

Base de Datos (75.15 / 95.05 / TA044)

Evaluación Integradora - 17 de julio de 2024 - 20241C3

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

1. (*SQL / CRT*) La compañía de envíos *Lentti* registra los datos de sus envíos realizados en una base relacional:

- **Envío** (id, fecha, hora, monto, cod_repartidor, direccion, barrio)
- **Repartidores** (cod_repartidor, nombre, fecha_ingreso)

Escriba una consulta que devuelva el código y nombre de los repartidores que efectuaron envíos al barrio de Palermo y nunca efectuaron envíos al barrio de Recoleta ni al de Belgrano

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

2. (*Procesamiento de consultas*) Estime la cardinalidad del resultado de la siguiente junta

SELECT * FROM R INNER JOIN S ON (R.a = S.a AND S.a >= 2)

Contando con los siguientes metadatos de las tablas

$$n(R) = 8,000$$

$$V(a, R) = 23$$

$$n(S) = 5,000$$

$$V(a, S) = 23$$

Y el histograma con los tres valores más frecuentes de cada tabla:

	1	2	3	4
R.a	1000	700	500	
S.a		1000	700	500

3. (Concurrencia y Transacciones) Dado el siguiente solapamiento de transacciones:

$b_{T1}; b_{T2}; b_{T3}; R_{T1}(X); W_{T2}(Y); W_{T2}(X); R_{T3}(X); W_{T1}(Z); R_{T3}(Z); W_{T3}(Z)$

- Agregue los commits de las transacciones de modo que sea recuperable (si hay más de una forma de hacerlo, alcanza con indicar una única forma).
- Indique cómo deberían haber sido los commits para que no sea recuperable (si hay más de una forma de hacerlo, alcanza con indicar una única forma).
- Indique si el solapamiento es serializable o no.

4. (NoSQL) La AAS (Asociación Argentina de Salud) recauda datos mensuales de cada provincia y los casos de enfermedades raras que se encontraron. Se ejemplifica la estructura de un documento con dichos datos:

```
{
  cod_provincia: "BA",
  mes: 04,
  año: 2024,
  casos: [
    {
      dni_enfermo: 27094817,
      enfermedad: "Esquierica Culkin"
    },
    {
      dni_enfermo: 37553132,
      enfermedad: "Gripe Unicorniar"
    },
    (...)
  ]
}
```

Escriba una consulta en MongoDB que devuelva el código de aquellas provincias que **en 2024** tuvieron al menos 3 enfermedades con más de 50 casos cada una.

5. (*NoSQL*) Mariano está haciendo un video con las mejores jugadas de su equipo de fútbol “Los Pupenautas”. Dado que la duración del video es de aproximadamente 7 minutos, precisa elegir dos temas para musicalizarlo. Entonces hizo lo que cualquiera haría: volcó en una base de datos Neo4J información del spotify de sus compañeros de equipo para poder elegir las dos canciones. Registró datos de los jugadores del equipo y de las canciones:

```
CREATE (j1: Jugador { casaca: 1, nombre: "Luigi Fulciatto" }),
CREATE (c1: Cancion { nombre: "Mami Daddy", banda: "El Chuky feat. el Burger King"}),
```

Hay dos posibles arcos que indican respectivamente si el jugador le dio “me gusta” o “no la soporto” a la canción. Considerar que es posible que el jugador no haya interactuado de ninguna de las dos formas con alguna canción en particular:

```
MATCH (j1: Jugador { casaca: 1 } ), (c1: Cancion {nombre: "Mami Daddy"} )
CREATE (j1)-[:LE_GUSTA]->(c1);
```

```
MATCH (j1: Jugador { casaca: 1 } ), (c2: Cancion {nombre: "Los días sin tu amor"} )
CREATE (j1)-[:NO_SOPORTA]->(c2);
```

Mariano quiere encontrar un par de canciones que en conjunto tengan la mayor cantidad de jugadores que le dieron “me gusta”. Es importante que si a un jugador le gustan ambas canciones no debe contarse dos veces sino una. Y finalmente, pero no por ello menos importante, quiere que para ninguna de las dos canciones haya un jugador que no la soporte.

Escriba una consulta en *Cypher* que resuelva dicho requerimiento

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

Previo a todas las acciones de la derecha, los valores de A, B, C y D en disco eran 0.

Para los cuatro ítems (A, B, C y D) indique si es seguro que tienen un valor en disco (indicando cuál es) o si pueden tener distintos valores (indicando cuáles son los posibles valores).

```
01 (BEGIN, T1);
02 (WRITE T1, A, 1);
03 (COMMIT, T1);
04 (BEGIN, T2);
05 (WRITE T2, B, 5);
06 (BEGIN, T3);
07 (BEGIN CKPT, T2, T3);
08 (WRITE T3, C, 10);
09 (COMMIT, T3);
10 (BEGIN, T4);
11 (WRITE T4, D, 15);
12 (END CKPT);
13 (COMMIT, T4);
```