

2do Parcial - 1er Cuatrimestre 2016 - Base de Datos 24 de Junio 2016

- Numere las hojas entregadas. Complete en la primera hoja la cantidad total de hojas.
- Cada ejercicio debe realizarse en **hojas separadas y numeradas**. Debe identificarse cada hoja con nombre, apellido y LU.
- Cada tema tiene un criterio de aprobación y se recupera por tema. **NO DEBEN COMETERSE ERRORES CONCEPTUALES GRAVES**.
- La devolución de los exámenes corregidos es personal. Los pedidos de revisión se realizarán por escrito, antes de retirar el examen corregido del aula.
- El parcial es a libro abierto. **Justifique** sus respuestas

1 NOSQL

Criterio de aprobación: se aprueba con 6.5 pts El ítem a) vale 4 pts, el ítem b) 3 pts y el ítem c) 3 pts. (debe tener al menos 1 punto en cada ítem)
Debe justificar los modelos elegidos.

- a) Dado el DER de la figura, donde se modela un conjunto de redes de sensores con sus mediciones.

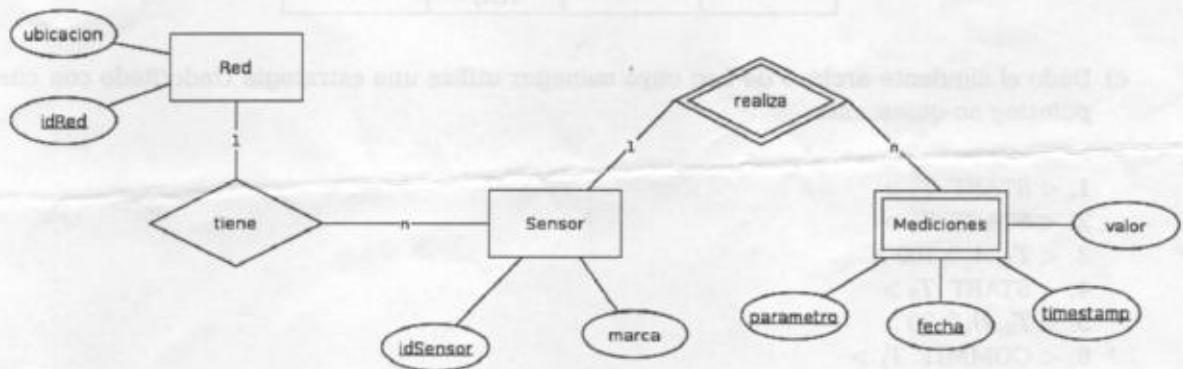


Figure 1: DER Artículos y Autores

Se pide que realice el un modelo para una base de datos de Column-Family tal que responda las siguientes consultas.

- Buscar todas las mediciones para un parametro dado en una ubicación determinada. El parametro es por ejemplo temperatura, presión, humedad y la ubicación es en donde se encuentra la Red.
 - El número de mediciones por sensor en un rango de Fechas y para una Red dada. O sea debo poder obtener, por ejemplo, que para la red 1 entre el 5/6/2015 y el 4/8/2015 que el sensor 1 tiene 10000 mediciones, el sensor 2 tiene 50000 mediciones, etc.
- b) Realice un modelo de documentos para el DER anterior asumiendo que la *cantidad de mediciones de un sensor es enormemente grande*, pero en cada fecha sólo hay una cantidad fija y manejable de mediciones. Las consultas a responder son: ¿Cuántas marcas diferentes de sensores hay en una red? ¿Cuales son las mediciones de un Sensor en un día particular?
- c) Asuma que con el mismo modelo conceptual del punto a) tiene un documento por cada sensor y fecha (es decir por ejemplo que para el sensor 1 el 10/6/2000 hay un sólo documento) con todas las mediciones. Realice una consulta map reduce que devuelva el valor más alto de cada parámetro en cada fecha a partir del 1/7/2015.

Justifique todas sus respuestas.

2 Concurrencia y Recuperabilidad

Criterio de aprobación: se aprueba con 6.5 pts. Valores parciales: a) 4, b) 3, c) 3. Debe tener **al menos algo correcto en cada ítem**

a) Se tienen las siguientes transacciones:

$$T_1 = rl_1(B); wl_1(A); u_1(A); u_1(B);$$

$$T_2 = rl_2(C); wl_2(A); u_2(A); u_2(C)$$

$$T_3 = rl_3(A); rl_3(B); u_3(A); u_3(B)$$

Se pide realizar una Historia (ubicando los commits) que sea No Serial, Serializable, Recuperable y No ACA. Justificar.

b) Dada la siguiente tabla con una historia incompleta, complétela agregando operaciones de tal forma que, con un planificador basado en *timeStamp* **sin multiversión** se produzca un *rollback* pero con un planificador **con multiversión**, no se produzca (**Justificar**):

T1 =100	T2 =200	T3 =300	T4 = 400
r(A)			
	w(X)		
			r(Z)
			w(Y)
		r(X)	

c) Dado el siguiente archivo de Log cuyo manager utiliza una estrategia Undo/Redo con *check-pointing no-quietescente*:

1. < START T_2 >
2. < START T_1 >
3. < $T_1, A, 0, 100$ >
4. < START T_3 >
5. < $T_2, B, 0, 90$ >
6. < COMMIT T_1 >
7. < $T_3, C, 40, 44$ >
8. < START CKPT (T_2, T_3) >
9. < START T_4 >
10. < $T_4, D, 5, 10$ >
11. < $T_3, C, 44, 50$ >
12. < COMMIT T_3 >
13. < $T_2, B, 90, 200$ >
14. < END CKPT >
15. < COMMIT T_4 >
16. < $T_2, E, 500, 400$ >

Se pide

- i) Suponiendo una falla y que el último registro escrito es el 16 indicar (justificando) los pasos a seguir para la recuperación, indicando que debe hacerse con cada transacción, que modificaciones ocurren en el log si las hubiera, y los valores de los ítems modificados durante la recuperación
- ii) Idem el punto i) si el último registro escrito fuera el 13.

3 Optimización

Criterio de aprobación: se aprueba con 6.5 ptos, sin errores conceptuales, y el **plan de ejecución propuesto en a) debe ser muy cercano al plan de menor costo posible.**

El gobierno de la ciudad ha decidido transformar el Zoo en un ecoparque. Los animales enfermos o viejos, serán conservados en el ecoparque para su cuidado, y los demás liberados en reservas naturales en el país. Se cuenta con las siguientes relaciones:

RESERVA (idReserva, nombre, provincia, descripción)

ANIMAL (idAnimal, apodo, idEspecie, historia, salud, edad)

ESPECIE (idEspecie, nombre, descripción, extincion, peligrosidad)

HOST (idReserva, idEspecie, capacidad)

La tabla RESERVA guarda las reservas naturales del país. Tiene la información del nombre, provincia, y un texto con una descripción ampliada de la reserva. La tabla ANIMAL guarda el apodo del animal, su especie, su historia (padres, historia clínica, etc.), su grado de salud (que es un número float entre 0 y 10), y su edad, que es un valor entre joven (1), adulto (2), y viejo (3). La tabla ESPECIE guarda el nombre de la especie, una descripción detallada, si está en peligro de extinción (0=NO, 1=SI), y su grado de peligrosidad para el hombre (que es un número entero entre 1 y 10). La tabla HOST guarda la cantidad de animales que pueden ser albergados de cierta especie en una reserva.

Se desea optimizar la siguiente consulta:

```
SELECT A.idAnimal, A.apodo, H.idReserva
FROM HOST H, ANIMAL A, ESPECIE E
WHERE H.idEspecie = E.idEspecie AND E.idEspecie = A.idEspecie AND H.capacidad < 50
AND A.salud > 9 AND A.edad < 3 AND E.peligrosidad > 8
```

Se sabe adicionalmente que:

- Los atributos idX son numéricos enteros. Salud es numérico flotante. Cantidad de bytes que ocupa cada atributo: idX (4), nombre (60), provincia (8), descripción (500), apodo (60), historia (1000), salud (8), edad (4), extincion (4), peligrosidad (4), capacidad (4).
- $T_{ANIMAL} = 10.000$; $T_{RESERVA} = 200$; $T_{ESPECIE} = 100$; $T_{HOST} = 20.000$.
- La capacidad de la reserva es como máximo 100 por especie.
- Se dispone de los índices: **I1**: índice B+ clustered sobre edad en ANIMAL; **I2**: índice B+ unclustered sobre idEspecie en ANIMAL; **I3**: índice Hash sobre salud en ANIMAL; **I4**: Índice Hash sobre peligrosidad en ESPECIE. **I5**: Índice Hash sobre capacidad en HOST.
- Los índices B+ tienen una altura $X = 3$. En los índices Hash se asume un máximo de 5 bloques por bucket. Asuma que los punteros a tupla que se necesiten recorrer en los índices B+ entran en una hoja, y que una hoja entra en un bloque.
- Tamaño de un bloque $L_{bloque} = 2048$ bytes
- Cantidad de bloques en memoria: 5
- Si no hay especificación, suponga distribución uniforme.

Resuelva los siguientes items, **justificando todas sus respuestas**:

- Proponga un plan de ejecución optimizado para la consulta dada, indicando solamente el árbol final. Justifique cada una de las decisiones que tome para su propuesta de plan de ejecución. Justifique también porque decide utilizar o no utilizar cada uno de los índices disponibles.
- Calcule el costo de a) hasta el primer join inclusive.

Nota: En cada plan de ejecución utilizar materialización como input de una junta.