**UNIVERSIDAD GALILEO**

**Programación VI (Base de Datos)**

**Examen Parcial 2**

**Primera serie (30 Puntos)**: Desarrolle cada una de las siguientes preguntas

1. ¿Cuáles son las propiedades ACID?

* A: Atomicidad
* C: Consistencia
* I: Isolation (aislamiento)
* D: Durabilidad

1. ¿Qué es un deadlock? De un ejemplo

Un deadlock es un error de calendarizaciones entre transacciones las cuales esperan bloqueos entre ellas por lo que ninguna logra finalizar.

Ejemplo:

|  |  |
| --- | --- |
| **T1** | **T2** |
| X(A) |  |
|  | X(B) |
| X(B) sleep |  |
|  | X(A) sleep |

1. Describa las políticas de robar frames y no forzar

Robar frames es cuando a pesar de que una transacción no haya hecho commit se permita escribir a disco. Esto se hace para poder trabajar con memoria limitada.

Forzar es que obligamos a escribir a disco cuando una transacción haya hecho commit únicamente.

1. Explique los pasos a seguir para la construcción de un índice

* Se debe considerar la carga de trabajo.
* Que tipo de índice?
* Clustered?
* Hash o Árbol?
* Dinámico o estático?
* Denso o disperso?

1. ¿Qué debe incluir la descripción de la carga de trabajo?
   * Las consultas más importantes y con que frecuencia se utilizan.
   * Las mas importantes actualizaciones y que tan frecuentemente se utilizan.
   * El tiempo de respuesta deseado para estas consultas y/o actualizaciones.
2. ¿En qué consiste la descomposición horizontal?

Consiste en dividir las tablas con el objetivo de disminuir las tuplas que existen en la tabla original con el objetivo de disminuir su tamaño. Para volver a unir esa data se puede usar un UNION.

1. ¿En que consiste la afinación de índices> ¿Por qué es necesaria?

Consiste en optimizar los índices según las consultas que son mas importantes o mas utilizadas. Esto es necesario dado que el objetivo de las vistas es optimizar consultas pero varios factores pueden hacer que los índices necesiten afinarse.

1. ¿Qué es denormalizacion?

Consiste dejar las relaciones en una forma normal menor, por ejemplo pasar de BCNF a 3FN. Esto se puede lograr con particionamiento.

1. Explique el control de acceso mandatorio

Esta basado en políticas del sistema que no pueden ser cambiadas por usuarios individuales, a cada objeto de la DB se le asigna un nivel de seguridad. Por lo que todo estará basado según esos niveles de seguridad.

1. Explique el control de acceso discrecional

Se basa en derechos de acceso o privilegios por objetos, se dan permisos a usuarios de select, insert, update, delete, etc.

**Segunda serie (70 Puntos)**: Resuelva los siguientes problemas

**Problema 1 (15 Puntos)**

Considera las siguientes clases de calendarizaciones:

* Serializable
* Serializable respecto a conflictos
* Recuperables
* Evitan aborts en cascada.

Para cada una de las calendarizaciones mostradas, diga a que clase pertenece y cual no, y si en algún caso no pude determinar si pertenece o no, explíquelo brevemente.

Las acciones están listadas de acuerdo con su orden en la calendarización, teniendo como prefijo el nombre de la transacción.

1. T1:R(X), T2:R(X), T1:W(X), T2:W(X)
2. T1:R(X), T2:R(Y), T3:W(X), T2:R(X), T1:R(Y)
3. T1:R(X), T2:W(X), T1:W(X), T2:ABORT, T1:COMMIT
4. T2:R(X), T3:W(X), T3:COMMIT, T1:W(Y), T1:COMMIT, T2:R(Y), T2;W(Z), T2:COMMIT
5. T1:R(X), T2:W(X), T1:W(X), T3:R(X), T1:COMMIT, T2:COMMIT, T3:COMMIT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Calendarización | Serializable | Serializable respecto a conflictos | Recuperables | Evitan aborts |
| A | No | No | No | No |
| B | Si | Si | No | No |
| C | Si | No | Si | No |
| D | Si | Si | Si | Si |
| E | No | Si | Si | Si |

**Problema 2 (15 Puntos)**

Suponga que un DBMS reconoce adicional a read y write las acciones increment la cual incremente el valor de un objeto integer en 1 y la acción decrement. Una transacción que incrementa un objeto no necesita saber el valor al igual que decrement por lo que son versiones de un blind write. Además de los bloqueos share y exclusive, dos bloqueos adicionales son soportados: Un objeto debe de estar bloqueado en modo I antes de incrementar y bloqueado en modo D antes de disminuir el valor. Un bloqueo I es compatible con otro bloqueo tipo I o D en el mismo objeto pero no con S y X.

Muestre como el uso de bloqueos I y D pueden incrementar concurrencia (Muestre una calendarización permitida por 2PL estricto que únicamente utilicé bloqueos S y X. Explique como el uso de bloqueos I y D permite que mas acciones sean intercaladas mientras se sigue respetando el protocolo de bloqueo.

**Problema 3 (15 Puntos)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **T** | **T1** | **T2** | **T3** | **T4** |
| 1 |  |  |  | R(tax) |
| 2 | R(salary) |  |  |  |
| 3 |  |  |  | W(tax) |
| 4 |  | R(tax) |  |  |
| 5 |  | W(tax) |  |  |
| 6 | R(tax) |  |  |  |
| 7 | W(salary) |  |  |  |
| 8 |  |  | R(salary) |  |
| 9 | W(tax) |  |  |  |
| 10 |  |  | W(salary) |  |
| 11 |  |  |  | R(salary) |
| 12 |  |  |  | W(salary) |

1. Dibuje la grafica de dependencias para esta calendarización
2. A que serialización equivale esta calendarización. Si no existe por favor indíquelo.
3. Asuma que la transacción T4 no se ejecuto para nada. ¿Cuál es la grafica de dependencia para este caso?
4. A que serialización equivale esta segunda calendarización. Si no existe por favor indíquelo.

**Problema 4 (15 Puntos)**

Considere una base de datos con la siguiente jerarquía de objetos: Base datos (D) la cual contiene dos archivos (F1 y F2) cada uno de estos contiene 1000 páginas (P1…P1000 y P1001…P2000, respectivamente). Cada página contiene 100 registros identificados por p:i donde i es el slot al que pertenece el registro y p es la página. Un bloqueo de granualidad múltiple es utilizado con bloqueos S, X, IS, IX e SIX para cada una de las siguientes operaciones indique la secuencia de bloqueo que se debe realizar

1. Leer de la página P2 a P1999

S(D) // bloqueo toda la db dado que se desea leer la mayoría de paginas.

1. Leer todas las paginas en F1 y modificar P1002

IS(D)

S(F1)

IX(D)

IX(F2)

X(P1002)

1. Leer todas las paginas en F1 y modificar 10 páginas que únicamente pueden ser inidentificadas luego de leer F1

IX(D)

IX(F1)

X(P1)

X(P2)

X(P3)

X(P4)

X(P5)

X(P6)

X(P7)

X(P8)

X(P9)

X(P10)

1. Borre todos los registros

X(D)

**Problema 5 (10 Puntos)**

Considere el siguiente árbol



20

FDGHFGHFGHFGHFGH

10

35

6

12

23

38 44

3\* 4\*

6\* 9\*

10\* 11\* 4\*

12\* 13\* 4\*

20\* 22\* 4\*

23\* 31\* 4\*

35\* 36\* 4\*

38\* 41\* 4\*

44\*

Describa los pasos necesarios para ejecutar cada una de las siguientes operaciones. Los incisos no son independientes.

- Buscar la entradas de datos 41\*

S(A)

S(C)

U(A)

S(G)

U(C)

S(O) // aquí se lee el 41

U(G)

U(O)

- Buscar todas las entradas de datos k\* con k<=20

S(A)

S(B),S(C)

U(A)

S(D),S(E)

U(B)

S(H),S(I)

U(D)

S(J),S(K)

U(E)

S(F)

U(C)

S(L) // aquí ya se leyeron todos los datos que solicitaron

U(H)

U(I)

U(J)

U(K)

U(L)

- Insertar la entrada de datos 23\*

X(A)

X(C)

U(A)

X(F)

U(C)

X(M) // M no tiene espacio por lo que se deberá dividir en 2 y no podemos liberar F dado que se modificara.

- Insertar la entradas de datos 7\*

X(A)

X(B)

U(A)

X(D)

U(B)

X(I) // I no tiene espacio por lo que se deberá dividir en 2 y no podemos liberar B dado que se modificara.

- Insertar las entradas de datos 23\* y 75\*

Insertamos 23:

X(A)

X(C)

U(A)

X(F)

U(C)

X(M) // M no tiene espacio por lo que se deberá dividir en 2 y no podemos liberar F dado que se modificara.

U(F) // después de insertar liberamos los objetos

U(M)

Insertamos 75

X(A)

X(C)

U(A)

X(G)

U(C)

X(P) // se inserta exitosamente el 75

U(G)

U(P)