

Base de Datos (75.15 / 95.05 / TA044)

Evaluación Integradora - 14 de agosto de 2024 - 20241C5

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

1. (*SQL / CRT*) La Asociación Argentina de Cinéfilos Aficionados guarda información sobre las películas que le encantan a sus socios:

- **Socios** (id_socio, nombre, fecha_nacimiento)
- **SocioAdoraPelícula** (id_socio, nombre_pelicula)

Escriba una consulta que devuelva el nombre de la/s película/s que le gustan tanto al socio más joven como al más viejo.

Importante: Para alumnos de 2024 debe resolverse en SQL, para alumnos de cuatrimestres anteriores en Cálculo Relacional de tuplas y únicamente buscar películas que le gustan al socio más joven (no es necesario en CRT buscar que también le gusten al más viejo).

2. (*NoSQL*) La famosa red social “*LinkedOut*” registra los datos de sus usuarios en una base de datos Mongo, con documentos de la siguiente estructura que indican los conocimientos que posee y las cuentas a las que sigue:

```
{
  usuario: "mbeiro",
  nombre: "Martín Bautista Eiroman",
  conocimientos: ["SQL", "NoSQL"]
  direccion: "Pi y Margall 750",
  sigueA: [
    { cuenta: "Tildenture", fecha_desde: 2024-04-01 },
    { cuenta: "Planobant", fecha_desde: 2024-04-03 },
    }, (...)
  ]
}
```

Escriba una consulta en MongoDB cuales son las 5 cuentas más seguidas por usuarios que tienen conocimientos en SQL.

3. (NoSQL) La famosa banda “Eruca Sativa” está buscando otra banda con la que hacer una colaboración. Para ello accedió a la base de datos Neo4J de música de la aplicación “EsTopify” que contiene datos sobre las bandas, sus canciones y los géneros de las mismas, y además de los usuarios de la aplicación:

```
CREATE (b1: Banda{ nombre: "La condena de Caín" , nacionalidad: "Argentina"});  
CREATE (c1: Cancion { nombre: "Def Ghi", año:2024});  
CREATE (g1: Genero { tipo: "Rock"});  
CREATE (u1: Usuario { login: "gianni", nombre: "Gianni Maschiutto", pais: "Italia"});
```

Los tres tipos de arcos existentes indican los géneros de las canciones, las bandas a las que les pertenecen las canciones (una canción es de una única banda) y los usuarios a los que les gusta cada canción (una canción puede gustarle a varios usuarios):

```
MATCH (g1: Genero { tipo: "Rock"}), (c1: Cancion { nombre: "Def Ghi"})  
CREATE (c1)-[:ES_DE]->(g1);
```

```
MATCH (b1: Banda{ nombre: "La condena de Caín" } ), (c1: Cancion { nombre: "Def Ghi"})  
CREATE (c1)-[:PERTENECE_A]->(b1);
```

```
MATCH (u1: Usuario { login: "gianni"}), (c1: Cancion { nombre: "Def Ghi"})  
CREATE (u1)-[:LE_GUSTA]->(c1);
```

Se pide encontrar las bandas que tengan al menos una canción propia con un mismo género musical que alguna canción de la banda “Eruca Sativa”.

De todas esas bandas, se deben devolver únicamente las 5 que tengan la mayor cantidad de usuarios a los que les gusta algún tema de esa banda y también algún tema de “Eruca Sativa”.

4. (Diseño Relacional) El salón de juegos “Jump Ignon” quiere armar una base de datos con las reservas del uso de su salón.

Cada reserva es para un día y horario en particular (tiene diversos slots de 1 hora cada uno desde las 9:00 hasta las 18:00 hs) y puede ser de servicio normal o premium (que incluye distintos snacks y refrigerios).

En la reserva se indican los niños que asistirán, de cada niño es necesario conocer su DNI, nombre, apellido, fecha de nacimiento y nombre y teléfono de un contacto de emergencia (que por suerte, hasta la fecha nunca fue necesario). Además se debe poder registrar un comentario libre por cada niño para indicar, por ejemplo, si tiene algún tipo de alergia.

Genere un esquema de base de datos relacional que permita registrar toda la información de las reservas, sin redundancias. Para cada relación indique sus atributos, las dependencias funcionales que tenga y las claves candidatas.

5. (Procesamiento de consultas) El sitio de publicaciones “Paper View” guarda la siguiente información de los papers de su base de datos:

- Papers (id, título, fecha_publicación, ciudad, categoría, contenido)
- AutoresPapers (id_paper, id_autor)
- Autores (id, nombre, apellido, nacionalidad)

```
SELECT *
FROM Papers p INNER JOIN AutoresPapers ap ON (p.id = ap.id_paper)
INNER JOIN Autores a ON (a.id = ap.id_autor)
WHERE p.categoria = 'Tecnología';
```

Estime el costo de realizar esta junta y 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:

PAPERS	AUTORESPAPERS	AUTORES
n(papers) = 10.000	n(autorespapers) = 20.000	n(autores) = 1.000
B(papers) = 1.000	B(autorespapers) = 200	B(autores) = 50
V(categoria,papers) = 100	V(id_autor, autorespapers) = 1.000	
	V(id_paper, autorespapers) = 10.000	

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):

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 (COMMIT, T1);
04 (BEGIN, T2);
05 (WRITE T2, A, 20);
06 (BEGIN CKPT, T2);
07 (BEGIN, T3);
08 (WRITE T3, B, 30);
09 (COMMIT, T3);
10 (COMMIT, T2);
11 (END CKPT);
10 (BEGIN, T4);
11 (WRITE T4, D, 40);
```