

FINAL DE GESTION DE DATOS (2/3/22)

Apellido y Nombre	Legajo	1a	1b	2a	2b	3a	3b

A efectos de aprobar este examen final deberá sumar como mínimo 6 (seis) puntos y al menos tener un ejercicio práctico correcto.

1.a y b) Respuesta correcta 1 punto, respuesta incorrecta resta 1 punto.

2.a) 2 puntos 2.b) 2 puntos. 3.a) 2 puntos 3.b) 2 puntos

Contestar por Verdadero o Falso

1.a) En un stored procedure, dentro de una transacción no se puede cambiar el nivel de aislamiento.

1.b) El método Heapsort, siempre es más rápido que el método de la Burbuja para ordenar un conjunto de datos desordenados.

Responder exactamente lo solicitado en la pregunta en no más de 20 renglones.

2.a) Indique en que casos considera que el método de Hashing sería mejor que el método de Arbol B para la creación de índices.

2.b) Indique dos ejemplos donde sería necesario el uso de un Grafo irrestricto para modelar una solución.

Parte Práctica

3.a) Se sabe que en el motor SQL SERVER de este ejercicio los bloqueos son **compartidos** independiente del nivel de aislamiento para las **lecturas**.

Se tiene una tabla STOCK (id int (PK) , id_producto int (FK), cantidad Int) , que posee 1 sola tupla de valores (1, 1, 100) respectivamente.

Suponiendo que corren 2 transacciones A y B, en los tiempos establecidos en el gráfico, (es decir Tiempo = 0 significa que se ejecuta en paralelo la instrucción correspondiente a A y B para dicho tiempo).

Con esta información, usted deberá marcar cual o cuales son las respuestas correctas.

Tabla de secuencia de comandos y tiempos:

Tiempo	TRANSACCION A	TRANSACCION B
0	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
1	BEGIN TRANSACTION	BEGIN TRANSACTION
2	SELECT * FROM STOCK WHERE id_producto = 1	SELECT * FROM STOCK WHERE id_producto = 1
3	UPDATE STOCK SET set cantidad = cantidad - 1 WHERE id_producto = 1	UPDATE STOCK SET set cantidad = cantidad - 1 WHERE id_producto = 1
4	COMMIT	ROLLBACK

Decir cuál o cuáles de las opciones es la correcta.

- a) La cantidad de stock para el producto 1 es 99, porque se hace ROLLBACK de la Transacción B.
- b) La cantidad de stock puede ser, al finalizar la ejecución 100 o 99, según qué decisión tome el motor de la base de datos.
- c) Debido a que se genera DEADLOCK, siempre el stock va a ser 100.
- d) Ninguna de las anteriores.

3.b) Se desea realizar una actualización de STOCK cada vez que se realiza una venta y no permitir que esa venta ni ninguna otra deje el stock en negativo. La venta está representada por un insert en la tabla ITEM_FACTURA, cuyos campos son: id PK, id_producto FK, cantidad (todos enteros). Suponiendo que la tabla STOCK posee la estructura (id_producto int (PK), cantidad Int) y representa la cantidad de unidades disponibles de un producto.

Sabemos que el nivel de aislamiento es por defecto READ COMMITED, además que el sistema recibe en paralelo muchos insert en la tabla item_factura, con mucha concurrencia sobre el mismo producto.

```
Create trigger tr_final ON ITEM_FACTURA FOR INSERT
AS
BEGIN TRANSACTION
DECLARE @prod int
DECLARE @cantidad int

DECLARE mi_cursor CURSOR FOR
    SELECT id_producto, cantidad FROM INSERTED

OPEN mi_cursor
FETCH mi_cursor INTO @prod, @cantidad
WHILE @@fetch_status = 0
BEGIN
    IF (select cantidad FROM stock where id_producto = @prod) >= @cantidad
    BEGIN
        UPDATE stock set cantidad = cantidad - @cantidad
        WHERE
            id_producto = @prod
    END
    ELSE
    BEGIN
        ROLLBACK
        RETURN
    END
    FETCH mi_cursor INTO @prod, @cantidad
END
CLOSE mi_cursor
DEALLOCATE mi_cursor
COMMIT
```

Luego de la salida en producción se detectó que la tabla STOCK quedó en negativo para algunos productos.

Explique por qué pudo ocurrir esta situación y determina qué acciones/cambios de código aplicaría para que esta situación no suceda.