CO572 Advanced Databases Course Work 3: Pig

Due in 12noon 24th November 2017

Background Material

The tables of the **usgs** database provided by US Geographical Survey are available as a set of TSV files, each containing a table of data, accessed from CSG Linux Lab machines in the directory /vol/automed/data/usgs. Three of the tables are illustrated below, each listed with a small fragment of the data from each table.

state

code	abbr	name
1	AL	ALABAMA
2	AK	ALASKA
4	AZ	ARIZONA
5	AR	ARKANSAS
6	CA	CALIFORNIA
8	CO	COLORADO
9	CT	CONNECTICUT
10	DE	DELAWARE
11	DC	DISTRICT OF COLUMBIA
12	FL	FLORIDA

populated_place(state_code) $\overset{fk}{\Rightarrow}$ state(code)
key state(name)
key state(abbr)
key state(code)
key populated_place(name,county,state_code,federal_status,cell_name)
key feature(name,county,state_name,type,latitude)

populated place

name	county	state_code	county_code	latitude?	longitude?	elevation?	population?	federal_status?	cell_name
A and F Trailer Park	Maricopa	4	13	33.49	-112.11	1115	null	null	Phoenix
Achi	Pima	4	19	32.34	-112.01	null	null	null	Santa Rosa
Adams	Hamilton	12	47	30.46	-83.05	85	null	null	Fort Union
Abra	Yavapai	4	25	34.90	-112.45	null	null	null	Paulden
Adam	Alachua	12	1	29.49	-82.51	70	null	null	Bronson NE
Achan	Polk	12	105	27.86	-81.97	123	null	null	Bradley Junction
A and F Trailer Court	Maricopa	4	13	33.30	-111.84	1215	null	null	Chandler
A-1 Trailer Park	Maricopa	4	13	33.45	-112.01	1120	null	null	Phoenix
Acline	Charlotte	12	15	26.88	-82.02	5	null	null	Punta Gorda
Abe Springs	Calhoun	12	13	30.37	-85.15	104	null	null	Frink

feature

name	type	county?	state_name	latitude?	longitude?	elevation?
Ak-Chin Indian Reservation	reserve	Pinal	Arizona	33.03	-111.92	null
Alford	ppl	Jackson	Florida	30.69	-84.61	146
Alcoma	ppl	Polk	Florida	27.89	-80.52	120
Agua Fria River	stream	Maricopa	Arizona	33.39	-111.64	null
Alachua County	civil	Alachua	Florida	29.68	-81.65	null
Ajo Range	range	Pima	Arizona	32.04	-111.32	null
Alapaha River	stream	Hamilton	Florida	30.44	-82.90	null
Ajo	ppl	Pima	Arizona	32.37	-111.14	1747
Aguila	ppl	Maricopa	Arizona	33.94	-112.83	2163
Alachua	ppl	Alachua	Florida	29.80	-81.51	85

The state table contains all states and administratively equivalent entities within the USA. Each state has a number of habitations recorded in the populated_place table. The type column of feature identifies the type of geographic feature, such as forest, dam, lake, and include some classified as populated places under type ppl. Note that the data is not very 'clean', meaning that there are foreign keys not present that intuitively might expected to be present, and there is a certain amount of inconsistency between data found in the ferature and populated_place tables.

The exercise below requires that Pig scripts be written that are named after the question number, and generate the result in a directory also named after the question number. The instructions are based on working at the command line on one of the CSG linux machines in the lab.

Running an Example Script

For example, suppose there was a question 0 which asks

> *Produce a CSV file with the scheme (feature_name,populated_place_name,latitude,longitude), that contains the names of feature and populated_place rows share the same latiude and longitude. The results should be sorted by latitude and longitude.*

Then a suitable script to store in file q0.pig would be:

```
-- Load the state, feature and populated_place tables
RUN /vol/automed/data/usgs/load_tables.pig

-- Project just the columns of feature we need later
feature_data =
   FOREACH feature
   GENERATE name AS feature_name,latitude,longitude;

-- Project just the columns of populated_place we need later
populated_place_data =
   FOREACH populated_place
   GENERATE name AS populated_place_name,latitude,longitude;

-- Find those features that match the latitude and longitude of a place
same_place =
   JOIN feature_data BY (latitude,longitude),
        populated_place_data BY (latitude,longitude);

-- Get rid of data not required for result
same_place_data =
   FOREACH same_place
   GENERATE feature_name, populated_place_name,
        feature_data::latitude, feature_data::longitude;

-- Provide the result sorted
sorted_same_place_data =
   ORDER same_place_data
   BY    latitude,longitude;

STORE sorted_same_place_data INTO 'q0' USING PigStorage(',');
```

To run this file, copy it into your working directory by typing on the command line:

```
cp /vol/automed/data/usgs/q0.pig .
```

Then run the q0.pig script using the command:

```
pig -x local q0.pig
```

After building the pipeline for the job, and running the pipeline, you should have a q0 directory created, containing a file part-r-0000 with the result of the STORE command in the script.

Note that if you wish to rerun the script, since the STORE command expects the directory is saves into not to exist, you must first delete the q0 directory using the command:

```
rm -Rf q0
```

Further information about Pig

The official manuals for Pig are found at `pig.apache.org`, the version of Pig installed in the labs is 0.12.0 and hence you should look the documentation for that release. Versions of Pig are available to install on your own linux computers from `www.cloudera.com`. There is information on various development environment options at cloudera.

For those users that like using emacs as their text editor, there is a syntax highlighting option for emacs, which you can enable on a CSG linux machine by adding the following two lines to your .emacs configuration file in your home directory:

```
(setq load-path (nconc '("/vol/automed/emacs") load-path))
(require 'pig-mode)
```

Debugging Pig Scripts

To debug a Pig script, it is normally better to use Pig in interactive mode. If you enter the command:

```
pig -x local
```

then you should have the interactive prompt returned:

```
grunt>
```

You may then cut and paste sections of the script you are editing in a text editor into the command line to determine if they parse correctly, and then use the DUMP command to see the intermediate results.

For example, you enter the command at the grunt prompt:

```
RUN /vol/automed/data/usgs/load_tables.pig
```

you can view the contents of the state.tsv file using the command:

```
DUMP state;
```

If you then cut and paste from q0.pig the definitions of the feature_data, populated_place_data, and same_place aliases, you may then type at the grunt prompt:

```
DUMP same_place;
```

to view what are the contents of the alias, and then enter subsequent aliases, and view the contents of those aliases with other DUMP commands. You may also list the scheme of same_place by typing at the grunt prompt:

```
DESCRIBE same_place;
```

Submission

To gain full marks, answers to the following questions should make full use of Pig commands to write compact and efficient scripts, and be laid out such that structure of the scripts is clear. The queries must also run correctly on the Pig installation provided by CSG Linux lab machines, and be submitted electronically by the coursework deadline to CATE four Pig scripts q1.pig, q2.pig, q3.pig, and q4.pig providing the answer to corresponding question below. Each script should output its answer to a corresponding directory named q1, q2, q3, and q4.

Questions

1. Write a Pig script that writes a CSV file with the scheme (state_name) containing all those state names in feature for which there are no corresponding records in state. The result must be ordered by state name, return the names found in upper case, should assume all records in state are in upper case, and ignore difference in case between the two tables.

   marks 20

2. Write a Pig script that writes a CSV file with the scheme (state_name,population,elevation) that returns in order of state name the sum of the population and the average elevation of all populated_place data in a given state. The result must be ordered by state name, and elevation data must be rounded to the nearest integer.

   marks 20

3. Write a Pig script that writes a CSV file with the scheme (state_name,county,no_ppl,no_stream) the number of populated places and the number of streams recorded in feature in each county. The result must be ordered by state name and county.

marks 30

4. Write a Pig script that writes a CSV file with the scheme (state_name,name,population) containing the state name and place name of each populated_place, returning only the five largest populated places in each state. The result must be ordered by state name, with places in each state listed in declining order of population. If populations agree, then order of name should be used.

marks 30