Part 1 - SQL

Given the following 3 tables

city_dim:

column name	type	semantics
city_sk	INT (P)	city id
name	VARCHAR	city name
state_sk	INT	state id

Example row:

city_sk	name	state_sk
5	Berlin	10

people:

column name	type	semantics
uuid	VARCHAR (P)	unique user id
first_name	VARCHAR	user's first name
last_name	VARCHAR	user's last name
city_sk	INT	city id

Example row:

uuid	first_name	last_name	city_sk
7e718f1p105a8da29k81ef3ea5f12872	Anna	Lugi	5

state_dim:

column name	type	semantics
`state_sk`	`INT`	state id
`name`	`VARCHAR`	state name
`is_in`	TINYINT (0 or 1)	boolean value for is in EU

Example row:

state_sk	name	is_in
`10`	`Germany`	1

Please provide the code for the following two tasks:

1. Write a SQL query that computes the amount of people per city
2. Write a SQL query that computes the amount of people from a state with is_in True.

Part 2 - Data Transformation

The `data.csv` dataset describes events that are recorded when a user registers.[1] It has ten entries, but can potentially be a lot bigger.

Input format:

column name	semantics
`first_name`	user's first name
`last_name`	user's last name
`created_at`	timestamp of event creation
`email`	user's email
`id`	user id

Example rows:

first_name	last_name	created_at	email	id
`Llewellyn`	`Monahan`	Sun Apr 24 1994 17:38:12 GMT+0200 (CEST)	Freeda@ford.ca	0
`Shany`	`Upton`	Fri Sep 02 2005 01:04:26 GMT+0200 (CEST)	Bessie_Fay@ian.net	1

Desired output format:

column name	type	semantics
`id`	`INT`	User id
`first_name`	`VARCHAR`	user's first name
`last_name`	`VARCHAR`	user's last name

[1] The dataset is real, but the PII data is not what we actually use in the backend.

`date_sk`	`INT`	Date in YYYYMMDD format

Example rows:

id	**first_name**	**last_name**	**date_sk**
0	`Llewellyn`	`Monahan`	19940424
1	`Shany`	`Upton`	20050902

Input-Output relation is N→N.

Using the ‘Part 2 - DataTransformation.py’ file, please provide the code for the following tasks:

1. Complete the decorator function 'transform' to transform the data to the desired output
2. Write code to read in the provided file data.csv and transform the data using the ‘transform’ function you created.

Part 3 - Query Chunks

We have a big MySQL table that we must backup before dropping it from a MySQL server.
Event_table_to_backup table:

column name	**type**	**semantics**
`event_id`	bigint(20) unsigned NOT NULL AUTO_INCREMENT (P)	unique event id
`event_name`	varchar(255) DEFAULT NULL	event name
`uuid`	varchar(255) DEFAULT NULL	unique user id

min(event_id) in the table is 2743139188
max(event_id) in the table is 5644096288

We have time for the backup, but of course, we don't want to query for 2900957100 rows at once. Instead, we want to execute the query in chunks of max 50k rows.

Our challenge is to query rows in chunks of 50000 rows.

```
toooo_big_query = SELECT  event_id,
                          event_name,
                          uuid
                  FROM    event_table_to_backup
                  WHERE   event_id >= 2743139188
                          AND event_id <= 5644096288
```

into several queries that return not more than chunksize rows:

```
query =  SELECT  event_id,
                 event_name,
                 uuid
         FROM    event_table_to_backup
         WHERE   event_id >= %(min_id)s
                 AND event_id <= %(max_id)s"""
```

Using the ‘Part 3 - QueryChunks.py’ file, please provide the code for the following tasks:
Given engine, query and result_handler,

1. Set the `query_params`, that we can pass to the query_runner function
2. Complete the query_runner function: output each query with `min_id` and `max_id` to stdout

Part 4 - Spark ETL

Using the Github events dataset:

wget http://data.githubarchive.org/2017-10-01-10.json.gz

Please provide the code for the following tasks:

1. Write an Apache Spark application in **Python/Scala** that reads the Github events json, extracts every events that has as
   type==PullRequestEvent
2. Clean the dataset, selecting only these fields for each event:
   created_at as created_at
   repo.name as repo_name
   actor.login as username
   payload.pull_request.user.login as pr_username
   payload.pull_request.created_at as pr_created_at
   payload.pull_request.head.repo.language as pr_repo_language
3. For each event, add another field, called pr_repo_language_type, based on the following criteria:
   - Procedural -> Basic, C
   - Object Oriented -> C#, C++, Java, Python,
   - Functional -> Lisp, Haskell, Scala
   - Data Science -> R, Jupyter Notebook, Julia
   - Others -> contains all the other languages that are not mention above
4. Save the final dataset as parquet with Snappy compression

Requirements

- Please submit both the spark job code and the dataset you produced
- Write unit tests
- Your Spark application could run in Stand-alone mode or it could run on YARN.

Part 5 - Data Modelling

WorkAwake have a system to simulate coffee machines usage for their large business clients. Let's assume their client has N machines in M office kitchens in their Headquarters. They can be in different floors and buildings.

You are in charge of supporting their performance evaluation of different coffee machines so you need to design the data model to capture the observed data and the measurements you would calculate on the data model.

You will use your detailed data model to compute a few simple measurements, for example average drink preparation time per user, average coffee consumption per user.

Assume your data is generated by a simulation that covers a 24 hours period, in which you can observe as much as you like (where each machine is, how many people are using it, when they arrive to use it, when the machines are refilled/cleaned, how many beans/milk/water is used and so on. If in doubt, assume you can observe it).

Please complete the following tasks:

1. List the different stakeholders who would be interested in the coffee machines' performance. (a "stakeholder" is any person or group who have an interest in or may be affected by some aspect of the machines' performance).
2. List other performance measures that it would be useful or important to measure – make sure these cover all of the stakeholders. (Hint: there are lots and lots of these. Aim for 10 or more...).
3. What would a suitable data representation look like?
    - Please design a series of tables (as would be suitable to put in a database). Make sure that the data representation (with very simple arithmetic calculations) is adequate to calculate the above measures, and any other measures that you deem important (and that those calculations are fairly easy and unambiguous).
    - Please point out any problems you might expect to arise with your data model.
4. For "Average preparation time per user" and at least 2 other performance measures, describe how they can be easily calculated from your data model. Please include the corresponding the SQL code you would use to calculate these measures.

(end of document)