Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Segundo Semestre de 2018

IIC 2413 – Bases de Datos
Examen

Pregunta 1: SQL y Recursión / Transacciones

Considere el siguiente esquema:

```
Usuarios(uid int PRIMARY KEY, nombre VARCHAR(100))

Transacciones(tid int PRIMARY KEY, schedule_id int,
    uid int FOREIGN KEY REFERENCES Usuarios(uid))

Conflictos(tid1 int, tid2 int, PRIMARY KEY(tid1, tid2))
```

En este esquema se presentan distintos usuarios de un sistema de bases de datos, junto a ciertas transacciones y conflictos entre ellas. En la relación `Transacciones` se almacena el `uid` del usuario que la creo y el número de *schedule* al que pertenece. En la relación `Conflictos` aparecerá la tupla (i, j) si en el grafo de precedencia del *schedule* al que pertenecen la transacción i y la j hay una arista desde i hacia j. Puede asumir que i y j siempre serán del mismo *schedule*. En esta pregunta se piden las siguientes consultas (para resolver esta pregunta puede usar vistas):

a) **[0.5 pts]** Entregue cada nombre de usuario junto a todas las transacciones que ha creado.

b) **[1.5 pts]** Entregue cada identificador de usuario junto al número de sus transacciones que forman parte de un conflicto (i.e. están mencionadas en alguna tupla de la relación `Conflictos`). Si una transacción con identificador i aparece más de una vez, **sólo se cuenta en una ocasión**.

c) **[3.5 pts]** Entregue cada identificador de *schedule* que no es *conflict serializable*. **Hint**: Use recursión.

d) **[0.5 pts]** Entregue el nombre de todos los usuarios que tienen transacciones que participan en un *schedule* no serializable.

Pregunta 2: Programación y SQL / Recuperación de fallas

En esta pregunta tendrás que implementar el algoritmo de *undo logging*. Considere que tiene una relación que se utilizó para guardar *logs* de una base de datos. Esta relación tiene la siguiente forma:

```
Logs(linea_log int PRIMARY KEY,
     nombre_transaccion VARCHAR(20),
     atributo_cambiado VARCHAR(20),
     valor_antiguo int,
     status VARCHAR(20))
```

La columna `status` puede tener tres valores:

- `W`: Indica que esa línea del *log* representa un cambio.
- `S`: Indica que esa línea del *log* representa una transacción que empieza.

- `E`: Indica que esa línea del *log* representa una transacción que finaliza.

En esta relación:

- Si tenemos la tupla `(23, 'T1', 'a', 2, 'W')` quiere decir que la línea 23 del *log* señala que la transacción `T1` cambió el atributo `a` y que su valor antiguo era el número 2.
- Si tenemos la tupla `(1, 'T1', null, null, 'S')` quiere decir que la transacción `T1` comenzó en aquella línea del log.
- Si tenemos la tupla `(30, 'T1', null, null, 'E')` quiere decir que la transacción `T1` terminó en aquella línea del log.
- En esta versión del *log* no hay *checkpoints*.
- No puede asumir que la relación está ordenada.

[6 pts] Hace un programa con Python + `psycopg2` (sin ninguna otra librería externa) que recorra el *log* como lo haría un algoritmo de *undo logging* sin checkpoints y que imprima en consola todas las operaciones de *undo* que se deberían hacer en el orden que las haría el algoritmo de *recovery*. Si una variable fue modificada varias veces en el proceso de *undo* se deben imprimir todos los cambios. Asuma que los datos caben en memoria.

MongoDB / Índices

MongoDB

Tu trabajo como desarrollador de la red social *Fotograma*[1] va mucho mejor de lo esperado y recientemente han agregado una *feature* al sistema, que es la posibilidad de comentar las fotos. El modelo de la red social es bastante sencillo:

- Existe una colección de usuarios con datos básicos que toda red social debe tener.
- Existe una colección de fotogramas. Estas son las imágenes que los usuarios suben a la red social. Entre otros atributos, se destaca la presencia de la URL de la imagen y la descripción, que corresponde a un texto.
- Existe una colección de comentarios, en donde cada documento contiene el id del usuario que emitió el comentario, el id del fotograma comentado y el texto del comentario
- Cada documento que corresponde a la colección de usuarios tiene una lista con los `id` de los fotogramas a los que ha marcado con "me gusta".

Un ejemplo de documentos que se pueden encontrar en la base de datos MongoDB son los siguientes:

```
// Usuarios
{ "uid": 23,
  "nombre": "Fernando Pieressa",
  "bio": {
    "estudia_en": "PUC",
    "edad": 23,
    "descr": "Link a mi último vídeo: yu.tuv/video1234"
  },
```

[1] Recordemos que el nombre es una mezcla de Fotolog con Instagram.

```
  "me_gusta": [3, 5, 6] }

// Fotogramas
{ "fid": 1,
  "posteada_por": 23,
  "fotograma": "fotograma.com/foto/12345.jpg",
  "descripción": "Foto de mi preparación para subir el Cerro El Plomo #Trekking #Outdoor" }

// Comentarios
{ "fid": 1,
  "uid": 24,
  "comentario": "Ten cuidado, va a llover en la precordillera" }
```

Utilizando PyMongo o JavaScript se pide un procedimiento para cada una de las consultas a continuación. Si necesitas un índice por texto debes indicar el comando para crearlo.

a) **[1 pt]** Dado el identificador de un fotograma (puedes suponer que este id es i), entrega el número de likes de ese fotograma junto a todos sus comentarios.

a) **[2.5 pts]** Entregue el nombre de cada usuario que haya dado like o haya comentado algún (o algunos) fotograma que contenga la frase "#futbolEnOtraDimensión" y no la frase "#Cobreloa".

Índices

a) **[1.5 pts]** Suponga que posee el siguiente esquema:

```
R(a int, b int PRIMARY KEY)

S(b int PRIMARY KEY, c int)
```

Y conoce el valor mínimo y máximo de la columna b en ambas relaciones. Explique con pseudocódigo cómo implementaría un algoritmo que compute el resultado de la consulta $R \bowtie_{R.b=S.b} S$ si ambas llaves primarias están indexadas con un Hash Index Clustered. Entregue cuál sería el costo I/O de dicho algoritmo.

b) **[1 pt]** Explique con palabras cómo funciona el índice invertido de MongoDB para hacer búsqueda por texto. Además indique brevemente por qué este índice es diferente de uno tradicional, como un B+Tree o un Hash Index.

Pandas / Cypher

Esta pregunta tiene nota máxima 8.

Pandas

Considere tres objetos de tipo `DataFrame` instanciados en Python, el primero llamado `usuarios` el segundo llamado `productos` y el tercero `compras`. Sus descripciones son las siguientes:

- El `DataFrame` usuarios contiene los atributos `uid`, `nombre` y `edad`.
- El `DataFrame` productos contiene los atributos `pid`, `nombre`, `descripción` y `precio`.

- El `DataFrame` compras contiene los atributos `pid`, `uid`, `cantidad` que indica que cierto producto fue comprado por cierto usuario en cierta cantidad.

Estos registros corresponden a una tienda en línea. Se pide que entregue las siguientes tres consultas utilizando operaciones de la librería Pandas:

a) **[0.5 pts]** Entregue el nombre de todos los usuarios mayores a 18 años y menores que 50.

b) **[1 pt]** Entregue cada nombre de usuario junto a todos los productos que ha comprado. Si un usuario no ha comprado ningún producto, debe aparecer igual.

c) **[2 pts]** Entregue cada nombre de usuario junto al total del dinero gastado en la tienda.

Hint: Puede ir creando *dataframes* auxiliares. Recuerde que si multiplica dos vectores el resultado es la multiplicación componente a componente.

Cypher

Considere un grafo que almacena información de la universidad. En concreto guarda alumnos, cursos y profesores. En este grafo se señala qué cursos tomó un alumno, de qué cursos es ayudante un alumno y qué curso dicta un determinado profesor según lo que muestra la figura.

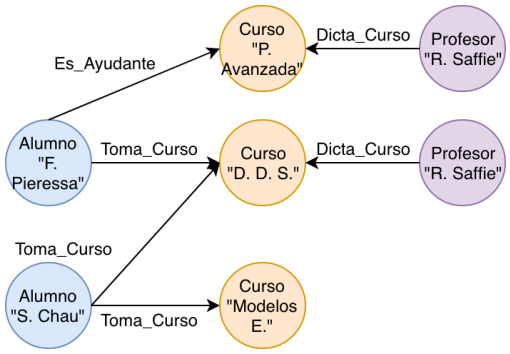

Figura 1: Grafo de ejemplo.

Considere que los atributos son:

- `Alumno`: posee un nombre y una edad.
- `Profesor`: posee un nombre y una edad.
- `Curso`: posee un nombre y una sigla.

y que las posible aristas son `Es_Ayudante`, `Toma_Curso` y `Dicta_Curso`.
Entregue las siguientes consultas en Cypher:

a) **[0.5 pts]** Todos los cursos que tomó el alumno con nombre "Fernando Pieressa".

b) **[0.5 pts]** El número de alumnos en el curso con la sigla "IIC2413".

c) **[1 pt]** Todos los ayudantes de todos los cursos dictados por el profesor con el nombre "Rodrigo Saffie".

d) **[1.5 pt]** Todos los pares de alumnos que son compañeros en algún curso y alguna ayudantía.