

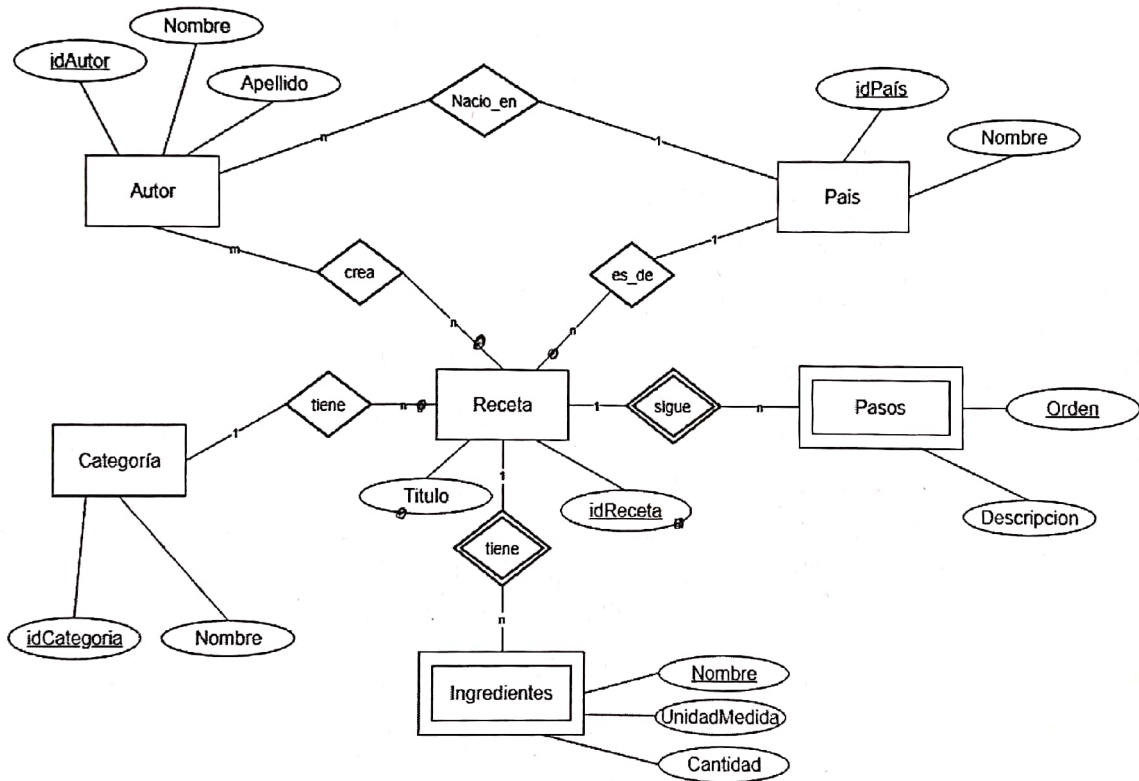
2do Parcial - 2do Cuatrimestre 2018 - Base de Datos
10 de Octubre de 2018

- Debe identificarse **cada** hoja con nombre, apellido, LU y su **número de orden**.
- Complete la primera hoja con la cantidad total de hojas entregadas y numere todas las hojas.
- Los pedidos de revisión se realizarán por escrito, antes de retirar el examen corregido del aula.
- Para que un ejercicio sume puntos **no deben cometerse errores conceptuales graves**.
- La **interpretación** del enunciado forma parte de la evaluación.
- El parcial es a libro **cerrado**. Justifique sus respuestas.

Criterio de Aprobación: Se aprueba con 7. Ejercicio 1 5ptos, Ejercicio 2 5ptos,

1 NOSQL

El DER de la figura es un modelo de datos para un sistema donde se almacenan recetas:



Se pide:

a) Document Database

- (i) Dibujar el diagrama de interrelación de documentos, justificando las decisiones tomadas. Es un requisito que cuando se accede a la base de datos debe poder obtenerse para una receta inmediatamente los nombres de los autores, la categoría, los pasos y los ingredientes.
- (ii) Especificar en JSON Schema el tipo de documento Receta

b) Realizar el diseño Column Family haciendo el diagrama de Chebotko para las siguientes consultas:

- (i) Las recetas de los autores de un país dado ordenadas por título.
- (ii) Las recetas ordenadas por categoría de la misma y dentro de ella por el título.

c) Suponga un sitio Web que permite acceder a los datos de las recetas utilizando un nombre de usuario y password. Diseñe una base de datos Clave-Valor que permita mantener el histórico de las recetas visitadas en una sesión.

2 Concurrency and Recoverability

a) Se tienen las siguientes transacciones:

$T_1 = rl_1(X); wl_1(Z); u_1(Z); u_1(X); c_1$

$T_2 = rl_2(Y); wl_2(Z); u_2(Z); u_2(Y); c_2$

$T_3 = rl_3(Z); rl_3(X); u_3(Z); u_3(X); c_3$

Se pide realizar una Historia que sea **Serializable** (pero no serial) y **No ACA**. Justificar.

b) Dada la siguiente tabla con una historia incompleta agregue al final una operación a alguna de las transacciones tal que dicha operación provoque un rollback si el planificador fuera basado en *timeStamp* **sin multiversión** pero no si fuera con **con multiversión**. Justificar:

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

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

1. < START T_1 >
2. < START T_2 >
3. < $T_1, A, 0, 100$ >
4. < $T_2, B, 0, 190$ >
5. < START T_3 >
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. < $T_2, B, 190, 200$ >
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, explicando 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 12.