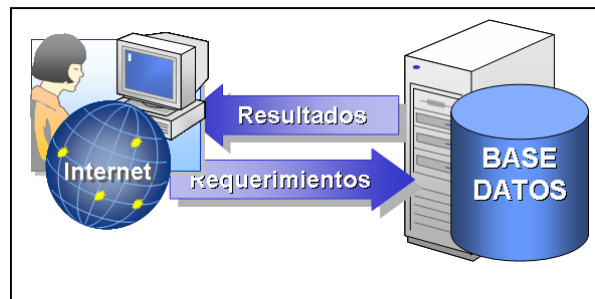


Diseño de Base de Datos

- Introducción al Diseño de Base de Datos.
 - El Diseño Conceptual de Datos
 - El Modelo Entidad-Relación
 - El Diseño Lógico
 - El Proceso de Normalización
-



MCP. Ing. Ricardo Mendoza Rivera

rimenri@hotmail.com

<http://rimenri.blogspot.com>

Introducción al Diseño de Base de Datos

El presente curso tiene como propósito introducir al estudiante en el Diseño e Implementación de Base de Datos como una herramienta de imprescindible ayuda como parte de la construcción de un sistema de información.

Proporcionando al participante un conjunto de técnicas de identificación de requerimientos operacionales, y modelado de bases de datos, pasando por las etapas de diseño conceptual, lógico y físico. Incluyendo el desarrollo de un caso práctico, que incluye desde la identificación de requerimientos hasta la implementación con un Administrador de Base de Datos, en donde se verán conceptos de integridad, concurrencia, optimización de consultas.

Diseño de Base de Datos

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.

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.

¿Que es una Base de Datos?

Es un conjunto de datos almacenados y organizados, que sirven para satisfacer las necesidades de información de una organización. Por ejemplo:

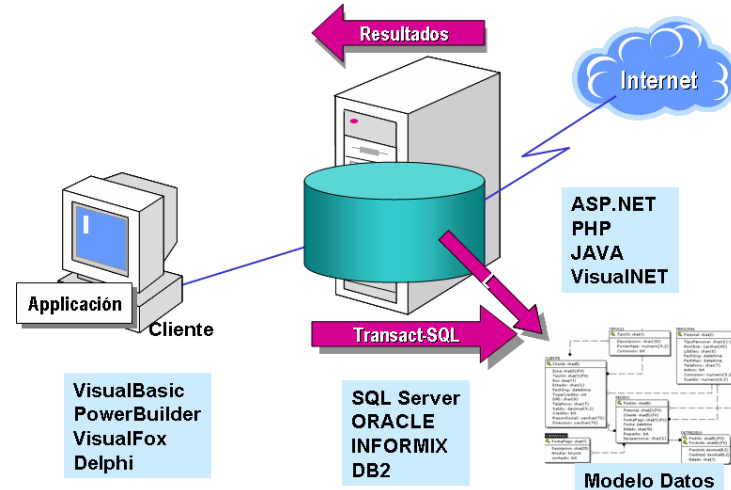
- Datos de un Cliente
- Datos de una Venta
- Datos de una Matricula
- Datos de un Operación (Depósito, Retiro, Transferencias, etc.) de una cuenta en un Banco.
- Compra de un Producto por Internet.



¿En donde se aplica una Base de Datos?

DISEÑO DE BASE DE DATOS

Constituyen una parte fundamental en el funcionamiento de una Sistema de Información, ya que a partir en ella se registran las transacciones de los diferentes procesos que desarrollan una empresa y trabaja al margen de la plataforma de desarrollo (Windows, Web, Móviles, etc)



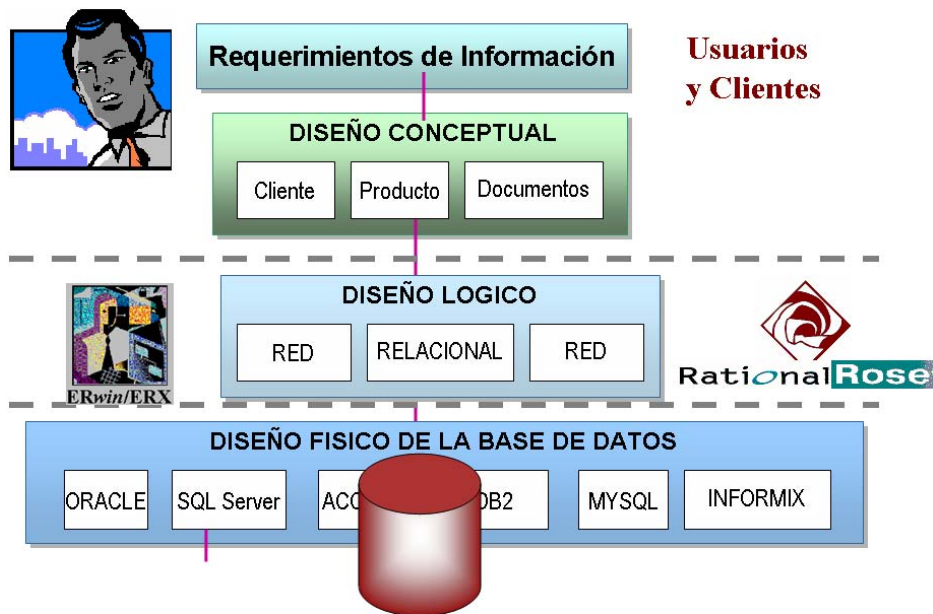
Dentro de la construcción de los sistemas se ubica desde los requerimientos – a partir del cual se construye- y el diseño de prototipos fundamentalmente:



Etapas en el diseño de una Base de Datos

DISEÑO DE BASE DE DATOS

Pasos para el Diseño de una Base de Datos



Cada etapa debe generar un Esquema:



La idea fundamental para llegar al esquema físico es trabajar con modelo de datos.

Modelo de Datos

Los modelos de datos son usados para describir la realidad. El bloque de construcción común a todos los modelos de datos es una pequeña colección de "mecanismos de abstracción" como las siguientes:

DISEÑO DE BASE DE DATOS

- Clasificación
- Agregación
- Generalización

Construcción de un Diseño de una Base de Datos

Estos se elaboran a partir de:

- ✓ Requerimientos
- ✓ Reglas de Negocio

Requerimientos: son necesidades de información que los actores (usuarios) de un sistema necesitan como soporte a las labores que realizan.

Ejemplo:

- Conocer los montos que compraron los clientes de la categoría A1 en Mayo del 2007 de la línea de productos Aceites.
- Identificar los 3 alumnos con mejores notas del 5to grado de secundaria

Reglas de Negocio: condiciones que un proceso de Negocios impone para la realización de sus operaciones.

Ejemplo:

- No dar un nuevo crédito a un cliente que tiene 2 documentos pendientes de pago
- Para registrar su matrícula previamente cancelar su derecho.

Existen diferentes formas de encontrar requerimientos:

- Realizando entrevistas.
- Analizando Documentos.
- Mediante el análisis de casos de uso.

En nuestro caso realizaremos la captura de requerimientos utilizando casos de uso.

A TENER EN CUENTA! Donde encontrar Información para la Base de Datos

Cuando se definen algunos de los requerimientos hay que analizarlo y preguntarse en que momento se “crea” o genera la Información.

DISEÑO DE BASE DE DATOS

Proceso de Ventas

- Controlar Ventas ->
 - Contado
 - Crédito
- Pagar Comisiones Vendedores
 - A partir de las ventas canceladas
- Controlar Clientes
 - Zonas
 - Definir perfil crediticio

Proceso Académico

- Listar Horarios
- Listar Matriculados
 - En que momento se genera la matricula
 - → Registrar Matricula

Proceso Cobranzas

- Conocer clientes morosos
- Reportar documentos vencidos.
 - En que momento se crea el documento
 - → Orden de Compra (compra)
 - → Registra el Documento (venta)
 - En que momento se convierte en vencido

Proceso de Producción de Espárragos

- Quiero conocer los productos elaborados en Mayo del 2007 en la Planta 2
 - Registro los productos elaborados
- Necesito conocer los insumos utilizados en Junio del 2007
 - Nota de Salida de Insumos.

Control de Asistencia

- Quiero conocer los 10 docentes con más tardanza.
 - Registro de la asistencia- marcación (Hora Entrada, Fecha Salida, Código del Docente)

Diseño Conceptual de Base de Datos

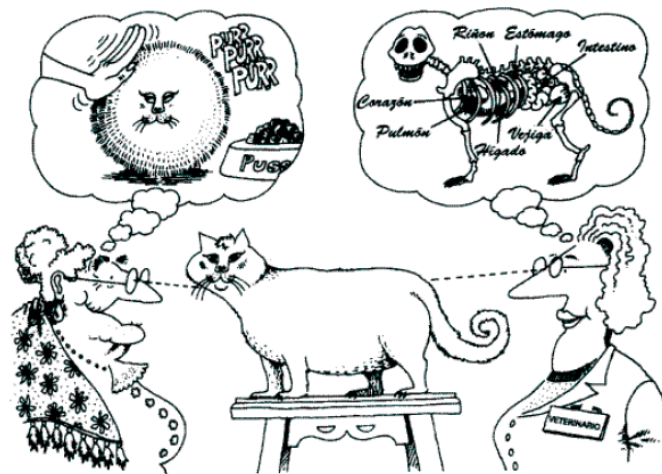
Introducción

El Diseño Conceptual de Base de Datos

En esta etapa se debe construir un esquema de la información que se usa en la empresa, independientemente de cualquier consideración física. A este esquema se le denomina *esquema conceptual*. Al construir el esquema, los diseñadores descubren la semántica (significado) de los datos de la empresa: encuentran entidades, atributos y relaciones. El objetivo es comprender:

- La perspectiva que cada usuario tiene de los datos.
- La naturaleza de los datos, independientemente de su representación física.
- El uso de los datos a través de las áreas de aplicación.

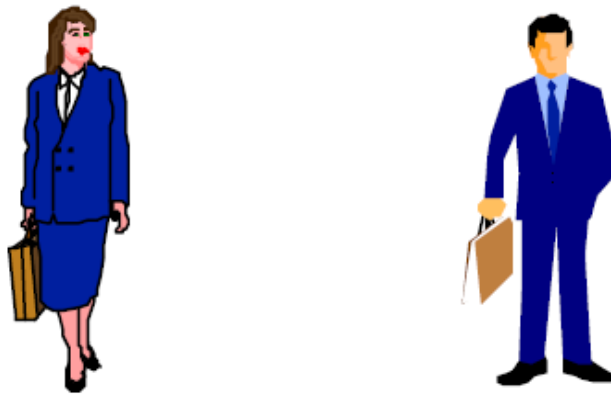
Abstracción juega un papel fundamental en esta etapa y abstraer es un proceso mental que consiste en aislar las cosas más relevantes de la realidad (características y propiedades) en función a la perspectiva del usuario.



La abstracción se centra en las características esenciales de algún objeto, en relación a la perspectiva del observador.

Por ejemplo, cual sería lo más relevante en una persona??

DISEÑO DE BASE DE DATOS



El concepto persona puede verse como resultado de un proceso de abstracción definiendo detalles como (nombre, talla, color de pelo, peso, dirección)

Desde la perspectiva del responsable de personal las propiedades podrían ser: Nombre, Fecha de Ingreso, código, documento de identificación, etc

Desde la perspectiva de un medico las propiedades podrían ser: Nombre, Nro de Historia Clinica, los síntomas, peso, etc.

Las abstracciones ayudan a entender, clasificar y modelar la realidad y permiten identificar clases

En el diseño conceptual de base de datos se usan tres tipos de abstracción: clasificación, agregación y generalización.

Abstracción de Clasificación

Consiste en definir un concepto común como una clase de objeto de la realidad caracterizado por sus propiedades comunes. Por ejemplo en el caso de una persona: Nombre, Talla, Dirección, Peso, Color de Pelo, número de calzado.

En el caso de una factura: nro, fecha, cliente, producto, cantidad, precio unitario, la unidad de medida.

En algunos casos un objeto puede tener varias clasificaciones por ejemplo un escritorio en un sistema de inventarios podría ser: escritorio de melanina, escritorios de madera, etc. .

Podrían existir facturas al contado o facturas al crédito

Se representan de esta manera:

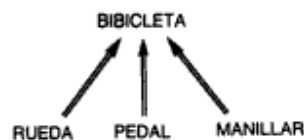
DISEÑO DE BASE DE DATOS



Abstracción de Agregación

Define una clase a partir de un conjunto de otras clases que representan sus partes componentes. Por ejemplo una factura se define a partir de la clase productos, vendedores y clientes.

Se representan de esta manera:

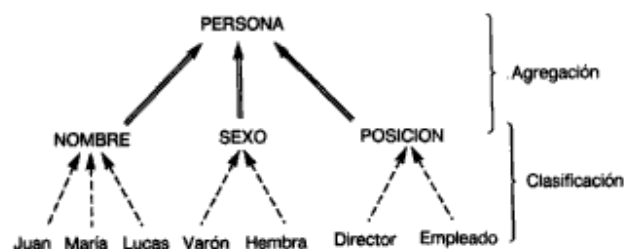


La clasificación y la agregación son las dos abstracciones básicas utilizadas para construir estructuras de datos. Recuerde que la clasificaciones es utilizado cuando partiendo de elementos individuales de información se identifican campos y atributos. Mientras que la agregación reúne los tipos de campo relacionados en grupos.

Una buena forma de ir definiéndolos es a partir de la funcionalidad de un caso de uso que permite definir datos transformados en campos y como es que las operaciones necesitan de otras clases para relacionarse.

Abstracción de Generalización

Define una relación de subconjunto entre elementos de dos o mas clases. De un supertipo a un subtipo. Por ejemplo: clientes naturales y clientes jurídicos: generalizamos como clientes



El Modelo Entidad- Relación

Introducción

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976.

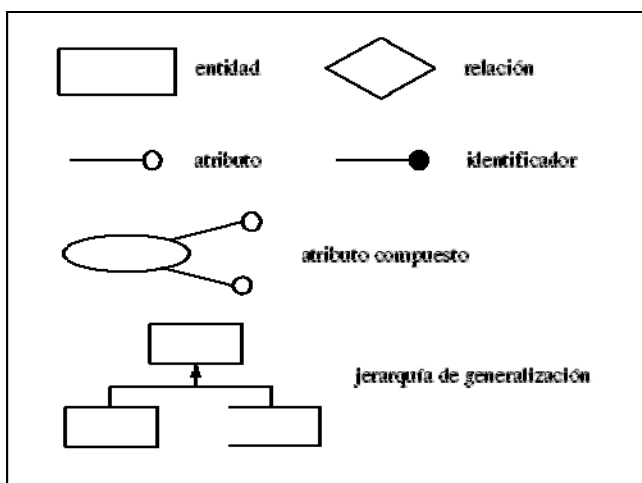
El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Elementos del Modelo Entidad-Relación

Se pueden clasificar en función al orden de aparición:

- Básicos
- Complementarios

Estos son los elementos generales del modelo



Elementos Básicos

- Entidades
- Atributos
- Relaciones

DISEÑO DE BASE DE DATOS

Entidades

- Un objeto de interés para los Negocios
- Una clase o categoría de cosas
- Una cosa con un nombre
- Un sustantivo

Aspecto importante acerca del cual se necesita tener o conocer información para los negocios

Ejemplos:



- Necesita conocer información de los clientes?
- Necesita conocer información de los productos?

Atributo

- Cualquier detalle que sirva para calificar, identificar, clasificar o expresar a una entidad.
- Piezas específicas de información que necesita ser conocida
- Una entidad debe tener atributos

Responde a la pregunta clave:

Qué información necesita conocer acerca de ...?

DISEÑO DE BASE DE DATOS



Ejemplos:

- **Que información necesita conocer acerca del cliente?**
 - Numero del cliente
 - Nombre
 - Estado de activación
 - Fecha de contrato
- **Que información necesita conocer acerca del producto?**
 - Código
 - Ubicación
 - Descripción
 - Unidad de medida
 - Stock Actual
 - Precio de Venta

Relaciones

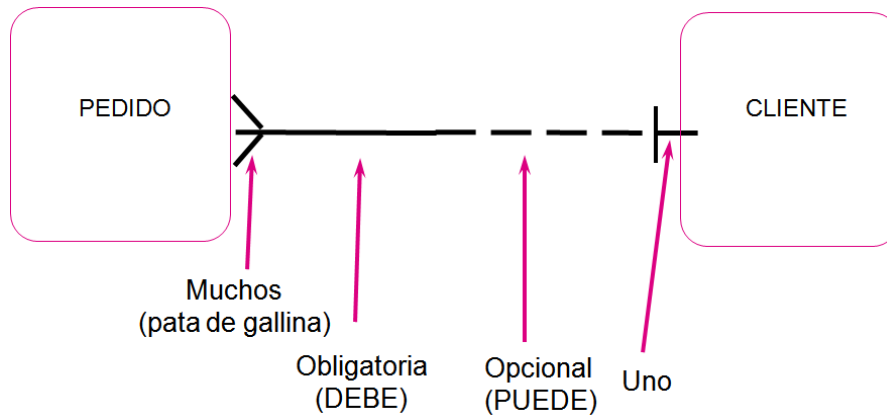
- Representan vínculos entre entidades y consigo mismos
- Una asociación o agregación nombrada por entidades

Ejemplos

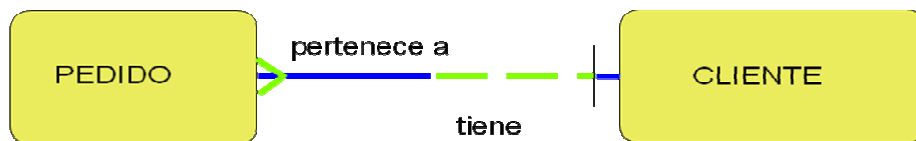
DISEÑO DE BASE DE DATOS



Nombrando Relaciones



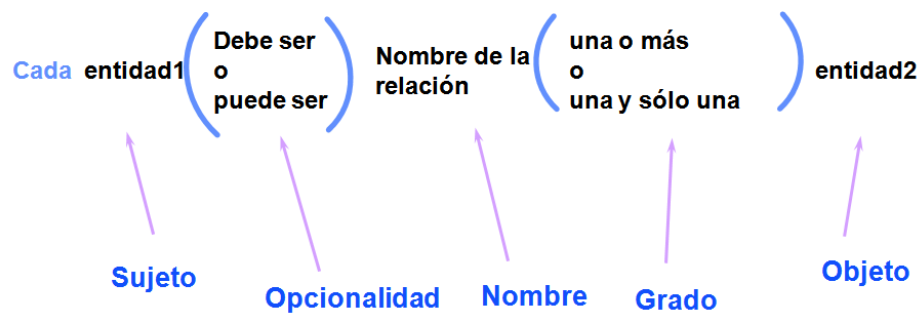
Por ejemplo:



Cada PEDIDO debe pertenecer a uno y sólo un CLIENTE

Cada CLIENTE puede tener uno o más PEDIDOS

Sintaxis General:



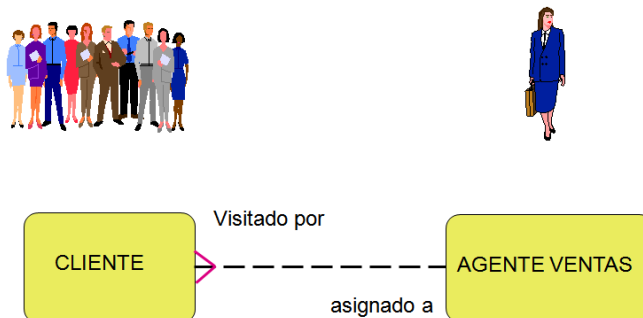
Tipos de Relación

DISEÑO DE BASE DE DATOS

Están representados por el grado de cardinalidad o correspondencia entre las entidades a relacionarse y encontramos los siguientes tipos:

- Uno a muchos
- Muchos a Muchos
- Uno a uno

Relaciones Uno a Muchos:



Relaciones Muchos a Muchos



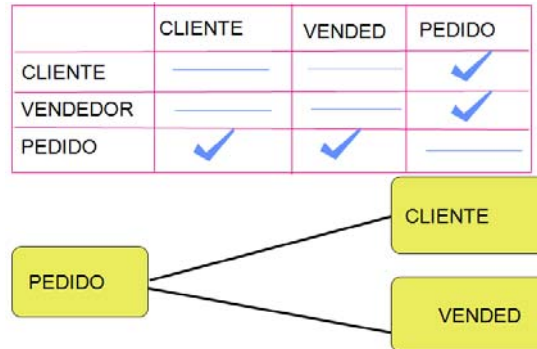
Relaciones Uno a Uno



DISEÑO DE BASE DE DATOS

Recomendaciones para Analizar y Modelar Relaciones

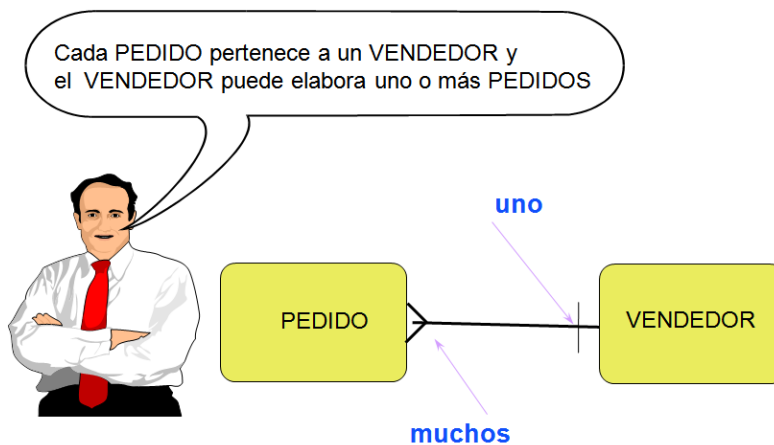
- Determine la existencia de una relación



- Nombre cada dirección de la relación

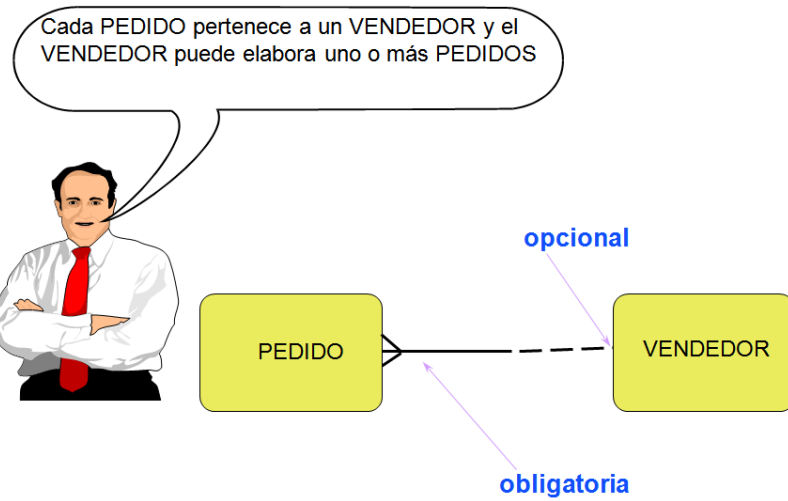


- Determine el grado de cardinalidad de cada entidad en las tablas relacionadas

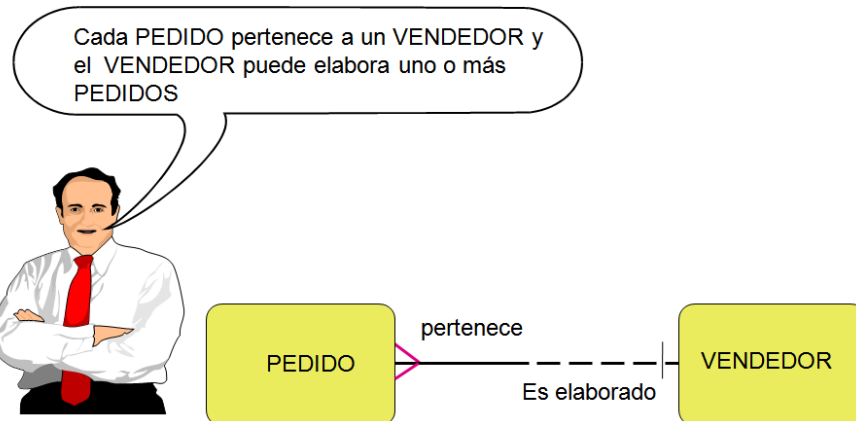


- Determine la opcionalidad de cada dirección de la relación

DISEÑO DE BASE DE DATOS



- Lea las relaciones para validarlas



Elementos Secundarios

- Identificadores
- Atributos Compuestos
- Jerarquías de Generalización

DISEÑO DE BASE DE DATOS

Identificadores

Cada instancia de una entidad debe ser identificada de forma única. Esto implica que no pueden existir dos instancias o registros de información iguales. Siempre se van a diferenciar por un atributo. Por ejm:

En el caso de un cliente que presenta los atributos: código, razonsocial, dirección, email : el código podría identificar a la entidad cliente. La idea es que no existan 2 clientes con el mismo código!



Los Identificadores pueden ser:

- Simples
- Compuestos

Pregunta:

ES SUFICIENTE CON EL (LOS) ATRIBUTO (S) PARA IDENTIFICAR UNIVOCAMENTE A LA ENTIDAD?

Ejemplo:

- ES SUFICIENTE CON el **Código del Producto** PARA IDENTIFICAR UNIVOCAMENTE A LA ENTIDAD **PRODUCTO**?

Identificadores Simple

Cuando existe un solo atributo como identificador. Por ejemplo en el caso de la tabla productos.

- Esquema

Producto	
IdProducto	
Descripcion	
UniMedida	
StockActual	

- Caso

DISEÑO DE BASE DE DATOS

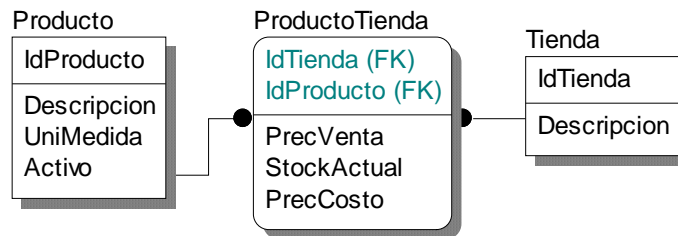
IdProducto	Descripcion	UnidMedidad	StockActual
PR1	CLAVOS	Unid	100
PR2	YESO	Bolsa	2000
PR3	CEMENTO	Bolsa	500

- ✓ Nótese que no pueden existir 2 productos con el mismo código.

Identificadores Compuesto

Es cuando existe más de un atributo como clave compuesta. Imagine que se manejan varios almacenes y cada almacén tiene su propio stock y precio de venta. Ejemplo en la entidad: **ProductoTienda**.

- Esquema



- Caso

IdTienda	IdProducto	PrecVenta	StockActual	PrecCosto
T1	PR1	12	100	10
T1	PR2	15	2000	12
T1	PR3	16	500	11
T2	PR1	12	23	10
T2	PR2	15	222	12
T2	PR3	16.5	12	11

- ✓ Notese que el atributo IdTienda por si solo no podría ser identificador porque se repite (Ejemplo: T1); lo mismo sucede con el IdProducto (Ejemplo: PR1) por lo que el identificador serian ambos: IdTienda con IdProducto

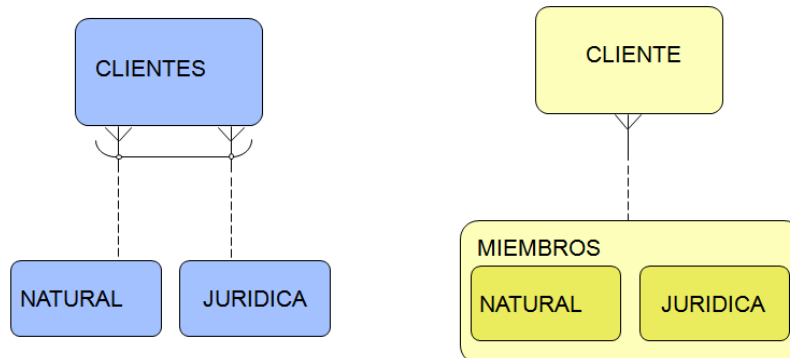
Jerarquías de Generalización

Con las entidades definidas hasta el momento debemos determinar las entidades que tienen operaciones y atributos comunes. Esto marcaria la creación de una entidad mayor que las agrupe.

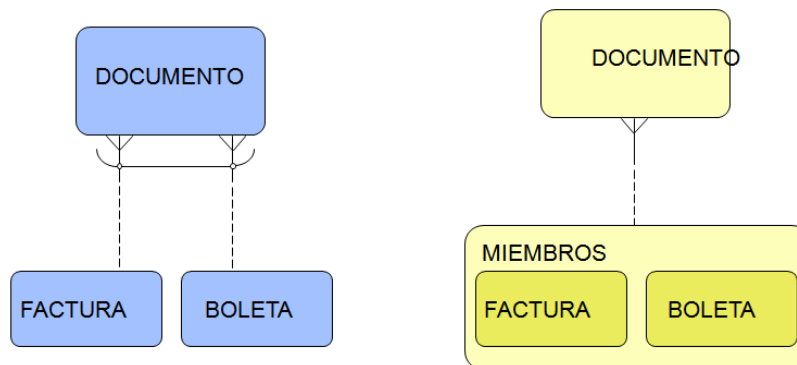
Ejemplos:

DISEÑO DE BASE DE DATOS

- Si una empresa administra clientes naturales y clientes jurídicos:



- Si manejamos facturas y boletas de venta

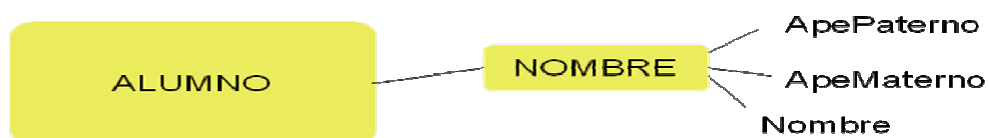


Atributos Compuestos

Son grupos de atributos que tienen afinidad en cuanto a su significado:



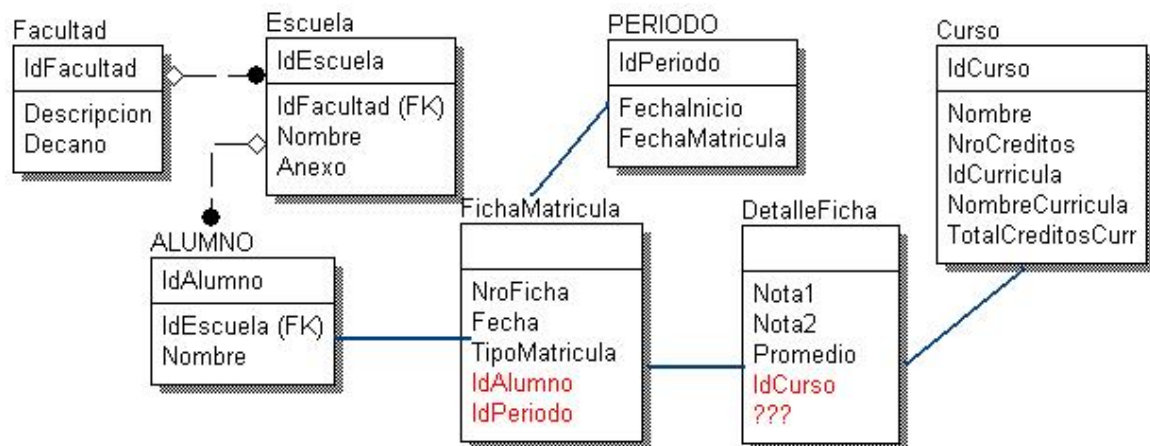
Los atributos deben descomponerse en simples:



DISEÑO DE BASE DE DATOS

Laboratorio: Prepara el Modelo Final en base al Modelo Inicial y las Condiciones Propuestas

Modelo Inicial:



Condiciones Propuestas.

1. Un alumno puede matricularse (FichaMatricula) una sola vez por período.
2. Un detalle de la Ficha de Matricula puede contener 1 ó más cursos diferentes.
3. El nro de la ficha en Ficha de Matricula se reinicia (Vuelve a cero) cada vez que se inicia un nuevo período académico.
4. Si es necesario incluir nuevas tablas o nuevos atributos es posible hacerlo siempre que tengan el sustento del caso.

Observe que en el diagrama se sugieren ciertas relaciones. Haga las relaciones correctas de tal manera que se ajusten al modelo presentado

A TENER EN CUENTA! Pasos recomendados para obtener un Diseño Conceptual representado por un Modelo Entidad-Relación

DISEÑO DE BASE DE DATOS

- ✓ Identificar las entidades.
- ✓ Identificar las relaciones.
- ✓ Identificar los atributos y asociarlos a entidades y relaciones.
- ✓ Determinar los identificadores.
- ✓ Determinar las jerarquías de generalización (si las hay).
- ✓ Dibujar el diagrama entidad-relación.
- ✓ Revisar el esquema conceptual local con el usuario.

El Diseño Lógico

Introducción

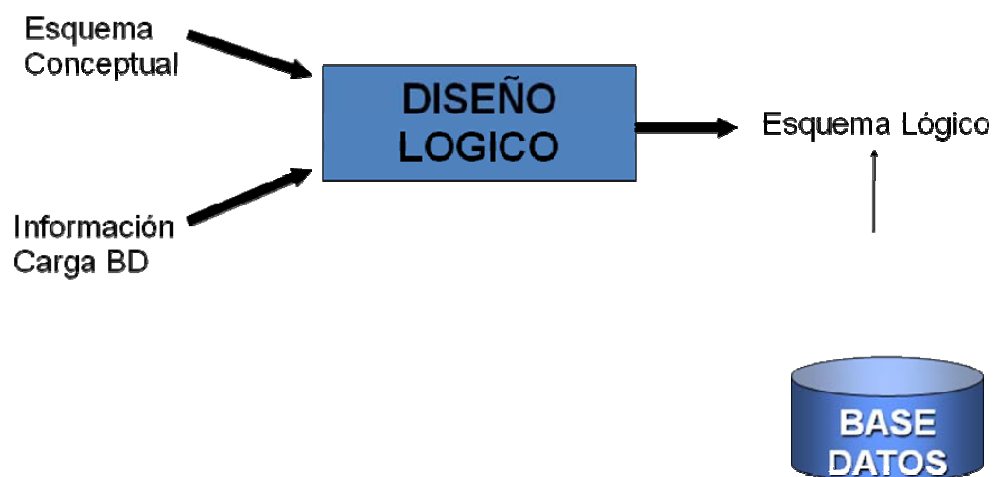
El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976.

El objetivo del diseño lógico es convertir el esquema conceptual en un esquema lógico que se ajuste al modelo de SGBD sobre el que se implementará el sistema. La idea es obtener una representación que use los recursos que el SGBD posee para estructurar los datos y para modelar las restricciones que aseguren una adecuada integridad de los datos.

Preparando el Esquema Lógico

El Esquema Lógico es el resultado del Diseño Lógico de Datos. La información requerida para empezar a trabajar es:

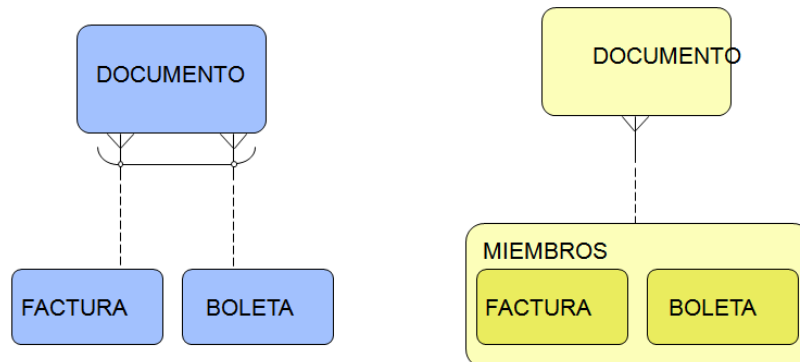
- Esquema Conceptual
- Información de la Carga Posible de la Base de Datos.



El esquema conceptual ya elaborado por medio del Modelo Entidad Relación es fundamental para la consecución del esquema lógico, pero además la Información de la Carga de Datos es vital para decisiones en las transformaciones que podrían darse.

DISEÑO DE BASE DE DATOS

Es en este momento donde debemos preocuparnos, sobre todo si nuestro modelo conceptual incluye **Jerarquías de Generalización o Relaciones 1 a 1**, el universo de los datos, ya que estas decisiones podrían tener implicancias enormes en el rendimiento de la base de datos puesta en operatividad. Por ejemplo:



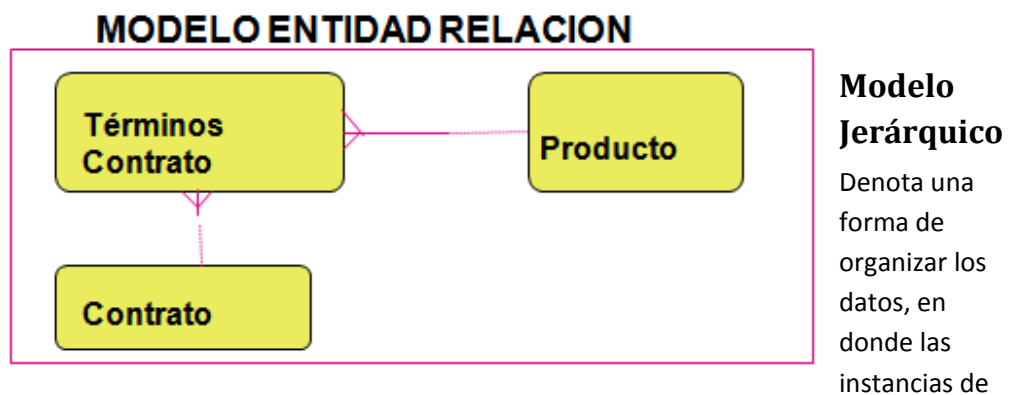
La idea de carga de datos es definir a partir de un número determinado de ocurrencias (documentos) cuantos corresponden a determinado Sub Tipo (cuantas son facturas o cuantas son boletas).

Organización de Datos del Modelo Lógico

Es posible organizar el Esquema Conceptual en diferentes estructuras de Almacenamiento, como:

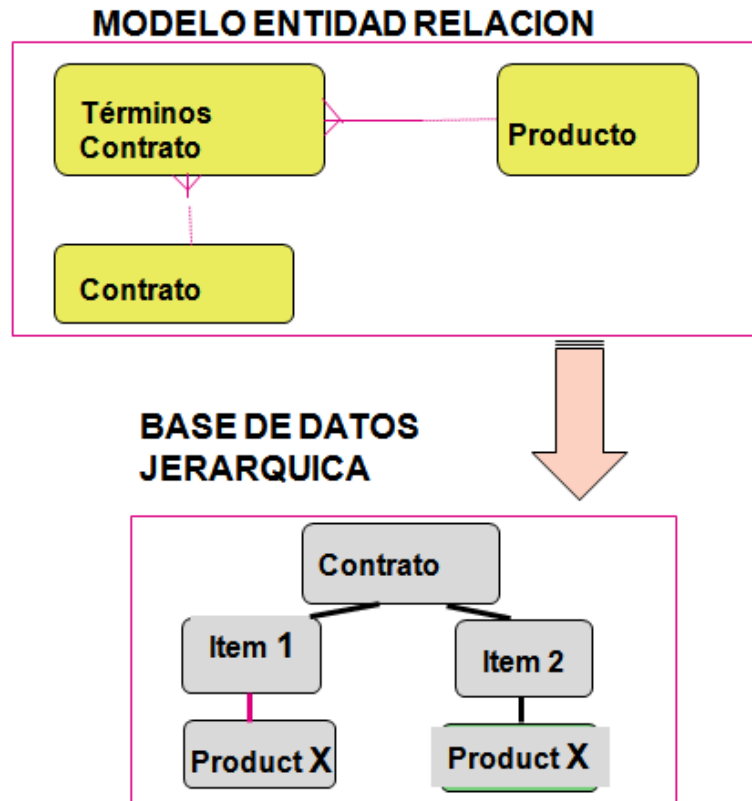
- Jerárquico
- Red
- Relacional

Imaginemos que se ha modelado, a nivel conceptual, un contrato en donde se definen las condiciones generales de pago y se detallan las características de los productos



DISEÑO DE BASE DE DATOS

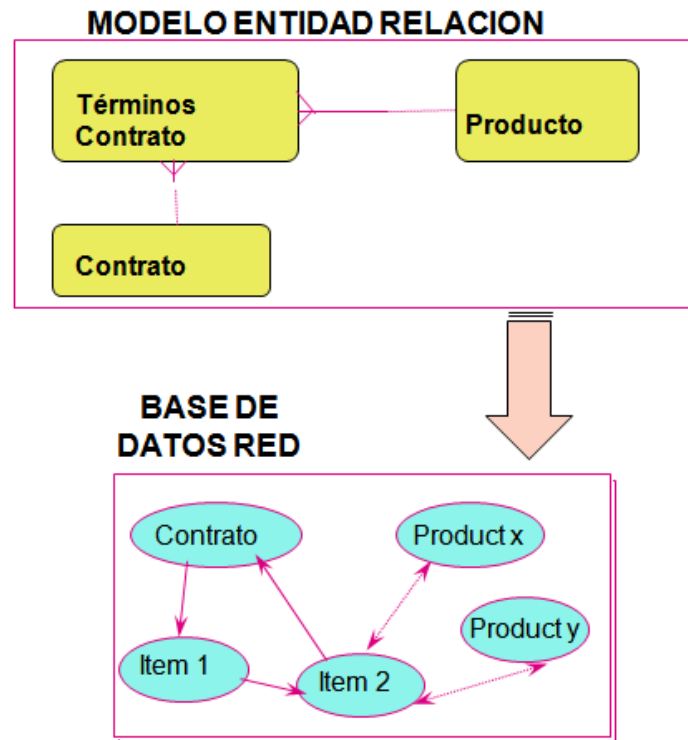
una entidad dependen exclusivamente de otra.



Modelo en Red

Denota una forma de organizar los datos sin una dependencia absoluta. En donde una determinada instancia de una entidad podría pertenecer a 1 o más instancias de otra entidad.

DISEÑO DE BASE DE DATOS

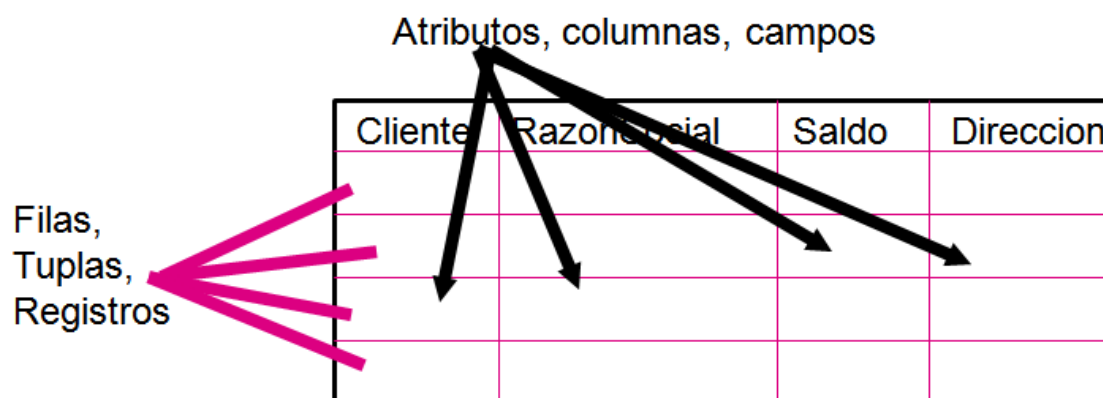


Modelo Relacional

Consiste en organizar al modelo conceptual en una matriz bidimensional

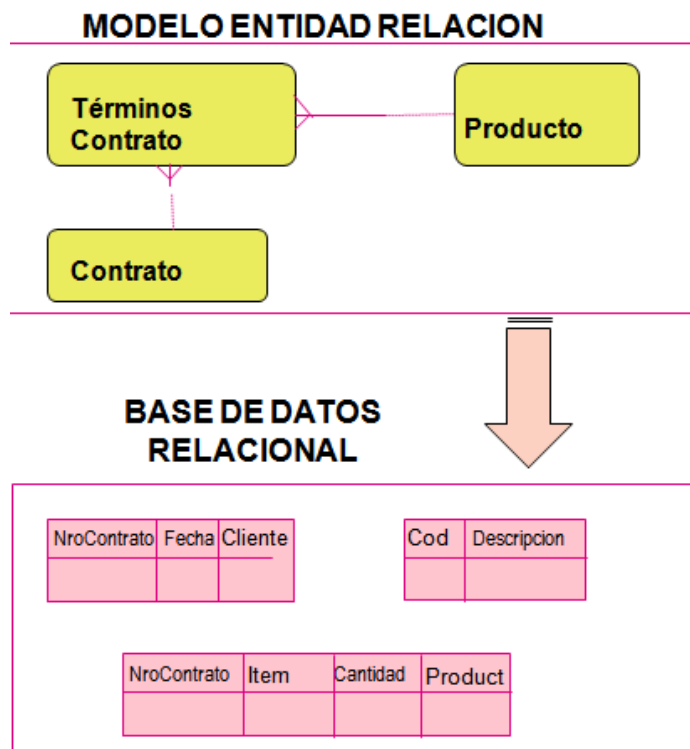
- Filas (llamados tuplas o registros)
- Columnas (llamados campos o atributos)

Este modelo relacional fue propuesto por Codd en 1970



DISEÑO DE BASE DE DATOS

El modelo E-R originalmente propuesta queda definido así:



Terminología en el Diseño Lógico

Aparecen términos propios de esta etapa del modelo de datos comparándolos con el Modelo Conceptual

Conceptual	Lógico
Análisis	Diseño
Entidad	Tabla
Relación	Clave Foránea (FK)
Atributo	Columna
Identificador	Clave Primaria (PK)

Transformaciones en el Diseño Conceptual en el Diseño Lógico

Relaciones 1 a 1

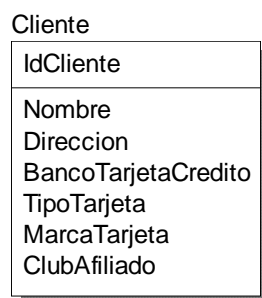
Deberán mantenerse siempre que permitan ahorros significativos de espacio de almacenamiento.

Por ejemplo el caso del cliente que tenga datos de la vida Social del Cliente adicionalmente. La decisión de mantener las entidades separadas va por el lado de la carga de datos:

- Si del 100% de clientes solo el 2% de ellos tiene datos en Vida Social, es mas conveniente mantener la relación.



- Si del 100% de clientes el 55% o más de ellos tiene datos en Vida Social, es conveniente eliminar la relación y tenerlo en una sola entidad.



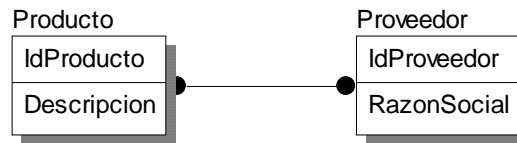
Relaciones Muchos a Muchos

Originan la creación de una nueva entidad llamada: entidad asociativa, que tendrá como clave primaria a las claves primarias de las entidades originalmente relacionadas.

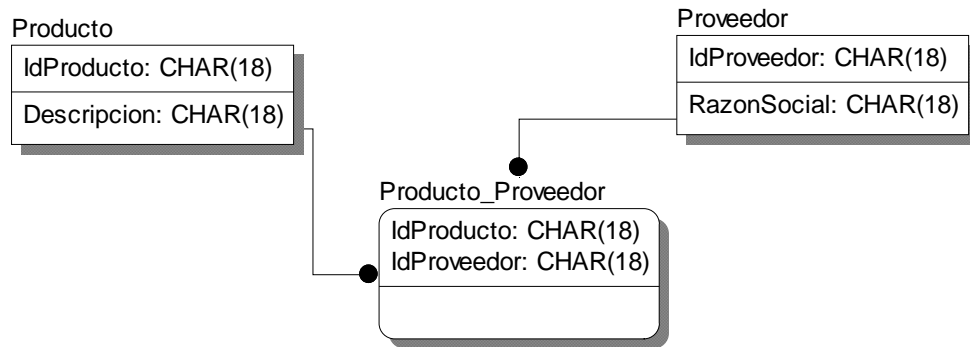
En el caso que se necesite conocer que proveedores distribuyen un producto determinado o que productos distribuye un proveedor.

La relación original corresponde a:

DISEÑO DE BASE DE DATOS



Realizadas las transformaciones del caso aparece una nueva entidad:

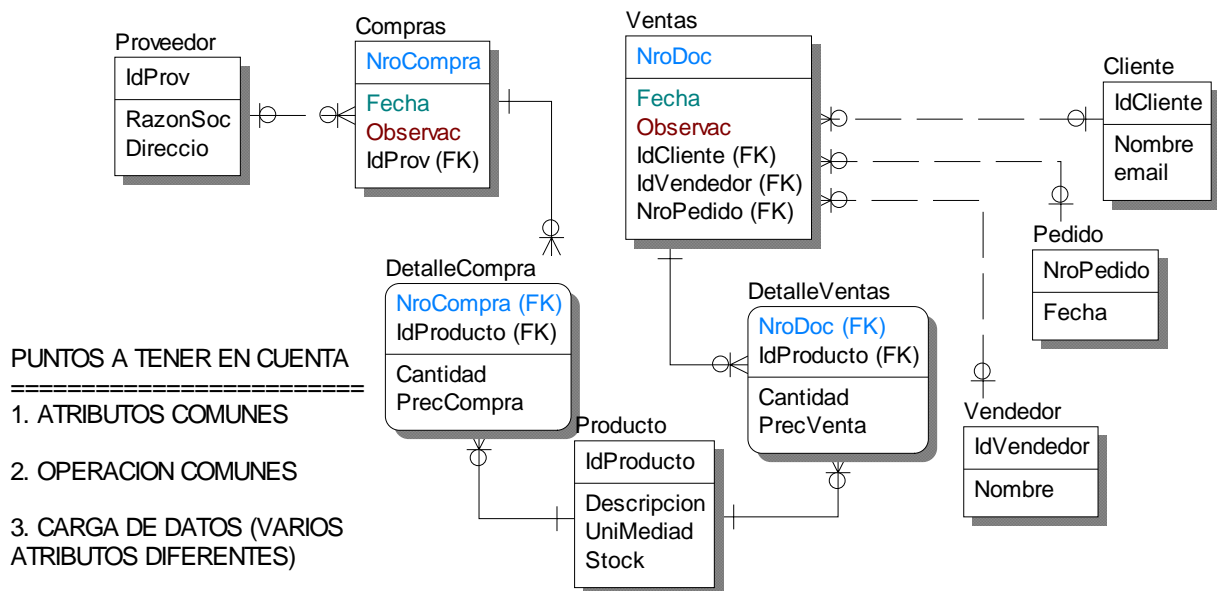


Atributos Compuestos

Hay que crear nuevos atributos individuales.

Jerarquías de Generalización

De acuerdo a las clase y subclases encontradas es posible: unir o dejar separados los subtipos, veamos el siguiente modelo y analice Compras y Ventas:



PUNTOS A TENER EN CUENTA

1. ATRIBUTOS COMUNES
2. OPERACION COMUNES
3. CARGA DE DATOS (VARIOS ATRIBUTOS DIFERENTES)

Como puede apreciar existen atributos comunes en compras y ventas. Veamos algunos casos.

DISEÑO DE BASE DE DATOS

Ventas

Documento	Cliente	Personsal	Fecha	Estado	Pedido
100000017	CLI2	P1	01/01/2003	C	100000014
100000023	CLI2	P2	19/02/2003	C	4
100000024	CLI2	P3	19/02/2003	C	5
100000026	CLI1	P4	19/02/2003	C	6
100000030	CLI4	P2	19/02/2003	C	7
100000031	cli5	P2	19/02/2003	C	8
100000033	CLI4	P1	27/02/2003	C	9
100000034	CLI2	P1	05/05/2000	C	10
100000035	CLI4	P5	07/07/2003	C	11
100000036	CLI2	P2	08/08/2003	C	12
100000037	CLI2	P3	05/05/2002	C	13
100000038	CLI2	P3	08/08/2003	C	13
100000039	CLI2	P1	08/08/2003	C	14
100000040	CLI2	P1	06/06/2003	C	14
100000041	CLI2	P3	06/06/2003	P	15
100000042	CLI4	P3	08/09/2003	P	16
100000043	CLI4	P3	08/09/2003	P	16
100000044	CLI2	P5	06/06/2000	P	17
100000045	CLI2	P5	09/09/2002	P	17

Compras

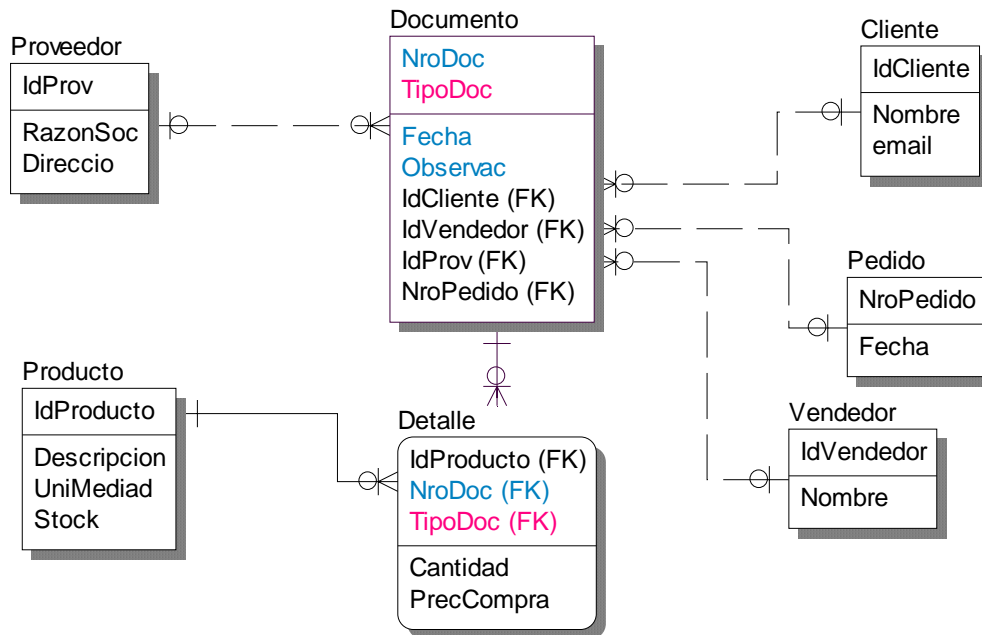
Documento	Proveedor	Fecha	Estado
1	PV01	01/01/2000	C
2	PV02	01/08/2000	C

De acuerdo a los puntos a tener en cuenta:

1. Si existen atributos comunes
2. También existen operaciones comunes, ya que ambos afectaran directamente al Stock del producto y permitirán construir el kardex de un producto determinado.
3. Analizando carga de datos:
 - a. Se tiene atributos mas diferenciados por el lado de la venta. Por lo que si hay un 60% a mas de ventas, sería conveniente unir las tablas
 - b. Se tiene atributos mas diferenciados por el lado de la venta. Por lo que si hay menos de 20% de ventas, sería conveniente separar las tablas

En el caso expuesto de acuerdo al análisis de los datos se puede apreciar que hay mas Ventas que compras, por lo que la carga de datos se ajusta mas al punto a., siendo asi el modelo quedaría:

DISEÑO DE BASE DE DATOS



Nótese que la tabla generalizada contiene los atributos comunes:

- NroDoc
- Fecha
- Observac

Adicionalmente se incluyen los atributos no comunes de ambas tablas

- IdCliente (Venta)
- IdVendedor (Venta)
- NroPedido (Venta)
- IdProveedor (Compra)

Como parte final se debe incluir un atributo que identifique al SubTipo, en este caso:

- TipoDoc

A continuación presentamos una muestra de los datos ya en una sola tabla:

DOCUMENTO

Documento	TipoDoc	Proveedor	Pedido	Cliente	Fecha	Observac	Personal
1	A	PV01	NULL	NULL	01/01/2000	C	NULL
2	A	PV02	NULL	NULL	01/08/2000	C	NULL
100000017	F	NULL	100000014	CLI2	01/01/2003	C	P1
100000023	F	NULL	4	CLI2	19/02/2003	C	P2

DISEÑO DE BASE DE DATOS

100000024	F	NULL	5	CLI2	19/02/2003	C	P3
100000026	F	NULL	6	CLI1	19/02/2003	C	P4
100000030	F	NULL	7	CLI4	19/02/2003	C	P2
100000031	F	NULL	8	cli5	19/02/2003	C	P2
100000033	F	NULL	9	CLI4	27/02/2003	C	P1
100000034	F	NULL	10	CLI2	05/05/2000	C	P1
100000035	F	NULL	11	CLI4	07/07/2003	C	P5
100000036	F	NULL	12	CLI2	08/08/2003	C	P2
100000037	F	NULL	13	CLI2	05/05/2002	C	P3
100000038	F	NULL	13	CLI2	08/08/2003	C	P3
100000039	F	NULL	14	CLI2	08/08/2003	C	P1
100000040	F	NULL	14	CLI2	06/06/2003	C	P1
100000041	F	NULL	15	CLI2	06/06/2003	P	P3
100000042	F	NULL	16	CLI4	08/09/2003	P	P3
100000043	F	NULL	16	CLI4	08/09/2003	P	P3
100000044	F	NULL	17	CLI2	06/06/2000	P	P5
100000045	F	NULL	17	CLI2	09/09/2002	P	P5

La ventaja fundamental por el lado de la unión de las tablas va por el rendimiento fundamentalmente.

A TENER EN CUENTA! Las 12 Reglas de Codd que determinan la fidelidad de un Sistema de Gestión Base Datos (SGBD) al Modelo Relacional – Codd 1984-

REGLA 0: Para que un sistema se denomine sistema de gestión de bases de datos relacionales, este sistema debe usar (exclusivamente) sus capacidades relacionales para gestionar la base de datos.

- **REGLA 1: REGLA DE LA INFORMACIÓN:** Toda la información en una base de datos relacional se representa explícitamente en el nivel lógico exactamente de una manera: con valores en tablas.
- **REGLA 2: REGLA DEL ACCESO GARANTIZADO:** Para todos y cada uno de los datos (valores atómicos) de una BDR se garantiza que son accesibles a nivel lógico utilizando una combinación de nombre de tabla, valor de clave primaria y nombre de columna.
- **REGLA 3: TRATAMIENTO SISTEMÁTICO DE VALORES NULOS:** Los valores nulos (que son distintos de la cadena vacía, blancos, 0, ...) se soportan en los SGBD totalmente relacionales para representar información desconocida o no aplicable de manera sistemática, independientemente del tipo de datos.
- **REGLA 4: CATÁLOGO DINÁMICO EN LÍNEA BASADO EN EL MODELO RELACIONAL:** La descripción de la base de datos se representa a nivel lógico de la misma manera que los datos normales, de modo que los usuarios autorizados pueden aplicar el mismo lenguaje relacional a su consulta, igual que lo aplican a los datos normales.
- **REGLA 5: REGLA DEL SUBLENGUAJE DE DATOS COMPLETO:** Un sistema relacional debe soportar varios lenguajes y varios modos de uso de terminal (ej: rellenar formularios, etc.). Sin embargo, debe existir al menos un lenguaje cuyas sentencias sean expresables, mediante una sintaxis bien definida, como cadenas de caracteres y que sea completo, soportando (Definición de datos, Definición de vistas, Manipulación de datos, Limitantes de integridad, Limitantes de transacción -iniciar, realizar, deshacer -Begin, commit, rollback-)

DISEÑO DE BASE DE DATOS

- **REGLA 6: REGLA DE ACTUALIZACIÓN DE VISTAS:** *Todas las vistas que son teóricamente actualizables se pueden actualizar por el sistema*
- **REGLA 7: INSERCIÓN, ACTUALIZACIÓN Y BORRADO DE ALTO NIVEL:** *La capacidad de manejar una relación base o derivada como un solo operando se aplica no sólo a la recuperación de los datos (consultas), si no también a la inserción, actualización y borrado de datos.*
- **REGLA 8: INDEPENDENCIA FÍSICA DE DATOS:** *Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cuandoquiera que se realicen cambios en las representaciones de almacenamiento o métodos de acceso.*
- **REGLA 9: INDEPENDENCIA LÓGICA DE DATOS:** *Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cuandoquiera que se realicen cambios a las tablas base que preserven la información.*
- **REGLA 11: INDEPENDENCIA DE DISTRIBUCIÓN:** *Los limitantes de integridad específicos para una determinada base de datos relacional deben poder ser definidos en el sublenguaje de datos relacional, y almacenables en el catálogo, no en los programas de aplicación.*
- **REGLA 12: REGLA DE LA NO SUBVERSIÓN:** *Una BDR tiene independencia de distribución. Las mismas órdenes y programas se ejecutan igual en una BD centralizada que en una distribuida.*

Afinamiento de los Modelos de Datos. Normalización de Datos

Introducción

Dentro del diseño de nuestras tablas muchas veces, estamos tentados a incluir atributos que dependen funcionalmente de otros. Por ejemplo en el caso de una factura al ver la razón social del cliente incluimos este atributo dentro de la tabla de ventas o documentos. Cuando realmente este dato depende de la tabla cliente, lo cual significa una redundancia de datos.

El proceso de normalización busca fundamentalmente: eliminar redundancia de datos basado en formas normales que son reglas que nos inducen a encontrar atributos que se repiten en una determinada tabla y que pueden ser parte de otra tabla existente o nueva.

El Proceso de Normalización

Busca fundamentalmente:

- Evitar redundancia de datos.

Como consecuencia de ello:

- Reduce el consumo de espacio de almacenamiento.
- Facilita el mantenimiento de la Base de Datos.

Formas Normales

Primera Forma Normal

“Eliminar los grupos de datos repetitivos”

Qué hacer?

- Crear una tabla separada por cada grupo de datos repetitivos
- Identificar la nueva tabla con una clave primaria la misma que deberá estar relacionada con la tabla base.

DISEÑO DE BASE DE DATOS

Panorama Inicial:

Imagine que desea almacenar la información de una factura. Los dato de la factura: 90, la cual contiene 3 productos. Vemos los datos:

Documento

nro	Fecha	IdCli	RazonSocial	Direcc	Obs	Tventa	IdPro	Descrip	u-med	cant	punit	stota	igv	total
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	Cr	p1	cemento	bsa	10	20	200	98	598
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	Cr	p2	yeso	bsa	20	10	200	98	598
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	Cr	p3	fierro	unid	5	10	100	98	598
...														
100	02/01/07	c1	LOS cocoz	Aven America										

Observe que por cada línea de detalle –por cada producto incluido en la factura- tendríamos que repetir todo un grupo de información común al documento.

Note que lo **marcado con amarillo** es información repetitiva. En este caso procedemos:

- Separarlos en una nueva tabla con una clave primaria respectiva –nro-.

nro	Fecha	IdCLiente	RazonSocial	Direcc	Obs	Tventa	igv	total
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	cr	98	598
100	02/01/07	c1	LOS cocoz	Aven America	Yy	co	19	100

- La tabla original quedaría:

Nro	IdProd	Descrip	u-med	cant	punit	stota
90	p1	Ceme	bsa	10	20	200
90	p2	Yeso	bsa	20	10	200
90	p3	fierro	unid	5	10	100

Segunda Forma Normal

“Eliminar los atributos que no dependan del identificador o clave primaria”

DISEÑO DE BASE DE DATOS

Qué hacer?

- Crear una tabla separada con los atributos que no dependan de la tabla base
- Identificar la nueva tabla con una clave primaria la misma que deberá estar relacionada con la tabla base.

Panorama Inicial

nro	Fecha	IdCLiente	RazonSocial	Direcc	Obs	Tventa	igv	total
90	01/01/07	c1	LOS COCOS	Av Amer	Xx	cr	98	598
100	02/01/07	c1	LOS cocoz	Aven America	Yy	co	19	100

Observe que los campos marcados con color celeste pueden existir al margen de que se registre una venta, no dependen funcionalmente de la venta.

En este caso:

- Separarlos en una nueva tabla, incluyendo un identificador el mismo que al relacionarse con la tabla base migraría.

IdCLiente	RazonSocial	Direcc
c1	LOS COCOS	Av Amer

- La tabla original quedaría:

nro	Fecha	Obs	Tventa	igv	total	IdCLiente
90	39083	xx	cr	98	598	c1

Tercera Forma Normal

“Eliminar los atributos que dependen de otro identificador o clave alterna”

Qué hacer?

- Crear una tabla separada con los atributos que no dependan de la tabla base
- Identificar la nueva tabla con una clave primaria la misma que deberá estar relacionada con la tabla base.

Panorama Inicial

DISEÑO DE BASE DE DATOS

Nro	IdProd	Descrip	u-med	cant	punit	stota
90	p1	ceme	bsa	10	20	200
90	p2	yeso	bsa	20	10	200
90	p3	fierro	unid	5	10	100

Note que los datos marcados corresponden a la información del Producto.

En este caso:

- Separarlos en una nueva tabla:

IdProd	Descrip	u-med	punit
p1	Ceme	bsa	20
p2	Yeso	bsa	10
p3	Fierro	unid	10

- La tabla original quedaría:

Nro	IdProd	cant	Punit
90	P1	10	20
90	P2	20	10
90	P3	5	10

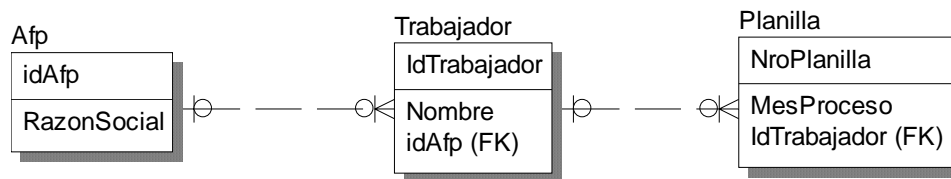
A TENER EN CUENTA! Mantener Datos historicamente de una transaccion

Note que la columna que hace referencia al Precio Unitario (punit) se ha mantenido en la tabla base. Esto es justificable en el caso de que **se interese mantener datos de una transacción** en el tiempo.

Imagine que no se haya quedado este campo precunit en la tabla y el precio en la tabla productos haya pasado del producto: p1 de 20 a 22, a cuanto ascendería ahora el documento??

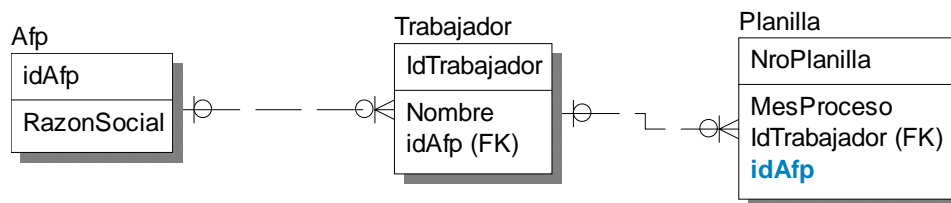
En general toda vez que vamos a mantener las condiciones de una transacción estas deben ser guardadas en las tablas de movimiento. Otro ejemplo correspondería al **código de la afp** donde se encuentra registrado un trabajador, si solo lo mantenemos en la tabla trabajador, al momento de que se cambie de afp, perderíamos la historia donde estuvo afiliado en el transcurso del tiempo

DISEÑO DE BASE DE DATOS



La idea sería que al momento del calculo el idAfp se copie en la planilla, de tal forma que cuando se requiera información de que afp se encuentra afiliado un trabajador lo tomemos de la tabla planilla.

El modelo quedaría así:



Consideraciones Adicionales

- El proceso de normalización incluye mas formas normales (4FN, Boyce –Codd, 5FN)