

# Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 10 de agosto de 2022

<b>TEMA 20221C5</b>						Padrón: _____
DML		Proc.		CyT		Apellido: _____
NoSQL		NoSQL		Ind.		Nombre: _____
Nota:						Cantidad de hojas: _____
						<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. Se aprueba con nota mayor o igual a 4(cuatro), equivalente a desarrollar el 60% del examen correctamente.

1. (*Lenguajes de manipulación de datos*) El operador de *antijoin* ( $\triangleright$ ) es un operador del álgebra relacional cuyo resultado puede declararse en Cálculo Relacional de Tuplas de la siguiente manera: dadas dos relaciones  $R(\bar{A})$  y  $S(\bar{B})$  que poseen un conjunto de atributos en común  $\bar{Y} = \bar{A} \cap \bar{B}$ ,

$$R \triangleright S = \{r | R(r) \wedge (\nexists s)(S(s) \wedge r[\bar{Y}] = s[\bar{Y}])\}$$

- (a) Exprese el operador de *antijoin* en términos de los operadores  $\pi, \sigma, \times, \cup, -, \cap, \bowtie, \div$ .
- (b) Considere ahora las siguientes relaciones que almacenan información sobre tenistas y los torneos que cada uno ganó:
- Tenistas(nombre\_tenista, país, altura, diestro)
  - Torneos(nombre\_torneo, tipo\_torneo)
  - Campeones(nombre\_tenista, nombre\_torneo, modalidad, año)

A partir de la siguiente expresión del álgebra relacional que utiliza el operador de *antijoin*:

$$Tenistas \triangleright (Campeones * \sigma_{tipo\_torneo='GRAND SLAM'}(Torneos))$$

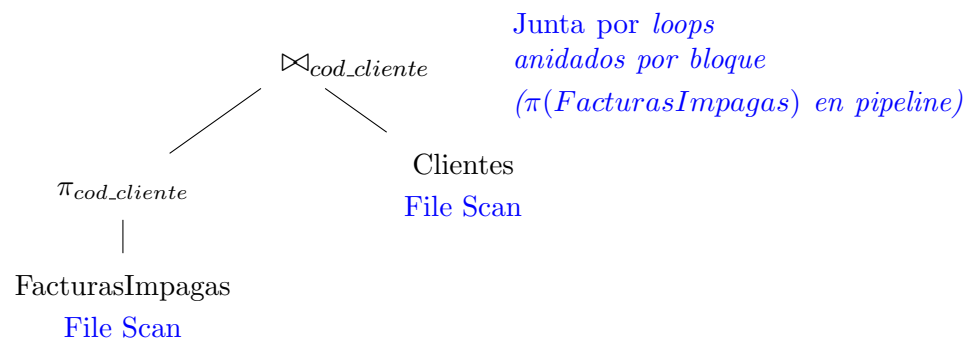
Se pide:

- Exprese en lenguaje coloquial el resultado de dicha consulta.
- Traduzca la consulta al lenguaje SQL.

2. (*Procesamiento de Consultas*) La *Compañía de Electricidad del Este* quiere contactar a aquellos clientes que poseen facturas impagas para informarles que a partir del próximo mes correrá una nueva tasa de financiación. Para ello dispone de las siguientes dos tablas en una base de datos relacional con información de contacto de todos sus clientes, y con los datos de aquellas facturas aún sin pagar:

- **Clientes**(cod\_cliente, apellido, nombre, domicilio, teléfono, mail)
- **FacturasImpagas**(cod\_factura, cod\_cliente, monto, fecha\_vencimiento)

Para encontrar los datos de los clientes que tienen alguna factura impaga se construye el siguiente plan de ejecución:



Ninguna de las dos tablas dispone de índices, por lo que se utilizarán *file scans* como método de acceso. La proyección se realizará por *sort externo*, y en la primera etapa de la misma se descartarán los atributos innecesarios. A medida que se obtengan resultados de la última etapa de la proyección, se irán juntando con la tabla de *Clientes* aprovechando la memoria de la mejor forma posible.

Considerando que la cantidad de memoria disponible es de  $M=20$  bloques y utilizando la información de catálogo que se brinda más abajo, se pide:

- Estime el costo del plan de ejecución generado, en términos de cantidad de bloques.
- Indique si cree que el costo sería mayor o menor si la proyección se hiciera luego de la junta, justificando su respuesta.
- Indique cuál sería el costo de la consulta si se dispusiera de un índice de clustering de altura 2 por el atributo **cod\_cliente** en *FacturasImpagas*, y el mismo se utilizara para reemplazar el *File Scan* de esa tabla por un *Index Scan*. Suponga que los nodos hoja de dicho árbol son los bloques de datos de la tabla.

FACTURASIMPAGAS	CLIENTES
$n(\text{FacturasImpagas}) = 100.000$	$n(\text{Clientes}) = 1.000.000$
$B(\text{FacturasImpagas}) = 10.000$	$B(\text{Clientes}) = 100.000$
$V(\text{cod\_cliente}, \text{FacturasImpagas}) = 20.000$	

Nota: Considere que el atributo **cod\_cliente** representa el 10% del tamaño de una tupla de *FacturasImpagas*.

3. (*Concurrencia y Transacciones*) Considere el siguiente solapamiento de transacciones:

Transacción $T_1$	Transacción $T_2$	Transacción $T_3$
begin leer_item(X)		
		begin leer_item(Y) leer_item(Z)
escribir_item(X)	begin	
leer_item(Z)	escribir_item(Z)	
commit		escribir_item(X) commit
	escribir_item(Y) commit	

Se pide:

- Indique si el solapamiento es serializable.
  - Indique si el solapamiento es recuperable.
  - Indique si este solapamiento –si bien no muestra instrucciones de *Lock()* y *Unlock()*– podría haberse obtenido aplicando el Protocolo de Lock de 2 Fases. Justifique su respuesta.
  - En caso de que se intentara ejecutar este solapamiento bajo el método de control de concurrencia basado en *timestamps*, indique si alguna transacción deberá ser abortada. Asuma que los timestamps asignados a las transacciones están ordenados en forma creciente por el instante en que se ejecuta el **begin**.
4. (*NoSQL*) Indique si las siguientes afirmaciones sobre la implementación de hashing consistente de *Dynamo* son verdaderas o falsas, justificando su respuesta.
- El valor asociado a una determinada clave puede estar registrado en más de un nodo.
  - Cuando un nodo es consultado por el valor asociado a una determinada clave, el mismo siempre devolverá el valor que posee registrado de esa clave, sin consultar a ningún otro nodo.
  - El uso de hashing consistente es lo que permite que Dynamo ofrezca un nivel de consistencia eventual.
  - Cuando un nodo se cae, las consultas sobre valores asociados a claves que eran manejadas por ese nodo ya no serán respondidas, hasta que ese nodo vuelva a levantarse.
  - Todos los nodos administran un rango de claves del mismo tamaño.

5. (*NoSQL*) En el juego conocido como el *Gran DT* cada usuario registrado arma un equipo compuesto de 15 jugadores, y suma puntos en función del desempeño de esos jugadores durante el campeonato. Los organizadores del torneo cargaron en una base en Neo4j todos los equipos de fantasía que se registraron. En esta base de datos existen 4 tipos de nodo: EQ\_FANTASIA, JUGADOR, ESCUADRA y PARTIDO, que poseen los atributos indicados a continuación:

```

1 (incons:EQ_FANTASIA {nombre: 'Los Inconsistentes', mail: 'araindo@ion.com'})
2 (armani:JUGADOR {nombre: 'Franco Armani', posición: 'arquero'})
3 (river:ESCUADRA {nombre: 'Club Atlético River Plate'})
4 (p5_3:PARTIDO {fecha: 5, nro.orden: 3})

```

La información sobre la composición de cada equipo de fantasía se registra a través de la interrelación INCLUYE:

```

1 (incons:EQ_FANTASIA)-[:INCLUYE]->(armani:JUGADOR)

```

Mientras que la información sobre los partidos disputados por los jugadores, las escuadras a las que pertenecen, y la puntuación recibida por cada jugador se registran en la interrelaciones JUEGA y JUEGA\_EN:

```

1 (armani:JUGADOR)-[:JUEGA {puntaje: 6.2}]->(p5_3:PARTIDO)
2 (armani:JUGADOR)-[:JUEGA_EN]->(river:ESCUADRA)
3 (river:ESCUADRA)-[:JUEGA]->(p5_3:PARTIDO)

```

Por último, cuando un jugador es expulsado en un partido, ésto se registra en la interrelación EXPULSADO\_EN:

```

1 (rojo:JUGADOR)-[:EXPULSADO_EN {minuto: 48}]->(p5_3:PARTIDO)

```

Habiendo concluido el torneo, los organizadores quieren armar la tabla final de puntajes de cada equipo de fantasía. El puntaje de cada equipo de fantasía se calcula sumando las puntuaciones recibidas por su plantel en los distintos partidos jugados. Sin embargo, para estimular el *Fair Play*, este año los organizadores impusieron que los jugadores expulsados durante un partido no otorgarán puntaje alguno al equipo de fantasía en ese partido (por más que tengan un puntaje mayor a cero asignado en el partido).

Escriba una consulta en *Neo4j* que encuentre para cada equipo de fantasía el total de puntos acumulados por dicho equipo de fantasía a lo largo del torneo, devolviendo los resultados ordenados en forma decreciente por el puntaje total del equipo de fantasía.

6. (*Índices*) Cuando se crea una nueva tabla, el gestor de bases de datos relacional *PostgreSQL* crea automáticamente un índice por la clave primaria de la misma. Explique con qué fin/és cree que *PostgreSQL* construye ese índice.

Por otra parte, una vez creada la tabla el usuario que posea permisos suficientes puede crear otros índices sobre otros conjuntos de atributos de esa misma tabla si lo desea. Explique qué ventajas y desventajas tiene la creación de nuevos índices para la realización de consultas o para ABM's (altas/bajas/modificaciones) sobre esa tabla.