

CAPITULO 3. CONCEPTO DE MODELO DE DATOS

3.1 Introducción

Entre las consideraciones fundamentales al organizar los datos en una base de datos se incluye, la determinación de los datos que deben recopilar, quienes tendrá acceso a la información y como podrían desear utilizar los datos almacenados. Con base en estas determinaciones se puede crear una base de datos. Su creación requiere de dos tipos de diseño diferentes: un diseño lógico y uno físico.

El diseño lógico de una base de datos muestra un modelo abstracto de cómo se debe estructurar y ordenar los datos para cumplir con las necesidades de información en una organización. El diseño lógico de una base de datos incluye la identificación de las relaciones entre las diferentes sesiones de datos y su agrupamiento en forma ordenada.

Una de las características fundamentales de los sistemas de bases de datos es que proporcionan cierto nivel de abstracción de datos, al ocultar las características sobre el almacenamiento físico que la mayoría de usuarios no necesita conocer. Los modelos de datos son el instrumento principal para ofrecer dicha abstracción.

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. Los modelos de datos contienen también un conjunto de operaciones básicas para la realización de consultas (lecturas) y actualizaciones de datos. Además, los modelos de datos más modernos incluyen conceptos para especificar comportamiento, permitiendo especificar un conjunto de operaciones definidas por el usuario.

Los modelos de datos se pueden clasificar dependiendo de los tipos de conceptos que ofrecen para describir la estructura de la base de datos. Los modelos de datos de alto nivel, o modelos conceptuales, disponen de conceptos muy cercanos al modo en que la mayoría de los usuarios percibe los datos, mientras que los modelos de datos de bajo nivel, o modelos físicos, proporcionan conceptos que describen los detalles de cómo se almacenan los datos en el ordenador.

Los conceptos de los modelos físicos están dirigidos al personal informático, no a los usuarios finales. Entre estos dos extremos se encuentran los modelos lógicos, cuyos conceptos pueden ser entendidos por los usuarios finales, aunque no están demasiado alejados de la forma en que los datos se organizan físicamente. Los modelos lógicos ocultan algunos detalles de cómo se almacenan los datos, pero pueden implementarse de manera directa en un ordenador.

Los modelos conceptuales utilizan conceptos como entidades, atributos y relaciones. Una entidad representa un objeto o concepto del mundo real como, por ejemplo, un empleado de la empresa inmobiliaria o una oficina. Un atributo representa alguna propiedad de interés de una entidad como, por ejemplo, el nombre o el salario del empleado. Una relación describe una interacción entre dos o más entidades, por ejemplo, la relación de trabajo entre un empleado y su oficina.

Cada SGBD soporta un modelo lógico, siendo los más comunes el relacional, el de red y el jerárquico. Estos modelos representan los datos valiéndose de estructuras de registros, por lo que también se denominan modelos orientados a registros. Hay una nueva familia de modelos lógicos, son los modelos orientados a objetos, que están más próximos a los modelos conceptuales.

Los modelos físicos describen cómo se almacenan los datos en el ordenador: el formato de los registros, la estructura de los ficheros (desordenados, ordenados, etc.) y los métodos de acceso utilizados (índices, etc.).

A la descripción de una base de datos mediante un modelo de datos se le denomina esquema de la base de datos. Este esquema se especifica durante el diseño, y no es de esperar que se modifique a menudo. Sin embargo, los datos que se almacenan en la base de datos pueden cambiar con mucha frecuencia: se insertan datos, se actualizan, etc. Los datos que la base de datos contiene en un determinado momento se denominan estado de la base de datos u ocurrencia de la base de datos.

La distinción entre el esquema y el estado de la base de datos es muy importante. Cuando definimos una nueva base de datos, sólo especificamos su esquema al SGBD. En ese momento, el estado de la base de datos es el "estado vacío", sin datos. Cuando se cargan datos por primera vez, la base de datos pasa al "estado inicial". De ahí en adelante, siempre que se realice una operación de actualización de la base de datos, se tendrá un nuevo estado. El SGBD se encarga, en parte, de garantizar que todos los estados de la base de datos sean estados válidos que satisfagan la estructura y las restricciones especificadas en el esquema.

Por lo tanto, es muy importante que el esquema que se especifique al SGBD sea correcto y se debe tener muchísimo cuidado al diseñarlo. El SGBD almacena el esquema en su catálogo o diccionario de datos, de modo que se pueda consultar siempre que sea necesario.

3.2 Definición de Modelos de datos.

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información.

Con respecto al diseño de bases de datos, el modelado de datos puede ser descrito así (Brodie 1984:20): "dados los requerimientos de información y proceso de una aplicación de uso intensivo de datos (por ejemplo, un sistema de información), construir una representación de la aplicación que capture las propiedades estáticas y dinámicas requeridas para dar soporte a los procesos deseados (por ejemplo, transacciones y consultas). Además de capturar las necesidades dadas en el momento de la etapa de diseño, la representación debe ser capaz de dar cabida a eventuales futuros requerimientos".

Un modelo de datos es por tanto una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo. Conceptualmente, una aplicación puede ser caracterizada por:

Propiedades estáticas: entidades (u objetos), propiedades (o atributos) de esas entidades, y relaciones entre esas entidades.

Propiedades dinámicas: operaciones sobre entidades, sobre propiedades o relaciones entre operaciones.

Reglas de integridad sobre las entidades y las operaciones (por ejemplo, transacciones).

Así, un modelo de datos se distingue de otro por el tratamiento que da a estas tres categorías. El resultado de un modelado de datos es una representación que tiene dos componentes: las propiedades estáticas se definen en un esquema y las propiedades dinámicas se definen como especificaciones de transacciones, consultas e informes.

Un esquema consiste en una definición de todos los tipos de objetos de la aplicación, incluyendo sus atributos, relaciones y restricciones estáticas. Correspondientemente, existirá una reposición de información, la base de datos, que es una instancia del esquema.

Un determinado tipo de procesos sólo necesita acceder a un subconjunto predeterminado de entidades definidas en un esquema, por lo que este tipo de procesos puede requerir sólo un subconjunto de las propiedades estáticas del esquema general. A este subconjunto de propiedades estáticas se le denomina subesquema.

Una transacción consiste en diversas operaciones o acciones sobre las entidades de esquema o subesquema. Una consulta se puede expresar como una expresión lógica sobre los objetos y relaciones definidos en el esquema; una consulta identifica un subconjunto de la base de datos. Las herramientas que se usan para realizar las operaciones de definición de las propiedades estáticas y dinámicas de la base de datos son los lenguajes de definición y manipulación de datos (DDL, DML), junto con los lenguajes de consulta (QL).

3.3 Clasificación de los Modelos de datos

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), es un método para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas; son abstracciones que permiten la implementación de un sistema eficiente de base de datos, por lo general se refieren a algoritmos, y conceptos matemáticos.

3.3.1 Modelos Lógicos Basados en Objetos

Los modelos lógicos basados en objetos se usan para describir los datos en los niveles conceptual y de visión. Se caracterizan porque proporcionan una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente.

3.3.1.2 Modelo Entidad-Relación

El modelo E-R (Entidad-Relación) es un modelo de datos conceptual de alto nivel y

que se suele utilizar bastante en el diseño de bases de datos. Se basa en una percepción del mundo real que consiste en un conjunto de objetos básicos denominados entidades y relaciones, y se desarrolló para facilitar el diseño de bases de datos.

El modelo E-R crea un modelo de la realidad que se asimila a la realidad que queremos modelar, y lo hace de forma que es independiente de la implementación posterior, ofreciendo un alto nivel de abstracción, y siendo una herramienta gráfica fácil de comprender.

El resultado del modelado E-R es un diagrama E-R que representa una estructura lógica general de la base de datos.

3.3.1.3 Modelo Orientado a Objetos

Este modelo también se basa en la percepción de una colección de objetos. Un objeto se caracteriza por tener un estado y un comportamiento. El estado corresponde a los valores que toman un conjunto de propiedades o variables de instancia, y el comportamiento es llevado a cabo mediante una serie de operaciones o funciones que operan sobre el objeto, y que se denominan métodos. Los objetos que tienen el mismo tipo de propiedades y el mismo comportamiento son agrupados en clases. Dichas clases se organizan en un diagrama o jerarquía de clases, en el que las clases pueden estar relacionadas mediante relaciones de asociación o mediante relaciones de herencia. La herencia permite la definición de clases a partir de clases existentes heredándose a las nuevas clases las propiedades y el comportamiento de las clases existentes, cumpliéndose también que todos los objetos de una subclase también es objeto de su superclase.

La única forma en la que un objeto puede acceder a los datos de otro objeto es a través de los métodos de este objeto. Esto se denomina envío de mensajes al objeto. De esta forma, la interfaz de llamada mediante los métodos de un objeto define la parte visible, mientras que la parte interna del objeto (variables y código de los métodos) no es visible externamente. De esta forma se tienen dos niveles de abstracción.

Por ejemplo, sea un objeto que representa a una cuenta corriente, y que dicho objeto contiene las variables de instancia numeroDeCuenta y saldo. Este objeto puede tener un método denominado Ingresar que añade una cantidad al saldo.

A diferencia del modelo E-R, en el modelo OO, cada objeto tiene su propia entidad que viene dado por un OID (identificador del objeto) asignado por el sistema.

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes de la programación orientada a objetos:

Encapsulación: Ocultar datos del resto de los datos, impidiendo así accesos incorrectos o conflictos.

Herencia: Reusabilidad del código.

Polimorfismo: Sobrecarga de operadores o de métodos.

3.3.2 Modelos Lógicos Basados en Registros

Los modelos lógicos basados en registros se utilizan para describir datos en los modelos conceptual y de visión. Los modelos basados en registros se llaman así porque la base de datos está estructurada en registros de formato fijo de varios tipos. Cada tipo de registro define un número fijo de campos o atributos, y cada campo normalmente es de longitud fija. Los tres modelos de datos más ampliamente extendidos son el modelo relacional, el modelo en red y el modelo jerárquico

3.3.2.1 Modelo Relacional

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postuladas su bases en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que esta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que esta compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario casual de la base de datos. La información puede ser recuperada o almacenada por medio de "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más común para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización de una base de datos.

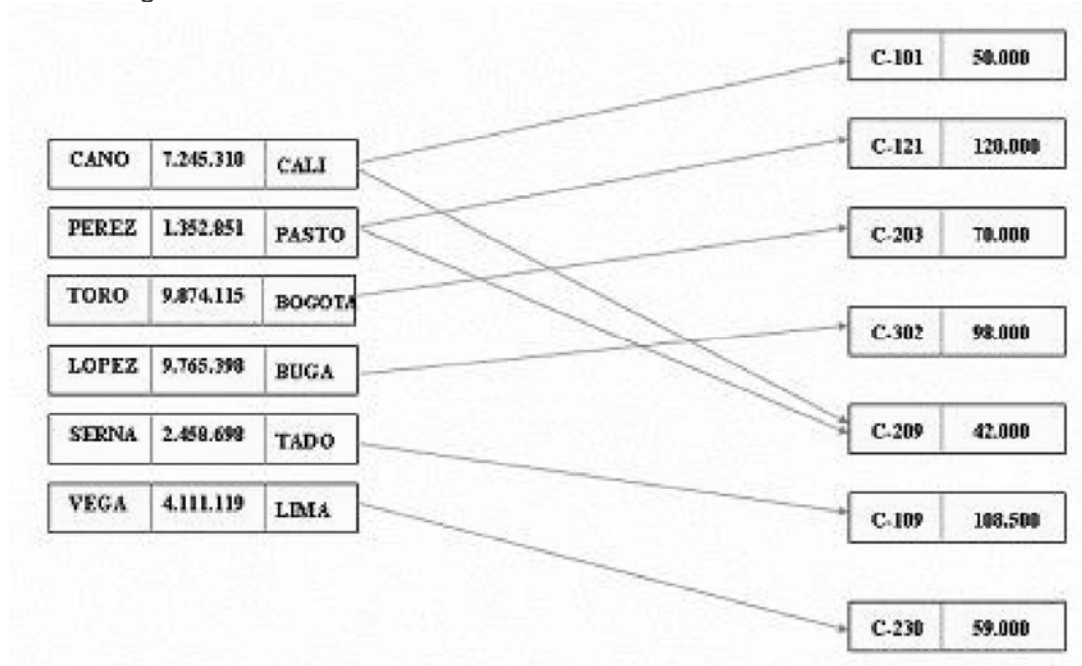
3.3.2.2 Modelo de Red

Éste es un modelo ligeramente distinto del jerárquico, en donde su diferencia fundamental es la modificación del concepto de un nodo, permitiendo que un mismo nodo tenga varios padres (algo no permitido en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos, pero aun así, la dificultad que significa administrar la información en una base de datos de red, ha significado

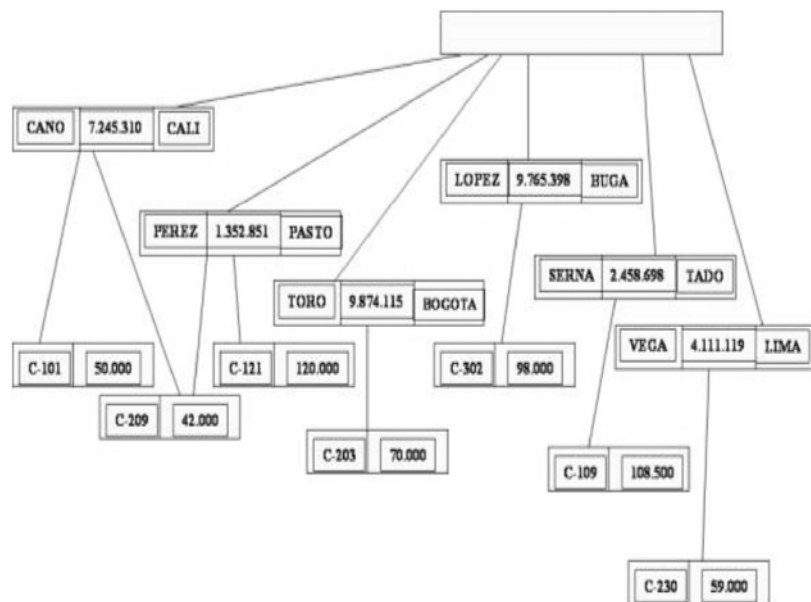
que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

Colecciones de registros y las relaciones entre datos se representan mediante enlaces dirigidos



3.3.2.3 Modelo Jerárquico

Estas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres se le conoce como raíz, y a los nodos que no tienen hijos se les conoce como hojas.



Una de las principales limitaciones de este modelo, es su incapacidad de representar eficientemente la redundancia de datos.

Los registros se organizan como colecciones de árboles, en lugar de grafos dirigidos

3.4 Lenguajes e interfaces de bases de datos

Como los usuarios de un SGBD pueden tener distintos privilegios y distintos conocimientos informáticos, es necesario disponer de diferentes lenguajes e interfaces para cada tipo de usuarios.

3.4.1 Lenguaje de definición de datos

Una vez que se ha finalizado la tarea de diseño de la base de datos, y que se ha seleccionado un SGBD para su implementación, el primer paso consiste en la especificación del esquema conceptual de la base de datos.

El esquema conceptual de la base de datos se especifica mediante una serie de definiciones expresadas en un Lenguaje de definición de datos (DDL, Data Definition Language). El SGBD contará con un compilador de DDL cuya función será procesar las sentencias en DDL para identificar las descripciones de los Elementos de los esquemas y guardar la descripción del esquema en un diccionario de datos.

El diccionario de datos es un archivo que contiene metadatos, es decir, datos acerca de los datos. Este archivo se consulta cada vez que se leen o modifican los datos del Sistema de base de datos.

3.4.2 Lenguaje de manipulación de datos

Una vez que se han compilado los esquemas de la base de datos, y que ya se han introducido datos en la base de datos, los usuarios necesitarán algún mecanismo para obtener información de la base de datos. Las operaciones más comunes de manipulación son la consulta, inserción, eliminación y modificación de datos. Para ello, el SGBD ofrece un Lenguaje de manipulación de datos (DML, Data Manipulation Language).

En general existen dos tipos de DML:

- Procedimentales. Requieren que el usuario especifique qué datos desea y cómo hay que obtenerlos.

- No procedimentales. Requieren que el usuario especifique qué datos desea sin tener que especificar cómo obtenerlos.

Los DML no procedimentales suelen ser más fáciles de utilizar para los usuarios, ya que no hay que especificar la forma en que se tienen que obtener los datos, pero esta ventaja se convierte en un inconveniente, puesto que el código que se genere puede que no sea tan eficiente como el producido por los lenguajes de consulta procedimentales.

Una consulta es una sentencia que solicita información de la base de datos, y un lenguaje de consulta es el subconjunto de un DML que se utiliza para la recuperación y actualización de información de la base de datos, pero nosotros obviaremos esta diferencia.

Por último, siempre que las sentencias del lenguaje de consulta se incluyan en un lenguaje de programación de propósito general, a este lenguaje se le denomina lenguaje anfitrión.

3.5 Interfaces para Sistemas de gestión de bases de datos

Normalmente, los usuarios de un Sistema de base de datos, utilizan un lenguaje de consulta de alto nivel, mientras que los programadores utilizan el DML para la creación de consultas. Para la mayoría de los usuarios se suelen definir interfaces de usuario amigables para la interacción con la base de datos. A continuación vamos a ver los tipos de interfaces que hay.

- Interfaces basadas en menús. Presentan al usuario una lista de opciones en forma de menús que guían al usuario en la petición de consultas. De esta forma no es necesario conocer la sintaxis de un lenguaje de consulta, pues permiten la creación de la consulta eligiendo las opciones que presenta la interfaz.
- Interfaces gráficas. Suelen presentar al usuario los esquemas en forma de diagrama, y las consultas se especifican manipulando el diagrama con el ratón.
- Interfaces basadas en formularios. Estas interfaces presentan un formulario al usuario en el que se rellenan los huecos del formulario para la modificación de los datos, o bien para especificar los parámetros de la consulta.
- Interfaces de lenguaje natural. Estas interfaces aceptan la especificación de una consulta descrita en términos de un idioma concreto y construyen expresiones DML a partir de dicha especificación.
- Interfaces parametrizadas. Se trata de interfaces para usuarios que siempre suelen realizar el mismo conjunto reducido de operaciones, reduciendo el número de pulsaciones para la creación de la consulta.

3.5.1 Usuarios y administradores de la base de datos

Uno de los objetivos primordiales de un Sistema de bases de datos es el proporcionar un entorno de recuperación de información y de almacenamiento de datos en la base de datos.

Podemos hacer una clasificación de los tipos de usuarios de una base de datos en función de la forma en que interactúan con el sistema.

Administradores de la base de datos. Persona que tiene centralizado el control del sistema.

Programadores de aplicaciones. Se trata de los profesionales que interactúan con el sistema a través de llamadas en DML, las cuales están incorporadas en un lenguaje anfitrión. A estos programas se les denominan programas de aplicación, como por ejemplo, los programas para la generación de cargos, abonos, transferencias de un sistema bancario. Como la sintaxis DML suele ser diferente de la sintaxis del lenguaje anfitrión, las llamadas en DML suelen ir precedidas de un carácter especial, de forma que se genere el código apropiado en el lenguaje anfitrión, lo cual se hace mediante un precompilador de DML, que convierte las sentencias DML en sentencias del lenguaje anfitrión. Una vez precompilado el programa, se compilaría mediante el compilador del lenguaje anfitrión, que generaría el código objeto apropiado.

Usuarios sofisticados. Son los que interactúan con el sistema sin escribir programas, escribiendo las consultas en el lenguaje de consulta de la base de datos.

Usuarios especializados. Se trata de usuarios sofisticados que crean aplicaciones de bases de datos especializadas para el procesamiento de la información.

Usuarios ingenuos. Son los usuarios que interactúan con el sistema llamando a uno de los programas desarrollados por los programadores de aplicaciones.

Como primer tipo de usuario hemos descrito la figura del administrador, un usuario vital en el enfoque de bases de datos, y que tiene unas funciones que merecen ser estudiadas más detalladamente. Estas son:

Definición del esquema conceptual. El esquema original de la base de datos se crea escribiendo un conjunto de definiciones que son traducidas por el compilador de DDL a un conjunto de metadatos que se guardan en el diccionario de datos.

Definición del esquema físico. Se trata de definir las estructuras de almacenamiento y los métodos de acceso adecuados (especificación de los tipos de índices)

Modificación del esquema y de la organización física. Si bien las modificaciones tanto del esquema de la base de datos como de la organización física no son demasiado habituales, éstas se realizan modificando el esquema conceptual y físico.

Creación de permisos para el acceso a los datos. El administrador de la base de datos es el encargado de definir los permisos que autorizan a los usuarios a acceder a ciertas partes de la base de datos.

Especificación de las restricciones de integridad. Estas restricciones se guardan en el diccionario de datos para ser consultado cada vez que se realice una actualización.

3.5.2 Gestión de transacciones

Una transacción es un conjunto de operaciones sobre una base de datos que forman una unidad lógica de trabajo, como por ejemplo una transferencia de fondos de una cuenta a otra. Es indispensable que se produzca el cargo en la cuenta origen y el abono en la cuenta destino, o que no se produzca ninguna operación. A esto se le llama atomicidad. Otro de los requisitos es que se mantenga la consistencia de la base de datos (que la suma de las dos cuentas sea constante), y un tercer requisito es que se garantice la durabilidad (que persistan los nuevos valores a pesar de la posibilidad de fallo del sistema).

La definición de las transacciones para que mantengan la consistencia es responsabilidad del programador de aplicaciones. El módulo de gestión de transacciones del SGBD es responsable de garantizar la atomicidad y la durabilidad. Necesita también un mecanismo de recuperación de fallos, para restaurar la base de datos al estado anterior al inicio de una transacción interrumpida por un fallo del sistema.

Cuando hay varias transacciones que actualizan la base de datos de forma concurrente, el módulo de control de concurrencia controla la interacción entre ellas para garantizar la consistencia.

3.6 Actividades Complementarias

- 1 Describa los siguientes tres tipos de bases de datos modelo jerárquico, modelo de red y modelo relacional.
- 2 Cuales son los propósitos del lenguaje de definición de datos (DDL) y de un diccionario de datos?
- 3 Haga un cuadro comparativo en donde resalte las funciones y responsabilidades de los usuarios y administradores frente al manejo y administración de las bases de datos.
- 4 Explique cada uno de los elementos que intervienen en un modelo de bases de datos orientado a objetos.
- 5 ¿Qué es y para que sirve el nivel externo de una base de datos?
- 6 ¿Explique cuál es el nivel conceptual de un sistema de base de datos?
- 7 ¿Para usted que significado tiene base de datos y que importancia tiene en el ámbito empresarial?