República Bolivariana de Venezuela

Ministerio del Poder Popular para la Educación Superior

Universidad Nacional Experimental de la Gran Caracas "UNEXCA"

Trayecto III, semestre II

Unidad Curricular: MODELADO DE BASE DE DATOS

MBD Taller No. 3 Integridad y Seguridad en BD y SGBD

Docente: Vladimir Peña

Estudiante: Bruno Palacios

C.I: 30150650

Caracas Enero del 2026

I. Preguntas Teóricas

Integridad de Datos

1. **Propiedades ACID:** * **Atomicidad:** Garantiza que la transacción se ejecute como una unidad irreducible; o se realizan todas las operaciones o ninguna.
    - **Consistencia:** Asegura que una transacción transforme la base de datos de un estado válido a otro, respetando todas las reglas de integridad.
    - **Aislamiento:** Permite que las transacciones se ejecuten de forma independiente, de modo que los efectos intermedios de una transacción no sean visibles para otras.
    - **Durabilidad:** Certifica que, una vez confirmada la transacción (commit), los cambios persistirán en el almacenamiento físico incluso ante fallos del sistema.
2. **Integridad Referencial:** Es la validez de las relaciones entre tablas, asegurando que una clave foránea siempre apunte a una clave primaria existente. Para evitar la eliminación de una fila padre con hijos asociados, se recomienda configurar la cláusula **`ON DELETE RESTRICT`** o **`ON DELETE NO ACTION`**.
3. **Integridad de Entidad vs. Dominio:** La integridad de entidad garantiza que cada fila sea única e identificable mediante una clave primaria no nula. La integridad de dominio define el conjunto de valores válidos (tipo, rango, formato) para una columna específica. La restricción principal para la integridad de entidad es la **`PRIMARY KEY`**.
4. **Disparador vs. Procedimiento Almacenado:** La diferencia reside en la forma de invocación: un disparador se ejecuta automáticamente ante un evento DML (INSERT, UPDATE, DELETE), mientras que un procedimiento debe ser llamado explícitamente por el usuario o aplicación. Se usaría un disparador para reglas de negocio que deban cumplirse obligatoriamente sin intervención humana, como el registro de auditoría.
5. **Control de Concurrencia:** Su objetivo es gestionar el acceso simultáneo de múltiples usuarios para evitar la pérdida de integridad de los datos. El

protocolo de bloqueo de dos fases (2PL) busca prevenir anomalías como la **Actualización Perdida** o la **Lectura Sucia**.

Seguridad de Bases de Datos

6. **Autenticación vs. Autorización:** La autenticación es el proceso de verificar la identidad del usuario (¿quién es?), generalmente mediante credenciales. La autorización determina los privilegios y permisos que dicho usuario tiene sobre los objetos de la base de datos (¿qué puede hacer?).
7. **Principio del Mínimo Privilegio:** Dicta que a un usuario se le deben otorgar únicamente los permisos estrictamente necesarios para desempeñar sus funciones. Es vital porque reduce la superficie de ataque y limita el daño potencial ante un compromiso de cuenta o error humano.
8. **RBAC vs. DAC:** El Control de Acceso Basado en Roles (RBAC) asigna permisos a funciones (roles) y luego usuarios a esos roles, facilitando la administración masiva. En el Control de Acceso Discrecional (DAC), el dueño de un objeto otorga permisos directamente a usuarios específicos, lo que se vuelve inmanejable en entornos grandes.
9. **Cifrado en Tránsito vs. Reposo:** El cifrado en tránsito protege los datos mientras viajan por la red (ej. **TLS/SSL**). El cifrado en reposo protege los datos almacenados en disco (ej. **AES-256** mediante Transparent Data Encryption o TDE).
10. **Prevención de Inyección SQL:** Además de las consultas parametrizadas, es esencial implementar la **validación y saneamiento estricto de entradas (input validation)** en la capa de aplicación para descartar caracteres sospechosos o formatos inesperados.
11. **Auditoría:** Su función es registrar y monitorear las acciones realizadas en la base de datos. Ayuda a detectar intrusiones al identificar patrones de acceso inusuales y abusos de privilegio al rastrear quién modificó datos sensibles o cambió configuraciones críticas.
12. **RPO y RTO:** El **RPO (Recovery Point Objective)** es el tiempo máximo de datos que la organización tolera perder tras un fallo. El **RTO (Recovery Time Objective)** es el tiempo máximo permitido para restaurar el servicio. El **RPO** se relaciona directamente con la cantidad de datos perdidos.

II. Estudios de Caso

Casos de Integridad

13. **Restricción de Monto Positivo:** Para asegurar que el `Monto_Transaccion` sea mayor a cero, se debe aplicar una restricción de comprobación a nivel de tabla. La instrucción técnica es: **`ALTER TABLE Movimientos ADD CONSTRAINT chk_monto_positivo CHECK (Monto_Transaccion > 0);`**. Esto garantiza que cualquier intento de insertar valores negativos o nulos sea rechazado por el motor.
14. **Actualización Perdida:** Ocurre cuando dos transacciones (T1 y T2) leen el mismo dato, T1 lo modifica y hace commit, y luego T2 sobreescribe el cambio de T1 con su propia modificación basada en el valor original. Se soluciona mediante el uso de **bloqueos pesimistas** o elevando el nivel de aislamiento a **SERIALIZABLE**.
15. **Sincronización de Stock:** El mecanismo ideal es un disparador, ya que debe reaccionar automáticamente ante un `INSERT` en la tabla de ubicaciones. Para esto, se ejecutaría: **`CREATE TRIGGER trg_actualizar_stock AFTER INSERT ON Ubicaciones_Almacen para cada fila UPDATE Productos SET Stock_Disponible = Stock_Disponible + NEW.Cantidad WHERE ID_Producto = NEW.ID_Producto;`**. Esto elimina el riesgo de desincronización manual.
16. **Propagación de Clave Primaria:** Si se modifica el `ID_Cliente` en la tabla maestra, la base de datos debe actualizar automáticamente las referencias en las tablas hijas. Para lograrlo, la clave foránea debe definirse con: **`ALTER TABLE Pedidos ADD CONSTRAINT fk_cliente FOREIGN KEY (ID_Cliente) REFERENCES Clientes(ID_Cliente) ON UPDATE CASCADE;`**.
17. **Transferencia Fallida:** La propiedad ACID que asegura la reversión de la resta inicial si la suma posterior falla es la **Atomicidad**. Mediante el uso de un bloque de transacción, el SGBD realiza un *rollback* automático, devolviendo el dinero a la cuenta origen.

18. **Ataque de Phishing:** Si el atacante descargó los datos, el mecanismo que los habría hecho inútiles es el **Cifrado de Datos en Reposo (TDE)** o el cifrado a nivel de columna. Sin las claves de descifrado almacenadas en el Hardware Security Module (HSM), los datos son solo ruido ilegible.

Casos de Seguridad

19. **Cuenta Compartida:** Se viola el principio de **No Repudio** y la **Responsabilidad Individual**. La auditoría se vuelve ineficaz porque todos los registros aparecerán bajo el usuario `app_dev`, impidiendo identificar qué persona física realizó una acción específica.
20. **Privilegios Excesivos:** Se ha violado el **Principio del Mínimo Privilegio**. Para corregirlo con RBAC, se debe crear un rol con permisos limitados y asignarlo al usuario: **`CREATE ROLE rol_ventas_basico; GRANT SELECT, INSERT, UPDATE ON Ventas TO rol_ventas_basico; GRANT rol_ventas_basico TO user_app; REVOKE DROP ANY TABLE FROM user_app;`**.
21. **Empleado Insatisfecho:** El mecanismo más adecuado es el **Monitoreo de Actividad de Base de Datos (DAM)** o la **Auditoría de Acceso a Objetos**. Estas herramientas alertan ante volúmenes atípicos de transferencia de datos (exfiltración) o accesos a tablas fuera del horario o perfil habitual.
22. **Privacidad de DBAs:** Para ocultar los números de tarjetas sin impedir las tareas de mantenimiento, se utilizan las **Vistas con Máscara de Datos** o **Dynamic Data Masking (DDM)**. La consulta para implementar esto sería: **`CREATE VIEW Vista_Mantenimiento_DBA AS SELECT Nombre, Direccion, 'XXXX-XXXX-XXXX-' + RIGHT(Numero_Tarjeta, 4) AS Numero_Tarjeta FROM Clientes;`**.
23. **Denegación de Servicio (DoS) Interna:** Para mitigar consultas masivas que agotan el CPU, se deben implementar **Resource Governors (Gobernadores de Recursos)** para limitar el porcentaje de CPU por sesión y el **Query Timeout** para abortar automáticamente procesos que excedan un tiempo razonable.

24. **Inundación en Centro de Datos:** El mecanismo de alta disponibilidad superior es la **Replicación Síncrona (Always On / Mirroring)**. Es superior a las copias tradicionales porque ofrece un RTO casi nulo, ya que el servidor de respaldo tiene una copia idéntica y activa de los datos lista para asumir el servicio de inmediato sin necesidad de procesos de restauración largos.