

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

Evaluación Integradora - 20 de diciembre de 2017

<b>TEMA 20172C2</b>						Padrón: _____  Apellido: _____  Nombre: _____  Cantidad de hojas: _____  <input type="checkbox"/> Aprobado <input type="checkbox"/> Insuficiente
<b>SQL</b>		<b>Proc.</b>		<b>CyT</b>		
<b>Rec.</b>		<b>Rec.II</b>		<b>NoSQL</b>		
Corrigió:  <b>Nota:</b>						

**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. (*SQL*) Una heladería almacena información sobre sus clientes, las ventas que realiza, y los gustos pedidos en cada venta.

- Clientes(nro\_cliente, nombre, domicilio)
- Ventas(nro\_factura, fecha, nro\_cliente, monto)
- Degustación(nro\_factura, gusto)

A la heladería le interesa saber si sus clientes son propensos a probar gustos variados, o si en general se aferran cada uno a unos pocos gustos. Para ello se desea construir un histograma que indique para cada número entero positivo  $i$  la cantidad de clientes que han probado  $i$  gustos distintos desde el 1 de enero de este año. Se pide:

- a) Escriba una consulta en *SQL* que construya una vista **NroGustosPorCliente(nro\_cliente, nro\_gustos)** que para cada cliente que probó algún gusto desde el 1 de enero de 2017 devuelva el número de cliente y la cantidad de gustos distintos que probó.
- b) A partir de la vista anterior, escriba una consulta en *SQL* que devuelva, para cada número entero positivo  $i$ , la cantidad de clientes que probó esa cantidad de gustos distintos desde el 1 de enero de 2017.

Nota: Omitiremos los  $i$  para los cuales la cantidad de clientes es cero.

2. (*Procesamiento de Consultas*) Indiana Jones está tras los pasos de una momia egipcia que acaba de ser robada por un contrabandista en El Cairo, y cuyo potencial comprador sería el famoso multimillonario Khia Salgufi.

Para encontrar al contrabandista, Indiana dispone de un listado mundial de contactos telefónicos vinculados al contrabando de piezas antiguas, entre los que se encontrarían tanto el delincuente buscado como Khia Salgufi. Por otra parte, acaba de conseguir el listado de todas las llamadas telefónicas realizadas desde El Cairo hacia números en todo el mundo durante los últimos 30 días, provistas por la compañía Cairofónica S.A.. Estos datos se resumen en las siguientes dos relaciones:

- `Contactos(telefono, nombre_sosp)`
- `LlamadasCairo(tel_origen, hora_inicio, tel_destino, duracion)`

Nota: Observe que, para proteger su identidad, estas figuras tienen típicamente múltiples teléfonos.

Para ayudar a Indy a encontrar al delincuente, se pide:

- a) Escriba una consulta en álgebra relacional que encuentre los nombres de los sospechosos que hayan establecido llamadas desde El Cairo hacia alguno de los teléfonos vinculados con Khia Salgufi.
- b) Dibuje el árbol de consulta optimizado aplicando todas las reglas de optimización heurística estudiadas. Puede considerar la siguiente información de catálogo si lo cree necesario:

CONTACTOS	LLAMADASCAIRO
$n(\text{Contactos}) = 5.000$	$n(\text{LlamadasCairo}) = 30.000.000$
$B(\text{Contactos}) = 500$	$B(\text{LlamadasCairo}) = 3.000.000$
$V(\text{nombre\_sosp}, \text{Contactos}) = 200$	$V(\text{tel\_origen}, \text{LlamadasCairo}) = 1.000.000$
	$V(\text{tel\_destino}, \text{LlamadasCairo}) = 10.000.000$

- c) Para el árbol de consulta que realizó, indique cuál de los siguientes pares de índices le sería útil para optimizar la ejecución en términos de cantidad de accesos a disco. (*Elija sólo uno de los siguientes 5 ítems*). Justifique su respuesta explicando cómo utilizaría dichos índices en el plan físico de ejecución.
  - Un índice por `Contactos.telefono` y otro por `LlamadasCairo.tel_origen`
  - Un índice por `Contactos.nombre_sosp` y otro por `LlamadasCairo.tel_destino`
  - Un índice por `LlamadasCairo.tel_origen` y otro por `LlamadasCairo.tel_dest`
  - Un índice por `Contactos.telefono` y otro por `Contactos.nombre_sosp`
  - Ninguno de los anteriores

Nota: Suponga que dispone de una cantidad de memoria mínima (2 bloques para *input* y 1 para *output*).

3. (*Concurrencia y transacciones*) Explique brevemente qué es un *deadlock* en el contexto de la ejecución concurrente de transacciones, y cómo puede ser detectado a través del grafo de asignación de recursos.
4. (*Recuperación*) El gestor de *log* de una base de datos utiliza un *log* de tipo UNDO, y por cada instrucción atómica de cada transacción  $T_i$  escribe el correspondiente registro en el archivo de *log*, de acuerdo al siguiente cuadro:

Instrucción de $T_i$	Registro escrito en el <i>log</i>
begin	( <i>BEGIN</i> , $T_i$ )
read_item(X)	( <i>READ</i> , $T_i$ , $X$ )
write_item(X)	( <i>WRITE</i> , $T_i$ , $X$ , $v_{old}$ )
commit	( <i>COMMIT</i> , $T_i$ )
abort	( <i>ABORT</i> , $T_i$ )
lock(X)	
unlock(X)	

Tanto los bloques de disco que guardan ítems como aquellos que guardan el *log* poseen *buffers* en memoria. El *flush* (*volcado*) de estos *buffers* de los bloques a disco es controlado por el sistema operativo, aunque el SGBD puede ordenar el *flush* de un bloque cuando lo desea. En base a ésto, indique si cada una de las siguientes afirmaciones es verdadera ó falsa, justificando su respuesta.

- a) El registro de *log* correspondiente a un *write\_item(X)* debe ser volcado a disco después de volcar el nuevo valor de  $X$  modificado por dicha instrucción a disco.
  - b) El registro de *log* correspondiente al *commit* de una transacción  $T_i$  debe ser volcado a disco después de volcar el nuevo valor de todo ítem  $X$  modificado por  $T_i$  a disco.
  - c) En caso de recuperación y necesidad de *rollback* de una transacción  $T_i$ , el registro de *log* correspondiente al *abort* de  $T_i$  debe ser volcado a disco antes de volcar a disco los valores  $v_{old}$  de cada ítem previos a la ejecución de  $T_i$ .
5. (*Recuperación II*) Explique brevemente cuál es la utilidad de colocar *checkpoints* en el archivo de *log* de una base de datos. Escoja uno de los tres algoritmos de recuperación estudiados (UNDO, REDO ó UNDO/REDO) y explique el proceso de construcción de un checkpoint activo desde que se escribe el (BEGIN CKPT, ...) hasta que se escribe el (END CKPT).

6. (*NoSQL*) Los vikingos se están preparando para las próximas celebraciones de verano y se han repartido la recolección de provisiones como leña, hongos, frutos y peces. Para llevar el control de lo recolectado instalaron una base de datos en *Cassandra* en la que definieron las familias de columnas *EspecialidadesPorVikingo*, *RecursosPorVikingoTipo* y *RecursosPorTipoVikingo* de la siguiente manera:

```
CREATE TABLE EspecialidadesPorVikingo (  
    nombre_vikingo text,  
    especialidad text,  
    asentamiento_origen text static,  
    primary key ((nombre_vikingo), especialidad));
```

```
CREATE TABLE RecursosPorVikingoTipo (  
    nombre_vikingo text,  
    tipo_recurso text,  
    cantidad int,  
    primary key ((nombre_vikingo), tipo_recurso));
```

```
CREATE TABLE RecursosPorTipoVikingo (  
    tipo_recurso text,  
    nombre_vikingo text,  
    cantidad int,  
    primary key ((tipo_recurso), nombre_vikingo));
```

Para cada una de las siguientes consultas, indique si la misma es válida en lenguaje *CQL* y si puede ser ejecutada o no a partir del diseño anterior. En aquellos casos en que no sea posible, reformule la consulta y/o construya una nueva familia de columnas para que se pueda obtener el resultado pedido.

- a) Listar los nombres de los vikingos especialistas en caza:

```
SELECT nombre_vikingo  
FROM EspecialidadesPorVikingo  
WHERE especialidad='CAZA';
```

- b) Encontrar la cantidad total de leña recolectada por vikingos provenientes de Kattegat:

```
SELECT sum(cantidad)  
FROM RecursosPorVikingoTipo, EspecialidadesPorVikingo  
WHERE RecursosPorVikingoTipo.nombre_vikingo =  
    EspecialidadesPorVikingo.nombre_vikingo  
AND EspecialidadesPorVikingo.asentamiento_origen = 'KATTEGAT'  
AND RecursosPorVikingoTipo.tipo_recurso = 'LEÑA';
```

- c) Para cada vikingo que proveyó hongos, encontrar su nombre y la cantidad provista:

```
SELECT nombre_vikingo, cantidad  
FROM RecursosPorTipoVikingo  
WHERE tipo_recurso = 'HONGOS';
```