

## 2. MODELOS DE DATOS

### Introducción

#### 2.1 Entidad – relación

#### 2.2 Jerárquico

#### 2.3 De red

#### 2.4 Relacional

### *Introducción*

Hoy en día las empresas manejan una gran cantidad de datos. Cualquier empresa que se precie debe tener almacenados todos estos datos en una base de datos para poder realizarlos mediante una aplicación profesional; sin esta funcionalidad resultaría imposible tratar y manejar en su totalidad los datos que lleva a cabo la empresa y se perdería un tiempo y un dinero muy valiosos

Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la base de datos, en donde lo más importante son **LOS MODELOS DE DATOS**.

Si los modelos no son definidos apropiadamente, podemos tener muchos dolores de cabeza al momento de ejecutar consultas a la base de datos para tratar de obtener algún tipo de datos.

No importa si nuestra base de datos tiene sólo 20 registros, o algunos cuantos miles, es importante asegurarnos que nuestra base de datos está correctamente diseñada para que tenga eficiencia y que se pueda seguir utilizando por largo del tiempo.

En este TEMA se mencionarán algunos principios básicos del diseño de base de datos y se tratarán algunas reglas que se deben seguir cuando se crean bases de datos.

Dependiendo de los requerimientos de la base de datos, el diseño puede ser algo complejo, pero con algunas reglas simples que tengamos en la cabeza será mucho más fácil crear una base de datos perfecta para nuestro siguiente proyecto.

### **Requerimientos para el Diseño de Bases de Datos**

Son muchas las consideraciones a tomar en cuenta al momento de hacer el diseño de la base de datos, quizá las más fuertes sean:

- La velocidad de acceso,
- El tamaño de la DB,
- El tipo de los DATOS,
- Facilidad de acceso a los datos,
- Facilidad para extraer los datos requeridos,
- El comportamiento del manejador de bases de datos con cada tipo de datos.

No obstante que pueden desarrollarse sistemas de procesamiento de archivo e incluso manejadores de bases de datos basándose en la experiencia del equipo de desarrollo de software logrando resultados altamente aceptables, siempre es recomendable la utilización de determinados estándares de diseño que garantizan el nivel de eficiencia mas alto en lo que se refiere a almacenamiento y recuperación de los datos.

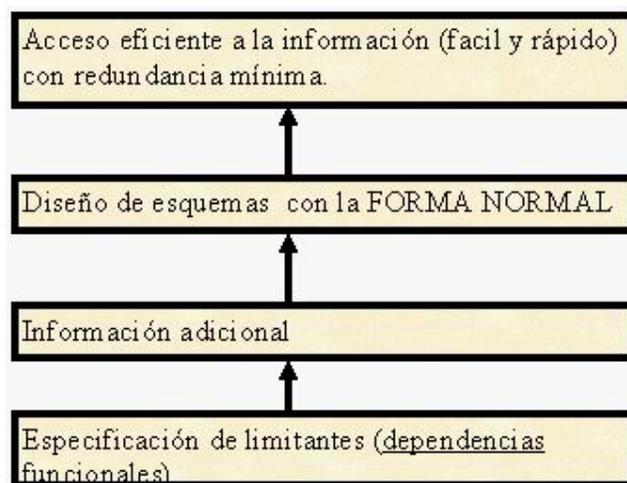
De igual manera se obtiene modelos que optimizan el aprovechamiento secundario y la sencillez y flexibilidad en las consultas que pueden proporcionarse al usuario.

Un modelo de datos es un conjunto de conceptos utilizados para organizar los datos de interés y describir su estructura en forma comprensible para un sistema informático.

Cada modelo de datos provee mecanismos de estructuración, que permiten definir nuevos tipos de datos a partir de tipos elementales predefinidos.

## OBJETIVOS DEL DISEÑO DE BASES DE DATOS

Entre las metas más importantes que se persiguen al diseñar un modelo de bases de datos, se encuentran las siguientes que pueden observarse en esta figura.



## Almacenar Solo DATOS Necesarios.

A menudo pensamos en todo lo que quisiéramos que estuviera almacenado en una base de datos y diseñamos la base de datos para guardar dichos datos. Debemos de ser realistas acerca de nuestras necesidades y decidir qué dato es realmente necesario.

Frecuentemente podemos generar algunos datos sobre la marcha sin tener que almacenarlos en una tabla de una base de datos. En estos casos también tiene sentido hacer esto desde el punto de vista del desarrollo de la aplicación.

Un modelo de datos es básicamente una "descripción" de algo conocido como *contenedor de datos* (algo en donde se guarda los datos), así como de los métodos para almacenar y recuperar dato 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.

El diseño de bases de datos es el proceso por el que se determina la organización de una base de datos, incluidos su estructura, contenido y las aplicaciones que se han de desarrollar. Durante mucho tiempo, el diseño de bases de datos fue considerado una tarea para expertos: más un arte que una ciencia. Sin embargo, se ha progresado mucho en el diseño de bases de datos y éste se considera ahora una disciplina estable, con métodos y técnicas propios. Debido a la creciente aceptación de las bases de datos por parte de la industria y el gobierno en el plano comercial, y a una variedad de aplicaciones científicas y técnicas, el diseño de bases de datos desempeña un papel central en el empleo de los recursos de datos en la mayoría de las organizaciones. El diseño de bases de datos ha pasado a constituir parte de la formación general de los informáticos, en el mismo nivel que la capacidad de construir algoritmos usando un lenguaje de programación convencional.

Las últimas dos décadas se han caracterizado por un fuerte crecimiento en el número e importancia de las aplicaciones de bases de datos. Las bases de datos son componentes esenciales de los sistemas de datos, usadas rutinariamente en todos los computadores [...]. El diseño de bases de datos se ha convertido en una actividad popular, desarrollada no sólo por profesionales sino también por no especialistas.

A finales de la década de 1960, cuando las bases de datos entraron por primera vez en el mercado del software, los diseñadores de bases de datos actuaban como artesanos, con herramientas muy primitivas: diagramas de bloques y estructuras de registros eran los formatos

comunes para las especificaciones, y el diseño de bases de datos se confundía frecuentemente con la implantación de las bases de datos.

Esta situación ahora ha cambiado: los métodos y modelos de diseño de bases de datos han evolucionado paralelamente con el progreso de la tecnología en los sistemas de bases de datos. Se ha entrado en la era de los sistemas relacionales de bases de datos, que ofrecen poderosos lenguajes de consulta, herramientas para el desarrollo de aplicaciones e interfaces amables con los usuarios. La tecnología de bases de datos cuenta ya con un marco teórico, que incluye la teoría relacional de datos, procesamiento y optimización de consultas, control de concurrencia, gestión de transacciones y recuperación, etc.

Según ha avanzado la tecnología de bases de datos, así se han desarrollado las metodologías y técnicas de diseño. Se ha alcanzado un consenso, por ejemplo, sobre la descomposición del proceso de diseño en fases, sobre los principales objetivos de cada fase y sobre las técnicas para conseguir estos objetivos.

Desafortunadamente, las metodologías de diseño de bases de datos no son muy populares; la mayoría de las organizaciones y de los diseñadores individuales confía muy poco en las metodologías para llevar a cabo el diseño y esto se considera, con frecuencia, una de las principales causas de fracaso en el desarrollo de los sistemas de datos.

Debido a la falta de enfoques estructurados para el diseño de bases de datos, a menudo se subestiman el tiempo o los recursos necesarios para un proyecto de bases de datos, las bases de datos son inadecuadas o ineficientes en relación a las demandas de la aplicación, la documentación es limitada y el mantenimiento es difícil.

Muchos de estos problemas se deben a la falta de una claridad que permita entender la naturaleza exacta de los datos, a un nivel conceptual y abstracto. En muchos casos, los datos se describen desde el comienzo del proyecto en términos de las estructuras finales de almacenamiento; no se da peso a un entendimiento de las propiedades estructurales de los datos que sea independiente de los detalles de la realización.

Ventajas del uso de la base de datos en la organización:

1. Independencia de datos y tratamiento.  
Cambio en datos no implica cambio en programas y viceversa (Menor coste de mantenimiento).
2. Coherencia de resultados.
  - Reduce redundancia :
  - Acciones lógicamente únicas.

- Se evita inconsistencia.
3. Mejora en la disponibilidad de datos
    - No hay dueño de datos (No igual a ser públicos).
    - Ni aplicaciones ni usuarios.
    - Guardamos descripción (Idea de catálogos).
  4. Cumplimiento de ciertas normas.
    - Restricciones de seguridad.
    - Accesos (Usuarios a datos).
    - Operaciones (Operaciones sobre datos).
  5. Otras ventajas:  
Más eficiente gestión de almacenamiento.  
Efecto sinérgico.

### **Metodología de diseño de bases de datos**

El diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente, utilizando técnicas específicas.

Así, el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico.

El diseño conceptual parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la base de datos. Un *esquema conceptual* es una descripción de alto nivel de la estructura de la base de datos, independientemente del DBMS que se vaya a utilizar para manipularla. Un *modelo conceptual* es un lenguaje que se utiliza para describir esquemas conceptuales. El objetivo del diseño conceptual es describir los datos de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar estos datos.

El diseño lógico parte del esquema conceptual y da como resultado un esquema lógico. Un *esquema lógico* es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de DBMS. Un *modelo lógico* es un lenguaje usado para especificar esquemas lógicos (modelo relacional, modelo de red, etc.). El diseño lógico depende del tipo de DBMS que se vaya a utilizar, no depende del producto concreto.

El diseño físico parte del esquema lógico y da como resultado un esquema físico. Un *esquema físico* es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Por ello, el diseño físico depende del DBMS

concreto y el esquema físico se expresa mediante su lenguaje de definición de datos.

## Modelos de datos

Un *modelo de datos* es entonces una serie de conceptos que puede utilizarse para describir un conjunto de datos y las operaciones para manipularlos. Hay dos tipos de modelos de datos: los modelos conceptuales y los modelos lógicos. Los *modelos conceptuales* se utilizan para representar la realidad a un alto nivel de abstracción.

Mediante los modelos conceptuales se puede construir una descripción de la realidad fácil de entender. En los *modelos lógicos*, las descripciones de los datos tienen una correspondencia sencilla con la estructura física de la base de datos.

En el diseño de bases de datos se usan primero los modelos conceptuales para lograr una descripción de alto nivel de la realidad, y luego se transforma el esquema conceptual en un esquema lógico. El motivo de realizar estas dos etapas es la dificultad de abstraer la estructura de una base de datos que presente cierta complejidad. Un *esquema* es un conjunto de representaciones lingüísticas o gráficas que describen la estructura de los datos de interés.

Los modelos conceptuales deben ser buenas herramientas para representar la realidad, por lo que deben poseer las siguientes cualidades:

- *Expresividad*: deben tener suficientes conceptos para expresar perfectamente la realidad.
- *Simplicidad*: deben ser simples para que los esquemas sean fáciles de entender.
- *Minimalidad*: cada concepto debe tener un significado distinto.
- *Formalidad*: todos los conceptos deben tener una interpretación única, precisa y bien definida.

En general, un modelo no es capaz de expresar todas las propiedades de una realidad determinada, por lo que hay que añadir aserciones que complementen el esquema.

Algunos modelos con frecuencia utilizados en las bases de datos son:

### Bases de datos jerárquicas

Éstas son bases de datos que, como su nombre indica, almacenan sus datos 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 datos puede tener varios *hijos*. El nodo que no tiene padres es llamado *raíz*, y a los nodos que no tienen hijos se los conoce como *hojas*.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de datos y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

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

### **Base de datos de red**

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de *nodo*: se permite que un mismo nodo tenga varios padres (posibilidad no permitida 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 los datos en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

### **Base de datos relacional**

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos 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 ésta 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 está 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 esporádico de la base de datos. Los datos pueden ser recuperados o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar los datos.

El lenguaje más habitual 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.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

Durante los años '80 (1980-1989) la aparición de dBASE produjo una revolución en los lenguajes de programación y sistemas de administración de datos. Aunque nunca debe olvidarse que dBase no utilizaba SQL como lenguaje base para su gestión.

### **Bases de datos multidimensionales**

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de **Cubos OLAP**. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos multidimensional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.

### **Bases de datos orientadas a objetos**

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 del paradigma de objetos:

- Encapsulación - Propiedad que permite ocultar los datos al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación

de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

Se está trabajando en **SQL3**, que es el estándar de SQL92 ampliado, que soportará los nuevos conceptos orientados a objetos y mantendrá compatibilidad con SQL92.

### **Bases de datos documentales**

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Tesouro es un sistema de índices optimizado para este tipo de bases de datos.

### **Base de datos deductivas**

Un sistema de **base de datos deductivas**, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas **base de datos lógica**, a raíz de que se basan en lógica matemática.

### **Bases de datos distribuida**

La base de datos está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así a distintas universidades, sucursales de tiendas, etcétera.

Resumiendo:

Un modelo de datos para las bases de datos es una colección de conceptos que se emplean para describir la estructura de una base de datos. Esa colección de conceptos incluyen: entidades, atributos y relaciones.

Las bases de datos almacenan datos, permitiendo manipularlos fácilmente y mostrarlos de diversas formas.

El proceso de construir una base de datos es llamado diseño de base de datos.

La mayoría de los modelos de datos poseen un conjunto de operaciones básicas para especificar consultas y actualizaciones de la base de datos,

donde los modelos de datos pueden clasificarse en:

\* Modelos de datos de alto nivel o conceptuales: disponen de conceptos cercanos a la forma en que los usuarios finales perciben una base de datos.

\* Modelos de datos de bajo nivel o físicos: disponen de conceptos que describen detalles sobre el almacenamiento de los datos en la computadora.

\* Modelos de datos de representación (o de implementación): disponen de conceptos que pueden entender los usuarios finales, pero que no están alejados de la forma en que se almacenan los datos en la computadora.

Los modelos de datos sirven para clasificar los distintos tipos de SGBD.

Los modelos más conocidos y utilizados son:

- \* Modelo entidad-relación
- \* Modelo jerárquico
- \* Modelo de red
- \* Modelo relacional

## **2.1 Entidad – relación**

Denominado por sus siglas como: E-R; Este modelo representa a la realidad a través de **entidades**, que son objetos que existen y que se distinguen de otros por sus características, por ejemplo: un alumno se distingue de otro por sus características particulares como lo es el nombre, o el número de control asignado al entrar a una institución educativa, así mismo, un empleado, una materia, etc. Las entidades pueden ser de dos tipos:

- ◆ Tangibles:  
Son todos aquellos objetos físicos que podemos ver, tocar o sentir.
- ◆ Intangibles:  
Todos aquellos eventos u objetos conceptuales que no podemos ver, aun sabiendo que existen, por ejemplo: la entidad materia, sabemos que existe, sin embargo, no la podemos visualizar o tocar.

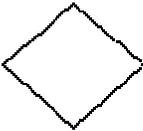
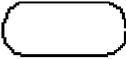
Las características de las entidades en base de datos se llaman **atributos**, por ejemplo el nombre, dirección teléfono, grado, grupo, etc. son atributos de la entidad alumno; Clave, número de seguro social, departamento, etc., son atributos de la entidad empleado. A su vez una entidad se puede asociar o relacionar con más entidades a través de **relaciones**.

Pero para entender mejor esto, veamos un ejemplo:

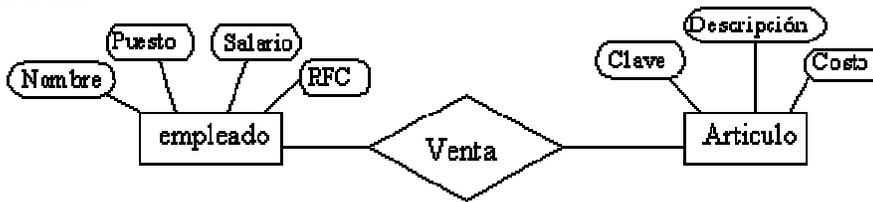
Consideremos una empresa que requiere controlar a los vendedores y las ventas que ellos realizan; de este problema determinamos que los objetos o entidades principales a estudiar son el empleado (vendedor) y el artículo (que es el producto en venta), y las características que los identifican son:

|                  |                  |
|------------------|------------------|
| <i>Empleado:</i> | <i>Artículo:</i> |
| Nombre           | Descripción      |
| Puesto           | Costo            |
| Salario          | Clave            |
| R.F.C.           |                  |

La relación entre ambas entidades la podemos establecer como Venta. Bueno, ahora nos falta describir como se representa un modelo E-R gráficamente, la representación es muy sencilla, se emplean símbolos, los cuales son:

| <b>Símbolo</b>  | <b>Representa</b> |
|---|-------------------|
|  | Entidad           |
|  | Relación          |
|  | Atributos         |
|  | Ligas             |

Así nuestro ejemplo anterior quedaría representado de la siguiente forma:



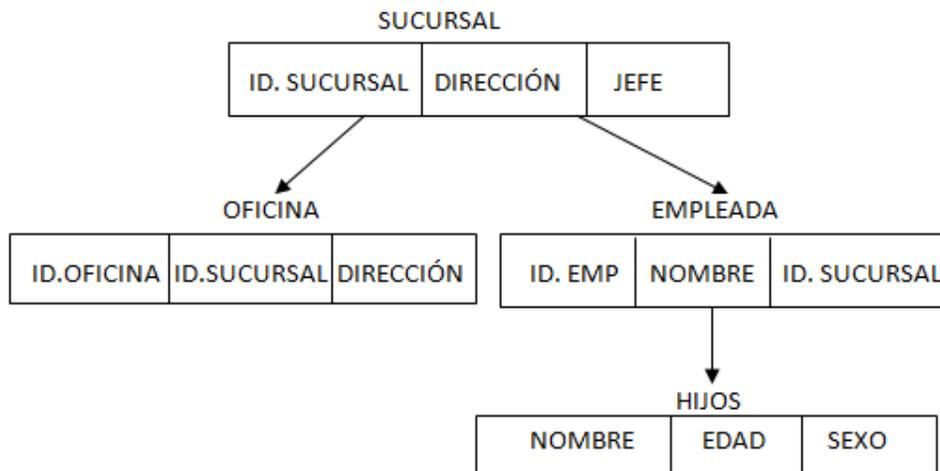
Existen más aspectos a considerar con respecto a los modelos entidad relación, estos serán considerados en el tema Modelo Entidad Relación.

## 2.2 Jerárquico

Este modelo utiliza árboles para la representación lógica de los datos.

Este árbol esta compuesto de unos elementos llamados nodos. El nivel más alto del árbol se denomina raíz. Cada nodo representa un registro con sus correspondientes campos.

La representación gráfica de este modelo se realiza mediante la creación de un árbol invertido, los diferentes niveles quedan unidos mediante relaciones.



En este modelo solo se pueden representar relaciones 1:M, por lo que presenta varios inconvenientes:

- No se admiten relaciones N:M
- Un segmento hijo no puede tener más de un padre.

- No se permiten más de una relación entre dos segmentos.
- Para acceder a cualquier segmento es necesario comenzar por el segmento raíz
- El árbol se debe de recorrer en el orden designado.

### 2.3 De red

En este modelo las entidades se representan como nodos y sus relaciones son las líneas que los unen. En esta estructura cualquier componente puede relacionarse con cualquier otro.

A diferencia del modelo jerárquico, en este modelo, un hijo puede tener varios padres.

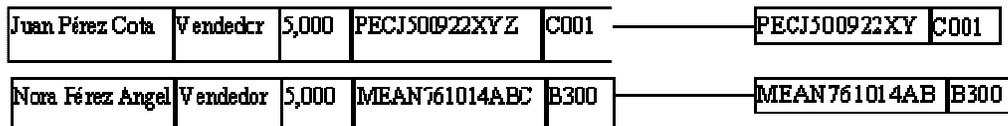
Los conceptos básicos en el modelo en red son:

- El tipo de registro, que representa un nodo.
- Elemento, que es un campo de datos.
- Agregado de datos, que define un conjunto de datos con nombre.

Este modelo de datos permite representar relaciones N:M

Aquí se representa los datos mediante colecciones de registros y sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros. Los registros se organizan en un conjunto de gráficas arbitrarias.

Ejemplo:



## 2.4 Relacional

Este modelo es el más utilizado actualmente ya que utiliza tablas bidimensionales para la representación lógica de los datos y sus relaciones.

Algunas de sus principales características son:

- Puede ser entendido y usado por cualquier usuario.
- Permite ampliar el esquema conceptual sin modificar las aplicaciones de gestión.
- Los usuarios no necesitan saber donde se encuentran los datos físicamente.

El elemento principal de este modelo es la relación que se representa mediante una tabla.

En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a los cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla;

Considerando nuestro ejemplo del empleado y el artículo:

Tabla del empleado

\* Cada una las columnas representa a los atributos de la entidad empleado

Tabla de empleado

| Nombre            | Puesto   | Salario | R.F.C         |
|-------------------|----------|---------|---------------|
| Juan Pérez Cota   | Vendedor | 5,000   | PECJ500922XYZ |
| Nora Méndez Angel | Vendedor | 5,000   | MEAN761014ABC |

\* Registros que contienen la información de la entidad empleado

Existen más aspectos a considerar con respecto a los modelos entidad relación, estos serán considerados en el tema de Modelo Relacional.