Advanced Data Systems - Fall 2023

Assignment 2 – Writing Code

Experiments in the IoTLab are performed to measure power consumption of IoT devices (or nodes) when executing various workloads. The power consumption data obtained from each node is formatted as an OML file (see https://iot-lab.github.io/docs/tools/consumption-monitoring/).

Here is an example OML file obtained from one node:

```
protocol: 5
domain: 381037
start-time: 1697457732
sender-id: st_lrwan1_11
app-name: control_node_measures
schema: 0 _experiment_metadata subject:string key:string value:string
schema:   1   control_node_measures_consumption   timestamp_s:uint32   timestamp_us:uint32
power:double voltage:double current:double
content: text

3.542297       1      1      1697457735     465462  0.283869       4.870000       0.058263
3.542469       1      2      1697457735     469978  0.283259       4.871250       0.058239
3.542522       1      3      1697457735     474494  0.283259       4.868750       0.058239
3.542565       1      4      1697457735     478980  0.283869       4.868750       0.058312
3.542608       1      5      1697457735     483496  0.283869       4.870000       0.058361
3.542650       1      6      1697457735     488012  0.283869       4.870000       0.058312
3.542766       1      7      1697457735     492528  0.283259       4.870000       0.058166
...
```

OML is an instrumentation tool that inserts measurement points in an application. Data from the measurement points are directed to storage via a collection server through oml files. Currently, the collection server relies on sqlite and postgres to store measurements (https://github.com/mytestbed/oml).
The goal of the assignment is to develop an oml extension to DuckDB.

1. **Out of Tree Extension** (20%)

Create an out of tree extension, named oml, using the DuckDB extension template:
https://github.com/duckdb/extension-template

Question 1. Describe in one paragraph the build process when you compile the extension.

2. **Database Load** (60%)

Your DuckDB extension should provide a function to load data into an existing table. Basically, this function should be functionally equivalent to the following SQL code:

```
CREATE TABLE IF NOT EXISTS Power_Consumption (
    experiment_id VARCHAR,
    node_id VARCHAR,
    node_id_seq VARCHAR,
    time_sec VARCHAR NOT NULL,
    time_usec VARCHAR NOT NULL,
    power REAL NOT NULL,
    current REAL NOT NULL,
    voltage REAL NOT NULL
```

```
);

COPY       Power_Consumption       FROM       '/Users/htl719/Dropbox/Class/ADS-
23/livecoding/st_lrwan1_11_oml.csv'
(AUTO_DETECT TRUE);

CREATE SEQUENCE IF NOT EXISTS Power_Consumption_id_seq;
CREATE VIEW PC AS (SELECT nextval('power_consumption_id_seq') AS id,
                  cast(time_sec AS real) + cast(time_usec AS real) AS ts,
                   power, current, voltage
FROM power_consumption);
```

Question 2 (20%). Describe how the single threaded version of the "read_csv" table function is defined in DuckDB (see src/include/duckdb/function/table/read_csv.hpp and src/function/table/read_csv.cpp).

Question 3 (40%). Design, implement and test a table function Power_Consumption_load(filename) that reads from the oml file 'filename' generated by an IoTLab measurement point and loads the tuples it contains into a table that corresponds to the Power_Consumption table above. (hint: iot_load is a simplified version of the single threaded csv reader).

3. Database Generation (20%)

Question 5. Design, implement and test a table function OmlGen(filename) that reads an oml file, create a schema based on the metadata the oml file contains and loads the tuples that the oml file contains.