

Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 15 de febrero de 2023

TEMA 20222C3						Padrón: _____
CRT		NoSQL		Proc.		Apellido: _____
SQL		CyT		DR		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. (CRT)

*“Los amigos de mis amigos
son mis amigos.”*

Marco Sacherberg quiere evaluar la exactitud del conocido refrán, y para ello utilizará datos de una red social a la que tiene acceso. Dispone en particular de las siguientes tablas:

- Usuarios(cod_usuario, nombre, localidad)
- Amistades(cod_usuario_1, cod_usuario_2)

En esta última tabla, cada relación de amistad $a \sim b$ se guarda en ambos sentidos, es decir, como par (a, b) y como par (b, a) . Escriba una consulta en Cálculo Relacional de Tuplas que permita encontrar los códigos de usuario de las personas que cumplen con que “los amigos de sus amigos son también sus amigos” sin excepciones.

2. (*NoSQL*) Luego de repasar la lógica del cálculo de tuplas, Marco Sacherberg decide cargar los datos de la red social a la que tiene acceso dentro de una base Neo4j. En esta base, Marco colocó un nodo por cada usuario, y un arco por cada amistad entre usuarios, siguiendo la siguiente nomenclatura:

```

1  CREATE (u1:Usuario {cod_usuario: '01981571', nombre: 'Alberto Vulnek',
2     localidad: 'Boulogne'});
3
4  CREATE (u2:Usuario {cod_usuario: '03134990', nombre: 'Claudia Remmini',
5     localidad: 'Pilar'});
6
7  MATCH (u1: Usuario {cod_usuario: '01981571'}),
8     (u2: Usuario {cod_usuario: '01981571'})
9  CREATE (u1)-[:AMIGO_DE]->(u2);

```

Escriba una consulta en *Cypher* que devuelva el listado de los usuarios para los cuales se cumple que “los amigos de sus amigos son también sus amigos”.

Nota: Le sugerimos utilizar la estructura `MATCH ... WHERE [NOT] EXISTS { MATCH pattern } ...`

3. (*Procesamiento de consultas*) El último domingo se registró en la Ciudad de Buenos Aires el día de febrero más caluroso de los últimos 61 años. Para corroborar esta información, el *Servicio Meteorológico Nacional* utilizará la siguiente tabla en la que almacena todos los registros de temperatura históricos del mes de febrero:

- Temperaturas_Feb(cod_estación, fecha, hora, latitud, longitud, localidad, temperatura)

Para encontrar los registros más altos de temperatura ocurridos en Buenos Aires durante el mes de febrero, se ejecutará la siguiente consulta:

$$\begin{array}{c}
 \tau_{\text{temperatura}} \\
 | \\
 \sigma_{\text{localidad} = \text{"Ciudad de Buenos Aires"}} \\
 | \\
 \text{Temperaturas_Feb}
 \end{array}$$

El operador τ del plan de ejecución representa el ordenamiento de la tabla por el atributo ‘temperatura’. Dicho ordenamiento se realiza en forma similar al ordenamiento de una tabla en una proyección.

La tabla dispone de un índice de clustering de tipo árbol por `localidad`, de altura 5. Además, puede considerar para sus cálculos la siguiente información de catálogo:

TEMPERATURAS FEB

n(Temperaturas_Feb) = 400.000

B(Temperaturas_Feb) = 20.000

V(localidades, Temperaturas_Feb) = 804

Histograma de localidades:

Ciudad de Buenos Aires: 60.000

Córdoba: 50.000

Rosario: 35.000

Bahía Blanca: 33.000

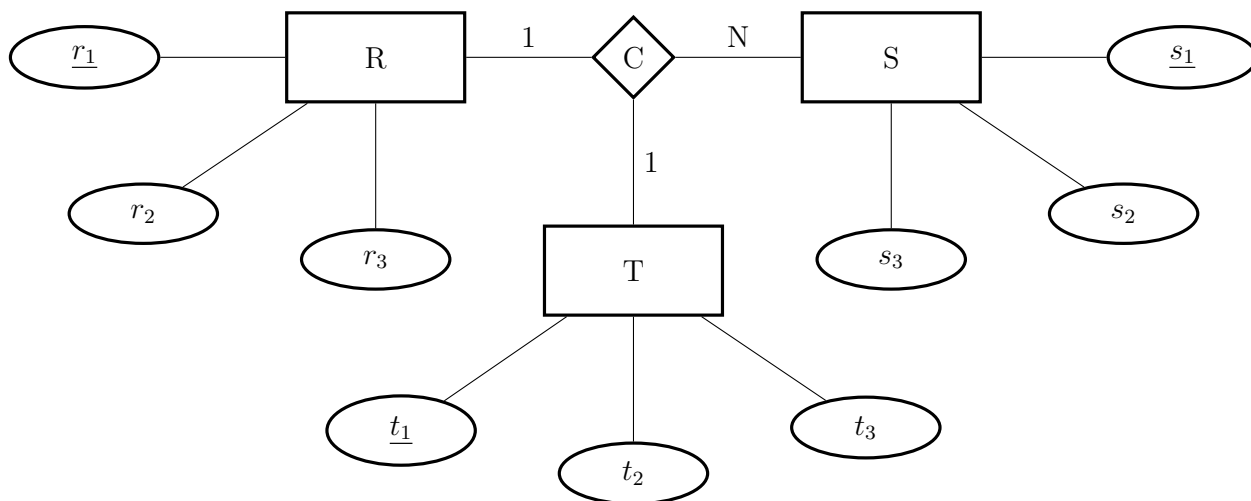
Otras: 222.000

Se pide:

- a) Teniendo la anterior información, y asumiendo que no hay problemas de memoria, estime el costo del plan de ejecución.
 - b) ¿Cuánta memoria estima que debería tener como mínimo para poder realizar el ordenamiento en memoria sin utilizar disco? Justifique su respuesta.
4. (*SQL*) Al ejecutar la consulta del *Ejercicio 3*, los científicos del *Servicio Meteorológico Nacional* encontraron que un mismo día es devuelto repetidas veces, dado que en la Ciudad de Buenos Aires existen numerosas estaciones meteorológicas, que además toman muchas mediciones por día. Escriba una consulta en lenguaje SQL que devuelva el ranking de los 10 días más calurosos registrados en la Ciudad de Buenos Aires, mostrando para cada uno de esos días la temperatura máxima registrada en alguna estación de la ciudad. Devuelva como resultado únicamente los campos de fecha y de temperatura máxima registrada.
5. (*Concurrencia y Transacciones*) Suponga que en una base de datos con un cierto estado inicial y sin transacciones en curso comienzan a ejecutarse en forma solapada tres transacciones T_1, T_2 y T_3 . Indique si las siguientes afirmaciones relativas a dicha ejecución concurrente son verdaderas ó falsas, justificando su respuesta.
- a) Si el solapamiento es equivalente por conflictos a la ejecución serial $T_1 \rightarrow T_3 \rightarrow T_2$, entonces no puede ser equivalente por conflictos también a la ejecución serial $T_2 \rightarrow T_3 \rightarrow T_1$.
 - b) Si el solapamiento es equivalente por resultado a la ejecución serial $T_1 \rightarrow T_3 \rightarrow T_2$, entonces es también equivalente por conflictos a dicha ejecución serial.
 - c) Si el grafo de precedencias resultante de la ejecución incluye el arco dirigido $T_2 \rightarrow T_3$, entonces no puede incluir el arco dirigido $T_3 \rightarrow T_2$.
 - d) Si cada vez que una transacción utiliza un recurso lo hace manteniendo un *lock exclusivo* sobre ese recurso, entonces el solapamiento será serializable.
 - e) Si el grafo de precedencias resultante de la ejecución es *acíclico*, el mismo puede tener más de un ordenamiento topológico.

Nota: Suponga que en principio desconocemos si el SGBD garantiza la serializabilidad ó no.

6. (Diseño Relacional) Considere el siguiente modelo entidad-interrelación:



Este modelo se traduce a una base de datos relacional compuesto por las siguientes 4 relaciones:

- $R(r_1, r_2, r_3)$
- $S(s_1, s_2, s_3)$
- $T(t_1, t_2, t_3)$
- $C(r_1, s_1, t_1)$

Si se quisiera respetar las restricciones impuestas por el modelo conceptual, entonces (indique V o F, y justifique su respuesta):

- a) La clave primaria de la relación C debe ser el atributo s_1 .
- b) Debe regir la dependencia funcional $(s_1, t_1) \rightarrow r_1$.
- c) Debe regir la dependencia funcional $(s_1, r_1) \rightarrow t_2$.
- d) El atributo s_1 en la relación S debe ser configurado como clave foránea que hace referencia a C, para asegurar que toda instancia de S se vincule con al menos una instancia de R.
- e) No debe haber nunca dos tuplas de C con igual valor de t_1 .