

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

Evaluación Integradora - 5 de marzo de 2025 - 20242C5

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

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*) La Asociación Argentina de Curling mantiene datos del resultado de sus miembros en su competencia anual:

- **Resultados** (año, legajo, nombre, apellido, puntaje, puesto)

Escriba una consulta que devuelva el nombre y apellido de aquellos integrantes que, participando en más de un año, siempre hayan salido en el mismo puesto. Considerar que puede haber distintos integrantes con mismo nombre (“Juan Pérez”), pero no con un mismo legajo.

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

2. (*Costos*) El sitio de publicaciones científicas “*Paper View*”, en un intento de “hacerse social”, creó una tabla UserTags(idUser, idTag) que vincula a los usuarios con los tags que les interesa. Quiere hacerle un join natural con la tabla PaperTags(idPaper, idTag), que vincula a cada paper con sus etiquetas, con el objetivo final de sugerir papers que pudieran interesar

El CTO de la empresa decidió, por el volumen de datos, que el join entre ambas tablas debe hacerse utilizando el método de **Sort Merge**, por lo que a la tabla UserTags la ordenó físicamente por id de Tag al momento de crearla. Con la tabla PaperTags no pudo hacerlo porque ya existía de antes y estaba sin orden alguno. Cuenta con los siguientes metadatos de las tablas:

USERTAGS	PAPERTAGS
n(UserTags) = 2,000,000	n(PaperTags) = 900,000
B(UserTags) = 20,000	B(PaperTags) = 9,000
V(idTag, UserTags) = 80	V(idTag, PaperTags) = 100

Sabiendo que se cuenta con  $M = 10,000$ , estimar el **costo** de resolver el join y la **cantidad de filas y bloques** devueltos por el mismo

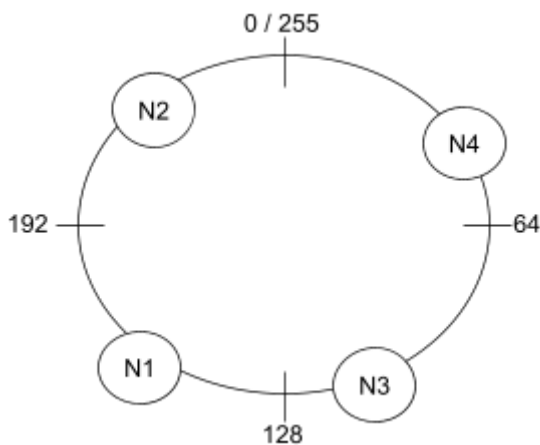
3. (NoSQL) Una aerolínea almacena información sobre sus vuelos diarios y los pasajeros que viajaron en ellos en una base de datos MongoDB. El siguiente documento JSON ejemplifica la estructura almacenada por cada vuelo, que incluye los siguientes campos:

```
{
  cod_vuelo: "AE2840",
  fecha_salida: ISODate("2022-01-18 11:30:00Z"),
  comandante: "GONZÁLEZ, AURELIANO",
  recorrido: [ "EZE", "PAN", "BOS" ],
  distancia_km: 4830,
  pasajeros: [
    {dni: 38291440, nombre: "AUZQUI, BELÉN", asiento: "36A" },
    {dni: 41219085, nombre: "RIZOU, ADOLF", asiento: "28F" },
    (...) ]
}
```

La empresa quisiera obtener, para cada persona que viajó al menos 5000 kilómetros en total, la cantidad de viajes que realizó y la cantidad de kilómetros totales recorridos, con el objetivo de sortear un premio entre ellos (un toallón con el logo de la empresa!).

Escriba una consulta que permita obtener este resultado

4. (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 155, el de N2 235, el de N3 120 y el de N4 fue 60.

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 **10**, **70** y **128** respectivamente.
- Indique en caso de que se caiga el nodo N3 qué cambios deben efectuarse con respecto a qué nodo se hace cargo de qué ítems.

5. (Recuperación) Un SGBD implementa el algoritmo de recuperación REDO con checkpoint activo. Los valores iniciales de A, B y C son 1, 2 y 3 respectivamente.

La transacción T1 intenta grabar el valor 10 en A y el 20 en B, la transacción T2 intenta grabar el valor 30 en A y 40 en C y la transacción T3 intenta grabar 50 en B.

Se produce un corte en el sistema y el log (incompleto) tiene los valores que se muestran a la derecha.

**Complete** el log llenando los valores que faltan y 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, __ );
03 (BEGIN, T2);
04 (WRITE T1, B, __ );
05 (BEGIN CKPT, T1, T2);
06 (COMMIT, T1);
07 (WRITE T2, A, __ );
08 (WRITE T2, C, __ );
09 (END CKPT);
10 (BEGIN, T3);
11 (WRITE T3, B, __ );
12 (COMMIT, T3);
13 (COMMIT, T2);
```

6. (Concurrencia) Dado el siguiente solapamiento:

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

Agregue, en caso de ser posible, los acquire y release de locks necesarios para que el solapamiento pueda producirse con el **protocolo de 2 fases (2PL)**. En caso de no poder hacerlo, justifique brevemente por qué.

Agregue, en caso de ser posible, los acquire y release de locks necesarios para que el solapamiento pueda producirse con el **protocolo de 2 fases estricto (S2PL)**. En caso de no poder hacerlo, justifique brevemente por qué.

Agregue al final del solapamiento (no en el medio) los commits de las transacciones de tal modo que el solapamiento sea **recuperable**. En caso de no poder hacerlo, justifique brevemente por qué.

Importante: Cada uno de los tres sub-ítems es independiente del resto y debe trabajarse con el solapamiento original. En particular en el tercer sub-ítem sólo se pide agregar los commits, independientemente si se puede o no lograr el solapamiento con algún protocolo de lockeo