



Arbutus Analyzer Intermediate Training Case Study Solutions

Compatible with Analyzer Version 6.3x



Arbutus Analyzer Intermediate Training

Copyright

Copyright © 2020 Arbutus Software Inc. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means (photocopying, electronic, mechanical, recording or otherwise), without permission in writing from the publisher.

This manual may contain dated information. Use of these materials is based on the understanding that this manual may not contain all the information, or the most current information, relevant to an intended application. Support and update services are available from Arbutus Software Inc., by separate written agreement.

Published December 2020 by:

Arbutus Software Inc.
270 - 6450 Roberts Street
Burnaby, British Columbia
Canada V5G 4E1

Telephone: +1-604-437-7873

Fax: +1-604-437-7872

E-mail: Training@ArbutusSoftware.com

Web: www.ArbutusSoftware.com

“Analyzer” is a registered trademark of Arbutus Software Inc., or one of its subsidiaries or licensors. Microsoft, Windows, and the Windows logo are registered trademarks of Microsoft Corporation. Other brands or products referenced in the software, or associated with it, are the trademarks or registered trademarks of their respective holders, and should be treated as such. Use of a term in this manual should not be regarded as affecting the validity of any trademark or service mark.

Solution

Case 1

Data Integrity Verification



Case 1: Data Integrity Verification Solution

Follow a data integrity verification plan.

The tags can all be devised using conditional computed fields where the default value is zero and the value for an exception is 1. Setting up the default value as numeric makes it easy to create the final computed field to tally up the number of exceptions in each record and generate a risk score.

DIV Objective	Steps		Results
	Milestone	Analyzer Approach	
1. Test for corrupt data and incorrect field definitions.	Identify any corrupt data or incorrect field definitions.	Table: Payments_DIV Command: Verify Fields to Verify: All Fields	The Verify command reports the first 20 errors (default) and stops processing. Errors on a date field in all or most of the records is an indication that the date format in table layout is incorrect. Edit the table layout and change the format for the Due_Date field to mm/dd/yyyy. Rerun the Verify command and there are four errors which are caused by blank dates.
2. Invoice numbers should be only 6 or 7 characters long consisting only of numbers.	Identify where the Invoice Number field is not 6 or 7 characters long and contains characters other than numbers.	Table: Payments_DIV Conditional Computed field: T2_Non_Conforming_InvoiceNo Default value: 0 Condition: NOT (Between(Length(Alltrim(Invoice_Number)) 6 7) AND Match(Format(Invoice_Number) "999999" "999999")) Filter: T2_Non_Conforming_InvoiceNo = 1	There are 8 exceptions comprised of 2 blanks and 6 invoice numbers containing 5 numeric digits. Alternate solution: Match(Format(Invoice_Number) "999999" "999999") Requirement: Check the box Exact character comparisons on the Table tab in Tools Options .
3. Vendor numbers should be 5 characters long and consist of only numbers.	Identify where the Vendor Number field is not 5 characters long and contains characters other than numbers.	Table: Payments_DIV Conditional Computed field: T3_Non_Conforming_VendorNo Default value: 0 Condition: Format(Alltrim(Vendor_Number)) <> "99999" Value: 1 Filter: T3_Non_Conforming_VendorNo = 1	There are 0 exceptions

Case 1 (Continued)

DIV Objective	Steps		Results
	Milestone	Analyzer Approach	
4. Transactions can occur on any day of the year. Every transaction must have a date.	Identify transactions with blank or invalid dates.	Table: Payments_DIV Conditional Computed field: T4_Blank_Invalid_Trans_Date Default value: 0 Condition: Trans_Date = `19000101` Value: 1 Filter: T4_Blank_Invalid_Trans_Date = 1	There are 2 exceptions. Editing the table layout with the filter still active confirms the 2 exceptions are blank dates. <div> Best practice: The IsBlank() function will identify blank dates but not invalid dates. Although using IsBlank() in this instance would yield the correct result, the best practice is to filter for dates = `19000101` which would identify a blank or invalid date. Invalid dates are dates before January 1, 1900, or dates that do not exist like April 31 or February 30. </div>
5. Transactions dates should fall within CY 2020	Identify transactions that fall outside January 1 to December 31, 2020	Table: Payments_DIV Conditional Computed field: T5_Trans_Date_Outside_2020 Default value: 0 Condition: NOT Between(Trans_Date `20200101` `20201231`) AND Trans_Date <> `19000101` Value: 1 Filter: T5_Trans_Date_Outside_2020 = 1	There are 5 records with dates in 2019
6. Product numbers should be 5 characters long, starting with two upper-case letters of the alphabet followed by 3 numbers	Identify product numbers that do not have two upper case letters followed by 3 numbers.	Table: Payments_DIV Conditional Computed field: T6_Non_Conforming_ProdNo Default value: 0 Condition: Format(Prod_No) <> "XX999" OR Length(Alltrim(Prod_No)) <> 5 Value: 1 Filter: T6_Non_Conforming_ProdNo = 1	There are 4 exceptions where the product numbers are blank.

Case 1 (Continued)

DIV Objective	Steps		Results
	Milestone	Analyzer Approach	
7. Due dates must be at least seven days after the transaction date.	Identify transactions where the due date is less than seven days after the transaction date.	Table: Payments_DIV Conditional Computed field: T7_Early_Due_Date Default value: 0 Condition: Due_Date - Trans_Date < 7 AND Verify (Due_Date) AND Verify(Trans_Date) Value: 1 Filter: T7_Early_Due_Date = 1	There are 50 exceptions
8. Amounts should be greater than zero.	Identify transactions where amounts are not greater than zero.	Table: Payments_DIV Conditional Computed field: T8_Amounts_Not_Positive Default value: 0 Condition: Amount <= 0 Value: 1 Filter: T8_Amounts_Not_Positive = 1	There are 5 exceptions
9. Risk-score each transaction.	Calculate an exception score for each record, allowing you to risk-score each transaction.	Table: Payments_DIV Computed field: T9_Score Default value: T2_Non_Conforming_InvoiceNo + T3_Non_Conforming_VendorNo + T4_Blank_Invalid_Trans_Date + T5_Trans_Date_Outside_2020 + T6_Non_Conforming_ProdNo + T7_Early_Due_Date + T8_Amounts_Not_Positive	

Case 1 (Continued)

DIV Objective	Steps		Results
	Milestone	Analyzer Approach	
Reporting			
Create exception reports showing totals by vendor and by product.	Isolate records with exceptions to a new table.	Table: Payments_DIV Command: Extract Command filter: T9_Score > 0	Result table: R_All_DIV_Exceptions This table contains 70 records totaling 3,463,834.29.
	Calculate the total number of exceptions by vendor and the amounts involved.	Table: R_All_DIV_Exceptions Command: Classify Key field: Vendor_Number Result sort options: By Size - All Items Fields to accumulate: T9_Score Amount Output: R_DIV_Exceptions_By_Vendor	Result table: R_DIV_Exceptions_By_Vendor There are 66 vendors with an exception for a total amount of 3,463,834.29. <div>Best practice: Classify calculates the percentage for only the first accumulate field. Be sure to select the field you want percentages calculated for first.</div>
	Calculate the total number of exceptions by product and the amounts involved.	Table: R_All_DIV_Exceptions Command: Classify Result sort options: By Size – All Items Key field: Prod_No Fields to accumulate: T9_Score Amount Output: R_DIV_Exceptions_By_Product	Result table: R_DIV_Exceptions_By_Product There are 43 products with an exception for a total amount of 3,463,834.29.

Solution

Case 2

Travel and Entertainment Expenses



Case 2: Travel and Entertainment Expenses Solution

Analysis Objective	Steps		Results
	Milestone	Analyzer Approach	
1. Merchant names must be entered—blanks are not acceptable except for mileage claims.	Identify where the Merchant Name is blank and the claim type is not mileage	Table: T_and_E_Claims View filter: Expense_Type <> 'Mileage' AND IsBlank (Merchant_Name)	There is 1 exception
	Output the results to a table/file	Extract the exceptions to a new Analyzer table; Export to an Excel file to provide to the business users	Result table: Test1_Missing_Merchant_Name This table contains 1 record. Excel file: T&E_Exceptions Excel worksheet: Test1_Missing_Merchant_Name
2. Individual Meals: Breakfast/lunch claims should not exceed \$25 each per day. Dinner claims should not exceed \$75 per day.	Create daily totals for individual meals	Table: T_and_E_Claims Command: Summarize Key fields: Employee_ID, Expense_Date and Expense_Type Fields to process: Expense_Amount TYPE SUM. ✓ Delete SUM from the AS parameter for the Expense_Amount. This is required in order to standardize the Expense_Amount field among all result tables in preparation for appending them together in Objective 8. Command filter: Expense_Type = "Individual"	Temp table: Temp_Meal_Claims This table contains 867 records
	Identify meals exceeding the limit	Table: Temp_Meal_Claims Conditional computed field: c_Meal_Limit Default value: 0.00 Condition1: Match(Expense_Type "Individual Breakfast" "Individual Lunch") Value1: 25 Condition2: Expense_Type = "Individual Dinner" Value2: 75 Unconditional computed field: c_Exceeds_Meal_Limit Default value: Expense_Amount - c_Meal_Limit Filter: c_Exceeds_Meal_Limit > 0	There are 234 exceptions with a total amount over meal limit of \$6,059.72. Best practice: Set the number of decimals accuracy for a numeric conditional computed field in the default value.
	Output the results to a table/file	Extract the exceptions to a new Analyzer table; Export to an Excel file to provide to the business users	Result table: Test2_Meals_Exceed_Limit This table contains 234 records. Excel file: T&E_Exceptions Excel worksheet: Test2_Meals_Exceed_Limit

Case 2: (Continued)

Analysis Objective	Steps		Results
	Milestone	Analyzer Approach	
3. No item exceeding \$100 may be claimed under the "Miscellaneous" category.	Filter for exceptions	Table: T_and_E_Claims View filter: Expense_Type = 'Miscellaneous' AND Expense_Amount > 100	There are 39 exceptions.
	Output the results to a table/file	Extract the exceptions to a new Analyzer table; Export to an Excel file to provide to the business users.	Result table: Test3_Misc_Claims_Exceed_100 This table contains 39 records. Excel file: T&E_Exceptions Excel worksheet: Test3_Misc_Claims_Exceed_100
4. Only expenses at merchants with approved SIC codes may be claimed.	Incorporate the SIC code into the claims data.	Tables: T_and_E_Claims, Merchant_Master and SIC_Codes_Master Command: Relations Alternative Command: Join (requires two joins to accomplish)	Best practice: When extracting from a parent table, remember to use the Data to Extract: Selected fields option on the Extract command and select the child fields required in output.
	Filter for exceptions	Filter: Approved = "N"	There are 417 transactions at merchants with SIC codes that are not approved.
	Output the results to a table/file	Extract the exceptions to a new Analyzer table; Export to an Excel file to provide to the business users.	Result table: Test4_TE_Claims_No_Approval This table contains 417 records. Excel file: T&E_Exceptions Excel worksheet: Test4_TE_Claims_No_Approval
5. Only 1 hotel stay per day may be claimed.	Create a file of hotel expenses grouped by employee, expense_type and date	Table: T_and_E_Claims Command: Summarize Key fields: Employee_ID, Expense_Type and Expense_Date Fields to process: Expense_Amount TYPE SUM ✓ Delete SUM_ from the AS parameter for the Expense_Amount. This is required in order to standardize the Expense_Amount field among all result tables in preparation for appending them together in Objective 8. Command filter: Expense_Type = "Hotel "	Temp table: Temp_Hotel_Claims_Summary This table contains 1,234 records
	Identify where an employee claimed for more than one Hotel expense per day	Table: Temp_Hotel_Claims_Summary View filter: Count > 1	There are 290 instances of multiple hotel charges per day.
	Output the results to a table/file	Extract the exceptions to a new Analyzer table; Export to an Excel file to provide to the business users.	Result table: Test5_Multiple_Hotel_Claims This table contains 290 records. Excel file: T&E_Exceptions Excel worksheet: Test5_Multiple_Hotel_Claims

Case 2 (Continued)

Analysis Objective	Steps		Results
	Milestone	Analyzer Approach	
6. The number of participants must be indicated. Only meals/ events with an average cost of \$75 per participant or less may be claimed.	Extract all relevant claims to a new file	Table: T_and_E_Claims Command: Extract Command filter: Match(Expense_Type "Business" "Conferences" "Department" "Entertainment" "International" "Team") AND NOT Find("Taxi" Expense_Type)	Temp table: Temp_Group_Meals_Events This file contains 4,259 records including 17 records where the number of attendees is 0.
	Identify the exceptions	Table: Temp_Group_Meals_Events Conditional computed field: c_Cost_Per_Attendee Default value: Expense_Amount Condition: Number_of_Attendees > 0 Value: Expense_Amount / Number_of_Attendees View filter: c_Cost_Per_Attendee > 75.	There are 665 instances of group claims where the cost per attendee exceeded \$75 or the number of attendees is 0. Best Practice: When creating a computed field that does division, make sure you consider if your divisor can be equal to zero. In this example, we know that some of the group events are entered with zero attendees which would cause a division by zero error. Make the computed field conditional and display the Expense_Amount when the number of attendees was not entered.
	Output the results to a table/file	Extract the exceptions to a new Analyzer table; Export to an Excel file to provide to the business users.	Result table: Test6_Group_Claims_Exceptions This table contains 665 records. Excel file: T&E_Exceptions Excel worksheet: Test6_Group_Claims_Exceptions
7. Expense claims must be reported within 90 days of the date of the expense.	Filter for exceptions	Table: T_and_E_Claims View filter: Report_Date – Expense_Date > 90	There are 67 exceptions
	Output the results to a table/file	Extract the exceptions to a new Analyzer table; Export to an Excel file to provide to the business users.	Result table: Test7_Stale_Claims This table contains 67 records. Excel filename: T&E_Exceptions Excel worksheet: Test7_Stale_Claims

Case 2 (Continued)

Analysis Objective	Steps		Results
	Milestone	Analyzer Approach	
8. Identify all employees whose share of policy violations exceeds their share of the entire claims population, in terms of the number of violations and in terms of the dollar value.	Combine the 7 exception files	Tables: Test1_Missing_Merchant_Name Test2_Meals_Exceed_Limit Test3_Misc_Claims_Exceed_100 Test4_TE_Claims_No_Approval Test5_Multiple_Hotel_Claims Test6_GroupClaimsExceptions Test7_Stale_Claims Command: Append Append options: Common fields only	Temp table: Temp_TE_All_Exceptions This table contains 1,713 records
	Calculate the percent of exceptions per employee	Table: Temp_TE_All_Exceptions Command: Classify Key field: Employee_ID Accumulate field: Expense_Amount	Temp table: Temp_Classify_Exceptions_By_Employee This table contains 14 records. Best practice: Use the Classify command when you need percent of total calculated because Classify calculates the percentages automatically.
	Calculate the percent of total expense claims per employee	Table: T_and_E_Claims Command: Classify Key field: Employee_ID Accumulate field: Expense_Amount	Table: Temp_Classify_All_Claims This table contains 15 records.
	Combine the two classified files.	Tables: Temp_Classify_Exceptions_By_Employee, Temp_Classify_All_Claims Command: Join Matched Key field: Employee_ID	Temp table: Temp_All_Claims_Join_Exceptions This table contains 14 records.
	Filter for employees with percent of exceptions higher than percent of total claims.	View filter: FIELD_PERCENTAGE2 > FIELD_PERCENTAGE	There are 5 employees whose share of exceptions is larger than their share of total expense claims.
	Output the results to a table/file	Extract the exceptions to a new Analyzer table; Export to an Excel file to provide to the business users.	Result table: Test8_Policy_Violations_Exceed_Expense_Share This table contains 5 records. Excel filename: T&E_Exceptions Excel worksheet: Test8_Policy_Violations_Exceed_Expense_Share

Solution

Case 3

Vendor Management



Case 3: Vendor Management Solution

Analysis Objective	Steps		Results
	Milestone	Analyzer Approach	
1. Identify Vendors and employees who may have the same address.	Create an address key in vendor table.	Table Vendor_Master Computed field: c_Vendor_Key Default value: SortNormalize(Vendor_Address "USPS_Address.txt")	Best practice: Use the SortNormalize() function with a substitution file for the best results when matching addresses with Join or for Duplicates testing.
	Create an address key in the employee table.	Table: Employee_Master Computed field: c_Employee_Key Default value: SortNormalize(STREET_ADDRESS "USPS_Address.txt") + Blanks(16)	
	Join the tables on the computed key fields	Tables: Vendor_Master, Employee_Master Command: Join Many to Many Matched Key fields: c_Vendor_Key c_Employee_Key Command filter: Zip_Code = Employee_Master.ZIP_CODE	Result table: R_Vendor_Employee_Address_Many_Join There are 4 records produced
2. Identify Price variations in excess of 2% above the standard price for the month of December 2020.	Group the Invoice_Line_Items by Vendor and Product	Table: Invoice_Line_Items Command: Summarize Key fields: Vendor_Number, Prod_No Fields to process: Amount TYPE SUM, Units TYPE SUM, Invoice_Date TYPE FIRST	Table: Temp_ILI_Summ_Vendor_Prodno This table contains 73,557 records.
	Recalculate the vendor unit price	Table: Temp_ILI_Summ_Vendor_Prodno Computed field: c_Vendor_Unit_Cost Default value: SUM_Amount / SUM_Units	
	Combine the summarized invoice line items file with the product master file	Tables: Temp_ILI_Summ_Vendor_Prodno, Product_Master Command: Join Type of Join: Matched primary records only Key fields: Prod_No	Result table: R_ILI_Summ_Join_Matched_Product_Master This table contains 72,599 records. Best practice: Review the Command Log after running Join to see important information. In this example, the log shows that there are 998 records bypassed representing the invoice line items where the product number did not occur in the Product_Master file.

Case 3 (Continued)

Analysis Objective	Steps		Results
	Milestone	Analyzer Approach	
2. (Continued)	Calculate the difference between the Unit_Cost and the computed vendor unit cost	Table: R_ILI_Summ_Join_Matched_Product_Master Computed field: c_Exceeds_Standard_Price Default value: (c_Vendor_Unit_Cost - Unit_Cost) * SUM_Units Condition: c_Vendor_Unit_Cost <= Unit_Cost Value: 0.00	
	Calculate the percent difference	Computed field: c_Percent_Exceeds_Standard_Price Default value: (c_Exceeds_Standard_Price * 100.00) / SUM_Amount	
	Create a file for December 2020 of products with price variations in excess of 2% above the standard price.	Command: Extract Command filter: c_Percent_Exceeds_Standard_Price > 2 AND Between(FIRST_Invoice_Date `20201201` `20201231`)	Table: Temp_Exceptions There are 54 records in December 2020 with a percent over standard price greater than 2.
	Calculate the overage by Product	Table: Temp_Exceptions Command: Summarize Key field: Prod_No Fields to process: c_Exceeds_Standard_Price TYPE SUM	Result table: R_Products_Exceeding_Standard_Price There are 24 products exceeding standard price for a total of \$269,088.84.
Extra Credit: Create a report showing the range of prices for each product	For each Product identify the highest, lowest and average Unit_Cost during the period (2020)	Table: R_ILI_Summ_Join_Matched_Primary_Product_Master Command: Summarize Key field: Prod_No Fields to process: c_Vendor_Unit_Cost TYPE MIN, c_Vendor_Unit_Cost TYPE MAX, c_Vendor_Unit_Cost TYPE AVG	Result table: R_ILI_Summ_Product_With_Min_Max_Avg_Price This table contains 91 records.

Case 3 (Continued)

Analysis Objective	Steps		Results
	Milestone	Analyzer Approach	
3. Identify instances and total amounts where the company did not avail itself of early payment discounts.	Summarize the invoice line items table.	Table: Invoice_Line_Items Command: Summarize Key Fields: Vendor_Number, Invoice_Number Fields to process: Amount TYPE SUM, Invoice_Date TYPE FIRST, Due_Date TYPE FIRST	Result table: R_ILI_Summ_Vendor_Invoice This table contains 951,706 records.
	Summarize the payments table.	Table: Payments Command: Summarize Key fields: Vendor_Number, Invoice_Number Fields to process: Net_Payment_Amount TYPE SUM, Payment_Date TYPE FIRST	Result table: R_Payments_Summ_Vendor_Invoice This table contains 951,705 records.
	Combine the two summarized tables.	Tables: R_ILI_Summ_Vendor_Invoice, R_Payments_Summ_Vendor_Invoice Command: Join Type of Join: All primary records Key fields: Vendor_Number, Invoice_Number	Result table: R_ILI_Summ_Join_Payments_Summ This table contains 951,706 records.
	Relate the Vendor_Master table to access the vendor terms	Parent: R_ILI_Summ_Join_Payments_Summ Child: Vendor_Master Key fields: Vendor_Number	Best Practice: Add a field from the Vendor_Master table to the view to make sure you can see the fields from the child table and that they are populated.
	For each vendor, calculate actual discounts taken. Assume that there were no pricing differences.	Computed field: c_Actual_Discount Default value: 0.00 Condition: Vendor_Master.Terms = "1-10-N30" AND FIRST_Payment_Date <= FIRST_Due_Date Value: SUM_Amount - SUM_Net_Payment_Amount Command: Total	The total of actual discounts is 53,466,958.89. There are 58,791 records where discounts were applied.
	For each vendor, calculate the discounts that should be taken.	Computed field: c_Calculated_Discount Default value: 0.00 Condition: Vendor_Master.Terms = "1-10-N30" AND FIRST_Payment_Date <= FIRST_Due_Date Value: SUM_Amount * .01 Command: Total	There are 64,709 records where discounts should have been applied. The total of calculated discounts is 59,163,901.38.

Case 3 (Continued)

Analysis Objective	Steps		Results
	Milestone	Analyzer Approach	
3. (Continued)	Identify the number and calculate the total amount of invoices where discounts were not taken.	Computed field: c_Discounts_Not_Taken Default value: c_Calculated_Discount – c_Actual_Discount Filter: c_Discounts_Not_Taken <> 0 Command: Total	The total of discounts not taken is 5,696,942.49. There are 5,918 records where discounts were not applied.
4. Identify instances and total amounts where late penalties were incurred but not paid.	For each vendor, calculate the actual penalty paid. Assume that there were no pricing differences.	Table: R_ILI_Summ_Join_Payments_Summ Computed field: c_Actual_Penalties Default value: 0.00 Condition: Match(Vendor_Master.Terms “N30” “N45”) AND FIRST_Payment_Date > FIRST_Due_Date Value: SUM_Net_Payment_Amount - SUM_Amount Condition 2: Vendor_Master.Terms = “1-10-N30” AND FIRST_Net_Payment_Date – FIRST_Invoice_Date > 30 Value 2: SUM_Net_Payment_Amount - SUM_Amount Command: Total	The total of penalties paid is 33,943,725.49. There are 36,780 records where penalties were paid.
	For each vendor, calculate the penalties that should have been paid.	Computed field: c_Calculated_Penalties Default value: 0.00 Condition: Match(Vendor_Master.Terms “N30” “N45”) AND FIRST_Payment_Date > FIRST_Due_Date Value: SUM_Amount * .01 Condition 2: Vendor_Master.Terms = “1-10-N30” AND FIRST_Net_Payment_Date – FIRST_Invoice_Date > 30 Value 2: SUM_Amount * .01 Command: Total	The total of penalties that should have been paid is 35,570,125.57. There are 38,289 records where penalties should have been paid.
	Identify the number and calculate the total amount of invoices where penalties were not paid.	Computed field: c_Penalties_Not_Paid Default value: c_Calculated_Penalties – c_Actual_Penalties Filter: c_Penalties_Not_Paid <> 0 Command: Total	The total of penalties incurred and not paid is 1,626,400.08. There are 1,509 records where penalties were not paid.
Create a result table showing the totals by Vendor.	Calculate the total discounts not taken and penalties incurred not paid by Vendor.	Command: Summarize Key field: Vendor_Number Fields to process: c_Discounts_Not_Taken TYPE SUM, c_Penalties_Incurred_Not_Paid TYPE SUM Filter: c_Discounts_Not_Taken <> 0 OR c_Penalties_Not_Paid <> 0	Result table: R_Discount_Penalty_Exceptions_By_Vendor There are 71 vendors with discounts not taken and/or penalties incurred and not paid.

Solution

Case 4

Accounts and Terminations



Case 4: Accounts and Terminations Solution

It is necessary to join the two tables using the Join command in order to have the hire and termination information together with the passive directory activation and deactivation information in one table. To do this, we must create a key field in each table since a common field to use as the Join key fields does not already exist.

It is important to understand if there are duplicate key fields when using the Join command. Part of preparing the tables requires testing for duplicate key fields. We find that there are duplicate keys in both tables in this case, since it's possible for two similar names to generate the same key. Thus, an HR record could be joined to an unrelated PD account.

To reduce the false positives, we made one assumption: All requests would eventually be addressed within 5 days, but not always within the service level agreement. This is why the condition for the join specifies 5 days or fewer in the solution. There may still be false matches, but their occurrence would be greatly reduced.

Objective	Steps		Result
	Milestone	Analyzer Approach	
Prepare			
Prepare HR_Data table	Create the key field	Table: HR_Data Computed field: c_HR_Key Default value: Substr(Lower(Substr(First_Name 1 1) + Alltrim>Last_Name))1 15)	
	Check for duplicate keys	Command: Duplicates Key field: c_HR_Key Output: Data	R_HR_Duplicate_Keys This table contains 7 records
Prepare PD_Accounts_List table	Create the key field	Table: PD_Accounts_List Computed field: c_PD_Accounts_Key Default value: Substr(Include(User_Name "a~z") 1 15)	
	Check for duplicate keys	Command: Duplicates Key field: c_PD_Accounts_Key	R_PD_Duplicate_Keys This table contains 39 records
Blend the data from the HR and PD_Accounts into one file.	Combine the files	Tables: HR_Data, PD_Accounts_List Command: Join Key fields: c_HR_Key, c_PD_Accounts_Key Join type: Many to Many Matched Command filter: Between (PD_Accounts_List.Account_Deactivate_Request - Term_Date_Time `e00` `e5/00`)	R_HR_Many_Join_PD_Accounts This table contains 250 records

Case 4 (Continued)

Objective	Steps		Result
	Milestone	Analyzer Approach	
Analysis			
The Identity Management team should create the new PD account within 18 hours and 30 minutes of the employee being added in the HR System.	Calculate time between account activation and start date. Create a file of exceptions.	Table: R_HR_Many_Join_PD_Accounts Computed field: c_Gap_Hired_Account_Create Default value: Account_Created - Start_Date_Time Command: Extract Command filter: c_Gap_Hired_Account_Create > `e18:30`	Result table: R_HR_PD_New_Account_Exceptions This file contains 93 records.
The Identity Management team should receive a deactivation request within 20 hours of the employee's status changing to "Terminated" in the HR system.	Calculate the time between employee terminate date in HR system and IM team receive deactivate request. Create a file of exceptions.	Table: R_HR_Many_Join_PD_Accounts Computed field: c_Gap_Term_Deactivate Default value: Account_Deactivate_Request - Term_Date_Time Command: Extract Command filter: c_Gap_Term_Deactivate > `e20`	Result table: R_HR_PD_Term_Request_Exceptions This table contains 47 records.
The employee PD Account should be disabled within 6 hours and 45 minutes after receipt of the deactivation request.	Calculate the time between receipt of deactivation request and account deactivation. Create a file of exceptions.	Table: R_HR_Many_Join_PD_Accounts Computed field c_Gap_Request_Deactivated Default value: Account_Deactivated - Account_Deactivate_Request Command: Extract Command filter: c_Gap_Request_Deactivated > `e06:45`	Result table: R_HR_PD_Deactivate_Exceptions This table contains 103 records.

Solution

Case 5

Automation



Case 5: Automation Solutions

The automation solution procedures are stored with the course files in a hidden folder. You must change your Analyzer Options to view the solutions in Analyzer.

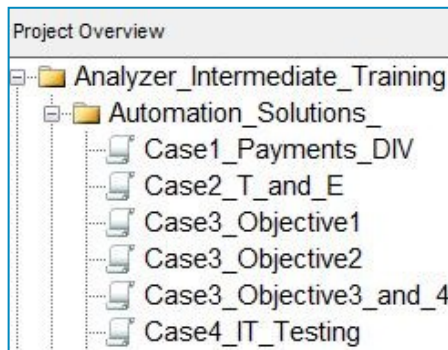
Step-By-Step

Select **Tools > Options** from the menu.

On the **Interface tab** check the box to **Show hidden overview Items**.

Click **OK**.

You will see the Automation_Solutions_ folder containing 6 procedures. Click on a procedure to display the contents in the procedure editor tab in the display area.



Please refer to the topic “*Run a procedure*” in the Analyzer Intermediate Training Workbook for information on how to run procedures.