Supplemental Digital Content 1

eTable. Raw Primary Data Fields for the Controlled Substances Monitoring Database Patient and Prescription SQL Tables Available for Study

Raw Field Name[a]	Field Type (SQL)	Field Definition	Field attributes (Expected Values*, codes)	Foreign Key	Primary Key	Identifier in Matching Algorithms
PATIENT TABLE						
PatientID	Int	Patient Identification Number (not a unique ID, may be duplicates for a patient entity)	Number		X	
AnimalName	varchar (30)	Name of animal (if veterinary patient)	Text Used if required by the PMP for prescriptions written by a veterinarian and the pharmacist has access to this information at the time of dispensing the prescription.			
City	varchar(50)	Patient's city of residence	Text			X
DateOfBirth	datetime	Patient's date of birth (at record creation)	Datetime			X
FirstName	varchar (50)	Patient's first name	Text			X
LastName	varchar (50)	Patient's last name	Text			X
MiddleName	varchar (50)	Patient's middle name	Text			X
GenderID	tinyint	Code indicating the patient's sex	0 = Unknown 1 = Female 2 = Male 6 = Unknown			
SpeciesCode	tinyint	Used to differentiate a prescription for an individual from one prescribed for an animal.	01 = Human 02 = Veterinary Patient			

State	varchar(50)	Patient’s state of residence	Text			
Street	varchar(100)	Patient’s street address	Numbers and text This is patient’s most recent street address in CSMD, not necessarily contemporaneous with the prescription fill record.			X
Street2	varchar(100)	Patient’s street address (continued)	Numbers and text Often contains notes rather than address.			X
Zip	varchar(9)	Patient’s zip code	5 or 9 digits Some zips listed as 99999 or 00000 This is patient’s most recent zip in CSMD, not necessarily contemporaneous with the prescription fill record.			X
CreateDate	smalldatetime	Date and time patient record was created	Datetime			
UpdateDate	smalldatetime	Date and time patient record was updated	Datetime			
NamePrefix	varchar (50)	Patient’s name prefix	Text			
NameSuffix	varchar (50)	Patient’s name suffix	Text			
customerID	varchar (20)	Identification number for the patient as indicated as indicated by IDqualifierID, not required.	Number or alpha-numeric			X
IDqualifierID	tinyint	Code to identify the type of ID in customerID	01 = military ID 02 = state issued ID 03 = unique system ID 04 = permanent resident card 05 = passportID 06 = driver’s license 07 = social security number 08 = tribal ID 99 = other type of ID			

PRESCRIPTION TABLE						
PrescriptionID	Int	Prescription Identification Number	Number		X	
PatientID	Int	Patient Identification Number (not a unique ID, may be duplicates for a patient entity)	Number	X		
PharmacyID	int	Pharmacy Identification Number (not a unique ID, may be duplicates for a pharmacy entity)	Number	X		
PractitionerID	int	Practitioner/Prescriber Identification Number (not a unique ID, may be duplicates for a practitioner entity)	Number	X		
DateFilled	date	Date prescription filled	Date			
DateRxWritten	datetime	Date prescription written	Datetime			
DaysSupply	smallint	Estimated number of days the prescription will cover	Non-negative number Generally, days' supply is calculated by the pharmacists/pharmacy technician dividing the maximum amount of the medication used in 1 day by the dispensed amount.			
CreateDate	smalldatetime	Date and time prescription record was created.	Datetime			
Updatedate	smalldatetime	Date and time prescription record was updated.	Datetime			
PartialFill	int	Indicates if prescription is only a partial fill	00 = Not a Partial Fill 01 = First Partial Fill Note: For additional fills per prescription, increment by 1. So the second partial fill would be reported as 02, up to a maximum of 99.			

			This field is used when the quantity is less than the metric quantity per dispensing authorized by the prescriber. This dispensing activity is often referred to as a split filling.			
Refillcode	varchar(3)	Number of the fill of the prescription	0 indicates original dispensing, 01-99 is the refill number			
PaymentPlanTypeID	tinyint	The form of payment for each prescription	1 = Private Pay 2 = Medicaid 3 = Medicare 4 = Commercial Insurance 5 = Military Insurance and VA 6 = Worker’s Comp. 7 = Indian Nations 99 = Other (according to operations, indicates coupons/discount cards)			
Quantity	Decimal(11,4)	Number of metric units dispensed	Non-negative number			
DrugDosageUnitID	Tinyint	Identifies the unit of measure for the quantity dispensed	01 = Each (used to report solid dosage units or indivisible package) 02 = Milliliters (ml) (for liters adjust to the decimal milliliter equivalent) 03 = Grams (gm) (for milligrams adjust to the decimal gram equivalent)			
NDCnum	varchar(15)	National Drug Code number for the prescribed drug (11 digit version)	11 digit number Allows for identification of drug in drug classification references.			
PHADEANum	varchar(15)	Identifier assigned to the pharmacy by the Drug Enforcement Administration.	Non-zero alpha numeric			
PHANABPNum	varchar(15)	Identifier assigned to pharmacy by the National Council for Prescription Drug	Non-zero alpha numeric			

		Programs				
PHANPINum	varchar(15)	Pharmacist’s National Provider Identifier number	Non-zero alpha numeric			
PRADEANum	varchar(15)	Identifying number assigned to a prescriber or an institution by the Drug Enforcement Administration.	Non-zero alpha numeric			
PRANPINum	varchar(15)	Practitioner’s National Provider Identifier number	Non-zero alpha numeric			
NumOfAuthRefills	Int	The number of refills authorized by the prescriber.	Non-zero number			
RxOriginCodeID	Tinyint	Code indicating how the pharmacy received the prescription	01 = Written Prescription 02 = Telephone Prescription 03 = Telephone Emergency Prescription 04 = Fax Prescription 05 = Electronic Prescription 99 = Other			
*Expected values, not necessarily actual values in the database due to data entry errors, lack of standardizations, and changes in type of information over time without updates to available data documentation.						

Supplemental Digital Content 2, eAppendix

We have created a detailed report describing our "in-house" data cleaning methods for PDMP data (Golladay M, Nechuta S. Lessons Learned: Deep Cleaning Procedure Design for Name Variables in the Tennessee CSMD. Found here: https://www.tn.gov/content/dam/tn/health/documents/opioid_response/CSMDNameCleaningReport.pdf). Below we include selected helpful sections from this report.

Key Elements of General Deep Cleaning Procedures

In our cleaning process, we apply three overarching philosophical principles. First, we do not overwrite original variables; all original information in the record must be preserved intact. Second, our procedure is inherently data-driven; making assumptions about data structure without reading the dataset is inefficient. Third, we practice conservative identification; when a record is ambiguous as to classification type, the default assumption is that the record is a person.

I. Record Flagging

In Tennessee, the CSMD tracks both human and non-human records, and one of our cleaning objectives is to classify records as human, animal (pet), business, or dummy records. We endorse the use of record flags in not only this classification process but also in identifying how individual records are being cleaned. This has two main purposes. First, it allows the internal team responsible for data cleaning and management to keep track of which records have already been cleaned while developing their cleaning algorithm. Second, it allows us to track cleaning issues in the database. This is critical for providing feedback to teams developing front-end solutions to data issues; by identifying which dispensers are making which specific errors, training to improve data entry could be implemented for those who have greatest need of these services.

Our procedure makes use of three main CSMD patient record classification flags *PetFlag*, *BusinessFlag*, and *DummyFlag*. An 'unflagged' record using this system, then, is a person. Additionally, we use placeholder flags as we are cleaning subsets of the database – for example, when cleaning a record of a suffix or prefix incorrectly placed in FirstName, we use FirstSuf as a flag that this record has been adjusted. These placeholder flags are generally dropped at the end of the cleaning procedure, as they are not particularly useful in the cleaned dataset. The only exception to this is for researchers analyzing the cleaning process itself, the placeholder flags can be used to generate descriptive statistics.

II. Data Profiling

The most data-driven piece of the cleaning process is the proper identification of keyword search terms. Because of this, keyword identification is one of the more time-consuming pieces of the procedure as well. As a first step, we recommend approaching the dataset in subsets of roughly 10,000 to 20,000 records at a time. Because reading the dataset is critical in not only identifying terms but also making sure that the algorithm is cleaning as it should, keeping each subset readably small is vital. This isn't the kind of routine where we can look at the first ten or twenty lines and see that it's doing what it should; because each record can be quite unique in its challenges, the whole set must be read.

must always be prepared either to edit prior routines or to simply add new sections as additional records are detected as the existing cleaning algorithm is implemented.

Keyword Search Techniques for Patient Names

There are several useful initial ways to subset the data to find useful search terms. These include but are not limited to:

- searching for special characters
- searching for entries containing numbers
- searching for multi-word entries
- searching for 'long' entries[3]
- searching for entries that contain no vowels

Many standard sources on data cleaning recommend stripping special characters and numbers out of entries as a matter of course, much as they recommend stripping out multiple spaces between words. We found that this is counterproductive for our patient name fields in the CSMD. Many pet and business records can be identified *using* special characters as a flag, as discussed below, and the numbers are often dates or other identification pieces that may be useful for entity resolution at a later time. While we do eventually remove special characters from the names and move numbers into a separate variable, we initially leave them in the records as searchable terms. Multi-word and 'long' entries are almost all either business records or records that have patient notations, and these searches are particularly useful for identifying further keywords that will be used to parse the records correctly.

A specific and surprisingly useful technique that we identified in the CSMD involved searching for entries which have a second or third word containing no vowels.[4] If the 'word' is longer than two letters, we can conclude that it is not a set of initials and must therefore be an abbreviation (e.g. CHK, MTM, RMR). Since it isn't feasible that we would be aware of every possible abbreviation, running a 'no vowel' search can help us pick up those abbreviations that we weren't aware of previously. We tended to find this search especially useful after parsing the first word of the name into a separate column – more discussion of parsing techniques below – but it's useful as one is finishing a routine and looking for any stray, uncleaned records.

Even though database-specific searches need to be implemented on the 'first pass' of cleaning any large dataset like the CSMD, we can also give some indications of very common keywords found in the CSMD. First, we focus on search words for human records that need parsing. Second, we discuss keywords that are helpful in record classification.

[3] For the CSMD, we defined a 'long' entry as one having more than 20 characters, including spaces

[4] These entries are consistently formatted as *NAME NOTATION*, as in *'MARY MTM'* or *'JOHN RMR.'*

Keyword Specifics: Prefixes and Suffixes

Possibly the most obvious phrases that need to be cleaned from our name variables are common **prefixes** and **suffixes**. These include, but are not limited to:

- Formal titles: Mr, Mrs, Ms, Miss
    - NOTE: Many usages of 'Miss' are in fact pets
- Academic titles and suffixes: Dr, MD, DVM, DDS, PhD, DC
    - NOTE: Some use of these titles can indicate offices instead of persons
- Religious titles: Sister/Sr[5], Brother/Br, Father/Fr, Rev, Rabbi
    - NOTE: 'Sister' is a possible first name, especially in the Southeastern US
- Common suffixes: Jr, Sr, III, IV, etc
    - NOTE: Watch out for all variations of these – we found III, 3rd, and THIRD in the CSMD

Keyword Specifics: Numbers in Name Fields

We have previously discussed the usefulness of searching for **numbers** in our name variables, but here we highlight in particular that the number of digits in the record can be helpful as well. In the CSMD, we found that most entries that only contained one digit were actually a typo, whereas entries with multiple digits usually contained a notation. We used this fact to create appropriate subsets in our cleaning algorithm.

Keyword Specifics: Patient Notations

The final set of important keywords used to parse human records are intended to identify **notations**. Dispensers commonly use FirstName in the CSMD to make record notations, so we see records such as:

- JANE CHECK DOB
- JOHN CID
- MARY MTM 1/3/14

We have decided to retain these notations in a separate variable, but they must be identified first. Our multi-word and long searches have proven useful in finding these terms, and we have also learned two additional techniques to help us find true instances of these notations without having to deal with too much over-identification. First, we use trailing and leading blanks to find terms; for example, we search for '_ID_' instead of just 'ID.' This excludes names like SIDNEY, which happen to contain the letter i beside the letter d. Second, we attempt to identify the most common keyword in several phrases to keep our code efficient. For example, if we look at the following phrases, we see a pattern:

- SEE ID
- CHECK ID
- MUST ID
- ID ONLY

All of these phrases will be identified by searching for 'ID.' So instead of running four separate searches, we only run one and then adjust our parsing algorithm accordingly.

[5] In the CSMD, we found that 'SR' at the beginning of a name tended to be 'Sister,' where 'SR' at the end of a name tended to be 'Senior.' Reading the record can help clarify by providing context.

There are hundreds of possible notation keywords. Here, we list a subset of useful ones, in addition to the previous example:

NOTE
WORKMAN
DECEASE
HOSPICE
BANNED
READ
PREFER
PAY
FILL
USE
DOB
TWIN
NO
AKA
SELF
ONLY

Additionally, there are many pharmacy notations, related either to medication packaging or therapy management that are important. Some of these include:

MTM
EZ
LOOSE
PHI
DYE
NSC
PIP
WC

We also found the following keywords useful in identifying dummy records:

ERROR
DUMMY
VOID
DO NOT USE
MISTAKE
HOUSE STOCK
TEST
PATIENT[6]

Identifying **businesses** can be quite complex, as there are a variety of ways the name can be entered. It is also important to keep in mind that some common business terms, like the word 'CENTER' are also fairly common *last names*, which makes implementing this search more challenging. Again, we present a subset of useful search terms:

HOSPITAL
OFFICE
SHELTER
RESCUE
INSTITUTE
ALLIANCE
CLINIC
SURGERY
COUNTY
PEDIATRIC
MEDICAL
OF[7]
DENTAL
METRO
*State name

Keyword identification is one of the pieces that makes deep cleaning procedures so complex. The goal of this section has been to give the reader a starting point for running these searches, as well as a set of useful keywords to start with. We end this section with the reminder that it is important to let the data guide these searches; every database will have a unique set of keywords.

[6] When listed as a first name with no other indicators. In the CSMD, a common dummy notation was FirstName: PATIENT, LastName: VOID.

[7] By searching for 'OF' with appropriate leading and trailing spaces, we can find entries like CITY OF, FRIENDS OF, and so on. In the CSMD, this search successfully identified only businesses, but it is possible in another state that this search might also indicate patient notations.

III. Record Parsing

Once a subset of data has been identified by keyword search as being in need of cleaning, the actual *process* of cleaning essentially consists of:

- parsing name information into the appropriate variable(s)
- parsing other retained information, such as notations, into the appropriate variable(s)
- compressing unwanted words or characters

We go back to the first guiding principle stated at the beginning of this section: we strongly recommend retaining all data. To ensure this, our first step is to create a new variable set: UncleanedFirstName, UncleanedMiddleName, UncleanedLastName. This set holds our original information. We have found this to be a good method of tracking our cleaning progress because it may be easier in writing code to overwrite variables for a variety of reasons, and this way our original data remains untouched.

Because of this principle of retaining as much data as possible, the only compressed terms are either non-alphanumeric characters or special cases of suffixes that are written phonetically, such as 'THE 3RD.' In that case, the suffix 'III' would be parsed into the appropriate variable and the original phrasing would not be retained outside of the uncleaned record. In the case of non-alphanumeric characters, we recommend removing all characters except a full stop, parentheses, dashes, and quote marks *before* running any keyword searches[8]. Characters like !@#$ are not useful in the kinds of searches that we run in the CSMD. Periods, parentheses, dashes, and quote marks are useful as search terms, however, and we recommend using PRXPARSE to index the position of any characters that are not either a space or alphabetic. We have found the following uses for these special characters in parsing records:

- periods can be used to separate initialed names such as C.J. if desired
- parentheses can be used to identify nicknames, notations, or pet names
- records beginning with quote marks are almost always pet names
- dashes and quote marks in names can be used either to separate names if desired (MARY-BETH) or can be compressed if names make more sense that way (O'HARA)

When parsing the names, we remind the reader that *placeholder* variables can be invaluable in this process. They are especially useful when an entry contains more than two words or when a subset needs multiple if-statements to sort through its records. Another reason we strongly recommend the usage of placeholder variables is because we have found it easier to clean FirstName and MiddleName simultaneously.

In the vast majority of the CSMD (91.0%), the MiddleName variable is blank. Because of this, it is more efficient to combine the first and middle name variables into a single placeholder and clean that placeholder. Once keyword searches have been used to identify notations, special characters, and prefixes, we use our placeholder variable in combination with an array to parse the names into the correct columns, including sample SAS code here because it makes more sense presented in context:

```
array First(3) $ 50 First1-First3;
placeholder = catx(" ", FirstName, MiddleName);

nameIndex = index(placeholder,' ');
i = 1;

do until(nameIndex <= 1);
    First[i] = substr(placeholder,1,nameIndex-1);
    placeholder = strip(substr(placeholder,nameIndex+1));
    nameIndex = index(placeholder,' ');
    i+1;
end;
```

Placeholders are also especially useful when notations are not uniformly formatted:

- H011 MARY L
- DONNA ROOM211 ANN
- SUSAN 30 DAY SUPPLY

In these cases, we would have identified all of these as records with notations due to the presence of numbers. Since the notations are not in the 'same place' in the string, we would parse the first word, use an if-statement to see which entry contains a number, and put the notation or name in the appropriate place, moving to the next word in the entry and repeating the process until we were out of words.

Parsing variables is a critical part of the data cleaning process. Using placeholder variables helps us not only preserve original data but also optimize our algorithms so that we can manipulate records with differing formats within the same keyword search. Due to the widely varying entry methods in the CSMD, this has proven to be crucial to our success.

Additional Resources

SAS techniques and code examples are available here:

https://www.tn.gov/content/dam/tn/health/documents/opioid_response/CSMDNameCleaningReport.pdf (see pages 19-34).

Below we also provide resources for further reading. Because of the specific challenges in cleaning the CSMD, the references listed below are essentially the documents we found useful in constructing SAS code. We are not treating this as a formal reference section for this reason, and this is by no means a comprehensive listing of literature on data cleaning.

[1] Prescription Drug Monitoring Program Training and Technical Assistance Center. (April 2015). *Technical Assistance Guide: PDMP Suggested Practices to Ensure Pharmacy Compliance and Improve Data Integrity.* Brandeis University. Retrieved from http://www.pdmpassist.org/pdf/Resources/Pharmacy_compliance_data_quality_TAG__FINAL_20150615_A.pdf

[2] Dusetzina, S., Tyree, S., & Meyer, A. (2014, September 4). Appendix 4.1, Useful SAS Functions and Procedures. In *Linking Data for Health Services Research: A Framework and Instructional Guide.* Rockville, MD, US. Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK253312/

[3] Christen, P. (2012). *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.* Springer Publishing Company, Incorporated.

[4] Cody, R. (2017). *Cody's Data Cleaning Techniques Using SAS®, Third Edition*. Cary, NC: SAS Institute, Incorporated.

We also found the following SUGI papers to be useful:

129-29, The Perks of PRX: http://www2.sas.com/proceedings/sugi29/129-29.pdf

247-31, An Introduction to Character Functions: http://www2.sas.com/proceedings/sugi31/247-31.pdf

059-30, A Clever Demonstration of the SAS SUBSTR
Function: http://www2.sas.com/proceedings/sugi30/059-30.pdf

Supplemental Digital Content 3

eFigure. Example Matched Record Set in the Controlled Substance Monitoring

Example 1: Matched record set: Identifying all patient records for one patient in the CSMD

Patient ID	Date of Birth	First Name	Middle name	Last Name	Gender	Street	City	State	Number of Prescriptions
3333	09 30 1955	Barb		Austin	2	4444 Main St.	Place	TN	1
3444444	09 30 1955	Barb		Austen-Short	6	4444 Main St.		TN	1
3555555	09 30 1955	Barb	T	Austen	2	4444 Main St.	Place	TN	17
3666666	09 30 1955	Barb	S	Austin	1	4444 Main St.	Place	TN	8
3777777	09 30 1955	Barb		Austen	2	4444 Main St.	Place	TN	8
3888888	09 30 1955	Barbara	T	Austen	6	4444 Main St.	Place	TN	1
39999999	09 30 1955	Barbra	T	Short	2	4444 Main St.	Place	TN	21

Example 2: Matched record set: Identifying all patient records for one patient in the CSMD

Patient ID	Date of Birth	First Name	Middle name	Last Name	Gender	Street	City	State	Number of Prescriptions
23232233	09 16 1975	Catherine		Book	2	2222 Main Dr.		MI	10
222222	09 16 1975	Cathy	Book	Smith	6	2222 Main Dr.	City	MI	4
37774444	09 16 1975	Catherine	George	Book	2	1726 George Rd.	Street	TN	12
377766	09 16 1975	Catherine		Book	1	1700 George Rd.	Street	TN	1
3888444	09 16 1975	Catherine	George	Book	2	333 Test Ave.	Rock	TN	3
3999555	09 16 1975	Cathy	S	Book	2	222 Main Dr.	City	MI	8

Supplemental Digital Content 4

eTable. Influence of Record Linkage Approaches on PDMP Prescription Measures

	Statistical Programming Approach (n=4,259)[a]	Most Accurate Probabilistic Approach (n=4,400)[a]	Difference in proportions (95% CI)
	n	n	
Active prescription at overdose death	2,388	2,502	0.0079 (-0.0129, 0.0288)
Any prescription in the last 60 days before overdose	2,973	3,117	0.0104 (-0.0089, 0.0296)
Opioid prescription in the last 60 days before overdose	2,478	2,606	0.0104 (-0.0103, 0.0312)
Oxycodone prescription in the last 60 days before overdose	1,371	1,443	0.0060 (-0.0137, 0.0258)
Hydrocodone prescription in the last 60 days before overdose	1,008	1,065	0.0054 (-0.0126, 0.0233)
Buprenorphine for MAT prescription in the last 60 days before overdose	139	145	0.0003 (-0.0072, 0.0078)
Benzodiazepines in the 60 days before overdose	1,887	1,991	0.0094 (-0.0115, 0.0304)
Active opioid prescription at overdose	1,774	1,875	0.0096 (-0.0112, 0.0304)
Active benzodiazepine prescription at overdose death	1,581	1,665	0.0072 (-0.0132, 0.0276)
Cash payment for a prescription in the year before overdose[b]	2,060	2,178	-0.0065 (-0.0273, 0.0143)

[a] Decedent in this analysis has to fill at least one prescription on or before the date of death within two years of death with an eligible prescription defined as at least 1 days' supply and NDC number linked to the 2017 CDC Oral morphine milligram equivalents Drug Classification Table.

[b] Excludes prescriptions with no payment information.

Table excludes true FP prescriptions (excludes only 1 death that had only FP prescriptions in the year before overdose).