Distributed analysis of data (V2)

Configure Hive with postgresql

For this we only need to change the mysql image to postgres, change connection urls to point to this new container using the `postgresql` protocol, and change the driver to org.postgresql.Driver.

Load data into Hive

```
docker cp information_households.csv
↪  hiveserver2:/opt/hive/data/warehouse/ # docker cli does not
↪  allow stdin as parameters, thus we need to copy, then sed
docker exec hiveserver2 sed -i '1d'
↪  /opt/hive/data/warehouse/information_households.csv # We
↪  remove the headers of the csv files, with the colnumn names
docker cp daily_dataset.csv hiveserver2:/opt/hive/data/warehouse/
docker exec hiveserver2 sed -i '1d'
↪  /opt/hive/data/warehouse/daily_dataset.csv
docker exec -it hiveserver2 beeline -u
↪  jdbc:hive2://hiveserver2:10000
```

```
> CREATE TABLE clientes (
        LCLid STRING,
        stdorToU STRING,
        Acorn STRING,
        Acorn_grouped STRING,
        file STRING
    )
    ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
    STORED AS TEXTFILE;
> LOAD DATA INPATH
↪  '/opt/hive/data/warehouse/informations_households.csv' INTO
↪  TABLE clientes;
```

```
> CREATE TABLE consumos (
    LCLid STRING,
    day DATE,
    energy_median DOUBLE,
    energy_mean DOUBLE,
    energy_max DOUBLE,
    energy_count INT,
    energy_std DOUBLE,
    energy_sum DOUBLE,
    energy_min DOUBLE
    )
    ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

```
> LOAD DATA INPATH '/opt/hive/data/warehouse/daily_dataset.csv'
↪  INTO TABLE consumos;
```

Questions

1. What are the 10 first registers of each table?

```
> SELECT *
  FROM clientes
  LIMIT 5;
```

lclid	stdortou	acorn	acorn_grouped	file
MAC005492	ToU	ACORN-	ACORN-	block_0
MAC001074	ToU	ACORN-	ACORN-	block_0
MAC000002	Std	ACORN-A	Affluent	block_0
MAC003613	Std	ACORN-A	Affluent	block_0
MAC003597	Std	ACORN-A	Affluent	block_0
MAC003579	Std	ACORN-A	Affluent	block_0
MAC003566	Std	ACORN-A	Affluent	block_0
MAC003557	Std	ACORN-A	Affluent	block_0
MAC003553	Std	ACORN-A	Affluent	block_0
MAC003482	Std	ACORN-A	Affluent	block_0

```
> SELECT *
  FROM consumos
  LIMIT 10;
```

Note: The correspoding table has float values trimed to 2 decimal for visualization purposes. The day column has been reduced to only dat as all 10 first entries start with "2011-12-" and the whole table does not fit. lclid has also been reduced to the las 6 numbers of the id for a similar reason.

lclid	day	e_median	e_mean	e_max	e_count	e_std	e_sum	e_min
131	15	0.48	0.43	0.86	22	0.23	9.50	0.072
131	16	0.14	0.29	1.11	48	0.28	14.2	0.031
131	17	0.10	0.18	0.68	48	0.18	9.11	0.064
131	18	0.11	0.21	0.67	48	0.20	10.5	0.065
131	19	0.19	0.32	0.78	48	0.25	15.6	0.066
131	20	0.21	0.35	1.07	48	0.28	17.1	0.066
131	21	0.13	0.23	0.70	48	0.22	11.2	0.066
131	22	0.08	0.22	1.09	48	0.26	10.6	0.062
131	23	0.16	0.29	0.74	48	0.24	13.9	0.065
131	24	0.10	0.16	0.61	47	0.15	7.94	0.065

2. How many households are in each socioeconomic group?

Socioeconomic groups are defined by `Acorn` column. Thus we can do a groupby and count.

```
> SELECT Acorn, COUNT(*) AS count
  FROM clientes
  GROUP BY Acorn;
```

acorn	count
ACORN-	2
ACORN-A	157
ACORN-B	25
ACORN-C	151
ACORN-D	292
ACORN-E	1567
ACORN-F	684
ACORN-G	205
ACORN-H	455
ACORN-I	51
ACORN-J	112
ACORN-K	165
ACORN-L	342
ACORN-M	113
ACORN-N	152
ACORN-O	103
ACORN-P	110
ACORN-Q	831
ACORN-U	49

3. Show the first 10 households with the most ammount of consumption registers.

We can do this by using `ORDER BY` and `LIMIT` by first calculating the total ammount of registers for each household, which are stored in the `energy_count` column.

```
> SELECT LCLid, SUM(energy_count) AS total
  FROM consumos
  GROUP BY LCLid
  ORDER BY total DESC
  LIMIT 10;
```

lclid	total
MAC000147	39724

lclid	total
MAC000145	39724
MAC000150	39719
MAC000152	39718
MAC000148	39717
MAC000149	39717
MAC000153	39713
MAC000156	39712
MAC000151	39710
MAC000155	39704

4. Total energy consumption per household.

To calculate this, we need to make use of the `energy_sum` column, by first grouping by `LCLid` and then summing the values.

```
> SELECT LCLid, SUM(energy_sum) AS total
  FROM consumos
  GROUP BY LCLid
  LIMIT 5;
> SELECT LCLid, SUM(energy_sum) AS total
  FROM consumos
  GROUP BY LCLid
  ORDER BY LCLid DESC
  LIMIT 5;
```

lclid	total
MAC000002	6095.671997562051
MAC000003	14080.862013287842
MAC000004	1119.8390001356602
MAC000005	2911.00600380823
MAC000006	2167.4479979783064
. . . .	. . . .
MAC005567	2266.4009990394115
MAC005566	8942.237986594439
MAC005565	5.789999961853027
MAC005564	2314.1690012402833
MAC005563	NULL

5. Mean consumption per tariff type.

Considering tariff type is given by `stdorToU` column, we must first perform a join between `clientes` and `consumos` tables, and then group by `stdorToU` and calculate the mean of `energy_mean` column.

```
> SELECT stdorToU, AVG(energy_mean) AS mean
  FROM clientes JOIN consumos ON clientes.LCLid = consumos.LCLid
  GROUP BY stdorToU;
```

stdortou	mean
Std	0.2150364198457096
ToU	0.19859910474893103

6. Which households have more then 5kWh of consumption on at least one measure?

If some household has more than 5kWh of consumption in at least one measure, then we can filter the rows where `energy_max` is greater than 5.

```
> SELECT COUNT(DISTINCT LCLid) AS
↪  number_of_households_with_consumption_greater_than_five_kilowatts_hour
↪
  FROM consumos
  WHERE energy_max > 5;
```

number_of_households_with_consumption_greater_than_five_kilowatts_hour
172

7. Average consumption per Acorn category.

We will consider that the average consumption is calculated per day. Thus we will use the `energy_sum` column, which represents the total daily energy consumption per household.

We well need to join again both tables by `LCLid`, then group by acorn and `MEAN` the values in the `enery_sum`.

```
> SELECT Acorn, AVG(energy_sum) AS mean
  FROM clientes JOIN consumos ON clientes.LCLid = consumos.LCLid
  GROUP BY Acorn;
```

acorn	mean
ACORN-B	11.902596611015543
ACORN-C	11.950990032022034
ACORN-H	11.007658101081573
ACORN-I	9.439642578605136
ACORN-J	11.347920754414371
ACORN-L	10.028332144232545
ACORN-M	9.98745462878744

acorn	mean
ACORN-N	9.218043374493446
ACORN-O	8.528725788763992
ACORN-	12.003773375391654
ACORN-A	19.06387563150888
ACORN-D	13.578131126798352
ACORN-E	10.353099058446276
ACORN-F	9.19145402086119
ACORN-G	10.169359705464755
ACORN-K	10.006529787821579
ACORN-P	6.611038371189515
ACORN-Q	7.564821794787263
ACORN-U	11.617295885777194

8. Compare the different energy consumption of households per tariff type.

For this question, we will calculate several metrics, including: - Average daily consumption - Standard deviation of daily consumption - Mean standard deviation of daily consumption - Median of daily consumption - Median of standard deviation of daily consumption

While purposely ignoring metrics such as maximum or minimum as they most likely represent outliers.

```
> SELECT stdorToU,
         AVG(energy_sum) AS mean_energy_sum,
         STDDEV(energy_sum) AS std_energy_sum,
         AVG(energy_std) AS mean_energy_std,
         PERCENTILE(CAST(energy_sum*1000 AS BIGINT), 0.5) / 1000
↪   AS median_energy_sum,
         PERCENTILE(CAST(energy_std*1000 AS BIGINT), 0.5) / 1000
↪   AS median_energy_std
  FROM clientes JOIN consumos ON clientes.LCLid = consumos.LCLid
  GROUP BY stdorToU;
```

Note: The correspoding table has float values trimed to 2 decimal for visualization purposes. Column names have also been shortened.

stdortou	mean_e_sum	std_e_sum	mean_e_std	med_e_sum	med_e_std
Std	10.28	9.36	0.17	7.90	0.13
ToU	9.49	8.07	0.16	7.50	0.12

From the obtain data, we find that the most significant different is found in (1) the mean energy consumption per day, which appears to be higher in the `Std` tariff type, meaning the price is fixed. This could mean that people with fixed tariffs are less careful on when and how they consume energy, leading to a higher consumption. We find the same in (2) the median energy consumption per day.

On the other hand, regarding (3) standard deviation, we consider, given the median standard deviation are almost entirely the same, we conclude that tariff types do not affect the variability on the consumtion of electricity.

9. Detect households with inconsistant consumption behaviour.

We understand inconsistant consumtion behaviour as having less than 0.1kWh consumption for three days in a row.

To detect this, we propose to construct a window starting from any day within a household, and compare the three days in the window with the target value.

To do this, this table must first first be grouped by `LCLid`, and apply the LAG function to get previous and following days. Then we can filter the rows where the target value is less than 0.1kWh.

```
> SELECT COUNT(DISTINCT LCLid)
  FROM (
      SELECT LCLid, day, energy_sum,
           LAG(energy_sum, 1, 0.2) OVER (PARTITION BY LCLid ORDER
           ↪  BY day) AS previous_day,
           LAG(energy_sum, 2, 0.2) OVER (PARTITION BY LCLid ORDER
           ↪  BY day) AS previous_previous_day
      FROM consumos
  ) AS t
  WHERE energy_sum < 0.1 AND previous_day < 0.1 AND
  ↪  previous_previous_day < 0.1;
```

_c0
213

10. Consumption per morning, afternoon and night.

Cannot be done with available data

11. (Final Boss) How much does consumption change per weekdays or weekends?

To calculate whether a day is a weekday or a weekend day, we can use the `DAYOFWEEK` function, which returns 1 for Sunday, 2 for Monday, and so on.

Then, with that information we can map 1 and 7 to weekend, and 2, 3, 4, 5, 6 to weekday.

Once that is done, we can group by, and finally calculate the indicators: - Average daily Consumption - Standard deviation of daily consumption - Mean standard deviation of daily consumption - Median of daily consumption - Median of standard deviation of daily consumption

```
> SELECT day_type,
     AVG(energy_sum) AS mean_energy_sum,
     STDDEV(energy_sum) AS std_energy_sum,
     AVG(energy_std) AS mean_energy_std,
     PERCENTILE(CAST(energy_sum*1000 AS BIGINT), 0.5) / 1000 AS
↪  median_energy_sum,
     PERCENTILE(CAST(energy_std*1000 AS BIGINT), 0.5) / 1000 AS
↪  median_energy_std
  FROM (
    SELECT CASE
        WHEN
            DAYOFWEEK(day) IN (1, 7) THEN
                'weekend'
            ELSE
                'weekday'
        END AS day_type, energy_sum, energy_std
    FROM consumos
    ) AS t
  GROUP BY day_type;
```

Note: The correspoding table has float values trimed to 2 decimal for visualization purposes. Column names have also been shortened.

d_type	mean_e_sum	std_e_sum	mean_e_std	med_e_sum	med_e_std
weekday	9.98	9.06	0.170	7.70	0.13
weekend	10.46	9.27	0.177	8.11	0.137

As expected, we find that the energy consumption, both mean and median, are higher during the weekends, indicating that people spend more time at home.

Challenges encountered

Overall, I found this deliverable quite straight forward and did not find much trouble. The only thing I would consider was a bit more challenging was the 9th question, which required the use of the `LAG` function, which I am not really used to use, but reading some documentation is enough to understand how to use it in this case.