

The logo consists of the letters 'EGCC' in a bold, white, sans-serif font, centered within a solid blue square.

Modelamiento de Datos

Eric Gustavo Coronel Castillo

Julio – 2007



Modelamiento de Datos

Lección 01 Introducción a los Sistemas de Información

Conceptos Básicos.....	2
Datos.....	2
Información.....	2
Características de la información	2
Importancia Socio-Económica de la información.....	3
Sistemas de Información.....	3
Definición de Sistema de Información (SI)	3
Componentes de un SI.....	3
Objetivos de un SI	4
Tipos de SI	4
SI Transaccionales	4
Sistemas para soporte de decisiones.....	5
Sistemas Estadísticos o Científicos.....	5
Sistemas Mixtos	5

Conceptos Básicos

Datos

Son los números y los factores en bruto, sin analizar, acerca de los sucesos, algo recogido directamente de la fuente.

Información

Es el resultado de la organización o análisis de los datos de alguna manera significativa con un objetivo preestablecido.

Características de la información

La información es evaluada de acuerdo a cuatro factores que deben formar parte de sus características:

Calidad	Cuanto más exacta sea la información mejor será su calidad y los usuarios recurrirán a ella con más confianza cuando tomen una decisión.
Oportunidad	La información debe estar siempre disponible, para la acción correcta y la toma de decisiones en el momento que se le necesite.
Cantidad	Los administradores casi nunca toman decisiones correctas y acertadas si la información no es acertada, oportuna y suficiente; la información debe ser la suficiente para cubrir el espectro del problema en la toma de decisiones.
Relevancia	Debe darse solo la información relevante para la toma de decisiones; toda la información proporcionada debe ser de utilidad; los datos no deben ser de relleno.

Importancia Socio-Económica de la información

La Información es clave para el porvenir de la humanidad e indispensable para modelar ese porvenir.

La investigación, la planificación y la toma de decisiones exigen una información precisa, oportuna, completa, coherente y adaptada a las necesidades específicas de cada usuario y de cada circunstancia.

La expansión de la información se realiza con la segunda revolución industrial, debido a la necesidad de conocer el entorno Socio-Económico y Cultural.

Así como la materia y la energía son componentes fundamentales de la naturaleza, la información es un componente fundamental para el desarrollo de los pueblos.

En la lucha de la selección de las especies, los que disponen de mayor información son los que sobreviven y vencen.

En las empresas la Información por lo tanto es considerada como un Recurso Fundamental, un Bien.

Finalmente la importancia de la información se refleja en la Declaración Universal de los Derechos Humanos.

Sistemas de Información

Definición de Sistema de Información (SI)

Un Sistema de Información es una colección integrada de hardware, software, procedimientos, datos y personas que trabajan en conjunto para generar información. Un SI. es un método formal de poner a disposición de los usuarios información confiable y oportuna que se necesita para facilitar el proceso de la toma de decisiones y permitir que las funciones de planeación, control y operaciones se realicen eficazmente en la organización.

Componentes de un SI

Un SI está conformado por los siguientes componentes:

Contenido	Es el conjunto de datos estructurados y almacenados en un soporte de computador
Equipo físico	Conformado por la Unidad Central de Procesos y los equipos periféricos.
Equipo lógico	Es el software que incluye el DBMS ó Sistema de Gestión de Base de Datos (por ejemplo SQL Server), las aplicaciones, el software de comunicaciones, y software para objetivos específicos.

Administrador	El responsable del área de datos, es el Administrador de de la BD y el Administrador del Area de Informática.
Usuarios	Conformado por el personal que usa directamente las aplicaciones del sistema, los cuales pueden ser informáticos ó no informáticos.

Objetivos de un SI

- Obtener de ventajas estratégicas para la organización.
- Ubicarse en el lugar correcto, tiempo correcto y forma correcta.
- Reducir de costos.
- Proveer funciones de automatización para tareas críticas.
- Brindar soporte a un número creciente de usuarios.
- Desarrollar nuevas aplicaciones en función a las aplicaciones existentes.
- Proveer soporte para periodos largos de tiempo.

Tipos de SI

SI Transaccionales

Están orientados a cubrir la operatividad de la organización, cumplir con cada uno de los procesos que la empresa realiza como objetivo de negocio. Por ejemplo el Sistema de Ventas.

Características:

- Se leen pocas columnas.
- La data es leída por medio de índices.
- SQL simple.
- Tiempo de respuesta pequeños.
- Orientados al nivel operativo.

Sistemas para soporte de decisiones

Deben cubrir la demanda de información par la toma de decisiones estos sistemas se orientan al nivel estratégico de la organización y al táctico en un menor grado.

Características

- Se leen muchas columnas.
- La data es leída secuencialmente.
- Tiempo de respuesta en minutos y horas.
- Se basan en programas de ejecución en lotes.

Sistemas Estadísticos o Científicos

Son SI complejos que generalmente apoyan a los procesos de investigación y explotación de sistemas complejos por la naturaleza de las operaciones a la que está orientada y diseñada. Por ejemplo, el sistema de un laboratorio minero.

Características:

- Se leen muchas columnas.
- La data es leída secuencialmente.
- SQL complejo.

Sistemas Mixtos

Combina cualquiera de los SI anteriores, por lo tanto las características que puede adoptar este tipo de sistema es la combinación de los otros tipos de SI.

Página en Blanco



Modelamiento de Datos

Lección 02 Introducción al Modelamiento de Datos

Bases de Datos (BD).....	2
Evolución Histórica de la Definición de Base de Datos	2
Sistema de Gestión de Base de Datos.	3
Modelamiento de Datos	3
Necesidad de Modelar los Datos.....	3
Planificación Estratégica.	3
Origen e Identificación de los Datos a Partir de la Realidad.....	4
El Modelo Entidad-Relación (ER)	4
Ciclo de Vida del Modelamiento	4

Bases de Datos (BD)

Evolución Histórica de la Definición de Base de Datos

A lo largo del tiempo el concepto de Base de Datos ha ido cambiando; en algunos casos, como lo fue en el antiguo Perú se usó un sistema de almacenamiento y procesamiento de datos usando los Quipus aunque todavía no se definía que era una Base de Datos, ya se la usó.

Veamos a continuación diferentes definiciones que se han ido planteando:

- "Colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados". (Martín, 1975).
- "Colección o depósito de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones". (Conference des Statisticiens Européens, 1977).
- "Conjunto de datos de la empresa memorizado en un ordenador, que es utilizado por numerosas personas y cuya organización está regida por un modelo de datos". (Flory, 1982).
- "Conjunto estructurado de datos registrados sobre soportes accesibles por ordenador para satisfacer simultáneamente a varios usuarios de forma selectiva y en tiempo oportuno". (Delobel, 1982).
- "Colección no redundante de datos que son compartidos por diferentes sistemas de aplicación". (Howe, 1983).
- "Colección integrada y generalizada de datos, estructurada atendiendo a las relaciones naturales de modo que suministre todos los caminos de acceso necesarios a cada unidad de datos con objeto de poder atender todas las necesidades de los diferentes usuarios". (Deen, 1985).
- "Conjunto de ficheros maestros, organizados y administrados de una manera flexible de modo que los ficheros puedan ser fácilmente adaptados a nuevas tareas imprevisibles". (Frank, 1988).
- "Colección de datos interrelacionados". (Elsuari y Navathe, 1989).

Actualmente se puede definir entonces a una Base de Datos como:

Una colección o depósito de datos integrados, almacenados en soporte secundario (no volátil) y con redundancia controlada. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos, y su definición (estructura de la base de datos) única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual ha de permitir captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos".

Sistema de Gestión de Base de Datos.

Conocido como DBMS (Data Base Manager System), es un Software que contiene una combinación ordenada y sincronizada de programas, procedimientos, lenguajes, etc.

Este software suministra los medios que permite a los usuarios mostrar, describir, recuperar y manipular los Datos de la Base de Datos, estos usuarios pueden ser analistas, programadores o al mismo Administrador de la Base de Datos.

Modelamiento de Datos

El Modelamiento de datos es la actividad de definir un mundo teórico y abstracto de manera tal, que las conclusiones que resulten coincidan con las manifestaciones de la realidad.

Por lo tanto un modelo de datos es una abstracción que permite ver la información contenida en los datos, en oposición a los valores individuales de los datos.

Necesidad de Modelar los Datos

- Identificar los requerimientos del usuario
- Incrementar la productividad
- Mantener la consistencia de los datos
- Optimizar la performance
- Permitir el diseño de información integrada

Planificación Estratégica

La planificación es un trabajo colectivo estratégico para determinar las necesidades de la organización para un largo periodo de tiempo. La planificación estratégica exitosa debe preceder al diseño e implementación de la BD que satisfaga las necesidades de información de la organización.

Las ventajas de una planificación estratégica son las siguientes:

- Expresa la comprensión que tienen los responsables de los recursos informáticos.
- Identifica y justifica los requisitos de recursos, y ayuda a asegurar que los recursos estén disponibles.
- Identifica las oportunidades para una efectiva gestión de recursos, incluyendo la colaboración entre departamentos y divisiones de la organización.
- Especifica los planes de acción para alcanzar los objetivos.
- Puede sensibilizar y servir de estímulo de cómo dirigir a los empleados, para que centren sus esfuerzos en aumentar la productividad y hacerlos sentir que son parte real de la empresa.

Origen e Identificación de los Datos a Partir de la Realidad.

El modelo debe ser una representación de la realidad. Que conserva los detalles relevantes. Para poder identificar los datos y detalles de la realidad hay que ponerse en contacto con ella a través de la identificación de documentos o de transacciones, lo cual conlleva a realizar un trabajo de campo en la realidad a estudiar.

La realidad involucra un sinnúmero de detalles cuya importancia o relevancia dependen de los usuarios. Por lo tanto, diferentes usuarios pueden tener diferentes modelos de la realidad.

Los modelos son herramientas poderosas para eliminar los detalles irrelevantes y comprender en forma unificada la realidad de los usuarios individuales.

El Modelo Entidad-Relación (ER)

Es uno de los modelos Conceptuales mas extendido en las metodologías de diseño de Bases de Datos. El modelo E-R puede usarse como una base para obtener una visión unificada de los datos, tomando el enfoque más natural de la realidad que consiste en **Entidades** e **Interrelaciones**.

El modelo ER nos permite proyectar la base de datos a un nivel superior de abstracción, separándolo de las consideraciones relativas a la máquina y a los usuarios, y enfocándolo en un plano en el cual la información tiene un papel fundamental. Este modelo se fundamenta en los conceptos de **Entidad**, que es algo que puede identificarse claramente, e **Interrelación** que representa la relación que existe entre las entidades.

Ciclo de Vida del Modelamiento

Requerimientos	Lógica de Negocios / Control de Flujo: Procesos y Secuencias
Análisis	Identificación de Datos: Reglas de Negocio y Datos Definición de Datos: Objetos Conceptuales
Diseño	Esquemas de Bases de Datos: Objetos Físicos y Procedimientos Almacenados
Implementación	Generación de Código del Servidor: Servidor Generación de Objetos Vinculados a Datos: Cliente



Modelamiento de Datos

Lección 03 Diagrama Entidad-Relación

Entidad	2
Entidad Independiente.....	2
Entidad Dependiente	2
Atributo	3
Dominio	3
Instancia	3
Claves	4
Clave Primaria.....	4
Clave Foránea.....	5
Relación	5
Cardinalidad	6
Cardinalidad de la Relación	6
Notación.....	7
Tipos de Relación.....	7
Relación Identificadora	7
Relación No Identificadora	8
Tipos Adicionales de Relaciones	9
Relación de Muchos a Muchos	9
Relaciones Enésimas.....	9
Relación Recursiva	11
Subtipo de Relaciones.....	12
Relaciones Inclusivas y Exclusivas.....	14
Notación para Subtipo en IE	15
Ejercicios Propuestos.....	15
Ejercicio 01	15
Ejercicio 02.....	16
Ejercicio 03.....	16
Ejercicio 04.....	17
Ejercicio 05.....	17
Ejercicio 06.....	18
Ejercicio 07.....	18

Entidad

Una entidad representa un objeto real ó abstracto que existe en alguna realidad, (gente, lugar, evento, etc.), que tiene propiedades o características que deseamos almacenar en una Base de Datos.

Toda entidad debe ser nombrada y dicho nombre se recomienda que sea en singular, por ejemplo: Factura, Cliente, Empleado, Pedido, Libro, etc.

Un **Tipo de Entidad** es la estructura genérica que describe un conjunto de entidades. Por ejemplo **CLIENTE** es un tipo de entidad que describe las características comunes de un conjunto de Clientes. Una Instancia del Tipo de Entidad **CLIENTE** podría ser un cliente en particular, por ejemplo el Sr. Ricardo Marcelo, otra instancia podría ser el Sr. Gustavo Coronel.

Cuando normalmente se sobreentiende que nos estamos refiriendo a un Tipo de Entidad, se simplifica la expresión y usamos únicamente la palabra **ENTIDAD**, y a cada ejemplar nos referiremos como **INSTANCIA** (*ejemplar de una instancia*).

Todos las instancias deberían tener las mismas propiedades, salvo en el caso que el modelo admita valores nulos.

Entidad Independiente

Una entidad cuyas instancias pueden ser identificadas unívocamente sin necesidad de determinar su relación con otra entidad. Tiene existencia propia. Por ejemplo: Cliente, Empleado, Pedido, etc.

A una entidad independiente también se la conoce como *Entidad Fuerte*, la cual tiene atributos propios, además de aquellos necesarios para establecer asociaciones y que tiene entidades asociadas cuyas ocurrencias dependen de la existencia de ciertas ocurrencias en ella.

Entidad Dependiente

Una entidad cuyas instancias no pueden ser identificadas unívocamente sin que se haya determinado su relación con otra entidad o entidades. No tiene existencia propia, su existencia depende de la existencia de otra u otras entidades.

A una Entidad Dependiente también se la conoce como *Entidad Débil*, la cual tiene atributos que le permiten establecer su relación con otras entidades y cuyas instancias dependen de la existencia de otras instancias de una entidad asociada.

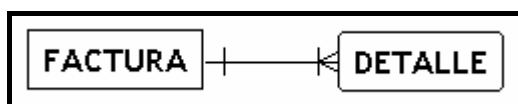


Figura 3.1 Ejemplo de Entidad Independiente y Dependiente.

Atributo

Es una característica o propiedad de una entidad, estas deben servir para describirla e identificarla. Los atributos toman valores de un *Dominio*.

Entre todos los atributos de una entidad debemos elegir uno o varios que identifiquen unívocamente cada una de sus ocurrencias.



Figura 3.2 Atributos de una Entidad.

Dominio

Esta conformado por todos los valores que puede tomar un atributo. Este conjunto de valores debe ser homogéneo.

Las instancias tienen sus atributos y cada uno de estos atributos toma sus valores de un dominio.

Instancia

Es una ocurrencia particular de un tipo de entidad. No pueden existir dos instancias iguales para una misma entidad.

Por ejemplo, para la entidad CLIENTE existe una ocurrencia o instancia como:

Código	Nombre	RUC	Teléfono
C2045	CEPS-UNI	20169004359	481-6693

Claves

Clave Primaria

Sirve para identificar unívocamente a cada instancia de la Entidad; puede estar formada por uno o más atributos. Cuando una Clave está formada por más de un atributo se la conoce como **Clave Compuesta**.

La clave primaria también se conoce como **Primary Key (PK)**

Para reconocer la PK se debe tener en cuenta los siguientes criterios:

Criterio de Unicidad	Los atributos que seleccione como PK deben identificar unívocamente a una instancia de la entidad, no pueden existir dos instancias que tengan un mismo valor para la PK.
Criterio de Minimidad	El número de atributos que se seleccione como la clave primaria (PK) debe ser el mínimo.

Las claves que solo cumplan con el criterio de unicidad solo son **claves candidatas**.

Tomemos como ejemplo a la entidad EMPLEADO con sus correspondientes atributos:

Empleado

IdEmpleado Paterno Materno Nombres Direccion Telefono Clave

En un primer intento podríamos afirmar que un candidato a PK es el atributo Paterno, pero podríamos tener una situación como la siguiente:

IdEmpleado	Paterno	Materno	Nombres	Direccion	Telefono	Clave
E101	Pérez	Quispe	Juan	Av Tacna 304	4550827	cazador
E105	Pérez	García	Jaime	Av. El Tumi 490	5440278	tigre
E108	Marcelo	Villalobos	Ricardo	Jr. El Pino 555	4221166	conde
E112	Pérez	Quispe	Juan	Jr Amazonas 245	2552349	jupiter

La primera, segunda, y cuarta instancias en el atributo Paterno tienen los mismos datos "Pérez"; si elegimos una clave compuesta con los atributos Paterno, Materno y Nombres inclusive, se estarían repitiendo los mismos datos en la primer y cuarta instancias. Luego esta combinación de atributos no nos sirve de PK evidentemente.

Podríamos intentar con la combinación de Paterno y Dirección, lo cual vendría a cumplir con el criterio de Unicidad; luego la dejamos por ahora como una Clave Candidata.

Podemos observar que los atributos DNI, IdEmpleado y RUC también cumplen con el criterio de unicidad; luego tenemos cuatro claves candidatas. Pero la que cumple con ser la mínima en tamaño (Criterio de Minimidad) es la que esta conformada por el atributo **IdEmpleado**; luego esta es la clave que se elige como PK.

Clave Foránea

También conocida como **Foreing Key (FK)**, esta formada por atributos que son PK en otras Entidades.

Tomemos el caso de la entidad FACTURA con sus siguientes atributos:

Factura	
Serie	Numero
IdEmpleado	IdCliente
Fecha	SubTotal
Impuesto	Total

Como se puede observar el atributo IdEmpleado es una PK en la entidad EMPLEADO luego en esta entidad FACTURA IdEmpleado es una FK.

Relación

Representa conexiones, vínculos o asociaciones entre entidades. Toda relación debe ser designada con verbos que impliquen acción de una entidad a otra. Por ejemplo:

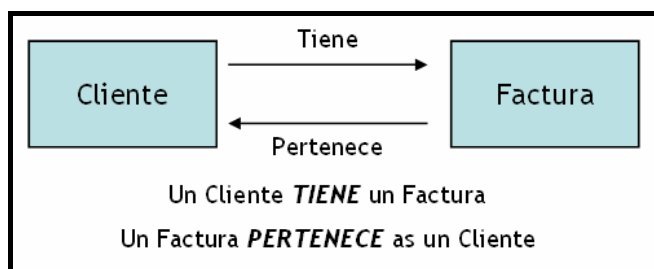


Figura 3.3 Relación entre entidades.

Las Relaciones son el corazón del Modelo de Datos, debido a que ellas describen las Reglas de Negocios y las Restricciones para crear, modificar y eliminar instancias.

Para que dos entidades se “Relacionen” deben tener datos en común; esto es, atributos en común.

Por ejemplo en el caso anterior de las entidades EMPLEADO y FACTURA, estas tienen en común al atributo IdEmpleado; luego este es el nexo entre estas dos entidades.

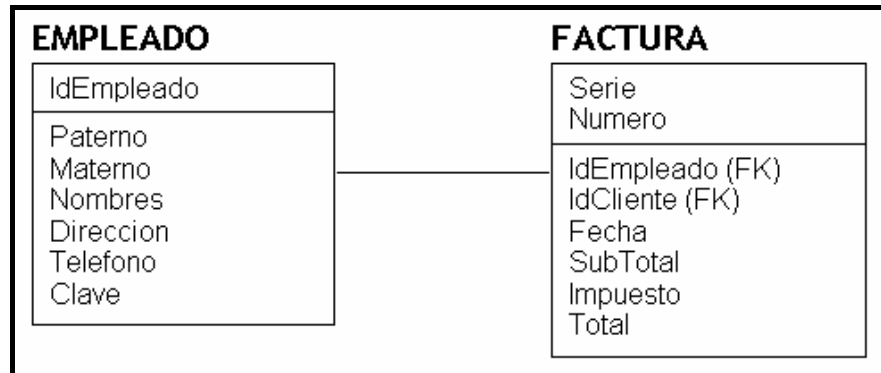


Figura 3.4 Relación entre entidades.

Se puede observar que el Código de un empleado puede aparecer en alguna instancia de la entidad FACTURA, en el atributo IdEmpleado. Si se quisiera conocer los datos correspondientes a un empleado en particular se tendría que recurrir a la entidad EMPLEADO, donde figura el Código de este empleado en el atributo IdEmpleado; y en los demás atributos los otros datos correspondientes a ese empleado en particular.

Nótese que el nombre del atributo en común para ambas entidades es IdEmpleado; en este caso figuran con el mismo nombre, aunque no necesariamente podría ser así; lo importante es que los datos que contengan estos atributos sean de la misma especie en cada entidad para poder establecer la relación entre ambas.

Cardinalidad

La Cardinalidad indica el número posible (máximo y mínimo) de instancias de una entidad que están relacionadas con una instancia en la otra entidad relacionada.

La Cardinalidad se puede usar para definir exactamente cuantas instancias están involucradas en la relación, tanto en la entidad hija como en la entidad padre. También se puede especificar con mayor precisión como se desea manipular las acciones de la base de datos, tales como, INSERT, UPDATE, y DELETE, usando las Reglas de Integridad Referencial.

Cardinalidad de la Relación

La idea de “muchos” en una relación de uno a muchos no significa que tiene que haber más de una instancia de la entidad hija conectada a una entidad padre dada. En vez de esto, “muchas” significa que hay cero, o una, o mas instancias de la entidad hija vinculadas a la entidad padre.

La Cardinalidad es la propiedad relacional que define exactamente cuantas instancias se vinculan en una entidad hija por cada instancia correspondiente en la entidad padre.

Notación

Para representar la cardinalidad de una relación debemos utilizar una notación, en este caso hemos elegido la notación **Information Engineering** (IE), que se detalla a continuación:

Tipo de Cardinalidad	Tipo de Relación	
	Identificativa	No Identificativa
De uno a cero, uno, o muchos		
De uno a uno, o muchos		
De uno a cero o uno		
De cero o uno a cero, uno, o muchos		
De cero o uno a cero o uno		

IE es una notación desarrollada por James Martín, Clive Finkelstein, y otras autoridades en IE.

Tipos de Relación

Relación Identificadora

La cardinalidad permite especificar reglas adicionales de negocio que se aplican a la relación. En el siguiente ejemplo se ha decidido identificar cada COPIA DE PELÍCULA basándose tanto en la Clave Foránea *NumeroDePelicula* y una clave substituta *NumeroDeCopia*. Además, cada PELÍCULA esta disponible como una o más copias de película. Se ha establecido que la **relación es Identificadora**, que la COPIA DE PELÍCULA no puede existir a menos que exista su correspondiente PELÍCULA.

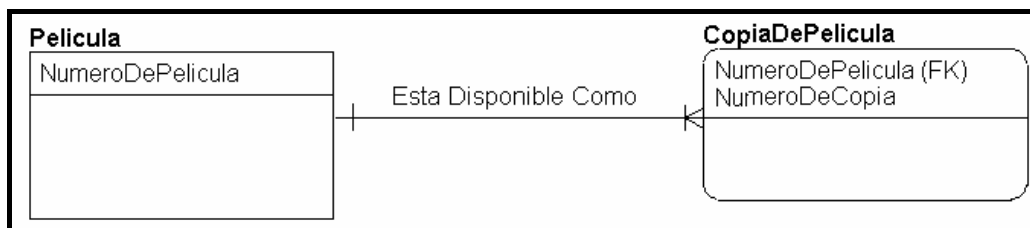


Figura 3.5 Relación Identificadora, cardinalidad de uno a uno o muchos.

El modelo PELÍCULA-COPIADEPELICULA también especifica la Cardinalidad para la relación. La línea de relación muestra que habrá exactamente una película, y solamente una, participando en una relación. Esto se debe a que la entidad PELÍCULA es la entidad Padre en esta relación.

El hecho de hacer a la entidad COPIADEPELICULA la entidad Hija en esta relación, define una COPIADEPELICULA como una o tal vez varias copias rentables de una película determinada. También se ha determinado que si se incluye una PELÍCULA en la Base de Datos es por que tiene por lo menos una COPIADEPELICULA. Esto hace que la Cardinalidad de la relación “**esta disponible como**” sea “de uno a uno o muchos”. Como consecuencia se determina que una PELÍCULA que no tenga copia no es una instancia permitida en esta Base de datos.

En contraste, se podría querer obtener información de todas las películas del mundo, aun cuando no se tengan copias de estas. Esto determinaría que la regla de negocio sea que para una PELÍCULA que existe puede haber cero, o una, o muchas copias.



Figura 3.6 Relación Identificadora, cardinalidad de uno a cero, uno, o muchos.

Relación No Identificadora

Las relaciones *no identificadoras* aportan atributos claves a las entidades Hijas procedentes de las entidades Padres. Por definición, algunas (o todas) las claves no forman parte de la PK de la entidad Hija. Esto significa que para identificar alguna instancia u ocurrencia en la entidad Hija no se necesitará a la entidad Padre.

También podrían existir situaciones en las cuales, una entidad en el terminal muchos de la relación, puede existir sin un “padre”; a estas instancias se las llama “huérfanas”.

Por ejemplo en la relación existente entre la entidad EMPLEADO y la entidad FACTURA (Fig. 3.4), para identificar una factura en particular solo se necesita conocer la serie y número de factura, no se requiere conocer que empleado fue el que realizo dicha factura; luego la relación es no identificadora por que no identifica instancias en la entidad hija (FACTURA).

Si la relación es obligatoria (*mandatory*) con respecto a la entidad hija, entonces la existencia de una instancia en la entidad hija depende de la existencia de otra instancia relacionada en la entidad padre. Si la relación es opcional, una instancia de la entidad hija no requiere de la existencia de la entidad padre para existir.

Veamos el siguiente ejemplo:

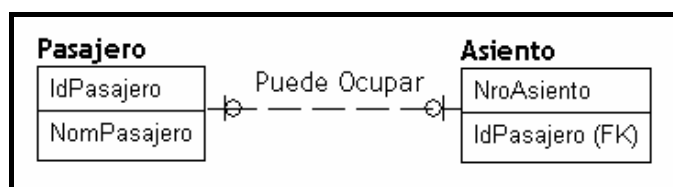


Figura 3.7 Relación Pasajero-Asiento

En este caso, el atributo **IdPasajero** (Código del pasajero) es una FK en la entidad **ASIENTO**. Debido a que **IdPasajero** no identifica al asiento, este identifica al pasajero que ocupa el asiento, las reglas de negocio han determinado que la relación es No Identificadora. Estas mismas reglas determinan que un asiento puede existir sin pasajero (asiento vacío), luego la relación es Opcional.

Tipos Adicionales de Relaciones

Las relaciones definen si la entidad hija es dependiente o independiente de la entidad padre y cuantas instancias están relacionadas en las entidades padre e hija. Conforme se vaya desarrollando el modelo se encontrarán relaciones que no cumplen con los estándares comunes.

Relación de Muchos a Muchos

Si se descubren relaciones que tengan cardinalidad de muchos a muchos en una relación de dos entidades, se debe fraccionar en dos relaciones de cardinalidad uno a muchos.

Por ejemplo a continuación se muestra dos entidades ESTUDIANTE y CURSO las cuales se relacionan con cardinalidad de muchos a muchos; esto quiere decir que un alumno se puede inscribir en diferentes cursos y que un curso puede ser llevado por diferentes alumnos.

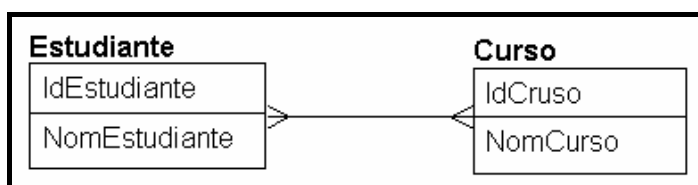


Figura 3.8 Relación de Muchos a Muchos.

Se puede eliminar esta relación creando una Entidad Asociativa, Matricula la cual es usada para la asociación entre las dos entidades anteriores.

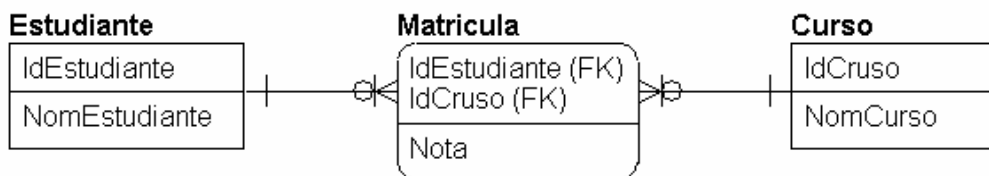


Figura 3.9 Resolución de una relación de Muchos a Muchos.

Relaciones Enésimas

Cuando existe una relación simple entre dos entidades Padre e Hija se dice que es una Relación Binaria. Sin embargo en muchos casos se pueden presentar Relaciones Enésimas.

En el siguiente ejemplo se tiene un caso de este tipo. Como en las relaciones de Muchos a Muchos, esta relación enésima debe ser resuelta usando un grupo de relaciones binarias hacia una entidad asociativa.

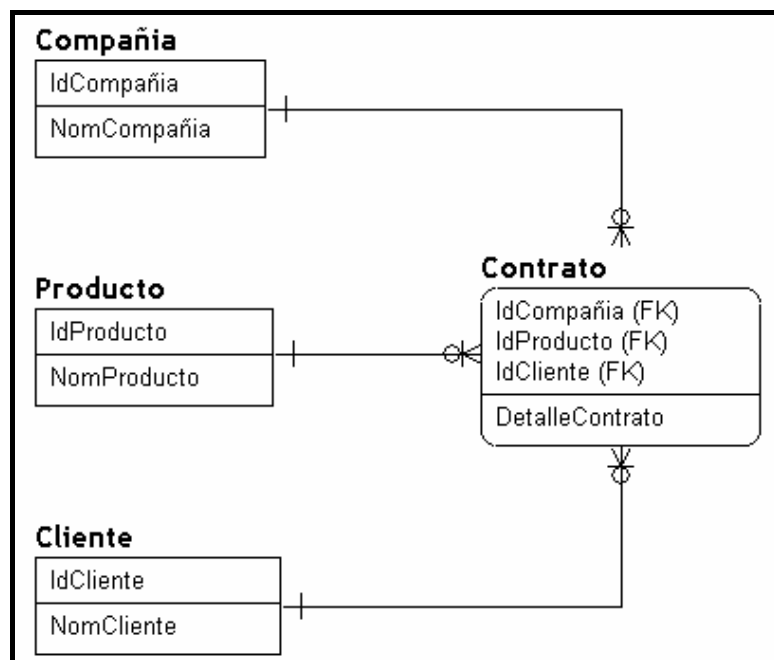


Figura 3.10 Relación Enésima.

Si se considera el diagrama anterior se puede observar que un CONTRATO representa tres formas de relaciones entre COMPAÑÍA, PRODUCTO y CLIENTE. La estructura indica que muchas Compañías venden muchos productos a muchos Clientes.

Cuando se observa una relación como esta se pueden tener varias interrogantes, por ejemplo:

- ¿Un producto debería ser ofrecido por una compañía antes de que este sea vendido?
- ¿Un cliente puede establecer un contrato único incluyendo productos de diferentes compañías?
- ¿Necesita mantenerse al tanto de cuales clientes pertenecen a cuales compañías?

Las estructuras pueden cambiar, dependiendo de las respuestas a estas interrogantes.

Si por ejemplo, la respuesta a la pregunta ¿Un producto debería ser ofrecido por una compañía antes de que este sea vendido? es si, entonces la estructura resultante seria la siguiente:

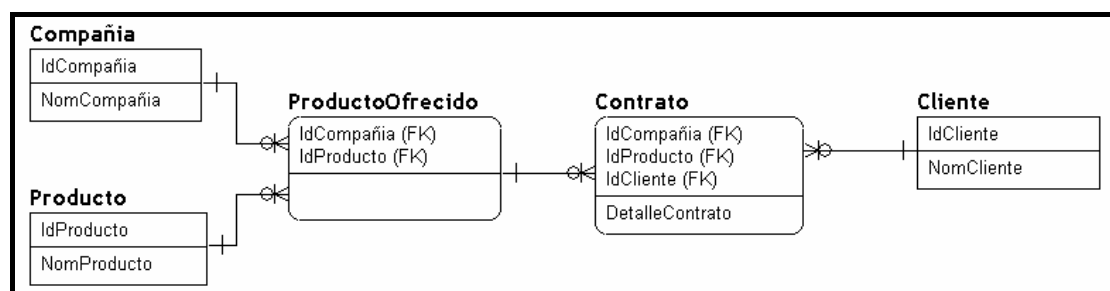


Figura 3.11 Resolución de una relación Enesima.

Por el motivo de que los productos deben ser ofrecidos por las compañías, se ha creado una entidad asociativa que captura esta relación. Como resultado, la relación original de “tres formas” es reemplazada por dos relaciones de “dos formas”.

Relación Recursiva

Una Entidad puede participar en una relación recursiva (también llamada “anzuelo”, Fishhook), en la cual la misma entidad es padre e hija a la vez.

Por ejemplo, una EMPRESA puede ser “padre de” otras EMPRESAS. De manera que se relaciona consigo misma mediante una relación No Identificativa; así la clave primaria aparece en el área de datos de la entidad hija.

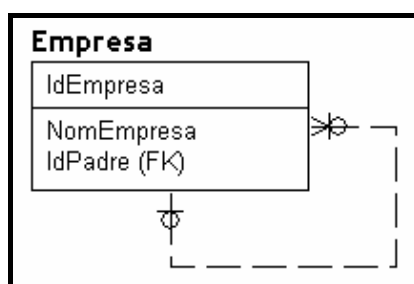


Figura 3.12 Relación Recursiva.

Una relación recursiva es una relación No Identificativa. En el ejemplo de la figura 3.12, en la entidad EMPRESA se puede observar que la FK puede contener NULL, luego, una Empresa puede ser que no tenga Padre, o que una empresa tenga uno o muchas empresas hijas, así como puede ser que una empresa no tenga ninguna empresa hija.

El atributo IdEmpresa migra por medio de la relación recursiva, y aparece en este caso con el RoleName “IdPadre”. Esto sucede por dos razones.

Primero, debido a una regla general de diseño, un atributo no puede aparecer dos veces en la misma entidad con un mismo nombre. Por lo tanto, para construir una relación recursiva, se le debe proporcionar un RoleName al atributo migrado. El **RoleName** es el nombre que se le da a una clave foránea, y este nombre debe describir la regla de negocio personificada por la relación en la que se encuentra esta clave foránea.

Segundo, el atributo “IdEmpresa” es la clave que identifica cada instancia de la entidad EMPRESA y no es lo mismo que el “IdEmpresa” que ha migrado a través de la relación, la cual identifica a la EMPRESA padre. Por lo tanto, no se puede usar la misma definición para ambos atributos, es así que al atributo migrado se le debe asignar un **RoleName**. En el ejemplo el RoleName es IdPadre.

A continuación se muestra la entidad EMPRESA con sus diferentes instancias:

IdEmpresa	NomEmpresa	IdPadre
E01	Empresa AlfaSA	NULL
E02	Repuestos AlfaSA	E01
E03	Servicios AlfaSA	E01
E04	Subsidiaria RAifa Chiclayo	E02
E05	Subsidiaria RAifa Pacherez	E02
E06	PeruSoft SAC	NULL

Se puede observar en esta entidad que Empresa AlfaSA es padre de Repuestos AlfaSA y de Servicios AlfaSA. A su vez Repuestos AlfaSA es padre de Subsidiaria RAifa Chiclayo y de Subsidiaria RAifa Pacherez.

La empresa PeruSoft SAC no es padre de ninguna otra empresa, y tampoco tiene padre.

Subtipo de Relaciones

Un subtipo de relación también se conoce como una Generalización, como una generalización de Jerarquía, o como una Jerarquía de Herencia.

Es una forma de agrupar un grupo de entidades que comparten características comunes. Por ejemplo se puede encontrar durante un trabajo de modelamiento que existen diferentes tipos de CUENTAS en un banco tales como Cuenta Corriente, Cuenta de ahorros, Cuenta de Préstamo.

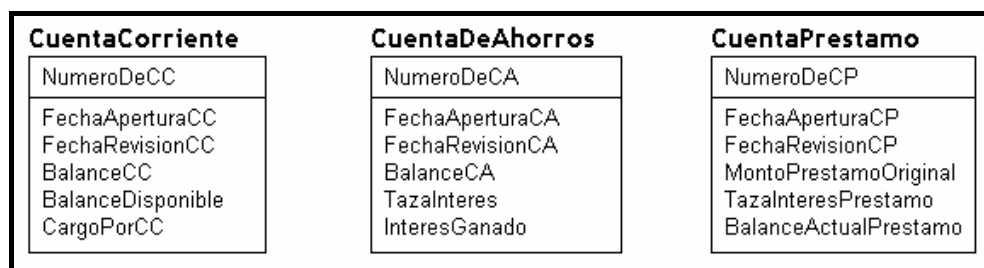


Figura 3.13 Entidades que pueden ser generalizadas.

Cuando se identifican similitudes entre estas entidades independientes, se pueden coleccionar atributos comunes a los tres tipos de cuentas dentro de una Estructura Jerárquica.

Se pueden mover estos atributos comunes dentro de una entidad de nivel más alto llamada *Entidad de SuperTipo* o *Entidad Generalizada*. Aquellos atributos que son específicos a los tipos de cuenta individual permanecen en las Entidades de *SubTipo*.

Entonces podemos crear un supertipo de entidad llamada CUENTA para representar la información que es común a los tres tipos de cuentas. El supertipo CUENTA incluye una PK *NúmeroDeCuenta*.

Se añaden tres Subtipos de Entidades, CTACORRIENTE, CTAdAHORRO, CTAdPRESTAMO, como Entidades Dependientes, las cuales son relacionadas a la Entidad CUENTA usando un SubTipo de Relación.

Luego se tiene como resultado la siguiente estructura:

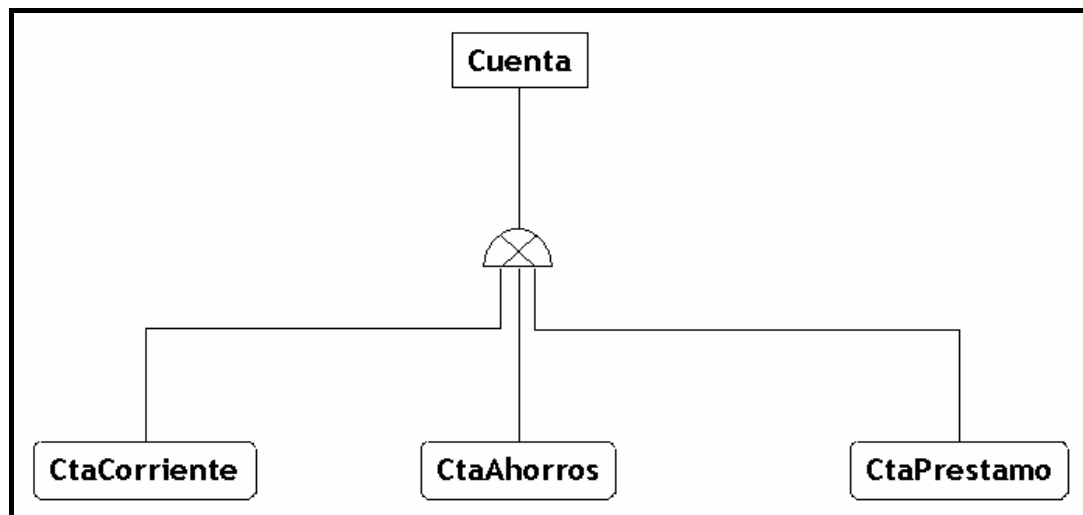


Figura 3.14 Generalización de las Cuentas.

En este ejemplo, una cuenta puede ser cualquiera, una Cuenta Corriente, una Cuenta de Ahorros, o una Cuenta de Prestamos. Cada subtipo de entidad es una cuenta y hereda las propiedades CUENTA. Los tres subtipos de entidades son mutuamente exclusivos.

Para poder distinguir un tipo de cuenta de otro, se puede añadir el atributo *TipoCuenta* como un **Discriminador de Subtipo**. Este discriminador de subtipo es un atributo del Supertipo (CUENTA) y su valor nos indicará el tipo CUENTA que es.

Una vez que se ha establecido el subtipo de relación, se puede observar cada atributo en el modelo original, uno por uno, para determinar si debería permanecer en la entidad de subtipo, ó trasladarse a la entidad de supertipo. Por ejemplo, cada entidad de subtipo tiene una *Fecha de Apertura*, si las definiciones de estas tres clases de *Fecha de Apertura* son las mismas, se las puede entonces trasladar a la entidad de supertipo, y eliminarlas de las entidades de subtipo.

Se debe analizar cada atributo para determinar si permanecerá en el subtipo o será removido al supertipo. En aquellos casos donde un solo atributo aparece en la mayoría, pero no en todas las entidades de subtipo, la decisión se hace difícil. Se podrá trasladar o dejar este atributo en las entidades de subtipo. Si este atributo aparece en la entidad de supertipo, el valor de dicho atributo será NULL cuando el atributo no este incluido en la correspondiente entidad de subtipo.

La decisión de cual alternativa elegir, depende de cuantas entidades de subtipo comparten el atributo común. Si la mayor parte lo hacen, sería mejor removerlos. Si son pocas las entidades que comparten este atributo, es mejor dejarlos en su lugar.

Después de un análisis, el resultado del modelo de nuestro ejemplo podría quedar así:

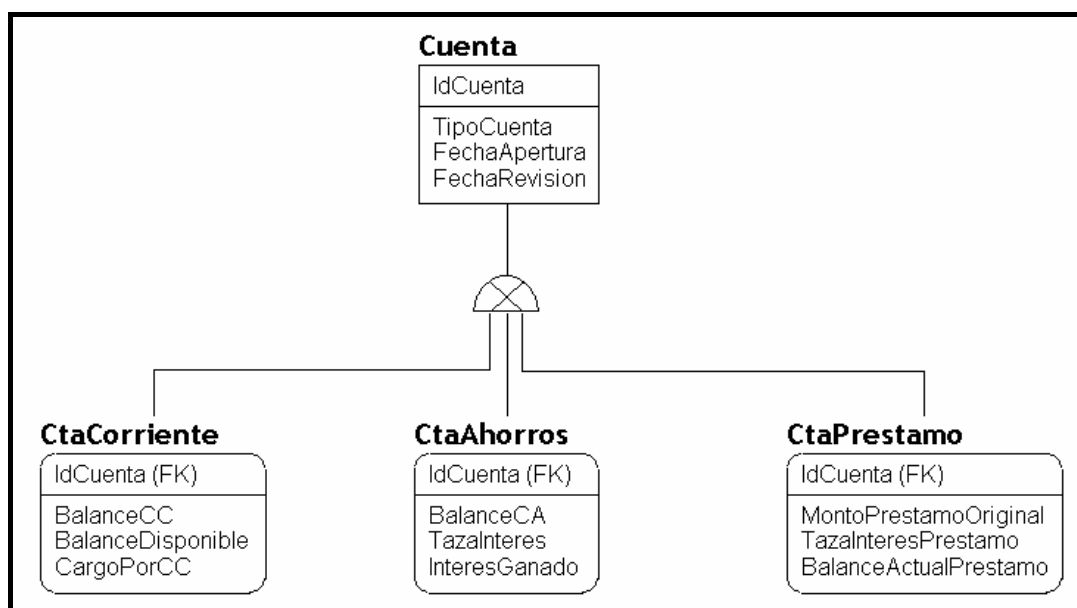


Figura 3. 15 Generalización de las Cuentas.

Cuando se crea un subtipo de relación, se debe tener en cuenta las reglas de negocio, que necesiten ser impuestas a nivel de subtipo, que no son pertinentes a los otros subtipos del supertipo. Por ejemplo, en la entidad CtaPrestamo, las cuentas de préstamos son eliminadas después que alcancen un balance cero. En cambio, bajo las mismas circunstancias, no podríamos realizar eliminaciones en las entidades CtaCorriente y CtaAhorro.

También podrían existir relaciones que son significativas para un solo subtipo y no para otros subtipos en la jerarquía. La entidad CtaPrestamo, por ejemplo, necesita ser examinada para asegurar que ninguna relación previa a los registros de pagos del cliente o asiento se haya perdido debido a una estructura organizacional diferente.

Relaciones Inclusivas y Exclusivas

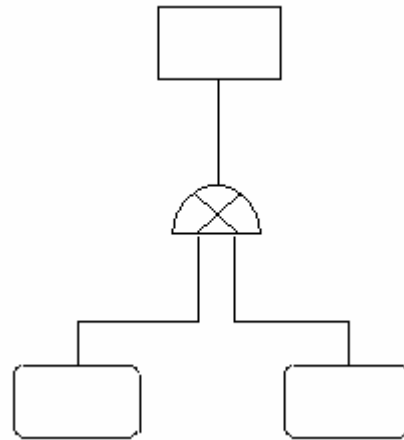
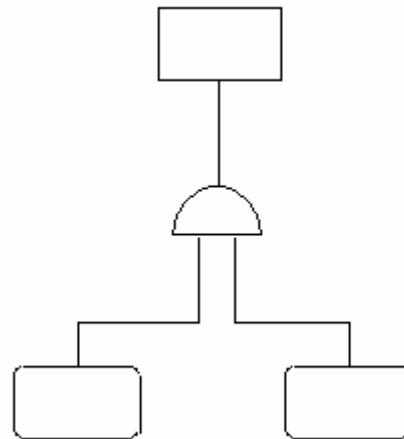
En un subtipo de relación exclusiva, cada instancia en el supertipo puede relacionarse con una y solo una del subtipo.

Por ejemplo, podríamos tener una regla de negocio que establece que un empleado podría ser un empleado de tiempo parcial, o de tiempo completo, pero no de los dos tipos.

Para crear el modelo, se deberá incluir una entidad de supertipo EMPLEADO, con entidades de subtipo TIEMPOCOMPLETO y TIEMPOPARCIAL y un atributo discriminador llamado *TipoEmpleado*. Además, se podría obligar a que el valor del discriminador muestre *C* para tiempo completo y *P* para tiempo parcial.

En una relación de subtipo Inclusiva, cada instancia en el supertipo puede relacionarse con uno o mas subtipos. En nuestro ejemplo, la regla de negocio podría establecer ahora que un empleado puede ser del tipo a tiempo completo o del tipo a tiempo parcial o de ambos tipos.

En este ejemplo, se podría obligar a que el valor del discriminador muestre los valores de validación, *C* para tiempo completo, *P* para tiempo parcial, y *B* para los dos casos (tiempo parcial y tiempo completo a la vez).

Notación para Subtipo en IE**Subtipo Exclusivo****Subtipo Inclusivo****Ejercicios Propuestos****Ejercicio 01**

Desarrollar el diagrama ER que cumpla con las siguientes proposiciones.

1. Cada EMPLEADO debe estar asignado a uno y solo un DEPARTAMENTO.
2. Cada DEPARTAMENTO puede ser responsable de uno ó más EMPLEADOS.
3. Cada EMPLEADO puede ser asignado a uno ó más ACTIVIDADES.
4. Cada ACTIVIDAD puede ser realizada por uno ó más EMPLEADOS.

Ejercicio 02

Desarrollar el diagrama ER que cumpla con las siguientes proposiciones con respecto a la estructura física de una base de datos SQL Server.

1. Cada DATABASE esta constituido por uno o más FILE GROUP.
2. Cada FILE GROUP es parte de uno y solo un DATABASE.
3. Un FILE GROUP esta constituido por uno o más DATA FILE.
4. Un DATA FILE es parte de uno y solo un FILE GROUP.
5. Cada DATABASE esta constituido por uno o más LOG FILE.
6. Cada LOG FILE es parte uno y solo un DATABASE.
7. Cada DATA FILE esta formado por uno o más EXTENTS.
8. Cada EXTENT debe ser incluido en uno y solo un DATA FILE.
9. Cada EXTENT debe estar compuesto por uno o más PAGES.
10. Cada PAGE debe ser parte de uno y solo un EXTENT.
11. Cada DATA FILE debe ser ubicado en uno y solo un HARD DISK.
12. Un HARD DISK puede almacenar uno o más DATA FILE.

Ejercicio 03

Desarrollar el diagrama ER que cumpla con las siguientes proposiciones con respecto a la estructura física de una base de datos Oracle.

1. Cada ORACLE DATABASE debe estar construido por uno o más TABLESPACES.
2. Cada TABLESPACE debe ser parte de uno y solo un ORACLE DATABASE.
3. Cada TABLESPACE esta constituido por uno o más FILES.
4. Cada FILE puede ser parte de uno y solo un TABLESPACE.
5. Cada TABLESPACE esta dividido en uno o más SEGMENTS.
6. Cada SEGMENT debe estar incluido en uno y solo un TABLESPACE.
7. Cada SEGMENT debe estar formado por uno o más EXTENTS.
8. Cada EXTENT debe pertenecer s uno y solo un SEGMENT.
9. Cada EXTENT debe estar formado por uno o más BLOCKS.
10. Cada BLOCK debe ser parte de uno y solo un EXTENT.
11. Cada FILE debe ser ubicado en uno y solo un HARD DISK.
12. Cada HARD DISK puede almacenar uno o más FILES.

Ejercicio 04

Identificar y modelar las entidades con el siguiente conjunto de requerimientos. Escriba una breve descripción de cada entidad. Mostrar por lo menos tres atributos de cada entidad. Realice las aclaraciones necesarias para entender correctamente su interpretación.

1. Soy el propietario de una tienda de alquiler de videos de nombre **vides Club**. Tenemos más de 3000 videos de los que necesitamos llevar un registro.
2. Cada uno de nuestros video esta identificado por un código (número de video), y lo podemos tener en formato CD ó DVD. Para cada película necesitamos conocer su titulo y su categoría (ejm. Comedia, Suspenso, Drama, Guerra, Ciencia Ficción, etc). Generalmente tenemos muchas copias de cada película, por lo menos debemos tener una, cada copia esta identificada también por un código.
3. Frecuentemente nos preguntan por los protagonistas de una determinada película. A los clientes también les interesa conocer el nombre real de los actores y su fecha de nacimiento. También preguntan por las películas de un actor y si es protagonista o no.
4. Tenemos muchos clientes. Solo alquilamos videos a la gente que sea socio del **Video Club**. Para pertenecer a nuestro club, debe tener buen crédito. Por cada socio del club nos interesa registrar su nombre y apellido, teléfono, dirección; y a cada socio se le asigna un número de membresía.
5. Necesitamos llevar el registro de que videos ha alquilado cada socio en tiempo real. Un cliente puede alquilar varios videos al mismo tiempo. También nos gustaría saber el histórico de cada video y cada cliente.

Ejercicio 05

Identificar y modelar las entidades con el siguiente conjunto de requerimientos. Escriba una breve descripción de cada entidad. Mostrar por lo menos tres atributos de cada entidad. Realice las aclaraciones necesarias para entender correctamente su interpretación.

La asociación educativa **EduTech**, desea llevar un control informatizado de los cursos que imparte así como de lo profesores y alumnos que participan en dichos cursos. Para ello, nos han dado las siguientes especificaciones:

1. Cada curso, del que se desea conocer el título, el número de horas y el tema o los temas que trata, se identifica por un código de curso.
2. Cada curso se puede impartir una o varias veces, en diferentes fechas y en cada edición del mismo pueden participar diferentes profesores.
3. Los profesores, de los que se desea conocer su código, nombre, DNI y fecha de antigüedad en la empresa, pueden impartir cursos pero con la restricción de que en una misma edición solo podrán impartir 3 cursos.
4. En cada edición se programan un grupo de cursos, tiene una fecha de inicio y una fecha de finalización.
5. Los alumnos deben ser registrados en la base de datos, y son libres de matricularse en cualquiera de los cursos programados en cada edición.

Ejercicio 06

Identificar y modelar las entidades con el siguiente conjunto de requerimientos. Escriba una breve descripción de cada entidad. Mostrar por lo menos tres atributos de cada entidad. Realice las aclaraciones necesarias para entender correctamente su interpretación.

Se trata de diseñar una base de datos para una red de agencias en franquicias de FAIRHOUSE, empresa especializada en el alquiler y venta de inmuebles.

1. Cada agencia tiene un titular propio y un conjunto de vendedores. Tanto el titular como los vendedores sólo pueden pertenecer a una agencia. Sobre las agencias interesa almacenar su dirección, teléfonos (que pueden ser varios), fax, etc. Además, cada agencia tiene asignada una zona de actuación que es única.
2. Las agencias disponen de inmuebles tanto para alquilar como para vender (o ambas cosas), en el primer caso figurará el precio de alquiler y la fianza a depositar, mientras que en el segundo caso, además del precio de venta, se indica si el inmueble está o no hipotecado.
3. Por otro lado, los inmuebles pueden ser locales comerciales, o pisos. En ambos casos se identifican por un código, interesando conocer el propietario, la dirección y la superficie en m2.
4. Además, en el caso de pisos interesa conocer el número de habitaciones (incluyendo el salón), el número de cuartos de baño, el tipo de gas (natural, ciudad, butano), y si es interior o exterior. Para los locales comerciales se debe conocer si dispone de licencia de apertura.
5. Un cliente puede acudir a varias agencias, en cada una se le asigna un vendedor, que es el encargado de seleccionar los inmuebles que cumplen las características deseadas, y en caso de estar interesado, el cliente debe dar una señal para reservar el inmueble (o los inmuebles) que desea.

Ejercicio 07

Identificar y modelar las entidades con el siguiente conjunto de requerimientos. Escriba una breve descripción de cada entidad. Mostrar por lo menos tres atributos de cada entidad. Realice las aclaraciones necesarias para entender correctamente su interpretación.

Se desea informatizar parcialmente determinado banco con los siguientes supuestos:

1. El banco tiene distintas sucursales que se identifican por un código.
2. Cada sucursal tiene una serie de cuentas corrientes asignadas a ella que se identifican por un código distinto para cada cuenta, aunque sea de distinta sucursal.
3. Una cuenta corriente tiene asociado uno o varios clientes. Es posible sin embargo, que las operaciones que puede realizar cada uno de estos clientes con la cuenta no sean las mismas. Por ejemplo, si la cuenta 110245 tiene como clientes asociados a Sergio Matsukawa y Ricardo Marcelo, es posible que el privilegio de cerrar la cuenta sólo la tenga Sergio, y no Ricardo.
4. Por otra parte, cada cliente, que se identifica por su DNI, puede tener varias cuentas y, por supuesto, unos privilegios distintos en cada una de ellas.
5. Los clientes pueden tener otorgados préstamos, sin que estos estén asociados con ninguna de las cuentas corrientes. Cada préstamo se otorga a nombre de un solo cliente, y a un cliente se le puede otorgar mas de un préstamo.



Modelamiento de Datos

Lección 04 Normalización

Dependencias Funcionales	2
Dependencia Funcional Completa	2
Descriptores Equivalentes.....	2
Dependencia funcional transitiva.....	2
Normalización.....	3
PRIMERA FORMA NORMAL (1FN).....	4
SEGUNDA FORMA NORMAL (2FN)	4
Dependencia Parcial.....	4
TERCERA FORMA NORMAL (3FN).....	5
Ejemplo Aplicativo	5
Enunciado	5
Identificar la Clave Primaria en la Entidad Compleja.....	5
Aplicación de la 1FN.....	6
Identificar los grupos repetitivos en la entidad compleja.....	6
Rescribir la entidad compleja original sin los grupos repetitivos	6
Crear una segunda entidad con los grupos repetitivos	6
Aplicación de la 2FN.....	7
Aplicación de la 3FN.....	8
Ejercicios	10
Ejercicio 01	10
Ejercicio 02.....	11
Ejercicio 03.....	12
Ejercicio 04.....	13

Dependencias Funcionales

Al intentar dibujar las relaciones entre los atributos de una entidad, se debe conocer cuáles atributos son dependientes de otros atributos.

La frase **funcionalmente dependiente** se debe interpretar de la siguiente manera:

Si Salario es funcionalmente dependiente de IdEmpleado, entonces IdEmpleado identifica a Salario. En otras palabras, si conocemos el valor del código del empleado IdEmpleado, podemos identificar el valor del Salario asociado con éste.

$\text{IdEmpleado} \rightarrow \text{Salario}$

Dependencia Funcional Completa

Se dice que B tiene **dependencia funcional completa o plena** de A, si depende funcionalmente de A, pero no depende de ningún subconjunto de A.

$(\text{IdPedido}, \text{IdArticulo}) \rightarrow \text{Precio}$

Descriptores Equivalentes

Se dice que los descriptores A y B son equivalentes, si se cumple:

$A \rightarrow B$

$B \rightarrow A$

Lo que puede representarse por:

$A \leftrightarrow B$

Por ejemplo, los atributos *Codigo_Libro* y *Titulo* son equivalentes, se supone que dos libros distintos no pueden tener ni el mismo código, ni el mismo título.

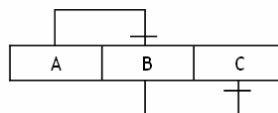
$\text{Codigo_Libro} \leftrightarrow \text{Titulo}$

Dependencia funcional transitiva

Se dice que C tiene dependencia transitiva respecto de A, a través de B si:

$A \rightarrow B$

$B \rightarrow C$



Normalización

La Normalización es un proceso que permite reemplazar las relaciones complejas entre los datos por relaciones más sencillas de tal manera que se facilita la comprensión y el desarrollo del procesamiento de datos.

La idea básica de la normalización es reducir una estructura de datos compleja (multidimensional) a un grupo de entidades y relaciones mas simples.

Los objetivos principales de la Normalización son facilitar la actualización de la Base de Datos, eliminar las redundancias y las anomalías de actualización, de manera tal que se maneje adecuadamente la integridad y consistencia de los datos. El modelo normalizado finalmente debe corresponder y reflejar la realidad a partir de la cual se diseño.

La integridad de datos, se puede asociar en general a restricciones que permitan asegurar que los cambios o actualizaciones que efectúan los usuarios en la base de datos, no resulten en una pérdida de consistencia de éstas, es decir se debe contar con mecanismos que permitan proteger la base de datos.

El concepto de Normalización de los datos tuvo su origen en la teoría relacional numérica y fue planteada formalmente por Edgard F. Codd. Codd encontró en esta teoría una forma elegante de presentar un método para el trabajo de modelar los datos; esta es la razón por la cual de debe seguir estos procedimientos, llamados *Formas Normales*, de manera rígida y en forma ordenada. Esto quiere decir que primero se debe pasar a la Primera Forma Normal, luego a la Segunda Forma Normal luego a la Tercera y así sucesivamente. No se debe alterar esta secuencia.

Codd planteo inicialmente la Primera Forma Normal (1FN), la Segunda Forma Normal (2FN) y la Tercera Forma Normal (3FN), posteriormente en 1971 Heath reformuló la 3FN y en 1977 Fagin definió la Cuarta Forma Normal (4FN).

La presentación formal de la teoría de la Normalización exige un bagaje matemático, con sus correspondientes algoritmos, y es solo indispensable en una formación académica universitaria. No pretendo desarrollar esta formalización, si no mas bien presentar estas ideas de una manera intuitiva para que los lectores a quienes va dirigido este trabajo, que por lo general son personas que quieren llegar directo "al grano", puedan comprender esta teoría de manera muy simple y práctica.

Si una entidad no satisface una determinada regla de normalización, se procede a descomponerla en otras entidades que sí las satisfagan. Esto usualmente requiere decidir qué atributos de la entidad original van a residir en una u otra de las nuevas entidades. La descomposición tiene que conservar dos propiedades fundamentales:

1. No pérdida de información.
2. No pérdida de dependencias funcionales.

A continuación plantearemos las principales formas normales, desde el punto de vista que nos hemos propuesto

PRIMERA FORMA NORMAL (1FN)

Una entidad se encuentra en 1FN si todos sus atributos son simples (no descomponibles), y si no tiene grupos repetitivos.

Un atributo que forma parte del grupo repetitivo es aquel que no permite presentar la entidad en forma *Planar*. Esto quiere decir que no se puede dibujar la entidad en un plano debido a que puede admitir varios valores en una misma instancia de la entidad; esto es en una misma celda.

Para obtener la 1FN debemos proceder de la siguiente manera:

1. Identificar la clave primaria en la entidad compleja
2. Identificar los grupos repetitivos en la entidad compleja.
3. Re-escribir la entidad compleja original sin los grupos repetitivos.
4. Crear una segunda entidad con los grupos repetitivos
5. Añadir la clave de la primera entidad (la original) a la segunda entidad.

SEGUNDA FORMA NORMAL (2FN)

Para pasar una entidad a 2FN, el primer requisito es que ya se encuentre en 1FN; luego se deben eliminar los atributos que tengan dependencias parciales.

Con estos atributos extraídos se deben formar otras entidades. Una entidad se encuentra en 2FN si ya está en la 1FN y además cada atributo no clave es completamente dependiente de la clave primaria.

Dependencia Parcial.

Supongamos que tenemos una entidad como la que se muestra en la siguiente figura, donde la PK es compuesta y está formada por los atributos A y B:

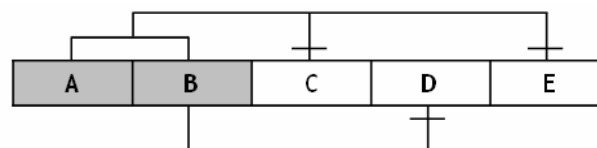


Figura 4.1 Dependencia Parcial.

El atributo C depende funcionalmente de la PK; esto quiere decir que para identificar plenamente el atributo C se necesitan conocer a todos los atributos que conforman la PK (A y B). Lo mismo se aplica para el atributo E.

Por otro lado el atributo D se puede identificar tan solo conociendo el atributo B, sin necesidad de conocer el atributo A. Luego se puede afirmar que el atributo E tiene una dependencia funcional parcial de la PK.

TERCERA FORMA NORMAL (3FN)

Una entidad se encuentra en 3FN si ya está en la 2FN, y además no tiene atributos con dependencia transitiva de la PK.

Para obtener la 3FN se deben seguir los siguientes pasos:

1. Remover los atributos transitivos
2. Formar nuevas entidades con estos atributos y encontrar en estas su clave primaria

Ejemplo Aplicativo

Enunciado

Para observar con mayor claridad la manera de aplicar las formas normales plantearemos un caso práctico, y en él desarrollaremos las diferentes formas normales. Supongamos que al estudiar una realidad de un negocio se nos presenta un documento como el que se muestra a continuación:

Bodega "Todo Barato"				Boleta de Venta	
<i>De Juan Pérez Contreras</i>				Nro. 001-00005347	
<i>Teléfono: 521-5566</i>					
Cliente: Sergio Matsukawa			Código: C567		
Vendedor: Julio Flores			Código: V014		
Fecha: 24-Febrero-2005					
Código	Producto	Unidad	Cantidad	Precio	Sub-Total
P104	Arroz del Norte	Saco	2	90.00	180.00
P098	Leche Gloria	Caja 24 Uni.	5	80.00	400.00
P133	Aceite "El Sol"	Caja 12 Uni.	3	24.00	72.00
Total Venta					652.00

Identificar la Clave Primaria en la Entidad Compleja

Lo primero que debemos hacer es, a partir del documento obtenido de la realidad capturar los atributos que se presentan en este en una entidad compleja en forma de datos, y establecer un nombre apropiado a cada atributo reconocido.

NroBol	IdCli	NomCli	IdVen	NomVen	Fecha	IdProd	NomProd	Unidad	Cant	PU	ST	TV
--------	-------	--------	-------	--------	-------	--------	---------	--------	------	----	----	----

Donde:

PU: Precio ST: Sub-Total TV: Total de la Venta

A continuación determinaremos la Clave Primaria (PK), usando los criterios de Unicidad y de Minimidad reconocemos como PK al atributo **NroBol**. Con el NroBol podemos identificar unívocamente a cualquiera de las diferentes boletas que conforman las instancias de la entidad compleja.

Aplicación de la 1FN

Identificar los grupos repetitivos en la entidad compleja

Luego debemos identificar a los atributos que forman el Grupo Repetitivo, para lo cual utilizaremos un método práctico que consiste en tratar de representar a la entidad compleja como una Tabla con filas y columnas, donde cada fila representa una instancia de la entidad, y las columnas sus atributos.

NroBol	IdCli	NomCli	IdVen	NomVen	Fecha	IdProd	NomProd	Unidad	Cant	PU	ST	TV
5347	C567	Sergio	V014	Ricardo	24-Feb-05	P104	Arroz del Norte	Saco	2	90	180	652
5348	C345	Hugo	V010	Claudia	24.Feb-05	P256	Azúcar Pomalca	Saco	3	70	210	760

Como se puede apreciar los atributos IdProd, NomProd, Unidad, Cant, PU y ST, no permiten representar la entidad compleja en forma planar por que tienen varios datos en la misma celda para una misma instancia, por lo tanto estos atributos forman el grupo repetitivo.

Por ejemplo en la celda que corresponde a la columna IdProd en la primera fila debe estar el dato P104, pero también en esta misma fila deberían estar los datos P089 y P133, ya que corresponden a la misma boleta o sea a la misma instancia; estos datos no se podrían ubicar mas abajo, en la siguiente fila, por que estos lugares corresponden a otra fila ó a otra instancia. Luego estos tres datos deben ubicarse en la misma celda yuxtaponiéndose entre ellos; por lo tanto los datos de este atributo no permiten representar a la entidad en un plano sino más bien en una representación espacial; luego este atributo se dice que forma parte del grupo repetitivo.

Lo anterior también se aplica para los atributos NomProd, Cant, PU y ST.

Rescribir la entidad compleja original sin los grupos repetitivos

La entidad compleja anterior quedaría sin los atributos que conforman el grupo repetitivo. A esta entidad resultante le llamaremos momentáneamente entidad A.

Entidad A:

NroBol	IdCli	NomCli	IdVen	NomVen	Fecha	TV
--------	-------	--------	-------	--------	-------	----

Crear una segunda entidad con los grupos repetitivos

Con los atributos extraídos se forma otra entidad, agregándole la PK de la entidad compleja, de manera que se conserva la relación entre estas dos entidades resultantes. A esta entidad la llamaremos Entidad B.

Entidad B:

NroBol	IdProd	NomProd	Unidad	Cant	PU	ST
--------	--------	---------	--------	------	----	----

Luego tenemos como resultado de aplicar la 1FN a la entidad compleja, las entidades A y B, las cuales se encuentran en primera 1FN, lo cual quiere decir que se pueden representar dichas entidades en *Forma Planar*.

Aplicación de la 2FN

En el caso de la entidad A, podemos observar que la PK está formada por un solo atributo, luego es imposible encontrar alguna dependencia parcial de cualquier atributo que se identifique con la PK.

Podemos concluir entonces que, si una entidad ya está en 1FN, y su PK es simple, (formada por un solo atributo), no tendrá dependencias parciales, luego ya se encuentra en 2FN. O como corolario, solo existirá posibilidad de encontrar dependencias parciales en una entidad, si su PK es compuesta (formada por más de un atributo).

En la entidad B debemos encontrar la PK para luego buscar dependencias parciales. La PK debe cumplir con el criterio de unicidad, graficamos la entidad en una tabla donde cada fila es una instancia:

NroBol	IdProd	NomProd	Unidad	Cant	PU	ST
5347	P104	Arroz del Norte	Saco	2	90.00	180.00
5347	P089	Leche Gloria	Caja 24 Uni.	5	80.00	400.00
5347	P133	Aceite "El Sol"	Caja 12 Uni.	3	24.00	72.00
5348	P256	Azúcar Pomalca	Saco	3	70.00	210.00
...
5380	P104	Arroz del Norte	Saco	4	90.00	360.00
...

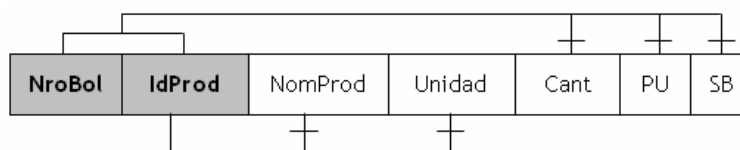
Aquí se puede observar que el **NroBol** se puede repetir en diferentes instancias de la entidad B, esto se debe a que es una Clave Foránea (FK). Pero con solo este atributo no se puede identificar plenamente a una instancia en particular de esta entidad. Por ejemplo si tuviéramos el **NroBol** 5347 tendríamos tres casos que cumplen con este valor. Debemos recurrir a otro atributo adicional para formar la PK.

Un posible candidato podría ser el atributo **NomProd**. Si tenemos el NroBol 5347 y el producto Arroz del Norte, identificamos inmediatamente a la primera fila de la tabla. Tenemos otra fila con el mismo producto, pero pertenece a otra boleta (5380). Por lo tanto la combinación **NroBol + NomProd** cumple con el criterio de Unicidad.

Otra posible combinación sería **NroBol + IdProd**, y como podemos observar también cumple con el primer criterio de Unicidad. Pero además es más pequeña que la primera candidata; por lo tanto también cumple con el criterio de Minimidad.

Finalmente elegimos a la combinación **NroBol + IdProd** como la PK de la entidad B.

A continuación graficamos la entidad identificando su PK, para investigar si existen dependencias parciales.



En el grafico se puede observar que los atributos **NomProd** (nombre del producto) y **Unidad** (unidad en que se comercializa el producto), se pueden identificar solo con conocer el **IdProd** (el código del producto debe ser único por producto). Luego no necesitan de toda la PK para ser identificados, por lo tanto tienen dependencia funcional parcial de la PK.

Para pasar a la **2FN** se deben eliminar los atributos que originen dependencias parciales:

Entidad B1:

NroBol	IdProd	Cant	PU	ST
---------------	---------------	------	----	----

Y con los atributos eliminados tenemos:

Entidad B2

IdProd	NomProd	Unidad
---------------	---------	--------

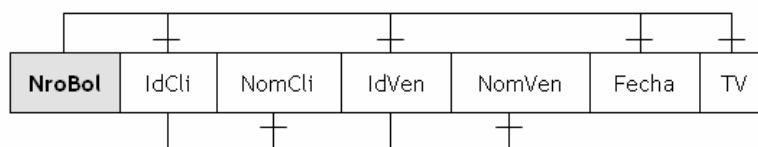
Aquí podemos observar que la entidad B2 también lo conforma el atributo del cual dependían funcionalmente en la entidad B, esto asegura la relación entre las entidades resultantes. Además identificamos al atributo **IdProd** como PK en esta nueva entidad resultante B2.

Hasta este momento, en este ejemplo, tenemos tres entidades resultantes en 2FN: **A**, **B1**, y **B2**.

Aplicación de la 3FN

En nuestro ejemplo anterior, notamos que las entidades B1 y B2 tienen dependencias funcionales directas, luego podemos deducir que no existen dependencias transitivas, por lo tanto estas entidades ya se encuentran en 3FN.

En el caso de la entidad A debemos investigar las dependencias funcionales de sus atributos, para ello trazamos las líneas de dependencia respectivas:



Podemos reconocer dos atributos con dependencias transitivas; **NomCli** (nombre del cliente) se puede identificar con el atributo **IdCli** (código del cliente), a su vez el **IdCli** se puede identificar con el atributo **NroBol** (número de boleta), que es la PK de la entidad; así podemos afirmar que el atributo **NomCli** tiene una *dependencia transitiva* de la PK.

Del mismo modo se puede afirmar que el atributo **NomVen** tiene también dependencia transitiva de la PK.

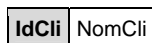
A continuación procedemos a eliminar estas dependencias transitivas de la entidad A y obtenemos las siguientes entidades resultantes:

Entidad A1:

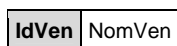
NroBol	IdCli	IdVen	Fecha	TV
---------------	-------	-------	-------	----

A continuación con los atributos extraídos formamos otras entidades.

Entidad A2:



Entidad A3:



Observaremos que, en la entidad A2 se ha incluido también el atributo **IdCli** del cual dependía en la entidad A. Esto asegura la relación entre las entidades resultantes. Además el atributo **IdCli** es ahora PK en la entidad A2.

Del mismo modo se observa que en la entidad A3 la PK es ahora **IdVend**.

Cuando se tiene que las relaciones resultantes solo tienen dependencias directas y totales (obsérvese las líneas de dependencia), podemos afirmar que el proceso de normalización ha terminado.

Como resultado final entonces tenemos en este caso cinco entidades:

A1 A2 A3 B1 B2

A continuación se deben nombrar las entidades resultantes utilizando nombres que reflejen su esencia, como se ilustra a continuación:

- A1** Boleta
- A2** Cliente
- A2** Vendedor
- B1** DetalleBoleta
- B2** Producto

El siguiente paso es construir el Diagrama ER con estas entidades resultantes del proceso de normalización, tal como se ilustra a continuación.

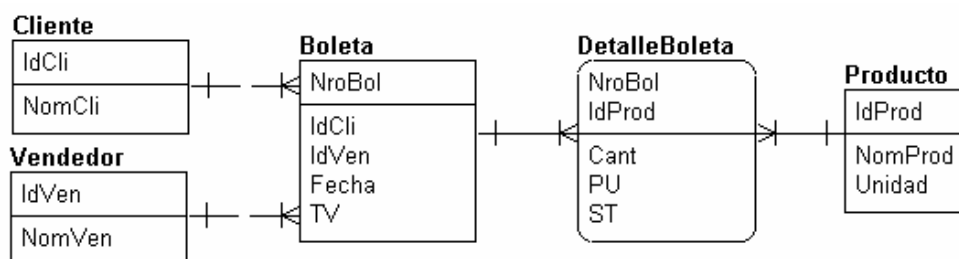


Figura 4.2 Diagrama Entidad-Relación.

Ejercicios

Ejercicio 01

Se tiene el siguiente documento correspondiente a la Planilla de Empleados de la empresa **Fashion House SAC**, obtener la estructura compleja correspondiente y luego aplicar las técnicas de normalización para llegar a un modelo E-R normalizado.

Fashion House SAC							Nro. 2003-09
Planilla de empleados de Agosto del 2003							
Código	Apellidos y Nombres	Sueldo Básico	Sistema Nacional Pensiones	AFP	Aporte AFP	Prest. Salud	Neto
01 - Logística							
95001	Coronel Castillo, Eric Gustavo	3000.00	300.00			210.00	2700.00
95010	Flores Ramirez, Julio	2500.00		INT - Integra	275.00	175.00	2225.00
96014	Marcelo Villalobos, Ricardo	2800.00		HOR - Horizonte	308.00	196.00	2492.00
...
...
02 - Contabilidad							
96601	Matsukawa Maeda, Sergio	3000.00		HOR - Horizonte	330.00	210.00	2370.00
97098	Carrasco Salas, Javier	2800.00		INT - Integra	308.00	196.00	2492.00
94073	Pérez Castro, Carmen	2500.00	250.00			175.00	2250.00
...
...
Totales		####.##	###.##		###.##	###.##	####.##

Ejercicio 02

Normalice el siguiente documento y obtenga su diagrama E-R

Instituto Superior "El Alma del Saber"		<i>Comprobante:</i> 7496	<i>Concepto:</i> Remuneración: Julio-2005
Nro. Planilla: 20030701	Fecha: 31-Jul-2005	Del 01-Jul-2005 Al 31-Jul-12005	
Código: 23456	Nombre: Julio E. Flores Manco	Ocupación: Docente	
DNI: 76893456	L.M.: 6784578587	Nro. Seg. Social: 6576451FLMAJ001	
Fecha Ingreso: 01-Ene-2001	Horas Dictadas: 64	Categoría: B	
AFP: 08 Nueva Vida	CUISPP: 657411EMCTR1		
Ingresos		Egresos	
Rem. Docente	1280.00	AFP. Ap. Obli.	239.36
Boni. Transporte	256.00	AFP. Seg. Inv.	43.52
Refrigerio	320.00	AFP Com. Var.	54.40
Vestuario	128.00		
Mat. Docente	192.00		
	<hr/>		<hr/>
	2176.00		337.28
		<u>Neto: 1838.72</u>	
Aportaciones del Empleador			
Prest. Salud	65.28		
<hr/>			
Director			


Ejercicio 03

Dado el siguiente documento correspondiente a la Ficha de Matricula de un Alumno de EduTec, obtener la Estructura compleja correspondiente y luego aplicar las técnicas de Normalización desarrolladas en clase.

EduTec <i>Educación & Tecnología</i>	FICHA DE MARTICULA	Nº 1568963		
Código: C005	Carrera: Computación e Informática	Periodo: 2005-2		
Código: 2003345	Alumno: Córdova Tello, Javier	Fecha: 02-Agosto-2005		
Código Curso	Nombre Curso	Ciclo	Créditos	Horario
FDP	Fundamentos de Programación	1	4	Mi: 20-22 Horas Ju: 20-22 Horas
MODBD	Modelamiento de Base de Datos	2	4	Ma: 19-22 Horas
LPVI	Lenguaje de Programación Visual	3	5	Mi: 18-20 Horas Ju: 18-20 Horas
Observaciones: Última oportunidad para nivelarse en sus cursos, de lo contrario solo podrá llevar los cursos que debe.				

Ejercicio 04

Dado el siguiente documento correspondiente al estado de cuenta de una tarjeta de crédito de una entidad bancaria, obtener la estructura compleja correspondiente y luego aplicar las técnicas de Normalización para obtener el diagrama E-R.

PeruBank	FoxCard <i>Tarjeta Internacional</i>	 234578962	
Cliente: 152345 Ramos Contreras, Juan Alberto CL Los Tulipanes 570 Los Olivos		Nro. Cuenta: 9768-78563423-98 Nro. Tarjeta: 3456-7645-5678-2365	
CREDITO UTILIZADO	CREDITO DISPONIBLE	LINEA DE CREDITO	LIMITE DISPOSICION EFECTIVO
903.27	496.73	1400.00	1100.00
FECHA CIERRE	DE PAGO MINIMO DEL MES	PAGO TOTAL DEL MES	ULTIMO DE PAGO DIA
30-Nov-2004	83.77	1005.22	08-Dic-2004

Número Operación	Fecha Operación	Descripción	Monto
		Saldo anterior	947.56
1559	05-Nov	Pago a cuenta	-450.00
1560	15-Nov	Metro – Los Olivos	45.37
1561	18-Nov	Ripley – San Miguel	230.35
1562	20-Nov	PlazaVea – Miraflores	129.99
1563	30-Nov	Int. Crédito Financiado	89.45
1564	30-Nov	Impuesto Transac. Financieras	0.50
1565	30-Nov	Seguro Desgravamen	12.00
Deuda Total del Mes			1005.22

Página en Blanco