Base de Datos (75.15 / 95.05 / TA044)

Evaluación Integradora - 26 de febrero de 2025 - **20242C4**

SQL/CRT		**NoSQL**		**NoSQL**		**Padrón:**
Recup		**Conc.**		**Costos**		**Apellido:**
Nota: ☐ Aprobado ☐ 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)* Una asociación de historia está revisando la proliferación de distintas etnias en el país. Pidió a la facultad ayuda, solicitando datos de los estudiantes que le fueron dados en la siguiente tabla:

- **Estudiantes** (padron, nombre, apellido, provincia_nacimiento)

Escriba una consulta que devuelva aquellos apellidos que son tenidos por al menos un estudiante en todas las provincias de nacimiento.
Importante: Para alumnos de 2024 debe resolverse en SQL, para alumnos de cuatrimestres anteriores en Cálculo Relacional de tuplas.

2. *(NoSQL)* Un conocido profesor quiere saber la cantidad de alumnos que aprobaron el parcial por cada valor de nota posible (i.e. un histograma). Guarda todas las notas registradas en una base MongoDB con la siguiente estructura de documento por cada alumno

```
{
   nombre: "Ramón Román", padron: 78888, cuatrimestre: "2do 2024",
   notas_parcial: [{oportunidad:"parcial" , nota:2},
                   {oportunidad:"1er recu", nota:8]
   nota_final: 8
}
```

Escriba una consulta que devuelva para cada nota de **aprobación del parcial** (se aprueba con 4 o más), la cantidad de alumnos que se la sacaron, con la siguiente estructura:

```
{
   nota: 8, cantidad_alumnos: 23
}
```

3. *(NoSQL)* Mabel tiene muchos problemas por ser celíaca. Para poder decidir qué comer, todo un inconveniente constante, hizo lo que cualquiera haría: cargó en una base Neo4J los platos de todos los restaurantes cercanos:

```
CREATE (p1:Plato{ nombre: "Suprema a la Suiza" , tipo:"Principal"} );
CREATE (r1:Restaurante{ nombre:"Negro el 33", direccion:"Chile 654"} );

MATCH (p1:Plato{nombre: "Suprema a la Suiza" })
      , (r1:Restaurante{ nombre:"Negro el 33"})
CREATE (p1)-[:SE_VENDE_EN { precio: 6500 } ]->(r1);
```

Un mismo plato puede venderse en varios restaurantes con distintos precios. Además Mabel carga los ingredientes de todos los platos, indicando si poseen gluten o no (si al menos un ingrediente del plato posee gluten, no es apto celíaco)

```
CREATE (i1:Ingrediente{ nombre:"Pan rallado", tieneGluten: true});
MATCH (p1:Plato{nombre: "Suprema a la Suiza" })
      , (i1:Ingrediente{ nombre:"Pan rallado"})
CREATE (p1)-[:USA]->(i1);
```

Mabel quiere elegir un restorán en el que comer por menos de 9,000 pesos un plato principal y postre. Escriba una consulta en Cypher que devuelva todas las combinaciones de restaurante, plato principal y postre que sean apto celíaco y que su costo total no superen los 9,000 pesos

4. *(Recuperación)* Un SGBD implementa el algoritmo de recuperación REDO con checkpoint activo. En un momento dado, el log tiene los valores que se muestran a la derecha. Indique para cada uno de los ítems (A y B) **qué valor tiene seguramente** en disco, o, en caso de que no haya un valor seguro, **qué valores puede tener**

Considere que inicialmente todos los ítems tenían valor 0

01 (BEGIN, T1);
02 (WRITE T1, A, 1);
03 (BEGIN, T2);
04 (WRITE T2, B, 2);
05 (COMMIT, T1);
06 (BEGIN CKPT, T2);
07 (WRITE T2, A, 3);
08 (COMMIT, T2);
09 (END CKPT);
10 (BEGIN, T3);
11 (WRITE T3, B, 4);

5. (*Concurrencia*) Dado el siguiente solapamiento:

b_{T1} ; b_{T2} ; $W_{T1}(X)$; $W_{T2}(Y)$; ____ ; $R_{T1}(Y)$; c_{T2} ; c_{T1}

Indique qué instrucción se puede agregar en el lugar donde hay una línea. para que el solapamiento **no quede recuperable**. Sólo se puede agregar un Read o Write sobre los ítems X o Y y sólo para transacciones T1 o T2. Agregando esa instrucción, **¿quedó serializable el solapamiento?**
En caso de no ser posible, justifíquelo

6. *(Costos)* El reconocido compositor Johan Sebastian Mastrobeiro está buscando obras con las que se pueda "inspirar" para sus nuevas creaciones. Lo hace con la siguiente consulta en una base de datos relacional:

```
SELECT *
FROM compositores c INNER JOIN obras o USING (id_compositor)
WHERE c.año_fallecimiento <= 1800
```

Tiene la siguiente metadata sobre ambas tablas

COMPOSITORES	**OBRAS**
n(compositores) = 100,000	n(obras) = 5,000,000
B(compositores) = 20,000	B(obras) = 500,000
V(id_compositor,compositores) = 100,000	V(id_compositor,obras) = 100,000

Y además sabe que aproximadamente un 90% de los compositores de la base fallecieron **luego** del año 1800.
No hay índices que puedan aprovecharse y se cuenta únicamente con 2,005 bloques de memoria para resolver la consulta
Calcule el **costo** de resolver la consulta y estime la **cantidad de filas** que son devueltas