

SANDIA REPORT

SAND2016-12693

Unlimited Release

December 2016

NMSBA-RS21

Mark K Kinnan, Richard Valerio, Tatiana P Flanagan, Donald N Shirah

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: <http://www.osti.gov/scitech>

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: <http://www.ntis.gov/search>



SAND2016-12693
Unlimited Release
December 2016

NMSBA–RS21

Mark K Kinnan
WMD Threats & Aerosol Science
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0782

Richard Valerio
Science Applications International Corporation
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1138

Tatiana P Flanagan
Resilience and Regulatory Effects
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1137

Donald N Shirah
Systems Research, Analysis, & Applications
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1138

Abstract

This report gives introductory guidance on the level of effort required to create a data warehouse for mining data. Numerous tutorials have been provided to demonstrate the process of downloading raw data, processing the raw data, and importing the data into a PostgreSQL database. Additional information and tutorial has been provided on setting up a Hadoop cluster for storing vast amounts of data. This report has been generated as a deliverable for a New Mexico Small Business Assistance (NMSBA) project.

ACKNOWLEDGMENTS

Technical Support: Tatiana Flanagan, Robert (Bobby) Jeffers, Alexander (Sasha) Outkin, Eric Vugrin, La Tonya Walker, Sarah Walsh, and Andrew (Andy) Wilson, are acknowledged for their enthusiastic, superior technical support of this project.

NMSBA Support: Juan (John) Martinez and Genaro Montoya are acknowledged for supporting this project.

Technical Review: The authors wish to thank Laura Sowko for reviewing this report before final submission.

CONTENTS

1	Introduction	9
2	Proposed Solution to Run Big Data For City Resiliency	11
2.1	Hadoop Map/Reduce and Apache Spark	11
2.2	The Hadoop Cluster	12
2.3	Server Costs.....	12
2.3.1	Physical.....	12
2.3.2	Cloud.....	12
2.4	On-going Running Costs.....	12
2.4.1	Physical.....	12
2.4.2	Cloud.....	12
3	Connection to AWS Server	13
4	Basic Terminal Commands	15
5	Basic PostgreSQL Commands	17
5.1	Information about the Database	17
6	NOAA Climate Data Tutorial	19
6.1	Weather Stations	19
6.2	Climate Data.....	20
6.3	Climate Data Transformation.....	21
7	FDIC Failed Bank List Tutorial	27
8	EPA FRS Tutorial	29
9	EPA Air Data Tutorial.....	33
9.1	EPA Monitoring Stations	33
9.2	EPA Hourly Criteria Gases (ozone).....	35
9.3	EPA Hourly Meterological (wind).....	36
10	BEA Tutorial.....	39
10.1	GDP by Metropolitan Area	39
10.2	Quarterly GDP by State.....	40
10.3	Annual GDP by State	42
11	CDC Chronic Disease Indicators	45
12	CDC Live Birth Data Tutorial.....	47
13	BLS Tutorial.....	57
13.1	Reference Tables	57
13.1.1	sm.area	57
13.1.2	sm.series.....	58
13.1.3	sm.industry.....	59
13.1.4	sm.data_type	60
13.1.5	sm.footnote	61
13.1.6	sm.supersector	62

13.1.7	sm.state	63
13.2	Data	64
13.3	Complex Data Query Examples	65
13.4	Write Query Data to File	65
14	NCES Tutorial	67
14.1	Institution Details	67
14.2	Admissions and Test Scores	69
14.2.1	Data from 2001	73
14.2.2	Data from 2002	79
14.2.3	Data from 2003	86
14.2.4	Data from 2004	92
14.2.5	Data from 2005	99
14.2.6	Data from 2006	106
15	CENSUS Data Tutorial	113
15.1	Data description	113
16	Questions for Data Mining	122
17	Data Integration and Warehousing	123
17.1	Recreating Data on AWS Server	123
17.2	Match Tables for Data Integration	123
17.3	Requirements Planning	123
18	Distribution	124

FIGURES

Figure 1.	Spark Hadoop Architecture	11
Figure 2.	Bitvise host key manager configuration screen	13
Figure 3.	Bitvise opening configuration screen for AWS server connection.	13

NOMENCLATURE

ACS	American Community Service
AWS	Amazon Web Services
BEA	Bureau of Economic Analysis
BLS	Bureau of Labor Statistics
CDC	Centers for Disease Control and Prevention
CPU	Central Processing Unit
DOE	Department of Energy
EPA	Environmental Protection Agency
FDIC	Integrated Postsecondary Education Data System
FRS	Facility Registry System
GDP	Gross Domestic Product
HDFS	Hadoop Distributed File System
IP	Internet Protocol
IPEDS	Integrated Postsecondary Education Data System
M&O	Maintenance and Operations
NCES	National Center for Education Statistics
NOAA	National Oceanic and Atmospheric Administration
PSQL	PostgreSQL
PUMS	Public Use Microdata Sample
SNL	Sandia National Laboratories
SQL	Structured Query Language

1 INTRODUCTION

City resiliency cuts across (e.g., physical, social, economic) domains, creating complex and interconnected systems with vast amount of data. To tackle this challenge, data must be efficiently collected across a large number of areas critical to resiliency from open sources, paid sources, and government sources. This information must be organized in a way so that the data can be quickly analyzed through databases, servers, models, etc. Users can glean insights from the information through the power of systems dynamics, modeling and computing, and machine learning. Using recent breakthroughs in cognitive science to create cutting edge visualizations, this data can be used to communicate information to current and prospective clients, both geospatially and through other mechanisms.

Big Data provides useful information for the user once the user has taken the time to work through the data's intricacies to formulate an answer to their question. While these instructions will enable a novice to build out a full cluster, database, and information service, it is highly recommended that only fully qualified personnel are utilized.

2 PROPOSED SOLUTION TO RUN BIG DATA FOR CITY RESILIENCY

Hadoop provides a framework to develop applications with the capability to run in a cluster environment allowing high performance statistical analysis for extremely large (Big Data) amounts of data.

2.1 Hadoop Map/Reduce and Apache Spark

Hadoop Map/Reduce allows for the fast processing of extremely large amounts of data, in the terabyte range, by breaking the data into logical groupings and distributing the problem across a Hadoop cluster. The cluster can consist of two computers to multiple thousands of computers.

Apache Spark accomplishes the same task as Hadoop Map/Reduce, the processing of large amounts of data. However, Apache Spark processes data much faster than Hadoop Map/Reduce due to processing mostly in memory.

While either Hadoop Map/Reduce or Apache Spark will work for processing data, the trade offs in budget and time have to be evaluated by the project manager.

1. Apache Spark is faster but requires systems with enough RAM to handle the queries efficiently.
2. Hadoop Map/Reduce may run slower but is able to work on lower cost hardware.

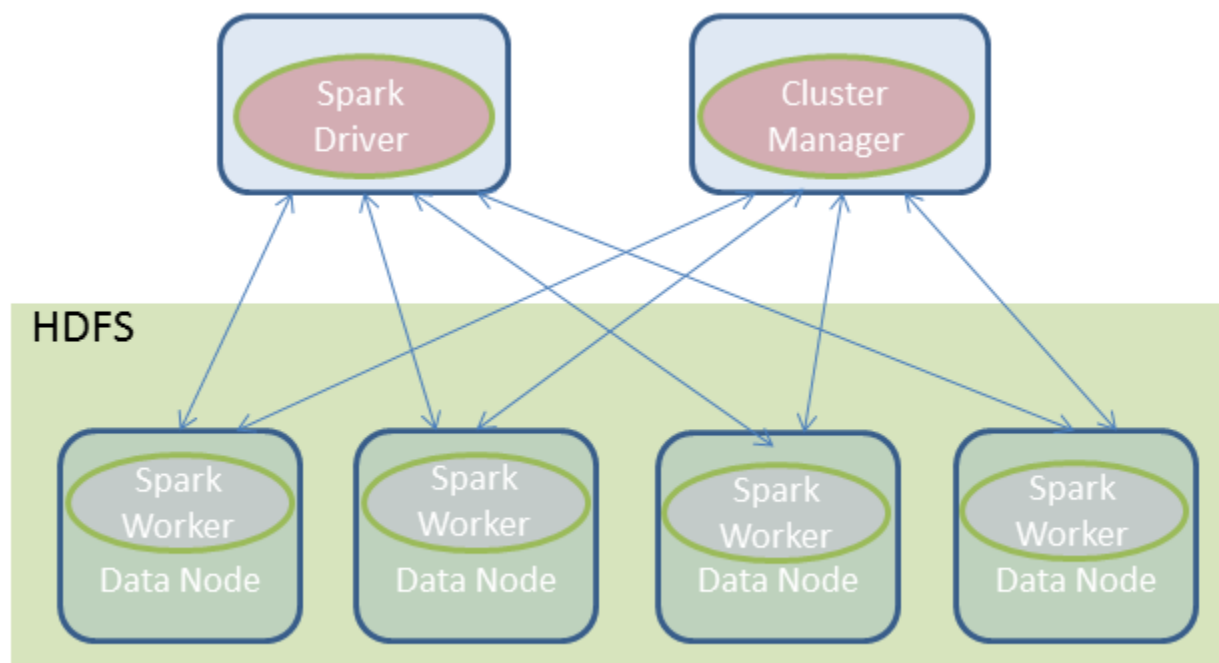


Figure 1. Spark Hadoop Architecture

2.2 The Hadoop Cluster

Building a Hadoop cluster is not a task for a novice due to the complexity of the operating system and the networking requirements. Several excellent tutorials are available on the Internet to guide the build process:

- https://www.tutorialspoint.com/hadoop/hadoop_multi_node_cluster.htm
- <http://hortonworks.com/hadoop-tutorial/hello-world-an-introduction-to-hadoop-hcatalog-hive-and-pig/>
- <http://gethue.com/hadoop-tutorial-how-to-create-a-real-hadoop-cluster-in-10-minutes/>

2.3 Server Costs

2.3.1 Physical

Dell PowerEdge R730xd Rack Server
12 TB of Storage
512 GB of Ram
Dual Processor
\$14,500 x four servers = \$58,000

2.3.2 Cloud

Amazon Elastic MapReduce is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data.

- Approximately \$10,000 per month
- Four instances 100% utilized Hadoop Distribution (MapR M7) Type (c3.8Large)

2.4 On-going Running Costs

2.4.1 Physical

Maintenance and Operations (M&O) at three hours per week per server covers patching, security reviews, and system backups.

2.4.2 Cloud

M&O at two hours per week per server covers patching, security reviews, and system backups.

3 CONNECTION TO AWS SERVER

There are many different software options to connect to an Amazon Web Services (AWS) server, such as Bitvise (<https://www.bitvise.com/>) or Putty (<http://www.putty.org/>). For demonstration here, Bitvise will be used for connecting to the server. The first task in Bitvise is to import the server key. This can be done by clicking the Import button shown in Figure 2 and selecting the key from your computer.

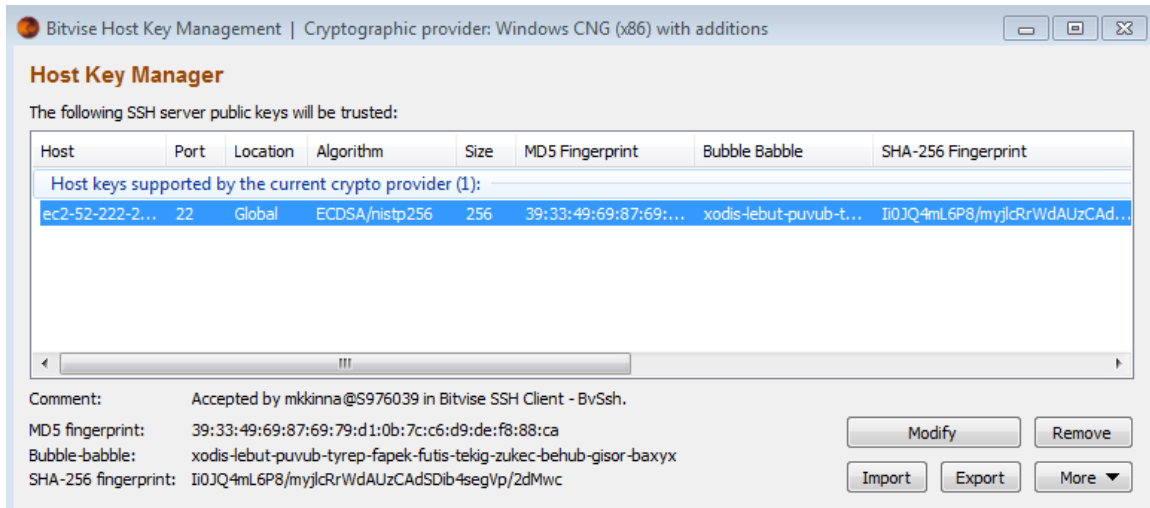


Figure 2. Bitvise host key manager configuration screen

Figure 3 illustrates the configuration settings for connecting to the AWS server. Note that the IP address of the server has been censored.

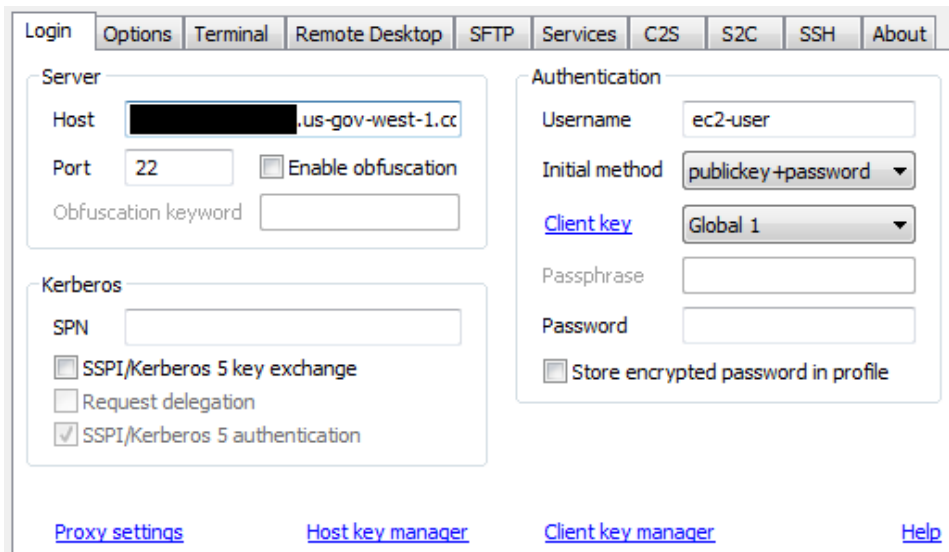


Figure 3. Bitvise opening configuration screen for AWS server connection

4 BASIC TERMINAL COMMANDS

Navigating the AWS server is straightforward and uses regular Linux terminal commands. AWS specific commands can be found at:

<https://aws.amazon.com/cli/>

Regular Linux terminal commands and tutorials for navigating the AWS server can be found at:

http://linuxcommand.org/lc3_learning_the_shell.php

5 BASIC POSTGRESQL COMMANDS

Currently, the data is stored in a PostgreSQL (<https://www.postgresql.org/>) database. Accessing the data can be achieved by using command line commands or via PGAdmin (<https://www.pgadmin.org/>). Tutorials on PostgreSQL can be found at the following links:

<https://www.postgresql.org/docs/9.5/static/tutorial.html>

<http://www.postgresguide.com/>

5.1 Information about the Database

Provided below is a query that can be used to get information about the tables and indexes currently in the PostgreSQL database.

```
-- This query returns a list of tables in the database
SELECT *, pg_size_pretty(total_bytes) AS total
      , pg_size_pretty(index_bytes) AS INDEX
      , pg_size_pretty(toast_bytes) AS toast
      , pg_size_pretty(table_bytes) AS TABLE
FROM (
  SELECT *, total_bytes-index_bytes-COALESCE(toast_bytes,0) AS table_bytes FROM (
    SELECT c.oid,nspname AS table_schema, relname AS TABLE_NAME
           , c.reltuples AS row_estimate
           , pg_total_relation_size(c.oid) AS total_bytes
           , pg_indexes_size(c.oid) AS index_bytes
           , pg_total_relation_size(reltoastrelid) AS toast_bytes
    FROM pg_class c
    LEFT JOIN pg_namespace n ON n.oid = c.relnamespace
    WHERE relkind = 'r' AND relname NOT LIKE 'pg_%'
  ) a
) a;
```


6 NOAA CLIMATE DATA TUTORIAL

6.1 Weather Stations

The National Oceanic and Atmospheric Administration (NOAA) provides a directory of all the weather stations in the world. To understand the variables of information associated with this dataset, it is recommended to review the readme.txt file. To recreate the NOAA directory of data on the AWS server, follow the tutorial below.

Server Terminal Commands

```
-- view readme.txt file for understanding the data (e.g., tavg, tmin, etc.)
-- ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/readme.txt

-- create the weather stations directory
sudo mkdir /nmsba/data-downloaded/ncdc-noaa-weather-stations

-- navigate to the newly created directory
cd /nmsba/data-downloaded/ncdc-noaa-weather-stations

-- get Weather Station Data:
sudo wget ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.txt

-- modify the ghcnd-stations.txt file in Microsoft Excel to create a "|" separated
file. See ghcnd-stations-cleaned.txt in /nmsba/data-downloaded/ncdc-noaa-weather-
stations

-- Log in to PostgreSQL:
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE climate.ncdc_noaa_weather_stations_2016100705;
-- create table in the climate schema for storing all the weather station data
CREATE TABLE climate.ncdc_noaa_weather_stations_2016100705
(
    station_id text NOT NULL,
    latitude numeric,
    longitude numeric,
    elevation numeric,
    state text,
    station_name text,
    gsn_flag text,
    hcn_crn_flag text,
    wmo_id integer,
    CONSTRAINT ncdc_noaa_weather_stations_pkey PRIMARY KEY (station_id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE climate.ncdc_noaa_weather_stations_2016100705
    OWNER TO postgres;

-- import weather station information into the newly created table
COPY climate.ncdc_noaa_weather_stations_2016100705 FROM '/nmsba/data-downloaded/ncdc-
noaa-weather-stations/ghcnd-stations-cleaned.txt' DELIMITER '|' QUOTE E'\b' NULL AS ''
CSV HEADER;

-- clean up table
```

```
vacuum (verbose, analyze) climate.ncdc_noaa_weather_stations_2016100705;

-- example query
SELECT * FROM climate.ncdc_noaa_weather_stations_2016100705 WHERE station_name LIKE
'%ALBUQUERQUE%' LIMIT 40;

-- exit PostgreSQL Command Line:
\q
```

6.2 Climate Data

NOAA provides daily climate summary information for each year for all weather stations in the world:

<https://www.ncdc.noaa.gov/data-access>

To understand the variables of information associated with this dataset, it is recommended to review the readme.txt file. To recreate the NOAA daily climate summary of data on the AWS server, follow the tutorial below.

Server Terminal Commands

```
-- FTP folder containing data:
-- ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/
-- see readme.txt file in folder for description of *.csv.gz files
-- ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/ghcn-daily-by_year-format.rtf

-- Create Directory
sudo /nmsba/mkdir ncdc-noaa-climate-daily-summaries

-- Navigate to new directory
cd /nmsba/ncdc-noaa-climate-daily-summaries

-- To download a file use the following command but substitute the XXXX for the year
you want to download for the NOAA Climate data
-- sudo wget ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/XXXX.csv.gz
-- So, to download 2014 download you would use the following:
sudo wget ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/2014.csv.gz

-- To unzip the file, you would use the following:
sudo gunzip 2014.csv.gz

-- Log in to PostgreSQL:
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- Connect to the raw database
\c raw

-- Create the SQL table
-- DROP TABLE climate.ncdc_noaa_daily_summaries_v01_20161006;
CREATE TABLE climate.ncdc_noaa_daily_summaries_v01_20161006
(
    station_id text,
    date integer,
    obs_type text,
    obs_value integer,
    unk1 text,
    unk2 text,
    unk3 text,
```

```

    obs_time text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE climate.ncdc_noaa_daily_summaries_v01_20161006
    OWNER TO postgres;

-- Import 2014.csv file data into database table
-- To import other years, substitute the year for other years downloaded
COPY climate.ncdc_noaa_daily_summaries_v01_20161006 FROM '/nmsba/data-downloaded/ncdc-
noaa-climate-daily-summaries/2014.csv' DELIMITER ',' CSV;

-- Create indexes
CREATE INDEX idx_ncdc_noaa_daily_summaries_stationid ON
climate.ncdc_noaa_daily_summaries_v01_20161006 USING btree(station_id);
CREATE INDEX idx_ncdc_noaa_daily_summaries_date ON
climate.ncdc_noaa_daily_summaries_v01_20161006 USING btree(date);
CREATE INDEX idx_ncdc_noaa_daily_summaries_obstype ON
climate.ncdc_noaa_daily_summaries_v01_20161006 USING btree(obs_type);
CREATE INDEX idx_ncdc_noaa_daily_summaries_stationidobstype ON
climate.ncdc_noaa_daily_summaries_v01_20161006 USING btree(station_id, obs_type);
CREATE INDEX idx_ncdc_noaa_daily_summaries_dateobstype ON
climate.ncdc_noaa_daily_summaries_v01_20161006 USING btree(date, obs_type);
CREATE INDEX idx_ncdc_noaa_daily_summaries_stationiddate ON
climate.ncdc_noaa_daily_summaries_v01_20161006 USING btree(station_id, date);

-- clean up table
vacuum (verbose, analyze) climate.ncdc_noaa_daily_summaries_v01_20161006;

-- Perform a Test query:
SELECT * FROM climate.ncdc_noaa_daily_summaries_v01_20161006 LIMIT 20;

-- Exit PSQL Command Line:
\q

```

6.3 Climate Data Transformation

One of the challenges associated with the NOAA daily summary data is that it is not convenient to query the data to get the desired results. Therefore, the data is transformed and inserted into a new table for better organization and efficiency. To recreate the transformation done on the server, follow the tutorial below.

PostgreSQL Terminal Commands

```

-- DROP TABLE climate.daily_weather;
CREATE TABLE climate.daily_weather
(
    station_id text,
    date integer,
    AWDR integer,
    AWND integer,
    DAPR integer,
    DASF integer,
    DATN integer,
    DATX integer,
    DWPR integer,
    EVAP integer,
    MDPR integer,
    MDSF integer,
    MDTN integer,
    MDTX integer,

```

```

MNPB integer,
MXPB integer,
PGTM integer,
PRCP integer,
PSUN integer,
SN31 integer,
SN32 integer,
SN33 integer,
SN35 integer,
SN36 integer,
SN51 integer,
SN52 integer,
SN53 integer,
SN55 integer,
SN56 integer,
SN57 integer,
SNOW integer,
SNWD integer,
SX31 integer,
SX32 integer,
SX33 integer,
SX35 integer,
SX36 integer,
SX51 integer,
SX52 integer,
SX53 integer,
SX55 integer,
SX56 integer,
SX57 integer,
TAVG integer,
THIC integer,
TMAX integer,
TMIN integer,
TOBS integer,
TSUN integer,
WDF2 integer,
WDF5 integer,
WDFG integer,
WDMV integer,
WESD integer,
WESF integer,
WSF2 integer,
WSF5 integer,
WSFG integer,
WSFI integer,
WT01 integer,
WT02 integer,
WT03 integer,
WT04 integer,
WT05 integer,
WT06 integer,
WT07 integer,
WT08 integer,
WT09 integer,
WT10 integer,
WT11 integer,
WT18 integer
)
WITH (
  OIDS=FALSE
);
ALTER TABLE climate.daily_weather
  OWNER TO postgres;

```

```

-- populate the new table with all the unique weather station data from the
ncdc_noaa_daily_summaries_v01_20161006 table and into the daily_weather table
do $$
    DECLARE
        arow climate.ncdc_noaa_daily_summaries_v01_20161006%ROWTYPE;
        iterator integer := 1;
    BEGIN
        FOR arow IN SELECT station_id, date FROM
climate.ncdc_noaa_daily_summaries_v01_20161006 GROUP BY station_id, date
            LOOP

                INSERT INTO climate.daily_weather (station_id, date) VALUES (arow.station_id,
arow.date);

                iterator := iterator + 1;
                RAISE NOTICE 'Iteration = % | Station ID = % | Date = %', iterator,
arow.station_id, arow.date;

            END LOOP;
    END;
$$;

-- create an index of the data
CREATE INDEX idx_daily_weather_stationid ON climate.daily_weather USING
btree(station_id);

CREATE INDEX idx_daily_weather_date ON climate.daily_weather USING btree(date);

-- update the records to input values
-- the following code snipped updates only the state of NM ... which is approximately
1.3M rows for years 2014,2015
do $$
    DECLARE
        arow climate.ncdc_noaa_daily_summaries_v01_20161006%ROWTYPE;
        iterator integer := 1;
    BEGIN
        FOR arow IN SELECT * FROM climate.ncdc_noaa_daily_summaries_v01_20161006 ds INNER
JOIN climate.ncdc_noaa_weather_stations_2016100705 ws ON ds.station_id = ws.station_id
WHERE ws.state = 'NM'
            LOOP

                UPDATE climate.daily_weather SET
                    awdr = (SELECT obs_value FROM climate.ncdc_noaa_daily_summaries_v01_20161006
WHERE station_id = arow.station_id AND date = arow.date AND obs_type = 'AWDR'),
                    awnd = (SELECT obs_value FROM climate.ncdc_noaa_daily_summaries_v01_20161006
WHERE station_id = arow.station_id AND date = arow.date AND obs_type = 'AWND'),
                    dapr = (SELECT obs_value FROM climate.ncdc_noaa_daily_summaries_v01_20161006
WHERE station_id = arow.station_id AND date = arow.date AND obs_type = 'DAPR'),
                    dasf = (SELECT obs_value FROM climate.ncdc_noaa_daily_summaries_v01_20161006
WHERE station_id = arow.station_id AND date = arow.date AND obs_type = 'DASF'),
                    datn = (SELECT obs_value FROM climate.ncdc_noaa_daily_summaries_v01_20161006
WHERE station_id = arow.station_id AND date = arow.date AND obs_type = 'DATN'),
                    datx = (SELECT obs_value FROM climate.ncdc_noaa_daily_summaries_v01_20161006
WHERE station_id = arow.station_id AND date = arow.date AND obs_type = 'DATX'),
                    dwpr = (SELECT obs_value FROM climate.ncdc_noaa_daily_summaries_v01_20161006
WHERE station_id = arow.station_id AND date = arow.date AND obs_type = 'DWPR'),
                    evap = (SELECT obs_value FROM climate.ncdc_noaa_daily_summaries_v01_20161006
WHERE station_id = arow.station_id AND date = arow.date AND obs_type = 'EVAP'),
                    mdpr = (SELECT obs_value FROM climate.ncdc_noaa_daily_summaries_v01_20161006
WHERE station_id = arow.station_id AND date = arow.date AND obs_type = 'MDPR'),
                    mdsf = (SELECT obs_value FROM climate.ncdc_noaa_daily_summaries_v01_20161006
WHERE station_id = arow.station_id AND date = arow.date AND obs_type = 'MDSF'),

```

[illegible]


```

    END LOOP;
  END;
$$;

-- perform a query to test view the data
SELECT ws.station_id, latitude, longitude, elevation, station_name, date, prcp, snow,
snwd, tmax, tmin FROM climate.daily_weather dw INNER JOIN
climate.ncdc_noaa_weather_stations_2016100705 ws ON dw.station_id = ws.station_id
WHERE tmax IS NOT NULL AND ws.station_id IN ('USC00290225', 'USC00290231',
'USW00023050') ORDER BY ws.station_id, date;

-- Exit PSQL Command Line:
\q

```

7 FDIC FAILED BANK LIST TUTORIAL

The Federal Deposit Insurance Corporation (FDIC) maintains a list of all the banks that have failed since October 1, 2000:

<https://www.fdic.gov/bank/individual/failed/banklist.html>

To recreate the data imported on the server, follow the tutorial below.

Server Terminal Commands

```
-- create the data directory
sudo /nmsba/mkdir fdic-failed-bank-list

-- navigate to the data directory
cd /nmsba/fdic-failed-bank-list

-- download the data to the data directory
sudo wget https://www.fdic.gov/bank/individual/failed/banklist.csv

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.fdic_bank_closures_v01_20161018;
CREATE TABLE economic.fdic_bank_closures_v01_20161018
(
    bank_name text,
    city character varying(64),
    state character varying(2),
    cert integer,
    acquiring_institution text,
    date_closed date,
    date_updated date,
    CONSTRAINT cert_pkey PRIMARY KEY (cert)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.fdic_bank_closures_v01_20161018
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.fdic_bank_closures_v01_20161018 FROM '/nmsba/data-downloaded/fdic-failed-bank-list/banklist.csv' DELIMITER ',' CSV HEADER;

-- create table indexes
CREATE INDEX idx_fdic_bank_closures_state ON economic.fdic_bank_closures_v01_20161018
    USING btree(state);
CREATE INDEX idx_fdic_bank_closures_city ON economic.fdic_bank_closures_v01_20161018
    USING btree(city);

-- clean up table and indexes
vacuum (verbose, full, analyze) economic.fdic_bank_closures_v01_20161018;

-- perform query to verify data looks correct
```

```
SELECT * FROM economic.fdic_bank_closures_v01_20161018 WHERE state = 'NM' ORDER BY
date_closed ASC;

-- view states with most bank closures
SELECT state, count(*) FROM economic.fdic_bank_closures_v01_20161018 GROUP BY state
ORDER BY count(*) DESC;

-- view cities with most bank closures
SELECT city, state, count(*) FROM economic.fdic_bank_closures_v01_20161018 GROUP BY
city, state ORDER BY count(*) DESC;

-- exit PostgreSQL terminal
\q
```

8 EPA FRS TUTORIAL

The EPA has a Facility Registry System (FRS) for facilities or sites subject to environmental regulation:

<https://www.epa.gov/enviro/epa-state-combined-csv-download-files>

To recreate the data import on the server, follow the tutorial below.

Server Terminal Commands

```
-- create data directory
sudo mkdir /nmsba/epa-facility-registry-service

-- navigate to the data directory
cd /nmsba/epa-facility-registry-service

-- download the data
sudo wget
https://www3.epa.gov/enviro/html/fii/downloads/state_files/national_combined.zip

-- unzip the file
sudo unzip national_combined.zip

-- change permissions of directory and files
sudo chmod -R 777 /nmsba/epa-facility-registry-service

-- remove header lines from each file of data to be imported
sudo sed -i '/REGISTRY_ID/d' NATIONAL_ALTERNATIVE_NAME_FILE.CSV
sudo sed -i '/REGISTRY_ID/d' NATIONAL_NAICS_FILE.CSV
sudo sed -i '/REGISTRY_ID/d' NATIONAL_SIC_FILE.CSV
sudo sed -i '/REGISTRY_ID/d' NATIONAL_FACILITY_FILE.CSV
sudo sed -i '/REGISTRY_ID/d' NATIONAL_SUPP_INTEREST_FILE.CSV

-- login to PostgreSQL command line terminal
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to raw database
\c raw

-- DROP TABLE economic.epa_frs_alternative_name_v01_20161018;
CREATE TABLE economic.epa_frs_alternative_name_v01_20161018
(
    registry_id bigint,
    pgm_sys_acrnm text,
    pgm_sys_id text,
    alternative_name text,
    alternative_name_type text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.epa_frs_alternative_name_v01_20161018
    OWNER TO postgres;

-- DROP TABLE economic.epa_frs_naics_v01_20161018;
CREATE TABLE economic.epa_frs_naics_v01_20161018
(
    registry_id bigint,
```

```

    pgm_sys_acrnm text,
    pgm_sys_id text,
    interest_type text,
    naics_code integer,
    primary_indicator text,
    code_description text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.epa_frs_naics_v01_20161018
    OWNER TO postgres;

-- DROP TABLE economic.epa_frs_sic_v01_20161018;
CREATE TABLE economic.epa_frs_sic_v01_20161018
(
    registry_id bigint,
    pgm_sys_acrnm text,
    pgm_sys_id text,
    interest_type text,
    sic_code integer,
    primary_indicator text,
    code_description text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.epa_frs_sic_v01_20161018
    OWNER TO postgres;

-- DROP TABLE economic.epa_frs_facility_v01_20161018;
CREATE TABLE economic.epa_frs_facility_v01_20161018
(
    frs_facility_detail_report_url text,
    registry_id bigint,
    primary_name text,
    location_address text,
    supplemental_location text,
    city_name text,
    county_name text,
    fips_code text,
    state_code text,
    state_name text,
    country_name text,
    postal_code text,
    federal_facility_code text,
    federal_agency_name text,
    tribal_land_code text,
    tribal_land_name text,
    congressional_dist_num text,
    census_block_code text,
    huc_code text,
    epa_region_code text,
    site_type_name text,
    location_description text,
    create_date text,
    update_date text,
    us_mexico_border_ind text,
    latitude83 text,
    longitude83 text,
    conveyor text,
    collect_desc text,
    accuracy_value text,

```

```

    ref_point_desc text,
    hdatum_desc text,
    source_desc text,
    CONSTRAINT facility_registry_id_pkey PRIMARY KEY (registry_id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.epa_frs_facility_v01_20161018
    OWNER TO postgres;

-- DROP TABLE economic.epa_frs_supp_interest_v01_20161018;
CREATE TABLE economic.epa_frs_supp_interest_v01_20161018
(
    registry_id bigint,
    pgm_sys_acrnm text,
    pgm_sys_id text,
    interest_type text,
    sup_pgm_sys_acrnm text,
    sup_pgm_sys_id text,
    sup_interest_type text,
    start_date text,
    start_date_qualifier text,
    end_date text,
    end_date_qualifier text,
    source_of_data text,
    last_reported_date text,
    reported_sup_interest_type text,
    create_date text,
    update_date text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.epa_frs_supp_interest_v01_20161018
    OWNER TO postgres;

-- import data into tables
COPY economic.epa_frs_alternative_name_v01_20161018 FROM '/nmsba/data-downloaded/epa-
facility-registry-service/NATIONAL_ALTERNATIVE_NAME_FILE.CSV' DELIMITER ',' CSV;

COPY economic.epa_frs_naics_v01_20161018 FROM '/nmsba/data-downloaded/epa-facility-
registry-service/NATIONAL_NAICS_FILE.CSV' DELIMITER ',' CSV;

COPY economic.epa_frs_sic_v01_20161018 FROM '/nmsba/data-downloaded/epa-facility-
registry-service/NATIONAL_SIC_FILE.CSV' DELIMITER ',' CSV;

COPY economic.epa_frs_facility_v01_20161018 FROM '/nmsba/data-downloaded/epa-facility-
registry-service/NATIONAL_FACILITY_FILE.CSV' DELIMITER ',' CSV;

COPY economic.epa_frs_supp_interest_v01_20161018 FROM '/nmsba/data-downloaded/epa-
facility-registry-service/NATIONAL_SUPP_INTEREST_FILE.CSV' DELIMITER ',' CSV;

-- create indexes
CREATE INDEX idx_epa_frs_facility_registryid ON economic.epa_frs_facility_v01_20161018
USING btree(registry_id);

-- clean up tables and indexes
vacuum (verbose, analyze) economic.epa_frs_alternative_name_v01_20161018;
vacuum (verbose, analyze) economic.epa_frs_naics_v01_20161018;
vacuum (verbose, analyze) economic.epa_frs_sic_v01_20161018;
vacuum (verbose, analyze) economic.epa_frs_facility_v01_20161018;
vacuum (verbose, analyze) economic.epa_frs_supp_interest_v01_20161018;

```

```
-- example query to test data integration
SELECT f.registry_id, f.primary_name, f.location_address, f.city_name, f.state_code,
n.naics_code, n.code_description FROM economic.epa_frs_facility_v01_20161018 f
LEFT JOIN economic.epa_frs_naics_v01_20161018 n ON f.registry_id = n.registry_id
WHERE f.registry_id = 110000701296;

-- exit PostgreSQL terminal
\q
```


9 EPA AIR DATA TUTORIAL

EPA maintains archival file downloads for climate data:

http://aqsdrl.epa.gov/aqsweb/aqstmp/airdata/download_files.html

To recreate the data imported on the server, follow the tutorial below.

9.1 EPA Monitoring Stations

Server Terminal Commands

```
-- create the data directory
sudo mkdir /nmsba/epa-airdata

-- change permissions of directory and files
sudo chmod -R 777 /nmsba

-- navigate to the data directory
cd /nmsba/epa-airdata

-- download the data to the data directory
sudo wget http://aqsdrl.epa.gov/aqsweb/aqstmp/airdata/aqs_sites.zip

sudo wget http://aqsdrl.epa.gov/aqsweb/aqstmp/airdata/aqs_monitors.zip

-- unzip the files
sudo unzip aqs_sites.zip

sudo unzip aqs_monitors.zip

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE climate.epa_airdata_aqs_monitors;
CREATE TABLE climate.epa_airdata_aqs_monitors
(
    state_code text,
    county_code integer,
    site_number integer,
    parameter_code integer,
    parameter_name text,
    poc text,
    latitude text,
    longitude text,
    datum text,
    first_year_of_data text,
    last_sample_date text,
    monitor_type text,
    networks text,
    reporting_agency text,
    pqao text,
    collecting_agency text,
    exclusions text,
    monitoring_objective text,
    last_method_code integer,
    last_method text,
```

```

naaqs_primary_monitor text,
local_site_name text,
address text,
state_name text,
county_name text,
city_name text,
cbsa_name text,
tribe_name text,
tribe_name_2 text, -- for some reason there are 2 fields for tribe names and the CSV
file is broken so this extra column was added
extraction_date date
)
WITH (
    OIDS=FALSE
);
ALTER TABLE climate.epa_airdata_aqs_monitors
    OWNER TO postgres;

-- DROP TABLE climate.epa_airdata_aqs_sites;
CREATE TABLE climate.epa_airdata_aqs_sites
(
    state_code text,
    county_code integer,
    site_number integer,
    latitude text,
    longitude text,
    datum text,
    elevation decimal,
    land_use text,
    location_setting text,
    site_established_date date,
    site_closed_date text,
    met_site_state_code text,
    met_site_county_code text,
    met_site_site_number text,
    met_site_type text,
    met_site_distance text,
    met_site_direction text,
    gmt_offset integer,
    owning_agency text,
    local_site_name text,
    address text,
    zip_code text,
    state_name text,
    county_name text,
    city_name text,
    cbsa_name text,
    tribe_name text,
    extraction_date date
)
WITH (
    OIDS=FALSE
);
ALTER TABLE climate.epa_airdata_aqs_sites
    OWNER TO postgres;

-- import data into tables
COPY climate.epa_airdata_aqs_monitors FROM '/nmsba/data-downloaded/epa-
airdata/aqs_monitors.csv' DELIMITER ',' CSV HEADER;

COPY climate.epa_airdata_aqs_sites FROM '/nmsba/data-downloaded/epa-
airdata/aqs_sites.csv' DELIMITER ',' CSV HEADER;

```

```
-- clean up tables and indexes
vacuum (verbose, analyze) climate.epa_airdata_aqs_sites;

-- example query to test data integration
SELECT state_code, county_code, site_number, city_name, address FROM
climate.epa_airdata_aqs_sites WHERE state_code::integer = 35 AND county_code::integer
= 1 LIMIT 50;

-- exit PostgreSQL terminal
\q
```

9.2 EPA Hourly Criteria Gases (ozone)

The tutorial is performed for only one year, 2015. To include all years repeat the tutorial below and change the year. For the PostgreSQL Terminal Commands section, skip the CREATE TABLE command for all other years after completing the tutorial for year 2015.

Server Terminal Commands

```
-- navigate to the data directory
cd /nmsba/data-downloaded/epa-airdata

-- create the data directories
sudo mkdir hourly
sudo mkdir hourly/criteria
sudo mkdir hourly/criteria/ozone

-- change permissions of directory and files
sudo chmod -R 777 /nmsba

-- navigate to the data directory
cd hourly/criteria/ozone

-- download the data to the data directory
sudo wget http://aqsdr1.epa.gov/aqsweb/aqstmp/airdata/hourly_44201_2015.zip

-- unzip the files
sudo unzip hourly_44201_2015.zip

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE climate.epa_airdata_hourly_criteria_gases;
CREATE TABLE climate.epa_airdata_hourly_criteria_gases
(
    state_code integer,
    county_code integer,
    site_num integer,
    parameter_code integer,
    poc integer,
    latitude decimal,
    longitude decimal,
    datum text,
    parameter_name text,
    date_local text,
    time_local text,
    date_gmt text,
    time_gmt text,
```

```

    sample_measurement decimal,
    units_of_measure text,
    mdl decimal,
    uncertainty text,
    qualifier text,
    method_type text,
    method_code integer,
    method_name text,
    state_name text,
    county_name text,
    date_of_last_change text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE climate.epa_airdata_hourly_criteria_gases
    OWNER TO postgres;

-- import data from file into newly created database table
COPY climate.epa_airdata_hourly_criteria_gases FROM '/nmsba/data-downloaded/epa-
airdata/hourly/criteria/ozone/hourly_44201_2015.csv' DELIMITER ',' CSV HEADER;

-- create indexes
CREATE INDEX idx_epa_airdata_hourly_criteria_gases_statecode ON
climate.epa_airdata_hourly_criteria_gases USING btree(state_code);

CREATE INDEX idx_epa_airdata_hourly_criteria_gases_countycode ON
climate.epa_airdata_hourly_criteria_gases USING btree(county_code);

CREATE INDEX idx_epa_airdata_hourly_criteria_gases_sitnum ON
climate.epa_airdata_hourly_criteria_gases USING btree(site_num);

CREATE INDEX idx_epa_airdata_hourly_criteria_gases_datetime ON
climate.epa_airdata_hourly_criteria_gases USING btree(date_local, time_local);

CREATE INDEX idx_epa_airdata_hourly_criteria_gases_datetime2 ON
climate.epa_airdata_hourly_criteria_gases ((date_local || ' ' || time_local));

-- clean up tables and indexes
vacuum (verbose, analyze) climate.epa_airdata_hourly_criteria_gases;

-- example query to test data integration
SELECT date_local, time_local, sample_measurement, units_of_measure FROM
climate.epa_airdata_hourly_criteria_gases WHERE state_code = 35 AND county_code = 1
AND site_num = 23 AND date_local IN ('2015-01-01','2015-01-02','2015-01-03');

-- exit PostgreSQL terminal
\q

```

9.3 EPA Hourly Meterological (wind)

The tutorial is performed for only one year, 2015. To include all years repeat the tutorial below and change the year. For the PostgreSQL Terminal Commands section, skip the CREATE TABLE command for all other years after completing the tutorial for year 2015.

Server Terminal Commands

```

-- navigate to the data directory
cd /nmsba/data-downloaded/epa-airdata/hourly

-- create the data directories
sudo mkdir meterological

```

```

sudo mkdir meterological/wind

-- change permissions of directory and files
sudo chmod -R 777 /nmsba

-- navigate to the data directory
cd meterological/wind

-- download the data to the data directory
sudo wget http://aqsdrl.epa.gov/aqsweb/aqstmp/airdata/hourly_WIND_2015.zip

-- unzip the files
sudo unzip hourly_WIND_2015.zip

-- login to the PostgreSQL command line
sudo -u postgres psql postgres

```

PostgreSQL Terminal Commands

```

-- connect to the raw database
\c raw

-- DROP TABLE climate.epa_airdata_hourly_meterological_wind;
CREATE TABLE climate.epa_airdata_hourly_meterological_wind
(
    state_code integer,
    county_code integer,
    site_num integer,
    parameter_code integer,
    poc integer,
    latitude decimal,
    longitude decimal,
    datum text,
    parameter_name text,
    date_local text,
    time_local text,
    date_gmt date,
    time_gmt text,
    sample_measurement decimal,
    units_of_measure text,
    mdl decimal,
    uncertainty text,
    qualifier text,
    method_type text,
    method_code integer,
    method_name text,
    state_name text,
    county_name text,
    date_of_last_change text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE climate.epa_airdata_hourly_meterological_wind
    OWNER TO postgres;

-- import data from file into newly created database table
COPY climate.epa_airdata_hourly_meterological_wind FROM '/nmsba/data-downloaded/epa-airdata/hourly/meterological/wind/hourly_WIND_2015.csv' DELIMITER ',' CSV HEADER;

-- create indexes
CREATE INDEX idx_epa_airdata_hourly_meterological_wind_statecode ON
climate.epa_airdata_hourly_meterological_wind USING btree(state_code);

```

```

CREATE INDEX idx_epa_airdata_hourly_meterological_wind_countycode ON
climate.epa_airdata_hourly_meterological_wind USING btree(county_code);

CREATE INDEX idx_epa_airdata_hourly_meterological_wind_sitnum ON
climate.epa_airdata_hourly_meterological_wind USING btree(site_num);

CREATE INDEX idx_epa_airdata_hourly_meterological_wind_datetime ON
climate.epa_airdata_hourly_meterological_wind USING btree(date_local, time_local);

CREATE INDEX idx_epa_airdata_hourly_meterological_wind_datetime2 ON
climate.epa_airdata_hourly_meterological_wind ((date_local || time_local));

-- clean up tables and indexes
vacuum (verbose, analyze) climate.epa_airdata_hourly_meterological_wind;

-- example query to test data integration
SELECT date_local, time_local, sample_measurement, units_of_measure FROM
climate.epa_airdata_hourly_meterological_wind WHERE state_code = 35 AND county_code =
1 AND site_num = 23 AND date_local IN ('2015-01-01','2015-01-02','2015-01-03');

-- exit PostgreSQL terminal
\q

```

10 BEA TUTORIAL

The Bureau of Economic Analysis (BEA) maintains archival file downloads for economic data:

<http://www.bea.gov/regional/downloadzip.cfm>

To recreate the data imported on the server, follow the tutorial below.

10.1 GDP by Metropolitan Area

It is necessary to download the file using a browser. Then the file needs to be transferred to the following location on the server: `/nmsba/data-download/bea-gdp-metro-area`

Server Terminal Commands

```
-- create the data directory
sudo mkdir /nmsba/data-download/bea-gdp-metro-area

-- copy the file to the directory after creating the directory

-- change permissions of directory and files
sudo chmod -R 777 /nmsba

-- clean the file for importing
sudo sed -i '/See the included footnote file/d' allgmp.csv

sudo sed -i '/Industry detail is based on the/d' allgmp.csv

sudo sed -i '/Regional Product Division/d' allgmp.csv

sudo sed -i '/GeoFIPS/d' allgmp.csv

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.bea_gdp_metro_areas_v01_20161027;
CREATE TABLE economic.bea_gdp_metro_areas_v01_20161027
(
    geofips integer,
    geoname text,
    region text,
    componentid text,
    componentname text,
    industryid text,
    industryclassification text,
    description text,
    year2001 text,
    year2002 text,
    year2003 text,
    year2004 text,
    year2005 text,
    year2006 text,
    year2007 text,
```

```

    year2008 text,
    year2009 text,
    year2010 text,
    year2011 text,
    year2012 text,
    year2013 text,
    year2014 text,
    year2015 text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.bea_gdp_metro_areas_v01_20161027
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.bea_gdp_metro_areas_v01_20161027 FROM '/nmsba/data-downloaded/bea-gdp-
metro-area/allgmp.csv' DELIMITER ',' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.bea_gdp_metro_areas_v01_20161027;

-- example queries to test data integration
SELECT geoname, componentname, industryid, industryclassification, description,
year2013, year2014, year2015 FROM economic.bea_gdp_metro_areas_v01_20161027 WHERE
(lower(geoname) LIKE lower('%albuquerque%') OR lower(geoname) LIKE lower('%santa
fe%')) AND industryid = '11';

SELECT * FROM economic.bea_gdp_metro_areas_v01_20161027 WHERE (lower(geoname) LIKE
lower('%albuquerque%') OR lower(geoname) LIKE lower('%santa fe%')) AND industryid =
'11';

-- exit PostgreSQL terminal
\q

```

10.2 Quarterly GDP by State

It is necessary to download the file using a browser. Then the file needs to be transferred to the following location on the server: `/nmsba/data-download/bea-gdp-state-quarterly`

Server Terminal Commands

```

-- create the data directory
sudo mkdir /nmsba/data-download/bea-gdp-state-quarterly

-- copy the file to the directory after creating the directory

-- change permissions of directory and files
sudo chmod -R 777 /nmsba

-- clean the file for importing
sudo sed -i '/See the included footnote file/d' qgsp_all.csv

sudo sed -i '/Industry detail is based on the/d' qgsp_all.csv

sudo sed -i '/Regional Product Division/d' qgsp_all.csv

sudo sed -i '/GeoFIPS/d' qgsp_all.csv

-- view the last few lines of the file
tail qgsp_all.csv

```



```
-- if there are extra lines of text that need to be removed, execute the following
command
sudo sed -i '$d' qgsp_all.csv

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.bea_gdp_state_quarter_v01_20161027;
CREATE TABLE economic.bea_gdp_state_quarter_v01_20161027
(
    geofips integer,
    geoname text,
    region text,
    componentid text,
    componentname text,
    industryid text,
    industryclassification text,
    description text,
    year2005q1 text,
    year2005q2 text,
    year2005q3 text,
    year2005q4 text,
    year2006q1 text,
    year2006q2 text,
    year2006q3 text,
    year2006q4 text,
    year2007q1 text,
    year2007q2 text,
    year2007q3 text,
    year2007q4 text,
    year2008q1 text,
    year2008q2 text,
    year2008q3 text,
    year2008q4 text,
    year2009q1 text,
    year2009q2 text,
    year2009q3 text,
    year2009q4 text,
    year2010q1 text,
    year2010q2 text,
    year2010q3 text,
    year2010q4 text,
    year2011q1 text,
    year2011q2 text,
    year2011q3 text,
    year2011q4 text,
    year2012q1 text,
    year2012q2 text,
    year2012q3 text,
    year2012q4 text,
    year2013q1 text,
    year2013q2 text,
    year2013q3 text,
    year2013q4 text,
    year2014q1 text,
    year2014q2 text,
    year2014q3 text,
```

```

year2014q4 text,
year2015q1 text,
year2015q2 text,
year2015q3 text,
year2015q4 text,
year2016q1 text
)
WITH (
  OIDS=FALSE
);
ALTER TABLE economic.bea_gdp_state_quarter_v01_20161027
  OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.bea_gdp_state_quarter_v01_20161027 FROM '/nmsba/data-downloaded/beat-gdp-state-quarterly/qgsp_all.csv' DELIMITER ',' CSV;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.bea_gdp_state_quarter_v01_20161027;

-- example query to test data integration
SELECT geoname, componentname, industryid, industryclassification, description,
year2015q1, year2015q2, year2015q3, year2015q4, year2016q1 FROM
economic.bea_gdp_state_quarter_v01_20161027 WHERE geoname LIKE 'New Mexico%' LIMIT 10;

-- exit PostgreSQL terminal
\q

```

10.3 Annual GDP by State

Download the file “NAICS All GDP Components”. It is necessary to download the file using a browser. Then the file needs to be transferred to the following location on the server: `/nmsba/data-download/beat-gdp-state-naics-annual`

Server Terminal Commands

```

-- create the data directory
sudo mkdir /nmsba/data-download/beat-gdp-state-naics-annual

-- copy the file to the directory after creating the directory

-- change permissions of directory and files
sudo chmod -R 777 /nmsba

-- unzip the files
sudo unzip gsp_naics_all.zip

-- clean the file for importing
sudo sed -i '/See the included footnote file/d' gsp_naics_all.csv

sudo sed -i '/Industry detail is based on the/d' gsp_naics_all.csv

sudo sed -i '/Regional Product Division/d' gsp_naics_all.csv

sudo sed -i '/GeoFIPS/d' gsp_naics_all.csv

-- view the last few lines of the file
tail gsp_naics_all.csv

-- if there are extra lines of text that need to be removed, execute the following
command

```

```
sudo sed -i '$d' gsp_naics_all.csv

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.bea_gdp_state_annual_naics_v01_20161027;
CREATE TABLE economic.bea_gdp_state_annual_naics_v01_20161027
(
    geofips integer,
    geoname text,
    region text,
    componentid text,
    componentname text,
    industryid text,
    industryclassification text,
    description text,
    year1997 text,
    year1998 text,
    year1999 text,
    year2000 text,
    year2001 text,
    year2002 text,
    year2003 text,
    year2004 text,
    year2005 text,
    year2006 text,
    year2007 text,
    year2008 text,
    year2009 text,
    year2010 text,
    year2011 text,
    year2012 text,
    year2013 text,
    year2014 text,
    year2015 text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.bea_gdp_state_annual_naics_v01_20161027
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.bea_gdp_state_annual_naics_v01_20161027 FROM '/nmsba/data-
downloaded/bea-gdp-state-naics-annual/gsp_naics_all.csv' DELIMITER ',' CSV;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.bea_gdp_state_annual_naics_v01_20161027;

-- example query to test data integration
SELECT geoname, componentname, industryid, industryclassification, description,
    year2012, year2013, year2014, year2015 FROM
economic.bea_gdp_state_annual_naics_v01_20161027 WHERE geoname LIKE 'New Mexico%'
LIMIT 10;

-- exit PostgreSQL terminal
\q
```


11 CDC CHRONIC DISEASE INDICATORS

The Centers for Disease Control and Prevention (CDC) maintain records on chronic disease indicators. The data can be exported from <https://chronicdata.cdc.gov/Chronic-Disease-Indicators/U-S-Chronic-Disease-Indicators-CDI-/g4ie-h725>. It is necessary to download the file using a browser. Then the data needs to be transferred to the following location on the server: `/nmsba/data-download/bea-gdp-state-naics-annual`.

Server Terminal Commands

```
-- create the data directory
sudo mkdir /nmsba/data-download/ cdc-cdi

-- copy the file to the directory after creating the directory

-- change permissions of directory and files
sudo chmod -R 777 /nmsba

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.cdc_cdi_v01_20160822;
CREATE TABLE economic.cdc_cdi_v01_20160822
(
    yearstart text,
    yearend text,
    locationabbr text,
    locationdesc text,
    datasource text,
    topic text,
    question text,
    response text,
    datavalueunit text,
    datavaluetypeid text,
    datavaluetype text,
    datavalue text,
    datavaluealt text,
    datavaluefootnotesymbol text,
    datavaluefootnote text,
    lowconfidencelimit text,
    highconfidencelimit text,
    stratificationcategory1 text,
    stratification1 text,
    stratificationcategory2 text,
    stratification2 text,
    stratificationcategory3 text,
    stratification3 text,
    geolocation text,
    topicid text,
    questionid text,
    responseid text,
    locationid text,
    stratificationcategoryid1 text,
    stratificationid1 text,
    stratificationcategoryid2 text,
```

```

    stratificationid2 text,
    stratificationcategoryid3 text,
    stratificationid3 text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.cdc_cdi_v01_20160822
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.cdc_cdi_v01_20160822 FROM '/nmsba/data-downloaded/cdc-
cdi/U.S._Chronic_Disease_Indicators__CDI.csv' DELIMITER ',' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.cdc_cdi_v01_20160822;

-- example query to test data integration
SELECT * FROM economic.cdc_cdi_v01_20160822 LIMIT 10;

-- exit PostgreSQL terminal
\q

```

12 CDC LIVE BIRTH DATA TUTORIAL

The CDC maintains extensive records on birth data that can be found at http://www.cdc.gov/nchs/data_access/Vitalstatsonline.htm. For this tutorial, birth data from 2015 was used.

Server Terminal Commands

```
-- create the data directory
sudo mkdir /nmsba/data-downloaded/cdc-birth

-- create the data directory
sudo chmod -R 777 /nmsba

-- navigate to the data directory
cd /nmsba/data-downloaded/cdc-birth

-- download the data to the data directory
sudo wget
ftp://ftp.cdc.gov/pub/Health_Statistics/NCHS/Datasets/DVS/natality/Nat2015us.zip

-- unzip the files
sudo unzip Nat2015us.zip

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- create temporary data holding table
CREATE TABLE economic.places_staging(data text);

-- import data from file into newly created database table
COPY economic.places_staging FROM '/nmsba/data-downloaded/cdc-birth/Nat2015PublicUS.c20160517.r20160907.txt' DELIMITER '|';

-- DROP TABLE economic.cdc_live_births_v01_20161024;
CREATE TABLE economic.cdc_live_births_v01_20161024
(
  dob_yy text,
  dob_mm text,
  dob_tt text,
  dob_wk text,
  bfacil text,
  f_facility text,
  bfacil3 text,
  mage_impflg text,
  mage_repflg text,
  mager text,
  mager14 text,
  mager9 text,
  mbstate_rec text,
  restatus text,
  mrace31 text,
  mrace6 text,
  mrace15 text,
  mbrace text,
  mraceimp text,
```

mhispl_r text,
f_mhispl text,
mracehisp text,
mar_p text,
dmar text,
mar_imp text,
f_mar_p text,
meduc text,
f_meduc text,
fagerpt_flg text,
fagecomb text,
fagerecl1 text,
frace31 text,
frace6 text,
frace15 text,
fbrace text,
fhisp_r text,
f_fhisp text,
fracehisp text,
feduc text,
filler_f text,
priorlive text,
priordead text,
priorterm text,
lbo_rec text,
tbo_rec text,
illb_r text,
illb_r11 text,
ilop_r text,
ilop_r11 text,
ilp_r text,
ilp_r11 text,
precare text,
f_mpcb text,
precare5 text,
previs text,
previs_rec text,
f_tpcv text,
wic text,
f_wic text,
cig_0 text,
cig_1 text,
cig_2 text,
cig_3 text,
cig0_r text,
cig1_r text,
cig2_r text,
cig3_r text,
f_cigs_0 text,
f_cigs_1 text,
f_cigs_2 text,
f_cigs_3 text,
cig_rec text,
f_tobaco text,
m_ht_in text,
f_m_ht text,
bmi text,
bmi_r text,
pwgt_r text,
f_pwgt text,
dwgt_r text,
f_dwgt text,
wtgain text,


```

wtgain_rec text,
f_wtgain text,
rf_pdiab text,
rf_gdiab text,
rf_phype text,
rf_ghype text,
rf_ehype text,
rf_ppterm text,
f_rf_pdiab text,
f_rf_gdiab text,
f_rf_phyper text,
r_rf_ghyper text,
r_rf_eclamp text,
f_rf_ppb text,
rf_inftr text,
rf_fedrg text,
rf_artec text,
f_rf_inf_drg text,
f_rf_inf_art text,
rf_cesar text,
rf_cesarn text,
f_rf_cesar text,
f_rf_ncesar text,
no_risks text,
ip_gon text,
ip_syph text,
ip_chlam text,
ip_hepb text,
ip_hepc text,
f_ip_gonor text,
f_ip_syph text,
f_ip_chlam text,
f_ip_hepatb text,
f_ip_hepatc text,
no_infec text,
ob_ecvs text,
ob_ecvf text,
f_ob_succ text,
f_ob_fail text,
ld_indl text,
ld_augm text,
ld_ster text,
ld_antb text,
ld_chor text,
ld_anes text,
f_ld_indl text,
f_ld_augm text,
f_ld_ster text,
f_ld_antb text,
f_ld_chor text,
f_ld_anes text,
no_lbrdlv text,
me_pres text,
me_rout text,
me_trial text,
f_me_pres text,
f_me_rout text,
f_me_trial text,
rdmeth_rec text,
dmeth_rec text,
f_dmeth_rec text,
mm_mtr text,
mm_plac text,

```

mm_rupt text,
mm_uhyst text,
mm_aicu text,
f_mm_mtr text,
f_mm_plac text,
f_mm_rupt text,
f_mm_uhyst text,
f_mm_aicu text,
no_mmorg text,
attend text,
mtran text,
pay text,
pay_rec text,
f_pay text,
f_pay_rec text,
apgar5 text,
apgar5r text,
f_apgar5 text,
apgar10 text,
apgar10r text,
dplural text,
imp_plur text,
setorder_r text,
sex text,
imp_sex text,
dlmp_mm text,
dlmp_yy text,
compgst_imp text,
obgest_flg text,
combgest text,
gestrec10 text,
gestrec3 text,
lmpused text,
oegest_comb text,
oegres_r10 text,
oegest_r3 text,
dbwt text,
bwtr12 text,
bwtr4 text,
ab_aven1 text,
ab_aven6 text,
ab_nicu text,
ab_surf text,
ab_anti text,
ab_seiz text,
f_ab_vent text,
f_ab_vent6 text,
f_ab_niuc text,
f_ab_surfac text,
f_ab_antibio text,
f_ab_seiz text,
no_abnorm text,
ca_anen text,
ca_mnsb text,
ca_cchd text,
ca_cdh text,
ca_omph text,
ca_gast text,
f_ca_anen text,
f_ca_menin text,
f_ca_heart text,
f_ca_hernia text,
f_ca_ompha text,

```

f_ca_gastro text,
ca_limb text,
ca_cleft text,
ca_clpal text,
ca_down text,
ca_disor text,
ca_hyop text,
f_ca_limb text,
f_ca_cleftlp text,
f_ca_cleff text,
f_ca_downs text,
f_ca_chrom text,
f_ca_hypos text,
no_congen text,
itran text,
ilive text,
bfed text,
f_bfed text,
ubfacil text,
urf_diab text,
urf_chyper text,
urf_phyper text,
urf_eclam text,
ume_forcp text,
ume_vac text,
uop_induc text,
uld_breech text,
uca_anen text,
uca_spina text,
uca_ompha text,
uca_celftlp text,
uca_hernia text,
uca_downs text
)
WITH (
  OIDS=FALSE
);
ALTER TABLE economic.cdc_live_births_v01_20161024
  OWNER TO postgres;

-- split data apart from the staging 1 column data into each separate column of the
holding table
INSERT INTO economic.cdc_live_births_v01_20161024 (dob_yy, dob_mm, dob_tt, dob_wk,
bfacil, f_facility, bfacil3, mage_impflg, mage_repflg, mager, mager14, mager9,
mbstate_rec, restatus, mrace31, mrace6, mrace15, mbrace, mraceimp, mhispl_r, f_mhispl,
mracehisp, mar_p, dmar, mar_imp, f_mar_p, meduc, f_meduc, fagerpt_flg, fagecomb,
fagerecl1, frace31, frace6, frace15, fbrace, fhisp_r, f_fhisp, fracehisp, feduc,
filler_f, priorlive, priordead, priorterm, lbo_rec, tbo_rec, illb_r, illb_r11, ilop_r,
ilop_r11, ilp_r, ilp_r11, precare, f_mpcb, precare5 , previs, previs_rec, f_tpcv, wic,
f_wic, cig_0, cig_1, cig_2, cig_3, cig0_r, cig1_r, cig2_r, cig3_r, f_cigs_0, f_cigs_1,
f_cigs_2, f_cigs_3, cig_rec, f_tobaco, m_ht_in, f_m_ht, bmi, bmi_r, pwgt_r, f_pwgt,
dwgt_r, f_dwgt, wtgain, wtgain_rec, f_wtgain, rf_pdiab, rf_gdiab, rf_phype, rf_ghype,
rf_ehype, rf_ppterm, f_rf_pdiab, f_rf_gdiab, f_rf_phyper, r_rf_ghyper, r_rf_eclam,
f_rf_ppb, rf_inftr, rf_fedrg, rf_artec, f_rf_inf_drg, f_rf_inf_art, rf_cesar,
rf_cesarn, f_rf_cesar, f_rf_ncesar, no_risks, ip_gon, ip_syph, ip_chlam, ip_hepb,
ip_hepc, f_ip_gonor, f_ip_syph, f_ip_chlam, f_ip_hepatb, f_ip_hepatc, no_infec,
ob_ecvs, ob_ecvf, f_ob_succ, f_ob_fail, ld_indl, ld_augm, ld_ster, ld_antb, ld_chor,
ld_ane, f_ld_indl, f_ld_augm, f_ld_ster, f_ld_antb, f_ld_chor, f_ld_ane, no_lbrdlv,
me_pres, me_rout, me_trial, f_me_pres, f_me_rout, f_me_trial, rdmeth_rec, dmeth_rec,
f_dmeth_rec, mm_mtr, mm_plac, mm_rupt, mm_uhyst, mm_aicu, f_mm_mtr, f_mm_plac,
f_mm_rupt, f_mm_uhyst, f_mm_aicu, no_mmorg, attend, mtran, pay, pay_rec, f_pay,
f_pay_rec, apgar5, apgar5r, f_apgar5, apgar10, apgar10r, dplural, imp_plur,
setorder_r, sex, imp_sex, dlmp_mm, dlmp_yy, compgst_imp, obgest_flg, combgest,

```

```

gestrec10, gestrec3, lmpused, oegest_comb, oegres_r10, oegest_r3, dbwt, bwtr12, bwtr4,
ab_aven1, ab_aven6, ab_nicu, ab_surf, ab_anti, ab_seiz, f_ab_vent, f_ab_vent6,
f_ab_niuc, f_ab_surfac, f_ab_antibio, f_ab_seiz, no_abnorm, ca_anen, ca_mnsb, ca_cchd,
ca_cdh, ca_omph, ca_gast, f_ca_anen, f_ca_menin, f_ca_heart, f_ca_hernia, f_ca_ompha,
f_ca_gastro, ca_limb, ca_cleft, ca_clpal, ca_down, ca_disor, ca_hyop, f_ca_limb,
f_ca_cleftlp, f_ca_cleff, f_ca_downs, f_ca_chrom, f_ca_hypos, no_congen, itran, ilive,
bfed, f_bfed, ubfacil, urf_diab, urf_chyper, urf_phyper, urf_eclam, ume_forcp,
ume_vac, uop_induc, uld_breech, uca_anen, uca_spina, uca_ompha, uca_celftlp,
uca_hernia, uca_downs)
SELECT
substring(data,9,4) AS dob_yy,
substring(data,13,2) AS dob_mm,
substring(data,19,4) AS dob_tt,
substring(data,23,1) AS dob_wk,
substring(data,32,1) AS bfacil,
substring(data,33,1) AS f_facility,
substring(data,50,1) AS bfacil3,
substring(data,73,1) AS mage_impflg,
substring(data,74,1) AS mage_repflg,
substring(data,75,2) AS mager,
substring(data,77,2) AS mager14,
substring(data,79,1) AS mager9,
substring(data,84,1) AS mbstate_rec,
substring(data,104,1) AS restatus,
substring(data,105,2) AS mrace31,
substring(data,107,2) AS mrace6,
substring(data,108,2) AS mrace15,
substring(data,110,1) AS mbrace,
substring(data,111,1) AS mraceimp,
substring(data,115,1) AS mhisp_r,
substring(data,116,1) AS f_mhisp,
substring(data,117,1) AS mracehisp,
substring(data,119,1) AS mar_p,
substring(data,120,1) AS dmar,
substring(data,121,1) AS mar_imp,
substring(data,123,1) AS f_mar_p,
substring(data,124,1) AS meduc,
substring(data,126,1) AS f_meduc,
substring(data,142,1) AS fagerpt_flg,
substring(data,147,2) AS fagecomb,
substring(data,149,2) AS fagerec11,
substring(data,151,2) AS frace31,
substring(data,153,1) AS frace6,
substring(data,154,2) AS frace15,
substring(data,156,1) AS fbrace,
substring(data,160,1) AS fhisp_r,
substring(data,161,1) AS f_fhisp,
substring(data,162,1) AS fracehisp,
substring(data,163,1) AS feduc,
substring(data,165,1) AS filler_f,
substring(data,171,2) AS priorlive,
substring(data,173,2) AS priordead,
substring(data,175,2) AS priorterm,
substring(data,179,1) AS lbo_rec,
substring(data,182,1) AS tbo_rec,
substring(data,198,3) AS illb_r,
substring(data,201,2) AS illb_r11,
substring(data,206,3) AS ilop_r,
substring(data,209,2) AS ilop_r11,
substring(data,214,3) AS ilp_r,
substring(data,217,2) AS ilp_r11,
substring(data,224,2) AS precare,
substring(data,226,1) AS f_mpcb,

```

```

substring(data,227,1) AS precare5 ,
substring(data,238,2) AS previs,
substring(data,242,2) AS previs_rec,
substring(data,244,1) AS f_tpcv,
substring(data,251,1) AS wic,
substring(data,252,1) AS f_wic,
substring(data,253,2) AS cig_0,
substring(data,255,2) AS cig_1,
substring(data,257,2) AS cig_2,
substring(data,259,2) AS cig_3,
substring(data,261,1) AS cig0_r,
substring(data,262,1) AS cig1_r,
substring(data,263,1) AS cig2_r,
substring(data,264,1) AS cig3_r,
substring(data,265,1) AS f_cigs_0,
substring(data,266,1) AS f_cigs_1,
substring(data,267,1) AS f_cigs_2,
substring(data,268,1) AS f_cigs_3,
substring(data,269,1) AS cig_rec,
substring(data,270,1) AS f_tobaco,
substring(data,280,2) AS m_ht_in,
substring(data,282,1) AS f_m_ht,
substring(data,283,4) AS bmi,
substring(data,287,1) AS bmi_r,
substring(data,292,3) AS pwgt_r,
substring(data,295,1) AS f_pwgt,
substring(data,299,3) AS dwgt_r,
substring(data,303,1) AS f_dwgt,
substring(data,304,2) AS wtgain,
substring(data,306,1) AS wtgain_rec,
substring(data,307,1) AS f_wtgain,
substring(data,313,1) AS rf_pdiab,
substring(data,314,1) AS rf_gdiab,
substring(data,315,1) AS rf_phype,
substring(data,316,1) AS rf_ghype,
substring(data,317,1) AS rf_ehype,
substring(data,318,1) AS rf_ppterm,
substring(data,319,1) AS f_rf_pdiab,
substring(data,320,1) AS f_rf_gdiab,
substring(data,321,1) AS f_rf_phyper,
substring(data,322,1) AS r_rf_ghyper,
substring(data,323,1) AS r_rf_eclamp,
substring(data,324,1) AS f_rf_ppb,
substring(data,325,1) AS rf_inftr,
substring(data,326,1) AS rf_fedrg,
substring(data,327,1) AS rf_artec,
substring(data,329,1) AS f_rf_inf_drg,
substring(data,330,1) AS f_rf_inf_art,
substring(data,331,1) AS rf_cesar,
substring(data,332,2) AS rf_cesarn,
substring(data,335,1) AS f_rf_cesar,
substring(data,336,1) AS f_rf_ncesar,
substring(data,337,1) AS no_risks,
substring(data,343,1) AS ip_gon,
substring(data,344,1) AS ip_syph,
substring(data,345,1) AS ip_chlam,
substring(data,346,1) AS ip_hepb,
substring(data,347,1) AS ip_hepc,
substring(data,348,1) AS f_ip_gonor,
substring(data,349,1) AS f_ip_syph,
substring(data,350,1) AS f_ip_chlam,
substring(data,351,1) AS f_ip_hepatb,
substring(data,352,1) AS f_ip_hepatc,

```

```

substring(data,353,1) AS no_infec,
substring(data,360,1) AS ob_ecvs,
substring(data,361,1) AS ob_ecvf,
substring(data,363,1) AS f_ob_succ,
substring(data,364,1) AS f_ob_fail,
substring(data,383,1) AS ld_indl,
substring(data,384,1) AS ld_augm,
substring(data,385,1) AS ld_ster,
substring(data,386,1) AS ld_antb,
substring(data,387,1) AS ld_chor,
substring(data,388,1) AS ld_anes,
substring(data,389,1) AS f_ld_indl,
substring(data,390,1) AS f_ld_augm,
substring(data,391,1) AS f_ld_ster,
substring(data,392,1) AS f_ld_antb,
substring(data,393,1) AS f_ld_chor,
substring(data,394,1) AS f_ld_anes,
substring(data,395,1) AS no_lbrdlv,
substring(data,401,1) AS me_pres,
substring(data,402,1) AS me_rout,
substring(data,403,1) AS me_trial,
substring(data,404,1) AS f_me_pres,
substring(data,405,1) AS f_me_rout,
substring(data,406,1) AS f_me_trial,
substring(data,407,1) AS rdmeth_rec,
substring(data,408,1) AS dmeth_rec,
substring(data,409,1) AS f_dmeth_rec,
substring(data,415,1) AS mm_mtr,
substring(data,416,1) AS mm_plac,
substring(data,417,1) AS mm_rupt,
substring(data,418,1) AS mm_uhyst,
substring(data,419,1) AS mm_aicu,
substring(data,421,1) AS f_mm_mtr,
substring(data,422,1) AS f_mm_plac,
substring(data,423,1) AS f_mm_rupt,
substring(data,424,1) AS f_mm_uhyst,
substring(data,425,1) AS f_mm_aicu,
substring(data,427,1) AS no_mmorg,
substring(data,433,1) AS attend,
substring(data,434,1) AS mtran,
substring(data,435,1) AS pay,
substring(data,436,1) AS pay_rec,
substring(data,437,1) AS f_pay,
substring(data,438,1) AS f_pay_rec,
substring(data,444,2) AS apgar5,
substring(data,446,1) AS apgar5r,
substring(data,447,1) AS f_apgar5,
substring(data,448,2) AS apgar10,
substring(data,450,1) AS apgar10r,
substring(data,454,1) AS dplural,
substring(data,456,1) AS imp_plur,
substring(data,459,1) AS setorder_r,
substring(data,475,1) AS sex,
substring(data,476,1) AS imp_sex,
substring(data,477,2) AS dlmp_mm,
substring(data,481,4) AS dlmp_yy,
substring(data,488,1) AS compgst_imp,
substring(data,489,1) AS obgest_flg,
substring(data,490,2) AS combgest,
substring(data,492,2) AS gestrec10,
substring(data,494,1) AS gestrec3,
substring(data,498,1) AS lmpused,
substring(data,499,2) AS oegest_comb,

```

```

substring(data,501,2) AS oegres_r10,
substring(data,503,1) AS oegest_r3,
substring(data,504,4) AS dbwt,
substring(data,509,2) AS bwtr12,
substring(data,511,1) AS bwtr4,
substring(data,517,1) AS ab_aven1,
substring(data,518,1) AS ab_aven6,
substring(data,519,1) AS ab_nicu,
substring(data,520,1) AS ab_surf,
substring(data,521,1) AS ab_anti,
substring(data,522,1) AS ab_seiz,
substring(data,524,1) AS f_ab_vent,
substring(data,525,1) AS f_ab_vent6,
substring(data,526,1) AS f_ab_niuc,
substring(data,527,1) AS f_ab_surfac,
substring(data,528,1) AS f_ab_antibio,
substring(data,529,1) AS f_ab_seiz,
substring(data,531,1) AS no_abnorm,
substring(data,537,1) AS ca_anen,
substring(data,538,1) AS ca_mnsb,
substring(data,539,1) AS ca_cchd,
substring(data,540,1) AS ca_cdh,
substring(data,541,1) AS ca_omph,
substring(data,542,1) AS ca_gast,
substring(data,543,1) AS f_ca_anen,
substring(data,544,1) AS f_ca_menin,
substring(data,545,1) AS f_ca_heart,
substring(data,546,1) AS f_ca_hernia,
substring(data,547,1) AS f_ca_ompha,
substring(data,548,1) AS f_ca_gastro,
substring(data,549,1) AS ca_limb,
substring(data,550,1) AS ca_cleft,
substring(data,551,1) AS ca_clpal,
substring(data,552,1) AS ca_down,
substring(data,553,1) AS ca_disor,
substring(data,554,1) AS ca_hyop,
substring(data,555,1) AS f_ca_limb,
substring(data,556,1) AS f_ca_cleftlp,
substring(data,557,1) AS f_ca_cleff,
substring(data,558,1) AS f_ca_downs,
substring(data,559,1) AS f_ca_chrom,
substring(data,560,1) AS f_ca_hypos,
substring(data,561,1) AS no_congen,
substring(data,567,1) AS itran,
substring(data,568,1) AS ilive,
substring(data,569,1) AS bfed,
substring(data,570,1) AS f_bfed,
substring(data,1330,1) AS ubfacil,
substring(data,1331,1) AS urf_diab,
substring(data,1332,1) AS urf_chyper,
substring(data,1333,1) AS urf_phyper,
substring(data,1334,1) AS urf_eclam,
substring(data,1335,1) AS ume_forcp,
substring(data,1336,1) AS ume_vac,
substring(data,1337,1) AS uop_induc,
substring(data,1338,1) AS uld_breech,
substring(data,1340,1) AS uca_anen,
substring(data,1341,1) AS uca_spina,
substring(data,1342,1) AS uca_ompha,
substring(data,1343,1) AS uca_celftlp,
substring(data,1344,1) AS uca_hernia,
substring(data,1345,1) AS uca_downs
FROM economic.places_staging;

```

```
-- clean up tables and indexes
vacuum (verbose, analyze) economic.cdc_live_births_v01_20161024;

-- example query to test data integration
SELECT * FROM economic.cdc_live_births_v01_20161024 LIMIT 10;

-- drop the temporary table
DROP TABLE economic.places_staging;

-- exit PostgreSQL terminal
\q
```


13 BLS TUTORIAL

The Bureau of Labor Statistics (BLS) maintains an extensive data repository of information located at: <http://www.bls.gov/data/>. For this tutorial, data on “Employment, Hours, and Earnings - State and Metro Area” was used. Data can be downloaded from <http://download.bls.gov/pub/time.series/sm/>. The help file for understanding the data files can be viewed at: <http://download.bls.gov/pub/time.series/sm/sm.txt>

13.1 Reference Tables

13.1.1 *sm.area*

Server Terminal Commands

```
-- create the data directory
sudo mkdir /nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/

-- navigate to the data directory
cd /nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/

-- download the data to the data directory
sudo wget http://download.bls.gov/pub/time.series/sm/sm.area

-- remove tag line tabs
sudo sed -i 's/\s*$//' sm.area
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.bls_employment_monthly_state_metro_def_area_20161021;
CREATE TABLE economic.bls_employment_monthly_state_metro_def_area_20161021
(
    area_code text,
    area_name text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.bls_employment_monthly_state_metro_def_area_20161021
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.bls_employment_monthly_state_metro_def_area_20161021 FROM '/nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/sm.area' DELIMITER E'\t' CSV
HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze)
economic.bls_employment_monthly_state_metro_def_area_20161021;

-- example query to test data integration
SELECT * FROM economic.bls_employment_monthly_state_metro_def_area_20161021 LIMIT 10;

-- exit PostgreSQL terminal
\q
```

13.1.2 sm.series

Server Terminal Commands

```
-- navigate to the data directory
cd /nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/

-- download the data to the data directory
sudo wget http://download.bls.gov/pub/time.series/sm/sm.series

-- remove tag line tabs
sudo sed -i 's/\s*$//' sm.series
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.bls_employment_monthly_state_metro_def_series_20161021;
CREATE TABLE economic.bls_employment_monthly_state_metro_def_series_20161021
(
    series_id text,
    state_code text,
    area_code text,
    supersector_code text,
    industry_code text,
    data_type_code text,
    seasonal text,
    benchmark_year text,
    footnote_codes text,
    begin_year text,
    begin_period text,
    end_year text,
    end_period text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.bls_employment_monthly_state_metro_def_series_20161021
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.bls_employment_monthly_state_metro_def_series_20161021 FROM
    '/nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/sm.series'
    DELIMITER E'\t' CSV HEADER;

-- data cleanup
UPDATE economic.bls_employment_monthly_state_metro_def_series_20161021 SET series_id =
    trim(series_id);

-- clean up tables and indexes
vacuum (verbose, analyze)
    economic.bls_employment_monthly_state_metro_def_series_20161021;

-- example query to test data integration
SELECT * FROM economic.bls_employment_monthly_state_metro_def_series_20161021 WHERE
    area_code = '10740' LIMIT 10;

-- exit PostgreSQL terminal
\q
```

13.1.3 sm.industry

Server Terminal Commands

```
-- navigate to the data directory
cd /nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/

-- download the data to the data directory
sudo wget http://download.bls.gov/pub/time.series/sm/sm.industry

-- remove tag line tabs
sudo sed -i 's/\s*$//' sm.industry
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.bls_employment_monthly_state_metro_def_industry_20161021;
CREATE TABLE economic.bls_employment_monthly_state_metro_def_industry_20161021
(
    industry_code text,
    industry_name text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.bls_employment_monthly_state_metro_def_industry_20161021
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.bls_employment_monthly_state_metro_def_industry_20161021 FROM
'/nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/sm.industry'
DELIMITER E'\t' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze)
economic.bls_employment_monthly_state_metro_def_industry_20161021;

-- example query to test data integration
SELECT * FROM economic.bls_employment_monthly_state_metro_def_industry_20161021 WHERE
industry_code = '10113300' LIMIT 10;

-- exit PostgreSQL terminal
\q
```

13.1.4 sm.data_type

Server Terminal Commands

```
-- navigate to the data directory
cd /nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/

-- download the data to the data directory
sudo wget http://download.bls.gov/pub/time.series/sm/sm.data_type

-- remove tag line tabs
sudo sed -i 's/\s*$//' sm.data_type
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.bls_employment_monthly_state_metro_def_datatype_20161021;
CREATE TABLE economic.bls_employment_monthly_state_metro_def_datatype_20161021
(
    data_type_code text,
    data_type_text text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.bls_employment_monthly_state_metro_def_datatype_20161021
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.bls_employment_monthly_state_metro_def_datatype_20161021 FROM
'/nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/sm.data_type'
DELIMITER E'\t' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze)
economic.bls_employment_monthly_state_metro_def_datatype_20161021;

-- example query to test data integration
SELECT * FROM economic.bls_employment_monthly_state_metro_def_datatype_20161021 LIMIT
10;

-- exit PostgreSQL terminal
\q
```

13.1.5 sm.footnote

Server Terminal Commands

```
-- navigate to the data directory
cd /nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/

-- download the data to the data directory
sudo wget http://download.bls.gov/pub/time.series/sm/sm.footnote

-- remove tag line tabs
sudo sed -i 's/\s*$//' sm.footnote
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.bls_employment_monthly_state_metro_def_footnote_20161021;
CREATE TABLE economic.bls_employment_monthly_state_metro_def_footnote_20161021
(
    footnote_code text,
    footnote_text text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.bls_employment_monthly_state_metro_def_footnote_20161021
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.bls_employment_monthly_state_metro_def_footnote_20161021 FROM
'/nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/sm.footnote'
DELIMITER E'\t' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze)
economic.bls_employment_monthly_state_metro_def_footnote_20161021;

-- example query to test data integration
SELECT * FROM economic.bls_employment_monthly_state_metro_def_footnote_20161021 LIMIT
10;

-- exit PostgreSQL terminal
\q
```

13.1.6 sm.supersector

Server Terminal Commands

```
-- navigate to the data directory
cd /nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/

-- download the data to the data directory
sudo wget http://download.bls.gov/pub/time.series/sm/sm.supersector

-- remove tag line tabs
sudo sed -i 's/\s*$//' sm.supersector
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.bls_employment_monthly_state_metro_def_supersector_20161021;
CREATE TABLE economic.bls_employment_monthly_state_metro_def_supersector_20161021
(
    supersector_code text,
    supersector_text text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.bls_employment_monthly_state_metro_def_supersector_20161021
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.bls_employment_monthly_state_metro_def_supersector_20161021 FROM
'/nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/sm.supersector'
DELIMITER E'\t' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze)
economic.bls_employment_monthly_state_metro_def_supersector_20161021;

-- example query to test data integration
SELECT * FROM economic.bls_employment_monthly_state_metro_def_supersector_20161021
LIMIT 10;

-- exit PostgreSQL terminal
\q
```

13.1.7 sm.state

Server Terminal Commands

```
-- navigate to the data directory
cd /nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/

-- download the data to the data directory
sudo wget http://download.bls.gov/pub/time.series/sm/sm.state

-- remove tag line tabs
sudo sed -i 's/\s*$//' sm.state
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.bls_employment_monthly_state_metro_def_state_20161021;
CREATE TABLE economic.bls_employment_monthly_state_metro_def_state_20161021
(
    state_code text,
    state_text text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.bls_employment_monthly_state_metro_def_state_20161021
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.bls_employment_monthly_state_metro_def_state_20161021 FROM '/nmsba/data-downloaded/bls-employment-monthly-state-metro/definitions/sm.state' DELIMITER E'\t'
CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze)
economic.bls_employment_monthly_state_metro_def_state_20161021;

-- example query to test data integration
SELECT * FROM economic.bls_employment_monthly_state_metro_def_state_20161021 LIMIT 10;

-- exit PostgreSQL terminal
\q
```

13.2 Data

Bureau of Labor Statistics (BLS) data can found at: <http://www.bls.gov/data/>

Server Terminal Commands

```
-- navigate to the data directory
cd /nmsba/data-downloaded/bls-employment-monthly-state-metro/

-- download the data to the data directory
sudo wget http://download.bls.gov/pub/time.series/sm/sm.data.1.AllData

-- remove tag line tabs
-- sudo sed -i 's/\s*$//' sm.data.1.AllData
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.bls_employment_monthly_state_metro_data_20161021;
CREATE TABLE economic.bls_employment_monthly_state_metro_data_20161021
(
    series_id text,
    year text,
    period text,
    value text,
    footnote_codes text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.bls_employment_monthly_state_metro_data_20161021
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.bls_employment_monthly_state_metro_data_20161021 FROM '/nmsba/data-downloaded/bls-employment-monthly-state-metro/sm.data.1.AllData' DELIMITER E'\t' CSV
HEADER NULL '';

-- create indexes
CREATE INDEX idx_bls_employment_monthly_state_metro_data_20161021 ON
economic.bls_employment_monthly_state_metro_data_20161021 USING btree(series_id);

CREATE INDEX idx_bls_employment_monthly_state_metro_def_series_20161021 ON
economic.bls_employment_monthly_state_metro_def_series_20161021 USING
btree(series_id);

-- remove blank space in data values
UPDATE economic.bls_employment_monthly_state_metro_data_20161021 SET series_id =
trim(series_id), year = trim(year), period = trim(period), value = trim(value),
footnote_codes = trim(footnote_codes);

-- clean up tables and indexes
vacuum (verbose, analyze) economic.bls_employment_monthly_state_metro_data_20161021;

-- example query to test data integration
SELECT * FROM economic.bls_employment_monthly_state_metro_data_20161021 LIMIT 10;

-- exit PostgreSQL terminal
\q
```


13.3 Complex Data Query Examples

```
-- example 1
SELECT data.series_id, area.area_name, data.year, data.period, data.value,
ind.industry_name, dt.data_type_text
FROM      economic.bls_employment_monthly_state_metro_data_20161021      data
INNER JOIN economic.bls_employment_monthly_state_metro_def_series_20161021 series ON
series.series_id      = data.series_id
INNER JOIN economic.bls_employment_monthly_state_metro_def_state_20161021 state ON
series.state_code      = state.state_code
INNER JOIN economic.bls_employment_monthly_state_metro_def_area_20161021 area ON
series.area_code      = area.area_code
INNER JOIN economic.bls_employment_monthly_state_metro_def_industry_20161021 ind ON
series.industry_code = ind.industry_code
INNER JOIN economic.bls_employment_monthly_state_metro_def_datatype_20161021 dt ON
series.data_type_code = dt.data_type_code
WHERE area.area_code = '10740' AND ind.industry_code = '15000000'
ORDER BY data.year DESC
LIMIT 30;
```

```
-- example 2
SELECT DISTINCT ON (ind.industry_name) ind.industry_name, ind.industry_code
FROM      economic.bls_employment_monthly_state_metro_data_20161021      data
INNER JOIN economic.bls_employment_monthly_state_metro_def_series_20161021 series ON
series.series_id      = data.series_id
INNER JOIN economic.bls_employment_monthly_state_metro_def_state_20161021 state ON
series.state_code      = state.state_code
INNER JOIN economic.bls_employment_monthly_state_metro_def_area_20161021 area ON
series.area_code      = area.area_code
INNER JOIN economic.bls_employment_monthly_state_metro_def_industry_20161021 ind ON
series.industry_code = ind.industry_code
INNER JOIN economic.bls_employment_monthly_state_metro_def_datatype_20161021 dt ON
series.data_type_code = dt.data_type_code
WHERE area.area_code = '10740';
```

13.4 Write Query Data to File

```
Copy (
  SELECT data.series_id, area.area_name, data.year, data.period, data.value,
ind.industry_name, dt.data_type_text
  FROM      economic.bls_employment_monthly_state_metro_data_20161021      data
  INNER JOIN economic.bls_employment_monthly_state_metro_def_series_20161021 series ON
series.series_id      = data.series_id
  INNER JOIN economic.bls_employment_monthly_state_metro_def_state_20161021 state ON
series.state_code      = state.state_code
  INNER JOIN economic.bls_employment_monthly_state_metro_def_area_20161021 area ON
series.area_code      = area.area_code
  INNER JOIN economic.bls_employment_monthly_state_metro_def_industry_20161021 ind ON
series.industry_code = ind.industry_code
  INNER JOIN economic.bls_employment_monthly_state_metro_def_datatype_20161021 dt ON
series.data_type_code = dt.data_type_code
  WHERE state.state_code = '35' AND ind.industry_code = '15000000'
) To '/tmp/bls2.csv' With CSV DELIMITER ',' HEADER;
```


14 NCES TUTORIAL

The National Center for Education Statistics (NCES) offers many freely available datasets. The example NCES data tutorial provided here is from the Integrated Postsecondary Education Data System (IPEDS). The IPEDS is a system of surveys designed to collect data from all primary providers of postsecondary education.

14.1 Institution Details

IPEDS data can be directly downloaded from: <https://nces.ed.gov/ipeds/datacenter/login.aspx>

Server Terminal Commands

```
-- create the data directory
sudo mkdir /nmsba/data-downloaded/nces-institution-details/directory-information

-- navigate to the data directory
cd /nmsba/data-downloaded/nces-institution-details/directory-information

-- there are many file options when viewing the website, filter by 2014 and select
Institutional Characteristics
-- HD2014 contains the institutional directory information
-- download the data to the data directory on the server
sudo wget https://nces.ed.gov/ipeds/datacenter/data/HD2014.zip

-- unzip the files
sudo unzip HD2014.zip

-- the file contains non UTF-8 characters so we must convert to UTF-8
iconv -f ISO-8859-15 -t UTF8 hd2014.csv > hd2014-encoded.csv

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.edu_stats_institute_dir_v0120161025;
CREATE TABLE economic.edu_stats_institute_dir_v0120161025
(
    unitid integer,
    instnm text,
    addr text,
    city text,
    stabbr text,
    zip text,
    fips integer,
    obereg integer,
    chfnm text,
    chftitle text,
    gentele text,
    ein integer,
    opeid integer,
    opeflag integer,
    webaddr text,
    adminurl text,
    faidurl text,
    applurl text,
```

```

npricurl text,
veturl text,
athurl text,
sector integer,
iclevel integer,
control integer,
hloffer integer,
ugoffer integer,
groffer integer,
hdegofr1 integer,
deggrant integer,
hbcu integer,
hospital integer,
medical integer,
tribal integer,
locale integer,
openpubl integer,
act text,
newid integer,
deathyr integer,
closedat text,
cyactive integer,
postsec integer,
pseflag integer,
pset4flg integer,
rptmth integer,
ialias text,
instcat integer,
ccbasic integer,
ccipug integer,
ccipgrad integer,
ccugprof integer,
ccenrprf integer,
ccsizset integer,
carnegie integer,
landgrnt integer,
instsize integer,
cbsa integer,
cbsatype integer,
csa integer,
necta integer,
flsystyp integer,
flsysnam text,
flsyscod text,
countycd integer,
countynm text,
cngdstcd integer,
longitud decimal,
latitude decimal,
dfrcgid integer,
dfrcuscg integer
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.edu_stats_institute_dir_v0120161025
    OWNER TO postgres;

-- import data from file into newly created database table
COPY economic.edu_stats_institute_dir_v0120161025 FROM '/nmsba/data-downloaded/nces-
institution-details/directory-information/HD2014/hd2014-encoded.csv' DELIMITER ',' CSV
HEADER;
```

```
-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_institute_dir_v0120161025;

-- example query to test data integration
SELECT unitid, instnm, addr, city, stabbr, zip FROM
economic.edu_stats_institute_dir_v0120161025 WHERE stabbr = 'NM';

-- exit PostgreSQL terminal
\q
```

14.2 Admissions and Test Scores

Every year of Admissions and Test Scores data has different variables. The first step is to create a master table of all variables. Then each year of data is imported into the master table.

Server Terminal Commands

```
-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- create master table of all column variables
-- DROP TABLE economic.edu_stats_admissions_tests_v01_20161025;
CREATE TABLE economic.edu_stats_admissions_tests_v01_20161025
(
    year integer,
    unitid text,
    peolistr text,
    peo2istr text,
    peo3istr text,
    peo4istr text,
    peo5istr text,
    peo6istr text,
    cntlaffi text,
    pubprime text,
    pubsecon text,
    relaffil text,
    level1 text,
    level2 text,
    level3 text,
    level4 text,
    level5 text,
    level6 text,
    level7 text,
    level8 text,
    level12 text,
    level17 text,
    level18 text,
    level19 text,
    openadmp text,
    admcon1 text,
    admcon2 text,
    admcon3 text,
    admcon4 text,
    admcon5 text,
    admcon6 text,
    admcon7 text,
    admcon8 text,
```

admcon9 text,
appdate text,
xapplcnm text,
applcnm text,
xapplcnw text,
applcnw text,
xadmssnm text,
admssnm text,
xadmssnw text,
admssnw text,
xenrlftm text,
enrlftm text,
xenrlftw text,
enrlftw text,
xenrlptm text,
enrlptm text,
xenrlptw text,
enrlptw text,
satactdt text,
xsatnum text,
satnum text,
xsatpct text,
satpct text,
xactnum text,
actnum text,
xactpct text,
actpct text,
xsatvr25 text,
satvr25 text,
xsatvr75 text,
satvr75 text,
xsatmt25 text,
satmt25 text,
xsatmt75 text,
satmt75 text,
xsatwr25 text,
satwr25 text,
xsatwr75 text,
satwr75 text,
xactcm25 text,
actcm25 text,
xactcm75 text,
actcm75 text,
xacten25 text,
acten25 text,
xacten75 text,
acten75 text,
xactmt25 text,
actmt25 text,
xactmt75 text,
actmt75 text,
xactwr25 text,
actwr25 text,
xactwr75 text,
actwr75 text,
credits1 text,
credits2 text,
credits3 text,
credits4 text,
slo5 text,
slo51 text,
slo52 text,
slo53 text,

slo6 text,
slo7 text,
slo8 text,
slo81 text,
slo82 text,
slo83 text,
slo9 text,
yrscoll text,
stusrv1 text,
stusrv2 text,
stusrv3 text,
stusrv4 text,
stusrv8 text,
stusrv9 text,
libfac text,
athassoc text,
assoc1 text,
assoc2 text,
assoc3 text,
assoc4 text,
assoc5 text,
assoc6 text,
sport1 text,
confno1 text,
sport2 text,
confno2 text,
sport3 text,
confno3 text,
sport4 text,
confno4 text,
calsys text,
xappfeeu text,
applfeeu text,
xappfeeg text,
applfeeg text,
ft_ug text,
ft_ftug text,
ftgdnidp text,
pt_ug text,
pt_ftug text,
ptgdnidp text,
docpp text,
docppsp text,
tuitvary text,
room text,
xroomcap text,
roomcap text,
board text,
xmealswk text,
mealswk text,
xroomamt text,
roomamt text,
xbordamt text,
boardamt text,
xrmbdamt text,
rmbdamt text,
alloncam text,
xenrlm text,
enrlm text,
xenrlw text,
enrlw text,
xenrlt text,
enrlt text,

```

xapplcn text,
applcn text,
xadmssn text,
admssn text,
xenrlft text,
enrlft text,
xenrlpt text,
enrlpt text,
tuitpl text,
tuitpl1 text,
tuitpl2 text,
tuitpl3 text,
tuitpl4 text,
disab text,
xdisabpc text,
disabpct text,
dstnced text,
dstnced1 text,
dstnced2 text,
dstnced3 text,
slo3 text,
ft_gd text,
pt_gd text,
pctpost text,
level9 text,
level10 text,
level11 text,
xappfeep text,
applfeep text,
ft_fp text,
pt_fp text,
apfee text,
accrd1 text,
accrd2 text,
regaccrd text,
accrd3 text,
accrd4 text,
sacchr text,
slo1 text,
slo2 text,
slo4 text,
insttoyr text,
fopna text,
fopna1 text,
fopna2 text,
ftstu text,
ptstu text
)
WITH (
  OIDS=FALSE
);
ALTER TABLE economic.edu_stats_admissions_tests_v01_20161025
  OWNER TO postgres;

```


14.2.1 Data from 2001

There are many file options when viewing the website, filter by All Years and select Admissions and Test Scores. Find the file named IC2001, which is for 2001.

Server Terminal Commands

```
-- create the data directory

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores/data

-- navigate to the data directory
cd /nmsba/data-downloaded/nces-admissions-and-test-scores/data

sudo wget https://nces.ed.gov/ipeds/datacenter/data/IC2001.zip

-- unzip the files
sudo unzip IC2001.zip

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.edu_stats_admissions_tests_2001;
CREATE TABLE economic.edu_stats_admissions_tests_2001
(
    unitid text,
    peo1istr text,
    peo2istr text,
    peo3istr text,
    peo4istr text,
    peo5istr text,
    peo6istr text,
    cntlaffi text,
    pubprime text,
    pubsecon text,
    relaffil text,
    level1 text,
    level2 text,
    level3 text,
    level4 text,
    level5 text,
    level6 text,
    level7 text,
    level8 text,
    level9 text,
    level10 text,
    level11 text,
    level12 text,
    fopna text,
    fopna1 text,
    fopna2 text,
    accrd1 text,
    accrd2 text,
    regacrd text,
    accrd3 text,
```

accrd4 text,
saccr text,
openadmp text,
admcon1 text,
admcon2 text,
admcon3 text,
admcon4 text,
admcon5 text,
admcon6 text,
admcon7 text,
admcon8 text,
appdate text,
xapplcnm text,
applcnm text,
xapplcnw text,
applcnw text,
xadmssnm text,
admssnm text,
xadmssnw text,
admssnw text,
xenrlftm text,
enrlftm text,
xenrlftw text,
enrlftw text,
xenrlptm text,
enrlptm text,
xenrlptw text,
enrlptw text,
satactdt text,
xsatnum text,
satnum text,
xsatpct text,
satpct text,
xactnum text,
actnum text,
xactpct text,
actpct text,
xsatvr25 text,
satvr25 text,
xsatvr75 text,
satvr75 text,
xsatmt25 text,
satmt25 text,
xsatmt75 text,
satmt75 text,
xactcm25 text,
actcm25 text,
xactcm75 text,
actcm75 text,
xacten25 text,
acten25 text,
xacten75 text,
acten75 text,
xactmt25 text,
actmt25 text,
xactmt75 text,
actmt75 text,
credits1 text,
credits2 text,
credits3 text,
credits4 text,
slo1 text,
slo2 text,

slo3 text,
slo4 text,
slo5 text,
slo51 text,
slo52 text,
slo53 text,
slo6 text,
slo7 text,
slo8 text,
slo81 text,
slo82 text,
slo83 text,
slo9 text,
yrscoll text,
stusrv1 text,
stusrv2 text,
stusrv3 text,
stusrv4 text,
stusrv8 text,
stusrv9 text,
libfac text,
athassoc text,
assoc1 text,
assoc2 text,
assoc3 text,
assoc4 text,
assoc5 text,
assoc6 text,
sport1 text,
confno1 text,
sport2 text,
confno2 text,
sport3 text,
confno3 text,
sport4 text,
confno4 text,
insttoyr text,
pctpost text,
calsys text,
apfee text,
xappfee text,
applfee text,
xappfeeg text,
applfeeg text,
xappfeep text,
applfeep text,
ftstu text,
ft_ug text,
ft_ftug text,
ft_gd text,
ft_fp text,
tuitvary text,
room text,
xroomcap text,
roomcap text,
board text,
xmealswk text,
mealswk text,
xroomamt text,
roomamt text,
xbordamt text,
boardamt text,
xrmbdamt text,

```

    rmbrdamt text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.edu_stats_admissions_tests_2001
    OWNER TO postgres;

-- import data from file into newly created database table for the 2001 data
COPY economic.edu_stats_admissions_tests_2001 FROM '/nmsba/data-downloaded/nces-
admissions-and-test-scores/data/ic2001.csv' DELIMITER ',' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_2001;

-- import 2001 data into master table
INSERT INTO economic.edu_stats_admissions_tests_v01_20161025 (year, unitid, peolistr,
peo2istr, peo3istr, peo4istr, peo5istr, peo6istr, cntlaffi, pubprime, pubsecon,
relaffil, level1, level2, level3, level4, level5, level6, level7, level8, level9,
level10, level11, level12, fopna, fopna1, fopna2, accrd1, accrd2, regacrd, accrd3,
acrd4, saccr, openadmp, admcon1, admcon2, admcon3, admcon4, admcon5, admcon6,
admcon7, admcon8, appdate, xapplcnm, applcnm, xapplcnw, applcnw, xadmssnm, admssnm,
xadmssnw, admssnw, xenrlftm, enrftm, xenrlftw, enrftw, xenrlptm, enrptm, xenrlptw,
enrlptw, satactdt, xsatnum, satnum, xsatpct, satpct, xactnum, actnum, xactpct, actpct,
xsatvr25, satvr25, xsatvr75, satvr75, xsatmt25, satmt25, xsatmt75, satmt75, xactcm25,
actcm25, xactcm75, actcm75, xacten25, acten25, xacten75, acten75, xactmt25, actmt25,
xactmt75, actmt75, credits1, credits2, credits3, credits4, slo1, slo2, slo3, slo4,
slo5, slo51, slo52, slo53, slo6, slo7, slo8, slo81, slo82, slo83, slo9, yrscoll,
stusrv1, stusrv2, stusrv3, stusrv4, stusrv8, stusrv9, libfac, athassoc, assoc1,
assoc2, assoc3, assoc4, assoc5, assoc6, sport1, confno1, sport2, confno2, sport3,
confno3, sport4, confno4, insttoyr, pctpost, calsys, apfee, xappfee, applfee,
xappfeeg, applfeeg, xappfeep, applfeep, ftstu, ft_ug, ft_ftug, ft_gd, ft_fp, tuitvary,
room, xroomcap, roomcap, board, xmealswk, mealswk, xroomamt, roomamt, xbordamt,
boardamt, xrmdbamt, rmbrdamt)
SELECT
2001 as year,
unitid AS unitid,
peolistr AS peolistr,
peo2istr AS peo2istr,
peo3istr AS peo3istr,
peo4istr AS peo4istr,
peo5istr AS peo5istr,
peo6istr AS peo6istr,
cntlaffi AS cntlaffi,
pubprime AS pubprime,
pubsecon AS pubsecon,
relaffil AS relaffil,
level1 AS level1,
level2 AS level2,
level3 AS level3,
level4 AS level4,
level5 AS level5,
level6 AS level6,
level7 AS level7,
level8 AS level8,
level9 AS level9,
level10 AS level10,
level11 AS level11,
level12 AS level12,
fopna AS fopna,
fopna1 AS fopna1,
fopna2 AS fopna2,
acrd1 AS accrd1,

```

```

accrd2 AS accrd2,
regaccrd AS regaccrd,
accrd3 AS accrd3,
accrd4 AS accrd4,
saccr AS saccr,
openadmp AS openadmp,
admcon1 AS admcon1,
admcon2 AS admcon2,
admcon3 AS admcon3,
admcon4 AS admcon4,
admcon5 AS admcon5,
admcon6 AS admcon6,
admcon7 AS admcon7,
admcon8 AS admcon8,
appdate AS appdate,
xapplcnm AS xapplcnm,
applcnm AS applcnm,
xapplcnw AS xapplcnw,
applcnw AS applcnw,
xadmssnm AS xadmssnm,
admssnm AS admssnm,
xadmssnw AS xadmssnw,
admssnw AS admssnw,
xenrlftm AS xenrlftm,
enrlftm AS enrlftm,
xenrlftw AS xenrlftw,
enrlftw AS enrlftw,
xenrlptm AS xenrlptm,
enrlptm AS enrlptm,
xenrlptw AS xenrlptw,
enrlptw AS enrlptw,
satactdt AS satactdt,
xsatnum AS xsatnum,
satnum AS satnum,
xsatpct AS xsatpct,
satpct AS satpct,
xactnum AS xactnum,
actnum AS actnum,
xactpct AS xactpct,
actpct AS actpct,
xsatvr25 AS xsatvr25,
satvr25 AS satvr25,
xsatvr75 AS xsatvr75,
satvr75 AS satvr75,
xsatmt25 AS xsatmt25,
satmt25 AS satmt25,
xsatmt75 AS xsatmt75,
satmt75 AS satmt75,
xactcm25 AS xactcm25,
actcm25 AS actcm25,
xactcm75 AS xactcm75,
actcm75 AS actcm75,
xacten25 AS xacten25,
acten25 AS acten25,
xacten75 AS xacten75,
acten75 AS acten75,
xactmt25 AS xactmt25,
actmt25 AS actmt25,
xactmt75 AS xactmt75,
actmt75 AS actmt75,
credits1 AS credits1,
credits2 AS credits2,
credits3 AS credits3,

```

```

credits4 AS credits4,
slo1 AS slo1,
slo2 AS slo2,
slo3 AS slo3,
slo4 AS slo4,
slo5 AS slo5,
slo51 AS slo51,
slo52 AS slo52,
slo53 AS slo53,
slo6 AS slo6,
slo7 AS slo7,
slo8 AS slo8,
slo81 AS slo81,
slo82 AS slo82,
slo83 AS slo83,
slo9 AS slo9,
yrscoll AS yrscoll,
stusrv1 AS stusrv1,
stusrv2 AS stusrv2,
stusrv3 AS stusrv3,
stusrv4 AS stusrv4,
stusrv8 AS stusrv8,
stusrv9 AS stusrv9,
libfac AS libfac,
athassoc AS athassoc,
assoc1 AS assoc1,
assoc2 AS assoc2,
assoc3 AS assoc3,
assoc4 AS assoc4,
assoc5 AS assoc5,
assoc6 AS assoc6,
sport1 AS sport1,
confno1 AS confno1,
sport2 AS sport2,
confno2 AS confno2,
sport3 AS sport3,
confno3 AS confno3,
sport4 AS sport4,
confno4 AS confno4,
insttoyr AS insttoyr,
pctpost AS pctpost,
calsys AS calsys,
apfee AS apfee,
xappfeeu AS xappfeeu,
applfeeu AS applfeeu,
xappfeeg AS xappfeeg,
applfeeg AS applfeeg,
xappfeep AS xappfeep,
applfeep AS applfeep,
ftstu AS ftstu,
ft_ug AS ft_ug,
ft_ftug AS ft_ftug,
ft_gd AS ft_gd,
ft_fp AS ft_fp,
tuitvary AS tuitvary,
room AS room,
xroomcap AS xroomcap,
roomcap AS roomcap,
board AS board,
xmealswk AS xmealswk,
mealswk AS mealswk,
xroomamt AS xroomamt,
roomamt AS roomamt,

```

```

xbordamt AS xbordamt,
boardamt AS boardamt,
xrmbdamt AS xrmbdamt,
rmbdamt AS rmbdamt
FROM economic.edu_stats_admissions_tests_2001;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_v01_20161025;

-- example query to test data integration
SELECT year, dir.unitid, instnm, addr, city, stabbr, zip, enrftm, enrftw, satnum,
actnum FROM economic.edu_stats_institute_dir_v0120161025 dir INNER JOIN
economic.edu_stats_admissions_tests_v01_20161025 ats ON dir.unitid::integer =
ats.unitid::integer WHERE dir.unitid = 187985;

-- exit PostgreSQL terminal
\q

```

14.2.2 Data from 2002

There are many file options when viewing the website, filter by All Years and select Admissions and Test Scores. Find the file named IC2002, which is for 2002.

Server Terminal Commands

```

-- create the data directory

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores/data

-- navigate to the data directory
cd /nmsba/data-downloaded/nces-admissions-and-test-scores/data

sudo wget https://nces.ed.gov/ipeds/datacenter/data/IC2002.zip

-- unzip the files
sudo unzip IC2002.zip

-- login to the PostgreSQL command line
sudo -u postgres psql postgres

```

PostgreSQL Terminal Commands

```

-- connect to the raw database
\c raw

-- DROP TABLE economic.edu_stats_admissions_tests_2002;
CREATE TABLE economic.edu_stats_admissions_tests_2002
(
    unitid text,
    peolistr text,
    peo2istr text,
    peo3istr text,
    peo4istr text,
    peo5istr text,
    peo6istr text,
    cntlaffi text,
    pubprime text,
    pubsecon text,
    relaffil text,
    levell text,

```

level2 text,
level3 text,
level4 text,
level5 text,
level6 text,
level7 text,
level8 text,
level9 text,
level10 text,
level11 text,
level12 text,
fopna text,
fopna1 text,
fopna2 text,
accrd1 text,
accrd2 text,
regaccrd text,
accrd3 text,
accrd4 text,
sacchr text,
openadmp text,
admcon1 text,
admcon2 text,
admcon3 text,
admcon4 text,
admcon5 text,
admcon6 text,
admcon7 text,
admcon8 text,
appdate text,
xapplcnm text,
applcnm text,
xapplcnw text,
applcnw text,
xadmssnm text,
admssnm text,
xadmssnw text,
admssnw text,
xenrlftm text,
enrlftm text,
xenrlftw text,
enrlftw text,
xenrlptm text,
enrlptm text,
xenrlptw text,
enrlptw text,
satactdt text,
xsatnum text,
satnum text,
xsatpct text,
satpct text,
xactnum text,
actnum text,
xactpct text,
actpct text,
xsatvr25 text,
satvr25 text,
xsatvr75 text,
satvr75 text,
xsatmt25 text,
satmt25 text,
xsatmt75 text,
satmt75 text,

xactcm25 text,
actcm25 text,
xactcm75 text,
actcm75 text,
xacten25 text,
acten25 text,
xacten75 text,
acten75 text,
xactmt25 text,
actmt25 text,
xactmt75 text,
actmt75 text,
credits1 text,
credits2 text,
credits3 text,
credits4 text,
slo1 text,
slo2 text,
slo3 text,
slo4 text,
slo5 text,
slo51 text,
slo52 text,
slo53 text,
slo6 text,
slo7 text,
slo8 text,
slo81 text,
slo82 text,
slo83 text,
slo9 text,
yrscoll text,
stusrv1 text,
stusrv2 text,
stusrv3 text,
stusrv4 text,
stusrv8 text,
stusrv9 text,
libfac text,
athassoc text,
assoc1 text,
assoc2 text,
assoc3 text,
assoc4 text,
assoc5 text,
assoc6 text,
sport1 text,
confno1 text,
sport2 text,
confno2 text,
sport3 text,
confno3 text,
sport4 text,
confno4 text,
insttoyr text,
pctpost text,
calsys text,
apfee text,
xappfeeu text,
applfeeu text,
xappfeeg text,
applfeeg text,
xappfeep text,

```

    applfeep text,
    ftstu text,
    ft_ug text,
    ft_ftug text,
    ft_gd text,
    ft_fp text,
    ptstu text,
    pt_ug text,
    pt_ftug text,
    pt_gd text,
    tuitvary text,
    room text,
    xroomcap text,
    roomcap text,
    board text,
    xmealswk text,
    mealswk text,
    xroomamt text,
    roomamt text,
    xbordamt text,
    boardamt text,
    xrmbdamt text,
    rmbdamt text,
    xenrlm text,
    enrlm text,
    xenrlw text,
    enrlw text,
    xenrlt text,
    enrlt text,
    xapplcn text,
    applcn text,
    xadmssn text,
    admssn text,
    xenrlft text,
    enrft text,
    xenrlpt text,
    enrpt text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.edu_stats_admissions_tests_2002
    OWNER TO postgres;

-- import data from file into newly created database table for the 2002 data
COPY economic.edu_stats_admissions_tests_2002 FROM '/nmsba/data-downloaded/nces-
admissions-and-test-scores/data/ic2002.csv' DELIMITER ',' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_2002;

-- import 2002 data into master table
INSERT INTO economic.edu_stats_admissions_tests_v01_20161025 (year, unitid, peolistr,
peo2istr, peo3istr, peo4istr, peo5istr, peo6istr, cntlaffi, pubprime, pubsecon,
relaffil, level1, level2, level3, level4, level5, level6, level7, level8, level9,
level10, level11, level12, fopna, fopna1, fopna2, accrd1, accrd2, regacrd, accrd3,
accrd4, saccr, openadmp, admcon1, admcon2, admcon3, admcon4, admcon5, admcon6,
admcon7, admcon8, appdate, xapplcnm, applcnm, xapplcnw, applcnw, xadmssnm, admssnm,
xadmssnw, admssnw, xenrlftm, enrftm, xenrlftw, enrftw, xenrlptm, enrptm, xenrlptw,
enrlptw, satactdt, xsatnum, satnum, xsatpct, satpct, xactnum, actnum, xactpct, actpct,
xsatvr25, satvr25, xsatvr75, satvr75, xsatmt25, satmt25, xsatmt75, satmt75, xactcm25,
actcm25, xactcm75, actcm75, xacten25, acten25, xacten75, acten75, xactmt25, actmt25,
xactmt75, actmt75, credits1, credits2, credits3, credits4, slo1, slo2, slo3, slo4,

```

```

slo5, slo51, slo52, slo53, slo6, slo7, slo8, slo81, slo82, slo83, slo9, yrscoll,
stusrv1, stusrv2, stusrv3, stusrv4, stusrv8, stusrv9, libfac, athassoc, assoc1,
assoc2, assoc3, assoc4, assoc5, assoc6, sport1, confno1, sport2, confno2, sport3,
confno3, sport4, confno4, insttoyr, pctpost, calsys, apfee, xappfee, applfee,
xappfeeg, applfeeg, xappfeep, applfeep, ftstu, ft_ug, ft_ftug, ft_gd, ft_fp, ptstu,
pt_ug, pt_ftug, pt_gd, tuitvary, room, xroomcap, roomcap, board, xmealswk, mealswk,
xroomamt, roomamt, xbordamt, boardamt, xrmbrdamt, rmbrdamt, xenrlm, enrlm, xenrlw,
enrlw, xenrlt, enrlt, xapplcn, applcn, xadmssn, admssn, xenrlft, enrlft, xenrlpt,
enrlpt)
SELECT
2002 as year,
unitid AS unitid,
peolistr AS peolistr,
peo2istr AS peo2istr,
peo3istr AS peo3istr,
peo4istr AS peo4istr,
peo5istr AS peo5istr,
peo6istr AS peo6istr,
cntlaffi AS cntlaffi,
pubprime AS pubprime,
pubsecon AS pubsecon,
relaffil AS relaffil,
level1 AS level1,
level2 AS level2,
level3 AS level3,
level4 AS level4,
level5 AS level5,
level6 AS level6,
level7 AS level7,
level8 AS level8,
level9 AS level9,
level10 AS level10,
level11 AS level11,
level12 AS level12,
fopna AS fopna,
fopna1 AS fopna1,
fopna2 AS fopna2,
accrd1 AS accrd1,
accrd2 AS accrd2,
regaccrd AS regaccrd,
accrd3 AS accrd3,
accrd4 AS accrd4,
saccr AS saccr,
openadmp AS openadmp,
admcon1 AS admcon1,
admcon2 AS admcon2,
admcon3 AS admcon3,
admcon4 AS admcon4,
admcon5 AS admcon5,
admcon6 AS admcon6,
admcon7 AS admcon7,
admcon8 AS admcon8,
appdate AS appdate,
xapplcnm AS xapplcnm,
applcnm AS applcnm,
xapplcnw AS xapplcnw,
applcnw AS applcnw,
xadmssnm AS xadmssnm,
admssnm AS admssnm,
xadmssnw AS xadmssnw,
admssnw AS admssnw,
xenrlftm AS xenrlftm,
enrlftm AS enrlftm,

```

xenrlftw AS xenrlftw,
enrlftw AS enrlftw,
xenrlptm AS xenrlptm,
enrlptm AS enrlptm,
xenrlptw AS xenrlptw,
enrlptw AS enrlptw,
satactdt AS satactdt,
xsatnum AS xsatnum,
satnum AS satnum,
xsatpct AS xsatpct,
satpct AS satpct,
xactnum AS xactnum,
actnum AS actnum,
xactpct AS xactpct,
actpct AS actpct,
xsatvr25 AS xsatvr25,
satvr25 AS satvr25,
xsatvr75 AS xsatvr75,
satvr75 AS satvr75,
xsatmt25 AS xsatmt25,
satmt25 AS satmt25,
xsatmt75 AS xsatmt75,
satmt75 AS satmt75,
xactcm25 AS xactcm25,
actcm25 AS actcm25,
xactcm75 AS xactcm75,
actcm75 AS actcm75,
xacten25 AS xacten25,
acten25 AS acten25,
xacten75 AS xacten75,
acten75 AS acten75,
xactmt25 AS xactmt25,
actmt25 AS actmt25,
xactmt75 AS xactmt75,
actmt75 AS actmt75,
credits1 AS credits1,
credits2 AS credits2,
credits3 AS credits3,
credits4 AS credits4,
slo1 AS slo1,
slo2 AS slo2,
slo3 AS slo3,
slo4 AS slo4,
slo5 AS slo5,
slo51 AS slo51,
slo52 AS slo52,
slo53 AS slo53,
slo6 AS slo6,
slo7 AS slo7,
slo8 AS slo8,
slo81 AS slo81,
slo82 AS slo82,
slo83 AS slo83,
slo9 AS slo9,
yrscoll AS yrscoll,
stusrv1 AS stusrv1,
stusrv2 AS stusrv2,
stusrv3 AS stusrv3,
stusrv4 AS stusrv4,
stusrv8 AS stusrv8,
stusrv9 AS stusrv9,
libfac AS libfac,
athassoc AS athassoc,

```

assoc1 AS assoc1,
assoc2 AS assoc2,
assoc3 AS assoc3,
assoc4 AS assoc4,
assoc5 AS assoc5,
assoc6 AS assoc6,
sport1 AS sport1,
confno1 AS confno1,
sport2 AS sport2,
confno2 AS confno2,
sport3 AS sport3,
confno3 AS confno3,
sport4 AS sport4,
confno4 AS confno4,
insttoyr AS insttoyr,
pctpost AS pctpost,
calsys AS calsys,
apfee AS apfee,
xappfeeu AS xappfeeu,
applfeeu AS applfeeu,
xappfeeg AS xappfeeg,
applfeeg AS applfeeg,
xappfeep AS xappfeep,
applfeep AS applfeep,
ftstu AS ftstu,
ft_ug AS ft_ug,
ft_ftug AS ft_ftug,
ft_gd AS ft_gd,
ft_fp AS ft_fp,
ptstu AS ptstu,
pt_ug AS pt_ug,
pt_ftug AS pt_ftug,
pt_gd AS pt_gd,
tuitvary AS tuitvary,
room AS room,
xroomcap AS xroomcap,
roomcap AS roomcap,
board AS board,
xmealswk AS xmealswk,
mealswk AS mealswk,
xroomamt AS xroomamt,
roomamt AS roomamt,
xbordamt AS xbordamt,
boardamt AS boardamt,
xrmbdamt AS xrmbdamt,
rmbdamt AS rmbdamt,
xenrlm AS xenrlm,
enrlm AS enrlm,
xenrlw AS xenrlw,
enrlw AS enrlw,
xenrlt AS xenrlt,
enrlt AS enrlt,
xapplcn AS xapplcn,
applcn AS applcn,
xadmsn AS xadmsn,
admsn AS admsn,
xenrlft AS xenrlft,
enrlft AS enrlft,
xenrlpt AS xenrlpt,
enrlpt AS enrlpt
FROM economic.edu_stats_admissions_tests_2002;

-- clean up tables and indexes

```

```
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_v01_20161025;

-- example query to test data integration
SELECT year, dir.unitid, instnm, addr, city, stabbr, zip, enrftm, enrftw, satnum,
actnum FROM economic.edu_stats_institute_dir_v0120161025 dir INNER JOIN
economic.edu_stats_admissions_tests_v01_20161025 ats ON dir.unitid::integer =
ats.unitid::integer WHERE dir.unitid = 187985;

-- exit PostgreSQL terminal
\q
```

14.2.3 Data from 2003

There are many file options when viewing the website, filter by All Years and select Admissions and Test Scores. Find the file named IC2003, which is for 2003.

Server Terminal Commands

```
-- create the data directory

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores/data

-- navigate to the data directory
cd /nmsba/data-downloaded/nces-admissions-and-test-scores/data

sudo wget https://nces.ed.gov/ipeds/datacenter/data/IC2003.zip

-- unzip the files
sudo unzip IC2003.zip

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.edu_stats_admissions_tests_2003;
CREATE TABLE economic.edu_stats_admissions_tests_2003
(
    unitid text,
    peolistr text,
    peo2istr text,
    peo3istr text,
    peo4istr text,
    peo5istr text,
    peo6istr text,
    cntlaffi text,
    pubprime text,
    pubsecon text,
    relaffil text,
    level1 text,
    level2 text,
    level3 text,
    level4 text,
    level5 text,
    level6 text,
    level7 text,
    level8 text,
```

level9 text,
level10 text,
level11 text,
level12 text,
fopna text,
fopna1 text,
fopna2 text,
accrd1 text,
accrd2 text,
regaccrd text,
accrd3 text,
accrd4 text,
sacchr text,
openadmp text,
admcon1 text,
admcon2 text,
admcon3 text,
admcon4 text,
admcon5 text,
admcon6 text,
admcon7 text,
admcon8 text,
appdate text,
xapplcnm text,
applcnm text,
xapplcnw text,
applcnw text,
xadmssnm text,
admssnm text,
xadmssnw text,
admssnw text,
xenrlftm text,
enrlftm text,
xenrlftw text,
enrlftw text,
xenrlptm text,
enrlptm text,
xenrlptw text,
enrlptw text,
satactdt text,
xsatnum text,
satnum text,
xsatpct text,
satpct text,
xactnum text,
actnum text,
xactpct text,
actpct text,
xsatvr25 text,
satvr25 text,
xsatvr75 text,
satvr75 text,
xsatmt25 text,
satmt25 text,
xsatmt75 text,
satmt75 text,
xactcm25 text,
actcm25 text,
xactcm75 text,
actcm75 text,
xacten25 text,
acten25 text,
xacten75 text,

acten75 text,
xactmt25 text,
actmt25 text,
xactmt75 text,
actmt75 text,
credits1 text,
credits2 text,
credits3 text,
credits4 text,
slo1 text,
slo2 text,
slo3 text,
slo4 text,
slo5 text,
slo51 text,
slo52 text,
slo53 text,
slo6 text,
slo7 text,
slo8 text,
slo81 text,
slo82 text,
slo83 text,
slo9 text,
yrscoll text,
stusrv1 text,
stusrv2 text,
stusrv3 text,
stusrv4 text,
stusrv8 text,
stusrv9 text,
libfac text,
athassoc text,
assoc1 text,
assoc2 text,
assoc3 text,
assoc4 text,
assoc5 text,
assoc6 text,
sport1 text,
confno1 text,
sport2 text,
confno2 text,
sport3 text,
confno3 text,
sport4 text,
confno4 text,
insttoyr text,
pctpost text,
calsys text,
apfee text,
xappfeeu text,
applfeeu text,
xappfeeg text,
applfeeg text,
xappfeep text,
applfeep text,
ft_ug text,
ft_ftug text,
ft_gd text,
ft_fp text,
pt_ug text,
pt_ftug text,


```

pt_gd text,
tuitvary text,
room text,
xroomcap text,
roomcap text,
board text,
xmealswk text,
mealswk text,
xroomamt text,
roomamt text,
xbordamt text,
boardamt text,
xrmbdamt text,
rmbdamt text,
xenrlm text,
enrlm text,
xenrlw text,
enrlw text,
xenrlt text,
enrlt text,
xapplcn text,
applcn text,
xadmssn text,
admssn text,
xenrlft text,
enrlft text,
xenrlpt text,
enrlpt text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.edu_stats_admissions_tests_2003
    OWNER TO postgres;

-- import data from file into newly created database table for the 2003 data
COPY economic.edu_stats_admissions_tests_2003 FROM '/nmsba/data-downloaded/nces-
admissions-and-test-scores/data/ic2003.csv' DELIMITER ',' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_2003;

-- import 2003 data into master table
INSERT INTO economic.edu_stats_admissions_tests_v01_20161025 (year, unitid, peolistr,
peo2istr, peo3istr, peo4istr, peo5istr, peo6istr, cntlaffi, pubprime, pubsecon,
relaffil, level1, level2, level3, level4, level5, level6, level7, level8, level9,
level10, level11, level12, fopna, fopna1, fopna2, accrd1, accrd2, regacrd, accrd3,
acrd4, saccr, openadmp, admcon1, admcon2, admcon3, admcon4, admcon5, admcon6,
admcon7, admcon8, appdate, xapplcnm, applcnm, xapplcnw, applcnw, xadmssnm, admssnm,
xadmssnw, admssnw, xenrlftm, enrlftm, xenrlftw, enrlftw, xenrlptm, enrlptm, xenrlptw,
enrlptw, satactdt, xsatnum, satnum, xsatpct, satpct, xactnum, actnum, xactpct, actpct,
xsatvr25, satvr25, xsatvr75, satvr75, xsatmt25, satmt25, xsatmt75, satmt75, xactcm25,
actcm25, xactcm75, actcm75, xacten25, acten25, xacten75, acten75, xactmt25, actmt25,
xactmt75, actmt75, credits1, credits2, credits3, credits4, slo1, slo2, slo3, slo4,
slo5, slo51, slo52, slo53, slo6, slo7, slo8, slo81, slo82, slo83, slo9, yrscoll,
stusrv1, stusrv2, stusrv3, stusrv4, stusrv8, stusrv9, libfac, athassoc, assoc1,
assoc2, assoc3, assoc4, assoc5, assoc6, sport1, confno1, sport2, confno2, sport3,
confno3, sport4, confno4, insttoyr, pctpost, calsys, apfee, xappfee, applfee,
xappfee, applfee, xappfee, applfee, ft_ug, ft_ftug, ft_gd, ft_fp, pt_ug, pt_ftug,
pt_gd, tuitvary, room, xroomcap, roomcap, board, xmealswk, mealswk, xroomamt, roomamt,
xbordamt, boardamt, xrmbdamt, rmbdamt, xenrlm, enrlm, xenrlw, enrlw, xenrlt, enrlt,
xapplcn, applcn, xadmssn, admssn, xenrlft, enrlft, xenrlpt, enrlpt)
SELECT

```

```

2003 as year,
unitid AS unitid,
peolistr AS peolistr,
peo2istr AS peo2istr,
peo3istr AS peo3istr,
peo4istr AS peo4istr,
peo5istr AS peo5istr,
peo6istr AS peo6istr,
cntlaffi AS cntlaffi,
pubprime AS pubprime,
pubsecon AS pubsecon,
relaffil AS relaffil,
level1 AS level1,
level2 AS level2,
level3 AS level3,
level4 AS level4,
level5 AS level5,
level6 AS level6,
level7 AS level7,
level8 AS level8,
level9 AS level9,
level10 AS level10,
level11 AS level11,
level12 AS level12,
fopna AS fopna,
fopna1 AS fopna1,
fopna2 AS fopna2,
accrd1 AS accrd1,
accrd2 AS accrd2,
regaccrd AS regaccrd,
accrd3 AS accrd3,
accrd4 AS accrd4,
saccr AS saccr,
openadmp AS openadmp,
admcon1 AS admcon1,
admcon2 AS admcon2,
admcon3 AS admcon3,
admcon4 AS admcon4,
admcon5 AS admcon5,
admcon6 AS admcon6,
admcon7 AS admcon7,
admcon8 AS admcon8,
appdate AS appdate,
xapplcnm AS xapplcnm,
applcnm AS applcnm,
xapplcnw AS xapplcnw,
applcnw AS applcnw,
xadmssnm AS xadmssnm,
admssnm AS admssnm,
xadmssnw AS xadmssnw,
admssnw AS admssnw,
xenrlftm AS xenrlftm,
enrlftm AS enrlftm,
xenrlftw AS xenrlftw,
enrlftw AS enrlftw,
xenrlptm AS xenrlptm,
enrlptm AS enrlptm,
xenrlptw AS xenrlptw,
enrlptw AS enrlptw,
satactdt AS satactdt,
xsatnum AS xsatnum,
satnum AS satnum,
xsatpct AS xsatpct,

```

```

satpct AS satpct,
xactnum AS xactnum,
actnum AS actnum,
xactpct AS xactpct,
actpct AS actpct,
xsatvr25 AS xsatvr25,
satvr25 AS satvr25,
xsatvr75 AS xsatvr75,
satvr75 AS satvr75,
xsatmt25 AS xsatmt25,
satmt25 AS satmt25,
xsatmt75 AS xsatmt75,
satmt75 AS satmt75,
xactcm25 AS xactcm25,
actcm25 AS actcm25,
xactcm75 AS xactcm75,
actcm75 AS actcm75,
xacten25 AS xacten25,
acten25 AS acten25,
xacten75 AS xacten75,
acten75 AS acten75,
xactmt25 AS xactmt25,
actmt25 AS actmt25,
xactmt75 AS xactmt75,
actmt75 AS actmt75,
credits1 AS credits1,
credits2 AS credits2,
credits3 AS credits3,
credits4 AS credits4,
slo1 AS slo1,
slo2 AS slo2,
slo3 AS slo3,
slo4 AS slo4,
slo5 AS slo5,
slo51 AS slo51,
slo52 AS slo52,
slo53 AS slo53,
slo6 AS slo6,
slo7 AS slo7,
slo8 AS slo8,
slo81 AS slo81,
slo82 AS slo82,
slo83 AS slo83,
slo9 AS slo9,
yrscoll AS yrscoll,
stusrv1 AS stusrv1,
stusrv2 AS stusrv2,
stusrv3 AS stusrv3,
stusrv4 AS stusrv4,
stusrv8 AS stusrv8,
stusrv9 AS stusrv9,
libfac AS libfac,
athassoc AS athassoc,
assoc1 AS assoc1,
assoc2 AS assoc2,
assoc3 AS assoc3,
assoc4 AS assoc4,
assoc5 AS assoc5,
assoc6 AS assoc6,
sport1 AS sport1,
confno1 AS confno1,
sport2 AS sport2,
confno2 AS confno2,

```

```

sport3 AS sport3,
confno3 AS confno3,
sport4 AS sport4,
confno4 AS confno4,
insttoyr AS insttoyr,
pctpost AS pctpost,
calsys AS calsys,
apfee AS apfee,
xappfeeu AS xappfeeu,
applfeeu AS applfeeu,
xappfeeg AS xappfeeg,
applfeeg AS applfeeg,
xappfeep AS xappfeep,
applfeep AS applfeep,
ft_ug AS ft_ug,
ft_ftug AS ft_ftug,
ft_gd AS ft_gd,
ft_fp AS ft_fp,
pt_ug AS pt_ug,
pt_ftug AS pt_ftug,
pt_gd AS pt_gd,
tuitvary AS tuitvary,
room AS room,
xroomcap AS xroomcap,
roomcap AS roomcap,
board AS board,
xmealswk AS xmealswk,
mealswk AS mealswk,
xroomamt AS xroomamt,
roomamt AS roomamt,
xbordamt AS xbordamt,
boardamt AS boardamt,
xrmbdamt AS xrmbdamt,
rmbdamt AS rmbdamt,
xenrlm AS xenrlm,
enrlm AS enrlm,
xenrlw AS xenrlw,
enrlw AS enrlw,
xenrlt AS xenrlt,
enrlt AS enrlt,
xapplcn AS xapplcn,
applcn AS applcn,
xadmsn AS xadmsn,
admssn AS admssn,
xenrlft AS xenrlft,
enrlft AS enrlft,
xenrlpt AS xenrlpt,
enrlpt AS enrlpt
FROM economic.edu_stats_admissions_tests_2003;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_v01_20161025;

-- example query to test data integration
SELECT year, dir.unitid, instnm, addr, city, stabbr, zip, enrlftm, enrlftw, satnum,
actnum FROM economic.edu_stats_institute_dir_v0120161025 dir INNER JOIN
economic.edu_stats_admissions_tests_v01_20161025 ats ON dir.unitid::integer =
ats.unitid::integer WHERE dir.unitid = 187985;

-- exit PostgreSQL terminal
\q

```

14.2.4 Data from 2004

There are many file options when viewing the website, filter by All Years and select Admissions and Test Scores. Find the file named IC2004, which is for 2004.

Server Terminal Commands

```
-- create the data directory

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores/data

-- navigate to the data directory
cd /nmsba/data-downloaded/nces-admissions-and-test-scores/data

sudo wget https://nces.ed.gov/ipeds/datacenter/data/IC2004.zip

-- unzip the files
sudo unzip IC2004.zip

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.edu_stats_admissions_tests_2004;
CREATE TABLE economic.edu_stats_admissions_tests_2004
(
    unitid text,
    peo1istr text,
    peo2istr text,
    peo3istr text,
    peo4istr text,
    peo5istr text,
    peo6istr text,
    cntlaffi text,
    pubprime text,
    pubsecon text,
    relaffil text,
    level1 text,
    level2 text,
    level3 text,
    level4 text,
    level5 text,
    level6 text,
    level7 text,
    level8 text,
    level9 text,
    level10 text,
    level11 text,
    level12 text,
    accrd1 text,
    accrd2 text,
    regacrd text,
    accrd3 text,
    accrd4 text,
    saccr text,
    openadmp text,
```

admcon1 text,
admcon2 text,
admcon3 text,
admcon4 text,
admcon5 text,
admcon6 text,
admcon7 text,
admcon8 text,
appdate text,
xapplcnm text,
applcnm text,
xapplcnw text,
applcnw text,
xadmssnm text,
admssnm text,
xadmssnw text,
admssnw text,
xenrlftm text,
enrlftm text,
xenrlftw text,
enrlftw text,
xenrlptm text,
enrlptm text,
xenrlptw text,
enrlptw text,
satactdt text,
xsatnum text,
satnum text,
xsatpct text,
satpct text,
xactnum text,
actnum text,
xactpct text,
actpct text,
xsatvr25 text,
satvr25 text,
xsatvr75 text,
satvr75 text,
xsatmt25 text,
satmt25 text,
xsatmt75 text,
satmt75 text,
xactcm25 text,
actcm25 text,
xactcm75 text,
actcm75 text,
xacten25 text,
acten25 text,
xacten75 text,
acten75 text,
xactmt25 text,
actmt25 text,
xactmt75 text,
actmt75 text,
credits1 text,
credits2 text,
credits3 text,
credits4 text,
slo1 text,
slo2 text,
slo3 text,
slo4 text,
slo5 text,

slo51 text,
slo52 text,
slo53 text,
slo6 text,
slo7 text,
slo8 text,
slo81 text,
slo82 text,
slo83 text,
slo9 text,
yrscoll text,
stusrv1 text,
stusrv2 text,
stusrv3 text,
stusrv4 text,
stusrv8 text,
stusrv9 text,
libfac text,
athassoc text,
assoc1 text,
assoc2 text,
assoc3 text,
assoc4 text,
assoc5 text,
assoc6 text,
sport1 text,
confno1 text,
sport2 text,
confno2 text,
sport3 text,
confno3 text,
sport4 text,
confno4 text,
insttoyr text,
pctpost text,
calsys text,
apfee text,
xappfeeu text,
applfeeu text,
xappfeeg text,
applfeeg text,
xappfeep text,
applfeep text,
ft_ug text,
ft_ftug text,
ft_gd text,
ft_fp text,
pt_ug text,
pt_ftug text,
pt_gd text,
pt_fp text,
tuitvary text,
room text,
xroomcap text,
roomcap text,
board text,
xmealswk text,
mealswk text,
xroomamt text,
roomamt text,
xbordamt text,
boardamt text,
xrmbdamt text,

```

    rmbrdamt text,
    alloncam text,
    xenrlm text,
    enrlm text,
    xenrlw text,
    enrlw text,
    xenrlt text,
    enrlt text,
    xapplcn text,
    applcn text,
    xadmssn text,
    admssn text,
    xenrlft text,
    enrlft text,
    xenrlpt text,
    enrlpt text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.edu_stats_admissions_tests_2004
    OWNER TO postgres;

-- import data from file into newly created database table for the 2004 data
COPY economic.edu_stats_admissions_tests_2004 FROM '/nmsba/data-downloaded/nces-
admissions-and-test-scores/data/ic2004.csv' DELIMITER ',' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_2004;

-- import 2004 data into master table
INSERT INTO economic.edu_stats_admissions_tests_v01_20161025 (year, unitid, peolistr,
peo2istr, peo3istr, peo4istr, peo5istr, peo6istr, cntlaffi, pubprime, pubsecon,
relaffil, level1, level2, level3, level4, level5, level6, level7, level8, level9,
level10, level11, level12, accrd1, accrd2, regaccrd, accrd3, accrd4, saccr, openadmp,
admcon1, admcon2, admcon3, admcon4, admcon5, admcon6, admcon7, admcon8, appdate,
xapplcnm, applcnm, xapplcnw, applcnw, xadmssnm, admssnm, xadmssnw, admssnw, xenrlftm,
enrlftm, xenrlftw, enrlftw, xenrlptm, enrlptm, xenrlptw, enrlptw, satactdt, xsatnum,
satnum, xsatpct, satpct, xactnum, actnum, xactpct, actpct, xsatvr25, satvr25,
xsatvr75, satvr75, xsatmt25, satmt25, xsatmt75, satmt75, xactcm25, actcm25, xactcm75,
actcm75, xacten25, acten25, xacten75, acten75, xactmt25, actmt25, xactmt75, actmt75,
credits1, credits2, credits3, credits4, slo1, slo2, slo3, slo4, slo5, slo51, slo52,
slo53, slo6, slo7, slo8, slo81, slo82, slo83, slo9, yrscoll, stusrv1, stusrv2,
stusrv3, stusrv4, stusrv8, stusrv9, libfac, athassoc, assoc1, assoc2, assoc3, assoc4,
assoc5, assoc6, sport1, confno1, sport2, confno2, sport3, confno3, sport4, confno4,
insttoyr, pctpost, calsys, apfee, xappfeeu, applfeeu, xappfeeg, applfeeg, xappfeep,
applfeep, ft_ug, ft_ftug, ft_gd, ft_fp, pt_ug, pt_ftug, pt_gd, pt_fp, tuitvary, room,
xroomcap, roomcap, board, xmealswk, mealswk, xroomamt, roomamt, xbordamt, boardamt,
xrmbdamt, rmbrdamt, alloncam, xenrlm, enrlm, xenrlw, enrlw, xenrlt, enrlt, xapplcn,
applcn, xadmssn, admssn, xenrlft, enrlft, xenrlpt, enrlpt)
SELECT
2004 as year,
unitid AS unitid,
peolistr AS peolistr,
peo2istr AS peo2istr,
peo3istr AS peo3istr,
peo4istr AS peo4istr,
peo5istr AS peo5istr,
peo6istr AS peo6istr,
cntlaffi AS cntlaffi,
pubprime AS pubprime,
pubsecon AS pubsecon,
relaffil AS relaffil,

```



```

level1 AS level1,
level2 AS level2,
level3 AS level3,
level4 AS level4,
level5 AS level5,
level6 AS level6,
level7 AS level7,
level8 AS level8,
level9 AS level9,
level10 AS level10,
level11 AS level11,
level12 AS level12,
accrd1 AS accrd1,
accrd2 AS accrd2,
regaccrd AS regaccrd,
accrd3 AS accrd3,
accrd4 AS accrd4,
saccr AS saccr,
openadmp AS openadmp,
admcon1 AS admcon1,
admcon2 AS admcon2,
admcon3 AS admcon3,
admcon4 AS admcon4,
admcon5 AS admcon5,
admcon6 AS admcon6,
admcon7 AS admcon7,
admcon8 AS admcon8,
appdate AS appdate,
xapplcnm AS xapplcnm,
applcnm AS applcnm,
xapplcnw AS xapplcnw,
applcnw AS applcnw,
xadmssnm AS xadmssnm,
admssnm AS admssnm,
xadmssnw AS xadmssnw,
admssnw AS admssnw,
xenrlftm AS xenrlftm,
enrlftm AS enrlftm,
xenrlftw AS xenrlftw,
enrlftw AS enrlftw,
xenrlptm AS xenrlptm,
enrlptm AS enrlptm,
xenrlptw AS xenrlptw,
enrlptw AS enrlptw,
satactdt AS satactdt,
xsatnum AS xsatnum,
satnum AS satnum,
xsatpct AS xsatpct,
satpct AS satpct,
xactnum AS xactnum,
actnum AS actnum,
xactpct AS xactpct,
actpct AS actpct,
xsatvr25 AS xsatvr25,
satvr25 AS satvr25,
xsatvr75 AS xsatvr75,
satvr75 AS satvr75,
xsatmt25 AS xsatmt25,
satmt25 AS satmt25,
xsatmt75 AS xsatmt75,
satmt75 AS satmt75,
xactcm25 AS xactcm25,
actcm25 AS actcm25,

```

```

xactcm75 AS xactcm75,
actcm75 AS actcm75,
xacten25 AS xacten25,
acten25 AS acten25,
xacten75 AS xacten75,
acten75 AS acten75,
xactmt25 AS xactmt25,
actmt25 AS actmt25,
xactmt75 AS xactmt75,
actmt75 AS actmt75,
credits1 AS credits1,
credits2 AS credits2,
credits3 AS credits3,
credits4 AS credits4,
slo1 AS slo1,
slo2 AS slo2,
slo3 AS slo3,
slo4 AS slo4,
slo5 AS slo5,
slo51 AS slo51,
slo52 AS slo52,
slo53 AS slo53,
slo6 AS slo6,
slo7 AS slo7,
slo8 AS slo8,
slo81 AS slo81,
slo82 AS slo82,
slo83 AS slo83,
slo9 AS slo9,
yrscoll AS yrscoll,
stusrv1 AS stusrv1,
stusrv2 AS stusrv2,
stusrv3 AS stusrv3,
stusrv4 AS stusrv4,
stusrv8 AS stusrv8,
stusrv9 AS stusrv9,
libfac AS libfac,
athassoc AS athassoc,
assoc1 AS assoc1,
assoc2 AS assoc2,
assoc3 AS assoc3,
assoc4 AS assoc4,
assoc5 AS assoc5,
assoc6 AS assoc6,
sport1 AS sport1,
confno1 AS confno1,
sport2 AS sport2,
confno2 AS confno2,
sport3 AS sport3,
confno3 AS confno3,
sport4 AS sport4,
confno4 AS confno4,
insttoyr AS insttoyr,
pctpost AS pctpost,
calsys AS calsys,
apfee AS apfee,
xappfeeu AS xappfeeu,
applfeeu AS applfeeu,
xappfeeg AS xappfeeg,
applfeeg AS applfeeg,
xappfeep AS xappfeep,
applfeep AS applfeep,
ft_ug AS ft_ug,

```

```

ft_ftug AS ft_ftug,
ft_gd AS ft_gd,
ft_fp AS ft_fp,
pt_ug AS pt_ug,
pt_ftug AS pt_ftug,
pt_gd AS pt_gd,
pt_fp AS pt_fp,
tuitvary AS tuitvary,
room AS room,
xroomcap AS xroomcap,
roomcap AS roomcap,
board AS board,
xmealswk AS xmealswk,
mealswk AS mealswk,
xroomamt AS xroomamt,
roomamt AS roomamt,
xbordamt AS xbordamt,
boardamt AS boardamt,
xrmbdamt AS xrmbdamt,
rmbdamt AS rmbdamt,
alloncam AS alloncam,
xenrlm AS xenrlm,
enrlm AS enrlm,
xenrlw AS xenrlw,
enrlw AS enrlw,
xenrld AS xenrld,
enrld AS enrld,
xapplcn AS xapplcn,
applcn AS applcn,
xadmsn AS xadmsn,
admsn AS admsn,
xenrlft AS xenrlft,
enrlft AS enrlft,
xenrlpt AS xenrlpt,
enrlpt AS enrlpt
FROM economic.edu_stats_admissions_tests_2004;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_v01_20161025;

-- example query to test data integration
SELECT year, dir.unitid, instnm, addr, city, stabbr, zip, enrldm, enrldw, satnum,
actnum FROM economic.edu_stats_institute_dir_v0120161025 dir INNER JOIN
economic.edu_stats_admissions_tests_v01_20161025 ats ON dir.unitid::integer =
ats.unitid::integer WHERE dir.unitid = 187985;

-- exit PostgreSQL terminal
\q

```

14.2.5 Data from 2005

There are many file options when viewing the website, filter by All Years and select Admissions and Test Scores. Find the file named IC2005, which is for 2005.

Server Terminal Commands

```

-- create the data directory

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores/data

```

```
-- navigate to the data directory
cd /nmsba/data-downloaded/nces-admissions-and-test-scores/data

sudo wget https://nces.ed.gov/ipeds/datacenter/data/IC2005.zip

-- unzip the files
sudo unzip IC2005.zip

-- login to the PostgreSQL command line
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE economic.edu_stats_admissions_tests_2005;
CREATE TABLE economic.edu_stats_admissions_tests_2005
(
    unitid text,
    peolistr text,
    peo2istr text,
    peo3istr text,
    peo4istr text,
    peo5istr text,
    peo6istr text,
    cntlaffi text,
    pubprime text,
    pubsecon text,
    relaffil text,
    level1 text,
    level2 text,
    level3 text,
    level4 text,
    level5 text,
    level6 text,
    level7 text,
    level8 text,
    level9 text,
    level10 text,
    level11 text,
    level12 text,
    openadmp text,
    admcon1 text,
    admcon2 text,
    admcon3 text,
    admcon4 text,
    admcon5 text,
    admcon6 text,
    admcon7 text,
    admcon8 text,
    admcon9 text,
    appdate text,
    xapplcnm text,
    applcnm text,
    xapplcnw text,
    applcnw text,
    xadmssnm text,
    admssnm text,
    xadmssnw text,
    admssnw text,
    xenrlftm text,
```

enrlftm text,
xenrlftw text,
enrlftw text,
xenrlptm text,
enrlptm text,
xenrlptw text,
enrlptw text,
satactdt text,
xsatnum text,
satnum text,
xsatpct text,
satpct text,
xactnum text,
actnum text,
xactpct text,
actpct text,
xsatvr25 text,
satvr25 text,
xsatvr75 text,
satvr75 text,
xsatmt25 text,
satmt25 text,
xsatmt75 text,
satmt75 text,
xactcm25 text,
actcm25 text,
xactcm75 text,
actcm75 text,
xacten25 text,
acten25 text,
xacten75 text,
acten75 text,
xactmt25 text,
actmt25 text,
xactmt75 text,
actmt75 text,
credits1 text,
credits2 text,
credits3 text,
credits4 text,
slo3 text,
slo5 text,
slo51 text,
slo52 text,
slo53 text,
slo6 text,
slo7 text,
slo8 text,
slo81 text,
slo82 text,
slo83 text,
slo9 text,
yrscoll text,
stusrv1 text,
stusrv2 text,
stusrv3 text,
stusrv4 text,
stusrv8 text,
stusrv9 text,
libfac text,
athassoc text,
assoc1 text,
assoc2 text,

```

    assoc3 text,
    assoc4 text,
    assoc5 text,
    assoc6 text,
    sport1 text,
    confno1 text,
    sport2 text,
    confno2 text,
    sport3 text,
    confno3 text,
    sport4 text,
    confno4 text,
    pctpost text,
    calsys text,
    apfee text,
    xappfeeu text,
    applfeeu text,
    xappfeeg text,
    applfeeg text,
    xappfeep text,
    applfeep text,
    ft_ug text,
    ft_ftug text,
    ft_gd text,
    ft_fp text,
    pt_ug text,
    pt_ftug text,
    pt_gd text,
    pt_fp text,
    tuitvary text,
    room text,
    xroomcap text,
    roomcap text,
    board text,
    xmealswk text,
    mealswk text,
    xroomamt text,
    roomamt text,
    xbordamt text,
    boardamt text,
    xrmbdamt text,
    rmbrdamt text,
    alloncam text,
    xenrlm text,
    enrlm text,
    xenrlw text,
    enrlw text,
    xenrlt text,
    enrlt text,
    xapplcn text,
    applcn text,
    xadmssn text,
    admssn text,
    xenrlft text,
    enrleft text,
    xenrlpt text,
    enrleft text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.edu_stats_admissions_tests_2005
    OWNER TO postgres;

```

```

-- import data from file into newly created database table for the 2005 data
COPY economic.edu_stats_admissions_tests_2005 FROM '/nmsba/data-downloaded/nces-
admissions-and-test-scores/data/ic2005.csv' DELIMITER ',' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_2005;

-- import 2005 data into master table
INSERT INTO economic.edu_stats_admissions_tests_v01_20161025 (year, unitid, peolistr,
peo2istr, peo3istr, peo4istr, peo5istr, peo6istr, cntlaffi, pubprime, pubsecon,
relaffil, level1, level2, level3, level4, level5, level6, level7, level8, level9,
level10, level11, level12, openadmp, admcon1, admcon2, admcon3, admcon4, admcon5,
admcon6, admcon7, admcon8, admcon9, appdate, xapplcnm, applcnm, xapplcnw, applcnw,
xadmssnm, admssnm, xadmssnw, admssnw, xenrlftm, enrlftm, xenrlftw, enrlftw, xenrlptm,
enrlptm, xenrlptw, enrlptw, satactdt, xsatnum, satnum, xsatpct, satpct, xactnum,
actnum, xactpct, actpct, xsatvr25, satvr25, xsatvr75, satvr75, xsatmt25, satmt25,
xsatmt75, satmt75, xactcm25, actcm25, xactcm75, actcm75, xacten25, acten25, xacten75,
acten75, xactmt25, actmt25, xactmt75, actmt75, credits1, credits2, credits3, credits4,
slo3, slo5, slo51, slo52, slo53, slo6, slo7, slo8, slo81, slo82, slo83, slo9, yrscoll,
stusrv1, stusrv2, stusrv3, stusrv4, stusrv8, stusrv9, libfac, athassoc, assoc1,
assoc2, assoc3, assoc4, assoc5, assoc6, sport1, confno1, sport2, confno2, sport3,
confno3, sport4, confno4, pctpost, calsys, apfee, xappfee, applfee, xappfee,
applfee, xappfee, applfee, ft_ug, ft_ftug, ft_gd, ft_fp, pt_ug, pt_ftug, pt_gd,
pt_fp, tuitvary, room, xroomcap, roomcap, board, xmealswk, mealswk, xroomamt, roomamt,
xbordamt, boardamt, xrmdbamt, rmbrdamt, alloncam, xenrlm, enrlm, xenrlw, enrlw,
xenrlt, enrlt, xapplcn, applcn, xadmssn, admssn, xenrlft, enrlft, xenrlpt, enrlpt)
SELECT
2005 as year,
unitid AS unitid,
peolistr AS peolistr,
peo2istr AS peo2istr,
peo3istr AS peo3istr,
peo4istr AS peo4istr,
peo5istr AS peo5istr,
peo6istr AS peo6istr,
cntlaffi AS cntlaffi,
pubprime AS pubprime,
pubsecon AS pubsecon,
relaffil AS relaffil,
level1 AS level1,
level2 AS level2,
level3 AS level3,
level4 AS level4,
level5 AS level5,
level6 AS level6,
level7 AS level7,
level8 AS level8,
level9 AS level9,
level10 AS level10,
level11 AS level11,
level12 AS level12,
openadmp AS openadmp,
admcon1 AS admcon1,
admcon2 AS admcon2,
admcon3 AS admcon3,
admcon4 AS admcon4,
admcon5 AS admcon5,
admcon6 AS admcon6,
admcon7 AS admcon7,
admcon8 AS admcon8,
admcon9 AS admcon9,
appdate AS appdate,

```

xapplcnm AS xapplcnm,
applcnm AS applcnm,
xapplcnw AS xapplcnw,
applcnw AS applcnw,
xadmssnm AS xadmssnm,
admssnm AS admssnm,
xadmssnw AS xadmssnw,
admssnw AS admssnw,
xenrlftm AS xenrlftm,
enrlftm AS enrlftm,
xenrlftw AS xenrlftw,
enrlftw AS enrlftw,
xenrlptm AS xenrlptm,
enrlptm AS enrlptm,
xenrlptw AS xenrlptw,
enrlptw AS enrlptw,
satactdt AS satactdt,
xsatnum AS xsatnum,
satnum AS satnum,
xsatpct AS xsatpct,
satpct AS satpct,
xactnum AS xactnum,
actnum AS actnum,
xactpct AS xactpct,
actpct AS actpct,
xsatvr25 AS xsatvr25,
satvr25 AS satvr25,
xsatvr75 AS xsatvr75,
satvr75 AS satvr75,
xsatmt25 AS xsatmt25,
satmt25 AS satmt25,
xsatmt75 AS xsatmt75,
satmt75 AS satmt75,
xactcm25 AS xactcm25,
actcm25 AS actcm25,
xactcm75 AS xactcm75,
actcm75 AS actcm75,
xacten25 AS xacten25,
acten25 AS acten25,
xacten75 AS xacten75,
acten75 AS acten75,
xactmt25 AS xactmt25,
actmt25 AS actmt25,
xactmt75 AS xactmt75,
actmt75 AS actmt75,
credits1 AS credits1,
credits2 AS credits2,
credits3 AS credits3,
credits4 AS credits4,
slo3 AS slo3,
slo5 AS slo5,
slo51 AS slo51,
slo52 AS slo52,
slo53 AS slo53,
slo6 AS slo6,
slo7 AS slo7,
slo8 AS slo8,
slo81 AS slo81,
slo82 AS slo82,
slo83 AS slo83,
slo9 AS slo9,
yrscoll AS yrscoll,
stusrv1 AS stusrv1,


```

stusrv2 AS stusrv2,
stusrv3 AS stusrv3,
stusrv4 AS stusrv4,
stusrv8 AS stusrv8,
stusrv9 AS stusrv9,
libfac AS libfac,
athassoc AS athassoc,
assoc1 AS assoc1,
assoc2 AS assoc2,
assoc3 AS assoc3,
assoc4 AS assoc4,
assoc5 AS assoc5,
assoc6 AS assoc6,
sport1 AS sport1,
confno1 AS confno1,
sport2 AS sport2,
confno2 AS confno2,
sport3 AS sport3,
confno3 AS confno3,
sport4 AS sport4,
confno4 AS confno4,
pctpost AS pctpost,
calsys AS calsys,
apfee AS apfee,
xappfeeu AS xappfeeu,
applfeeu AS applfeeu,
xappfeeg AS xappfeeg,
applfeeg AS applfeeg,
xappfeep AS xappfeep,
applfeep AS applfeep,
ft_ug AS ft_ug,
ft_ftug AS ft_ftug,
ft_gd AS ft_gd,
ft_fp AS ft_fp,
pt_ug AS pt_ug,
pt_ftug AS pt_ftug,
pt_gd AS pt_gd,
pt_fp AS pt_fp,
tuitvary AS tuitvary,
room AS room,
xroomcap AS xroomcap,
roomcap AS roomcap,
board AS board,
xmealswk AS xmealswk,
mealswk AS mealswk,
xroomamt AS xroomamt,
roomamt AS roomamt,
xbordamt AS xbordamt,
boardamt AS boardamt,
xrmbdamt AS xrmbdamt,
rmbordamt AS rmbordamt,
alloncam AS alloncam,
xenrlm AS xenrlm,
enrlm AS enrlm,
xenrlw AS xenrlw,
enrlw AS enrlw,
xenrlt AS xenrlt,
enrlt AS enrlt,
xapplcn AS xapplcn,
applcn AS applcn,
xadmsn AS xadmsn,
admssn AS admssn,
xenrlft AS xenrlft,

```

```

enrlft AS enrlft,
xenrlpt AS xenrlpt,
enrlpt AS enrlpt
FROM economic.edu_stats_admissions_tests_2005;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_v01_20161025;

-- example query to test data integration
SELECT year, dir.unitid, instnm, addr, city, stabbr, zip, enrlftm, enrlftw, satnum,
actnum FROM economic.edu_stats_institute_dir_v0120161025 dir INNER JOIN
economic.edu_stats_admissions_tests_v01_20161025 ats ON dir.unitid::integer =
ats.unitid::integer WHERE dir.unitid = 187985;

-- exit PostgreSQL terminal
\q

```

14.2.6 Data from 2006

There are many file options when viewing the website, filter by All Years and select Admissions and Test Scores. Find the file named IC2006, which is for 2006.

Server Terminal Commands

```

-- create the data directory

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores

sudo mkdir /nmsba/data-downloaded/nces-admissions-and-test-scores/data

-- navigate to the data directory
cd /nmsba/data-downloaded/nces-admissions-and-test-scores/data

sudo wget https://nces.ed.gov/ipeds/datacenter/data/IC2006.zip

-- unzip the files
sudo unzip IC2006.zip

-- login to the PostgreSQL command line
sudo -u postgres psql postgres

```

PostgreSQL Terminal Commands

```

-- connect to the raw database
\c raw

-- DROP TABLE economic.edu_stats_admissions_tests_2006;
CREATE TABLE economic.edu_stats_admissions_tests_2006
(
    unitid text,
    peolistr text,
    peo2istr text,
    peo3istr text,
    peo4istr text,
    peo5istr text,
    peo6istr text,
    cntlaffi text,
    pubprime text,
    pubsecon text,
    relaffil text,
    levell text,

```

level2 text,
level3 text,
level4 text,
level5 text,
level6 text,
level7 text,
level8 text,
level9 text,
level10 text,
level11 text,
level12 text,
openadmp text,
admcon1 text,
admcon2 text,
admcon3 text,
admcon4 text,
admcon5 text,
admcon6 text,
admcon7 text,
admcon8 text,
admcon9 text,
appdate text,
xapplcnm text,
applcnm text,
xapplcnw text,
applcnw text,
xadmssnm text,
admssnm text,
xadmssnw text,
admssnw text,
xenrlftm text,
enrlftm text,
xenrlftw text,
enrlftw text,
xenrlptm text,
enrlptm text,
xenrlptw text,
enrlptw text,
satactdt text,
xsatnum text,
satnum text,
xsatpct text,
satpct text,
xactnum text,
actnum text,
xactpct text,
actpct text,
xsatvr25 text,
satvr25 text,
xsatvr75 text,
satvr75 text,
xsatmt25 text,
satmt25 text,
xsatmt75 text,
satmt75 text,
xsatwr25 text,
satwr25 text,
xsatwr75 text,
satwr75 text,
xactcm25 text,
actcm25 text,
xactcm75 text,
actcm75 text,

xacten25 text,
acten25 text,
xacten75 text,
acten75 text,
xactmt25 text,
actmt25 text,
xactmt75 text,
actmt75 text,
credits1 text,
credits2 text,
credits3 text,
credits4 text,
slo3 text,
slo5 text,
slo51 text,
slo52 text,
slo53 text,
slo6 text,
slo7 text,
slo8 text,
slo81 text,
slo82 text,
slo83 text,
slo9 text,
yrscoll text,
stusrv1 text,
stusrv2 text,
stusrv3 text,
stusrv4 text,
stusrv8 text,
stusrv9 text,
libfac text,
athassoc text,
assoc1 text,
assoc2 text,
assoc3 text,
assoc4 text,
assoc5 text,
assoc6 text,
sport1 text,
confno1 text,
sport2 text,
confno2 text,
sport3 text,
confno3 text,
sport4 text,
confno4 text,
pctpost text,
calsys text,
apfee text,
xappfeeu text,
applfeeu text,
xappfeeg text,
applfeeg text,
xappfeep text,
applfeep text,
ft_ug text,
ft_ftug text,
ft_gd text,
ft_fp text,
pt_ug text,
pt_ftug text,
pt_gd text,

```

pt_fp text,
tuitvary text,
room text,
xroomcap text,
roomcap text,
board text,
xmealswk text,
mealswk text,
xroomamt text,
roomamt text,
xbordamt text,
boardamt text,
xrmbdamt text,
rmbdamt text,
alloncam text,
xenrlm text,
enrlm text,
xenrlw text,
enrlw text,
xenrlt text,
enrlt text,
xapplcn text,
applcn text,
xadmsn text,
admsn text,
xenrlft text,
enrlft text,
xenrlpt text,
enrlpt text
)
WITH (
    OIDS=FALSE
);
ALTER TABLE economic.edu_stats_admissions_tests_2006
    OWNER TO postgres;

-- import data from file into newly created database table for the 2006 data
COPY economic.edu_stats_admissions_tests_2006 FROM '/nmsba/data-downloaded/nces-
admissions-and-test-scores/data/ic2006.csv' DELIMITER ',' CSV HEADER;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_2006;

-- import 2006 data into master table
INSERT INTO economic.edu_stats_admissions_tests_v01_20161025 (year, unitid, peolistr,
peo2istr, peo3istr, peo4istr, peo5istr, peo6istr, cntlaffi, pubprime, pubsecon,
relaffil, level1, level2, level3, level4, level5, level6, level7, level8, level9,
level10, level11, level12, openadmp, admcon1, admcon2, admcon3, admcon4, admcon5,
admcon6, admcon7, admcon8, admcon9, appdate, xapplcnm, applcnm, xapplcnw, applcnw,
xadmsnm, admsnm, xadmsnw, admssnw, xenrlftm, enrlftm, xenrlftw, enrlftw, xenrlptm,
enrlptm, xenrlptw, enrlptw, satactdt, xsatnum, satnum, xsatpct, satpct, xactnum,
actnum, xactpct, actpct, xsatvr25, satvr25, xsatvr75, satvr75, xsatmt25, satmt25,
xsatmt75, satmt75, xsatwr25, satwr25, xsatwr75, satwr75, xactcm25, actcm25, xactcm75,
actcm75, xacten25, acten25, xacten75, acten75, xactmt25, actmt25, xactmt75, actmt75,
credits1, credits2, credits3, credits4, slo3, slo5, slo51, slo52, slo53, slo6, slo7,
slo8, slo81, slo82, slo83, slo9, yrscoll, stusrv1, stusrv2, stusrv3, stusrv4, stusrv8,
stusrv9, libfac, athassoc, assoc1, assoc2, assoc3, assoc4, assoc5, assoc6, sport1,
confno1, sport2, confno2, sport3, confno3, sport4, confno4, pctpost, calsys, apfee,
xappfee, applfee, xappfee, applfee, xappfee, applfee, ft_ug, ft_ftug, ft_gd,
ft_fp, pt_ug, pt_ftug, pt_gd, pt_fp, tuitvary, room, xroomcap, roomcap, board,
xmealswk, mealswk, xroomamt, roomamt, xbordamt, boardamt, xrmbdamt, rmbdamt,
alloncam, xenrlm, enrlm, xenrlw, enrlw, xenrlt, enrlt, xapplcn, applcn, xadmsn,
admsn, xenrlft, enrlft, xenrlpt, enrlpt)

```

```

SELECT
2006 as year,
unitid AS unitid,
peolistr AS peolistr,
peo2istr AS peo2istr,
peo3istr AS peo3istr,
peo4istr AS peo4istr,
peo5istr AS peo5istr,
peo6istr AS peo6istr,
cntlaffi AS cntlaffi,
pubprime AS pubprime,
pubsecon AS pubsecon,
relaffil AS relaffil,
level1 AS level1,
level2 AS level2,
level3 AS level3,
level4 AS level4,
level5 AS level5,
level6 AS level6,
level7 AS level7,
level8 AS level8,
level9 AS level9,
level10 AS level10,
level11 AS level11,
level12 AS level12,
openadmp AS openadmp,
admcon1 AS admcon1,
admcon2 AS admcon2,
admcon3 AS admcon3,
admcon4 AS admcon4,
admcon5 AS admcon5,
admcon6 AS admcon6,
admcon7 AS admcon7,
admcon8 AS admcon8,
admcon9 AS admcon9,
appdate AS appdate,
xapplcnm AS xapplcnm,
applcnm AS applcnm,
xapplcnw AS xapplcnw,
applcnw AS applcnw,
xadmsnm AS xadmsnm,
admssnm AS admssnm,
xadmsnw AS xadmsnw,
admssnw AS admssnw,
xenrlftm AS xenrlftm,
enrlftm AS enrlftm,
xenrlftw AS xenrlftw,
enrlftw AS enrlftw,
xenrlptm AS xenrlptm,
enrlptm AS enrlptm,
xenrlptw AS xenrlptw,
enrlptw AS enrlptw,
satactdt AS satactdt,
xsatnum AS xsatnum,
satnum AS satnum,
xsatpct AS xsatpct,
satpct AS satpct,
xactnum AS xactnum,
actnum AS actnum,
xactpct AS xactpct,
actpct AS actpct,
xsatvr25 AS xsatvr25,
satvr25 AS satvr25,

```

```
xsatvr75 AS xsatvr75,  
satvr75 AS satvr75,  
xsatmt25 AS xsatmt25,  
satmt25 AS satmt25,  
xsatmt75 AS xsatmt75,  
satmt75 AS satmt75,  
xsatwr25 AS xsatwr25,  
satwr25 AS satwr25,  
xsatwr75 AS xsatwr75,  
satwr75 AS satwr75,  
xactcm25 AS xactcm25,  
actcm25 AS actcm25,  
xactcm75 AS xactcm75,  
actcm75 AS actcm75,  
xacten25 AS xacten25,  
acten25 AS acten25,  
xacten75 AS xacten75,  
acten75 AS acten75,  
xactmt25 AS xactmt25,  
actmt25 AS actmt25,  
xactmt75 AS xactmt75,  
actmt75 AS actmt75,  
credits1 AS credits1,  
credits2 AS credits2,  
credits3 AS credits3,  
credits4 AS credits4,  
slo3 AS slo3,  
slo5 AS slo5,  
slo51 AS slo51,  
slo52 AS slo52,  
slo53 AS slo53,  
slo6 AS slo6,  
slo7 AS slo7,  
slo8 AS slo8,  
slo81 AS slo81,  
slo82 AS slo82,  
slo83 AS slo83,  
slo9 AS slo9,  
yrscoll AS yrscoll,  
stusrv1 AS stusrv1,  
stusrv2 AS stusrv2,  
stusrv3 AS stusrv3,  
stusrv4 AS stusrv4,  
stusrv8 AS stusrv8,  
stusrv9 AS stusrv9,  
libfac AS libfac,  
athassoc AS athassoc,  
assoc1 AS assoc1,  
assoc2 AS assoc2,  
assoc3 AS assoc3,  
assoc4 AS assoc4,  
assoc5 AS assoc5,  
assoc6 AS assoc6,  
sport1 AS sport1,  
confno1 AS confno1,  
sport2 AS sport2,  
confno2 AS confno2,  
sport3 AS sport3,  
confno3 AS confno3,  
sport4 AS sport4,  
confno4 AS confno4,  
pctpost AS pctpost,  
calsys AS calsys,
```

```

apfee AS apfee,
xappfeeu AS xappfeeu,
applfeeu AS applfeeu,
xappfeeg AS xappfeeg,
applfeeg AS applfeeg,
xappfeep AS xappfeep,
applfeep AS applfeep,
ft_ug AS ft_ug,
ft_ftug AS ft_ftug,
ft_gd AS ft_gd,
ft_fp AS ft_fp,
pt_ug AS pt_ug,
pt_ftug AS pt_ftug,
pt_gd AS pt_gd,
pt_fp AS pt_fp,
tuitvary AS tuitvary,
room AS room,
xroomcap AS xroomcap,
roomcap AS roomcap,
board AS board,
xmealswk AS xmealswk,
mealswk AS mealswk,
xroomamt AS xroomamt,
roomamt AS roomamt,
xbordamt AS xbordamt,
boardamt AS boardamt,
xrmbdamt AS xrmbdamt,
rmbdamt AS rmbdamt,
alloncam AS alloncam,
xenrlm AS xenrlm,
enrlm AS enrlm,
xenrlw AS xenrlw,
enrlw AS enrlw,
xenrlt AS xenrlt,
enrlt AS enrlt,
xapplcn AS xapplcn,
applcn AS applcn,
xadmsn AS xadmsn,
admssn AS admssn,
xenrlft AS xenrlft,
enrlft AS enrlft,
xenrlpt AS xenrlpt,
enrlpt AS enrlpt
FROM economic.edu_stats_admissions_tests_2006;

-- clean up tables and indexes
vacuum (verbose, analyze) economic.edu_stats_admissions_tests_v01_20161025;

-- example query to test data integration
SELECT year, dir.unitid, instnm, addr, city, stabbr, zip, enrlftm, enrlftw, satnum,
actnum FROM economic.edu_stats_institute_dir_v0120161025 dir INNER JOIN
economic.edu_stats_admissions_tests_v01_20161025 ats ON dir.unitid::integer =
ats.unitid::integer WHERE dir.unitid = 187985;

-- exit PostgreSQL terminal
\q

```


15 CENSUS DATA TUTORIAL

15.1 Data description

The American Community Service (ACS) provides detailed information about people living in the United States. Their data sets are numerous and the number of variables are large. Due to the large size of their data files, ACS provides data in several different formats, and the tools to search and read the data. We started studying the data by recreating the AWS directory with microdata samples (PUMS) for all population variables obtained from the census in 2010. We downloaded the data onto the AWS server by State. However, it is worth noting that ACS also offers the data in “giant file” format, which has all data in one file. ACS provides detailed documentation describing 2010 PUMS data: <https://www.census.gov/programs-surveys/acs/technical-documentation/pums/documentation.2010.html>

To recreate the ACS PUMS directory of data on the AWS server, follow the tutorial below.

Server Terminal Commands

```
-- create the data storage directory
sudo mkdir /nmsba/data-downloaded/census2010pums

-- navigate to the newly created directory
cd /nmsba/data-downloaded/census2010pums

-- get Data:
sudo wget -r
ftp://ftp2.census.gov/acs2010_1yr/summaryfile/2010_ACSSF_By_State_All_Tables/*. * -o
log

-- Log in to PostgreSQL:
sudo -u postgres psql postgres
```

PostgreSQL Terminal Commands

```
-- connect to the raw database
\c raw

-- DROP TABLE census2010pums;
CREATE TABLE census.census2010pums
(
    RT varchar(3),
    SERIALNO varchar(10),
    SPORDER varchar(10),
    PUMA varchar(10),
    ST varchar(10),
    ADJINC varchar(10),
    PWGTP varchar(10),
    AGEP varchar(10),
    CIT varchar(10),
    CITWP varchar(10),
    COW varchar(10),
    DDRS varchar(10),
    DEAR varchar(10),
    DEYE varchar(10),
    DOUT varchar(10),
    DPHY varchar(10),
    DRAT varchar(10),
```

DRATX varchar(10),
DREM varchar(10),
ENG varchar(10),
FER varchar(10),
GCL varchar(10),
GCM varchar(10),
GCR varchar(10),
HINS1 varchar(10),
HINS2 varchar(10),
HINS3 varchar(10),
HINS4 varchar(10),
HINS5 varchar(10),
HINS6 varchar(10),
HINS7 varchar(10),
INTP varchar(10),
JWMNP varchar(10),
JWRIP varchar(10),
JWTR varchar(10),
LANX varchar(10),
MAR varchar(10),
MARHD varchar(10),
MARHM varchar(10),
MARHT varchar(10),
MARHW varchar(10),
MARHYP varchar(10),
MIG varchar(10),
MIL varchar(10),
MLPA varchar(10),
MLPB varchar(10),
MLPC varchar(10),
MLPD varchar(10),
MLPE varchar(10),
MLPF varchar(10),
MLPG varchar(10),
MLPH varchar(10),
MLPI varchar(10),
MLPJ varchar(10),
MLPK varchar(10),
NWAB varchar(10),
NWAV varchar(10),
NWLA varchar(10),
NWLK varchar(10),
NWRE varchar(10),
OIP varchar(10),
PAP varchar(10),
RELP varchar(10),
RETP varchar(10),
SCH varchar(10),
SCHG varchar(10),
SCHL varchar(10),
SEMP varchar(10),
SEX varchar(10),
SSIP varchar(10),
SSP varchar(10),
WAGP varchar(10),
WKHP varchar(10),
WKL varchar(10),
WKW varchar(10),
WRK varchar(10),
YOEP varchar(10),
ANC varchar(10),
ANC1P varchar(10),
ANC2P varchar(10),

DECADE varchar(10),
 DIS varchar(10),
 DRIVESP varchar(10),
 ESP varchar(10),
 ESR varchar(10),
 FOD1P varchar(10),
 FOD2P varchar(10),
 HICOV varchar(10),
 HISP varchar(10),
 INDP varchar(10),
 JWAP varchar(10),
 JWDP varchar(10),
 LANP varchar(10),
 MIGPUMA varchar(10),
 MIGSP varchar(10),
 MSP varchar(10),
 NAICSF varchar(10),
 NATIVITY varchar(10),
 NOP varchar(10),
 OC varchar(10),
 OCCP varchar(10),
 PAOC varchar(10),
 PERNP varchar(10),
 PINCP varchar(10),
 POBP varchar(10),
 POVPIP varchar(10),
 POWPUMA varchar(10),
 POWSP varchar(10),
 PRIVCOV varchar(10),
 PUBCOV varchar(10),
 QTRBIR varchar(10),
 RAC1P varchar(10),
 RAC2P varchar(10),
 RAC3P varchar(10),
 RACAIAN varchar(10),
 RACASN varchar(10),
 RACBLK varchar(10),
 RACNHPI varchar(10),
 RACNUM varchar(10),
 RACSOR varchar(10),
 RACWHT varchar(10),
 RC varchar(10),
 SCIENGP varchar(10),
 SCIENGRLP varchar(10),
 SFN varchar(10),
 SFR varchar(10),
 SOCP varchar(10),
 VPS varchar(10),
 WAOB varchar(10),
 FAGEP varchar(10),
 FANCP varchar(10),
 FCITP varchar(10),
 FCITWP varchar(10),
 FCOWP varchar(10),
 FDDRSP varchar(10),
 FDEARP varchar(10),
 FDEYEP varchar(10),
 FDOUTP varchar(10),
 FDPHP varchar(10),
 FDRATP varchar(10),
 FDRATXP varchar(10),
 FDREMP varchar(10),
 FENGP varchar(10),

```

FESRP varchar(10),
FFERP varchar(10),
FFODP varchar(10),
FGCLP varchar(10),
FGCMP varchar(10),
FGCRP varchar(10),
FHINS1P varchar(10),
FHINS2P varchar(10),
FHINS3C varchar(10),
FHINS3P varchar(10),
FHINS4C varchar(10),
FHINS4P varchar(10),
FHINS5C varchar(10),
FHINS5P varchar(10),
FHINS6P varchar(10),
FHINS7P varchar(10),
FHISP varchar(10),
FINDP varchar(10),
FINTP varchar(10),
FJWDP varchar(10),
FJWMNP varchar(10),
FJWRIP varchar(10),
FJWTRP varchar(10),
FLANP varchar(10),
FLANXP varchar(10),
FMARHDP varchar(10),
FMARHMP varchar(10),
FMARHTP varchar(10),
FMARHWP varchar(10),
FMARHYP varchar(10),
FMARP varchar(10),
FMIGP varchar(10),
FMIGSP varchar(10),
FMILPP varchar(10),
FMILSP varchar(10),
FOCCP varchar(10),
FOIP varchar(10),
FPAP varchar(10),
FPOBP varchar(10),
FPOWSP varchar(10),
FRACP varchar(10),
FRELP varchar(10),
FRETP varchar(10),
FSCHGP varchar(10),
FSCHLP varchar(10),
FSCHP varchar(10),
FSEMP varchar(10),
FSEXP varchar(10),
FSSIP varchar(10),
FSSP varchar(10),
FWAGP varchar(10),
FWKHP varchar(10),
FWKLP varchar(10),
FWKWP varchar(10),
FWRKP varchar(10),
FYOEP varchar(10),
pwgtp1 varchar(10),
pwgtp2 varchar(10),
pwgtp3 varchar(10),
pwgtp4 varchar(10),
pwgtp5 varchar(10),
pwgtp6 varchar(10),
pwgtp7 varchar(10),

```

```
pwgtp8 varchar(10),
pwgtp9 varchar(10),
pwgtp10 varchar(10),
pwgtp11 varchar(10),
pwgtp12 varchar(10),
pwgtp13 varchar(10),
pwgtp14 varchar(10),
pwgtp15 varchar(10),
pwgtp16 varchar(10),
pwgtp17 varchar(10),
pwgtp18 varchar(10),
pwgtp19 varchar(10),
pwgtp20 varchar(10),
pwgtp21 varchar(10),
pwgtp22 varchar(10),
pwgtp23 varchar(10),
pwgtp24 varchar(10),
pwgtp25 varchar(10),
pwgtp26 varchar(10),
pwgtp27 varchar(10),
pwgtp28 varchar(10),
pwgtp29 varchar(10),
pwgtp30 varchar(10),
pwgtp31 varchar(10),
pwgtp32 varchar(10),
pwgtp33 varchar(10),
pwgtp34 varchar(10),
pwgtp35 varchar(10),
pwgtp36 varchar(10),
pwgtp37 varchar(10),
pwgtp38 varchar(10),
pwgtp39 varchar(10),
pwgtp40 varchar(10),
pwgtp41 varchar(10),
pwgtp42 varchar(10),
pwgtp43 varchar(10),
pwgtp44 varchar(10),
pwgtp45 varchar(10),
pwgtp46 varchar(10),
pwgtp47 varchar(10),
pwgtp48 varchar(10),
pwgtp49 varchar(10),
pwgtp50 varchar(10),
pwgtp51 varchar(10),
pwgtp52 varchar(10),
pwgtp53 varchar(10),
pwgtp54 varchar(10),
pwgtp55 varchar(10),
pwgtp56 varchar(10),
pwgtp57 varchar(10),
pwgtp58 varchar(10),
pwgtp59 varchar(10),
pwgtp60 varchar(10),
pwgtp61 varchar(10),
pwgtp62 varchar(10),
pwgtp63 varchar(10),
pwgtp64 varchar(10),
pwgtp65 varchar(10),
pwgtp66 varchar(10),
pwgtp67 varchar(10),
pwgtp68 varchar(10),
pwgtp69 varchar(10),
pwgtp70 varchar(10),
```

```

    pwgtp71 varchar(10),
    pwgtp72 varchar(10),
    pwgtp73 varchar(10),
    pwgtp74 varchar(10),
    pwgtp75 varchar(10),
    pwgtp76 varchar(10),
    pwgtp77 varchar(10),
    pwgtp78 varchar(10),
    pwgtp79 varchar(10),
    pwgtp80 varchar(10)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE census2010pums OWNER TO postgres;

-- import census information into the newly created table
COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pak.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pal.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10par.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10paz.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pca.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pco.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pct.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pdc.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pde.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pfl.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pga.csv' DELIMITER ',' CSV HEADER;

```

[illegible]

```

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pnc.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pnd.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pne.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pnh.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pnj.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pnm.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pnv.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pny.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10poh.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pok.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10por.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10ppa.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10ppr.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10pri.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ss10psc.csv' DELIMITER ',' CSV HEADER;

```



```

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0psd.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0ptn.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0ptx.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0pusa.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0pusb.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0put.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0pva.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0pvt.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0pwa.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0pwi.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0pwv.csv' DELIMITER ',' CSV HEADER;

COPY census.census2010pums FROM '/nmsba/data-
downloaded/census2010/ftp2.census.gov/programs-surveys/acs/data/pums/2010/1-
Year/ssl0pwy.csv' DELIMITER ',' CSV HEADER;

-- create indexes
CREATE INDEX idx_RT ON census.census2010pums (RT);
CREATE INDEX idx_SN ON census.census2010pums (SERIALNO);
CREATE INDEX idx_SP ON census.census2010pums (SPORDER);
CREATE INDEX idx_PU ON census.census2010pums (PUMA);
CREATE INDEX idx_ST ON census.census2010pums (ST);

-- clean up table
vacuum (verbose, analyze) census.census2010pums;

-- perform a test query
SELECT * FROM census.census2010pums LIMIT 20;

-- exit PostgreSQL Command Line:
\q

```


16 QUESTIONS FOR DATA MINING

Once the various datasets that have been outlined in this tutorial have been fully populated in a database, it is possible to mine the data to answer questions such as:

1. Does climate affect enrollment at postsecondary schools?
2. Does climate affect the test scores of the SAT scores of students applying to a university?
3. Does climate affect employment and wages in different industries?
4. Have local pollution levels increased from changes in EPA reporting facilities in the local area?
5. What is the yearly trend of the relationship between employment in a sector of a metropolitan area and the GDP of a metropolitan area as a function of average climate temperatures?
6. Is there an impact to GDP from low rainfall and high atmospheric particulate count in a region or metropolitan area?
7. What is the effect to GDP over time based on rainfall, atmospheric particulation count, and population growth?
8. Which postsecondary schools are the best choice based on acceptance scores, weather, pollution, housing, the outdoor experience, general education in the area, and local income?

17 DATA INTEGRATION AND WAREHOUSING

17.1 Recreating Data on AWS Server

To populate the server with the data outlined in this tutorial required approximately 200 hours. Each dataset has its own unique challenges that need to be addressed. Thus, every dataset has to be manually reviewed first before uploading the data. Data import can be challenging when inconsistencies in the data are encountered. These inconsistencies must be corrected before data import can be accomplished.

Once a dataset has been reviewed and tested for import, it is quick and easy to redownload and reimport using the commands outlined in this tutorial. It is estimated that it would take 6 hours (a reduction of 33 times) to recreate and populate the database server with the data from this tutorial.

This tutorial can be used to help expand the scope of data included in this effort.

17.2 Match Tables for Data Integration

Working with multiple datasets can be challenging for data integration. For example, some data sets use the value “NM” to represent New Mexico. Other data sets may use “New Mexico”, “35”, or some other identifier. This introduces complexities when attempting to join two or more tables from different datasets in a query. Creating “match” tables simplifies the joining process. A match tables provides a master value to represent ‘New Mexico’ and all the different identifiers used by the various datasets.

17.3 Requirements Planning

Before beginning to expand to create a large data warehouse, it is necessary to do extensive review to determine the best options that not only fit the business model but ensure requirements are not missed. Below are examples of questions that should be answered before tackling a data integration and warehousing capability:

- Where will the servers be located?
- How will scaling be achieved?
- Is there a preferred O/S platform?
- Who does server maintenance?
- Who installs patches and updates?
- Who is responsible for maintaining and restoring backups/disaster recovery?
- Who holds the master password(s)?
- Who downloads data and updates existing data?

18 DISTRIBUTION

Hard Copies

- 1 Cultivating Coders
Charles Ashley
219 Central Ave SW
STE 200
Albuquerque, NM 87102
- 2 High Water Mark
Phoebe Suina
PO Box 146
Cochiti Pueblo, NM 87072
- 3 Lautman Economic Architecture, LLC
Mark Lautman
7601 Jefferson St.NE
STE 100
Albuquerque, NM 87109
- 4 Resilient Solutions 21, LLC
Charles Rath
7601 Jefferson St. NE
STE 100
Albuquerque, NM 87109
- 5 Wallbuilder Project
Brian Burnett
12904 Sand Cherry Pl NE
Albuquerque, NM 87111
- 6 Why Water NM, LLC
Dale Dekker
7601 Jefferson St NE
STE 100
Albuquerque, NM 87109

Electronic Copies

- | | | | |
|---|--------|-------------------|------------------------|
| 1 | MS0782 | Mark K Kinnan | 6633 |
| 1 | MS1138 | Donald Shirah | 6132 |
| 1 | MS0899 | Technical Library | 9536 (electronic copy) |

