Sistemas Informáticos I

Control Intermedio 3

20 diciembre 2018

1	2	3	4	5	6	7	Total

Apellidos:

Nombre:

Las respuestas se entregarán en las hojas de enunciados. Ten en cuenta que el espacio reservado para cada pregunta es suficiente para proporcionar la solución esperada.

Las preguntas tipo test son preguntas de selección múltiple en las que **al menos una** respuesta/afirmación es correcta. Si se selecciona alguna de las falsas, la respuesta se dará por incorrecta. Si falta alguna de las verdaderas, la calificación dependerá de la importancia de la respuesta faltante.

---

1. (0.75 puntos) Selecciona las opciones correctas respecto al plan de acceso/ejecución de una consulta SQL:

1. Se calcula *a posteri* una vez la consulta finaliza satisfactoriamente.
2. Se calcula *a priori* antes de ejecutar la consulta.
3. Puede cambiar entre distintas ejecuciones de la misma consulta dependiendo de la volumetría de las tablas implicadas.
4. Gestores de bases de datos avanzados como Oracle, MySQL o PostgreSQL permiten definir reglas de configuración para adaptar el plan de acceso de una misma consulta a la optimización de distintos parámetros.
5. Viene determinado por el orden de las tablas en los cruces y de las condiciones de las consultas y subconsultas.
6. Depende del optimizador/planificador de consultas, pero en general existen mecanismos para que los programadores SQL fuercen la ejecución de un determinado plan de acceso.

2. (0.75 puntos) Selecciona las opciones correctas:

1. Utilizar códigos de redundancia cíclica de los mensajes enviados entre los distintos componentes de un sistema distribuido evita que éstos puedan ser manipulados por un atacante
2. Utilizar protocolos de cifrado de clave pública/privada como RSA garantiza la confidencialidad de los mensajes enviados en un sistema distribuido permitiendo que solo el emisor y el receptor conozcan su contenido.
3. Utilizar protocolos de cifrado de clave pública/privada como RSA permite garantizar la propiedad de integridad de los mensajes que se intercambian en un sistema distribuido.
4. En el contexto de una aplicación web, se recomienda utilizar POST en lugar de GET porque, aunque no es del todo seguro, evita ataques de tipo *man in the middle*.
5. Cuando hablamos de seguridad informática nos referimos al cifrado de datos, tanto en la transmisión como en el almacenamiento, que evita que un atacante pueda ver información para la que no está autorizado.
6. Las cookies presentan serios problemas de seguridad que se solucionan utilizando SSL para su transmisión entre servidor y cliente, y viceversa.

3. (1 punto) En una aplicación web de venta online con una arquitectura en tres capas desarrollada con *Flask*, se ha monitorizado que el tiempo de respuesta promedio del servicio que devuelve el catálogo de productos (con una volumetría aproximada de 10000 productos) son 46 segundos, siendo el número de usuarios concurrentes conectados a la aplicación 4728. Monitorizando la base de datos en tiempo real, se obtiene que el tiempo medio de ejecución de la consulta que obtiene los datos del catálogo (SELECT * FROM Producto) es 0.98 segundos. Asumiendo el correcto funcionamiento de las herramientas de monitorización, ¿a qué podría deberse la diferencia en los tiempos registrados? Si te encargaran optimizar el servicio, ¿qué medidas tomarías para ello?

4. (2.5 puntos) Dada la siguiente definición de tabla:

```
CREATE TABLE MiTabla(
    campo1   INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    campo2   VARCHAR(256) NOT NULL DEFAULT 'vacio',
    campo3   DATE
);
```

donde AUTO_INCREMENT significa que el valor del campo se va a tomar automáticamente de una secuencia, indicar razonadamente si el rendimiento general de las siguientes consultas en una situación de carga mejorará, empeorará o será aproximadamente el mismo tras definir el índice que acompaña a la consulta:

`SELECT * FROM MiTabla`	`CREATE INDEX USING BTREE ON MiTabla(campo1 ASC)`

`SELECT * FROM MiTabla WHERE campo1 < 7`	`CREATE INDEX USING BTREE ON MiTabla(campo1 DESC)`

`SELECT * FROM MiTabla WHERE DAY(campo3)=1`	`CREATE INDEX ON MiTabla(campo3)`

`SELECT * FROM MiTabla`
`  WHERE campo3 BETWEEN`
`  '2018-01-01' AND '2018-12-31'`
`  AND campo2 LIKE 'z%'`
	`CREATE INDEX ON MiTabla(campo2 DESC, campo3 ASC)`

`SELECT * FROM MiTabla` `  WHERE UPPER(campo2) LIKE 'A%'`	`CREATE INDEX ON MiTabla(campo2 DESC)`

5. (1,5 puntos): Dada la siguiente secuencia de operaciones pertenecientes a una transacción

Read x
Read y
Write y
Write x
Write y
Read y

a) (0,5 puntos): Describa la secuencia de locks y unlocks (y de qué tipo) que será necesaria si se requiere un nivel de aislamiento grado 1 y maximizar la posibilidad de acceso concurrente a los objetos.

b) (0,5 puntos): Describa la secuencia de locks y unlocks (y de qué tipo) que será necesaria si se requiere un nivel de aislamiento grado 2 y maximizar la posibilidad de acceso concurrente a los objetos.

c) (0,5 puntos): Describa la secuencia de locks y unlocks (y de qué tipo) que será necesaria si se requiere un nivel de aislamiento grado 3 y maximizar la posibilidad de acceso concurrente a los objetos.

6. (1,5 puntos): Data la siguiente secuencia de ejecución de acciones pertenecientes a 3 transacciones distintas, explicar el grado máximo de aislamiento que puede tener cada una de ellas y por qué, asumiendo que todas tienen por lo menos aislamiento grado 0.

Transacción 1	Transacción 2	Transacción 3
	Begin	
		Begin
Begin		
Read y		
	Write x	
Read x		
		Write x
		Read z
	Read x	
	Write z	
	End	
End		
		End

7. (2 puntos): Suponga que el siguiente fragmento de pseudocódigo de una aplicación web. Este fragmento es ejecutado cuando el usuario logueado solicita cambiar su clave.

```
function cambia_clave(request):
    new_passwd = request.getParameter("new_passwd");
    user_name = session['name'];
    sql = "UPDATE USERS SET passwd='"+ escape(new_passwd) +
             "' WHERE uname='" + uname + "'";

    database.connection.execute(sql)
```

Suponga también que la función "escape()" escapa las comillas y demás caracteres especiales de la cadena recibida como argumento. Explicar cómo se podría producir un ataque por Inyección de SQL