Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Primer Semestre de 2023

IIC 2440 – Procesamiento de Datos Masivos
MidTerm

Instrucciones. El examen tiene una duración de dos horas.

Resuelve cada uno de los problemas en una hoja separada. Las preguntas cortas generalmente tienen una mayor probabilidad de estar correctas.

Todas las preguntas tienen 6 puntos, la nota final se calcula como el promedio de los puntos de cada pregunta, más un punto base.

Las preguntas se hacen en voz alta; solo en los siguientes intervalos de tiempo:

- De 15:50 a 16:10
- De 17:00 a 17:20

Problema 1 Esta pregunta es sobre el Algoritmo A-priori que resolvemos con un sample. Mas especificamente, dado un conjunto de canastas con elementos, y un umbral de soporte s, se procede de acuerdo al siguiente esquema:

1. Primero se extrae un sample de canastas, digamos C.
2. Se corre A-priori sobre C
3. Los conjuntos con umbral de soporte mayor a s reportados en el sample C son reportados como conjuntos con umbral de soporte mayor a s en la muestra general.

Explica por que este algoritmo tiene falsos negativos: conjuntos reportados como frecuentes (con umbral de soporte mayor o igual a s), pero que realmente no lo son. Explica como puedes eliminar los falsos negativos sin incurrir en un costo mayor a el de pasar una vez por todas las canastas.

Problema 2 Una de tus amigas ha abierto una tienda de venta de artículos musicales en línea. El modelo de datos es el siguiente.

```
Usuarios(uid PRIMARY KEY, nombre, direccion, edad)

Productos(pid PRIMARY KEY, nombre, tipo, precio)

Compras(cid PRIMARY KEY, uid, fecha_compra, fecha entrega)

Compras_Productos(pid, cid, cantidad)
```

En este esquema tenemos usuarios, productos, compras y una relación intermedia que nos dice qué producto se adquirió en qué compra, y la cantidad del mismo; por ejemplo, podemos señalar que el producto 1 se compró 10 veces en la compra 3. Los tipos de datos y las llaves foraneas son las que esperarías.

Tu amiga tiene problemas con unas vistas que toman mucho tiempo, y sugiere indexar todas las columnas de todas las tablas en su base de datos.

Pregunta 2.1. Explícale a tu amiga por qué esto es mala idea.

Luego tu amiga te cuenta que las vistas de la página web que quiere optimizar son las siguientes:

1. La vista que le muestra a un usuario todas las compras que ha realizado.

2. La vista que le permite a tu amiga seleccionar un producto y ver todas las compras en las que se adquirió ese producto.

3. La vista que le permite a tu amiga contar el número de instrumentos comprado por tipo.

4. La vista que le muestra a un usuario todos los productos adquiridos en una compra, junto con el total de la compra.

Pregunta 2.2 Para cada una de las consultas explica cómo la indexarías para optimizar la consulta asociada indicando el tipo de índice. Además, explica cómo crees que el motor de consultas hace uso de esos índices, apoyandote en cómo crees que va a ser el plan de la consulta.

Problema 3 Después de aceptar tus consejos, el negocio de tu amiga empezó a crecer bastante. Por lo mismo, quiere hacer análisis de datos para poder construir un sistema recomendador que le permita aumentar aún más sus ventas. Ahora nesita hacer consultas más complejas como:

1. El total de dinero gastado por cada usuario, dividido por tipo de instrumento.

2. Para cada producto, un histograma que muestra la edad de los clientes que han adquirido el producto.

3. El total de ventas mensual por producto, que probablemente va a tener que reportar a la entidad fiscalizadora del negocio.

Tu amiga se dio cuenta que al hacer este tipo de consultas la base de datos funciona muy lento, y como le fue bien con la indexación propuesta anteriormente, te pide ayuda para optimizar estas consultas.

Supón que las ventas van bien, por lo que tu amiga tiene espacio para más desarrollo. Explícale a tu amiga lo que debería hacer. Apóyate en lo discutido en clases.

Problema 4 Una de las cosas que tu amiga probó fue Big Query, y como el sistema tenía buenas recomendaciones de otros amigos, se interesó en saber cómo funciona. En concreto, explícale los siguientes puntos:

1. ¿En qué se basa la capacidad de cómputo de Big Query?

2. Explica cómo lo hace Big Query para responder una consulta de join.

3. Explica cómo lo hace Big Query para responder una consulta de agregación.

4. Explica cómo lo hace Big Query para responder una consulta que utiliza Window Functions.

Problema 5 En el algoritmo A-priori, un costo importante se da entre etapas. Si al final de la etapa i se ha identificado un conjunto A de conjuntos de elementos frecuentes, cada uno con i elementos, para la etapa siguiente debemos generar todos los conjuntos candidatos de $i+1$ elementos que podrían ser frecuentes. Para simplificar el análisis, digamos que estos candidatos se generan sobre todos los conjuntos formados con la unión de dos conjuntos de A tales que su cardinalidad es exactamente $i+1$.

Supón que al correr este algoritmo, ves que no tienes espacio en memoria para iterar sobre todos los pares (a_1, a_2), para $a_i \in A$, de forma de verificar si el conjunto $a_1 \cup a_2$ tiene cardinalidad $i+1$. ¿Como podrías encontrar los candidatos sin hacer esa doble iteración? Estarías dispuesto a intentar alguna solución que tuviera alguna probabilidad de error (pero solo para la parte donde buscas y encuentras conjuntos candidatos).