

## Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 19 de diciembre de 2018

<b>TEMA 20182C2</b>						Padrón: _____
CRT		Proc.		DR		Apellido: _____
Rec.		NoSQL		NoSQL2		Nombre: _____
Corrigió:  <b>Nota:</b>						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*) A días de la Navidad, *Papá Noel* ya ha enviado sus regalos a 103 países del mundo, y le queda muy poco para terminar su tarea. Para realizarla se ayudó de las siguientes tablas que contienen los datos de todos los niños y niñas del mundo que le hicieron llegar sus cartas, el/los regalo/s que cada uno solicitó y una descripción breve de cada regalo:

- `Cartas(id_carta, nombre, domicilio, código_postal, país)`
- `Deseos(id_carta, cod_regalo)`
- `Regalos(cod_regalo, descripción, precio_rupias)`

Así, por ejemplo, para enviar el *convoy* con destino a un país  $X$ , *Papá Noel* calculaba el conjunto de los  $(id\_carta, cod\_regalo, precio\_rupias)$  de todos los regalos pedidos desde el país  $X$ , lo que en *Cálculo Relacional de Tuplas (C.R.T.)* podría expresarse como:

$$\{ c.id\_carta, d.cod\_regalo, r.precio\_rupias \mid$$

$$Cartas(c) \wedge Deseos(d) \wedge Regalos(r) \wedge c.país = X \wedge$$

$$c.id\_carta = d.id\_carta \wedge d.cod\_regalo = r.cod\_regalo$$

$$\}$$

Lamentablemente *Papá Noel* se está quedando sin presupuesto, y para su visita a Brasil ha decidido regalar a cada niño/a únicamente el regalo más económico de entre los que pidió.

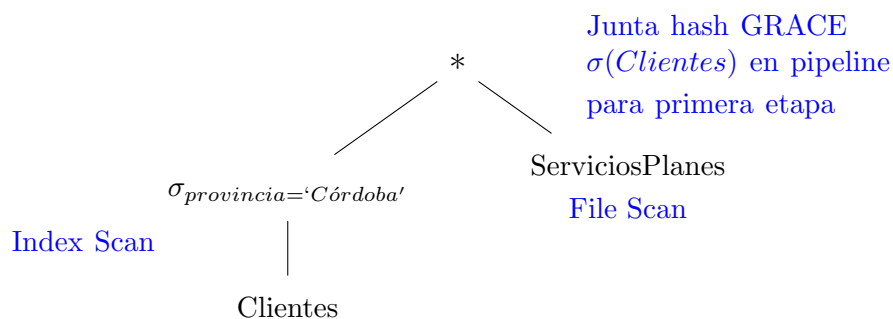
Escriba una consulta en *Cálculo Relacional de Tuplas (C.R.T.)* que devuelva el conjunto de las ternas  $(id\_carta, cod\_regalo, precio\_rupias)$  de los regalos a cargar en el *convoy* a Brasil, teniendo en cuenta la nueva limitación financiera.

Nota: En caso de que haya más de un regalo con precio mínimo entre los que algún niño/a pidió, simplemente desempate la situación utilizando el código de regalo, suponiendo que el mismo es comparable por  $<$ .

2. (*Procesamiento de Consultas*) Una obra social posee un registro de sus clientes, el plan que cada cliente contrata, los planes disponibles y los servicios que cada plan ofrece, a través de las siguientes tablas:

- **Clientes**(CUIL, apellido, nombre, domicilio, CP, provincia, cod\_plan)
- **Planes**(cod\_plan, costo, descripción)
- **ServiciosPlanes**(cod\_plan, servicio)

A los efectos de saber cuántos clientes de la provincia de Córdoba poseen cada servicio, se requiere como primer paso calcular la junta entre **Clientes** y **ServiciosPlanes**. Se utilizará el método de *junta hash GRACE* para calcular la junta, y un índice de *clustering* por el atributo **provincia** para acceder a la tabla **Clientes**. Esto se resume en el siguiente plan de ejecución:



Se pide:

- a) Calcule el costo del plan de ejecución en términos de cantidad de accesos a disco.
- b) Estime la cardinalidad del resultado de la junta en términos de cantidad de bloques.

Considere para sus cálculos la siguiente información de catálogo:

CLIENTES	SERVICIOSPLANES
$n(\text{Clientes}) = 2.000.000$	$n(\text{ServiciosPlanes}) = 60.000$
$B(\text{Clientes}) = 200.000$	$B(\text{ServiciosPlanes}) = 6.000$
$V(\text{provincia}, \text{Clientes}) = 20$	$V(\text{cod\_plan}, \text{ServiciosPlanes}) = 200$
$V(\text{cod\_plan}, \text{Clientes}) = 200$	
$H(I(\text{provincia}, \text{Clientes})) = 2$	

3. (*Diseño Relacional*) Dado un esquema de relación  $R(\bar{A})$  con un conjunto de dependencias funcionales  $F$ , decimos que el conjunto de esquemas de relación  $\{R_1(\bar{Z}_1), R_2(\bar{Z}_2), \dots, R_k(\bar{Z}_k)\}$  es una *descomposición* de  $R$  si y sólo si

$$\bar{Z}_1 \cup \bar{Z}_2 \cup \dots \cup \bar{Z}_k = \bar{A}$$

En otras palabras, debe cumplirse que:

- Cada atributo de  $R$  forme parte de al menos uno de los nuevos esquemas de relación.
- El conjunto de atributos de cada esquema de relación  $R_i$  sea un subconjunto del conjunto de atributos de  $R$ .

En este contexto, explique qué quiere decir que la descomposición  $\{R_1(\bar{Z}_1), R_2(\bar{Z}_2), \dots, R_k(\bar{Z}_k)\}$  preserve la información contenida en  $R$ , y de qué manera puede determinarse si una descomposición preserva la información o no.

Nota: La notación  $\bar{X} = \{X_1, X_2, \dots, X_{n_x}\}$  representa el conjunto de atributos de una relación.

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

```
01 (BEGIN, T1);
02 (BEGIN, T2);
03 (WRITE, T1, X, 4);
04 (WRITE, T2, Y, 8);
05 (COMMIT, T2);
06 (BEGIN CKPT, T1);
07 (WRITE, T1, Y, 3);
08 (BEGIN, T3);
09 (COMMIT T1);
10 (BEGIN, T4);
11 (WRITE, T3, Y, 10);
12 (END CKPT);
13 (WRITE, T4, X, 2);
14 (COMMIT, T4);
```

Explique cómo se llevará a cabo el procedimiento de recuperación, indicando hasta qué línea se deberá retroceder, y qué cambios deberán ser realizados en disco y en el archivo de log.

5. (*NoSQL*) Explique cuáles son las limitaciones de las bases de datos relacionales y que las bases de datos NoSQL intentan resolver. Comente brevemente alguna técnica o idea en particular utilizada por estas bases de datos para superar esas limitaciones.

6. (*NoSQL2*) La red social *FilmedIn* es una red de contactos entre personajes del mundo del cine destinada a facilitarles la búsqueda de empleo, y que cuenta con una base de datos en Neo4J sobre la cual los usuarios pueden realizar consultas. En esta base de datos se almacenan tres tipos de nodos: *Personas* (que pueden ser Actores, Directores, Guionistas, etc.), *Producciones* (Series, Películas, etc.) y *Empresas* (Estudios, Productoras, etc.).

```

1  (p1:Persona:Actor {nombre: 'Tom Hawks', fecha_nac: date('1963-02-20')})
2  (p1:Producción:Película {nombre: 'Madness', idioma: 'Inglés', duración: 98})
3  (a1:Empresa:Estudio {nombre: 'Píxeles', empleados: 730})

```

A su vez se almacenan dos tipos de interrelaciones:

- (p1:Persona)-[:CONOCE {fecha\_desde: date('2016-05-15')}]-(p2:Persona)  
Indica que p1 y p2 están en contacto.
- (p:Persona)-[:PARTICIPO]->(pr:Producción), ó (e:Empresa)-[:PARTICIPO]->(pr:Producción)  
Indica que la Persona p o la Empresa e participó en la Producción pr.

Tom Hawks está intentando obtener un papel en la próxima película del estudio de animación *Píxeles*, y acaba de ejecutar la siguiente consulta en *Neo4J* para averiguar si alguno de sus contactos  $c_1$  participó en alguna producción de dicho estudio:

```

1  MATCH p=(yo: Actor {nombre: 'Tom Hawks'})-[:CONOCE]-
2         (c1: Persona)-[:PARTICIPO]->(pr: Producción)<-[:PARTICIPO]-
3         (e: Estudio {nombre: 'Píxeles'})
4  RETURN c1.nombre;

```

Lamentablemente esta consulta devolvió un resultado vacío, por lo que Tom intentará ahora buscar a algún contacto suyo  $c_1$  que a su vez tenga un contacto  $c_2$  que haya participado en una producción de *Píxeles*. Dado que Tom posee 300 contactos, espera que esta consulta devuelva varios resultados, y por lo tanto quisiera ordenar a los contactos  $c_1$  en forma descendente por la cantidad de contactos distintos  $c_2$  de cada uno que hayan participado en producciones de *Píxeles*.

Escriba una consulta que devuelva dichos nombres de contacto  $c_1$  y la cantidad de contactos que cada uno tiene que hayan participado en producciones de *Píxeles*, ordenados en forma descendente por este último campo.