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

Actualizado a 1/02/2022

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 (estructura + semántica) 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... Es un elemento vivo, al que se van incorporando nuevos elementos (entradas), y a su vez estas entradas se componen de un nombre que la identifique, su tipología, un alias, los posibles valores que el dato puede tomar, quién y cuándo lo creó y quién y cuándo lo modificó por última vez, así como la información adicional que defina mejor el dato.

Si un diccionario (más restrictivo) está conectado a un compilador y el programador debe utilizar la definición que en él aparezca, se habla de diccionario activo. Si, por el contrario, se permite que el programador defina datos que no estén el diccionario (menos restrictivo), se habla de diccionario pasivo.

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 (gestión de bloqueos y consistencia de la información).
- Independencia física y lógica.
- Redundancia "controlada" de los datos.
- Integridad y consistencia de los datos.
- Seguridad y control de acceso a los datos según roles y permisos/privilegios.
- Alto rendimiento funcional.
- Alto volumen de datos.

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, vistas y esquemas, y de todo lo relacionado con sus 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.

  Aparte de los tres anteriores, a veces se considera un 4º tipo de lenguaje de los SGBD:

- **Transaction Control Language (TCL):** Permite controlar las transacciones para mantener la integridad de los datos. Una transacción es una unidad de trabajo que engloba una o varias sentencias SQL, por lo general, un grupo de sentencias DML.

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 garantiza que la BD vuelve a un estado íntegro anterior.
- **Privacidad**: los usuarios sólo pueden acceder a los datos según los privilegios definidos.

Según el teorema de CAP o la conjetura de Brewer, en un SGBD distribuido no se pueden garantizar simultáneamente más que 2 aspectos de entre los 3 siguientes:

1. Consistencia (C): Cualquier lectura recibe como respuesta la escritura más reciente o un error.
2. Disponibilidad (A): Cualquier petición recibe una respuesta no errónea, pero sin la garantía de que contenga la escritura más reciente.
3. Tolerancia al Particionado (P): el sistema sigue funcionando incluso si un número arbitrario de mensajes son descartados (o retrasados) entre nodos de la red.

CONCEPTOS DE TRANSACCIÓN Y CONCURRENCIA

Transacción: unidad elemental de trabajo delimitada por las sentencias Begin-transaction (comienzo de la transacción) y Commit/Rollback (final exitoso o fallido de la transacción). Entre ambas se sitúan las sentencias Read, Write a ejecutar como un bloque único.

Propiedades **ACID** que deben presentar las transacciones:

- **Atomicidad** (Atomicity): La transacción debe tener efecto en su totalidad, no se permite que se ejecute parcialmente. Si se realiza un COMMIT, todas las instrucciones se completan; si se realiza un ROLLBACK, ninguna de las instrucciones se completa.
- **Consistencia** (Consistency): Sólo se ejecutan aquellas operaciones que respetan las reglas de integridad de la base de datos. Se asegura así que los datos sean exactos y los esperados cuando un usuario los consulte.
- **Aislamiento** (Isolation): Una operación no afectará a las demás. Dos transacciones ejecutadas sobre la misma información son independientes y no generarán errores. Los cambios en la BD no serán visibles hasta la ejecución de un Commit.
- **Persistente** (Durability): Una vez ejecutado el Commit, los cambios se hacen persistentes y sobreviven a fallos del sistema.

Concurrencia: intercalación de operaciones de distintas transacciones con el objeto de mejorar el rendimiento, pero asegurando siempre la integridad y consistencia de la BD.

- Los problemas de interferencias entre transacciones son:
  - **Lectura no repetible**: ocurre cuando una transacción T1 lee dos veces un valor y no coinciden porque una segunda transacción T2 ha modificado dicho valor entre ambas lecturas.

- **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 inserta datos entre la realización de ambas consultas, provocando que la segunda lectura muestre más datos que la primera.

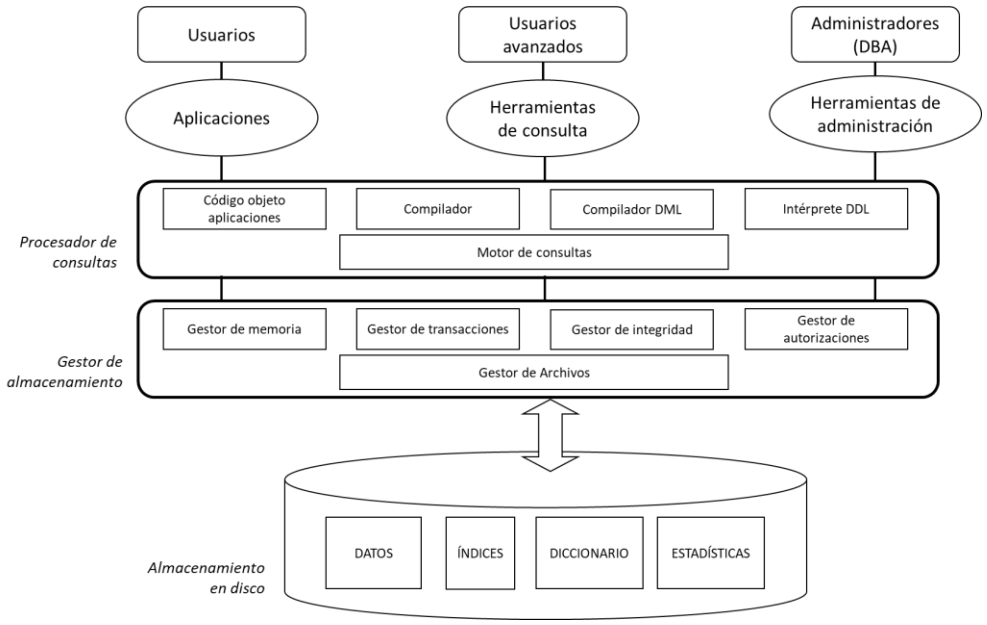
Según el nivel de error que se considere aceptable en nuestro sistema, se establecerá el nivel de aislamiento necesario para evitar alguno o todos los problemas anteriores. Los posibles niveles de aislamiento de las transacciones son:

- **Lectura no comprometida**: Los cambios realizados por las transacciones se encuentran disponibles inmediatamente.
- **Lectura comprometida**: Los cambios realizados por las transacciones sólo 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 todas las interferencias entre transacciones.

EVITA LA INTERFERENCIA			
NIVEL DE AISLAMIENTO	L. SUCIA	L. NO REPETIBLE	L. FANTASMA
LECTURA NO COMPROMETIDA	NO	NO	NO
LECTURA COMPROMETIDA	SI	NO	NO
LECTURA REPETIBLE	SI	SI	NO
SERIALIZABLE	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 recursos adquiridos por una transacción solo serán liberados después de ejecutarse una operación Commit o Rollback.

ARQUITECTURA INTERNA DE UN SGBD



Otros autores, destacan como más importantes una serie de elementos pertenecientes a la arquitectura 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, Rollback).
- **Gestor de Transacciones (TM)**: recibe múltiples transacciones concurrentes y ordena las operaciones Read, Write, Commit y Rollback 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, Rollback 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 Rollback. 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.

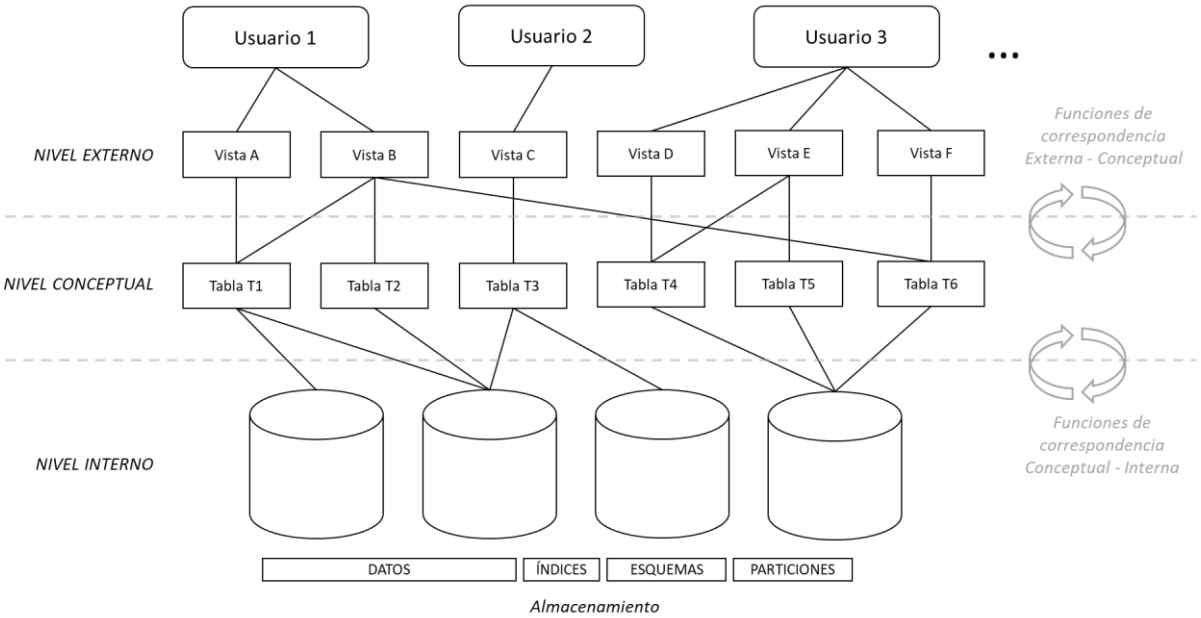
3. EL MODELO DE REFERENCIA ANSI/X3/SPARC

Según el Database Architecture Framework Task Group (DAFTG) del Grupo de Estudio de Sistemas de Base de Datos de ANSI/X3/SPARC, la arquitectura de un SGBD que sigue su estándar presenta 3 niveles diferenciados de representación de la información gestionada (definición de 1975):

- **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 especifica la definición de un Esquema Conceptual conforme a un determinado modelo de datos, que se denomina **Esquema de BD** (entidades, atributos, interrelaciones entre entidades, restricciones de integridad…)
- **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. Un esquema externo corresponde a una ***Vista***.

En el siguiente gráfico pueden apreciarse los tres niveles de representación de la información comentados, así como las **funciones de correspondencia** que garantizan la transferencia de información entre niveles y la independencia entre ellos.

- Correspondencia **externa-conceptual**: permite el intercambio de información entre un esquema externo y un esquema conceptual en ambos sentidos.
- Correspondencia **conceptual-interna**: permite el intercambio de información entre un esquema conceptual y un esquema interno, también en ambos sentidos.



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.

La correspondencia entre niveles de ANSI y el modelo relacional es la siguiente:

- NIVEL **LÓGICO** ←→ NIVEL **EXTERNO** + NIVEL **CONCEPTUAL**
- NIVEL **FÍSICO** ←→ NIVEL **INTERNO**

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.

Diez años después (1985) de la presentación del modelo de 3 niveles comentado, ANSI evolucionó al modelo de referencia DAFTG (RM) para los SGBD. Este nuevo modelo propone a 2 partes diferenciadas para actuar sobre los datos:

- La que se ocupa de la **DEFINICIÓN** de los datos. En esta parte, el Administrador (DBA) de la BD o el Administrador de las aplicaciones que usan la BD, definen el diccionario de datos (metadatos) y definen los procesos que harán uso de los datos.

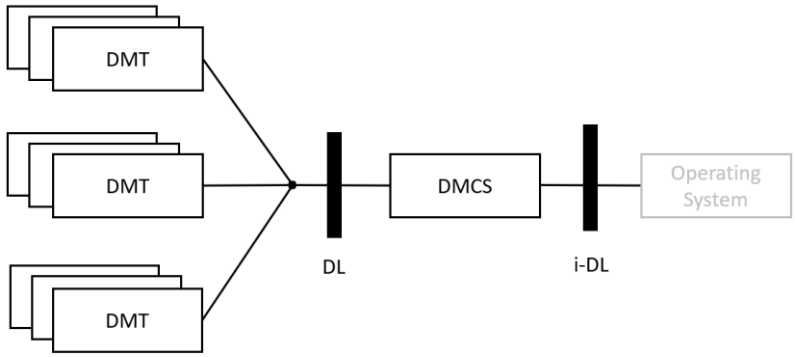
- La que se ocupa de la **MANIPULACION** de los datos. En esta parte se encuentran los propios datos, los procesos de transformación de los datos entre los niveles (externo, conceptual, interno) y los usuarios finales que hacen uso de los datos.

Se distinguen 3 conceptos en el DAFGT RM:

- Funciones
- Metadatos
- Interfaces

Por último, desde el punto de vista de **los componentes**, el RM propone los siguientes componentes y relaciones entre ellos:

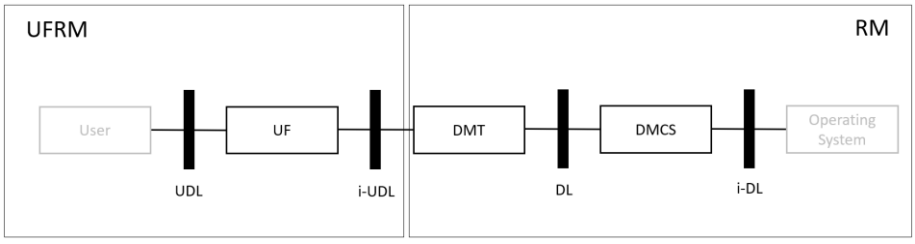
- Data Management Tools (**DMT**): Ejemplos de estas herramientas serían los lenguajes de tratamiento de datos, o las herramientas que utiliza el DBA para el tuning de la BD, o las utilizadas para realizar cargas y descargas de datos.
- Data Mapping Control System (**DMCS**): Corresponde al "core" del SGBD que proporciona operadores tanto para las operaciones de DEFINICIÓN como de MANIPULACIÓN de datos.
- Interfaces: Intercambian información entre el resto de componentes. Los hay a dos niveles:
    - Data Language Interface (**DL**). Lenguaje de manipulación para el modelo de datos del DMCS. Después, cada DMT puede tener sus interfaces más específicos que el DL de propósito general. Por ejemplo, el que permite a los usuarios o a las aplicaciones realizar peticiones para insertar o leer un dato.
    - Internal Data Language Interface (**i-DL**). Interfaz para el intercambio de datos entre el DMCS y el Sistema Operativo.



Tres años más tarde (1988), se produce una nueva actualización del modelo. En este caso, es una ampliación que añade los componentes de lo que corresponde al User Facility Reference Model (UFRM). Los 3 componentes que se añaden son:

- User Facility (**UF**): Hace de conexión entre el usuario y las herramientas DMT transformando las peticiones de los usuarios en peticiones entendibles por las DMT y en el sentido contrario, presentando los datos devueltos al usuario.
- Interfaces: Nuevamente en dos niveles:
    - User Data Language (**UDL**). Interfaz entre el entorno de usuario y la UF.
    - Internal User Data Language (**i-UDL**). Interfaz entre la UF y las DMT.

En el gráfico siguiente se aprecia la visión conjunta del Modelo de Referencia DAFTG RM y el UFRM.



4. TIPOS DE BASES DE DATOS