DATA ENGINEER & BI Analyst - EXAM

Ejercicio 1. Bases de Datos

Como parte del equipo de análisis de datos de HappyData, el equipo de analistas necesita una lista de todos los clientes y los protocolos de tráfico que han utilizado. Escribe en una consulta en SQL que genere el resultado.

El resultado debe tener el siguiente formato: client, protocol.
- *protocol* es una lista separada por comas de todos los protocolos para un cliente en particular, ordenados de forma descendente por el tráfico total, que se calcula como la suma de *traffic_in* y *traffic_out*.
- Los resultados deben estar ordenados de forma ascendente por cliente.

Input

```sql
-- create
CREATE TABLE traffic (
  client VARCHAR(17),
  protocol VARCHAR(17),
  traffic_in INTEGER,
  traffic_out INTEGER
);

-- inserts
INSERT INTO traffic VALUES ('19-58-33-40-6E-66',    'BGP',  233109, 974446);
INSERT INTO traffic VALUES ('19-58-33-40-6E-66',    'DNS',  260151, 56050);
INSERT INTO traffic VALUES ('19-58-33-40-6E-66',    'DNS',  808450, 78154);
INSERT INTO traffic VALUES ('19-58-33-40-6E-66',    'POP',  626847, 432101);
INSERT INTO traffic VALUES ('19-58-33-40-6E-66',    'SNP',  156130, 861098);
INSERT INTO traffic VALUES ('9E-43-EA-54-0A-E7',    'BGP',  931533, 393935);
INSERT INTO traffic VALUES ('9E-43-EA-54-0A-E7',    'DNS',  322727, 767978);
INSERT INTO traffic VALUES ('9E-43-EA-54-0A-E7',    'HTTP', 519008, 114712);
INSERT INTO traffic VALUES ('9E-43-EA-54-0A-E7',    'HTTPS',    997873, 660955);
INSERT INTO traffic VALUES ('A6-B6-94-1E-07-FE',    'BGP', 16598,  460181);
INSERT INTO traffic VALUES ('A6-B6-94-1E-07-FE',    'DHCP', 932759, 636364);
INSERT INTO traffic VALUES ('A6-B6-94-1E-07-FE',    'DNS', 311364, 189234);
INSERT INTO traffic VALUES ('A6-B6-94-1E-07-FE',    'HTTPS',    364181, 193177);
INSERT INTO traffic VALUES ('A6-B6-94-1E-07-FE',    'TCP',  309463, 301272);
INSERT INTO traffic VALUES ('BB-0B-0C-1D-24-F4',    'IMAP', 822503, 793792);
INSERT INTO traffic VALUES ('BB-0B-0C-1D-24-F4',    'POP',  440950, 157635);
INSERT INTO traffic VALUES ('BB-0B-0C-1D-24-F4',    'SNP',  94997,  660654);
INSERT INTO traffic VALUES ('BB-0B-0C-1D-24-F4',    'TCP',  554635, 361496);
INSERT INTO traffic VALUES ('E4-00-CE-46-3F-26',    'DNS',  478782, 523512);
INSERT INTO traffic VALUES ('E4-00-CE-46-3F-26',    'IMAP', 381783, 938555);
```

Entregables

- Resultado de la consulta
- SQL de la consulta

Schema

traffic		
name	**type**	**description**
client	VARCHAR(17)	Client MAC address
protocol	VARCHAR(64)	Protocol name
traffic_in	INT	Traffic in
traffic_out	INT	Traffic out

Data Sample

traffic			
client	**protocol**	**traffic_in**	**traffic_out**
19-58-33-40-6E-66	BGP	233109	974446
19-58-33-40-6E-66	DNS	260151	56050
19-58-33-40-6E-66	DNS	808450	78154
19-58-33-40-6E-66	POP	626847	432101
19-58-33-40-6E-66	SNP	156130	861098
9E-43-EA-54-0A-E7	BGP	931533	393935
9E-43-EA-54-0A-E7	DNS	322727	767978
9E-43-EA-54-0A-E7	HTTP	519008	114712
9E-43-EA-54-0A-E7	HTTPS	997873	660955
A6-B6-94-1E-07-FE	BGP	16598	460181
A6-B6-94-1E-07-FE	DHCP	932759	636364
A6-B6-94-1E-07-FE	DNS	311364	189234
A6-B6-94-1E-07-FE	HTTPS	364181	193177
A6-B6-94-1E-07-FE	TCP	309463	301272
BB-0B-0C-1D-24-F4	IMAP	822503	793792
BB-0B-0C-1D-24-F4	POP	440950	157635
BB-0B-0C-1D-24-F4	SNP	94997	660654
BB-0B-0C-1D-24-F4	TCP	554635	361496
E4-00-CE-46-3F-26	DNS	478782	523512
E4-00-CE-46-3F-26	IMAP	381783	938555

Ejemplo de la salida:

client ▲	**protocol**
19-58-33-40-6E-66	DNS,BGP,SNP
9E-43-EA-54-0A-E7	HTTPS,BGP,HTTP

Ejercicio 2. Programación

El número de goles conseguidos por dos equipos de fútbol en los partidos de una liga se da en forma de dos listas. Para cada partido del equipo B, calcula el número total de partidos del equipo A en los que el equipo A ha marcado menos o igual que el número de goles marcados por el equipo B en ese partido.

Ejemplo

```
equipoA = [1, 2, 3]
equipoB = [2, 4]
```

El equipo A ha jugado tres partidos y ha marcado `equipoA = [1, 2, 3]` goles en cada uno de ellos. El equipo B ha jugado dos partidos y ha marcado `equipoB = [2, 4]` goles en cada partido respectivamente. Para 2 goles marcados por el equipo B en su primer partido, el equipo A tiene 2 partidos con puntuaciones 1 y 2. Para los 4 goles marcados por el equipo B en su segundo partido, el equipo A tiene 3 partidos con puntuaciones 1, 2 y 3.
Por lo tanto, la respuesta es `[2, 3]`.

Escribe una función **counts** con las siguientes condiciones:

counts recibe los siguientes parámetros:

`int equipoA[n]`: primera lista de enteros positivos
`int equipoB[m]`: segunda lista de enteros positivos

Devuelve

int[m]: una matriz de m enteros positivos, uno por cada `equipoB[i]` que representa el número total de elementos del `equipoA[j]` que satisface `equipoA[j] ≤ equipoB[i]` donde `0 ≤ j < n y 0 ≤ i < m`, en el orden dado.

Restricciones

```
2 ≤ n, m ≤ 105
1 ≤ equipoA[j] ≤ 109, donde 0 ≤ j < n.
1 ≤ equipoB[i] ≤ 109, donde 0 ≤ i < m.
```

Ejemplo:

```
equipoA[] tamaño n = 5
equipoA = [2, 10, 5, 4, 8]
equipoB[] tamaño m = 4
equipoB = [3, 1, 7, 8]
```

Salida

```
res = [1,0,3,4]
```

Explicación

```
Los valores dados son n = 5, equipoA = [2, 10, 5, 4, 8], m = 4, y equipoB = [3, 1, 7, 8].

1. Para equipoB[0] = 3, tenemos 1 elemento en equipoA (equipoA[0] = 2) que es ≤ equipoB[0].
```

```
2. Para equipoB[1] = 1, tenemos 0 elementos en equipoA que son ≤ equipoB[1].
3. Para equipoB[2] = 7, tenemos 3 elementos en equipoA (equipoA[0] = 2, equipoA[2] = 5 y
equipoA[3] = 4) que son ≤ equipoB[2].
4. Para equipoB[3] = 8, tenemos 4 elementos en equipoA (equipoA[0] = 2, equipoA[2] = 5,
equipoA[3] = 4, y equipoA[4] = 8) que son ≤ equipoB[3].

Así, la función devuelve la matriz [1, 0, 3, 4] como respuesta.
```

Entregables

- Código de la función
- Pruebas unitarias (Opcionales pero deseables)

Consideraciones

- Aunque una respuesta por fuerza bruta soluciona el problema, nos gustaría ver otra implementación con una complejidad algorítmica menor.

Ejercicio 3.

Pipeline y orquestacion

El rol de Data Engineer es crear y administrar la Arquitectura para disponibilizar procesos de Data. Esta prueba técnica evalúa tu capacidad de aprender nuevos conceptos y tecnologías, tu capacidad de leer documentación y aplicarla para lograr resultados.

Descripción: Después del análisis, el equipo de Data te envía una matriz de relaciones entre personas, un 1 cuando tienen relación y un 0 cuando no tienen ningún tipo de relación. (Archivo de Excel adjunto a este email).

		1	2	3	4	5	6	7	8	9	10	11
		A	B	C	D	E	F	G	H	I	J	K
1	A	0	0	0	0	0	1	0	0	0	1	0
2	B	0	0	0	0	0	0	0	0	0	0	0
3	C	0	0	0	1	1	1	1	1	1	1	1
4	D	0	0	1	0	1	1	1	1	1	1	1
5	E	0	0	1	1	0	1	1	1	1	1	1
6	F	0	0	1	1	1	0	1	1	1	1	1
7	G	0	0	1	1	1	1	0	1	1	1	1
8	H	0	0	1	1	1	1	1	0	1	1	1
9	I	0	0	1	1	1	1	1	1	0	1	1
10	J	0	0	1	1	1	1	1	1	1	0	1
11	K	0	0	1	1	1	1	1	1	1	1	0
12	L	0	0	1	1	1	1	1	1	1	1	1
13	M	0	0	0	0	0	0	0	0	0	0	0
14	N	0	1	1	1	1	1	1	1	1	1	1
15	Ñ	0	0	0	0	0	0	0	0	0	1	0
16	O	0	0	0	0	0	0	0	0	0	0	0
17	P	0	0	0	0	0	0	0	0	0	0	0
18	Q											

El archivo tiene información hasta la row 17, así que faltan relaciones. El equipo de Data nos enviará este archivo actualizado diariamente, y te solicitan montar un pipeline de datos usando Dagster (Orquestador de Pipelines de Datos).

Reto: Como Data Engineer debes tomar estos datos y generar un Pipeline que cargue esta info en una nueva tabla de mysql (genera una Base de Datos Mysql local) y que este proceso se ejecute cada 24h actualizando la tabla de Mysql utilizando Dagster.

Resultado esperado:

Código en Python con la configuración del Pipeline de datos, junto con los querys utilizados para generar la DB y tablas. En este código esperamos se evidencien los pasos donde se lee el excel, organizan los datos y carga a mysql en una nueva tabla junto con su respectiva automatización empleando Dagster.

Documentación Recomendada:

- Adjunto a este email encontrarás el paso a paso detallando de como configurar Dagster en local y crear tu propio ETL.
- Documentación Dagster job: https://docs.dagster.io/tutorial/intro-tutorial/single-op-job

Código de Ejemplo de Automatización:

```python
from dagster import schedule
from datetime import datetime, time, date

@schedule(
        cron_schedule= "*/20 * * * *",
        pipeline_name= "load_bigquery_table_pipeline",
        execution_timezone="America/Bogota"
)
```

Modalidad de Calificación:

Tomaremos el código y lo ejecutaremos en un nuevo entorno, ejecutaremos los querys para generar la DB y el pipeline, esperamos que el pipeline se ejecute cada 24 h y actualice la tabla en Mysql, cargaremos diferentes versiones del archivo de excel y esperamos que la tabla de la base de datos se actualice con los cambios.

Envía tu solución a julilopez@palaceresorts.com

Tiempo para realizar la prueba: Tienes 3 días habiles para enviar la solución de este reto después de recibir este email. Envía la prueba lo más pronto posible, se tendrá en cuenta como puntos adicionales en la calificación de la prueba entre más pronto envíes la solución.

Punto Opcional: Puede complementar su pipeline de datos utilizando otras tecnologías y librerías. Adjuntando la documentación correspondiente explicando el proceso.