hmda_tools Documentation

Release 0.1.2

Clinton Dreisbach, CFPB

May 07, 2017

Contents

hmda_tools makes working with HMDA data much easier.

"HMDA" refers to the Home Mortgage Disclosure Act, a law that requires financial institutions to maintain and annually disclose data about home purchases, home purchase pre-approvals, home improvement, and refinance applications. This data is made public and is available from the US Government at the FFIEC HMDA Products site.

CHAPTER 1

Creating Databases

To create a database to hold HMDA data, run the script `hmda_create_schemas` with a database URL as your argument. The database URL should be specified as it would be in SQLAlchemy.

Examples:

```
hmda_create_schemas sqlite://hmda.db
hmda_create_schemas mysql://root@localhost/hmda
hmda_create_schemas postgresql://peter:rabbit@10.0.0.34/hmda
```

You will need to install the Python libraries for your database separately. `hmda_tools` does not require any DB libraries, as it tries to stay agnostic.

You can also create these schemas programatically using *`hmda_tools.data.create_schemas()`*.

The created schema will look like the following (dependent on database):

```
CREATE TABLE "action_taken" (
  "id" int(11) NOT NULL,
  "action_taken" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "agency" (
  "id" int(11) NOT NULL,
  "agency_abbr" varchar(10) NOT NULL,
  "agency" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "cbsa" (
  "cbsa_code" int(11) NOT NULL,
  "parent_code" int(11) DEFAULT NULL,
  "name" varchar(255) NOT NULL,
  PRIMARY KEY ("cbsa_code")
);
CREATE TABLE "county" (
  "county_fips_code" int(11) NOT NULL,
  "state_fips_code" int(11) NOT NULL,
```

```
  "ansi_code" varchar(8) NOT NULL,
  "cbsa_code" int(11) DEFAULT NULL,
  "name" varchar(255) NOT NULL,
  "population" int(11) DEFAULT NULL,
  "housing_units" int(11) DEFAULT NULL,
  "land_area" bigint(20) DEFAULT NULL,
  "water_area" bigint(20) DEFAULT NULL,
  "latitude" varchar(20) DEFAULT NULL,
  "longitude" varchar(20) DEFAULT NULL,
  PRIMARY KEY ("county_fips_code","state_fips_code")
);
CREATE TABLE "denial_reason" (
  "id" int(11) NOT NULL,
  "denial_reason" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "edit_status" (
  "id" int(11) NOT NULL,
  "edit_status" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "ethnicity" (
  "id" int(11) NOT NULL,
  "ethnicity" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "hmda" (
  "year" int(11) NOT NULL,
  "respondent" varchar(10) DEFAULT NULL,
  "agency" int(11) DEFAULT NULL,
  "loan_type" int(11) DEFAULT NULL,
  "property_type" int(11) DEFAULT NULL,
  "loan_purpose" int(11) DEFAULT NULL,
  "occupancy" int(11) DEFAULT NULL,
  "loan_amount" int(11) DEFAULT NULL,
  "preapproval" int(11) DEFAULT NULL,
  "action_type" int(11) DEFAULT NULL,
  "msa_md" int(11) DEFAULT NULL,
  "state_code" int(11) DEFAULT NULL,
  "county_code" int(11) DEFAULT NULL,
  "census_tract_number" varchar(8) DEFAULT NULL,
  "applicant_ethnicity" int(11) DEFAULT NULL,
  "co_applicant_ethnicity" int(11) DEFAULT NULL,
  "applicant_race_1" int(11) DEFAULT NULL,
  "applicant_race_2" int(11) DEFAULT NULL,
  "applicant_race_3" int(11) DEFAULT NULL,
  "applicant_race_4" int(11) DEFAULT NULL,
  "applicant_race_5" int(11) DEFAULT NULL,
  "co_applicant_race_1" int(11) DEFAULT NULL,
  "co_applicant_race_2" int(11) DEFAULT NULL,
  "co_applicant_race_3" int(11) DEFAULT NULL,
  "co_applicant_race_4" int(11) DEFAULT NULL,
  "co_applicant_race_5" int(11) DEFAULT NULL,
  "applicant_sex" int(11) DEFAULT NULL,
  "co_applicant_sex" int(11) DEFAULT NULL,
  "applicant_income" int(11) DEFAULT NULL,
  "purchaser_type" int(11) DEFAULT NULL,
  "denial_reason_1" int(11) DEFAULT NULL,
```

```
  "denial_reason_2" int(11) DEFAULT NULL,
  "denial_reason_3" int(11) DEFAULT NULL,
  "rate_spread" varchar(10) DEFAULT NULL,
  "hoepa_status" int(11) DEFAULT NULL,
  "lien_status" int(11) DEFAULT NULL,
  "edit_status" int(11) DEFAULT NULL,
  "sequence_number" int(11) DEFAULT NULL,
  "population" int(11) DEFAULT NULL,
  "minority_population" float DEFAULT NULL,
  "hud_median_family_income" int(11) DEFAULT NULL,
  "tract_to_msa" float DEFAULT NULL,
  "number_of_owner_occupied_units" int(11) DEFAULT NULL,
  "number_of_family_units" int(11) DEFAULT NULL,
  "application_date_indicator" int(11) DEFAULT NULL,
  KEY "state_code" ("county_code"),
  KEY "ix_hmda_occupancy" ("occupancy"),
  KEY "ix_hmda_state_code" ("state_code"),
  KEY "ix_hmda_year" ("year"),
  KEY "ix_hmda_msa_md" ("msa_md"),
  KEY "ix_hmda_applicant_ethnicity" ("applicant_ethnicity"),
  KEY "ix_hmda_loan_amount" ("loan_amount"),
  KEY "ix_hmda_census_tract_number" ("census_tract_number")
);
CREATE TABLE "hoepa" (
  "id" int(11) NOT NULL,
  "hoepa" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "lien_status" (
  "id" int(11) NOT NULL,
  "lien_status" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "loan_purpose" (
  "id" int(11) NOT NULL,
  "loan_purpose" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "loan_type" (
  "id" int(11) NOT NULL,
  "loan_type" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "owner_occupancy" (
  "id" int(11) NOT NULL,
  "owner_occupancy" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "preapproval" (
  "id" int(11) NOT NULL,
  "preapproval" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "property_type" (
  "id" int(11) NOT NULL,
  "property_type" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
```

```
CREATE TABLE "purchaser_type" (
  "id" int(11) NOT NULL,
  "purchaser_type" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "race" (
  "id" int(11) NOT NULL,
  "race" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "sex" (
  "id" int(11) NOT NULL,
  "sex" varchar(255) NOT NULL,
  PRIMARY KEY ("id")
);
CREATE TABLE "state" (
  "fips_code" int(11) NOT NULL,
  "abbr" varchar(2) NOT NULL,
  PRIMARY KEY ("fips_code")
);
```

CHAPTER 2

Loading Data

HMDA Code Sheet

The HMDA data set contains a code sheet that explains what each of the numbers it uses for sex, race, ethnicity, regulating authority, and other lookup columns means. `hmda_tools` helps you load join tables for each of those columns into your database by using the script `hmda_load_code_sheet`.

You can also load these join tables programatically by calling *`hmda_tools.data.load_code_sheet()`*.

CBSA

Individual mortgages in the HMDA data set may be associated with a Metropolitan Statistical Area (MSA), using a code in the `msa_md` column. You can load a join table for this column using the `hmda_load_cbsa` script.

The `msa_md` column will only join against the `cbsa_code` column in the `cbsa` table. The `parent_code` column is there only for your use in determining the child areas of a larger combined statistical area.

This data is taken from the December 2009 Core-Based Statistical Area data from the US Census. HMDA data from before 2010 may not use this data.

You can also load these join tables programatically by calling *`hmda_tools.data.cbsa.load_cbsa()`*.

State and County Data

Individual mortgages in the HMDA data set have a county code and a state code. You can load a join table for these columns, using data from the 2010 county gazetteer from the US Census, by running the script *hmda_load_geo*. A crosswalk file will also be downloaded and used to populate the CBSA each county is in.

You can also load these join tables programatically by calling *`hmda_tools.data.geo.load_all()`*.

Actual HMDA Data

There is no current functionality to automatically load HMDA data. Given its size, it is easiest to load using your database's bulk import facilities.

To load the 2011 HMDA data into MySQL, run the following:

```
wget http://www.ffiec.gov/hmdarawdata/LAR/National/2011HMDALAR%20-%20National.zip -O␣
↪hmda11.zip
unzip -p hmda11.zip | sed 's/NA//g' | sed 's/ //g' > hmd11c.csv
mysql -e 'load data local infile 'hmda11c.csv' into table hmda fields terminated by ',
↪' lines terminated by "\n";'
```

CHAPTER 3

API

hmda_tools Package

hmda_tools Package

hmda_tools.__init__.**download_file**(*uri*)

unicode_csv Module

class hmda_tools.unicode_csv.**UTF8Recoder**(*f*, *encoding*)

Iterator that reads an encoded stream and reencodes the input to UTF-8

next()

class hmda_tools.unicode_csv.**UnicodeReader**(*f*, *dialect=<class csv.excel>*, *encoding='utf-8'*, ***kwds*)

A CSV reader which will iterate over lines in the CSV file "f", which is encoded in the given encoding.

next()

Subpackages

data Package

data Package

hmda_tools.data.**create_schemas**(*db_uri*)

hmda_tools.data.**csv_row_to_dict**(*headers*, *row*)

hmda_tools.data.**load_code_sheet**(*db_uri*)

hmda_tools.data.**load_hmda**(*db_uri*, *year*)

cbsa Module

hmda_tools.data.cbsa.**download_cbsa**()

hmda_tools.data.cbsa.**insert_cbsa_data**(*engine*, *table*, *cbsa_file*)

hmda_tools.data.cbsa.**load_cbsa**(*db_uri*)

geo Module

hmda_tools.data.geo.**csv_row_to_dict**(*headers*, *row*)

hmda_tools.data.geo.**download_crosswalk**()

hmda_tools.data.geo.**download_gazetteer**()

hmda_tools.data.geo.**insert_crosswalk_data**(*db_uri*, *filename*)

hmda_tools.data.geo.**insert_gaz_data**(*db_uri*, *gaz_file*)

hmda_tools.data.geo.**load_all**(*db_uri*)

hmda_tools.data.geo.**load_crosswalk**(*db_uri*)

hmda_tools.data.geo.**load_gazetteer**(*db_uri*)

hmda_tools.data.geo.**unzip_gazetteer**(*gaz_zip_file*)

schemas Module

hmda_tools.data.schemas.**CodeTable**(*name*, *metadata*)

hmda_tools.data.schemas.**action_taken**(*metadata*)

hmda_tools.data.schemas.**agency**(*metadata*)

hmda_tools.data.schemas.**cbsa**(*metadata*)

hmda_tools.data.schemas.**county**(*metadata*)

hmda_tools.data.schemas.**denial_reason**(*metadata*)

hmda_tools.data.schemas.**edit_status**(*metadata*)

hmda_tools.data.schemas.**ethnicity**(*metadata*)

hmda_tools.data.schemas.**hmda**(*metadata*)

hmda_tools.data.schemas.**hoepa**(*metadata*)

hmda_tools.data.schemas.**lien_status**(*metadata*)

hmda_tools.data.schemas.**loan_purpose**(*metadata*)

hmda_tools.data.schemas.**loan_type**(*metadata*)

hmda_tools.data.schemas.**owner_occupancy**(*metadata*)

hmda_tools.data.schemas.**preapproval**(*metadata*)

hmda_tools.data.schemas.**property_type**(*metadata*)

hmda_tools.data.schemas.**purchaser_type**(*metadata*)

hmda_tools.data.schemas.**race**(*metadata*)

hmda_tools.data.schemas.**sex**(*metadata*)

hmda_tools.data.schemas.**state**(*metadata*)

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

h

Index