

Documentation for the MIMIC-III  
Critical Care Database  
(version 1.4)

MIT Laboratory for Computational Physiology,  
Cambridge, MA

+++ date = "2015-09-01T19:33:17-04:00" title = "MIMIC" linktitle = "MIMIC"  
weight = 1 toc = "true"

[menu][menu.main] parent = "About"

+++

## MIMIC-III Critical Care Database

MIMIC-III (**M**edical **I**nformation **M**art for **I**ntensive **C**are **III**) is a large, freely-available database comprising deidentified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012.

The database includes information such as demographics, vital sign measurements made at the bedside (~1 data point per hour), laboratory test results, procedures, medications, caregiver notes, imaging reports, and mortality (both in and out of hospital).

MIMIC supports a diverse range of analytic studies spanning epidemiology, clinical decision-rule improvement, and electronic tool development. It is notable for three factors:

- it is freely available to researchers worldwide
- it encompasses a diverse and very large population of ICU patients
- it contains high temporal resolution data including lab results, electronic documentation, and bedside monitor trends and waveforms.

+++ date = "2015-09-01T19:33:17-04:00" title = "Acknowledgements" linktitle = "Acknowledgements" weight = 5 toc = "true"

[menu][menu.main] parent = "About"

+++

## Acknowledgements

MIMIC is made available largely through the work of researchers at the MIT Laboratory for Computational Physiology and collaborating research groups.

Our work is supported by grants from the National Institute of Biomedical Imaging and Bioengineering (NIBIB) of the National Institutes of Health (NIH) under award numbers R01-EB001659 (2003-2013) and R01-EB017205 (2014-2018).

If you use our data, code, or algorithms, please provide a citation to this project.

## MIMIC-III citation

If you use our data, code, or algorithms in your work, please cite this project. Our recommended publication is:

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

The data citation for MIMIC-III, which we also kindly ask you cite, is:

Pollard, T. J. & Johnson, A. E. W. The MIMIC-III Clinical Database <http://dx.doi.org/10.13026/C2XW26> (2016).

## PhysioNet citation

An additional citation for all data hosted on PhysioNet is:

Physiobank, physiokit, and physionet components of a new research resource for complex physiologic signals. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov P, Mark RG, Mietus JE, Moody GB, Peng C, and Stanley HE. *Circulation*. 101(23), pe215–e220. 2000.

## Research groups

- Beth Israel Deaconess Medical Center
- MIT Clinical Decision Making
- MIT Computational Physiology and Clinical Inference Group
- MIT Lab for Computational Physiology
- Philips Health Care

```
+++ date = "2015-09-01T19:34:46-04:00" title = "Overview" linktitle =  
"Overview" weight = 1 toc = "true"
```

```
[menu][menu.main] parent = "Getting started"
```

```
+++
```

## Overview of the MIMIC-III data

MIMIC is a relational database containing tables of data relating to patients who stayed within the intensive care units at Beth Israel Deaconess Medical

Center. A table is a data storage structure which is similar to a spreadsheet: each column contains consistent information (e.g., patient identifiers), and each row contains an instantiation of that information (e.g. a row could contain the integer 340 in the patient identifier column which would imply that the row's patient identifier is 340).

The tables are linked by identifiers which usually have the suffix "ID". For example HADM\_ID refers to a unique hospital admission and SUBJECT\_ID refers to a unique patient. One exception is ROW\_ID, which is simply a row identifier unique to that table.

Tables pre-fixed with "D\_" are dictionaries and provide definitions for identifiers. For example, every row of OUTPUTEVENTS is associated with a single ITEMID which represents the concept measured, but it does *not* contain the actual name of the drug. By joining OUTPUTEVENTS and D\_ITEMS on ITEMID, it is possible to identify what concept a given ITEMID represents.

## List of tables

The following tables are used to define and track patient stays:

- **ADMISSIONS:** Every unique hospitalization for each patient in the database (defines HADM\_ID)
- **CALLOUT:** Information regarding when a patient was cleared for ICU discharge and when the patient was actually discharged
- **ICUSTAYS:** Every unique ICU stay in the database (defines ICUSTAY\_ID)
- **PATIENTS:** Every unique patient in the database (defines SUBJECT\_ID)
- **SERVICES:** The clinical service under which a patient is registered
- **TRANSFERS:** Patient movement from bed to bed within the hospital, including ICU admission and discharge

Each ICUSTAY\_ID corresponds to a single HADM\_ID and a single SUBJECT\_ID. Each HADM\_ID corresponds to a single SUBJECT\_ID. A single SUBJECT\_ID can correspond to multiple HADM\_ID (multiple hospitalizations of the same patient), and multiple ICUSTAY\_ID (multiple ICU stays either within the same hospitalization, or across multiple hospitalizations, or both).

The following tables contain data collected in the critical care unit:

- **CAREGIVERS:** Every caregiver who has recorded data in the database (defines CGID)
- **CHARTEVENTS:** All charted observations for patients
- **DATETIMEEVENTS:** All recorded observations which are dates, for example time of dialysis or insertion of lines.
- **INPUTEVENTS\_CV:** Intake for patients monitored using the Philips CareVue system while in the ICU

- **INPUTEVENTS\_\_MV**: Intake for patients monitored using the iMDSofT Metavision system while in the ICU
- **NOTEEVENTS**: Deidentified notes, including nursing and physician notes, ECG reports, imaging reports, and discharge summaries.
- **OUTPUTEVENTS**: Output information for patients while in the ICU
- **PROCEDUREEVENTS\_\_MV**: Patient procedures for the subset of patients who were monitored in the ICU using the iMDSofT MetaVision system.

The following tables contain data collected in the hospital record system:

- **CPTEVENTS**: Procedures recorded as Current Procedural Terminology (CPT) codes
- **DIAGNOSES\_ICD**: Hospital assigned diagnoses, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system
- **DRGCODES**: Diagnosis Related Groups (DRG), which are used by the hospital for billing purposes.
- **LABEVENTS**: Laboratory measurements for patients both within the hospital and in out patient clinics
- **MICROBIOLOGYEVENTS**: Microbiology measurements and sensitivities from the hospital database
- **PRESCRIPTIONS**: Medications ordered, and not necessarily administered, for a given patient
- **PROCEDURES\_ICD**: Patient procedures, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system

The following tables are dictionaries:

- **D\_CPT**: High-level dictionary of Current Procedural Terminology (CPT) codes
- **D\_ICD\_DIAGNOSES**: Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD) codes relating to diagnoses
- **D\_ICD\_PROCEDURES**: Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD) codes relating to procedures
- **D\_ITEMS**: Dictionary of ITEMIDs appearing in the MIMIC database, except those that relate to laboratory tests
- **D\_LABITEMS**: Dictionary of ITEMIDs in the laboratory database that relate to laboratory tests

## Derived tables

The MIMIC-II database contained a variety of derived tables which simplified use of the database. For example, a commonly used table was the ICUSTAY\_DETAIL table, which provided additional information summarizing a patient's ICU stay. The database also contained derived parameters commonly required by studies, such as severity scores. In MIMIC-III, we have made a conscious decision to *not* include any derived tables or calculated parameters as far as is possible. Instead, we encourage the community to produce and share scripts which can be run to create these tables or parameters. This has many advantages: it keeps the distinction between raw data and calculated data, it encourages users to validate the scripts which derive the data, and allows for as many scripts as is conceivable without cluttering the database for all users. We have provided a set of scripts at the mimic-code repository, which can be found here:

<http://github.com/MIT-lcp/mimic-code>

We will continue to update this repository both with code which we produce as well as with code produced by the community. We encourage users to make pull requests (a feature of git which allows us to integrate community created code) or raise issues regarding code found in the repository. The creation of an active international community building openly available code for capturing a variety of concepts will increase the speed of research on MIMIC-III exponentially - we hope you take the time to investigate the mimic-code repository for anything which may be of use to you, and further contribute any work of your own!

```
+++ date = "2015-09-01T14:02:13-04:00" title = "Requesting access" linktitle = "Requesting access" weight = 2 toc = "true"
```

```
[menu][menu.main] parent = "Getting started"
```

```
+++
```

## Requesting access

The latest version of MIMIC is MIMIC-III, which comprises over 58,000 hospital admissions for 38,645 adults and 7,875 neonates. The data spans June 2001 - October 2012. The database, although de-identified, still contains detailed information regarding the clinical care of patients, so must be treated with appropriate care and respect.

Researchers seeking to use the database must formally request access with the steps below:

## Complete the required training course

Prior to requesting access to MIMIC, you will need to complete the CITI “Data or Specimens Only Research” course:

- First register on the CITI program website, selecting “Massachusetts Institute of Technology Affiliates” as your organization affiliation: <https://www.citiprogram.org/index.cfm?pageID=154&icat=0&ac=0>
- Follow the links to add a Massachusetts Institute of Technology Affiliates course. In the Human Subjects training category, select the “Data or Specimens Only Research” course
- Complete the course and save a copy of your certificate.

## Request access to MIMIC-III:

- Create an account on PhysioNet using the following link: <https://physionet.org/pnw/login>. If you already have a PhysioNetWorks account, log in to it.
- Follow the instructions on PhysioNet to apply for access to the MIMIC-III project, remembering to provide your CITI training certificate: <https://physionet.org/works/MIMICIIIClinicalDatabase/access.shtml>
- When your application has been approved you will receive emails containing instructions for downloading the database from PhysioNetWorks. Approval may take several business days, and will be delayed if your request is missing any required information.

```
+++ date = “2015-09-01T14:02:13-04:00” title = “Downloading the database”  
linktitle = “Downloading the database” weight = 3 toc = “false”
```

```
[menu][menu.main] parent = “Getting started”
```

```
+++
```

## Downloading the MIMIC-III database

### PhysioNet

Once your application to access MIMIC has been approved, you will be granted access to the ‘MIMIC-III Clinical Database’ project page on PhysioNet: <https://physionet.org/works/MIMICIIIClinicalDatabase/>

### Data and build scripts

MIMIC is provided as a collection of comma-separated (CSV) files, along with scripts to help users import the data into popular database systems. Scripts are

available in the MIMIC code repository for Postgres, MySQL, and Oracle:  
<https://github.com/MIT-LCP/mimic-code/tree/master/buildmimic>

## Tutorial

Tutorials for installing MIMIC in a local Postgres database are provided for Mac OSX, Unix, and MS Windows systems. For more detail, select the ‘Tutorial’ item in the left menu.

```
+++ date = "2016-09-01T19:34:46-04:00" title = "Querybuilder" linktitle =  
"Querybuilder" weight = 4 toc = "true"
```

```
[menu][menu.main] parent = "Getting started"
```

```
+++
```

## MIMIC Querybuilder

The MIMIC Querybuilder is a graphical interface for running SQL queries on the MIMIC-III database. It is intended for light exploration of the data, in particular for researchers who are new to MIMIC-III and seeking to gain an understanding of its structure. Researchers seeking to carry out research projects with MIMIC should follow our instructions for building a local version of the database.

## Notes and limitations

Queries should be written using PostgreSQL syntax. Two key limitations have been implemented to manage server load:

- The system has been intentionally limited to return only the first 5000 rows of a query
- All queries have a maximum running time of 15 minutes. Queries that run for over 15 minutes will time out, returning no results.

While we will aim to provide a stable service, our laboratory has limited resources and so we cannot commit to providing continuous uptime.

## Access Querybuilder

Querybuilder is available via the following link, using the PhysioNetWorks username (email address) and password provided to you upon being granted access: <https://querybuilder-lcp.mit.edu/>

```
+++ date = "2017-01-01T00:00:00-04:00" title = "CareVue" linktitle = "CareVue"
weight = 2 toc = "false"
```

```
[menu][menu.main] parent = "Data details"
```

```
+++
```

## CareVue

CareVue is a clinical information system provided by Philips which archived and displayed clinical data at the bedside for intensive care unit patients admitted to the Beth Israel Deaconess Medical Center. Part of the MIMIC-III database is extracted from the CareVue system, the other part is extracted from the MetaVision system. The immediate consequence of this is that data for the entire duration of the MIMIC-III database needs to be extracted using special steps of different ITEMID in order to get data from both systems.

CareVue is frequently referenced in the documentation because:

- Patients in the CareVue system are admitted between 2001-2008 (one example of a consequence is that these patients have a longer follow-up for date of death)
- Patients in the CareVue system have data archived a different format
- All ITEMID for CareVue patients will differ from MetaVision patients

```
+++ date = "2015-09-01T19:34:46-04:00" title = "Data sources" linktitle =
"Data sources" weight = 3 toc = "true"
```

```
[menu][menu.main] parent = "Data details"
```

```
+++
```

## Patient level data: SUBJECT\_ID

The following is a list of static data available in the database for SUBJECT\_ID in the PATIENTS table:

- GENDER
- DOB
- DOD
- DOD\_HOSP
- DOD\_SSN

This list contains all the static data available for a single patient. Note that the DOD\_SSN (which also contributes to the DOD column) is acquired from the social security death registry, i.e. an external source. It contains dates of death up to

90 days in the future for Metavision patients. It contains dates of death up to 4 years in the future for CareVue patients.

## Hospital level data: HADM\_ID

The following tables were sourced from the hospital database, and contain information recorded in the hospital, but not necessarily during the patient's ICU stay:

- ADMISSIONS
- CALLOUT
- CPTEVENTS
- DIAGNOSES\_ICD
- DRGCODES
- ICUSTAYS
- LABEVENTS
- MICROBIOLOGYEVENTS
- PATIENTS
- PRESCRIPTIONS
- PROCEDURES\_ICD
- SERVICES
- TRANSFERS

The following is a list of static data available in the database for HADM\_ID in the ADMISSIONS table. This data is constant for a single hospital admission.

- ADMITTIME - The hospital admission time
- DISCHTIME - The hospital discharge time
- DEATHTIME - The date of death of the patient if they died within the hospital
- ADMISSION\_TYPE - The type of admission: ELECTIVE, EMERGENCY, NEWBORN or URGENT (note that the NEWBORN value here does not perfectly identify newborns)
- ADMISSION\_LOCATION - The location of the patient prior to hospital admission
- DISCHARGE\_LOCATION - The location of the patient after hospital discharge
- INSURANCE - The patient's type of medical insurance
- LANGUAGE - The patient's primary language
- RELIGION - The patient's stated religion
- MARITAL\_STATUS - The patient's marital status
- ETHNICITY - The patient's stated ethnicity
- DIAGNOSIS - A short description of the reason for the patient's admission

## ICU level data: ICUSTAY\_ID

The following tables were sourced from the ICU databases, and contain information only during a patient's ICU stay:

- DATETIMEEVENTS
- INPUTEVENTS\_CV
- INPUTEVENTS\_MV
- NOTEEVENTS
- OUTPUTEVENTS
- PROCEDUREEVENTS\_MV

The following is a list of static data available in the database for ICUSTAY\_ID in the ICUSTAYS table:

- DBSOURCE - The ICU database from which the patient exists in
- FIRST\_CAREUNIT - The first unit that cared for the patient (all ICUs except NWARD)
- LAST\_CAREUNIT - The last care unit that cared for the patient (all ICUs except NWARD)
- FIRST\_WARDID - An integer representing the first physical location of the patient
- LAST\_WARDID - An integer representing the last physical location of the patient
- INTIME - Time entered the ICU
- OUTTIME - Time left the ICU
- LOS - The patient's ICU length of stay

+++ date = "2015-09-01T14:02:13-04:00" title = "Glossary" linktitle = "Glossary"  
weight = 9 toc = "true"

[menu][menu.main] parent = "Data details"

+++

## Glossary of terms

Term	Description
BIDMC	Beth Israel Dea- coness Medi- cal Center

Term	Description
CCU	Coronary Care Unit
CVICU	Cardiovascular Inten- sive Care Unit
EHR	Electronic Health Record
ICU	Intensive Care Unit
LCP	Laboratory for Com- puta- tional Physiology
MICU	Medical Inten- sive Care Unit

Term	Description
MIMIC	Medical Infor- ma- tion Mart for Inten- sive Care ( <i>previ- ously Multi- pa- rame- ter Intel- ligent Moni- toring in Inten- sive Care</i> )
NICU	Neonatal Inten- sive Care Unit
SICU	Surgical Inten- sive Care Unit

+++ date = "2015-09-01T19:34:46-04:00" title = "Patient identifiers" linktitle = "Patient identifiers" weight = 3 toc = "true"

[menu][menu.main] parent = "Data details"

+++

## Types of data in the database

Data within MIMIC were recorded during routine clinical care and *not* explicitly for the purpose of retrospective data analysis. This is a key point to keep in mind when analyzing the data.

There are two types of data in the database: static data and dynamic data. Static data is recorded once for a given identifier. An example of static data is the DOB column in the PATIENTS table. Each patient has only one date of birth, which does not change over time and is not recorded with an associated timestamp. An example of dynamic data is a patient's blood pressure, which is periodically measured during a hospital stay.

This distinction between static data and dynamic data is merely a helpful conceptual construct: there is *no* strict technical distinction between date of birth and heart rate. However, static data tends to not have an associated ITEMID (as there is no need to repeatedly record values for static data), whereas dynamic data have an ITEMID to facilitate efficient storage of repeated measurements.

## Static data

### SUBJECT\_ID

The following is a list of static data available in the database for SUBJECT\_ID in the PATIENTS table:

The only static data available for patients are their gender (GENDER), date of birth (DOB) and various dates of death (DOD, DOD\_HOSP, DOD\_SSN). These columns all occur in the PATIENTS table.

### HADM\_ID

The following is a list of static data available in the database for HADM\_ID in the ADMISSIONS table:

- Admission time
- Discharge time
- Death time
- Admission type

### ICUSTAY\_ID

The following is a list of static data available in the database for ICUSTAY\_ID in the ICUSTAYS table:

- INTIME - Time entered the ICU
- OUTTIME - Time left the ICU
- First care unit
- Last care unit

## Hospital acquired data

The following tables were sourced from the hospital database, and contain information recorded in the hospital, but not necessarily during the patient's ICU stay:

- ADMISSIONS
- CALLOUT
- CPTEVENTS
- DIAGNOSES\_ICD
- DRGCODES
- ICUSTAYS
- LABEVENTS
- MICROBIOLOGYEVENTS
- PATIENTS
- PRESCRIPTIONS
- PROCEDURES\_ICD
- SERVICES
- TRANSFERS

## ICU acquired data

The following tables were sourced from the ICU databases, and contain information only during a patient's ICU stay:

- DATETIMEEVENTS
- INPUTEVENTS\_CV
- INPUTEVENTS\_MV
- NOTEVENTS
- OUTPUTEVENTS
- PROCEDUREEVENTS\_MV

## Externally acquired data

The DOD\_SSN (which also contributes to the DOD column) is acquired from the social security death registry. It contains dates of death up to 90 days in the

future for Metavision patients. It contains dates of death up to 4 years in the future for CareVue patients.

```
+++ date = "2015-09-08T06:58:18-04:00" title = "Inputs and outputs" linktitle = "Inputs and outputs" weight = 7 toc = "true"
```

```
[menu][menu.main] parent = "Data details"
```

```
+++
```

## Inputs and outputs

Inputs and outputs are extremely useful when studying intensive care unit patients. Inputs are any fluids which have been administered to the patient: such as oral or tube feedings or intravenous solutions containing medications. Outputs are fluids which have either been excreted by the patient, such as urine output, or extracted from the patient, for example through a drain. These data were the most complicated to handle technically in the MIMIC-III data.

The MIMIC-III database contains information from two distinct critical care information systems: Philips CareVue and iMDSoft Metavision. These two databases store data in different ways. In the descriptions in this document, data will be referred to as being sourced either from "CareVue" or "Metavision" to differentiate between the different systems.

## INPUTEVENTS\_CV, INPUTEVENTS\_MV, OUTPUTEVENTS

Inputs exist in two separate tables: INPUTEVENTS\_CV and INPUTEVENTS\_MV. INPUTEVENTS\_CV contains CareVue inputs, while INPUTEVENTS\_MV contains Metavision inputs. Results from these tables can be unioned as the same patient stay never occurs in both tables. Concretely, a unique ICUSTAY\_ID only occurs in *one* out of the two tables. However, if a patient has two ICU stays, one in CareVue and one in Metavision, then they could appear in both tables. As a result, a unique SUBJECT\_ID could occur in both tables. All outputs for both Metavision and CareVue patients have been merged into the OUTPUTEVENTS table.

### Outputs

Outputs are recorded in a reasonably consistent manner in both the CareVue and Metavision data. The time at which the output is measured is recorded in the CHARTTIME column. There is no start time recorded with outputs - CHARTTIME

corresponds to the time that the volume had been output by. The volume of output is recorded in the `VALUE` column, and the unit of measurement is provided in the `VALUEUOM` column (usually milliliters, or mL). It is usually reasonable to assume that any output recorded is for the interval between the current `CHARTTIME` and the previous `CHARTTIME` for the same item.

## Inputs

Inputs are handled differently by CareVue and Metavision. For CareVue data, only the `CHARTTIME` is available. Second, the `RATE` and `AMOUNT` columns are *asynchronous*, and originally stored in different tables. Volumes of input (e.g. 50 mL of normal saline) would be stored in one table (the original `IOEVENTS`), and would usually be recorded every hour (though sometimes the period was longer). Conversely, the `RATE` of the drug would be stored separately (in `MEDEVENTS`), and only updated when a change or verification of the rate was made by clinical staff. As a result, the raw data looked something similar to:

CHARTTIME	VOLUME	VOLUMEUOM	CHARTTIME	RATE	RATEUOM
		09:00	1		
10:00	60	mL		mL/min	
11:00	60	mL			
		11:30	0.5		
12:00	45	mL		mL/min	

Here, the volume is recorded only every hour, and no start time is available. However, it's reasonable to assume that the volume measurement corresponds to an hour. Next, we can see that the rate was titrated to 0.5, and for the period between 11:00 to 12:00 there was half an hour of delivery at 1 mL/min, and half an hour of delivery at 0.5 mL/min, resulting in a total volume of 45 mL delivered for the past hour.

Summing up, for CareVue data, the rate and volume will be asynchronous, and only the `CHARTTIME` will be available. For rates, the `CHARTTIME` will correspond to a start time (when the drug was set to that rate). For volumes, the `CHARTTIME` will correspond to an end time.

For Metavision data, there is no concept of a volume in the database: only a `RATE`. All inputs are recorded with a `STARTTIME` and an `ENDTIME`. As a result, the volumes in the database for Metavision patients are *derived* from the rates. Furthermore, exact start and stop times for the drugs are easily deducible.

## Details of the merging process

The difficulty in merging the databases arose due primarily to two factors: the lower resolution of information archiving in the CareVue system, and the different definition of an ‘order’ in the databases.

The aim of this section is to provide all the detail into how these data were merged: this information is not necessary to understand for the purposes of using the database, but will provide insight into the format of the IOEVENTS table. We welcome suggestions from the community on improving the format and usability of the table.

### Philips CareVue

The CareVue system stored input/output (IO) data across five tables: IOEVENTS, MEDEVENTS, ADDITIVES, SOLUTIONS and DELIVERIES. Each time a new order for a drug was recorded in the database, the ADDITIVES, SOLUTIONS and DELIVERIES tables would be populated with information regarding the order. The data archival format is best described with an example.

Patient A has been recently admitted to the ICU and is to be administered noradrenaline to restore their blood pressure to a value of at least 60 mmHg. The route of administration is intravenous, i.e. the drug is to be pumped into the patient’s blood stream directly through a line inserted in a vein. The nurse would prepare a solution of 250 mL sodium chloride (NaCl) to contain the drug. The SOLUTIONS table would consequently contain an entry of 250 mL NaCl for patient A at the time the nurse prepared the solution. The nurse would then mix in the drug into the solution: in this example let’s say 8 mg of noradrenaline. The 8 mg of noradrenaline would be recorded in the ADDITIVES table, and an identifier would be recorded which linked the 8 mg of noradrenaline to the 250 mL solution of NaCl. Finally, the nurse would optionally set an initial delivery rate and route: in this case it could be 10 mL/hr intravenously. These would populate the RATE and ROUTE columns in the DELIVERIES table. Note that DELIVERIES data is less frequently present and was not consistently recorded.

Now that the solution has been prepared, the nurse can begin administering the drug to the patient. The time is now 18:20. An initial entry of 0 mL is recorded in the IOEVENTS table at 18:00 (this usually occurs, but it has not been verified that this *always* occurs). The nurse begins at a rate of 1 mcg/kg/min. The MEDEVENTS contains an entry at a CHARTTIME of 18:20 for a rate of 1 mcg/kg/min. Five minutes later, at 18:25, the nurse notes that the blood pressure is still lower than the desired 60 mmHg and increases the dose. The nurse raises the dose to 2 mcg/kg/min, and MEDEVENTS records the new dose of 2 mcg/kg/min at 18:25. The nurse checks again and notes that the blood pressure has reached the target value of 60 mmHg and ceases titration of the drug. At 19:00, the volume of drug administered to the patient is recorded. If

the patient weighed 100 kg, then the amount of drug administered would be equal to:

$$5 \text{ min} * 1 \text{ mcg/kg/min} * 100 \text{ kg} + 25 \text{ min} * 2 \text{ mcg/kg/min} * 100 \text{ kg} = 500 \text{ mcg} + 5000 \text{ mcg} = 5500 \text{ mcg} = 5.5 \text{ mg}$$

The amount of solution administered would be equal to:

$$(5 \text{ min} * 1 \text{ mcg/kg/min} * 100 \text{ kg} + 25 \text{ min} * 2 \text{ mcg/kg/min} * 100 \text{ kg}) * 250 \text{ mL} / 8 \text{ mg} = 5.5 \text{ mg} * 250 \text{ mL} / 8 \text{ mg} = 171.875 \text{ mL}$$

Consequently, IOEVENTS would record 171.875 mL at 19:00. If the patient continued at the same rate for this drug, then IOEVENTS would record  $(60 * 2 * 100 * 250 / 8000) = 375 \text{ mL}$  at 20:00. Note that:

- Unless the rate is updated, no new entry in MEDEVENTS exists
- IOEVENTS always records a value on an hourly basis, regardless of how long the IO event has been present
- 375 mL is larger than the original 250 mL bag: somewhere during administration the nurse would need to replace the empty bag with a new solution of the same formulation

Note that the changing of the bag is sometimes, but not always, recorded in the ADDITIVES/SOLUTIONS/DELIVERIES table as a new order. It usually occurs if the drug has been discontinued, and then re-prescribed later in the patient's stay.

## Metavision

Metavision records IO data using two tables: RANGESIGNALS and ORDER-ENTRY. These tables do not appear in MIMIC-III as they have been merged to form the INPUTEVENTS\_MV table. RANGESIGNALS contains recorded data elements which last for a fixed period of time. Furthermore, the RANGESIGNALS table recorded information for each component of the drug separately. For example, for a noradrenaline administration there would be two components: a main order component (noradrenaline) and a solution component (NaCl). The STARTTIME and ENDTIME of RANGESIGNALS indicated when the drug started and finished. *Any* change in the drug rate would result in the current infusion ending, and a new STARTTIME being created.

Let's return to our example above of the patient being given noreadrenaline. The STARTTIME for the solution (NaCl) and the drug (noradrenaline) would be 18:20. The rate of the drug would be listed as 1 mcg/kg/min, and the rate of the solution would be listed as 10 mL/hr. The nurse, as before, decides to increase the drug rate at 18:25 to 2 mcg/kg/min. As a result, the ENDTIME for the two rows corresponding to the solution (NaCl and noreadrenaline) is set to 18:25. Two new rows are generated with a STARTTIME of 18:25. These two new rows would continue until either (i) the drug rate was changed or (ii) the

drug was delivery was discontinued. The `ORDERID` column would be identical for each instantiation of NaCl and noradrenaline which corresponded to the same solution/rate. That is, for the infusion given between 18:20 and 18:25, both NaCl and noreadrenaline would have the same `ORDERID`. `LINKORDERID` would further link the drug across all administrations, even when the rate is changed. The following table demonstrates these concepts:

Item	STARTTIME	ENDTIME	RATE	RATEUOM	ORDERID	LINKORDERID
NaCl	18:20	18:25	1	mcg/kg/min	8003	8003
Noradrenaline	18:20	18:25	10	ml/hr	8003	8003
NaCl	18:25	20:00	2	mcg/kg/min	8020	8003
Noradrenaline	18:25	20:00	20	ml/hr	8020	8003

Note that `ORDERID` links items occurring at the same time which correspond to the same solution, while `LINKORDERID` links all these solutions together across time. Note also that `LINKORDERID` is equal to the first `ORDERID` which occurs for the solution, as above.

```
+++ date = "2017-01-01T00:00:00-04:00" title = "MetaVision" linktitle =
"MetaVision" weight = 1 toc = "false"
```

```
[menu][menu.main] parent = "Data details"
```

```
+++
```

## MetaVision

MetaVision is a clinical information system provided by iMDSOft which archives and displays data at the bedside for patients cared in intensive care units as the Beth Israel Deaconess Medical Center. Part of the MIMIC-III database is extracted from the MetaVision system, the other part is extracted from the CareVue system. The immediate consequence of this is that data for the entire duration of the MIMIC-III database needs to be extracted using special steps of different `ITEMID` in order to get data from both systems.

MetaVision is frequently referenced in the MIMIC-III documentation because:

- Patients in the MetaVision system are admitted at a later date (2008 and onward)
- Patients in the MetaVision system have data archived a different format
- All `ITEMID` for MetaVision patients will differ from CareVue patients

```
+++ date = "2015-09-08T06:58:18-04:00" title = "Times" linktitle = "Times"
weight = 5 toc = "true"
```

```
[menu][menu.main] parent = "Data details"
```

+++

## Time types

Time in the database is stored with one of two suffixes: `TIME` and `DATE`. If a column has `TIME` as the suffix, e.g. `CHARTTIME`, then the data resolution is down to the minute. If the column has `DATE` as the suffix, e.g. `CHARTDATE`, then the data resolution is down to the day. That means that measurements in a `CHARTDATE` column will always have 00:00:00 as the hour, minute, and second values. This does *not* mean it was recorded at midnight: it indicates that we do not have the exact time, only the date.

## Date shifting

All dates in the database have been shifted to protect patient confidentiality. Dates will be internally consistent for the same patient, but randomly distributed in the future. Dates of birth which occur in the present time are *not* true dates of birth. Furthermore, dates of birth which occur before the year 1900 occur if the patient is older than 89. In these cases, the patient's age at their first admission has been fixed to 300.

## Time columns in the database

### `CHARTTIME` vs `STORETIME`

Most data, with the exception of patient related demographics, are recorded with a time indicating when the observation was made: `CHARTTIME`. `CHARTTIME` dates back to the use of paper charts: in order to facilitate efficient observations by nursing staff, the day was separated into hourly blocks, and observations were recorded within these hourly blocks. Thus, any time one performed a measurement between the hours of 04:00 and 05:00, the data would be charted in the 04:00 block, and so on. This concept has carried forward into the electronic recording of data: even if data is recorded at 04:23, in many cases it is still charted as occurring at 04:00.

`STORETIME` provides information on the recording of the data element itself. All observations in the database must be validated before they are archived into the patient medical record. The `STORETIME` provides the exact time that this validation occurred. For example, a heart rate may be charted at 04:00, but only validated at 04:40. This indicates that the care provider validated the measurement at 4:40 and indicated that it was a valid observation of the patient

at 04:00. Conversely, it's also possible that the `STORETIME` occurs *before* the `CHARTTIME`. While a Glasgow Coma Scale may be charted at a `CHARTTIME` of 04:00, the observation may have been made and validated slightly before (e.g. 3:50). Again, the validation implies that the care staff believed the measurement to be an accurate reflection of the patient status at the given `CHARTTIME`.

### **Summing up: CHARTTIME vs. STORETIME**

`CHARTTIME` is the time at which a measurement is *charted*. In almost all cases, this is the time which best matches the time of actual measurement. In the case of continuous vital signs (heart rate, respiratory rate, invasive blood pressure, non-invasive blood pressure, oxygen saturation), the `CHARTTIME` is usually exactly the time of measurement. `STORETIME` is the time at which the data is recorded in the database: logically it occurs after `CHARTTIME`, often by hours, but usually not more than that.

### **CHARTDATE**

`CHARTDATE` is equivalent to `CHARTTIME`, except it does not contain any information on the time (all hour, minute, and seconds are 0 for these measurements).

### **ADMITTIME, DISCHTIME, DEATHTIME**

`ADMITTIME` and `DISCHTIME` are the hospital admission and discharge times, respectively. `DEATHTIME` is the time of death of a patient if they died *in* hospital. If the patient did not die within the hospital for the given hospital admission, `DEATHTIME` will be null.

### **CREATETIME, UPDATETIME, ACKNOWLEDGETIME, OUTCOMETIME, FIRSTRESERVATIONTIME, CURRENTRESERVATIONTIME**

`CREATETIME` is the time at which an ICU discharge was requested for a given patient. `UPDATETIME` is the time which the ICU discharge request was updated. `ACKNOWLEDGETIME` was the time at which the discharge request was acknowledged by the transfers team. `OUTCOMETIME` is the time at which the ICU discharge request was completed (with an outcome of 'Discharged' or 'Canceled'). `FIRSTRESERVATIONTIME` and `CURRENTRESERVATIONTIME` only occur for patients who require certain locations in the hospital.

## INTIME, OUTTIME

INTIME and OUTTIME provide the time at which a patient entered and exited the given unit. In the ICUSTAYS table, the unit is always an ICU. In the TRANSFERS table, the unit can be any ward in the hospital.

## STARTTIME, ENDTIME

For events which occur over a period of time, STARTTIME and ENDTIME provide the beginning and end time of the event. For medical infusions, these columns indicate the period over which the substance was administered.

## COMMENTS\_DATE

COMMENTS\_DATE provides the time at which a cancel or edit comment was made for a given order.

## DOB, DOD, DOD\_HOSP, DOD\_SSN

DOB is the patient's date of birth. If the patient is older than 89, their date of birth is set to 300 at their first admission. DOD is the patient's date of death: sourced either from the hospital database (DOD\_HOSP) or the social security database (DOD\_SSN).

## TRANSFERTIME

TRANSFERTIME is the time at which the patient's service changes.

```
+++ date = "2015-10-25T13:37:00-04:00" title = "Waveforms" linktitle = "Waveforms" weight = 8 toc = "true"
```

```
[menu][menu.main] parent = "Data details"
```

```
+++
```

## Waveforms vs. clinical data

This website provides documentation for the MIMIC-III Clinical Database. A separate, complementary resource named the "MIMIC-III Waveform Database" contains high resolution waveforms and numerics (second by second derived values from the waveforms). **Work to link waveform data with the MIMIC-III**

**Clinical Database is still underway. Please check here for updates on our progress.**

```
+++ date = "2015-09-01T13:37:00-04:00" title = "MIMIC-II to MIMIC-III"
linktitle = "MIMIC-II to MIMIC-III" weight = 6 toc = "true"
```

```
[menu][menu.main] parent = "Data details"
```

```
+++
```

## MIMIC-II vs MIMIC-III

MIMIC-III is an extension of MIMIC-II: it incorporates the data contained in MIMIC-II (collected between 2001 - 2008) and augments it with newly collected data between 2008 - 2012. In addition, many data elements have been regenerated from the raw data in a more robust manner to improve the quality of the underlying data.

One of the challenges of adding new data resulted from a change in data management software at the Beth Israel Deaconess Medical Center. The original Philips CareVue system (which archived data from 2001 - 2008) was replaced with the new Metavision data management system (which continues to be used to the present). This page aims to facilitate the transition for researchers familiar with MIMIC-II who would like to continue their research with the updated MIMIC-III.

## ITEMID mapping

In MIMIC-II there were multiple tables containing the same column name, ITEMID, but referring to different concepts. In attempt to alleviate confusion, we have merged all these tables into a single table, as follows:

Old table	Merged into
D_CHARTITEMS	D_ITEMS
D_IOITEMS	D_ITEMS
D_MEDITEMS	D_ITEMS
D_CODEDITEMS	<i>deleted</i>
D_PARAMMAP_ITEMS	<i>deleted</i>
D_LABITEMS	D_LABITEMS

D\_CHARTITEMS, D\_IOITEMS, and D\_MEDITEMS were all sourced from the same data source: the ICU database (specifically Philips CareVue). In contrast, iMDSoft Metavision only has a single table to define most ITEMID

concepts. In order to simplify the schema and coalesce the databases, it was decided to merge together all the ITEMID dictionary tables into a single table, *except* D\_LABITEMS (see below).

As the ITEMID ranges in the original D\_ tables overlapped, they were offset by constant values. The following table maps the old ITEMID value ranges to the new ranges:

MIMIC-II source	Old range	New range	Offset
D_CHARTITEMS	1 - 20009	1-20009	<i>None</i>
D_MEDITEMS	1 - 405	30001 - 30405	+ 30000
D_IOITEMS	1 - 6807	40001 - 46807	+ 40000

For example, the charted item “Heart Rate” had an ITEMID of 211 in MIMIC-II (in D\_CHARTITEMS). The new ITEMID for this is, again, 211, as there was no offset for D\_CHARTITEMS. This means that any ITEMIDs used in the D\_CHARTITEMS table will be directly portable to MIMIC-III (caveat: you will still need to extract additional ITEMID for the new metavision patients).

Conversely, if we look at the ITEMID = 51 in MIMIC-II D\_MEDITEMS (vasopressin), we can find the same concept in MIMIC-III as  $51 + 30000 = 30051$ .

## D\_CODEDITEMS

D\_CODEDITEMS contained many concepts - most of these have been unchanged, simply moved to the new table D\_ITEMS. The following table provides details.

ITEMID range	Concept	MIMIC-III?
60001 - 61018	DRG codes	Deleted
		DRGs are stored with descriptions in DRGCODES

---

	Where
	is
	it in
ITEMID	MIMIC-
range	Concept?
70001	Microbiology
-	specimens
70093	ITEMID
	val-
	ues
	are
	<b>un-</b>
	<b>changed</b>
80001	Microbiology
-	organisms
80312	ITEMID
	val-
	ues
	are
	<b>un-</b>
	<b>changed</b>
90001	Microbiology
-	antibacterium
90031	ITEMID
	val-
	ues
	are
	<b>un-</b>
	<b>changed</b>
100001	ICD- Deleted
-	9 -
101885	pro- ICD9
	cedure codes
	are
	codes stored
	with
	descriptions

---

## Lab ITEMIDs - not merged into D\_ITEMS

The ITEMID for laboratory measurements in the D\_LABITEMS and LABEVENTS tables in MIMIC-II do *not* match the ITEMID for laboratory measurements in MIMIC-III. We have provided a mapping table to facilitate the updating of queries which use this table. The mapping can be found in the

MIMIC Code Repository:

Much of the data has been mapped to LOINC codes, which provide a standard ontology for recorded lab values. Careful inspection shows that the LOINC code for an ITEMID in MIMIC-III is, in rare occasions, different from the LOINC code for the same concept in MIMIC-II. This is usually attributable to the laboratory assigning a new LOINC code, which is done for many reasons, including changing the reagents of a laboratory test, changing the technique used to acquire the result or because the previous LOINC code was discontinued.

## New ITEMIDs

While almost all concepts are unified into a single table (D\_ITEMS), the concepts within that table are *not* unified. As patients admitted between 2001-2008 used a different ICU system to patients admitted between 2008-2012, the ITEMID values for the same concept differ in these two time periods. That is, for heart rates between 2001-2008, the ITEMID value 211 is appropriate. For heart rates for patients admitted between 2008 and onward, the ITEMID 220045 is appropriate. Most concepts will require the use of multiple ITEMID values in order to completely extract the data.

## Schema changes

### ADMISSIONS

ADMISSIONS is now sourced from the hospital database, rather than the ICU database. Changes include:

- Admission and discharge dates now have the time component
- Discharge location is now available
- Diagnosis on admission is now available
- ED registration and exit time are now available

### CENSUSEVENTS replaced by TRANSFERS

The CENSUSEVENTS table was used in MIMIC-II to track patient hospital admissions. The original source of this table was the ICU database which recorded when patients were admitted and discharged from the ICU. In MIMIC-III, this table has been removed and replaced with the TRANSFERS table. The TRANSFERS table is sourced from the hospital admission, discharge, transfer (ADT) data. This has a number of advantages:

- The ADT data tracks a patient throughout the *entire* hospital stay, providing greater granularity and easier tracking of a patient's hospital course

- The ADT data provides information regarding ward location
- The ADT data has fewer erroneous admissions: frequently the ICU database contained erroneous entries corresponding to accidental admission/discharges
- The ADT data is available for all patients in the ICU database

## **DEMOGRAPHIC\_DETAIL merged into ADMISSIONS**

The DEMOGRAPHIC\_DETAIL provided extra static information regarding a patient which rarely changed throughout an admission (e.g. age, ethnicity). This data was originally sourced from the ICU database. The new ADMISSIONS table is sourced entirely from the hospital database, and contained the same set of demographics. Instead of creating a new table for these demographics, it was decided to maintain the demographics within the ADMISSIONS table in the same format as it exists in the raw data.

## **DRGEVENTS renamed DRGCODES**

The DRGEVENTS table has been renamed DRGCODES. The table still corresponds to diagnostic related groups, however the clarity of the data has been improved. First, a column corresponding to the DRG system has been provided. Second, if available, additional information regarding the DRG severity and mortality risk have been added. Finally, the DRG code has been mapped to a description based upon the underlying version. Due to frequent updates in the DRG coding system, it is non-trivial to map DRG codes to a description. To ease the use of this data, each DRG code has been mapped to a free text description which has been added to the table.

## **ICD9 renamed DIAGNOSES\_ICD**

To clarify the content of the table, the ICD9 table has been renamed to DIAGNOSES\_ICD. The table contains diagnoses sourced from the hospital database codified by ICD, usually ICD-9.

## **IOEVENTS, ADDITIVES, DELIVERIES and MEDEVENTS**

Data in the IOEVENTS and MEDEVENTS tables is now contained in the OUTPUTEVENTS, INPUTEVENTS\_CV and INPUTEVENTS\_MV tables. As all the medications in the MEDEVENTS table were continuous infusions, they were all associated with an entry in IOEVENTS. MEDEVENTS would

specify the drug rate, while IOEVENTS would specify the volume given. These tables have been consolidated to ease querying for drug deliveries.

## **POE\_MED and POE\_ORDER merged into PRESCRIPTIONS**

The term POE, or provider order entry, references a hospital specific database which users may not be familiar with. To clarify the content of these tables, they have been merged into a single table named PRESCRIPTIONS.

## **Identifier changes**

### **SUBJECT\_ID, HADM\_ID and ICUSTAY\_ID**

SUBJECT\_ID between MIMIC-II v2.6 and MIMIC-III have been kept consistent, for example, SUBJECT\_ID 788 corresponds to the same patient in MIMIC-II v2.6 as it does in MIMIC-III.

HADM\_ID have been regenerated in MIMIC-III. HADM\_ID in MIMIC-II v2.6 will *not* match any HADM\_ID in MIMIC-III. The newly generated HADM\_ID range from 100,000 - 199,999 to help differentiate these IDs from others.

ICUSTAY\_ID have been regenerated in MIMIC-III. ICUSTAY\_ID in MIMIC-II v2.6 will *not* match any ICUSTAY\_ID in MIMIC-III. Note that the newly generated ICUSTAY\_ID range is between 200,000 - 299,999 to prevent confusion with other IDs.

## **New tables**

### **CALLOUT**

The CALLOUT table contains information about ICU discharge planning and execution. Each patient is “called out” of the ICU: which involves alerting hospital administrative staff that a bed, usually on the floor, is required for a patient currently in the ICU. The call out event also includes any precautions that the patient may require (such as susceptibility to MRSA or respiratory support). The table provides information both on when the patient was deemed ready for discharge and when the patient actually left the ICU.

## **PROCEDURES\_ICD**

ICD-9 codes for procedures are now available in the PROCEDURES\_ICD table.

## **INPUTEVENTS\_CV and INPUTEVENTS\_MV**

Two different monitoring systems were operating in the hospital over the data collection period. The systems - Metavision and CareVue - recorded data in very different ways. We therefore made the decision not to merge the data, and instead provided two separate tables (INPUTEVENTS\_CV and INPUTEVENTS\_MV).

## **OUTPUTEVENTS**

Data about outputs were recorded in a consistent fashion for the Metavision and CareVue databases. Therefore, we merged this data into one table: OUTPUTEVENTS.

## **Removed tables**

There are many tables in MIMIC-II which are no longer present in MIMIC-III. In most cases, these tables were generated from the raw data for user convenience. We have transitioned from the approach of creating flat files of these tables to providing the source code necessary to regenerate them. This has two advantages: first it is much more efficient in terms of data transfer, and second it clarifies that these data are not “raw” in that they are not acquired directly from the databases but rather synthesized views of this data.

## **COMORBIDITY\_SCORES**

This table is frequently used to define the comorbid status for patients. Code for generating this table will be provided in the MIMIC Code Repository. The comorbidities will be defined using ICD-9 codes and DRG codes as proposed by Elixhauser et al.

## **DEMOGRAPHICEVENTS, D\_DEMOGRAPHICITEMS**

These tables were specific to the older CareVue database. As these tables are not present in the Metavision data, and the same information has been acquired

from the hospital database (see the ADMISSIONS and D\_PATIENTS tables), these tables have been removed.

## **D\_CAREUNITS**

The care unit identifier, CUID, has been removed from the various tables as it was unavailable in the Metavision ICU database. Care unit can be determined at the patient level, rather than the observation level, using the ICUSTAYEVENTS and TRANSFERS tables.

## **D\_CODEDITEMS**

This was a generated table to facilitate the interpretation of various coding systems, including microbiology, DRG, etc. The database has been restructured to have explicit definitions for these codes where appropriate.

## **D\_PARAMMAP\_ITEMS**

This table is no longer needed as ITEMID concepts have been consolidated in the D\_ITEMS table.

## **ICUSTAY\_DETAILS**

A script to generate ICUSTAY\_DETAILS will be provided in the MIMIC Code Repository shortly.

## **PARAMETER\_MAPPING**

This table is no longer needed as ITEMID concepts have been consolidated in the D\_ITEMS table.

## **WAVEFORM\_\*, D\_WAVEFORM\_SIGNALS**

D\_WAVEFORM\_SIGNALS, WAVEFORM\_METADATA, WAVEFORM\_SEGMENTS, WAVEFORM\_SEG\_SIG, WAVEFORM\_SIGNALS, WAVEFORM\_TRENDS,

WAVEFORM\_TREND\_SIGNALS have been removed. The mapping to the waveform data is no longer provided within the relative database for clarity.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "ADMISSIONS" linktitle =  
"ADMISSIONS" weight = 10 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC" +++
```

## The admissions table

**Table source:** Hospital database.

**Table purpose:** Define a patient's hospital admission, HADM\_ID.

**Number of rows:** 58976

**Links to:**

- PATIENTS on SUBJECT\_ID

## Brief summary

The ADMISSIONS table gives information regarding a patient's admission to the hospital. Since each unique hospital visit for a patient is assigned a unique HADM\_ID, the ADMISSIONS table can be considered as a definition table for HADM\_ID. Information available includes timing information for admission and discharge, demographic information, the source of the admission, and so on.

## Important considerations

- The data is sourced from the admission, discharge and transfer database from the hospital (often referred to as 'ADT' data).
- Organ donor accounts are sometimes created for patients who died in the hospital. These are distinct hospital admissions with very short, sometimes negative lengths of stay. Furthermore, their DEATHTIME is frequently the same as the earlier patient admission's DEATHTIME.
- All text data, except for that in the INSURANCE column, is stored in upper case.

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
ADMITTIME	TIMESTAMP(0)
DISCHTIME	TIMESTAMP(0)
DEATHTIME	TIMESTAMP(0)
ADMISSION_TYPE	VARCHAR(50)
ADMISSION_LOCATION	VARCHAR(50)
DISCHARGE_LOCATION	VARCHAR(50)
INSURANCE	VARCHAR(255)
LANGUAGE	VARCHAR(10)
RELIGION	VARCHAR(50)
MARITAL_STATUS	VARCHAR(50)
ETHNICITY	VARCHAR(200)
EDREGTIME	TIMESTAMP(0)
EDOUTTIME	TIMESTAMP(0)
DIAGNOSIS	VARCHAR(300)
HOSPITAL_EXPIRE_FLAG	TINYINT
HAS_CHARTEVENTS_DATA	TINYINT

## Detailed description

The `ADMISSIONS` table defines all `HADM_ID` present in the database, covering an admission period between 1 June 2001 and 10 October 2012.

### `SUBJECT_ID`, `HADM_ID`

Each row of this table contains a unique `HADM_ID`, which represents a single patient's admission to the hospital. `HADM_ID` ranges from 1000000 - 1999999. It is possible for this table to have duplicate `SUBJECT_ID`, indicating that a single patient had multiple admissions to the hospital. The `ADMISSIONS` table can be linked to the `PATIENTS` table using `SUBJECT_ID`.

### `ADMITTIME`, `DISCHTIME`, `DEATHTIME`

`ADMITTIME` provides the date and time the patient was admitted to the hospital, while `DISCHTIME` provides the date and time the patient was discharged from the hospital. If applicable, `DEATHTIME` provides the time of in-hospital death for the patient. Note that `DEATHTIME` is only present if the patient died in-hospital, and is almost always the same as the patient's `DISCHTIME`. However, there can be some discrepancies due to typographical errors.

## **ADMISSION\_TYPE**

ADMISSION\_TYPE describes the type of the admission: 'ELECTIVE', 'URGENT', 'NEWBORN' or 'EMERGENCY'. Emergency/urgent indicate unplanned medical care, and are often collapsed into a single category in studies. Elective indicates a previously planned hospital admission. Newborn indicates that the HADM\_ID pertains to the patient's birth.

## **ADMISSION\_LOCATION**

ADMISSION\_LOCATION provides information about the previous location of the patient prior to arriving at the hospital. There are 9 possible values:

- EMERGENCY ROOM ADMIT
- TRANSFER FROM HOSP/EXTRAM
- TRANSFER FROM OTHER HEALT
- CLINIC REFERRAL/PREMATURE
- \*\* INFO NOT AVAILABLE \*\*
- TRANSFER FROM SKILLED NUR
- TRSF WITHIN THIS FACILITY
- HMO REFERRAL/SICK
- PHYS REFERRAL/NORMAL DELI

The truncated text occurs in the raw data.

## **INSURANCE, LANGUAGE, RELIGION, MARITAL\_STATUS, ETHNICITY**

The INSURANCE, LANGUAGE, RELIGION, MARITAL\_STATUS, ETHNICITY columns describe patient demographics. These columns occur in the ADMISSIONS table as they are originally sourced from the admission, discharge, and transfers (ADT) data from the hospital database. The values occasionally change between hospital admissions (HADM\_ID) for a single patient (SUBJECT\_ID). This is reasonable for some fields (e.g. MARITAL\_STATUS, RELIGION), but less reasonable for others (e.g. ETHNICITY).

## **EDREGTIME, EDOUTTIME**

Time that the patient was registered and discharged from the emergency department.

## DIAGNOSIS

The `DIAGNOSIS` column provides a preliminary, free text diagnosis for the patient on hospital admission. The diagnosis is usually assigned by the admitting clinician and does not use a systematic ontology. As of MIMIC-III v1.0 there were 15,693 distinct diagnoses for 58,976 admissions. The diagnoses can be very informative (e.g. chronic kidney failure) or quite vague (e.g. weakness). Final diagnoses for hospital admissions are coded and can be found in the `DIAGNOSES_ICD` table.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "CALLOUT" linktitle =  
"CALLOUT" weight = 11 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The callout table

**Table source:** Hospital database.

**Table purpose:** Provides information when a patient was `READY` for discharge from the ICU, and when the patient was actually discharged from the ICU.

**Number of rows:** 34499

**Links to:**

- `PATIENTS` on `SUBJECT_ID`
- `ADMISSIONS` on `HADM_ID`

## Brief summary

The `CALLOUT` table provides information about ICU discharge planning. When a patient is deemed ready to leave the ICU, they are “called out”. This process involves: (i) a care provider registering that the patient is ready to leave the ICU and detailing any specialized precautions required, (ii) a coordinator acknowledging the patient requires a bed outside the ward, (iii) a variable period of time in order to coordinate the transfer, and finally (iv) an outcome: either the patient is called out (discharged) or the call out event is canceled. This table provides information for all of the above.

## Important considerations

- Call out data is *not* available for all adult patients, as the data collection only began part way through the collection of the MIMIC database
- Call out data is *never* available for neonates

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
SUBMIT_WARDID	INT
SUBMIT_CAREUNIT	VARCHAR(15)
CURR_WARDID	INT
CURR_CAREUNIT	VARCHAR(15)
CALLOUT_WARDID	INT
CALLOUT_SERVICE	VARCHAR(10)
REQUEST_TELE	SMALLINT
REQUEST_RESP	SMALLINT
REQUEST_CDIFF	SMALLINT
REQUEST_MRSA	SMALLINT
REQUEST_VRE	SMALLINT
CALLOUT_STATUS	VARCHAR(20)
CALLOUT_OUTCOME	VARCHAR(20)
DISCHARGE_WARDID	INT
ACKNOWLEDGE_STATUS	VARCHAR(20)
CREATETIME	TIMESTAMP(0)
UPDATETIME	TIMESTAMP(0)
ACKNOWLEDGETIME	TIMESTAMP(0)
OUTCOMETIME	TIMESTAMP(0)
FIRSTRESERVATIONTIME	TIMESTAMP(0)
CURRENTRESERVATIONTIME	TIMESTAMP(0)

## Detailed description

The CALLOUT table provides information for “call out events”. A call out event occurs as follows: first, a patient is ready for discharge from the ICU. A care provider creates a call out request (occurring at `CREATETIME`) stating the desired ward and any necessary precautions. The call out request is acknowledged, usually within a short period of time (`ACKNOWLEDGETIME`). When the patient is actually

transferred out of the ICU, the `CALLOUT_OUTCOME` becomes 'Discharged' and the `OUTCOMETIME` column contains the time at which the patient was discharged.

#### **SUBJECT\_ID, HADM\_ID**

`SUBJECT_ID` and `HADM_ID` define the patient and hospital admission corresponding to the given call out event. ? Unique

#### **SUBMIT\_WARDID, SUBMIT\_CAREUNIT**

`SUBMIT_WARDID` identifies the ward from which the request was submitted. `SUBMIT_CAREUNIT` indicates whether the `SUBMIT_WARDID` corresponds to an ICU cost center, and if so, what type of ICU cost center.

#### **CURR\_WARDID, CURR\_CAREUNIT**

`CURR_WARDID` identifies the ward in which the patient resides when called out (i.e. prior to discharge/transfer). `CURR_CAREUNIT` indicates which ICU cost center the `CURR_WARDID` corresponds to (note: since all patients are being discharged from an ICU, all patients should reside in an ICU cost center).

#### **CALLOUT\_WARDID, CALLOUT\_SERVICE**

`CALLOUT_WARDID` identifies the ward to which the patient should be discharged. Note that `CALLOUT_WARDID = 0` represents 'Home' and `CALLOUT_WARDID = 1` represents 'First available ward'. The remaining IDs correspond to distinct wards in the hospital. `CALLOUT_SERVICE` is the service under which the patient should be discharged.

#### **REQUEST\_TELE, REQUEST\_RESP, REQUEST\_CDIFF, REQUEST\_MRSA, REQUEST\_VRE**

The request columns are binary indicators which request certain precautions for the patient in the subsequent ward where they will reside. For example, `MRSA` means that the patient is colonized or infected with MRSA, a drug resistant source of hospital acquired infections.

## **CALLOUT\_STATUS, CALLOUT\_OUTCOME**

CALLOUT\_STATUS indicates whether the call out is still active or not: if a call out is answered it should be flagged as inactive. CALLOUT\_OUTCOME is either ‘Discharged’ or ‘Cancelled’, indicating whether the patient finally called out (i.e. discharged) or not.

## **DISCHARGE\_WARDID**

DISCHARGE\_WARDID indicates the ward to which the patient was actually discharged. DISCHARGE\_WARDID = 0 indicates home and other values correspond to distinct wards in the hospital.

## **ACKNOWLEDGE\_STATUS**

ACKNOWLEDGE\_STATUS indicates the response to the callout event: ‘Acknowledged’, ‘Revised’, ‘Unacknowledged’ or ‘Reactivated’.

## **CREATETIME, UPDATETIME, ACKNOWLEDGETIME, OUTCOMETIME, FIRSTRESERVATIONTIME, CURRENTRESERVATIONTIME**

CREATETIME provides the time and date that the call out was initiated. UPDATETIME provides the last time at which the call out event was updated. ACKNOWLEDGETIME is the time at which the call out was first acknowledged. OUTCOMETIME is the time at which the CALLOUT\_OUTCOME occurred. FIRSTRESERVATIONTIME and CURRENTRESERVATIONTIME provide information regarding ward reservations.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "CAREGIVERS" linktitle =  
"CAREGIVERS" weight = 12 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## **The caregivers table**

**Table source:** CareVue and Metavision ICU databases.

**Table purpose:** Defines the role of caregivers.

**Number of rows:** 7567

**Links to:**

- CHARTEVENTS on CGID

## Brief summary

This table provides information regarding care givers. For example, it would define if a care giver is a research nurse (RN), medical doctor (MD), and so on.

## Table columns

Name	Postgres data type
ROW_ID	INT
CGID	INT
LABEL	VARCHAR(15)
DESCRIPTION	VARCHAR(30)

## Detailed Description

The CAREGIVERS table provides information regarding the type of caregiver. Each caregiver is represented by a unique integer which maps to this table.

### CGID

CGID is a unique identifier for each distinct caregiver present in the database. CGID is sourced from two tables in the raw data: the CareVue and Metavision ICU databases. Due to imprecision in the storage of unique identifiers across the database, it is possible that two distinct caregivers with the same names (e.g. RN Sarah Jones and MD Sarah Jones) would be considered as the same caregiver. However, this is an unlikely occurrence.

### LABEL

LABEL defines the type of caregiver: e.g. RN, MD, PharmD, etc. Note that LABEL is a free text field and as such contains many typographical errors and spelling variants of the same concept (e.g. MD, MDs, M.D.).

## DESCRIPTION

DESCRIPTION is less frequently present than LABEL, and provides additional information regarding the caregiver. This column is much more structured, and contains only 17 unique values as of MIMIC-III v1.0.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "CHARTEVENTS" linktitle = "CHARTEVENTS" weight = 13 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC" +++
```

## The chartevents table

**Table source:** CareVue and Metavision ICU databases.

**Table purpose:** Contains all charted data for all patients.

**Number of rows:** 330,712,483

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- ICUSTAYS on ICUSTAY\_ID
- D\_ITEMS on ITEMID
- CAREGIVERS on CGID

## Brief summary

CHARTEVENTS contains all the charted data available for a patient. During their ICU stay, the primary repository of a patient's information is their electronic chart. The electronic chart displays patients' routine vital signs and any additional information relevant to their care: ventilator settings, laboratory values, code status, mental status, and so on. As a result, the bulk of information about a patient's stay is contained in CHARTEVENTS. Furthermore, even though laboratory values are captured elsewhere (LABEVENTS), they are frequently repeated within CHARTEVENTS. This occurs because it is desirable to display the laboratory values on the patient's electronic chart, and so the values are copied from the database storing laboratory values to the database storing the CHARTEVENTS.

## Important considerations

- Some items are duplicated between the labevents and chartevents tables. In cases where there is disagreement between measurements, labevents should be taken as the ground truth.

## Table columns

Name	Postgres data type	In CareVue	In Metavision
ROW_ID	INT	Y	Y
SUBJECT_ID	NUMBER(7,0)	Y	Y
HADM_ID	NUMBER(7,0)	Y	Y
ICUSTAY_ID	NUMBER(7,0)	Y	Y
ITEMID	NUMBER(7,0)	Y	Y
CHARTTIME	DATE	Y	Y
STORETIME	DATE	Y	Y
CGID	NUMBER(7,0)	Y	Y
VALUE	VARCHAR2(200 BYTE)	Y	Y
VALUENUM	NUMBER	Y	Y
VALUEUOM	VARCHAR2(20 BYTE)	Y	Y
WARNING	NUMBER(1,0)		Y
ERROR	NUMBER(1,0)		Y
RESULTSTATUS	VARCHAR2(20 BYTE)	Y	
STOPPED	VARCHAR2(20 BYTE)	Y	

## Detailed Description

### SUBJECT\_ID, HADM\_ID, ICUSTAY\_ID

Identifiers which specify the patient: SUBJECT\_ID is unique to a patient, HADM\_ID is unique to a patient hospital stay and ICUSTAY\_ID is unique to a patient ICU stay.

### ITEMID

Identifier for a single measurement type in the database. Each row associated with one ITEMID (e.g. 212) corresponds to an instantiation of the same measurement (e.g. heart rate).

## **CHARTTIME, STORETIME**

**CHARTTIME** records the time at which an observation was made, and is usually the closest proxy to the time the data was actually measured. **STORETIME** records the time at which an observation was manually input or manually validated by a member of the clinical staff.

## **CGID**

**CGID** is the identifier for the caregiver who validated the given measurement.

## **VALUE, VALUENUM**

**VALUE** contains the value measured for the concept identified by the **ITEMID**. If this value is numeric, then **VALUENUM** contains the same data in a numeric format. If this data is not numeric, **VALUENUM** is null. In some cases (e.g. scores like Glasgow Coma Scale, Richmond Sedation Agitation Scale and Code Status), **VALUENUM** contains the score and **VALUE** contains the score and text describing the meaning of the score.

## **VALUEUOM**

**VALUEUOM** is the unit of measurement for the **VALUE**, if appropriate.

## **WARNING, ERROR**

**WARNING** and **ERROR** are Metavision specific columns which specify if a warning for the value was raised and if an error occurred during the measurement.

## **RESULTSTATUS, STOPPED**

**RESULTSTATUS** and **STOPPED** are CareVue specific columns which specify the type of measurement (**RESULTSTATUS** is 'Manual' or 'Automatic') and whether the measurement was stopped.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "CPTEVENTS" linktitle =  
"CPTEVENTS" weight = 14 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The cpevents table

**Table source:** Hospital database.

**Table purpose:** Contains current procedural terminology (CPT) codes, which facilitate billing for procedures performed on patients.

**Number of rows:** 573146

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID

## Brief summary

The CPTEVENTS table contains a list of which current procedural terminology codes were billed for which patients. This can be useful for determining if certain procedures have been performed (e.g. ventilation).

## Important considerations

- The respiratory cost center bills for ventilation *regardless* of duration. That means that 30 minutes of mechanical ventilation for a single day would result in same billing code as a full 24 hours of mechanical ventilation.
- Non-invasive ventilation and mechanical ventilation use the same CPT code. Differentiating these concepts requires selecting them using the DESCRIPTION column.

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
COSTCENTER	VARCHAR(10)
CHARTDATE	TIMESTAMP(0)
CPT_CD	VARCHAR(10)
CPT_NUMBER	INT
CPT_SUFFIX	VARCHAR(5)
TICKET_ID_SEQ	INT
SECTIONHEADER	VARCHAR(50)

Name	Postgres data type
SUBSECTIONHEADER	VARCHAR(300)
DESCRIPTION	VARCHAR(200)

## Detailed Description

CPTEVENTS contains current procedural terminology (CPT) codes for patients as billed through either the ICU cost center or the respiratory cost center. Each code represents a distinct procedure performed on the patient during their ICU stay.

### **SUBJECT\_ID, HADM\_ID**

Identifiers which specify the patient: **SUBJECT\_ID** is unique to a patient and **HADM\_ID** is unique to a patient hospital stay.

### **COSTCENTER**

**COSTCENTER** is the cost center which billed for the corresponding CPT codes. There are two possible cost centers: 'ICU' and 'Resp'. 'Resp' codes correspond to mechanical or non-invasive ventilation and were billed by the respiratory therapist. 'ICU' codes correspond to the procedures billed for by the ICU.

### **CHARTDATE**

The date at which the procedure occurred.

### **CPT\_CD, CPT\_NUMBER, CPT\_SUFFIX**

**CPT\_CD** contains the original CPT code. **CPT\_NUMBER** is a numeric version of the **CPT\_CD** column, which allows for easier range comparisons in querying. However, note not all **CPT\_CD** are fully numeric. The **CPT\_SUFFIX** column contains the text suffix when the **CPT\_CD** contains non-numeric characters.

### **TICKET\_ID\_SEQ**

The order of the **CPT\_CD**.

## SECTIONHEADER, SUBSECTIONHEADER

The section headers provide a category for the given CPT code. These headers were assigned using the D\_CPT table.

## DESCRIPTION

In the case of a CPT\_CD corresponding to the respiratory cost center, the description provides information about the meaning of the CPT code. Otherwise, the field is null.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "D_CPT" linktitle = "D_CPT"  
weight = 15 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The d\_cpt table

**Table source:** Online definitions.

**Table purpose:** High-level definitions for current procedural terminology (CPT) codes.

**Number of rows:** 134

**Links to:**

- CPTEVENTS on CPT\_CD between MINCODEINSUBSECTION and MAXCODEINSUBSECTION

## Brief summary

This table gives some high level information regarding current procedural terminology (CPT) codes. Unfortunately, detailed information for individual codes is unavailable.

## Important considerations

- Unlike all other definition tables, D\_CPT does *not* have a one to one mapping with the corresponding CPT\_CD in CPTEVENTS, rather each row of D\_CPT maps to a range of CPT\_CD.

## Table columns

Name	Postgres data type
ROW_ID	INT
CATEGORY	SMALLINT
SECTIONRANGE	VARCHAR(100)
SECTIONHEADER	VARCHAR(50)
SUBSECTIONRANGE	VARCHAR(100)
SUBSECTIONHEADER	VARCHAR(300)
CODESUFFIX	VARCHAR(5)
MINCODEINSUBSECTION	INT
MAXCODEINSUBSECTION	INT

## Detailed Description

D\_CPT provides information about CPT codes, specifically it provides the overall purpose of the procedure and in some cases the body system related to the procedure.

### CATEGORY

CATEGORY is a integer which identifies the category of the CPT code.

### SECTIONRANGE, SECTIONHEADER

SECTIONRANGE defines the range of codes for the given section, and SECTIONHEADER provides the description of the given section. There are 8 possible sections:

- Evaluation and management
- Surgery
- Radiology
- Anesthesia
- Emerging technology
- Pathology and laboratory
- Performance measurement
- Medicine

Note that the evaluation and management tends to represent administrative or generic costs.

## SUBSECTIONRANGE, SUBSECTIONHEADER

Similarly as for the sections, `SUBSECTIONRANGE` defines the range of codes for the given subsection, and `SUBSECTIONHEADER` provides the description of the given subsection. The subsection provides extra detail which can be useful, for example, when the section header is ‘Anesthesia’ the subsection provides information on the general anatomical region for the anesthesia.

## MINCODEINSUBSECTION, MAXCODEINSUBSECTION

These columns provide numeric representations of minimum and maximum value in the `SUBSECTIONRANGE` column. This facilitates joining the `CPTEVENTS` table to the `D_CPT` table on `CPT_CD`.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "D_ICD_DIAGNOSES" link-  
title = "D_ICD_DIAGNOSES" weight = 16 toc = "true"  
[menu][menu.main] parent = "Tables in MIMIC"  
+++
```

## The `d_icd_diagnoses` table

**Table source:** Online sources.

**Table purpose:** Definition table for ICD diagnoses.

**Number of rows:** 14,567

**Links to:**

- `DIAGNOSES_ICD ON ICD9_CODE`

## Brief summary

This table defines International Classification of Diseases Version 9 (ICD-9) codes for **diagnoses**. These codes are assigned at the end of the patient’s stay and are used by the hospital to bill for care provided.

## Table columns

Name	Postgres data type
<code>ROW_ID</code>	<code>INT</code>

Name	Postgres data type
ICD9_CODE	VARCHAR(10)
SHORT_TITLE	VARCHAR(50)
LONG_TITLE	VARCHAR(300)

## Detailed Description

### ICD9\_CODE

ICD9\_CODE is the International Coding Definitions Version 9 (ICD-9) code. Each code corresponds to a single diagnostic concept.

### SHORT\_TITLE, LONG\_TITLE

The title fields provide a brief definition for the given diagnosis code in ICD9\_CODE.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "D_ICD_PROCEDURES"
linktitle = "D_ICD_PROCEDURES" weight = 17 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The d\_icd\_procedures table

**Table source:** Online sources.

**Table purpose:** Definition table for ICD procedures.

**Number of rows:** 3,882

**Links to:**

- PROCEDURES\_ICD on ICD9\_CODE

## Brief summary

This table defines International Classification of Diseases Version 9 (ICD-9) codes for **procedures**. These codes are assigned at the end of the patient's stay and are used by the hospital to bill for care provided. They can further be used to identify if certain procedures have been performed (e.g. surgery).

## Table columns

Name	Postgres data type
ROW_ID	INT
ICD9_CODE	VARCHAR(10)
SHORT_TITLE	VARCHAR(50)
LONG_TITLE	VARCHAR(300)

## Detailed Description

### ICD9\_CODE

ICD9\_CODE is the International Coding Definitions Version 9 (ICD-9) code. Each code corresponds to a single procedural concept.

### SHORT\_TITLE, LONG\_TITLE

The title fields provide a brief definition for the given procedure code in ICD9\_CODE.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "D_ITEMS" linktitle =  
"D_ITEMS" weight = 18 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The d\_items table

**Table source:** CareVue and Metavision ICU databases.

**Table purpose:** Definition table for all items in the ICU databases.

**Number of rows:** 12,487

### Links to:

- CHARTEVENTS on ITEMID
- DATETIMEEVENTS on ITEMID
- INPUTEVENTS\_CV on ITEMID
- INPUTEVENTS\_MV on ITEMID
- MICROBIOLOGYEVENTS on ITEMID
- OUTPUTEVENTS on ITEMID
- PROCEDUREEVENTS\_MV on ITEMID

## Important considerations

- D\_ITEMS is sourced from two *distinct* ICU databases. The main consequence is that there are duplicate ITEMID for each concept. For example, heart rate is captured both as an ITEMID of 212 (CareVue) and as an ITEMID of 220045 (Metavision). As a result, it is necessary to search for multiple ITEMID to capture a single concept across the entire database. This can be tedious, and it is an active project to coalesce these ITEMID - one which welcomes any and all help provided by the community!
- Another source of duplicate ITEMID is due to the free text nature of data entry in CareVue - as a result there are additional ITEMID which correspond to misspellings or synonymous descriptions of a single concept. It is important to search for all possible abbreviations and descriptions of a concept to capture all associated ITEMID.
- If the LINKSTO column is null, then the data is currently unavailable, but planned for a future release.

## Table columns

Name	Postgres data type
ROW_ID	INT
ITEMID	INT
LABEL	VARCHAR(200)
ABBREVIATION	VARCHAR(100)
DBSOURCE	VARCHAR(20)
LINKSTO	VARCHAR(50)
CATEGORY	VARCHAR(100)
UNITNAME	VARCHAR(100)
PARAM_TYPE	VARCHAR(30)
CONCEPTID	INT

## Detailed Description

The D\_ITEMS table defines ITEMID, which represents measurements in the database. Measurements of the same type (e.g. heart rate) will have the same ITEMID (e.g. 211). The ITEMID column is an alternate primary key to this table: it is unique to each row.

Note that the D\_ITEMS table is sourced from two ICU databases: Metavision and CareVue. Each system had its own set of ITEMID to identify concepts. As a result, there are multiple ITEMID which correspond to the same concept. For CareVue data, ITEMID = 211 is used to identify heart rates, whereas for

Metavision data, ITEMID = 220045 is used. All Metavision ITEMIDs will have a value > 220000.

Note that the D\_ITEMS table does *not* link to the LABEVENTS table, as this data was acquired separately from the hospital database. The D\_ITEMS table was acquired from the ICU databases.

## **ITEMID**

As an alternate primary key to the table, ITEMID is unique to each row.

## **LABEL, ABBREVIATION**

The LABEL column describes the concept which is represented by the ITEMID. The ABBREVIATION column, only available in Metavision, lists a common abbreviation for the label.

## **DBSOURCE**

The DBSOURCE column was generated to clarify which database the given ITEMID was sourced from: 'carevue' indicates the ITEMID was sourced from CareVue, while 'metavision' indicated the ITEMID was sourced from Metavision.

## **LINKSTO**

LINKSTO provides the table name which the data links to. For example, a value of 'chartevents' indicates that the ITEMID of the given row is contained in CHARTEVENTS. A single ITEMID is only used in one event table, that is, if an ITEMID is contained in CHARTEVENTS it will *not* be contained in any other event table (e.g. IOEVENTS, CHARTEVENTS, etc).

## **CATEGORY**

CATEGORY provides some information of the type of data the ITEMID corresponds to. Examples include 'ABG', which indicates the measurement is sourced from an arterial blood gas, 'IV Medication', which indicates that the medication is administered through an intravenous line, and so on.

## UNITNAME

UNITNAME specifies the unit of measurement used for the ITEMID. This column is not always available, and this may be because the unit of measurement varies, a unit of measurement does not make sense for the given data type, or the unit of measurement is simply missing. Note that there is sometimes additional information on the unit of measurement in the associated event table, e.g. the VALUEUOM column in CHARTEVENTS.

## PARAM\_TYPE

PARAM\_TYPE describes the type of data which is recorded: a date, a number or a text field.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "D_LABITEMS" linktitle =  
"D_LABITEMS" weight = 19 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The d\_labitems table

**Table source:** Hospital database.

**Table purpose:** Definition table for all laboratory measurements.

**Number of rows:** 753

**Links to:**

- LABEVENTS on ITEMID

## Important considerations

- The ITEMID from MIMIC-III v1.0 does *not* match the ITEMID from MIMIC-II v2.6. If a mapping between the two is necessary, please contact the guardians of the database.
- Many of the LOINC codes were assigned during a project to standardize the ontology of lab measurements in the MIMIC database. Consequently, the codes were assigned post-hoc, and may not be present for every lab measurement. We welcome improvements to the present codes or assignment of LOINC codes to unmapped data elements from the community.

## Table columns

Name	Postgres data type
ROW_ID	INT
ITEMID	INT
LABEL	VARCHAR(100)
FLUID	VARCHAR(100)
CATEGORY	VARCHAR(100)
LOINC_CODE	VARCHAR(100)

## Detailed Description

D\_LABITEMS contains definitions for all ITEMID associated with lab measurements in the MIMIC database. All data in LABEVENTS link to the D\_LABITEMS table. Each unique LABEL in the hospital database was assigned an ITEMID in this table, and the use of this ITEMID facilitates efficient storage and querying of the data. Note that lab items are kept separate while most definitions are contained in the D\_ITEMS table, and there were good reasons to keep the lab items separate.

As the laboratory data is acquired from the hospital database, the data is consistent across all years in the database. Consequently, there is usually only one ITEMID associated with each concept in the database. Furthermore, the data contains information collected in departments outside the ICU. This includes both wards within the hospital and clinics outside the hospital. Most concepts in this table have been mapped to LOINC codes, an openly available ontology which provides a rich amount of information about the laboratory measurement including reference ranges, common units of measurement and other further detail regarding the measurement.

### ITEMID

As a candidate key in the table, ITEMID is unique to each row.

### LABEL

The LABEL column describes the concept which is represented by the ITEMID.

## FLUID

FLUID describes the substance on which the measurement was made. For example, chemistry measurements are frequently performed on blood, which is listed in this column as 'BLOOD'. Many of these measurements are also acquirable on other fluids, such as urine, and this column differentiates these distinct concepts.

## CATEGORY

CATEGORY provides higher level information as to the type of measurement. For example, a category of 'ABG' indicates that the measurement is an arterial blood gas.

## LOINC\_CODE

LOINC\_CODE contains the LOINC code associated with the given ITEMID. LOINC is an ontology which originally specified laboratory measurements but has since expanded to cover a wide range of clinically relevant concepts. LOINC openly provide a table which contains a large amount of detail about each LOINC code. This table is freely available online or can be provided by the guardians of the database.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "DATETIMEEVENTS" linktitle = "DATETIMEEVENTS" weight = 20 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The datetimeevents table

**Table source:** CareVue and Metavision ICU databases.

**Table purpose:** Contains all date formatted data.

**Number of rows:** 4,485,937

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- ICUSTAYS on ICUSTAY\_ID
- D\_ITEMS on ITEMID
- CAREGIVERS on CGID

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
ICUSTAY_ID	INT
ITEMID	INT
CHARTTIME	TIMESTAMP(0)
STORETIME	TIMESTAMP(0)
CGID	INT
VALUE	TIMESTAMP(0)
VALUEUOM	VARCHAR(50)
WARNING	SMALLINT
ERROR	SMALLINT
RESULTSTATUS	VARCHAR(50)
STOPPED	VARCHAR(50)

## Detailed Description

DATETIMEEVENTS contains all date measurements about a patient in the ICU. For example, the date of last dialysis would be in the DATETIMEEVENTS table, but the systolic blood pressure would not be in this table. As all dates in MIMIC-III are anonymized to protect patient confidentiality, all dates in this table have been shifted. Note that the chronology for an individual patient has been unaffected however, and quantities such as the difference between two dates remain true to reality.

### **SUBJECT\_ID, HADM\_ID, ICUSTAY\_ID**

Identifiers which specify the patient: **SUBJECT\_ID** is unique to a patient, **HADM\_ID** is unique to a patient hospital stay and **ICUSTAY\_ID** is unique to a patient ICU stay.

### **ITEMID**

### **CHARTTIME, STORETIME**

**CHARTTIME** records the time at which an observation was charted, and is usually the closest proxy to the time the data was actually measured. **STORETIME** records

the time at which an observation was manually input or manually validated by a member of the clinical staff.

#### **CGID**

CGID is the identifier for the caregiver who validated the given measurement.

#### **VALUE**

#### **VALUEUOM**

#### **WARNING, ERROR**

#### **RESULTSTATUS, STOPPED**

```
+++ date = "2015-09-01T19:34:46-04:00" title = "DIAGNOSES_ICD" linktitle  
= "DIAGNOSES_ICD" weight = 21 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## **The diagnoses\_icd table**

**Table source:** Hospital database.

**Table purpose:** Contains ICD diagnoses for patients, most notably ICD-9 diagnoses.

**Number of rows:** 651,047

#### **Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- D\_ICD\_DIAGNOSES on ICD9\_CODE

## **Important considerations**

- The ICD codes are generated for billing purposes at the end of the hospital stay.
- All ICD codes in MIMIC-III are ICD-9 based. The Beth Israel Deaconess Medical Center will begin using ICD-10 codes in 2015.

- The code field for the ICD-9-CM Principal and Other Diagnosis Codes is six characters in length, with the decimal point implied between the third and fourth digit for all diagnosis codes other than the V codes. The decimal is implied for V codes between the second and third digit.

## Table columns

Name	PostgreSQL data type	Modifiers
ROW_ID	INT	not null
SUBJECT_ID	INT	not null
HADM_ID	INT	not null
SEQ_NUM	INT	
ICD9_CODE	VARCHAR(10)	

## Detailed Description

### SUBJECT\_ID, HADM\_ID

Identifiers which specify the patient: SUBJECT\_ID is unique to a patient and HADM\_ID is unique to a patient hospital stay.

### SEQ\_NUM

SEQ\_NUM provides the order in which the ICD diagnoses relate to the patient. ICD diagnoses are ordered by priority - and the order does have an impact on the reimbursement for treatment.

### ICD9\_CODE

ICD9\_CODE contains the actual code corresponding to the diagnosis assigned to the patient for the given row. Note that all codes, as of MIMIC-III v1.0, are ICD-9 codes.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "DRGCODES" linktitle =
"DRGCODES" weight = 22 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The drgcodes table

**Table source:** Hospital database.

**Table purpose:** Contains diagnosis related groups (DRG) codes for patients.

**Number of rows:** 125,557

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID

## Important considerations

- HCFA-DRG codes have multiple descriptions as they have changed over time. Sometimes these descriptions are similar, but sometimes they are *completely different diagnoses*. Users will need to select rows using both the code *and* the description.
- Since there are multiple versions of DRG codes, queries will need to incorporate both the type of DRG and the code when filtering for a certain diagnosis.
- All patients have an HCFA-DRG code, but not all patients have an APR-DRG code. Note that APR-DRG is believed to be an alternative, more specific, code which could be used in conjunction with the HCFA codes.

## Table columns

Name	PostgreSQL data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
DRG_TYPE	VARCHAR(20)
DRG_CODE	VARCHAR(20)
DESCRIPTION	VARCHAR(300)
DRG_SEVERITY	SMALLINT
DRG_MORTALITY	SMALLINT

## Detailed Description

### **SUBJECT\_ID, HADM\_ID**

Identifiers which specify the patient: **SUBJECT\_ID** is unique to a patient and **HADM\_ID** is unique to a patient hospital stay.

### **DRG\_TYPE**

**DRG\_TYPE** provides the type of DRG code in the entry. There are two types of DRG codes in the database which have overlapping ranges but distinct definitions for the codes. The two types of DRG codes in the MIMIC-III database are ‘HCFA’ (Health Care Financing Administration) and ‘APR’ (All Payers Registry).

### **DRG\_CODE**

**DRG\_CODE** contains a code which represents the diagnosis billed for by the hospital.

### **DESCRIPTION**

**DESCRIPTION** provides a human understandable summary of the meaning of the given DRG code. The description field frequently has acronyms which represent comorbidity levels (comorbid conditions or “CC”). The following table provides a definition for some of these acronyms:

Acronym	Description
w CC/MCC	with CC or Major CC
w MCC	with Major CC
w CC	with CC and without Major CC
w NonCC	with NonCC and without CC or Major CC
w/o MCC	with CC or Non CC and without Major CC
w/o CC/MCC	with nonCC and without CC or Major CC

Note that there are three levels of comorbidities: none, with comorbid conditions, and with major comorbid conditions. These acronyms are primarily used in HCFA/MS DRG codes.

## DRG\_SEVERITY, DRG\_MORTALITY

DRG\_SEVERITY and DRG\_MORTALITY provide additional granularity to DRG codes in the ‘APR’ DRG type. Severity and mortality allow for higher billing costs when a diagnosis is more severe, and vice versa.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "ICUSTAYS" linktitle =  
"ICUSTAYS" weight = 23 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC" +++
```

## The icustays table

**Table source:** Hospital database.

**Table purpose:** Defines each ICUSTAY\_ID in the database, i.e. defines a single ICU stay.

**Number of rows:** 61,532

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID

## Important considerations

- ICUSTAY\_ID is a *generated* identifier that is *not* based on any raw data identifier. The hospital and ICU databases are not intrinsically linked and so do not have any concept of an ICU encounter identifier.
- The ICUSTAYS table is derived from the TRANSFERS table. Specifically, it groups the TRANSFERS table based on ICUSTAY\_ID, and excludes rows where no ICUSTAY\_ID is present.

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
ICUSTAY_ID	INT
DBSOURCE	VARCHAR(20)
FIRST_CAREUNIT	VARCHAR(20)

Name	Postgres data type
LAST_CAREUNIT	VARCHAR(20)
FIRST_WARDID	SMALLINT
LAST_WARDID	SMALLINT
INTIME	TIMESTAMP(0)
OUTTIME	TIMESTAMP(0)
LOS	DOUBLE

## Detailed Description

### **SUBJECT\_ID, HADM\_ID, ICUSTAY\_ID**

Identifiers which specify the patient: **SUBJECT\_ID** is unique to a patient, **HADM\_ID** is unique to a patient hospital stay and **ICUSTAY\_ID** is unique to a patient ICU stay.

### **DBSOURCE**

**DBSOURCE** contains the original ICU database the data was sourced from. Patients admitted between 2001 - 2008 had their data managed by the CareVue information system, represented in this column as 'carevue'. Patients admitted between 2008 - 2012 had their data managed by the Metavision system, represented in this column as 'metavision'. Knowing the database source is important as the data archiving for these two databases differs in some cases.

### **FIRST\_CAREUNIT, LAST\_CAREUNIT**

**FIRST\_CAREUNIT** and **LAST\_CAREUNIT** contain, respectively, the first and last ICU type in which the patient was cared for. As an **ICUSTAY\_ID** groups all ICU admissions within 24 hours of each other, it is possible for a patient to be transferred from one type of ICU to another and have the same **ICUSTAY\_ID**.

Care units are derived from the **TRANSFERS** table, and definition for the abbreviations can be found in the documentation for **TRANSFERS**.

### **FIRST\_WARDID, LAST\_WARDID**

**FIRST\_WARDID** and **LAST\_WARDID** contain the first and last ICU unit in which the patient stayed. Note the grouping of physical locations in the hospital database is referred to as ward. Though in practice ICUs are not referred to as wards, the

hospital database technically tracks ICUs as “wards with an ICU cost center”. As a result, each ICU is associated with a WARDID.

## INTIME, OUTTIME

INTIME provides the date and time the patient was transferred into the ICU. OUTTIME provides the date and time the patient was transferred out of the ICU.

## LOS

LOS is the length of stay for the patient for the given ICU stay, which may include one or more ICU units.

```
+++ date = “2015-09-01T19:34:46-04:00” title = “INPUTEVENTS_CV” linkti-  
tle = “INPUTEVENTS_CV” weight = 24 toc = “true”
```

```
[menu][menu.main] parent = “Tables in MIMIC” +++
```

## The inputevents\_cv table

**Table source:** CareVue ICU databases.

**Table purpose:** Input data for patients.

**Number of rows:** 17,527,935

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- ICUSTAYS on ICUSTAY\_ID
- D\_ITEMS on ITEMID
- CAREGIVERS on CGID

A high level description of the data is available [here](#).

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
ICUSTAY_ID	INT

Name	Postgres data type
CHARTTIME	TIMESTAMP(0)
ITEMID	INT
AMOUNT	DOUBLE PRECISION
AMOUNTUOM	VARCHAR(30)
RATE	DOUBLE PRECISION
RATEUOM	VARCHAR(30)
STORETIME	TIMESTAMP(0)
CGID	BIGINT
ORDERID	BIGINT
LINKORDERID	BIGINT
STOPPED	VARCHAR(30)
NEWBOTTLE	INT
ORIGINALAMOUNT	DOUBLE PRECISION
ORIGINALAMOUNTUOM	VARCHAR(30)
ORIGINALROUTE	VARCHAR(30)
ORIGINALRATE	DOUBLE PRECISION
ORIGINALRATEUOM	VARCHAR(30)
ORIGINALSITE	VARCHAR(30)

## Detailed Description

### SUBJECT\_ID, HADM\_ID, ICUSTAY\_ID

Identifiers which specify the patient: **SUBJECT\_ID** is unique to a patient, **HADM\_ID** is unique to a patient hospital stay and **ICUSTAY\_ID** is unique to a patient ICU stay.

### CHARTTIME

**CHARTTIME** represents the time at which the measurement was charted - that is - recorded on the clinical information system at the bedside. For amounts received (usually volumes), the **CHARTTIME** represents the time at which that volume was received. That is, it can be considered an “end time”, i.e. X millilitres of solution was administered to the patient by this **CHARTTIME**. For rates, the **CHARTTIME** represents the time at which that rate was set. That is, it can be considered a “start time”, i.e. the patient is now receiving X mcg/kg/min of a drug at this **CHARTTIME**.

See the understanding IOEVENTS page for a bit more detail about how this table was formed from the raw data.

## **ITEMID**

Identifier for a single measurement type in the database. Each row associated with one ITEMID (e.g. 212) corresponds to an instantiation of the same measurement (e.g. heart rate). Metavision ITEMID values are all above 220000. A subset of commonly used medications in CareVue data have ITEMID values are between 30000-39999. The remaining input/output ITEMID values are between 40000-49999.

## **AMOUNT, AMOUNTUOM**

AMOUNT and AMOUNTUOM list the amount of a drug or substance administered to the patient either between the STARTTIME and ENDTIME (if both are available) or at the ENDTIME (when the exact start time is unknown, but usually up to an hour before).

## **RATE, RATEUOM**

RATE and RATEUOM list the rate at which the drug or substance was administered to the patient either between the STARTTIME and ENDTIME (if both are available), or it lists the rate at which the drug is *currently* administered at the ENDTIME.

## **STORETIME**

STORETIME records the time at which an observation was manually input or manually validated by a member of the clinical staff.

## **CGID**

CGID is the identifier for the caregiver who validated the given measurement.

## **ORDERID, LINKORDERID**

ORDERID links multiple items contained in the same solution together. For example, when a solution of noradrenaline and normal saline is administered both noradrenaline and normal saline occur on distinct rows but will have the same ORDERID.

LINKORDERID links the same order across multiple instantiations: for example, if the rate of delivery for the solution with noradrenaline and normal saline is changed, two new rows which share the same new ORDERID will be generated, but the LINKORDERID will be the same.

## STOPPED, NEWBOTTLE

STOPPED indicates whether the infusion has been disconnected or continued. NEWBOTTLE indicates if a new preparation of the solution was hung at the bedside.

## ORIGINALAMOUNT, ORIGINALAMOUNTUOM, ORIGINALROUTE, ORIGINALRATE, ORIGINAL- RATEUOM , ORIGINALSITE

These columns provide information about the solution the medication was a part of when it was first entered into the information system.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "INPUTEVENTS_MV" link-  
title = "INPUTEVENTS_MV" weight = 25 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC" +++
```

## The inputeevents\_mv table

**Table source:** Metavision ICU databases.

**Table purpose:** Input data for patients.

**Number of rows:** 3,618,991

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- ICUSTAYS on ICUSTAY\_ID
- D\_ITEMS on ITEMID
- CAREGIVERS on CGID

A high level description of the data is available [here](#).

## Important considerations

- A bolus will be listed as ending one minute after it started, i.e. ENDTIME = STARTTIME + 1 minute

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
ICUSTAY_ID	INT
STARTTIME	TIMESTAMP(0)
ENDTIME	TIMESTAMP(0)
ITEMID	INT
AMOUNT	DOUBLE PRECISION
AMOUNTUOM	VARCHAR(30)
RATE	DOUBLE PRECISION
RATEUOM	VARCHAR(30)
STORETIME	TIMESTAMP(0)
CGID	BIGINT
ORDERID	BIGINT
LINKORDERID	BIGINT
ORDERCATEGORYNAME	VARCHAR(100)
SECONDARYORDERCATEGORYNAME	VARCHAR(100)
ORDERCOMPONENTTYPEDESCRIPTION	VARCHAR(200)
ORDERCATEGORYDESCRIPTION	VARCHAR(50)
PATIENTWEIGHT	DOUBLE PRECISION
TOTALAMOUNT	DOUBLE PRECISION
TOTALAMOUNTUOM	VARCHAR(50)
ISOPENBAG	SMALLINT
CONTINUEINNEXTDEPT	SMALLINT
CANCELREASON	SMALLINT
STATUSDESCRIPTION	VARCHAR(30)
COMMENTS_STATUS	VARCHAR(30)
COMMENTS_TITLE	VARCHAR(100)
COMMENTS_DATE	TIMESTAMP(0)
ORIGINALAMOUNT	DOUBLE PRECISION
ORIGINALRATE	DOUBLE PRECISION

## Detailed Description

### **SUBJECT\_ID, HADM\_ID, ICUSTAY\_ID**

Identifiers which specify the patient: **SUBJECT\_ID** is unique to a patient, **HADM\_ID** is unique to a patient hospital stay and **ICUSTAY\_ID** is unique to a patient ICU stay.

## **STARTTIME, ENDTIME**

STARTTIME and ENDTIME record the start and end time of an input/output event.

## **ITEMID**

Identifier for a single measurement type in the database. Each row associated with one ITEMID which corresponds to an instantiation of the same measurement (e.g. norepinephrine). MetaVision ITEMID values are all above 220000. Since this data only contains data from MetaVision, it only contains ITEMID above 220000 (see here for details about MetaVision)

## **AMOUNT, AMOUNTUOM**

AMOUNT and AMOUNTUOM list the amount of a drug or substance administered to the patient either between the STARTTIME and ENDTIME.

## **RATE, RATEUOM**

RATE and RATEUOM list the rate at which the drug or substance was administered to the patient either between the STARTTIME and ENDTIME.

## **STORETIME**

STORETIME records the time at which an observation was manually input or manually validated by a member of the clinical staff.

## **CGID**

CGID is the identifier for the caregiver who validated the given measurement.

## **ORDERID, LINKORDERID**

ORDERID links multiple items contained in the same solution together. For example, when a solution of noradrenaline and normal saline is administered both noradrenaline and normal saline occur on distinct rows but will have the same ORDERID.

LINKORDERID links the same order across multiple instantiations: for example, if the rate of delivery for the solution with noradrenaline and normal saline is

changed, two new rows which share the same new ORDERID will be generated, but the LINKORDERID will be the same.

### **ORDERCATEGORYNAME, SECONDARYORDERCATEGORYNAME, ORDERCOMPONENTTYPEDESCRIPTION, ORDERCATEGORYDESCRIPTION**

These columns provide higher level information about the order the medication/solution is a part of. Categories represent the type of administration, while the ORDERCOMPONENTTYPEDESCRIPTION describes the role of the substance in the solution (i.e. main order parameter, additive, or mixed solution)

### **PATIENTWEIGHT**

The patient weight in kilograms.

### **TOTALAMOUNT, TOTALAMOUNTUOM**

Intravenous administrations are usually given by hanging a bag of fluid at the bedside for continuous infusion over a certain period of time. These columns list the total amount of the fluid in the bag containing the solution.

### **STATUSDESCRIPTION**

STATUSDESCRIPTION states the ultimate status of the item. 'Stopped' indicates that the caregiver stopped the item or the programmed volume came to an end. 'Finished running' indicates that the programmed volume has come to an end. 'Rewritten' indicates that the caregiver rewrote the item, for example making an amendment to the starttime. 'Changed' indicates that the caregiver changed an item, for example setting a new rate or dose. 'Flushed' indicates that a line was flushed.

### **ISOPENBAG**

Whether the order was from an open bag.

### **CONTINUEINNEXTDEPT**

If the order ended on patient transfer, this field indicates if it continued into the next department (e.g. a floor).

## CANCELREASON

If the order was canceled, this column provides some explanation.

## COMMENTS\_STATUS, COMMENTS\_TITLE, COMMENTS\_DATE

Specifies if the order was edited or canceled, and if so, the date and job title of the care giver who canceled or edited it.

## ORIGINALAMOUNT

Drugs are usually mixed within a solution and delivered continuously from the same bag. This column represents the amount of the drug contained in the bag at **STARTTIME**. For the first infusion of a new bag, **ORIGINALAMOUNT** = **TOTALAMOUNT**. Later on, if the rate is changed, then the amount of the drug in the bag will be lower (as some has been administered to the patient). As a result, **ORIGINALAMOUNT** < **TOTALAMOUNT**, and **ORIGINALAMOUNT** will be the amount of drug leftover in the bag at that **STARTTIME**.

## ORIGINALRATE

This is the rate that was input by the care provider. Note that this may differ from **RATE** because of various reasons: **ORIGINALRATE** was the original planned rate, while the **RATE** column will be the true rate delivered. For example, if a bag is about to run out and the care giver decides to push the rest of the fluid, then **RATE** > **ORIGINALRATE**. However, these two columns are usually the same, but have minor non-clinically significant differences due to rounding error.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "LABEVENTS" linktitle =  
"LABEVENTS" weight = 26 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The labevents table

**Table source:** Hospital database.

**Table purpose:** Contains all laboratory measurements for a given patient, including out patient data.

**Number of rows:** 27,854,055

### Links to:

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- D\_LABITEMS on ITEMID

## Important considerations

- Note that the time associated with this result is the time of the fluid *acquisition*, not the time that the values were made available to the clinical staff.
- The labevents table contains both in-hospital laboratory measurements *and* out of hospital laboratory measurements from clinics which the patient has visited (since the patient is not “in” a hospital when visiting a clinic, these patients often referred to as “outpatients” and the data is often called “outpatient” data). Laboratory measurements for outpatients **do not have a HADM\_ID**.
- In MIMIC-III v1.0, there is a subset of patients for which the outpatient lab data is not available. They can be identified by checking for patients whose data *always* has an HADM\_ID.
- In MIMIC-III v1.0, there is a subset of patients for which text laboratory data is missing. This primarily affects the blood gas type recorded with blood gases.
- Some items are duplicated between the labevents and chartevents tables. In cases where there is disagreement between measurements, labevents should be taken as the ground truth.

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
ITEMID	INT
CHARTTIME	TIMESTAMP(0)
VALUE	VARCHAR(200)
VALUENUM	DOUBLE PRECISION
VALUEUOM	VARCHAR(20)
FLAG	VARCHAR(20)

## Detailed Description

The LABEVENTS data contains information regarding laboratory based measurements. The process for acquiring a lab measurement is as follows: first, a member of the clinical staff acquires a fluid from a site in the patient's body (e.g. blood from an arterial line, urine from a catheter, etc). Next, the fluid is bar coded to associate it with the patient *and* timestamped to record the time of the fluid acquisition. The lab analyses the data and returns a result within 4-12 hours.

### SUBJECT\_ID, HADM\_ID

Identifiers which specify the patient: SUBJECT\_ID is unique to a patient and HADM\_ID is unique to a patient hospital stay.

### ITEMID

Identifier for a single measurement type in the database. Each row associated with one ITEMID (e.g. 212) corresponds to an instantiation of the same measurement (e.g. heart rate).

### CHARTTIME

CHARTTIME records the time at which an observation was charted, and is usually the closest proxy to the time the data was actually measured. Note that because the data is directly sourced from the laboratory database, it is *not* validated by ICU clinical staff, and as a result there is no associated STORETIME.

### VALUE, VALUENUM

VALUE contains the value measured for the concept identified by the ITEMID. If this value is numeric, then VALUENUM contains the same data in a numeric format. If this data is not numeric, VALUENUM is null. In some cases (e.g. scores like Glasgow Coma Scale, Richmond Sedation Agitation Scale and Code Status), VALUENUM contains the score and VALUE contains the score and text describing the meaning of the score.

### VALUEUOM

VALUEUOM is the unit of measurement for the VALUE, if appropriate.

## FLAG

FLAG indicates whether the laboratory value is considered abnormal or not, using pre-defined thresholds.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "MICROBIOLOGYEVENTS"  
linktitle = "MICROBIOLOGYEVENTS" weight = 27 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The microbiologyevents table

**Table source:** Hospital database.

**Table purpose:** Contains microbiology information, including tests performed and sensitivities.

**Number of rows:** 631,726

### Links to:

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- D\_ITEMS on SPEC\_ITEMID
- D\_ITEMS on ORG\_ITEMID
- D\_ITEMS on AB\_ITEMID

## Important considerations

- The MICROBIOLOGYEVENTS table does not contain cultures from samples taken outside the ICU
- If the specimen is null, then the culture had no growth reported.

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
CHARTDATE	TIMESTAMP(0)
CHARTTIME	TIMESTAMP(0)

Name	Postgres data type
SPEC_ITEMID	INT
SPEC_TYPE_CD	VARCHAR(10)
SPEC_TYPE_DESC	VARCHAR(100)
ORG_ITEMID	INT
ORG_CD	INT
ORG_NAME	VARCHAR(100)
ISOLATE_NUM	SMALLINT
AB_ITEMID	INT
AB_CD	INT
AB_NAME	VARCHAR(30)
DILUTION_TEXT	VARCHAR(10)
DILUTION_COMPARISON	VARCHAR(20)
DILUTION_VALUE	DOUBLE PRECISION
INTERPRETATION	VARCHAR(5)

## Detailed Description

### **SUBJECT\_ID, HADM\_ID**

Identifiers which specify the patient: **SUBJECT\_ID** is unique to a patient and **HADM\_ID** is unique to a patient hospital stay.

### **CHARTDATE, CHARTTIME**

**CHARTTIME** records the time at which an observation was charted, and is usually the closest proxy to the time the data was actually measured. **CHARTDATE** is the same as **CHARTTIME**, except there is no time available.

**CHARTDATE** was included as time information is not always available for microbiology measurements: in order to be clear about when this occurs, **CHARTTIME** is null, and **CHARTDATE** contains the date of the measurement.

In the cases where both **CHARTTIME** and **CHARTDATE** exists, **CHARTDATE** is equal to a truncated version of **CHARTTIME** (i.e. **CHARTTIME** without the timing information). Not all observations have a **CHARTTIME**, but all observations have a **CHARTDATE**.

### **SPEC\_ITEMID, SPEC\_TYPE\_CD, SPEC\_TYPE\_DESC**

Details the itemid, code, and description for the specimen.

ORG\_ITEMID, ORG\_CD, ORG\_NAME

ISOLATE\_NUM

AB\_ITEMID, AB\_CD, AB\_NAME

DILUTION\_TEXT, DILUTION\_COMPARISON, DILUTION\_VALUE

INTERPRETATION

INTERPRETATION indicates the results of the test. “S” is sensitive, “R” is resistant, “I” is intermediate, and “P” is pending.

+++ date = “2015-09-01T19:34:46-04:00” title = “NOTEEVENTS” linktitle = “NOTEEVENTS” weight = 28 toc = “true”

[menu][menu.main] parent = “Tables in MIMIC”

+++

## The noteevents table

**Table source:** Hospital database.

**Table purpose:** Contains all notes for patients.

**Number of rows:** 2,083,180

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- CAREGIVERS on CGID

## Important considerations

- TEXT is often large and contains many newline characters: it may be easier to read if viewed in a distinct program rather than the one performing the queries.
- Echo reports, ECG reports, and radiology reports are available for both inpatient and outpatient stays. If a patient is an outpatient, there will not be an HADM\_ID associated with the note. If the patient is an inpatient, but was not admitted to the ICU for that particular hospital admission, then there will *not* be an HADM\_ID associated with the note.

- Echos are generated using templates and in some cases there may be discrepancies in severity. For example one report may contain: “Mild PA systolic hypertension. Severe PA systolic hypertension.” indicating that the caregiver may not have removed the appropriate item from the template.

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
CHARTDATE	TIMESTAMP(0)
CATEGORY	VARCHAR(50)
DESCRIPTION	VARCHAR(300)
CGID	INT
ISERROR	CHAR(1)
TEXT	TEXT

## Detailed Description

### SUBJECT\_ID, HADM\_ID

Identifiers which specify the patient: **SUBJECT\_ID** is unique to a patient and **HADM\_ID** is unique to a patient hospital stay.

### CHARTDATE

**CHARTDATE** records the date at which the note was charted.

### CATEGORY, DESCRIPTION

**CATEGORY** and **DESCRIPTION** define the type of note recorded. For example, a **CATEGORY** of ‘Discharge’ indicates that the note is a discharge note, and a **DESCRIPTION** of ‘Summary’ in conjunction with this indicates that the note is a discharge summary.

## CGID

CGID is the identifier for the caregiver who input the note.

## ISERROR

A '1' in the ISERROR column indicates that a physician has identified this note as an error.

## TEXT

TEXT contains the note text.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "OUTPUTEVENTS" linktitle  
= "OUTPUTEVENTS" weight = 29 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC" +++
```

## The outpotevents table

**Table source:** CareVue and Metavision ICU databases.

**Table purpose:** Output data for patients.

**Number of rows:** 4,349,218

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- ICUSTAYS on ICUSTAY\_ID
- D\_ITEMS on ITEMID
- CAREGIVERS on CGID

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
ICUSTAY_ID	INT
CHARTTIME	TIMESTAMP(0)
ITEMID	INT

Name	Postgres data type
VALUE	DOUBLE PRECISION
VALUEUOM	VARCHAR(30)
STORETIME	TIMESTAMP(0)
CGID	BIGINT
STOPPED	VARCHAR(30)
NEWBOTTLE	INT
ISERROR	SMALLINT

## Detailed Description

### SUBJECT\_ID, HADM\_ID, ICUSTAY\_ID

Identifiers which specify the patient: **SUBJECT\_ID** is unique to a patient, **HADM\_ID** is unique to a patient hospital stay and **ICUSTAY\_ID** is unique to a patient ICU stay.

### CHARTTIME

**CHARTTIME** is the time of an output event.

### ITEMID

Identifier for a single measurement type in the database. Each row associated with one **ITEMID** (e.g. 212) corresponds to an instantiation of the same measurement (e.g. heart rate).

Metavision **ITEMID** values are all above 220000. A subset of commonly used medications in CareVue data have **ITEMID** values are between 30000-39999. The remaining input/output **ITEMID** values are between 40000-49999.

### VALUE, VALUEUOM

**VALUE** and **VALUEUOM** list the amount of a substance at the **CHARTTIME** (when the exact start time is unknown, but usually up to an hour before).

### STORETIME

**STORETIME** records the time at which an observation was manually input or manually validated by a member of the clinical staff.

## CGID

CGID is the identifier for the caregiver who validated the given measurement.

## STOPPED, NEWBOTTLE, ISERROR

STOPPED indicates if the order was disconnected at the given CHARTTIME. NEWBOTTLE indicates that a new bag of solution was hung at the given CHARTTIME. ISERROR is a Metavision checkbox where a care giver can specify that an observation is an error. No other details are provided.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "PATIENTS" linktitle =  
"PATIENTS" weight = 30 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The patients table

**Table source:** CareVue and Metavision ICU databases.

**Table purpose:** Contains all charted data for all patients.

**Number of rows:** 46,520

**Links to:**

- ADMISSIONS on SUBJECT\_ID
- ICUSTAYS on SUBJECT\_ID

## Important considerations

- DOB has been shifted for patients older than 89. The median age for the patients whose date of birth was shifted is 91.4.

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
GENDER	VARCHAR(5)
DOB	TIMESTAMP(0)

Name	Postgres data type
DOD	TIMESTAMP(0)
DOD_HOSP	TIMESTAMP(0)
DOD_SSN	TIMESTAMP(0)
EXPIRE_FLAG	VARCHAR(5)

## Detailed Description

### **SUBJECT\_ID**

SUBJECT\_ID is a unique identifier which specifies an individual patient. SUBJECT\_ID is a candidate key for the table, so is unique for each row. Information that is consistent for the lifetime of a patient is stored in this table.

### **GENDER**

GENDER is the genotypical sex of the patient.

### **DOB**

DOB is the date of birth of the given patient. Patients who are older than 89 years old at any time in the database have had their date of birth shifted to obscure their age and comply with HIPAA. The shift process was as follows: the patient's age at their first admission was determined. The date of birth was then set to exactly 300 years before their first admission.

### **DOD, DOD\_HOSP, DOD\_SSN**

DOD is the date of death for the given patient. DOD\_HOSP is the date of death as recorded in the hospital database. DOD\_SSN is the date of death from the social security database. Note that DOD merged together DOD\_HOSP and DOD\_SSN, giving priority to DOD\_HOSP if both were recorded.

### **EXPIRE\_FLAG**

EXPIRE\_FLAG is a binary flag which indicates whether the patient died, i.e. whether DOD is null or not. These deaths include both deaths within the hospital (DOD\_HOSP) and deaths identified by matching the patient to the social security master death index (DOD\_SSN).

```

+++ date = "2015-09-01T19:34:46-04:00" title = "PRESCRIPTIONS" linktitle
= "PRESCRIPTIONS" weight = 31 toc = "true"

[menu][menu.main] parent = "Tables in MIMIC"

+++

```

## The prescriptions table

**Table source:** Hospital provider order entry database.

**Table purpose:** Contains medication related order entries, i.e. prescriptions.

**Number of rows:** 4,156,450

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- ICUSTAYS on ICUSTAY\_ID

## Important considerations

- The table does *not* specify if an order was later cancelled as of MIMIC-III v1.0.

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
ICUSTAY_ID	INT
STARTTIME	TIMESTAMP(0)
ENDTIME	TIMESTAMP(0)
DRUG_TYPE	VARCHAR(100)
DRUG	VARCHAR(100)
DRUG_NAME_POE	VARCHAR(100)
DRUG_NAME_GENERIC	VARCHAR(100)
FORMULARY_DRUG_CD	VARCHAR(120)
GSN	VARCHAR(200)
NDC	VARCHAR(120)
PROD_STRENGTH	VARCHAR(120)

Name	Postgres data type
DOSE_VAL_RX	VARCHAR(120)
DOSE_UNIT_RX	VARCHAR(120)
FORM_VAL_DISP	VARCHAR(120)
FORM_UNIT_DISP	VARCHAR(120)
ROUTE	VARCHAR(120)

## Detailed Description

### **SUBJECT\_ID, HADM\_ID, ICUSTAY\_ID**

Identifiers which specify the patient: **SUBJECT\_ID** is unique to a patient, **HADM\_ID** is unique to a patient hospital stay and **ICUSTAY\_ID** is unique to a patient ICU stay.

### **STARTTIME, ENDTIME**

**STARTTIME** and **ENDTIME** specify the time period for which the prescription was valid.

### **DRUG\_TYPE**

**DRUG\_TYPE** provides the type of drug prescribed.

### **DRUG, DRUG\_NAME\_POE, DRUG\_NAME\_GENERIC**

These columns are various representations of the drug prescribed to the patient.

### **FORMULARY\_DRUG\_CD, GSN, NDC**

These columns provide a representation of the drug in various coding systems. **GSN** is the Generic Sequence Number. **NDC** is the National Drug Code

PROD\_STRENGTH

DOSE\_VAL\_RX, DOSE\_UNIT\_RX

FORM\_VAL\_DISP, FORM\_UNIT\_DISP

ROUTE

The route prescribed for the drug.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "PROCEDUREEVENTS_MV"  
linktitle = "PROCEDUREEVENTS_MV" weight = 32 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The procedureevents\_mv table

**Table source:** Metavision ICU database.

**Table purpose:** Contains procedures for patients

**Number of rows:** 258,066

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- ICUSTAYS on ICUSTAY\_ID
- D\_ITEMS on ITEMID

## Table columns

Name	Postgres data type
ROW_ID	INT NOT NULL
SUBJECT_ID	INT NOT NULL
HADM_ID	INT NOT NULL
ICUSTAY_ID	INT
STARTTIME	TIMESTAMP(0)
ENDTIME	TIMESTAMP(0)
ITEMID	INT
VALUE	DOUBLE PRECISION
VALUEUOM	VARCHAR(30)

Name	Postgres data type
LOCATION	VARCHAR(30)
LOCATIONCATEGORY	VARCHAR(30)
STORETIME	TIMESTAMP(0)
CGID	INT
ORDERID	INT
LINKORDERID	INT
ORDERCATEGORYNAME	VARCHAR(100)
SECONDARYORDERCATEGORYNAME	VARCHAR(100)
ORDERCATEGORYDESCRIPTION	VARCHAR(50)
ISOPENBAG	SMALLINT
CONTINUEINNEXTDEPT	SMALLINT
CANCELREASON	SMALLINT
STATUSDESCRIPTION	VARCHAR(30)
COMMENTS_EDITEDBY	VARCHAR(30)
COMMENTS_CANCELEDBY	VARCHAR(30)
COMMENTS_DATE	TIMESTAMP(0)

```
+++ date = "2015-09-01T19:34:46-04:00" title = "PROCEDURES_ICD" linktitle = "PROCEDURES_ICD" weight = 33 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The procedures\_icd table

**Table source:** Hospital database.

**Table purpose:** Contains ICD procedures for patients, most notably ICD-9 procedures.

**Number of rows:** 240,095

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- D\_ICD\_PROCEDURES on ICD9\_CODE

## Important considerations

- The ICD codes are generated for billing purposes at the end of the hospital stay.

- In MIMIC-III v1.0, only ICD-9 codes are used for recording procedures.

## Table columns

Name	PostgreSQL data type	Modifiers
ROW_ID	INT	not null
SUBJECT_ID	INT	not null
HADM_ID	INT	not null
SEQ_NUM	INT	
ICD9_CODE	VARCHAR(10)	

## Detailed Description

### SUBJECT\_ID, HADM\_ID

Identifiers which specify the patient: SUBJECT\_ID is unique to a patient and HADM\_ID is unique to a patient hospital stay.

### SEQ\_NUM

PROC\_SEQ\_NUM provides the order in which the procedures were performed.

### ICD9\_CODE

CODE provides the code for the given procedure.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "SERVICES" linktitle =
"SERVICES" weight = 34 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The services table

**Table source:** Hospital database.

**Table purpose:** Lists services that a patient was admitted/transferred under.

**Number of rows:** 73,343

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID

## Brief summary

The services table describes the service that a patient was admitted under. While a patient can be physically located at a given ICU type (say MICU), they are not necessarily being cared for by the team which staffs the MICU. This can happen due to a number of reasons, including bed shortage. The SERVICES table should be used if interested in identifying the type of service a patient is receiving in the hospital. For example, if interested in identifying surgical patients, the recommended method is searching for patients admitted under a surgical service.

Each service is listed in the table as an abbreviation - this is exactly how the data is stored in the hospital database. For user convenience, we have provided a description of each service type.

<u>ServiceDescription</u>
CMEDCardiac Med- ical - for non- surgical car- diac re- lated admissions
CSURCardiac Surgery - for sur- gi- cal car- diac admissions

---

ServiceDescription

DENT Dental  
- for  
den-  
tal/jaw  
re-  
lated  
admissions

ENT Ear,  
nose,  
and  
throat  
-  
con-  
di-  
tions  
pri-  
mar-  
ily  
af-  
fect-  
ing  
these  
areas

GU Genitourinary  
- re-  
pro-  
duc-  
tive  
or-  
gans/urinary  
system

GYN Gynecological  
- fe-  
male  
re-  
pro-  
duc-  
tive  
sys-  
tems  
and  
breasts

---

ServiceDescription

MED Medical

-

gen-  
eral  
ser-  
vice  
for  
in-  
ter-  
nal  
medicine

NB Newborn

- in-  
fants  
born  
at  
the  
hospital

NBB Newborn

baby  
- in-  
fants  
born  
at  
the  
hospital

NMEDNeurologic

Med-  
ical

-

non-  
surgical,  
re-  
lat-  
ing  
to  
the  
brain

---

ServiceDescription

NSURC Neurologic

Sur-  
gi-  
cal -  
sur-  
gi-  
cal,  
re-  
lat-  
ing  
to  
the  
brain

OBS Obstetrics

-  
con-  
cerned  
with  
child-  
birth  
and  
the  
care  
of  
women  
giv-  
ing  
birth

ORTH Orthopaedic

-  
sur-  
gi-  
cal,  
re-  
lat-  
ing  
to  
the  
mus-  
cu-  
loskele-  
tal  
system

---

ServiceDescription

OMEDOrthopaedic  
medicine

-

non-  
surgical,  
re-  
lat-  
ing  
to  
mus-  
cu-  
loskele-  
tal  
system

PSURPlastic

-

restor-  
ta-  
tion/reconstruction  
of  
the  
hu-  
man  
body  
(in-  
clud-  
ing  
cos-  
metic  
or  
aesthetic)

---

ServiceDescription

PSYCHPsychiatric

-

men-  
tal  
dis-  
or-  
ders

re-  
lat-  
ing

to  
mood,

be-  
haviour,

cog-  
ni-  
tion,

or

perceptions

SURGSurgical

-

gen-  
eral  
sur-  
gi-  
cal

ser-  
vice

not  
clas-

si-  
fied

elsewhere

---

ServiceDescription

TRAUM Trauma  
- in-  
jury  
or  
dam-  
age  
caused  
by  
phys-  
ical  
harm  
from  
an  
ex-  
ter-  
nal  
source

TSUR Thoracic  
Sur-  
gi-  
cal -  
surgery  
on  
the  
tho-  
rax,  
lo-  
cated  
be-  
tween  
the  
neck  
and  
the  
abdomen

Service	Description
VSURV	Vascular Sur- gi- cal - surgery re- lat- ing to the cir- cu- la- tory system

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
TRANSFERTIME	TIMESTAMP(0)
PREV_SERVICE	VARCHAR(20)
CURR_SERVICE	VARCHAR(20)

## Detailed Description

### SUBJECT\_ID, HADM\_ID

Identifiers which specify the patient: SUBJECT\_ID is unique to a patient and HADM\_ID is unique to a patient hospital stay.

### TRANSFERTIME

TRANSFERTIME is the time at which the patient moved from the PREV\_SERVICE (if present) to the CURR\_SERVICE.

## PREV\_SERVICE, CURR\_SERVICE

PREV\_SERVICE and CURR\_SERVICE are the previous and current service that the patient resides under.

```
+++ date = "2015-09-01T19:34:46-04:00" title = "TRANSFERS" linktitle =  
"TRANSFERS" weight = 35 toc = "true"
```

```
[menu][menu.main] parent = "Tables in MIMIC"
```

```
+++
```

## The transfers table

**Table source:** Hospital database.

**Table purpose:** Physical locations for patients throughout their hospital stay.

**Number of rows:** 261,897

**Links to:**

- PATIENTS on SUBJECT\_ID
- ADMISSIONS on HADM\_ID
- ICUSTAYS on ICUSTAY\_ID

## Important considerations

- The ICUSTAYS table is derived from this table.
- Care units are defined based off the WARDID being associated with an ICU cost center.
- ICUs in the Beth Israel have moved throughout the years, and consequently the same WARDID may be considered as an ICU for patient A but not an ICU for patient B.

## Table columns

Name	Postgres data type
ROW_ID	INT
SUBJECT_ID	INT
HADM_ID	INT
ICUSTAY_ID	INT
DBSOURCE	VARCHAR(20)

Name	Postgres data type
EVENTTYPE	VARCHAR(20)
PREV_CAREUNIT	VARCHAR(20)
CURR_CAREUNIT	VARCHAR(20)
PREV_WARDID	SMALLINT
CURR_WARDID	SMALLINT
INTIME	TIMESTAMP(0)
OUTTIME	TIMESTAMP(0)
LOS	INT

## Detailed Description

### **SUBJECT\_ID, HADM\_ID, ICUSTAY\_ID**

Identifiers which specify the patient: **SUBJECT\_ID** is unique to a patient, **HADM\_ID** is unique to a patient hospital stay and **ICUSTAY\_ID** is unique to a patient ICU stay.

### **DBSOURCE**

**DBSOURCE** contains the original ICU database the data was sourced from. Patients admitted between 2001 - 2008 had their data managed by the CareVue information system, represented in this column as 'carevue'. Patients admitted between 2008 - 2012 had their data managed by the Metavision system, represented in this column as 'metavision'. Knowing the database source is important as the data archiving for these two databases differs in some cases.

### **EVENTTYPE**

**EVENTTYPE** describes what transfer event occurred: 'admit' for an admission, 'transfer' for an inter-hospital transfer and 'discharge' for a discharge from the hospital.

### **PREV\_CAREUNIT, CURR\_CAREUNIT**

**PREV\_CAREUNIT** contains the care unit in which the patient previously resided. **CURR\_CAREUNIT** contains the care unit in which the patient currently resides. The care unit is defined based upon the ward: if the ward is an ICU cost center, then the care unit defines the type of ICU. If the ward is *not* an ICU then in most cases the care unit is null. There are one or two exceptions to this rule. For example, **NWARD** is a ward for newborns.

The `INTIME` and `OUTTIME` of the transfer event correspond to the `CURR_CAREUNIT`. The `PREV_CAREUNIT` for each row is provided for convenience, and is identical to the `CURR_CAREUNIT` of the previous row (assuming the event is not an admission).

Care units include the following:

Care unit	Description
CCU	Coronary care unit
CSRU	Cardiac surgery recovery unit
MICU	Medical intensive care unit
NICU	Neonatal intensive care unit
NWARD	Neonatal ward
SICU	Surgical intensive care unit
TSICU	Trauma/surgical intensive care unit

### **PREV\_WARDID, CURR\_WARDID**

`PREV_WARDID` and `CURR_WARDID` contain the previous and current ward in which the patient stayed. Note that the grouping of physical locations in the hospital database is referred to as a ward. Though in practice ICUs are not referred to as wards, the hospital database technically tracks ICUs as “wards with an ICU cost center”. As a result, each ICU is associated with a `WARDID`, but not every `WARDID` is an ICU.

### **INTIME, OUTTIME**

`INTIME` provides the date and time the patient was transferred into the current care unit from the previous care unit. `OUTTIME` provides the date and time the patient was transferred out of the current care unit.

### **LOS**

`LOS` is the length of stay for the patient for the given ward stay, which may be within or outside of the ICU.

```
+++ date = “2015-09-01T19:33:17-04:00” title = “Help” linktitle = “Help”
weight = 1 toc = “false”
```

```
+++
```

## Help and support

MIMIC is provided through the work of researchers at the MIT Laboratory for Computational Physiology and our collaborators. We have limited resources and cannot provide individual support to researchers worldwide.

We therefore encourage MIMIC users to work together as a community. Please read our notes on how to get involved for further information.

### How can I get help with my analysis?

If you are seeking advice for a specific question, we would suggest raising an issue on the MIMIC Code Repository.

### How can I highlight an issue relating to the MIMIC dataset?

MIMIC is a complex dataset and we understand that there are many open questions, both in terms of the data and its analysis. If you identify an issue with the MIMIC data (for example, errors in the data, unusual characteristics, etc), we suggest raising an issue on the MIMIC Code Repository.

### How do I raise an issue on the MIMIC Code Repository?

To raise an issue, first navigate to the MIMIC Code Repository. After logging in to GitHub, click “New issue”, add a title and description of the problem, and then select the “Submit new issue” button.

Public discussion reaches a larger audience, enables wider discourse, and reduces workload on our laboratory. One exception is where a topic is sensitive, for example relating to patient health information. **If you need to raise a private issue, please contact us using the method noted below.**

## Contact us

If you need to raise an issue privately, outside the public forums, please contact us at: [mimic-support@physionet.org](mailto:mimic-support@physionet.org).