HISTORIA Y EVOLUCION DE LOS SGBD

El término bases de datos fue escuchado por primera vez en un simposio celebrado en California en 1963.  
En una primera aproximación, se puede decir que una base de datos es un conjunto de información relacionada que se encuentra agrupada o estructurada.

Desde el punto de vista informático, una base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.  
Por su parte, un sistema de Gestión de Bases de datos es un tipo de software muy específico dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan; o lo que es lo mismo, una agrupación de programas que sirven para definir, construir y manipular una base de datos, permitiendo así almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

Actualmente, están teniendo un impacto decisivo sobre el creciente uso de las  
computadoras.

Los orígenes de las bases de datos se remontan a la Antigüedad donde ya existían bibliotecas y toda clase de registros. Sin embargo, su búsqueda era lenta y poco eficaz y no se contaba con la ayuda de máquinas que pudiesen reemplazar el trabajo manual.

Posteriormente, el uso de las bases de datos se desarrolló a partir de las necesidades de almacenar grandes cantidades de información o datos.

En 1884 Herman Hollerith creó la máquina automática de tarjetas perforadas,

siendo nombrado así el primer ingeniero estadístico de la historia.

Posteriormente, en la década de los cincuenta se da origen a las cintas magnéticas, para automatizar la información y hacer respaldos. Esto sirvió para suplir las necesidades de información de las nuevas industrias. Y a través de este mecanismo se empezaron a automatizar información, con la desventaja de que solo se podía hacer de forma secuencial.

Las aplicaciones informáticas de los años sesenta acostumbraban a darse totalmente por lotes (batch) y estaban pensadas para una tarea muy específica relacionada con muy pocas entidades tipo.

Cada aplicación (una o varias cadenas de programas) utilizaba ficheros de movimientos para actualizar (creando una copia nueva) y/o para consultar uno o dos ficheros maestros o, excepcionalmente, más de dos.

Cada programa trataba como máximo un fichero maestro, que solía estar sobre cinta magnética y, en consecuencia, se trabajaba con acceso secuencial.

Cada vez que se le quería añadir una aplicación que requería el uso de algunos de los datos que ya existían y de otros nuevos, se diseñaba un fichero nuevo con todos los datos necesarios (algo que provocaba redundancia). Para evitar que los programas tuviesen que leer muchos ficheros.

A medida que se fueron introduciendo las líneas de comunicación, los terminales y los discos, se fueron escribiendo programas que permitían a varios usuarios consultar los mismos ficheros on-line y de forma simultánea. Más adelante fue surgiendo la necesidad de hacer las actualizaciones también on-line.

A medida que se integraban las aplicaciones, se tuvieron que interrelacionar sus ficheros y fue necesario eliminar la redundancia.

El nuevo conjunto de ficheros se debía diseñar de modo que estuviesen interrelacionados; al mismo tiempo, las informaciones redundantes (como por ejemplo, el nombre y la dirección de los clientes o el nombre y el precio de los productos), que figuraban en los ficheros de más de una de las aplicaciones, debían estar ahora en un solo lugar. El acceso on-line y la utilización eficiente de las interrelaciones exigían estructuras físicas que diesen un acceso rápido, como por ejemplo los índices, las multilistas, etc.

Estos conjuntos de ficheros interrelacionados, con estructuras complejas y compartidas por varios procesos de forma simultánea (unos on-line y otros por lotes), recibieron al principio el nombre de Data Banks, y después, a inicios de los años setenta, el de Data Bases. El software de gestión de ficheros era demasiado elemental para dar satisfacción a todas estas necesidades.

La utilización de estos conjuntos de ficheros por parte de los programas de aplicación era  excesivamente compleja, de modo que, especialmente durante la segunda mitad de los años setenta, fue saliendo al mercado software más sofisticado:

Los Data Base Management Systems, que aquí denominamos Sistemas de Gestión de BD (SGBD).

En otras palabras, una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelaciones. La representación será única e integrada, a pesar de que debe permitir utilizaciones varias y simultáneas.

En los sesenta, las computadoras bajaron los precios para que las compañías privadas las pudiesen adquirir; dando paso a que se popularizara el uso de los discos, cosa que fue un adelanto muy efectivo en la época, debido a que a partir de este soporte se podía consultar la información directamente, sin tener que saber la ubicación exacta de los datos.

En esta misma época se dio inicio a las primeras generaciones de bases de datos de red y las bases de datos jerárquicas, ya que era posible guardar estructuras de datos en listas y arboles.

En esta misma década, se llevo a cabo el desarrollo del IDS desarrollado por Charles Bachman (que formaba parte de la CODASYL) supuso la creación de un nuevo tipo de sistema de bases de datos conocido como modelo en red que permitió la creación de un estándar en los sistemas de bases de datos gracias a la creación de nuevos lenguajes de sistemas de información.

CODASYL (**Conference on Data Systems Languages**) era un consorcio de industrias informáticas que tenían como objetivo la regularización de un lenguaje de programación estándar que pudiera ser utilizado en multitud de ordenadores.

Los primeros SGBD de los años sesenta todavía no se les denominaba así. Estaban orientados a facilitar la utilización de grandes conjuntos de datos en los que las interrelaciones eran complejas.

Los SGBD estaban íntimamente ligados al software de comunicaciones y de gestión de transacciones.

Aunque para escribir los programas de aplicación se utilizaban lenguajes de alto nivel como Cobol o PL/I, se disponía también de instrucciones y de subrutinas especializadas para tratar las BD que requerían que el programador conociese muchos detalles del diseño físico, y que hacían que la programación fuese muy compleja.

Puesto que los programas estaban relacionados con el nivel físico, se debían modificar continuamente cuando se hacían cambios en el diseño y la organización de la BD. La preocupación básica era maximizar el rendimiento: el tiempo de respuesta y las transacciones por segundo.

Por lo que respecta la década de los setenta, Edgar Frank Codd definió el modelo relacional a la par que publicó una serie de reglas para los sistemas de datos relacionales a través de su artículo “Un modelo relacional de datos para grandes bancos de datos compartidos”. Este hecho dio paso al nacimiento de la segunda generación de los Sistemas Gestores de Bases de Datos.

Como consecuencia de esto, durante la década de 1970, Lawrence J. Ellison, más conocido como Larry Ellison, desarrolló el Relational Software System, o lo que es lo mismo, lo que actualmente se conoce como Oracle Corporation, desarrollando así un sistema de gestión de bases de datos relacional con el mismo nombre que dicha compañía.

Posteriormente en la época de los ochenta también se desarrollará el **SQL** (**Structured Query Language**) o lo que es lo mismo un lenguaje de consultas o lenguaje declarativo de acceso a bases de datos relacionales que permite efectuar consultas con el fin de recuperar información de interés de una base de datos y hacer cambios sobre la base de datos de forma sencilla; además de analiza grandes cantidades de información y permitir especificar diversos tipos de operaciones frente a la misma información, a diferencia de las bases de datos de los años ochenta que se diseñaron para aplicaciones de procesamiento de transacciones.

En la década de los años 80’, se desarrolló el SQL (Structured Query Language), un lenguaje de consultas que permite consultar, valga la redundancia, con el fin de recuperar información de una base de datos y a su vez, hacer cambios sobre esa misma base, de forma sencilla. Permitía analizar gran cantidad de información y especificar varios tipos de operaciones con la misma

información, a diferencia de los años anteriores, cuando se diseñaron aplicaciones de procesamientos de transacciones.

SQL comenzó a ser el modelo estándar de las industrias, con su base de datos bajo un sistema de tablas (filas y columnas), pudo competir con las bases jerárquicas y de redes, ya que su nivel de programación era sencillo y el nivel era relativamente bajo.

Al acabar la década de los ochenta, los SGBD relacionales ya se utilizaban prácticamente en todas las empresas. A pesar de todo, hasta la mitad de los noventa, cuando se ha necesitado un rendimiento elevado se han seguido utilizando los SGBD pre-relacionales.

A finales de los ochenta y principios de los noventa, las empresas se han encontrado con el hecho de que sus departamentos han ido comprando ordenadores departamentales y personales, y han ido haciendo aplicaciones con BD. El resultado ha sido que en el seno de la empresa hay numerosas BD y varios SGBD de diferentes tipos o proveedores.

Este fenómeno de multiplicación de las BD y de los SGBD se ha visto incrementado por la fiebre de las fusiones de empresas.

Esta distribución ideal se consigue cuando las diferentes BD son soportadas por una misma marca de SGBD, es decir, cuando hay homogeneidad.

Sin embargo, esto no es tan sencillo si los SGBD son heterogéneos. En la actualidad, gracias principalmente a la estandarización del lenguaje SQL, los SGBD de marcas diferentes pueden darse servicio unos a otros y colaborar para dar servicio a un programa de aplicación. No obstante, en general, en los casos de heterogeneidad no se llega a poder dar en el programa que los utiliza la apariencia de que se trata de una única BD.

Además de esta distribución “impuesta”, al querer tratar de forma integrada distintas BD preexistentes, también se puede hacer una distribución “deseada”, diseñando una BD distribuida físicamente, y con ciertas partes replicadas en diferentes sistemas.

Las razones básicas por las que interesa esta distribución son las siguientes:

* Disponibilidad y Coste.

Los tipos de datos que se pueden definir en los SGBD relacionales de los años ochenta y noventa son muy limitados. La incorporación de tecnologías multimedia –imagen y sonido – en los SI hace necesario que los SGBD relacionales acepten atributos de estos tipos.

Sin embargo, algunas aplicaciones no tienen suficiente con la incorporación de tipos especializados en multimedia. Necesitan tipos complejos que el desarrollador pueda definir a medida de la aplicación.

En definitiva, se necesitan tipos abstractos de datos: TAD.

Esto nos lleva a la orientación a objetos (OO). El éxito de la OO al final de los años ochenta, en el desarrollo de software básico, en las aplicaciones de ingeniería industrial y en la construcción de

interfaces gráficas con los usuarios, ha hecho que durante la década de los noventa se extendiese en prácticamente todos los campos de la informática. En los SI se inicia también la adopción, tímida de momento, de la OO.

La utilización de lenguajes como C++ o Java requiere que los SGBD relacionales se adapten a ellos con interfaces adecuadas.

La rápida adopción de la web a los SI hace que los SGBD incorporen recursos para ser servidores de páginas web, como por ejemplo la inclusión de SQL en guiones HTML, SQL incorporado en Java, etc.

Durante estos últimos años se ha empezado a extender un tipo de aplicación de las BD denominado Data Warehouse, o almacén de datos, que también produce algunos cambios en los SGBD relacionales del mercado.

A lo largo de los años que han trabajado con BD de distintas aplicaciones, las empresas han ido acumulando gran cantidad de datos de todo tipo. Si estos datos se analizan convenientemente pueden dar información muy valiosa.

Por lo tanto, se trata de mantener una gran BD con información proveniente de toda clase de aplicaciones de la empresa (e, incluso, de fuera). Los datos de este gran almacén, el Data Warehouse, se obtienen por una replicación más o menos elaborada de las que hay en las BD que se utilizan en el trabajo cotidiano de la empresa. Estos almacenes de datos se utilizan exclusivamente para hacer consultas, de forma especial para que lleven a cabo estudios los analistas financieros, los analistas de mercado, etc.

Actualmente, los SGBD se adaptan a este tipo de aplicación, incorporando, por ejemplo, herramientas como las siguientes:

* La creación y el mantenimiento de réplicas, con una cierta elaboración de los datos.
* La consolidación de datos de orígenes diferentes.
* La creación de estructuras físicas que soporten eficientemente el análisis multidimensional.

**GESTORES DE BASE DE DATOS.**

Hoy día la mayoría de las bases de datos se presentan en formato digital, gracias a los avances tecnológicos en la informática y la electrónica.

Los gestores de bases de datos, Database Management System o DBMS (SGBD) son programas que permiten almacenar y luego acceder a los datos de forma estructurada y rápida. Las aplicaciones más usadas son para gestiones de empresas e instituciones públicas, así como en entornos científicos, para almacenar la información experimental.

Una base de datos es un sistema compuesto por un conjunto de datos, los cuales están almacenados en discos, a los que se accede directamente y un conjunto de programas que regulen o manejen ese conjunto de datos.

Mientras que un sistema de Gestión de Bases de Datos es un software que sirve de interfaz entre la base de datos, el usuario y las aplicaciones que se utilizan.

El principal lenguaje de base de datos y el más utilizado desde que se conoce la programación de gestión, es el Structured Query Language (SQL). Este, de consulta estructurada, facilita el acceso a la gestión de las bases de datos relaciones, lo que permite realizar tareas en ellas y realizar consultas, que sirvan para obtener, agregar, eliminar o modificar información.

Para el desarrollo de este lenguaje hay que utilizar un gestor de base de datos, de los que hay muchos, unos de acceso libre y otros de pago.

Los principales son:

* Oracle.
* SQL Server.
* MySQL.
* Fire Bird.

Sistemas de Información orientados a procesos.

Antes de aparecer los SGBD (década de los setenta), la información se trataba y se gestionaba utilizando los típicos sistemas de gestión de archivos que iban soportados sobre un sistema operativo. Éstos consistían en un conjunto de programas que definían y trabajaban sus propios datos. Los datos se almacenan en archivos y los programas manejan esos archivos para obtener la información. Si la estructura de los datos de los archivos cambia, todos los programas que los manejan se deben modificar. En estos sistemas de información se crean diversas aplicaciones (software) para gestionar diferentes aspectos del sistema. Cada aplicación realiza unas determinadas operaciones. Los datos de dichas aplicaciones se almacenan en archivos digitales dentro de las unidades de almacenamiento. Cada programa almacena y utiliza sus propios datos de forma un tanto caótica. La ventaja de este sistema es que los procesos son independientes por lo que la modificación de uno no afectaba al resto.

Esto supone un gran inconveniente a la hora de tratar grandes volúmenes de información. Surge así la idea de separar los datos contenidos en los archivos de los programas que los manipulan, es decir, que se pueda modificar la estructura de los datos de los archivos sin que por ello se tengan que modificar los programas con los que trabajan. Se trata de estructurar y organizar los datos de forma que se pueda acceder a ellos con independencia de los programas que los gestionan. Inconvenientes de un sistema de gestión de archivos:

* Redundancia e Inconsistencia de los Datos.
* Dependencia de los datos Físicas-Lógicas.
* Dificultad para tener acceso a los datos.
* Separación y aislamiento de los datos.
* Dificultad para el acceso concurrente.
* Dependencia de la estructura del archivo con el lenguaje de programación.
* Problemas en la seguridad de los datos.
* Problemas de integridad de datos (datos inconsistentes).

Sistemas de Información orientados a los datos. Bases de Datos.

Todos estos inconvenientes hacen posible el fomento y desarrollo de SGBD. El objetivo primordial de un gestor es proporcionar eficiencia y seguridad a la hora de extraer o almacenar información en las BD. Los sistemas gestores de BBDD están diseñados para gestionar grandes bloques de información, que implica tanto la definición de estructuras para el almacenamiento como de mecanismos para la gestión de la información. En este tipo de sistemas los datos se centralizan en una base de datos común a todas las aplicaciones. En esos sistemas los datos se almacenan en una única estructura lógica que es utilizable por las aplicaciones. A través de esa estructura se accede a los datos que son comunes a todas las aplicaciones. Cuando una aplicación modifica un dato, dicho dato la modificación será visible para el resto de aplicaciones. Una BD es un gran almacén de datos que se define una sola vez; los datos pueden ser accedidos de forma simultánea por varios usuarios; están relacionados y existe un número mínimo de duplicidad; además en las BBDD se almacenarán las descripciones de esos datos, lo que se llama metadatos en el diccionario de datos.

Ventajas:

* Independencia de los datos y los programas y procesos.
* Menor redundancia.
* Integridad de los datos.
* Mayor seguridad en los datos.
* Datos más documentados.
* Acceso a los datos más eficiente.
* Menor espacio de almacenamiento.
* Acceso simultáneo a los datos.

Desventajas:

* Instalación costosa.
* Requiere personal cualificado.
* Implantación larga y difícil.
* Ausencia de estándares reales.

**Arquitectura de los Sistemas Gestores de Bases de Datos.**

Un sistema gestor de bases de datos o SGBD es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos. En estos sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos. El éxito del SGBD reside en mantener la seguridad e integridad de los datos. Lógicamente tiene que proporcionar herramientas a los distintos usuarios. Entre las herramientas que proporciona están:

* Herramientas para la creación y especificación de los datos.
* Herramientas para administrar y crear la estructura física.
* Herramientas para la manipulación de los datos.
* Herramientas de recuperación.
* Herramientas para la creación de copias de seguridad.
* Herramientas para la gestión de la comunicación.
* Herramientas para la creación de aplicaciones.
* Herramientas de instalación.
* Herramientas para la exportación e importación.

**Niveles de abstracción de una base de datos.**

Introducción En cualquier sistema de información se considera que se pueden observar los datos desde dos puntos de vista:

Nivel externo: Esta es la visión de los datos que poseen los usuarios del Sistema de Información.

**Arquitectura de los Sistemas Gestores de Bases de Datos**

Nivel físico: Esta es la forma en la que realmente están almacenados los datos. Realmente la base de datos es la misma, pero se la puede observar desde estos dos puntos de vista.

En todo sistema de información digital, los usuarios ven los datos desde las aplicaciones creadas por los programadores. A ese nivel se manejan formularios, informes en pantalla o en papel, Pero la realidad física de esos datos, tal cual se almacenan en los discos queda oculta a los usuarios. Esa forma de ver la base de datos está reservada a los administradores. Es el nivel físico el que permite ver la base de datos en función de cómo realmente se están almacenando en el ordenador, en qué carpeta, qué archivos se usan… En el caso de los Sistemas de Base de datos, se añade un tercer nivel, un tercer punto de vista, es el nivel conceptual. Ese nivel se sitúa entre el físico y el externo.

En 1975, el comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles para los SGBD cuyo objetivo principal era el de separar los programas de aplicación de la BD física. En esta arquitectura el esquema de una BD se define en tres niveles de abstracción distintos:

* Nivel interno o físico: el más cercano al almacenamiento físico, es decir, tal y como están almacenados en el ordenador. Este esquema se especifica con un modelo físico y describe los detalles de cómo se almacenan físicamente los datos: los archivos que contienen la información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, los campos que los componen, etcétera. Esta visión sólo la requiere el administrador/a. El administrador la necesita para poder gestionar más eficientemente la base de datos.
* Nivel externo o de visión: es el más cercano a los usuarios, es decir, es donde se describen varios esquemas externos o vistas de usuarios. Se trata de la visión de los datos que poseen los usuarios y usuarias finales.
* Nivel conceptual: describe la estructura de toda la BD para un grupo de usuarios mediante un esquema conceptual. Este esquema describe las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento. Representa la información contenida en la BD. Se trata de un esquema teórico de los datos en el que figuran organizados en estructuras reconocibles del mundo real y en el que también aparece la forma de relacionarse los datos. Este esquema es el paso que permite modelar un problema real a su forma correspondiente en el ordenador. Este esquema es la base de datos de todos los demás. Como se verá más adelante, es el primer paso a realizar al crear una base de datos. En definitiva es el plano o modelo general de la base de datos. El esquema conceptual lo realizan diseñadores/as o analistas.

**Componentes de los Sistemas Gestores de Bases de Datos**

Los SGBD son paquetes de software muy complejos que deben proporcionar una serie de servicios que van a permitir almacenar y explotar los datos de forma eficiente. Los componentes principales son los siguientes.

**Lenguajes de los SGBD.**

Todos los SGBD ofrecen lenguajes e interfaces apropiadas para cada tipo de usuario: administradores, diseñadores, programadores de aplicaciones y usuarios finales. Los lenguajes van a permitir al administrador de la BD especificar los datos que componen la BD, su estructura, las relaciones que existen entre ellos, las reglas de integridad, los controles de acceso, las características de tipo físico y las vistas externas de los usuarios. Los lenguajes del SGBD se clasifican en:

Lenguaje de definición de datos (LDD o DDL):

se utiliza para especificar el esquema de la BD, las vistas de los usuarios y las estructuras de almacenamiento. Es el que define el esquema conceptual y el esquema interno. Lo utilizan los diseñadores y los administradores de la BD.

Permite al diseñador de la base de datos crear las estructuras apropiadas para integrar adecuadamente los datos. Se dice que esta función es la que permite definir las tres estructuras de la base de datos (relacionadas con los tres niveles de abstracción).

• Estructura interna

• Estructura conceptual

• Estructura externa

Realmente esta función trabajo con metadatos. Los metadatos es la información de la base de datos que realmente sirve para describir a los datos.

Es decir, Sánchez Rodríguez y Crespo son datos; pero Primer Apellido es un metadato. También son datos decir que la base de datos contiene Alumnos o que el dni lo forman 9 caracteres de los cuales los 8 primeros son números y el noveno un carácter en mayúsculas. La función de definición sirve pues para crear, eliminar o modificar metadatos. Para ello permite usar un lenguaje de descripción de datos o DDL. Mediante ese lenguaje:

• Se definen las estructuras de datos.

• Se definen las relaciones entre los datos.

• Se definen las reglas que han de cumplir los datos.

Lenguaje de manipulación de datos (LMD o DML):

Se utilizan para leer y actualizar los datos de la BD. Es el utilizado por los usuarios para realizar consultas, inserciones, eliminaciones y modificaciones. Los hay procedurales, en los que el usuario será normalmente un programador y especifica las operaciones de acceso a los datos llamando a los procedimientos necesarios. Estos lenguajes acceden a un registro y lo procesan. Las sentencias de un DML procedural están embebidas en un lenguaje de alto nivel llamado anfitrión. Las BD jerárquicas y en red utilizan estos DML procedurales. Mediante ese lenguaje se puede:

• Añadir datos

• Eliminar datos

• Modificar datos

• Buscar datos

Actualmente se suele distinguir aparte la función de buscar datos en la base de datos (función de consulta). Para lo cual se proporciona un lenguaje de consulta de datos o DQL.

Lenguaje de control de datos (LCD o DCL):

Mediante esta función los administradores poseen mecanismos para proteger los datos; de modo que se permite a cada usuario ver ciertos datos y otros no; o bien usar ciertos recursos concretos de la base de datos y prohibir otros. Es decir simplemente permite controlar la seguridad de la base de datos. El lenguaje que implementa esta función es el lenguaje de control de datos o DCL. No procedurales son los lenguajes declarativos.

En muchos SGBD se pueden introducir interactivamente instrucciones del LMD desde un terminal, también pueden ir embebidas en un lenguaje de programación de alto nivel. Estos lenguajes permiten especificar los datos a obtener en una consulta, o los datos a modificar, mediante sentencias sencillas. Las BD relacionales utilizan lenguajes no procedurales como SQL (Structured Query Language) o QBE (Query By Example). La mayoría de los SGBD comerciales incluyen lenguajes de cuarta generación (4GL) que permiten al usuario desarrollar aplicaciones de forma fácil y rápida, también se les llama herramientas de desarrollo. Ejemplos de esto son las herramientas del SGBD ORACLE: SQL Forms para la generación de formularios de pantalla y para interactuar con los datos; SQL Reports para generar informes de los datos contenidos en la BD; PL/SQL lenguaje para crear procedimientos que interractuen con los datos de la BD.

**Recursos humanos de las bases de datos.**

Intervienen (como ya se ha comentado) muchas personas en el desarrollo y manipulación de una base de datos. Habíamos seleccionado cuatro tipos de usuarios (administradores/as, desarrolladores, diseñadores/as y usuarios/as). Ahora vamos a desglosar aún más esta clasificación.

**Informáticos**

Lógicamente son los profesionales que definen y preparan la base de datos. Pueden ser:

* Directivos/as.
* Analistas.
* Administradores/as de las bases de datos.
* Desarrolladores/as o programadores/as.
* Equipo de mantenimiento.

**Usuarios**

* Expertos.
* Habituales.
* Ocasionales.

**El Administrador de la Base de Datos.**

El DBA tiene una gran responsabilidad ya que posee el máximo nivel de privilegios. Será el encargado de crear los usuarios que se conectarán a la BD. En la administración de una BD siempre hay que procurar que haya el menor número de administradores, a ser posible una sola persona. El objetivo principal de un DBA es garantizar que la BD cumple los fines previstos por la organización, lo que incluye una serie de tareas como:

* Instalar SGBD en el sistema informático.
* Crear las BBDD que se vayan a gestionar.
* Crear y mantener el esquema de la BD.
* Crear y mantener las cuentas de usuario de la BD.
* Arrancar y parar SGBD, y cargar las BBDD con las que se ha de trabajar.
* Colaborar con el administrador del S.O. en las tareas de ubicación, dimensionado y control de los archivos y espacios de disco ocupados por el SGBD.
* Colaborar en las tareas de formación de usuarios.
* Establecer estándares de uso, políticas de acceso y protocolos de trabajo diario para los usuarios de la BD.
* Suministrar la información necesaria sobre la BD a los equipos de análisis y programación de aplicaciones.

**Efectuar tareas de explotación como:**

* Vigilar el trabajo diario colaborando en la información y resolución de las dudas de los usuarios de la BD.
* Controlar en tiempo real los accesos, tasas de uso, cargas en los servidores, anomalías, etcétera.
* Llegado el caso, reorganizar la BD.
* Efectuar las copias de seguridad periódicas de la BD.
* Restaurar la BD después de un incidente material a partir de las copias de seguridad.
* Estudiar las auditorías del sistema para detectar anomalías, intentos de violación de la seguridad, etcétera.
* Ajustar y optimizar la BD mediante el ajuste de sus parámetros, y con ayuda de las herramientas de monitorización y de las estadísticas del sistema.
* En su gestión diaria, el DBA suele utilizar una serie de herramientas de administración de la BD.
* Con el paso del tiempo, estas herramientas han adquirido sofisticadas prestaciones y facilitan en gran medida la realización de trabajos que, hasta no hace demasiado, requerían de arduos esfuerzos por parte de los administradores.

**Tipos de Sistemas Gestores de Bases de Datos.**

Cada SGBD puede utilizar un modelo diferente para los datos. Por lo que hay modelos conceptuales diferentes según que SGBD utilicemos.

Diferencias entre el modelo lógico y el conceptual

El modelo conceptual es independiente del DBMS que se vaya a utilizar.

El lógico depende de un tipo de SGBD en particular.

El modelo lógico está más cerca del modelo físico, el que utiliza internamente el ordenador. El modelo conceptual es el más cercano al usuario, el lógico es el encargado de establecer el paso entre el modelo conceptual y el modelo físico del sistema.

Tipos de Sistemas Gestores de Bases de Datos

Algunos ejemplos de modelos conceptuales son:

* Modelo Entidad Relación
* Modelo RM/T
* Modelo UML

Ejemplos de modelos lógicos son:

* Modelo relacional
* Modelo Codasyl
* Modelo Jerárquico