

Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 20 de julio de 2022

TEMA 20221C2						Padrón: _____
CRT		Proc.		NoSQL		Apellido: _____
NoSQL		Rec.		DW		Nombre: _____
Nota:						Cantidad de hojas: _____
						<input type="checkbox"/> Aprobado <input type="checkbox"/> Insuficiente

Criterio de aprobación: El examen está compuesto por 6 ítems, cada uno de los cuales se corrige como B/B-/Reg/Reg-/M. Se aprueba con nota mayor o igual a 4(cuatro), equivalente a desarrollar el 60% del examen correctamente.

1. (*Cálculo Relacional de Tuplas*) La fabricante de vehículos belga *Bruggeot* ha detectado una falla en el sistema de producción de la pieza de freno ‘IR2031H8-UPT’, y necesita contactar a los propietarios de vehículos con esa pieza para que envíen su auto a una revisión gratuita. Para esta tarea, se dispone de las siguientes tablas:

- Clientes(id_cliente, nombre, apellido, dirección, país, teléfono, mail)
- VehículosVendidos(número_serie, modelo, año, id_cliente_comprador, fecha_venta)
- ComposicionesVehículos(modelo, año, cod_pieza, cantidad)

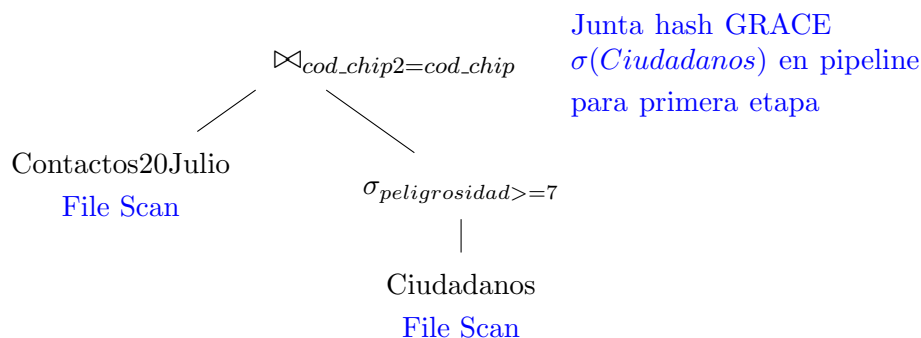
En donde la primera tabla posee datos de los compradores de vehículos, la segunda tabla indica de qué modelo y año era cada vehículo vendido y a qué cliente se vendió, y por último la tercera tabla indica qué piezas están presentes en cada vehículo de un determinado modelo y año de fabricación.

Escriba una consulta en *Cálculo Relacional de Tuplas* que encuentre el nombre, apellido, teléfono y mail de aquellos clientes que han comprado algún vehículo poseedor de la pieza ‘IR2031H8-UPT’.

2. (*Procesamiento de Consultas*) Año 2048. La República Inteligente de Belgandia, con una población de 100 millones de personas, se ha convertido en el país más avanzado de la Tierra. Todos sus ciudadanos son permanentemente vigilados a través de un sistema de ciberseguridad desarrollado por la empresa *Bruggeot*. Cada ciudadano posee una peligrosidad entre 0 y 10, y si en un día determinado un ciudadano c_1 es detectado en contacto con al menos un ciudadano de peligrosidad mayor o igual a 7, la peligrosidad de c_1 se aumenta en 0.01 unidades. Para hacer este seguimiento, el *Ministerio de Seguridad Ciudadana* dispone de las siguientes dos tablas:

- **Ciudadanos**(cod_chip, nombre, apellido, domicilio, peligrosidad)
- **Contactos20Julio**(cod_chip1, cod_chip2) --Si contiene (c1,c2) también contiene (c2,c1)

Se desea ejecutar una consulta que encuentre los códigos de chip de aquellos ciudadanos que en el día de hoy estuvieron en contacto con al menos un ciudadano de peligrosidad mayor o igual a 7, tal como se muestra en el siguiente plan de ejecución:



Se pide:

- Estime el costo del plan propuesto en términos de cantidad de bloques, utilizando la información de catálogo indicada debajo y suponiendo que se dispone de $M=10000$ bloques de memoria.
- Indique cuál sería el atributo de hashing para el armado de las particiones, y sugiera una cantidad razonable de particiones a construir para la junta hash GRACE.
- Si luego quisiera proyectar la salida a *cod_chip1* para eliminar los ciudadanos duplicados, ¿cree que esto se podría realizar en pipeline con la junta? En caso afirmativo justifique por qué es factible, y en caso negativo explique cómo lo haría.

CIUDADANOS	CONTACTOS20JULIO
$n(\text{Ciudadanos}) = 100.000.000$	$n(\text{Contactos20Julio}) = 300.000.000$
$B(\text{Ciudadanos}) = 10.000.000$	$B(\text{Contactos20Julio}) = 30.000.000$
Histograma para peligrosidad :	$V(\text{cod_chip1}, \text{Contactos20Julio}) = 50.000.000$
[0 – 2) : 40.000.000	$V(\text{cod_chip2}, \text{Contactos20Julio}) = 50.000.000$
[3 – 5) : 50.000.000	
[5 – 7) : 9.000.000	
[7 – 9) : 800.000	
[9 – 10] : 200.000	

3. (*NoSQL*) Indique si las siguientes afirmaciones sobre los LSM-trees (*log-structured merge trees*), son verdaderas o falsas, justificando brevemente su respuesta.

- (a) La estructura C_0 del LSM-tree es mantenida en memoria volátil. Por lo tanto, el gestor no le garantiza al usuario la durabilidad de los datos que se encuentran escritos en C_0 .
- (b) Si se buscan los datos asociados a una clave k en C_0 y no se encuentra ninguna entrada para k , entonces se deberá buscar en C_1 .
- (c) Si el usuario intenta modificar el valor asociado a una clave k que no se encuentra en C_0 sino en C_1 , se prohibirá la modificación debido a que C_1 es inmutable.
- (d) Las estructuras en disco –como C_1 , por ejemplo– pueden implementarse con un B-tree.

4. (*NoSQL*) La plataforma de empleos *LinkedOut* almacena datos sobre los empleos de sus usuarios y las competencias que poseen, entre otras cosas, en una base de datos Neo4J. Los (*Usuarios*) y las (*Competencias*) tienen la siguiente estructura:

```

1 CREATE (u1:Usuario {nombre: 'Pablo', ciudad: 'Mar del Plata'})
2      (u2:Usuario {nombre: 'Luisa', ciudad: 'Escobar'})
3      ...
4      (c1:Competencia {nombre: 'Docker', categoría: 'Infraestructura'})
5      (c2:Competencia {nombre: 'Python', categoría: 'Programación'})
6      ...

```

Las relaciones de amistad entre los usuarios se almacenan a través de la interrelación **SIGUE_A**:

```

1 MATCH (u1:Usuario {nombre: 'Pablo'}),
2      (u2:Usuario {nombre: 'Luisa'})
3 CREATE (u1)-[:SIGUE_A]->(u2)
4 MATCH ... CREATE ...

```

Mientras que la información sobre qué competencias posee cada usuario se almacena a través de la interrelación **CONOCE**:

```

1 MATCH (u1:Usuario {nombre: 'Pablo'}),
2      (c1:Competencia {nombre: 'Docker'})
3 CREATE (u1)-[:CONOCE]->(c1)
4 MATCH (u1:Usuario {nombre: 'Pablo'}),
5      (c2:Competencia {nombre: 'Python'})
6 CREATE (u2)-[:CONOCE]->(c2)
7 MATCH ... CREATE ...

```

La siguiente consulta en Cypher obtiene la cantidad de competencias en común de Pablo con cada uno de sus amigos (usuarios seguidos):

```

1 MATCH (amigo:Usuario)-[:SIGUE_A]-(p:Usuario),
2      comp_comun_amigo = (amigo:Usuario)-[:CONOCE]->(c:Competencia)-[:CONOCE]-(p)
3 WHERE p.nombre='Pablo'
4 RETURN amigo.nombre, COUNT(comp_comun_amigo)

```

La red social quisiera recomendar a Pablo nuevas personas a seguir. Modifique la consulta anterior para obtener la cantidad de competencias en común de Pablo con los amigos de sus amigos, pero que no son amigos de Pablo. El formato del resultado debería ser (*nombre_amigo_amigo*, *competencias_común*).

5. (*Recuperación*) Un SGBD implementa el algoritmo de recuperación REDO con checkpoint activo. Luego de una falla, el sistema encuentra el siguiente archivo de log:

```
01 (BEGIN, T1);
02 (WRITE, T1, A, 6);
03 (BEGIN, T2);
04 (WRITE, T2, B, 12);
05 (WRITE, T2, C, 3);
06 (COMMIT, T1);
07 (BEGIN CKPT, T2);
08 (BEGIN, T3);
09 (BEGIN, T4);
10 (WRITE, T3, A, 12);
11 (COMMIT T2);
12 (WRITE, T3, C, 21);
13 (END CKPT);
14 (WRITE, T4, B, 24);
15 (COMMIT, T3);
```

Explique cómo se llevará a cabo el procedimiento de recuperación, indicando qué cambios deben ser realizados en disco y en el archivo de log.

6. (*Data Warehousing*) Explique cuál es el papel que juegan los procesos ETL (*Extract-Transform-Load*) en la construcción de un Data Warehouse, y dé algunos ejemplos de los mismos.