

Base de Datos (75.15 / 95.05 / TA044)Evaluación Integradora - 18 de diciembre de 2024 - **20242C2**

SQL/CRT		DR		NoSQL		Padrón: Apellido: Nombre: Hojas entregadas:
NoSQL		Recup.		Proc.		
Nota: <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. El examen se aprueba con nota mayor o igual a 4 (cuatro) y la condición de aprobación es desarrollar al menos un ítem bien (B/B-) entre los 3 ejercicios de procesamiento de consultas y NoSQL y tener al menos un 60% del examen correcto.

1. (*SQL / CRT*) La Asociación Argentina de Tenistas Aficionados guarda información sobre los resultados de los partidos que se efectúan entre sus jugadores:

- **Partidos** (id_partido, id_jug_1, sets_jug_1, id_jug_2, sets_jug_2)
- **Jugadores** (id_jugador, nombre_con_apodo, fecha_inscripcion)

Escriba una consulta que devuelva los id de partidos en que el tenista Jose Luis “Batata” Clerc le ganó al tenista Juan Carlos “Papa” Noel (se gana teniendo más sets que el otro jugador).

Importante: Para alumnos de 2024 debe resolverse en SQL, para alumnos de cuatrimestres anteriores en Cálculo Relacional de tuplas.

2. (*Diseño Relacional*) La Asociación Argentina de Regalos Navideños quiere registrar en una base de datos todos los pedidos de los niños del país. Tiene los siguientes datos:

PedidosNavideños (dni_niño, nombre, apodo, puntaje_conducta, direccion, cod_juguete, nombre_juguete, nombre_tarjeta, direccion_entrega)

Sabiendo que:

- De los niños se conoce el nombre, el apodo que le dan los padres (ejemplo: “Pupe”), un puntaje del 1 al 10 que indica cómo se portó en el año y la dirección donde vive
- Puede haber varios pedidos de juguete por niño, para cada uno de ellos se pondrá en el nombre de la tarjeta quien lo regala (ejemplo: “Abuelo Pocho”)
- Pero para un niño la misma persona que lo regala no le hace más de un regalo
- La dirección de entrega no es siempre la misma para cada niño, a veces un familiar pide que le entreguen el regalo en su casa para luego entregarlo en persona

Descomponga la relación PedidosNavideños a **tercera forma normal** para evitar redundancias

3. (*NoSQL*) Nicolas posee una gran cantidad de ahorros en criptomonedas, y quiere poner al resguardo la clave de su wallet compartiéndola con dos personas de extrema confianza. En concreto, quiere dividir la clave de 128 bits en dos partes, y entregar cada parte a dos amigos de manera que:

- Dichos dos amigos de Nicolás sean amigos entre ellos también
- Ninguno de dichos amigos sea amigo de un enemigo de Nicolás
- El par de amigos escogidos maximice la confianza que Nicolás posee en ellos, de entre todos los pares de amigos que cumplan las dos condiciones anteriores.

Para poder encontrar a la pareja ideal, Nicolas ha desarrollado un operativo de inteligencia por el que capturó todas las relaciones de amistad y enemistad propias y de sus amigos, y las volcó en un grafo en Neo4j con la siguiente estructura:

```
MATCH (sn:Persona { nombre: 'Nicolas' } ) ,
      (no:Persona { nombre: 'Noelia' } ) ,
      (g:Persona { nombre: 'Gino Renni' } )
CREATE (sn)-[:AMIGO_DE {confianza: 0.75}]->(no) ,
      (no)-[:ENEMIGO_DE]->(g)
```

En donde el nivel de confianza de una relación de amistad es un número real entre 0 y 1, mientras que las relaciones de enemistad no poseen ningún atributo adicional.

Escribí una consulta en Neo4j que devuelva el nombre de cada persona integrante de la pareja ideal a ser seleccionada por Nicolás (si hay empate, con devolver una pareja alcanza).

4. (*NoSQL*) Santa Claus decidió que era hora de ganar más dinero y le vendió su base de datos de pedidos de regalo a una empresa para que pueda hacer ingeniería de datos sobre ella. Muy actualizado a las nuevas tendencias, Santa usa una base MongoDB con la siguiente estructura de documentos:

```
{
  "nombre": "Martincito", "direccion": "Paseo Colon 850",
  "ciudad": "Buenos Aires", "pais": "Argentina",
  "Regalos" : [ "Pelota Yellowey"
    , "Colchoneta Peppa la Galletita"
    , "Vaquia Feliz La Granja de Zurron" ]
}
```

El sector de ingeniería de datos quiere conocer los regalos que hayan sido pedidos por al menos 10 hermanos de Argentina, considerando como hermanos a dos niños que viven en la misma dirección. El pedido definitivamente es raro, pero nadie desconfía de la capacidad del sector. Escriba una consulta para poder devolver el nombre de los regalos que cumplen lo pedido

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

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

01 (BEGIN, T1);
 02 (WRITE T1, A, 1, 2);
 03 (COMMIT, T1);
 04 (BEGIN, T2);
 05 (WRITE T2, A, 2, 3);
 06 (BEGIN CKPT, T2);
 07 (BEGIN, T3);
 08 (BEGIN, T4);
 09 (WRITE T3, B, 4, 5);
 10 (END CKPT);
 11 (WRITE T3, C, 6, 7);
 12 (COMMIT, T3);
 13 (WRITE T4, C, 7, 8);

6. (Procesamiento de consultas) El viejito Pascuero quiere asignar a sus renos para distribuir los presentes navideños. Para ello efectúa un join natural entre las siguientes tablas, utilizando el método de junta HASH GRACE (se cuenta con memoria suficiente para hacerlo):

- Criaturitas (id_criaturita, nombre_criaturita, apellido, direccion, ciudad)
- Renos (id_reno, nombre_reno, ciudad)

Conociendo la siguiente información de las tablas:

CRIATURITAS	RENOS
$n(\text{Criaturitas}) = 50,000$	$n(\text{Renos}) = 1,000$
$B(\text{Criaturitas}) = 5,000$	$B(\text{Renos}) = 100$
$V(\text{ciudad}, \text{Criaturitas}) = 503$	$V(\text{ciudad}, \text{Renos}) = 303$

Se posee un histograma en ambas tablas con los tres valores más frecuentes del campo “ciudad”:

Valor	CRIATURITAS	RENOS
Santiago	10,000	100
Valparaíso	5,000	50
Puerto Montt	5,000	50

Calcule el **costo** de efectuar el join natural. Estime la **cantidad de filas** y de **bloques** devueltos.