1. **¿Cuál es el concepto de Base de Datos Consistente?**

Una base de datos es Consistente cuando sus datos están protegidos y resguardados en caso de algún inconveniente.

1. **Al ejecutar una o más instrucciones SQL ¿Cuáles son las posibles razones para que las instrucciones SQL NO terminen de ejecutarse?**
   * Falla del programa
   * Falla del software de la BD
   * Falla del Sistema Operativo
   * Falla del hardware
   * Falla de energía eléctrica
   * Situaciones excepcionales detectadas que hace que el programa no pueda continuar
2. **Al ejecutar dos o más instrucciones SQL. Defina el concepto de Ejecución Serializable.**

La ejecución serializable se define como aislar las instrucciones que se están ejecutando.

Es el nivel más alto de aislamiento.

1. **Al ejecutar dos o más instrucciones SQL. Defina el concepto de Atomicidad.**

Atomicidad es cuando todas las instrucciones se ejecutan es decir no se puede ejecutar un conjunto de instrucciones, Se ejecuta todo o nada.

1. **¿Qué es una Transacción?**

Es un programa que lee o modifica el contenido de la base de datos, también se dice que es un conjunto de instrucciones que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica.

1. **¿Las Transacciones resuelven el problema de Seriabilidad y Atomicidad?**

Si

1. **¿Cuándo usar Transacciones?**

Las transacciones se utilizan para proteger los datos de la base de datos ante cualquier falla y evitar la inconsistencia.

1. **¿Cuáles son los beneficios de usar Transacciones?**
   * Proteger los datos de las fallas del software, hardware y potencia eléctrica.
   * Permitir el aislamiento(serializable) de datos de tal forma que varios usuarios puedan acceder simultáneamente a los datos sin interferencia.
2. **¿Cuáles es el Rol de las Transacciones en una Base de Datos?**
   * Proteger los datos de las fallas del software, hardware y potencia eléctrica.
   * Permitir el aislamiento(serializable) de datos de tal forma que varios usuarios puedan acceder simultáneamente a los datos sin interferencia.
3. **¿Cómo se inicia una transacción?**

Se inicia con la instrucción

**Begin Transaction**

1. **¿Defina el concepto de COMMIT en las transacciones?**

Es el estado final de las transacciones es decir todas las instrucciones ya están comprometidas.

1. **¿Defina el concepto de ROLLBACK en las transacciones?**

La instrucción ROLLBACK permite abortar una transacción y las instrucciones que están parcialmente comprometidas son anuladas además toda la transacción es cancelada y la base de datos se revierte a un estado inicial.

1. **¿Describa las propiedades ACID de las transacciones?**

**A**tomicidad: Se deben ejecutar toda o ninguna instrucción de la transacción.

**C**onsistencia: Las instrucciones no deben violar, romper las reglas de la BD.

**I**solation(aislamiento): Una operación no puede afectar a otra, la transacción debe aislar sus instrucciones.

**D**urabilidad: Una vez ejecutada la operación persistirá para siempre.

1. **¿Describa los diferentes estados de las Transacciones?**
   * **Activa**: Es el estado inicial, permanece en ese estado durante la ejecución.
   * **Parcialmente Comprometida**: Después de ejecutarse la última instrucción.
   * **Fallida**: Se descubre que no puede continuar con la ejecución normal.
   * **Abortada**: Después de haber retrocedido la transacción y restablecido la BD a su estado anterior al comienzo de la transacción.
2. **¿Describa el concepto de BITACORA en una Base de Datos?**

Se utiliza para la recuperación del sistema después de una falla y para garantizar las propiedades **ACID** de las transacciones, además se guarda en los archivos Log de transacciones incluso llegan a ser 4 veces mas grande que la misma base de datos.

1. **¿Cuál es el rol de la BITACORA en una Base de Datos?**

El rol de la BITACORA es guardar todo lo que se hace, anotar cada detalle que en la base de datos.

1. **Escriba las instrucciones básicas que debe incluir una Transacción.**

**BEGIN**

**BEGIN TRAN**

**BEGIN TRY**

INSTRUCCIONES……

**COMMIT TRAN**

**END TRY**

**BEGIN CATCH**

**ROLLBACK TRAN**

**END CATCH**

**END**