



UNIVERSIDAD POPULAR AUTONOMA DEL
ESTADO DE PUEBLA

UPAEP
BIBLIOTECA CENTRAL
TESIS
USO UNICAMENTE EN SALA



DISEÑO DE BASES DE DATOS

TESIS PROFESIONAL

Que para obtener el Título de

Licenciado en Sistemas Computacionales

Presentan:

CLAUDIA MORAN MORENO

y

MA. SONIA MENDEZ CADENA

Asesorados por:

Lic. Jaime F. Castillo Rodríguez

H. Puebla de Z., junio de 1996.



UPAEP – Secretaría General

Dirección General de Apoyos Académicos

Dirección del Centro de Recursos para el Aprendizaje y la Investigación.

Biblioteca Central - **Karol Wojtyła**

Tesis Digitales Restricciones de uso:

DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de textos, imágenes, gráficas, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente de donde la obtuvo mencionando el autor o autores involucrados en el documento.

Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



Dedico este trabajo a:

Dios por la oportunidad que me dio de conocer su existencia caminando siempre a mi lado, a El porque cuando siento desfallecer siempre me ha tendido la mano fortaleciendo mi espíritu y mi fe. A ti Señor que siempre escuchas mis oraciones y me amas, te ofrezco cuanto soy.

Mi papá, Medardo Morán Castañeda, porque con tu ejemplo me has enseñado que en esta vida triunfa el que trasciende, aquel que logra avanzar sin detenerse; a ti porque has estado presente cuando te he necesitado compartiendo momentos de alegría y, consolándome y aconsejándome en momentos de tristeza Te respeto y admiro.

Mi mamá, Ma. Esther Moreno Ramírez, gracias por enseñarme a amar a Dios y el mundo que me rodea, por tus desvelos y preocupaciones que muchas noches te impidieron dormir, por tu paciencia y ternura infinita, porque me has entregado tu vida sin reservas ni salvedades y, porque siempre has querido mi felicidad. Bendita seas.

José Luis Sánchez Juárez, debido a que has cambiado mi mundo con tu amor, ayudándome a crecer y ser mejor cada día; a ti que te interesaste en conocer mis sueños y en compartir mis ilusiones; porque un día nuestras miradas se cruzaron y desde entonces miraron hacia la misma dirección, gracias por tu apoyo y confianza. Te amo.

Mi hermana Ma. Esther y mi cuñado Gumercindo Ignacio, por su apoyo, consejos y por la confianza que en mi han depositado. Les manifiesto mi cariño.

Mis hermanos Blanca Gloria, Aydee y Medardo, porque de cada uno siempre he tenido algo que aprender. Los quiero mucho.

Mis sobrinos Manuel Ignacio, Jacqueline Rethsee, Sthepany Lisset y Jennyfer Aydee porque cuando estoy a su lado, a través de su inocencia y ternura vivo momentos inolvidables. Los adoro.

*Miguel Angel Juárez Andrade, por tu amistad y apoyo incondicional
Sonia Méndez Cadena, gracias por tu amistad.*

Claudia Morán Moreno

Agradezco a:

Dios porque me dio la vida y me acompaña en todo momento, ya que sin El no sería nadie.

Mi papá, Román Méndez Domínguez, por ser un ejemplo de superación, por su amor, apoyo moral y económico.

Mi mamá, Gloria Cadena de Méndez, por su comprensión, paciencia y amor.

A Román, Fer, Bony y Edgar porque como hermanos siempre estaremos juntos.

La familia Cadena Robles porque me han acompañado en cada escalón de mi superación.

La familia Cadena Gómez por haber compartido momentos agradables y por su ayuda incondicional.

A Jaime Javier Hernández Morales por sus consejos, ayuda y por ser un gran amigo.

La familia Morán Moreno por sus facilidades prestadas y convivencias agradables.

Sonia Méndez Cadena.

RECONOCIMIENTOS

Los autores manifiestan su agradecimiento a todas las personas que contribuyeron a nuestra preparación física y mental aportando conocimientos, comentarios y sugerencias para ser siempre mejores y que jugaron un papel muy importante en nuestro desarrollo.

Al ingeniero José Ma. Enrique Bedolla Cordero por el interés y dedicación que ha demostrado en el desarrollo y crecimiento de la escuela de Sistemas Computacionales.

Queremos expresar nuestra gratitud por la ayuda especial que durante el tiempo de la elaboración de esta tesis recibimos del licenciado Jaime F. Castillo Rodríguez cuya guía, orientación y conocimientos nos ayudaron en esta etapa de difícil proceso manifestamos nuestra admiración y respeto.

Al licenciado Jaime Javier Hernández Morales le agradecemos su colaboración por facilitarnos el equipo de hardware necesario para la impresión final de este trabajo.

CONTENIDO

	<i>página</i>
INTRODUCCION	1
Capítulo 1. FUNDAMENTOS DE BASE DE DATOS	3
1.1. ¿Qué es una base de datos?	3
1.1.1. Clasificación de las Relaciones	4
1.2. El DBMS	5
1.3. Modelo de una base de datos relacional	6
1.3.1. Tablas	6
1.3.2. Llave primaria	7
1.3.3. Llaves externas	9
1.4. Redundancia de datos	11
1.5. Sistema de base de datos	12
1.6. Sistema de Información	13
1.7. ¿Por qué el diseño de la base de datos también es importante?	13
P r o b l e m a s	14
Capítulo 2. MODELANDO Y DISEÑANDO BASES DE DATOS	19
2.1. Modelando una entidad relación (E-R): algunos antecedentes	19
2.2. Diagrama E-R y sus componentes	20
2.2.1. Entidades, atributos y relaciones	20
2.2.2. Conectividades y cardinales	21
2.2.3. Relaciones: opcional y obligatoria	24
2.2.4. Entidades compuestas	25
2.2.5. Entidades débiles	27
2.2.6. Supertipos y subtipos	28
2.2.7. Diseño con atributos	30
2.3. Poniendo las piezas juntas: diseño de base de datos simples	32
2.4. El diseño del proceso	36
2.4.1. Descripción de operaciones	36
2.4.2. Diseño inicial	42
2.4.3. Verificación del diseño	44
2.5. Exclusiones del diseño	47
P r o b l e m a s	49
Capítulo 3. NORMALIZACION	53
3.1. Dependencia	53
3.2. Formas normales	55

	<i>página</i>
3.3. Diagramas de dependencia y conversiones forma normal	56
3.3.1. Conversiones de 1NF a 2NF	57
3.3.2. Conversiones de 2NF a 3NF	59
3.3.3. Forma normal Boyce-Codd	60
3.4. Denormalización: un mundo de advertencia	62
3.5. Normalización y modelando E-R	64
Problemas	66
Capítulo 4. ESTUDIO DE CAMPO, ENCUESTAS Y RESULTADOS	70
1.1. Definición del Problema	70
1.2. Investigación de campo y resultados estadísticos	70
1.3. Explicación general	76
Capítulo 5. DESARROLLO E IMPLEMENTACION DEL SISTEMA TUTORIAL	77
5.1. Reglas del Negocio	77
5.2. Diagrama Entidad-Relación	78
5.3. Diccionario de datos	79
5.4. Diagrama de Estructura	85
5.4.1. Representación del diagrama de estructura	85
5.4.2. Descripción del contenido del diagrama de estructura	86
5.5. Diagrama de Transición de Estados	88
5.5.1. Nomenclatura utilizada	88
5.5.2. Diccionario de menús	88
5.5.3. Representación de los Diagramas de Transición de Estados	89
5.6. Manual de Usuario	93
5.6.1. Sección mantenimiento catedráticos	93
5.6.2. Sección generar un nuevo tutorial	95
5.6.3. Sección uso de tutoriales	97
CONCLUSION	99
BIBLIOGRAFIA	102
GLOSARIO	104

INTRODUCCION

La historia de la informática es también la historia del progreso en los sistemas de comunicación entre el hombre y la máquina, en donde la información es considerada como el soporte de los conocimientos y las comunicaciones. Es así como aparecen lenguajes de programación provistos de metodologías apropiadas que proporcionan herramientas para el diseño e implementación de los sistemas; el problema se complica en tanto que cada sistema es de naturaleza y estructura propia, además de que cada analista los plantea de diferente forma para realizarlos. A pesar de ello todos buscan un común denominador: calidad, eficiencia y productividad.

Hablamos entonces de que probablemente no haya un tema tan diversificado como el de los sistemas, pero nos enfocaremos a los sistemas administradores de base de datos por medio de los que podemos organizar, buscar y presentar información rápidamente; es decir, un usuario desea información de buena calidad, en el momento en que la necesita y al menor costo posible.

Un procedimiento laborioso al formar una base de datos relacional es identificar las relaciones, cada una deberá componerse de un grupo de atributos, de manera que sea esencialmente independiente de las otras bases de datos e incorporar los atributos a ésta para que se formen relaciones óptimas; pero para la construcción de este conjunto de relaciones se requiere de un conocimiento adecuado de las técnicas de diseño de base de datos. Teniendo en cuenta esta consideración planteamos el objetivo principal de este trabajo que es el de dar a conocer diferentes técnicas para el diseño de base de datos, a través de un *Tutorial de Base de Datos*, generado a partir de un *Sistema Generador de Tutoriales* que se desarrolla en el presente trabajo.

Para el estudio de esta tesis, se requiere que el lector tenga conocimientos de computación en ambiente Windows; si no está familiarizado con él, le sugerimos que consulte primero el manual de Windows para conocer sus características antes de usar este software. Para su aprendizaje se divide en cinco capítulos, en donde la información de los tres primeros son una síntesis de los conceptos de Peter Rob y Treyton Williams de la Universidad Central de Tennessee, mencionados en su libro *Database Design and Applications Development with Microsoft Access 2.0*; y a continuación se describen.

El primero de ellos introduce conceptos fundamentales de bases de datos que servirán para identificar los elementos precisos que han de formar una estructura, garantizando el correcto análisis y diseño de la base de datos.

El segundo capítulo plantea cómo modelar y diseñar bases de datos por medio de diagramas y sus componentes. En el caso de una estructura de base de datos relacional los usuarios pueden utilizar archivos, seleccionar campos de ellos y asociarlos fácilmente; las características de la base de datos relacional debe cumplir con ciertas condiciones para que la computadora efectúe operaciones sobre ella, donde su desarrollo comienza con el punto de vista del usuario y cuya perspectiva es una representación gráfica de los datos que se quieren para tomar una decisión o contestar una pregunta.

El concepto de normalización es presentado en el capítulo tres, técnica utilizada para obtener estructuras de datos eficientes y optimar un sistema de base de datos relacional, generando tablas correctas con la ventaja de evitar la dependencia de inserciones, actualizaciones y borrados con los que

se evitan redundancias e inconsistencias; además reduce la reestructuración de la base de datos debida a la introducción de nuevos datos, y no se establecen restricciones artificiales en la estructura de los datos, con lo que aseguramos estructuras de tablas que contienen requerimientos específicos y que controlan la redundancia de los datos.

El estudio de los tres primeros capítulos son tan importantes que no se pueden omitir, de manera que no se justifica la exclusión de alguno de ellos, además de que el texto sigue un orden lógico y no arbitrario, por lo que se recomienda la lectura en forma ordenada para comprender cada concepto y manejarlo con mayor facilidad. Al final se adjunta una lista de preguntas y ejercicios cuya práctica reforzará los conocimientos adquiridos.

En el cuarto capítulo se presenta la definición del problema, la investigación de campo, así como el estudio estadístico llevado a cabo con su respectiva interpretación, por medio del que conoceremos el porcentaje de conocimientos de base de datos que poseen los alumnos de los últimos semestres de la Licenciatura de Sistemas Computacionales de la Universidad Popular Autónoma del Estado de Puebla.

El quinto y último capítulo expone el desarrollo e implementación del *Sistema Generador de Tutoriales* en el que se aplican los conocimientos teóricos y las técnicas de análisis y diseño de base de datos adquiridos. Este sistema tendrá un ejemplo de aplicación con la información de los tres primeros capítulos de la tesis formando un *Tutorial de base de datos*.

Se concluye con la adición de un glosario de los vocablos técnicos mencionados a lo largo de la tesis.

Existe además un *anexo* formado por tres capítulos, para el cual recomendamos que el lector domine los términos de las tres primeras lecciones de la tesis debido a que hará más fácil su comprensión y aprendizaje. El primero de ellos se fundamenta en la información que la Corporación Microsoft presenta en los manuales de Microsoft Access para su aprendizaje, así como en el libro Microsoft Access 2 de Mariano Flores Gutiérrez; el segundo y tercer capítulo condensa los apuntes de Sandra Dewitz y Michael Olson de su libro titulado *Semantic Object Modeling with Salsa*. La explicación del contenido es la siguiente:

Los conceptos sobre Microsoft Access y su entorno se describen claramente en el primer capítulo. En él se explica, mediante sencillos ejemplos, la forma de diseñar una base de datos y aprovechar todas sus características, como su poder gráfico, capacidad de consulta, administración de datos y demás herramientas de diseño.

El segundo capítulo presenta un panorama general de cómo modelar objetos semánticos y la sintaxis del lenguaje modelador usado para dibujar diagramas de objetos semánticos. Los conceptos son explicados mediante ejemplos y una vez dominados sirven para construir el modelo.

Y el tercer capítulo titulado *modelando-datos usando Salsa*, se distingue por describir las características del Registrador Salsa como son sus funciones, fórmulas, modelos, herramientas de validación para la documentación, creación de reportes, incluyendo las ventajas de cómo transformar un modelo de objetos semánticos al diseño de una base de datos relacional; también presenta los recursos de Salsa como herramienta gráfica usada para dibujar diagramas de objetos semánticos y generar los esquemas correspondientes de una base de datos.

En resumen, los problemas fundamentalmente siguen siendo los mismos, pero es preciso que los sistemas generados cuenten con un análisis y diseño adecuado, que proporcionen la información que el usuario requiere para la toma de decisiones.

Capítulo 1. FUNDAMENTOS DE BASE DE DATOS

Dado el objetivo principal de este trabajo, que es el de dar a conocer diferentes técnicas para el diseño de base de datos, en este capítulo se señalan los conceptos de las bases de datos necesarios para diseñar y organizar una base de datos maximizando su utilidad y minimizando la posibilidad que ésta contenga errores.

Además de entender los conceptos, hay dos razones para que conozca el vocabulario básico de las bases de datos: Primero, evitará confusión acerca de detalles del diseño de la base de datos, cuando use técnicamente un lenguaje preciso. Segundo, conocerá como responder cada pregunta como "¿quiere generar en Access una llave primaria? o ¿imponer una integridad referencial?".

El material de este capítulo está basado en el trabajo de Peter Rob y Treyton Williams, (1995).

1.1. ¿Qué es una base de datos?

En sentido básico, una base de datos es un depósito de almacenamiento electrónico de datos. Más precisamente es una estructura que contiene un conjunto de entidades y relaciones entre ese conjunto de entidades.

Una entidad es una persona, lugar, cosa o evento en la cual se quieren guardar hechos conocidos como **datos**. Por ejemplo, si es un gerente, probablemente quiera guardar datos acerca de cada persona empleada en la organización. Cada uno de esos empleados es una entidad. Cada entidad tiene características identificables conocidas como **atributos**. Por ejemplo, los atributos de la entidad EMPLEADO son probablemente incluir su nombre, apellido, una inicial, el número del seguro social, su número telefónico y otra información pertinente.

Un **conjunto de entidades** es una colección de entidades relacionadas. En un conjunto todas las entidades están nombradas. Para ilustrar, el conjunto de entidades nombrada EMPLEADO contiene sólo datos acerca de los empleados; éste no debe contener datos acerca de otra cosa. Dentro de las entidades, el conjunto de entidades del empleado están relacionadas por el hecho de que todos ellos son empleados. Similamente una biblioteca puede guardar un conjunto de entidades llamada LIBRO. Todas las entidades de LIBRO están relacionadas por el hecho de que todos ellos son libros. Evidentemente, sería inapropiado almacenar los datos de un cliente o colocar los datos de ingeniería en un conjunto de entidades EMPLEADO o almacenar los datos de un empleado de una librería en un conjunto de entidades de libro.

La relación entre un conjunto de entidades reflejan hechos que, dentro de una base de datos, muchas entidades son probablemente la interacción con otras entidades. Tales interacciones son descritas fácilmente si usamos verbos activos. Por ejemplo, un cliente pide un libro de la librería, la

relación entre *libro* y *cliente* es pedir. Similarmente, la relación entre *cliente* y *producto* es comprar (un cliente compra un producto), mientras la relación entre *piloto* y *avión* es volar (un piloto vuela un avión).

1.1.1. Clasificación de las Relaciones

En el capítulo dos podrá darse cuenta que el diseño de la base de datos requiere que defina usted todas las relaciones entre el conjunto de entidades de la base de datos. Específicamente, siendo hábil e inteligente para clasificar cada relación como uno a uno, uno a muchos, o muchos a muchos. Exploraremos cada una de las anteriores:

Una **relación uno a uno**, escrita como 1:1, existe si una entidad está dentro de un conjunto de entidades. (Uno puede corresponder a una sola entidad en un conjunto de entidades B y una entidad en un conjunto de entidades B puede corresponder a una entidad dentro de un conjunto de entidades A). Ejemplo: Un gerente administra un departamento y un departamento es administrado por un solo gerente.

Una **relación uno a muchos**, escrita como 1:M, existe si una entidad en un conjunto de entidades A puede corresponder a una sola entidad dentro del conjunto de entidades B, pero una entidad en un conjunto de entidades B puede corresponder a muchas entidades del conjunto A. Ejemplo: Un trabajador es empleado en un solo departamento pero, un departamento tiene muchos trabajadores.

Una **relación muchos a muchos**, escrita como M:N, existe si una entidad en un conjunto de entidades A puede corresponder a muchas entidades en un conjunto B, y una entidad en un conjunto de entidades B puede corresponder a muchas entidades de un conjunto A. Ejemplo: un estudiante toma muchos cursos y cada curso es tomado por muchos estudiantes.

Es fácil determinar el tipo de relación entre el conjunto de entidades, se requiere de contestar dos preguntas:

- 1.-¿Una entidad puede en un conjunto de entidades A corresponder a más de una entidad en un conjunto de entidades B?
- 2.-¿Una entidad puede en un conjunto de entidades B corresponder a más de una entidad en un conjunto de entidades A?

Si las dos respuestas son "no" indica una relación 1:1. Una respuesta "si" y una respuesta "no" indica una relación 1:M. Y dos respuestas "si" indican una relación M:N.

Utilizando las reglas anteriores, puede determinar la relación entre las entidades TIENDA y REGION preguntando: ¿Puede una tienda estar localizada en más de una región? (No). ¿Una región puede localizar muchas tiendas? (Si). Puesto que un "si" y un "no" indican que hemos descubierto una relación 1:M entre TIENDA y REGION. Específicamente, una tienda puede estar localizada en una sola región, pero una región puede tener muchas tiendas.

El conjunto de entidades y sus relaciones son específicas y definidas por usted. En el capítulo dos cuando lea como usar los componentes de las bases de datos en un buen diseño de esta

misma, podrá darse cuenta de definiciones propias de las relaciones entre entidades y son importantes para la correcta operación de la base de datos.

1.2. El DBMS

Sistema de administración de bases de datos (DBMS), es el software que adquiere usted, como es Microsoft Access. El DBMS administra y relaciona la base de datos almacenada. Afortunadamente, esto es de manera transparente, puesto que la complejidad de las acciones son suyas. Por ejemplo, cuando implementa el diseño de una base de datos incluye un apropiado conjunto de entidades y sus relaciones, El DBMS crea un *diccionario de datos* a través del cual administra transacciones entre los datos que son tomados de un lugar de la base de datos.

Un **diccionario de datos** es una estructura que almacena "meta datos", esto es datos acerca de los datos. Los meta datos incluyen tipos de datos como (numéricos, caracteres, lógicos, etcétera), requiere de un espacio físico para, conjunto de entidades nombradas, los nombres de los atributos encontrados en cada conjunto de entidades y la relación entre éstas.

^{Todo} Ningún cambio que haga en la base de datos es registrado automáticamente en el diccionario de datos, de esta manera le libera a usted la dificultad y el consumo de tiempo, al guardar el seguimiento de todo lo que realice. Desde el DBMS puede encausar todos los datos que requiera a través de un diccionario de datos, ninguna de sus aplicaciones (reportes, preguntas, formas, etcétera) necesita contener información relacionada con las características de los datos. A corto plazo, los DBMS permiten datos independientes al nivel de aplicación.

Además de efectuar importantes funciones en el diccionario de datos un DBMS lleva a cabo de manera transparente servicios que son propios de la base de datos. Idealmente un DBMS:

- Cuida todos los detalles de los datos almacenados, produciendo apropiadas (y muy complejas) estructuras físicas requeridas para cada tarea.
- Transformar su lógica de los datos que requiere de estructuras físicas para los datos. Esencialmente el DBMS trabaja de una forma parecida a un traductor. Por ejemplo, si usted quiere conocer cuántos empleados tienen un salario superior de \$30,000 en un año, el DBMS cambia su lógica requerida por un código físico complejo y pasa a través de un diccionario de datos para acceder los datos del empleado, efectuado por un filtro de acciones (sólo empleados con salarios arriba del estado deseado), y transforma estos resultados en algo que pueda entenderse.
- Le permite especificar la *validación de reglas* dirigiendo sus datos o imponerle esas reglas. Como las indicaciones del nombre, validación de reglas y uso para checar la validación de los datos. Por ejemplo, si sólo acepta pagar de forma CC para tarjeta de crédito y CK para cheque, una regla de validación puede ser usada específicamente tan sólo para que CC o CK sea aceptada y que alguna otra entrada sea rechazada.

- Crea acceso seguro de datos a través de una *contraseña de protección* (password) y otros dispositivos.
- Contribuye a la seguridad física de los datos a través de respaldos y recuperación de procedimientos.
- Efectúa traducciones de las estructuras de datos permitiendo que utilice datos de otras bases de datos o de hojas de cálculo.
- Produce estructuras complejas que permiten múltiples accesos a los datos.

En otras palabras, cuando adquiere su DBMS podrá utilizar un número increíble de servicios en una base de datos que permite que usted produzca prácticas y aplicaciones útiles en la bases de datos.

1.3. Modelo de una base de datos relacional

Los datos necesitan ser desarrollados sobre el tiempo, diferentes modelos de las bases de datos son desarrollados hasta encontrar lo que se necesite. Se debe tener claro: ¿cuál es el fin de la base de datos?, ¿cuáles son las relaciones necesarias para un modelo de base de datos? o teóricamente ¿qué queremos hacer?

E. F. Codd en 1970 su manera de ganar fue debido a unos factores, particularmente su concepto de realizar las consultas (la habilidad de contestar preguntas al momento.)

Fundamentalmente, en un modelo de una base de datos relacional, la construcción básica de las partes de la base de datos son implementadas a través de relaciones. Las relaciones o secuencia de lo que deseamos realizar no debe ser confundida con las relaciones; en su lugar, representa construcciones derivadas de una teoría matemática fija que proporciona modelos teóricos underpinning.

Afortunadamente, las relaciones están representadas conceptualmente por *tablas*, por eso eliminamos la necesidad de conocer detalles matemáticos. El medio ambiente de la base de datos relacional está dirigido por el *sistema de administración de bases de datos relacionales (RDBMS)*.

1.3.1. Tablas

Una *tabla* es una matriz en la cual los renglones corresponden a entidades y las columnas corresponden a atributos de las entidades. En otras palabras, la tabla contiene un conjunto de entidades. Por lo tanto, los términos tabla y conjunto de entidades son frecuentemente usados de manera intercambiable. La figura 1.1. muestra un ejemplo de la tabla relacional, compuesta por cinco entidades (renglones) y tres nombres de atributos (columnas). Los renglones en las tablas relacionales son referidos como **tuplas**.

Las tablas son solamente construcciones lógicas. Los datos encontrados en la tabla están escritos sobre el disco en forma tabular. Sin embargo, desde el RDBMS cuida de todos los detalles físicos. No necesitamos tratar de nuestras estructuras físicas. Algo importante es que la gente encuentre representaciones lógicas para visualizarlas.

Figura 1.1. Tabla de una base de datos relacional.

Tabla: DEPARTAMENTO		
DEPT COD	DEPT DESCRIP	DEPT LOCALIZ
1	Mercadotecnia	Walter Mitty Bldg. 304
2	Nómina	Bluehill Center 201
3	Personal	Annex 105A
4	Producción	Fromm Bldg. 325
5	Recursos	Faraday Center 1297

Fuente: Rob, 1995

Muchos manuales de software contienen terminología que toman de estructuras anteriores. Por ejemplo, "archivo" o "fichero" es algunas veces usado para indicar "tabla"; "registro" es usado para indicar "tupla" o "renglón"; "campo" es usado para denotar "atributo" o "columna", etcétera. Aunque cada terminología es técnicamente incorrecta en el medio ambiente de una base de datos relacional, muestras de entendimiento es si está escribiendo en su manual de software el siguiente resumen:

Terminología relacional

conjunto de entidades = tabla = relación
entidad = tupla = renglón
atributo = columna

Terminología del sistema de archivo

archivo
registro
campo

1.3.2. Llave primaria

Cada renglón de la tabla (entidad) debe tener únicamente una identificación. Por ejemplo, un empleado. Cada identificación única es conocida como **llave primaria**. La razón de usar un número del empleado es porque es más simple que un nombre: el apellido sirve para este propósito ya que hay potencialmente muchos empleados con nombres similares José, por ejemplo. Incluso la combinación del apellido y nombre no pueden hacer el trabajo -es potencialmente menor, José y Luisa Sánchez. (Note que la llave primaria de la figura 1.1. de la tabla departamento es DEPT_CODIGO).

Idealmente una llave primaria consiste de un solo atributo, pero algunas veces es una combinación de múltiples atributos que deben ser usados para producir una identificación única. Una llave primaria compuesta por una combinación de atributos es conocida como **llave primaria compuesta**. La llave primaria compuesta requiere de ser clara, cuando mire una factura simplificada, mostrada en la figura 1.2., note que el número de la factura por si mismo no es identificado para cada factura que entre, porque la factura lista cuatro diferentes artículos. Sólo usando la llave primaria compuesta el número de la factura 132245 y el código del artículo 2085V puede que dirijamos la identidad para especificar la entrada "renta del avión, 2085V" en esta factura.

En el medio ambiente de una base de datos relacional, los cuatro artículos de la factura figura 1.2. son conocidos como líneas de la factura, probablemente son almacenadas en una tabla como es mostrado en la figura 1.3. Note que no toda la información de la factura, que es mostrada en la figura 1.2., es almacenada en esta tabla: descubrirá en las secciones 1.3.3. y 1.4. del por qué los precios de los artículos y las descripciones están almacenados mejor en otra tabla. (Note también que la tabla mostrada en la figura 1.3. representa datos de dos facturas: 132245 y 132246).

Figura 1.2. Muestra de una factura

Factura Número: 132245					
Código del Artículo	Descripción	Unidades	Precio por unidad	Cargo	
FI/COMM	Instructor de vuelo, comercial	1.2	15.00	18.00	
2085V	Renta del avión, 2085V	1.2	65.75	78.90	
CS/11814	Gráfico, 12 pulgadas en plástico	1.0	16.95	16.95	
CS/11743	Mapa local	1.0	6.95	6.95	
				Subtotal	120.80
				Iva 8%	9.66
				TOTAL	130.46
Pago: tarjeta de crédito (VISA)				Pago	\$130.46
				Saldo	\$0.00

Fuente: Rob, 1995

Figura 1.3. Ejemplo de factura compuesta por una tabla de líneas.

Nombre de la tabla: INV_LINEA		
INV_NUM	ART_COD	INV_CANT
132245	FI/COMM	1.2
132245	2085V	1.2
132245	CS/11814	1.0
132246	CS/11743	1.0
132246	FI/PVT	0.9
132246	2085V	0.9

Fuente: Rob, 1995

Nosotros tenemos en negrita INV_NUM y ART_COD en la figura 1.3 INV_LINEA de la tabla para clasificar los componentes de la llave primaria. Tenga en mente que una llave primaria, no es materia de esta composición, es una identificación única para todos los valores del atributo, para cada una de las entidades (renglones). Por lo tanto la llave primaria es única. La llave primaria de la tabla debe ser única y es desplegada como entidad referencial.

Todos los valores de la llave primaria deben ser únicos, una llave primaria nunca debe contener entradas de valor nulo o nulos. Nulo es carencia de valor, cero o blanco; cuando usamos la barra de espacios del teclado esto no es un nulo. Nulo significa que no hay entrada en todo. El problema con los nulos es que un nulo puede significar que no hay ninguno de un número de cosas:

1. Este puede representar algo conocido, pero abstenerse de valor.
2. Este puede representar una condición "no aplicable".
3. Este puede representar un atributo desconocido.

El valor de un atributo nulo puede crear problemas cuando los atributos no son parte de la llave primaria, porque pueden crear problemas de apuntador cuando las funciones del sistema son usados y cuida pequeños detalles buscando así los datos correctos. Por lo tanto si pudiera evitar nulos cuando no sean atributos de llaves primarias que pudieran ser. Sin embargo, evitar esto no siempre es posible o apropiado. Por ejemplo, hay personas que tienen un nombre y sin iniciales, así es que un nulo podría ser una "entrada" apropiada para esta misma.

1.3.3. Llaves externas

Donde mostramos la factura en la figura 1.2. ¿obtuvimos precio y descripción de los datos? Nosotros recibimos esta información del artículo mostrado en la figura 1.4., observará que las tablas INV_LINEA y ARTICULO están conectadas o unidas (join) a través de un atributo común llamado ART_CODIGO. Específicamente, el ART_CODIGO en la tabla concuerda con el ART_CODIGO en la tabla de ARTICULO. Por ejemplo, utilizando la figura 1.4. como guía, podemos crear de la figura 1.2. una tercera factura siguiendo este camino:

Fuente (tabla de la figura 1.4.)	INV_LINEA	ARTICULO	ARTICULO
	CS/11814	Gráfico, 12 pulgadas en plástico	16.95

Los artículos de la figura 1.2. son generados cuando el vendedor da entrada a los valores a través de un dispositivo de entrada como es el teclado, y la figura 1.2. cambia porque se calcularon las unidades de entrada al multiplicarlas con el precio recibido. En este caso el cambio no es almacenado en la tabla. Sin embargo, el atributo se calculó o derivó de otros atributos que pueden ser almacenados en la tabla como un **atributo derivado**. Que una tabla contenga o no los atributos derivados es en gran parte a una función de usar requisitos finales.

Figura 1.4. Llaves externas y primarias en una pequeña base de datos.

INV NUM	ART COD	INV CANT
132245	FI/COMM	1.2
132245	2085V	1.2
132245	CS/11814	1.0
132246	CS/11743	1.0
132246	FI/PVT	0.9
132246	2085V	0.9

ART CODI	ART DESCRIPT	ART PRECI
CS/11814	Gráfico, 12 pulgadas en plástico	16.95
2085V	Renta del avión, 2085V	65.75
FI/PVT	Instructor de vuelo, privado	12.00
CS/11743	Mapa local	6.95
FI/COMM	Instructor de vuelo, comercial	15.00

Fuente: Rob, 1995

A pesar de que tenemos, en algo, simplificado el mundo real del entorno de la factura, ésta es un ejemplo del rendimiento que puntos esenciales del medio ambiente relacional puede juntar o enlazar (join) tablas a través de atributos comunes. Un atributo en una tabla es referido por referencias de otras tablas (así ponemos el campo para unir las) es conocido como **llave externa**. Si examina la tabla 1.4., note especialmente que la llave externa en una tabla debería ser la llave primaria en la tabla para el cual éste apunte. Por ejemplo, ART_CODIGO es la llave externa en la tabla INV_LINEA y es la llave primaria en la tabla ARTICULO. (Note que el valor CS/11814 es la llave externa de la tabla INV_LINEA y apunta al valor de la llave primaria CS/11814 de la tabla ARTICULO).

NOTA. Puede preguntarse que por qué estamos refiriéndonos al apuntador a cada detalle de los hechos que pueden unir tablas a través del uso de la llave externa. Aprenderá en esta sección 1.4. que esta clase de enlace producen algunas eficiencias importantes y ventajas en la integridad de los datos. De hecho, la unión relacional es la llave que da fuerza a la base de datos relacional.

El valor de la llave externa está formado por dos requisitos importantes:

- unir los valores de la llave primaria en la tabla(s) para el cual ellos apuntan o
- ser nulo

Si las llaves externas encuentran esos requisitos, las tablas en la base de datos tienen una **integridad referencial**. La importancia de la integridad referencial se aplica fuertemente y puede ser exagerado. Por ejemplo, si intenta unir un CLIENTE a cliente representante de ventas (tabla VENTAREP), esto definitivamente no es conveniente asignar un código a representante de ventas en la tabla CLIENTE en donde no un apuntador. De otra manera, si un cliente no tiene asignado todavía un representante de venta, él o ella, el código de representante de venta en la tabla CLIENTE podría ser nulo.

Nota: Aunque un nulo satisface la referencia requerida, muchos diseños de bases de datos usualmente intentan evitar nulos en todo lo posible. Así, en lugar de la asignación nulo al código representante de ventas del cliente, que habíamos asignado, asignarían un código especial indicando "no representante de ventas". Por ejemplo, se podría usar el código 999 mostrado en la figura 1.5.

Figura 1.5. Usando un código ejemplo para evitar nulos

Tabla: CLIENTE				
CLI COD	REP COD	CLI APELL	CLI NOMB	...etcétera
143256	451	Rafferty	Carlos	
121458	213	Washington	Luisa	
221347	999	Lahey	Roberto	
176699	326	Katingal	Susana	

Tabla: VENTAREP				
REP COD	REP OFICINA	REP TEL	REP APELL	...etcétera
999	NINGUNO	NINGUNO	NINGUNO	
451	KL-506A	899-1234	Olingo	
213	KL-114C	899-2314	Smithson	
326	GR-100F	899-4456	Rodríguez	

Fuente: Rob, 1995

Pero la llave primaria juega una regla crucial en una base de datos relacional, desarrollando para la gente que tenga problemas de recordar sus valores, por ejemplo probablemente recuerda el número del cliente 3121279; ¿cuándo llamar para enviar una orden de negocios?. Por esta razón, un "humano orientado" busca la llave para ser usada. Cada vez que la llave busca, conocida como **llave secundaria**, estaría compuesta de una combinación de atributos para la gente que probablemente recuerda. Por ejemplo, un cliente probablemente recuerda el apellido y los últimos cuatro dígitos del número telefónico, o el código postal y el apellido.

Una llave secundaria puede producir sólo un registro. Casi siempre, ésta será fácil en llegar a un registro en específico cuando sea listado por unos cuantos. Por ejemplo, una venta de una región puede contener miles de gentes quienes se apelliden Smith, pero serían unos cuantos llamados Smith con un código postal 37129. Por lo tanto, la llave secundaria "37129smith" puede ser una llave adecuada para buscarlo.

1.4. Redundancia de datos

Volviendo hacia atrás por un momento (backtracking). Ahora conoce las reglas básicas del juego, llaves externas, y conoce las tablas relacionales unidas. Pero ¿por qué son importantes las tablas relaciones unidas? La respuesta a esta pregunta está más adelante. Lo habitual para unir las tablas con la base de datos relacional, con un desarrollo reducido, es la redundancia de datos. Por ejemplo, suponiendo que guarda toda la información compartida entre las dos tablas mostradas en la figura 1.4 sólo una tabla se muestra en la figura 1.6. Cuando examine el contenido de la tabla, algunas problemas mayores llegan a ser obvios rápidamente.

Figura 1.6 Redundancia de datos

Tabla: FACTURA				
INV NUM	ART COD	DESCRIPCION	ART PRECIO	ART CANT.
132245	FI/COMM	Instructor de vuelo, comercial	15.00	1.2
132245	2085V	Renta del avión, 2085V	65.75	1.2
132245	CS/11814	Gráfica, 12 puigadas en plásticos	16.95	1.0
132245	CS/11743	Mapa local	6.95	1.0
132246	FI/PVT	Instructor de vuelo, privado	12.00	0.9
132246	2085V	Renta del avión, 2085V	65.75	0.9

Fuente: Rob, 1995.

Por ejemplo, suponiendo que quiera hacer una corrección a la descripción y al precio de la "renta del avión" tiene que enviarla a sus clientes, tendría que hacer correcciones en cada una de las descripciones de la renta del avión y el precio ocurrente en la tabla. Un aumento en el precio real, significaría miles de correcciones. Si requiere hacer correcciones en unos cuantos renglones de la tabla, tiene que concluir con datos inconsistentes porque hay ahora dos versiones de algunas entidades para entradas y otras en conjunto.

En cambio, en el medio ambiente con lo que la pequeña base de datos muestra en la figura 1.4. cualquier cambio en el precio del producto o en la descripción tendría que ser de hecho una sola vez y en un solo lugar. El hecho es que la redundancia, figura 1.4, es la existencia de la llave externa y ésta aparenta ser cambiada, porque también ayuda a la llave primaria en la tabla ARTICULO. En conclusión, un buen diseño relacional de una base de datos hace menor a la redundancia.

La redundancia de datos no es un problema fundado en tener que hacer múltiples correcciones a los datos cuando ocurran errores o cuando los cambios fueron hechos, a pesar de que el problema es substancial. Esto puede ser fácil al mandar un reporte de errores u omisiones. Supongamos que falla al encontrar todos los registros en el cual se están haciendo correcciones o cambios ¿qué debería hacer?, ¿conocería cuáles registros son correctos y cuáles no? o suponiendo que quiera los reportes de las actividades que tiene la "renta del avión" en el mes pasado. Si la descripción de la "renta del avión, 2085V" no está en la tabla con el mismo camino en cada momento, probablemente es que no ha acumulado los valores apropiados. El hecho es que si la descripción es para ser introducida en cualquier momento se crea un registro, los cambios que verá son con muchas variaciones. En cada entrada como las rentas de los aviones: ac renta 2085V, renta del avión 2085V, ac rntl 2085v, y 2085V AC son algunas posibilidades. Aclarando, si el programa de la computadora está buscando corresponder a la entrada "rentada del avión 2085V", muchas de las expectativas pueden no ser materializadas. Si no corresponde, ¿sería útil que hiciera un reporte?

La estructura redundante de los datos en la tabla FACTURA, se muestra en la figura 1.6. también hace difícil utilizar una entrada automática de los datos a través de la lectura del código de barras. Por una cosa u otra, el código de barras no puede contener mucha información descrita. Si la descripción del objeto y el precio no están guardados por separado, el código de barras no es muy útil. (Aún si guardara cada información con la etiqueta del código de barras, tendría etiquetas de desperdicio cada vez que cambie la descripción o el precio). Sin embargo, si la base de datos consiste de dos tablas mostradas en la figura 1.4, el código de barras CS/11814 sería usado para traer la descripción (gráfico, 12 pulgadas en. plástico) y precio (16.95) desde la tabla ARTICULO que ordena automáticamente cuando esta información entra en la forma de la factura. Si la descripción o precio es cambiado esto puede ser de suma importancia ya que se hizo una corrección en la tabla ARTICULO porque esta información cambia en un sólo lugar. Desde la llave primaria, contenida en un código de barras, esta barra hace comparaciones con datos correctos.

Evidentemente, la minimización de la redundancia de datos produce eficiencia y mejor integridad en los datos, y ésta determina una entrada automática de los datos. La entrada automática de datos es buena por dos razones:

- Es más rápida y eficiente.
- Elimina la introducción de errores humanos en datos en los que probables se haga, como un precio de \$61.95 en vez de \$16.95.

Una base de datos que falla en acomodar automáticamente la entrada de los datos no será aceptada en un negocio moderno en desarrollo.

1.5. Sistema de base de datos

La base de datos es el corazón del sistema de base de datos. Un sistema de base de datos es creado cuando un DBMS es adquirido (tal como Microsoft Access, o un paquete de software similar) es combinable con la base de datos que designe e implemente y las aplicaciones (las formas, los reportes, etiquetas enviadas, y demás) que haya desarrollado. Con mucho más sentido, dos factores adicionales deben ser admitidos: el hardware y la gente.

Primero, desde que toma una computadora (o dos) permite realizar una base de datos con funciones aplicadas a la reposición de datos, el hardware en el cual se encuentra la base de datos es considerado parte del sistema de la base de datos. El hardware incluye microcomputadoras, minicomputadoras (minis), y mainframes. Los componentes del hardware están unidos a través de una

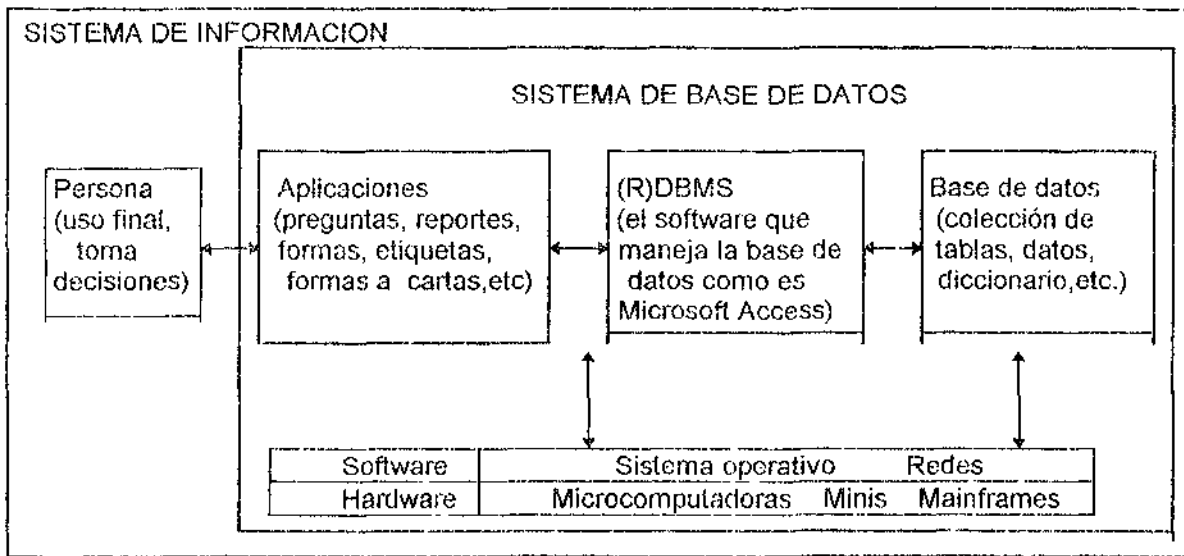
red, el hardware combinado con el software que maneja ésta (es operado por sistemas como lo es DOS, Windows, System7, OS/2, UNIX, así como la red que dirige el software) constituye el ambiente en el cual existe la base de datos. Segundo, éste toma gente que haga trabajos del sistemas.

Nosotros dirigiríamos parte de la gente a la ecuación indirecta cuando tratamos de desarrollar aplicaciones finales. Porque el diseño básico de la base de datos tiene una independencia de hardware, y puede mejorar los detalles cuando se especializa para quien se vende o instituciones educativas.

1.6. Sistema de Información

Los datos son simplemente el nuevo factor almacenado en la base de datos. La información es creada cuando los datos son transformados. La transformación de datos toma lugar cuando la gente los usa en la aplicación para sumar, combinar gráficos, o cualquier otra manipulación de los datos para crear resultados. Cuando el sistema de base de datos es usado, en las formas básicas, en una decisión formal a esto se refiere como un *sistema de información* (ver figura 1.7).

Figura 1.7 Sistema de información/Base de datos



Fuente: Rob, 1995

1.7. ¿Por qué el diseño de la base de datos también es importante?

Finalmente el usuario, usualmente está en contacto sólo con la porción de aplicaciones del sistema de base de datos y ellos casi siempre tienen una lista corta de requerimientos del sistema de la base de datos: estos requerimientos pueden producir información rápida, eficiente, fácil, y con un bonito formato.

Pocos alguna vez consideran la base de datos sobre la acción de la aplicación. Sin embargo algunas personas no la consideran porque tenga un diseño propio, y probablemente no esté bien implementada. Si la base de datos no está bien implementada, aunque las aplicaciones tengan un bonito diseño seguramente dará resultados incorrectos. Un desastre puede ocurrir porque un sistema de

base de datos es diseñado y los usuarios (accidentalmente o por ignorancia) alteren los datos. La computadora nos confundiría otra vez y esto es una queja común.

Para evitar serios defectos y desgracias en la base de datos, debería conocer el modelo básico y diseño de la base de datos. Estudiaremos el material presentado en los siguientes capítulos antes de crear e implementar su propia base de datos. Apresúrese a seguir usted mismo: primero diseñe y entonces construya.

Problemas: . . .

Suponga que está trabajando con una base de datos que contiene las tablas mostradas en la Figura 1.p1.

Dar los componentes de la base de datos, respondiendo a las siguientes preguntas:
(Nota: el EMP_COD en la tabla DIVISION indica que un empleado dirige la división).

- ¿Cuántos conjuntos de entidades contiene la base de datos?
- ¿Cuántas entidades están contenidas en GANAR?
- ¿Cuántos atributos contiene EMPLEADOS?
- ¿Cuántos archivos contiene EMPLEADOS?
- Identificar las llaves primarias y las llaves externas para cada de las siguientes tablas. Si no hay llaves externas, no hay ninguna entrada por debajo de la llave externa principal. Nota: Si una tabla está tiene una llave primaria compuesta, identificar todos de estos componentes.

Tabla	Llave primaria	Llave exterior
EMPLEADO		
REQUISITO		
TRABAJO		
GANAR		
DIVISION		
DEPENDIENTE		

F.-Para cada una de las tablas en esta base de datos, indica (respondiendo SI o NO). Si o no es muestra las referencias y entidad referencial. Usar NA para "No Aplicable".

Tabla	Integridad referencial	Entidad referencial
DEPENDIENTE		
EMPLEADO		
REQUISITO		
TRABAJO		
GANAR		
DIVISION		

G.-Identificar las relaciones (1:1, 1:M, M:N) para todos los conjuntos de entidades de esta base de datos.

Figura 1.p1. Tablas EMPLEADO.

EMP COD	EMP APELL	EMP NOMB	EMP INIC	TRAB COD	EMP TEL	DIV COD
104	Loandas	Ekiri	U	3	456-1234	3
105	Holmes	George	H	1	456-3390	1
106	Washington	Theresa	T	1	456-3389	4
107	Smith	Jerome		2	514-2216	2
108	Roberts	Anne	M	2	821-6675	2
109	Stovall	Peter	G		249-8816	
110	Steinbauer	Jeanne	M	1	514-9007	1
111	Colby	Georgette	W		514-0181	1
112	Koczisko	Paul	K	3	456-2556	4
113	Rosenthal	Herman	R	1	514-1128	1
114	Williamson	Mary	G	2	456-2887	3
116	Rockwell	William		1	574-9113	3

EST-COD	EST-NOM
1	Grado de bachiller
2	Grado de maestría
3	Grado de Ph.D
4	Certificado técnico
5	No certificado técnico

TRAB-COD	TRAB-DESCRIP	TRAB-CAMB
1	Soporte	\$ 32.50
2	Ingeniero	\$ 56.00
3	Admon. de base de datos	\$ 49.90

COD-EMP	REQ-COD
104	1
104	5
105	1
107	1
107	2
107	3
107	4
108	1
108	2
108	3
110	1
112	1
112	4
113	1
113	5
114	1
114	2
114	4
115	4

COD-DIV	DIV-DESCRIP	EMP COD
1	vendiendo	106
2	investigando	108
3	producción	114
4	inf. de sistemas	112

COD-EMP	DEP-NUM	FECHA
105	1	09/15/70
105	2	02/12/89
105	3	12/23/93
107	1	11/14/49
108	1	04/05/65
108	2	10/16/79
111	1	12/17/65
111	2	12/30/93
112	1	09/02/80
114	1	01/29/68
114	2	06/22/87

Fuente: Rob, 1995

Figura 1.p2. Tablas de ROBWil Aviación

Tabla: AVION							
AC-NUM	MODELO	AC-TTAF	AC-TTEL	AC-TTER	AC-100HRS	AC-YTDHR	AC-SINCE100
2085V	PA28-181	1785 3	1785 3	0 0	11/14/94	400.2	50 7
1998Y	C-172	4118 2	1422.0	0 0	10/05/94	521 9	87 4
2873R	C-152	3512 7	1948 5	0 0	10/23/94	497.8	72.1
1550P	C-172	3002 5	1119.2	0 0	11/05/94	600 5	89 3
5827Y	PA28-20	2259 9	385.1	0 0	10/29/94	114 9	23 8
2284W	C-152	3984 4	3984.4	0 0	09/12/94	508 7	91.9
1085Q	C-152	4029 1	114 9	0 0	10/03/94	208 9	34 5

Tabla: RENTA				
RENT-NUM	RENT-FECHA	AC-NUM	RENT-HRS	CUST-NUM
4891	12/12/94	2873R	1.1	1018
4892	12/12/94	1550P	2.7	1024
4893	12/12/94	2284W	0.9	1015
4894	12/12/94	2873R	2.4	1016
4895	12/12/94	1550P	1.6	1017
4896	12/13/94	2805V	4.9	1020
4897	12/13/94	1085Q	1.2	1017
4898	12/13/94	1550P	2.0	1016
4899	12/13/94	2284W	1.5	1025
4900	12/13/94	1550P	1.3	1019
4901	12/13/95	5827Y	2.3	1021
4902	12/14/94	1998Y	0.8	1016
4903	12/14/94	2085V	1.7	1015
4904	12/14/94	2873R	1.5	1019
4905	12/14/94	1085Q	1.2	1016

Tabla: MODELO			
MOD-CODIGO	MOD-MANUF	MOD-ASIENTO	MOD-CHG
Be76	Beechcraft	4	\$125.75
C-152	Cessna	2	\$ 42.50
C-172	Cessna	4	\$ 48.00
PA28-181	Pipero	4	\$ 54.00
PA28-200	Pipero	4	\$ 62.50

Figura 1.p2. Tablas de ROBWil Aviación (continuación)

Tabla: CLIENTE						
CIEN-NUM	CIEN-REQ	CIEN-CHEG	LIMITE	BALANCE	APELLIDO	NOMBRE
1015	Estudiante		\$1200 00	\$456 74	Robertson	Steven
1016	Estudiante	08/12/94	\$1500 00	\$450 38	Drapina	Anna
1017	Pvt/SEL	12/09/93	\$1800 00	\$0 0	Francisco	Mark
1018	Estudiante		\$1200 00	\$0 0	Robertson	John
1019	Estudiante		\$1200 00	\$205 57	Wilson	George
1020	Pvt/SEL*	11/09/94	\$2000 00	\$0 0	Smith	LouAnne
1021	Pvt/Instr/SEL	03/09/94	\$2000 00	\$1219 94	Rosten	Peter
1022	Pvt/SEL	12/09/93	\$1500 00	\$956 72	Darmanov	Elise
1023	Pvt/SEL	10/09/93	\$1500 00	\$205 71	Lopez	Jorge
1024	ATP/MEL	06/09/93	\$2500 00	\$0 0	Smith	William
1025	Pvt/Instr/SEL	04/09/94	\$2000 00	\$0 0	Treynor	Helen
1026	Pvt/SEL	07/09/94	\$1200 00	\$0 0	Erbaum	Joseph
1027	Comm/Instr/SMEL	10/09/93	\$1200 00	\$0 0	William	George
1028	Estudiante		\$1200 00	\$0 0	Smithson	Anne
1029	Comm/Instr/SEL	03/09/93	\$2000 00	\$864 38	Rakkof	William

Tabla: CLIENTE (continuación)						
CIEN INIC	DIRECC	CIUDAD	ESTADO	C. P.	AREA	TELEFONO
T	1235 East Barner Road	Murfreesboro	TN	37456	615	899-1234
G	987 Traverse Lane	Narsville	TN	21778	615	456-2315
	Box1245B	Smyrna	TN	34567	615	904-9981
H	145 East Main Str.	Murfreesboro	TN	34456	615	891-3245
D	985 Meadow Song Dr	Murfreesboro	TN	37456	615	899-0095
E	2314 Lake Shore Rd.	Lavergne	TN	38894	615	915-9982
M	Box 231	Murfreesboro	TN	37456	615	893-6667
H	2316 Unicorn Circle	Murfreesboro	TN	37456	615	899-1292
W	769 Kelly Drive	Nashville	TN	21772	615	450-0093
	Box1234 H	Murfreesboro	TN	37456	615	899-2667
K	983 Auto Circle	Murfreesboro	TN	37465	615	899-4528
D	218 GreatView Lane	Smyrna	TN	34567	615	904-6670
B	3425 Rowland Dr	Bowling Green	TN	32119	615	324-0893
J	453 Meadow Song	Murfreesboro	TN	37457	615	891-3346
A	2769 Airport Rd.	Neshville	TN	21772	615	456-8891

Fuente: Rob, 1995

H.-Sugerir mínimo una llave secundaria de la tabla de EMPLEADO (mínimo una llave secundaria que cumpla los datos solicitados mientras está usando el mínimo número de atributos).

1.2 Un fragmento de la base de datos ROBWil Aviación contiene las tablas mostradas en la figura 1.p2.

- 1 Dar los componentes de la base de datos ROBWil, respondiendo las siguientes preguntas.
- 2 ¿Cuántos conjuntos de entidades contiene esta base de datos?
- 3 ¿Cuántas entidades están contenidas en la tabla RENTA?
- 4 ¿Cuántos atributos contiene la tabla CLIENTE?
- 5 ¿Cuántos archivos contiene la tabla AVION?
- 6 Identificar las llaves: primaria y externa para cada de las siguientes tablas: Si no hay llave externa, introducir NINGUNO debajo de la llave externa principal. Nota. si una tabla tiene una llave primaria compuesta, identificar todos sus componentes.

TABLA	LLAVE PRIMARIA	LLAVE EXTERNA
AVION		
RENTA		
CLIENTE		
MODELO		

7. Para cada una de las tablas en la de datos, indica (respondiendo SI o NO) cualquiera de las referencias y entidad referencial. Usar NA para "No Aplicable".

TABLA	INTEGRIDAD REFERENCIAL	ENTIDAD REFERENCIAL
AVION		
RENTA		
CLIENTE		
MODELO		

8. Identificar las relaciones (1:1,1:M, M:N) para todo el conjunto de entidades dentro de la base de datos.
9. Sugerir mínimo una llave secundaria para la tabla CLIENTE.

Capítulo 2. MODELANDO Y DISEÑANDO BASES DE DATOS

Dado el objetivo principal de este trabajo, que es el de dar a conocer diferentes técnicas para el diseño de base de datos, en este capítulo se señalan los conceptos de los modelos de base de datos, símbolos, técnicas y algo más importante: el vocabulario.

Como un buen plano ayuda a evitar problemas en la construcción de casas, un buen diseño ayudará a evitar más problemas que puedan dañar el sistema de base de datos. Semejante al plano de un arquitecto, el diseño de la base de datos es un modelo (esto es, una abstracción) real de los datos. Y semejante al plano del arquitecto, un diseño de base de datos es dibujado con la ayuda de una colección de símbolos para definir su construcción.

El material de este capítulo está basado en el trabajo de Peter Rob y Treyton Williams, (1995).

2.1. Modelando una entidad relación (E-R): algunos antecedentes

Un modelo es una representación simplificada del mundo real del objeto o condición. Es decir, un modelo es una abstracción, que representa el concepto visto del objeto o condición que se estudia; justo como un plano, es una abstracción que representa una construcción. Nosotros usaremos modelos porque son fáciles para nuestro real entendimiento de la composición y funciones de lo que pensamos.

Desde entonces la base de datos es compleja y podemos usar modelos para representar componentes y relaciones entre esos componentes.

En el primer capítulo se habló del conjunto de entidades y de relaciones determinantes para el uso de gráficas de los modelos de **entidad relación (E-R)** que representan abstracciones del entorno real de los datos. Simplemente un modelo E-R es un modelo **conceptual** que puede llegar a ser básico para un proyecto de base de datos.

El modelo E-R es sólo uno de muchos modelos de base de datos. Hemos elegido usar el modelo E-R como nuestro diseño estándar porque:

- La representación gráfica del modelo E-R simplifica el proceso de diseño de la base de datos.
- Análogo a un proyecto, el modelo E-R es una excelente comunicación de herramientas.
- El modelo E-R ha sido bien establecido y de exitoso diseño histórico.
- El modelo E-R es básico, en gran parte, para la disponibilidad de la base de datos en el diseño del software.

El modelo E-R ha existido por mucho tiempo. Peter Chen, generalmente reconocido como el padre del modelo E-R, describe por primera vez los componentes básicos en 1976. Chen ha trabajado al mando para muchas investigaciones que produjo el modelo E-R y ahora lo conocemos como el diseño estándar. Existen muchos diseños de herramientas para la base de datos como es ERwinLogic y Exellerator, que son basados en ésta. Teniendo la ayuda para producir el diseño estándar, Chen y

asociados, ahora lo proveen a bajo costo. Si diseña su base de datos con un lápiz o un modelo E-R y diseña herramientas entonces debería usted conocer los componentes básicos del modelo E-R ligados a la producir un proyecto de base de datos que puede ser implementado.

2.2. Diagrama E-R y sus componentes

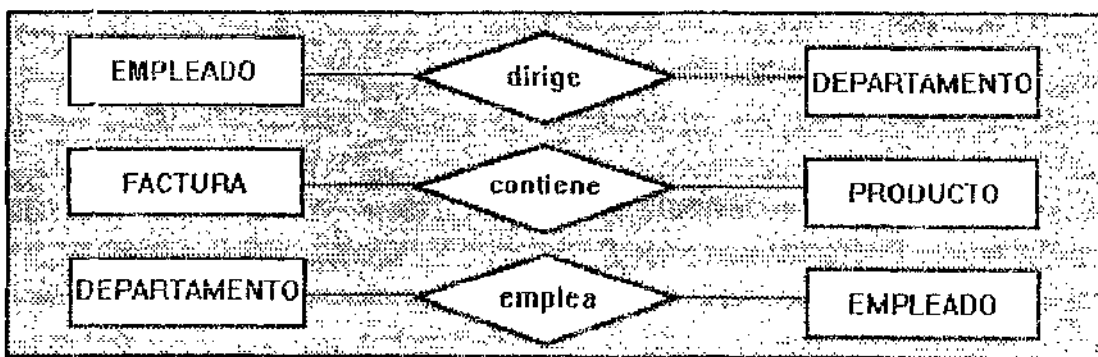
El modelo E-R está compuesto por conjuntos de entidades y relaciones. Desde entonces el modelo es representado por un diagrama E-R, no le sorprenda que después contenga los mismos componentes. Casi siempre hay confusiones cuando trabaje con modelos E-R. Por razones desconocidas el modelo E-R usa la palabra *entidad* cuando ellos se refieren a un conjunto de entidades. En otras palabras, las entidades en el diagrama E-R están representadas por una tabla y uno de sus renglones está etiquetado por entidades ocurrentes.

2.2.1 Entidades, atributos y relaciones

Una entidad (recuerde que actualmente es un conjunto de entidades) es representado por un rectángulo que contiene los nombres de entidades (ver figura 2.1). Siempre las entidades están nombradas como CLIENTE, PARTE, FACTURA, LIBRO, y demás. Recuerde que para usar una entidad debe ser singular como CLIENTE en vez de CLIENTES, por ejemplo.

Las relaciones entre las entidades son representadas por rombos que contienen relaciones de nombres, como se muestra en la figura 2.1. los nombres de las relaciones son siempre representados en cajas rotuladas y éste es siempre un verbo, preferentemente uno activo, como trabaja, dirige, reportes, contenidos. Aunque los verbos activos son preferidos, cada descripción pasiva como es un empleado, este relato, o esta descripción es también aceptada.

Figura 2.1. Entidades y Relaciones

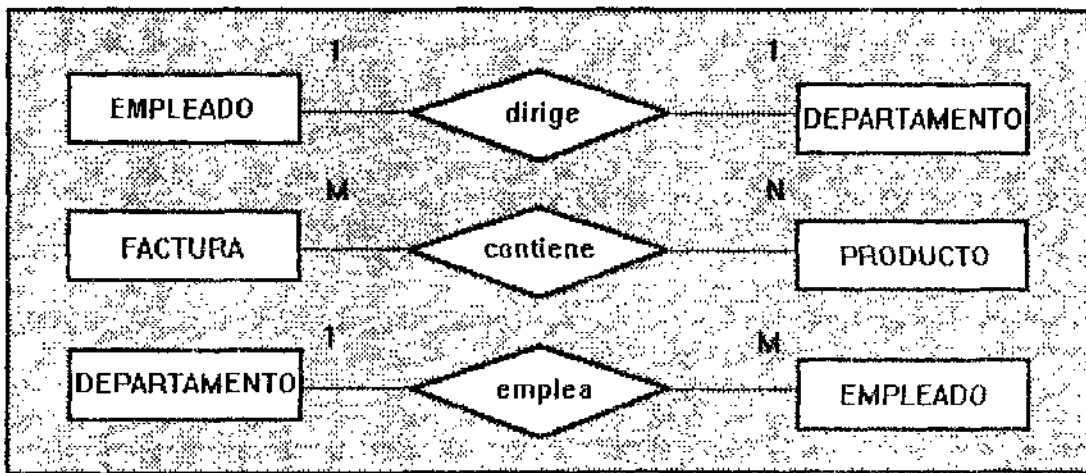


Fuente, Rob 1995

2.2.2 Conectividades y cardinales

La **conectividad** se refiere al tipo de relación (1:1, 1:M o M:N) entre las entidades, como se ilustra en la figura 2.2 la cual muestra que conectividad en la relación EMPLEADO dirige DEPARTAMENTO es 1:1; indicando que un departamento es dirigido por sólo un empleado y un empleado puede dirigir sólo un departamento. Similarmente, la conectividad en la relación FACTURA contiene PRODUCTO es M:N, reflejando el hecho de que una factura puede contener muchos productos en efecto un código del producto puede ser encontrado por muchas facturas. Finalmente, la conectividad en la relación DEPARTAMENTO emplea EMPLEADO es 1:M, indicando que un departamento emplea muchos empleados, pero cada empleado es empleado por sólo un departamento.

Figura 2.2 Conectividades



Fuente: Rob, 1995

Deberá de recordar que la conectividad está escrita a un punto próximo de la entidad apuntando hacia otra entidad con la cual esté relacionada. Observe porqué esta semejanza es apropiada, también observe la figura 2.3. Note que en la tabla EMPLEADO contiene muchas referencias de la tabla DEPARTAMENTO. Por ejemplo, los empleados 1005,1007 y 1008 todos apuntan al departamento 2. En otras palabras, el departamento 2 ocurre muchas veces en la tabla EMPLEADO. Por lo tanto, la "M" se escribe para el siguiente EMPLEADO en el diagrama E-R. El diagrama E-R también indica que un DEPARTAMENTO emplea muchos EMPLEADOS, por otro lado, cada empleado es empleado por un solo departamento. Por lo tanto cada referencia del departamento ocurre sólo una vez en la tabla DEPARTAMENTO, el "1" es escrito al siguiente DEPARTAMENTO en el diagrama E-R.

Examine en la figura 2.3 que una entidad (actualmente un conjunto de entidades) se representa por una tabla y la conexión entre DEPARTAMENTO y EMPLEADO es mantenido a través del atributo común DEPT-COD. Esta conexión está ilustrada en el tercer esquema relacional mostrado más abajo en la figura 2.3. Observe que el software que puede usar produce cada esquema para algunas consultas, formas, y reportes que son basados en más de una entidad (note que DEPT-COD es la llave primaria en DEPARTAMENTO y la llave externa en EMPLEADO).

Figura 2.3. Diagrama E-R, estructura de la tabla y el esquema relacional

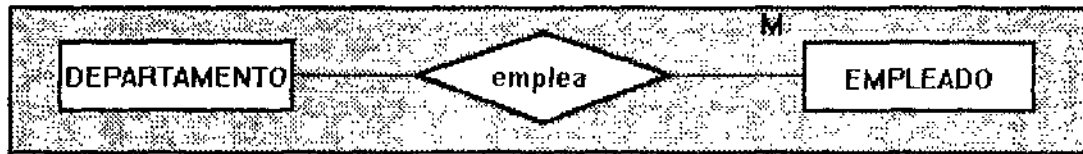
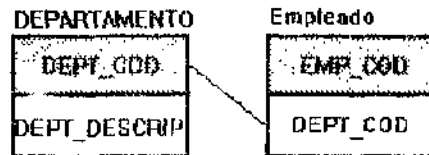


Tabla: DEPARTAMENTO

DEPT_COD	DEPT_DESCRIP
1	Mercado
2	Nómina

Tabla: EMPLEADO

EMP-COD	DEPT-COD
1005	2
1006	1
1007	2
1008	2
1009	1

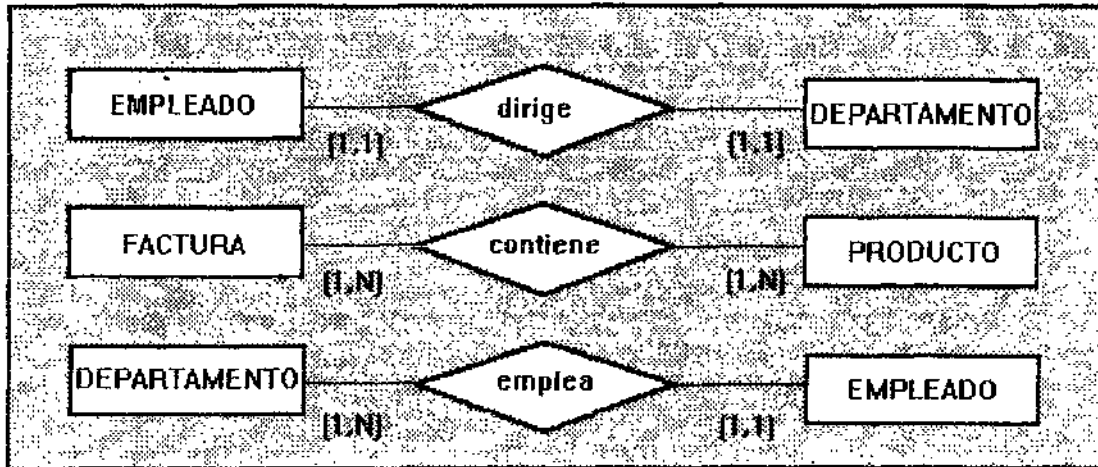


Fuente: Rob, 1995

Las **cardinalidades** definen la parte más baja y la más alta de la relación. Por ejemplo, la figura 2.4 muestra la etiqueta cardinal (1:1) de la relación EMPLEADO dirige DEPARTAMENTO. El cardinal (1:1) indica que un empleado, y no más de uno, puede dirigir un departamento. Similarmente, la etiqueta (1:1) próxima a DEPARTAMENTO en la misma relación indica que un departamento es dirigido por un empleado, y no más de uno. Por otro lado, la cardinalidad (1,N) localizada próxima a el DEPARTAMENTO en la relación DEPARTAMENTO emplea EMPLEADO indica que un departamento puede emplear un mínimo de un empleado, y un máximo de N empleados. Note que, parecido a las conectividades, las cardinalidades que están próximas a una entidad describe los límites sobre las relaciones con otra entidad.

Los cardinales no necesariamente es abrir-cerrar como se quiere decir (1,N). Por ejemplo, si determinamos que un departamento no puede contener menos que cinco empleados y si contiene más de cincuenta empleados, los cardinales deberían ser representados así (5,50)

Figura 2.4 Cardinalidades



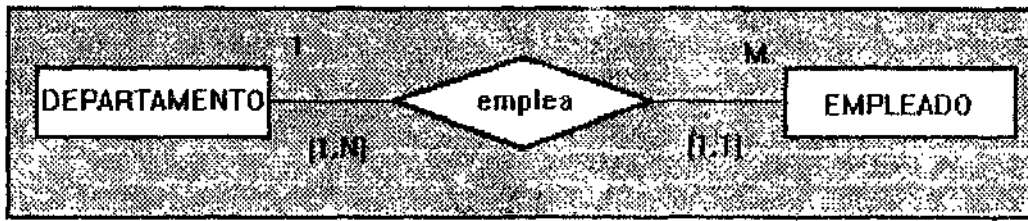
Fuente: Rob, 1995

No necesariamente, las relaciones entre las entidades son determinadas por algunas entidades. Los determinan el medio ambiente para operar entidades. Dentro de éste se describe por un conjunto de reglas llamadas **reglas de negocios**. Por ejemplo, en una organización, un empleado puede ser asignado para sólo un departamento y éste puede ser compartido para varios departamentos. Similarmente, una universidad permite enseñar a un conjunto de personas, de este modo definiendo que un profesor puede enseñar muchos cursos y un curso puede ser enseñado por muchos profesores. Desde las reglas de negocios se determinan ambas conectividades y cardinalidades, no le sorprenda descubrir que la determinación de reglas de negocios es parte del diseño de procesos.

El término *regla de negocios* implica que esas reglas están limitadas al entorno del negocio, pero nada puede ser exacto. Cualquier operación en la base de datos es dirigida por la regla de negocios, sea que ésta sirve a negocios, al Administración, a la universidad o a un ecosistema. De hecho, en algunos casos (como un ecosistema) no somos libres para definir las reglas y adaptarlas a nuestras necesidades, pero debemos descubrir que esas reglas son. Actualmente, es probable que esté trabajando en la creación del diseño de una base de datos para la gente o para usted, puede descubrir que algunas organizaciones, creadas por humanos, pueden desarrollar un conjunto de reglas para operarlas, que son tan inflexibles como las creadas por la naturaleza. En cualquier caso, encontraremos que las reglas de negocios son cruciales para el éxito de su diseño de base de datos.

Las conectividades y cardinalidades ambas son mostradas en el diagrama E-R. Por ejemplo note que el diagrama E-R, mostrado en la figura 2.3, debe ser modificado como se encuentra en la figura 2.5.

Figura 2.5 Conectividades y Cardinales



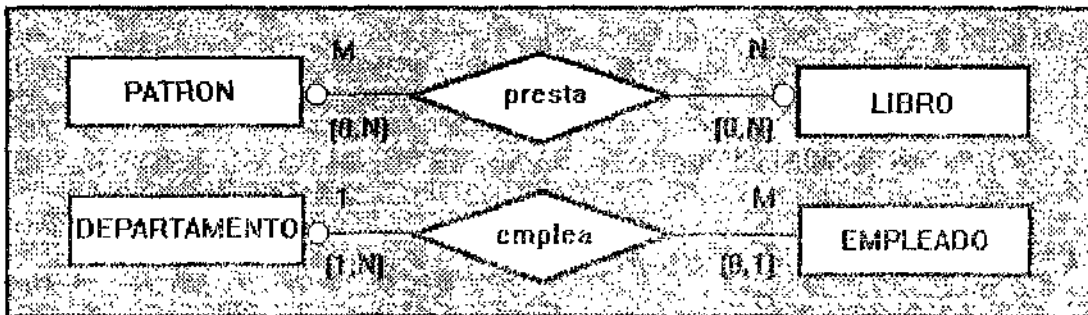
Fuente: Rob, 1995

2.2.3 Relaciones: opcional y obligatoria

Muchas relaciones son opcionales. Por ejemplo, no necesariamente un empleado es asignado a un departamento, DEPARTAMENTO es opcional para EMPLEADO. Las relaciones son opcionales u obligatorias dependiendo de las reglas de negocios.

Las relaciones opcionales están marcadas por un círculo, como lo muestra en la figura 2.6. Note que las reglas especifican que un empleado no necesariamente es asignado a un "DEPARTAMENTO" y también que "un empleado no puede ser asignado a más de un departamento", DEPARTAMENTO sería opcional para EMPLEADO y, por lo tanto, el círculo pequeño aparece al siguiente DEPARTAMENTO. Note también que las reglas de cardinalidad han cambiado (0,1) porque ahora un empleado puede ser asignado a ningún departamento.

Figura 2.6 Representando relaciones opcionales



Fuente: Rob, 1995

Examine en la figura 2.6. que las relaciones opcionales pueden ocurrir ya sea en la relación 1:1, 1:M o M:N y éstas pueden ocurrir en cualquiera de los dos lados, o en ambos lados de una relación. Por ejemplo, un patrón de biblioteca no tiene que pedir libros para tener una tarjeta de ésta misma, no es necesario que un LIBRO esté asociado a un patrón. Después de todo algunos libros sobre la repisa nunca son revisados o checados fuera de la biblioteca, por lo tanto esos libros no son aprovechados si no están prestados. Entonces, en un lapso de tiempo, un patrón no puede checar ningún libro que esté fuera, la cardinalidad en específico sería la condición (0,N) al lado de libro. Similarmente, durante algún tiempo, un libro puede ser prestado a todos o a muchos patrones, por lo tanto la cardinalidad sería también (0,N) al lado del patrón.

Si una relación es opcional u obligatoria, no es exactamente algo de sumo interés. Después descubrirá que una relación obligatoria debe ser protegida por una construcción segura, existiendo en un lado de la entidad y no puede ser borrada si una entidad ocurre en otro lado dependiendo de su existencia. Por ejemplo, si conserva su propio inventario para una organización no podrá aceptar que borren una localidad de almacenamiento, debido a que tiene muchos artículos almacenados en un punto de esa localidad de almacenamiento. Para evitar problemas deberá exigir la propiedad de los bienes inmuebles en el que se encuentra lo almacenado para ser borrado y exigir otra localidad de almacenamiento para después reconstruir lo borrado. De otra manera si una relación es opcional, cada restricción podría no ser necesaria. Por ejemplo, si debe borrar un patrón o un libro de la base de datos de la biblioteca sin tener la protección de cada patrón o libro en ese conjunto de relaciones opcionales.

2.2.4 Entidades compuestas

Hemos discutido que las relaciones M:N son completamente aceptadas a nivel conceptual, ellas no pueden ser implementadas apropiadamente. La relación en la base de datos no acepta fácilmente la relación M:N, rompemos el conjunto de relaciones 1:M. Las relaciones 1:M están conectadas a través de entidades compuestas, conocidas también como entidades puente, éstas contienen un mínimo de llaves primarias que sirven como conexiones. La combinación de estas llaves primarias sería la llave primaria compuesta de la entidad compuesta.

Figura 2.7 La relación M:N entre EMPLEADO y CLASE

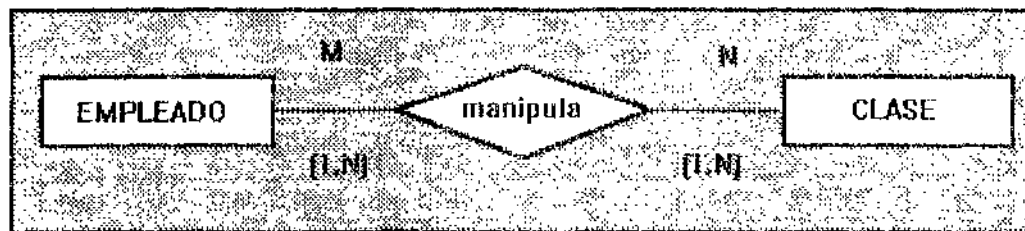


Tabla: EMPLEADO		
EMP-COD	CLAS-COD	EMP-APELLIDO
10	B	Lindsey
10	G	Lindsey
11	M	Cavazos
11	G	Cavazos
11	H	Cavazos
12	M	Robertson
13	G	Simpson
14	M	Washington
14	G	Washington
15	A	Jones
15	G	Jones
16	H	Claymore
17	H	Maryke
17	G	Maryke
18	A	Jonhson
18	M	Johnson

Tabla: CLASE		
CLAS-COD	CLAS-DESCRIP	EMP-COD
A	viación	15
A	viación	18
G	dministración	10
G	dministración	11
G	dministración	13
G	dministración	14
G	dministración	15
G	dministración	17
B	clinación Lateral	10
H	ospital y Serv. Rel	11
H	ospital y Serv. Rel	16
H	ospital y Serv. Rel	17
M	anufactura	11
M	anufactura	12
M	anufactura	14
M	anufactura	18

Fuente: Rob, 1995

Para una mejor idea de las transformaciones desde una relación M:N hacia un conjunto de relaciones 1:M, examinemos el siguiente ejemplo. Suponiendo que la compañía consultora ROBUIL proporciona varios servicios en varias clasificaciones (G=Administración, B=Inclinación Lateral, M=Manufactura, A=Aviación, H=Hospitales y servicios relacionados). El personal de ROBUIL cuyos atributos están almacenados en la tabla EMPLEADO, incluye varias personas que están calificadas para proporcionar servicios en diferentes clasificaciones. (La clasificación detallada de los servicios es almacenada en una tabla llamada CLASE). Cada clasificación sirve a un o más empleados. En otras palabras, una relación M:N existe entre EMPLEADO y CLASE (ver en la figura 2.7).

Figura 2.8 Creando una entidad compuesta

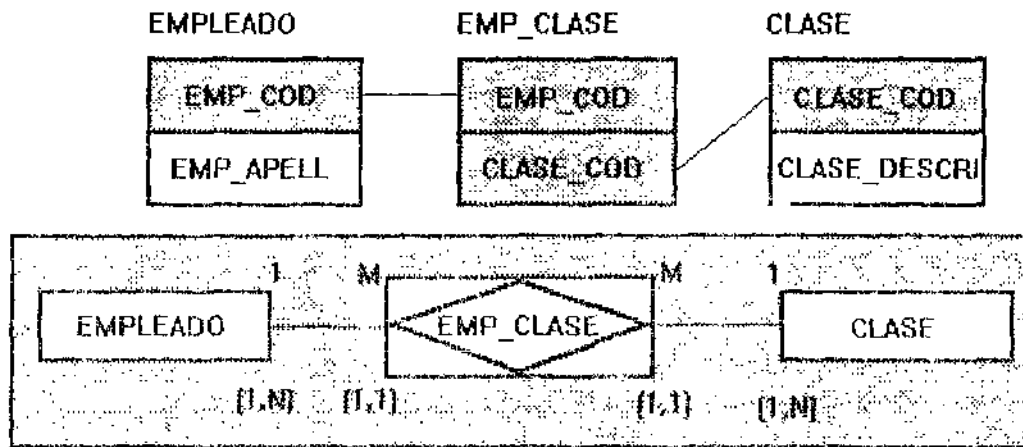


Tabla: EMPLEADO	
EMP-COD	EMP-APELL
10	Lindsey
11	Cavazos
12	Robertson
13	Simpson
14	Washington
15	Jones
16	Claymore
17	Maryke
18	Jonhson

Tabla: EMP-CLASE	
EMP-COD	CLASE-COD
10	B
10	G
11	M
11	G
11	H
12	M
13	G
14	M
14	G
15	A
15	G
16	H
17	H
17	G
18	A
18	M

Tabla: CLASE	
CLASE-COD	CLASE-DESCRIP
A	Aviación
B	Inclinación Lateral
G	Administración
H	Hospital y Serv. Rel.
M	Manufactura

Examine las tablas de la figura 2.7, ponga atención en la relación M:N la cual crea redundancia porque hay muchos acontecimientos de empleados en la tabla CLASE y muchos acontecimientos de clases en la tabla EMPLEADO. Y éste es un ejemplo poco riguroso: imagine la redundancia si (como debe ser realista) la tabla EMPLEADO tiene muchos atributos como nombres, iniciales, número del seguro social, dirección, número telefónico, etcétera, éste es mayor, si agregamos atributos que probablemente se encuentran en una tabla CLASE más real. Evidentemente, la relación M:N crea una redundancia real que no puede ser tolerada.

Afortunadamente, la relación M:N representada en la figura 2.7 puede ser descompuesta en dos relaciones que se muestran en la figura 2.8. Como puede ver, las dos entidades EMPLEADO y CLASE están ahora conectadas a través de una nueva entidad compuesta llamada EMP-CLASE, de quién la llave primaria es la combinación de la llave primaria de EMPLEADO: EMP-COD y la llave primaria de CLASE: CLASE-COD. De la tabla EMP-CLASE: EMP-COD y CLASE-COD apuntan a las tablas EMPLEADO y CLASE, respectivamente. Por lo tanto, EMP-COD y CLASE-COD son también llaves externas en la tabla EMP-CLASE. Note que la entidad compuesta es mostrada como un diamante en el interior del rectángulo. Aunque hay algunas redundancias, no hay comparación de la que existía antes, porque esas redundancias están ahora limitadas a múltiples ocurrencias de los valores de la llave externa en la tabla EMP-CLASE.

2.2.5 Entidades débiles

Una entidad débil no puede existir por sí misma. Por ejemplo, DEPENDIENTE en la relación EMPLEADO es una entidad débil. (Evidentemente, ¿una organización no llevaría dependientes en estos libros para gente que no emplea!).

La entidad débil nombrada DEPENDIENTE, está indicada por un rectángulo con doble línea, como se presenta en la figura 2.9. Note que la conectividad y la cardinalidad es mostrada ahora. Entonces no podemos requerir de empleados para los dependientes. La entidad débil es opcional para EMPLEADO, tener en mente que un dependiente está relacionado a sólo un empleado. (¡Seguramente la organización no estará dispuesta a llevar beneficios a dependientes más de una vez!). Finalmente, desde empleados puede tener sólo un dependiente, la relación entre EMPLEADO y DEPENDIENTE es de 1:M.

En la figura 2.9, observe las siguientes condiciones:

- No todos los empleados tienen dependientes (observe el valor de EMP-COD en la tabla DEPENDIENTE y observe que éstos no incluyen todos los valores de EMP-COD en la tabla EMPLEADO.)
- La llave primaria de la entidad débil es una llave compuesta de la combinación EMP-COD y DEP-COD. (La entidad débil EMP-COD, no puede ser un valor de atributo único, porque un empleado puede tener más de un dependiente).
- El EMP-COD, entidad débil, es una llave externa que apunta hacia la tabla EMPLEADO.

Figura 2.9 Representando a una entidad débil.

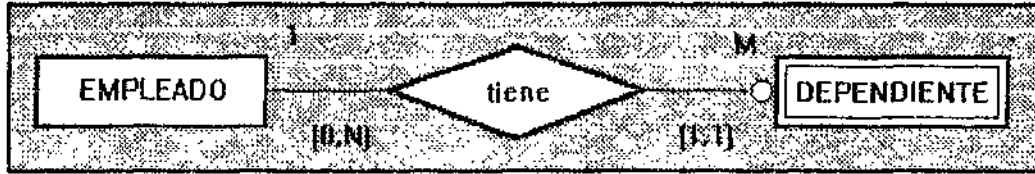
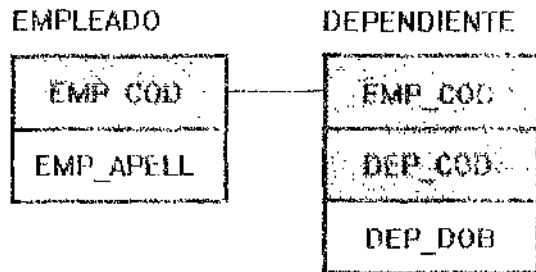


Tabla: EMPLEADO	
EMP-COD	EMP-APELL
10	Lindsey
11	Cavazos
12	Robertson
13	Simpson
14	Washington
15	Jones
16	Claymore
17	Maryke
18	Jonhson

Tabla: DEPENDIENTE		
EMP-COD	DEP-COD	DEP-DOB
11	1	11/15/46
12	1	12/27/89
12	2	07/16/93
17	1	02/19/61
17	2	10/30/85
17	3	03/11/88
17	4	07/14/93



Fuente: Rob, 1995

2.2.6 Supertipos y subtipos

Suponiendo que diseña una entidad llamada EMPLEADO para un negocio de aviación conocido como Operador de Base Fija (FBO). Los empleados de FBO incluyen personas con características profesionales diferentes. Por ejemplo, un atributo de un piloto podría incluir su tipo de licencia, la fecha del último vuelo adelantado, el número total de horas de vuelo, etcétera; al contrario, una secretaria no debe tener atributos específicos de un piloto, pero debería compartir cada atributo como apellido, número telefónico y demás. Claro está que si su tabla de EMPLEADO contiene personas que comparten sólo un número limitado de atributos, se tendría un grupo de nulos en la tabla (ver en la figura 2 10).

Para evitar nulos en un diseño de base de datos, podríamos simplificar asignando un valor imitado para las horas de vuelo (0,0) para Robertson, Kochenko y Jones. Similarmente podemos asignar un valores imitados como NA (No Aplicable) para checar la fecha de vuelo y el atributo de licencia para Robertson, Kochenko y Jones. De hecho, cada valor es usado frecuentemente.

Figura 2.10 Una tabla con muchos nulos.

Tabla: EMPLEADO						
EMP-COD	EMP-APELL	EMP-OFIC	EMP-TEL	EMP-VUELO	EMP-CHKFECHA	EMP-LIC
1001	Wachile	K-203	1185	6002.7	11/13/94	ATP/MEL
1002	Robertson	M-328	2298			
1003	Smith	K-205	1326	5294.6	10/15/94	COMM/MEL
1004	Kocherko	M-211	2501			
1005	Jones	M-292	2443			

Fuente: Rob, 1995

Figura 2.11 Supertipo y Subtipo.

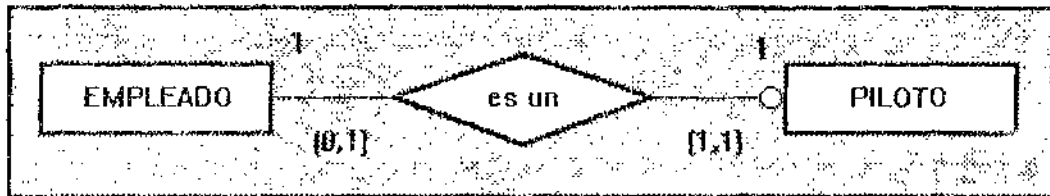
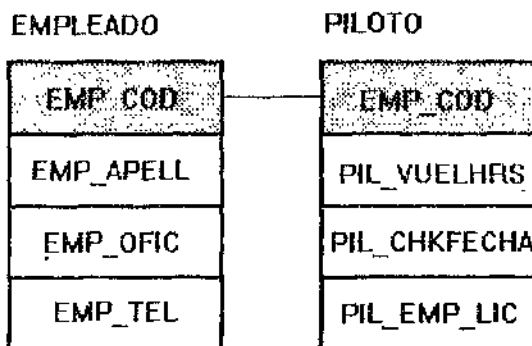


Tabla: EMPLEADO			
EMP-COD	EMP-APELL	EMP-OFIC	EMP-TEL
1001	Wachile	K-203	1185
1002	Robertson	M-328	2298
1003	Smith	K-205	1326
1004	Kochenko	M-211	2501
1005	Jones	M-292	2443

Tabla: PILOTO			
EMP-COD	PIL-HRSVUELO	PIL-CHKFECHA	PIL-LIC
1001	6002.7	11/13/94	ATP/ME
1003	5294.6	10/15/94	COMM/MEL



Fuente: Rob, 1995

Sin embargo, hay un punto donde un diseñador encuentra que el uso de valores imitados es excesivo. Si el apuntador es largo de un asunto que falla, el diseñador puede partir de la tabla EMPLEADO o más tablas, todas unidas por una relación 1:1, como se muestra en la figura 2.11. Note que en la tabla EMPLEADO contiene atributos comunes y la tabla PILOTO sólo tiene atributos asociados con pilotos. Las entidades basadas en atributos comunes es conocida como **supertipo**, mientras que la entidad basada en la especialización de atributos se les llama **subtipo**. Porque un empleado no requiere ser piloto, PILOTO es opcional de EMPLEADO.

En la figura 2.11. demostramos, las relaciones supertipo/subtipo y así eliminar la necesidad de nulos o de valores imitados. El precio pagado por cada diseño grande y elegante es un número largo en entidades y relaciones, haciéndolo un sistema más complejo. El diseñador debe hacer un buen juicio para hacer un acercamiento más considerable.

2.2.7. Diseño con atributos

Aunque se mencionan los atributos en el primer capítulo, los veremos otra vez para saber cómo se adaptan al diseño en la construcción de un trabajo. Estos se adaptan en dos categorías: simples y compuestas:

1. Los **atributos simples** no pueden ser subdivididos. Ejemplo: género, ciudad de origen, apellido, etcétera.

2. Los **atributos compuestos** pueden ser subdivididos. Ejemplo: nombre (puede ser subdividido en nombre, apellido e inicial), dirección (puede ser subdividido en calle, ciudad, estado y código postal).

Generalmente se usa el atributo simple porque permite al usuario buscar más fácilmente los datos. Por ejemplo, para hacer un estudio de investigaciones, pudo requerir buscar datos del cliente por código postal, ciudad, estado o por alguna combinación de cada atributo simple. Cada tarea podría ser mucho más compleja, si el atributo compuesto tuvo todas esas piezas.

Atributos simples:

DIRECCION	2134 Boulevard Norte Ridge
CIUDAD	Murfreesboro
ESTADO	TN
CODIGO POSTAL	37129

puede ser conveniente más de un atributo compuesto.

DIRECCION	Boulevard Norte Ridge 2134, Murfreesboro, TN 37129.
-----------	---

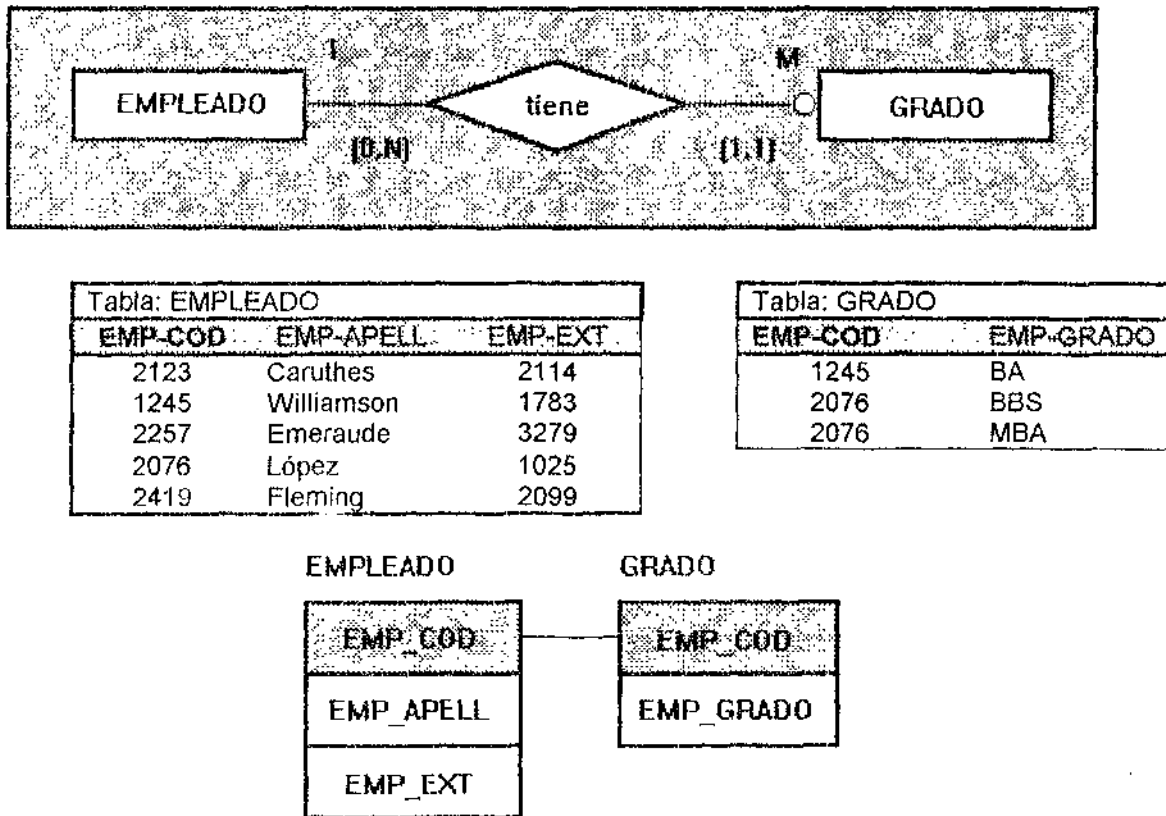
Naturalmente, dentro de su opinión, el dato puede ser subdividido.

Los atributos también son clasificados por sus datos característicos:

- El valor del **atributo simple** puede tener sólo un valor. Por ejemplo, una persona solamente tiene un número de seguro social y un género. Debe tener en mente que un *valor singular* no necesariamente son atributos *simples*. Por ejemplo, puede tener sólo un número de teléfono en casa, pero el número puede ser subdividido dentro del código de área, el tercer dígito del número cambia y el cuarto lo rastrea. (Otra vez la subdivisión depende de nuestro uso final y de nuestros requerimientos).

- **Atributos multi-valores** puede tener más de un valor. Por ejemplo, el atributo GRADO puede producir valores como: BA, BS, MA, MS, Ph.D, y demás.

Figura 2.12. Interpretación del multi-atributo



Fuente: Rob, 1995

La existencia del atributo multi-valor requiere de algunas planeaciones:

- El atributo puede ser definido como una cadena de caracteres simple. Ejemplo, se define GRADO para incluir todos los grados y certificados profesionales, y sería especificado como una cadena de caracteres de veinte y cinco para recibir entradas diferentes como:

ninguno
 BA
 BBS, MBA
 BA, BS, MA, MBA, Ph.D, CPA, CDP

- Cada dato almacenado es a favor de una simplicidad. Sin embargo, si su plan ha sido dirigido para llevar una búsqueda de los datos basada en títulos específicos ("¿Cuántos empleados tiene el grado de BA?"), esta solución no es considerable.

- Los atributos múltiples debieron ser definidos. Por ejemplo, GRADO.1, GRADO.2, GRADO.3, etcétera. Este registro tiende a ser más satisfactorio. Suponiendo que define cinco atributos que son para recibir grados y valores de certificados profesionales. Muy pocos empleados tendrían cinco conjuntos reales de datos para unir los atributos, y concluiríamos con un grupo de nulos o valores simulados como es ninguno, NA, etc. Y si un empleado aprende un sexto grado, tendría que crear un atributo *adicional*.

- Defina una entidad por separado (GRADO) para tener el dato del grado y unirlo a la organización (EMPLEADO) usando la relación 1:M. Si el uso final requiere de un bosquejo detallado basándose en el antecedente educacional, preferimos acercarnos, mostrándolo en la figura 2.12. Tenemos que eliminar el problema nulo/imitado, debemos proporcionar los requerimientos de acuerdo a los requerimientos de cada posibilidad. (Note que el grado es opcional para EMPEADO).

2.3. Poniendo las piezas juntas: diseño de base de datos simple.

Para ilustrar el modelado E-R creamos un modelo E-R basándonos en las operaciones de negocios. Esta descripción viene de las reglas de negocios apropiadas. Entonces, ésta regla de negocios crea el diagrama E-R.

La Corporación ROBWIL proporciona servicios de consulta en ocho regiones. Estas regiones están clasificados como sudeste, noreste, medio oeste, noroeste, oeste, Alaska y Hawaii. Los servicios de consulta cubre áreas como servicios para clientes que están clasificados como Administración (incluyendo estado, local y federal), Inclinación Lateral, Manufacturas, Aerolíneas, Hospitales y servicios relacionados a la salud.

Los asesores de ROBWIL están dirigidos a responder en una pequeña área o a la área de servicios para clientes. Los asesores de ROBWIL son asignados dependiendo de los requerimientos que el cliente solicite. (En otras palabras, un cliente que requiere de los servicios de Inclinación Lateral es asignado a un asesor que tiene los requisitos de entrenamiento y certificación). Aunque cada cliente tiene asignado un asesor singular, cada asesor proporciona servicios a diferentes clientes. Por ejemplo, un asesor certificado en Inclinación Lateral y en Administración puede tener tres clientes que requieran consultas de servicios de Inclinación Lateral, y cinco clientes que requieran de servicios de Administración. Para proporcionar el diseño de relaciones cliente-asesor, los asesores de ROBWIL sólo separan clientes quienes son localizados dentro de su propia región. (Esto es un asesor de la región sudoeste proporciona servicios sólo en esa región).

El gerente de ROBWIL quiere unir un cliente nuevo con un asesor de manera rápida y eficiente. Por ejemplo, una llamada ingresada debe ser asignada a un asesor apropiado sin demora. (Los clientes pueden requerir un asesor específico. Si el cliente no hace tal especificación un asesor certificado puede ser asignado por ROBWIL). Naturalmente la cuenta del cliente está en función del tipo de servicio requerido.

NOTA: Hemos abreviado y alterado la versión de una base de datos real. Nuestro enfoque sería desarrollar los componentes básicos de los modelos E-R, sobre los detalles específicos encontrarás un mundo real dentro del desarrollo de la base de datos. Sin embargo, en la simplificación de la descripción de operaciones dadas aquí, el resto del diseño básico de la base de datos está intacto.

La descripción detallada y exacta de operaciones es un componente de la llave del diseño en proceso. Dicha descripción está basada en entrevistas a dirigentes, en usuarios de bajo nivel, cuidadosa observación de prácticas y procedimientos, exámenes de reportes y consulta de formatos, etcétera.

Hay de hecho, algunas preguntas que destacan para ser respondidas por el diseño de base de datos. Por ejemplo, ¿qué relación podría ser considerada opcional y, de ellas cuál obligatoria?. La respuesta para cada pregunta tendría mucho impacto en la implementación del diseño. Por ejemplo, si especificamos que cada cliente debe tener un asesor asignado, entonces el ASESOR es una entidad obligatoria para CLIENTE. Cada relación no puede ser razonable. Pero tendría muchas razones para ser CLIENTE obligatorio a ASESOR, porque la relación nos forzaría a asignar un cliente a un asesor, quién ha empleado a quién; por lo tanto, este probablemente no tiene un cliente. (Por supuesto, podríamos asignar un valor imitado para indicar que un asesor no tiene un cliente. Pero ¿por qué obtenerlo afecta cada atributo innecesariamente?). Por otro lado, la REGION es obligatoria para ambos ASESOR y CLIENTE, porque un cliente es obviamente localizado en alguna región, como es un asesor.

Probablemente encuentre situaciones complejas en la que las relaciones esperadas no son fáciles de determinar. Nuestro mensaje es simple: si no está seguro de una relación, responda si es razonable o cómo tales relaciones actualmente son manipuladas. Reflexione sobre la respuesta y llévelo a cabo, después describa a su entendimiento varias relaciones en la organización.

Reglas de negocios:

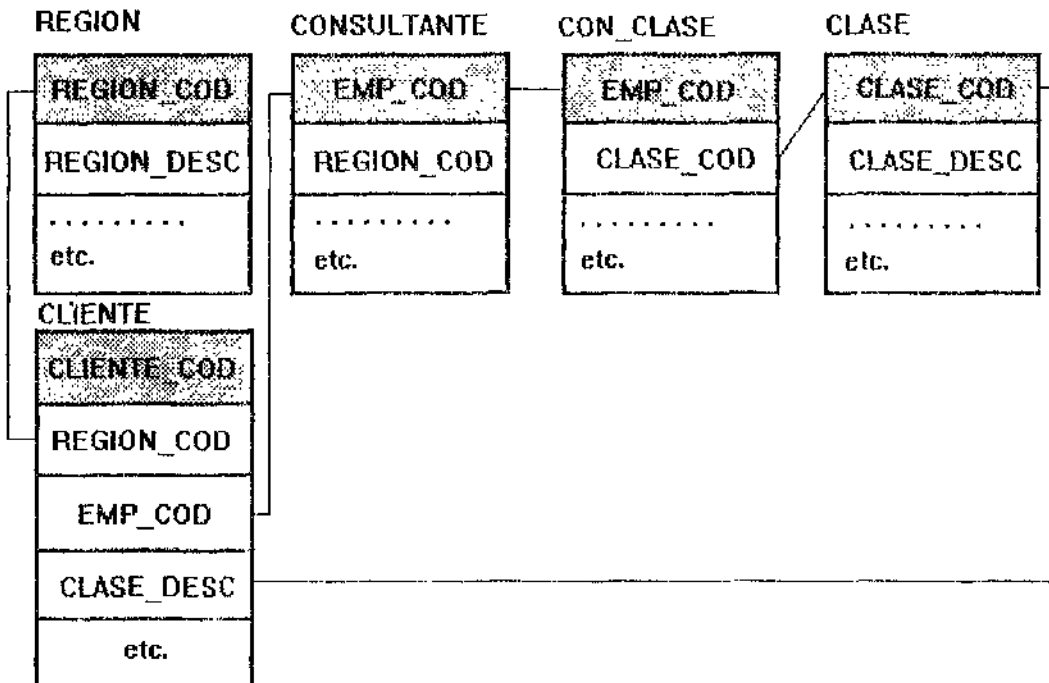
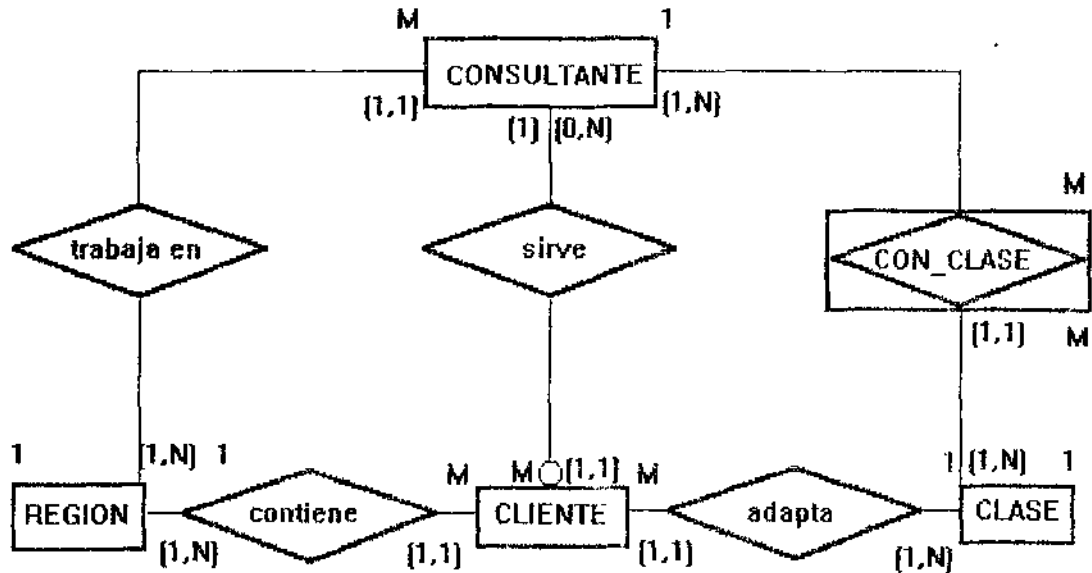
Leyendo la descripción del procedimiento de operaciones sería bueno ver las siguientes reglas exactas de negocios descritas en los datos de ROBWil.

1. Un asesor trabaja sólo en una región, pero cada región sirve a muchos asesores.
2. Un cliente es localizado sólo en una región, pero cada región puede tener muchos clientes.
3. Un asesor puede ser certificado para servir a los clientes en diferentes clasificaciones y cada clasificación puede ser manipulada por muchos asesores.
4. Un cliente da una clasificación, pero cada clasificación puede incluir muchos clientes.

El diagrama E-R, basado en éstas reglas de negocios, está representada en la figura 2.13. Note que la relación M:N entre ASESOR y CLASE ha sido resuelta por la creación de la entidad CON-CLASE y que la relación opcional ha sido marcada.

El esquema relacional mostrado en la figura 2.13 refleja sólo los más importantes atributos que permite el sistema para su función. (La tabla de datos de ROBWil contiene muchos atributos que son mostrados aquí). También, es muy probable que ROBWil tenga otros asesores. Por ejemplo, puede querer adicionar una entidad EMPLEADO para ser la superclase de la entidad subclase ASESOR. En resumen, ROBWil probablemente tenga un programa en beneficio del empleado, que requiera guardar la pista de los dependientes de empleados, así es que la entidad débil (opcional) DEPENDIENTE debería ser adicionada al diseño de base de datos

Figura 2.13 Base de datos de ROBWIL: Modelo E-R.



Fuente: Rob, 1995

Sin embargo, el diseño general es suficientemente flexible para permitir adicionar esas entidades y otras posteriormente.

Para darle una pequeña idea acerca del contenido de las tablas dentro de la base de datos ROBWil, tenemos ejemplos presentados en la figura 2.14. Para examinar el contenido de estas tablas fortalecemos la conectividad presentando la figura 2.13. Ejemplo, la figura 2.14 muestra lo siguiente:

- La tabla CON-CLASE tiene múltiples valores EMP-COD y CLASE-COD. (Note que la CLASE-BOD 'B' ocurre dos veces, como el código 'M'. Similarmente, el EMP-COD '10' ocurre dos veces y '11' ocurre tres veces). En otras palabras, CON-CLASE representa 'muchos' lados para ambos ASESOR y CLASE. (Examine esta relación en la figura 2.13).

- ASESOR y CLASE cada uno tiene el género EMP-COD y el valor CLASE-COD, respectivamente. La tabla CLASE contiene el valor del género CLASE-COD. Por lo tanto, estas entidades representan de '1' al lado de su relación para la entidad CON-CLASE. (Otra vez mire y observa que cada conectividad se muestra en la figura 2.13).

Figura 2.14 Ejemplo de tablas de la base de datos de ROBWil

Tabla: REGION	
REGION-COD	REGION-DESC
1	sudeste
2	sudoeste
3	noreste
4	medio oeste
5	noroeste
.....
etc.	etc.

Tabla: ASESOR	
EMP-COD	REGION-COD
10	2
11	1
12	2
13	3
14	1
15	5
.....
etc.	etc.

Tabla: CLASE	
CLASE-COD	CLASE-DESC
A	Aerolíneas
B	Inclinación Lateral
G	Administración
H	Hospitales y Serv. Fel.
M	Manufactura

Tabla: CON-CLASE	
EMP-COD	CLASE-COD
10	B
10	G
11	M
11	G
11	H
12	A
13	M
13	B
14	A
.....
etc.	etc.

Tabla: CLIENTE		
CLIENTE-COD	REGION-COD	EMP-COD
5003	2	10
5004	1	11
5005	5	15
5006	1	11
5007	1	14
5008	2	12
.....
etc.	etc.	etc.

Fuente: Rob, 1995.

2.4. Diseño del proceso

Los buenos modelos no se producen exactamente. Estos están basados en muchos detalles de la *descripción de operaciones* desde las cuales las reglas de negocios son derivadas. El modelo inicial E-R está sometido a un proceso conocido como *diseño de verificación*, para ver si la base de datos basada en éste contiene todos los requerimientos de operación y asegurar que la base de datos minimice la redundancia de datos.

Nota. En esta sección, exploraremos los mejores procesos de verificación de procesos. Sin embargo, *temporalmente* ignoraremos un importante procedimiento que detiene la redundancia de datos conocida como normalización. Este procedimiento es también importante, así que dedicaremos todo el capítulo tres a él. Aunque con la normalización detenemos, no podemos certificar la terminación de la verificación de procesos, este capítulo está enfocado a los mínimos puntos que requiere el diseño de la base de datos junto al diseño de los procesos.

Ilustrar el diseño de procesos es tan simple como posible, exploraremos una pequeña biblioteca con un sistema básico que checa las operaciones de entrada y salida de los libros. Nos dispondremos a examinar los defectos del sistema y brindar un nuevo entorno a la base de datos, dirigiendo el curso de las limitaciones del sistema. Después produciremos un diseño inicial, y lo modificaremos para encontrar la comprobación de los requerimientos. Nos enfocaremos siempre al desarrollo práctico de técnicas sobre el diseño de bases de datos y desarrollo de aplicaciones.

2.4.1. Descripción de operaciones

Una descripción de operaciones es un documento que establece la jerarquía organizacional y entorno básico de los datos. La definición de la jerarquía organizacional ayuda a establecer cambios y requerimientos en los reportes. La definición del entorno de los datos incluye requerimientos de los datos, el hardware disponible, y el software apropiado.

Una descripción del mundo real de los requerimientos en las operaciones y, una increíble cantidad de detalles aseguran que el diseño de la base de datos refleje precisamente los requerimientos en operaciones capaces de funcionar. La gente de los diferentes niveles de una organización perciben los requerimientos de manera distinta. Por ejemplo, los gerentes de niveles superiores cuidan el enfoque de las consideraciones estratégicas, los gerentes de niveles intermedios usualmente concentran sus tácticas en aspectos de operación organizacional, y los obreros se aburren encontrando diariamente detalles para los requerimientos operacionales. Una descripción precisa de operaciones está basada en intervenciones numerosas de personas que abarcan la jerarquía organizacional, la colección de formas, reportes precisos, y observaciones cuidadosas de las operaciones actuales.

La descripción de operaciones está sujeta a revisión hasta que personas de la jerarquía organizacional estén de acuerdo que éstas reflejan exactamente las operaciones de la organización. Esta es corregida con el propósito de incluir cambios por el avance técnico para mejorar el funcionamiento. Otras proposiciones son generadas internamente, y diseñarlas puede también proponer cambios para mejorar el funcionamiento de forma eficiente y flexible.

Una descripción de las funciones que usualmente incluyen esos componentes es:

- Una descripción, frecuentemente está basada en una tabla organizacional de la *jerarquía organizacional*. Propósito: definir lo extenso y complejo de los cambios reportados.

- Una descripción básica del entorno o *medio ambiente de los datos*. Propósito: definir lo extenso y complejo de la disponibilidad de los datos y anticipar los cambios en este entorno. Una organización con cien transacciones diarias y diez variables probablemente requiera un nivel diferente de diseño y aplicación, al complejo desarrollo de una organización con diez mil transacciones diarias y cien variables.
- Una descripción de los *recursos disponibles* incluyendo personal, hardware y software para generar, almacenar y administrar datos. Propósito: definir el diseño con sus límites y oportunidades. Una Computadora Intel 386 con 1 MB de RAM y un disco duro de 100 MB operado en parte por un empleado inexperto, podría desarrollar diferentes posibilidades para un ambiente en red y empleando personal profesional.
- Una descripción de *operaciones generales*. Propósito: definir los fundamentos en los que se basan para el modelo inicial E-R (entidad-relación). ¿Cuántas entidades hay? ¿Cuáles son sus relaciones? ¿Qué atributos deben ser incluidos? Este es el lugar donde muchas de las reglas de negocios se incluyen al sistema.
- Una descripción de *problemas y limitaciones* generales. Propósito: ayudar a definir el modelo inicial E-R
- Una descripción de *operaciones propuestas*. Propósito: realizar el modelo inicial E-R y verificar si éste es hábil para encontrar el diseño operacional estándar.

Porque si separamos las restricciones podremos dar una descripción de las limitaciones de las operaciones, describiendo parte del entorno de la biblioteca. Sin embargo, podremos llegar a tener un conocimiento amplio de los procesos y ver cómo éstos nos llevan a un apropiado diagrama E-R.

Jerarquía organizacional

La Biblioteca es fundada por el condado y emplea un bibliotecario que supervisa a tres empleados con sueldo y a tres empleados voluntarios sin pago. El bibliotecario trabaja con un voluntario, quinto miembro de la biblioteca, que encartona y realiza reportes a los dirigentes del condado, mínimo una vez al mes. El dirigente del condado requiere reportes periódicos concernientes al uso de la biblioteca, adquisición de libros, y condiciones de los mismos. El bibliotecario quiere monitorear los procedimientos de entrada y salida de los libros, y producir un reporte periódico de los libros usados por categorías (ciencia ficción, computación, historia y demás) y ayudar como guía en las decisiones de adquisición.

Medio ambiente de la base de datos

La biblioteca del condado ha coleccionado más de ochenta mil libros y sirve a más de veinticinco mil residentes del condado, casi cuatro mil tienen su propia credencial de biblioteca. La biblioteca posee credenciales que son conocidas como *patrones*. La biblioteca diariamente realiza la función de servicio al cliente manejando el proceso de envío de libros y el servicio de escritorio recordándoles cuándo deben regresarlo a la biblioteca. En cada patrón del libro se checa su entrada y salida contactando si está considerado como una transacción. La biblioteca en el servicio de escritorio también da salida a nuevas credenciales. Los principales datos de la biblioteca están resumidos en la tabla 2.1.

Tabla 2.1. Entorno de la base de datos

	1993	1994	1995 (estimado)
Número de libros	78,124	80,012	85,000
Número de patrones	3,876	3,991	4,350
Número de transacciones por día	80 (est)	85(est)	95

Fuente: Rob, 1995.

La biblioteca conserva un sistema de clasificación manual en el cual los registros patrón y libro son guardados. Los registros patrón incluyen nombre, dirección, teléfono, el número de tarjeta de la biblioteca y fecha de expedición. Los registros libro incluyen título, autor, fecha de compra y editorial. Un catálogo de tarjetas libro contiene un conjunto de tarjetas listadas alfabéticamente por título y, un conjunto de tarjetas listadas alfabéticamente por autor.

Recursos disponibles

La biblioteca emplea un bibliotecario, un asistente de bibliotecario, y tres empleados de personal. Tres empleados voluntarios están disponibles diariamente. Los comisionados del condado aprobaron la contratación adicional de asistentes de bibliotecario durante el próximo año fiscal.

La biblioteca tiene tres computadoras propias, que son usadas para manejar la nómina y el presupuesto. Las microcomputadoras están también equipadas con un procesador de palabras básico, una hoja de cálculo, y comparten una sola impresora láser. La tabla 2.2. resume las características de hardware de las microcomputadoras.

Tabla 2.2. Características de hardware.

PROCESADOR	RAM	DISCO DURO	MONITOR	MODEM	FAX
Intel 486SX	4MB	200MB	VGA, 28 mm	no	no
Intel 486SX	4MB	200MB	VGA, 28 mm	no	no
Intel 486DX2	16MB	500MB	Super VGA, 28 mm	si	si

Fuente: Rob, 1995.

La biblioteca no tiene personal asignado a manejar el recurso de las computadoras, ni está usando formalmente software especializado. Además no está autorizada a utilizar fondos para aumentar computadoras o impresoras. Sin embargo, después de una discusión se propuso el proceso de automatización, la biblioteca consintió invertir fondos para el software necesario de la bases de datos y para un lector de código de barras. El bibliotecario ha aceptado llegar a ser el administrador de la base de datos del proyecto de la biblioteca.

Operaciones generales

Al tomar prestado un libro de la biblioteca, el cliente (patrón) debe presentar su credencial de la biblioteca (un ejemplo es mostrado en la figura 2.15) y el libro(s) es checado por un policía que vigila las salidas, esto es simple: no presenta credencial, no se le prestan libros.

Figura 2.15. Credencial de biblioteca

Credencial Número: 2238	Fecha de Expedición:
02/12/91	
Nombre: José Luis	Apellido: Ramírez
Inicial: S	
Dirección: 33 Oriente 3006. C.P. 72510	
Firma: _____	

Fuente: Rob, 1995

Cuando la biblioteca expide una credencial, un empleado registra los datos del patrón en orden alfabético (nombre, apellido, inicial) y los anexa al fichero de tarjetas. El fichero de entradas incluye los datos encontrados en la credencial de la biblioteca.

El procedimiento de salida de un libro sigue la secuencia siguiente:

- Una credencial patrón es checada contra el fichero de números de credenciales extraviadas.
- Si el número de credencial no es reportada como perdida, el número es usado para checar libros atrasados. Si se atrasó en un libro, no se le pueden dar libros adicionales, a no ser que se regrese el libro atrasado y pague la multa.
- Si no hay cambios contra la credencial patrón, la tarjeta del libro es usada para el préstamo del libro, y se escribe la información correspondiente. El personal empleado verifica la información, estampa en la tarjeta del libro la fecha de vencimiento, quita la tarjeta del libro que está sujeta a él por un clip, colocándola con vencimientos iguales en el fichero de tarjetas libro, y da el libro a la persona. El proceso es repetido para cada libro que va a salir.

Un ejemplo de tarjeta libro es mostrado en la figura 2.16.

Las tarjetas de los libros son recogidas al iniciar cada mañana, son almacenadas en un archivo con el código "libro de salida" en orden y por fecha. La tarjeta es almacenada, en los agujeros del lado de la tarjeta se localiza la notación de martes, jueves y sábado para cada semana, y se guarda en el archivero con la cara hacia arriba y de lado.

Cada mañana, un miembro del personal checa el archivo, para los libros vencidos corre una barra de metal directamente por los agujeros y levanta la barra. Si un libro no está vencido, se encaja sujetándolo y sale la tarjeta del libro permitiendo a la barra deslizarla directamente. Desde el encajamiento no se sujeta a ninguna fecha pasada, debido a que los libros se localizan sólo cuando la tarjeta se levantó. Un miembro del personal recolecta cada una de las tarjetas de libros vencidos y le

envía una nota al patrón a la dirección de su tarjeta. La tarjeta del libro es clasificada por el número del patrón en un archivo de libros vencidos.

Figura 2.16. Ejemplo de tarjeta para los libros

Libro código:	Patrón número:	4 ^o Sem. Martes <input type="radio"/>
		Jueves <input type="radio"/>
Vencimiento:	Regresado por:	Sábado <input type="radio"/>
Cantidad debida: 5		3 ^o Sem. Martes <input type="radio"/>
		Jueves <input type="radio"/>
Nombre del autor:		Sábado <input type="radio"/>
Apellido del autor:		2 ^o Sem. Martes <input type="radio"/>
Inicial del autor:		Jueves <input type="radio"/>
Título del libro:		Sábado <input type="radio"/>
Editorial:		1 ^o Sem. Martes <input type="radio"/>
		Jueves <input type="radio"/>
		Sábado <input type="radio"/>

Fuente: Rob, 1995

Los libros regresados siguen el siguiente procedimiento:

- Cuando un libro es regresado, el personal de la biblioteca recoge la tarjeta del libro del fichero libros de salida. Si la tarjeta del libro no es encontrada en el fichero tarjetas del libro, el libro atrasado es detenido.
- Una vez que la tarjeta del libro ha sido recogida, el personal registra la fecha de entrega del libro en la tarjeta y, si es necesario, se hace acreedor a una multa.
- Si una tarjeta del libro se ha unido de manera equivocada a la de control, el libro es llevado a un estante de libros reservados y la persona que reserva el libro lo notifica por teléfono.
- Si la tarjeta no se ha unido a la tarjeta de control, el libro es llevado al estante libros que se tienen que volver a archivar, el lugar de apoyo en la biblioteca es un estante en el que se apilan los libros al final del día.

Si un patrón quiere averiguar los libros que actualmente tiene en préstamo, el dependiente llena una tarjeta de control, mostrada en la figura 2.17. La tarjeta es unida a la tarjeta del libro y es devuelta cuando el procedimiento de regreso de un libro es seguido. Si un libro es reservado se coloca en el estante de reserva y el patrón es notificado. Si un libro reservado no ha sido recogido después de tres días a la notificación, es regresado a los estantes.

Figura: 2.17. Tarjeta de control

Nº de libro: _____	Reservado por: _____
Título: _____	
Autor: _____	
Categoría: _____	
Fecha reservación: _____	Teléfono: _____
Notificación fecha: _____	

Fuente: Rob, 1995

Limitaciones y problemas

El sistema manual actual es lento, incómodo, y con numerosos errores. El personal se queja de lo siguiente:

- Muchos de los errores de las operaciones son causados por el sistema de tarjetas, el cual presenta equivocaciones. La información equivocada causa problemas en la notificación de reserva de un libro, siendo inexacto su registro y colección.
- El manual registra los códigos del libro, las credenciales patrón con su número y fecha de expedición, produciendo datos de entrada erróneos que son difíciles de encontrar y corregir.
- El sistema no permite generar reportes detallados, así hace difícil tomar decisiones apropiadas concernientes a la adquisición de libros, hacer una muestra de libros, reunir las multas, y demás.
- No es posible encontrar ¿cuántos patrones tienen libros detenidos? y actualmente ¿cuántos libros tiene cada patrón en préstamo?.
- A pesar de que la biblioteca quiere hacer compras, las decisiones son basadas en categorías por la popularidad de los libros, y la información por categorías no está disponible. Si estuvo disponible, el sistema de clasificación general no puede manejar otro sistema de clasificación.
- Los libro y los datos del patrón existen en varios ficheros, generando redundancia de datos, actualizando y borrando anomalías, así administrar los datos esenciales es difícil.
- El sistema general es un trabajo extenso. Por ejemplo, escribir tarjetas atrasadas que toman mucho tiempo coger y relacionar dentro de los límites permitidos al empleado.
- La biblioteca agrega miembros que quieren iniciar una campaña para el aumento de los fondos y esto requiere de escribir muchas cartas solicitando donaciones y agradeciendo contribuciones. Otro grupo

de archivos son generados al guardar las contribuciones y los contribuyentes. Generalmente el personal no puede manejar todas las actividades adicionales.

Operaciones propuestas

Limitar al personal a lo más general, ocupando el tiempo en la faena diaria, generar un reporte detallado y cuestionar su capacidad; el sistema manual debe ser reemplazado por un sistema basado en la computadora. Ayudar a eliminar datos de entrada erróneos, el procedimiento de entrada de datos debe ser automatizado. Las credenciales y libros deben tener un código de barras que permita al personal dar entrada a los datos al leer el código de barras. Los datos de entrada deben ser automatizados a través del uso de un sistema de cómputo. Sin embargo, el sistema debe ser lo suficientemente flexible que permita dar entradas por medio del teclado para modificar datos incorrectos del sistema, lectura de un código de barras defectuoso, o modificar datos de entrada por excepciones.

Manejar los recursos de la biblioteca y proveer de servicios más eficientes, el sistema debe generar formas de entrada y salida de libros, escribir e imprimir direcciones a cartas automáticamente, generar reportes detallados, y suministrar datos eficientes por la administración del sistema. Porque la biblioteca quiere generar fondos adicionales a través del aumento de algunas actividades, el sistema debe poder guardar los datos de patrones donadores que aumentan sus dádivas para escribirles una carta de agradecimiento. El personal de la biblioteca no está familiarizado con operaciones computacionales, pero no hay fondos para un empleado que maneje el sistema de información administrativa (MIS). El entrenamiento de los requisitos debe ser mínimo.

La capacidad de reporte del sistema debe ser incorporado con un sistema de menú. El personal de la biblioteca debe generar reportes con los datos clasificados. El reporte debe incluir:

- Libro por categoría, autor o título.
- Presentar una colección total y subtotal por categoría del libro.
- Datos de los patrones donadores.
- Libro de inventario, seleccionado por categoría y fecha de compra.
- Listas generales de teléfonos de los patrones.

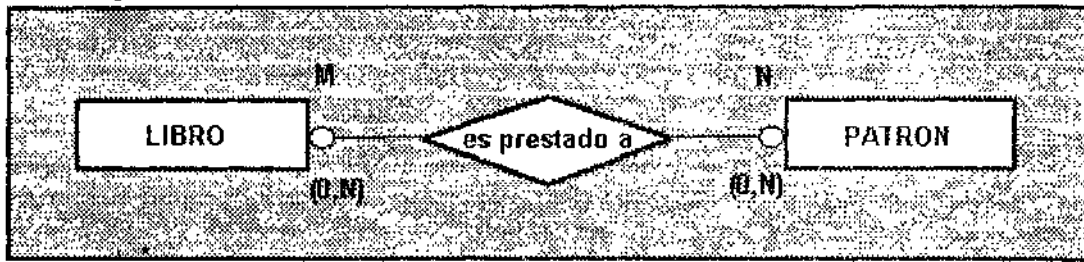
2.4.2. Diseño inicial

Dando esta breve y simplificada descripción de las operaciones, es claro que las operaciones de entrada y salida implican una relación entre patrones y libros. Especialmente, podemos escribir las reglas de negocios como siguen:

- Un libro puede ser prestado a muchos patrones y un patrón puede pedir prestado muchos libros. Hay una relación M:N entre LIBRO y PATRON.
- Un patrón no está pidiendo libros, así que LIBRO es opcional al PATRON.
- Un libro puede existir sin ser sacado en préstamo. De hecho, bastantes libros son sólo de referencia, nunca salen de la biblioteca. Otros libros pueden ser simplemente no escogidos por algún patrón interesado, así PATRON es opcional a LIBRO.

Este conjunto de reglas de negocios producen un diagrama inicial E-R mostrado en la figura 2.18.

Figura: 2.18. Diagrama inicial E-R de la biblioteca



Fuente: Rob, 1995

Atributos requeridos

Encontrar los requisitos de operación, la entidad LIBRO debe contener los atributos:

LIBRO_CODIGO, LIBRO_TITULO
 LIBRO_A_NOMBRE, LIBRO_A_APELLIDO, LIBRO_A_INICIAL
 LIBRO_PUBLICACION
 LIBRO_CATEGORIA
 LIBRO_FECHACOMP
 LIBRO_ESTADO

El atributo LIBRO-CODIGO produce propiamente una llave primaria. El nombre del autor es descompuesto en nombre, apellido e inicial porque frecuentemente los clientes preguntan de libros por autor y el personal debe efectuar una búsqueda conveniente. El atributo LIBRO_CATEGORIA es requerido porque la librería quiere poder reportar libros utilizados por cada categoría como ciencia de la naturaleza, computación, suspenso y demás. El atributo LIBRO-ESTADO es requerido porque el personal de la librería quiere poder contestar a los clientes preguntas acerca de los libros disponibles. Si el libro está disponible, el atributo debe tener el valor de "si". Si el libro ha sido prestado, el sistema debe insertar "no".

La entidad PATRON debe contener los atributos:

PATRON_CÓDIGO
 PATRON_TITULO, PATRON_NOMBRE, PATRON_APELLIDO, PATRON_INICIAL
 PATRON_DIRECCION, PATRON_CIUADAD, PATRON_ESTADO, PATRON_CP
 PATRON_TELEFONO
 PATRON_TOT_C, PATRON_ULT_C, PATRON_C_FECHA

El código del patrón sería la llave primaria. El uso del PATRON_TITULO (Sr., Sra., Srita., Dr.) refleja una de nuestras propias recomendaciones. Si periódicamente preguntan a los patrones si desean contribuir o le pide libros, ¿por qué no ser cortés?. Poca gente se opone a una excelente confianza. ¿Por qué no dirigir las cartas de la forma Dr. Sánchez, más bien que Luis? Se vería más profesional que "Apreciado Luis" si no conoce personalmente al Dr. Luis Sánchez.

Los últimos tres atributos de PATRON: PATRON_TOT_C, PATRON_ULT_C, y PATRON_C_FECHA, reflejan el hecho de que la biblioteca agrega miembros (no lo mencionan al bibliotecario) a los que les interesan las actividades para el aumento de fondos. Especialmente se preguntan y contestan periódicamente.

- ¿Qué patrones han contribuido aumentando fondos para la biblioteca? Teniendo el atributo de la contribución total del patrón, nombrado como PATRON_TOT_C, podríamos responder estas preguntas fácilmente. Claramente, si el valor es más de \$ 0.00, el patrón ha contribuido.
- ¿Con cuánto dinero en total contribuyó un patrón? Nuevamente el uso de PATRON_TOT_C contestaría esta pregunta fácilmente.
- ¿Cuándo fue la última contribución del patrón? La fecha presentada por PATRON_C_FECHA respondería esta pregunta.
- ¿Con cuánto un patrón contribuyó la última vez? Ese ¿por qué? podemos incluirlo a cantidad contribuida recientemente PATRON_ULT_C.

A pesar de que ahora definimos las entidades principales del sistema y sus atributos, la pregunta crucial es "¿este diseño puede ser implementado adecuadamente?" la contestación es respondiendo "¡no!".

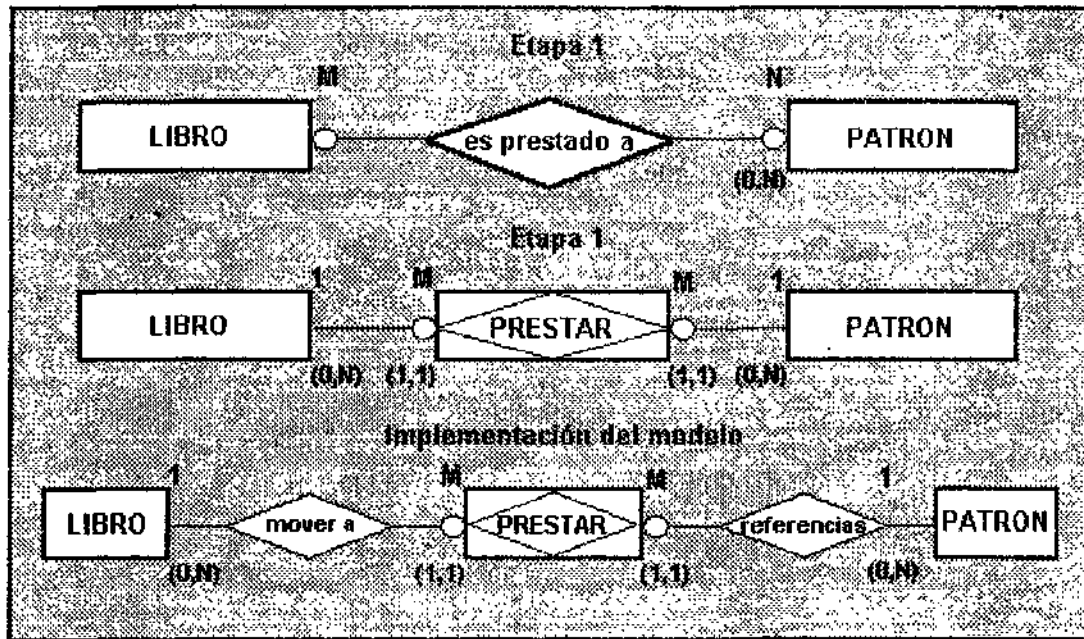
2.4.3. Verificación del diseño

Uno de nuestros diseños tomados es intentar minimizar la redundancia de datos. Desafortunadamente, la relación M:N presentada en la figura 2.18 produce muchas redundancias. Por ejemplo, dando el diagrama inicial E-R mostrado en la figura 2.18, nosotros potencialmente encontraríamos múltiples entidades libro para cada patrón, y cada libro potencialmente tiene muchas entidades patrón asociadas a ésta. Por lo tanto, debemos convertir la relación M:N produciendo la figura 2.19, la etapa uno hasta el conjunto de las dos relaciones 1:M, mostradas en la etapa dos. Representa que la actividad primaria de la biblioteca es prestar libros a patrones, la etapa dos está compuesta por la entidad que ha sido nombrada como PRESTAR.

La figura 2.19 merece un cuidadoso examen, porque la composición de las entidades presenta cambios esenciales en los cardinales y en la ubicación de las relaciones opcionales. Especialmente, note que:

- Cada libro tiene lugar sólo en la entidad LIBRO. Por lo tanto, la conexión LIBRO es uno. Similarmente, cada patrón tiene lugar en la entidad PATRON, así la conexión de PATRON es también uno. Dando estos cambios en las conexiones, es claro que la redundancia de datos es reducida substancialmente.
- PRESTAR es opcional a ambos LIBRO y PATRON. No hay requisito en el que un LIBRO deba ser pedido en préstamo para existir, no hay un requisito para que el PATRON deba pedir prestado libros. Porque PEDIR es insertado entre LIBRO y PATRON, es opcional el símbolo mover a PRESTAR.
- Cada ocurrencia en la entidad PRESTAR remite sólo un LIBRO y un PATRON. Por lo tanto, los cardinales de PRESTAR son (1,1) relativos a las entidades LIBRO y PATRON.
- A pesar de que la mayor parte de los diagramas E-R se detienen en la etapa dos, el uso de algunas herramientas de ingeniería de sistemas asistido por computadora (CASE) requieren que la relación entre LIBRO y PRESTAR, y entre PRESTAR y PATRON sea especificada como se muestra en la implementación del modelo.

Figura 2.19. Creando un modelo E-R que puede ser implementado



Fuente: Rob, 1995.

El diseño inicial también hace impráctico guardar el camino de quienes sacaron prestado libros, cuando sacan prestado libros están en deuda, qué multan si así se valoró, y de todos modos saber si la multa fue pagada. Evidentemente, el diseño inicial es incapaz de reunir los requisitos del usuario. Esas actividades, también, pueden acomodarse a través de la entidad compuesta **PRESTAR**. En otras palabras, **PRESTAR** debe servir dos propósitos:

1. Debe substancialmente decrecer la redundancia de datos.
2. Debe ayudar a reunir los requerimientos del usuario.

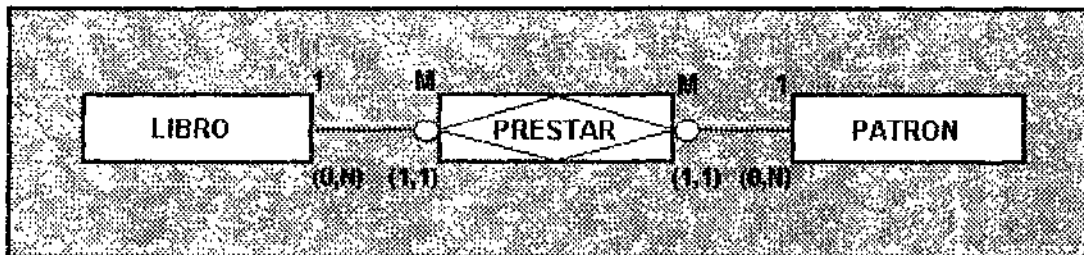
Note que la inclusión del seguimiento de los atributos cumple dos propósitos:

- La entidad compuesta **PRESTAR** uniría las tablas **LIBRO** y **PATRON** a través de las dos llaves externas de la tabla **PRESTAR**: **PATRON_CODIGO** y **LIBRO_CODIGO**.
- La tabla **PRESTAR** debe tener su propia llave primaria **PRESTAR_NUMERO**. Aunque la entidad compuesta que contiene el **PATRON_CODIGO** del **PATRON** y el **LIBRO_CODIGO** del **LIBRO**, la combinación de esas llaves externas no ordenan en este caso, constituyen una llave primaria suficiente. Por ejemplo, si un patrón pide prestado un libro una vez más, la combinación **PATRON_CODIGO** y **LIBRO_CODIGO** no es única, así violaría el requisito que todos los valores de las llaves primarias deben ser únicos. Pensando en cada ocurrencia **PRESTAR** como *transacción* única.

- PRESTAR_FECHA muestra la fecha en la que el pedido fue hecho. Minimizando la posibilidad de errores humanos en la entrada de datos, permitiendo al sistema introducir la fecha corriente del mismo.
- PRESTAR_DEUDA indica la fecha en la cual el libro es devuelto. Nuevamente, minimizar la posibilidad de errores humanos en los datos de entrada, el sistema puede agregar el número de días al pedido normal a PRESTAR_FECHA produciendo la fecha de devolución. Naturalmente, quisiéramos poder modificar esta fecha a través del uso del teclado que permita al propietario tener libros en diferentes periodos de tiempo.
- PRESTAR_REGRESO es registrada por el sistema cuando el libro es regresado realmente.
- La multa para cada libro es de veinticinco centavos por día, PRESTAR_MULTA es calculado por el sistema, tomando la diferencia entre la fecha de regreso y fecha de deuda, y multiplicando el resultado por veinticinco centavos. En otras palabras, PRESTAR_MULTA es producido por la transformación de $(PRESTAR_REGRESO - PRESTAR_DEUDA) * 0.25$.
- PRESTAR_PAGADO indica si o no, dependiendo si la multa fue pagada.

El proceso de verificación produce la revisión del diagrama E-R mostrado en la figura 2.20.

Figura 2.20. La revisión del diagrama E-R de la base de datos de la BIBLIOTECA.



LIBRO
LIBRO_CODIGO
LIBRO_TITULO
LIBRO_A_NOMBRE
LIBRO_A_APELLIDO
LIBRO_A_INICIAL
LIBRO_PUBLICACION
LIBRO_CATEGORIA
LIBRO_FECHACOMP
LIBRO_ESTADO

PRESTAR
PRESTAR_NUMERO
PATRON_CODIGO
LIBRO_CODIGO
PRESTAR_FECHA
PRESTAR_DEUDA
PRESTAR_REGRESO
PRESTAR_MULTA
PRESTAR_PAGADO

PATRON
PATRON_CODIGO
PATRON_TITULO
PATRON_NOMBRE
PATRON_APELLIDO
PATRON_INICIAL
PATRON_DIRECCION
PATRON_CIUADAD
PATRON_ESTADO
PATRON_CP
PATRON_TELEFONO
PATRON_TOT_C
PATRON_ULT_C
PATRON_C_FECHA

Fuente: Rob, 1995.

Si examina la figura 2.20, tenga en mente que la entidad PRESTAR requiere a LIBRO igual que a PATRON. Después de todo, la entrada del dato PRESTAR no es hecha a no ser que alguna persona pida un libro. Por lo tanto, cada entidad PRESTAR requiere de un LIBRO, así la cardinalidad (1,1) es apropiada para PRESTAR un LIBRO. Cada entidad PRESTAR ocurre del mismo modo cuando requiere un libro el PATRON. Así la cardinalidad (1,1) es apropiada para PRESTAR a PATRON. Sin embargo, desde un PATRON no podemos sacar prestado LIBROS, un PATRON jamás puede hacer resaltar la entidad PRESTAR, así la cardinalidad (0,N) es apropiada para PATRON a PRESTAR. La cardinalidad entre LIBRO y PRESTAR es también (0,N) porque un LIBRO no requiere ser sacado en préstamo.

Pudo haber ocurrido que el procedimiento de sacar un libro en préstamo puede parecerse a la operación de facturación. Por ejemplo, algún libro sacado en préstamo puede requerir un patrón (uno o más libros), justo como en una factura que requiere de un cliente que compre uno o más artículos. Por lo tanto el diagrama E-R presentado en la figura 2.21 es completamente apropiado. Note que la composición de la entidad PRESTAR ha sido partida en dos entidades: PRESTAR y PRESTAR_LINEA, justo como en una factura puede ser dividida en FACTURA e INV_LINEA. De la figura 2.21 la entidad PRESTAR contiene los atributos:

PRESTAR_NUMERO
PRESTAR_FECHA
PATRON_CODIGO

Y la composición de la entidad PEDIR_LINEA contiene los atributos:

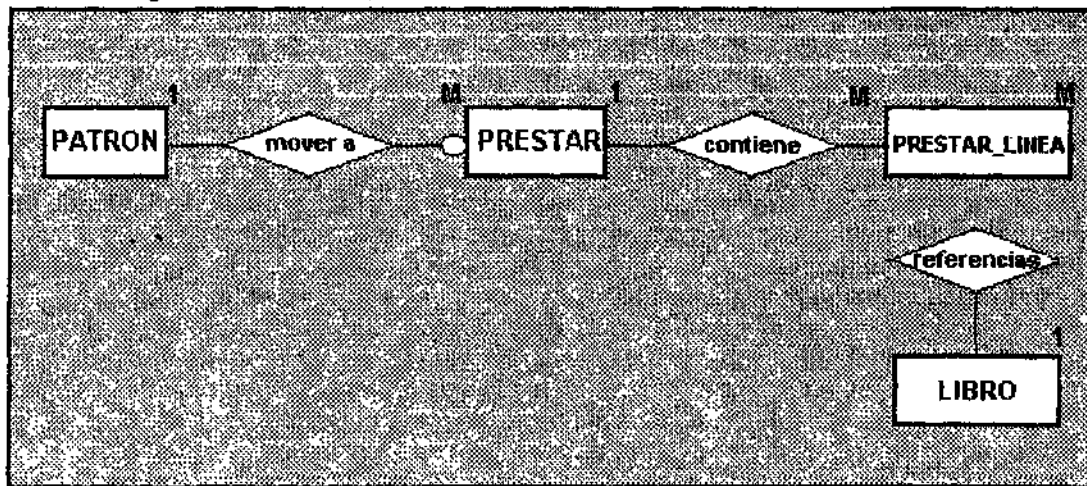
PRESTAR_NUMERO
LIBRO_CODIGO
PRESTAR_FECHA
PRESTAR_DEUDA
PRESTAR_REGRESO
PRESTAR_MULTA
PRESTAR_PAGADO

Los diagramas E-R presentados en las figuras 2.20 y 2.21 cada uno reúne los requisitos del diseño. Sin embargo, es distinta a la típica operación de facturación en la que se vio un producto o servicio en el cual usualmente no regresa a ser procesado, en la biblioteca la operación de prestar requiere del regreso de los libros prestados. El diagrama de la figura 2.20 produce menos puntos de operación que el diagrama de la figura 2.21, cuando éste llega al procedimiento de devolver el libro, la forma es mejorar la conveniencia de la operación de la biblioteca y hemos seleccionado éste como el diseño final.

2.5. Exclusiones del diseño

Debemos insistir que el modelo E-R no puede y no incluye todos los atributos del mundo real, muchos de los cuales podemos derivar de atributos existentes. Por ejemplo, la edad de una persona puede ser calculada vía almacenar la fecha de nacimiento, y una factura en dólares puede ser calculada al multiplicar el número de unidades y el precio por unidad. Pregúntese usted mismo si el sistema es más o menos eficiente si un atributo derivado es incluido en el diseño. Usar un diseño simple, regla. no incluir ninguna cosa en el diseño de base de datos, ese es más eficiente y efectivo ocupando aplicaciones de software.

Figura 2.21. Un diagrama E-R alterno, de la biblioteca



Fuente: Rob, 1995

En un diseño tampoco es necesario incluir todas las posibles entidades del mundo real. Conceptualmente la simplicidad es crucial, como lo es la eficiencia. Por ejemplo, ningún sistema de base de datos empieza a perder funcionalidad cuando carga a éste de tablas largas innecesarias. Por ejemplo, examinaremos el diseño de datos de la biblioteca: una vez que se pide prestado un libro se tiene que regresar y ningún pago debe ser aplicado, hacer esto en el sentido de que se guarda el registro PRESTAR en una tabla archivada. Una simple tabla archivada conserva la historia de los registros de las transacciones. Teniendo tal tabla almacenada, substancialmente decremента el número de registros en la misma tabla activa PRESTAR, por eso incrementa (misma actividad) la eficiencia de la tabla PRESTAR.

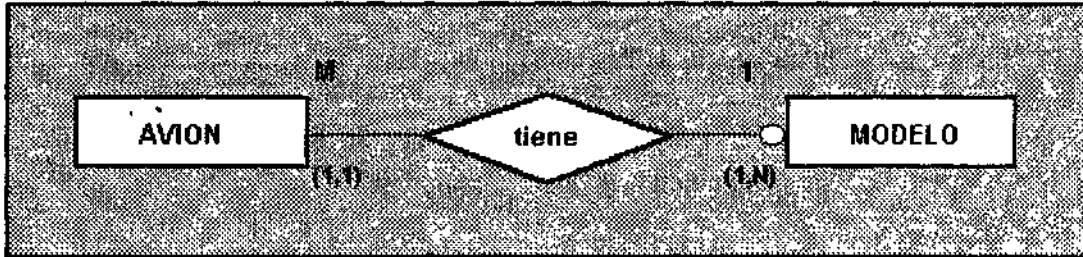
Una pequeña biblioteca puede generar un largo número de transacciones de PRESTAR durante el año, así el archivo tabla es una importante función en el mundo real del campo de la base de datos. Naturalmente, puede poner en duda un archivo tabla justo como usted haría otra tabla.

Y esta discusión nos hace reforzar el primer punto: pocas actividades que van dentro de la base de datos son efectuadas por el software externo del diseño básico. Algunas actividades, como el respaldar y archivar, simplemente no se prestan bien al modelado E-R y no es incluido en el modelo E-R. Por ejemplo, la actividad de guardar simples copias de tuplas seleccionadas de la tabla PRESTAR del archivo tabla no está definida la relación entre esas dos tablas; la revisión del modelo E-R no ha sido hecha, por lo tanto, incluir el archivo tabla.

Problemas:

2.1. Escriba la regla(s) de negocios reflejadas en el siguiente diagrama:

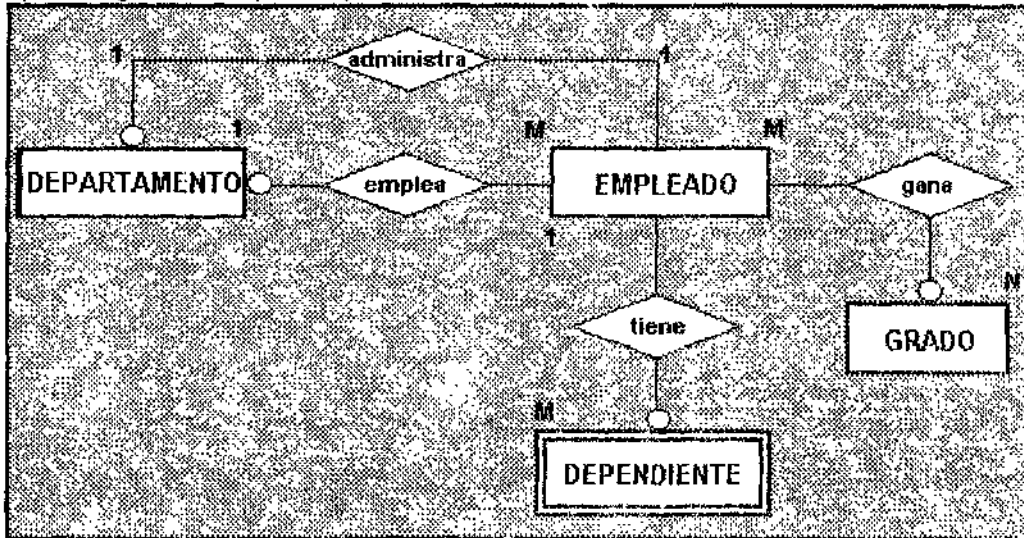
Figura 2.p1. Diagrama E-R para el problema 2.1.



Fuente: Rob, 1995

2.2. Indique todas las cardinalidades en el siguiente diagrama E-R, escriba todas las reglas apropiadas al negocio.

Figura 2.p2. Diagrama E-R para el problema 2.2.



Fuente: Rob, 1995

2.3. Convierta la relación M:N del problema 2.2. a una relación 1:M, uniéndolo a través de una entidad llamada EMP_DEG. Vuelva a dibujar el diagrama en el problema 2.2 reflejando las modificaciones que hizo.

2.4. Realice un conjunto de estructuras de tablas que sean razonables de acuerdo a la base de datos que usted representaste en el diagrama E-R del problema 2.3.

2.5. Dibuje la relación del esquema para la base de datos representada por usted de los problemas 2.3 y 2.4.

2.6. Dando la siguiente descripción de operaciones, escriba las reglas apropiadas al negocio, creando un diagrama E-R basado en esas reglas de negocios. Hacer seguro su diagrama E-R conteniendo todas las entidades apropiadas, relaciones, conexiones y cardinalidades:

Consultando una compañía suministrando servicios especializados sin perder la administración del ingeniero civil, ingeniero electrónico, e intervención/ diseño/ implementación/ aplicaciones desarrolladas en la base de datos. Cada vez que consultamos un contrato es escrito cubriendo una especial y específica consulta. (Cada consulta especializada a un área es dirigida por una apropiada visión de la compañía). Cada contrato asciende a proyecto y a cada proyecto se le asigna un nombre código, como es Sobre, Ultra, y demás.

Cuando un nuevo contrato es recibido, es asignado a la división, el gerente asigna el personal apropiado para trabajar en cada proyecto. El gerente también coordina la división de las actividades para asegurar el éxito en la complementación del proyecto. Asegurar que la división de cada proyecto reúna el talento al máximo de los empleados dentro de la división que se les asignó en más de un proyecto en el tiempo. Sin embargo, los empleados que son asignados a una división específica no se distribuyen en otras divisiones. Cada uno de los proyectos son dirigidos por un director del proyecto y un coordinador técnico.

A pesar de que muchos empleados son asignados a una división específica, algunos, principalmente recepcionistas, son "exploradores". Esto es, ellos son parte de los miembros que son asignados como necesarios. Todos los empleados y sus dependientes reciben un beneficio estándar en conjunto.

Guardar el camino de los detalles de la cuenta de cada proyecto, esto debe ser posible asignando las horas generadas por cada empleado asignado al proyecto. El valor del dólar para cada número de horas del empleado depende del trabajo primario en el que se le clasificó, como es un analista de sistemas, un especialista en red, analista en transportación, ingeniero electrónico, especialista en el diseño de base de datos, y demás.

2.7. Escriba la relación del esquema en la estructura de la tabla para la base de datos abajo del diagrama E-R que forme del esquema 2.6.

2.8. Dando la base de datos mostrada en el capítulo uno problema 1.1., complemente el diagrama E-R. Incluya todas las entidades, conectividades, cardinalidades, y relaciones.

2.9. Dibuje el esquema relacional para el diagrama E-R del problema 2.8.

2.10. Dando la siguiente descripción de operaciones, realice un apropiado diagrama E-R. (Para no tener una idea errónea visita un aeropuerto, preferentemente uno pequeño, que genere detalles adicionales en la renta básica de un avión y describa las operaciones de instrucciones de vuelo).

La compañía de aviación ROBWIL es un pequeño operador de base armada (OBF) localizado en un pequeño aeropuerto. El OBF renta aviones para clientes calificados y proporciona instructores de vuelos que ayudan a la gente a aprender la licencia de piloto privado o licencia de piloto comercial. La instrucción puede también ser dada para aprender en la reserva adicional, esto como un instrumento y valoración de múltiples máquinas. La instrucción de vuelo puede ser dada en un avión rentado de ROBWIL o en un avión de un cliente.

El propio ROBWil tiene cinco aviones que son descritos en la figura 2.p10.

Figura 2.p10

Nº avión	Modelo Código	Nombre Modelo	Fabricante	Asientos	Motors	Equipo	Modelo año	TTAF	TTEL	TTER	Tarifa/hr
2085V	PA28-181	Arquero II	Aeronaves Piper	4	1	Fijo	1979	1876.5	2876.5	0.0	\$ 56.75
1189X	C-152	Viajero	Aeronaves Cessna	2	1	Fijo	1981	4987.3	784.5	0.0	\$ 45.00
2310U	C-172	Skyhawk	Aeronaves Cessna	4	1	Fijo	1975	3890.2	1752.9	0.0	\$ 49.50
4899D	C-152	Viajero	Aeronaves Cessna	2	1	Fijo	1983	5241.9	421.8	0.0	\$ 45.00
9733E	PA34-220	Séneca III	Aeronaves Piper	6	2	Retractor	1983	2783.4	986.2	1023.6	\$132.50

Fuente: Rob, 1995

El gerente de ROBWil puede guardar la trayectoria del tiempo total, midiendo la décima parte más cercana a una hora, acumular las horas de vuelo y motor. El tiempo es registrado en base a un instrumento conocido como Hobb, contador que es instalado en cada cabina del avión. El tiempo total de vuelo es etiquetado como TTAF. El tiempo total de motor es etiquetado como TTEL para los motores del lado izquierdo y TTER para los motores del lado derecho. Un motor, el avión de un motor tiene un tiempo registrado inferior TTEL, salida de 0.0 horas para TTER. Todos los aviones ROBWil están sujetos a una inspección periódica de cien horas, así ROBWil debe guardar el camino del tiempo total desde la última inspección tan bien como la fecha de la inspección. La tarifa de la renta varía con la renta del avión.

Los clientes que rentan deben ser calificado para volar el avión rentado. Un nuevo cliente debe presentar su certificado médico, su diario de vuelo, su licencia de piloto y deben recibir un chequeo de vuelo de los instructores de ROBWil. La credencial de un cliente debe ser registrada dándola al personal de ROBWil. Los clientes pueden solicitar crédito. Si ellos obtuvieron su crédito, pueden cargar la renta y ROBWil la cargará a su cuenta. Cuando los clientes cargan la renta y otros gastos a su cuenta, el personal de ROBWil debe verificar qué balance máximo tiene para no excederse. ROBWil carga un porcentaje sobre el monto de la cuenta. Los cargos en la cuenta son considerados como pagos vencidos y deben ser pagados dentro de los sesenta días posteriores a la fecha en que se hizo el cargo.

Los cargos en la instrucción de vuelo varía por el tipo de instructor y son registradas en esta forma:

TIPO	TARIFA/HORA
Privada	\$ 18.50
Comercial	\$ 22.50
Instrumental	\$ 25.00
Multi-motor	\$ 27.85

Los cargos son hechos de acuerdo al precio estimado por la instrucción dada. Por ejemplo, si la instrucción instrumental es dada en un avión de multi-motor, el precio por hora será de veintisiete pesos con ochenta y cinco centavos.

ROBWil emplea cinco instructores de vuelo. Los instructores registran sus tipos de licencia y la categoría que tienen, la fecha de la última examinación médica en vuelo, el tipo de certificado médico sostenido, el tiempo total de instrucción, fecha y año (YTD) tiempo de instrucción, ambos total y tiempo de instrucción YTD por categoría (privada, comercial, instrumental y multi-motor). Esos registros deben

ser levantados a la fecha. El gerente de ROBUIL debe recibir cada información de manera rápida y eficiente.

ROBUIL debe de crear facturas, preguntas rápidas, y reportes. Las típicas preguntas deben incluir lo siguiente:

- ¿Cuántas horas tuvo de vuelo el instructor X en este mes?
- ¿Cuántas horas tuvo de vuelo el avión Y este mes?
- ¿Cuál fue la fecha de la última inspección de cien horas del avión X?
- ¿Cuántas horas son registradas en la inspección de cien horas?
- El cliente Z sacó prestado para volar un avión(es) ¿cuánto fue de renta? Incluir el típico reporte.
- El vuelo del avión reporta datos seleccionados. Como incluir un reporte resumido por categoría (instrucción o renta directa) y por avión.
- Reportar las actividades del instructor seleccionadas por fechas. Cada reporte incluir un resumen por categoría (privado, comercial, instrumental, y multi-motor).

Capítulo 3. NORMALIZACION

Dado el objetivo principal de este trabajo, que es el de dar a conocer diferentes técnicas para el diseño de base de datos, en este capítulo mostramos cómo puede pronosticar la fuente de la redundancia de datos, cómo detectar y describir la forma normal, y cómo asegurar estructuras de tablas que no encuentran requerimientos específicos de una forma normal. A pesar de que esas formas normales superiores son, generalmente, más convenientes que formas normales inferiores, también señalamos por qué el diseño de base de datos algunas veces tiene que denormalizarse a estructura de tablas que encuentren requerimientos especiales. Advirtiendo que la denormalización debe realizarse con gran precaución y con algunas reservas.

Cada vez que usa una tecnología apropiada modelando E-R diseña su base de datos, pero puede ser posible encontrar algunas redundancias de datos indeseables en ésta. La normalización es el proceso que detiene la redundancia y produce técnicas a través de las cuales esas redundancias son controladas. Finalmente, mencionamos como modelando E-R y trabajando con técnicas de normalización puede producir buenos diseños.

El material de este capítulo está basado en el trabajo de Peter Rob y Treyton Williams, (1995).

3.1. Dependencia

En el capítulo dos aprendió que, idealmente, una tabla tiene una llave primaria, así como también de que algunas tablas tienen una llave primaria compuesta, ésta formada de dos o más atributos, además que una llave primaria únicamente define valores de atributos de una entidad. Por ejemplo, si EMP_COD es la llave primaria de EMPLEADO, conociendo el valor de EMP_COD de un empleado podremos también conocer del empleado otros atributos almacenados, como su nombre, apellido, teléfono, etc. En otras palabras, los atributos del empleado están determinados por la llave primaria. Presentando esto de manera más precisa, los atributos que no son parte de una llave primaria dependen de ésta.

Idealmente, los atributos que no son parte de la llave primaria son desplegados dependiendo con respecto a su llave primaria. Así, observe la tabla 3.1., y note la existencia de diferentes tipos de dependencias:

- ORDEN tiene una llave primaria compuesta que consiste de la combinación de ORDEN_NUM y la descripción del producto, PROD_DESCR. (Permitiendo asumir que la descripción del producto es único, ¡nunca tener en mente que la descripción hace una llave primaria pobre!). La ORDEN_NUM por ella misma, no puede ser una llave primaria, porque una orden puede consistir de muchos productos.

Por ejemplo, la orden número diez mil dos incluye cuatro productos diferentes. Similarmente, PROD_DESCR no puede ser una llave primaria por sí misma, porque un producto puede ser incluido en varias órdenes diferentes. Por ejemplo, el producto "destornillador" se presenta tres veces en la tabla. Pero uniendo ORDEN_NUM y PROD_DESCR puede tener una identificación única de entre todos los demás atributos. Todos los atributos que no son parte de la llave primaria (atributos **no-llave** o **no-principal**) son dependientes de ésta.

Nota. Cuando mostramos el ejemplo de la tabla 3.1, la gente que está empezando a utilizar el concepto de llave primaria, frecuentemente encontramos que ellos identifican la combinación de ORDEN_NUM y CLIENT_COD como llave primaria. Sin embargo, si observa los datos mostrados en la tabla 3.1, fácilmente puede ver que la combinación de la orden número diez mil dos y el código del cliente cinco mil veintiuno no es la única identificación de un producto. (La combinación pudo referirse a un esquilador, un destornillador, una pala, o una sierra.)

Tabla 3.1. Componentes y contenido de la tabla ORDEN

ORDEN_NUM	CLIENT_COD	ORDEN_FECHA	CLIENT_NOMB	PROD_DESCR	PROD_PRECIO	ORDEN_CANT
10001	5217	11/22/94	Williams	martillo	\$ 8.99	2
10001	5217	11/22/94	Williams	destornillador	\$ 4.45	1
10002	5021	11/22/94	Johnson	esquilador	\$ 18.22	1
10002	5021	11/22/94	Johnson	destornillador	\$ 4.45	3
10002	5021	11/22/94	Johnson	pala	\$ 11.07	1
10002	5021	11/22/94	Johnson	sierra	\$ 14.99	1
10003	4118	11/22/94	Lorenzo	martillo	\$ 8.99	1
10004	6002	11/22/94	Kopiusko	sierra	\$ 14.99	1
10004	6002	11/22/94	Koprusko	destornillador	\$ 4.45	2
0005	5021	11/23/94	Johnson	taladro	\$ 34.95	1

Fuente: Rob, 1995

- El PROD_PRECIO es dependiente del PROD_DESCR, el cual es solamente parte de la llave primaria. Cada dependencia es conocida como **dependencia parcial**.
- El código del cliente (CLIENT_COD) determina el nombre del cliente (CLIENT_NOMB). Por ejemplo, el cliente número cinco mil veintiuno se llama Johnson. CLIENT_COD no es parte de la llave primaria, vemos una dependencia conocida como **dependencia transitiva**. En otras palabras, una dependencia transitiva existe cuando un atributo es dependiente de otro atributo que ~~es~~ no es parte de la llave primaria, ni es parte de una llave primaria compuesta.

3.2. Formas normales

Al examinar las fechas de la tabla 3.1. note su innecesaria duplicación. Por ejemplo, la orden número diez mil dos muestra que se repitió (y por lo tanto redundancia) el nombre del cliente, la fecha de la orden, también como (necesaria) la descripción del producto y el precio para cada uno de los cuatro artículos ordenados. La redundancia innecesaria usualmente significa que la forma normal de la tabla puede ser inapropiada. La tabla 3.1. refleja una condición conocida como **primera forma normal**, escrita también como **1NF**.

Primera Forma Normal (1NF)

La etiqueta 1NF es usada para describir una estructura en la cual:

- Todos los componentes de la llave primaria están definidos. Ambos componentes de la llave primaria ORDEN_NUM y PROD_DESCR, son conocidos y mostrados en cada uno de los registros.
- Todos los atributos no-llave son dependientes de la llave primaria. Por ejemplo, si conoce la combinación ORDEN_NUM igual a diez mil dos y PROD_DESCR igual a sierra, también conoce que el PROD_PRECIO es de catorce pesos noventa y nueve centavos, y ORDEN_CANT es de uno para el CLIENT_COD número cinco mil veintiuno, CLIENT_NOMB es Johnson, que ordenó el producto en la fecha (ORDEN_FECHA) veintidós de noviembre de mil novecientos noventa y cuatro.
- El contenido de la tabla no está repetido por grupos. Esto es, cada intersección renglón/columna contiene un solo valor, más bien que un grupo de valores. En otras palabras, desde varias intersecciones renglón/columna mostradas en la tabla 3.2. contiene *grupos* de valores, esta tabla no está en 1NF pero la tabla 3.1. tiene las condiciones de la 1NF.

Tabla 3.2. Repitiendo grupos en una tabla

ORDEN_NUM	CLIENT_COD	ORDEN_FECHA	CLIENT_NOMB	PROD_DESCR	PROD_PRECIO	ORDEN_CANT
10001	5217	11/22/94	Williams	martillo	\$ 8.99	2
				destornillador	\$ 4.45	1
10002	5021	11/22/94	Johnson	esquilador	\$ 18.22	1
				destornillador	\$ 4.45	3
				pala	\$ 11.07	1
				sierra	\$ 14.99	1
10003	4118	11/22/94	Lorenzo	martillo	\$ 8.99	1
10004	6002	11/22/94	Kopiusko	sierra	\$ 14.99	1
				destornillador	\$ 4.45	2
10005	5021	11/23/94	Johnson	taladro	\$ 34.95	1

Fuente: Rob, 1995

La tabla que está en 1NF está sujeta a una *innecesaria* redundancia de datos, así debemos encontrar un camino que reduzca tal redundancia. Fundamentalmente, la disminución de la redundancia es una forma de controlar la cantidad, asegurando la minimización de anomalías en los datos, descritos en el capítulo uno. Afortunadamente, esto es más fácil de crear con un procedimiento de control de calidad cuando acordemos formas normales superiores, indicando la disminución de la redundancia de datos que en formas normales inferiores. En otras palabras, la **segunda forma normal (2NF)** produce menor redundancia de datos que la 1NF y la **tercera forma normal (3NF)** produce menor redundancia que la 2NF. Mostraremos en la sección 3.3. cómo puede ser cambiada esta forma normal.

A pesar de que la forma normal existe más allá de la 3NF, hay en gran parte el interés sólo de aquellos quienes siguen fronteras teóricas de estructuras de bases de datos y de aquellos quienes producen muy especiales y raras aplicaciones.¹ Nos enfocaremos en prácticas finales y desarrollo de aplicaciones, no nos dirigiremos más allá de la forma 3NF.

Segunda Forma Normal (2NF)

La etiqueta 2NF es usada para describir una tabla en la cual:

- Todos los requerimientos de 1NF son encontrados.
- No hay dependencias parciales.

Una dependencia parcial está basada sólo en una *parte* de la llave primaria, ésta puede existir cuando la tabla tiene una llave primaria compuesta. Por lo tanto una tabla cuya llave primaria es un atributo único y en el cual se encuentran los requerimientos de la 1NF está automáticamente en 2NF. (Pero una tabla en 2NF puede contener dependencias transitivas.)

Tercera Forma Normal (3NF)

Finalmente, la etiqueta 3NF es usada para describir una tabla en la cual:

- Todos los requerimientos de 2NF son encontrados.
- No hay dependencias transitivas.

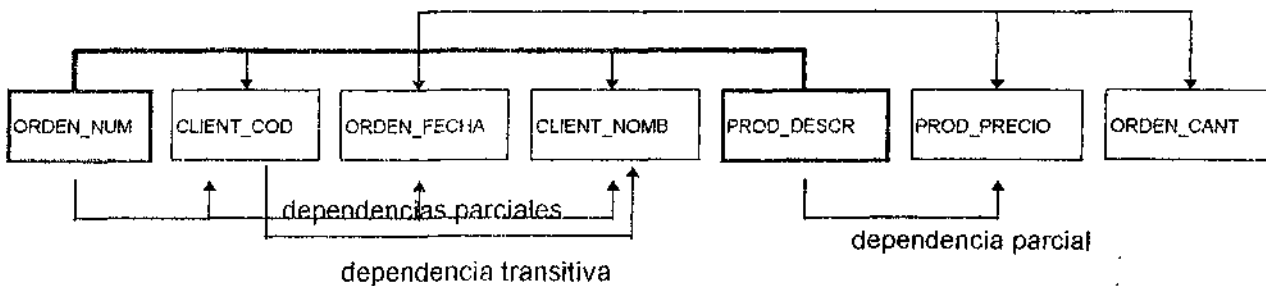
3.3. Diagramas de dependencia y conversiones forma normal

Las conversiones de forma normal, de 1NF a 2NF a 3NF son fácilmente realizadas cuando examina la estructura de la tabla con la ayuda de un diagramas de dependencia. Un **diagrama de dependencia** es usado para identificar visualmente diferentes tipos de dependencias (llave primaria, parcial o transitiva). Examine la figura 3.1., note que un diagrama de dependencia muestra los atributos de la tabla, los componentes de la llave primaria, y todas las dependencias. Específicamente:

¹ Si está interesado en formas normales superiores, puede encontrar información examinando los textos: Peter Rob y Carlos M. Coronel, Database Systems: Desing, Implementation, y Management (Belmont, CA: Wadsworth Publishing Company, 1993); Catherine M. Ricardo, Database Systems: Principles, Design, and Implementation (New York: Macmillan Publishing Company, 1990); y Davir M. Kroenke, Database Processing: Fundamentals, Design, Implementation (New York: Macmillan Publishing Company, 1994).

- Los componentes de la llave primaria son mostrados dentro de las cajas con bordes más prominentes (más oscuro) que otros atributos. Los componentes de la llave primaria son conectados por una línea que los relaciona.
- Las dependencias basadas en la llave primaria son indicadas por una flecha dibujada arriba del diagrama.
- Las dependencias transitivas y parciales son indicadas por una flecha dibujada abajo del diagrama.

Figura 3.1. Diagrama de dependencia basado en la tabla 3.1.



Fuente: Rob, 1995

Usando las indicaciones visuales de la figura 3.1., es fácil reconocer los diferentes tipos de dependencias y hacer las conversiones deseadas. Los componentes de la llave primaria son presentados visualmente, es fácil distinguirlos entre las dependencias parciales y transitivas.

3.3.1. Conversiones de 1NF a 2NF

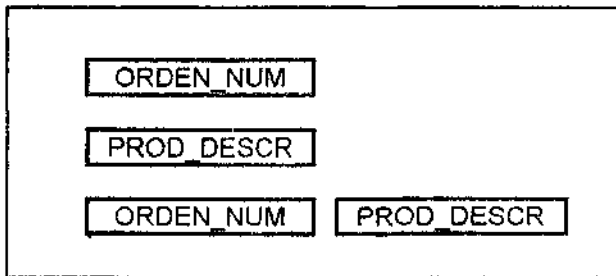
Usando la figura 3.1. como guía, note que la llave primaria compuesta consiste de ORDEN_NUM y PROD_DESCR. La conversión de 1NF a 2NF requiere que:

- Escriba los componentes de la llave primaria ORDEN_NUM y PROD_DESCR en líneas separadas. Cada una llegará a ser una llave primaria en una nueva tabla.
- Escriba la llave primaria compuesta en la última línea, que será la llave primaria compuesta de una tercera tabla.

Estos puntos deben ser creados al inicio de los tres diagramas de dependencia mostrados en la figura 3.2.

La figura 3.2. muestra tres conjuntos de llaves primarias. Ahora regresemos a observar la figura 3.1. en la cual los atributos serán unidos a cada una de esas llaves primarias. Use la información presentada en la figura 3.1. para completar los tres diagramas de dependencia que se basan en la figura 3.2.

Figura 3.2. Primer paso de la normalización

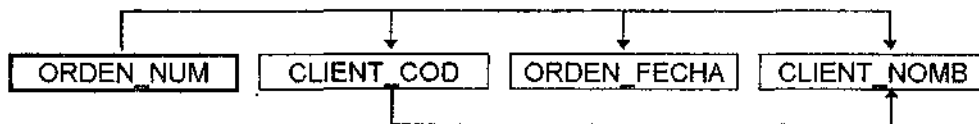


Fuente: Rob, 1995

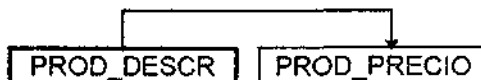
- ORDEN_FECHA, CLIENT_COD, y CLIENT_NOMB son dependientes de ORDEN_NUM, uniéndolos a ORDEN_NUM que se mostró en la figura 3.2. produce la estructura de tabla ORDEN mostrada en la figura 3.3.
- PROD_PRECIO es dependiente de PROD_DESCR, uniéndolo a PROD_DESCR mostrado en la figura 3.2. produce la estructura de la tabla PRODUCTO mostrados en la figura 3.3.
- ORDEN_CANT es dependiente de la combinación de ORDEN_NUM y PROD_DESCR, uniendo ese a los dos componentes de la llave primaria mostrada en la figura 3.2. produce la estructura de la tabla ORDLINEA mostrada en la figura 3.3. (Note que la ORDEN_CANT fue sólo un atributo que no ha sido considerado todavía para los primeros dos pasos).

Figura 3.3. Una base de datos en el pequeño 2NF

Nombre de la tabla: ORDEN



Nombre de la tabla: PRODUCTO



Nombre de la tabla: ORDLINEA



Fuente: Rob, 1995

Examine la figura 3.3, observe dentro de la tabla la 2NF: en la tabla ORDEN, CLIENT_NOMB es dependiente del CLIENT_COD, el cual es un atributo no-llave. En otras palabras, la tabla ORDEN contiene una dependencia transitiva y es por lo tanto 2NF.

3.3.2. Conversión de 2NF a 3NF

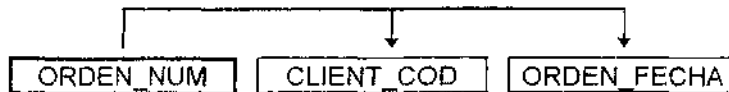
Convertir una tabla 2NF a 3NF, simplemente rompe la dependencia transitiva creando una nueva tabla, guardar la llave primaria de la nueva tabla en la tabla original para que sirva como llave externa. Usando la tabla ORDEN 2NF mostrada en la figura 3.3 como un ejemplo, haga lo siguiente:

- Rompa la dependencia transitiva (CLIENT_COD determina CLIENT_NOMB) creando una nueva estructura de tabla CLIENTE, mostrada en la figura 3.4. que contiene CLIENT_COD y CLIENT_NOMB.
- Guarde CLIENT_COD en la tabla ORDEN para que sirva de llave externa a CLIENTE.

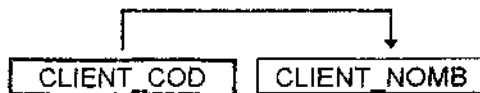
Los resultados de esas acciones son mostrados en la figura 3.4.

Figura 3.4. Una base de datos cuya tabla está en 3NF

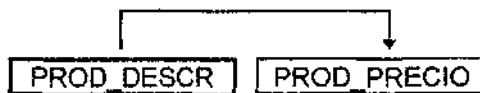
Nombre de la tabla: ORDEN



Nombre de la tabla: CLIENTE



Nombre de la tabla: PRODUCTO



Nombre de la tabla: ORDLINEA



Fuente: Rob, 1995

La base de datos mostrada en la figura 3.4. ahora contiene cuatro tablas, dada una de las cuales está en 3NF. Imagine cuanta redundancia de datos hemos eliminado en este punto, si como es usual en este caso, la tabla CLIENTE contiene cada detalle como su dirección, número telefónico, fecha del crédito, etcétera. No mencionamos la tabla PRODUCTO, ¡la cual probablemente contiene mucho más atributos que unos mostrados aquí!

3.3.3. Forma normal Boyce-Codd

El procedimiento de la 1NF a 2NF a 3NF fue ilustrado aquí para producir la 3NF conocida como la **Forma Normal Boyce Codd**, o **BCNF**. Fundamentalmente, el BCNF es muy estricta 3NF. Eso es, todas las relaciones en BCNF están en 3NF, pero no todas las relaciones 3NF están basadas en una llave primaria compuesta de BCNF. (Sin embargo, todas las relaciones 3NF están basadas en un atributo singular de llave primaria que está también en BCNF.)

Encontrar los requerimientos para una relación en 3NF:

- No debe contener grupos repetidos.
- No debe contener dependencias transitivas.
- No debe contener dependencias parciales.
- Debe conformar los requerimientos de todos los atributos no llave siendo completa y funcionalmente dependiente a la llave primaria.

La BCNF añade otra condición: *todo elemento determinante en la relación debe ser candidato a llave*. Esto es, una dependencia es apropiada sólo si esa dependencia está basada en una llave primaria o en algún atributo(s) que puede servir para el mismo propósito como llave primaria.

Nota. Una **llave candidata** es alguna llave que puede servir como llave primaria. En otras palabras, cada llave es un candidato a ser llave primaria. Así, si la relación tiene más de una llave candidata, la llave primaria es simplemente la llave candidata que fue seleccionada como llave primaria.

La razón de nuestro interés en el BCNF es simple: una tabla 3NF que tiene una llave primaria compuesta puede todavía ser sometida a anomalías en los datos, si la condición BCNF no es encontrada. Vemos ahora como esto es posible, considerando este escenario:

- La compañía XYZ proporciona software de uso final con capacitación en bases de datos, redes y hojas de cálculo.
- XYZ emplea varios instructores para cada una de esas tres áreas.
- Cada instructor enseña sólo un área, esto es, un instructor de bases de datos enseña solamente eso.
- Los clientes de la corporación pueden elegir para adquirir un contrato de una o más áreas.

Dando este escenario, suponga usted que la compañía XYZ guarda la base de datos que se muestra en la tabla 3.3.

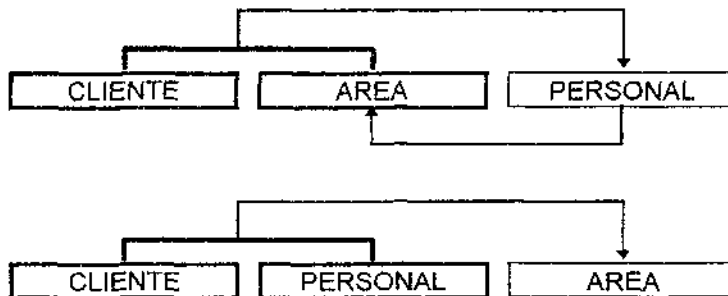
Tabla 3.3. Una tabla en 3NF que no está en BCNF

CLIENTE	AREA	PERSONAL
10001	Base de datos	Rodríguez
10001	Redes	McDowell
10002	Base de datos	Rodríguez
10003	Hoja de cálculo	Johnson
10004	Base de datos	Brodinski

Fuente: Rob, 1995

Examine el contenido de la tabla 3.3. note que conociendo el número del cliente, no es suficiente para identificar otros atributos. Por ejemplo, el CLIENTE igual a mil uno puede identificar cada base de datos o redes. Similarmente, AREA no puede identificar al PERSONAL, porque el AREA es igual a base de datos puede identificar Rodríguez o Brondinski. Sólo la combinación de CLIENTE y AREA identificará siempre al PERSONAL. Pero la tabla 3.3. también tiene otra llave primaria candidata, porque la combinación de CLIENTE y PERSONAL puede identificar siempre el AREA. Por ejemplo, el CLIENTE número mil uno y el PERSONAL llamado Rodríguez identificará al AREA igual a base de datos. Por lo tanto dos combinaciones de atributos son disponibles para servir como llaves primarias. Estas llaves primarias candidatas se *traslapan*, porque ambos contienen el atributo CLIENTE. (Note que hemos usado nombres, más bien que números, para identificar el PERSONAL y el AREA haciendo visible la relación. Nunca tener en mente que los nombres no pueden ser buenas llaves primarias.)

Figura 3.5. Un posible diagrama de dependencia para la tabla 3.3.



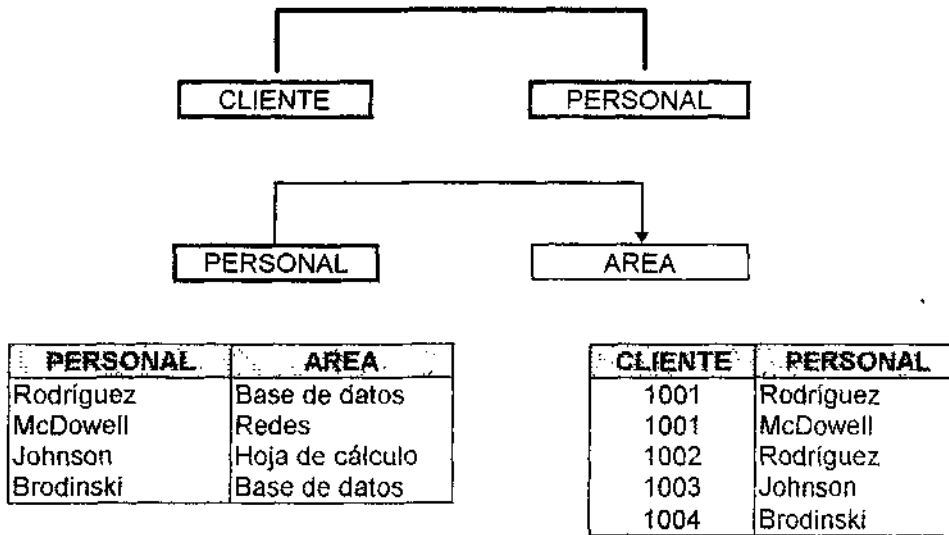
Fuente: Rob, 1995

Examine la figura 3.5., note que el primer diagrama de dependencia satisface la condición de 3NF, porque la dependencia "PERSONAL determina AREA" ninguna es parcial o transitiva. Pero el diagrama no está en BCNF, porque el atributo PERSONAL no es llave candidata. Así ¿por qué es éste un problema? Observe la tabla 3.3. nuevamente note que, la supresión del cliente mil cuatro también puede borrar que Brodinski enseñe base de datos. Si el cliente mil uno es suprimido, perderemos el hecho de que McDowell enseña redes. Sintetizando, la estructura de la tabla 3.3. produce **supresión de anomalías**. Y supone que la compañía XYZ contrata a Sánchez para enseñar base de datos: no

podemos almacenar este hecho hasta que tengamos para un cliente al que Sánchez le sea asignado como maestro de base de datos, así producimos una **inserción anómala**.

Para eliminar las anomalías tenemos que citarlo, debemos de descomponer el primer diagrama de dependencia mostrado en la figura 3.5. para crear un diagrama de dependencia mostrado en la figura 3.6. Note que el contenido de la tabla muestra abajo los diagramas de dependencia, las áreas no son largas para las anomalías y descubriendo que la redundancia de datos se reduce mucho, que cuando son comparadas con lo encontrado en la tabla 3.3.

Figura: 3.6 Conociendo los requerimientos del BCNF.



Fuente: Rob, 1995

3.4. Denormalización: una palabra de advertencia.

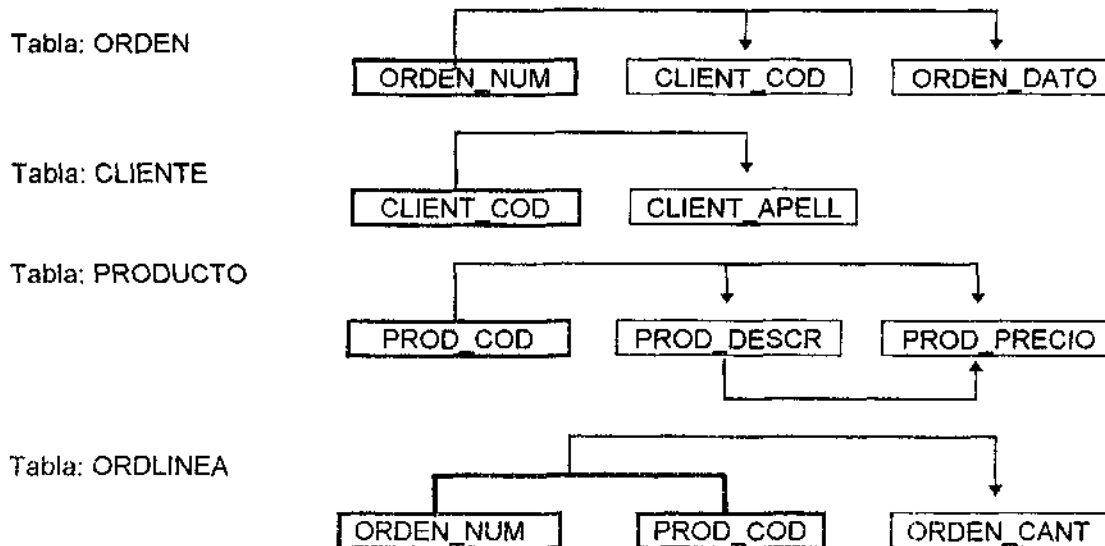
Aunque la 3NF generalmente se refiere para un control razonable de la redundancia de datos, algunas veces la denormalización es apropiada. La **denormalización** reduce la forma normal, que es, moviendo desde 3NF a 2NF representa la denormalización. La intervención de la base de datos muestra que la denormalización es a veces efectuada por la razón de que la base de datos contiene más estructuras de tablas, apuntando hacia una base de datos más compleja que puede llegar a tener el sistema. (Como el nombre lo sugiere, un apuntador "apunta" para localizar el dato. El ejemplo sencillo de un apuntador es una llave externa de una tabla en la que se usa el apuntador para unir los datos de una tabla relacionada). Si una base de datos está sujeta a miles de transacciones diarias; reduce el movimiento del apuntador haciéndola en términos diferentes de transacción rápida. Desafortunadamente, el beneficio rápido es comprado a un costo de la redundancia de datos pudiendo producir anomalías de datos. Dado el mejoramiento de la rapidez del Software y Hardware, la denormalización efectúa por si sola razones que están perdiendo su lado favorable.

La denormalización es veces hecha para simplificar la estructura de la base de datos, cuando el costo de las anomalías de datos en la estructura 2NF aparenta ser mínima. Por ejemplo, el director de una biblioteca probablemente encontrará útil guardar los datos de la editorial y los del autor para cada publicación. Sin embargo, algunos autores escriben pocos libros o artículos, y como hay muchas

editoriales diferentes, esto puede ser apropiado para guardar cada dato en la tabla LIBRO que fue creado por separado de las tablas de AUTOR y EDITORIAL para que fueran ligadas a la tabla LIBRO. La decisión para guardar los datos del autor y editorial sin la tabla LIBRO crea las condiciones 2NF sujetas a la redundancia de datos y anomalías de datos que se habló en el capítulo dos. Dado al medio ambiente de las operaciones de una biblioteca, estas redundancias aparentemente no son destructivas.

Otra razón de la denormalización es crear más llaves primarias eficientes. Por ejemplo, la tabla PRODUCTO en la figura 3.4 su llave primaria es llamada PROD_DESCR. Porque es difícil de garantizar que las descripciones del producto son siempre turnadas al mismo tiempo, aparentemente la llave primaria fallará muy rápido. Además para evitar errores humanos en la entrada de datos se usa mejor el código de barras u otro procedimiento de código automático y dispositivos. En parte, los códigos del producto aparentemente son usados como llaves primarias que toda la descripción del producto. Por lo tanto, podríamos usar menos modificaciones como se muestra en la figura 3.7., la llave primaria del PRODUCTO es ahora PROD_COD la cual sirve como una llave externa en ORDLINEA.

Figura 3.7 Conclusión de la Base de Datos



Fuente: Rob, 1995

Sin embargo, la unión de PROD_COD a la estructura de la tabla PRODUCTO mostrada en la figura 3.4 crea una dependencia transitiva, debido a que PROD_DESCR no es una parte de la llave primaria, como lo determina PROD_PRECIO. En otras palabras, para producir mejor una llave primaria, tuvimos que aceptar una estructura de la tabla de 2NF. (Asumimos que la descripción del producto es suficientemente precisa por ser única y por servir, por lo tanto, está como una llave primaria. Naturalmente si la descripción del producto es genérica, la tabla PRODUCTO en la figura 3.7 sería en 3NF. Después de todo, ¡la descripción genérica de "martillo" puede ser asociado con una larga variedad de precios!).

Aunque el procedimiento de la denormalización usado en el ejemplo anterior es conveniente, especialmente en la entrada automática de datos, aproxímese a la denormalización con alguna precaución. Siempre tenga en mente que el precio de la denormalización es la redundancia de datos, la

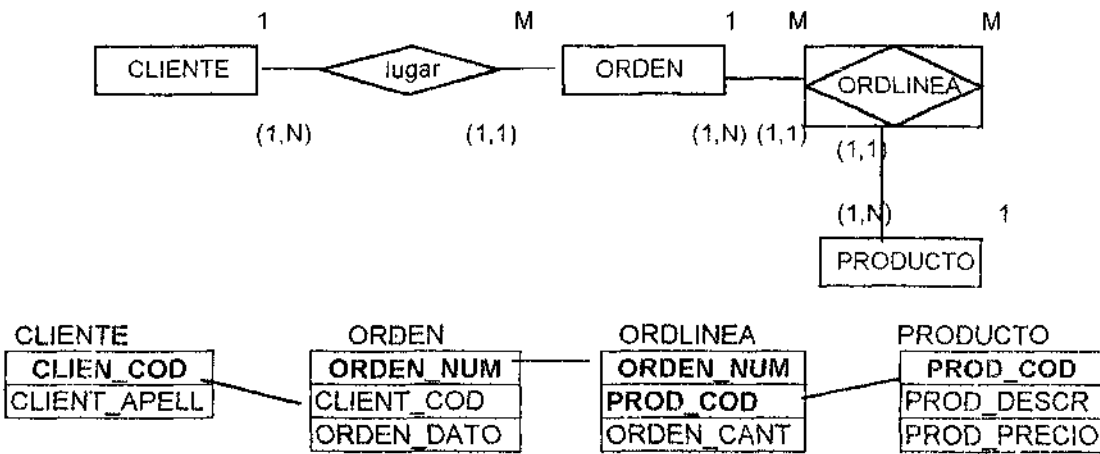
cual puede crear anomalías de datos como se ilustró en el primer capítulo. Pocas cosas destruyen un ciclo de una base de datos como un depósito seguro de datos, como un rápido receptor que *no controla* la redundancia de datos.

3.5 Normalización y modelo E-R

No queremos dejar la impresión de que la normalización está limitada algunas veces para que haga antes o después el modelo E-R. En efecto, en un mundo real de diseño de la base de datos, la normalización y el modelo E-R toma un lugar frecuente. En cada etapa del proceso de modelo E-R, la normalización puede indicar la creación de adiciónar entidades y relaciones. En parte, el modelo E-R genera entidades en la cual las tablas normalizadas son construidas. No puede crear una buena estructura a menos que ambas actividades sean llevadas para tener un buen proceso de diseño. El modelo E-R reorganiza las relaciones entre entidades más explícitamente y el proceso de la normalización interviene en las estructuras con redundancia de datos.

Como se examinan los diagramas de dependencia en la figura 3.7, debe recordar que estas estructuras de tablas representan entidades en las cuales las relaciones están definidas en un diagrama E-R. Por ejemplo, la figura 3.7 no sugiere que la relación entre ORDEN y PRODUCTO sea M:N. (Una orden puede contener muchos productos y un producto puede aparecer con muchas ordenes). Solamente el diagrama E-R mostrada en la figura 3.8 puede desplegar esa información.

Figura: 3.8 Diagrama E-R para semejar la estructura en la figura 3.7



Fuente: Rob, 1995

Los datos originalmente mostrados en la tabla 3.1 deben ser ahora distribuidos a través de la estructura de la tabla mostrada en la figura 3.8. La tabla nueva es mostrada en la figura 3.9. Note que los datos originales presentados en la tabla 3.1 están todavía dispuestos y que la redundancia ha sido reducida porque los datos están ahora guardados en las tablas que son conocidas por los requerimientos de 3NF. La disminución de la redundancia de datos pudo ser más espectacular si tenemos que usar los datos reales de la tabla CLIENTE que incluye: apellidos de los clientes, nombre, inicial, dirección, datos de crédito y otros más que son de interés para los vendedores. La tabla PRODUCTO puede ser expandida para conocer el típico requerimiento para incluir la descripción completa del producto, cantidades comunes disponibles, información del vendedor, y demás. (Podrá entonces adiciónar a la tabla VENDEDOR la relación de PRODUCTO provistos del VENDEDOR.)

Figura: 3.9 Encontrando datos con la base de datos de la figura 3.8.

Tabla: CLIENTE	
CLIENT_COD	CLIENT_APELL
5217	Williams
5021	Johnson
4118	Lorenzo
6002	Kopiusko

Tabla: ORDEN		
ORDEN_NUM	CLIENT_COD	ORDEN_FECHA
10001	5217	11/22/94
10002	5021	11/22/94
10003	4118	11/22/94
10004	6002	11/22/94
10004	5021	11/23/94

Tabla: PRODUCTO		
PROD_COD	PROD_DESCR	PROD_PRECIO
231-AD/45	Martillo	\$ 8.99
341-QW/23	Destornillador	\$ 4.45
554-AQ/67	Recortador	\$ 18.22
325-HT/51	Barra	\$ 11.07
112-TR/33	Sierra	\$ 14.99
349-SE/87	Taladro	\$ 34.95

Tabla: ORDLINEA		
ORDEN_NUM	PROD_COD	ORDEN_CANT
10001	231-AD/45	2
10001	341-QW/23	1
10002	554-AQ/67	1
10002	341-QW/23	3
10002	325-HT/51	1
10002	112-TR/33	1
10003	231-AD/45	1
10004	112-TR/33	1
10004	341-QW/23	2
10005	349-SE/87	1

Fuente: Rob, 1995

Examine cuidadosamente la tabla presentada en la figura 3.9 para descubrir los beneficios de un buen diseño:

- Las tablas CLIENTE, ORDEN y PRODUCTO cada una contiene sólo una entrada para cada conjunto de entidades, en otras palabras, excepto para la presencia de la llave externa CLIENT_COD en la tabla ORDEN, aquí no hay una redundancia total. La ventaja de cada distribución de datos es obvia: un dato puesto al día (adición, borrado o cambiado) en cualquier tabla requiere de solamente una acción en esa tabla. Por lo tanto *no hay camino para diferenciar el valor existente para alguna entidad.*
- La tabla más activa, ORDLINEA, contiene un número mínimo de atributos y por lo tanto requiere lo menos posible de una cantidad de entrada de datos. Mejor aún, desde el ORDEN_NUM será probablemente generado por la computadora y el PROD_COD aparentemente es buscado, la fuente principal de la entrada del dato erróneo es ORDEN_CANT. Casi siempre, el dato ORDEN_CANT

solamente consiste de uno o dos dígitos, los errores aparentan ser raros y son fáciles de localizar porque están en un lugar (y fijos) cuando el cliente recibe la factura.

- El diseño es flexible porque las funciones de la base de datos pueden ser fáciles de levantar. Por ejemplo, si la tabla de PRODUCTO es expandido para incluir cantidades disponibles, esto puede ser levantado automáticamente por la entrada de ORDEN_CANT hecha en la tabla ORDLINEA. Por lo tanto, sería fácil de crear un sistema de control basado en cada interacción. Puede también adicionar una cantidad mínima disponible y datos del vendedor para la tabla PRODUCTO, entonces usted crea un sistema de ordenado automático que será provocado cuando un cliente hace una compra dejando pendiente la cantidad mínima disponible de este valor crítico.

Recuerde: para un buen diseño use E-R, y la técnica de normalización es para *minimizar la redundancia de datos y maximizar la flexibilidad del diseño*.

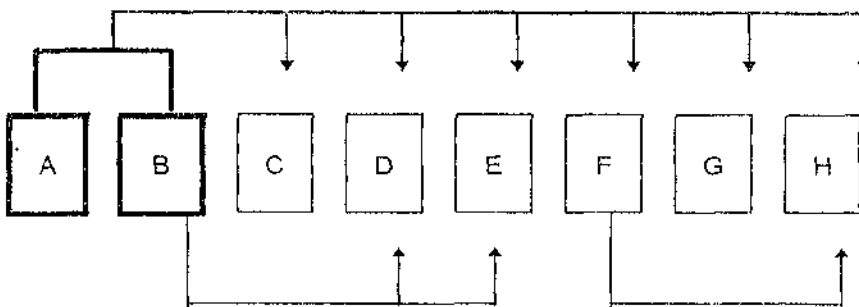
Problemas

3.1. Dibuje el diagrama de dependencias para la tabla LIBRO localizada en la base de datos de LIBRERIA.

TABLA: LIBRO	DESCRIPCION	EJEMPLO
LIBRO_COD	Identifica el código del libro	20-5 21-1
LIBRO_TITULO	Título del libro	Recrear una base de datos y todo eso
LIBRO_A_APELL	Apellido del autor	López
LIBRO_A_NOM	Nombre del autor	Anne
LIBRO_A_INIC	Inicial del autor	G
LIBRO_EDITO	Editorial del libro	McGraw-Hill
LIBRO_CATEG	Categoría (ciencia, negocios, ficción)	Negocios
LIBRO_COMPFECH	¿Cuándo fue comprado el libro?	7/21/92
LIBRO_DISP	¿Está el libro disponible? (S o N)	S
LIBRO_USADO	Número de veces que ha sido checado el libro	37

3.2 Usando el siguiente diagrama de dependencia:

Figura: 3.p2 Diagrama de dependencia para el problema 3.2



Fuente: Rob, 1995.

- Identificar todas las dependencias parciales.
- Identificar la dependencia transitiva.
- Convertir la estructura a un conjunto de diagramas de dependencia que muestra una base de datos cuyas tablas están en 2NF o en 3NF.
- Convertir el diagrama de dependencia del problema 3.2c que muestra una base de datos en el cual todas las tablas están en 3NF.

3.3 Dibuje el diagrama de dependencia para la estructura mostrada en la tabla 3.p3. La tabla contiene datos que pueden ser guardados en partes locales.

Tabla: 3.p3

PROD_COD	DESCRIPCIÓN	PRECIO	PASILLO	DEPOSITO	COLGADOR
QW-2314/YT	Aro del pistón, 0-360	\$ 23.67	3	6	12
REW-TYR00	Válvula de escape, 3/4"	\$ 56.93	1	8	12
569980-WEQ	Tenazas de alambre, 12/ torcidos	\$ 15.95	9	2	8
.....
.....
etc.	etc	etc.	etc.	etc.	etc.

Fuente: Rob, 1995

3.4 Dibuje el diagrama de dependencia para la estructura de la tabla 3.p4, usando la técnica de normalización para producir una base de datos en la cual todas las tablas están en 3NF. Forme la llave primaria apropiada donde sea necesaria.

Tabla: 3.p4

CLIENTE	CIEN TEL	CLASE	REGION	CONSULTANTE	CON TEL
K.L. Dorsey	545-324-9983	Banco	Sudeste	D.R. Whiteside	549-231-4452
G.F. Rathers	901-789-4439	Manufactura	Sudeste	D.R. Whiteside	549-231-4452
H.U. Raphael	678-324-8320	Banco	Medio oeste	M.H. Donner	342-990-0056
.....
.....
.....

Fuente: Rob, 1995

3.5 Se ha preguntado para evaluar el invento del sistema que fue desarrollado por PERDIDO de empleado de la Corporación. La pista del sistema es diseñado para guardar la localización de PERDIDO o los muebles de la oficina de la corporación y computadoras. La sola tabla de la "base de datos" contiene los siguientes datos. (Ver tabla 3.p5)

Tabla: 3.p5

EQUIPAMIENTO	VALOR	CONSTRUCTOR	CONST. LOC.	CONST. MGR.	PISO	CUARTO
Lámpara para piso	\$ 123.99	Ramsey	123 Western Blvd	E.G. Browning	2	219-A
Computadora 486DX2	\$ 2367.93	Bayonne	2318 Lod Circle	R.H. Bellamy	1	23
Escritorio para oficina	\$ 989.06	Ramsey	123 Western Blvd.	E.G. Browning	3	303
Sillón para oficina	\$ 234.82	Annex	125 Western Blvd	T.F. Rowland	1	25
Lámpara para piso	\$ 123.99	Bayonne	2318 Lod Circle	R.H. Bellamy	1	87
.....
.....
.....
etc.						

Fuente: Rob, 1995.

Dando esta información:

- Dibuje el diagrama de dependencia.
- Usando el procedimiento apropiado de normalización, convertir la estructura mostrada aquí a una colección de estructuras de tabla que están en 3NF.
- Dibuje el diagrama E-R para unir la estructura que creó en (b).
- Dibuje el diagrama de dependencia para unir la estructura que creó en (b).

3.6 Una base fija de operación (FBO) es un negocio de aviones que puede vender combustible, avión rentado, proporciona el piloto, y demás. El dirigente del FBO le gustaría guardar los datos de las instrucciones que da el (piloto privado, comercial, instrumento) para el rastro de las horas de vuelo y renta generada por cada FBO del propio avión y de cada piloto, para guardar el dato de renta generada por el cliente, y demás. Uno de los empleados del FBO ha creado una "base de datos" que contiene la siguiente tabla la cual genera un reporte por semana. Se ha preguntado ¿Cómo interviene una estructura de la base de datos y cómo improvisarla?

AC_NUM.- es un avión federal con número de registro único.

MODEL.- es un modelo de manufactura con diseño único.

MANUF.- es la manufactura del avión.

TIPO.- es el tipo del avión: SEL/P = máquina de tierra, pistón de la máquina

MEL/P.- multimáquina de tierra, pistón de máquina.

AC_CHG.- cargo del avión por hora (\$0.00 si un cliente usa su propio avión).

FECHA.- Fecha del vuelo.

TIPO.- tipo de instrucción:

PVT=enfocado al piloto privado.

COMM.- entrenamiento piloto comercial.

ME.- entrenamiento en avión multi-máquina.

INSTR.- entrenamiento en vuelo instrumental.

Un nulo indica no entrenamiento (Si renta un piloto es competente para volar sólo).

HRS.- horas voladas.

INSTR.- instructor de vuelo (un nulo indica que el vuelo fue sólo por la renta del piloto).

1-CHR.- cargo de instrucción.

TTAF.- tiempo total (horas) sobre el avión.

TTEL.- tiempo total (horas) sobre la máquina izquierda. (Los aviones con una sola máquina usan

TTEL para registrar el tiempo de máquina).

TTER.- tiempo total (horas) sobre la máquina derecha.

(el tiempo del avión y de la máquina no son registradas para el cliente que lleva su propio avión).

CLIENT.- número del cliente.

Tabla: 3.p6

AC NUM	MOD.	MANUF	TIPO	AC. CHG	FECHA	TIPO	HRS	INSTR	I. CHG	TAF	TTEL	TER	CUEN
2099Y	PA28-181	Pipa	SEL/P	\$65.00	11/23/94	Pvt	1.4	T.R. Smith	\$16.00	2009.3	203.8	0.0	1002
1005U	C-172	Cesna	SEL/P	\$58.00	11/23/94	Pvt	0.9	W.T. Poole	\$16.00	5980.1	1432.5	0.0	1005
2099Y	PA28-181	Pipa	SEL/P	\$65.00	11/24/94		2.3			2010.7	205.2	0.0	1003
3489P	BE76	Artesania	MEL/P	\$124.50	11/24/94	ME	1.2	D.D. Rose	\$27.50	4001.2	1023.4	1456.9	1001
1005U	C-172	Cesna	SEL/P	\$58.00	11/24/94	instr	1.0	T.R. Smith	\$23.75	5981.1	1433.5	0.0	1004
2287T	C-152	Cesna	SEL/P	\$0.00	11/24/94	Comm	2.1	W.T. Poole	\$20.00				1007
2099Y	PA28-181	Pipa	SEL/P	\$65.00	11/24/94		1.8			2012.5	207.0	0.0	1002
1005U	C-172	Cesna		\$58.00	11/25/94		1.5			5982.6	1435.0	0.0	1004
etc.						

Fuente: Rob, 1995.

- Usando la tabla 3.p6, dibuje el diagrama de dependencia.
- Usando el diagrama de dependencia dibujado en el problema 3.6a. levante la estructura de la tabla para crear un conjunto de estructuras de la tabla que conforman los requerimientos en 3NF.
- Evalúe la estructura que creó en el problema 3.6b para determinar si es o no apropiado para crear una o más estructuras en 2NF.
- Dibuje el diagrama E-R para completar la base de datos.
- Dibuje el esquema relacional para completar la base de datos.
- Muestre el dato con la tabla nuevamente construida.

3.7 Usando la figura 2.19, del capítulo dos, dibuje el diagrama de dependencia para LIBRO, PATRON y PRESTAMO e identifique sus formas normales.

3.8 Usando la figura 2.19 del capítulo dos, identificar las llaves candidatas.

Capítulo 4. ESTUDIO DE CAMPO, ENCUESTAS Y RESULTADOS

Dado el objetivo principal de este trabajo, que es el de dar a conocer diferentes técnicas para el diseño de base de datos, en este capítulo se describe el desarrollo de la investigación llevada a cabo para conocer el nivel académico de los alumnos que egresan de la Licenciatura de Sistemas Computacionales, de la Universidad Popular Autónoma del Estado de Puebla.

4. 1. Definición del Problema

Los estudiantes de la carrera de la Licenciatura de Sistemas Computacionales de la Universidad Autónoma del Estado de Puebla, necesitan conocer las diferentes técnicas de base de datos que le permiten incrementar la posibilidad de tomar mejores decisiones en el análisis y diseño que todo sistema requiere.

Este objetivo se logrará con una aplicación que tendrá un *Sistema Generador de Tutoriales* desarrollado utilizando las ventajas que presenta el software Microsoft Access, y en particular es un sistema *tutorial* de base de datos completo, eficiente y ameno con el que el alumno puede aprender, comprender y manejar fácilmente los conceptos de base de datos.

Además este *Tutorial de base de datos*, también podrá ser utilizado por alumnos que cursan dicha materia, para aprenderla o reforzar los conocimientos adquiridos en el aulas de clases.

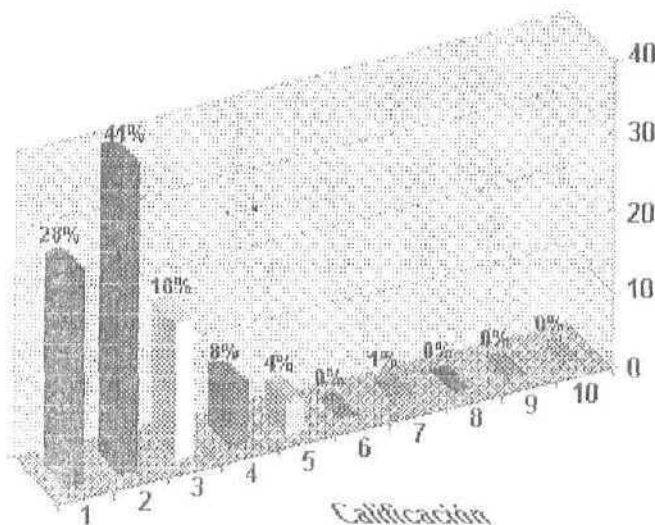
4. 2. Investigación de campo y resultados estadísticos.

De acuerdo con las características citadas en el punto 4.1 realizamos una encuesta en el área de base de datos, abarcando conocimientos teóricos y prácticos, a estudiantes de séptimo y noveno semestre de la carrera de Sistemas Computacionales de la Universidad Popular Autónoma del Estado de Puebla, abarcando una muestra de cien alumnos. Las preguntas fueron las siguientes:

1. ¿Qué porcentaje en conocimientos de base de datos, cree tener?
2. ¿Qué es un sistema de base de datos?
3. ¿Qué es una base de datos?
4. ¿Por qué utilizar una base de datos?
5. ¿Qué es un RDBMS?
6. ¿Cuáles son los objetivos de una organización de base de datos?
7. ¿Cuáles son los objetivos de un DBMS?
8. ¿Qué es un Administrador de Base de Datos?
9. ¿Cuáles son las funciones de un DBA?
10. ¿Cuáles son las ventajas del enfoque de una base de datos?
11. ¿Qué es una entidad?
12. ¿Cómo se representa una entidad?

13. ¿Qué es un conjunto de entidades?
14. ¿Qué es una entidad débil?
15. ¿Cómo se representa?
16. ¿Qué es una entidad compuesta?
17. ¿Qué es un atributo?
18. ¿Cómo se representa?
19. ¿Qué es un esquema?
20. ¿Qué es un subesquema?
21. ¿Qué es una tupla?
22. ¿Qué es cardinalidad?
23. ¿Qué es una relación?
24. ¿Cómo se representan?
25. ¿Cuál es la diferencia entre asociación y relación?
26. ¿Cómo se representa una relación entre entidades?
27. ¿Qué es una base de datos relacional?
28. ¿Cuáles son los tipos de relación?
29. ¿Qué es una llave candidata?
30. ¿Qué es una llave de archivo?
31. ¿Cuáles son los requisitos que debe tener una llave primaria?
32. ¿Qué es una llave externa?
33. ¿Qué es un diccionario de datos?
34. ¿Qué es una tabla?
35. ¿Qué es un modelo E-R?
36. ¿Qué es conectividad?
37. ¿Qué es un tipo?
38. ¿Qué es un subtipo?
39. ¿Qué es un supertipo?
40. ¿Qué es normalización?
41. ¿Qué es una forma no normalizada?
42. ¿Cuál es la forma 1NF?
43. ¿Cuál es la forma 2NF?
44. ¿Cuál es la forma 3NF?
45. ¿Cuál es la forma BCNF?
46. ¿Hasta que Forma Normal se puede llegar?
47. ¿Qué es denormalización?
48. Significado de QBE.
49. ¿Qué es un Query?
50. ¿Cuáles son las características de un QBE?
51. ¿Qué bases de datos conoce?
52. ¿Por qué considera que son importantes?
53. ¿Qué lenguajes de bases de datos domina?
54. ¿En qué las ha aplicado?
55. ¿Qué prácticas en base de datos le gustaría realizar?
56. ¿Cuáles le gustaría conocer?
57. ¿Cómo le gustaría que se las enseñaran?
58. ¿Qué porcentaje en conocimientos de base de datos, cree tener?

De las que se obtuvieron los siguientes resultados estadísticos:

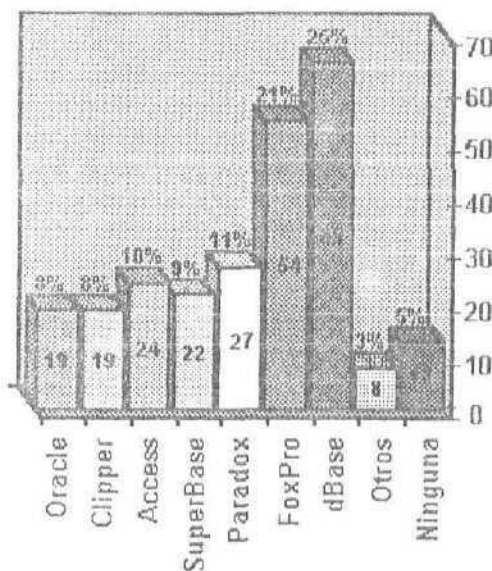


Preguntas números dos a la cincuenta.
Conocimientos teóricos de Base de Datos.

Observe en la gráfica que la muestra obtuvo resultados reprobatorios en un 99%, y el 1% restante aprobó con calificación regular.

Si consideramos que se aplicó a séptimo y noveno semestre nos damos cuenta que recibieron los cursos correspondientes de base de datos, pero tuvieron una enseñanza deficiente o mal enfocada, olvidando conceptos básicos y sus posibles aplicaciones.

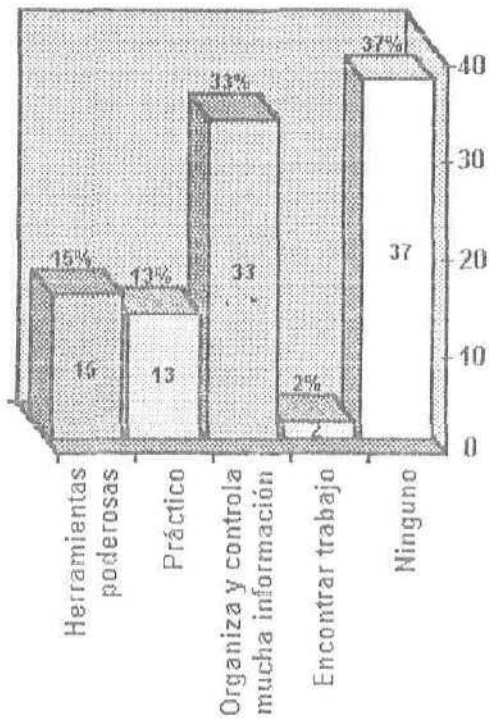
Con lo anterior concluimos que la mayor parte de las personas pertenecientes al área de **Sistemas Computacionales necesitan capacitación** sobre fundamentos y aplicaciones en el área de base de datos.



Pregunta número cincuenta y uno.
Sistemas de Administración de Bases de Datos que conocen.

La gráfica refleja falta de conocimiento sobre la existencia de Sistemas de Administración de BD, e inclusive algunos alumnos los confunden con sistemas de otras aplicaciones tales como Lotus 1-2-3, Quattro Pro, Corel Draw, Image, Works.

También observe, que **dBase** con sólo el 25% de la muestra es el sistema más conocido; **Microsoft Access** solamente lo es en un 10% y existen personas que **no conocen** ninguna Base de Datos. Por lo que concluimos que a pesar de que los alumnos recibieron cursos de base de datos no les dieron un enfoque correcto para su aprendizaje y futura aplicación.

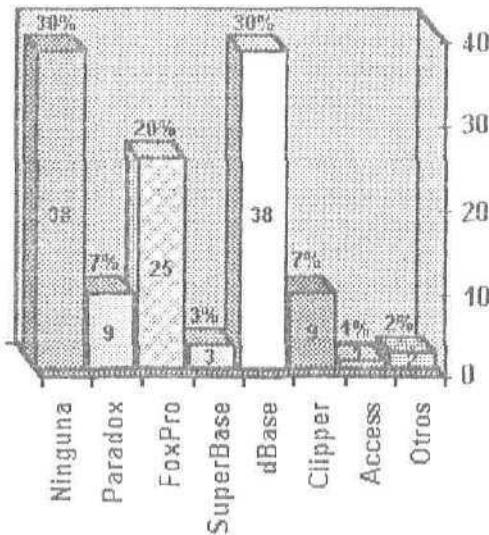


Pregunta número cincuenta y dos

Sistemas de Administración de Bases de Datos que consideran más importantes.

El **concepto** que tienen algunos alumnos acerca de las bases de datos para su aplicación, en realidad **no** es el **correcto** porque la importancia de un DBMS **radica** en *administrar la información y hacer que esté disponible cuando se solicite*, como lo manifiesta el 33% de la muestra.

Observe el porcentaje más alto, note que está concentrado en las personas que no tienen ninguna idea de su uso y posible aplicación y añadido a los porcentajes de aplicaciones incorrectas suman el 67%, lo que indica la falta de conocimiento sobre sistemas administradores de base de datos.

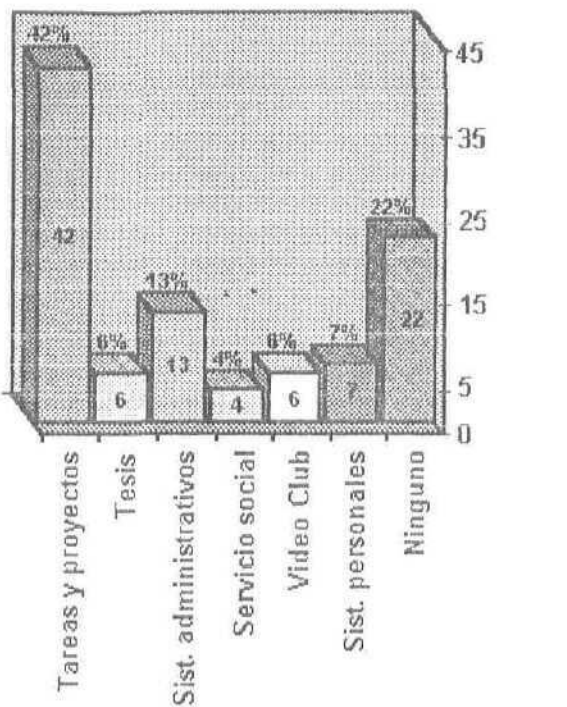


Pregunta número cincuenta y tres

Sistemas de Administración de Base de datos que los estudiantes dominan.

Mire un momento la gráfica, y repare que **dBase** dominado por el 30% de la muestra es igual porcentaje al de alumnos que no dominan ningún sistema de base de datos. Analizando este primer porcentaje, tal vez se deba a que recibieron el curso correspondiente, pero no fueron informados adecuadamente para utilizar las ventajas que proporcionan los demás DBMS, y el que existan alumnos de los últimos semestres de la carrera y no dominen ninguno es preocupante porque no egresan con los conocimientos necesarios.

Se observa también, que los demás sistemas tienen un porcentaje muy **bajo**, como el de **Microsoft Access** herramienta poderosa para realizar todas las tareas de gestión de base de datos.

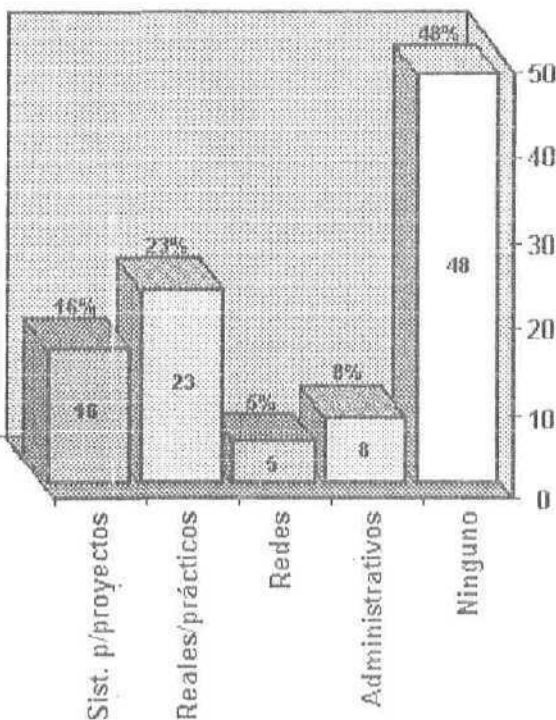


Pregunta número cincuenta y cuatro

Aplicaciones de los Sistemas de Administración de Bases de Datos.

Dentro de las aplicaciones nos damos cuenta que en los *cursos de base de datos*, no se incluyen prácticas para su futuro manejo, ni proporcionan las herramientas necesarias para el análisis y diseño de los sistemas. Lo anterior se refuerza con un alto porcentaje de alumnos no aplican los conocimientos recibidos.

Por otra parte, aunque se han aplicado los conocimientos sobre bases de datos para desarrollar algunos sistemas, en su mayoría sólo son utilizados para almacenar información, desconociendo u omitiendo las herramientas que debieron ser empleadas.

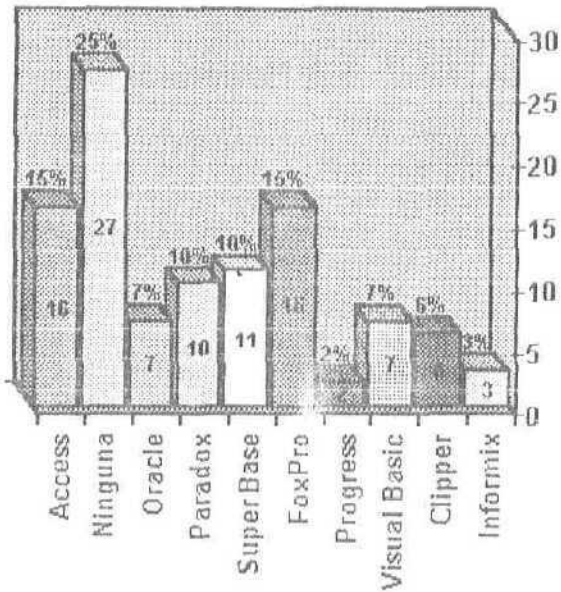


Pregunta número cincuenta y cinco

Prácticas en Bases de Datos que les gustaría realizar.

Un 48% muestra **indiferencia** por aplicaciones y uso de software de **Sistemas Administradores de BD**, correspondiente a la mitad de la muestra; y los demás porcentajes reflejan inclinación para aplicaciones prácticas.

Pregúntese ¿por qué si la muestra encuestada recibió cursos sobre Base de Datos, no aplican estos temas a proyectos reales?, reflexione un momento sobre la posible causa; nosotros suponemos que se debe a un enfoque erróneo. Pero no sólo hay que considerar el problema, sino la posible solución.

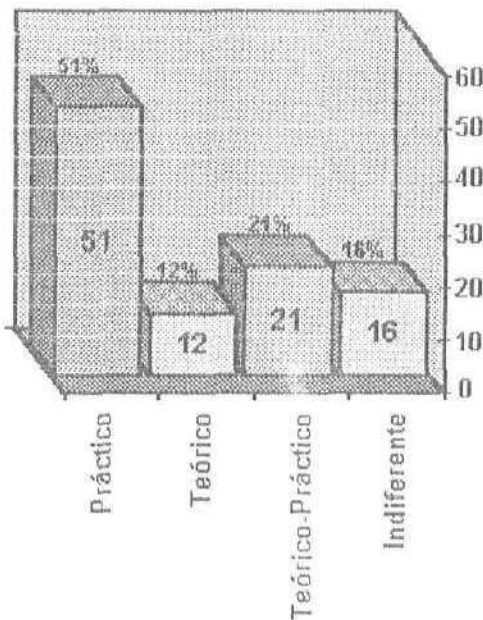


Pregunta número cincuenta y seis

Sistemas Administradores de Bases de Datos que les gustaría conocer.

En esta gráfica apreciamos que el porcentaje más alto corresponde a la **indiferencia** por **conocer nuevos** Sistemas Administradores de BD, motivo que puede ser sustentado por desconocer sus alcances y magnitudes para realizar sistemas con calidad.

Sin embargo la gráfica refleja la inquietud de conocer Sistemas Administradores de BD, por estudiantes con verdadera vocación.



Pregunta número cincuenta y siete

Formas de aprendizaje para los Sistemas Administradores de Base de Datos

Un alto porcentaje de estudiantes anhelan que sus cursos de base datos sean totalmente **prácticos**, esto no puede ser posible, sin embargo así lo manifiestan porque quieren aplicar los conocimientos recibidos y enfrentarse desde las aulas a problemas reales. Lo anterior lo refleja en el 21% de la muestra, a comparación del 12% **teórico**.

De una u otra forma existe el interés común de *aprender*, pero es preocupante que el 16% muestra displicencia por su aprendizaje, porcentaje que en términos normales debiera estar repartido entre los demás.

4. 3. Explicación general.

Los alumnos de los últimos semestres de la Licenciatura de Sistemas Computacionales no tienen sus conocimientos muy claros, presentando deficiencias en el área de base de datos y manifestándose al momento de utilizar un software administrador de base de datos; es decir, el alumno no posee las herramientas necesarias para realizar sus tareas y proyectos y por consiguiente no proporcionan la calidad necesaria que todo sistema requiere para ser productivo.

El problema anterior es porque en muchas ocasiones el alumno no le da suficiente importancia al curso(s), concretándose a realizar solamente las tareas que el profesor exige. Y cuando se enfrenta al problema de elaborar un proyecto para algún comercio o empresa, se da cuenta que le hacen falta conocimientos para el análisis y diseño del sistema que va a proponer; además que por lo general omiten esta fase y directamente van a programar sin tener una solución escrita y mucho menos sin realizar los diagramas correspondientes.

Capítulo 5. DESARROLLO E IMPLEMENTACION DEL SISTEMA GENERADOR DE TUTORIALES

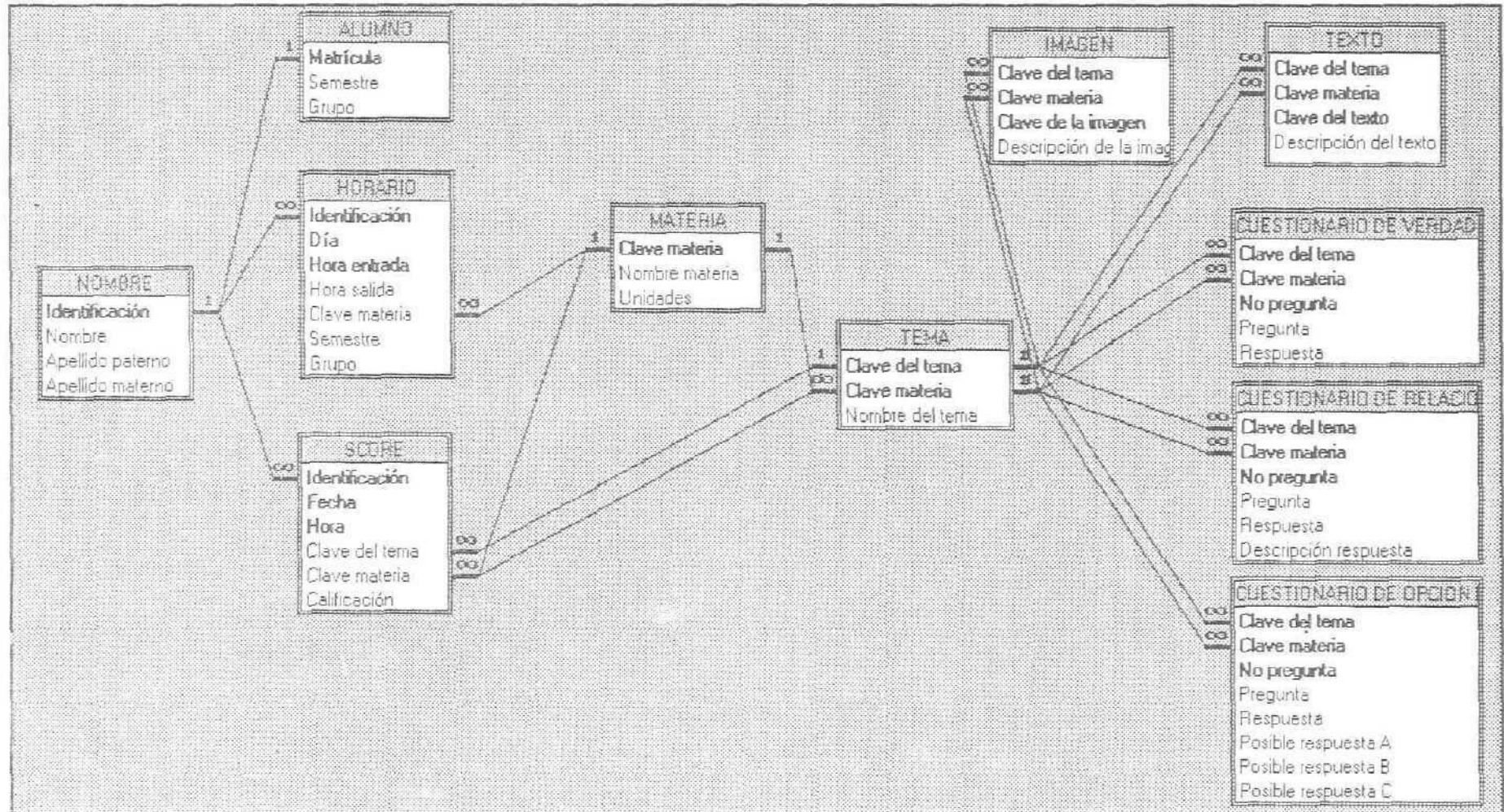
Dado el objetivo principal de este trabajo, que es el de dar a conocer diferentes técnicas para el diseño de base de datos, en este capítulo se sintetiza el análisis y diseño del sistema *Generador de Tutoriales*, el cual tiene como ejemplo un tutorial con conceptos de base de datos descritos en los tres primeros capítulos de la tesis.

5.1. Reglas del Negocio

1. El *Sistema Tutorial* debe manejar las materias dentro de las cuales se clasifican los diferentes temas a aprender, considerando el nombre de la materia y número de unidades correspondiente.
2. El *Sistema Tutorial* debe permitir manejar temas y subtemas, relacionando para cada uno de ellos una imagen y un texto.
3. Las imágenes y textos deberán conservar un estándar para facilitar el mantenimiento dentro del sistema.
4. El *Sistema Tutorial* debe controlar el acceso, así como el registro de los alumnos; en éste último indicando su nombre completo, semestre y grupo.
5. El *Sistema Tutorial* debe registrar, en caso de ser profesor, su nombre completo y horario en el que imparte clase.
6. En los horarios se debe considerar el nombre del profesor que imparte clase, la materia, el semestre y el grupo, así como el día y la hora en que se impartirá. Los días en que se impartirá clases abarcará de lunes a sábado; la hora será dentro del horario de siete de la mañana a diez de la noche, y las clases tendrán duración de cincuenta y cinco minutos.
7. El *Sistema Tutorial* debe ser flexible para poder adicionar, insertar y borrar temas o subtemas dentro de su estructura.
8. El *Sistema Tutorial* debe permitir, opcionalmente, manejar requisitos que condicionen el acceso a ciertos módulos del sistema.
9. El *Sistema Tutorial* debe manejar después de cada capítulo evaluaciones. Estas pueden ser de tres tipos: verdadero o falso, preguntas de relación y opción múltiple, generando el módulo de ejercicio y práctica.
10. El *Sistema Tutorial* debe permitir obtener resultados SCORE, por alumno, por grupo, por materia, etcétera, y además cruzar estas relaciones para obtener resultados que podrán ser impresos a través de un reporte o una gráfica.

DIAGRAMA ENTIDAD-RELACION

5.2. Diagrama Entidad-Relación





5.3. Diccionario de datos

NOMBRE DEL SISTEMA	TUTORIAL	Nº BD	1
NOMBRE(S)	Claudia Morán Moreno Sonia Méndez Cadena	FECHA	17/Marzo/96
DESCRIPCION	<i>Registro del semestre y grupo de los alumnos</i>	NOMBRE BD	ALUMNO

No. DE CAMPO	DESCRIPCION DE CAMPO	NOMBRE VARIABLE	LLAVE	TIPO	LONG	DEC
1	Matrícula de inscripción de un alumno de la Licenciatura de Sistema Computacionales de la UPAEP.	Matrícula	K	N Numérico Entero largo	6	0
2	Número de semestre del alumno.	Semestre		N Numérico Byte	1	0
3	Grupo al que pertenece el alumno.	Grupo		T Texto	1	

NOMBRE DEL SISTEMA	TUTORIAL	Nº BD	2
NOMBRE(S)	Claudia Morán Moreno Sonia Méndez Cadena	FECHA	17/Marzo/96
DESCRIPCION	<i>Preguntas y respuestas tipo opción múltiple sobre un tema o o subtema</i>	NOMBRE BD	CUESTIONARIO DE OPCION MULTIPLE

No. DE CAMPO	DESCRIPCION DE CAMPO	NOMBRE VARIABLE	LLAVE	TIPO	LONG	DEC
1	Clave del tema o subtema	Clave del tema	K	N Numérico Entero	4	0
2	Clave de la materia	Clave materia	K	T Texto	5	
3	Número de pregunta del cuestionario sobre un tema o subtema	No pregunta	K	N Numérico Byte	2	0
4	Pregunta al usuario sobre un tema o subtema	Pregunta		T Texto	80	
5	Respuesta correcta de la pregunta realizada	Respuesta		T Texto	1	
6	Descripción de la probable respuesta A	Posible respuesta A		T Texto	20	
7	Descripción de la probable respuesta B	Posible respuesta B		T Texto	20	
8	Descripción de la probable respuesta C	Posible respuesta C		T Texto	20	



DICCIONARIO DE DATOS

NOMBRE DEL SISTEMA	TUTORIAL	Nº BD	3
NOMBRE(S)	Claudia Morán Moreno	FECHA	17/Marzo/96
	Sonia Méndez Cadena	NOMBRE BD	CUESTIONARIO
DESCRIPCION	<i>Preguntas y respuestas tipo cuestionario de relación sobre un tema o subtema</i>		DE RELACION

No. DE CAMPO	DESCRIPCION DE CAMPO	NOMBRE VARIABLE	LLAVE	TIPO	LONG	DEC
1	Clave del tema o subtema	Clave del tema	K	Numérico Entero	4	0
2	Clave de la materia	Clave materia	K	Texto	5	
3	Número de pregunta del cuestionario sobre un tema o subtema	No pregunta	K	Numérico Byte	2	0
4	Pregunta al usuario sobre un tema o subtema	Pregunta		Texto	80	
5	Respuesta correcta de la pregunta realizada	Respuesta		Numérico Byte	2	0
6	Descripción de la respuesta que se relaciona	Descripción respuesta		Texto	20	

NOMBRE DEL SISTEMA	TUTORIAL	Nº BD	4
NOMBRE(S)	Claudia Morán Moreno	FECHA	17/Marzo/96
	Sonia Méndez Cadena	NOMBRE BD	CUESTIONARIO
DESCRIPCION	<i>Preguntas y respuestas de un cuestionario tipo verdadero/falso sobre un tema o subtema</i>		DE VERDADERO FALSO

No. DE CAMPO	DESCRIPCION DE CAMPO	NOMBRE VARIABLE	LLAVE	TIPO	LONG	DEC
1	Clave del tema o subtema	Clave del tema	K	Numérico Entero	4	0
2	Clave de la materia	Clave materia	K	Texto	5	
3	Número de pregunta del cuestionario sobre un tema o subtema	No pregunta	K	Numérico Byte	2	0
4	Pregunta al usuario sobre un tema o subtema	Pregunta		Texto	80	
5	Respuesta afirmativa/negativa de la pregunta realizada	Respuesta		Booleano Si/no	1	



SISTEMA GENERADOR DE TUTORIALES

DICCIONARIO DE DATOS

NOMBRE DEL SISTEMA	TUTORIAL	Nº BD	5
NOMBRE(S)	Claudia Morán Moreno Sonia Méndez Cadena	FECHA	17/Marzo/96
DESCRIPCION	<i>Registro del horario de materias y profesores</i>	NOMBRE BD	HORARIO

No. DE CAMPO	DESCRIPCION DE CAMPO	NOMBRE VARIABLE	LLAVE	TIPO	LONG	DEC
1	Clave de un profesor de la Licenciatura en Sistemas Computacionales de la UPAEP	Identificación	K	Numérico entero largo	6	0
2	Día de la semana en que se imparte la clase	Día	K	Numérico Byte	1	0
3	Hora en que inicia la clase	Hora entrada	K	Hora hora corta	8	
4	Hora en que termina la clase	Hora salida		Hora hora corta	8	
5	Clave de la materia	Clave materia		Texto	5	
6	Semestre en que imparte clase.	Semestre		Numérico Byte	1	0
7	Grupo en que imparte clase.	Grupo		Texto	1	

NOMBRE DEL SISTEMA	TUTORIAL	Nº BD	6
NOMBRE(S)	Claudia Morán Moreno Sonia Méndez Cadena	FECHA	17/Marzo/96
DESCRIPCION	<i>Registra cada imagen o diagrama mediante una clave única</i>	NOMBRE BD	IMAGEN

No. DE CAMPO	DESCRIPCION DE CAMPO	NOMBRE VARIABLE	LLAVE	TIPO	LONG	DEC
1	Clave del tema o subtema	Clave del tema	K	Numérico Entero	4	0
2	Clave de la materia	Clave materia	K	Texto	5	
3	Clave de la imagen de un tema o subtema	Clave de la imagen	K	Texto	7	
4	Imagen o diagrama de un tema o subtema	Descripción de la imagen		Objeto OLE		



DICCIONARIO DE DATOS

NOMBRE DEL SISTEMA	TUTORIAL	Nº BD	7
NOMBRE(S)	Claudia Morán Moreno	FECHA	17/Marzo/96
	Sonia Méndez Cadena	NOMBRE BD	MATERIA
DESCRIPCION	<i>Registro de las materias que imparte cada profesor</i>		

No. DE CAMPO	DESCRIPCION DE CAMPO	NOMBRE VARIABLE	LLAVE	TIPO	LONG	DEC
1	Clave de la materia	Clave materia	K	Texto	5	
2	Nombre de la materia	Nobre materia		Texto	50	
3	Unidades por semestre de la materia	Unidades		Numérico Byte	2	0

NOMBRE DEL SISTEMA	TUTORIAL	Nº BD	8
NOMBRE(S)	Claudia Morán Moreno	FECHA	17/Marzo/96
	Sonia Méndez Cadena	NOMBRE BD	NOMBRE
DESCRIPCION	<i>Registro de profesores y alumnos</i>		

No. DE CAMPO	DESCRIPCION DE CAMPO	NOMBRE VARIABLE	LLAVE	TIPO	LONG	DEC
1	Matrícula de inscripción de un alumno de la Licenciatura de Sistema Computacionales de la UPAEP.	Identificación	K	Numérico Entero largo	6	0
2	Nombre del alumno.	Nombre		Texto	20	
3	Apellido paterno del alumno.	Apellido Paterno		Texto	12	
4	Apellido materno del alumno.	Apellido Materno		Texto	12	



SISTEMA GENERADOR DE TUTORIALES

DICCIONARIO DE DATOS

NOMBRE DEL SISTEMA *	TUTORIAL	Nº BD	9
NOMBRE(S)	Claudia Morán Moreno Sonia Méndez Cadena	FECHA	17/Marzo/96
DESCRIPCION	<i>Puntuación obtenida por cada alumno o profesor, en el módulo ejercicio y práctica de un tema o subtema</i>	NOMBRE BD	SCORE

No. DE CAMPO	DESCRIPCION DE CAMPO	NOMBRE VARIABLE	LLAVE	TIPO	LONG	DEC
1	Matrícula de inscripción o clave de un profesor de la Licenciatura de Sistema Computacionales de la UPAEP.	Identificación	K	Numérico entero largo	6	0
2	Fecha en que realiza el cuestionario	Fecha	K	Fecha Fecha mediana	8	
3	Hora del sistema operativo en que el usuario contesta un cuestionario del tutorial	Hora	K	Hora	8	
4	Clave del tema o subtema	Clave del tema		Numérico Entero	4	0
5	Clave de la materia	Clave materia		Texto	5	
6	Calificación de evaluación sobre algún tema o subtema	Calificación		Numérico Simple	2	2

NOMBRE DEL SISTEMA	TUTORIAL	Nº BD	10
NOMBRE(S)	Claudia Morán Moreno Sonia Méndez Cadena	FECHA	17/Marzo/96
DESCRIPCION	<i>Registra los nombres de los temas pertenecientes a una materia determinada</i>	NOMBRE BD	TEMA

No. DE CAMPO	DESCRIPCION DE CAMPO	NOMBRE VARIABLE	LLAVE	TIPO	LONG	DEC
1	Clave del tema o subtema	Clave del tema	K	Numérico Entero	4	0
2	Clave de la materia	Clave materia	K	Texto	5	
3	Descripción del título del tema o subtema	Nombre del tema		Texto	60	



NOMBRE DEL SISTEMA	TUTORIAL	Nº BD	11
NOMBRE(S)	Claudia Morán Moreno	FECHA	17/Marzo/96
	Sonia Méndez Cadena	NOMBRE BD	TEXTO
DESCRIPCION	<i>Registra un párrafo de texto de un tema o subtema mediante un identificador único</i>		

No. DE CAMPO	DESCRIPCION DE CAMPO	NOMBRE VARIABLE	LLAVE	TIPO	LONG	DEC
1	Clave del tema o subtema	Clave del tema	K	Numérico Entero	4	0
2	Clave de la materia	Clave materia	K	Texto	5	
3	Clave del texto de un tema o subtema	Clave del texto	K	Texto	7	
4	Información sobre un tema o subtema	Descripción del texto		Texto Memo		

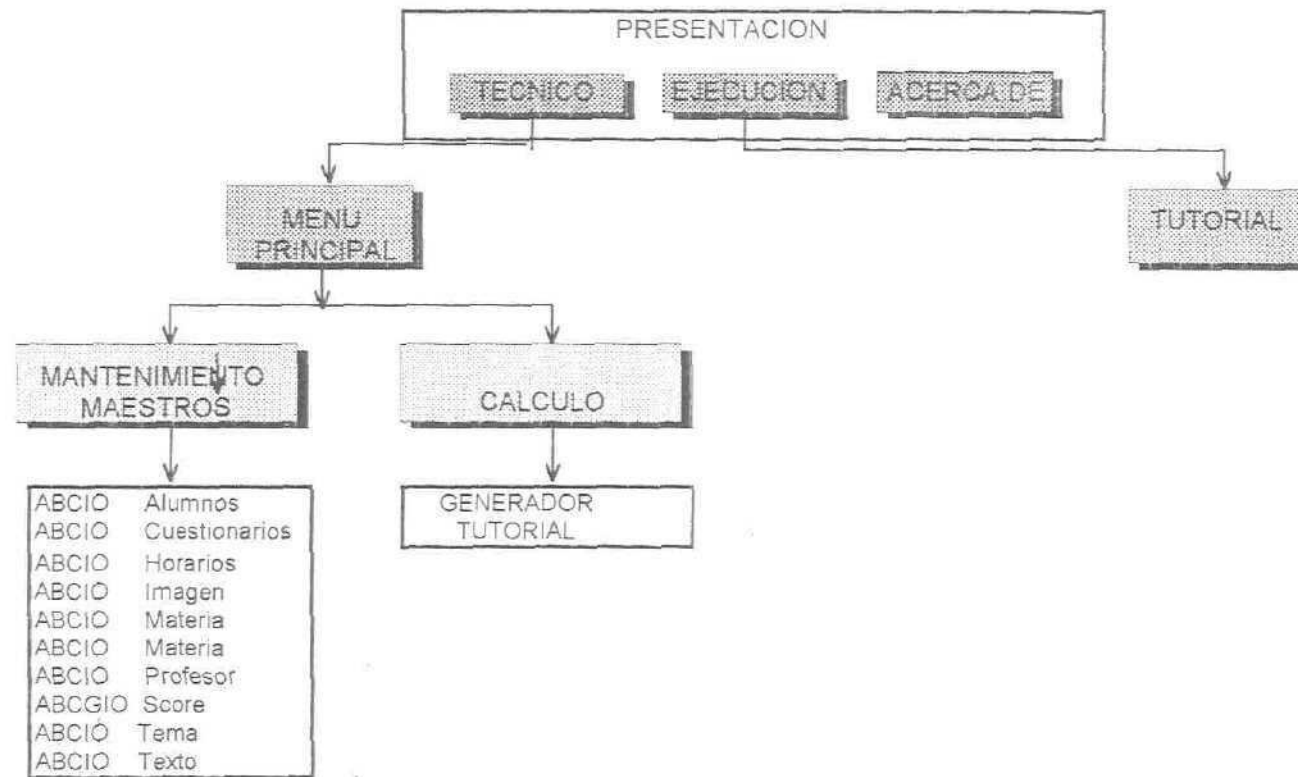


SISTEMA GENERADOR DE TUTORIALES

DIAGRAMA DE ESTRUCTURA

5.4. Diagrama de Estructura

5.4.1. Representación del Diagrama de Estructura



Nomenclatura utilizada:

A: Altas B: Bajas C: Cambios I: Informes O: Consultas G: Gráficas

5.4.2. Descripción del contenido del diagrama de estructura

Informes	
<i>Alumnos por:</i>	materia semestre semestre y grupo
<i>Cuestionario de opción múltiple por:</i>	materia y tema materia y grupo
<i>Cuestionario de relación por:</i>	materia y tema materia y grupo
<i>Cuestionario de verdadero / falso por:</i>	materia y tema materia y grupo
<i>Horario:</i>	horario por profesor horario por semestre y grupo horarios
<i>Imagen:</i>	imagen por materia y tema imágenes por materia
<i>Materia:</i>	materia y temas materias y temas materias
<i>Profesor:</i>	profesor por nombre profesores que dan clase
<i>Score</i>	Alumnos por semestre Alumnos por semestre y grupo Alumnos por semestre, grupo y materia Alumnos por semestre y materia
<i>Texto:</i>	texto por materia y tema textos por materia

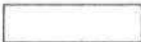


Consultas	
<i>Alumnos:</i>	nombre de alumnos alumnos por semestre alumnos por semestre y grupo
<i>Cuestionario de opción múltiple por:</i>	materia y tema materia y grupo

Consultas	
<i>Cuestionario de relación por:</i>	materia y tema materia y grupo
<i>Cuestionario de verdadero / falso por:</i>	materia y tema materia y grupo
<i>Horario:</i>	horario por profesor horario por semestre y grupo horarios
<i>Imagen:</i>	imagen por materia y tema imágenes por materia
<i>Materia:</i>	materia y temas materias y temas materias
<i>Profesor:</i>	profesor por apellidos profesor por identificación profesores que dan clase profesores
<i>Score</i>	alumnos por apellidos alumnos por matrícula alumnos por semestre alumnos por semestre y grupo alumnos por semestre y materia alumnos por semestre, grupo y materia
<i>Texto:</i>	texto por materia y tema textos por materia


Gráficas	
<i>Catedráticos:</i>	catedráticos por apellidos catedráticos por identificación catedráticos
<i>Score</i>	alumnos por semestre alumnos por semestre y grupo alumnos por semestre y materia alumnos por semestre, grupo y materia

5.5. Diagrama de Transición de Estados

5.5.1. Nomenclatura utilizada

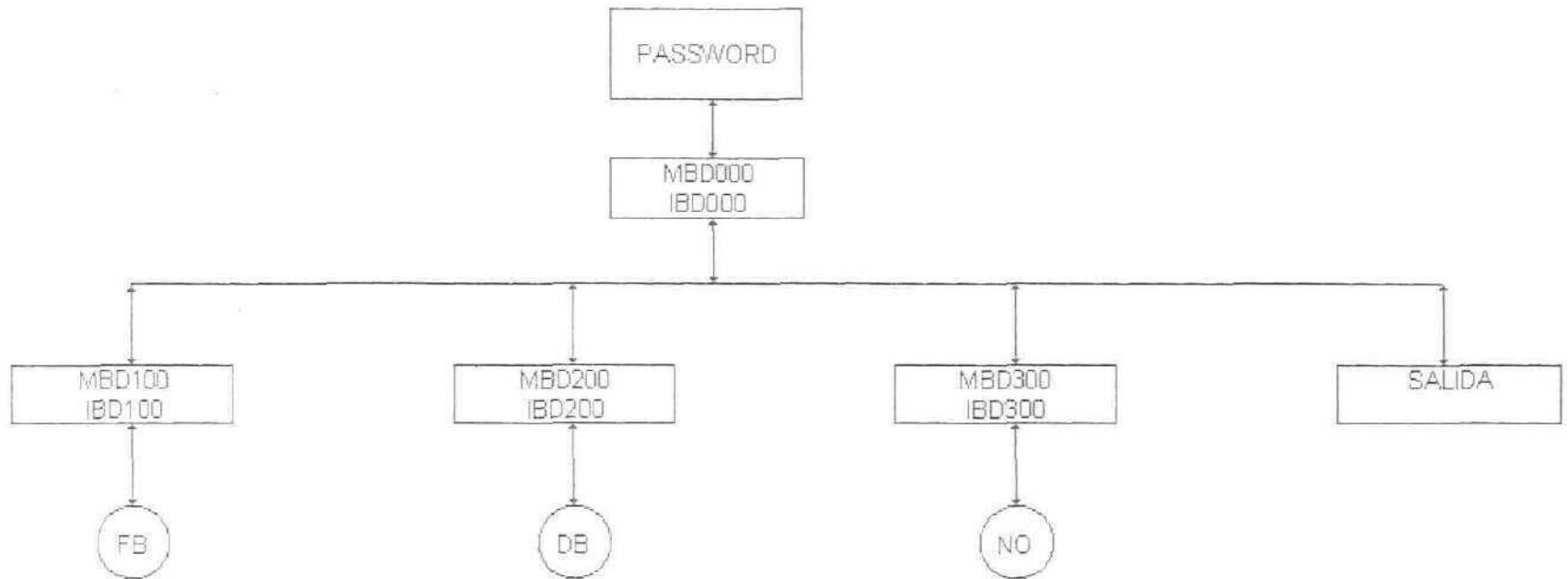
SIMBOLO	SIGNIFICADO
	Proceso
	Flujo de información
	Conector lógico

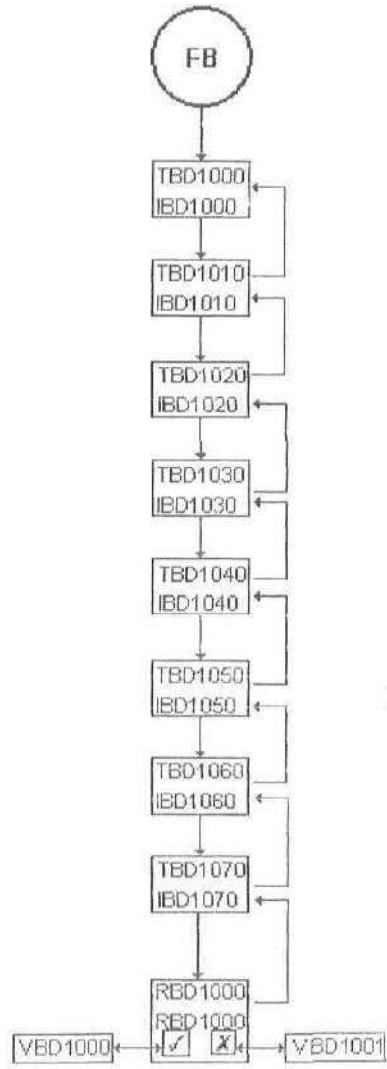
5.5.2. Diccionario de Menús

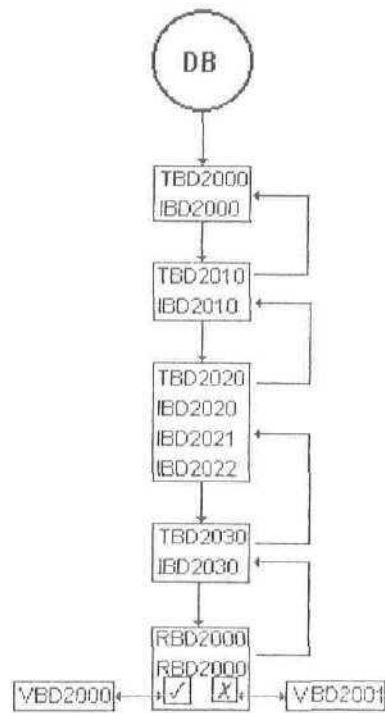
X BD N1 N2 N3 N4	
Donde: X	
M	Menú
T	Texto
I	Imagen
V	Motivador
H	Ayuda
P	Preguntas
R	Respuestas
y:	
BD	Base de Datos
N1	Número de capítulo
N2	Número de submenú de cada capítulo
N3, N4	Conceptos teóricos de cada submenú
	conector lógico
FB	Fundamentos de Base de Datos
BD	Modelando y diseñando Base de Datos
NO	Normalización

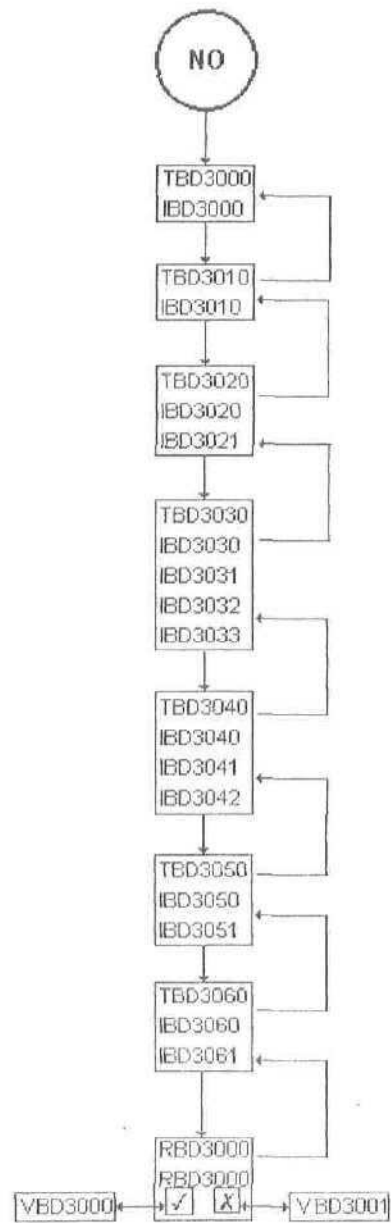


5.5.3. Representación de los Diagramas de Transición de Estados









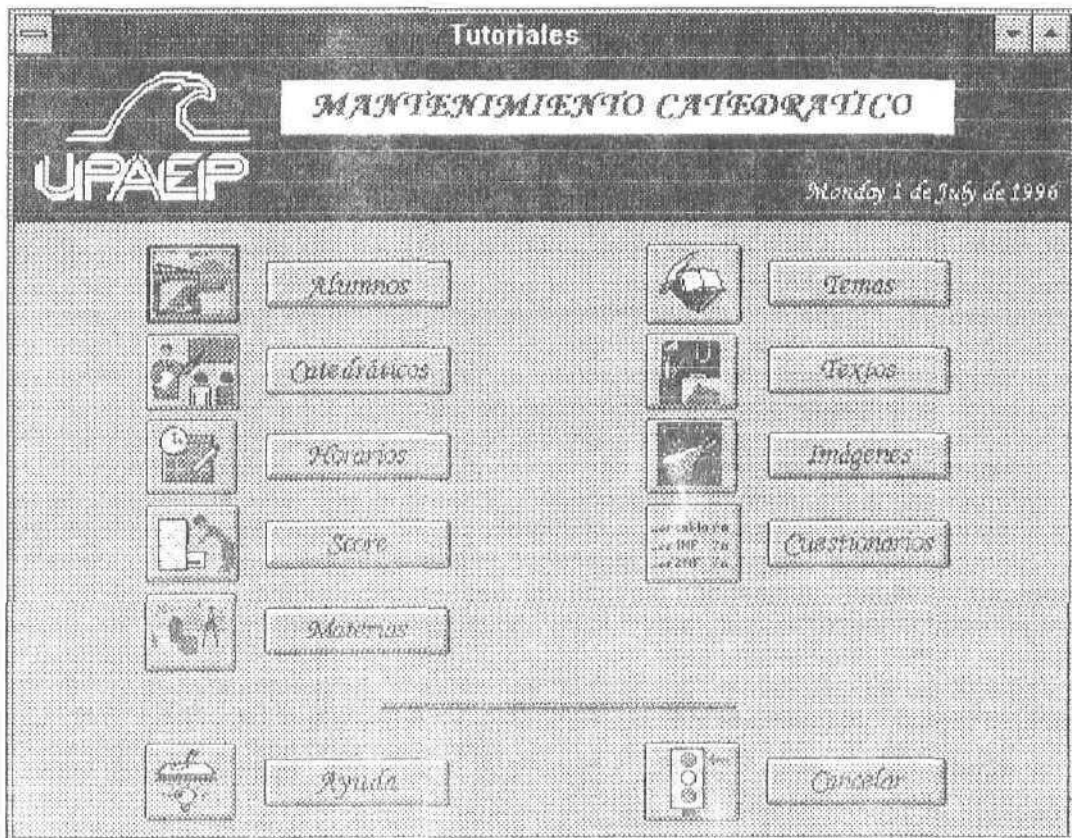
5.6. Manual de Usuario

5.6.1. Sección Mantenimiento Catedráticos

¿Cómo comenzar?

El apartado de mantenimiento catedrático está formado por diez secciones como se muestra en la figura 5.1., que serán utilizadas para la administración de los sistemas tutoriales, cada una realiza diversas tareas que a continuación se describen:

Figura 5.1.



ALUMNOS

- Registra la matrícula, nombre completo del alumno, semestre y grupo.
Acciones que pueden realizarse:
- Undo, guardar, borrar, imprimir (un registro, todos los registros e informes), consultas diversas, editar y aplicar filtros, y buscar registros.



CUESTIONARIOS

A) VERDADERO/FALSO

- Registra la clave de la materia y la del tema, número de pregunta, y la pregunta, indicando al final si la respuesta es verdadera o falsa.

Acciones que pueden realizarse:

- Undo, guardar, borrar, imprimir (un registro, todos los registros e informes), consultas diversas, editar y aplicar filtros, y buscar registros.

B) OPCION MULTIPLE

- Registra la clave de la materia y tema, el número de la pregunta, tres alternativas de respuesta y la respuesta correcta a la pregunta.

Acciones que pueden realizarse:

- Undo, guardar, borrar, imprimir (un registro, todos los registros e informes), consultas diversas, editar y aplicar filtros, y buscar registros.

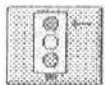
C) RELACION

- Registra la clave de la materia y tema, número de pregunta y la pregunta, respuesta de la pregunta con la que se relaciona y la descripción de ésta.

Acciones que pueden realizarse:

- Undo, guardar, borrar, imprimir (un registro, todos los registros e informes), consultas diversas, editar y aplicar filtros, y buscar registros.

- Cierra la ventana del formulario actual.



CERRAR

5.6.2. Sección Generar un Nuevo Tutorial

¿Cómo comenzar?

A continuación se enumera una serie de pasos para indicar cómo adicionar un nuevo tutorial de la materia que desee, representado en la figura 5.2.

Adicionar un tutorial.

A. Presiones el botón MATERIA

1. Indique la clave de la materia (cinco caracteres máximo)
2. Escriba el nombre de la materia (texto, máximo cincuenta caracteres)
3. Introduzca el número de unidades

B. Presiones el botón TEMA

1. Seleccione la clave de la materia (haciendo clic en el botón del menú desplegable)
2. Indique la clave del tema
3. Escriba el nombre del tema (texto, máximo cincuenta caracteres)

C. Presiones el botón TEXTO

1. Seleccione la clave de la materia (haciendo clic en el botón del menú desplegable)

2. Seleccione de la lista desplegable la clave del tema
3. Indique la clave del texto (máximo 7 caracteres)
4. Escriba la descripción del tema (longitud variable)

D. Presiones el botón IMAGEN

1. Seleccione la clave de la materia (haciendo clic en el botón del menú desplegable)
2. Seleccione de la lista desplegable la clave del tema
3. Indique la clave de la imagen (máximo 7 caracteres)
4. Ubíquese haciendo clic en el campo descripción de la imagen e incruste el objeto mediante la opción insertar objeto de la barra de menú.

E. Presiones el botón CUESTIONARIOS

Aparece un cuadro de diálogo en donde tiene que seleccionar cómo quiere presentar su tipo de cuestionario:

a) verdadero o falso

1. Seleccione la clave de la materia (haciendo clic en el botón del menú desplegable).
2. Seleccione de la lista desplegable la clave del tema.
3. Escriba el número de pregunta.
4. Escriba la pregunta (ochenta caracteres máximo).
5. Indique si la respuesta es verdadera o falsa, según el caso.

b) opción múltiple

1. Seleccione la clave de la materia (haciendo clic en el botón del menú desplegable).
2. Seleccione de la lista desplegable la clave del tema.
3. Escriba el número de pregunta.
4. Escriba la pregunta (ochenta caracteres máximo).
5. Indique tres respuestas alternativas (cada una con veinte caracteres máximo).

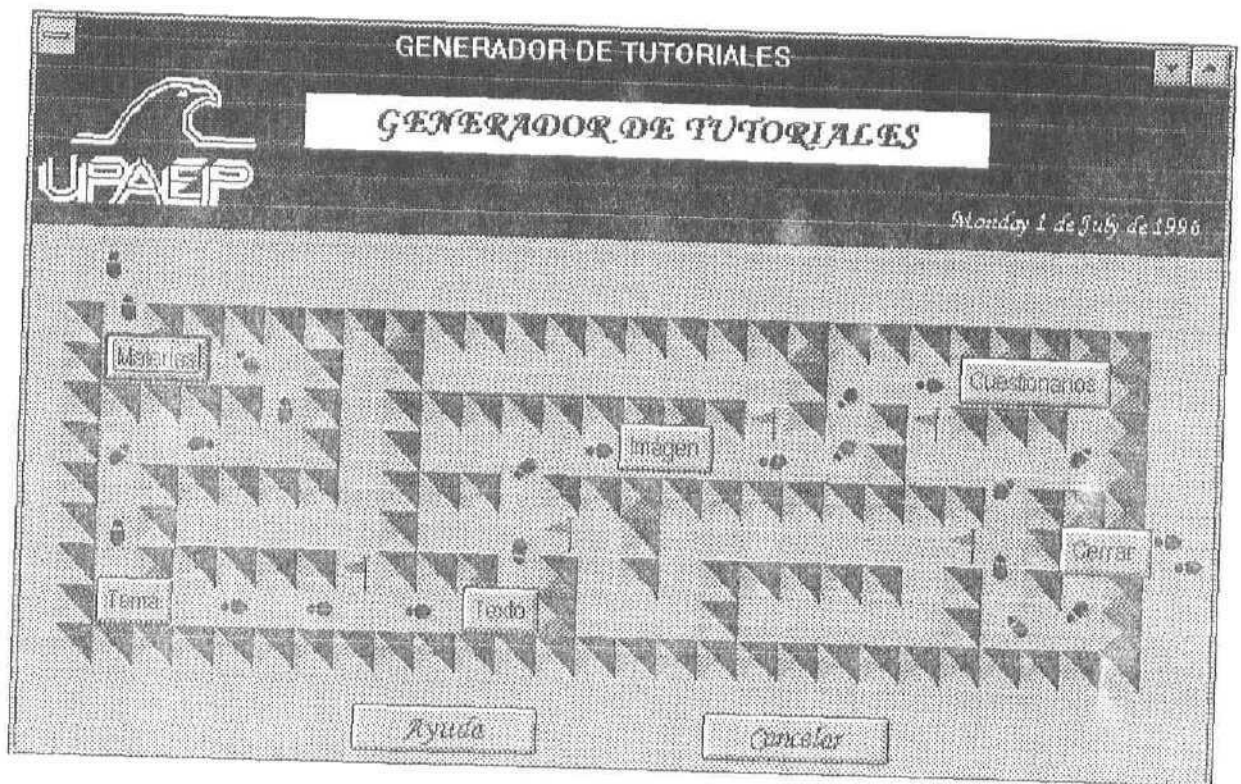
b) relación

1. Seleccione la clave de la materia (haciendo clic en el botón del menú desplegable).
2. Seleccione de la lista desplegable la clave del tema.
3. Escriba el número de pregunta.
4. Escriba la pregunta (ochenta caracteres máximo).
5. En el campo "respuesta" indique el número de la pregunta de la cual se relaciona.
NOTA: Tome en cuenta que debe relacionar las preguntas previamente.
6. Escriba la descripción de la respuesta que corresponde el número de la pregunta que se indicó en el punto anterior.

F. Para finalizar presiones el botón Cerrar

NOTA: Recuerde que la calidad del tutorial depende de los temas e imágenes que en él se presenten.

Figura 5.2.



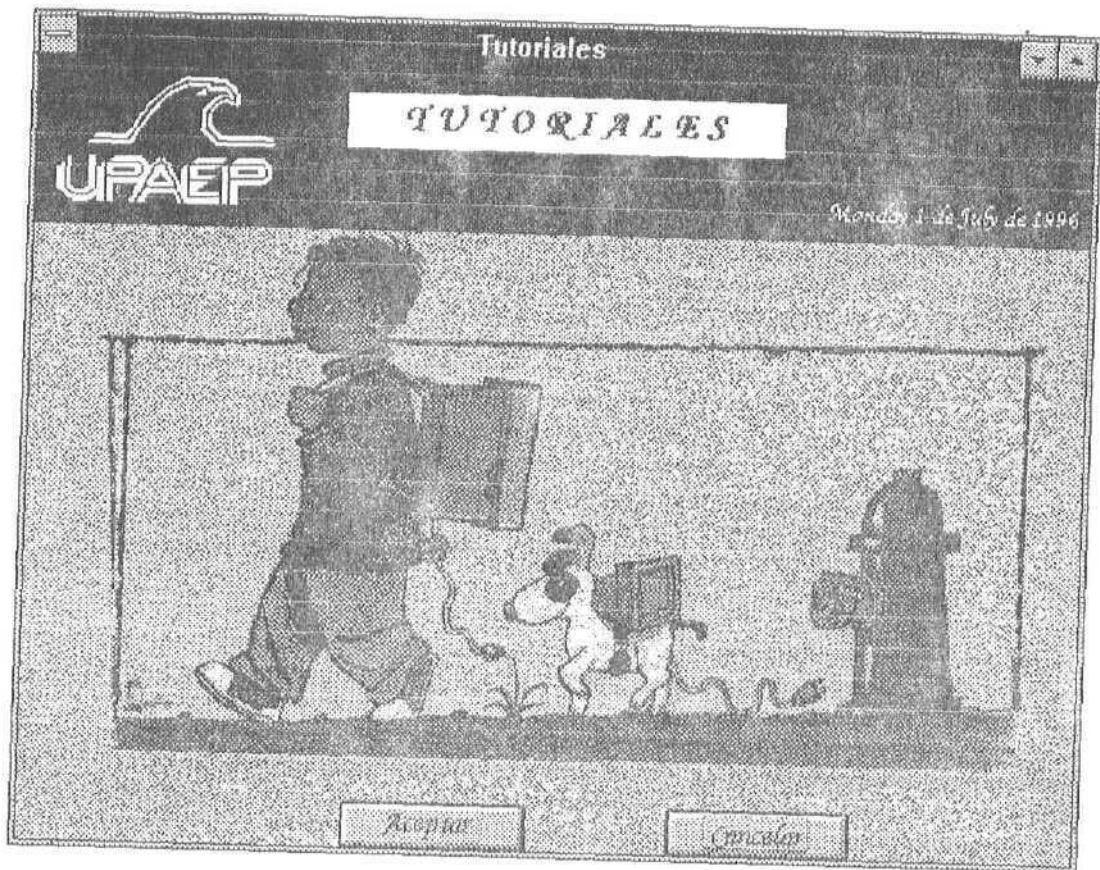
5.6.3. Sección uso de Tutoriales

¿Cómo comenzar?

Cuando usted decide entrar en esta sección le aparecerá la figura 5.3. la cual le confirma su deseo de consultarlo, y en caso de que así sea, podrá aprender un nuevo tutorial realizando los siguientes pasos:

- 1.-Seleccionar la materia deseada.
- 2.-Indicar qué tema desea consultar.
- 3.-A continuación se representan una serie de textos con su imagen correspondiente.
- 4.-Después de haber aprendido la lección se le presenta un cuestionario, que deberá responder.
- 5.-Finalmente se le felicitará en caso de haber comprendido correctamente el tema, y en caso contrario le invitamos a leerlo nuevamente.

Figura 5.3.



CONCLUSIONES

Los sistemas administradores de base de datos relacionales son una poderosa herramienta utilizada para el manejo y administración de la información, la cual para ser obtenida requiere de técnicas apropiadas que proporcionen una visión amplia de su manejo y administración.

La posibilidad de almacenar y recuperar la información de forma rápida y eficaz son dos de las principales razones para utilizar una base de datos interactiva, con este pensamiento se realizó el análisis y diseño de un *Sistema Generador de Tutoriales*; el cual funciona en microcomputadoras bajo el ambiente de Microsoft Windows y con el software de Microsoft Access; donde se utilizan los conocimientos de base de datos explicados en las tres primeras lecciones y teniendo como aplicación final un *Tutorial de bases de datos*.

En cuanto al objetivo sólo podremos saber si se cumplió una vez que el Sistema Generador de Tutoriales sea aplicado a un grupo piloto, en el que el catedrático tome como instrumento principal el Tutorial de base de datos, y con esta guía complementar la comunicación que se establece entre el catedrático y el alumno durante el proceso de enseñanza - aprendizaje. Para su aplicación debe considerar las siguientes situaciones:

- a) Necesita un aula de clases o un centro de cómputo, donde existan tantas máquinas de hardware como alumnos tiene. Es importante que este proceso sea individual para poder captar la completa atención de los educandos y facilitar el proceso de instrucción.
- b) Ofrecer una herramienta indispensable para comprender, analizar y enfrentar los retos de análisis y diseño que todo sistema requiere para su elaboración, no consiste únicamente en proporcionar conocimientos teóricos, sino que debe enseñar al alumno a aplicarlos correctamente. Es por ello que no sólo debe cuestionarlo para verificar si ha comprendido adecuadamente los conceptos teóricos, sino también la parte práctica; le sugerimos entonces que considere los ejercicios que se presentan en al final de cada capítulo de la tesis.
- c) En un aula de clase, el catedrático es como el capitán de un barco y los estudiantes su tripulación, y sólo de acuerdo a las instrucciones u órdenes del primero los discípulos realizarán las actividades que le hayan sido asignadas, pero le sugerimos que exhorte al alumno a tomar iniciativa propia para desarrollar otros ejemplos y de esta manera verificar que puede resolver los problemas que más tarde podrían presentársele. Este proceso es sencillo pero debe tomarlo en consideración para óptimos resultados.

Otros objetivos alcanzados al realizar este trabajo son:

- a) Tener una guía de fácil manejo para conocer las características del sistema de gestión de bases de datos relacionales: Microsoft Access para Windows.

También se tuvieron que solventar diversos gastos económicos que se generaron a lo largo de la realización de este trabajo, como la adquisición de libros, material de papelería, transporte, alimentación, luz, entre otros; factores necesarios y básicos que serían trascendentales para otras personas.

Antes de concluir queremos mencionar que nos enfrentamos a problemas de distinta naturaleza que tuvimos que resolver en cuanto se nos presentaron; pero hubo uno que ha estado fuera de nuestro control, como diversos problemas de salud.

Finalizamos mencionando que el número de horas invertidas en el presente trabajo fue de dos mil.

BIBLIOGRAFIA

Libros

Cómo usar Power Point 4

Chordá, Ramón M.

Addison-Wesley Iberoamericana / Rama, E.U.A., 1994.

Fox Pro 2.5 para DOS y Windows

Chordá, Ramón M

Addison - Wesley Iberoamericana, RAMA México 1993

Introducción a los sistemas de bases de datos

Date, C. J.

Volumen uno, quinta edición, Addison - Wesley Iberoamericana S. A. 1993.

Semantic Object Modeling with SALSA

Dewitz, Sandra / Olson Michael

McGraw Hill; USA 1994

Microsoft Access 2 Paso a paso

Flores Gutiérrez, Mariano / Perea De Nalda Alicia

McGraw Hill, Catapult Inc. y Microsoft Corporation, México, 1994.

Introducción a las bases de datos

Gillenson, Mark L.

McGraw Hill primera edición, México 1987.

Fundamentos de bases de datos

Korth, Henry F. / Siberchatz Abraham

McGraw Hill, segunda edición, Interamericana de España S.A., 1993.

Organización de las bases de datos

Martin PHH, James.

Prentice Hall Inc.1992

Microsoft Access-Introducción

Microsoft Corporation, U.S.A., 1993

Microsoft Access-Introducción a la programación

Microsoft Corporation, U.S.A., 1993

- Microsoft Access Manual del usuario
Microsoft Corporation, U.S.A., 1993
- Microsoft Graph v. 3.0
Microsoft Corporation, U.S.A., 1993
- Users Manual
NX-1000II Multifont
Star Micronics Co., Ltd. Japan, 1989.
- dBASE IV
Osborne
McGraw Hill, México 1990
- Database design and applications development with Microsoft Access 2.0
Rob, Peter / Treyton Williams
McGraw Hill, México 1995
- Diseño de bases de datos
Wiederhold Gio
McGraw Hill, segunda edición, México 1994.

Revistas

- PC Magazine
Volumen 7, número 12
- PC Magazine
Volumen 3, número 9
- PC Magazine
Volumen 4, número 4
- PC Magazine
Volumen 2, número 11
- PC Magazine
Volumen 4, número 9
- PC Magazine
Volumen 5, número 10

GLOSARIO

TERMINO	SIGNIFICADO
1:1	Relación uno a uno. Existe si una entidad está dentro de un conjunto de entidades
1:M	Relación de uno a muchos.
1NF	Primera forma normal
2NF	Segunda forma normal
3NF	Tercera forma normal
ACCESS BASIC	Lenguaje estructurado en cual es una variante del lenguaje BASIC de Microsoft incorporado en Access
ANSI	Instituto Nacional Americano de Normalización
ARCHIVO	Terminología del sistema de archivo utilizada para representar un conjunto de entidades, una tabla o una relación en terminología relacional
ARGUMENTO	Define el dato que debe utilizarse en las acciones de una macro o la forma de utilizarlos
ASISTENTE	En Access es como un experto en base de datos que hace preguntas sobre el objeto que deseemos crear y lo construye de acuerdo a las instrucciones recibidas
ATRIBUTO	Entidad identificada con características conocidas como atributos
ATRIBUTO NO LLAVE	Atributos que no son parte de la llave primaria
ATRIBUTO DERIVADO	Atributo que se calculó o derivó de otros atributos almacenados en la tabla
ATRIBUTO SIMPLE	El atributo no puede ser subdividido, puede tener sólo un valor
ATRIBUTO COMPUESTO	El atributo pueden ser subdividido
BARRA DE HERRAMIENTAS	Conjunto de botones que al pulsarlos ejecutan una determinada acción

TERMINO	SIGNIFICADO
BASE DE DATOS	Estructura que contiene un conjunto de entidades y relaciones entre estas entidades
BOYCE CODD	Fundamentalmente es una estricta 3NF; debe estar basada en una llave primaria compuesta y todo elemento determinante en la relación debe ser candidato a llave
CAJA DE HERRAMIENTAS	Contiene todos los controles que podemos añadir a un formulario o un informe
CAMPO	Terminología del sistema de archivo utilizada para representar una columna o atributo en terminología relacional
CARDINALIDAD	Define la parte más baja y la parte más alta de la relación
CODIGO	Término utilizado para designar las instrucciones en un lenguaje de programación
COMPACTAR BASE DE DATOS	Elimina la fragmentación que presenta un archivo de base de datos en el disco duro cuando se han realizado muchas modificaciones en él, ocupará menos espacio y optimará el rendimiento
CONECTIVIDAD	Se refiere al tipo de relación en las entidades
CONJUNTO DE ENTIDADES	Colección de entidades relacionados
CONSTANTE	Representa un valor numérico o un valor de cadena que permanece inalterable en la ejecución de un programa
CONSULTA	Objeto que proporciona una información particular de los datos a partir de una o mas tablas y se puede crear mediante la gráfica QBE o escribiendo la sentencia en SQL
CONTROL	Objeto gráfico incluido en los formularios o informes
CUADRICULA	Malla de puntos visibles en el área de trabajo de la ventana de diseño de formularios o informes que nos ayudan a colocar o alinear los controles en un alto grado de precisión
DB	Bases de datos
DBMS	Sistema de administración de bases de datos
DD	Diccionario de datos
DER	Diagrama entidad relación

TERMINO	SIGNIFICADO
DENORMALIZACION	Reduce la forma normal, se realiza para simplificar la estructura de la base de datos cuando el costo de las anomalías de los datos en la estructura 2NF aparenta ser mínima; o para crear llaves primarias mas eficientes
DEPENDENCIA PARCIAL	Es dependiente solamente de parte de la llave primaria
DEPENDENCIA TRANSITIVA	Cuando un atributo es dependiente de otro atributo que no es parte de la llave primaria, ni de una llave primaria compuesta
DESCRIPCION DE OPERACIONES	Documento que establece la jerarquía organizacional y entorno básico de los datos
DIAGRAMA DE DEPENDENCIA	Esquema usado para identificar diferentes tipos de dependencia (llave primaria, parcial o transitiva)
DICCIONARIO DE DATOS	Estructura que almacena metadatos, requiere de un espacio físico para los nombres de los atributos encontrados en cada conjunto de entidades y la relación entre éstas
E-R	Entidad Relación
ENTIDAD	Persona, cosa o evento en la cual se quieren guardar datos
ENTIDAD COMPUESTA	Conocida también como puente de entidades y sirve conexión entre entidades
ENTIDAD DEBIL	La entidad no puede existir por si misma
ESQUEMA RELACIONAL	Conexión mantenida por el atributo común
EVENTO	Acción reconocida por un objeto, por ejemplo pulsar el ratón o presionar una tecla para la que puede definirse una respuesta
FILTRO	Tipo especial de consulta que permite aplicar un conjunto de criterios o condiciones a los registros representados en un formulario con el fin de ordenarlos o mostrar un subgrupo de ellos
FORMULARIO	Objeto diseñado para introducir o visualizar los datos con los que deseamos trabajar y la disposición que tendrán cuando se visualicen en pantalla
FUNCION	Realiza operaciones o acciones sin tener que programarlas, es decir realiza cálculos sobre los datos, introducidos como argumentos y devuelve un resultado
GENERADOR	Herramienta de Microsoft Access que simplifica una tarea

TERMINO	SIGNIFICADO
HOJA DE DATOS	Modo de presentación de los datos que muestra los registros en una cuadrícula, semejante a las celdas de una hoja de cálculo, muestra los nombres de los campos como encabezados de columnas y los registros en filas
IDENTIFICADOR	Se utiliza en una expresión para hacer referencia el valor de un campo, control o propiedad de un formulario o informe
LLAVE CANDIDATA	En cada entidad puede existir mas de un conjunto de atributos que satisfagan los requisitos de una llave primaria
LLAVE EXTERNA	Un atributo en una tabla es referido por referencia de otras tablas
LLAVE PRIMARIA	Identificación única para cada entidad de una tabla
LLAVE PRIMARIA COMPUESTA	Llave primaria formada por una combinación de atributos
LLAVE SECUNDARIA	Compuesta de una combinación de atributo que la gente probablemente recuerda, produce sólo un registro
M:N	Relación de muchos a muchos
MACRO	Objeto que define en forma estructurada la acción de una o mas tareas que realice en Microsoft Access
MER	Modelo entidad relación
METADATOS	Datos acerca de los datos
MICROSOFT ACCESS	Sistema interactivo de administración de base de datos para Microsoft Windows que permite realizar tareas de gestión de base de datos
MODELO	Representación simplificada del mundo real del objeto o condición
MODELO ENTIDAD RELACION	Representa abstracciones del entorno real de los datos
MODULO	Objeto que contiene procedimientos personales que se configuran utilizando Access BASIC
NORMALIZACION	Proceso que detiene la redundancia y produce técnicas a través de las cuales éstas son controladas
OBJETO	Cualquier entidad del entorno de Access que puede ser manipulada o controlada, como tablas, consultas, gráficos, etc.



TERMINO	SIGNIFICADO
ODBC	Conectividad de base de datos abierta. Es un interfaz de conexión a bases de datos que viene con el programa Access y contiene los controladores apropiados para acceder a datos de la base de datos SQL e importar, exportar o adjuntar los datos almacenados en dichas bases de datos
OLE	Vinculación e incrustación de objetos. Protocolo por medio del cual un objeto (tal como un gráfico) contenido en una aplicación o documento puede vincularse o incrustarse en otro documento (formulario o informe)
PALETA DE COLORES	Proporciona elementos de diseño para dar vistosidad y colorido a los formularios uniformes
PRIMERA FORMA NORMAL	Los componentes de la llave primaria están definidos, los atributos no llave son dependientes de la llave primaria y el contenido de una tabla no está repetido por grupos
PROCEDIMIENTO	Es un programa creado por el lenguaje de programación Access BASIC incorporado en Access
PROPIEDADES	Atributos específicos de pantallas o controles que definen las características de los objetos
QBE	Consulta mediante ejemplo
RDBMS	Sistema de administración de bases de datos relacionales
REDUNDANCIA DE DATOS	Innecesaria duplicación de los datos
REGISTRO	Terminología del sistema de archivo utilizada para describir una entidad, tupla o renglón en terminología relacional
REGLAS DE NEGOCIOS	Descripciones exactas de los procedimientos de operaciones de un negocio o empresa
RELACION	Asociación entre varias entidades
RELACION OBLIGATORIA	Depende de las reglas de negocio la relación siempre va a ocurrir para cualquier caso
RELACION OPCIONAL	Depende de las reglas de negocio en donde se especifica que no necesariamente una relación es asignada; la relación se marca con círculo
SECCIONES	Apartados en los que se dividen los formularios e informes, que proporcionan el formato final y contienen la información que se presentará en él



TERMINO	SIGNIFICADO
SISTEMA DE ADMINISTRACION DE BASE DE DATOS	Es el software que adquiere como Microsoft Access; administra y relaciona la base de datos almacenada.
SISTEMA DE ADMINISTRACION DE BASE DE DATOS RELACIONALES	Dirige el medio ambiente de la base de datos relacional, donde las relaciones están representadas conceptualmente por tablas
SISTEMA DE BASE DE DATOS	Es creado cuando un DBMS es adquirido y es combinable con la base de datos que designe e implemente para las aplicaciones que haya desarrollado.
SISTEMA DE INFORMACION	Se refiere cuando el sistema de base de datos es usado, en las formas básicas, en una decisión formal
SUBTIPO	Entidad basada en la especialización de atributos
SEGUNDA FORMA NORMAL	No hay dependencia parciales
SUPERTIPOS	Entidades basadas en atributos comunes
SQL	Lenguaje de consulta estructurada. Lenguaje que se utiliza para consultar, actualizar y gestionar base de datos relacionales
TABLA	Conjunto de entidades
TABLA	Objeto que se utiliza para almacenar los datos en Microsoft Access
TERCERA FORMA NORMAL	No debe contener grupos repetidos, dependencias parciales, ni transitivas
TUPLA	Renglones de las tablas relacionadas
UNDERPINING VALOR PREDETERMINADO	Modelo teórico proporcionando por una teoría matemática física. Introduce en el campo un valor cuando se agregan nuevos registros
VARIABLE	Guarda temporalmente los datos obtenidos al ejecutar las líneas de código escritas en un procedimiento