

# **TEMA 63: LOS SISTEMAS DE GESTIÓN DE BASES DE DATOS SGBD. EL MODELO DE REFERENCIA DE ANSI**

Actualizado a 31/03/2019

## 1. BASES DE DATOS

**Base de datos (BD):** conjunto de datos almacenado en un soporte informático no volátil. Los datos deben estar estructurados e interrelacionados conforme a un modelo capaz de recoger el máximo contenido semántico.

**Diccionario de datos:** almacena información acerca de todos los objetos que conforman la BD: estructura física y lógica, definiciones de los objetos, restricciones de integridad, espacio utilizado, privilegios y roles...

## 2. SISTEMA DE GESTIÓN DE BASES DE DATOS (SGBD)

**Sistema de Gestión de Bases de Datos (SGBD):** conjunto de programas, procedimientos y lenguajes que permiten almacenar, actualizar y consultar datos contenidos en una BD, manteniendo su integridad, confidencialidad y seguridad.

### CARACTERÍSTICAS DE UN SGBD

- Modelo de datos soportado por el mismo.
- Uso de lenguajes de alto nivel (DDL, DML, DCL) y lenguajes 4GL.
- Acceso concurrente.
- Independencia física y lógica.
- Redundancia "controlada" de los datos.
- Integridad y consistencia de los datos.
- Seguridad de los datos.
- Alto rendimiento funcional.

### TIPOS DE LENGUAJE DE UN SGBD

- **Data Definition Language (DDL):** Permite definir la estructura de la base de datos del SGBD mediante la creación, manipulación o borrado de tablas y de todo lo relacionado con ellas como los atributos, índices o reglas de integridad.
- **Data Manipulation Language (DML):** Permite realizar consultas, inserciones, modificaciones o borrados de los datos en el SGBD.
- **Data Control Language (DCL):** Permite controlar el acceso a los datos del SGBD estableciendo, modificando y borrando los privilegios de los usuarios.

### ASPECTOS QUE DEBE GARANTIZAR UN SGBD

- **Concurrencia:** permitir accesos simultáneos a la BD sin conflictos y garantizando la consistencia de los datos.
- **Consistencia:** los valores de los datos no deben presentar contradicciones.
- **Integridad:** los valores de los datos son conformes a las reglas semánticas establecidas por el diseño mediante el uso del DDL.
- **Recuperación:** en caso de fallo se debe garantizar que la BD vuelve a un estado íntegro anterior

- **Privacidad:** los usuarios solo pueden acceder a los datos según los privilegios definidos.

#### CONCEPTOS DE TRANSACCIÓN Y CONCURRENCIA

**Transacción:** unidad elemental de trabajo delimitada por las sentencias Begin-transaction (comienzo de la transacción) y Commit/Abort (final exitoso o fallido de la transacción). Entre ellas se sitúan las sentencias Read y Write a ejecutar. **Propiedades ACID** que deben presentar las transacciones:

- **Atomicidad:** La transacción debe tener efecto en su totalidad, no se permite que se ejecute parcialmente.
- **Consistencia:** deben preservar la integridad y consistencia de la BD
- **aislamiento:** los cambios en la BD no serán visibles hasta la ejecución de un Commit
- **Duradero o persistente:** los cambios se hacen persistentes una vez ejecutado el Commit

**Concurrencia:** intercalación de operaciones de distintas transacciones con el objeto de mejorar el rendimiento, siempre que se asegure la integridad y consistencia de la BD.

- Problemas de interferencias entre transacciones:
  - **Lectura no repetible:** ocurre cuando una transacción T1 lee dos veces un valor y no coincide porque una segunda transacción T2 ha modificado dicho valor entre ambas lecturas, provocando que dos valores que deberían ser iguales no lo sean.
  - **Lectura sucia:** ocurre cuando una transacción T1 lee un dato modificado por otra transacción T2 antes de que haya realizado el Commit. Si T2 falla o realiza otra modificación del dato el valor leído por T1 nunca ha llegado a ser válido.
  - **Lectura fantasma:** ocurre cuando una transacción T1 realiza varias consultas iguales y una segunda transacción T2 introduce datos entre la realización de ambas, provocando que la segunda lectura tenga más datos que la primera.
- Niveles de aislamiento de las transacciones:
  - **Lectura no comprometida:** Los cambios realizados por las transacciones se encuentran disponibles inmediatamente.
  - **Lectura comprometida:** Los cambios realizados por las transacciones solo se encuentran a disposición del resto cuando se comprometen (se realiza un commit). Previene las lecturas sucias.
  - **Lectura repetible:** Las filas leídas o actualizadas por una transacción quedan bloqueadas hasta que dicha transacción termina. Previene la lectura sucia y la lectura no repetible
  - **Serializable:** Las transacciones ejecutadas de manera simultánea producen los mismos efectos que si se ejecutaran en serie. Previene las interferencias entre transacciones.

PERMITE LA VIOLACIÓN			
NIVEL DE AISLAMIENTO	L. SUCIA	L. NO REPETIBLE	L. FANTASMA

<b>SERIALIZABLE</b>	NO	NO	NO
<b>LECTURA REPETIBLE</b>	NO	NO	SI
<b>LECTURA COMPROMETIDA</b>	NO	SI	SI
<b>LECTURA NO COMPROMETIDA</b>	SI	SI	SI

- **Two-phase-locking:** mecanismo de control de la concurrencia usado por la mayoría de SGBD que obliga a que durante la ejecución de una transacción existan dos fases: en la primera se adquieren los recursos y en la segunda se liberan. Los locks adquiridos por una transacción solo serán liberados después de ejecutarse una operación Commit o Abort.

#### ARQUITECTURA INTERNA DE UN SGBD

- **Procesador de Consultas (QP):** analiza la sintaxis y semántica de las consultas, utilizando para ello los metadatos del diccionario de datos. Posteriormente las optimiza y transforma en un conjunto de instrucciones de bajo nivel (Begin-transaction, Read, Write, Commit, Abort).
- **Gestor de Transacciones (TM):** recibe múltiples transacciones concurrentes y ordena las operaciones Read, Write, Commit y Abort a ejecutar. Proporciona aislamiento entre transacciones.
- **Planificador (SC):** controla la concurrencia y planifica la ejecución de las transacciones, restringiendo el orden en el que el Gestor de Datos ejecutará las operaciones Read, Write, Commit, Abort de varias transacciones. Proporciona integridad y consistencia. Cuando el control de la concurrencia es de tipo 2PL (two-phase-locking), al Planificador se le denomina también como **Gestor de Bloqueos (LM, Lock Manager)**.
- **Gestor de Datos (DM):**
  - **Gestor de Recuperación (RM):** ejecuta las operaciones Commit y Abort. Asegura la atomicidad, persistencia y recuperación frente a fallos. Las transacciones deben ejecutarse completamente, en caso de fallo debe llevar la BD a un estado anterior consistente.
  - **Gestor de Buffers (cache):** ejecuta las operaciones Read y Write sobre la BD.

### 3. EL MODELO DE REFERENCIA ANSI/X3/SPARC

La arquitectura de un SGBD que sigue el estándar ANSI presenta 3 niveles diferenciados de representación de la información gestionada:

- **Nivel Interno o Físico:** se encarga de los aspectos más internos y relacionados con el servidor (generalmente sistema operativo y sistema de gestión de ficheros). Al diseñar el sistema interno se pretende conseguir un mejor tiempo de respuesta, minimizar el espacio de almacenamiento y evitar la redundancia de información.
- **Nivel Conceptual:** materializa la representación de los datos independientemente de su estructura física. El resultado del diseño de una BD específica la definición de un Esquema Conceptual conforme a un determinado modelo de datos, que se denomina **Esquema de BD**.
- **Nivel Externo o Lógico:** "filtra" los Esquemas de las BD conforme a la parte de los mismos que es de interés para un usuario concreto. Cada parte visible es una **Vista**.

Una arquitectura como la anterior aporta numerosas ventajas, siendo la más importante garantizar la independencia física y lógica de los datos, de manera que tanto si ocurren cambios a nivel físico como a nivel lógico, las BD puedan seguir funcionando como lo venían haciendo antes del cambio.

El diseño de una BD se define conforme a un **Modelo de Datos**, que permite definir la estructura y las restricciones de los datos de la misma. El estándar ANSI establece 3 familias de modelos de datos:

- **Modelo jerárquico:** presenta una estructura en árbol donde nodos y ramas siguen una relación del tipo 1:n.
- **Modelo Codasyl:** estructura en red donde se establecen relaciones n:m. Es más flexible que el jerárquico.
- **Modelo relacional:** presenta estructuras de la teoría matemática de conjuntos (álgebra relacional) y/o de la lógica de predicados (cálculo relacional). Permite el procesamiento de conjuntos de datos y no simples registros como en los anteriores. Se caracteriza por disponer los datos organizados en tablas (relaciones) que cumplen ciertas restricciones.