**Assignment 2B**

**Generate Dimensions – Date**

**Initial Questions**

Q. What do the arguments to the gendates function mean?

A. There are 2 arguments in the gendates function. First (days\_back) is the number of days we need to subtract from today and the second (days\_total) is the total number of days upto which you want to generate dates for.

Q. What is the granularity of the date dimension (as we start)?

A. At the very start the granularity is upto the Month of the date.

Q. What is the format of the date "key"?

A. Auto Increasing integer value appended by the word “key”

Q. What are the formats of the date hierarchies?

A. key k, fulldate yyyy-mm-dd, year y, month m

**Final Questions**

Q. How did you format your additional hierarchy levels and why?

A. key k, fulldate yyyy-mm-dd 20:00:00:00000, year y, quarter q, month m, week w, day d, hour h. Since hour is the most granular value, I have kept it at the very end and all other values preceded it in a logical manner.

Q. Why did you change Python code the way you did?

A. These were the changes and the reason behind the change that I had to do in the code:

1. I had to make the date object into a datetime object since the hour granularity was supposed to be added.
2. The inner loop had to be run for the (number of days \* 24) times since every day has 24 hours.
3. Inside the for loop I incremented the date by an hour instead of my a day in the previous code.
4. Total number of entries generated were captured as well as the total number of entries logged were captured.

**Generate Dimensions – People**

**Questions**

Q. What is pd?

A. “pd” is an alias variable which is created while importing the pandas library. “pd” can be used to access all the different attributes and methods the pandas library can offer.

Q. How can you find the names of the columns?

A. titanic\_ppl.columns.values will give the all the column names of the data frame

Q. How can you find the number of rows?

A. The len() function can be used to find the number of rows in the data frame.

Q. Why did you program this the way you did?

A. For generating a 3-level hierarchy, Python pandas library provided a direct function named set\_index(). This function takes a mandatory argument which is the list of column names which needed to be indexed in the order of their priority. Also the fare updated column was added in the data frame by dividing the entire Fare column by 3. Since a normal list doesn’t have the capability to divide by 3, I used the numpy array. At the end I am printing the entire output in a html file and later for every 20 entries in the newly generated data frame I am printing the values in the log. I am logging the passenger Id and its rank value. Additionally I have floored the fare and fareupdated values.

Q. How did you structure your "person key" and why?

A. Meaning of person key remains the same as given in the database. Also used as Unique ID. Rank value notation created for the hierarchy for further clarity.

**Load data from database**

**Questions**

Q. What is the difference between stateInfo and datFrame?

A. stateInfo fetches all the records in the query using the execute and fetchAll statements of the cursor. The output of fetchAll() is a List. datFrame is formed using the pandas package and the read\_sql() function. The output of this function is as a dataframe hence it provides some more functionality than a list. Both the varibles have the same query data but in different formats.

Q. Could we implement the is\_it\_a\_state function more easily using datFrame?

A. It might be easy to implement the is\_it\_a\_state function easily with datframe because in that case there won’t be any need to make a different dictionary to check if the abbreviation is a state. The check can be directly performed on the datFrame object.

Q. What happens if we run is\_it\_a\_state with something that is not a state abbreviation at all?

A. It will throw an exception stating there is a Key error since there is no abbreviation in the dictionary.

Q. How did you address not finding a state in the table?

A. If the state was not found (Either full name or abbreviation) I would return back a relevant message to the user stating that there is no abbreviation or full state name with that value.

Q. Why did you choose the type of query that you did for the non-state household?

A. It seemed to be easy for joining tables first and then performing a <> operation on the join to check if the value IS NOT A STATE.

Q. Why did you choose or not choose to use Pandas for the non-state household query?

A. Since the only job to performed was to find the output of the query and not perform any operations with it, there was no need to make a dataframe and using the pandas library. Hence I used the normal cursor execute function to get the list and I just wrote all the values to a CSV file.

**Extract – Transform – Load**

Q. what does the resulting table tell us?

A. It tells us that there is some hierarchical information associated to every person which is mapped to the location dimension key. Every location dimension key can be linked to different values in the person\_economic\_info table

Q. what programming choices did you make and why?

A. I read the entire CSV data into a dataframe and extracted only the relevant fields from it which were required by the assignment. Then a location dimension table was created using the create table command. I made sure to add an ID field to all the tables I create. The fact table contains all the information from the CSV file which I extracted above as well as some randomly generated values of location dimension between 0 and 51 (corresponding to values in location dimension table). A foreign key constraint was applied on this key in the fact table.

Q. what does your query tell us?

A. The query counts by state how many persons have access to a wireless connection.