CS 327E Final Project Guidelines – Spring 2018

Scenario:

The city of Austin has requested a data-driven analysis of Airbnb activity. They want to know how many short-term rentals are available, what their occupancy rate is, how much revenue is being generated, and how have these metrics changed over time. They are concerned about the affordability of housing in the city and want to understand the impact of Airbnb earnings on long-term rentals.

To answer these questions, they have collected Airbnb data since 2015 for Austin and four US-cities that have similar characteristics as Austin. Those cities are Boston, Denver, Nashville, and Portland. They have also collected nation-wide Zillow data for the same time periods to enable comparison between short-term and long-term rentals.

Minimum Requirements:

- Devise a method to ingest the Airbnb and Zillow data into BigQuery.

- Perform any necessary transformations in Dataflow to make the Zillow data queryable.

- Analyze the data in SQL, joining Airbnb and Zillow data and comparing multiple data points across cities and over time periods.

- Visualize the analysis in Data Studio.

Milestones:

Milestone 1 (Lab 7 in syllabus): ingest the Airbnb data into BigQuery, for Austin and two other cities; create an ERD for the ingested data (due Fri 03/30).

Milestone 2 (Lab 8 in syllabus): develop 4 metrics to analyze the three cities and draw comparisons between them; implement the metrics in SQL over BigQuery and visualize the results in Data Studio (due Fri 04/06).

Milestone 3 (Lab 9 in syllabus): transform the Zillow data in Dataflow and ingest it into BigQuery; create an ERD for the ingested data (due Fri 04/13). See the **Beam/Dataflow jobs** section below for further details.

Milestone 4 (Lab 10 in syllabus): analyze and visualize the Airbnb and Zillow data in SQL and Data Studio (due Fri 04/20). See the **Cross-Dataset Join** section below for further details.

Milestone 5 (Demo in syllabus): deliver a 10-minute live demo of your solution (on Fri 04/27).

Milestone 6 (Report in syllabus): produce a report that summarizes the key aspects of your solution (due Fri 05/04).

Data Sources:

All data sources for the project are stored in the bucket `utcs-spr2018-datasets` in Cloud Storage.

Airbnb Datasets:

There are 5 Airbnb datasets total, one for each city (Austin, Boston, Denver, Nashville, and Portland). You are only required to analyze 2 cities besides Austin.

Each dataset consist of 9 files:

```
amenity.csv
calendar_summary.csv
calendar.csv
host.csv
listing.csv
neighborhood.csv
review.csv
summary_listing.csv
summary_review.csv
```

The directory path to the Airbnb datasets is:

```
gs://utcs-spr2018-datasets/airbnb/$city/clean
```

where `$city` = {austin, boston, denver, nashville, portland}

For example, the full path to the Austin dataset is:

```
gs://utcs-spr2018-datasets/airbnb/austin/clean
```

Therefore, to access the file amenity.csv for Austin, use the full path:

```
gs://utcs-spr2018-datasets/airbnb/austin/clean/amenity.csv
```

Zillow Dataset:

The Zillow dataset consists of the following 5 files:

```
Zip_MedianRentalPrice_1Bedroom.csv
Zip_MedianRentalPrice_2Bedroom.csv
Zip_MedianRentalPrice_3Bedroom.csv
Zip_MedianRentalPrice_4Bedroom.csv
Zip_MedianRentalPrice_5BedroomOrMore.csv
```

These files are provided in 2 forms: files that have column headings in the first line of the file and files that do not contain column headings. The files with column headings are intended to be downloaded, opened in a text editor, and read by a human. The files without the column headings are intended to be read by an Apache Beam job and processed with Dataflow.

The download link to the files with column headings is:

https://storage.googleapis.com/utcs-spr2018-datasets/zillow/Zip_MedianRentalPrice_with_headings.zip

The download link to the files without column headings is:

https://storage.googleapis.com/utcs-spr2018-datasets/zillow/Zip_MedianRentalPrice_no_header.zip

Beam/Dataflow Jobs:

Implement an Apache Beam pipeline to ingest the raw CSV data into BigQuery. The pipeline consists of several individual Beam jobs written in Python. Each job processes one or more source CSV files, transforms the source data into the appropriate format, and loads a BigQuery table with the transformed data. All jobs must run on Dataflow using the Dataflow Runner.

There should be **6 jobs** total, one per target table. The target tables and data mappings are given below. The jobs must satisfy these specifications and produce the correct output in order to receive full credit.

Target Tables (BigQuery):

```
zillow.Rental_Price_1Bedroom(zipcode INTEGER, date DATE,
 price FLOAT);

zillow.Rental_Price_2Bedroom(zipcode INTEGER, date DATE,
 price FLOAT);

zillow.Rental_Price_3Bedroom(zipcode INTEGER, date DATE,
 price FLOAT);

zillow.Rental_Price_4Bedroom(zipcode INTEGER, date DATE,
 price FLOAT);

zillow.Rental_Price_5Bedroom(zipcode INTEGER, date DATE,
 price FLOAT);

zillow.Region(zipcode INTEGER, city STRING, state STRING,
 metro STRING,county STRING);
```

Data Mappings:

Source File(s) (CSV format)	**Target Table (BigQuery)**
`Zip_MedianRentalPrice_1Bedroom.csv`	`zillow.Rental_Price_1Bedroom`
`Zip_MedianRentalPrice_2Bedroom.csv`	`zillow.Rental_Price_2Bedroom`
`Zip_MedianRentalPrice_3Bedroom.csv`	`zillow.Rental_Price_3Bedroom`
`Zip_MedianRentalPrice_4Bedroom.csv`	`zillow.Rental_Price_4Bedroom`
`Zip_MedianRentalPrice_5Bedroom.csv`	`zillow.Rental_Price_5Bedroom`
`Zip_MedianRentalPrice_*.csv`	`zillow.Region`

Note: The `zillow.Region` table is populated from all 5 CSV files.

Translations and Filters:

- The "RegionName" column in the CSV files should be labeled `zipcode` in the Beam jobs and `Rental_Price` tables.
- The year-month columns in the CSV files (e.g. 2015-01, etc.) should be used to construct a date type, formatted as YYYY-MM-DD (e.g. 2015-01-01, etc.). Use "01" for the day component of the date since it is not provided in the source data.
- Extract from the CSV files only the data for the years 2015 and onward, starting with 2015-01 and ending with 2018-01. Omit data for 2014 and earlier.

Uniqueness Property:

The records in the target tables must be unique, including the records in the `zillow.Region` table. This implies that the job must remove any duplicate records before loading them into the appropriate target table.

Naming Conventions:

Name each job according to the target table that it populates. More specifically, the file name for the job should be made up of the table name converted to lower case. For example, the job for the `zillow.Region` table should be named `region.py`.

Cross-Dataset Joins:

Join the Airbnb and Zillow datasets on the `date`, `zipcode`, and `bedroom` fields to compute the *Revenue Crossover Point* metric.

The *Revenue Crossover Point* = ceiling of (Zillow's median rental price per month / Airbnb's median rental price per day).

This value represents the number of days per month that an Airbnb host would need to rent out his/her property in order to earn the same amount of revenue as from a median long-term rental.

The *Revenue Crossover Point* should be calculated for each combination of date, zipcode, and bedroom number.

Since the Zillow dates are given in monthly granularity, summarize the Airbnb bookings dates (from `Calendar.date`) to match this granularity. For example, a booking date of '2017-04-16' should be converted to '2017-04-01'. Hint: use the `date_trunc()` function to perform the conversion.

Additionally, the Airbnb rental price for a listing should come from `Calendar.price` if the value exists or `Listing.price` if it doesn't. Note that the `Calendar.price` value is more accurate than the `Listing.price` because it accounts for seasonal variability, but it is frequently not included. Hint: use a SQL case expression to perform the conditional logic.

Filter out all Airbnb listings for shared housing or apartment rentals. Use `Listing.room_type = 'Entire home/apt'` and `Listing.bedrooms > 0` as the filter criteria. Also, filter out any Airbnb dates, zipcodes or bedrooms which are NULL.

Calculate Airbnb's median rental price per day using the windowed function `percentile_cont()`. Refer to the BigQuery documentation for usage details.

Save the query results as a BigQuery view. The view should have the following definition:

```
v_Revenue_Crossover(date DATE, zipcode INT, bedrooms INT,
  airbnb_price_day FLOAT, zillow_price_month FLOAT,
  crossover_pt FLOAT)
```

Create a `v_Revenue_Crossover` view for each of your 3 cities and store the views in the same dataset as the city (e.g. `austin.v_Revenue_Crossover`, etc.).

Finally, create some interesting visualizations in Data Studio. The visualizations do not need to be limited to the `v_Revenue_Crossover` views and can access additional related tables/views from the database. For example, `zillow.Region`.

Submission Instructions:

Create a **final-project** folder in your git repo. Create a subfolder for each milestone and name the subfolders **milestone1**, **milestone2**, **milestone3**, etc. Place your work in the appropriate subfolder.

All code and documentation must be submitted to receive credit for a milestone. This includes SQL, ERDs, visualizations, and even errors. For example, if you were unable to load a particular file into BigQuery and you did this work through the BigQuery Console, provide a **readme.txt** file that explains the steps taken and issues encountered. This is especially important if you were unable to resolve the issue by the submission deadline and wish to receive partial credit for the milestone.

Follow our normal submission procedure through Canvas.

Use the following format for your submission.json:

```
{

    "commit_id": "[commit id]",
    "project_id": "[project id]"
}
```

The **commit_id** is the git commit identifier, just like for earlier assignments.

The **project_id** is a globally unique identifier assigned to your Google Cloud project. You can find out your project_id by going to your GCP console's home page. Note that the project_id that you list in the submission file will be used for grading your group's milestone. You should only list one project_id per submission (either yours or your partner's, but not both).

Rubrics:

Milestone 1: http://www.cs.utexas.edu/~scohen/project/fp_milestone1.pdf

Milestone 2: http://www.cs.utexas.edu/~scohen/project/fp_milestone2.pdf

Milestone 3: http://www.cs.utexas.edu/~scohen/project/fp_milestone3.pdf

Milestone 4: http://www.cs.utexas.edu/~scohen/project/fp_milestone4.pdf

Milestone 5: http://www.cs.utexas.edu/~scohen/project/fp_milestone5.pdf

Milestone 6: http://www.cs.utexas.edu/~scohen/project/fp_milestone6.pdf

Quick Links:

BigQuery Console: https://bigquery.cloud.google.com

BigQuery Functions: https://cloud.google.com/bigquery/docs/reference/standard-sql/functions-and-operators

Data Studio Console: https://datastudio.google.com/u/0/navigation/reporting

Cloud Storage Console: https://console.cloud.google.com/storage/browser

Dataflow Setup Guide: https://github.com/cs327e-spring2018/snippets/wiki/Dataflow-Setup-Guide

Dataflow Console: https://console.cloud.google.com/dataflow

Beam Programming Guide: https://beam.apache.org/documentation/programming-guide/

Code Samples: https://github.com/cs327e-spring2018/snippets/tree/master/beam (beam1.py – beam5.py)