

**Base de Datos (75.15 / 75.28 / 95.05)**

Evaluación Integradora - 25 de julio de 2018

<b>TEMA 20181C4</b>						Padrón: _____
<b>CRT</b>		<b>DR</b>		<b>Proc.</b>		Apellido: _____
<b>CyT</b>		<b>Rec.</b>		<b>DW</b>		Nombre: _____
Corrigió:						Cantidad de hojas: _____
<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. Se aprueba con nota mayor o igual a 4(cuatro), equivalente a desarrollar el 60 % del examen correctamente.

1. (*Cálculo Relacional de Tuplas*) El comité organizador del festival *Cerveza Abierta* conserva el registro de todas las degustaciones que se realizaron durante la última edición en una base de datos con las siguientes tablas:

- Visitantes(numero\_inscripción, apellido, nombre, teléfono)
- Cervezas(marca, nombre, tipo)
- Degustaciones(numero\_inscripción, fecha\_hora, marca, nombre)

Escriba una consulta en Cálculo Relacional de Tuplas que encuentre el número de inscripción, apellido y nombre de aquellos visitantes para los cuales todas las cervezas que hayan degustado sean de una misma marca.

Nota: En otras palabras, el visitante debe haber probado a lo sumo una marca de cerveza.

2. (*Diseño Relacional*) Dada una relación  $R(A, B, C, D)$ :
- a) Proponga un conjunto de dependencias funcionales  $F$  asociado a  $R$  de manera tal que  $R$  no satisfaga la Segunda Forma Normal (2FN).
  - b) A partir del conjunto de dependencias definido, obtenga una descomposición de  $R$  a Forma Normal Boyce-Codd (FNBC) sin pérdida de información.

3. (*Procesamiento de Consultas*) La AFIP está investigando el patrimonio de los funcionarios públicos y analizando la composición de sus carteras de inversiones en Fondos Comunes de Inversión. Para ello cuenta con las siguientes dos tablas:

- `Suscripciones(CUIT_funcionario, nombre_fondo, cuotapartes)`
- `Composiciones(nombre_fondo, tipo_activo, nombre_activo, porcentaje)`

La relación *Suscripciones* indica qué Fondos Comunes de Inversión (identificados a través del atributo `nombre_fondo`) suscribió cada funcionario (identificado a través de su CUIT), y más concretamente cuántas cuotapartes suscribió en cada uno de ellos.

La relación *Composiciones* indica para cada Fondo Común de Inversión cuáles son los activos que lo componen (identificados por su tipo y nombre), y en qué porcentaje.

Para encontrar el CUIT, apellido y nombre de los funcionarios que poseen acciones del activo *Banco Merluza S.A.*, un empleado de la AFIP comienza por ejecutar la siguiente consulta SQL (*Consulta #1*) para construir una tabla que indique los activos que cada funcionario posee:

```
CREATE TABLE ActivosFuncionarios AS
(SELECT *
FROM Suscripciones s, Composiciones c
WHERE s.nombre_fondo=c.nombre_fondo);
```

Su objetivo es ejecutar luego la *Consulta #2*:

```
SELECT DISTINCT CUIT_funcionario, apellido, nombre
FROM ActivosFuncionarios a
WHERE a.tipo_activo='Acciones' AND a.nombre_activo='Banco Merluza S.A.';
```

El SGBD ofrece la siguiente información de catálogo, que incluye histogramas sobre la distribución del atributo `nombre_fondo`:

SUSCRIPCIONES	COMPOSICIONES
$n(\text{Suscripciones}) = 200000$	$n(\text{Composiciones}) = 40000$
$B(\text{Suscripciones}) = 20000$	$B(\text{Composiciones}) = 4000$
$V(\text{nombre\_fondo}, \text{Suscripciones}) = 1000$	$V(\text{nombre\_fondo}, \text{Composiciones}) = 1000$

	Fondo Caimán	ExtraFondo BancaT	LibreFondo	otros
Suscripciones.nombre_fondo	30000	50000	20000	100000
Composiciones.nombre_fondo	500	300	400	38800

A partir de esta información, el funcionario estimó la cardinalidad de la tabla *ActivosFuncionarios* en  $n(\text{ActivosFuncionarios}) = \frac{200000 \cdot 40000}{1000} = 8$  millones de tuplas, su factor de bloque como  $F(\text{ActivosFuncionarios}) = \left(\frac{1}{F(S)} + \frac{1}{F(C)}\right)^{-1} = 5$ , y finalmente su tamaño en bloques en  $B(\text{ActivosFuncionarios}) = 1,6$  millones de bloques.

El SGBD cuenta con 10000 bloques disponibles en memoria RAM y 5 millones de bloques disponibles en disco. Sin embargo, cuando el funcionario intenta ejecutar la **Consulta #1**, obtiene un error.

Se pide:

- Explique por qué motivo no fue posible hallar la junta entre *Suscripciones* y *Composiciones*, justificando numéricamente su respuesta.
  - Indique si el problema podría solucionarse: (i) aumentando la cantidad de memoria RAM; (ii) aumentando la cantidad de espacio en disco; o (iii) construyendo un índice de clustering sobre una de las tablas. Justifique su respuesta.
  - Proponga una solución alternativa para encontrar el CUIT, apellido y nombre de los funcionarios que poseen acciones de *Banco Merluza S.A.*, que no requiera de ninguna de las opciones mencionadas en el ítem anterior.
4. (*Concurrencia y transacciones*) Considere el siguiente solapamiento de transacciones:

Transacción $T_1$	Transacción $T_2$	Transacción $T_3$	Transacción $T_4$
begin		begin	
leer_item(X)		leer_item(X)	
	begin		
	leer_item(Y)		
	escribir_item(Y)		
		leer_item(Y)	
		escribir_item(Y)	
	commit		
			begin
			leer_item(Y)
			leer_item(Z)
leer_item(Z)			
escribir_item(Z)			
			escribir_item(Y)
			commit
commit		commit	

Responda los siguientes ítems, justificando su respuesta.

- Indique si el solapamiento es serializable.
- Indique si es posible que un solapamiento de lecturas y escrituras como éste se produzca en el contexto de la aplicación del *Protocolo de Lock de 2 Fases (2PL)*.
- Indique si el solapamiento es recuperable.
- Indique si el solapamiento evita *rollbacks* en cascada.

5. (*Recuperación*) Un SGBD implementa el algoritmo de recuperación UNDO/REDO con checkpoint activo. Indique si las siguientes afirmaciones sobre el funcionamiento del algoritmo son verdaderas ó falsas, justificando su respuesta:
- Los ítems de datos modificados por una transacción  $T_i$  deben ser *flushados* a disco antes de escribir (*COMMIT*,  $T_i$ ) en el archivo de *log*.
  - Cuando se modifica un ítem de datos  $X$  es necesario registrar en el *log* tanto su valor anterior como su nuevo valor.
  - Cuando se modifica un ítem de datos  $X$  es obligatorio *flushear* el nuevo valor de  $X$  a disco antes de *flushear* el registro de *log* correspondiente a disco.
  - Si el sistema se reinicia y el algoritmo detecta que una transacción  $T_i$  no había llegado a *commit*, se debe escribir (*ABORT*,  $T_i$ ) en el archivo de *log* y *flusharlo* a disco antes de deshacer las modificaciones realizadas por  $T_i$  en disco.
6. (*Data Warehousing*) Describa el diagrama de estrella (*star schema*) utilizado en el diseño conceptual de *data warehouses*, explicando a través de un ejemplo de qué manera vincula a los *hechos* y las *dimensiones*.