

Core concepts

A few key concepts when working with MIMIC-IV.

Patient identifiers

Patients are identified in the database using three possible identifiers: `subject_id`, `hadm_id`, and `stay_id`. Every unique patient is assigned a unique `subject_id`, all unique hospitalizations are assigned a unique `hadm_id`, and finally all unique ward stays are assigned a unique `transfer_id`. In this context, a ward is a distinct area of the hospital, and a new `transfer_id` is assigned to a patient if the hospital patient tracking system records that they have been moved from one room to another.

However, many patients will move from one specific location to another, but practically their type of care has not changed. A good example is a patient moving bed locations within an ICU: these changes result in the patient having a new `transfer_id`, but the patient never left the ICU and we would consider this as a continuous episode of care. In order to alleviate this issue, we have created a `stay_id`, which is retained across all ward stays of the same type occurring within 24 hours of each other. That is, if a patient leaves and returns to the ICU within 24 hours, they will have the same `stay_id` for the second ICU stay.

`subject_id`

The *patients* table contains information for each unique `subject_id`. `subject_id` is sourced from the hospital, and is an anonymized version of a patient's medical record number.

`hadm_id`

The *admissions* table contains information for each unique `hadm_id`. `hadm_id` is sourced from the hospital, and is an anonymized version of an identifier assigned to each patient hospitalization.

`transfer_id`

The *transfers* table contains information for each unique `transfer_id`. `transfer_id` is an artificially generated identifier which is uniquely assigned to a ward stay for an individual patient.

stay_id

The *transfers* table also contains the `stay_id`. This is an artificially generated identifier which groups reasonably contiguous episodes of care.

The `stay_id` present in *icustays* is derived from the `stay_id` values in the *transfers* table.

date and times

Columns which store a date and time in the database are stored with one of two suffixes: `time` or `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.

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`.

To recap:

- `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.

Other date and time columns present in the database

`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`, `disctime`, `deathtime`

`admittime` and `disctime` 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.

`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.

dod

`dod` is the patient's date of death sourced from one of two sources: the hospital database or a state death database. See the [patients](#) documentation for more detail.

transfertime

`transfertime` is the time at which the patient's service changes.

What's new in MIMIC-IV?

Changes from MIMIC-III to MIMIC-IV.

Many users will be familiar with [MIMIC-III](#), the predecessor of MIMIC-IV. A number of improvements have been made, including simplifying the structure, adding new data elements, and improving the usability of previous data elements.

Table-wise changes are summarized below.

MIMIC-III table	MIMIC-IV module	Removed columns	Additional columns
patients	hosp	row_id, dob, dod_ssn, expire_flag	dod_hosp, anchor_age, anchor_year_group, anchor_year
admissions	hosp	row_id, religion, has_chartevents_data	
transfers	hosp	row_id, icustay_id, prev_careunit, prev_wardid, curr_wardid, los	dbsource, transfer_id, careunit, curr_careunit
d_hcpcs	hosp	N/A, was d_cpt	
d_icd_diagnoses	hosp	row_id, icd9_code, short_title	icd_code, icd_version

MIMIC-III table	MIMIC-IV module	Removed columns		Additional columns	
d_icd_procedures	hosp	row_id, icd9_code, short_title		icd_code, icd_version	
d_labitems	hosp	row_id			
diagnoses_icd	hosp	row_id, icd9_code		icd_code, icd_version	
drgcodes	hosp	row_id			
emar	hosp	N/A, new table.			
emar_detail	hosp	N/A, new table.			
hcupcsevents	hosp	N/A, was cptevents.			
labevents	hosp	row_id		labevent_id, specimen_id, storetime, ref_range_lower, ref_range_upper, priority, comments	
microbiologyevents	hosp	row_id		microevent_id, micro_specimen_id, test_seq, storedate, storetime, test_item_name, test_name, quantity, comments	
pharmacy	hosp	N/A, new table.			
poe	hosp	N/A, new table.			
poe_detail	hosp	N/A, new table.			
prescriptions	hosp	row_id, startdate, enddate, starttime, stoptime, drug_name_poe, drug_name_generic, doses_per_24_hrs, formulary_drug_cd		formulary_drug_cd	
procedures_icd	hosp	row_id, icd9_code		icd_code, icd_version	
services	hosp	row_id			
d_items	icu	row_id, dbsource, conceptid		lownormalvalue, highnormalvalue	
chartevents	icu	row_id, icustay_id, resultstatus, stopped		cgid, error, stay_id	
datetimeevents	icu	row_id, icustay_id, cgid, error, stay_id			

MIMIC-III table	MIMIC-IV module	Removed columns	Additional columns
		resultstatus, stopped	
icustays	icu	row_id, icustay_id	stay_id
inputevents	icu	row_id, icustay_id, cgid, error	stay_id
outputevents	icu	row_id, icustay_id, cgid, error	stay_id
procedureevents	icu	row_id, icustay_id, comments_editedby, comments_canceledby	cgid, stay_id, patientweight, totalamountuom, originalamount

Structure

The structure of MIMIC-IV is necessarily different than MIMIC-III. In MIMIC-III, the set of tables were given as one large set, with no obvious differentiation between them. In MIMIC-IV, we explicitly state the source database of each table. Not only does this clarify the data provenance, but it answers many questions regarding data coverage. For example, as the CHARTEVENTS table is sourced from the ICU clinical information system, it will only provide data for patients while they are in an ICU. Conversely, the LABEVENTS table is sourced from the hospital database, and consequently contains information for a patient's entire hospital stay.

Contemporary

MIMIC-IV contains data from 2008 - 2019 (inclusive). Biomarkers which have been more recently introduced will be available.

CareVue is no more

As the update covers the years 2008 - 2019, the CareVue clinical information system is no longer relevant, as it was not used during that time period. The implications are:

- All `itemid` in `d_items` with a value less than 220000 are no longer relevant.
- The `inputevents_cv` table has been removed. The `inputevents_mv` table was renamed `inputevents`. The structure is otherwise unchanged.
- The `procedureevents_mv` table has been renamed `procedureevents`.

icustay_id is now stay_id

Eventually, stays across different areas of the hospital will be indexed by a unique `stay_id`, such that a stay in the emergency department, ICU, and operating room will all be distinct and referred to by the same identifier. In preparation for this change, `icustay_id` has been renamed `stay_id`.

Years are included

The date-shift strategy in MIMIC has changed. Instead of releasing the day of the week and the season, we have released the approximate year of patient admission. This allows studying patients over time as care practices change.

Audit trails are removed

MIMIC-III contained a number of rows associated with auditing clinical documentation. These rows were marked as erroneous in various ways (`error = 1`, `statusdescription = 'Rewritten'`). These rows have been removed in MIMIC-IV.

Chest x-ray data

Imaging data is also an entirely new addition to MIMIC. The MIMIC-CXR database is [publicly available](#). Notably, the `subject_id` identifier used in the MIMIC-CXR database is consistent with the `subject_id` used in MIMIC-IV. Therefore, all chest x-rays in MIMIC-CXR are linkable to patient stays in MIMIC-IV.

Table-wise improvements over MIMIC-III

A number of enhancements have been made to tables which may be familiar to you from MIMIC-III. Entirely new tables have also been added.

Hospital data

emar and emar_detail

Two entirely new tables are made available, sourced from the relatively newly installed electronic Medicine Administration Record (eMAR) system. Bedside staff will scan barcodes for each individual formulary unit of a medication when administering it. This allows for a granular, high resolution record of when a medication was given.

labevents

- Reference ranges are now available.
- A specimen identifier (`specimen_id`) allows users to group all measurements made for a single specimen (e.g. all blood gas measurements from the same sample of blood).
- A priority column indicates the priority level of the laboratory measure.

microbiologyevents

- Now contains the name of the test performed.

prescriptions

- Instead of `startdate` and `enddate`, *prescriptions* now has `starttime` and `stoptime`.
 - This means all prescriptions now have the date **and** time of start/stop
 - In an internal assessment, only 10 prescriptions were missing the start hour, and 1650 prescriptions were missing the stop hour (there are over 17 million rows in this table).
 - We cannot guarantee the start time is the first instance of patient administration (as these are *prescriptions*), but the added resolution should help in research studies.
- `drug_name_generic`, `drug_name_poe`, and `formulary_drug_cd` have been removed.
 - `drug_name_poe`, when not null, was always equal to `drug`.
 - `drug` is the displayed drug name in the EHR, and is more reliable than `drug_name_generic`.
 - `formulary_drug_cd` was an internal ontology that did not provide additional information over `drug`.
- New columns!
 - `pharmacy_id` - to link to the *pharmacy* table which has additional information about the prescription
 - `form_rx`.
 - `doses_per_24_hrs` provides the number of doses per 24 hours prescribed by this row.

MIMIC-IV Change log

Changes between releases of MIMIC-IV.

The latest version of MIMIC-IV is v2.2.

This page lists changes implemented in sequential updates to the MIMIC-IV database. Issues are tracked using a unique issue number, usually of the form #100, #101, etc. Note that some of these issues are only accessible in a private 'building' repository.

MIMIC-IV v2.2

MIMIC-IV v2.2 was released on January 6, 2023. It added provider identifiers, imputed `hadm_id` for a number of rows in *emar*, and changed the subset of `subject_id` which are held out. Final row counts are available in the validation scripts published with the [MIMIC Code Repository](#). For clarity, after removal of the test set, the row counts are as follows:

- *patients*: 299,712 (was 315,460 in v2.0)
- *admissions*: 431,231 (was 454,324 in v2.0)
- *icustays*: 73,181 (was 76,943 in v2.0)

icu module

- *caregiver*
 - New table in v2.2. Contains one column: **caregiver_id**, a deidentified integer which uniquely represents a single caregiver or provider. These identifiers are sourced from the MetaVision ICU system. When present in a table, it indicates the user who documented the data into MetaVision. For example, the **caregiver_id** associated with a row indicating mechanical ventilation in the *procedureevents* table represents the user who documented the event, and not the provider who performed the procedure.
- *chartevents*, *datetimeevents*, *ingredientevents*, *inpuvents*, *outputevents*, *procedureevents*
 - Added the **caregiver_id** column. This column is a deidentified integer representing the care provider who documented the data for the given row.

hosp module

- *provider*
 - New table in v2.2. Contains one column: **provider_id**, a deidentified string which uniquely represents a single caregiver or provider. These identifiers are sourced from the hospital wide EHR system, and used in a variety of contexts across tables in the module.
- *admissions*
 - New column: **admit_provider_id**, a deidentified string representing the provider who admitted the patient.
- *emar*
 - New column: **enter_provider_id**, a deidentified string representing the provider who entered the medication administration information into the database.
 - Fixed a bug where a subset of *emar* rows (713,117, ~2.5%) did not have an **hadm_id** even though they were associated with a given hospitalization. These rows occur outside of the administratively documented admission and discharge times for a hospitalization, but are still considered as administered during that hospitalization in the raw data.
- *labevents*, *microbiologyevents*, *poe*, *prescriptions*

- New column: `order_provider_id`, a deidentified string representing the provider who ordered the corresponding event (e.g. the lab test in the case of *labevents*, or the medication in the case of *prescriptions*).

MIMIC-IV v2.1

MIMIC-IV v2.1 was released on November 14, 2022. It removed a subset of `subject_id` which will be retained internally as a test set. Future data releases will exclude these patients.

Major changes

- A subset of patients were removed from the dataset. 15,748 `subject_id` were removed from the patients table. 23,093 `hadm_id` were removed from the admissions table. 3,762 `stay_id` were removed from the *icustays* table.

MIMIC-IV v2.0

MIMIC-IV v2.0 was released on June 12, 2022. It focused on expanding the data elements available for patients within MIMIC-IV v1.0. Additional data available includes out-of-hospital date of death, information from the online medical record system (which includes height and weight), and more detail for continuous infusions in the ICU.

Major changes

- The core module has been removed to simplify the schema. The *admissions*, *patients*, and *transfers* tables are now in the hosp module.
- Neonates have been removed from the dataset. Neonatal data will be released in a separate project with data from the neonatal intensive care unit.

icu module

- *icustays*
 - Around 700 stays (~1%) have changed due to the changes in the *patients* table.
- *chartevents*, *d_items*
 - The problem list from MetaVision has been added. All problems are documented with the same `itemid` now present in *d_items*: 220001. There are just over 1,000 unique problems. Most documented problems are related to the care plan for the patient and documented during nurse shift changes (either 7am or 7pm). Less frequently, the ongoing issues are documented here.
- *ingredientevents*
 - This is a new table associated with *inpuvents*. Each intravenous administration tracked in *inpuvents* is associated with a set of ingredients. These ingredients include water content, caloric information, and so on. The goal of the *inpuvents* table is to support nutrition research and to provide a mechanism for estimating fluid input via summing all instances of the water ingredient. These

ingredients have been separated from the *inpu*tevents table to simplify analysis and reduce the size of *inpu*tevents.

- *inpu*tevents
 - Removed a single column which contained only null values: `cancelreason`.
- *procedureevents*
 - Removed columns which contained only null values: `totalamount`, `totalamountuom`, `cancelreason`, `comments_editedby`, `comments_canceledby`, `comments_date`, `secondaryordercategoryname`.

hosp module

- *admissions*
 - Fixed an issue where hospitalizations were missing *edregtime* and *edouttime* when the patient was admitted via the ED (reported in [#1247](#), thanks [@MEladawi](#)).
- *patients*
 - `dod` is now populated with out-of-hospital mortality from state death records. For patients admitted to the ICU, this change has increased capture of date of death from 8,223 records to 23,844 (i.e. we now have out-of-hospital mortality for an additional 15,621 ICU patients).
 - The mechanism for determining patients included in MIMIC was changed. For the most part this has resulted in an improvement, particularly regarding the logic for merging patients who had distinct medical record numbers. As a result of this change, most tables have had a change in the data content. Approximately 1% of stays were affected.
- *transfers*
 - Fixed a bug where the `outtime` for ED stays with no associated `hadm_id` (i.e. an ED stay where the individual was not admitted to the hospital) was incorrect. This resulted in all *transfers* rows with a NULL `hadm_id` having an apparent stay of minutes or less. The `outtime` column has now been corrected.
- *labevents*, *d_labitems*
 - The `itemid` for *d_labitems* has been changed for 43 items. These are extremely infrequently documented and each `itemid` has fewer than 100 observations in *labevents*. The exact `itemid` are provided in the changelog file `CHANGELOG.txt`.
 - Errors were found in the current values of `loinc_code` (reported in [#938](#), thanks [@Mauvila](#)). In order to enable collaborative improvement, the `loinc_code` column has been removed, and will now be collaboratively developed in the [MIMIC Code Repository](#). Initial values will be sourced from the hospital system.
 - A number of labs which previously had the value in the comments field now have the value in the value field (reported in [#941](#), thanks [@Mauvila](#)). This change makes the *labevents* table more consistent with MIMIC-III, which had these values in the value field.
- *microbiologyevents*
 - New organisms, tests, specimens, and antibiotics have been added.
- *omr*
 - A new table has been added: *omr*. The source of this data is the Online Medical Record, and it contains miscellaneous information useful for understanding an

individual's health. As of v2.0, the *omr* table has the following information: blood pressure, height, weight, body mass index, and Estimated Glomerular Filtration Rate (eGFR). These values are available from both inpatient and outpatient visits, and in many cases a "baseline" value from before a patient's hospitalization is available.

- *prescriptions*
 - The **formulary_drug_cd** table has been added back (was previously in MIMIC-III). This column has the same set of values as the **product_code** column of *emar_detail*.

MIMIC-IV v1.0

MIMIC-IV v1.0 was released March 16th, 2021.

core

- *admissions*
 - A number (~1000, <1%) of erroneous **hadm_id** have been removed.
- *patients*
 - 942 **subject_id** have been removed as they were only associated with the above erroneous **hadm_id**.
 - **dod** is now populated using the patient's **deathtime** from their latest hospitalization (reported in [#71](#), thanks [@jinjinzhou](#)).
 - At the moment, out-of-hospital mortality is **not** captured by **dod**
- *transfers*
 - Removed erroneous transfers included in the previous version.
 - **transfer_id** has been regenerated. **transfer_id** in MIMIC-IV v1.0 are **not compatible** with **transfer_id** from v0.4. We do not intend to change **transfer_id** when updating MIMIC-IV, but had to update it due to an error in its generation.
 - All **hadm_id** in transfers are also present in *admissions* and vice-versa (reported in [#84](#), thanks [@kokoko12305](#)).

icu

- *icustays*
 - ICU stays were inappropriately assigned in the previous version due to an error in the preprocessing code. Previously, non-ICU ward transfers were included in the ICU stays, and certain ward stays were not treated as ICU stays (reported in [#67](#), thanks [@JHLiu7](#) and [@stefanhgm](#)). The assignment of **stay_id** has been regenerated.
 - The mapping between hospital transfers and ICU stays has been updated.
 - **stay_id** in MIMIC-IV v1.0 are **not compatible** with **stay_id** from v0.4. We do not intend to change **stay_id** when updating MIMIC-IV, but had to update it due to an error in its generation.
- The change in *icustays* has re-assigned values to new **stay_id**, as a result all tables have had their content changed (due to a change in **stay_id**), but the structure is unchanged.

hosp

- *hcupsevents*
 - Data has been added for a number of previously excluded hospitalizations.
 - The table now has a **chartdate** column, containing the date associated with the code. Every row is associated with a date.
- *drgcodes*
 - Data has been added for a number of previously excluded hospitalizations.
 - Duplicate DRG codes have been removed from the table.
 - Descriptions have been updated using the latest dictionaries made available from [the Massachusetts government website](#) and [HCUP](#).
- *diagnoses_icd, d_icd_diagnoses*
 - Data has been added for a number of previously excluded hospitalizations (reported in [#27](#), thanks [@yugangjia](#)).
 - The icd_code column is now trimmed and stored as a VARCHAR, i.e. codes no longer contain trailing whitespaces (**850** -> **850**).
 - Missing ICD codes have been added to the dictionary. All ICD codes in the diagnoses_icd table have an associated reference in *d_icd_diagnoses*.
- *labevents*
 - The **comments** field has been updated, fixing a bug where comments longer than 4096 characters were truncated. Due to the deidentification, it's unlikely users will see much difference, as these comments will appear as ____.
- *procedures_icd*
 - Data has been added to *procedures_icd* for a number of previously excluded hospitalizations.
 - The table now has a chartdate column, containing the date associated with each billed procedure.
 - The icd_code column is now trimmed and stored as a VARCHAR, i.e. codes no longer contain trailing whitespaces (**850** -> **850**).
 - Missing ICD codes have been added to the dictionary. All ICD codes in the *procedures_icd* table have an associated reference in *d_icd_procedures*.

MIMIC-IV v0.4

MIMIC-IV v0.4 was released August 13th, 2020.

- *d_micro*
 - This table has been removed
- *microbiologyevents*
 - Added the **spec_type_desc**, **test_name**, **org_name**, and **ab_name** columns
 - These columns contain the textual name of the organism/antibiotic/test/specimen
 - Added the **comments** column
 - this column contains information about the test, and in some cases (e.g. viral load tests), contains the result
 - **micro_specimen_id** has been regenerated; the values will not match previous versions.

MIMIC-IV v0.3

MIMIC-IV v0.3 was released July 13th, 2020.

- Fixed an alignment issue in shifted dates/times

MIMIC-IV v0.2

MIMIC-IV v0.2 was released June 23rd, 2020.

- Updated demographics in the patient table
 - `anchor_year` -> `anchor_year_group`
 - `anchor_year_shifted` -> `anchor_year`
 - See the [patients table](#) for detail on these columns
- *transfers*
 - Deleted the `los` column
- *emar*
 - `mar_id` -> `emar_id`
 - `emar_id` is now a composite of `subject_id` and `emar_seq`, and has form "subject_id-emar_seq"
 - `emar_seq` column - a monotonically increasing integer starting with the first eMAR administration
 - Added `poe_id` and `pharmacy_id` columns for linking to those tables
- *emar_detail*
 - `mar_id` -> `emar_id` (changed as above)
 - Deleted the `mar_detail_id` column
- *hcpcsevents*
 - `ticket_id_seq` -> `seq_num`
- *labevents*
 - Many previously NULL values are now populated - these were removed originally due to deidentification
 - Added the `comments` column. This contains deidentified free-text comments with labs. PHI is replaced with three underscores (___). If an entire comment is ___, then the entire comment was scrubbed.
 - `spec_id` -> `specimen_id`
- *microbiologyevents*
 - `stay_id` column removed
 - `spec_id` -> `micro_specimen_id`
- Added the [poe](#) and [poe_detail](#) tables
 - Documentation of provider orders for various treatments and other aspects of patient management
- Added the [prescriptions](#) table
 - Documentation of medicine prescriptions via the provider order interface
- Added the [pharmacy](#) table
 - Detailed information regarding prescriptions provided by the pharmacy including formulary dose, route, frequency, dose, and so on.
- *inpuvents*
 - Fixed an error in the calculation of the *amount* column

- *icustays*
 - Re-derived *stay_id* - the new *stay_id* are distinct from the previous version.
- *diagnosis*
 - Added diagnosis table with similar schema as the *diagnosis_icd* table.
- *main*
 - Removed diagnosis columns from this table (inserted into *diagnosis* above)

MIMIC-IV v0.1

MIMIC-IV v0.1 was released on 15 August 2019.

Modules

Description of the data contained in each of the the MIMIC-IV modules.

Data within the modules are available on PhysioNet:

- hosp, icu: [MIMIC-IV](#)
- ed: [MIMIC-IV-ED](#)
- note: [MIMIC-IV-Note](#)
- cxr: [MIMIC-CXR](#)

The sections below describe data within each module.

[Hosp](#)

The Hosp module provides all data acquired from the hospital wide electronic health record. Information covered includes patient and admission information, laboratory measurements, microbiology, medication administration, and billed diagnoses.

[ICU](#)

The ICU module contains information collected from the clinical information system used within the ICU. Documented data includes intravenous administrations, ventilator settings, and other charted items.

[ED](#)

The ED module contains data for emergency department patients collected while they are in the ED. Information includes reason for admission, triage assessment, vital signs, and medicine reconciliation.

The *subject_id* and *hadm_id* identifiers allow MIMIC-IV-ED to be linked to other MIMIC-IV modules.

[CXR](#)

The CXR module provides lookup tables linking patient identifiers with MIMIC-CXR `study_id` and `dicom_id`, allowing analysis of patient chest x-rays to be linked with the clinical data from other MIMIC-IV modules.

[ECG](#)

The ECG module provides waveform data, and lookup tables which can be used to link subjects to other MIMIC-IV modules via `subject_id`. A `study_id` allows for linking within the ECG module while `note_id` allows linking to the cardiologist note in the MIMIC-IV-Note module.

[Note](#)

(NOT PUBLICLY AVAILABLE): The Note module contains deidentified free-text clinical notes for hospitalized patients.

Hosp

The Hosp module provides all data acquired from the hospital wide electronic health record. Information covered includes patient and admission information, laboratory measurements, microbiology, medication administration, and billed diagnoses.

The hosp module contains data derived from the hospital wide EHR. These measurements are predominantly recorded during the hospital stay, though some tables include data from outside the hospital as well (e.g. outpatient laboratory tests in `labevents`). Information includes patient and admission details (`patients`, `admissions`, `transfers`), laboratory measurements (`labevents`, `d_labitems`), microbiology cultures (`microbiologyevents`), provider orders (`poe`, `poe_detail`), medication administration (`emar`, `emar_detail`), medication prescription (`prescriptions`, `pharmacy`), hospital billing information (`diagnoses_icd`, `d_icd_diagnoses`, `procedures_icd`, `d_icd_procedures`, `hcupsevents`, `d_hcupcs`, `drgcodes`), and hospital service related information (`services`).

[omr table](#)

The Online Medical Record (OMR) table contains miscellaneous information from the EHR.

[provider table](#)

The provider table lists deidentified provider identifiers used in the database.

[admissions table](#)

Detailed information about hospital stays.

[d_hcpcs](#)

Dimension table for *hcpcsevents*; provides a description of CPT codes.

[d_icd_diagnoses](#)

Dimension table for *diagnoses_icd*; provides a description of ICD-9/ICD-10 billed diagnoses.

[d_icd_procedures](#)

Dimension table for *procedures_icd*; provides a description of ICD-9/ICD-10 billed procedures.

[d_labitems](#)

Dimension table for *labevents* provides a description of all lab items.

[diagnoses_icd](#)

Billed ICD-9/ICD-10 diagnoses for hospitalizations.

[drgcodes](#)

Billed diagnosis related group (DRG) codes for hospitalizations.

[emar](#)

The Electronic Medicine Administration Record (eMAR); barcode scanning of medications at the time of administration.

[emar_detail](#)

Supplementary information for electronic administrations recorded in *emar*.

[hpcsevents](#)

Billed events occurring during the hospitalization. Includes CPT codes.

[labevents](#)

Laboratory measurements sourced from patient derived specimens.

[microbiologyevents](#)

Microbiology cultures.

[patients table](#)

Patients' gender, age, and date of death if information exists.

[pharmacy](#)

Formulary, dosing, and other information for prescribed medications.

[poe](#)

Orders made by providers relating to patient care.

[poe_detail](#)

Supplementary information for orders made by providers in the hospital.

[prescriptions](#)

Prescribed medications.

[procedures_icd](#)

Billed procedures for patients during their hospital stay.

[services](#)

The hospital service(s) which cared for the patient during their hospitalization.

[transfers table](#)

Detailed information about patients' unit transfers.

ICU

The ICU module contains information collected from the clinical information system used within the ICU. Documented data includes intravenous administrations, ventilator settings, and other charted items.

The ICU module contains data sourced from the clinical information system at the BIDMC: MetaVision (iMDSOFT). MetaVision tables were denormalized to create a star schema where the icustays and d_items tables link to a set of data tables all suffixed with "events". Data documented in the icu module includes intravenous and fluid inputs (inputevents), ingredients of the aforementioned inputs (ingredientevents), patient outputs (outputevents), procedures (procedureevents), information documented as a date or time (datetimeevents), and other charted information (chartevents). All events tables contain a stay_id column allowing identification of the associated ICU patient in icustays, and an itemid column allowing identification of the concept documented in d_items.

[caregiver table](#)

The caregiver table lists deidentified provider identifiers used in the ICU module.

[d_items](#)

Dimension table describing itemid. Defines concepts recorded in the events table in the ICU module.

[chartevents](#)

Charted items occurring during the ICU stay. Contains the majority of information documented in the ICU.

[datetimeevents](#)

Documented information which is in a date format (e.g. date of last dialysis).

[ICU stays](#)

Tracking information for ICU stays including admission and discharge times.

[Ingredientevents](#)

Ingredients of continuous or intermittent administrations including nutritional and water content.

[Inputevents](#)

Information documented regarding continuous infusions or intermittent administrations.

[outputevents](#)

Information regarding patient outputs including urine, drainage, and so on.

[procedureevents](#)

Procedures documented during the ICU stay (e.g. ventilation), though not necessarily conducted within the ICU (e.g. x-ray imaging).

ED

The ED module contains data for emergency department patients collected while they are in the ED. Information includes reason for admission, triage assessment, vital signs, and medicine reconciliation. The **subject_id** and **hadm_id** identifiers allow MIMIC-IV-ED to be linked to other MIMIC-IV modules.

[diagnosis table](#)

diagnosis table

[edstays table](#)

edstays table

[medrecon table](#)

medrecon table

[pyxis table](#)

pyxis table

[triage table](#)

triage table

[vitalsign table](#)

vitalsign table

CXR

The CXR module provides lookup tables linking patient identifiers with MIMIC-CXR **study_id** and **dicom_id**, allowing analysis of patient chest x-rays to be linked with the clinical data from other MIMIC-IV modules.

In order to access this module, you must sign the data use agreement for MIMIC-CXR and request access to MIMIC-CXR data on BigQuery via the [MIMIC-CXR PhysioNet project page](#).

MIMIC-CXR contains 227,835 imaging studies for 64,588 patients presenting to the Beth Israel Deaconess Medical Center Emergency Department between 2011 - 2016. A total of 377,110 images are available in the dataset. Each imaging study can contain one or more images, usually a frontal view and a lateral view. Studies are made available with a semi-structured free-text radiology report that describes the radiological findings of the images, written by a practicing radiologist contemporaneously during routine clinical care. All images and reports have been de-identified to protect patient privacy.

A records file, `cxr-record-list.csv.gz`, provides a mapping between the image (`dicom_id`), the study (`study_id`), and the patient (`subject_id`). Another records file, `cxr-study-list.csv.gz`, provides a mapping between the studies (`study_id`) and patients (`subject_id`).

All patient identifiers begin with the digit 1 and have a total length of 8 digits. All study identifiers begin with the digit 5 and have a total length of 8 digits. DICOM file names are unique 40 character hexadecimal strings with dashes separating groups of eight characters.

Images are provided in DICOM format; see the [image](#) section for more information about the images.

Reports are provided as plain text files; see the [reports](#) section for more information about the reports.

Data Organization

Data files are made available in a hierarchical structure. The following block lists the first patient's records as an demonstrative example (MIMIC-CXR v2.0.0):

```
files/  
p10/  
  p10000032/  
    s50414267/  
      02aa804e-bde0afdd-112c0b34-7bc16630-4e384014.dcm.gz  
      174413ec-4ec4c1f7-34ea26b7-c5f994f8-79ef1962.dcm.gz  
    s53189527/  
      2a2277a9-b0ded155-c0de8eb9-c124d10e-82c5caab.dcm.gz  
      e084de3b-be89b11e-20fe3f9f-9c8d8dfe-4cfd202c.dcm.gz  
    s53911762/  
      68b5c4b1-227d0485-9cc38c3f-7b84ab51-4b472714.dcm.gz  
      fffabebf-74fd3a1f-673b6b41-96ec0ac9-2ab69818.dcm.gz  
    s56699142/  
      ea030e7a-2e3b1346-bc518786-7a8fd698-f673b44c.dcm.gz  
    s50414267.txt  
    s53189527.txt  
    s53911762.txt  
    s56699142.txt  
  ...
```

You will note a high level folder: p10. This is done to avoid having many files in a single directory. All patient folders are stored in a higher level folder which is identical to the first 3 characters of their folder name, i.e. p10000032 will be in folder p10, p11000011 will be in p11, and so on.

Above, this patient (10000032) has four studies. Most of the studies have two scans (usually a frontal and a lateral chest x-ray), but one study 56699142 has only one image. Each study is associated with a de-identified free-text radiology report (e.g. s56699142.txt). Note that the identifiers are random, and do not indicate order of the studies in any way.

Images

Chest radiographs were sourced from the hospital picture archiving and communication system (PACS) in Digital Imaging and Communications in Medicine (DICOM) format. DICOM is a common format which facilitates interoperability between medical imaging devices. Put simply, the DICOM format contains structured meta-data associated with one or more images, and the DICOM standard stipulates strict rules around the structure of this information. The DICOM standard is updated regularly each year. MIMIC-CXR is built according to the [DICOM Standard version 2017e](#).

Deidentified free-text radiology reports

During routine care, radiologists will review chest radiographs and document their interpretation electronically. When reviewing a radiograph, radiologists have access to: (1) brief text written by another clinician summarizing the underlying medical condition, (2) the reason for examination, and (3) prior imaging studies performed.

Reports in MIMIC-CXR are semi-structured, and have linebreaks to ensure individual lines are no longer than 79 characters. As reports are templated, structure is seeded in the reports, but radiologists are free to modify it as they will before saving. Most reports will contain a FINDINGS'' and IMPRESSION''. The findings section details the radiologists assessment of the image, while the impression section acts as a summary of the most pertinent findings.

Reports sometimes have addendums at the top. Addenums are added after the radiology report has already been written, and are intended to clarify language as necessary. Addendums are delimited from the original report by underscores which span an entire line.

Radiology reports have been de-identified to protect patient privacy. All patient information has been replaced with three underscores (___). Provider information has also been removed.

The following is an example radiology report from MIMIC-CXR:

```

                                FINAL REPORT
EXAMINATION:  CHEST (PA AND LAT)

INDICATION:  ___ year old woman with ?pleural effusion  //  ?pleural effusion

TECHNIQUE:  Chest PA and lateral

COMPARISON:  ___

FINDINGS:

Cardiac size cannot be evaluated.  Large left pleural effusion is new.  Small
right effusion is new.  The upper lungs are clear.  Right lower lobe
opacities
are better seen in prior CT.  There is no pneumothorax.  There are mild
degenerative changes in the thoracic spine

IMPRESSION:

Large left pleural effusion
```

MIMIC-CXR v2.0.0

The current version of the database is v2.0.0. When referencing this version, we recommend using the full title: MIMIC-CXR v2.0.0.

MIMIC-CXR v2.0.0 is the first release of the images in their native format, DICOM, and the first release of the free-text radiology reports associated with these images. All data has been de-identified prior to release to protect patient privacy.

Past versions

MIMIC-CXR v1.0.0

MIMIC-CXR v1.0.0 was released on 22 January 2019. The data contains only JPG format images and 14 structured labels extracted from an NLP tool. The images are identical to MIMIC-CXR v2.0.0, and we no longer distribute v1.0.0.

Researchers interested in JPG format images with NLP derived labels can find them in the [MIMIC-CXR-JPG project](#).

ECG

The ECG module provides waveform data, and lookup tables which can be used to link subjects to other MIMIC-IV modules via **subject_id**. A **study_id** allows for linking within the ECG module while **note_id** allows linking to the cardiologist note in the MIMIC-IV-Note module.

The MIMIC-IV-ECG module contains approximately 800,000 diagnostic electrocardiograms across nearly 160,000 unique patients.

There are three lookup tables in this module:

- A records file, `record-list.csv`, provides a mapping between the study (`study_id`), the path to the diagnostic ECG waveform (`path`), and the patient (`subject_id`).
- A machine measurements file, `machine_measurements.csv`, provides the information generated from the ECG machine including global (across all 12 leads) summary measurements (RR interval, QRS onset, etc.), the machine generated report, as well as machine and cart information. It also provides a study (`study_id`) and patient (`subject_id`) identifier.
- A note linking file, `waveform_note_links.csv`, provides a `note_id` for each waveform that has an available free-text cardiologist note in the MIMIC-IV-Note module. It also provides a study (`study_id`) and patient (`subject_id`) identifier.

Additional details for each lookup table are provided in the pages below.

Please note that the free-text cardiologist notes for the ECG module will be publicly released in the MIMIC-IV-Note module at a later date.

For additional information about this project please see the MIMIC-IV-ECG project on physionet.org.

[ECG Record Table](#)

The table which lists all available records in this module.

[Machine Measurement](#)

A table with ECG machine information and output measurements / reports.

[Waveform Note Links Table](#)

A table which provides links between waveforms and available cardiologist notes.

Note

(NOT PUBLICLY AVAILABLE): The Note module contains deidentified free-text clinical notes for hospitalized patients.

MIMIC-Note is currently not publicly available and the structure is subject to change.

[Discharge](#)

Discharge summaries for hospitalizations.

[Discharge detail](#)

Auxiliary information for discharge summaries.

[Radiology](#)

Radiology reports

[Radiology detail](#)

Auxiliary information for radiology notes.

BigQuery Tutorial

BigQuery UI

The aim of this tutorial is to get you familiarized with BigQuery web UI to query/filter/aggregate/export data.

This tutorial is based upon one from the [Google Healthcare datathon repository](#). It is written with the *old* BigQuery interface, and focuses on the MIMIC-III/eICU-CRD datasets.

Prerequisites

- You should already have had a valid Google account with access to the datasets.
 - If you do not have a Google account, you can create one at <http://www.gmail.com>. You need to add this e-mail to your PhysioNet account.
- Access to MIMIC-III/eICU-CRD can be done via the their PhysioNet project pages.
- All users have access to the demo datasets.

PhysioNet does not cover the cost of queries against `physionet-data` (though this cost is mostly trivial). In order to run queries, you will need to configure a *project* for your account, which BigQuery can then use to bill for your usage of the cloud platform. For more information on GCP projects, see [the documentation on creating and managing projects](#).

All PhysioNet data is hosted on the `physionet-data` project. You will only have read-access privileges to these datasets. As a result, if you would like to save the results of any queries, you will need to save them to a dataset created on your own project.

TLDR

In this section we are going to run a query to briefly showcase BigQuery's capability. The goal is to aggregate the mimic demo data to find out the correlation between age and the average length of stay in hours in ICU.

Run the following query from BigQuery web interface (See "Executing Queries" section above for how to access BigQuery web interface).

New Query ?

```
1 #standardSQL
2 WITH re AS (
3   SELECT
4     TIMESTAMP_DIFF(icu.outtime, icu.intime, HOUR) AS icu_length_of_stay,
5     DATE_DIFF(DATE(icu.intime), DATE(pat.dob), YEAR) AS age
6   FROM `datathon-datasets.mimic_demo.icustays` AS icu
7   INNER JOIN `datathon-datasets.mimic_demo.patients` AS pat
8     ON icu.subject_id = pat.subject_id
9   SELECT
10     icu_length_of_stay AS stay,
11     age
```

RUN QUERY

Save Query

Save View

Format Query

Show Options

Query complete (1.8s elapsed, 4.75 KB processed)

```
#standardSQL
WITH re AS (
```

```

SELECT
  DATETIME_DIFF(icu.outtime, icu.intime, HOUR) AS icu_length_of_stay,
  DATE_DIFF(DATE(icu.intime), DATE(pat.dob), YEAR) AS age
FROM `physionet-data.mimiciii_demo.icustays` AS icu
INNER JOIN `physionet-data.mimiciii_demo.patients` AS pat
  ON icu.subject_id = pat.subject_id)
SELECT
  age,
  AVG(icu_length_of_stay) AS stay
FROM re
WHERE age < 100
GROUP BY age
ORDER BY age

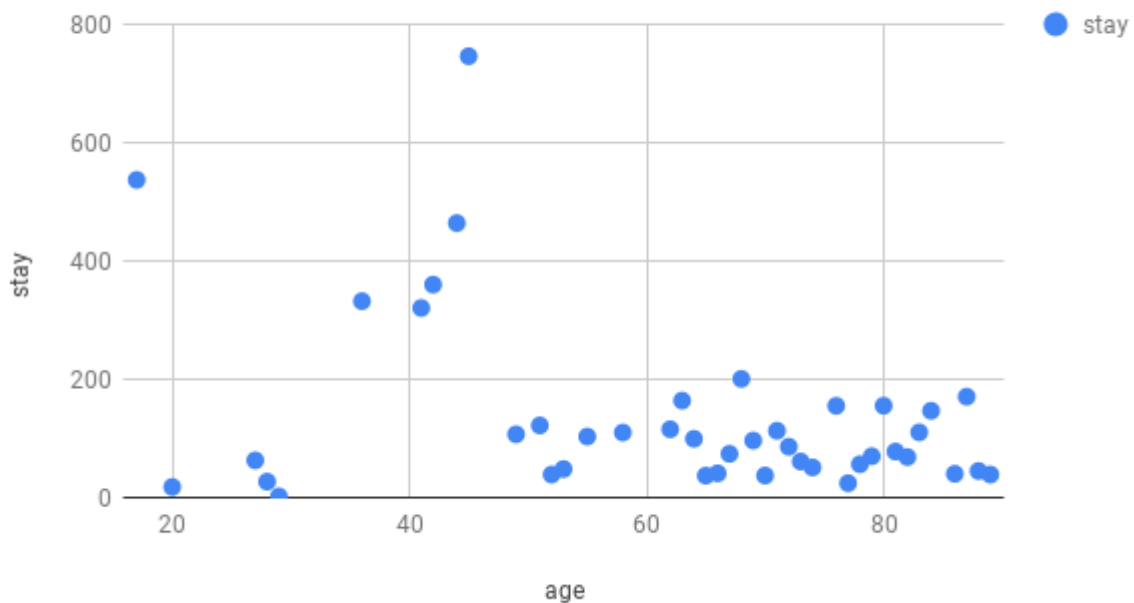
```

You can download the returned result as a CSV file and generate a chart with your preferred tools.

Results		Details		Download as CSV	Download as JSON	Save as Table	Save to Google Sheets
Row	stay	age					
1	537	17					
2	18	20					
3	63	27					
4	27	28					
5	2	29					
6	332	36					
7	469	41					
8	172	41					
9	360	42					
10	30	44					

The following is a scatter chart plotted from the result with Google Sheets.

stay vs. age



Detailed Tutorial

BigQuery Basics

Feel free to skip this section if you are already familiar with BigQuery.

BigQuery Table Name

A BigQuery table is uniquely identified by the three-layer hierarchy of project ID, dataset ID and table name. For example in the following query:

```
SELECT
  subject_id
FROM
  `physionet-data.mimiciii_demo.icustays`
LIMIT 10
```

`physionet-data.mimiciii_demo.icustays` specifies the table we are querying, where `physionet-data` is the project that hosts the datasets, `mimiciii_demo` is the name of the dataset, and `icustays` is the table name. Backticks (```) are used as there is a non-standard character (`-`) in the project name. If the dataset resides in the same project, you can safely omit the project name, e.g. `my-project.my_dataset.my_table` can be written as `my_dataset.my_table` instead.

SQL Dialect

BigQuery supports 2 SQL dialects, legacy and standard. During this datathon we highly recommend using standard SQL dialect.

Follow the steps below to make sure the StandardSQL dialect is used:

1. Click "COMPOSE QUERY" on top left corner;
2. Click "Show Options" below the input area;
3. Lastly, make sure "Use Legacy SQL" is **NOT** checked, and click "Hide Options".

The screenshot shows the BigQuery query options panel. The 'SQL Dialect' section is highlighted with a red circle around the 'Use Legacy SQL' checkbox, which is currently unchecked. The panel includes various configuration options for the query execution, such as 'Destination Table', 'Write Preference', 'Results Size', 'Results Schema', 'Query Caching', 'Query Priority', 'UDF Source URIs', 'Maximum Bytes Billed', 'Destination Encryption', and 'Processing Location'. At the bottom, there are buttons for 'RUN QUERY', 'Save Query', 'Save View', 'Format Query', and 'Hide Options'.

Alternatively, ["#standardSQL" tag](#) can be prepended to each query to tell BigQuery the dialect you are using, which is what we used in the TLDR section above.

MIMIC-III Basics

Dataset Exploration

As mentioned previously, the datasets are hosted in a different project, which can be accessed [here](#). On the left panel, you will see the `mimiciii_demo` dataset, under which you will see the table names.

To view the details of a table, simply click on it (for example the `icustays` table). Then, on the right side of the window, you will have to option to see the schema, metadata and preview of rows tabs.

Queries

Most of the following queries are adapted from the [MIMIC cohort selection tutorial](#).

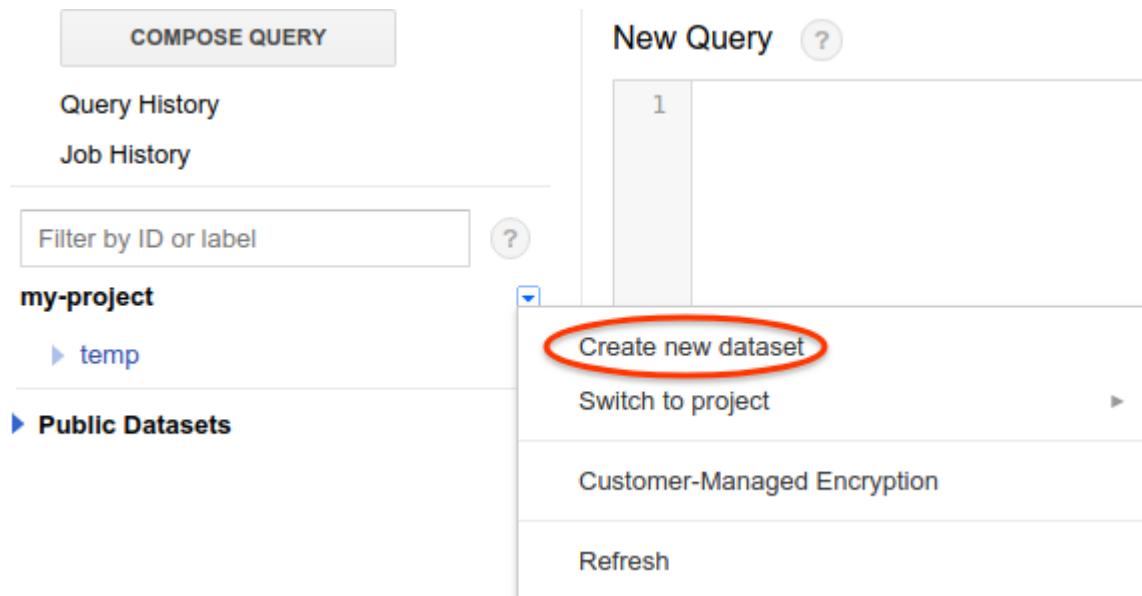
Analysis

Let's take a look at a few queries. To run the queries yourself, copy the SQL statement to the input area on top of the web interface and click the red "RUN QUERY" button.

```
SELECT
  subject_id,
  hadm_id,
  icustay_id,
  intime,
  outtime,
  DATETIME_DIFF(outtime, intime, DAY) AS icu_length_of_stay
FROM `physionet-data.mimiciii_demo.icustays`
```

Let's save the result of previous query to an intermediate table for later analysis:

1. Create a dataset by clicking the caret below the search box on the left sidebar, and choose "Create new dataset";
 - Set dataset ID to "temp" and data expiration to 2 days;
 - Click "OK" to save the dataset.
2. Click "Save to table" button on the right;
 - Set destination dataset to "temp" and table to "icustays", use the default value for project;
 - Click "OK" to save the table, it usually takes less than a few seconds for demo tables.



Now let's take a look at a query that requires table joining: include the patient's age at the time of ICU admittance. This is computed by the date difference in years between the ICU intime and the patient's date of birth. The former is available in the icustays table, and the latter resides in the dob column of the patients table.

```
SELECT
  icu.subject_id,
  icu.hadm_id,
  icu.icustay_id,
  pat.dob,
  icu.icu_length_of_stay,
  DATE_DIFF(DATE(icu.intime), DATE(pat.dob), YEAR) AS age
FROM `physionet-data.mimiciii_demo.patients` AS pat
INNER JOIN `temp.icustays` AS icu
  ON icu.subject_id = pat.subject_id
```

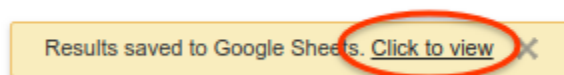
Again, let's save the table as "pat_icustays" in the "temp" dataset for use later. Briefly look at the age of patients when they are admitted with the following query.

Now let's run the following query to produce data to generate a histogram graph to show the distribution of patient ages in ten-year buckets (i.e. [0, 10), [10, 20), ..., [90, ∞).

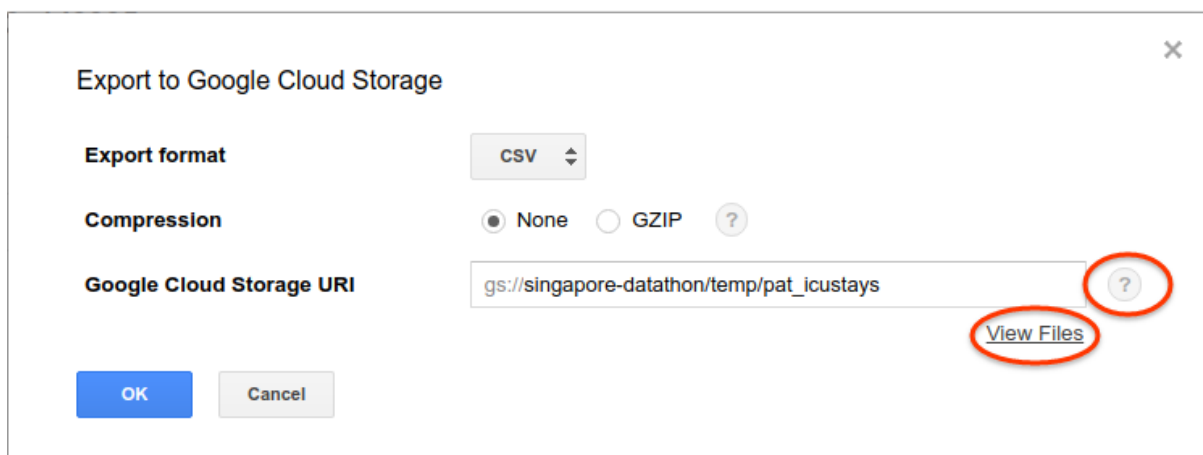
```
WITH bu AS (
  SELECT
    CAST(FLOOR(age / 10) AS INT64) AS bucket
  FROM `temp.pat_icustays`)
SELECT
  IF(bucket >= 9, ">= 90", FORMAT("%d - %d", bucket * 10, (bucket + 1) * 10))
  AS age,
  COUNT(bucket) AS total
```

```
FROM bu
GROUP BY bucket
ORDER BY bucket ASC
```

Now click "Save to Google Sheets" button and wait 1-2 seconds until a yellow notification shows up, click "Click to view" which leads you to Google Spreadsheet in a new browser window. As you can see, the data from our last query is dumped into a spreadsheet. By clicking "Insert -> Chart" from the menu bar on top, a nice histogram graph is automatically created for us!



If you prefer using other tools to process the final result, a CSV file can be downloaded by clicking the "Download as CSV" button. If downloading fails because the file is too large (we highly recommend aggregating the data to a small enough result before downloading though), you can save it to a temporary table, click the caret then "Export table" button from the dropdown menu and save it to Google Cloud Storage, then you can download the file from [GCS](#).



Now let's see if there is correlation between age and average length of stay in hours. Since we are using the age of patients when they get admitted, so we don't need to worry about multiple admissions of patients. Note that we treat the redacted ages (> 90) as noises and filter them out.

```
WITH re AS (
SELECT
  DATETIME_DIFF(icu.outtime, icu.intime, HOUR) AS icu_length_of_stay,
  DATE_DIFF(DATE(icu.intime), DATE(pat.dob), YEAR) AS age
FROM `physionet-data.mimiciii_demo.icustays` AS icu
INNER JOIN `physionet-data.mimiciii_demo.patients` AS pat
  ON icu.subject_id = pat.subject_id)
SELECT
```

```
icu_length_of_stay AS stay,  
age  
FROM re  
WHERE age < 100
```

Follow the same steps to save the result to Google Spreadsheet, by default a linear chart is generate. We will need to change the chart type to scatter chart through the chart editor on the right.

Useful Tips

Saving View

Datathon organizers might not allow you to create new tables. However, you can save a view of a query's output to then use in later queries.

1. **Create a temporary dataset in the datathon project.** Next to the datathon project in the left side of the BigQuery UI, click the arrow and then Create new dataset. Give the dataset a temporary name that can be identified to your team (like 'team6temp').
2. **Save the view.** After running your query, click the button next to Run Query that says Save view. Select the temporary dataset you created and then give the view a name.
3. **Query your view.** Now you can perform a query using the syntax project.dataset.view like the following:

```
SELECT * FROM `datathon_project.team6temp.our_custom_view`;
```

Working with DATETIME

The times in the tables are stored as DATETIME objects. This means you cannot use operators like <, =, or > for comparing them.

- Use the [DATETIME functions](#) in BigQuery. An example would be if you were trying to find things within 1 hour of another event. In that case, you could use the native DATETIME_SUB() function. In the example below, we are looking for stays of less than 1 hour (where the admit time is less than 1 hour away from the discharge time). [...] WHERE ADMITTIME BETWEEN DATETIME_SUB(DISCHTIME, INTERVAL 1 HOUR) AND DISCHTIME
- If you are more comfortable working with timestamps, you can cast the DATETIME object to a TIMESTAMP object and then use the [TIMESTAMP functions](#).

Input / Output Options

There are a few cases where you may want to work with files outside of BigQuery. Examples include importing your own custom Python library or saving a dataframe. [This tutorial](#) covers importing and exporting from local filesystem, Google Drive, Google Sheets, and Google Cloud Storage.

Data Description

MIMIC-IV is grouped into two modules: *hosp*, and *icu*. Organization of the data into these modules reflects their provenance: data in the *hosp* module is sourced from the hospital wide EHR, while data in the *icu* module is sourced from the in-ICU clinical information system (MetaVision). A total of 364,627 unique individuals are in MIMIC-IV v3.0, each represented by a unique `subject_id`. These individuals had 546,028 hospitalizations and 94,458 unique ICU stays.

hosp

The *hosp* module contains detailed data regarding 546,028 unique hospitalizations for 223,452 unique individuals. Measurements in the *hosp* module are predominantly recorded during the hospital stay, though some tables include data from outside an admitted hospital stay as well (e.g. outpatient or emergency department laboratory tests in *labevents*). Patient demographics (*patients*), hospitalizations (*admissions*), and intra-hospital transfers (*transfers*) are recorded in the *hosp* module. Other information in the *hosp* module includes laboratory measurements (*labevents*, *d_labitems*), microbiology cultures (*microbiologyevents*, *d_micro*), provider orders (*poe*, *poe_detail*), medication administration (*emar*, *emar_detail*), medication prescription (*prescriptions*, *pharmacy*), hospital billing information (*diagnoses_icd*, *d_icd_diagnoses*, *procedures_icd*, *d_icd_procedures*, *hcupcsevents*, *d_hcpcs*, *drgcodes*), online medical record data (*omr*), and service related information (*services*).

Provider information is available in the *provider* table. The `provider_id` column is a deidentified character string which uniquely represents a single care provider. As `provider_id` is used in different contexts across the module, a prefix is usually present in data tables to contextualize how the provider relates to the event. For example, the provider who admits the patient to the hospital is documented in the *admissions* table as `subject_id`. All columns which have a suffix of `provider_id` may be linked to the *provider* table.

Deidentified dates and aligning stays to year groups

All dates in MIMIC-IV have been deidentified by shifting the dates into a future time period between 2100 - 2200. This shift is done independently for each patient, and as a result two patients admitted in the deidentified year 2120 cannot be assumed to be admitted in the same year. To provide information about the original time period when a patient was admitted, the *patients* table provides a set of columns with the "anchor_" prefix. The `anchor_year` column is a deidentified year occurring sometime between 2100 - 2200, and the `anchor_year_group` column is one of the following values: "2008 - 2010", "2011 - 2013", "2014 - 2016", "2017 - 2019", and "2020 - 2022". These pieces of information allow researchers to infer the approximate year a patient received care. For example, if a patient's `anchor_year` is 2158, and their `anchor_year_group` is 2011 - 2013, then any hospitalizations for the patient occurring in the year 2158 actually occurred sometime between 2011 - 2013. In order to minimize accidental release of information, only a single `anchor_year` is provided per `subject_id`. Consequently, individual stays must be aligned to the anchor year using the respective date (e.g. `admittime`). Finally, the `anchor_age` provides the patient age in the given `anchor_year`. If the patient was over 89 in the `anchor_year`, this `anchor_age` has been set to 91 (i.e. all patients over 89 have been grouped together into a single group with value 91, regardless of what their real age was).

Out of hospital linkage of date of death

Date of death is available within the `dod` column of the *patients* table. Date of death is derived from hospital records and state records. If both exist, hospital records take precedence. State records were matched using a custom rule based linkage algorithm based on name, date of birth, and social security number. State and hospital records for date of death were collected two years after the last patient discharge in MIMIC-IV, which should limit the impact of reporting delays in date of death.

Dates of death occurring more than one year after hospital discharge are censored as a part of the deidentification process. As a result, the maximum time of follow up for each patient is exactly one year after their last hospital discharge. For example, if a patient's last hospital discharge occurs on 2150-01-01, then the last possible date of death for the patient is 2151-01-01. If the individual died on or before 2151-01-01, and it was captured in either state or hospital death records, then the `dod` column will contain the deidentified date of death. If the individual survived for at least one year after their last hospital discharge, then the `dod` column will have a NULL value.

icu

The *icu* module contains data sourced from the clinical information system known as MetaVision (iMDSoft). MetaVision tables were denormalized to create a star schema

where the *icustays* and *d_items* tables link to a set of data tables all suffixed with "events". Data documented in the *icu* module includes intravenous and fluid inputs (*inpuvents*), ingredients for the aforementioned inputs (*ingredientevents*), patient outputs (*outpuvents*), procedures (*procedureevents*), information documented as a date or time (*datetimeevents*), and other charted information (*chartevents*). All events tables contain a *stay_id* column allowing identification of the associated ICU patient in *icustays*, and an *itemid* column allowing identification of the concept documented in *d_items*. Additionally, the *caregiver* table contains *caregiver_id*, a deidentified integer representing the care provider who documented data into the system. All events tables

(*chartevents*, *datetimeevents*, *ingredientevents*, *inpuvents*, *outpuvents*, *procedureevents*) have a *caregiver_id* column which links to the *caregiver* table.

The *icu* module contains a total of 94,458 ICU stays for 65,366 unique individuals as of MIMIC-IV v3.0. An ICU stay is defined as a contiguous sequence of transfers within a unit of the hospital classified as an ICU, and the *icustays* table is derived from the *transfers* table. During the creation of the *icustays* table, consecutive transfers within an ICU were merged into the same *stay_id* for analytical convenience, as these transfers are often bed number changes. Importantly, non-consecutive ICU stays remain as unique *stay_id* in the *icustays* table. In some cases, these could be considered the "same" ICU stay as the patient was transferred out for a planned procedure. In other cases, these are unanticipated readmissions to the ICU. As there was no systematically perfect method to differentiate these cases, we did not attempt to merge non-consecutive *stay_id*, and it is up to the investigator to appropriately handle these cases.

Usage Notes

The data described here are collected during routine clinical practice and reflect the idiosyncrasies of that practice. Implausible values may be present in the database as an artifact of the archival process. Researchers should follow best practice guidelines when analyzing the data.

Documentation

Up to date documentation for MIMIC-IV is available on the MIMIC-IV website [4]. We have created an open source repository for the sharing of code and discussion of the database, referred to as the MIMIC Code Repository [5, 6]. The code repository provides a mechanism for shared discussion and analysis of all versions of MIMIC, including MIMIC-IV.

Linking MIMIC-IV to emergency department, note, and chest x-ray data

MIMIC-IV is linkable to other MIMIC projects published on PhysioNet. Where possible, we have prefixed the other projects with "MIMIC-IV" to make this clear such as MIMIC-IV-ED. Note that MIMIC-CXR is also linkable although it is not prefixed with MIMIC-IV. Free-text clinical notes are available in MIMIC-IV-Note [7], observations made in the emergency department are available in MIMIC-IV-ED [8], and chest x-rays in MIMIC-CXR [9].

Linking the other datasets to MIMIC-IV requires two steps. The first step is to match the data using `subject_id`, taking care to note that MIMIC-IV is a superset of other modules, and sampling biases may be introduced by the linking process. For example, MIMIC-CXR is only available between 2011 - 2016 for patients who were admitted to the emergency department, and this selection bias impacts the patient cohort. The second step involves aligning the dates. Since all modules are deidentified by the same shift, the time periods for measurements overlap. For example, if a patient is admitted to the hospital on 2105-01-01, discharged on 2105-01-03, and has an x-ray in MIMIC-CXR on 2105-01-02, then it is correct to assume the x-ray was taken while the patient was admitted to the hospital.

Patient composition

MIMIC-IV contains patients admitted to the emergency department and the intensive care unit. While patients admitted to the intensive care unit must have an associated hospitalization, patients may be admitted to the emergency department without being subsequently admitted to the hospital. As a result, the number of patients in MIMIC-IV is much higher than the number of unique patients with hospitalizations. As of MIMIC-IV v3.0 there are 364,627 unique patients, of whom 223,452 had at least one hospitalization (i.e. at least one record in the *admissions* table). The remaining 141,175 patients were only seen in the emergency department, which can be verified using the *transfers* table.

Release Notes

MIMIC-IV v3.1

MIMIC-IV v3.1 was released in October, 2024. This release fixed minor bugs raised by the community:

- The `itemid` values in the *d_labitems* and *labevents* tables had changed for a subset of laboratory measurements between v2.2 and v3.0. This change was not intentional. The tables

have been updated, and the *d_labitems* and *labevents* *itemid* values have been verified to be consistent with v2.2.

- Two *subject_id* were present in various data tables but were not present in the *patients* table. These subjects have been removed from the data tables. Database constraints with a foreign key to the *subject_id* column in the *patients* table should now work correctly.

If upgrading from v3.0, note that only the following tables were modified (and thus require updating):

- *d_labitems*
- *diagnoses_icd*
- *drgcodes*
- *labevents*
- *microbiologyevents*
- *omr*
- *transfers*
- *icustays*

MIMIC-IV v3.0

MIMIC-IV v3.0 was released on July 23, 2024. Stays occurring between 2020 and 2022, inclusive, were added to the database. Out of hospital mortality is available for up to 1-year post hospital or ED discharge. The number of additional patients, admissions, and stays are highlighted by the increased row counts of their respective tables:

- *patients*: 364,627 (was 299,712 in v2.2)
- *admissions*: 546,028 (was 431,231 in v2.2)
- *icustays*: 94,458 (was 73,181 in v2.2)

Other changes include:

- Improved language data. The language column of *admissions* now provides a standardized primary language, if non-English, rather than "?" as before.
- Improved insurance data. The categories of the insurance column of *admissions* have been expanded to "Medicare", "Medicaid", "Private", "Self-pay", "No charge", and "Other". This change better aligns the field with other databases such as the National Inpatient Sample.

MIMIC-IV v2.2

MIMIC-IV v2.2 was released in January 2023. It added provider identifiers, imputed *hadm_id* for a number of rows in *emar*, and changed the subset of *subject_id* which are held out. Final row counts are available in the validation scripts published with the MIMIC Code Repository [6]. For clarity, after removal of the test set, the row counts are as follows:

- *patients*: 299,712 (was 315,460 in v2.0)
- *admissions*: 431,231 (was 454,324 in v2.0)
- *icustays*: 73,181 (was 76,943 in v2.0)

icu module

- *caregiver*
 - New table in v2.2. Contains one column: **caregiver_id**, a deidentified integer which uniquely represents a single caregiver or provider. These identifiers are sourced from the MetaVision ICU system. When present in a table, it indicates the user who documented the data into MetaVision. For example, the **caregiver_id** associated with a row indicating mechanical ventilation in the *procedureevents* table represents the user who documented the event, and not the provider who performed the procedure.
- *chartevents*, *datetimeevents*, *ingredientevents*, *inpuvents*, *outputevents*, *procedureevents*
 - Added the **caregiver_id** column. This column is a deidentified integer representing the care provider who documented the data for the given row.

hosp module

- *provider*
 - New table in v2.2. Contains one column: **provider_id**, a deidentified string which uniquely represents a single caregiver or provider. These identifiers are sourced from the hospital wide EHR system, and used in a variety of contexts across tables in the module.
- *admissions*
 - New column: **admit_provider_id**, a deidentified string representing the provider who admitted the patient.
- *emar*
 - New column: **enter_provider_id**, a deidentified string representing the provider who entered the medication administration information into the database.
 - Fixed a bug where a subset of *emar* rows (713,117, ~2.5%) did not have an **hadm_id** even though they were associated with a given hospitalization. These rows occur outside of the administratively documented admission and discharge times for a hospitalization, but are still considered as administered during that hospitalization in the raw data.
- *labevents*, *microbiologyevents*, *poe*, *prescriptions*
 - New column: **order_provider_id**, a deidentified string representing the provider who ordered the corresponding event (e.g. the lab test in the case of *labevents*, or the medication in the case of *prescriptions*).

MIMIC-IV v2.1

MIMIC-IV v2.1 was released on November 16, 2022. It removed a subset of **subject_id** which will be retained internally as a test set. Future data releases will exclude these patients.

- *patients* - Removed 15,748 **subject_id** from the table
- *admissions* - Removed 23,093 **hadm_id** from the table.

- icustays - Removed 3,762 *stay_id* from the table.
- Other tables will have rows removed to reflect the removal of the aforementioned *subject_id*, *hadm_id*, and *stay_id*. Final row counts are available in the validation scripts published with the MIMIC Code Repository [6].

MIMIC-IV v2.0

MIMIC-IV v2.0 was released on June 12, 2022. It focused on expanding the data elements available for patients within MIMIC-IV v1.0. Additional data available includes out-of-hospital date of death, information from the online medical record system (which includes height and weight), and more detail for continuous infusions in the ICU.

Major changes

- The *core* module has been removed to simplify the schema. The *admissions*, *patients*, and *transfers* tables are now in the *hosp* module.
- Neonates have been removed from the dataset. Neonatal data will be released in a separate project with data from the neonatal intensive care unit.

icu module

- *icustays*
 - Around 700 stays (~1%) have changed due to the changes in the *patients* table.
- *chartevents*, *d_items*
 - The problem list from MetaVision has been added. All problems are documented with the same *itemid* now present in *d_items*: 220001. There are just over 1,000 unique problems. Most documented problems are related to the care plan for the patient and documented during nurse shift changes (either 7am or 7pm). Less frequently, the ongoing issues are documented here.
- *ingredientevents*
 - This is a new table associated with *inpuvents*. Each intravenous administration tracked in *inpuvents* is associated with a set of ingredients. These ingredients include water content, caloric information, and so on. The goal of the *inpuvents* table is to support nutrition research and to provide a mechanism for estimating fluid input via summing all instances of the water ingredient. These ingredients have been separated from the *inpuvents* table to simplify analysis and reduce the size of *inpuvents*.
- *inpuvents*
 - Removed a single column which contained only null values: *cancelreason*.
- *procedureevents*
 - Removed columns which contained only null values: *totalamount*, *totalamountuom*, *cancelreason*, *comments_editedby*, *comments_canceledby*, *comments_date*, *secondaryordercategoryname*.

hosp module

- *admissions*

- Fixed an issue where hospitalizations were missing *edregtime* and *edouttime* when the patient was admitted via the ED (reported in [#1247](#), thanks [@MEladawi](#)).
- *patients*
 - *dod* is now populated with out-of-hospital mortality from state death records. For patients admitted to the ICU, this change has increased capture of date of death from 8,223 records to 23,844 (i.e. we now have out-of-hospital mortality for an additional 15,621 ICU patients).
 - The mechanism for determining patients included in MIMIC was changed. For the most part this has resulted in an improvement, particularly regarding the logic for merging patients who had distinct medical record numbers. As a result of this change, most tables have had a change in the data content. Approximately 1% of stays were affected.
- *transfers*
 - Fixed a bug where the *outtime* for ED stays with no associated *hadm_id* (i.e. an ED stay where the individual was not admitted to the hospital) was incorrect. This resulted in all *transfers* rows with a NULL *hadm_id* having an apparent stay of minutes or less. The *outtime* column has now been corrected.
- *labevents, d_labitems*
 - The *itemid* for *d_labitems* has been changed for 43 items. These are extremely infrequently documented and each *itemid* has fewer than 100 observations in *labevents*. The exact *itemid* are provided in the changelog file CHANGELOG.txt.
 - Errors were found in the current values of *loinc_code* (reported in [#938](#), thanks [@Mauvila](#)). In order to enable collaborative improvement, the *loinc_code* column has been removed, and will now be collaboratively developed in the [MIMIC Code Repository](#). Initial values will be sourced from the hospital system.
 - A number of labs which previously had the value in the comments field now have the value in the value field (reported in [#941](#), thanks [@Mauvila](#)). This change makes the *labevents* table more consistent with MIMIC-III, which had these values in the value field.
- *microbiologyevents*
 - New organisms, tests, specimens, and antibiotics have been added.
- *omr*
 - A new table has been added: *omr*. The source of this data is the Online Medical Record, and it contains miscellaneous information useful for understanding an individual's health. As of v2.0, the *omr* table has the following information: blood pressure, height, weight, body mass index, and Estimated Glomerular Filtration Rate (eGFR). These values are available from both inpatient and outpatient visits, and in many cases a "baseline" value from before a patient's hospitalization is available.
- *prescriptions*
 - The *formulary_drug_cd* table has been added back (was previously in MIMIC-III). This column has the same set of values as the *product_code* column of *emar_detail*.

MIMIC-IV v1.0

MIMIC-IV v1.0 was released March 16, 2021.

core

- *admissions*

- A number (~1000, <1%) of erroneous `hadm_id` have been removed.
- *patients*
 - `dod` is now populated using the patient's `deathtime` from their latest hospitalization (reported in #71, thanks @jinjinzhou).
 - At the moment, out-of-hospital mortality is **not** captured by ``dod``.
- *transfers*
 - Removed erroneous transfers included in the previous version.
 - `transfer_id` has been regenerated. `transfer_id` in MIMIC-IV v1.0 are **not compatible** with `transfer_id` from v0.4. We do not intend to change `transfer_id` when updating MIMIC-IV, but had to update it due to an error in its generation.
 - All `hadm_id` in transfers are also present in *admissions* and vice-versa (reported in #84, thanks @kokoko12305).

icu

- *icustays*
 - ICU stays were inappropriately assigned in the previous version due to an error in the preprocessing code. Previously, non-ICU ward transfers were included in the ICU stays, and certain ward stays were not treated as ICU stays (reported in #67, thanks @JHLiu7 and @stefanhgm). The assignment of `stay_id` has been regenerated.
 - The mapping between hospital transfers and ICU stays has been updated.
 - `stay_id` in MIMIC-IV v1.0 are **not compatible** with `stay_id` from v0.4. We do not intend to change `stay_id` when updating MIMIC-IV, but had to update it due to the error identified above.
- The change in *icustays* has re-assigned values to new `stay_id`, as a result all tables have had their content changed (due to a change in `stay_id`), but the structure is unchanged.

hosp

- *hospcsevents*
 - Data has been added for a number of previously excluded hospitalizations.
 - The table now has a `chartdate` column, containing the date associated with the code. Every row is associated with a date.
- *drgcodes*
 - Data has been added for a number of previously excluded hospitalizations.
 - Duplicate DRG codes have been removed from the table.
 - Descriptions have been updated using the latest dictionaries made available from mass.gov and HCUP.
- *diagnoses_icd, d_icd_diagnoses*
 - Data has been added for a number of previously excluded hospitalizations (reported in #27, thanks @yugangjia).
 - The `icd_code` column is now trimmed and stored as a VARCHAR, i.e. codes no longer contain trailing whitespaces ('850 ' -> '850').
 - Missing ICD codes have been added to the dictionary. All ICD codes in the *diagnoses_icd* table have an associated reference in *d_icd_diagnoses*.
- *labevents*

- The **comments** field has been updated, fixing a bug where comments longer than 4096 characters were truncated. Due to the deidentification, it's unlikely users will see much difference, as these comments will appear as ____.
- *procedures_icd*
 - Data has been added to *procedures_icd* for a number of previously excluded hospitalizations.
 - The table now has a **chartdate** column, containing the date associated with each billed procedure.
 - The **icd_code** column is now trimmed and stored as a VARCHAR, i.e. codes no longer contain trailing whitespaces ('850 ' -> '850').
 - Missing ICD codes have been added to the dictionary. All ICD codes in the *procedures_icd* table have an associated reference in *d_icd_procedures*.

v0.4

- *d_micro*
 - This table has been removed
- *microbiologyevents*
 - Added the column *spec_type_desc*, *test_name*, *org_name*, and *ab_name* columns
 - These columns contain the textual name of the organism/antibiotic/test/specimen
 - Added the *comments* column: this column contains information about the test, and in some cases (e.g. viral load tests), contains the result

v0.3

- Fixed a bug in the timing between *hosp* and *icu*

v0.2

- Updated demographics in the patient table
 - **anchor_year** -> **anchor_year_group**
 - **anchor_year_shifted** -> **anchor_year**
 - See the patients table in the MIMIC online documentation for detail on these columns
- *transfers*
 - Deleted the **los** column
- *emar*
 - **mar_id** -> **emar_id**
 - **emar_id** is now a composite of **subject_id** and **emar_seq**, and has form "subject_id-emar_seq"
 - **emar_seq** column - a monotonically increasing integer starting with the first eMAR administration
 - Added **poa_id** and **pharmacy_id** columns for linking to those tables
- *emar_detail*
 - **mar_id** -> **emar_id** (changed as above)
 - Deleted the **mar_detail_id** column
- *hpcsevents*
 - **ticket_id_seq** -> **seq_num**
- *labevents*

- Many previously NULL values are now populated - these were removed originally due to deidentification
- Added the **comments** column. This contains deidentified free-text comments with labs. PHI is replaced with three underscores (___). If an entire comment is ___, then the entire comment was scrubbed.
- *microbiologyevents*
 - **stay_id** column removed
 - **spec_id** -> **micro_specimen_id**
- Added the *poe* and *poe_detail* tables
 - Documentation of provider orders for various treatments and other aspects of patient management
- Added the *prescriptions* table
 - Documentation of medicine prescriptions via the provider order interface
- Added the *pharmacy* table
 - Detailed information regarding prescriptions provided by the pharmacy including formulary dose, route, frequency, dose, and so on.
- *inpatientevents*
 - Fixed an error in the calculation of the *amount* column
- *icustays*
 - Re-derived **stay_id** - the new **stay_id** are distinct from the previous version.