Prepare_data

October 31, 2017

1 Pre-process data from live bike_ped.data table

1.1 Database diagram

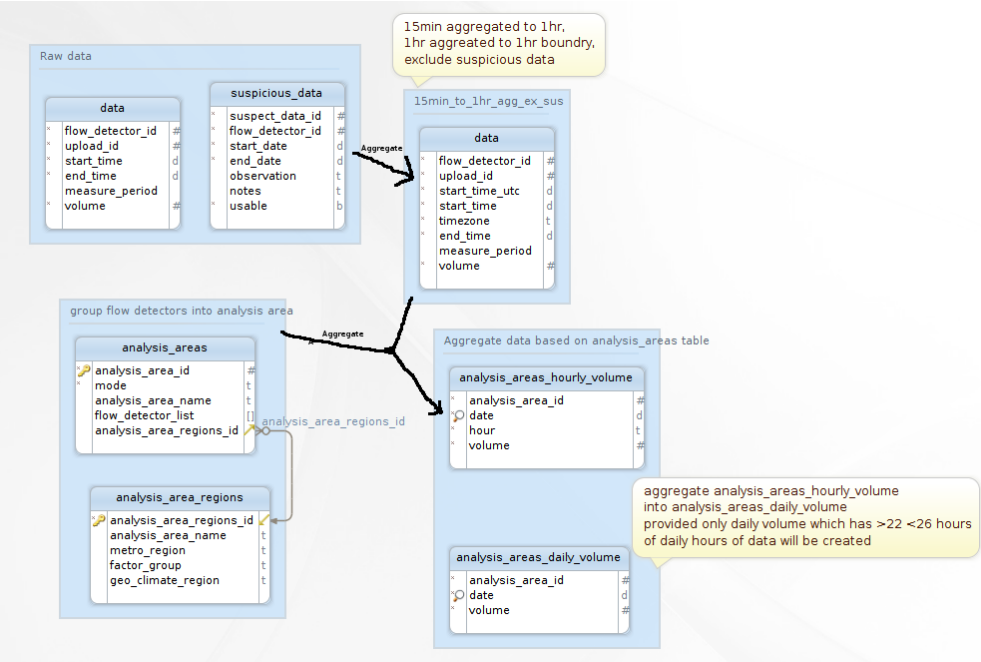

Figure 1. Data preparation

1.2 Aggregate 15min and hourly live data into hourly data

- Query to aggregate 15 min data into hourly data

```
insert into baa_temp.data (
with fltz as (
        select
          flow_detector_id,
          bike_ped.get_flow_detector_timezone(flow_detector_id) as timezone
        from
          bike_ped.flow_detectors
```

```
  )
  select
    bpd.flow_detector_id,
    bpd.upload_id,
    date_trunc('hour', bpd.start_time) as start_time_utc,
    date_trunc('hour', bpd.start_time at time zone fltz.timezone)
      as start_time,
    fltz.timezone as timezone,
    date_trunc('hour', bpd.start_time at time zone fltz.timezone)
      +'1 hour'::interval as end_time,
    '1 hour'::INTERVAL as measure_period,
    sum(volume) as volume
from
    bike_ped.data as bpd inner join fltz using(flow_detector_id)
where
    measure_period='00:15:00'
    group by flow_detector_id,
             upload_id, date_trunc('hour', start_time),
             fltz.timezone ,
             date_trunc('hour', bpd.start_time at time zone fltz.timezone)
    -- make sure we have 4 15-min records each hour
    having count(start_time) = 4
)
```

- Query to convert hourly data to houly data at hourly boundry (some hourly data not recorded at hour boundry)

```
insert into baa_temp.data (
  with fltz as (
        select
          flow_detector_id,
          bike_ped.get_flow_detector_timezone(flow_detector_id) as timezone
        from
          bike_ped.flow_detectors
  )
  select
    bpd.flow_detector_id,
    bpd.upload_id,
    date_trunc('hour', bpd.start_time) as start_time_utc,
    date_trunc('hour', bpd.start_time at time zone fltz.timezone)
      as start_time,
    fltz.timezone as timezone,
    date_trunc('hour', bpd.start_time at time zone fltz.timezone)
      +'1 hour'::interval as end_time,
    '1 hour'::INTERVAL as measure_period,
    sum(volume) as volume
from
    bike_ped.data as bpd inner join fltz using(flow_detector_id)
```

```
where
    measure_period='01:00:00'
    group by flow_detector_id,
             upload_id, date_trunc('hour', start_time),
             fltz.timezone ,
             date_trunc('hour', bpd.start_time at time zone fltz.timezone)
    )
```

```
In [1]: from utility import db_connect, query2csv
        from settings import  DBNAME, DBPASS, DBUSER, DBHOST

        qsql="""
        select * from baa_temp.suspicious_data
        """
        csvfile='suspicious_data.csv'
        query2csv(qsql,csvfile)
```

```
<IPython.core.display.HTML object>
```

1.2.1 Remove suspicious data from data table

- query

```
insert into baa_ex_sus.data (
with susdata as (select
 bd.flow_detector_id,
 bd.start_time,
 bd.end_time,
 bs.start_date,
 bs.end_date,
 bd.measure_period,
 bd.volume,
 bs.suspect_data_id,
 bs.usable
from
 baa_temp.data as bd, baa_temp.suspicious_data as bs
where
 bd.flow_detector_id = bs.flow_detector_id
 and bs.usable=FALSE
 and bd.start_time between bs.start_date and bs.end_date +'23 hour'::interval
 )
select
  bd.flow_detector_id,
  bd.upload_id,
  bd.start_time_utc,
  bd.start_time,
  bd.timezone,
  bd.end_time,
```

```
  bd.measure_period,
  bd.volume
from
  susdata as sd right join baa_temp.data as bd
    using(flow_detector_id, start_time, end_time)
where
  sd.usable is NULL
)
```

1.2.2 Verify that total data rows = data_rows_excluding_suspicious_data + suspicious_data_rows

- suspicious_data_rows **count**

```
with susdata as (select
 bd.flow_detector_id,
 bd.start_time,
 bd.end_time,
 bs.start_date,
 bs.end_date,
 bd.measure_period,
 bd.volume,
 bs.suspect_data_id,
 bs.usable
from
 baa_temp.data as bd, baa_temp.suspicious_data as bs
where
 bd.flow_detector_id = bs.flow_detector_id
 and bs.usable=FALSE
 and bd.start_time between bs.start_date and bs.end_date +'23 hour'::interval
 )
 select count(*) as sus_rows_count from susdata
 -- result --
 473830
```

- data_rows_excluding_suspicious_data **count**

```
 select count(*) as row_count_excluding_suspicious_data  from baa_ex_sus.data
 -- result --
 12072542
```

- total data rows **count**

```
select count(*) as hourly_data_row_count  from baa_temp.data
 -- result --
 12546372
```

- Verification:

  ```
  12072542 + 473830 = 12546372
  ```

1.2.3 Creating analysis_areas_hourly_volume and analysis_areas_daily_volume

```
CREATE TABLE baa_temp.analysis_areas_daily_volume_ex_sus
(
  analysis_area_id integer NOT NULL,
  date timestamp with time zone NOT NULL,
  volume integer NOT NULL
);

CREATE INDEX baa_temp_analysis_areas_daily_volume_date_ex_sus_idx
  ON baa_temp.analysis_areas_daily_volume_ex_sus
  USING btree
  (date);

CREATE TABLE baa_temp.analysis_areas_hourly_volume_ex_sus
(
  analysis_area_id integer NOT NULL,
  date timestamp with time zone NOT NULL,
  hour character varying(10) NOT NULL,
  volume integer NOT NULL
);

CREATE INDEX baa_temp_analysis_areas_hourly_volume_date_ex_sus_idx
  ON baa_temp.analysis_areas_hourly_volume_ex_sus
  USING btree
  (date);
```

1.2.4 Populate analysis_areas_hourly_volume table with a join of analysis_areas and hourly aggregated data table

```
insert into baa_ex_sus.analysis_areas_hourly_volume (
select
  baaa.analysis_area_id,
  date_trunc('day', bpd.start_time) as date,
  to_char(bpd.start_time, 'HH24') as hour,
  sum(bpd.volume) as volume
from
  baa.analysis_areas as baaa
    inner join baa_ex_sus.data as bpd
      on bpd.flow_detector_id = Any(baaa.flow_detector_list::int[])
  group by analysis_area_id, bpd.start_time
)
```

1.2.5 Populate analysis_areas_daily_volume table with aggregation of analysis_areas_hourly_volume into daily volume providing number of hourly records for a day is greater than 22 hours and less than 26 hours

```
insert into baa_ex_sus.analysis_areas_daily_volume (
with predaily as (
```

```
select
  baaa.analysis_area_id,
  date_trunc('day', bpd.start_time) as date,
  count(to_char(bpd.start_time, 'HH24')) as hour,
  -- count number of flow detectors in the group
  array_length(baaa.flow_detector_list, 1) as fd_cnt,
  sum(bpd.volume) as volume
from
  baa.analysis_areas as baaa
    inner join baa_ex_sus.data as bpd
      on bpd.flow_detector_id = Any(baaa.flow_detector_list::int[])
  group by 1,2
)
select
  analysis_area_id,
  date,
  volume
from predaily
where
-- make sure flow detector daily hour count is >22 and < 26
hour > fd_cnt*22
and hour < fd_cnt*26
)
```