Preparing Work-Related Injuries and Illness Data for MiTracking

Process Overview and User's Guide

Prepared for:

Kenneth D. Rosenman, MD
Principal Investigator
Michigan State University
East Lansing, MI

Project Closeout
June 19, 2019

Table of Contents

Chapter 1   Introduction

This project was carried out by investigators from two groups, one at Michigan State University (MSU) and one at the Michigan Department of Health and Human Services (MDHHS).

A long-term contractor to the State of Michigan and MDHHS, Kunz, Leigh & Associates (KL&A) was contracted to aid in technical aspects of the project. The project harnesses the MiTracking system that was developed by KL&A, which is currently in place at MDHHS, in order to provide access to the publicly available data.

The work completed by KL&A developed a process for identifying data matches, calculating the final de-identified workers' compensation rates, and preparing files to be loaded on the MiTracking web site.

This document describes the steps used by KL&A. This information, combined with the development deliverables provided to MSU during project closeout, can be used to create new data files as data becomes available for future years.

Chapter 2   Initial Matching Process

The initial matching process combines two datasets: One set contains wage loss claims from the State of Michigan workers' compensation office; the second set contains quarterly company data. The purpose of matching is to align the company records from each dataset. Subsequent years datasets are combined to build a comprehensive automated matching process based off of the previously verified matches.

2.1 Create the Database and Populate Data

Complete the following steps using the scripts provided in MSU_WC_Database_Data_Scripts.zip.

1. Run 01-msu-wc.Database.sql first to create the database.

Run the remaining scripts. The order does not matter:

2. dbo.Emp_2014_Condensed.Table.sql

3. dbo.Emp_2014_Q1.Table.sql

4. dbo.Emp_2014_Q2.Table.sql

5. dbo.Emp_2014_Q3.Table.sql

6. dbo.Emp_2014_Q4.Table.sql

7. dbo.Emp_2014.Table.sql

8. dbo.Emp_2015_Condensed.Table.sql

9. dbo.Emp_2015_Q1.Table.sql

10. dbo.Emp_2015_Q2.Table.sql

11. dbo.Emp_2015_Q3.Table.sql

12. dbo.Emp_2015_Q4.Table.sql

13. dbo.Emp_2015.Table.sql

14. dbo.Emp_2016_Condensed.Table.sql

15. dbo.Emp_2016_Q1.Table.sql

16. dbo.Emp_2016_Q2.Table.sql

17. dbo.Emp_2016_Q3.Table.sql

18. dbo.Emp_2016_Q4.Table.sql

19. dbo.Emp_2016.Table.sql

20. dbo.Emp_All_Years.Table.sql

21. dbo.fips.Table.sql

22. dbo.Levenshtein.UserDefinedFunction.sql

23. dbo.Match_2014_Confirmed.Table.sql

24. dbo.Match_2014_FEINOnly.Table.sql

25. dbo.Match_2014.Table.sql

26. dbo.Match_2015_Confirmed.Table.sql

27. dbo.Match_2015_FEINOnly.Table.sql

28. dbo.Match_2015.Table.sql

29. dbo.Match_2016_Confirmed.Table.sql

30. dbo.Match_2016_FEINOnly.Table.sql

31. dbo.Match_2016_Manual.Table.sql

32. dbo.Match_2016.Table.sql

33. dbo.Match_All.Table.sql

34. dbo.MaxOf.UserDefinedFunction.sql

35. dbo.MinOf.UserDefinedFunction.sql

36. dbo.Other_Filter_Naics.Table.sql

37. dbo.Other_Filter_Nora_Sectors.Table.sql

38. dbo.UI_Acct_Employer.Table.sql

39. dbo.WC_2014.Table.sql

40. dbo.WC_2015.Table.sql

41. dbo.WC_2016.Table.sql

2.2 Prepare Data for New Year

When the claims and QCEW data files are ready for a new year the data can be imported and prepared using the following steps. All scripts will be run on the database created in section 2.1.

1. Import the quarterly data using SQL Server Management Studio data import utility. A table should be created for each quarter of the new year. For example, if adding data for 2017 the following tables should be created and populated: Emp_2017_Q1, Emp_2017_Q2, Emp_2017_Q3, Emp_2017_Q4.

2. Open 01-Add-New-QCEW-File.sql and update all references from the previous year to the new year. For example, if the file were previously configured for Emp_2016, the references should be updated to Emp_2017 for the new year. Save changes and run the script.

3. Open 02-Combine-QCEW-Quarters.sql and update all references from the previous year to the new year. Save changes and run the script.

4. Open 03-Combine-All-Matches.sql to add another insert statement for the new year of data. Save changes and run the script.

5. Open 04-Combine-QCEW.sql to add another insert statement for the new year of data. Save changes and run the script.

6. Open 05-QCEW-Compression.sql and replace references to the previous year with the current year. Save changes and run the script.

7. Open 06-Matching-Script.sql and replace references to the previous year with the current year. Save changes and run the script.

8. Open 07-Insert-Previous-Company-Matches.sql and update to the most recently completed year. For example, if the new year being added is 2017, the script should be run for Match_2016.

Chapter 3   Secondary Matching Processes

The results of the automated matching process are exported to an Access Database for review and verification. The MSU reviewer should sign off on any new, automated matches. Claims without a matching company can be manually matched. Historical matches are automatically verified but may be reviewed if desired.

3.1 Prepare the Access file for MSU reviewer

1. From the msu-wc database, export all records from the Match_20xx table for the current year. You should generate an Excel file that matches the database table.

2. In Microsoft Access, open the provided file named MSU-WC-QCEW-Reviewer.accdb. Verify that the Manual_Match and Match_Year tables are empty. If records remain from a previous year you should delete the rows before continuing.

3. Import the Match_20xx file generated in the step 1 and append the records to the Match_Year table.

   In Access, go to the External Data tab and select New Data Source. Choose the From File option, then select Excel. From the Get External Data dialogue window, choose to Append a copy of the records to the table Match_Year.

4. Run the macro named RunPrepareNewYear. This will first delete rows from Manual_Match if any remain from a previous year. Next a record will be created in Manual_Match for each distinct WC_Seq number. Finally, if the imported Match_Year data contains records that have a previous year match the MSU Reviewed status will be set to true.

The Access tool is now ready for the MSU reviewer to verify matches. If multiple potential matches were found for a record, the reviewer must select a single option or set a manual match. If no potential matches are returned, the reviewer should set a manual match.

3.2 Reviewing Suggested Matches

The MSU reviewer should open the Access file that was prepared in the previous step. All of the reviewer's work is performed from a single form. There are three queries that can be run to report on progress.

1. With the Access file open, double click the Form titled 01-WC/QCEW Match Reviewer.

2. Records are returned in the order of most potential matches to fewest potential matches. You may use the record navigation tools or filters to move between individual records.

3. On a record, the desire would be to have one item checked in the Potential Matches list.

   If none of the suggested employer data matches the worker info, a manual match can be made. Enter a UI Account number to select an existing employer in the database or enter a NAICS code. The NAICS code should be 2 to 6 digits.

4. When a match has been identified or manually entered, the MSU Reviewed checkbox should be marked. This indicates the record has been completed.

3.3 Import Reviewer Results Back to Database

Once the MSU reviewer has completed their manual matching process, the results must be inserted into the SQL database.

1. In SQL Server Management Studio, use the SQL Server Import Tool to import the Match_Year table from the Access database. The SQL table should be named Match_20xx_Confirmed replacing 20xx with the current year.

2. Then, import Manual_Match into SQL as Match_20xx_Manual, replacing 20xx with the current year.

3. Open 08-Manual-Match-Import.sql and update the references to the previous year to the new year. Save changes and run the script.

3.4 Helper Reports

In the Access tool, the Queries section contains three reports that can be run:

Not Reviewed, No Matches Assigned
Returns records that do not have the MSU Reviewed flag set.

Reviewed, Manual Match Assigned
Returns records where the MSU Reviewed flag is set and a manual match was made.

Reviewed, No Matches Assigned
Returns records that have been marked as reviewed by mistake.

Chapter 4   Rate Calculations

The next steps are to modify the denominators and calculate FTE modifiers. Per the recommendation of NIOSH, data from the US Census website should be used for calculating the denominator of the rate calculation for Age and Gender.

4.1 Obtain Census Data

The following data is obtained using the LED extraction tool:

1. Use a web browser to open
    https://ledextract.ces.census.gov/static/data.html

2. In the geography level column on the left, select Michigan. Then, select all counties and continue to Firm Characteristics. Select all industries on this page.

    Navigate to Worker Characteristics. Check all the boxes in both the Sex column and Age column.

    Continue to the Indicators section. Check the Emp and EmpEnd columns.

    Continue to the Quarters section. Check all the quarters for the year(s) you are interested in.

    Continue to the Summary and Export section. Submit your request and wait for results. Once it's done, download the file as a CSV.

3. Copy all the data from the CSV file into the Raw Data sheet on the LEDCalculator spreadsheet.

4. Return back through the LED Extraction tool, repeating steps 1 – 3. but in the Firm Characteristics section, select NAICS 3-digit Subsectors. From there, select the NAICS codes for 211 only.

5. Run the tool and download this file as a CSV as well.

4.2 Import Census Data

If this is the first time processing the LED data the QWI table will need to be created:

1. Run 09-CreateQWITable.sql

Next, import both of these CSV files into the QWI table in the database. Do not add duplicate records to this table. If you're unsure if duplicate data will be inserted into this table, run TRUNCATE TABLE QWI on the database to remove all the data from the table first, then it should be safe to insert a fresh copy of the data from the CSV files you downloaded earlier.

2. In SQL Server Management Studio, open the Import Data tool.

3. Change the data source to Flat File Source

4. Select the QWI file

5. Change the Text qualifer from <none> to "

6. Switch to the Advanced section

7. Click the Suggest Types... button

8. Uncheck the Suggest smallest integer data type box

9. Uncheck the Suggest smallest real data type box

10. Check the Pad string columns box

11. Set Percent padding to 1000

12. Continue through the import process as normal. When you get to the Select Source Tables and Views screen, make sure to change the destination table to the QWI table instead of the qwi_lotsofcharactershere table.

If you run into an error while importing, the tool probably reset your Text Qualifer back to <none>. Resetting that back to " should help.

13. Run 10-MiTrackingDenominatorModifiers.sql script to generate the MiTracking_Modifiers table using the data in the QWI table. This script is safe to run multiple times, as it will remove all data in the MiTracking_Modifiers table before inserting data into it.

FTE Ratio Calculation

Rates must be adjusted to account for part-time employees. This is accomplished using data provided by Martha Jones. The rates vary from year to year. When a new year is calculated, the All Years modifier is also regenerated.

1. Open the FTE Ratio Calculator spreadsheet.

2. Insert a new column for the new year to the left of each Sum column.

3. Populate the FTE and EE totals for the new year. The data should align with the appropriate county code row.

4. Insert a new column to the left of the Combined column.

5. Copy the calculations by sector into the new column and verify the forumlas are updated to use the new year's data.

6. Save the spreadsheet changes. The ratios will be used in later steps.

7. Open the FTE Ratio Calculator – 3 Digit NAICS spreadsheet

8. Insert new columns to the left of each Sum column.

9. Populate the FTE and EE totals for the new year. The data should align with the appropriate NAICS code row.

10. Save the spreadsheet changes. The ratios will be used in later steps.

Chapter 5   Preparation of MiTracking Files

All updates have now been made to generate the MiTracking files. If any of the Other Filters options on MiTracking are changed from previous years (e.g. Age, Gender, Industry Type, Company Size, or Injury Type), or if any of those options become incompatible with one another (e.g. arm strain and gender), the 11-Infrequently-Updated-Tables.sql script should be run.

After these scripts are run, a number of csv files will be created. These files can be provided to the MiTracking team for upload. The following files will be created plus an additional two files for each new year that has been added:

- MSU injuries fte rates all.csv

- MSU injuries fte rates 2016.csv

- MSU injuries fte rates 2015.csv

- MSU injuries fte rates 2014.csv

- MSU injuries count all.csv

- MSU injuries count 2016.csv

- MSU injuries count 2015.csv

- MSU injuries count 2014.csv

5.1 Count All Years

Make sure the Match_All table has been filled with any data you want to be included.

1. Open 12-mitracking-count all years.sql.

2. Update the @YearString variable. Save and run the script.

3. Set output to CSV and generate the file.

4. Remove the (X rows affected) statements from the start and end of the generated file.

5.2 Count by NAICS All Years

This script outputs extra rows with 1 and 2 digit NAICS codes, where we don't have anything more than that in the data. MiTracking will ignore these rows when they import the file, so it's safe to leave them in the output.

1. Open 13-mitracking-count by naics all years.sql.

2. Update the references to 2014 – 2016 to include the new year. Save and run the script.

3. Set output to CSV and generate the file.

4. Remove the (X rows affected) statements from the start and end of the generated file.

5.3 Count by Single Year

This script should be run once for each individual year to be generated. Repeat all steps for each year.

1. Open 14-mitracking – count single year.sql.

2. Update the year references to the desired year. Save and run the script.

3. Set output to CSV and generate the file.

4. Remove the (X rows affected) statements from the start and end of the generated file.

5. Repeat all steps for each year.

5.4 Rate by NAICS All Years

The script outputs extra rows with 1 and 2-digit NAICS codes, where we don't have anything more than that in the data. MiTracking will ignore these rows when they import the file, so it's safe to leave them in the output.

1. Open 15-mitracking – rate by naics all years.sql.

2. Update the references to 2014 – 2016 to include the new year. Save and run the script.

3. Set output to CSV and generate the file.

4. Remove the (X rows affected) statements from the start and end of the generated file.

5.5 Rate Per FTE All Years

The FTE ratios generated by the spreadsheet calculator are required in this step.

1. Open 16-mitracking – ratePerFTE all years.sql.

2. Update the FTE ratios to include new years of data. Save and run the script.

3. Set output to CSV and generate the file.

4. Remove the (X rows affected) statements from the start and end of the generated file.

5.6 Rate Per FTE Single Year

The scripts for existing years can be run without any modifications.

1. Run 17-mitracking – ratePerFTE2014.sql

2. Set output to CSV and generate the file.

3. Remove the (X rows affected) statements from the start and end of the generated file.

4. Repeat steps 1-3 for 18-mitracking – ratePerFTE2015.sql.

5. Repeat steps 1-3 for 19-mitracking – ratePerFTE2016.sql.

Create a new script for each new year of data. The FTE ratios generated by the spreadsheet calculator are required in this step.

6. Copy 19-mitracking – ratePerFTE2016.sql and rename for the appropriate year to add. Open the new file.

7. Update the @Year variable to the new year being added.

8. Update the FTE ratio modifiers for the new year. Save and run the script.

9. Set output to CSV and generate the file.

10. Remove the (X rows affected) statements from the start and end of the generated file.

5.7 Submit Files

The csv files are ready to send to MiTracking for installation.

Appendix A: Data Dictionary

Table descriptions

WC_2014, WC_2015, WC_2016
These tables contain the raw workers' compensation claims.

Emp_2014_Q1, Emp_2014_Q2, Emp_2014_Q3, Emp_2014_Q4
Emp_2015_Q1, Emp_2015_Q2, Emp_2015_Q3, Emp_2015_Q4
Emp_2016_Q1, Emp_2016_Q2, Emp_2016_Q3, Emp_2016_Q4
These tables contain the raw company data from the QCEW files.

Emp_2014, Emp_2015, Emp_2016
These tables combine data from the quarterly tables into a single table by UI_ACCT and RPT_UNIT

Emp_2014_Condensed, Emp_2015_Condensed, Emp_2016_Condensed
These tables combine data from the quarterly tables into a single table, to be used in the matching process. Each UI_ACCT number has a single row, and a NAICS code is assigned to it based on the number of employees working in each NAICS code for that company. The compressionRule column indicates how the NAICS code was chosen for that row.

> compressionRule = 1: 75% of employees had the same 6 digit NAICS code
>
> compressionRule = 2: 75% of employees had the same first 4 digits in their NAICS code
>
> compressionRule = 3: 75% of employees had the same first 3 digits in their NAICS code
>
> compressionRule = 4: 75% of employees had the same first 2 digits in their NAICS code
>
> compressionRule = 5: The 6 digit NAICS code has the most employees, but fewer than 75%.

Emp_All_Years
Contains all the data from Emp_2014, Emp_2015, and Emp_2016 combined into a single table.

Fips
Contains a mapping from ZIP code to most likely FIPS code.

Match_2014, Match_2015, Match_2016

Contains the output of the matching algorithm, combining the matching WC_XXXX and Emp_XXXX_Condensed tables. A few columns are added to assist with debugging. The distance column shows the Levenshtein distance between all the matching columns (only on Levenshtein matches), the confirmedMatch column indicates 1 if we believe this to be a high-confidence match, and the matching_algorithm column shows the algorithm that was used for this specific match.

Old Match: Employer match was found based on previous year matches, from the UI_Acct_Employer table.

Levenshtien5 ñ 1: Using the Levenshtein algorithm, this record was matched by matching employer name, FEIN, street address, city, state, and zip code.

Levenshtien5 ñ 2: Using the Levenshtein algorithm, this record was matched by matching FEIN, street address, city, state, and zip code.

Levenshtien5 ñ 3: Using the Levenshtein algorithm, this record was matched by matching employer name, and FEIN.

Soundex ñ 1: Using the Soundex algorithm, this record was matched by matching employer name, FEIN, street address, city, state, and zip code.

Soundex ñ 2: Using the Soundex algorithm, this record was matched by matching employer name, and FEIN.

Soundex ñ 3: Using the Soundex algorithm, this record was matched by matching FEIN, street address, city, state, and zip code.

Soundex ñ 4: Using the Soundex algorithm, this record was matched by matching employer name, and requiring an exact match on street address, city, state, and zip code.

Soundex ñ 5: Using the Soundex algorithm, this record was matched by matching street address, city, state, zip code, and requiring an exact match on employer name.

Soundex ñ 6: Using the Soundex algorithm, this record was matched by matching employer name, street address, city, state, and zip code.

FEIN – Only: This record was matched exclusively based off of FEIN.

Match_2014_FEINOnly, Match_2015_FEINOnly, Match_2016_FEINOnly

Contains the FEIN - Only matches from the matching algorithm.

Match_2014_Confirmed, Match_2015_Confirmed, Match_2016_Confirmed
Contains the resulting output of the matching algorithm, after a manual review from MSU using the Access tool.

Match_All
Combines the Match_2014_Confirmed, Match_2015_Confirmed, and Match_2016_Confirmed tables.

UI_Acct_Employer
Contains historical matches of employer information from WC tables to UI_Acct on Emp tables.

Other_Filter_Naics, Other_Filter_Nora_Sectors
Contains every possible combination of MiTracking other filters, to be used when generating MiTracking files to generate 0 rows.

Functions

Levenshtein(left nvarchar(4000), right nvarchar(4000), maxDistance int)
Computes the levenshtein distance of the left and right strings, up to maxDistance. If maxDistance is exceeded, returns null.

MinOf(val1 int, val2 int)
Returns the minimum of the two values. If either value is null, returns the other value. If both values are null, returns null.

MaxOf(val1 int, val2 int)
Returns the maximum of the two values. If either value is null, returns the other value. If both values are null, returns null.

Appendix B: Development Deliverables

The following development resources have been provided to MSU for the completion of the steps outlined above:

Presentations.zip
Contains the PowerPoint files from meetings with NIOSH and the Workers' Compensation office.

MSU_WC_SQL_Scripts.zip
Contains scripts numbered 01 – 19 referenced in this manual.

MSU_WC_Database_Data_Scripts.zip
Contains all files required to generate the msu-wc database and populate data for the years of 2014 – 2016. Referenced in Chapter 2 of this manual.

MSU_WC_QCEW_Reviewer.accdb
Access database for MSU reviewer to verify matches and set manual matches.

MiTracking Files 2014 – 2016.zip
Final version of the files uploaded to MiTracking.

FTE Ratio Calculator.xlsx

FTE Ratio Calculator – 3 Digit NAICS.xlsx