

# Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 6 de diciembre de 2017

<b>TEMA 20172C1</b>						Padrón: _____
<b>CRT</b>		<b>Proc.</b>		<b>DR</b>		Apellido: _____
<b>Seg.</b>		<b>CyT</b>		<b>NoSQL</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*) Uno de los *hobbies* de Alberto es visitar galerías de arte por el mundo. Para organizar sus visitas compiló las siguientes relaciones que describen distintos museos de arte, las obras pictóricas que los mismos poseen, los artistas que las realizaron y la/s corriente/s artística/s a la/s que cada artista perteneció:

- Museos(nombre\_museo, ciudad, país, precio)
- Pinturas(nombre\_pintura, nombre\_artista, nombre\_museo, año)
- Artistas(nombre\_artista, país\_nacimiento, año\_nacimiento)
- Corrientes(nombre\_artista, nombre\_corriente)

Alberto es fanático de la obra del pintor español Pablo Picasso, y por eso escribió la siguiente consulta en Cálculo Relacional de Tuplas (C.R.T.) que le permitía encontrar los museos del mundo que exhiben obras de dicho artista:

$$\{ m.nombre\_museo, m.ciudad, m.país, m.precio \mid Museos(m) \wedge$$

$$(\exists p)(Pinturas(p) \wedge p.nombre\_museo = m.nombre\_museo \wedge$$

$$p.nombre\_artista = \text{'PABLO PICASSO'})$$

$$\}$$

Habiendo visitado todos ellos, se propone ahora encontrar museos que, sin contener obras de Pablo Picasso, contengan al menos una obra de algún pintor que comparta alguna corriente artística con Picasso. Escriba la consulta en Cálculo Relacional de Tuplas (C.R.T.) que le permitirá a Alberto continuar con su exploración.

Nota: Observe que un artista puede estar vinculado con más de una corriente artística.

2. (*Procesamiento de consultas*) La red social *Bareando* conecta a personas que frecuentan bares. Para ello almacena las siguientes relaciones:

- $Visitas(\underline{\text{nombre\_usuario}}, \underline{\text{nombre\_bar}}, \text{cantidad})$
- $Usuarios(\underline{\text{nombre\_usuario}}, \text{localidad}, \text{fecha\_alta})$
- $Bares(\underline{\text{nombre\_bar}}, \text{dirección}, \text{ciudad}, \text{teléfono})$
- $Amistades(\underline{\text{nombre\_usuario\_1}}, \underline{\text{nombre\_usuario\_2}})$

Se desea invitar a conectarse a aquellas personas que hayan visitado bares en común. Como primer paso, se calcula la siguiente junta por igual con el método de junta *hash GRACE*:

$$Encuentros \leftarrow Visitas \bowtie_{Visitas1.nombre\_bar=Visitas2.nombre\_bar} Visitas$$

Para la tabla de *Visitas* poseemos las siguientes estadísticas:

VISITAS
$n(Visitas) = 5.000.000$
$B(Visitas) = 200.000$
$V(\text{nombre\_usuario}, Visitas) = 500.000$
$V(\text{nombre\_bar}, Visitas) = 50.000$

Se pide:

- a) Estime la cardinalidad del resultado de esta junta en términos de cantidad de tuplas.
- b) Estime el costo de la operación en términos de cantidad de accesos a disco.
- c) Si elijo  $m = 100$  particiones para la función de *hash*, ¿cuántos bloques de espacio en memoria debería tener disponibles para poder realizar la operación de junta?
- d) Si a continuación se quisiera hacer la proyección  $\pi_{\text{nombre\_usuario1}, \text{nombre\_usuario2}}(Encuentros)$ , ¿cree que la misma podría hacerse en *pipeline* (es decir, procesando una a una las tuplas a la salida de la junta) ó sería necesario materializar el resultado? Justifique su respuesta.

Nota: Asumimos que el esquema de la relación resultante de la junta es  $Encuentros(\text{nombre\_usuario1}, \text{nombre\_bar1}, \text{cantidad1}, \text{nombre\_usuario2}, \text{nombre\_bar2}, \text{cantidad2})$ .

3. (*Diseño relacional*) Dado un esquema de relación  $R(A, B, C, D)$  con su conjunto de dependencias funcionales  $F = (A \rightarrow B, BD \rightarrow A, C \rightarrow D)$ , indique cuáles de las siguientes desigualdades *deben necesariamente cumplirse* en toda instancia del esquema, cuáles *pueden eventualmente cumplirse* y cuáles *deben necesariamente ser falsas*. Justifique sus respuestas.

- a)  $n(\pi_B(R)) \leq n(\pi_A(R))$
- b)  $n(\pi_{B,C}(R)) \geq n(\pi_A(R))$
- c)  $n(\pi_{A,D}(R)) > n(\pi_B(R))$
- d)  $n(\pi_{B,C}(R)) < n(R)$

Nota: El símbolo  $n(X)$  denota la cardinalidad (cantidad de tuplas) de la relación  $X$ .

4. (*Concurrencia y transacciones*) Se dice que una transacción  $T_2$  realiza una Lectura Sucia ó *Dirty Read* cuando la misma lee un ítem cuyo último valor fue escrito por una transacción  $T_1$  que aún no hizo su *commit*. La Lectura Sucia da lugar a potenciales anomalías si no se toman medidas preventivas. Indique si las siguientes afirmaciones sobre la misma son verdaderas ó falsas. Justifique cada una de sus respuestas.

- a) En un solapamiento recuperable de transacciones puede ocurrir una Lectura Sucia.
- b) Aplicando el Protocolo de Lock de Dos Fases (2PL) puede ocurrir una Lectura Sucia.
- c) Aplicando el Protocolo de Lock de Dos Fases Riguroso (R2PL) puede ocurrir una Lectura Sucia.
- d) Bajo el nivel de aislamiento *Read Committed* definido en el estándar SQL puede ocurrir una Lectura Sucia.

Definiciones:

Protocolo 2PL: Una transacción no puede adquirir ningún *lock* luego de haber liberado un *lock* que había adquirido.

Protocolo R2PL: Una transacción sólo puede liberar un *lock* después de su *commit*.

Recuperabilidad: Un solapamiento es recuperable si y sólo si ninguna transacción  $T$  del mismo realiza su *commit* hasta tanto todas las transacciones que escribieron datos antes de que  $T$  los leyera hayan *commit*-teado.

5. (*Seguridad*) En una base de datos *PostgreSQL*, el usuario *Juan* es el dueño de la tabla *Compras* y otorga todos los privilegios sobre la misma al usuario *Rolo* a través del siguiente comando: `GRANT ALL PRIVILEGES ON TABLE Compras TO Rolo;`. *Juan* tiene roles de *Empleado de Compras* y *Analista* mientras que *Rolo* tiene roles de *Analista* y *Auditor*. Indique si, una vez realizada dicha concesión, las siguientes afirmaciones son verdaderas ó falsas, justificando su respuesta.

- a) El usuario *Rolo* puede ejecutar `DROP Table Compras;` y eliminar la tabla.
- b) El usuario *Rolo* puede hacer un `INSERT` de una nueva fila en *Compras*.
- c) El usuario *Rolo* puede otorgarle privilegios a un tercer usuario *María* ejecutando `GRANT ALL PRIVILEGES ON TABLE Compras TO María;`.
- d) El usuario *Rolo* puede ejecutar `REVOKE ALL PRIVILEGES ON TABLE Compras FROM Juan;` y revocarle los permisos a *Juan*.
- e) Un superusuario puede ejecutar `DROP Table Compras;`.
- f) Cualquier usuario con rol de *Analista* puede hacer un `INSERT` de una nueva fila en *Compras*.

Nota: Suponga que el usuario *Juan* es también el dueño del esquema al que la tabla *Compras* pertenece, y que todos los usuarios tienen permiso de `USAGE` de dicho esquema, aunque nadie tenía inicialmente ningún privilegio adicional.

6. (*NoSQL*) Explique en qué consiste el modelado orientado a las consultas (*query-driven modeling*) en *Cassandra*. Justifique por qué se adopta esta metodología, relacionando brevemente su respuesta con el diseño de *Cassandra* y sus limitaciones.