

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

Evaluación Integradora - 3 de julio de 2024 - 20241C1

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

1. (*SQL / CRT*) Una empresa mantiene una tabla de precios donde indica los cambios de precios que se hacen para cada artículo. El precio del artículo es desde la fecha de la fila (inclusive) hasta un día antes de la próxima fecha que haya para el mismo artículo. A continuación el esquema de la tabla y un ejemplo de posibles datos de la misma:

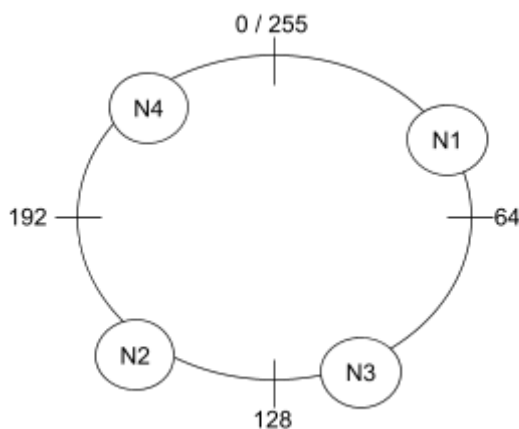
- **Precios** (cod\_art, fecha\_desde, precio)

cod_art	fecha_desde	precio
1111	2020-01-01	800
2222	2020-02-15	1000
2222	2022-08-20	1500
3333	2023-01-10	950

Escriba una consulta que devuelva el precio de todos los artículos el 13 de mayo de 2021 (devolviendo el código de artículo y el precio).

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

2. (*NoSQL*) Se está utilizando hashing consistente para determinar a qué nodos irá cada ítem de la base de datos. Luego de hashear los 4 servidores con una precisión de 8 bits se generó la siguiente estructura:



El hash de N1 fue 59, el de N2 160, el de N3 120 y el de N4 fue 235.

Se sabe que para evitar fallas se guardará cada ítem de la base en dos nodos.

- Indique en qué nodos deberían guardarse los ítems de la base I1, I2 e I3 sabiendo que al hashearles la clave se obtienen los valores 100, 130 y 240 respectivamente.
- Indique en caso de que se caiga el nodo N1 qué cambios deben efectuarse con respecto a qué nodo se hace cargo de qué ítems.

3. (NoSQL) La famosa red social *LinkedOut* registra en su base de datos en *Neo4J* a distintos usuarios y las empresas en las que trabajaron. Los nodos en dicha base pueden ser usuarios:

```
CREATE (u1: User { username: "mbeiro", name:"Marcos Brian Eirosky" } ),
```

o bien empresas:

```
CREATE (c1: Company { companyname: "Universidad de San Juan" }),
```

Los posibles arcos indican si un usuario trabaja actualmente en una empresa:

```
MATCH (u1: User { username: "mbeiro" },
      (c1: Company { companyname: "Universidad de San Juan" })
CREATE (u1)-[:WORKS_IN {start:"2015-01-02", end:"2020-05-31"}]->(c1);
```

y si un usuario conoce a otro usuario:

```
MATCH (u1: User { username: "mbeiro" },
      (u2: User { username: "lramon" })
CREATE (u1)-[:KNOWS]->(u2);
```

Se sospecha que algunas personas tienen más de un usuario en la red. Para intentar detectarlas se busca encontrar pares de usuario que ambos hayan trabajado en al menos 3 mismas empresas y tengan al menos 5 conocidos en común.

Escriba una consulta en *Cypher* que devuelva los pares de usuarios que cumplen lo anterior, sin repetir el mismo par.

4. (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):

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 (BEGIN, T2);
03 (WRITE, T1, A, 10, 14);
04 (WRITE, T2, B, 5, 8);
05 (COMMIT, T1);
06 (BEGIN CKPT, T2);
07 (BEGIN, T3);
08 (WRITE, T3, D, 18, 20);
09 (WRITE, T2, C, 22, 50);
10 (COMMIT, T2);
11 (END CKPT);
12 (BEGIN, T4);
13 (WRITE, T4, A, 14, 30);
14 (WRITE, T4, B, 8, 10);
```

5. (*Procesamiento de consultas*) El Banco *McBancon* está contactando a aquellos clientes que tienen préstamos vencidos para intentar una conciliación previa a la judicialización. Para ello dispone de dos tablas, con los datos de todos los clientes y los datos de los préstamos que ya han vencido:

- **Clientes** (DNI, apellido, nombre, domicilio, telefono)
- **Préstamos Vencidos** (cod\_prestamo, concepto, monto, DNI)

Se desea juntar estas dos tablas por DNI para extraer todos los datos de los clientes con préstamos impagos. Se dispone para ello de dos índices:

- Un índice secundario por DNI en la tabla *Clientes*.
- Un índice de clustering por DNI en la tabla *Préstamos Vencidos*.

Proponga un plan de ejecución que sea eficiente para la junta, indicando cómo será el método de acceso utilizado para cada tabla, y con qué método se realizará la junta. Luego estime el costo del plan propuesto en términos de cantidad de bloques transferidos desde/hacia disco. Asuma que dispone de toda la memoria que necesita para su plan y considere utilizar la siguiente información de catálogo:

CLIENTES	PRESTAMOS_VENCIDOS
$n(\text{Clientes}) = 5.000.000$	$n(\text{Préstamos Vencidos}) = 10.000$
$B(\text{Clientes}) = 500.000$	$B(\text{Préstamos Vencidos}) = 1.000$
$V(\text{DNI}, \text{Clientes}) = 5.000.000$	$V(\text{DNI}, \text{Préstamos Vencidos}) = 5.000$
$H(I(\text{DNI}, \text{Clientes})) = 4$	$H(I(\text{DNI}, \text{Préstamos Vencidos})) = 2$

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

$$b_{T1}; b_{T2}; b_{T3}; b_{T4}; W_{T4}(X); R_{T3}(Y); W_{T1}(X); W_{T2}(Y); c_{T4}; W_{T3}(Z); \\ R_{T1}(Y); W_{T2}(X); c_{T3}; W_{T1}(Z); R_{T4}(X); W_{T2}(Z); c_{T2}; W_{T1}(Y); c_{T1}; c_{T4};$$

- Dibuje el grafo de precedencias del solapamiento.
- Indique si el solapamiento es serializable. Justifique su respuesta.
- Indique si el solapamiento es recuperable. Justifique su respuesta.