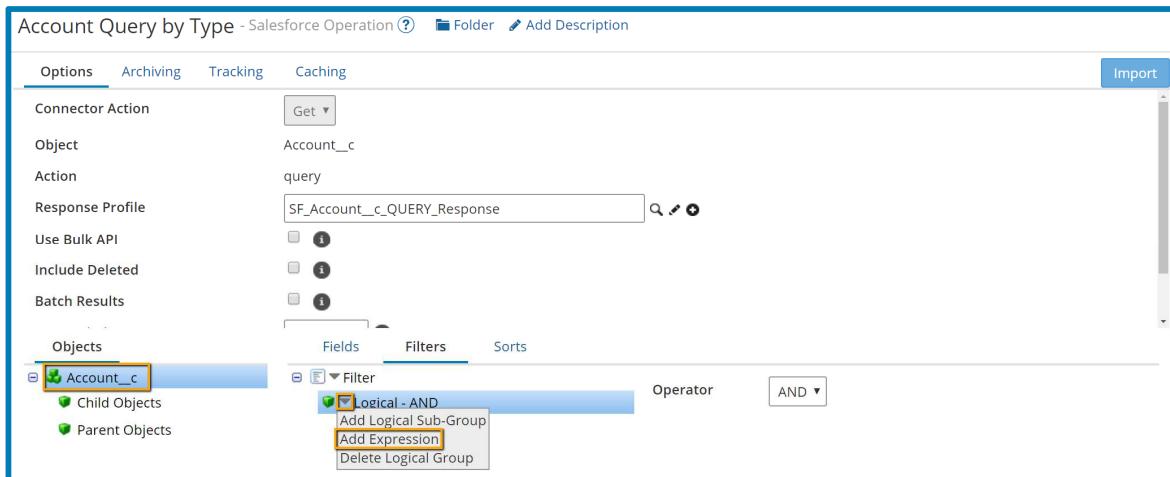


9. Review the results and then click **Finish**.



## Exercise 4: Create a Salesforce query Operation

10. At the bottom of the query window, under the **Objects** tab, highlight **Account\_\_c**.
11. Click the **Filters** tab, click the blue dropdown arrow next to **Logical - AND**, and then choose **Add Expression**.



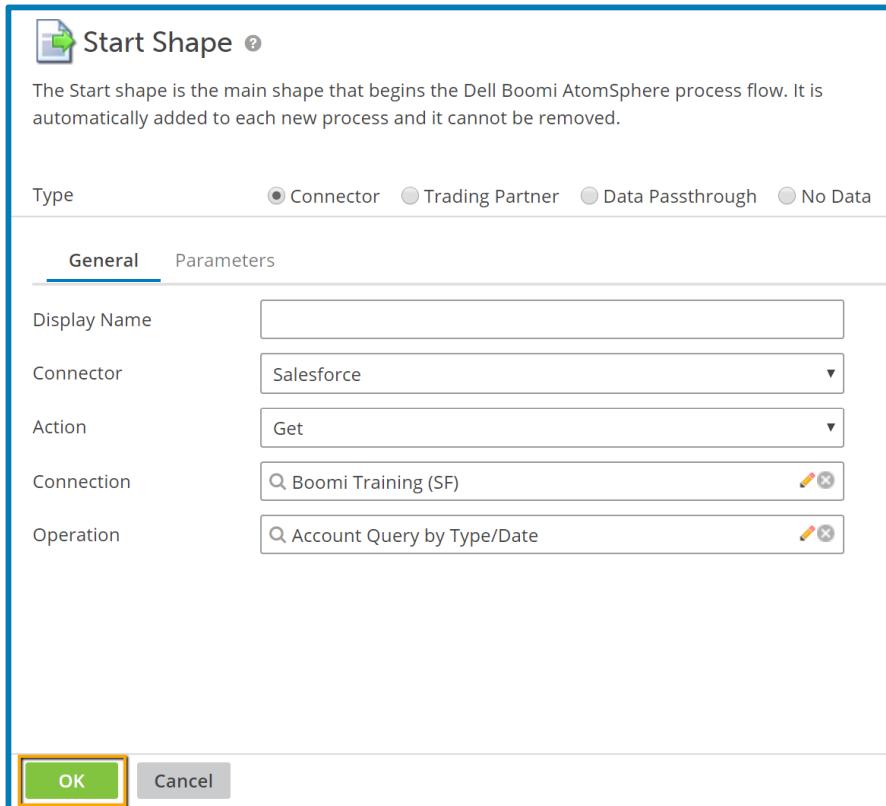
12. Highlight the new expression and define the following for the filter instance:

Filter Name	Type=
Field	Type__c
Operator	Equal To



13. Click **Save and Close** to return to the process. The Operation is now loaded into the Start shape.
14. Click **OK** to accept the configured Start shape.

## Exercise 4: Create a Salesforce query Operation

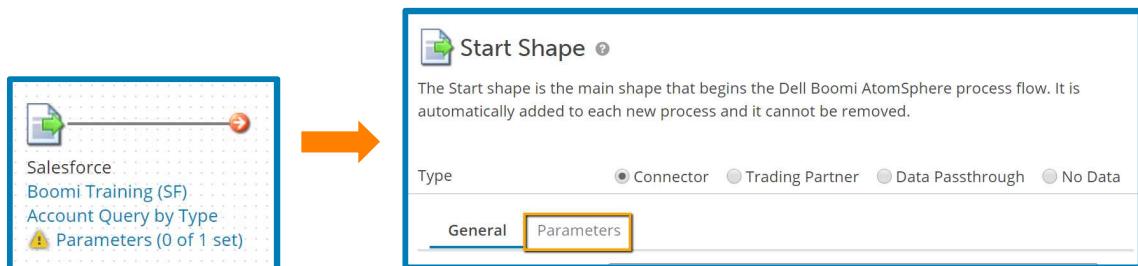


15. Click **Save** to save the current version of the process. "Prospect Tracking saved successfully" displays in the lower right corner of the Process Canvas.

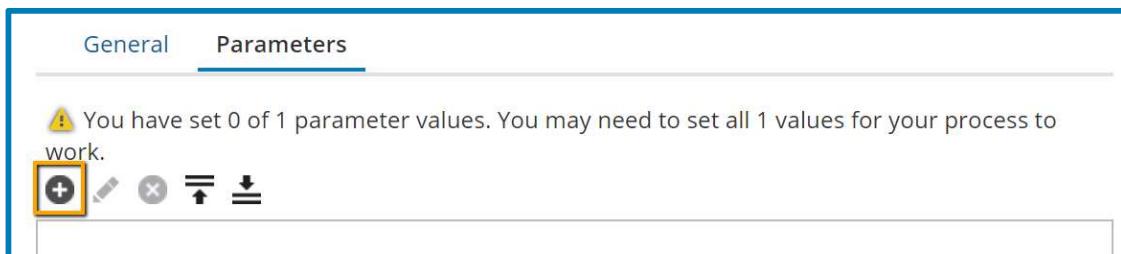
## Exercise 5: Set the Salesforce runtime parameter

To populate the operation filter for your Salesforce Account request, set a static input parameter on the Start shape. This input ensures the account response XML documents are a type Prospect defined by the Salesforce business user.

1. Open the **Start** shape and then click the **Parameters** tab.



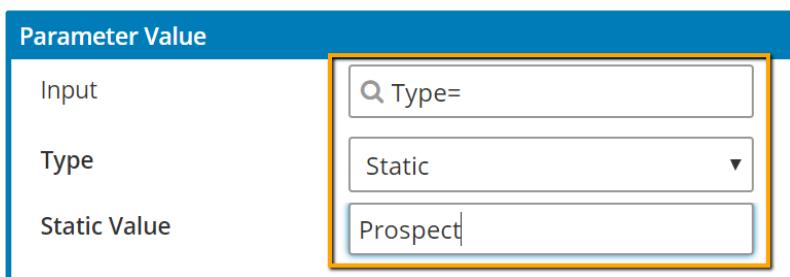
2. Click **Add Parameter (+)**.



3. In the Parameter Value window, define the following:

Input	Type=
Type	Static
Static Value	Prospect

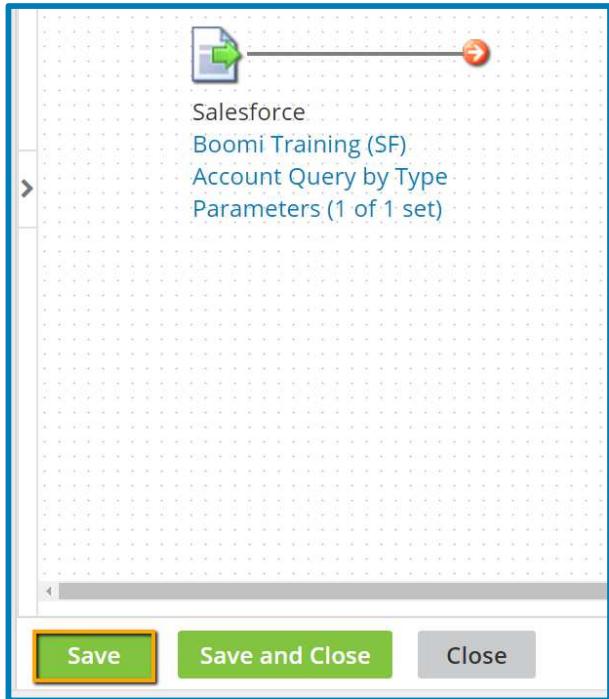
- ✓ Be sure to spell the static value correctly. Stored values on the destination data system must match identically.



4. Click **OK** to return to the Parameters tab. The created parameter is now displayed.

## Exercise 5: Set the Salesforce runtime parameter

5. Click **OK** to return to the process.
6. **Save** the process.



## Exercise 6: Test the process

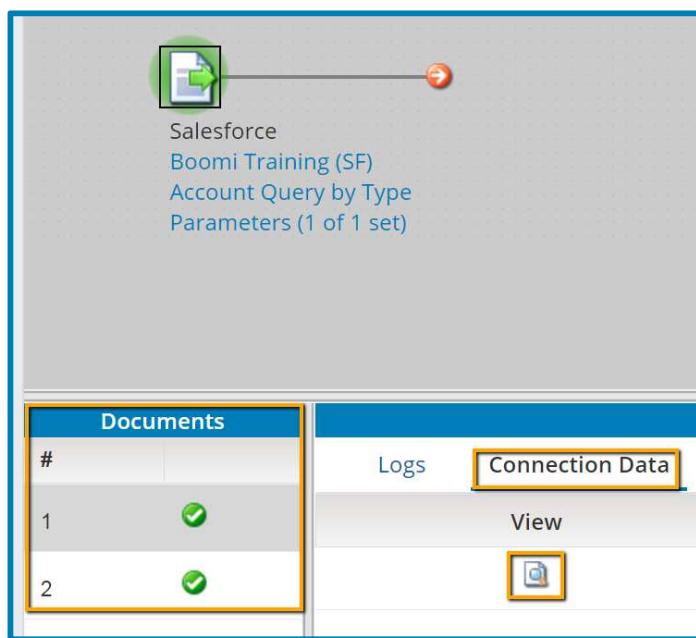
Before continuing, check if the two Salesforce records are successfully picked up. It is a best practice to continually test your process at various stages throughout development.

1. From the dropdown arrow, choose **Test Atom Cloud**, and then click **Run Test**.



2. Note the two documents retrieved and highlight each document individually.

3. Click the **Connection Data** tab and select the icon to view the document contents.



Notice there is a separate entry for each document that is retrieved. To view the contents of the second document, click on it in the Documents pane.

## Exercise 6: Test the process

Document Viewer

```
1 <Account_c>
2   <Id>a6f0q0000004CgqAAE</Id>
3   <OwnerId>00540000003O16YAAS</OwnerId>
4   <IsDeleted>false</IsDeleted>
5   <Name>GenePoint</Name>
6   <CurrencyIsoCode>USD</CurrencyIsoCode>
7   <CreatedDate>2017-06-16T16:22:16.000Z</CreatedDate>
8   <CreatedBy>00540000003O16YAAS</CreatedBy>
9   <LastModifiedDate>2017-10-11T17:50:47.000Z</LastModifiedDate>
10  <LastModifiedBy>00540000003O16YAAS</LastModifiedBy>
11  <SystemModstamp>2017-10-11T17:50:47.000Z</SystemModstamp>
12  <LastViewedDate>2017-10-08T18:31:33.000Z</LastViewedDate>
13  <LastReferencedDate>2017-10-08T18:31:33.000Z</LastReferencedDate>
14  <ConnectionReceivedId></ConnectionReceivedId>
15  <ConnectionSentId></ConnectionSentId>
16  <Account_Number_c>CC978213</Account_Number_c>
17  <Description_del_c>Genomics company engaged in mapping and sequencing of the human genome
18  <Phone_c>(650) 867-3450</Phone_c>
19  <Rating_c>Cold</Rating_c>
20  <Type_c>Prospect</Type_c>
21  <Website_c>http://www.genepoint.com</Website_c>
22  <Annual_Revenue_c>3.0E7</Annual_Revenue_c>
23  <Account_Currency_c>U.S.Dollar</Account_Currency_c>
24  <Account_Site_c></Account_Site_c>
25  <Industry_c>Biotechnology</Industry_c>
26  <Billing_Address_c>345 Shoreline Park</Billing_Address_c>
27  <SI_A_Expiration Date_c>2017-10-06</SI_A_Expiration Date_c>
```

Formatted view is on. Formatted view hides some data for easier reading.

Size 4008 bytes

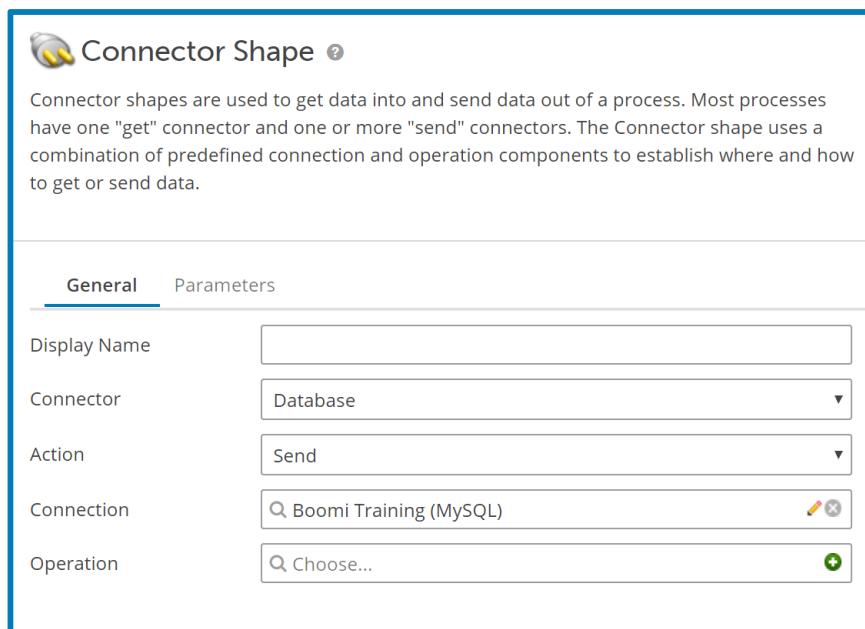
[Close Document Viewer](#)



## Exercise 7: View the database write Connector

It is a best practice to design and build the workflow of a new process from the outside in. The endpoint Connectors were preloaded in the process you installed from the Process Library. The database write connector defines the different ways requests can be sent to database tables and/or stored procedures. Connection components can be reused to create database queries and SQL commands.

1. Return to Edit Mode and click on the **Database Connector** to view the General tab. Notice the Action is set to Send (Write).



2. In the Connection field, **Boomi Training (MySQL)** has already been loaded. To view the Connection settings, click the **Edit** icon to the right of the Connection name.

## Exercise 8: Create a database write Operation

The database write operation identifies how to commit the SQL transactions and houses the database field set(s) which are structured in the profile component, resulting in a database profile to be reused in the Map shape.

1. Next to Operation in the Database Connector, click **Create (+)** to open a new tab.

The screenshot shows the 'General' tab of a configuration dialog. It includes fields for Label (set to 'Optional'), Connector (set to 'Database'), Action (set to 'Send'), Connection (set to 'Boomi Training (MySQL)'), and Operation (with a '+' icon). The 'Operation' field has a yellow box around its '+' icon.

2. In the Name window, enter **Org Insert**.
3. In the Commit Options section, accept the default values.

The screenshot shows the 'Org Insert' database operation configuration. It includes tabs for Options, Archiving, Tracking, and Caching. Under Options, the Connector Action is set to 'Send'. The Profile field has a 'Choose...' button. A yellow box highlights the 'Commit Options' section, which contains fields for Commit Option (set to 'Commit By Profile') and Batch Count (set to '0').

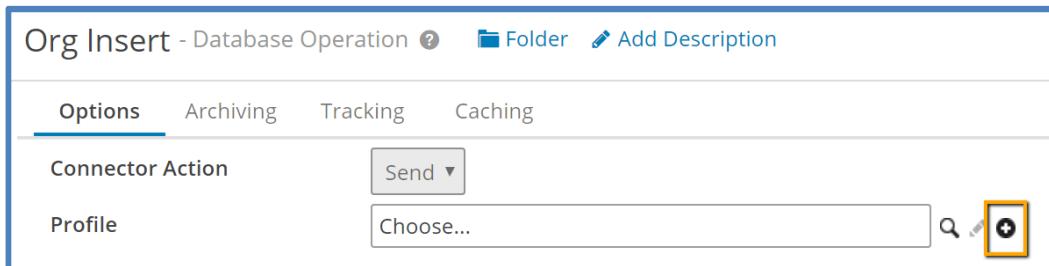
4. Click **Save**.



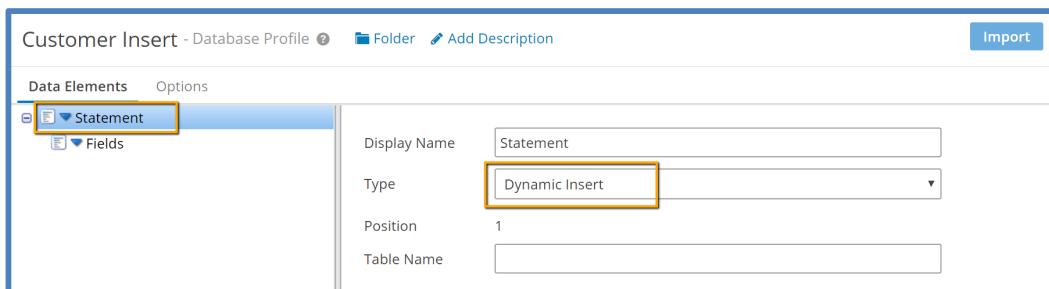
## Exercise 9: Create a database write Profile

Use a Dynamic Insert to define the field set for the customer table to poll the database server and auto-list the table's field names and types. The quickest way to dynamically build the SQL based insert from the XML records mapped from Salesforce is to use Dynamic Insert.

- Under the Options tab, next to Profile, click (+) **Create a new component** to create a new Profile.

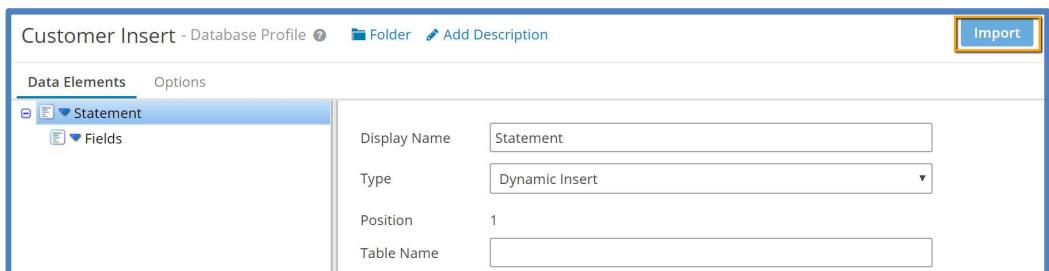


- In the new tab, enter **Customer Insert** in the Name window.
- Under the Data Elements tab, click **Statement**.
- In the dropdown window next to **Type**, select **Dynamic Insert**.



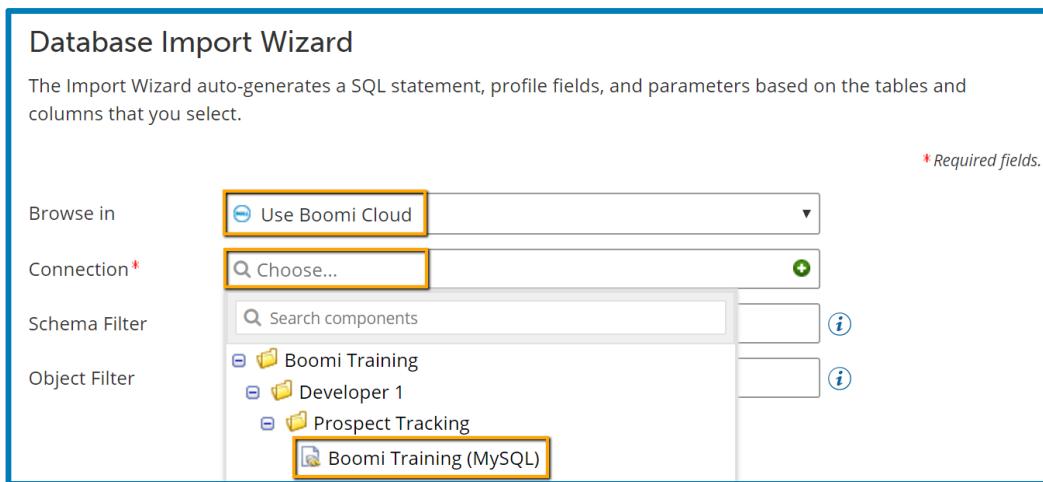
*The various database types include: Standard Insert / Update / Delete which allow you to enter the SQL statements. Dynamic Insert, Dynamic Update, and Dynamic Delete allow you to dynamically build the SQL statement. Stored Procedure Write allows you to enter the name of the stored procedure to be executed.*

- Click **Import**.



## Exercise 9: Create a database write Profile

6. In the Database Import Wizard window, Browse in the **Boomi Cloud**.
7. Choose the **Boomi Training (MySQL)** Connection.

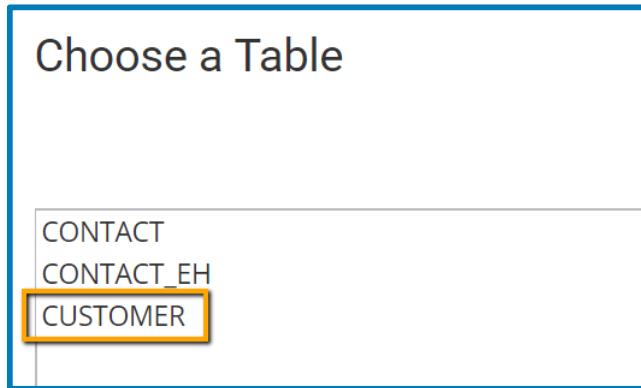


8. Click **Next**.
- ✓ If the *Import Wizard* hangs on “Connecting to Atom....” click on the back button to re-launch the *Import Wizard*.



When a connection to the database is made, the tables load into the Database Import Wizard window, however, the messages “Loading database tables....” and “Connecting to Atom....” open regardless of whether there is a connection.

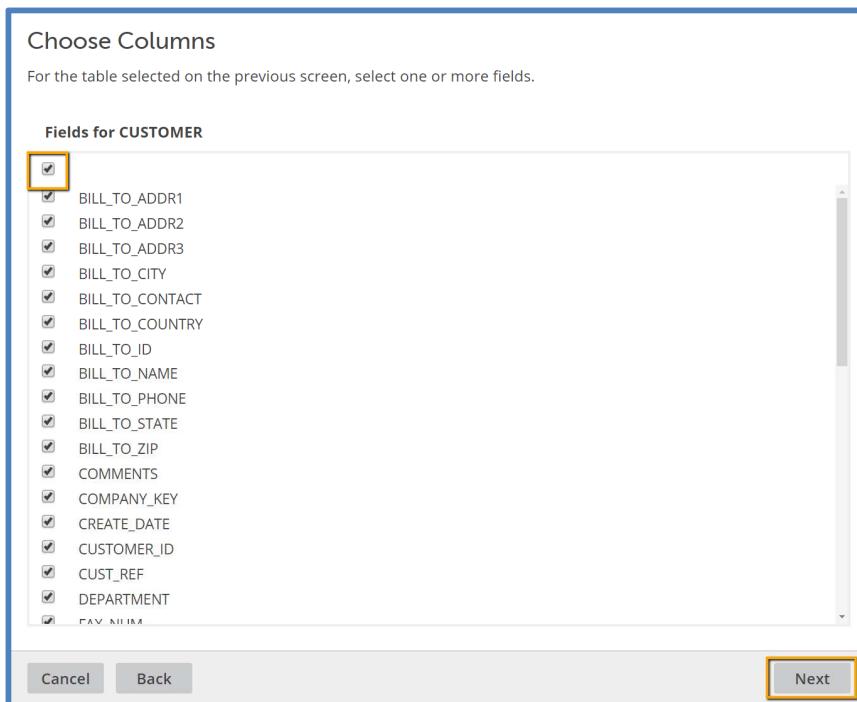
9. Under **Choose a Table**, highlight the **CUSTOMER** table then click **Next**.



10. Under **Choose Columns**, check the “Select All” check box, click **Next**, and then click **Finish**.



## Exercise 9: Create a database write Profile



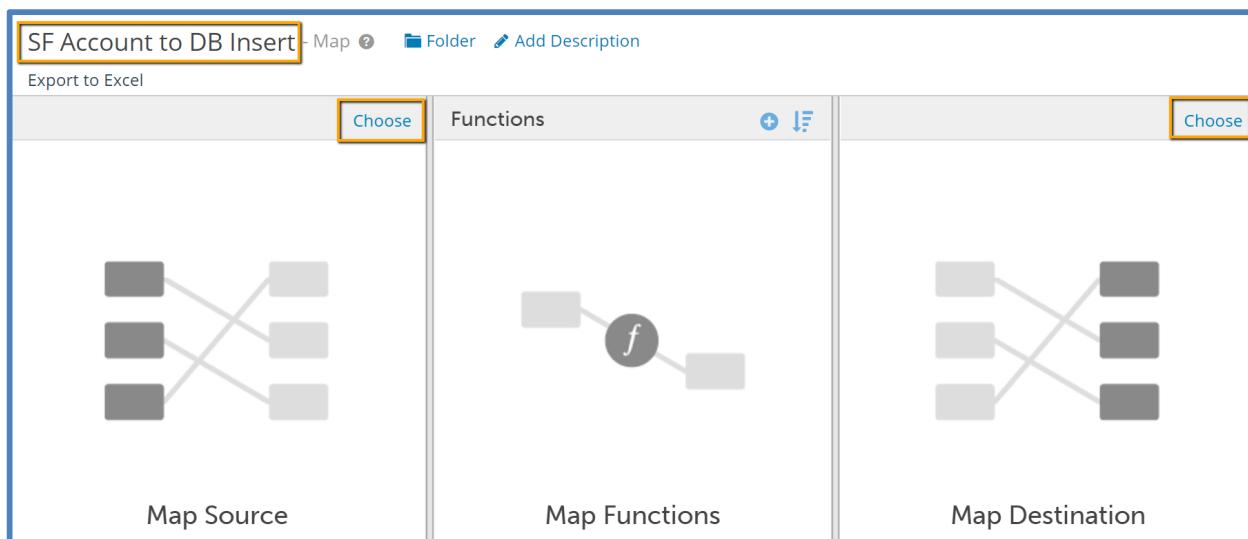
11. Click **Save and Close** to exit the Database Profile and return to the Operation tab.
12. Click **Save and Close** to exit the Database Operation and return to the Process Canvas.
13. Click **Save** to save the process.



## Class Activity 1: Create a Salesforce to database insert Map

Although you constructed both the inbound Salesforce and outbound database document structures, you must bridge the gap by adding the Map component to connect the proper fields. For this activity, create a Salesforce to database Map, keeping in mind we are only loading the profiles into the map and are not yet mapping the fields.

1. Drag and drop a Map shape onto the Process Canvas.
2. Create a new mapping component and name this map **SF Account to DB Insert**.
3. Load both the source and destination profiles into the map.



Both the source (XML) profile and destination (DB) profile (with Dynamic Inserts) have been created using the Import feature. It is important to select the profiles created in your Salesforce and Database Connector Operations.

## Exercise 10: Map the source and destination fields

With the source and destination profiles loaded in the Map, it is time to link the corresponding fields.

- Map the following fields from **Account** (source) to **Customer** columns (destination).

Salesforce Account SOURCE	DB Customer Insert DESTINATION
Name	BILL_TO_NAME
Name	NAME
Name	SHIP_TO_NAME
Phone_c	BILL_TO_PHONE
Phone_c	PHONE
Phone_c	SHIP_TO_PHONE
Website_c	URL
Industry_c	INDUSTRY
Billing_Address_c	BILL_TO_ADDR1
Fax_c	FAX_NUM
Shipping_Address_c	SHIP_TO_ADDR1
Billing_City_c	BILL_TO_CITY
Billing_State_c	BILL_TO_STATE
ZipCode_c	BILL_TO_ZIP
Billing_Address_Country_c	BILL_TO_COUNTRY
Shipping_Address_City_c	SHIP_TO_CITY
Shipping_Address_State_c	SHIP_TO_STATE
Shipping_Address_Country_c	SHIP_TO_COUNTRY
Shipping_Address_Zipcode_c	SHIP_TO_ZIP

- If using Boomi Suggest, accept only the **19 mappings** listed in the table. Boomi Suggest lists the fields alphabetically according to the DESTINATION profile, so it may be helpful to use the following table.

Salesforce Account SOURCE	DB Customer Insert DESTINATION
Billing_Address_c	BILL_TO_ADDR1
Billing_City_c	BILL_TO_CITY
Billing_Address_Country_c	BILL_TO_COUNTRY
Name	BILL_TO_NAME
Phone_c	BILL_TO_PHONE
Billing_State_c	BILL_TO_STATE
ZipCode_c	BILL_TO_ZIP
Fax_c	FAX_NUM
Industry_c	INDUSTRY
Name	NAME
Phone_c	PHONE
Shipping_Address_c	SHIP_TO_ADDR1
Shipping_Address_City_c	SHIP_TO_CITY
Shipping_Address_Country_c	SHIP_TO_COUNTRY



Exercise 10: Map the source and destination fields

Name	SHIP_TO_NAME
Phone_c	SHIP_TO_PHONE
Shipping_Address_State_c	SHIP_TO_STATE
Shipping_Address_Zipcode_c	SHIP_TO_ZIP
Website_c	URL

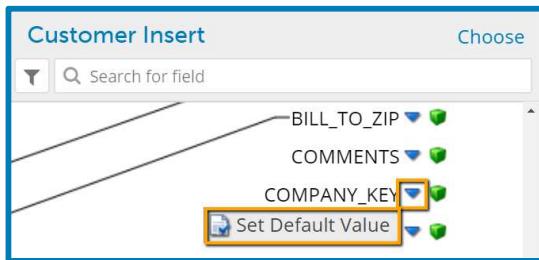
2. Click **Save and Close** and then click **OK** to return to the Process Canvas.
3. Click **Save** to save the process.



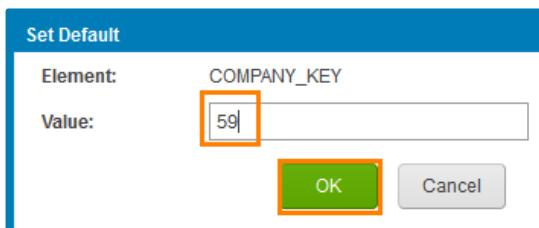
## Exercise 11: Set default mapping values

Sometimes the source profile does not contain all of the required information, so hard-coded values need to be applied to some destination fields.

1. Open the Map “SF Account to DB Insert.”
2. Click the blue dropdown arrow to the right of the **COMPANY\_KEY** field in the destination window, then choose **Set Default Value**.



3. In the **Value** field, enter **59** and then click **OK**.



The default value is appended to the element in the list.



4. Repeat steps 1-3 above for the following three fields to set their default values:

DEPARTMENT	IT
STATUS	0
USER_ID	***Enter Your Name***

The default values should now be appended to the elements in the list.

5. Click **Save and Close** to save the Map.
6. Click **Save** to save the process.



## Class Activity 2: Populate the CREATE\_DATE field

The database requires the CREATE\_DATE field be populated with the current system date/time. A map function is used to supply this data for each account record.



## Additional Challenge 1: Update the Salesforce query operation with contact information

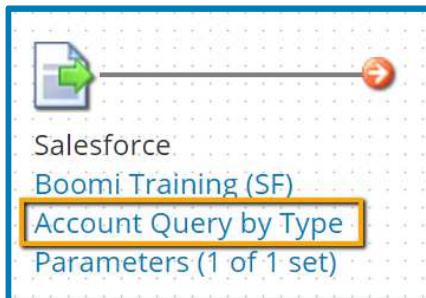
In mapping the account fields to the customer columns, you realize the main Contact's Name and Email address cannot be sourced from the base account object. Contact Name refers to the main contact of the account record which could be a sales rep, an account manager, etc. There are two options to handle this:

- Add a connector call map function to the map to query the Salesforce contact object on-the-fly, or
- Update the original account query operation to include the parent contact fields.

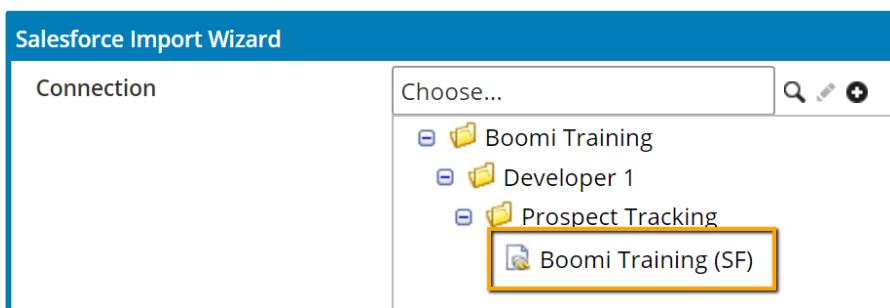
The second option is better because it exposes all required fields up front and prevents an extra API request for each account record, which is required for the Connector Call. For all Connector types, it is best to append any additional objects to the profile by using the import feature again.

**Refresh the account query operation and the source XML profile with the new contact fields:**

1. From the Process Canvas, click **Account Query by Type** to open the Operation component.



2. Click **Import**.
3. In the Salesforce Import Wizard, choose the **Boomi Training (SF)** Salesforce connection component in your account.

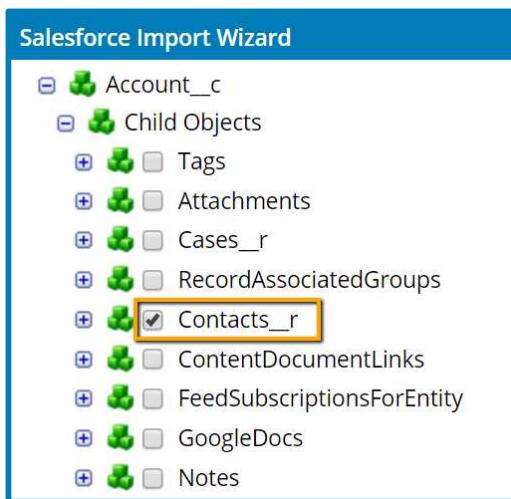


4. Click **Next**. The Salesforce connection is established.



## Additional Challenge 1: Update the Salesforce query operation with contact information

5. Configure the following from the drop-down menus:  
Object Type = **Account\_c**  
Action = **Query**
6. Click **Next**.
7. Expand the **Account\_c > Child Objects** section and check **Contacts\_\_r**.



- ✓ No child object was imported earlier, in the previous import exercise. However, anytime you select a child or parent node in an import, you **must re-select the node(s)** to maintain the fields in any operations or mappings previously built.

8. Click **Next**.
9. Review the results and then click **Finish**.

- ✓ The **Contacts\_\_r** child object has been added to the XML Profile.



10. Click **Save and Close** to return to the Process Canvas.



Additional Challenge 2: Map the contact information

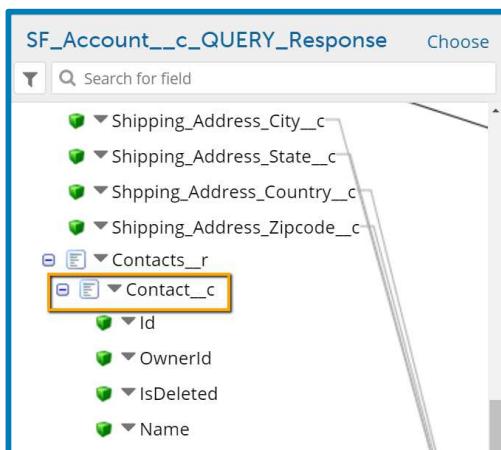
## Additional Challenge 2: Map the contact information

The field listing in the Salesforce Profile component in the Operation shows the changes in the data Map. Expand the Contact\_\_c field set and then map the Contact Name and Contact Email to the corresponding fields.

1. Open the **SF Account to DB Insert** map from the Process Canvas.



2. Scroll to the bottom of the source Profile window and expand the **Contact\_\_c** section.



3. Map the **Contact > Name** field in the Salesforce source profile to the **CONTACT\_NAME** field in the database destination profile.



4. Map the **Contact > Email\_\_c** field in the Salesforce source profile to the **CONTACT\_EMAIL** field in the database destination profile.



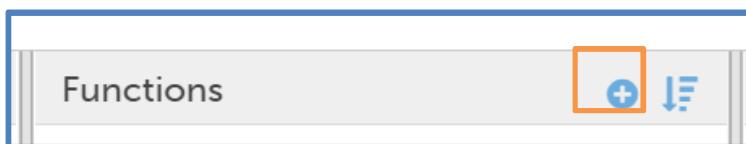
5. Click **Save**.



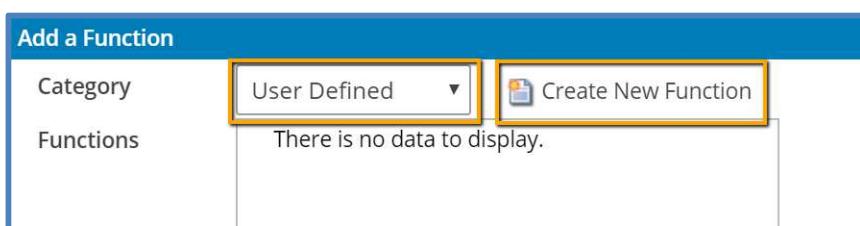
## Exercise 12: Create a user-defined map function

The database insert requires the **CUSTOMER\_ID** field to be populated with a unique alphanumeric key for each customer record. For our training purposes, we assume the Customer Name and Phone Number do not change, and we can concatenate these values to create a unique key. However, this requires more than one map function to create a valid ID. In this exercise, we will create a re-usable User Defined Map Function to string together multiple function steps resulting in the proper ID field.

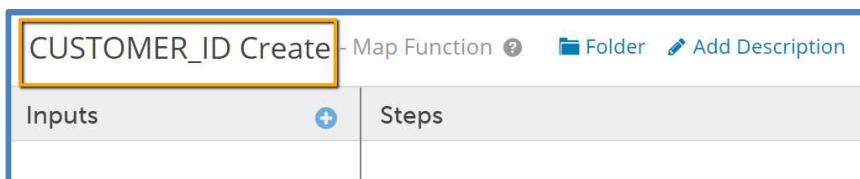
1. Open the **SF Account to DB Insert** map.
2. In the **Functions** column, click the **(+)** Add icon.



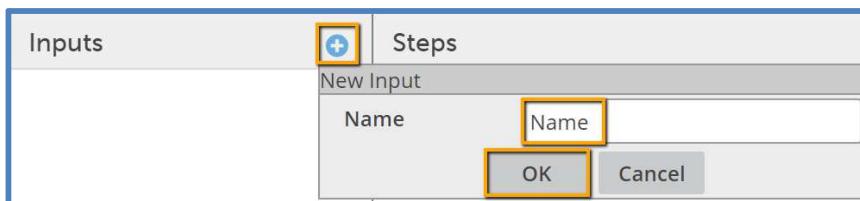
3. From the Category dropdown, choose **User Defined**, and then click **Create New Function** at the top of the window.



4. On the new tab, type the name of the map function: **CUSTOMER\_ID Create**.

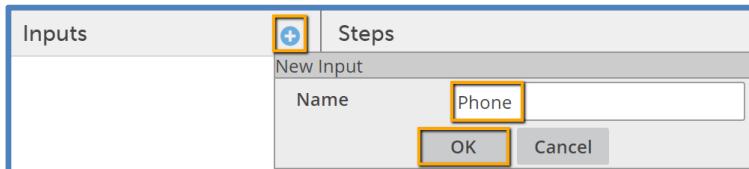


- ✓ Always provide a descriptive name for all user defined map functions.
- 5. In the left Inputs column, click **Add (+)**. For **Input Name**, enter **Name** and then click **OK**.

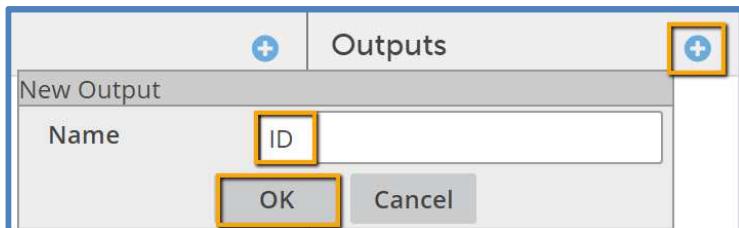


## Exercise 12: Create a user-defined map function

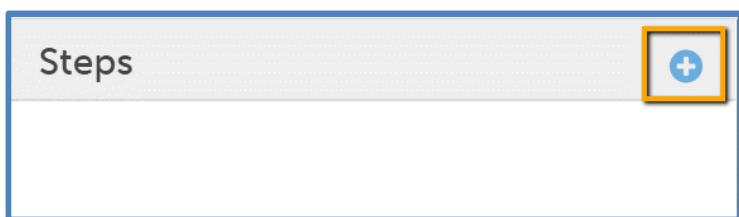
6. Repeat step 5 to add another input. For **Input Name**, enter **Phone** and then click **OK**.



7. In the right Outputs column, click **Add (+)**. For **Output Name**, enter **ID** and then click **OK**.

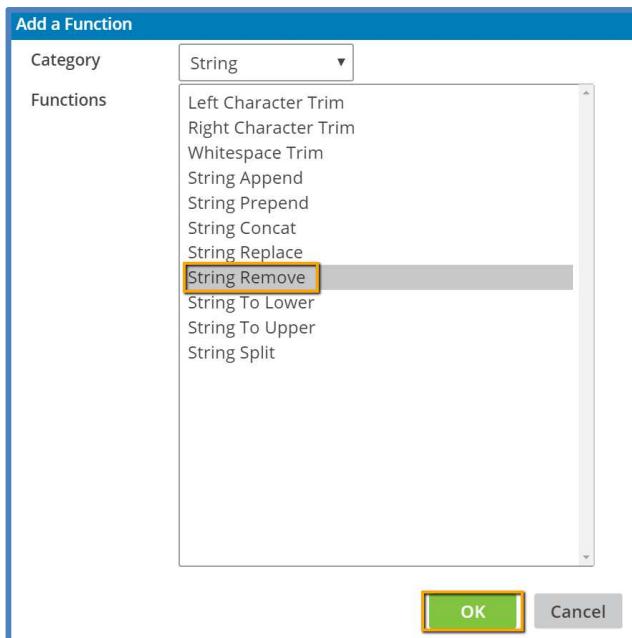


8. In the Steps column click **Add (+)**.



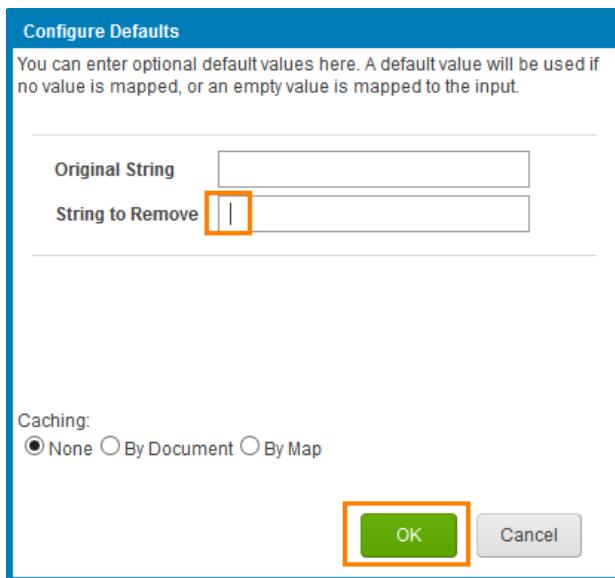
9. From the dropdown arrow next to **Category**, choose **String**.

10. Highlight **String Remove** and click **OK**. The Configure Defaults window opens.



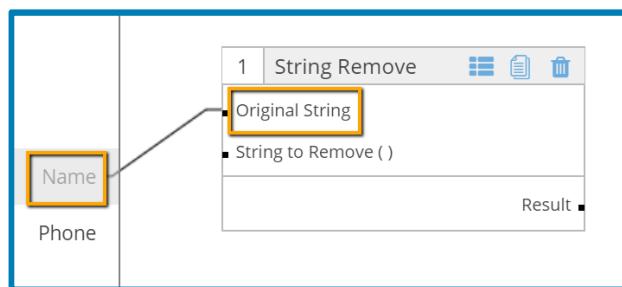
## Exercise 12: Create a user-defined map function

11. Leave the **Original String** field blank. In the **String to Remove** field, type a <space> and click **OK**.



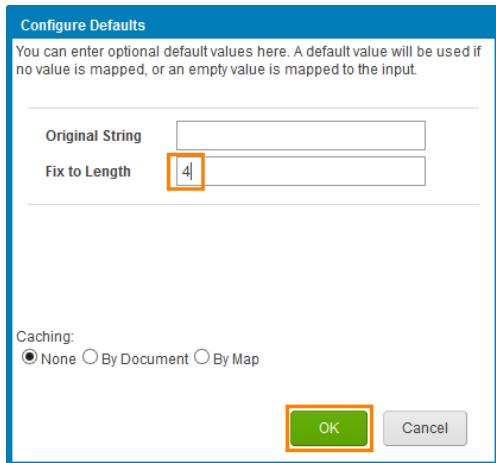
- ✓ Be sure to enter a blank space in the **String to Remove** field. This ensures the name portion of the key generation will consist of one string by stripping away any blank spaces making up the name.

12. Click and drag **Name** to connect to the **1 String Remove > Original String** input.



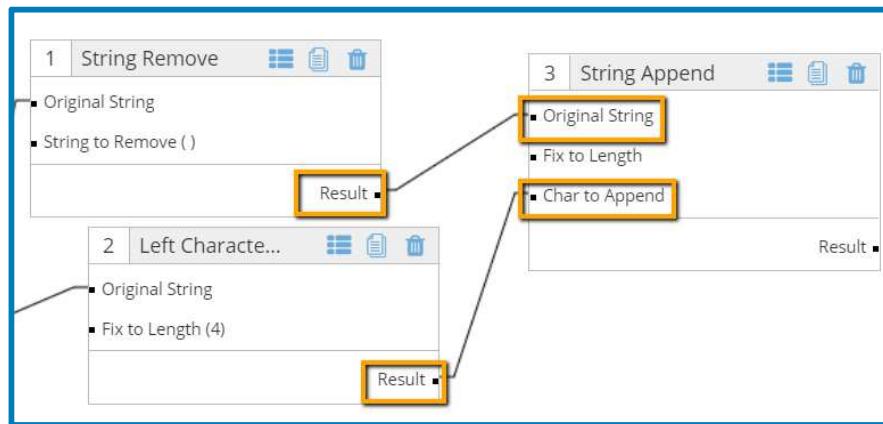
13. In the middle Steps window, click **Add (+)** to add another step to the user-defined function.
14. From the dropdown, choose **String**.
15. Highlight **Left Character Trim** and then click **OK**. The Configure Defaults window opens.
16. Leave the **Original String** field blank. In the **Fix to Length** field, enter **4** and then click **OK**.

## Exercise 12: Create a user-defined map function



**The Left Character Trim** function removes characters from the left side of the string so the remaining string is fixed to a specified length. For example, applying a Left Character Trim, Fixed to Length of 6, trims the string AtomSphere to Sphere. We want to use the last four digits of the telephone number as part of the unique ID, so use the **Left Character Trim** function to strip characters (from left to right) until the string length is set to 4.

17. Click and drag from the main **Phone** input to connect to the **2 Left Character Trim > Original String** input.
18. In the middle Steps window, click **Add (+)** to add another step to the user-defined function.
19. From the dropdown, choose **String**.
20. Highlight **String Append** and click **OK**. The Configure Defaults window displays.
21. Leave all fields blank and click **OK**.
22. Click and drag from the **1 String Remove > Result** output to connect to the **3 String Append > Original String** input.
23. Click and drag from the **2 Left Character Trim > Result** output to connect to the **3 String Append > Char to Append** input.



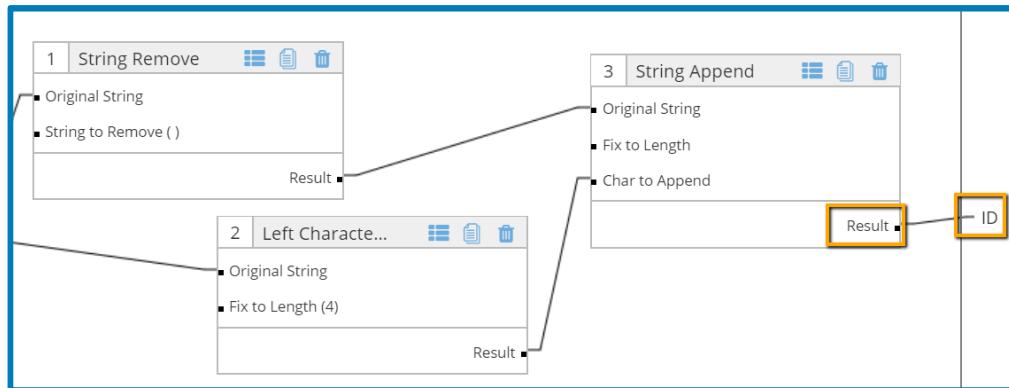
## Exercise 12: Create a user-defined map function

- ✓ To review, within the Account Name remove all white space separators, and append the last four digits of the phone number to it to create a unique CUSTOMER\_ID.



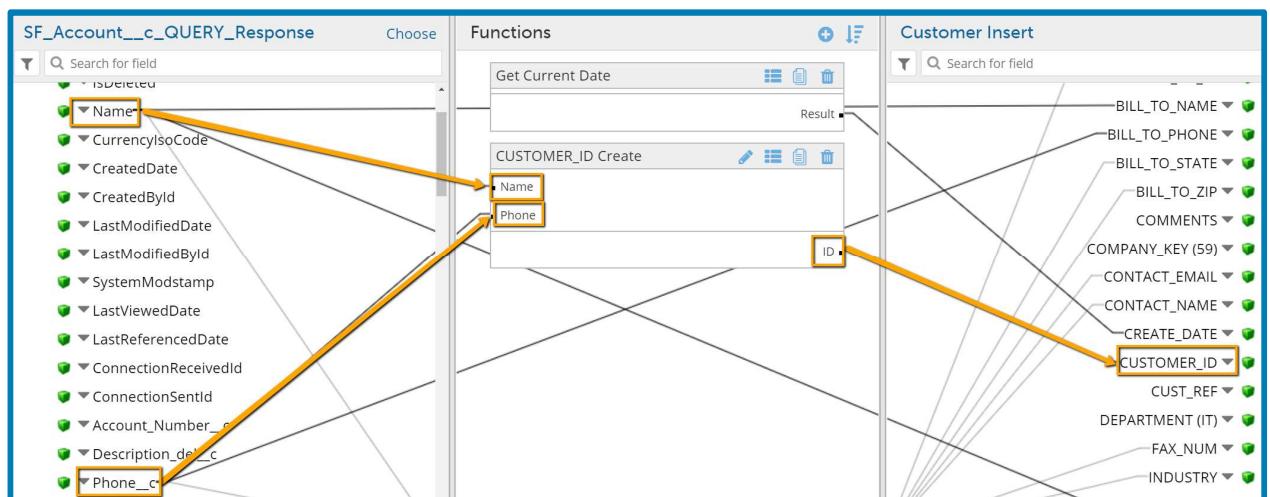
Observe the changes on the numbering when the function shapes are moved on the canvas.

24. Click and drag from the 3) String Append >> Result output to connect to the main ID Output.



25. Save the CUSTOMER\_ID Create function by clicking **Save and Close**.

26. In the map window, connect the **Account> Name and Account> Phone\_c** source XML fields to the corresponding inputs in the CUSTOMER\_ID Create function. Then connect the output ID to the destination database profile element CUSTOMER\_ID.



27. Save the SF Account to DB Insert Map by clicking **Save and Close**.



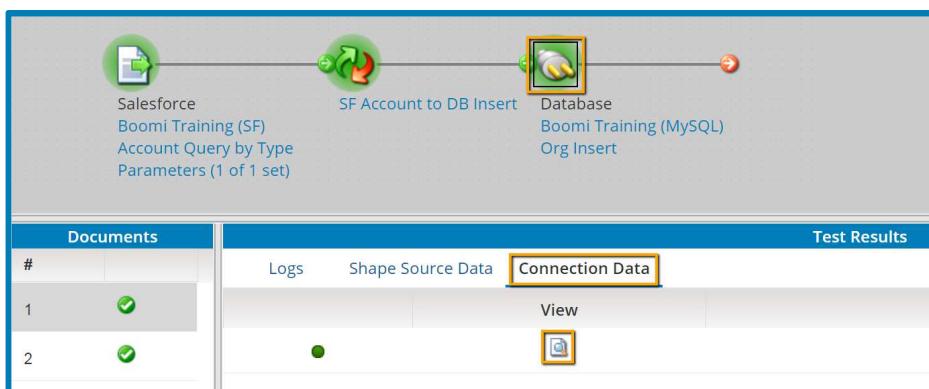
## Exercise 13: Test the process

The necessary components to complete the main workflow are now created, so link the components together on the canvas and execute the process in Test Mode to review if the documents were successfully committed to the database.

1. Connect the Start shape to the Map and the Map to the Database. **Save** the process.



2. Execute the process in Test Mode by selecting **Run Test** and choosing the **Test Atom Cloud**.
3. Click on the Database Connector, then select the **Connection Data** tab and click **View the contents**.



The screenshot shows the 'Document Viewer' window displaying the raw XML output of the database insertions. The XML content includes various place names and their coordinates, along with database connection details and timestamps.

```
DBSTART|d271b5aa-156f-4d60-b08a-7d3ca5b18cef|2|@|BEGIN|2|@|OUT_START|3|@|312
Constitution Place|^|_||^|_|Austin|^|_|^USA|^|_|^Edge Communications|^|(512) 757-
6000|^|TX|^|78767.0|^|||^59|^|_||^|20170619
193324.162|^|EdgeCommunications6000|^|_|IT|^|(512) 757-9000|^|Electronics|^|_|^Edge
Communications|^|_||^|(512) 757-6000|^|||^312 Constitution
Place|^|_||^|Austin|^|_|^USA|^|_|^Edge Communications|^|(512) 757-
6000|^|TX|^|78767|^|||^0|^|_|www.edgecomm.com|^|BoomiTrainer|^|_|#||#|OUT_END
|3|@|END|2|@|DBEND|d271b5aa-156f-4d60-b08a-7d3ca5b18cef|2|@|
```

4. After viewing the results in the Document Viewer, click **Close Document Viewer** and return to edit mode.

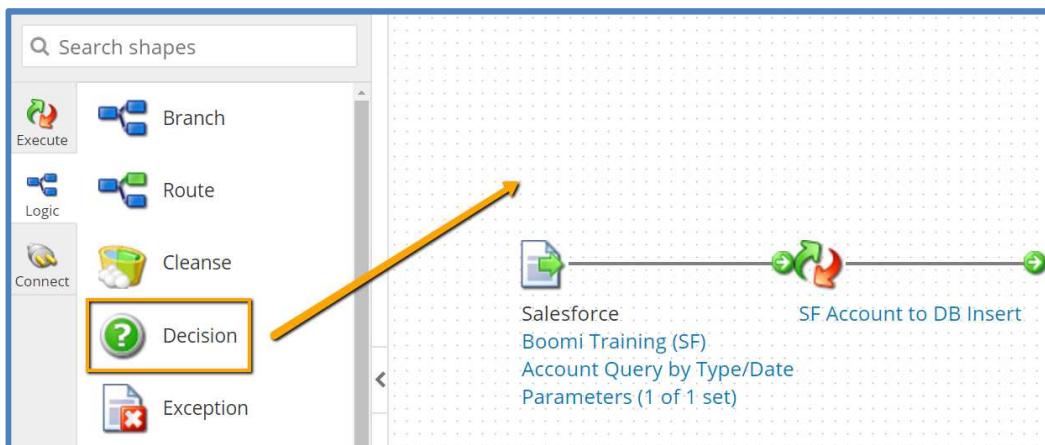


## Exercise 14: Add a Decision shape to query the database

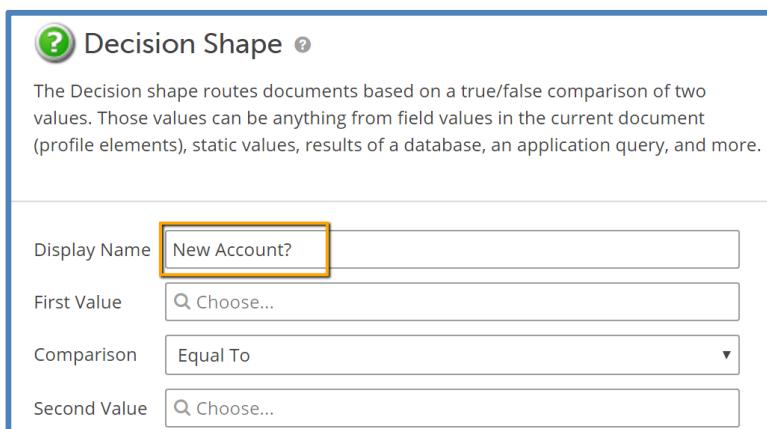
Many databases produce an error when attempting to enter duplicate records. Most AtomSphere integrations can programmatically prevent duplicating records and/or failures against the destination system. The Decision shape is a simple way to prevent duplication as it compares values within a document. The result of the comparison forces the document down a true-or-false path.

In this exercise, add a Decision to query how many records (matching those coming from your source) exist in the database. If the return value is greater than 0, the document is filtered out of the process flow.

1. From the **Logic** tab of the shapes palette, drag and drop a **Decision** shape onto the Process Canvas.



2. In the Decision Properties window's Display Name field, enter **New Account?**
  - ✓ *Unlike most other shapes, a decision should have a Display Name to make the logic clearly perceptible when viewing the Process Canvas.*



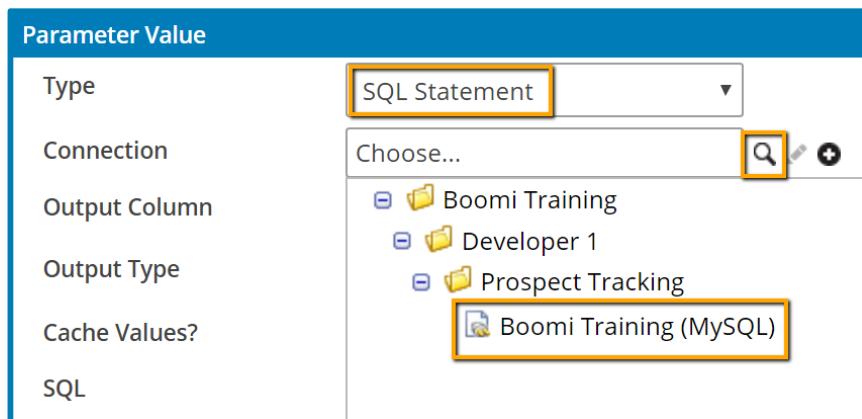
## Exercise 14: Add a Decision shape to query the database

3. For **First Value**, click **Choose...** inside the field.

4. In the Parameter Value window select:

Type = **SQL Statement**

Connection = **Boomi Training (MySQL)**



It is important to understand the logic contained in this SQL statement. Within the CUSTOMER table, return a COUNT (the number) of records where the NAME of the record is equal to the Salesforce Account Name and the USER\_ID equal to 'My Name' (the name you entered in the map).

The overall decision logic is:

If no records exist in the database, with the criteria of COUNT=0, the record from Salesforce is not in the database. The Decision shape will yield a TRUE result (0=0), and the document will flow down the path and be inserted in the database.

If a record exists in the database, with the criteria of COUNT!=0 (not equal to zero), the record from Salesforce already exists in the database. The Decision shape will yield a FALSE result (1≠0), and the document will flow down the false path and not be inserted in the database, thus preventing a duplicate database record.

5. Enter the following:

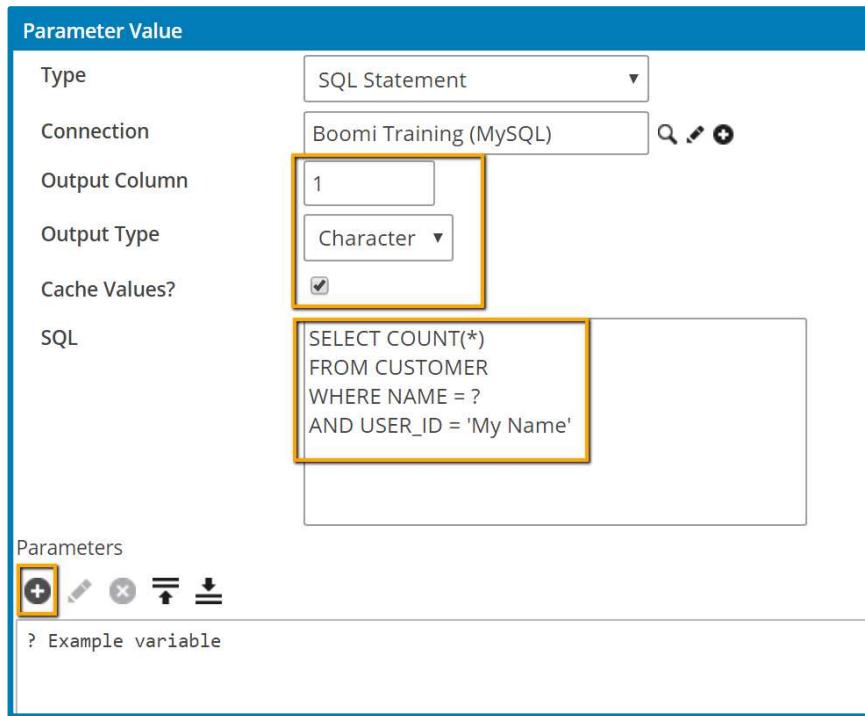
Output Column	1
Output Type	Character
Cache Values?	Checked
SQL	<pre>SELECT COUNT(*) FROM CUSTOMER WHERE NAME = ? AND USER_ID = 'My Name'</pre>

- ✓ The **My Name** value should correspond exactly to the default value entered into the

## Exercise 14: Add a Decision shape to query the database

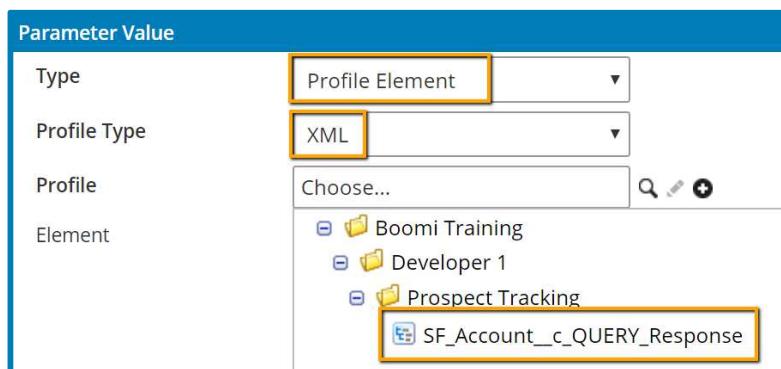
*USER\_ID field in the database destination profile in the SF Account to DB Insert Map.*

6. Click **Add Parameter (+)**.



*In SQL, the question mark ?, denotes the parameter entered below. The literal string in this example is 'My Name' (or the default value set earlier).*

7. In the **Parameter Value** window, set:  
Type = **Profile Element**  
Profile Type = **XML**
8. For **Profile**, click the **Browse** icon and select **SF\_Account\_\_c\_QUERY\_Response**.



9. For **Element**, click **Browse** and select **Account\_\_c > Name**.

## Exercise 14: Add a Decision shape to query the database

10. Click **OK** to accept and then **OK** to close .
11. Note the **Name (Account\_c/Name)** parameter listed, then press **OK** to return to the Decision shape's properties window.
12. Set Compare Type = **Equal To**. Then, for **Second Value**, click **Choose...** inside the field.

The screenshot shows the 'Decision Properties' window with the following fields:

- Display Name: New Account?
- First Value: SQL - SELECT COUNT(\*) FRO...
- Comparison: Equal To
- Second Value: Choose...

The 'Second Value' field is highlighted with a yellow box.

13. In the Parameter Value window select:

Type = **Static**

Static Value = **0**

Then click **OK**.

The screenshot shows the 'Parameter Value' window with the following entries:

Parameter Value	
Type:	Static
Static Value:	0

The 'Type' and 'Static Value' fields are highlighted with orange boxes.

14. Note the Decision Properties entries and then click **OK** to return to the process.

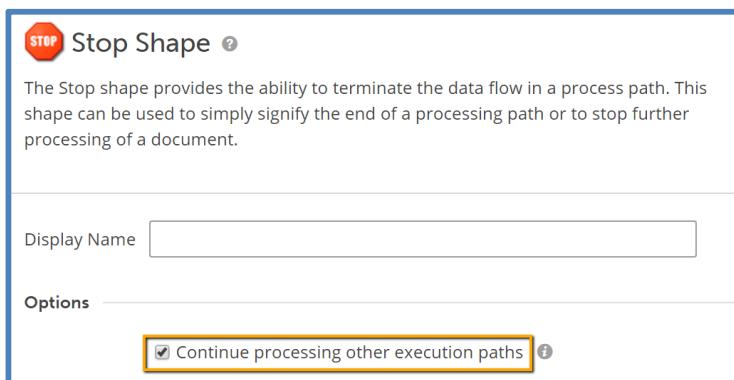
15. Click **Save** to save the process.



## Exercise 15: Test the decision lookup

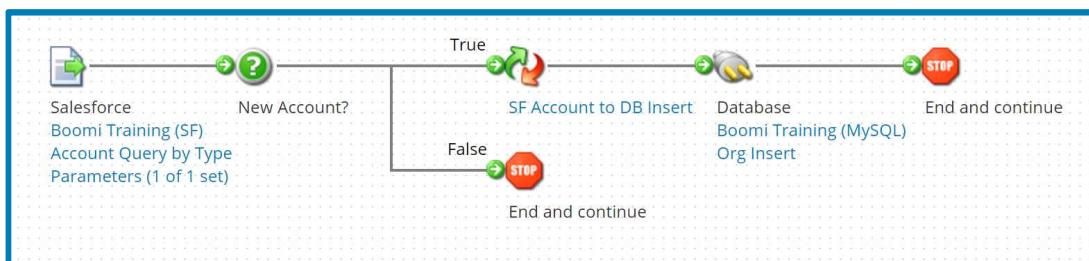
Since our Decision shape is configured, we must reorganize the existing flow to meet the result of the decision's logic (New Account?) to prevent new test documents from entering the Customer table.

1. From the Logic tab of the shapes palette, drag and drop a **Stop** shape onto the canvas.
2. The Stop shape's properties window opens. By default, **Continue processing other execution paths** is checked.



*The Stop shape's command defaults to continue processing other execution paths. If unchecked, the process stops. This is useful for unit testing large processes, as well as for viewing documents in Test Mode.*

3. Click **OK**.
4. Create another Stop shape by repeating steps 1-3.
5. Reconnect the **Start** shape to the **Decision** shape.
6. Connect the Decision shape's **True** path to the **SF Account to DB Insert Map**.
7. Connect the Decision shape's **False** path to one of the **Stop** shapes.
8. Connect the Database Connector to the other **Stop** shape.



9. Execute the process in Test Mode and view the results.

Congratulations on successfully completing the process Prospect Tracking!



## Exercise 16: Create a database read process

The following exercises deal with our second process, Daily Customer Wins, which builds a notification process and sets flags in both Salesforce and the database to complete the Account record synchronization.

### Create a subfolder to house the database read process

1. Click on the dropdown arrow next to the **Developer 1** folder.
2. Choose **New Folder** and name it **Daily Customer Wins**.

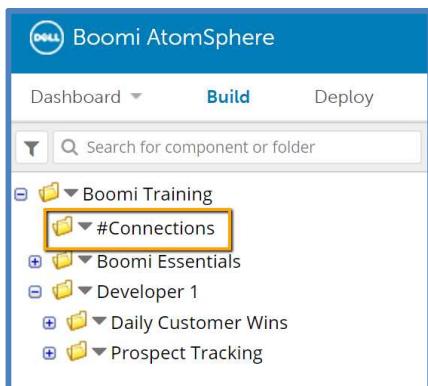
### Download the process from the Process Library

3. Open the Process Library by clicking on **Browse Process Library** at the bottom of the Component Explorer.
4. In the right column search bar, enter **Dev1** and press Enter.
5. Click **Install** next to **Daily Customer Wins**.
6. Click **Choose...** next to Select Installation Location and navigate to the Daily Customer Wins folder.
7. Click **Install** in the lower-right corner.
8. The process is now installed in your account. Click **View Process**.

### Create a folder to house Connections

As you continue to add processes to your account, it will be easier to organize and locate reusable Connections if they are stored in a **#Connections** folder. There are three Connections in your **Developer 1** folder: the Database Connection (Boomi Training - MySQL), the Salesforce Connection (Boomi Training - SF), and the Mail Connection (Boomi Training - Mail) that you loaded from the Process Library and placed in the “Daily Customer Wins” folder. Move all of these Connections to a **#Connections** folder.

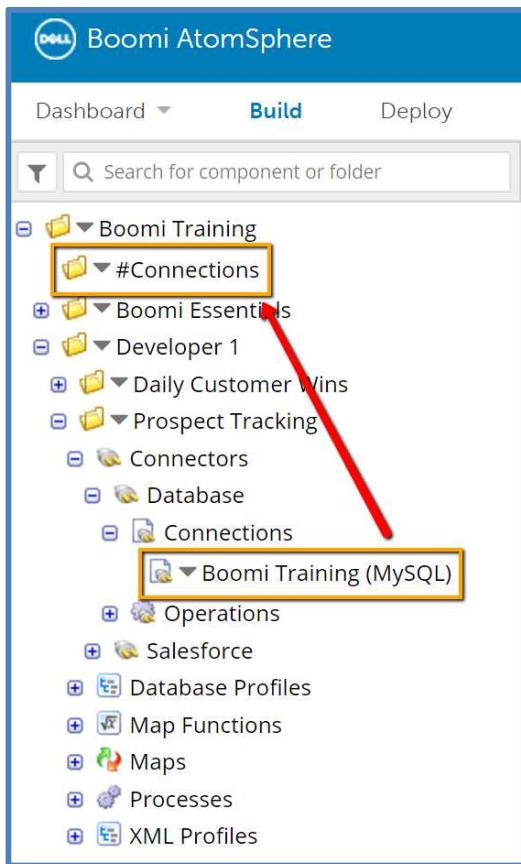
9. Click on the dropdown arrow next to the **Boomi Training** folder.
10. Choose **New Folder** and name it **#Connections**.



## Exercise 16: Create a database read process

11. In the **Prospect Tracking** folder, expand Connectors to expose **Boomi Training (MySQL)**.

12. Click and drag **Boomi Training (MySQL)** to the **#Connections** folder.



13. Repeat steps 11-12 to move **Boomi Training (SF)** and **Boomi Training (Mail)** to the **#Connections** folder.

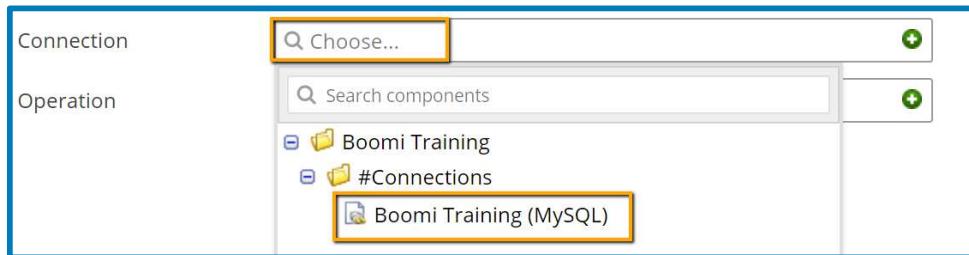


*Note: Boomi Training (Mail) is currently housed in the Daily Customer Wins folder.*

### Configure the Database Start shape

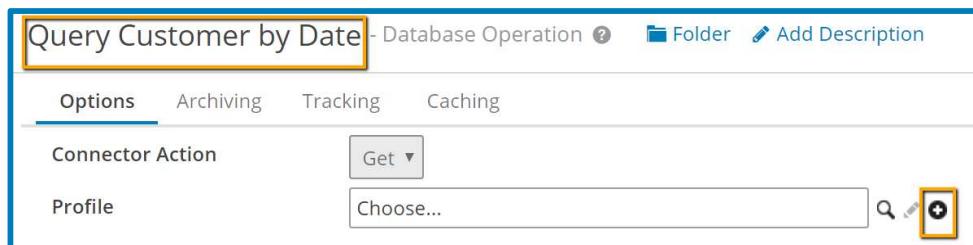
14. Click the **Database Start** shape to open the properties window.

15. For the Connection, click **Choose...**, navigate to the **#Connections** folder, and select the **Boomi Training (MySQL)** Connection.

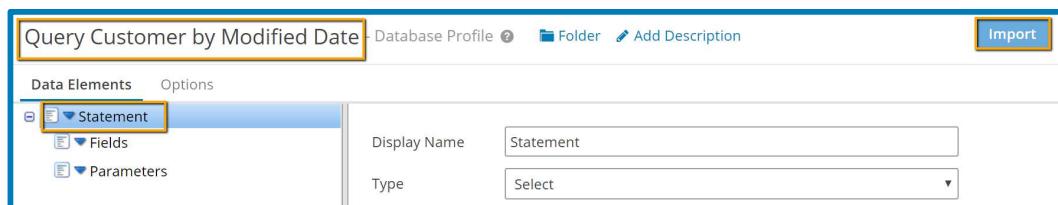


## Exercise 16: Create a database read process

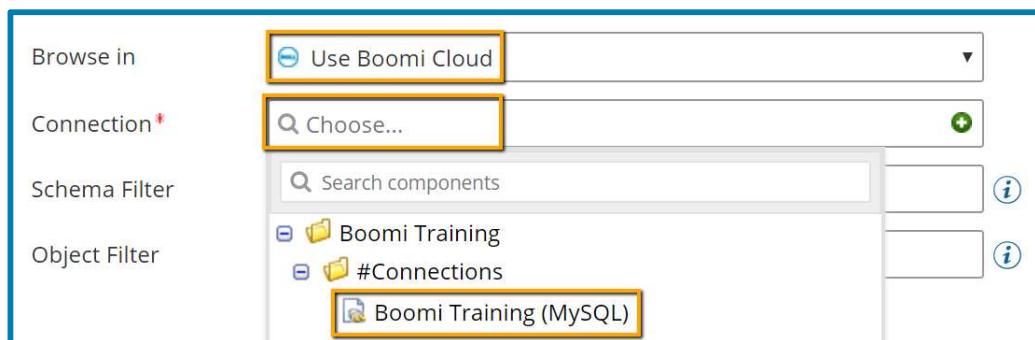
16. Click the **Create (+)** icon in the Operation field, and enter **Query Customer by Date** in the Name window.
17. Next to the Profile field, click **Create (+)** to open a new component tab.



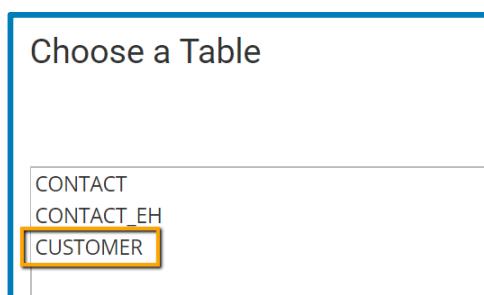
18. In the Name window next to **Database Profile**, enter **Query Customer by Modified Date**.
19. Click **Statement** and then click **Import**. The Database Import Wizard opens.



20. From the “Browse in” dropdown, select **Boomi Cloud**.
21. For Connection, browse to and select the **Boomi Training (MySQL)** connection and then click **Next**.



22. When the tables load, choose the **CUSTOMER** table and click **Next**.



## Exercise 16: Create a database read process

23. Select only these three fields: **CUSTOMER\_ID**, **NAME**, and **PHONE**, click **Next**, and then click **Finish**.

The screenshot shows a list of fields for a 'CUSTOMER' entity. The selected fields are CUSTOMER\_ID, NAME, and PHONE, which are highlighted with orange boxes around their respective checkboxes.

- COMPANY\_KEY
- CREATE\_DATE
- CUST\_REF
- CUSTOMER\_ID
- DEPARTMENT
- FAX\_NUM
- INDUSTRY
- MODIFIED\_DATE
- NAME
- NOTIFY\_STATUS
- OWNER NAME
- PHONE

24. The fields are now in both the fields list and the SQL Script.

The screenshot shows the 'Data Elements' configuration screen. Under the 'Statement' section, the 'Fields' list contains CUSTOMER\_ID, NAME, and PHONE, which are highlighted with orange boxes. The 'SQL Script' field contains the generated SQL query: 'SELECT CUSTOMER\_ID, NAME, PHONE FROM CUSTOMER'.

Data Elements		Options
<input type="checkbox"/> Statement		Type: Select
<input type="checkbox"/> Fields		Position: 1
SQL Script:		
SELECT CUSTOMER_ID, NAME, PHONE FROM CUSTOMER		



*Enhance this SQL script to include make the CUSTOMER\_ID DISTINCT and the WHERE clause to determine the reported items (CUSTOMER\_ID, NAME and PHONE) and the records having specific field values (MODIFIED\_DATE, USER\_ID, COMPANY\_KEY, and NOTIFY\_STATUS).*

25. Replace the existing SQL Script with the SQL Script below. Use the name entered for the **USER\_ID** in the map of Prospect Tracking for the 'My Name' field:

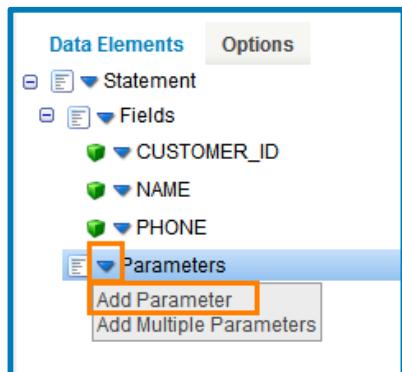
```
SELECT DISTINCT CUSTOMER_ID, NAME, PHONE
FROM CUSTOMER
WHERE MODIFIED_DATE >= ?
AND USER_ID = 'My Name'
AND COMPANY_KEY = '59'
AND NOTIFY_STATUS = 0
```

## Exercise 16: Create a database read process

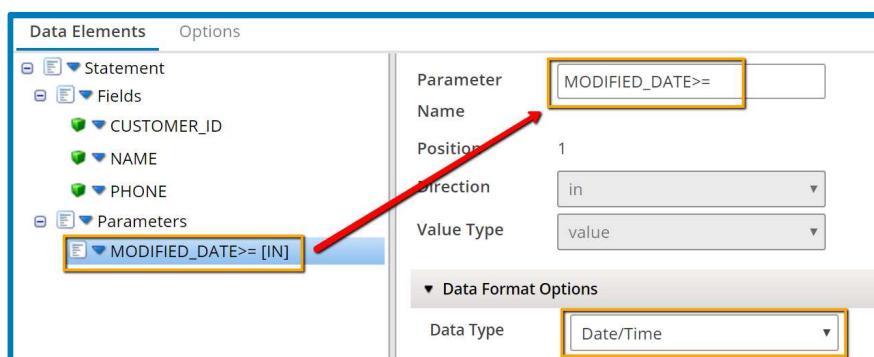


*In the database, Prospect Salesforce account records have been won and updated as Customer - Direct. These are the accounts that will be read into the process.*

26. In the left window next to Parameters, click the blue dropdown arrow and choose **Add Parameter**.



27. Highlight the new parameter on the left.
28. In the Parameter Name field, on the right, enter **MODIFIED\_DATE>=**
29. Expand the Data Format Options, and for Data Type select **Date/Time**.

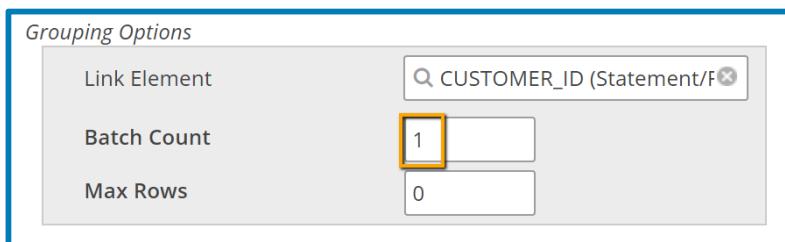
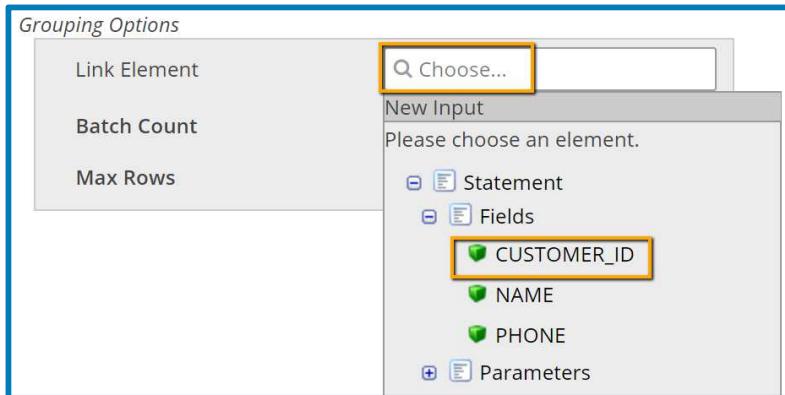


30. Click **Save and Close** to return to the Database Read Operation.
31. In the Operation's Grouping Options, next to Link Element click **Choose...**, expand the Fields set and select **CUSTOMER\_ID**.
32. Click **OK**, and then check the following values are entered for the Grouping Options:

Link Element:	<b>Statement &gt;&gt; Fields &gt;&gt; CUSTOMER_ID</b>
Batch Count:	<b>1</b>
Max Rows:	<b>0</b>



## Exercise 16: Create a database read process



33. Click **Save and Close**, then click **OK** to return to the Process Canvas.

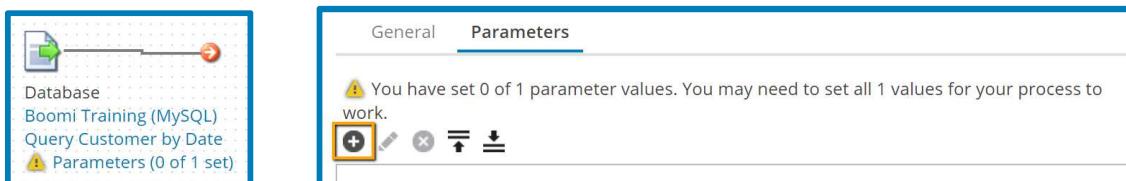
34. Click **Save** to save the Daily Customer Wins process.



## Exercise 17: Set a database runtime parameter

Set a dynamic input parameter on the Start shape set a dynamic input parameter on the Start shape to populate the database parameter in the SQL query. This ensures the database records returned are unique and inserted in the desired timeframe.

1. Click the **Start** shape, then move to the Parameters tab and click **Add (+)**.



2. For the Input value, click **Browse** and select the MODIFIED\_DATE>= parameter created earlier.
3. Enter the following information to complete the Parameter Value entries:

Type	Date/Time
Date Mask	yyyyMMdd HHmmss.SSS
Date Type	Relative Date
Relative to Current Date	- 1 Day(s)

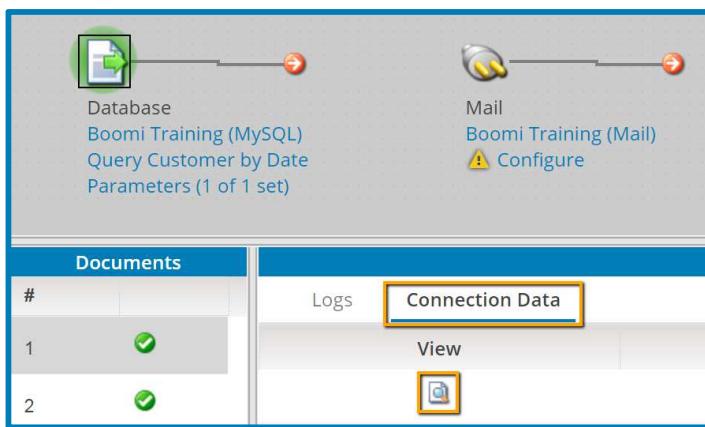
This screenshot shows the 'Parameter Value' dialog box. It contains five input fields: 'Input' with a search bar containing 'MODIFIED\_DATE>='; 'Type' set to 'Date/Time'; 'Date Mask' set to 'yyyyMMdd HHmmss.SSS'; 'Date Type' set to 'Relative Date'; and 'Relative To Current Date' with dropdowns for '-', '1', and 'Day(s)'. The entire dialog box is highlighted with a blue border.

4. Click **OK** to close the dialog and then **OK** to return to the process.
5. Click **Save** to save the process.

## Exercise 18: Test the process

It is time to test the process to see if the two database records were successfully retrieved. A Boomi best practice is to continuously test your process at various stages throughout development.

1. Click **Test** in the upper-right of the Process Canvas.
2. Select the **Test Atom Cloud** Atom from the dropdown and click **Run Test**.
3. Two documents are retrieved, so individually highlight each document, click the **Connection Data** tab and select **View the contents icon**.



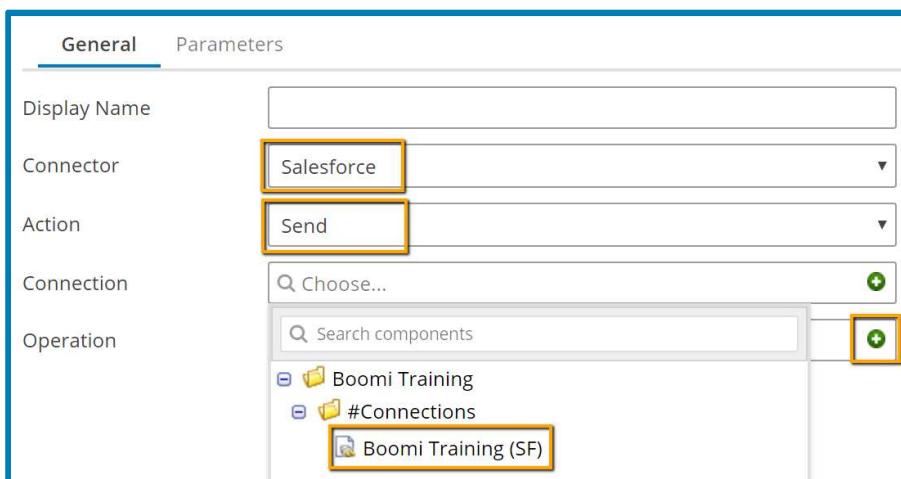
- ✓ A separate entry displays for each Document retrieved.



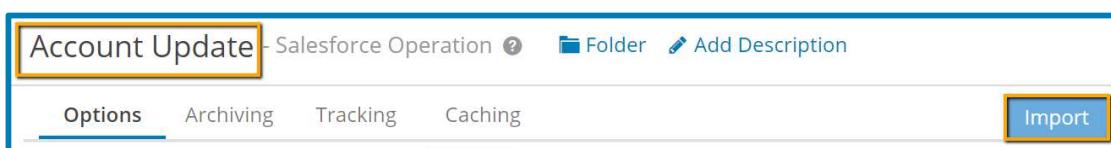
## Exercise 19: Create a Salesforce update action

The Salesforce users monitoring customers that have been successfully entered as Closed/Won in the organization tracking system need to update the accounts' Service Level Agreements (SLA) to a status of 'Gold.' In the Daily Customer Wins process, add a Salesforce Update operation to change the SLA from its current setting to *Gold*.

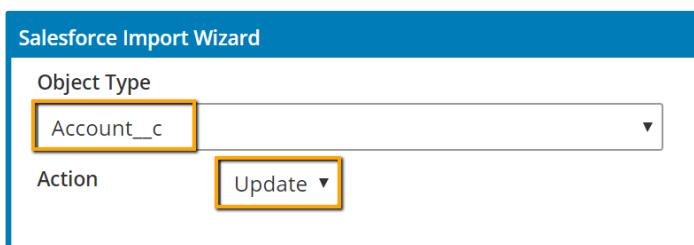
1. Drag and drop a Connector shape onto the Process canvas.
2. Set Connector to **Salesforce** and set Action to **Send**.
3. For the Connection, choose the Salesforce connection component **Boomi Training (SF)** used earlier and now housed in #Connections.
4. Next to Operation, click **Add (+)** to create a new component tab.



5. Enter **Account Update** in the name window and click **Import**.



6. In the Salesforce Import Wizard, choose the Salesforce connection component in #Connections.
7. Click **Next**.
8. Set Object Type to **Account\_\_c**, set Action to **Update**, and then click **Next**.



## Exercise 19: Create a Salesforce update action

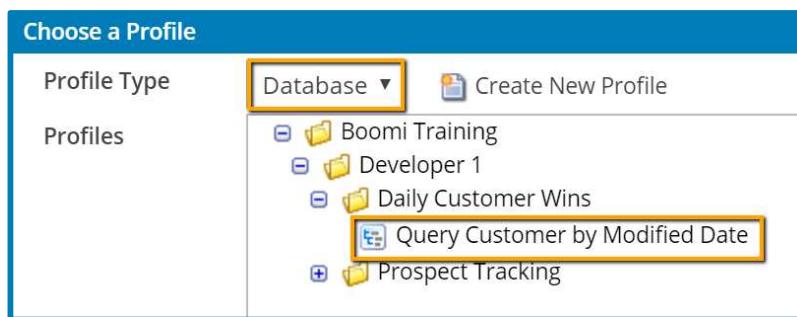
9. Ignore the Account's Objects (do not select anything).
10. Click **Next**, and then click **Finish**.
11. Click **Save and Close** and then click **OK** to return to the process.
12. Click **Save** to save the process.



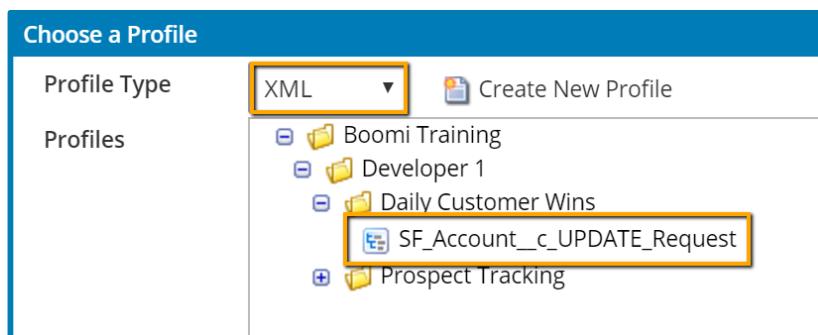
## Exercise 20: Create a Salesforce update map

To update the proper account records in Salesforce, create a Map using the database customer account name and phone information as a unique value to update each record. Salesforce uses an object's internal ID (Id) field to identify the records needing to be updated. This ID field is not available in the database source data, so a connector call function needs to be added to query the account based on the name and phone and extract the record's Internal ID. The Service Level Agreement (SLA) field can then be hard coded as **Gold** to reflect the new contract level in Salesforce.

1. Drag and drop a **Map** shape onto the process canvas.
2. Click **Create (+)** to open a new Map tab.
3. Enter **DB to SF Account Update** in the name window.
4. In the left mapping window, click **Choose**.
5. Select the Database Profile Type, highlight the **Query Customer by Modified Date** profile, and then click **OK**.



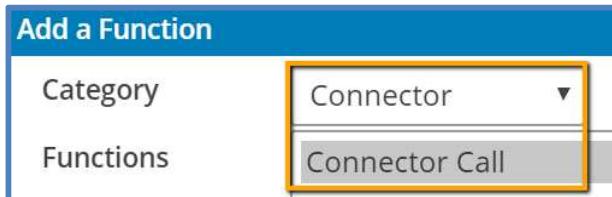
6. In the right mapping window, click **Choose**.
7. Select the **XML Profile Type**, highlight the newly created **SF\_Account\_\_c\_UPDATE\_Request Profile**, and then click **OK**.



8. Click **Add a function (+)** in the center pane of the Map.

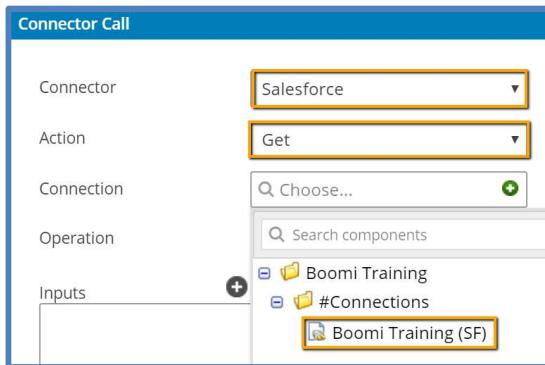
## Exercise 20: Create a Salesforce update map

9. From the Category dropdown menu, choose **Connector**, highlight **Connector Call**, and click **OK**.



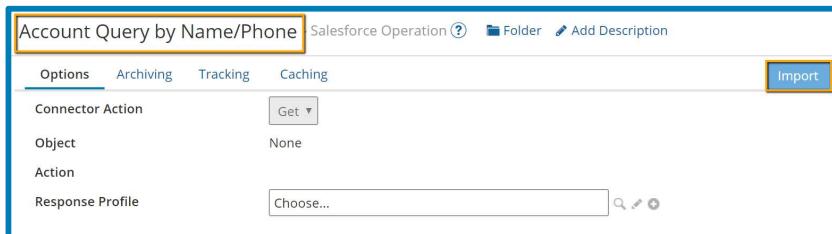
10. In the Connector Call window, choose the following options:

Connector:	Salesforce
Action:	Get
Connection:	Boomi Training (SF)



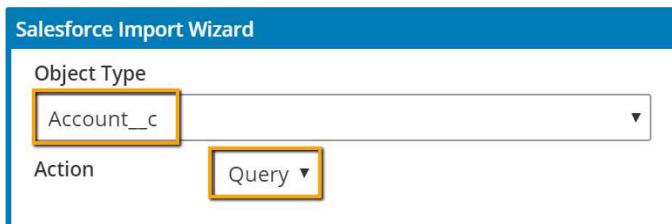
11. Next to Operation, click **Create (+)** to open a new Operation tab.

12. In the Name window, enter **Account Query by Name/Phone**, then click **Import**.



13. Click **Browse**, choose the Salesforce connection **Boomi Training (SF)** component in your account, then click **Next**.

14. Set Object Type to **Account\_\_c**, set Action to **Query**, then click **Next**.



## Exercise 20: Create a Salesforce update map

15. Ignore the Account's Object Tree nodes and click **Next**. Click **Finish**.
16. Notice the **SF\_Account\_\_c\_QUERY\_Response** profile is loaded in the Operation and click **Finish**.
17. At the bottom of the query window under the Objects tab, highlight **Account**.

Objects	Fields	Filters	Sorts
Account	<input checked="" type="checkbox"/> Id <input checked="" type="checkbox"/> IsDeleted <input checked="" type="checkbox"/> MasterRecordId <input checked="" type="checkbox"/> Name <input checked="" type="checkbox"/> Type <input type="checkbox"/> ParentId		

18. On the Fields tab:
  - a. Check the main **Account\_\_c** node to select all the Account fields
  - b. Uncheck the node to de-select all of the Account fields
  - c. Check the top **Id** field.

Fields	Filters	Sorts
<input checked="" type="checkbox"/> Account__c <input checked="" type="checkbox"/> Id <input checked="" type="checkbox"/> OwnerId <input checked="" type="checkbox"/> IsDeleted <input checked="" type="checkbox"/> Name		
<input type="checkbox"/> Account__c <input checked="" type="checkbox"/> Id <input checked="" type="checkbox"/> OwnerId <input checked="" type="checkbox"/> IsDeleted <input checked="" type="checkbox"/> Name		
<input type="checkbox"/> Account__c <input checked="" type="checkbox"/> Id <input type="checkbox"/> OwnerId <input type="checkbox"/> IsDeleted <input type="checkbox"/> Name		

19. Click the **Filters** tab, then click the blue dropdown arrow next to **Logical - AND**, then choose **Add Expression**.

Fields	Filters	Sorts
	<input type="checkbox"/> Filter <input type="checkbox"/> Logical - AND Add Logical Sub-Group <input checked="" type="checkbox"/> Add Expression Delete Logical Group	

20. Highlight the new expression and define the following for the filter instance:

Filter Name:	Name=
Field:	Name
Operator:	Equal To



## Exercise 20: Create a Salesforce update map

The screenshot shows the 'Filters' tab of a configuration interface. A logical AND filter is selected. It contains one condition: 'Name=' with the operator 'Equal To'. The 'Field' dropdown shows 'Name'.

21. Add a second expression and define the following for the filter instance:

Filter Name:	<b>Phone=</b>
Field:	<b>Phone_c</b>
Operator:	<b>Equal To</b>

The screenshot shows the 'Filters' tab of a configuration interface. A logical AND filter is selected. It contains two conditions: 'Name=' and 'Phone=' (highlighted with an orange box). Both conditions have the operator 'Equal To'. The 'Field' dropdown shows 'Phone\_c' for the 'Phone=' condition.

22. Click **Save and Close** to return to the Connector Call map function.

23. In the Connector Call's Input window, click **Add (+)**, highlight **Name =**, and then click **Add**.

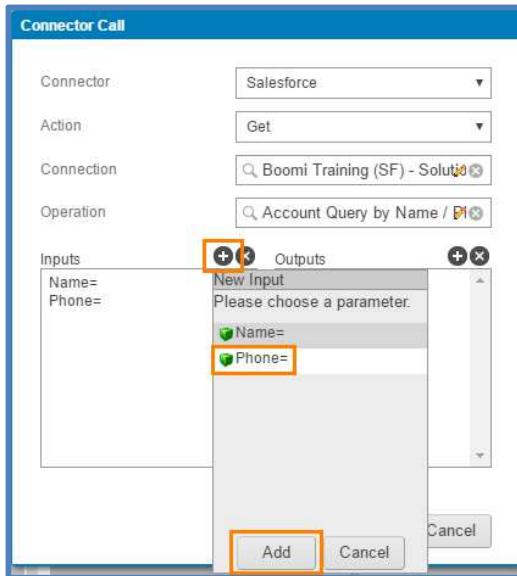
The screenshot shows the 'Connector Call' input window. Under the 'Inputs' section, there is a list of parameters: 'Name=' and 'Phone=' (highlighted with an orange box). A 'New Input' dialog is open, listing 'Name=' and 'Phone=' (also highlighted with an orange box). The 'Add' button in the dialog is highlighted with an orange box.

24. In the Connector Call's Input window, click **Add (+)**.

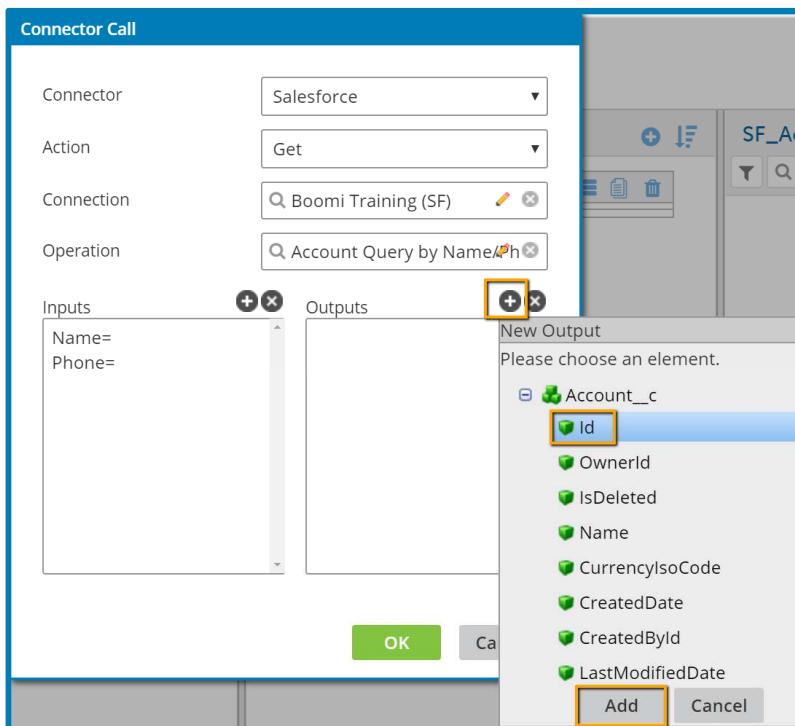
25. Highlight **Phone=**, and then click **Add**.



## Exercise 20: Create a Salesforce update map



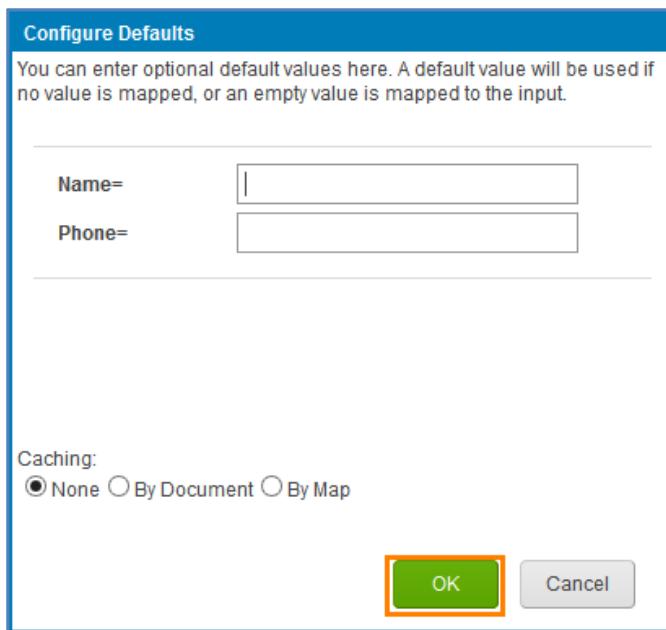
26. In the Connector Call's Output window, click **Add Output (+)**, highlight the **Account\_\_c > Id** field and then click **Add**.



27. In the Connector Call window, click **OK** to display the Configure Defaults screen.
28. Ignore the optional default value setting, because the NAME and PHONE are required fields in the database.
29. Click **OK** to return to the main Map.

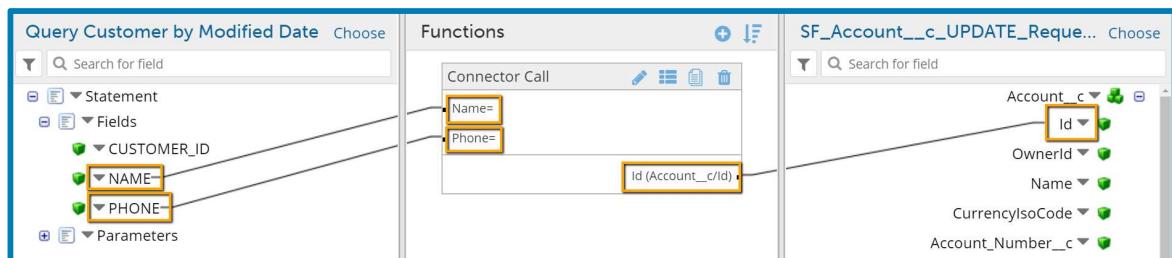


## Exercise 20: Create a Salesforce update map



30. In the source profile, map the following:

- the Database NAME field to the Name in the Connector Call function Input
- the PHONE field to the Phone in the Connector Call function Input
- the Id (Account\_\_c/Id) Connector Call result to the Id field in the SF\_Account\_\_c\_UPDATE\_Request destination profile

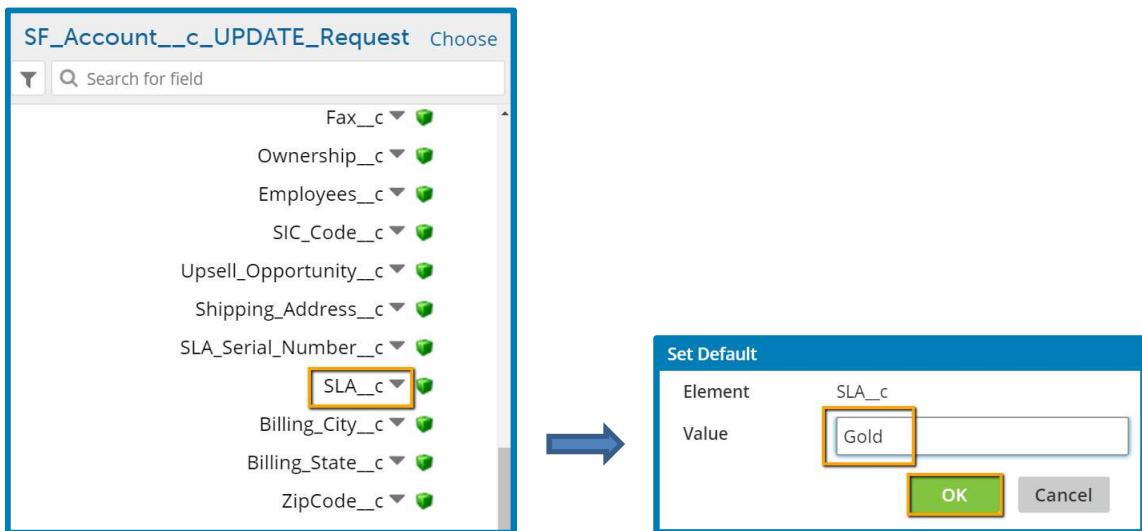


31. On the destination profile in the destination window, click the blue dropdown arrow to the right of the **SLA\_\_c** field, then choose **Set Default Value**.

32. In the Value field, enter **Gold**, then click **OK**.

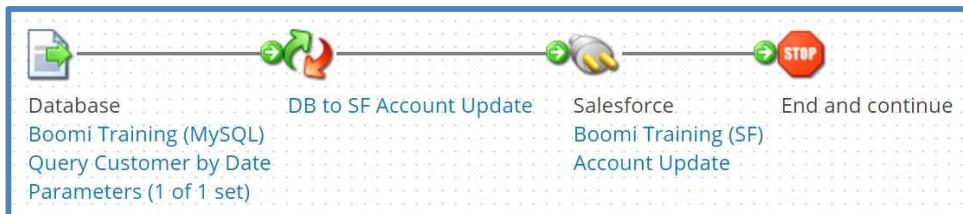


## Exercise 20: Create a Salesforce update map



Since we want this value to coincide with the exact SLA value (stored as a pick list item) in Salesforce, it is important to enter the value exactly as it appears in the Salesforce record.

33. Click **Save and Close** to save the Map.
34. If the Map Properties window is still open, click **OK** to return to the process, then click **Save** to save the process.
35. Connect the **Start shape** to the **Map**, connect the **Map** to the **Salesforce Connector**, and add a **Stop shape** to complete the path. The resulting complete process should look like this:



36. Execute the process in Test Mode by clicking on **Test** and note the results.



## Exercise 21: Create a mail operation

An email needs to be sent to the account executive to tell them when the account records have been updated to *closed/won* with an SLA contract level of *Gold*. This is completed by using a Mail Connector to send an email through an SMTP server to one or many **To** addresses.



We are using a single Gmail account that multiple users access simultaneously. The connection settings may be valid ONLY for short intervals during training sessions and may cause your processes to error if used outside our training session.

Google closely monitors account access. If they discover multiple IP addresses are attempting to simultaneously access a specific Gmail account, they may temporarily block account access.

To test this process outside of training, reconfigure the email connection settings and credentials in AtomSphere to use your own mail client account and enable POP access in your mail client.

The connection component to a hosted SMTP server is preconfigured, and now we need to pair the operation with the connection to define the Subject and From/To Addresses for the final report.

1. Click the **Mail Connector** to open the Action window. Next to Operation, click **Create (+)** to open a new tab.

The screenshot shows the 'General' tab of the Mail Action window. It includes fields for Display Name, Connector (set to 'Mail'), Action (set to 'Send'), Connection (set to 'Boomi Training (Mail)'), and Operation. The 'Operation' field has a small green '+' button with an orange border to its right, indicating where to click to create a new operation.

2. In the Name window, enter **Customer Win Email**.
3. Enter the following information for the **Mail Operation** details:

From:	<b>boomitrain@gmail.com</b>
To:	<b>*** Your email address***</b>
Subject:	<b>New Customer</b>
Disposition:	<b>Inline</b>
Content Type:	<b>text/plain</b>



## Exercise 21: Create a mail operation

Customer Win Email - Mail Operation ? Folder Add Description

Options Archiving Tracking Caching

Connector Action Send

From boomitrain@gmail.com

To \*\*\*Your email address\*\*\*

Subject New Customer

Disposition Inline

Content Type text/plain

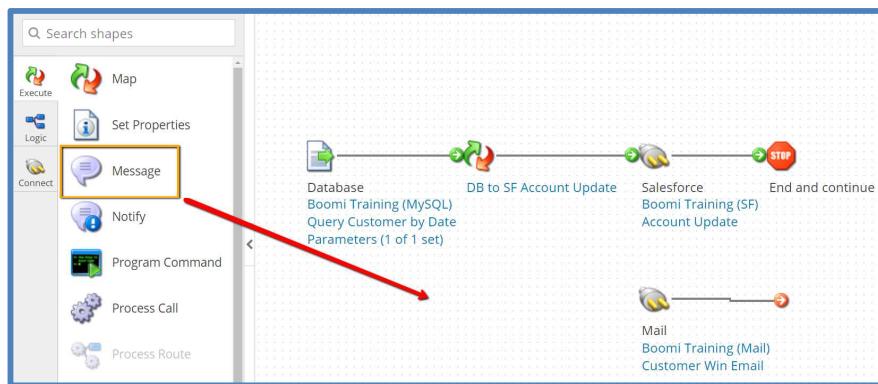
4. Click **Save and Close**, then click **OK** to return to the Process Canvas.



## Exercise 22: Configure a message shape

Similar to how a Map shape converts a document from one format to another, the Message shape overwrites the document data and builds it into a message to send in the body of an email. The following exercise extracts dynamic content from each document to build a custom report message.

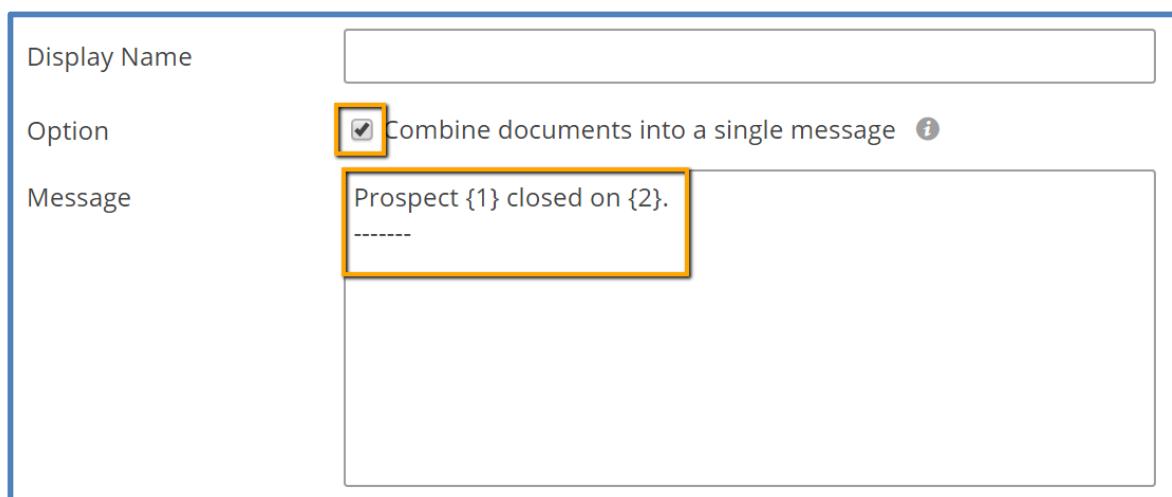
1. From the Execute tab of the shapes palette, drag and drop a **Message** shape onto the Process Canvas.



2. Check the **Combine documents into a single message** option, then enter the following lines in the Message text box:

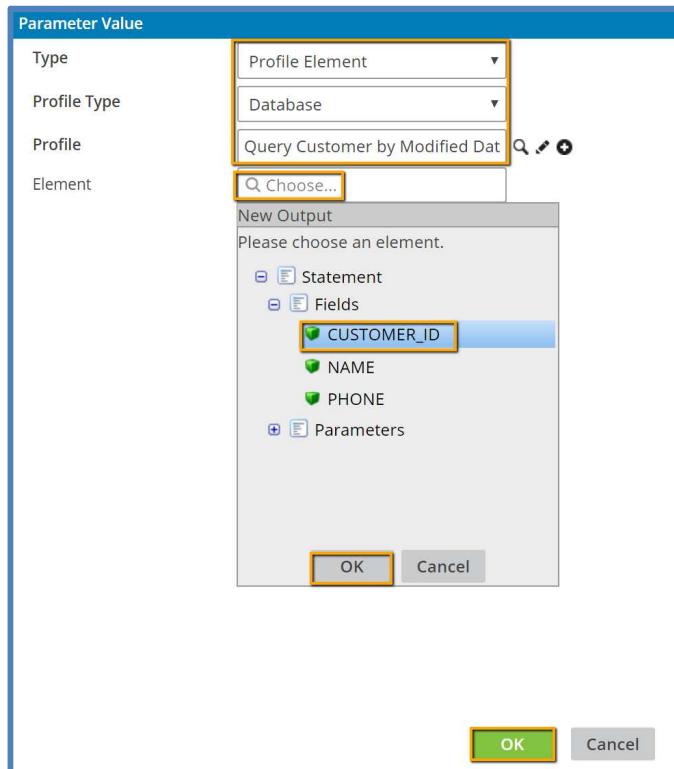
**Line 1)** Prospect {1} closed on {2}.  
**Line 2)** ----- and press <Enter>  
**Line 3)** <Blank Line>

- ✓ In the Message text box include a hard return after ‘-----’, so the cursor moves to the next line. This creates a blank line between records for easier viewing.



## Exercise 22: Configure a message shape

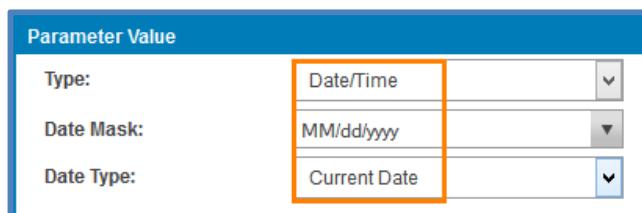
3. Next to Variable, click **Add (+)**. The Parameter Value window opens.
4. Set Type to **Profile Element**, then set Profile Type to **Database**.
5. For Profile, click **Browse** to navigate to and select the **Query Customer by Modified Date** profile.
6. For Element, click **Choose...** to navigate to and select **CUSTOMER\_ID**.
7. Click **OK** to close the dialog and then **OK** again to return to the Message Properties window.



8. Click **Add (+)** to add a second parameter value.
9. Configure the following for the second parameter value:

Type:	Date/Time
Date Mask:	MM/dd/yyyy
Date Type:	Current Date

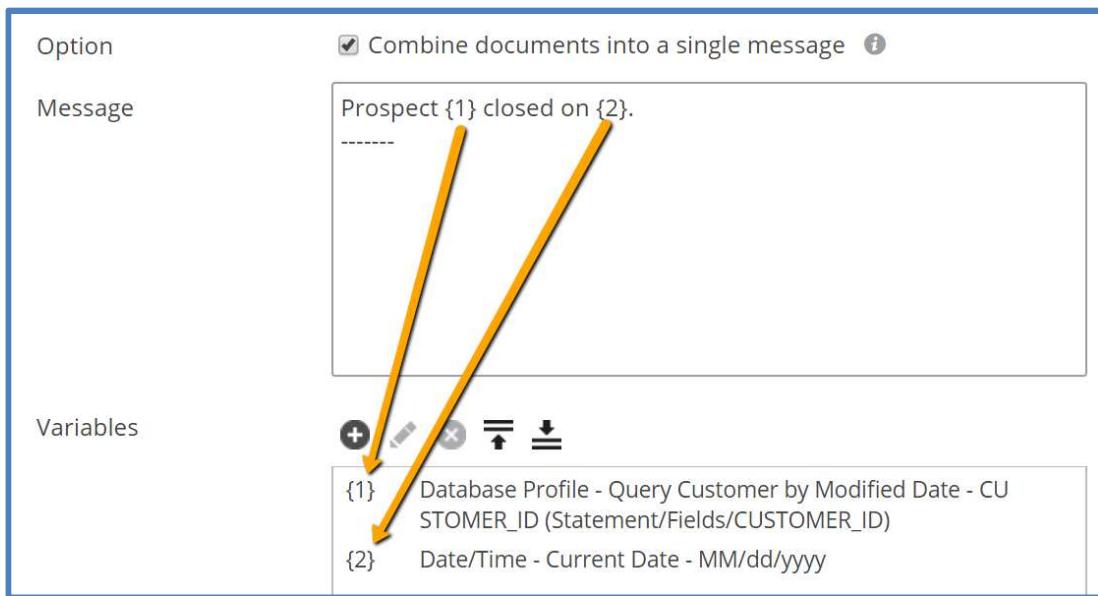
- ✓ If the date format you want to use is not listed, you can enter it by typing in the Date Mask field.



## Exercise 22: Configure a message shape

10. Click **OK** to return to the Message Properties.

Note there are now two parameters included as dynamic parts of the Message.

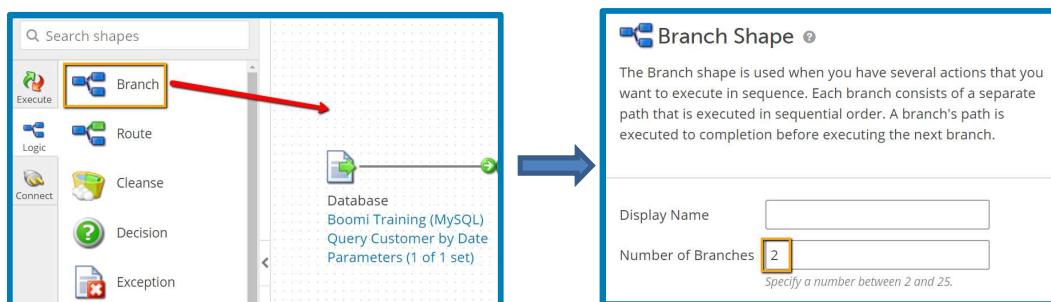


11. Click **OK** to return to the Process Canvas.

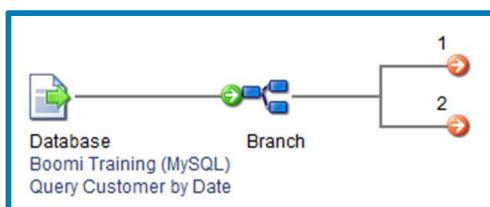
## Exercise 23: Test the mail connectivity

The Message is now constructed, and the Mail connector is enabled to send messages to your email address. Executing the process will confirm records are extracted and updated in Salesforce, and the process will then email formalized message. To do this dual action, we will use the Branch shape.

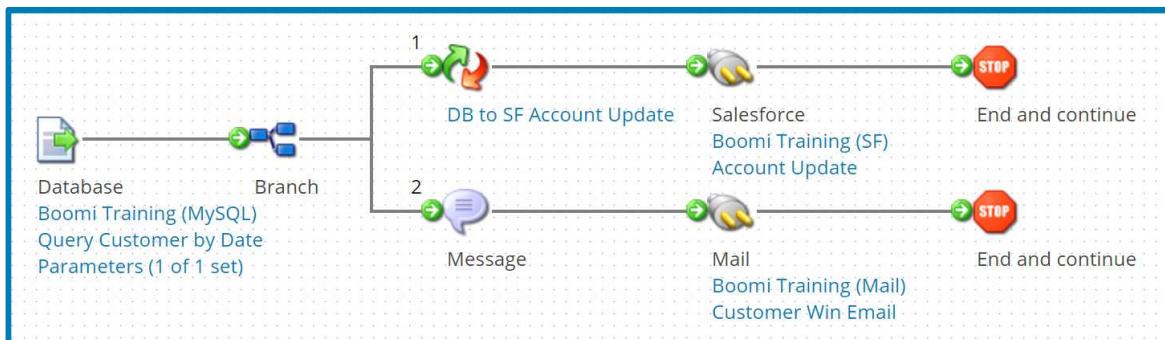
1. From the Logic tab of the shapes palette, add a **Branch** shape to the process flow.
2. Set the Number of branches to **2** and then click **OK**.



3. Disconnect the Start shape from the **DB to SF Account Update Map** and connect it to the Branch shape.



4. Connect the 1st Branch path to the Map and the 2nd Branch path to the Message shape.
5. Connect the Message shape to the Mail Connector.
6. Add a Stop shape to the 2nd Branch path. The resulting process should look like this:



7. Execute the process in Test Mode by clicking on **Test**.
8. Select the Test Atom Cloud atom and click **Run Test**.

## Exercise 23: Test the mail connectivity

9. Navigate to your email inbox to see the results.

Document Viewer

Prospect GenePoint3450 closed on 06/04/2017.  
-----  
Prospect EdgeCommunications6000 closed on 06/04/2017.  
-----

Sun 6/4/2017 9:37 AM  
 boomitrain@gmail.com  
New Customer  
To   
  
Prospect GenePoint3450 closed on 06/04/2017.  
-----  
Prospect EdgeCommunications6000 closed on 06/04/2017.  
-----



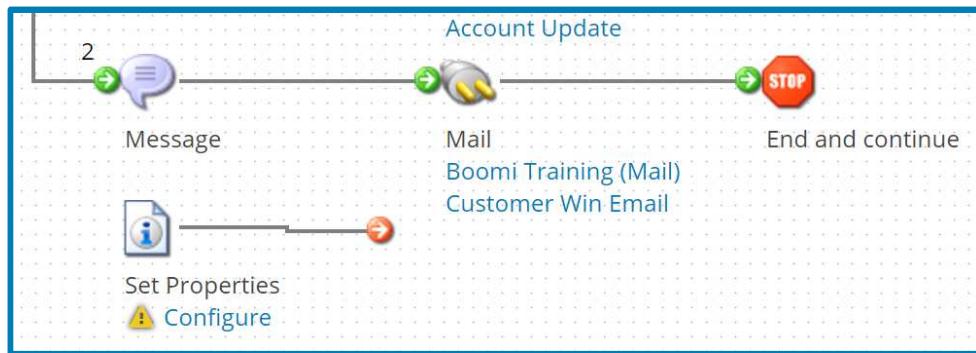
*Check your spam folder if you do not receive an email.*



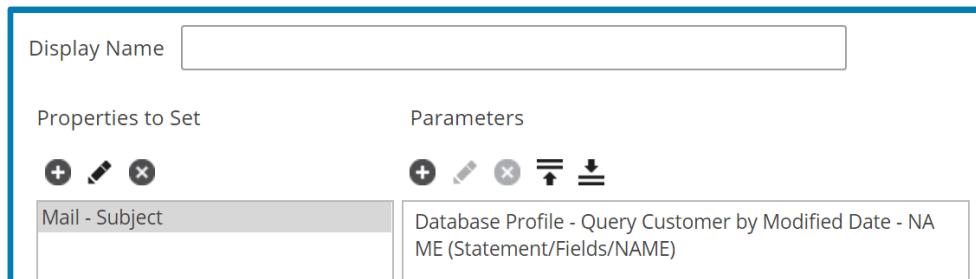
## Additional Challenge 3: Format output to email

When sending email notifications, dynamic content can be used in the subject or to send messages to a variable set of **To** addresses. This exercise uses the Set Properties shape to make the email subject dynamic, containing each database record's name.

1. Add a Set Properties shape to the Process Canvas
  - ✓ Consider where the Set Properties shape should be positioned in the process flow.



2. Configure a Profile Element Parameter Type value that dynamically sets the Mail Subject by NAME. The resulting Set Properties window should look like this:



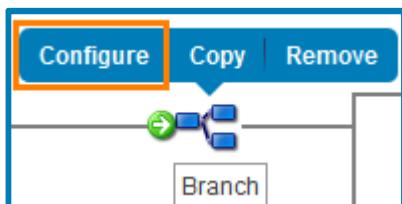
3. Uncheck the Combine documents into a single message option in the Message shape. The result should yield separate emails, titled by customer name, each containing the record's NAME, Customer ID, and Date, with a '-----' separator in the body.



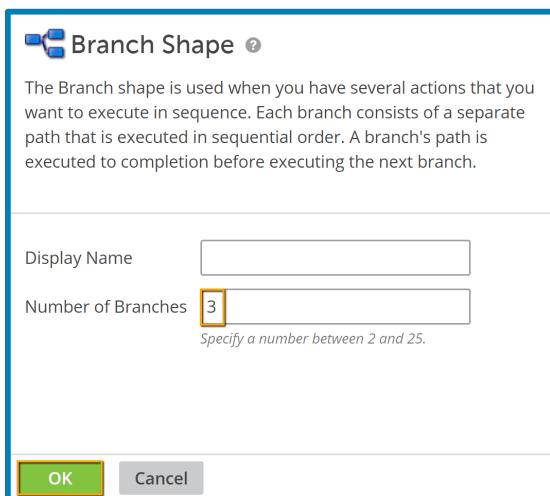
## Exercise 24: Add a SQL command

After creating an email notification, update the Organization Tracking System's corresponding record's NOTIFY\_STATUS field to confirm the email was sent successfully. Use the Program Command shape as a quick alternative to executing an update statement with dynamic content.

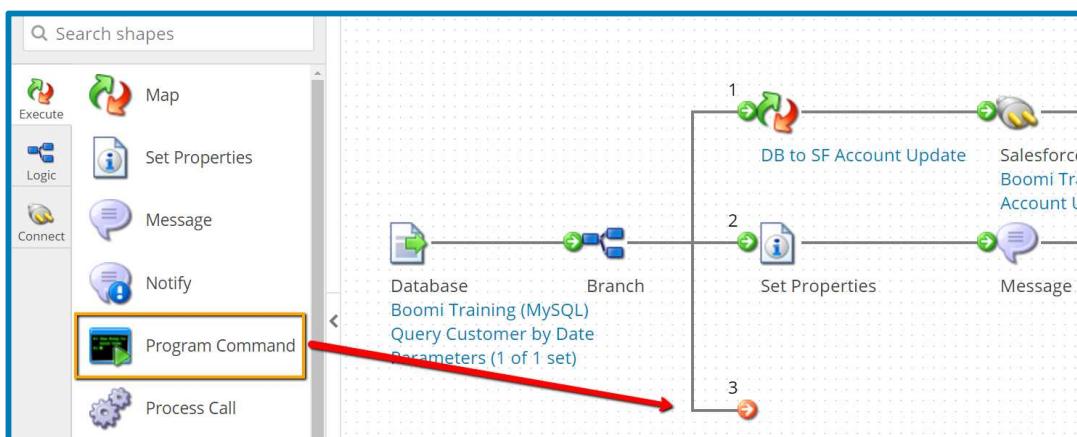
1. Hover over the Branch shape and choose **Configure**.



2. Increase the Number of Branches to **3** and click **OK**.



3. From the Execute tab of the shapes palette, drag and drop a **Program Command** shape onto the canvas.



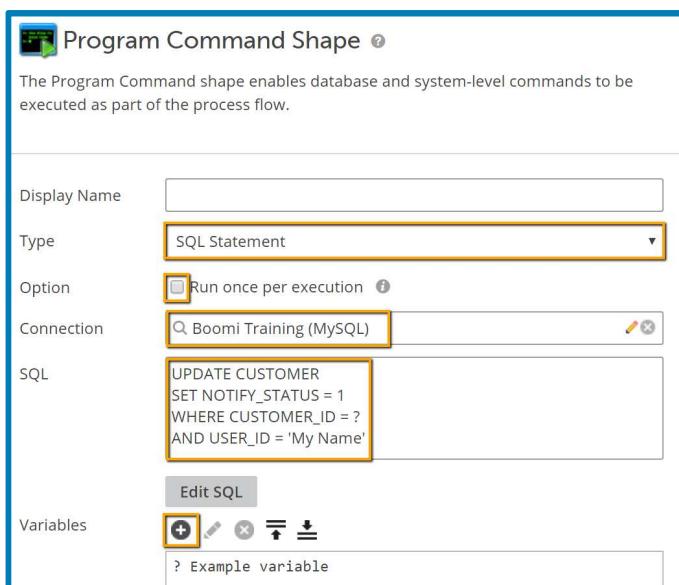
## Exercise 24: Add a SQL command

4. In the Program Command Properties window, configure the following:

Type:	<b>SQL Statement</b>
Run Once Per Execution:	<b>Unchecked</b>
Connection:	<b>Boomi Training (MySQL)</b>
SQL:	<b>UPDATE CUSTOMER SET NOTIFY_STATUS = 1 WHERE CUSTOMER_ID = ? AND USER_ID = 'My Name'</b>

- ✓ Enter '**My Name**' exactly the same as you entered it as the default value on the **USER\_ID** field in the SF Account to DB Insert Map.

5. Click **Add (+)**.



*It is important to understand the logic of the SQL statement:*

**UPDATE CUSTOMER  
SET NOTIFY\_STATUS = 1  
WHERE CUSTOMER\_ID = ?  
AND USER\_ID = 'My Name'**

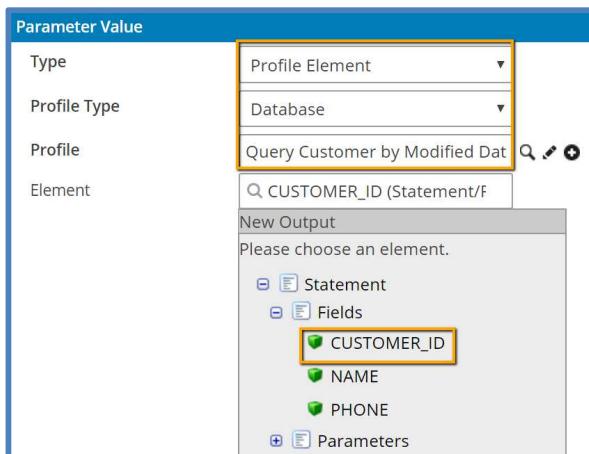
*In the CUSTOMER table of the DB, UPDATE the NOTIFY\_STATUS field to 1 where the CUSTOMER\_ID of the Database record matches the one queried at the front end of the process and the USER\_ID is My Name (or the default value set in the map).*

*If the document flows successfully through both branch 1 (updating the record in Salesforce from **Prospect** to **Customer - Direct**) and then branch 2 (emailing the monitoring team), then branch 3 executes and updates the database record (NOTIFY\_STATUS=1) with the new Account status.*

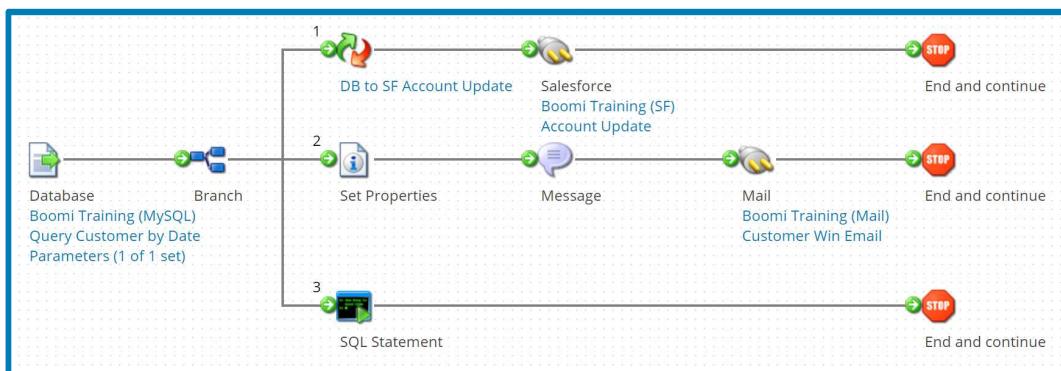
## Exercise 24: Add a SQL command

6. Configure the following for the first parameter value:

Type:	<b>Profile Element</b>
Profile Type:	<b>Database</b>
Profile:	<b>Query Customer by ModifiedDate</b>
Element:	<b>CUSTOMER_ID (Statement/Fields/CUSTOMER_ID)</b>



7. Click **OK** to close the dialog and return to the Process Canvas.
8. Connect branch 3 to the Program Command shape.
9. Drag and drop another **Stop** shape onto the canvas.
10. Leave the **Continue processing other execution paths?** checked and click **OK**.
11. Connect the SQL Statement shape to the Stop shape. The resulting process should look like the following:



12. Click **Save**.
13. Execute the process in Test Mode and note the results.



*This process can only be executed once. Updating the NOTIFY\_STATUS will not retrieve processes in future executions.*