

## Tarea N° 2

La tarea 2 está basada en el contenido visto en la primera clase de laboratorio. El data mart al que se hace referencia aquí es el utilizado en ese labo.

Son **dos ejercicios**, de los cuales se requiere **hacer sólo uno de ellos**, según lo siguiente:

- Quienes instalaron el servidor postgresSQL+postgis junto con el datamart en su equipo local, podrán elegir cuál hacer. Tener en cuenta que para el ejercicio 1 tendrían que usar jupyter notebooks en su equipo local, porque desde google colab no van a poder ver el postgresql local.
- Quienes no lo hayan hecho, tendrán que hacer el ejercicio 2, utilizando el sitio <https://sqliteonline.com/>, previa ejecución del script **"dwh\_recorte\_dm\_sube\_sin\_geom, TAREA 2v2.sql"** que se encuentra en el campus **adjunto a esta tarea**. (este script es una versión del datamart sin datos de tipo geometry, ya que la base psotgresql del sitio sqliteonline de la nube no cuenta con postgis).

### Ejercicio 1

1.- En un jupyter notebook, generar el código que lea la tabla DIM\_HORA del data mart utilizado en el labo, y la persista como un archivo jsonl.

- ✍ **TIP 1:** Si quisiera convertir la lista de tuplas de valores de campos en lista de diccionarios (combinándolas con sus títulos), podría usar la combinación de funciones **dict(zip(tupla1, tupla2))**, que aparea dos tuplas de igual longitud y la convierte en un diccionario.
- ✍ **TIP 2:** Tener en cuenta igualmente que hay formas alternativas de resolver el ejercicio que no requieren esa conversión.

### Ejercicio 2

Genere las consultas SQL que devuelvan lo solicitado a partir del data mart utilizado en el labo

2.a.- ¿Cuál es la zona de interés con mayor cantidad de transacciones a la hora 12 con tipo de tarifa "BONIFICADA" para la empresa TRANSPORTES 25 DE MAYO SRL?

- o Columnas de salida: Zona\_interes, cant\_trx

2.b.- ¿Cuántas y cuáles líneas de transporte distintas tuvieron alguna transacción en los barrios AREA CENTRO, ESTACION TERMINAL y PLAYA CHAPADMALAL?

- Columnas de salida: Nombre\_barrio, cant\_lineas\_transp, lista\_lineas\_transp
- El campo lista\_líneas\_transp tiene que mostrar los nombres de las líneas separados por coma, ordenados por nombre de la línea, y no tiene que tener nombres repetidos

- ✍ **TIP:** Usar la función de agregación **STRING\_AGG**
- ✍ **Si no está familiarizado con esta función**, vea en estas páginas de referencias (en ambas, son complementarias): <https://www.postgresql.org/docs/current/functions-aggregate.html> y <https://www.postgresql.org/docs/current/sql-expressions.html#SYNTAX-AGGREGATES>

