

# Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 27 de febrero de 2019

<b>TEMA 20182C5</b>						Padrón: _____
SQL		Proc.I		Proc.II		Apellido: _____
CyT		Rec.		NoSQL		Nombre: _____
Corrigió:						Cantidad de hojas: _____
Nota:						<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. (*SQL*) Una pareja de desarrolladores de software ha decidido optimizar sus compras en supermercados de manera de gastar lo mínimo posible. Para ésto programaron una serie de *scripts* que realizan el procesamiento –en inglés, *scraping*– de los sitios web de varios supermercados, generando las siguientes dos tablas:

- Productos(cod\_barras, nombre\_producto)
- Precios(nombre\_supermercado, cod\_barras, precio)

Mientras los *scripts* corrían, ellos prepararon la siguiente tabla con el listado de compras para mañana:

- Compras(cod\_barras, unidades)

Escriba una consulta en lenguaje SQL que encuentre para cada producto del listado de compras el supermercado en que el mismo se encuentre al precio más económico, indicando: *el código de barras, el nombre del producto, el nombre del supermercado en que el mismo se encuentra al precio más conveniente y la cantidad de unidades a comprar.*

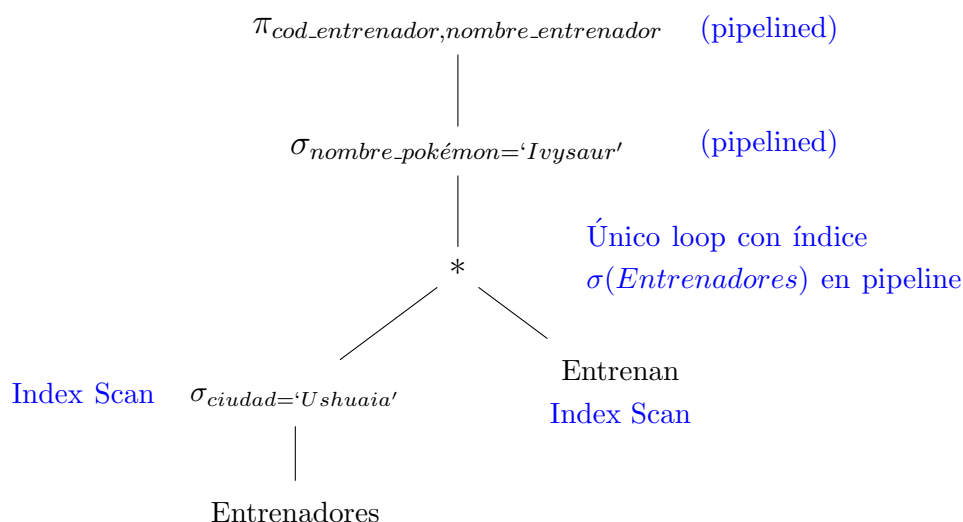
Muestre el resultado ordenado por nombre del supermercado y, en caso de que un producto se encuentre al precio más económico en más de un supermercado a la vez, desempate entre estos últimos suponiendo que sus nombres son comparables por  $<$ .

2. (*Procesamiento de Consultas I*) Mencione una ventaja y una desventaja de estimar la cardinalidad del resultado de una junta  $R \bowtie_{R.A=S.A} S$  utilizando histogramas en lugar de utilizar la fórmula  $\frac{n(R) \cdot n(S)}{\max\{V(A,R), V(A,S)\}}$ .

3. (*Procesamiento de Consultas II*) Leonor es fanática de Pokémon y quiere encontrar otros entrenadores en su ciudad que, al igual que ella, estén entrenando un Ivysaur. Acaba de acceder a una base de datos que contiene información sobre todos los entrenadores de Pokémon y las especies de pokémon que cada uno de ellos entrena:

- `EspeciesPokemons(nombre_pokémon, generación, peso, altura)`
- `Entrenan(cod_entrenador, nombre_pokémon, fecha_desde)`
- `Entrenadores(cod_entrenador, nombre_entrenador, ciudad)`

Además, cuenta con un *índice de clustering* para la tabla *Entrenadores* por el atributo *ciudad*, mientras que la tabla *Entrenan* cuenta con un *índice secundario* sobre *cod\_entrenador*. Para buscar otros entrenadores de Ivysaurs en la ciudad de Ushuaia, preparó el siguiente plan de ejecución:



Estime el costo del plan de ejecución en términos de cantidad de accesos a bloques de disco, suponiendo que dispone de una cantidad mínima de memoria. Considere para el cálculo la siguiente información de catálogo:

ENTRENADORES	ENTRENAN
$n(\text{Entrenadores}) = 1.000.000$	$n(\text{Entrenan}) = 4.000.000$
$B(\text{Entrenadores}) = 100.000$	$B(\text{Entrenan}) = 400.000$
$V(\text{ciudad}, \text{Entrenadores}) = 1000$	$V(\text{cod\_entrenador}, \text{Entrenan}) = 1.000.000$
	$V(\text{nombre\_pokémon}, \text{Entrenan}) = 800$

*Nota:* Asuma que los índices existentes son de tipo árbol y poseen una altura de 4.

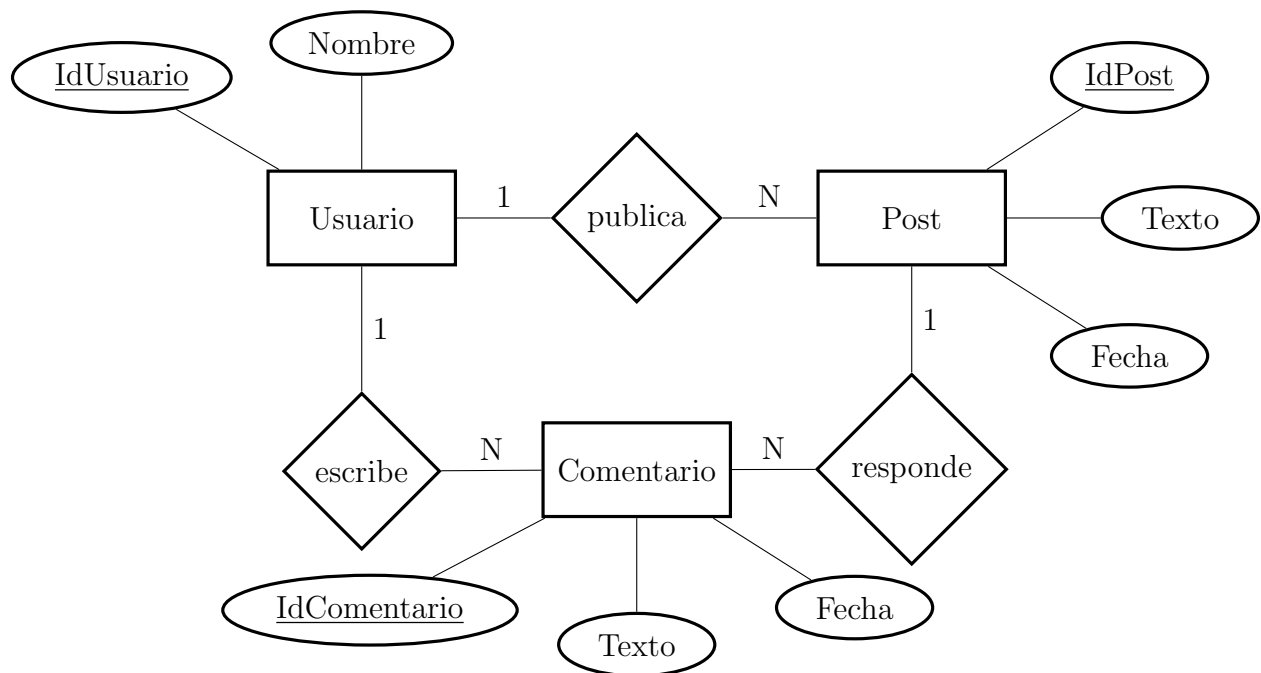
4. (*Concurrencia y Transacciones*) Suponga que en una base de datos con un cierto estado inicial y sin transacciones en curso comienzan a ejecutarse en forma solapada tres transacciones  $T_1, T_2$  y  $T_3$ . Indique si las siguientes afirmaciones relativas a dicha ejecución concurrente son verdaderas ó falsas, justificando su respuesta.

- a) Si el solapamiento es equivalente por conflictos a la ejecución serial  $T_1 \rightarrow T_3 \rightarrow T_2$ , entonces no puede ser equivalente por conflictos también a la ejecución serial  $T_2 \rightarrow T_3 \rightarrow T_1$ .
- b) Si el solapamiento es equivalente por resultado a la ejecución serial  $T_1 \rightarrow T_3 \rightarrow T_2$ , entonces es también equivalente por conflictos a dicha ejecución serial.
- c) Si el grafo de precedencias resultante de la ejecución incluye el arco dirigido  $T_2 \rightarrow T_3$ , entonces no puede incluir el arco dirigido  $T_3 \rightarrow T_2$ .
- d) Si cada vez que una transacción utiliza un recurso lo hace manteniendo un *lock exclusivo* sobre ese recurso, entonces el solapamiento será serializable.
- e) Si el grafo de precedencias resultante de la ejecución es *acíclico*, el mismo puede tener más de un ordenamiento topológico.

Nota: Suponga que en principio desconocemos si el SGBD garantiza la serializabilidad ó no.

5. (*Recuperación*) Cuando un SGBD inicia un *checkpoint* sobre el archivo de *log*, escribe la línea (BEGIN CKPT,  $\{T_{act}\}$ ) indicando en  $\{T_{act}\}$  el listado de transacciones activas en ese momento. Explique qué deberá hacer el SGBD previo a poder concluir el *checkpoint* (es decir, a escribir (END CKPT) en el archivo de *log*), según se trate de un *log* UNDO, REDO ó UNDO/REDO.

6. (NoSQL) Considere el siguiente diseño de una red social en que los usuarios pueden publicar contenido y/o comentar el contenido posteoado por otros usuarios:



Se pide:

- a) Diseñe un conjunto de familias de columnas en *Cassandra* que permita resolver las siguientes dos consultas:
- 1) Dado un IdUsuario, encontrar sus últimos 3 posts, indicando el texto del post y la fecha.
  - 2) Dado un IdPost, encontrar los últimos 5 comentarios realizados sobre el post, indicando el *id* del usuario que realizó el comentario, el texto del comentario y la fecha.

Para resolver este ítem puede ya sea mostrar el diagrama *Chebotko* de cada familia de columnas ó escribir las sentencias `CREATE COLUMNFAMILY` respectivas.

- b) A partir del diseño realizado, traduzca cada una de las dos consultas al lenguaje *CQL* (*Cassandra Query Language*).