

Examen

Instrucciones

El examen es individual, y el plazo (impostergable) para entregar el examen es el miércoles 5 de Julio, a las 23:59 hrs. La entrega se realiza por canvas.

Preguntas

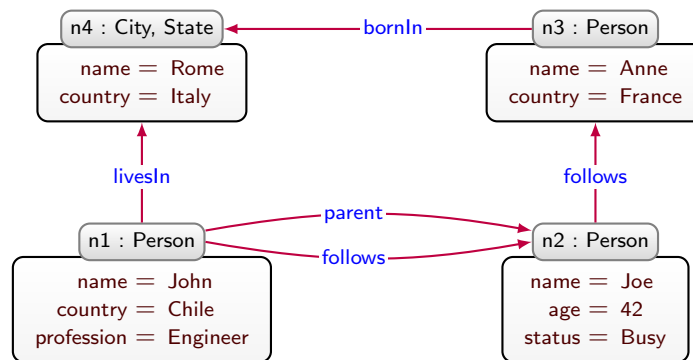


Figura 1: Una base de datos property graph.

En primeras dos preguntas trabajaremos con una representación relacional de la base de datos property graph que se muestra en Figura 1. Para esto ocuparemos el siguiente esquema relacional:

- `Nodes(nodeID)`
- `Edges(startNodeID,label,endNodeID)`
- `NodeLabel(nodeID,label)`
- `NodeProperties(nodeID,property,value)`

Las llaves vienen subrayadas en nuestro esquema. Para una ilustración, en Figura 1 podemos observar una base de datos de grafos de personas y ciudades. Aquí nuestros nodos tienen identificadores (e.g. `n1`), etiquetas (e.g. `Person`), y una serie de atributos (también llamados `properties`) con valores asociados, cómo, por ejemplo, `country = Chile` para el nodo `n1`. Por otro lado, las aristas conectan a dos nodos, y siempre tienen una etiqueta asociada. Cómo pueden observar, pueden existir varias aristas entre dos nodos, pero cada una debe tener una etiqueta distinta. Nuestro esquema adicionalmente permite varios `labels` para un nodo (observen cual es la llave para la tabla `NodeLabel`), cómo, por ejemplo, en el nodo `n4`. A continuación mostramos las tablas de nuestro esquema para la base de datos de Figura 1 (en `NodeProperties` faltan filas):

Node	nodeID	Edges	startnodeID	label	endNodeID
	n1		n1	parent	n2
	n2		n1	follows	n2
	n3		n1	livesIn	n4
	n4		n2	follows	n3
			n3	bornIn	n4

NodeLabel	nodeID	label	NodeProperties	nodeID	property	value
	n1	Person		n1	name	John
	n2	Person		n1	country	Chile
	n3	Person		n1	profession	Engineer
	n4	City		n2	name	Joe
	n4	State		n2	age	42
			

Si no entienden bien que es una base de datos de grafos no importa, porque para este examen solamente modelaremos este caso de uso como una base de datos relacional con el esquema que se indicó arriba. En lo que sigue cada pregunta vale 1 punto.

- Suponga que nos interesaría extraer desde nuestra base de datos la información sobre todas las personas que siguen a alguien viviendo en un lugar que es al mismo tiempo una ciudad y un estado. Quiere decir, nos interesan nodos que tienen una arista con la etiqueta **follows** hacia un nodo que tiene una arista etiquetada **livesIn** a un tercer nodo, y además este tercer nodo es etiquetado con **City** y **State**. Expresa esta consulta en **álgebra relacional**. Basta con devolver el identificador de los nodos solicitados.
- Usando el esquema de arriba, expresa la siguiente consulta en SQL:
 - Para cada nodo de tipo persona devuelve el número de personas que sigue este nodo. Quiere decir, para un nodo etiquetado como **Person** hay que contar el número de aristas salientes desde este nodo con la etiqueta **follows**. Adicionalmente, el nodo al cual lleva la arista debe ser etiquetado con **Person** (no validar esto significa una reducción en su puntaje). Si un nodo de tipo persona no sigue a nadie hay que devolver 0 asociado a este nodo.
- Suponga que $R(A \text{ int}, X \text{ char}(20))$ y $S(B \text{ int}, Y \text{ char}(10))$ son dos relaciones con llaves primarias A y B , respectivamente. Explica qué índices te puede ayudar en responder mejor la siguiente consulta:
 - $\text{SELECT } R.A, S.B \text{ FROM } R, S \text{ WHERE } R.A \geq 50 \text{ AND } S.B = 32$

Aquí debes también explicar cómo este índice (o estos índices) ayuda(n) en la ejecución de la consulta.
- Considere el schedule **S** de el Cuadro 1. Argumente lo siguiente:
 - (0.3 puntos) **S** no es serial.
 - (0.7 puntos) **S** no es conflict-serializable.
- Recuerde que en el protocolo Strict 2-phase locking (strict 2PL) visto en clases, una transacción acumula locks (compartidos o exclusivos) sobre recursos en la base de datos, y los suelta todos al hacer Commit. En Cuadro 1 se presenta un schedule que no permite a protocolo strict 2PL avanzar porque se encuentra en un deadlock. Para esta pregunta debes responder lo siguiente:
 - (0.2 puntos) ¿Por qué el schedule **S** de Cuadro 1 está en el estado de deadlock?

- b) (0.8 puntos) Presente un schedule para las transacciones T1, T2 y T3 de Cuadro 1 que **no es** serial, pero si está permitido por el protocolo strict 2PL y que no está en el estado de deadlock. Quiere decir, muestre un schedule para las transacciones T1, T2 y T3 de Cuadro 1 tal que hay una mezcla de operaciones entre distintas transacciones, pero no hay un deadlock. Este schedule no necesita ser equivalente al schedule **S**.

T1	T2	T3
R(a) R(b)	R(a) R(c) W(c)	R(d) W(d) Commit
R(c) W(c) Commit	W(a) R(e) W(e) Commit	

Cuadro 1: Schedule **S**.

6. Suponga que su sistema tuvo una falla. Al reiniciar el sistema, el sistema se encuentra con el *log file* que se muestra a continuación, en la tabla “Log Undo”. Suponiendo que la política de *recovery* es la de *Undo Logging*:
- (0.8 puntos) indique, paso a paso, cómo se verá el proceso de recovery.
 - (0.2 puntos) después de realizar el recovery, cual es el valor en la variable/recurso a?

Log Undo
<START T1>
<T1, a, 27>
<T1, a, 29>
<START T2>
<T1, a, 31>
<COMMIT T1>
<T2, a, 33>
<START CKPT (T2)>
<T2, b, 2>
<START T3>
<T3, c, 2>
<T2, b, 79>
<START T4>
<T4, f, 7>
<T3, e, 22>
<COMMIT T4>
<T3, e, 24>
<T2, b, 77>