Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 17 de julio de 2019

TEMA 20191C3					
DML		**Proc.**		**DR**	
Esp.		**Rec.**		**DW**	
Corrigió: **Nota:**					

Padrón: _____

Apellido: _____

Nombre: _____

Cantidad de hojas: _____

☐ Aprobado ☐ 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 *semijoin* ($\ltimes$) 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 $\bar{Y} = \bar{A} \cap \bar{B}$,

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

   a) Exprese el operador de *semijoin* en términos de los operadores $\pi, \sigma, \times, \cup, -, \cap, \bowtie, \div$.

   b) Considere ahora las siguientes relaciones que almacenan información sobre músicos, bandas musicales y álbumes:

   - `Artistas(id_artista, nombre_artista)`
   - `Músicos(id_artista, fecha_nac)`
   - `Bandas(id_artista, fecha_creación)`
   - `Integra(id_músico, id_banda)`
   - `Álbumes(id_álbum, nombre_álbum, id_artista)`

   Tanto los *Músicos* como las *Bandas* se identifican a través de un `id_artista`, mientras que la composición de cada banda se refleja en la relación `Integra`. Cada álbum es obra de un único artista, que puede ser o bien una *Banda* o bien un *Músico*, y que se identifica a través de la clave foránea `id_artista` que hace referencia a la relación `Artistas`.
   A partir de la siguiente expresión del álgebra relacional que utiliza el operador de *semijoin*:

$$\mathit{Álbumes} \ltimes (\sigma_{\mathit{fecha\_nac} > 01-01-2000}(\mathit{Músicos}) \bowtie_{\mathit{id\_artista}=\mathit{id\_músico}} \mathit{Integra})$$

   Se pide:

   i) Exprese en lenguaje coloquial el resultado de dicha consulta.

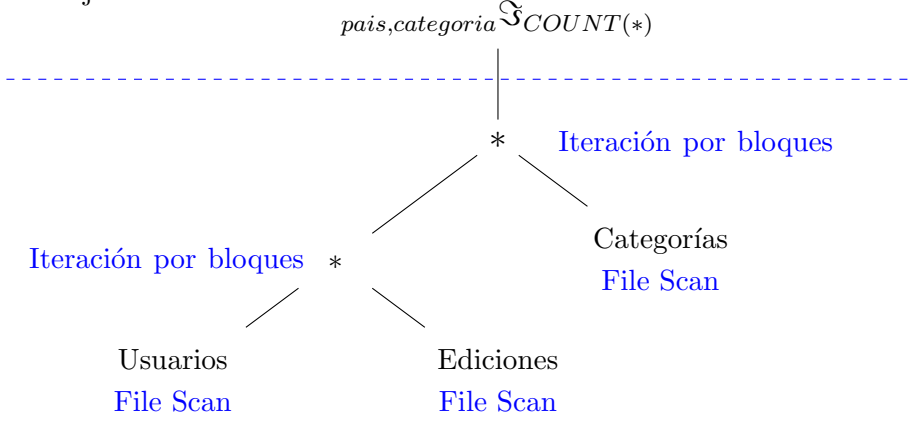
   ii) Traduzca la consulta al lenguaje SQL.

Nota: La relación entre *Artistas* y *(Músicos, Bandas)* es una generalización/especialización total y disjunta.

2. *(Procesamiento de Consultas)* Los revisores de *Wikipedia* poseen las siguientes tablas con información sobre todas las ediciones realizadas sobre todos los artículos en español, indicando qué usuarios las hicieron, de qué países provienen, y a qué categoría/s pertenece cada artículo:

   - `Usuarios(id_usuario, país)`
   - `Ediciones(id_documento, num_rev, id_usuario, diff)`
   - `Documentos(id_documento, título, texto)`
   - `Categorías(id_documento, categoría)`

   A los revisores les interesa obtener una tabla que indique cuántas modificaciones sobre cada categoría se hicieron desde cada país. Dicha estadística puede interpretarse como el resultado del siguiente plan de ejecución:



   En donde el operador $\Im$ del álgebra relacional extendida representa la agregación sobre los atributos *(país, categoría)* con el objetivo de contar la cantidad de ediciones dentro de cada grupo. Se pide:

   a) Estime la cardinalidad del resultado a la altura de la línea punteada (es decir, previo a la agregación) en términos de cantidad de tuplas y de cantidad de bloques.

   b) Suponiendo que dispone de 1 millón de bloques de memoria, indique de qué modo podría administrar su uso en el plan de ejecución que muestra la figura (considerando únicamente la ejecución hasta la línea punteada[1]), de manera de minimizar el costo de la junta de las 3 tablas en términos de entrada/salida. Señale si en algún momento puede hacer *pipelining*, e indique cuál sería el costo mencionado.

   [1] Es decir, omitiendo la agregación final.

   c) *(Bonus track)* Proponga un procedimiento para calcular la agregación final a partir de la salida de la junta. ¿Cree que dicha agregación podría hacerse en pipeline? Realice las hipótesis que considere necesarias e indique cuál sería a grandes rasgos el costo del procedimiento que propone.

   Para la resolución, considere que ninguna de las tablas dispone de un índice, y utilice la siguiente información de catálogo:

USUARIOS	EDICIONES	CATEGORIAS
n(Usuarios) = 5.000.000	n(Ediciones) = 100.000.000	n(Categorías) = 4.000.000
B(Usuarios) = 100.000	B(Ediciones) = 50.000.000	B(Categorías) = 400.000
V(país, Usuarios) 200	V(id_usuario, Ediciones) = 4.000.000	V(id_documento, Categorías) = 2.000.000
	V(id_documento, Ediciones) = 2.000.000	V(categoría, Categorías) = 40.000

3. *(Diseño relacional)* Traduzca el esquema de base de datos relacional del *Ejercicio 1* a un diagrama *Entidad-Interrelación (ER)*. Para mayor facilidad se transcribe a continuación el esquema:

   - `Artistas(id_artista, nombre_artista)`
   - `Músicos(id_artista, fecha_nac)`
   - `Bandas(id_artista, fecha_creación)`
   - `Integra(id_músico, id_banda)`
   - `Álbumes(id_álbum, nombre_álbum, id_artista)`

4. *(Bases de datos espaciales)* Mencione dos estructuras de datos que puedan utilizarse para indexar datos espaciales multidimensionales, explicando brevemente su funcionamiento y qué tipo de objetos geométricos pueden indexar.

5. *(Recuperación)* Un SGBD implementa el algoritmo de recuperación REDO con checkpoint activo. Luego de una falla, el sistema encuentra el siguiente archivo de log:

```
01 (BEGIN, T1);
02 (WRITE T1, X, 8);
03 (WRITE T1, Y, 3);
04 (BEGIN, T2);
05 (WRITE T2, Z, 2);
06 (WRITE T2, W, 4);
07 (COMMIT, T2);
08 (BEGIN CKPT, T1);
09 (BEGIN, T3);
10 (WRITE T3, W, 1);
11 (COMMIT, T1);
12 (END CKPT);
13 (WRITE T3, X, 2);
```

   Explique cómo se llevará a cabo el procedimiento de recuperación, indicando qué cambios deben ser realizados en disco y en el archivo de log.

6. *(Data Warehousing)* Explique brevemente en qué consisten las cláusulas `GROUPING SETS`, `ROLLUP` y `CUBE` del lenguaje SQL, indicando cuál es su funcionalidad y qué ventajas ofrece su uso.