

# **FUNDAMENTOS Y ADMINISTRACIÓN DE BD**

**AUTOR: ORLANDO ESPINOZA BARBOZA**

**NOVIEMBRE: 2020**



**San Marcos**

## Introducción

Las bases de datos son útiles para el almacenamiento masivo y estructurado de datos. Todas las aplicaciones de información sean simples o multiusuario utilizan bases de datos para el funcionamiento de los mismos y para la integridad de los datos permitiendo a los usuarios el procesamiento de la información.

## Fundamentos y administración de BD

Conforme fue creciendo el uso de los sistemas de información evolucionando las tecnologías, se han usado desde archivos ficheros hasta los sistemas de almacenamiento de bases de datos. Estos datos son almacenados en estructuras complejas e integrales. Se tienen sistemas de gestión de bases de datos especializados para la administración y almacenamiento de estas estructuras.



## Tabla de contenido

Introducción.....	1
Fundamentos y administración de BD .....	1
Introducción a las bases de datos.....	3
Aplicaciones de los sistemas de bases de datos .....	3
Diferencia entre sistema de ficheros y bases de datos .....	3
Abstracción de datos .....	4
Modelo de datos relacional .....	4
Lenguaje estándar SQL.....	4
Transaccion .....	5
Arquitectura de un SGBD.....	5
Gestor de almacenamiento .....	5
Procesador de consultas .....	6
Seguridad .....	6
Flexibilidad e independencia .....	6
Problemas de la redundancia .....	7
Integridad de los datos .....	7
Concurrencia de usuarios .....	7
Conclusiones y recomendaciones.....	8
Referencias bibliográficas .....	8

## Introducción a las bases de datos

Una base de datos es una colección de datos interrelacionados, una colección de datos que contiene información relevante para una empresa.

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente.

Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos.

## Aplicaciones de los sistemas de bases de datos

Las bases de datos son ampliamente usadas. Las siguientes son algunas de sus aplicaciones más representativas:

- Banca. Para información de los clientes, cuentas y préstamos, y transacciones bancarias.
- Líneas aéreas. Para reservas e información de planificación. Las líneas aéreas fueron de los primeros en usar las bases de datos de forma distribuida geográficamente (los terminales situados en todo el mundo accedían al sistema de bases de datos centralizado a través de las líneas telefónicas y otras redes de datos).
- Universidades. Para información de los estudiantes, matrículas de las asignaturas y cursos.
- Transacciones de tarjetas de crédito. Para compras con tarjeta de crédito y generación mensual de extractos.
- Telecomunicaciones. Para guardar un registro de las llamadas realizadas, generación mensual de facturas, manteniendo el saldo de las tarjetas telefónicas de prepago y para almacenar información sobre las redes de comunicaciones.
- Finanzas. Para almacenar información sobre grandes empresas, ventas y compras de documentos formales financieros, como bolsa y bonos.
- Ventas. Para información de clientes, productos y compras.
- Producción. Para la gestión de la cadena de producción y para el seguimiento de la producción de elementos en las factorías, inventarios de elementos en almacenes y pedidos de elementos.
- Recursos humanos. Para información sobre los empleados, salarios, impuestos y beneficios, y para la generación de las nóminas.

## Diferencia entre sistema de ficheros y bases de datos

La información es almacenada en varios archivos y las bases de datos la almacena integrada. Las bases de datos evitan inconvenientes como Redundancia e inconsistencia de datos, Dificultad en el acceso a los datos, Aislamiento de datos, Problemas de integridad, Problemas de atomicidad, acceso concurrente y Problemas de seguridad.

- Redundancia e inconsistencia de datos. Debido a que los archivos y programas de aplicación son creados por diferentes programadores en un largo período de tiempo, los diversos archivos tienen probablemente diferentes formatos y los programas pueden estar escritos en diferentes lenguajes. Más aún, la misma información puede estar duplicada en diferentes lugares (archivos)
- Aislamiento de datos. Debido a que los datos están dispersos en varios archivos, y los archivos pueden estar en diferentes formatos, es difícil escribir nuevos programas de aplicación para recuperar los datos apropiados.
- Problemas de integridad. Los valores de los datos almacenados en la base de datos deben satisfacer ciertos tipos de

restricciones de consistencia. Por ejemplo, el saldo de una cuenta bancaria no puede nunca ser más bajo de una cantidad predeterminada (por ejemplo 25 €). Los desarrolladores hacen cumplir esas restricciones en el sistema añadiendo el código apropiado en los diversos programas de aplicación. Sin embargo, cuando se añaden nuevas restricciones, es difícil cambiar los programas para hacer que se cumplan. El problema es complicado cuando las restricciones implican diferentes elementos de datos de diferentes archivos.

- **Problemas de atomicidad.** Un sistema de un computador, como cualquier otro dispositivo mecánico o eléctrico, está sujeto a fallo. En muchas aplicaciones es crucial asegurar que, una vez que un fallo ha ocurrido y se ha detectado, los datos se restauran al estado de consistencia que existía antes del fallo.
- **Anomalías en el acceso concurrente.** Conforme se ha ido mejorando el conjunto de ejecución de los sistemas y ha sido posible una respuesta en tiempo más rápida, muchos sistemas han ido permitiendo a múltiples usuarios actualizar los datos simultáneamente.
- **Problemas de seguridad.** No todos los usuarios de un sistema de bases de datos deberían poder acceder a todos los datos.

## Abstracción de datos

Para que el sistema sea útil debe recuperar los datos eficientemente. Esta preocupación ha conducido al diseño de estructuras de datos complejas para la representación de los datos en la base de datos. Como muchos usuarios de sistemas de bases de datos no están familiarizados con computadores, los desarrolladores esconden la complejidad a los usuarios a través de varios niveles de abstracción para simplificar la interacción de los usuarios con el sistema:

- **Nivel físico:** El nivel más bajo de abstracción describe cómo se almacenan realmente los datos. En el nivel físico se describen en detalle las estructuras de datos complejas de bajo nivel.
- **Nivel lógico:** El siguiente nivel más alto de abstracción describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos. La base de datos completa se describe así en términos de un número pequeño de estructuras relativamente simples. Aunque la implementación de estructuras simples en el nivel lógico puede involucrar estructuras complejas del nivel físico, los usuarios del nivel lógico no necesitan preocuparse de esta complejidad. Los administradores de bases de datos, que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.
- **Nivel de vistas:** El nivel más alto de abstracción describe sólo parte de la base de datos completa. A pesar del uso de estructuras más simples en el nivel lógico, queda algo de complejidad, debido a la variedad de información almacenada en una gran base de datos. Muchos usuarios del sistema de base de datos no necesitan toda esta información. En su lugar, tales usuarios necesitan acceder sólo a una parte de la base de datos. Para que su interacción con el sistema se simplifique, se define la abstracción del nivel de vistas. El sistema puede proporcionar muchas vistas para la misma base de datos.

## Modelo de datos relacional

El modelo entidad-relación (E-R) es un modelo de datos de alto nivel. Está basado en una percepción de un mundo real que consiste en una colección de objetos básicos, denominados entidades, y de relaciones entre estos objetos.

El modelo relacional es un modelo de menor nivel. Usa una colección de tablas para representar tanto los datos como las relaciones entre los datos. Su simplicidad conceptual ha conducido a su adopción general; actualmente, una vasta mayoría de productos de bases de datos se basan en el modelo relacional. Los diseñadores formulan generalmente el diseño del esquema de la base de datos modelando primero los datos en alto nivel, usando el modelo E-R, y después traduciéndolo al modelo relacional.

## Lenguaje estándar SQL

Es un lenguaje de definición y manipulación de datos para expresar las consultas a la base de datos y las modificaciones y la forma de almacenarlos. se divide en LDD (lenguaje de definición de datos) o DDL que es el que define la estructura de los datos, y el LMD (lenguaje de manipulación de datos) o DML que lleva a cabo las tareas de consulta o modificación de los datos contenidos en las Bases de Datos del Sistema Gestor de Bases de Datos.

El lenguaje SQL tiene varios componentes:

- Lenguaje de definición de datos (LDD). El LDD de SQL proporciona órdenes para la definición de esquemas de relación, borrado de relaciones, creación de índices y modificación de esquemas de relación.
- Lenguaje interactivo de manipulación de datos (LMD). El LMD de SQL incluye un lenguaje de consultas, basado tanto en el álgebra relacional como en el cálculo relacional de tuplas. Incluye también órdenes para insertar, borrar y modificar tuplas de la base de datos.
- Definición de vistas. El LDD de SQL incluye órdenes para la definición de vistas.
- Control de transacciones. SQL incluye órdenes para la especificación del comienzo y final de transacciones.

## Transaccion

Se llama transacción a una colección de operaciones que forman una única unidad lógica de trabajo. Un sistema de base de datos debe asegurar que la ejecución de las transacciones se realice adecuadamente a pesar de la existencia de fallos: o se ejecuta la transacción completa o no se ejecuta en absoluto. Además, debe gestionar la ejecución concurrente de las transacciones evitando introducir inconsistencias.

## Arquitectura de un SGBD

Un sistema de bases de datos se divide en módulos que se encargan de cada una de las responsabilidades del sistema completo. Los componentes funcionales de un sistema de bases de datos se pueden dividir a grandes rasgos en los componentes gestor de almacenamiento y procesador de consultas. El gestor de consultas es importante porque las bases de datos requieren normalmente una gran cantidad de espacio de almacenamiento. Las bases de datos corporativas tienen un tamaño de entre cientos de gigabytes y, para las mayores bases de datos, terabytes de datos. Un gigabyte son 1.000 megabytes (1.000 millones de bytes), y un terabyte es 1 millón de megabytes (1 billón de bytes). Debido a que la memoria principal de los computadores no puede almacenar esta gran cantidad de información, esta se almacena en discos. Los datos se trasladan entre el disco de almacenamiento y la memoria principal cuando es necesario. Como la transferencia de datos a y desde el disco es lenta comparada con la velocidad de la unidad central de procesamiento, es fundamental que el sistema de base de datos estructure los datos para minimizar la necesidad de movimiento de datos entre el disco y la memoria principal. El procesador de consultas es importante porque ayuda al sistema de bases de datos a simplificar y facilitar el acceso a los datos. Las vistas de alto nivel ayudan a conseguir este objetivo. Con ellas, los usuarios del sistema no deberían ser molestados innecesariamente con los detalles físicos de implementación del sistema. Sin embargo, el rápido procesamiento de las actualizaciones y de las consultas es importante. Es trabajo del sistema de bases de datos traducir las actualizaciones y las consultas escritas en un lenguaje no procedimental, en el nivel lógico, en una secuencia de operaciones en el nivel físico.

## Gestor de almacenamiento

Un gestor de almacenamiento es un módulo de programa que proporciona la interfaz entre los datos de bajo nivel en la base de datos y los programas de aplicación y consultas emitidas al sistema. El gestor de almacenamiento es responsable de la interacción con el gestor de archivos. Los datos en bruto se almacenan en disco usando un sistema de archivos, que está disponible habitualmente en un sistema operativo convencional. El gestor de almacenamiento traduce las diferentes instrucciones LMD a órdenes de un sistema de archivos de bajo nivel. Así, el gestor de almacenamiento es responsable del almacenamiento, recuperación y actualización de los datos en la base de datos. Los componentes del gestor de almacenamiento incluyen:

- Gestor de autorización e integridad, que comprueba que se satisfagan las restricciones de integridad y la autorización de los usuarios para acceder a los datos.
- Gestor de transacciones, que asegura que la base de datos quede en un estado consistente (correcto) a pesar de los fallos del sistema, y que las ejecuciones de transacciones concurrentes ocurran si conflictos.
- Gestor de archivos, que gestiona la reserva de espacio de almacenamiento de disco y las estructuras de datos usadas



para representar la información almacenada en disco.

- Gestor de memoria intermedia, que es responsable de traer los datos del disco de almacenamiento a memoria principal y decidir qué datos tratar en memoria caché. El gestor de memoria intermedia es una parte crítica del sistema de bases de datos, ya que permite que la base de datos maneje tamaños de datos que son mucho mayores que el tamaño de la memoria principal. El gestor de almacenamiento implementa varias estructuras de datos como parte de la implementación física del sistema:
- Archivos de datos, que almacenan la base de datos en sí.
- Diccionario de datos, que almacena metadatos acerca de la estructura de la base de datos, en particular, el esquema de la base de datos.
- Índices, que proporcionan acceso rápido a elementos de datos que tienen valores particulares.

## Procesador de consultas

Los componentes del procesador de consultas incluyen:

- Intérprete del LDD, que interpreta las instrucciones del LDD y registra las definiciones en el diccionario de datos.
- Compilador del LMD, que traduce las instrucciones del LMD en un lenguaje de consultas a un plan de evaluación que consiste en instrucciones de bajo nivel que entiende el motor de evaluación de consultas. Una consulta se puede traducir habitualmente en varios planes de ejecución alternativos que proporcionan el mismo resultado. El compilador del LMD también realiza optimización de consultas, es decir, elige el plan de evaluación de menor coste de entre todas las alternativas.
- Motor de evaluación de consultas, que ejecuta las instrucciones de bajo nivel generadas por el compilador del LMD.

## Seguridad

Los SGBD permiten definir autorizaciones o derechos de acceso a diferentes niveles: al nivel global de toda la BD, al nivel entidad y al nivel atributo.

Estos mecanismos de seguridad requieren que el usuario se pueda identificar. Se acostumbra a utilizar códigos de usuarios (y grupos de usuarios) acompañados de contraseñas (passwords), pero también se utilizan tarjetas magnéticas, identificación por reconocimiento de la voz, etc.

Nos puede interesar almacenar la información con una codificación secreta; es decir, con técnicas de encriptación (como mínimo se deberían encriptar las contraseñas). Muchos de los SGBD actuales tienen prevista la encriptación.

Prácticamente todos los SGBD del mercado dan una gran variedad de herramientas para la vigilancia y la administración de la seguridad. Los hay que, incluso, tienen opciones (con precio separado) para los SI con unas exigencias altísimas, como por ejemplo los militares

## Flexibilidad e independencia

Dar flexibilidad a los cambios para ir adaptándose a la evolución de los sistemas de información. Obtener el máximo de independencia posible entre los procesos de usuarios y los datos, para esto los desarrolladores deben poder desconocer las características físicas de la BD con que trabajan para hacer cambios de tecnología y cambios físicos sin afectaciones. Este tipo de independencia

Recibe el nombre de independencia física de los datos. Por otro lado, también se requiere que los desarrolladores no tengan que hacer cambios en la lógica del SI cuando se modifica la estructura o esquema de la BD. Este tipo de independencia se denomina independencia lógica de los datos

## Problemas de la redundancia

En el mundo de los ficheros tradicionales, cada aplicación utilizaba su fichero. Sin embargo, puesto que se daba mucha coincidencia de datos entre aplicaciones, se producía también mucha redundancia entre los ficheros. El verdadero problema es el grave riesgo de inconsistencia o incoherencia de los datos; es decir, la pérdida de integridad que las actualizaciones pueden provocar cuando existe redundancia. Por lo tanto, convendría evitar la redundancia. En principio, nos conviene hacer que un dato sólo figure una vez en la BD. Sin embargo, esto no siempre será cierto.

## Integridad de los datos

La integridad se refiere a que los datos sean correctos, vigentes y de calidad. Que no haya duplicidad de datos, es decir redundancia. Datos dañados también por errores de programas o humanos, averías de discos o por otros motivos.

Nos interesará que los SGBD aseguren el mantenimiento de la calidad de los datos en cualquier circunstancia. Acabamos de ver que la redundancia puede provocar pérdida de integridad de los datos, pero no es la única causa posible. Se podría perder la corrección o la consistencia de los datos por muchas otras razones: errores de programas, errores de operación humana, avería de disco, transacciones incompletas por corte de alimentación eléctrica, etc.

Aparte de las reglas de integridad que el diseñador de la BD puede definir y que el SGBD entenderá y hará cumplir, el mismo SGBD tiene reglas de integridad inherentes al modelo de datos que utiliza y que siempre se cumplirán. Son las denominadas reglas de integridad del modelo. Las reglas definibles por parte del usuario son las reglas de integridad del usuario.

El concepto de integridad de los datos va más allá de prevenir que los programas usuarios almacenen datos incorrectos. En casos de errores o desastres, también podríamos perder la integridad de los datos. El SGBD nos debe dar las herramientas para reconstruir o restaurar los datos estropeados.

## Concurrencia de usuarios

Permitir que varios usuarios puedan acceder concurrentemente a la misma BD. Cuando el acceso es para modificación se da el problema de interferencia y puede perder integridad la BD. Para esto se definen las transacciones que son ejecutadas completas.

Cuando los accesos concurrentes son todos de lectura (es decir, cuando la BD sólo se consulta), el problema que se produce es simplemente de rendimiento, causado por las limitaciones de los soportes de que se dispone: pocos mecanismos de acceso independientes, movimiento del brazo y del giro del disco demasiado lentos, buffers locales demasiado pequeños, etc.

Cuando un usuario o más de uno están actualizando los datos, se pueden producir problemas de interferencia que tengan como consecuencia la obtención de datos erróneos y la pérdida de integridad de la BD.

Para tratar los accesos concurrentes, los SGBD utilizan el concepto de transacción de BD, concepto de especial utilidad para todo aquello que hace referencia a la integridad de los datos, como veremos a continuación.

Para indicar al SGBD que damos por acabada la ejecución de la transacción, el programa utilizará la operación de COMMIT. Si el programa no puede acabar normalmente (es decir, si el conjunto de operaciones se ha hecho sólo de forma parcial), el SGBD tendrá que deshacer todo lo que la transacción ya haya hecho. Esta operación se denomina ROLLBACK.

Nos interesará que el SGBD ejecute las transacciones de forma que no se interfieran; es decir, que queden aisladas unas de otras. Para conseguir que las transacciones se ejecuten como si estuviesen aisladas, los SGBD utilizan distintas técnicas. La más conocida es el bloqueo (lock). El bloqueo de unos datos en beneficio de una transacción consiste en poner limitaciones a los accesos que las demás transacciones podrán hacer a estos datos



## Conclusiones y recomendaciones

Que el estudiante comprenda la a fondo las estructuras, funciones e instrucciones que componen una base de datos, así como su administración, aplicación y uso

## Referencias bibliográficas

1. Silberschatz, Korth, Sudarshan. (2002). FUNDAMENTOS DE BASES DE DATOS. Cuarta edición. Madrid, España: McGRAW-HILL/INTERAMERICANA DE ESPAÑA.
2. Rafael Camps Paré, Luis Alberto Casillas Santillán, Dolors Costal Costa, Marc Gibert Ginestà. (2005). Bases de datos. Barcelona, España, ISBN: 84-9788-269-5



[www.usanmarcos.ac.cr](http://www.usanmarcos.ac.cr)

San José, Costa Rica