

**Base de Datos (75.15 / 95.05 / TA044)**

Evaluación Integradora - 19 de febrero de 2025 - 20242C3

SQL/CRT		NoSQL		Conc.		Padrón:
Proc.		Recup		NoSQL		Apellido:
Nota: <input type="checkbox"/> Aprobado <input type="checkbox"/> Insuficiente						Nombre:
						Hojas entregadas:

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 Curling mantiene datos de los ganadores de distintos torneos anuales:

- **Ganadores** (nombre\_torneo, año, ganador)

Escriba una consulta que devuelva a quienes hayan ganado al menos una vez el torneo "Champions Liga" pero nunca hayan ganado ni el torneo "Ke Torneo" ni el torneo "Torneo en Bloque".

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

2. (NoSQL) El registro nacional de las personas posee en una base Neo4J datos de las personas, identificadas por su DNI, y los parentescos entre ellas:

```
CREATE (p1:Persona{ dni: 28000111, nombre: "Lucas Ramón" });
CREATE (p2:Persona{ dni: 7222333 , nombre: "Raimundo Ramón" });

MATCH (p1:Persona{dni: 28000111}), (p2:Persona{dni: 7222333})
CREATE (p1)-[:HIJO_DE]->(p2);
```

Escriba una consulta en Cypher que devuelva a todos los primos hermanos de una persona con DNI 12345678

**Aclaraciones por si acaso:** Primo hermano = Hijo/a de un tío/a de una persona. Tío/a = Hermano/a de uno de los padres de una persona

3. (Concurrencia) Una transacción con un timestamp de valor 8 quiere utilizar el ítem X de una base de datos que trabaja con control de concurrencia basado en timestamps. Se encuentra con que read\_TS(X) es 9 y write\_TS(X) es 5. Indique si la transacción **puede leer** el ítem. Indique si **puede escribirlo**. Para cada negativo, explique brevemente por qué. Para cada afirmativo, indique los **cambios a efectuar** luego en el read y write de la transacción.

4. (*Proc. Cons.*) La compañía de envíos *Lentti* quiere premiar a sus mejores repartidores. Para ello, inicialmente ejecuta la siguiente consulta en su base de datos con todos los pedidos:

```
SELECT *
FROM repartidores r INNER JOIN pedidos p USING (id_repartidor)
WHERE p.calificacion = 10 OR p.calificacion = 9
```

Posteriormente, agrupará por id de repartidor y premiará a los que mayor cantidad de pedidos hicieron con un hermoso señalador de libros con el logo de la empresa.

Estime el **costo** de resolver la primera consulta, teniendo la siguiente información de catálogo:

REPARTIDORES	PEDIDOS
n(repartidores) = 10,000	n(pedidos) = 1,000,000
B(repartidores) = 2,000	B(pedidos) = 100,000
	V(id_repartidor, pedidos) = 10,000
	V(calificacion, pedidos) = 8

No hay índices en las tablas y se cuenta con 100 bloques de memoria, el join quiere hacerse usando el método de **loops anidados por bloque** (aunque no resultara ser la mejor opción).

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) incompleto:

**Complete los valores faltantes** y 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, __);
03 (BEGIN CKPT, __);
04 (COMMIT, T1);
05 (BEGIN, T2);
06 (WRITE T2, A, 2, 4);
07 (COMMIT, T2);
08 (END CKPT);
09 (BEGIN, T3);
10 (BEGIN CKPT, __);
11 (WRITE T3, A, __, 3);
12 (WRITE T3, B, 6, 7);
```

6. (*NoSQL*) Indique si la siguiente secuencia de instrucciones en tres procesos tiene **consistencia secuencial**, justificando por qué no la tiene en caso negativo, o mostrando una ejecución secuencial equivalente en caso afirmativo. Los valores iniciales de A y B son 0

P1	Ra (0)	Wa (10)	Rb (20)	Wb (30)
P2	Rb (0)	Wb (20)	Ra (10)	Wa (50)
P3	Rb (0)	Ra (10)	Rb (20)	Wb (40)