

- a. En no más de 15 renglones explique todo lo relacionado con objeto Vista.
- b. En una carilla explique la Funcionalidad de Seguridad de un RDBMS detallando someramente los objetos relacionados con la misma.

1. Base de Datos – Teoría (50 pts)

- a. Explique el Concepto de Dominio y tres formas diferentes de implementarlo en un motor. (30 pts)
- b. Describa brevemente los Niveles de Aislamiento.(20 pts)

- Describa 3 diferentes formas de implementar el concepto de Dominio definido por Edgar Codd, en un motor de Base de datos Relacional.
- Defina que es una transacción y explique su relación con la funcionalidad de consistencia en un motor de BD Relacional.

1. Base de Datos – Teoría (50 pts)

- a. Explique el objeto de BD Índice, cuando usarlo, ventajas y desventajas. (25 pts)
- b. Describa brevemente la Arquitectura ANSI/SPARK.(25 pts)

- a. Explique en menos de 15 renglones qué es Dominio y las diferentes formas de implementarlo en una BD.
- b. En una carilla explique Índices: Qué son, para qué sirven, tipos, ventajas, desventajas, y su relación con la funcionalidad de integridad.

Vistas:

A)

Las vistas es una utilidad de las bases de datos que nos permite visualizar determinados campos provenientes de una o mas tablas, esta misma no es una tabla en si, sino una visualización de campos de otras tablas. No se pueden insertar, eliminar ni modificar datos mediante vistas (excepcion a triggers). La utilidad de esta puede ser para evitar complejidad o hacer mas simple un código (evita repetir el mismo select, where, etc) cada vez que queramos consultar un dato especifico.

No se permiten crear indices en base a las views, para que se pueda crear una view el usuario debe tener permiso sobre los select que se estan haciendo.

Tambien cuenta con la funcionalidad de checkoption que en resumidas cuentas: realiza un chequeo de integridad de los datos a insertar o modificar

Seguridad de un RDBMS:

B) Se centra en la protección de datos y la administración de acceso. Los principales objetos relacionados con la seguridad incluyen:

1. **Usuarios y Roles:** Permiten definir cuentas y grupos de usuarios con permisos específicos.
2. **Permisos:** Controlan el acceso a datos y operaciones, asignados a usuarios o roles.

3. **Esquemas:** Agrupan objetos de base de datos (tablas, vistas, procedimientos) y ayudan a organizar y administrar permisos.
4. **Vistas:** Restringen el acceso a datos sensibles al mostrar solo la información autorizada.
5. **Triggers y Procedimientos Almacenados:** Pueden implementar reglas de seguridad y auditoría de cambios.
6. **Encriptación:** Protege datos sensibles tanto en tránsito como en reposo.

Estos elementos trabajan juntos para garantizar que solo los usuarios autorizados puedan acceder y modificar los datos, cumpliendo con las políticas de seguridad establecida

A) **Dominio:** Conceptualmente se refiere a todas las formas que puede presentar un campo/ atributo

Forma de implementación de dominio:

- Nombre de la columna
- Triggers
- Constrains

B) **Tipos y niveles de aislamiento:**

Read Uncommitted (Lectura No Confirmada):

- Permite leer datos que aún no han sido confirmados por otras transacciones.
- Problemas: lecturas sucias, no repetibles y fantasmas.

Read Committed (Lectura Confirmada):

- Solo permite leer datos que han sido confirmados.
- Problemas: lecturas no repetibles y fantasmas.

Repeatable Read (Lectura Repetible):

- Garantiza que los datos leídos durante una transacción no cambien hasta que la transacción finalice.
- Problemas: fantasmas.

Serializable:

- Aisla completamente una transacción de todas las demás, asegurando la ejecución secuencial.
- No hay problemas de concurrencia (lecturas sucias, no repetibles ni fantasmas).

B) **Transacción:**

Una transacción se refiere en SQL a las operaciones que se realizan en forma conjunta de manera atómica, estas transacciones sirven para garantizar la integridad de la información y su consistencia. Son seguras y funcionan como una única acción, es decir se ejecutan todas las acciones que incluye la transacción o no se ejecuta ninguna. En el caso de que falle 1 se vuelve todo para atrás.

INDICES

Los índices son estructuras opcionales asociadas a una tabla. La función de los índices es la de permitir un acceso más rápido a los datos de una tabla, se pueden crear distintos tipos de índices sobre uno o más campos. Los índices son lógicamente y físicamente independientes de los datos en la tabla asociada. Se puede crear o borrar un índice en cualquier momento sin afectar a las tablas base o a otros índices.

Btree Index

Estructura de índice estándar y más utilizada.

Btree Cluster Index

Este tipo de índice provoca al momento de su creación que físicamente los datos de la tabla sean ordenados por el mismo.

Bitmap Index

Son utilizados para pocas claves con muchas repeticiones. Cada bit en el Bitmap corresponde a una fila en particular.

Si el bit está en on significa que la fila con el correspondiente rowid tiene el valor de la clave.

Hash Index

Están implementados en tablas de hash y se basan en otros índices Btree existentes para una tabla. Si una tabla entra íntegramente en memoria, la manera más rápida de ejecutar consultas sobre ella es usando un hash index.

Cuando usarlos

- Indexar columnas que intervienen en Joins
- Indexar las columnas donde frecuentemente se realizan filtros
- Indexar columnas que son frecuentemente usadas en orders by
- Evitar duplicación de índices
- Sobre todo en columnas con pocos valores diferentes Ej: Sexo, Estado Civil,
- Verificar que el tamaño de índice debería ser pequeño comparado con la fila
- Tratar sobre todo en crear índices sobre columnas cuya longitud de atributo sea pequeña
- No crear índices sobre tablas con poca cantidad de filas, no olvidar que siempre se recupera de a páginas. De esta manera evitaríamos que el sistema lea el árbol de índices

Arquitectura ANSI/PARK

Nivel interno (Internal Level):

- También conocido como nivel físico.
- Describe cómo los datos están realmente almacenados en la base de datos.
- Incluye estructuras de almacenamiento físico y caminos de acceso.
- Se encarga de la optimización del almacenamiento y acceso a los datos.

- Los detalles de los índices, métodos de compresión y organización de los datos están en este nivel.

Nivel conceptual (Conceptual Level):

- También conocido como nivel lógico.
- Describe qué datos están almacenados en la base de datos y las relaciones entre ellos.
- No se preocupa de cómo los datos están físicamente almacenados.
- Es el nivel donde se define el esquema de la base de datos, incluyendo tablas, vistas y restricciones.
- Proporciona una visión de los datos que es independiente de la estructura de almacenamiento físico.

Nivel externo (External Level):

- También conocido como nivel de vista.
- Describe cómo los usuarios individuales ven los datos.
- Permite múltiples vistas de la base de datos, cada una adaptada a las necesidades específicas de diferentes usuarios o aplicaciones.
- Cada vista puede incluir parte de la base de datos o una combinación de datos de diferentes partes.
- No afecta ni se ve afectado por el nivel conceptual o interno.