



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
BASES DE DATOS

Tema 2:
Diseño de Bases de Datos
(Diseño Conceptual)

Profa. Mercy Ospina T.
Caracas, 03 de septiembre de 2009
Centro de Investigación en Sistemas de Información CISI.

INDICE

CONTENIDO

INTRODUCCIÓN.....	3
DISEÑO DE BASE DE DATOS.....	4
1. Fases del Diseño de Base de Datos.....	4
2. Modelo de Datos.....	5
2.1. Modelo de datos basico (M).....	6
Generación de las estructuras:.....	6
Operaciones:.....	10
2.2. Modelo Entidad/Relación.....	14
Conceptos:.....	14
Diagrama Entidad – Relación.....	17
Entidad Débil.....	17
Modelo E/R Extendido [1].....	19
BIBLIOGRAFÍA.....	22

INTRODUCCIÓN

Una vez que manejamos los conceptos básicos de BD y conocemos las ventajas de utilizar un SDBD es necesario saber diseñar los esquemas que representan los datos de la organización, para poder sacar provecho de estas ventajas y que la BD sea un soporte adecuado para los datos de la organización para la cual ha sido diseñada.

Este documento se ha realizado con el objetivo de servir de guía a los estudiantes de la cátedra Base de Datos, y se tomo como base el segundo capítulo de la publicación de docencia ND 2001- 02 de las Profesoras Concettina Di Vasta* y Rossana Díaz, realizada en Abril de 2001 (Lecturas en Ciencias de la Computación ISSN 1316-6239), la cual fue actualizada con las últimas ediciones de la bibliografía.

En esta guía se define el diseño de base de datos en fases, y se estudian varios modelos de datos, referidos a la fase conceptual, de manera que los estudiantes comiencen a tener conocimientos sobre cómo se estructura una base de datos y que herramientas puede utilizar para obtener una estructura bien diseñada.

DISEÑO DE BASE DE DATOS.

Hasta ahora hemos dado por supuesto que los datos de la base de datos tienen una estructura, por ejemplo los empleados tienen nombre, apellido, dirección, entre otros. Pero ¿cómo obtenemos esta estructura? La respuesta es muy simple, la estructura de la base de datos se determina durante el **diseño de la base de datos** sin embargo esta tarea puede ser muy compleja.

Para producir un sistema que satisfaga a toda la organización se debe usar un enfoque distinto al usado en el procesamiento de archivos donde se dirigía el trabajo hacia un departamento y las aplicaciones que necesitaba. Ahora se debe pensar primero en los datos y luego en las aplicaciones, por lo que una base de datos mal diseñada puede llevar a errores que pueden llevar a decisiones incorrectas.

1. FASES DEL DISEÑO DE BASE DE DATOS

El proceso de diseño está compuesto por una serie de fases que servirán como guía al diseñador acerca de las técnicas apropiadas en cada una.

a. Diseño conceptual de la base de datos:

Es la primera fase del diseño de BD e implica la creación de un modelo de datos conceptual (donde se reflejen los conceptos) para aquellos procesos de la empresa que se desean modelar. Este modelo se construye usando la información documentada en la especificación de requisitos de los usuarios, y es “independiente” de los detalles de implementación tales como el tipo de SMBD, los programas de aplicación, los lenguajes de programación o la plataforma de Hardware, es decir, en esta fase no importa como o donde se va a implementar la base de datos. Este modelo se valida de acuerdo a los requerimientos de los usuarios y es una fuente de información para la siguiente fase.

b. Diseño lógico de la base de datos:

En esta segunda fase, el diseñador traduce el modelo conceptual a un modelo lógico, donde se toma en cuenta la implementación del SMBD (si es relacional, orientado a objetos, entre otros), pero ninguna otra consideración física, igualmente se valida con los requerimientos de los usuarios y se verifica que no tenga problemas de redundancia.

c. Diseño físico de la base de datos.

En esta fase el diseñador decide como implementar la base de datos, por lo tanto se debe tener claro cuál es el SMBD a usar, ya que el diseño físico estará adaptado a este, en esta fase se implementan las estructuras de datos, sus relaciones, sus restricciones y las medidas de seguridad necesarias para resguardar los datos.

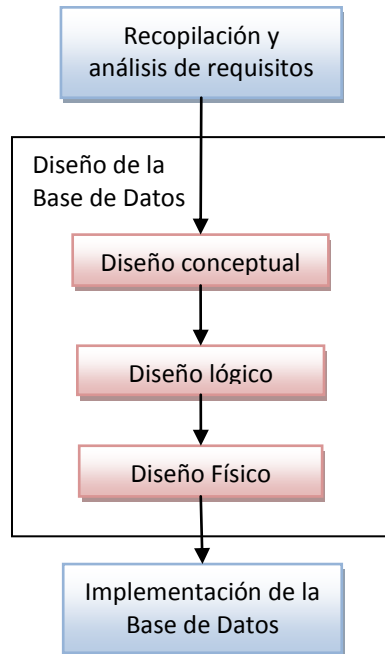


FIGURA 1: FASES DE DISEÑO DE UNA BASE DE DATOS

2. MODELO DE DATOS

Una característica fundamental del enfoque de bases de datos es que proporciona cierto nivel de abstracción de los datos al ocultar detalles de almacenamiento que la mayoría de los usuarios no necesitan conocer.

Los modelos de datos son el principal instrumento para ofrecer dicha abstracción. Un modelo de datos es un conjunto de conceptos que pueden servir para describir la estructura de la base de datos. Es decir, un Modelo de Datos no es más que una colección de herramientas conceptuales que se utilizan para describir los datos, las relaciones existentes entre ellos, la semántica asociada a los mismos y las restricciones de consistencia.

En este curso vamos a estudiar dos tipos de modelos de datos de la fase conceptual y uno de la fase lógica, que corresponden a:

1. Modelo de datos básico
2. Modelo Entidad-Relación
3. Modelo relacional

2.1. MODELO DE DATOS BASICO (M)

Herramienta intelectual / conceptual que prevee una interpretación del mundo real lo suficientemente abstracta para lograr un entendimiento de cómo se relacionan los datos.

Este modelo de datos esta definido como una tupla

$$M = \langle G, O \rangle$$

G: representan las estructuras de datos a modelar, y las restricciones asociadas a estos. Representa las propiedades estáticas de un modelo de datos, es decir, el esquema conceptual.

O: especifica las operaciones que se pueden realizar sobre los datos modelados en las estructuras. Expresa las propiedades dinámicas de la base de datos, es decir, su instancia.

GENERACIÓN DE LAS ESTRUCTURAS:

Categorías: Representan conjuntos de entes u objetos con características comunes y para generarlas se usaran los conceptos asociados a los procesos de abstracción.

Agregación: Proceso de abstracción mediante el cual se construye una categoría en función de las partes que la componen, las cuales se denominarán atributos. Contrario particularización

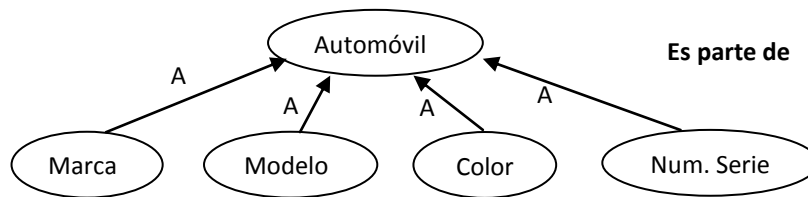


FIGURA 2 AGREGACIÓN EN CATEGORÍAS

Generalización: Proceso de abstracción que define una categoría general en función de otras categorías con características comunes. Lo contrario Especialización.

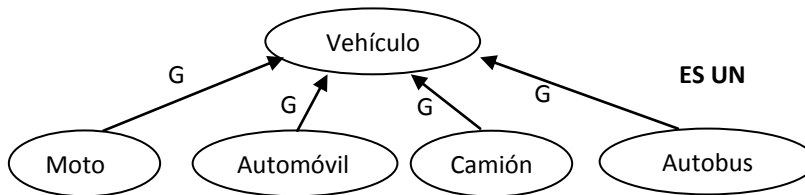


FIGURA 3 GENERALIZACIÓN EN CATEGORÍAS

Clasificación: define una categoría en términos de ejemplos de instancias asociadas a los atributos de dicha categoría. Lo contrario Instanciación

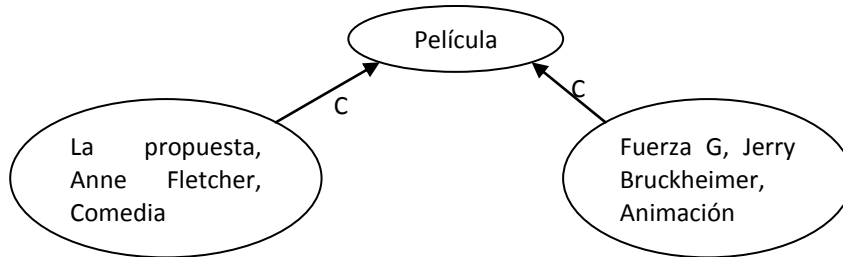


FIGURA 4 CLASIFICACIÓN EN CATEGORÍAS

Se ha definido como estándar que los nombres de las categorías sean en singular, ya que representan a cada uno de los elementos o instancias que pueden tomar los atributos.

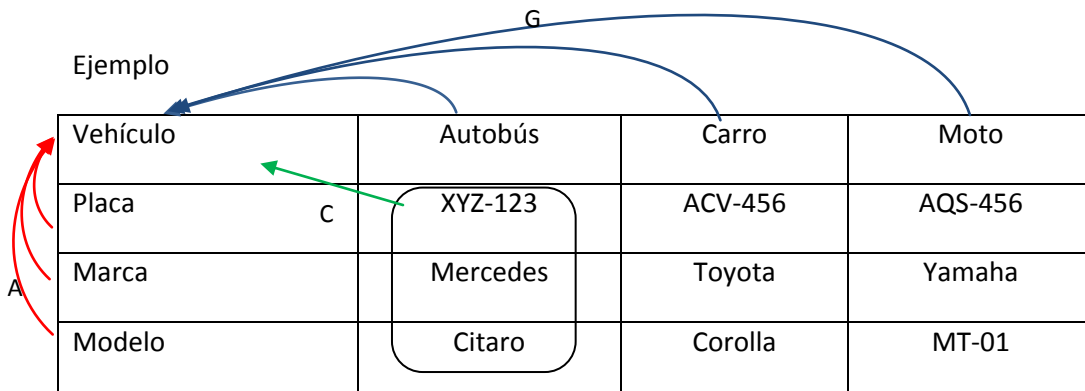


FIGURA 5 EJEMPLO DE GENERALIZACIÓN, AGREGACIÓN Y CLASIFICACIÓN

Otros conceptos:

Clave: conjunto de atributos (uno o más) que identifican de manera única (univoca) cada elemento de una categoría. Se denota subrayando, o colocando un asterisco, en el o los, atributos clave

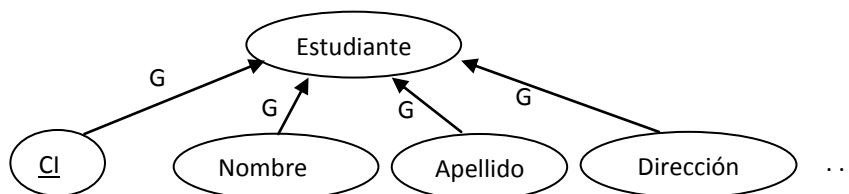
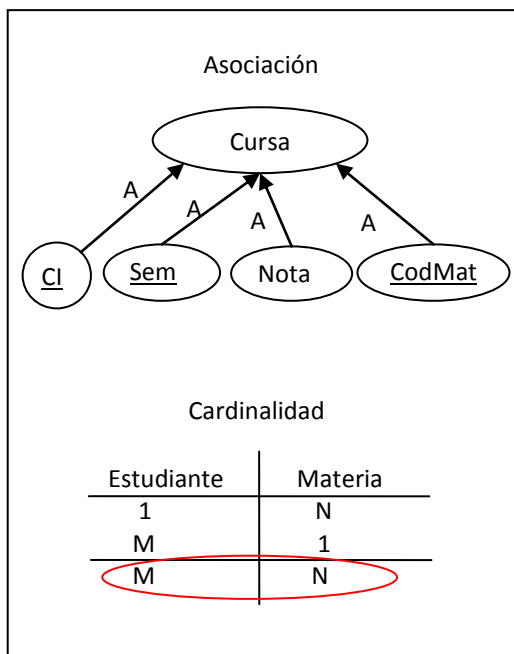
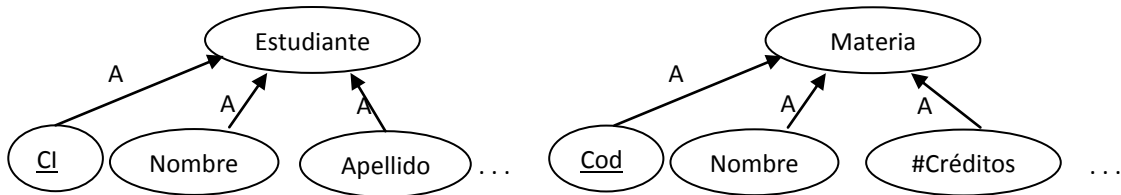


FIGURA 6 CLAVE EN UNA CATEGORÍA

Asociación: Relación que se establece entre 2 o más categorías, con el fin de definir correspondencias entre estas, por ejemplo entre Estudiante y Materia se puede definir una asociación cursa, debido a que un estudiante cursa materias y una materia es cursada por estudiantes.



Toda asociación tiene como atributos los atributos claves de las categorías que está asociando, aunque no necesariamente deben llevar el mismo nombre. *Estos atributos permiten la navegación entre categorías.*

Además podrá tener atributos propios que sean necesarios para identificar los elementos de la asociación de manera unívoca (claves) y atributos adicionales

Cardinalidad

La asociación tiene una Cardinalidad que representa cuantas instancias o elementos de una categoría, están relacionados con elementos de la(s) otra(s) categoría(s)

FIGURA 7 ASOCIACION ENTRE ESTUDIANTE Y MATERIA

En la figura 7 podemos calcular la cardinalidad preguntándonos, cuantas materias cursa **un** estudiante, y cuantos estudiantes cursan **una** materia, la respuesta a ambas preguntas representan la cardinalidad de la asociación.

Las cardinalidades como se verá en detalle más adelante pueden ser de uno a uno (1:1), de uno a muchos (1:n) y de muchos a muchos (n:m) como en el ejemplo anterior.

Restricciones:

Conjunto de reglas que se aplican a los distintos componentes de un modelo de datos para garantizar la semántica¹ e integridad de los datos.

Son lineamientos, reglas, limitaciones o acotaciones que norman o restringen las características de la situación a modelar, determinando los valores válidos de los elementos representados por el modelo (Base de Datos).

Tipos de restricciones

- *Restricciones Inherentes:* Nacen con la estructura, son parte integral de las estructuras del modelo de datos.
- *Restricciones explícitas:* Provee un mecanismo flexible para aumentar la especificación de la estructura de la BD.
- *Restricciones de dominio:* establecen que valores puede tomar un atributo.

Dominio: Conjunto de valores posibles que puede tomar un atributo, podemos tener dominios conocidos, como los Enteros, String, Fecha, o conjuntos definidos por el usuario como (A .. Z). Se establece una relación de generalización entre los atributos y el dominio.

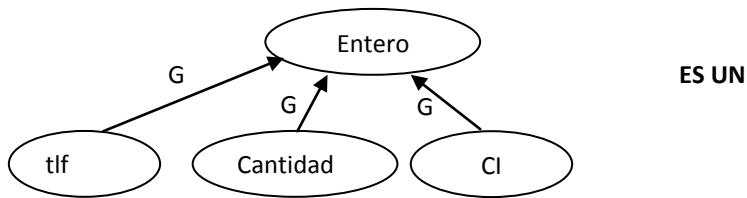


FIGURA 8 EJEMPLO DE RESTRICCIÓN DE DOMINIO

- *Restricciones de Cardinalidad:* Son aquellas establecidas en asociaciones

Cardinalidad: Sean dos conjuntos A y B y R una asociación entre ellos, la Cardinalidad se refiere al número de objetos de A que pueden estar relacionados con B y viceversa.

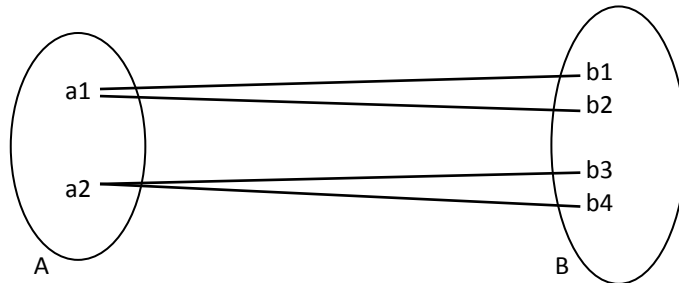


FIGURA 9 CARDINALIDAD ENTRE CONJUNTOS

¹ El término **semántica** se refiere a los aspectos del significado, sentido o interpretación del significado de un determinado elemento, símbolo, palabra, expresión o representación formal.

OPERACIONES:

Se definen como un conjunto de acciones que consultan o producen un cambio de estado en la base de datos

Patrón de una operación:

Selección: Especifica la parte de la Base de datos donde se va a aplicar la acción

Acción: especifica que es lo que se va a hacer, que pueden ser: Recuperación, Inserción, Eliminación, actualización, navegación.

Especificación de una operación:

Nombre: Un identificador de la operación en lenguaje natural.

Entrada: Porción de la base de datos (categorías y asociaciones), y valores externos necesarios para llevar a cabo la operación

Salida: Datos o modificaciones a los datos que se tendrán como resultado.

Acción: Proceso que se llevara a cabo para, a partir de la entrada, obtener la salida, descrito en lenguaje natural.

Restricciones: Reglas o limitaciones que deben cumplir las operaciones.

Ejemplos: Dado un modelo de datos donde existen las categorías Estudiante y Materia, y la asociación cursa, se pueden definir las siguientes operaciones:

1. Listado de los estudiantes que cursan Base de Datos en el semestre 02-2009.
2. Cantidad de estudiantes que reprobaron la materia con código 6306
3. Modificar la nota del estudiante CI 123456 en la materia 8456 en la fecha 02-2009 a 12

Operación 1:

Nombre: Estudiantes de BD 2-2009

Entradas: variables externas: Nombre_materia = 'Base de Datos', Semestre = '2_2009', categorías: Estudiante y Materia, Asociaciones: Cursa. #todas las estructuras necesarias para hacer la consulta.

Salida: Listado de los datos de los Estudiantes que cumplan con los requisitos que cursan base de datos en el semestre 2-2009

Accion:

- En la categoría Materia se consulta la materia cuyo nombre sea igual a la variable externa y se toma el código, si no existe la materia se da un mensaje de error y se detiene la ejecución
- En la Asociación Cursa se buscan los registros cuyo codMat sea igual al código anterior, y el Sem sea igual a la variable externa semestre, de no existir el registro se envía un mensaje de error y se detiene la ejecución.
- Por cada CI obtenida en el paso anterior, se navega a la categoría estudiante se obtienen los demás atributos y se muestran.

Restricciones: Que existan el nombre de la materia y el semestre indicados en las variables de entrada.

Operación 3:

Nombre: Modificación nota estudiante

Entrada: variables externas: Cedula del estudiante=1234, código de la materia=6306, nota=12, categoría: ninguna, asociación: Cursa.

Salida: mensaje indicando que la modificación se realizo de manera correcta o no

Accion:

- En la Asociación Cursa se obtiene el registro que coincida con la cedula y el código de materia de las variables externas.
- Si no coincide ningún registro se envía un mensaje de error y se detiene la ejecución
- En la nota del registro se coloca la nota ingresada.
- Se guarda el registro en la base de datos y se envía el mensaje que la modificación se realizo con éxito.

Restricciones:

- Debe existir el estudiante al que pertenece la cédula y el código de materia.
- La nota debe estar entre 0 y 20.

Se deja como ejercicio la operación 2.

Nombre : _____

Entrada: _____

Salida: _____

Acción: _____

Restricciones: _____

Actividades: Realice las siguientes operaciones para la próxima clase:

- Listado de materias del 5to semestre.
- Cantidad de estudiantes que cursaron sistemas operativos en los semestres 1_2009 y 2_2009.
- Promedio de la nota de los estudiantes que cursaron la materia 6104 el semestre pasado.
- Listado de los estudiantes que aprobaron todas las materias que cursaron en el semestre 1_2009

Pasos para realizar un modelo de datos básico (a partir de los requerimientos del usuario)

- 1) Identificar las categorías (nombres)
- 2) Identificar los componentes de las categorías o atributos (agregación)
- 3) Identificar las claves
- 4) Verificar generalización entre categorías.
- 5) Identificar asociaciones entre categorías
- 6) Identificar las restricciones
 - a. Dominio
 - b. Cardinalidad
 - c. Explicitos (adicionales al modelo)
- 7) Identificar y describir las operaciones.

2.2. MODELO ENTIDAD/RELACIÓN

Este modelo se puede considerar un refinamiento del modelo de datos básico, aunque aun se halla en la primera fase del diseño de BD, ya que se considera un modelo semántico, es decir, que le da significado a los datos.

CONCEPTOS:

Entidad: Es un ente u objeto del mundo real que es distinguible de todos los demás objetos, por ejemplo cada persona de una empresa. Una entidad tiene un conjunto de propiedades o atributos que lo distinguen de las demás entidades, Ejemplo: Pedro Pérez con C.I: 10.503.806, es una entidad ya que identifica únicamente a una persona específica. Igualmente, el Número de Cuenta 900 del Banco Venezuela con monto de 20.000 es una entidad.

Un conjunto de entidades es un grupo de entidades del mismo tipo que comparten los mismos atributos, por ejemplo los clientes de un banco. [1]

Relación: asociación entre varias entidades, por ejemplo relación entre el cliente López con el préstamo P-15, un conjunto de relaciones es una relación entre dos o más conjuntos de entidades. Cada relación tiene, por lo menos, los atributos claves de las entidades, además puede tener atributos que den información adicional o que contribuyan a identificar unívocamente cada elemento de la relación.

Por ejemplo la relación venta tendrá como atributos la clave del cliente, la clave del producto, la fecha y la cantidad de venta, donde la clave de la relación será el conjunto (ci, codprod, fecha_venta) ya que identifica cada venta de manera univoca, a menos que un cliente compre el mismo producto en la misma fecha, en más de una ocasión.

Grado de una relación: Cantidad de entidades asociadas a la relación, se recomienda hasta un máximo de 3.

Atributo: elemento de dato que caracteriza a una entidad, para cada atributo hay un conjunto de valores permitidos, por ejemplo el dominio del atributo nombre es el conjunto de cadenas de caracteres (string)

Tipos de atributos [1]:

- *Atributos simples y compuestos:* los atributos simples no están divididos en subpartes, los compuestos pueden dividirse en otros atributos, por ejemplo el atributo *nombre* podría subdividirse en *nombre*, *primer_apellido*, *segundo_apellido* o *dirección* puede referirse a *calle*, *ciudad*, *estado* y *código_postal*. Los atributos compuestos ayudan a agrupar atributos relacionados

- *Atributos monovalorados y multivalorados*: Los monovalorados tienen un único valor para cada entidad, los multivalorados pueden tener un conjunto de valores para dicha entidad, por ejemplo teléfonos.
- *Atributos derivados*: El valor de este atributo se puede obtener a partir de del valor de otros atributos o entidades relacionadas, por ejemplo, el atributo *edad* se puede obtener a partir del atributo *fecha_nacimiento* y la fecha actual. El valor de los atributos derivados no se almacenan sino que se calculan cada vez que hace falta.

Nota: En este curso se utilizarán atributos simples, monovalorados y no derivados.

Restricciones

Claves candidatas: conjunto de atributos que pueden identificar unívocamente una entidad, por ejemplo, la cedula, el conjunto nombre y apellido.

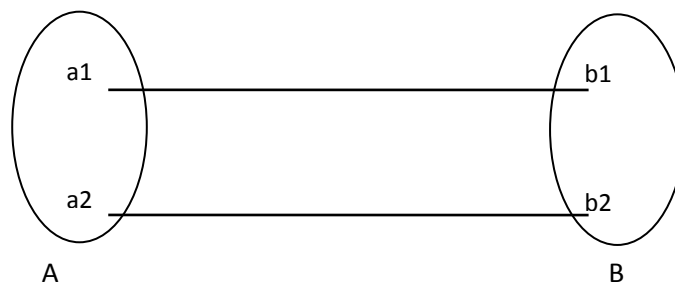
Clave primaria: Conjunto de atributos escogido entra las claves candidatas, para identificar las entidades como únicas, se debe escoger de manera que su valor no cambie, o cambie muy raramente, por ejemplo una dirección no puede ser clave primaria porque puede cambiar con frecuencia.

Cardinalidad: Expresa el número de entidades a las que otra entidad se puede asociar mediante un conjunto de relaciones.

Para un conjunto de relaciones binarias A y B, la correspondencia de cardinalidades puede ser la siguiente:

- Uno a uno (1: 1)

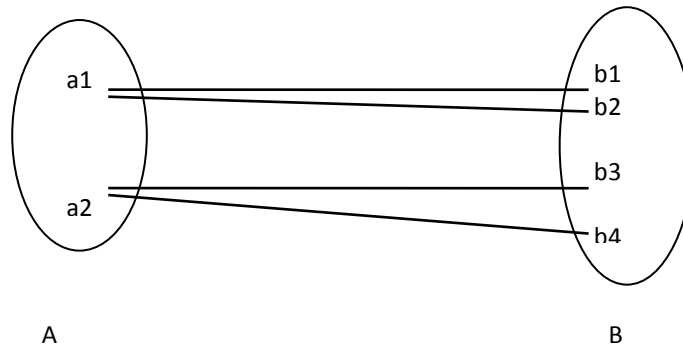
Una entidad en A está relacionada con una entidad en B, y una entidad en B está relacionada con una entidad en A.



Por ejemplo, si se asume que cada persona casada tiene sólo un conyugue, la cardinalidad de la relación *ESTA_CASADO_CON* es 1 en cada dirección. Es una relación 1: 1 (uno a uno).

➤ Uno a muchos (1: M)

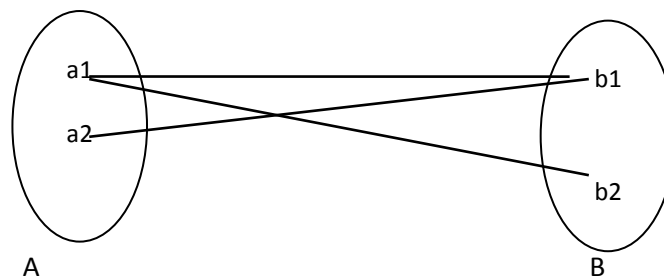
Una entidad en A está relacionada con cualquier número de entidades en B, pero una entidad en B puede asociarse únicamente con una entidad en A.



Por ejemplo, en la relación DORMITORIO-OCUPANTE, una ocurrencia única de Dormitorio se relaciona con muchas ocurrencias de Estudiantes. Si la relación es 1: M (uno a muchos), en un dormitorio hay muchos estudiantes, pero un estudiante sólo tiene un dormitorio.

➤ Muchos a muchos (M:N)

Una entidad en A está relacionada con cualquier número de entidades en B, y una entidad en B está relacionada con cualquier número de entidades en A.



Por ejemplo, en la relación ESTUDIANTE-CLUB, donde se relaciona las ocurrencias de Estudiante con las ocurrencias de Club. Un estudiante puede inscribirse en más de un club, y en un club puede haber como miembros muchos estudiantes. Esta es una relación N: M (muchos a muchos).

DIAGRAMA ENTIDAD – RELACIÓN

Expresa gráficamente la estructura lógica general de la base de datos, usando los conceptos definidos anteriormente.

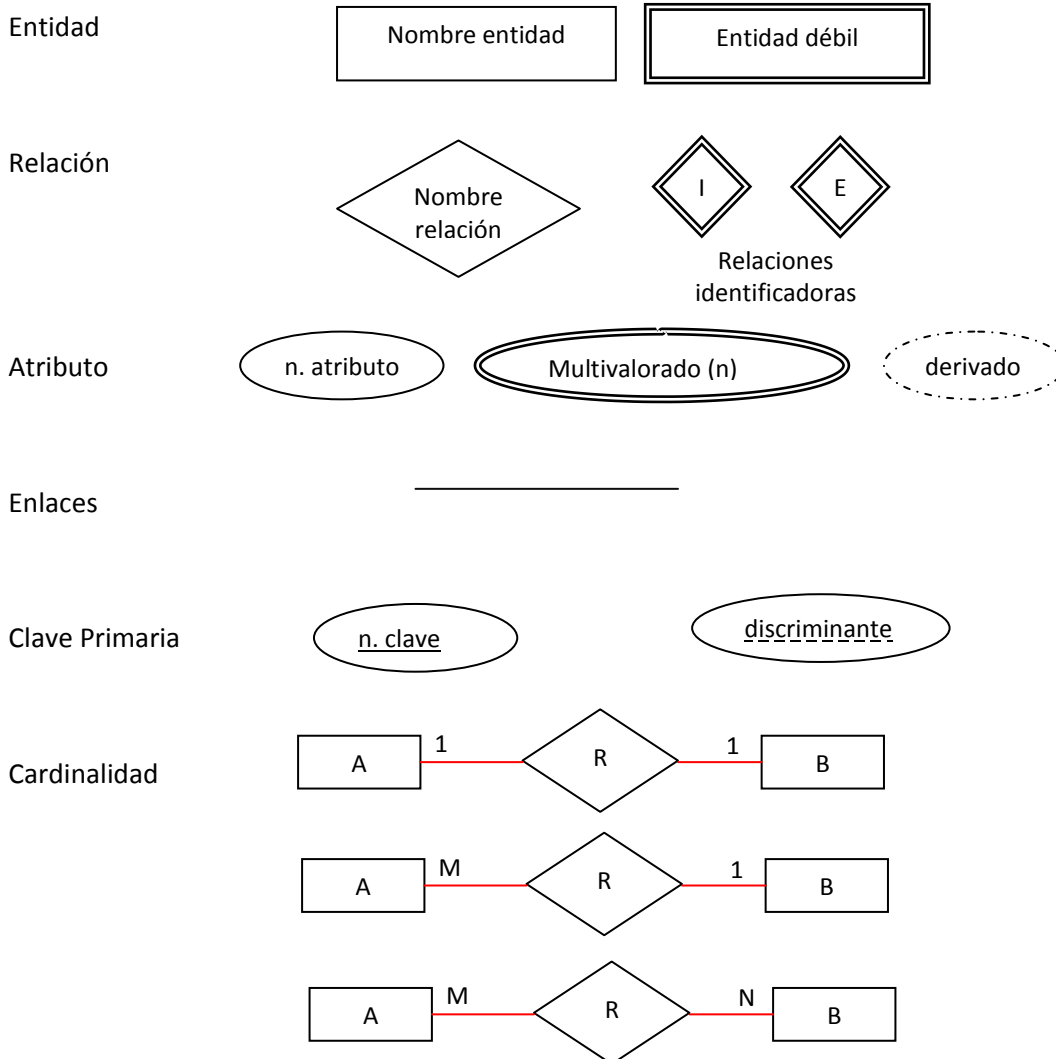


FIGURA 10 SIMBOLOS USADOS EN LA NOTACION E-R

ENTIDAD DÉBIL

Puede que un conjunto de entidades no tenga suficientes atributos para formar una clave primaria, ese conjunto se denomina entidades débiles, y para que tengan sentido deben estar asociadas a otro conjunto de entidades las que se llaman entidades identificadoras o propietarias.

Tipos de dependencia de una entidad débil:

- Por identificación: en conjunto de entidades débiles tiene que ser identificados a través de sus relaciones con otros conjuntos de entidades.

Por ejemplo: Edificio → Apartamento

Libro → Ejemplares

Producto → Versión

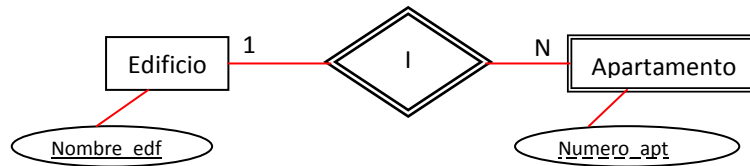


FIGURA 11 DEPENDENCIA POR IDENTIFICACION

- Por existencia: La existencia de la entidad débil depende de la existencia de la entidad propietaria, por lo tanto si se elimina la entidad propietaria se elimina la entidad débil.

Ejemplo: Proyecto → Actividad.

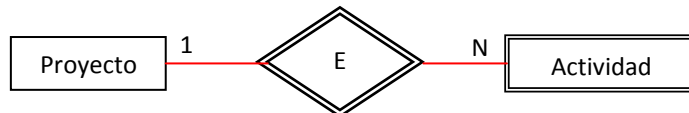


FIGURA 12 DEPENDENCIA POR EXISTENCIA

Las dependencias por identificación lo son también por existencia, sin embargo las dependencias por existencia no son necesariamente por identificación

La relación que asocia el conjunto de entidades débiles con el conjunto de entidades propietarias se denomina relación **identificadora** y su Cardinalidad es de muchos (entidad débil) a uno (entidad propietaria)

Aunque los conjuntos de entidades débiles no tienen clave primaria, se utiliza un conjunto de atributos como **discriminante** para distinguirlos, la clave primaria se forma con la unión del discriminante de las entidades débiles y la clave primaria de las entidades propietarias. Por ejemplo la clave primaria de apartamento sería (nombre_edf, numero_apt)

MODELO E/R EXTENDIDO [1]

Aunque los conceptos básicos del modelo E-R pueden modelar la mayor parte de las características de las bases de datos, algunos aspectos se pueden expresar mejor mediante ciertas extensiones del modelo E-R básico, tales como especialización, generalización, herencia de atributos y agregación.

Especialización

Los conjuntos de entidades pueden tener subgrupos de entidades que se diferencian, por ejemplo en algunos atributos, es el caso del conjunto de entidad *persona* que pudiera clasificarse como *cliente* o *empleado*, ambos tienen atributos comunes a *persona*, pero además otros atributos adicionales, por ejemplo el *cliente* puede tener una cuenta, y el *empleado* un sueldo.

La especialización se puede aplicar repetidamente para refinar el esquema de diseño, por ejemplo los empleados se pudieran clasificar en *supervisor*, *secretaria* y *mensajero*, y representa un diseño descendente.

Generalización

Si procedemos de manera ascendente, podemos ver características comunes en diferentes conjuntos de entidades para crear entidades de nivel superior. Por ejemplo el diseñador pudo haber identificado las entidades *cliente* y *empleado* y luego generalizarlos en una entidad *persona*.

A efectos prácticos la generalización es el inverso de la especialización, por lo que en el diagrama E-R no se distinguirá entre ellos.

Herencia de atributos

Esta es una propiedad de las entidades de nivel superior e inferior creadas mediante la especialización y la generalización, se dice que los atributos de la entidad de nivel superior, son heredados por las entidades de nivel inferior, así por ejemplo, empleado y cliente heredarán los atributos de persona.

Los conjuntos de entidades de nivel inferior (o subclases) también heredan la participación en los conjuntos de relaciones en los que participa la entidad de nivel superior (o superclase). Por ejemplo las entidades *supervisor*, *secretaria* y *mensajero* participaran en el conjunto de relaciones *trabaja_para* en la que participan las entidades *empleado*. La herencia de atributos y participación se aplica a todas las capas de nivel inferior.

Restricción de pertenencia

Una entidad, puede pertenecer a uno o más conjuntos de entidades de nivel superior de la generalización, por lo que los conjuntos de entidades de nivel inferior pueden ser uno de los siguientes tipos:

- Disjuntos: La restricción exige que cada entidad solo pertenezca a un conjunto de entidades de nivel inferior

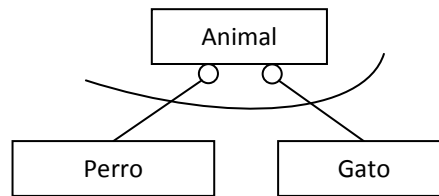


FIGURA 13 PERTENENCIA DISJUNTA

- Solapados: La misma entidad puede pertenecer a más de un conjunto de entidades de nivel inferior.

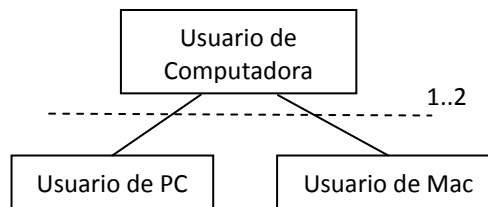


FIGURA 14 PERTENENCIA SOLAPADA

Agregación

Una limitación del modelo E-R es que no es posible expresar relaciones entre las relaciones, para entender mejor este problema consideremos la relación ternaria *trabaja_en* entre *empleado*, *trabajo*, *sucursal*, supongamos que se desea registrar el *director* responsable de las tareas realizadas por cada *empleado* de cada *sucursal*.

Una alternativa para representar esta relación es la siguiente (para simplificar se han omitido los atributos)

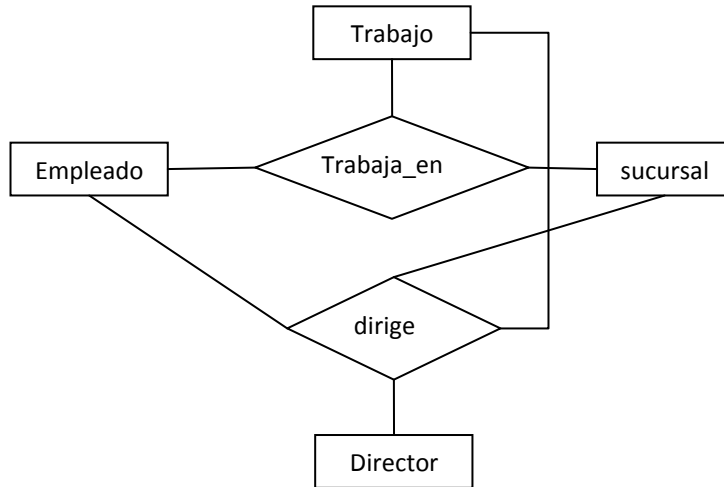


FIGURA 15 DIAGRAMA E-R CON RELACIONES REDUNDANTES

Parece que los conjuntos de relaciones *trabaja_en* y *dirige* se pueden combinar en una sola, sin embargo si esto se hace existirán algunas combinaciones *empleado, sucursal, trabajo* que no tengan director. No obstante hay información redundante en el diagrama obtenido, ya que cada combinación de *empleado, sucursal y trabajo*, en *dirige* también estará en *trabaja_en*.

La mejor forma de modelar la situación ya descrita es usar la agregación, la cual es una abstracción a través de la cual las relaciones se tratan como entidades de nivel superior.

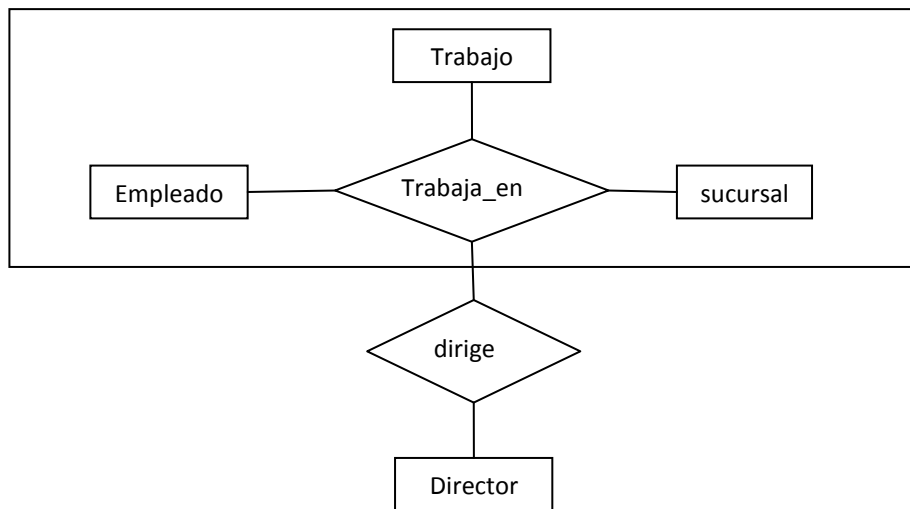


FIGURA 16 DIAGRAMA E-R CON AGREGACION

Así en este ejemplo el conjunto de relaciones *trabaja_en* se considera como entidad de nivel superior, y se trata como cualquier otro conjunto de entidades.

BIBLIOGRAFÍA

[1] Silbershatz A, Korth H. "Fundamentos de bases de datos". Quinta edición, 2006, McGraw-Hill

[2] DATE, C.J. "Introducción a los Sistemas de Bases de Datos". Vol I. Quinta Edición. Addison-Wesley Iberoamericana. 2000.

[3] CONNOLLY T. M, BEGG C. E. "Sistemas de bases de datos". 4ta Edición, Pearson educación S.A., 2005,

Di Vasta C., Díaz R. "Sistemas de Base de datos". ND 2001 -02, Abril 2001.