Base de Datos (75.15 / 95.05 / TA044)

Evaluación Integradora - 11 de diciembre de 2024 - **20242C1**

AR/CRT		**NoSQL**		**NoSQL**		**Padrón:**
Conc.		**Recup.**		**Proc.**		**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. *(AR / CRT)* La Asociación Argentina de Cinéfilos Aficionados guarda información sobre sus socios y la calificación que hacen sobre las películas:

- **Socios** (id_socio, nombre, fecha_nacimiento)
- **Calificaciones**(id_socio, nombre_pelicula, estrellas_calificacion)

Escriba una consulta que devuelva el nombre de la/s pelicula/s que han recibido una calificación de 5 estrellas por todos los socios de la asociación.

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

2. *(NoSQL)* La famosa banda "*Mac y los McMaccers*" quiere recompensar a sus fans. Para ello se conectó a la base Neo4J de la plataforma musical "*EsTopIfy*" que le brinda la información de qué usuarios le dieron "Me gusta" a las canciones que tiene la plataforma:

```
CREATE (c1:Cancion { nombre: "Text Raño", banda: "Mac y los McMaccers", año: 2023} );
CREATE (u1:Usuario { alias:"afasce", nombre:"Adolfo Fabian Asceri"} );

MATCH (u1:Usuario{ alias: "afasce"}), (c1:Cancion{nombre:"Text Raño"})
CREATE (u1)−[:LEGUSTA]−>(c1);
```

La banda quiere regalarle entradas para su próximo show a aquellos usuarios que le dieron "Me gusta" a al menos un 80% de los temas que la banda sacó en el año 2024.
Escriba una consulta en lenguaje Cypher que resuelva el requerimiento. Asuma que no hay dos canciones distintas con mismo nombre.

3. *(NoSQL)* Una reconocida facultad decidió utilizar Cassandra para agilizar la resolución de consultas de notas de los alumnos. Para ello creó una column family con la siguiente definición:

```
CREATE COLUMNFAMILY alumnos(
  padron INT,
  nombre TEXT STATIC,
  apellido TEXT STATIC,
  codigo_materia TEXT,
  fecha DATE,
  nota INT,
  PRIMARY KEY ((padron), codigo_materia, fecha)
);
```

Indique (con una mínima justificación) si es posible o no con dicho esquema resolver las siguientes consultas:

a) Obtener todas las notas del alumno con padrón 100.000.
b) Obtener todas las notas de los alumnos apellidados “Román”.
c) Obtener las notas del alumno con padrón 100.000 en el año 2024.
d) Obtener las notas del alumno con padrón 100.000 en la materia “TA044”.
e) Obtener las notas del alumno con padrón 100.000 en el año 2024 y la materia “TA044”.

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

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

a. Agregue un WRITE de una transacción para que el solapamiento no sea serializable.
b. Agregue un READ de una transacción para que el solapamiento no sea serializable.
c. Agregue los commits de las tres transacciones de modo que el solapamiento sea recuperable.

Importante: Cada ítem es independiente y debe realizarse con el solapamiento original, no con el modificado por ítems anteriores del ejercicio

5. *(Recuperación)* Un SGBD implementa el algoritmo de recuperación UNDO 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, 10);
03 (BEGIN, T2);
04 (WRITE T1, B, 20);
05 (COMMIT, T1);
06 (WRITE T2, B, 30);
07 (BEGIN CKPT, T2);
08 (BEGIN, T3);
09 (WRITE T3, C, 40);
10 (COMMIT, T3)
11 (BEGIN, T4);
12 (WRITE T4, D, 50);
```

6. (*Procesamiento de consultas)* El sitio de apuestas pseudolegales "*Bet Ina*" guarda la siguiente información de las apuestas de sus usuarios en distintos eventos deportivos:

- Usuarios (id, nombre_usuario, apodo, fecha_inscripcion)
- Apuestas (id_evento,id_usuario, tipo_apuesta, monto_apostado)
- Eventos (id, fecha, descripcion, tipo)

```
SELECT *
FROM Usuarios u INNER JOIN Apuestas a ON (u.id = a.id_usuario)
INNER JOIN Eventos e ON (a.id = a.id_evento)
WHERE e.tipo = 'Fútbol Americano';
```

Estime el **costo** de realizar la consulta anterior que busca las apuestas hechas para partidos de fútbol americano. Estime también la **cantidad de filas** que serán devueltas, contando la siguiente información de catálogo y sabiendo que se dispone de mil bloques de memoria disponibles para la operación:

USUARIOS	**APUESTAS**	**EVENTOS**
n(Usuarios) = 2,000	n(Apuestas) = 40.000	n(Eventos) = 10.000
B(Usuarios) = 50	B(Apuestas) = 400	B(Eventos) = 1.000
	V(id_usuario, Apuestas) = 2.000	V(tipo, Eventos) = 100
	V(id_evento, Apuestas) = 10.000	