Graduate Database Project

Reconstruction of the MIMIC-III Database for Data Analytics

Joseph Miles, PharmD

SUNY Oswego

BHI600IS: Database Applications

Table of Contents

Please address all comments or questions to: Joseph Miles, jmiles3@oswego.edu

Reconstruction of the MIMIC-III database for Data Analytics

Introduction

The MIMIC database is in its third iteration and is referred to as the "Medical Information Mart for Intensive Care" (previously called the "Multiparameter Intelligent Monitoring in Intensive Care" while in its second iteration - MIMIC-II, Johnson et al, 2016). MIMIC-III is a comprehensive collection of deidentified data from 53,423 distinct critical care hospital admissions from 38,597 distinct adult patients at the Beth Israel Deaconess Medical Center in Boston, Massachusetts (Johnson et al, 2016). The data has been compiled into 26 tables which contain, for example, an average of 4579 charted observations and 380 laboratory measurements for each hospital admission as well as a total of 3.8 gigabytes of unstructured textual data from various healthcare provider notes and analyses (ibid). An excellent figure from Johnson and associates (2016) summarizing the MIMIC-III database is included in Figure 1. Historically, the MIMIC database has been used in industrial research, quality improvement initiatives, and higher education coursework (ibid).

Project Overview

MIMIC-III is a static warehouse of patient data recorded between 2001 and 2012 and is maintained by Massachusetts Institute of Technology (MIT). All patient information has been thoroughly deidentified.

In addition to deidentifying patient data, MIT requires training in the protection of patient data for anyone requesting access to the MIMIC dataset. After completing the prescribed training, data can be downloaded as 26 comma separated values (csv) files representing the 26 tables in the MIMIC-III database. Sample SQL code can be acquired from GitHub (https://github.com/MIT-LCP/mimic-code) for establishing relationships between the tables. Additionally, there is a published data dictionary which can be found at https://mimic.physionet.org/mimictables/admissions/.

There is variability in the usage of "unique" attributes and definition of primary keys between the sample SQL code and the published data dictionary. For example, every table has an attribute called "ROW_ID", and the sample SQL code consistently declares this attribute as "unique" and/or as a "primary key" for every table despite the fact that tables like the "PATIENTS" table have a unique identifier (SUBJECT_ID) that is intended to be the primary key and serve as foreign key in child relations that refer to the "PATIENTS" table. Because of the variability of past instantiations of MIMIC, the MIMIC database was carefully reverse-engineered to establish a logical process for reassembling the database for SUNY Oswego.

After downloading and analyzing the MIMIC source tables, implementation occurs in 5 additional steps:

1. Create tables with attribute rules (data types) and identify the primary key for each table.
2. Load records from csv files into each table.
3. Declare the indexes for each table.
4. Define foreign keys in each table and establish table relationships.
5. Implement user interface (with appropriately granted permissions) for the database.

Project Scope and Functionality

Due to the complexity of the MIMIC-III database and the massive size of the available source data (over 40 gigabytes), the scope of this project will be to create a database capable of being queried to accomplish the tasks of project creation, project validation, and other forms of data mining. In contrast to the standard of creating database views that are intended to be intuitive and specific for a naïve end-user, a single "view" which excepts ad hoc SQL queries will be necessary since all users will fall under the category of researcher/data analyst.

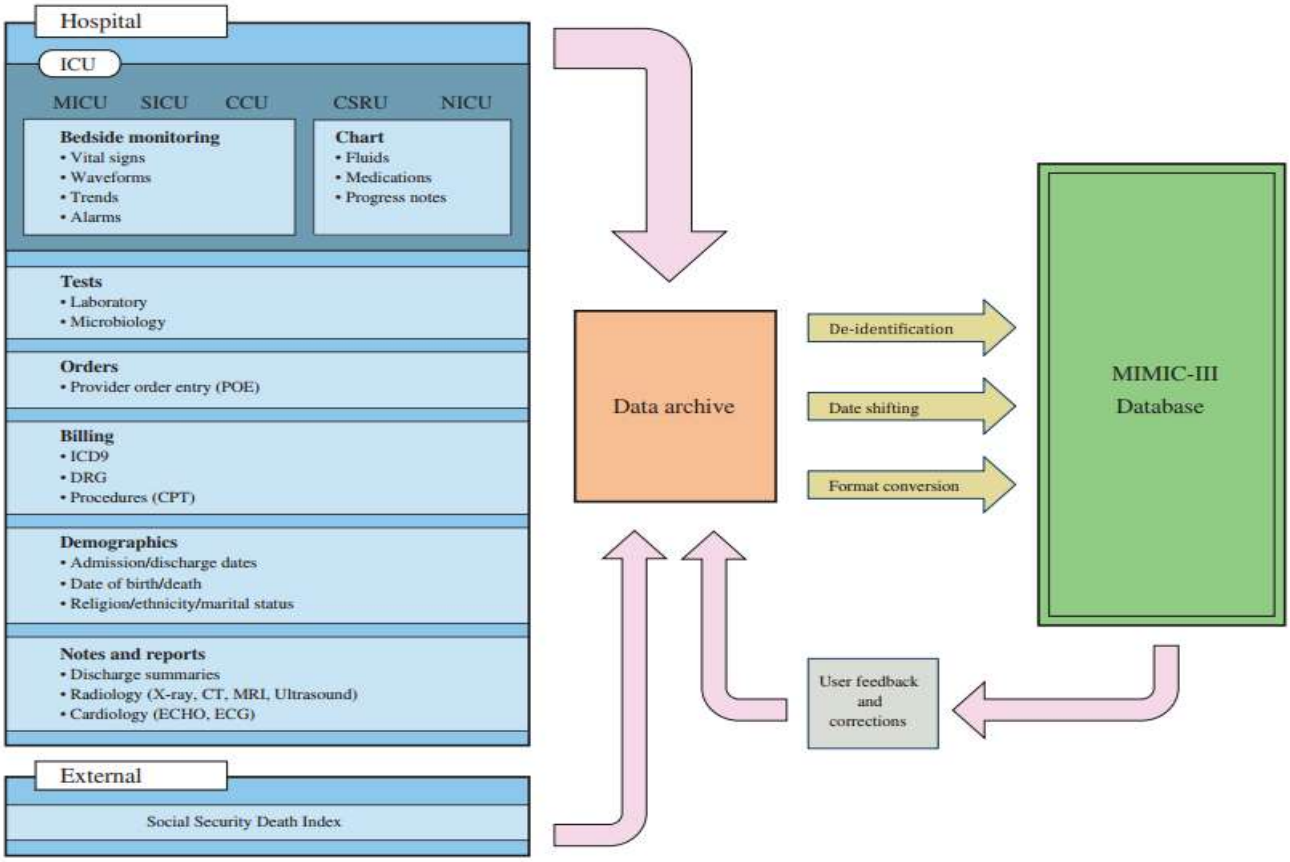
Client/User Background

Establishing a local copy of the MIMIC-III database will allow students and faculty involved with Biomedical and Health Informatics at SUNY Oswego to analyze the data for project creation (example: create hypotheses), project validation (example: compare study data with data in MIMIC-III) and various data mining exercises (example: semantic analysis of provider notes). As mentioned above, all users of this MIMIC implementation could be singly categorized as "researcher/data analyst."

Document Overview

This document includes the conceptual model used recreate the MIMIC-III database, a description of the interface created for the users of the database, an explanation of results and a discussion concerning the importance of data analytics.

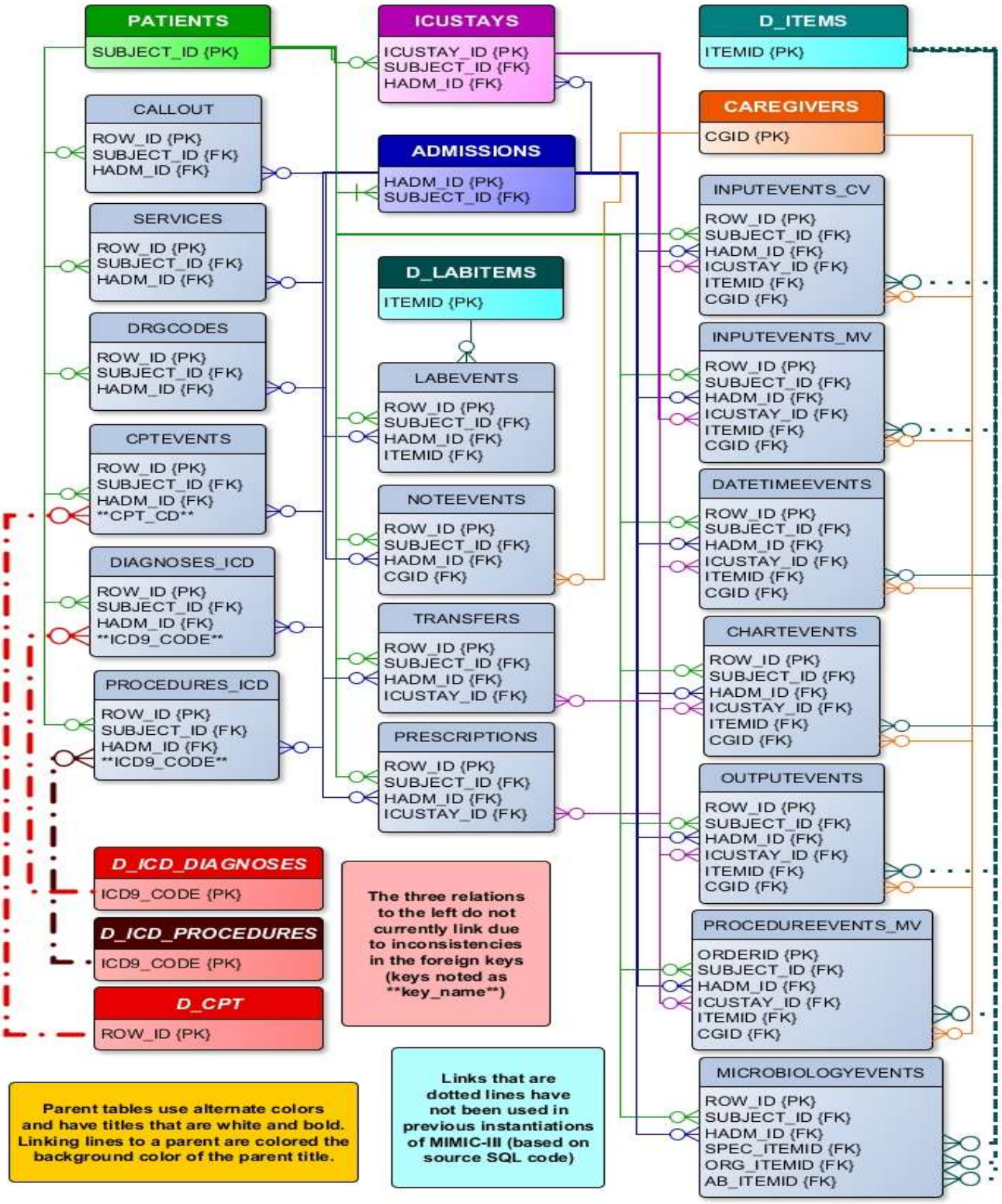
Figure 1: "Overview of the MIMIC-III critical care database" from Johnson and Associates, 2016.



Conceptual Data Model

Based on the analysis of the database and the relationships between the 26 tables in the database, the following Entity Relationship (ER) class diagram was created:

Figure 2: MIMIC-III Entity Relationship Class Diagram



Important considerations from the ER diagram (Figure 2):

- Only the attributes used as primary key and foreign key are listed in the ER diagram.
    - For a complete list of attributes, consult the Data Dictionary.
- Although every table has an attribute called "ROW_ID", only the tables using "ROW_ID" as a primary key list this unique attribute.
    - In consideration for the performance of the reconstructed database, it was decided that it was unnecessary for the database management system (DBMS) to index a unique "ROW_ID" for the tables where the "ROW_ID" attribute is never referred to by any other relation and is not utilized as a primary key.
    - The eight parent tables plus the table named "PROCEDUREEVENTS_MV" all use a primary key other than "ROW_ID".
    - Composite primary keys could have been created in many of the tables that would be more meaningful, but in the interest of time and minimizing changes from previous successful MIMIC instantiations, "ROW_ID" will still be used as primary key in 17 tables.
- Several foreign keys referring to the "D_ITEMS" table were not identified in the SQL code provided on GitHub and were added for the SUNY Oswego MIMIC-III instantiation.
- The foreign keys "CPTEVENTS.CPT_CD", "DIAGNOSES_ICD.ICD9_CODE", and "PROCEDURES_ICD.ICD9_CD" all contain strings that do not match any value in the parent table's primary key and will not be enforced in the SUNY Oswego MIMIC instantiation.
    - As a result, the following tables have no established relationships: "D_CPT," "D_ICD_DIAGNOSES" and "D_ICD_PROCEDURS".
- Finally, since the dataset is so large, it was impossible to refine the cardinality to know which one-to-many relationships are "zero or more"  or "one or more"  on the "many" side of the relationship. For purposes of the ER diagram, it was known that all patients have at least one admission, so the "one or more" relationship was used. For all other relationships, there is an assumption that there may be individual ICU stays that may not have data represented in some of the other tables even though, realistically, there are probably many more "one or more" notations that should have been used in the ER diagram.
    - As always, any errors in the above interpretations are the fault of this document's author alone.

Creation of the Data Dictionary was an important part of the conceptual understanding of the MIMIC-III database. With 324 different attributes in the 26 tables, much care was exercised for understanding the structure of the MIMIC-III database.

In the process of reverse engineering the MIMIC-III database, it was discovered that most tables would fail third normal form in database design. For example, most tables refer to both the admission ID ("HADM_ID") and the patient ID ("SUBJECT_ID") despite the fact that only one "SUBJECT_ID" is associated with each "HADM_ID" (meaning that "SUBJECT_ID" is dependent on "HADM_ID"). Due to the size and complexity of the MIMIC-III database, it is assumed that denormalization was purposeful and will lead to more efficient analysis of the data. Since the database is not intended for data insertion nor data updates, denormalization should not present a problem in the SUNY Oswego instantiation of MIMIC-III.

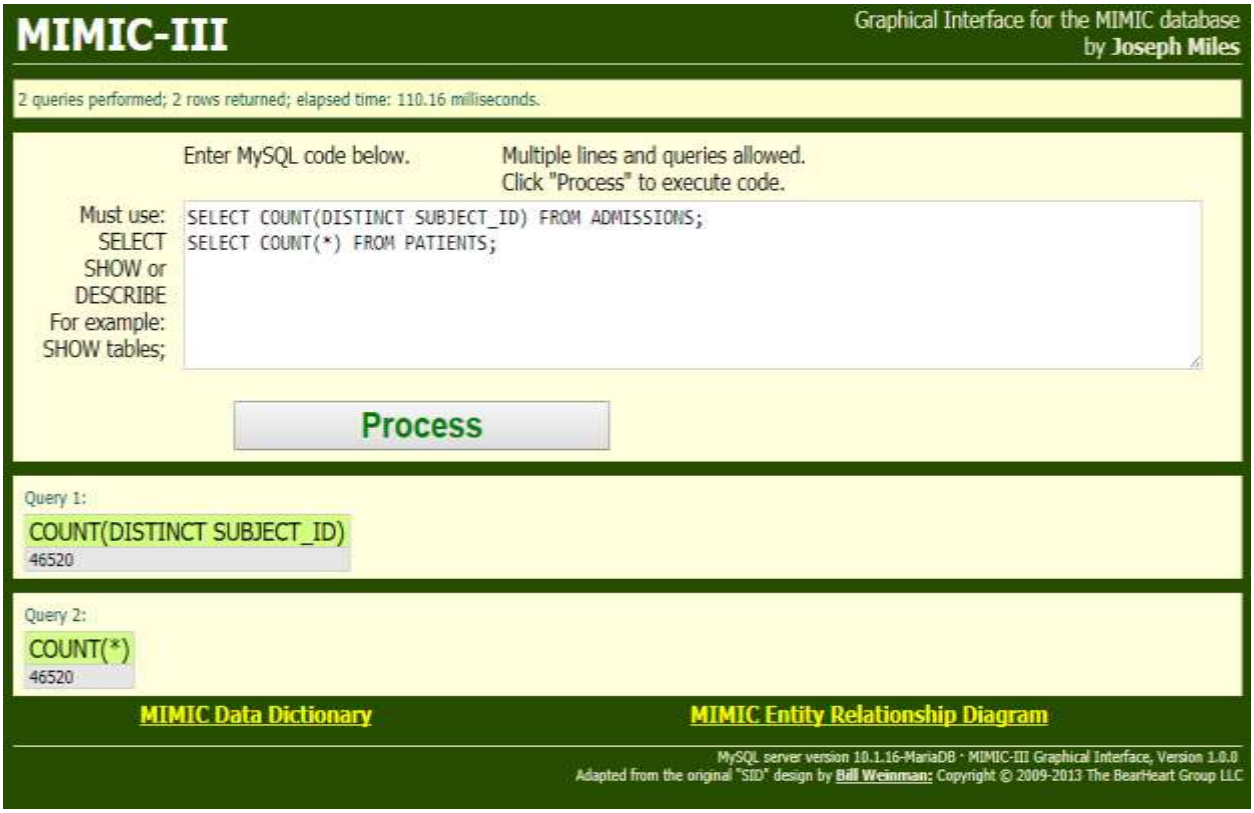
In consideration of a use case for the SUNY Oswego MIMIC instantiation, all users of the database will fall under a single actor role of "Researcher/Data Analyst." The database is being implemented as "read only," so there shouldn't need to be any ongoing support of the database once it is instantiated. The user (actor) will interface using ad hoc SQL queries and will be allowed to copy any results for further analysis outside and beyond the database environment.

Database Interface (View)

No traditional relational database "views" are going to be created for this database implementation. However, having a common tool to interface with the database will be essential for database access by multiple analysts from various locations.

Using a database interface template (called "SID" by its creator, Bill Weinman), a versatile interface was created, using the PHP programming language, to facilitate interaction with MIMIC-III. A screen shot of the interface can be seen in Figure 3.

Figure 3: Graphical Interface for the MIMIC database



- A database username was created in MySQL giving the user of this interface only "SELECT" privileges: SQL command: GRANT SELECT ON mimic.* TO 'createduser'@'localhost';
- As a further safeguard, the PHP code for the interface was altered to only allow commands that start with 'SELECT', 'DESCRIBE', 'SHOW', 'DESC', 'EXPLAIN'.
    - 'WITH' is also a way of selecting data, but MySQL does not yet recognize the 'WITH' command and is ignored by this interface.

- Row striping and mouse hover highlighting was added for easier interpretation of the result table:

ROW_ID	ITEMID	LABEL	ABBREVIATION	DBSOURCE	LINKSTO
15001	227666	Immobilizer Location	Immobilizer Location	metavision	chartevents
15002	227667	Reason for Immobilizer	Reason for Immobilizer	metavision	chartevents
15003	227668	Side Rails_V2	Side Rails_V2	metavision	chartevents
15004	227669	Side Rails (Restraint)	Side Rails (Restraint)	metavision	chartevents

- In the area below the page header, feedback about the query is provided. As can be seen in Figure 3, after query completion, the utility will display the number of results and the time elapsed for the query (or queries).
    - In the event of an error, the feedback will relay the error message from MySQL:
      query #1: SQLSTATE[42000]: Syntax error or access violation: 1064 You have MariaDB server version for the right syntax to use near 'attributes' at line 1
- Data in the result tables can be easily copied and pasted into other programs, such as Microsoft Excel, for further analysis and data comparisons.
- For efficient results with simple 'SELECT' commands, it is suggested to add a 'LIMIT' clause to the end of many non-aggregated SQL queries since the tables contain a large number of records (for example: "CHARTEVENTS" has over 250 million records). Similarly, you can limit results with "'WHERE' results are 'BETWEEN' values", etc.
    - In general, refine queries to ensure that the most usable results are produced.

Results

The database was implemented in a test environment on a personal computer using XAMPP Control Panel v3.2.2 as an Apache server for MySQL MariaDB server version 10.1.16-MariaDB. Due to the limited capabilities of the personal computer, not all records were loaded into every table, but every table contained at least hundreds of records. (A python script was used to segment larger csv files into smaller files for database upload.) Indexes were then added for each table and foreign keys were added for each table. Sample SQL code used for creating the "NOTEEVENTS" table is shown below:

Figure 4: Sample SQL code for uploading MIMIC data to MySQL

```
USE mimic;
tee noteevent-index.log

DROP TABLE IF EXISTS NOTEEVENTS;
CREATE TABLE NOTEEVENTS (    -- rows=2078705
   ROW_ID MEDIUMINT UNSIGNED NOT NULL,
   SUBJECT_ID MEDIUMINT UNSIGNED NOT NULL,
   HADM_ID MEDIUMINT UNSIGNED,
   CHARTDATE DATE NOT NULL,
   CHARTTIME DATETIME,
   STORETIME DATETIME,
   CATEGORY VARCHAR(50) NOT NULL,    -- max=17
   DESCRIPTION VARCHAR(255) NOT NULL,    -- max=80
   CGID SMALLINT UNSIGNED,
   ISERROR TINYINT UNSIGNED,
   TEXT MEDIUMTEXT, -- max=55725
-- *  UNIQUE KEY NOTEEVENTS_ROW_ID (ROW_ID) -- nvals=2078705
   INDEX NOTEEVENTS_INDX01 (ROW_ID),
   PRIMARY KEY NOTEEVENTS_PK_ROW_ID (ROW_ID)
  )
  CHARACTER SET = UTF8;

LOAD DATA LOCAL INFILE 'NOTEEVENTS.csv' INTO TABLE NOTEEVENTS
   FIELDS TERMINATED BY ',' ESCAPED BY '\\' OPTIONALLY ENCLOSED BY '"'
   LINES TERMINATED BY '\n'
   IGNORE 1 LINES
   (@ROW_ID,@SUBJECT_ID,@HADM_ID,@CHARTDATE,@CHARTTIME,@STORETIME,
    @CATEGORY,@DESCRIPTION,@CGID,@ISERROR,@TEXT)
 SET
   ROW_ID = @ROW_ID,
   SUBJECT_ID = @SUBJECT_ID,
   HADM_ID = IF(@HADM_ID='', NULL, @HADM_ID),
   CHARTDATE = @CHARTDATE,
   CHARTTIME = IF(@CHARTTIME='', NULL, @CHARTTIME),
   STORETIME = IF(@STORETIME='', NULL, @STORETIME),
   CATEGORY = @CATEGORY,
   DESCRIPTION = @DESCRIPTION,
   CGID = IF(@CGID='', NULL, @CGID),
   ISERROR = IF(@ISERROR='', NULL, @ISERROR),
   TEXT = IF(@TEXT='', NULL, @TEXT);

```

(SQL code is continued on the next page)

```
42  alter table NOTEEVENTS
43        add INDEX NOTEEVENTS_INDX02 (SUBJECT_ID),
44        add INDEX NOTEEVENTS_INDX03 (HADM_ID),
45        add INDEX NOTEEVENTS_INDX04 (CHARTDATE),
46        add INDEX NOTEEVENTS_INDX05 (CATEGORY),
47        add INDEX NOTEEVENTS_INDX06 (CGID),
48        add INDEX NOTEEVENTS_INDX07 (DESCRIPTION);
49
50  -- subject_id
51  ALTER TABLE NOTEEVENTS DROP FOREIGN KEY noteevents_fk_subject_id;
52  ALTER TABLE NOTEEVENTS
53  ADD CONSTRAINT noteevents_fk_subject_id
54    FOREIGN KEY (SUBJECT_ID)
55    REFERENCES PATIENTS(SUBJECT_ID);
56
57  -- hadm_id
58  ALTER TABLE NOTEEVENTS DROP FOREIGN KEY noteevents_fk_hadm_id;
59  ALTER TABLE NOTEEVENTS
60  ADD CONSTRAINT noteevents_fk_hadm_id
61    FOREIGN KEY (HADM_ID)
62    REFERENCES ADMISSIONS(HADM_ID);
63
64  -- cgid
65  ALTER TABLE NOTEEVENTS DROP FOREIGN KEY noteevents_fk_cgid;
66  ALTER TABLE NOTEEVENTS
67  ADD CONSTRAINT noteevents_fk_cgid
68    FOREIGN KEY (CGID)
69    REFERENCES CAREGIVERS(CGID);
70
71  notee
```

All tables loaded successfully. As can be seen on line 17, 'UNIQUE KEY' designation is commented out in lieu of the 'PRIMARY KEY' designation on line 19. 'INDEX' is declared, redundantly, for the 'PRIMARY KEY' as the only declared index before data upload. Remaining 'INDEX' commands can be seen starting on line 42 above.

phpMyAdmin (version 4.5.1) was used to analyze the structure of the test database. The resulting data dictionary created by phpMyAdmin matched the intended Data Dictionary in this document.

The data dictionary created by phpMyAdmin for the test database environment can be viewed at: http://pi.cs.oswego.edu/~jmiles3/mimic/DataDictionary

The working interface for the database can be seen in Figure 3. Test queries can be found at https://mimic.physionet.org/tutorials/intro-to-mimic-iii/. All tested queries completed successfully. Test query results were easily and flawlessly copied into Microsoft Excel for further analysis.

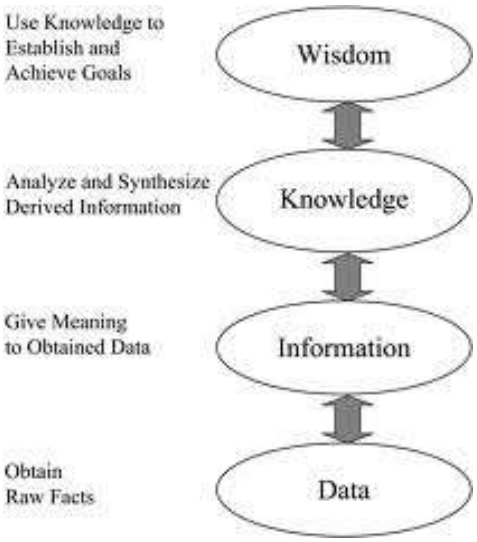
SQL code for recreating the database on the SUNY Oswego MySQL servers is contained in 3 separate files (one for table creation and data upload, second for index creation, and third for foreign key relationship creation).
The three SQL files can be downloaded from: http://pi.cs.oswego.edu/~jmiles3/mimic/sql/

Part of the process of patient deidentification was to change all dates to future dates to further protect patient identity. Altered admission dates preserved the actual day of the week and approximate season of the year, but no other information can be derived from the altered date. Date and time differentials for each patient are accurate, such as comparing a patient's admit date to their altered date of birth will give an accurate age-on-admission. An exception to this age algorithm, in accordance with established rules of patient deidentification (United States Department of Health and Human Services), patients over the age of 89 are given a non-specific age greater than 299 years old. For all patients, comparing future dated timestamps of measured patient data (such as blood pressure or blood chemistry measurement) is accurate in relation to their ICU stay timeline.

Discussion

The practice of informatics can be defined by Charles Friedman's (2009) example of a human utilizing technology to produce greater results than either the human or the computer alone. The goal of health informatics could be defined as refining **data** to produce new **information** which can be synthesized into better **knowledge** which develops into **wisdom** utilized for future success (Figure 5).

Figure 5: Data-Information-Knowledge-Wisdom Continuum (Rhona Carretas, 2012)



Future success in health informatics will be in producing wisdom for improved patient care.

MIMIC-III represents a decade of patient data from intensive care units of a single medical facility, and yet millions of rows of data exist from this small sliver of history for analysis and for deriving information. The "NOTEEVENTS" table alone contains are over 2-million rows holding several paragraphs of free text data in each row from medical providers concerning the care for the patients referenced in the database. The MIMIC-III database will provide resources for project creation and validation as well as material for various data mining exercises, especially in the realm of analyzing unstructured textual data.

Giuseppe Polese wrote in 2014: "data mining and data warehousing provide tools to acquire medical data, to extract relevant information from them, and to make this knowledge available to all the people involved in health care." Beyond the hope that medical wisdom can be derived from the multiple instantiations of MIMIC databases utilized around the world, hopefully the existence of the MIMIC database and the success it continues to create for those who utilize the data's information will lead other institutions to similarly share and utilize patient data towards the goal of increasing patient-care-wisdom.

References:

Carretas, R. (2012). Nursing Informatics. Retrieved 8/9/2017 from http://rhonacarretas.blogspot.com/2012/05/theories-in-nursing-informatics.html

Friedman, CP. (2009). A "fundamental theorem" of biomedical informatics. *Journal Of The American Medical Informatics Association: JAMIA.* 16(2): 169-170. doi:10.1197/jamia.M3092

Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data* DOI: 10.1038/sdata.2016.35. Available from: http://www.nature.com/articles/sdata201635

Polese, G. (2014). A decision support system for Evidence Based Medicine. *Journal of Visual Languages and Computing*. 25(2014): 858-867. http://dx.doi.org/10.1016/j.jvlc.2014.09.013

United States Department of Health and Human Services. Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule. Subsection: Age deidentification: "What are examples of dates that are not permitted according to the Safe Harbor Method?" retrieved 8/8/2017 from https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html

Weinman, B. SQL Essential Training. *Lynda.com*. "SID" source code retrieved 8/6/2017 from https://www.lynda.com/SQL-tutorials/SQL-Essential-Training/139988-2.html

Data Dictionary for MIMIC-III

Table Name **Page number**

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
ADMISSIONS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Hospital admission associated with ICU stay**
ROW_ID			INT	Y		(Obsolete) Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID	Y		INT	Y	Y	Unique identifier for each hospital stay
ADMITTIME			TIMESTAMP(0)	Y		Time of admission
DISCHTIME			TIMESTAMP(0)	Y		Time of discharge
DEATHTIME			TIMESTAMP(0)			Time of death
ADMISSION_TYPE			VARCHAR(50)	Y	Y	Type of admission [example: emergency or elective]
ADMISSION_LOCATION			VARCHAR(50)	Y		Admission location
DISCHARGE_LOCATION			VARCHAR(50)	Y		Discharge location
INSURANCE			VARCHAR(255)	Y		Insurance type
LANGUAGE			VARCHAR(10)			Language
RELIGION			VARCHAR(50)			Religion
MARITAL_STATUS			VARCHAR(50)			Marital status
ETHNICITY			VARCHAR(200)	Y		Ethnicity
EDREGTIME			TIMESTAMP(0)			Time patient was registered in the emergency department
EDOUTTIME			TIMESTAMP(0)			Time patient was discharged from the emergency department
DIAGNOSIS			VARCHAR(300)			Diagnosis
HOSPITAL_EXPIRE_FLAG			TINYINT	Y		
HAS_CHARTEVENTS_DATA			TINYINT	Y		Has at least one observation in CHARTEVENTS table
CALLOUT	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Patient ready for discharge and outcome information**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT	Y	Y	REFERENCES ADMISSIONS(HADM_ID)
SUBMIT_WARDID			INT			Identifies ward where call out request was submitted
SUBMIT_CAREUNIT			VARCHAR(15)			Ward where call out was submitted, if care unit, ICU type listed here
CURR_WARDID			INT			Ward where patient is currently residing

Data Dictionary Key: FK = Foreign Key, Indx = Indexed attribute, NN = Not NULL, PK = Primary Key; for FK: REFERENCES TABLE_NAME(ATTRIBUTE_NAME)

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
CURR_CAREUNIT			VARCHAR(15)		Y	If currently in a care unit, ICU type listed here
CALLOUT_WARDID			INT	Y		Where patient is to be discharged, '0' = home, '1' = first available ward
CALLOUT_SERVICE			VARCHAR(10)	Y	Y	Identifies service that the patient is called out to
REQUEST_TELE			SMALLINT	Y		Indicates if special precautions are required [telemetry]
REQUEST_RESP			SMALLINT	Y		Indicates if special precautions are required [respiratory]
REQUEST_CDIFF			SMALLINT	Y		Indicates if special precautions are required [CDiff infection]
REQUEST_MRSA			SMALLINT	Y		Indicates if special precautions are required [MRSA infection]
REQUEST_VRE			SMALLINT	Y		Indicates if special precautions are required [VRE infection]
CALLOUT_STATUS			VARCHAR(20)	Y		Current status of the call out request
CALLOUT_OUTCOME			VARCHAR(20)	Y		Result [cancellation or a discharge]
DISCHARGE_WARDID			INT			The ward to which the patient was discharged
ACKNOWLEDGE_STATUS			VARCHAR(20)	Y		Status of the response to the call out request
CREATETIME			TIMESTAMP(0)	Y		Time and date that the call out was initiated
UPDATETIME			TIMESTAMP(0)	Y		Last time the call out event was updated
ACKNOWLEDGETIME			TIMESTAMP(0)			Time at which the call out request was acknowledged
OUTCOMETIME			TIMESTAMP(0)	Y		Time at which (cancellation or discharge) occurred
FIRSTRESERVATIONTIME			TIMESTAMP(0)			First time at which a ward was reserved for the call out request
CURRENTRESERVATIONTIME			TIMESTAMP(0)			Latest time which a ward was reserved for the call out request
CAREGIVERS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**List of caregivers associated with an ICU stay**
ROW_ID			INT	Y		(Obsolete) Unique row identifier
CGID	Y		INT	Y	Y	Unique caregiver identifier
LABEL			VARCHAR(15)			Title of the caregiver [example: MD or RN]
DESCRIPTION			VARCHAR(30)			More detailed description of the caregiver
CHARTEVENTS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Events occurring on a patient chart**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	NUMBER(7,0)	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	NUMBER(7,0)		Y	REFERENCES ADMISSIONS(HADM_ID)

Data Dictionary Key: FK = Foreign Key, Indx = Indexed attribute, NN = Not NULL, PK = Primary Key; for FK: REFERENCES TABLE_NAME(ATTRIBUTE_NAME)

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
ICUSTAY_ID		Y	NUMBER(7,0)		Y	REFERENCES ICUSTAYS(ICUSTAY_ID)
ITEMID		Y	NUMBER(7,0)	Y	Y	REFERENCES D_ITEMS(ITEMID)
CHARTTIME			DATE	Y		Time the event occurred
STORETIME			DATE			Time the event was recorded in the system
CGID		Y	NUMBER(7,0)		Y	REFERENCES CAREGIVERS(CGID)
VALUE			VARCHAR2(200 BYTE)			Value of the event as a text string
VALUENUM			NUMBER			Value of the event as a number
VALUEUOM			VARCHAR2(20 BYTE)			Unit of measurement
WARNING			NUMBER(1,0)			Flag to highlight that the value has triggered a warning
ERROR			NUMBER(1,0)			Flag to highlight an error with the event
RESULTSTATUS			VARCHAR2(20 BYTE)			Result status of lab data
STOPPED			VARCHAR2(20 BYTE)			Text string indicating the stopped status of an event
CPTEVENTS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Events recorded in Current Procedural Terminology**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT	Y	Y	REFERENCES ADMISSIONS(HADM_ID)
COSTCENTER			VARCHAR(10)	Y		Center recording the code [example: ICU or respiratory unit]
CHARTDATE			TIMESTAMP(0)			Date the event occurred
CPT_CD		*Y	VARCHAR(10)	Y	Y	*Broken relationship with D_CPT(_,_,_CODEINSUBSETION)
CPT_NUMBER			INT			Current Procedural Terminology code number
CPT_SUFFIX			VARCHAR(5)			Text element of CPT code, indicates code category
TICKET_ID_SEQ			INT			Sequence number of the event, derived from the ticket ID
SECTIONHEADER			VARCHAR(50)			High-level section of the CPT code
SUBSECTIONHEADER			VARCHAR(300)			Subsection of the CPT code
DESCRIPTION			VARCHAR(200)			Description of the Current Procedural Terminology
D_CPT	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Dictionary of Current Procedural Terminology**
ROW_ID	Y		INT	Y	Y	Unique row identifier

Data Dictionary Key: FK = Foreign Key, Indx = Indexed attribute, NN = Not NULL, PK = Primary Key; for FK: REFERENCES TABLE_NAME(ATTRIBUTE_NAME)

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
CATEGORY			SMALLINT	Y	Y	Code category
SECTIONRANGE			VARCHAR(100)	Y		Range of codes within the high-level section
SECTIONHEADER			VARCHAR(50)	Y		Section header
SUBSECTIONRANGE			VARCHAR(100)	Y		Range of codes within the subsection
SUBSECTIONHEADER			VARCHAR(300)	Y		Subsection header
CODESUFFIX			VARCHAR(5)			Text element of Current Procedural Terminology
MINCODEINSUBSECTION			INT	Y		Minimum code within the subsection
MAXCODEINSUBSECTION			INT	Y		Maximum code within the subsection
D_ICD_DIAGNOSES	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Dictionary of Internatnl. Classification of Disease-Diagnoses**
ROW_ID			INT	Y		(Obsolete) Unique row identifier
ICD9_CODE	Y		VARCHAR(10)	Y	Y	Fixed length field (whitespaces included), uniquely ID ICD codes
SHORT_TITLE			VARCHAR(50)	Y	Y	Short title associated with the ICD code
LONG_TITLE			VARCHAR(300)	Y		Long title associated with the ICD code
D_ICD_PROCEDURES	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Dictionary of Internatnl. Classification of Disease-Procedures**
ROW_ID			INT	Y		(Obsolete) Unique row identifier
ICD9_CODE	Y		VARCHAR(10)	Y	Y	Fixed length field (whitespaces included), uniquely ID ICD codes
SHORT_TITLE			VARCHAR(50)	Y	Y	Short title associated with the ICD code
LONG_TITLE			VARCHAR(300)	Y		Long title associated with the ICD code
D_ITEMS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Dictionary of non-laboratory-related charted items**
ROW_ID			INT	Y		(Obsolete) Unique row identifier
ITEMID	Y		INT	Y	Y	Unique identifier for the charted item
LABEL			VARCHAR(200)			Label identifying the item
ABBREVIATION			VARCHAR(100)			Abbreviation associated with the item
DBSOURCE			VARCHAR(20)	Y		Source database of the item
LINKSTO			VARCHAR(50)			Table which contains data for the given ITEMID
CATEGORY			VARCHAR(100)		Y	Category of data which the concept falls under
UNITNAME			VARCHAR(100)			Unit associated with the item

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
PARAM_TYPE			VARCHAR(30)			Type of item [example: solution or ingredient]
CONCEPTID			INT			ID used to harmonize concepts identified by ITEMIDs [not used?]
D_LABITEMS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Dictionary of laboratory-related items**
ROW_ID			INT	Y		(Obsolete) Unique row identifier
ITEMID	Y		INT	Y	Y	Unique identifier for the charted item
LABEL			VARCHAR(100)	Y		Label identifying the item
FLUID			VARCHAR(100)	Y		Fluid associated with the item [example: blood or urine]
CATEGORY			VARCHAR(100)	Y		Category of item [example: hematology or chemistry]
LOINC_CODE			VARCHAR(100)		Y	Logical Observation Identifiers Names and Codes for item
DATETIMEEVENTS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Events relating to a datetime**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT		Y	REFERENCES ADMISSIONS(HADM_ID)
ICUSTAY_ID		Y	INT		Y	REFERENCES ICUSTAYS(ICUSTAY_ID)
ITEMID		Y	INT	Y	Y	REFERENCES D_ITEMS(ITEMID)
CHARTTIME			TIMESTAMP(0)	Y	Y	Time the event occurred
STORETIME			TIMESTAMP(0)	Y		Time the event was recorded in the system
CGID		Y	INT	Y	Y	REFERENCES CAREGIVERS(CGID)
VALUE			TIMESTAMP(0)		Y	Value of the event as a text string
VALUEUOM			VARCHAR(50)	Y		Unit of measurement
WARNING			SMALLINT			Flag to highlight that the value has triggered a warning
ERROR			SMALLINT			Flag to highlight an error with the event
RESULTSTATUS			VARCHAR(50)			Result status of lab data
STOPPED			VARCHAR(50)			Event was explicitly marked as stopped (rarely used)
DIAGNOSES_ICD	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Diagnosis on admission coded using ICD9 system**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
HADM_ID		Y	INT	Y	Y	REFERENCES ADMISSIONS(HADM_ID)
SEQ_NUM			INT			Priority of the code. Sequence 1 is the primary code
ICD9_CODE		*Y	VARCHAR(10)		Y	*Referential integrity is not intact in source data (ICD9 Code)
DRGCODES	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Hospital stay classified using DRG system**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT	Y	Y	REFERENCES ADMISSIONS(HADM_ID)
DRG_TYPE			VARCHAR(20)	Y	Y	Type of Diagnosis-Related Group [ex: APR = All Patient Refined]
DRG_CODE			VARCHAR(20)	Y	Y	Diagnosis-Related Group code
DESCRIPTION			VARCHAR(300)			Description of the DRG
DRG_SEVERITY			SMALLINT			Relative severity, available for type APR only
DRG_MORTALITY			SMALLINT			Relative mortality, available for type APR only
ICUSTAYS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**List of ICU admissions**
ROW_ID			INT	Y		(Obsolete) Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT	Y	Y	REFERENCES ADMISSIONS(HADM_ID)
ICUSTAY_ID	Y		INT	Y	Y	Unique identifier for the ICU stay
DBSOURCE			VARCHAR(20)	Y		Source database of the item
FIRST_CAREUNIT			VARCHAR(20)	Y	Y	First careunit associated with the ICU stay
LAST_CAREUNIT			VARCHAR(20)	Y	Y	Last careunit associated with the ICU stay
FIRST_WARDID			SMALLINT	Y		Identifier for the first ward location for the patient
LAST_WARDID			SMALLINT	Y		Identifier for the last ward location for the patient
INTIME			TIMESTAMP(0)	Y		Time of admission to the ICU
OUTTIME			TIMESTAMP(0)			Time of discharge from the ICU
LOS			DOUBLE		Y	Length Of Stay in the ICU in minutes
INPUTEVENTS_CV	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Events relating to fluid input, CareVue system**
ROW_ID	Y		INT	Y	Y	Unique row identifier

Data Dictionary Key: FK = Foreign Key, Indx = Indexed attribute, NN = Not NULL, PK = Primary Key; for FK: REFERENCES TABLE_NAME(ATTRIBUTE_NAME)

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT		Y	REFERENCES ADMISSIONS(HADM_ID)
ICUSTAY_ID		Y	INT		Y	REFERENCES ICUSTAYS(ICUSTAY_ID)
CHARTTIME			TIMESTAMP(0)	Y		Time that the input was started or received
ITEMID		Y	INT	Y	Y	REFERENCES D_ITEMS(ITEMID)
AMOUNT			DOUBLE PRECISION			Amount of the item administered to the patient
AMOUNTUOM			VARCHAR(30)			Unit of measurement for the amount
RATE			DOUBLE PRECISION			Rate at which the item is being administered to the patient
RATEUOM			VARCHAR(30)			Unit of measurement for the rate
STORETIME			TIMESTAMP(0)	Y		Time when the event was recorded in the system
CGID		Y	BIGINT		Y	REFERENCES CAREGIVERS(CGID)
ORDERID			BIGINT	Y	Y	Identifier linking items which are grouped in a solution
LINKORDERID			BIGINT	Y		Identifier linking orders across multiple administrations
STOPPED			VARCHAR(30)			Event was explicitly marked as stopped (rarely used)
NEWBOTTLE			INT			Indicates when a new bottle of the solution was hung at bedside
ORIGINALAMOUNT			DOUBLE PRECISION			Amount of the item which was originally charted
ORIGINALAMOUNTUOM			VARCHAR(30)			Unit of measurement for the original amount
ORIGINALROUTE			VARCHAR(30)			Route of administration originally chosen for the item
ORIGINALRATE			DOUBLE PRECISION			Rate of administration originally chosen for the item
ORIGINALRATEUOM			VARCHAR(30)			Unit of measurement for the rate originally chosen
ORIGINALSITE			VARCHAR(30)			Anatomical site for the original administration of the item
INPUTEVENTS_MV	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Events relating to fluid input, MetaVision system**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT	Y	Y	REFERENCES ADMISSIONS(HADM_ID)
ICUSTAY_ID		Y	INT		Y	REFERENCES ICUSTAYS(ICUSTAY_ID)
STARTTIME			TIMESTAMP(0)	Y	Y	Time the event started
ENDTIME			TIMESTAMP(0)	Y	Y	Time the event ended

Data Dictionary Key: FK = Foreign Key, Indx = Indexed attribute, NN = Not NULL, PK = Primary Key; for FK: REFERENCES TABLE_NAME(ATTRIBUTE_NAME)

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
ITEMID		Y	INT	Y	Y	REFERENCES D_ITEMS(ITEMID)
AMOUNT			DOUBLE PRECISION	Y		Amount of the item administered to the patient
AMOUNTUOM			VARCHAR(30)	Y		Unit of measurement for the amount
RATE			DOUBLE PRECISION			Rate at which the item is being administered to the patient
RATEUOM			VARCHAR(30)			Unit of measurement for the rate
STORETIME			TIMESTAMP(0)	Y		Time when the event was recorded in the system
CGID		Y	BIGINT	Y	Y	REFERENCES CAREGIVERS(CGID)
ORDERID			BIGINT	Y	Y	Identifier linking items which are grouped in a solution
LINKORDERID			BIGINT	Y		Identifier linking orders across multiple administrations
ORDERCATEGORYNAME			VARCHAR(100)	Y		A group to which the item corresponds
SECONDARYORDERCATEGORYNAME			VARCHAR(100)			A secondary group for those items with more than one grouping
ORDERCOMPONENTTYPEDESCRIPTION			VARCHAR(200)	Y		The role of the item administered in the order
ORDERCATEGORYDESCRIPTION			VARCHAR(50)	Y		The type of item administered
PATIENTWEIGHT			DOUBLE PRECISION	Y		Value of the patient weight used for medication calculation
TOTALAMOUNT			DOUBLE PRECISION			The total amount in the solution for the given item
TOTALAMOUNTUOM			VARCHAR(50)			Unit of measurement for the total amount in the solution
ISOPENBAG			SMALLINT	Y		Indicates whether the bag containing the solution is open
CONTINUEINNEXTDEPT			SMALLINT	Y		Indicates whether the item will be continued if transferred
CANCELREASON			SMALLINT	Y		Reason for cancellation
STATUSDESCRIPTION			VARCHAR(30)	Y		Current status of the order: stopped, rewritten, running, cancelled
COMMENTS_EDITEDBY			VARCHAR(30)			Title of the caregiver who edited the order
COMMENTS_CANCELEDBY			VARCHAR(40)			Title of the caregiver who canceled the order
COMMENTS_DATE			TIMESTAMP(0)			Time at which the caregiver edited or cancelled the order
ORIGINALAMOUNT			DOUBLE PRECISION	Y		Amount of the item which was originally charted
ORIGINALRATE			DOUBLE PRECISION	Y		Rate of administration originally chosen for the item
LABEVENTS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Events relating to laboratory tests**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)

Data Dictionary Key: FK = Foreign Key, Indx = Indexed attribute, NN = Not NULL, PK = Primary Key; for FK: REFERENCES TABLE_NAME(ATTRIBUTE_NAME)

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
HADM_ID		Y	INT		Y	REFERENCES ADMISSIONS(HADM_ID)
ITEMID		Y	INT	Y	Y	REFERENCES D_LABITEMS(ITEMID)
CHARTTIME			TIMESTAMP(0)	Y	Y	Time when the event occurred
VALUE			VARCHAR(200)			Value of the event as a text string
VALUENUM			DOUBLE PRECISION			Value of the event as a number
VALUEUOM			VARCHAR(20)			Unit of measurement
FLAG			VARCHAR(20)			Flag indicating whether lab test value is abnormal (NULL = normal)
MICROBIOLOGYEVENTS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Events relating to microbiology tests**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT		Y	REFERENCES ADMISSIONS(HADM_ID)
CHARTDATE			TIMESTAMP(0)	Y	Y	Date when the event occurred
CHARTTIME			TIMESTAMP(0)			Time when the event occurred
SPEC_ITEMID		Y	INT		Y	REFERENCES D_ITEMS(ITEMID) [Identifies specimen]
SPEC_TYPE_DESC			VARCHAR(100)	Y	Y	Description of the specimen
ORG_ITEMID		Y	INT		Y	REFERENCES D_ITEMS(ITEMID) [Identifies organism]
ORG_NAME			VARCHAR(100)		Y	Name of the organism
ISOLATE_NUM			SMALLINT			Isolate number associated with the test
AB_ITEMID		Y	INT		Y	REFERENCES D_ITEMS(ITEMID) [Identifies antibody]
AB_NAME			VARCHAR(30)		Y	Name of the antibody used
DILUTION_TEXT			VARCHAR(10)			The dilution amount tested and the comparison which was made
DILUTION_COMPARISON			VARCHAR(20)			The comparison component of DILUTION_TEXT
DILUTION_VALUE			DOUBLE PRECISION			The value component of DILUTION_TEXT
INTERPRETATION			VARCHAR(5)			Interpretation of the test
NOTEEVENTS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Notes associated with hospital stay**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)

Data Dictionary Key: FK = Foreign Key, Indx = Indexed attribute, NN = Not NULL, PK = Primary Key; for FK: REFERENCES TABLE_NAME(ATTRIBUTE_NAME)

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
HADM_ID		Y	INT		Y	REFERENCES ADMISSIONS(HADM_ID)
CHARTDATE			TIMESTAMP(0)	Y	Y	Date when the note was charted
CHARTTIME			TIMESTAMP(0)			Date and time when the note was charted
STORETIME			TIMESTAMP(0)			Time the event was recorded in the system
CATEGORY			VARCHAR(50)	Y	Y	Category of the note [example: discharge summary]
DESCRIPTION			VARCHAR(300)	Y	Y	More detailed categorization for the note [free text]
CGID		Y	INT		Y	REFERENCES CAREGIVERS(CGID)
ISERROR			CHAR(1)			Flag to highlight an error with the note
TEXT			TEXT			Content of the note
OUTPUTEVENTS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Output data for patients**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT		Y	REFERENCES ADMISSIONS(HADM_ID)
ICUSTAY_ID		Y	INT		Y	REFERENCES ICUSTAYS(ICUSTAY_ID)
CHARTTIME			TIMESTAMP(0)	Y	Y	Time of an output event
ITEMID		Y	INT	Y	Y	REFERENCES D_ITEMS(ITEMID)
VALUE			DOUBLE PRECISION			The amount of substance at the CHARTTIME
VALUEUOM			VARCHAR(30)			Unit of measurement for the substance
STORETIME			TIMESTAMP(0)	Y		Time the event was recorded in the system
CGID		Y	BIGINT	Y	Y	REFERENCES CAREGIVERS(CGID)
STOPPED			VARCHAR(30)			Indicates if the order was stopped at the given CHARTTIME
NEWBOTTLE			INT			Indicates that a new bag of solution was hung at given CHARTTIME
ISERROR			SMALLINT			In Metavision, checkbox indicator for an observation error
PATIENTS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Patients associated with an ICU admission**
ROW_ID			INT	Y		(Obsolete) Unique row identifier
SUBJECT_ID	Y		INT	Y	Y	Unique identifier for each patient
GENDER			VARCHAR(5)	Y		Gender

Data Dictionary Key: FK = Foreign Key, Indx = Indexed attribute, NN = Not NULL, PK = Primary Key; for FK: REFERENCES TABLE_NAME(ATTRIBUTE_NAME)

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
DOB			TIMESTAMP(0)	Y		Date of birth
DOD			TIMESTAMP(0)			Date of death
DOD_HOSP			TIMESTAMP(0)			Date of death recorded in the hospital records
DOD_SSN			TIMESTAMP(0)			Date of death recorded in social security records
EXPIRE_FLAG			VARCHAR(5)	Y	Y	Flag indicating that the patent has died
PRESCRIPTIONS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Medications prescribed**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT	Y	Y	REFERENCES ADMISSIONS(HADM_ID)
ICUSTAY_ID		Y	INT		Y	REFERENCES ICUSTAYS(ICUSTAY_ID)
STARTDATE			TIMESTAMP(0)			Date when the prescription started
ENDDATE			TIMESTAMP(0)			Date when the prescription ended
DRUG_TYPE			VARCHAR(100)	Y	Y	Type of drug
DRUG			VARCHAR(100)	Y	Y	Name of the drug
DRUG_NAME_POE			VARCHAR(100)			Name of drug on the Provider Order Entry interface
DRUG_NAME_GENERIC			VARCHAR(100)			Generic name of drug
FORMULARY_DRUG_CD			VARCHAR(120)			Formulary drug code
GSN			VARCHAR(200)			Generic Sequence Number
NDC			VARCHAR(120)			National Drug Code
PROD_STRENGTH			VARCHAR(120)			Strength of the drug (product)
DOSE_VAL_RX			VARCHAR(120)			Dose of the drug prescribed
DOSE_UNIT_RX			VARCHAR(120)			Unit of measurement associated with the dose
FORM_VAL_DISP			VARCHAR(120)			Amount of the formulaiton dispensed
FORM_UNIT_DISP			VARCHAR(120)			Unit of measurement associated with the formulation
ROUTE			VARCHAR(120)			Route of administration [example: oral or intravenous]
PROCEDUREEVENTS_MV	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Contains procedures for patients from MetaVision**
ROW_ID			INT	Y		(Obsolete) Unique row identifier

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT	Y	Y	REFERENCES ADMISSIONS(HADM_ID)
ICUSTAY_ID		Y	INT		Y	REFERENCES ICUSTAYS(ICUSTAY_ID)
STARTTIME			TIMESTAMP(0)	Y		
ENDTIME			TIMESTAMP(0)	Y		
ITEMID		Y	INT	Y	Y	REFERENCES D_ITEMS(ITEMID)
VALUE			DOUBLE PRECISION	Y		
VALUEUOM			VARCHAR(30)	Y		
LOCATION			VARCHAR(30)			
LOCATIONCATEGORY			VARCHAR(30)			
STORETIME			TIMESTAMP(0)	Y		
CGID		Y	INT	Y	Y	REFERENCES CAREGIVERS(CGID)
ORDERID	Y		INT	Y	Y	
LINKORDERID			INT	Y		
ORDERCATEGORYNAME			VARCHAR(100)	Y	Y	
SECONDARYORDERCATEGORYNAME			VARCHAR(100)			
ORDERCATEGORYDESCRIPTION			VARCHAR(50)	Y		
ISOPENBAG			SMALLINT	Y		
CONTINUEINNEXTDEPT			SMALLINT	Y		
CANCELREASON			SMALLINT	Y		
STATUSDESCRIPTION			VARCHAR(30)	Y		
COMMENTS_EDITEDBY			VARCHAR(30)			
COMMENTS_CANCELEDBY			VARCHAR(30)			
COMMENTS_DATE			TIMESTAMP(0)			
PROCEDURES_ICD	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Procedures relating to an admission coded in ICD9**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT	Y	Y	REFERENCES ADMISSIONS(HADM_ID)

Data Dictionary Key: FK = Foreign Key, Indx = Indexed attribute, NN = Not NULL, PK = Primary Key; for FK: REFERENCES TABLE_NAME(ATTRIBUTE_NAME)

MIMIC-III Data Dictionary (Created by Joseph Miles for SUNY Oswego implementation)

Table Name / **Attribute Name**	PK	FK	PostgreSQL data type	NN	Indx	Description of **Table** / **Attribute**
SEQ_NUM			INT	Y		Lower procedure numbers occurred earlier
ICD9_CODE		*Y	VARCHAR(10)	Y	Y	*Referential integrity is not intact in source data (ICD9 Code)
SERVICES	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Hospital services received by patients**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT	Y	Y	REFERENCES ADMISSIONS(HADM_ID)
TRANSFERTIME			TIMESTAMP(0)	Y		Time when the transfer occurred
PREV_SERVICE			VARCHAR(20)		Y	Previous service type
CURR_SERVICE			VARCHAR(20)	Y	Y	Current service type
TRANSFERS	**PK**	**FK**	**Data Type**	**NN**	**Indx**	**Location of patients during their hospital stay**
ROW_ID	Y		INT	Y	Y	Unique row identifier
SUBJECT_ID		Y	INT	Y	Y	REFERENCES PATIENTS(SUBJECT_ID)
HADM_ID		Y	INT	Y	Y	REFERENCES ADMISSIONS(HADM_ID)
ICUSTAY_ID		Y	INT		Y	REFERENCES ICUSTAYS(ICUSTAY_ID)
DBSOURCE			VARCHAR(20)			Source database of the item
EVENTTYPE			VARCHAR(20)			Type of event [example: admission or transfer]
PREV_CAREUNIT			VARCHAR(20)		Y	Previous careunit
CURR_CAREUNIT			VARCHAR(20)		Y	Current careunit
PREV_WARDID			SMALLINT			Identifier for the patient's previous ward
CURR_WARDID			SMALLINT			Identifier for the patient's current ward
INTIME			TIMESTAMP(0)			Time when the patient was transferred into the unit
OUTTIME			TIMESTAMP(0)			Time when the patient was transferred out of the unit
LOS			INT		Y	Length Of Stay in the unit in minutes