**Loading MIMIC-III into PostgreSQL**

Contents

[1 PhysioNet Notes 2](#_Toc533338080)

[2 PostgreSQL Installation 2](#_Toc533338081)

[3 Start/Stop PostgreSQL Server 2](#_Toc533338082)

[4 Run SQL Shell (psql) 2](#_Toc533338083)

[4.1 Create database 4](#_Toc533338084)

[4.2 Connect to the database 4](#_Toc533338085)

[4.3 Create the database tables 4](#_Toc533338086)

[4.4 Load the data 5](#_Toc533338087)

[4.5 Build indexes 7](#_Toc533338088)

[4.6 Test the build 8](#_Toc533338089)

[4.7 Verify everything loaded correctly 8](#_Toc533338090)

[4.8 Add Comments 10](#_Toc533338091)

[5 psql command line 10](#_Toc533338092)

[5.1 cmd.exe /c chcp 1252 10](#_Toc533338093)

[5.2 psql -d mimic -U postgres -W 10](#_Toc533338094)

[5.3 \? 10](#_Toc533338095)

[5.4 \d+ MIMICIII.ADMISSIONS; 13](#_Toc533338096)

[6 pgAdmin3 14](#_Toc533338097)

[7 Schema 14](#_Toc533338098)

# 

# PhysioNet Notes

From “Installing MIMIC-III in a local Postgres database on Windows”:  
<https://mimic.physionet.org/tutorials/install-mimic-locally-windows/>

*These are relatively brief instructions provided to ease installation of PostgreSQL with MIMIC on a Windows machine. If you feel there are key details missing, please raise an issue with your suggested improvements - we would love to incorporate them!*

*Note that before proceeding with this guide you will need to:*

*1.Download the MIMIC-III Clinical Database (*[*see here for details on gaining access*](https://mimic.physionet.org/gettingstarted/access/)*).*

*2.Extract the MIMIC-III Clinical Database as .csv files somewhere on your local computer.*

*3.Download the PostgreSQL scripts from* [*here*](https://github.com/MIT-LCP/mimic-code/tree/master/buildmimic/postgres) *- only the files which end in .sql are required*

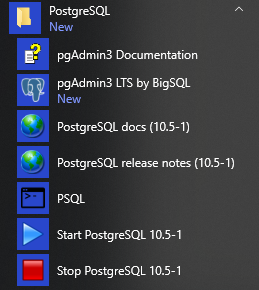
Indented text below in italics is from this same source.

# PostgreSQL Installation

See separate document, “PostgreSQL Installation on Windows.”

# Start/Stop PostgreSQL Server

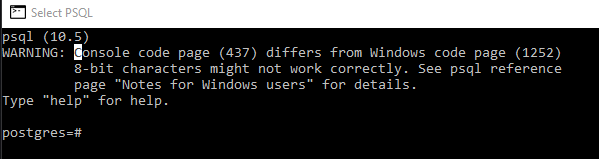
In Windows start menu …



Select **Start PostgreSQL** or **Stop PostgreSQL**.

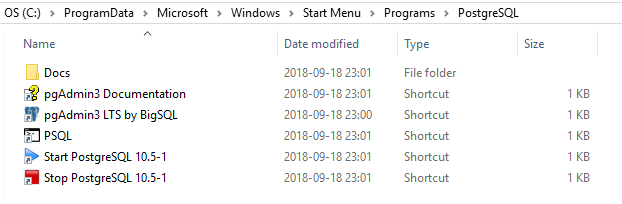
# Run SQL Shell (psql)



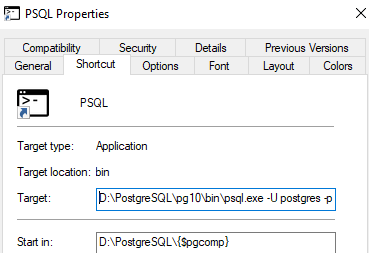


PostgreSQL: Warning: Console code page (437) differs from Windows code page (1252)  
<https://stackoverflow.com/questions/20794035/postgresql-warning-console-code-page-437-differs-from-windows-code-page-125>

From start menu, find the PSQL. Right click on PSQL, More, Open File Location



Right click on PSQL link, Properties.



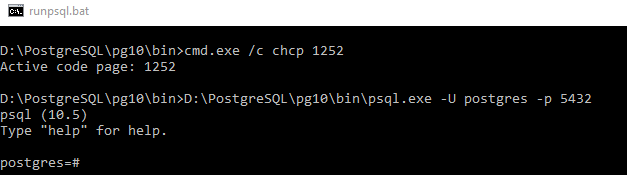
Create a **runpsql.bat** in D:\PostgreSQL\pg10\bin



Create desktop link to run PSQL with correct code page:



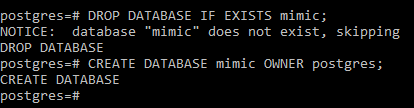
Right click | Run as administrator



## Create database

DROP DATABASE IF EXISTS mimic;

CREATE DATABASE mimic OWNER postgres;



This creates the database mimic, owned by user postgres. Of course you are welcome to change these values if you like - just note that any changes here will require further changes in the subsequent steps.

Next, connect to the mimic database.

## Connect to the database

\c mimic;



## Create the database tables

*Note that postgres uses the public schema by default. While it is personal preference, we recommend creating an independent schema to host the data. To do this, create the mimiciii schema:*

CREATE SCHEMA mimiciii;



*Then, inform postgres that it should by default use the mimiciii schema. You will need to do this every time you launch psql.*

set search\_path to mimiciii;

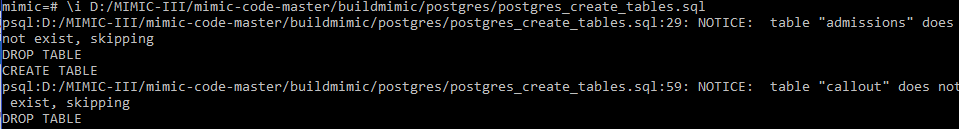


Build script from GitHub is stored here:

D:\MIMIC-III\mimic-code-master\buildmimic\postgres\postgres\_create\_tables.sql

Since not in the local directory, specify the path to the script (change to forward slashes):

\i D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_create\_tables.sql



*. . .*

*If you see a lot of “NOTICE: table does not exist” don’t worry, that’s normal. The script tries to delete the table before it creates it.*

*mimic=# \i D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_create\_tables.sql*

*psql:D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_create\_tables.sql:29: NOTICE: table "admissions" does not exist, skipping*

*DROP TABLE*

*CREATE TABLE*

*psql:D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_create\_tables.sql:59: NOTICE: table "callout" does not exist, skipping*

*DROP TABLE*

*CREATE TABLE*

*psql:D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_create\_tables.sql:93: NOTICE: table "caregivers" does not exist, skipping*

*DROP TABLE*

*CREATE TABLE*

*psql:D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_create\_tables.sql:108: NOTICE: table "chartevents" does not exist, skipping*

*DROP TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE TABLE*

*CREATE FUNCTION*

*CREATE TRIGGER*

*psql:D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_create\_tables.sql:189: NOTICE: table "cptevents" does not exist, skipping*

. . .

## Load the data

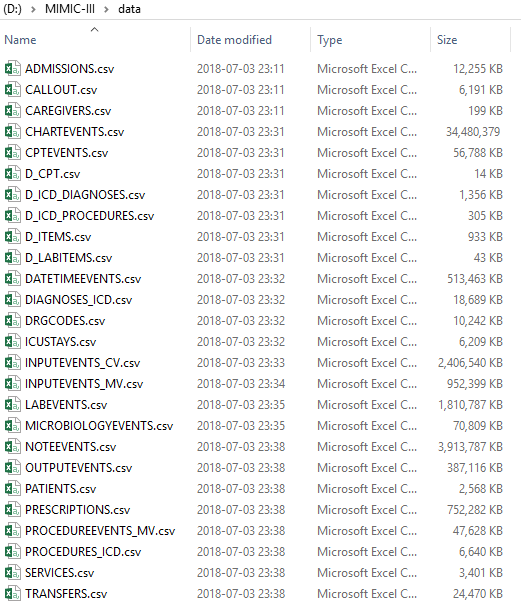
\set ON\_ERROR\_STOP 1



Identify folder with all the CSV files

\set mimic\_data\_dir 'D:/MIMIC-III/DATA'

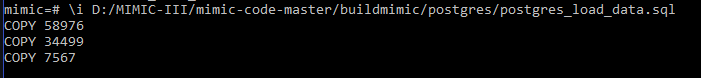




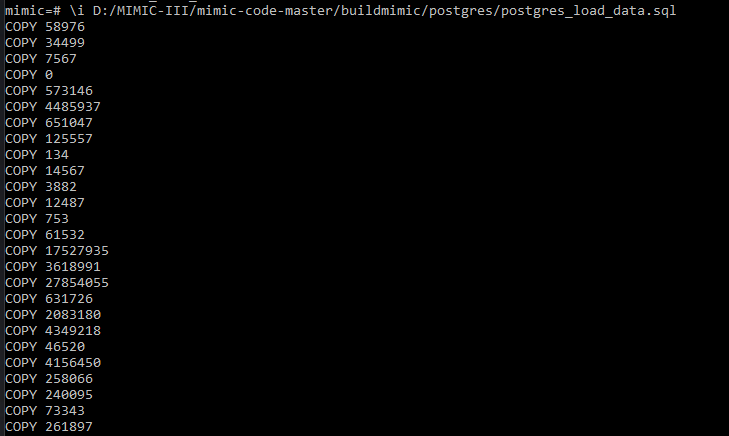
Run the load script:

\i D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_load\_data.sql

*Be aware that this can take some time, as there are almost 500 million rows in the entire database. Typical loading times are 4-6 hours. Usually the load will appear to pause after printing the following three lines:*



*This is expected - the fourth table is chartevents, and is by far the largest, and therefore takes the longest to load. Note also that eventually the fourth line will read COPY 0. This is expected: CHARTEVENTS acts as a “mapping” table to multiple sub-tables, and no data is actually stored within it, so postgres reports that 0 rows were inserted. This is expected behaviour for chartevents. When querying, do not query subtables chartevents\_1, chartevents\_2, etc, only query chartevents itself.*



mimic=# \i D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_load\_data.sql

COPY 58976

COPY 34499

COPY 7567

COPY 0

COPY 573146

COPY 4485937

COPY 651047

COPY 125557

COPY 134

COPY 14567

COPY 3882

COPY 12487

COPY 753

COPY 61532

COPY 17527935

COPY 3618991

COPY 27854055

COPY 631726

COPY 2083180

COPY 4349218

COPY 46520

COPY 4156450

COPY 258066

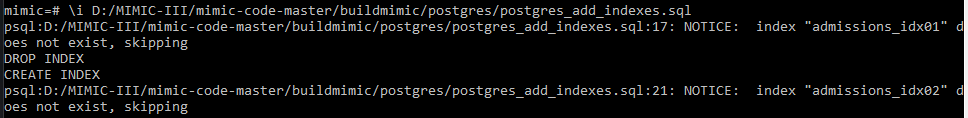
COPY 240095

COPY 73343

COPY 261897

## Build indexes

\i D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_add\_indexes.sql



mimic=# \i D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_add\_indexes.sql

psql:D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_add\_indexes.sql:17: NOTICE: index "admissions\_idx01" does not exist, skipping

DROP INDEX

CREATE INDEX

psql:D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_add\_indexes.sql:21: NOTICE: index "admissions\_idx02" does not exist, skipping

DROP INDEX

CREATE INDEX

. . .

## Test the build

Try the following simple query:

select

icustay\_id, intime, outtime

from icustays

limit 10;

mimic=# select

mimic-# icustay\_id, intime, outtime

mimic-# from icustays

mimic-# limit 10;

icustay\_id | intime | outtime

------------+---------------------+---------------------

280836 | 2198-02-14 23:27:38 | 2198-02-18 05:26:11

206613 | 2170-11-05 11:05:29 | 2170-11-08 17:46:57

220345 | 2128-06-24 15:05:20 | 2128-06-27 12:32:29

249196 | 2120-08-07 23:12:42 | 2120-08-10 00:39:04

210407 | 2186-12-25 21:08:04 | 2186-12-27 12:01:13

241507 | 2141-04-19 06:12:05 | 2141-04-20 17:52:11

254851 | 2114-06-28 22:28:44 | 2114-07-07 18:01:16

219649 | 2170-10-07 11:28:53 | 2170-10-14 14:38:07

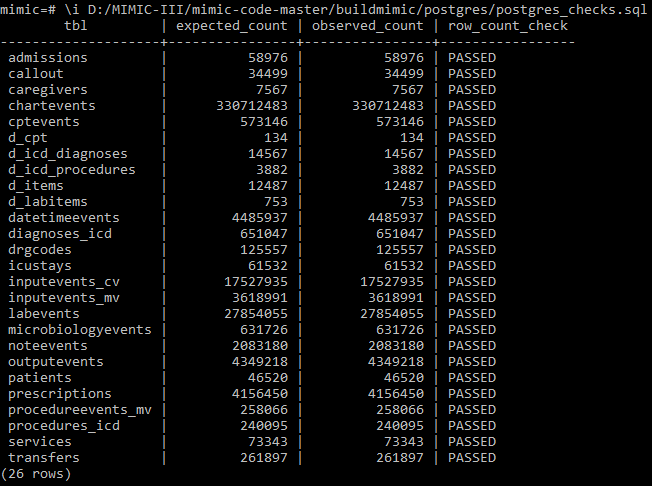
206327 | 2147-11-20 09:02:23 | 2147-11-21 17:08:52

272866 | 2132-10-21 21:11:46 | 2132-10-22 14:44:48

(10 rows)

## Verify everything loaded correctly

\i D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_checks.sql



mimic=# \i D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_checks.sql

tbl | expected\_count | observed\_count | row\_count\_check

--------------------+----------------+----------------+-----------------

admissions | 58976 | 58976 | PASSED

callout | 34499 | 34499 | PASSED

caregivers | 7567 | 7567 | PASSED

chartevents | 330712483 | 330712483 | PASSED

cptevents | 573146 | 573146 | PASSED

d\_cpt | 134 | 134 | PASSED

d\_icd\_diagnoses | 14567 | 14567 | PASSED

d\_icd\_procedures | 3882 | 3882 | PASSED

d\_items | 12487 | 12487 | PASSED

d\_labitems | 753 | 753 | PASSED

datetimeevents | 4485937 | 4485937 | PASSED

diagnoses\_icd | 651047 | 651047 | PASSED

drgcodes | 125557 | 125557 | PASSED

icustays | 61532 | 61532 | PASSED

inputevents\_cv | 17527935 | 17527935 | PASSED

inputevents\_mv | 3618991 | 3618991 | PASSED

labevents | 27854055 | 27854055 | PASSED

microbiologyevents | 631726 | 631726 | PASSED

noteevents | 2083180 | 2083180 | PASSED

outputevents | 4349218 | 4349218 | PASSED

patients | 46520 | 46520 | PASSED

prescriptions | 4156450 | 4156450 | PASSED

procedureevents\_mv | 258066 | 258066 | PASSED

procedures\_icd | 240095 | 240095 | PASSED

services | 73343 | 73343 | PASSED

transfers | 261897 | 261897 | PASSED

(26 rows)

## Add Comments

Add comments to all the tables/columns in MIMIC-III

\i D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_add\_comments.sql

mimic=# \i D:/MIMIC-III/mimic-code-master/buildmimic/postgres/postgres\_add\_comments.sql

COMMENT

COMMENT

. . .

# psql command line

Open command window

## cmd.exe /c chcp 1252

C:\Users\efg>cmd.exe /c chcp 1252

Active code page: 1252

## psql -d mimic -U postgres -W

C:\Users\efg>psql -d mimic -U postgres -W

Password for user postgres:

psql (10.5)

Type "help" for help.

mimic=# help

You are using psql, the command-line interface to PostgreSQL.

Type: \copyright for distribution terms

\h for help with SQL commands

\? for help with psql commands

\g or terminate with semicolon to execute query

\q to quit

## \?

mimic=# \?

General

\copyright show PostgreSQL usage and distribution terms

\crosstabview [COLUMNS] execute query and display results in crosstab

\errverbose show most recent error message at maximum verbosity

\g [FILE] or ; execute query (and send results to file or |pipe)

\gexec execute query, then execute each value in its result

\gset [PREFIX] execute query and store results in psql variables

\gx [FILE] as \g, but forces expanded output mode

\q quit psql

\watch [SEC] execute query every SEC seconds

Help

\? [commands] show help on backslash commands

\? options show help on psql command-line options

\? variables show help on special variables

\h [NAME] help on syntax of SQL commands, \* for all commands

Query Buffer

\e [FILE] [LINE] edit the query buffer (or file) with external editor

\ef [FUNCNAME [LINE]] edit function definition with external editor

\ev [VIEWNAME [LINE]] edit view definition with external editor

\p show the contents of the query buffer

\r reset (clear) the query buffer

\w FILE write query buffer to file

Input/Output

\copy ... perform SQL COPY with data stream to the client host

\echo [STRING] write string to standard output

\i FILE execute commands from file

\ir FILE as \i, but relative to location of current script

\o [FILE] send all query results to file or |pipe

\qecho [STRING] write string to query output stream (see \o)

Conditional

\if EXPR begin conditional block

\elif EXPR alternative within current conditional block

\else final alternative within current conditional block

\endif end conditional block

Informational

(options: S = show system objects, + = additional detail)

\d[S+] list tables, views, and sequences

\d[S+] NAME describe table, view, sequence, or index

\da[S] [PATTERN] list aggregates

\dA[+] [PATTERN] list access methods

\db[+] [PATTERN] list tablespaces

\dc[S+] [PATTERN] list conversions

\dC[+] [PATTERN] list casts

\dd[S] [PATTERN] show object descriptions not displayed elsewhere

\dD[S+] [PATTERN] list domains

\ddp [PATTERN] list default privileges

\dE[S+] [PATTERN] list foreign tables

\det[+] [PATTERN] list foreign tables

\des[+] [PATTERN] list foreign servers

\deu[+] [PATTERN] list user mappings

\dew[+] [PATTERN] list foreign-data wrappers

\df[antw][S+] [PATRN] list [only agg/normal/trigger/window] functions

\dF[+] [PATTERN] list text search configurations

\dFd[+] [PATTERN] list text search dictionaries

\dFp[+] [PATTERN] list text search parsers

\dFt[+] [PATTERN] list text search templates

\dg[S+] [PATTERN] list roles

\di[S+] [PATTERN] list indexes

\dl list large objects, same as \lo\_list

\dL[S+] [PATTERN] list procedural languages

\dm[S+] [PATTERN] list materialized views

\dn[S+] [PATTERN] list schemas

\do[S] [PATTERN] list operators

\dO[S+] [PATTERN] list collations

\dp [PATTERN] list table, view, and sequence access privileges

\drds [PATRN1 [PATRN2]] list per-database role settings

\dRp[+] [PATTERN] list replication publications

\dRs[+] [PATTERN] list replication subscriptions

\ds[S+] [PATTERN] list sequences

\dt[S+] [PATTERN] list tables

\dT[S+] [PATTERN] list data types

\du[S+] [PATTERN] list roles

\dv[S+] [PATTERN] list views

\dx[+] [PATTERN] list extensions

\dy [PATTERN] list event triggers

\l[+] [PATTERN] list databases

\sf[+] FUNCNAME show a function's definition

\sv[+] VIEWNAME show a view's definition

\z [PATTERN] same as \dp

Formatting

\a toggle between unaligned and aligned output mode

\C [STRING] set table title, or unset if none

\f [STRING] show or set field separator for unaligned query output

\H toggle HTML output mode (currently off)

\pset [NAME [VALUE]] set table output option

(NAME := {border|columns|expanded|fieldsep|fieldsep\_zero|

footer|format|linestyle|null|numericlocale|pager|

pager\_min\_lines|recordsep|recordsep\_zero|tableattr|title|

tuples\_only|unicode\_border\_linestyle|

unicode\_column\_linestyle|unicode\_header\_linestyle})

## \d+ MIMICIII.ADMISSIONS;

mimic=# \d+ MIMICIII.ADMISSIONS;

Table "mimiciii.admissions"

Column | Type | Collation | Nullable | Default | Storage | Stats target | Description

----------------------+--------------------------------+-----------+----------+---------+----------+--------------+---------------------------------------------

------------------------------

row\_id | integer | | not null | | plain | | Unique row identifier.

subject\_id | integer | | not null | | plain | | Foreign key. Identifies the patient.

hadm\_id | integer | | not null | | plain | | Primary key. Identifies the hospital stay.

admittime | timestamp(0) without time zone | | not null | | plain | | Time of admission to the hospital.

dischtime | timestamp(0) without time zone | | not null | | plain | | Time of discharge from the hospital.

deathtime | timestamp(0) without time zone | | | | plain | | Time of death.

admission\_type | character varying(50) | | not null | | extended | | Type of admission, for example emergency or

elective.

admission\_location | character varying(50) | | not null | | extended | | Admission location.

discharge\_location | character varying(50) | | not null | | extended | | Discharge location

insurance | character varying(255) | | not null | | extended | | Insurance type.

language | character varying(10) | | | | extended | | Language.

religion | character varying(50) | | | | extended | | Religon.

marital\_status | character varying(50) | | | | extended | | Marital status.

ethnicity | character varying(200) | | not null | | extended | | Ethnicity.

edregtime | timestamp(0) without time zone | | | | plain | |

edouttime | timestamp(0) without time zone | | | | plain | |

diagnosis | character varying(255) | | | | extended | | Diagnosis.

hospital\_expire\_flag | smallint | | | | plain | |

has\_chartevents\_data | smallint | | not null | | plain | | Hospital admission has at least one observat

ion in the CHARTEVENTS table.

Indexes:

"adm\_rowid\_pk" PRIMARY KEY, btree (row\_id)

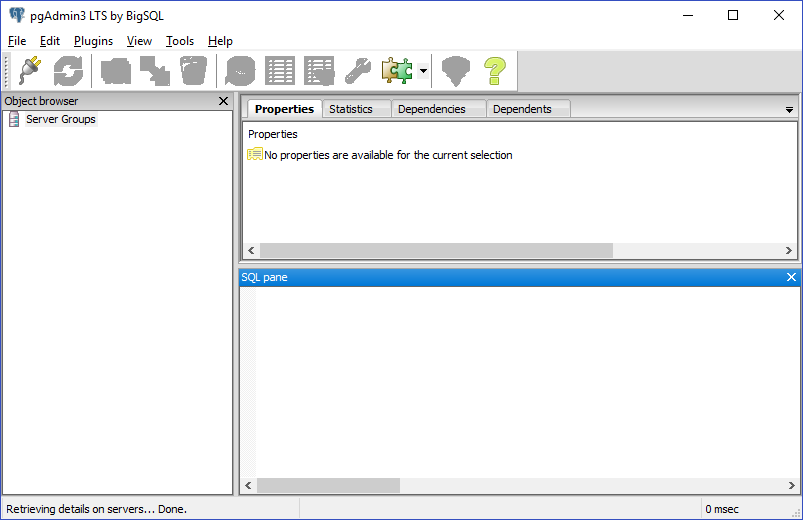
"adm\_hadm\_unique" UNIQUE CONSTRAINT, btree (hadm\_id)

"admissions\_idx01" btree (subject\_id)

"admissions\_idx02" btree (hadm\_id)

# pgAdmin3





# Schema

<https://mit-lcp.github.io/mimic-schema-spy/>

