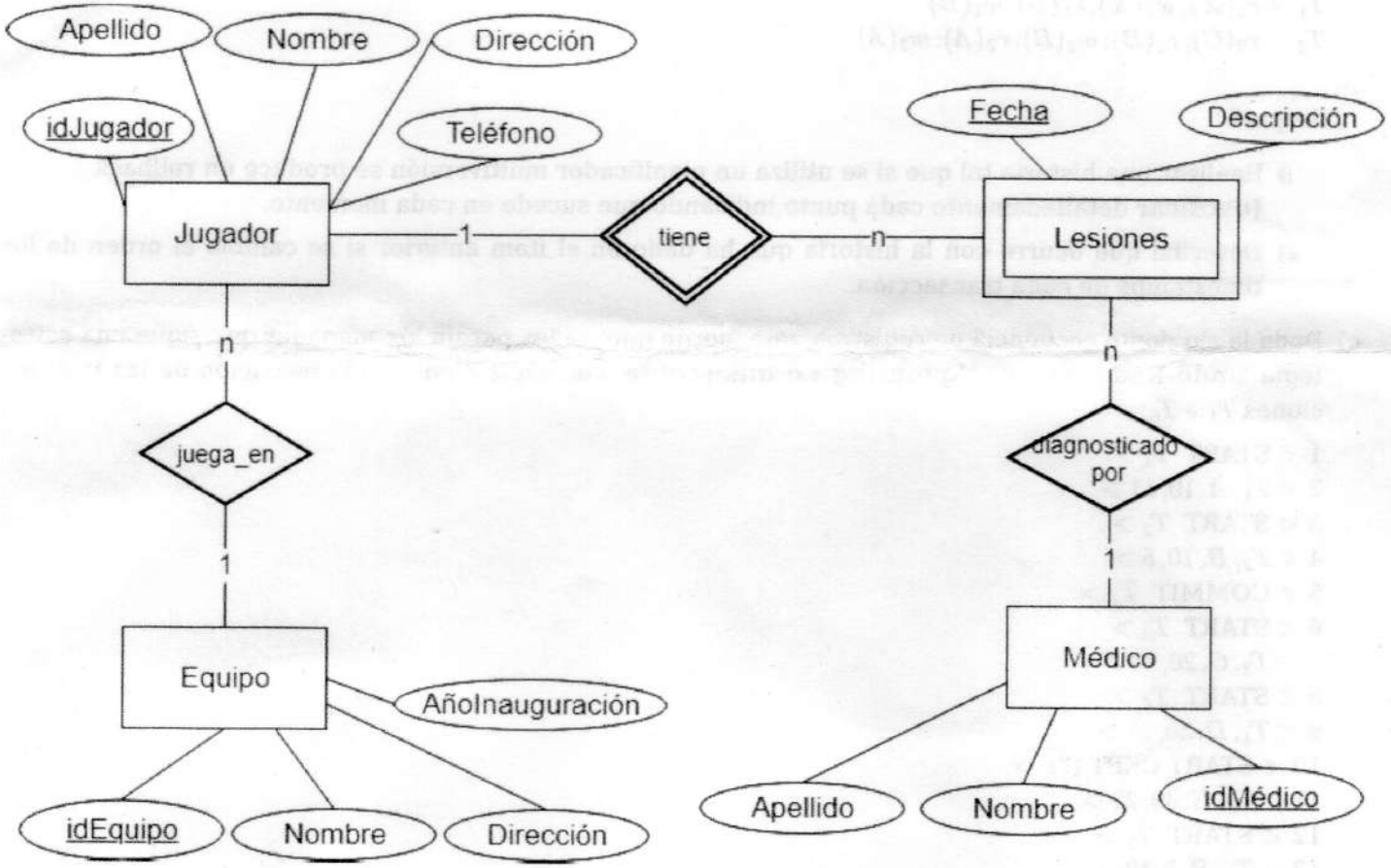
Recuperatorio del 2do Parcial 1 Cuatrimestre 2019
3 de julio de 2019

- 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*. Además debe sumar al menos 2.5 en cada ejercicio.

1. NOSQL

Dado el siguiente DER (que modela la información necesaria para guardar jugadores y equipos de una liga pequeña):



Se pide:

a) Dibujar el diagrama de interrelación de documentos, justificando las decisiones tomadas. Especificar en JSON Schema el tipo de documento Médico. La consulta más común, cuando se accede a un equipo, es tener la lista de jugadores (nombre y apellido de cada uno). No es tan importante acceder a las lesiones del jugador ni tener en el momento el nombre del médico, pero obviamente deber poder accederse a esta información.

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

   (i) Los nombres y apellidos de jugadores de los equipos que fueron inaugurados en un rango de años determinados.

   (ii) Todos los nombres, direcciones y años de inauguración de los equipos ordenados por nombre.

c) Suponga que la información del DER se guarda en una base de datos KV, realice un diseño adecuado. Sólo modele la parte del DER correspondiente a las entidades Jugador, Equipo y Lesiones. No hace falta la información del Médico.

2. Concurrencia y Recuperabilidad

a) Dada la siguiente historia:

$$H = r_1(D); r_3(E); r_2(A); w_3(B); w_1(E)\ r_2(E); r_3(C); w_2(C); r_4(B); r_1(A); r_4(E); r_3(D)$$

Se pide:

i) Hacer el $SG(H)$, indicar si es SR, y en caso afirmativo indicar todas las historias seriales equivalentes.

ii) Agregar los correspondientes commits de tal forma que H sea RC pero no ACA. ¿Es posible agregar los commits de tal forma que H también sea ACA? Justificar la respuesta.

iii) Suponga que cada operción de lectura se transforma en un lock de lectura y cada operación de escritura en un lock de escritura. ¿Es posible ubicar los unlocks de tal manera que cada transacción sea 2PL? ¿Cómo deben ubicarse los unlocks para que la historia sea legal?

b) Dadas las siguientes transacciones:

$T_1 = r_1(X); w_1(X); r_1(B); w_1(B)$
$T_2 = r_2(C); r_2(B); w_2(B); r_2(A); w_2(A)$

Se pide:

i) Realizar una historia tal que si se utiliza un planificador multiversión se produce un rollback. **Justificar** detalladamente cada punto,indicando que sucede en cada momento.

ii) Describa que ocurre con la historia que ha dado en el ítem anterior si se cambia el orden de los timestamps de cada transacción.

c) Dada la siguiente secuencia de registros, que fueron generados por un log manager que sigue una estrategia **Undo-Redo** con **checkpointing no-quiescente**, correspondientes a la ejecución de las transacciones T_1 a T_6:

1 < START T_1 >
2 < $T_1, A, 10, 11$ >
3 < START T_2 >
4 < $T_2, B, 10, 5$ >
5 < COMMIT T_1 >
6 < START T_3 >
7 < $T_3, C, 20, 45$ >
8 < START T_4 >
9 < $T_4, D, 30, 33$ >
10 < START CKPT (?) >
11 < $T_3, E, 40, 22$ >
12 < START T_5 >
13 < $T_2, B, 5, 12$ >
14 < COMMIT T_2 >
15 < $T_5, G, 20, 3$ >
16 < COMMIT T_4 >
17 < END CKPT >
18 < START T_6 >
19 < $T_5, G, 3, 4$ >
20 < $T_6, I, 50, 2$ >
21 < START CKPT (?) >
22 < $T_6, I, 2, 66$ >

i) Complete los registros de log con **"?"**, con los valores correspondientes. El registro < END CKPT > del primer checkpoint, ¿podría ocurrir antes del registro de < COMMIT T_4 > ? Justifique

ii) Describa las acciones que debería realizar el recovery manager para recuperarse, incluyendo los cambios tanto de los items en disco como en el log, si ocurriera un *crash* y el último registro de log en disco fuera: < $T_6, I, 2, 66$ >