

Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 1 de marzo de 2023

TEMA 20222C5						Padrón: _____
CRT		Proc.		SQL		Apellido: _____
NoSQL		NoSQL		CyT		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*) En una plataforma de fútbol online, cada jugador está caracterizado por una medición de las siguientes habilidades: velocidad, defensa, pase, disparo, resistencia. Esta información, junto con el equipo al que pertenece el jugador, se encuentra relevada en la siguiente tabla:

- `Jugadores(nombre_jugador, nombre_equipo, velocidad, defensa, pase, disparo, resistencia)`
(‘Angel Correa’, ‘Atlético de Madrid’, 89, 63, 90, 93, 85)

Escriba una consulta en *Cálculo Relacional de Tuplas* que encuentre los nombres de los jugadores que superan en todas las habilidades a todos los demás jugadores de su mismo equipo.

2. (*Procesamiento de Consultas*) La FIFA quiere analizar la repercusión que tuvieron los premios *The Best* en distintos países del mundo. Para ello dispone de datos de la red social *Twitter* extraídos en la última semana. La información proveniente de esta red social se encuentra concentrada en las siguientes dos tablas:

- **Users**(user_id, nombre_usuario, ciudad, país)
- **Usos_Hashtags**(tweet_id, hashtag, user_id)

La tabla **Users** cuenta además con un índice secundario por el atributo **país**, mientras que la tabla **Usos_Hashtags** cuenta con un índice de clustering por el atributo **hashtag**, ambos de altura 4. Se dispone también de la siguiente información de catálogo:

USERS	USOS_HASHTAGS
$n(\text{Users}) = 10.000.000$	$n(\text{Usos_Hashtags}) = 5.000.000$
$B(\text{Users}) = 1.000.000$	$B(\text{Usos_Hashtags}) = 1.000.000$
$V(\text{país, Users}) = 200$	$V(\text{hashtag, Usos_Hashtags}) = 50.000$
	$V(\text{user_id, Usos_Hashtags}) = 500.000$

Adicionalmente, la tabla **Usos_Hashtags** cuenta con un histograma para el atributo **hashtag**, y ese histograma indica que el hashtag **#TheBest** fue uno de los trending topics, con 500.000 usos en la base de datos.

Los analistas de la FIFA quisieran calcular la cantidad de veces que se utilizó el hashtag **#TheBest** en cada país del mundo, mostrando para cada país su nombre y la cantidad de usos del hashtag **#TheBest**.

Considerando que sólo dispone de $M=1000$ bloques de memoria, se pide:

- Diseñe un plan de ejecución eficiente para la consulta, que permita resolverla con un costo razonablemente bajo.
 - Estime el costo del plan de ejecución propuesto.
3. (*SQL*) Emil Zatopek fue uno de los más grandes corredores de la historia del atletismo, habiendo conseguido el oro olímpico en los 5.000 y 10.000 metros llanos, y en la maratón. En las siguientes tablas, la *Federación Internacional de Atletismo* tiene registro de las marcas de los corredores en las distintas competiciones oficiales realizadas:

- **Atletas**(cod_atleta, nombre_apellido, fecha_nacimiento, nacionalidad)
- **Competiciones**(cod_competicion, tipo, ciudad, fecha)
- **Marcas**(cod_atleta, cod_competicion, marca)

Escriba una consulta en SQL que liste los códigos y nombres de aquellos atletas que alguna vez superaron a Emil Zatopek en una competición de tipo '10000 metros llanos'.

4. (*NoSQL*) Indique si las siguientes afirmaciones sobre *Dynamo* son verdaderas o falsas, justificando su respuesta.

- (a) En el mecanismo de hashing consistente no sólo las claves deben ser hasheadas, sino también los nodos.
- (b) En Dynamo, si un nodo se cae temporalmente, las consultas sobre claves ese nodo almacena no podrán ser respondidas por la base de datos hasta tanto ese nodo no vuelva a levantarse.
- (c) El uso de Merkle trees les permite a los nodos de Dynamo resolver los problemas de inconsistencias en sus datos.
- (d) El mecanismo de hashing consistente es lo que le permite a Dynamo garantizar un nivel de consistencia eventual.

5. (*NoSQL*) Teobaldo es apasionado de la historia, y le gusta analizar datos sobre los reinados de la Edad Media. Para ello construyó una base en Neo4j con información sobre los miembros de las distintas familias reales, sus filiaciones, matrimonios, y reinados.

En esta base, los nodos representan a las personas y a los territorios, como muestra el siguiente ejemplo:

```

1 CREATE (p1:Persona {nombre: 'Alfonso III de Aragón', año_nacimiento: 1265,
2                      ciudad_nacimiento: 'Valencia', año_fallecimiento: 1291,
3                      ciudad_fallecimiento: 'Barcelona'});
4 CREATE (t1:Territorio {nombre: 'Reino de Aragón', año_formación: 1035,
5                       año_disolución: 1707});

```

Asimismo, las aristas de este grafo representan interrelaciones de tipo `SE_CASÓ_CON`, `HIJO_DE` y `REINÓ`, como se ilustra a continuación:

```

1
2 MATCH (p1: Persona {nombre: 'Alfonso III de Aragón'}),
3      (p2: Persona {nombre: 'Leonor de Inglaterra'}),
4      (p3: Persona {nombre: 'Enrique II'})
5 CREATE (p1)-[:SE_CASÓ_CON]->(p2),
6      (p2)-[:HIJO_DE]->(p3);
7
8 MATCH (p1: Persona {nombre: 'Alfonso III de Aragón'}),
9      (t1: Territorio {nombre: 'Reino de Aragón'})
10 CREATE (p1)-[:REINÓ {inicio_reinado: 1285, fin_reinado: 1291}]->(t1);

```

Teobaldo quisiera identificar las situaciones en que un hijo/a de un/a rey/reina de Inglaterra, sin llegar nunca a ser rey/reina de Inglaterra, se casó con un rey/reina de Francia.

Escriba una consulta en *Cypher* que encuentre los nombres de estas personas que, siendo hijas de alguien que reinó en el Reino de Inglaterra pero sin nunca haberlo reinado, se hayan casado con alguien que reinó en el Reino de Francia.

6. (*Concurrencia y Transacciones*) Indique si las siguientes afirmaciones sobre concurrencia son verdaderas o falsas, justificando brevemente su respuesta.

- (a) En el método de control de concurrencia *Snapshot Isolation* no es posible que ocurra la anomalía del fantasma.
- (b) En el método de control de concurrencia basado en timestamps no pueden ocurrir deadlocks.
- (c) Bajo *Snapshot Isolation* nunca es necesario abortar una transacción para garantizar serializabilidad.
- (d) Bajo 2PL (*two-phase locking*) todo solapamiento de transacciones resulta recuperable.
- (e) Bajo *Snapshot Isolation* nunca puede ocurrir una lectura sucia.