

The background of the page features a large, light blue letter 'U' that frames the central text. Above the 'U', there are five orange circles of varying sizes arranged in a semi-circular pattern, resembling a sun or a decorative element.

“UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA”

ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA

PROGRAMA TECNOLOGÍA E INGENIERIA DE SISTEMAS

301330 –Bases de Datos Básicos

MARIA DORA ALBA SÁNCHEZ GÓMEZ¹

MARIA DORA ALBA SÁNCHEZ GÓMEZ
(Directora Nacional)

ANYELO QUINTERO REYES
(Acreditador)

Medellín, Julio de 2011

¹ Ingeniera de Sistemas, Magistra en Administración de Empresas. Medellín, Julio de 2011

DEDICATORIA

Este módulo, fruto de la experiencia profesional de once años de trabajo y quince años en la academia, se lo dedico a Dios, quien nos da la sabiduría, la inteligencia y la fortaleza, para emprender trabajos nobles. A mi madre, quien con su amor y dedicación, posibilitó que yo fuera una profesional honesta y transparente. A mis hijos, por su paciencia y acompañamiento, aún en momentos en que a veces estando con ellos, me encontraba ausente.

Dios los bendiga a todos



AGRADECIMIENTOS

A la UNAD, por darme la oportunidad de servirle y poner a su disposición todos mis conocimientos. A los compañeros, quienes me ayudaron para que este material hoy sea visible para la comunidad académica Unadista.

A todos los autores, quienes con sus libros ayudaron la construcción conceptual del módulo, lo mismo que a todos los instructores quienes a través de sus cursos y diplomados, me ayudaron a comprender y a aplicar todos los conceptos.



ASPECTOS DE PROPIEDAD INTELECTUAL Y VERSIONAMIENTO

El presente módulo es la segunda edición, fue elaborado en el año 2011 por María Dora Alba Sánchez Gómez, Docente auxiliar de tiempo completo de la UNAD, actualmente laborando en la zona de Occidente y específicamente en el Cead de Medellín.

Este módulo contiene las lecciones 2, 3 y 4 del capítulo 1, unidad 1; de la primera edición, realizada por el ingeniero Jhon Alejandro Figueredo Luna.

La autora es “ingeniera de sistemas” y “magistra en administración de empresas”. Se ha desempeñado como docente auxiliar de la UNAD desde febrero de 2009 hasta la fecha y ha sido docente de tiempo completo, medio tiempo y catedrática de en varias instituciones de educación superior en Cartagena: Universidad San Buenaventura, Instituto Universitario Comfenalco, Corporación Universitaria Rafael Núñez y Tecnar y está participando en el grupo de investigación registrado en Colciencias que se denomina “Ciencia y tecnología con Don Fuan”.

El presente módulo es el producto de mis once años como profesional en diferentes empresas y con mi experiencia académica de quince años.

El autor espera mejorar y actualizar este material de estudio en el 2011 y para ello espera sus aportes, sugerencias, inquietudes. Felicidades en este bello viaje.

INTRODUCCIÓN

El presente módulo (Bases de Datos Básico) está dirigido a estudiantes de los programas de pregrado (tecnología e ingeniería de sistemas) que oferta la UNAD, bajo la modalidad de educación superior abierta y a distancia.

El material está estructurado en tres (3) unidades que son las temáticas macro del curso académico. El contenido de cada una de las partes fue seleccionado, teniendo en cuenta los saberes mínimos que se esperaba debe alcanzar un estudiante de la UNAD (Universidad Nacional Abierta y a Distancia) al culminar sus estudios en el curso de “Bases de Datos Básico”.

La propuesta permite que los estudiantes reconozcan los conocimientos fundamentales del curso, que les permita resolver situaciones propias del mismo y además, abordar posteriores temáticas que requieran de estos conocimientos.

Los ingenieros o tecnólogos en sistemas, además de un sólido soporte en teoría de conjuntos deben tener una gran capacidad lógica para interpretar y diseñar adecuadamente los estudios de casos que se proponen, con base en la técnica de modelado “entidad-relación”. Estudiar, analizar, socializar, los diferentes casos, es importante para tener un aprendizaje significativo, que permita comprender todos los conceptos necesarios para un diseño razonable de éstas.

El estudio de las bases de datos, es fundamental para el diseño de sistemas de información, pues éstas son la columna vertebral, la estructura y, un mal diseño lleva en poco tiempo a degenerarlo, a brindar información errónea e inconsistente.

El módulo se caracteriza porque en cada unidad, se presentan estudios de casos y ejercicios, donde se configura un problemática específica en cada una de las temáticas abordadas. Al final de cada unidad se presenta una Autoevaluación de un nivel medio-alto, las cuales permiten verificar los alcances de los estudiantes en las temáticas analizadas y detectar las debilidades y así centrarse en éstas, con el fin de alcanzar las competencias propuestas.

Finalmente, el material pretende servir como guía de aprendizaje autónomo y se recomienda apoyar este proceso con lecturas especializadas, ayudas audiovisuales, visitas a sitios Web o realización de prácticas significativas (entre otras), para lograr una efectiva comprensión, interiorización y aplicación de las temáticas estudiadas en el desarrollo del curso de “Bases de Datos Básico”.

INDICE DE CONTENIDO

UNIDAD UNO: **FUNDAMENTOS DE BASE DE DATOS**

- **CAPÍTULO 1. INTRODUCCIÓN A LAS BASES DE DATOS**

Lección 1: Conceptos Generales

Lección 2: Tipos de Bases de Datos

Lección 3: Sistema Gestor de Bases de Datos

Lección 4: Arquitectura de Bases de Datos

Lección 5: Planificación de Proyectos de Análisis, Diseño e Implementación de Bases de Datos.

- **CAPÍTULO 2. FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE BASES DE DATOS RELACIONAL**

Lección 6: Concepto de Modelo de Datos

Lección 7: Procedimiento para elaborar un Modelo

Lección 8: Desarrollo Caso de Empleados

Lección 9: Análisis Caso de Cursos

Lección 10: Análisis Caso de Órdenes de Pedidos

- **CAPÍTULO 3. MODELO DE DATOS BASADOS EN FORMATOS**

Lección 11: Introducción y Tipos de Formatos

Lección 12: Procedimiento para elaborar Modelo basados en formatos

Lección 13: Análisis formato Datos Básico Proveedores

Lección 14: Análisis formato Vacaciones Empleados

Lección 15: Análisis formato Créditos Bancarios

UNIDAD DOS: **MODELO RELACIONAL**

- **CAPÍTULO 4. DISEÑO MODELO RELACIONAL DE DATOS**

Lección 16: Concepto del Modelo Relacional

Lección 17: Paso de un Modelo lógico de Datos a un Modelo Relacional

Lección 18: Paso Modelo lógico Datos Básico Proveedores a Modelo Relacional

Lección 19: Paso Modelo lógico Órdenes de Pedidos a Modelo Relacional

Lección 20: Paso Modelo lógico Créditos Bancarios a Modelo Relacional

- **CAPÍTULO 5. ÁLGEBRA RELACIONAL**

Lección 21: Conceptos de Álgebra Relacional

Lección 22: Operaciones Básica

Lección 23: Operaciones Avanzadas

Lección 24: Análisis de Ejecución de operaciones Caso de Reserva

Lección 25: Análisis de Ejecución de operaciones Caso de Certámenes

- **CAPÍTULO 6. NORMALIZACIÓN**

Lección 26: Conceptos de Normalización

Lección 27: Formas Normales

Lección 28: Análisis de Normalización Caso Matrícula

Lección 29: Análisis de Normalización Caso Mensajería

Lección 30: Análisis de Normalización Caso Biblioteca

UNIDAD TRES: LENGUAJE ESTANDAR DE CONSULTA Y HERRAMIENTAS

- **CAPÍTULO 7. LENGUAJE DE DEFINICIÓN DE DATOS**

Lección 31: Creación de Bases de Datos y Tablas

Lección 32: Creación de Índices

Lección 33: Creación de Vistas

Lección 34: Modificación de Tablas

Lección 35: Borrados de Bases de Datos, Tablas, Índices y Vistas

- **CAPÍTULO 8. LENGUAJE DE MANIPULACIÓN DE DATOS**

Lección 36: Inserción de Filas

Lección 37: Actualización de Filas

Lección 38: Borrado de Filas

Lección 39: Consultas Básicas

Lección 40: Consultas Avanzadas

- **CAPÍTULO 9. HERRAMIENTAS MySQL**

Lección 41: Generalidades de la Herramienta

Lección 42: Creación de Bases de Datos y Tablas

Lección 43: Inserción, Actualización y Borrado de Filas

Lección 44: Consultas Básicas

Lección 45: Consultas Avanzada

LISTADO DE GRÁFICOS Y FIGURAS

- Figura No 1: Modelo Jerárquico
- Figura No 2: Modelo en Red
- Figura No 3: Modelo en Red
- Figura No 4: Modelo Orientado a Objetos
- Figura No 5: Bases de datos centralizadas
- Figura No 6: Red de comunicaciones
- Figura No 7: Bases de datos distribuida
- Figura No 8: Bases de datos distribuida en redes
- Figura No 9: Entidad fuerte
- Figura No 10: Entidad débil
- Figura No 11: Entidad asociativa
- Figura No 12: Tipos de cardinalidad
- Figura No 13: Matriz relación caso simple empleados
- Figura No 14: Matriz relación caso empleados
- Figura No 15: DER- 1 caso empleados
- Figura No 16: DER- 2 caso empleados
- Figura No 17: DER- 3 caso empleados
- Figura No 18: DER- 4 caso empleados
- Figura No 19: DER- 5 caso empleados
- Figura No 20: DER- 6 caso empleados
- Figura No 21: DER- 7 caso empleados

Figura No 22: **Matriz relación caso curso**

Figura No 23: **DER-1 caso curso**

Figura No 24: **DER-2 caso curso**

Figura No 25: **DER-3 caso curso**

Figura No 26: **DER-4 caso curso**

Figura No 27: **DER-5 caso curso**

Figura No 28: **DER-6 caso curso**

Figura No 29: **DER-7 caso curso**

Figura No 30: **Matriz relación caso órdenes**

Figura No 31: **DER-1 caso órdenes**

Figura No 32: **DER-2 caso órdenes**

Figura No 33: **DER-3 caso órdenes**

Figura No 34: **DER-4 caso órdenes**

Figura No 35: **DER-5 caso órdenes**

Figura No 36: **DER-6 caso órdenes**

Figura No 37: **DER-7 caso órdenes**

Figura No 38: **DER-8 caso órdenes**

Figura No 39: **DER-9 caso órdenes**

Figura No 40: **Formato de un solo cuerpo**

Figura No 41: **Formato de dos cuerpo**

Figura No 42: **Formato de varios cuerpo**

Figura No 43: **Entidades iniciales formato un solo cuerpo**

Figura No 44: **Entidades iniciales formato dos cuerpos**

Figura No 45: Entidades iniciales formato varios cuerpos

Figura No 46: Asignación de atributos

Figura No 47: Formato actualización datos proveedores

Figura No 48: Identificación entidades caso proveedores

Figura No 49: Asignación de atributos caso proveedores

Figura No 50: Identificación de mas entidades y atributos



UNIDAD 1

Nombre de la Unidad	FUNDAMENTOS DE BASES DE DATOS
Introducción	<p>En el mundo de las organizaciones, los sistemas de información son vitales como apoyo a la gestión de los procesos administrativos y operativos. Es así, como las bases de datos se consideran la columna vertebral de éstos. De allí la importancia de aprender bien los conceptos para analizar y diseñar el modelo lógico de datos.</p> <p>En este capítulo se presentan los conceptos fundamentales necesarios para dimensionar la importancia de éstas, así como los conceptos para analizarlas y diseñarlas en el mundo de las organizaciones.</p> <p>Sin lugar a dudas, el estudio de las bases de datos es un tema interesante y necesario, de actualidad tecnológica, que es base para estudios posteriores de sistemas más complejos.</p>
Intencionalidades Formativas	<ul style="list-style-type: none"> • Fundamentar los conceptos de las bases de datos relacionales para su análisis y diseño lógico. • Aportar a los estudiantes experiencias significativas a través del estudio de casos, el cual contribuye a desarrollar sus habilidades para argumentar, razonar o dar explicaciones o justificaciones a los diseños planteados por ellos mediante la técnica “entidad-relación” • Potenciar en los estudiantes la capacidad de comprensión y aprehensión de los conceptos específicos del diseño lógico de las bases de datos relacional. • Contribuir al desarrollo de habilidades de pensamiento en los estudiantes de los programas de tecnología e ingeniería de sistemas que oferta la UNAD mediante la estimulación cognitiva de modelado que faciliten la apropiación de nociones, conceptos, experiencias y técnicas que fundamentan el diseño lógico de las bases de datos relacional. • Fortalecer en el participante las características que

	<p>deben identificarlo en su desempeño y actuación como tecnólogo(a) e ingeniero(a) de sistemas.</p> <ul style="list-style-type: none"> • Desarrollar en el estudiante las aptitudes y las actitudes que le permitan analizar, comprender y diseñar el modelo lógico de datos en las organizaciones.
Denominación de capítulos	<ul style="list-style-type: none"> • Introducción a las Bases de Datos • Fundamentos de Análisis y Diseño de Bases de Datos • Modelo de Datos basado en Formatos

UNIDAD 1.

FUNDAMENTOS DE BASE DE DATOS

En este capítulo se presentan los conceptos fundamentales necesarios para dimensionar la importancia de las bases de datos relacional en las organizaciones, así como los conceptos y desarrollo de estudio de casos para analizarlas y diseñarlas lógicamente.

Para el análisis y diseño lógico de las bases de datos relacional, se utilizará la técnica “entidad-relación” con notación de “patagallina”; esta notación es sencilla y ofrece menos dificultad que la notación de Chen y Cood, debido a que solo maneja cuatro tipos de elementos como son la entidad, el atributo, la relación y la cardinalidad.

Por otro lado, para abordar las problemáticas de la estructuración de los datos en una organización y el diseño adecuado de ésta, se presentarán y desarrollarán estudio de casos que son analizados y diseñados. Así mismo se abordarán desde el estudio y análisis de los formatos que utiliza la organización y que hacen de estos un material importante de recolección de información.

Por lo tanto, esta unidad presenta tres capítulos, donde el primero fundamenta en conceptos generales, los tipos, los gestores, su arquitectura para terminar en la planificación de proyectos de diseño de bases de datos. En el segundo capítulo, se fundamenta en los conceptos del modelo de datos lógico basado en la técnica entidad relación, el procedimiento para elaborarlo y el desarrollo de dos estudios de caso. En el tercer capítulo, se realiza una introducción a los tipos de formatos que se manejan en la organización, el procedimiento para elaborar el modelo de datos a partir de éstos y el desarrollo de tres estudios de casos.

CAPÍTULO 1: “INTRODUCCIÓN A LAS BASES DE DATOS”

Lección 1: “Conceptos Generales”

- **propósitos de los sistemas**

La información hoy en día es el motor de toma de decisiones y el tesoro maspreciado de las organizaciones, ya que al conocer de una manera adecuada y de una forma acertada los procesos internos y externos se puede en determinado momento ayudar a tomar decisiones, esto con fin de lograr una mejor producción posición en el mercado.

El manejo de gran cantidad de datos es la consecuencia del aumento de información que se maneja en el transcurso de nuestras vida y mas aun en el mundo empresarial, al incrementar todo este volumen de información que diariamente se puede acumular en el manejo de una empresa u organización se hace necesario organizarla para poder encontrar resultados rápidos y óptimos en el momento de utilizarla.

Debido a esta necesidad en los años setenta, para manejar toda esta información surgen las bases de datos, en la cual se integran archivos individuales para poder ser compartidos por todos los usuarios de la empresa. Para el desarrollo de esta técnica de almacenamiento y ordenamiento de información es necesario conocer la causa que llevo el desenvolvimiento de dicho proceso, uno de ellos fue la transmisión de la información o los datos, es decir que el usuario tenga capacidad y facilidad de acceder a los datos de una forma remota. Que el proceso de dialogo que se genere entre el usuario y la maquina sea el mas amigable y compatible en donde se pueda consultar y borrar, modificar e insertar datos en cualquier momento que se necesite en el manejo de la información.

El diseño de la base de datos es de gran importancia en el manejo de la información, ya que tiene como principal objetivo que los datos almacenados se puedan utilizar por una gran numero de aplicaciones. Los predecesores de los sistemas de bases de datos fueron los sistemas de ficheros o archivos. No hay un momento concreto en que los sistemas de archivos hayan cesado y hayan dado comienzo los sistemas de bases de datos. De hecho, todavía existen sistemas de archivos en uso.

Por otro lado, como se ha observado, son muchas las aplicaciones que requieren acceder a datos, bien sea un sencillo programa doméstico, o bien una suite para la gestión empresarial. Estos datos se deben almacenar en algún soporte permanente, y las aplicaciones deben disponer de un medio para acceder a ellos. Normalmente, la forma en que un programa accede a un fichero es a través del Sistema operativo. Éste provee de funciones como abrir archivo, leer información

del archivo, guardar información, etc. No obstante, este procedimiento de acceso a ficheros es altamente ineficaz cuando se trata con un volumen elevado de información. Es aquí donde aparecen los Sistemas Gestores de Bases de Datos: proporcionan un interfaz entre aplicaciones y sistema operativo, consiguiendo, entre otras cosas, que el acceso a los datos se realice de una forma más eficiente, más fácil de implementar, y, sobre todo, más segura.

Un sistema gestor de bases de datos (SGDB) consiste en un conjunto de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGDB es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente.

Por lo tanto, los sistemas de gestión de bases de datos, se diseñan para almacenar y manipular grandes cantidades de datos con el fin de brindar información.

- Definición de de datos e información

[Ver artículo.](#)

Ejemplo de datos serían:

Vista: los colores (amarillo, rojo, verde, etc.); tamaño (alto, bajo, mediano), morfología (delgado, grueso, gordo); símbolos (número, letras, grafos)

Tacto: Textura (Áspero, suave, etc); Peso (liviano, pesado, etc.); temperatura (caliente, frío, normal)

Oído: Contraste (Suave, fuerte, movido, etc.); Volumen (Alto, medio, bajo, etc.)

Olfato: Olores (agradable, desagradable, etc.)

Boca: Sabores (dulce, salado, amargo, picante, etc.)

Ejemplos de información sería:

La fecha de cumpleaños de los amigos, los familiares, la fecha de ingreso a trabajar, etc.; el dato sería la fecha.

El color amarillo, azul y rojo de la bandera de Colombia; los datos serían amarillo, azul y rojo.

El olor del perfume en el aire, le recuerda a una persona: el dato sería olor agradable.

Y así podríamos quedarnos dando ejemplos. Pero desde el ámbito de las organizaciones, los datos y la información es todavía más compleja, puesto que en la primera, deben estar bien estructurados y la segunda debe cumplir con unas características.

En cuanto a la estructura de los datos, existen muchas formas de realizarlo y una de ellas es sobre el concepto de las bases de datos, que es el estudio, que nos ocupará en este curso.

En cuanto a las características de la información, esta debe ser:

- Relevante
- Precisa
- Disponible

- Antecedentes de las bases de datos

[Ver documento.](#)

- Ventajas y desventajas de las bases de datos

Ventajas

- Reduce la redundancia de datos, esto debido a que un datos o conjunto de datos, solo se define “conceptualmente” una vez, es decir reduce los sinónimos y homónimos.
- Reduce la programación y mantenimiento de los datos. Esto como consecuencia, de que solo un datos o conjunto de datos, es definido una vez en el modelo. Por lo tanto, cada vez que necesite ser modificado se debe ir a una sola parte del modelo.
- Incrementa la seguridad provista a datos almacenados, esto es, que como se tiene totalmente identificado los datos, solo existe un lugar donde pueden ser accedidos y por lo tanto su control es más fácil.
- Incrementa la integridad de los datos. Como todos los datos se encuentran relacionados en forma lógica, hace que cualquier actualización en forma ilógica sea fácilmente detectada.
- Incrementa compartir datos, ya que se encuentran en forma lógica centralizados, todos los procesos que desean acceder a los datos, van al mismo sitio.
- Incrementa la disponibilidad de datos. Todos pueden acceder al mismo datos en un sitio determinado, su restricción solo se basa a permisos de acceso y no a un lugar.

- Incrementa la independencia de datos, esto es, los datos son independiente del programa o persona que lo utilice.
- Mejora el manejo y control de datos. Es mas fácil el manejo y control de datos que estén centralizados y relacionados, de otros, cuya estructura se encuentra en cada uno de los procesos que lo utilizan (tercera generación).
- Mejora el acceso de datos. Es más fácil el acceso de datos centralizados que el acceso de datos descentralizados, que muchas veces no se saben dónde se encuentran definidos ni almacenados.

Desventaja

- Mayor consumo de recursos de hardware y software.

Lección 2: “Tipos de bases de datos”

a) Modelo Jerárquico

Estas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres se le conoce como raíz, y a los nodos que no tienen hijos se les conoce como hojas. Una de las principales limitaciones de este modelo, es su incapacidad de representar eficientemente la redundancia de datos. Los registros se organizan como colecciones de árboles, en lugar de grafos dirigidos.

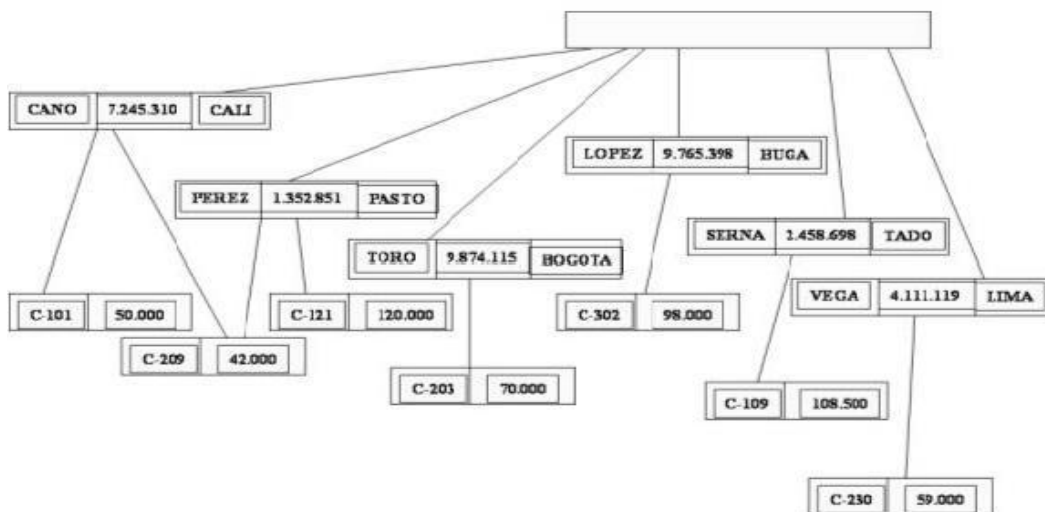


Figura 1- Modelo Jerárquico

b) Modelo en Red

Éste es un modelo ligeramente distinto del jerárquico, en donde su diferencia fundamental es la modificación del concepto de un nodo, permitiendo que un mismo nodo tenga varios padres (algo no permitido en el modelo jerárquico). Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos, pero aun así, la dificultad que significa administrar la información en una base de datos de red, ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales. Colecciones de registros y las relaciones entre datos se representan mediante enlaces dirigidos.

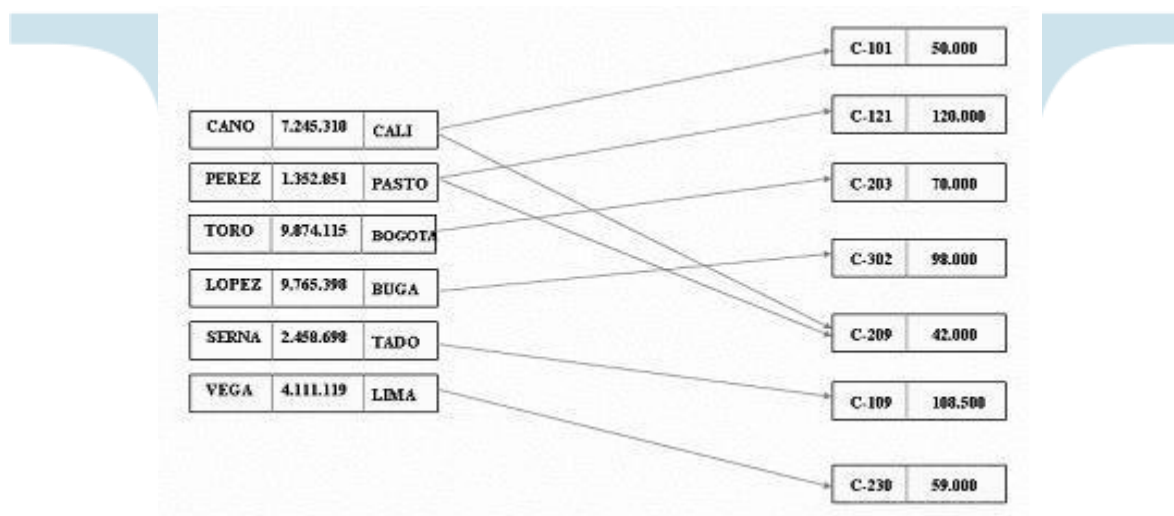


Figura 2 – Modelo en Red

c) Modelo Relacional

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postuladas su bases en 1970 por IBM, Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que esta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que esta compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario casual de la base de datos. La información puede ser recuperada o almacenada por medio de "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más común para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales. Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización de una base de datos.

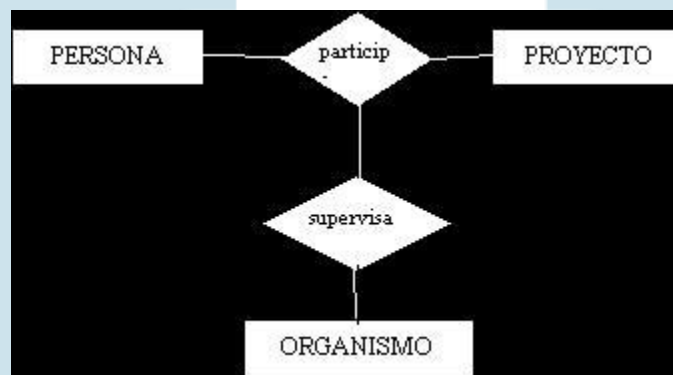


Figura 3 – Modelo relacional

d) Orientado a Objetos

Como sabemos una Base de Datos es un conjunto de datos y relaciones que representa una interfaz uniforme de usuario, que se describe por sí sola. La BD Relacional es un conjunto de relaciones formada por un esquema y un cuerpo que se describen en términos de dominios, atributos, asociaciones, tuplas; y la Base de Datos Relacional es una base de datos autodescriptiva por medio de sus tablas de sistema. Donde el modelo relacional satisface el espíritu de la definición introductoria. Contiene elementos de datos (tuplas) y relaciones entre ellos (por medio de atributos comunes). El SQL proporciona la interfaz uniforme.

Sin embargo Una BD Orientada a Objetos (BDOO) es una base de datos en el sentido de la definición introductoria, donde los elementos de datos son objetos y las relaciones se mantienen por medio inclusión lógica. Las entidades de aplicación están representadas como clases. La auto descripción se obtiene

porque las clases son meta objetos que contiene los nombres de atributos y métodos de señal. Una BDOO contiene un método sistemático de representación de relación, y la interfaz uniforme de usuario es un sistema de mensajes que puede explorar los objetos y sus interconexiones.

En una BDOO, las entidades de aplicación son las clases, las instancias de entidad son objetos creados desde las clases, y las relaciones se mantienen por medio de inclusión lógica. Un sistema de señales y métodos para procesarlas contiene una interfaz uniforme para la base de datos.



Figura 4 - Modelo Orientado a Objetos

- Tipos de bases de datos según la variabilidad
 - Bases de datos estáticas

Estas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar

proyecciones y tomar decisiones; ejemplo de éstas, son los Data Ware House ó Bodegas de Datos.

- Bases de datos dinámicas

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, etc. Éstas también se conocen como bases de datos Transaccionales.

- Tipos de bases de datos según su contenido

- Bases de datos bibliográficas

Solo contienen un surrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino estaríamos en presencia de una base de datos a texto completo (o de fuentes primarias—ver más abajo). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

- Bases de datos de texto completo

Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

- Directorios

Un ejemplo son las guías telefónicas en formato electrónico.

- Bases de datos o "bibliotecas" de información Biológica

Son bases de datos que almacenan diferentes tipos de información proveniente de las ciencias de la vida o médicas. Se pueden considerar en varios subtipos:

Aquellas que almacenan secuencias de nucleótidos o proteínas.

- Bases de datos de rutas metabólicas
- Bases de datos de estructura, comprende los registros de datos experimentales sobre estructuras 3D de biomoléculas
- Bases de datos clínicas
- Bases de datos bibliográficas (biológicas)

Lección 3: “Sistema gestor de bases de datos”

a) Definición y Objetivo

Sistema de gestión de base de datos ó en inglés Database management system (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una bases de datos. El objetivo de estos programas, consiste en proporcionar las interfaces entre los datos de bajo nivel en la base de datos y los programas de aplicación y consultas emitidas al sistema.

El gestor de almacenamiento es responsable de la interacción con el gestor de archivos. Éste incluye:

- Gestor de autorización e integridad, que comprueba que se satisfagan las restricciones de integridad y la autorización de los usuarios para acceder a los datos.
- Gestor de transacciones, que asegura que la base de datos quiere un estado consistente a pesar de los fallos del sistema, y que las ejecuciones de transacciones concurrentes ocurren sin conflictos.
- Gestor de archivos, que gestiona la reserva de espacio de almacenamiento de disco y las estructuras de datos usadas para representar la información almacenada en disco.

Gestor de memoria intermedia, que es responsable de trae los datos del disco de almacenamiento a memoria principal y describir que datos tratar en memoria caché.

Gestor de almacenamiento, también implementa varias estructuras de datos como parte de la implementación física el sistema:

- Archivos de datos.
- Diccionario de datos.
- Índices.

Por otro lado, los gestores tienen un procesador de consultas. Los componentes del procesador de consultas incluyen:

- Intérprete del LDD, que interpreta las instrucciones del LDD y registra las definiciones en el diccionario de datos.
- Compilador del LMD, que traduce las instrucciones del LMD en un lenguaje consultas a un plan de evaluación que consiste en instrucciones de bajo nivel que entiende el motor de evaluación de consultas.
- Motor de evaluación de consultas, que ejecutan las instrucciones de bajo nivel generadas por el compilador del LMD.

b) Tipos de Sistema Gestor de Bases de Datos

- Bases de datos centralizadas

Los sistemas de bases de datos centralizados son aquellos que se ejecutan en un único sistema informático sin interaccionar con ninguna otra computadora. Tales sistemas comprenden el rango desde los sistemas de bases de datos monousuario ejecutándose en computadoras personales hasta los sistemas de bases de datos de alto rendimiento ejecutándose en grandes sistemas.

Una computadora moderna de propósito general consiste en una o unas pocas CPU's y un número determinado de controladores para los dispositivos que se encuentren conectados a través de un bus común, el cual proporciona acceso a la memoria compartida. Las CPU's poseen memorias caché locales donde se almacenan copias de ciertas partes de la memoria para acelerar el acceso a los datos. Cada controlador de dispositivo se encarga de un tipo específico de dispositivos (por ejemplo, una unidad de disco, una tarjeta de sonido o un monitor). La CPU y los controladores de dispositivo pueden ejecutarse concurrentemente, compitiendo así por el acceso a la memoria. La memoria caché reduce el acceso a la memoria, ya que la CPU necesita acceder a la memoria compartida un número de veces menor. Ver figura:

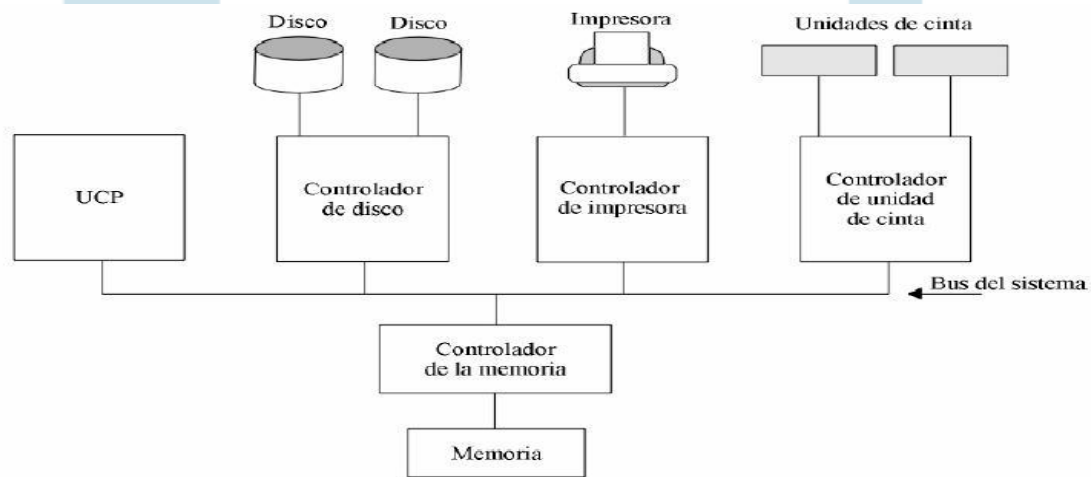


Figura 5 – bases de datos centralizadas

Se distinguen dos formas de utilizar las computadoras: como sistemas mono usuario o como sistemas multiusuario. En la primera categoría están las computadoras personales y las estaciones de trabajo. Un sistema mono usuario típico es una unidad de sobremesa utilizada por una única persona que dispone de una sola CPU, de uno o dos discos fijos y que trabaja con un sistema operativo que sólo permite un único usuario. Por el contrario, un sistema multiusuario típico tiene más discos y más memoria, puede disponer de varias CPU y trabaja con un sistema operativo multiusuario. Se encarga de dar servicio a un gran número de usuarios que están conectados al sistema a través de terminales. Estos sistemas se denominan con frecuencia sistemas servidores.

Normalmente, los sistemas de bases de datos diseñados para funcionar sobre sistemas mono usuario, como las computadoras personales, no suelen proporcionar muchas de las facilidades que ofrecen los sistemas multiusuario. En particular, no tienen control de concurrencia, que no es necesario cuando solamente un usuario puede generar modificaciones. Las facilidades de recuperación en estos sistemas, o no existen o son primitivas; por ejemplo, realizar una copia de seguridad de la base de datos antes de cualquier modificación. La mayoría de estos sistemas no admiten SQL y proporcionan un lenguaje de consulta muy simple, que en algunos casos es una variante de QBE (Query By Example).

Aunque hoy en día las computadoras de propósito general tienen varios procesadores, utilizan paralelismo de grano grueso, disponiendo de unos pocos procesadores (normalmente dos o cuatro) que comparten la misma memoria principal. Las bases de datos que se ejecutan en tales máquinas habitualmente no intentan dividir una consulta simple entre los distintos procesadores, sino que ejecutan cada consulta en un único procesador, posibilitando la concurrencia de varias consultas. Así, estos sistemas soportan una mayor productividad, es decir, permiten ejecutar un mayor número de transacciones por segundo, a pesar de que cada transacción individualmente no se ejecuta más rápido.

Las bases de datos diseñadas para las máquinas monoprocesador ya disponen de multitarea, permitiendo que varios procesos se ejecuten a la vez en el mismo procesador, usando tiempo compartido, mientras que de cara al usuario parece que los procesos se están ejecutando en paralelo. Ver ejemplo base de datos centralizada.



Figura 6 – Red de comunicaciones

- **Bases de datos Distribuidas**

Los sistemas de bases de datos distribuidas son un caso particular de los sistemas de cómputo distribuido en los cuales un conjunto de elementos de procesamiento autónomos (no necesariamente homogéneos) se interconectan por una red de comunicaciones y cooperan entre ellos para realizar sus tareas asignadas. Entre los términos más comunes que se utilizan para referirse al cómputo distribuido podemos encontrar: funciones distribuidas, procesamiento distribuido de datos, multiprocesadores, multicomputadoras, procesamiento satelital, procesamiento tipo "backend", computadoras dedicadas y de propósito específico, sistemas de tiempo compartido, sistemas funcionalmente modulares.

En el cómputo distribuido existen muchos componentes para realizar una tarea. Los elementos que se pueden distribuir son:

- Control. Las actividades relacionadas con el manejo o administración del sistema.
- Datos. La información que maneja el sistema.
- Funciones. Las actividades que cada elemento del sistema realiza.
- Procesamiento lógico. Las tareas específicas involucradas en una actividad de procesamiento de información.



Figura 7 – Bases de datos distribuida

Una base de datos distribuida (BDD) es un conjunto de múltiples bases de datos lógicamente relacionadas las cuales se encuentran distribuidas entre diferentes sitios interconectados por una red de comunicaciones. Un sistema de bases de datos distribuidas (SBDD) es un sistema en el cual múltiples sitios de bases de datos están ligados por un sistema de comunicaciones, de tal forma que, un usuario en cualquier sitio puede acceder los datos en cualquier parte de la red exactamente como si los datos estuvieran almacenados en su sitio propio.

Un sistema de manejo de bases de datos distribuidas (SMBDD) es aquel que se encarga del manejo de la BDD y proporciona un mecanismo de acceso que hace que la distribución sea transparente a los usuarios. El término transparente

significa que la aplicación trabajaría, desde un punto de vista lógico, como si un solo SMBD ejecutado en una sola máquina, administrara esos datos.

Un sistema de base de datos distribuida (SBDD) es entonces el resultado de la integración de una base de datos distribuida con un sistema para su manejo. Dada la definición anterior, es claro que algunos sistemas no se pueden considerar como SBDD. Por ejemplo, un sistema de tiempo compartido no incluye necesariamente un sistema de manejo de bases de datos y, en caso de que lo haga, éste es controlado y administrado por una sola computadora.

Un sistema de multiprocesamiento puede administrar una base de datos pero lo hace usualmente a través de un solo sistema de manejo de base de datos; los procesadores se utilizan para distribuir la carga de trabajo del sistema completo o incluso del propio SMBD pero actuando sobre una sola base de datos. Finalmente, una base de datos la cual reside en un solo sitio de una red de computadoras y que es accedida por todos los nodos de la red no es una base de datos distribuida. Este caso se trata de una base de datos cuyo control y administración esta centralizada en un solo nodo pero se permite el acceso a ella a través de la red de computadoras.

El medio ambiente típico de un SMBDD consiste de un conjunto de sitios o nodos los cuales tiene un sistema de procesamiento de datos completo que incluye una base de datos local, un sistema de manejo de bases de datos y facilidades de comunicaciones. Si los diferentes sitios pueden estar geográficamente dispersos, entonces, ellos están interconectados por una red de tipo WAN. Por otro lado, si los sitios están localizados en diferentes edificios o departamentos de una misma organización pero geográficamente en la misma ubicación, entonces, están conectados por una red local (LAN). Ver figura.

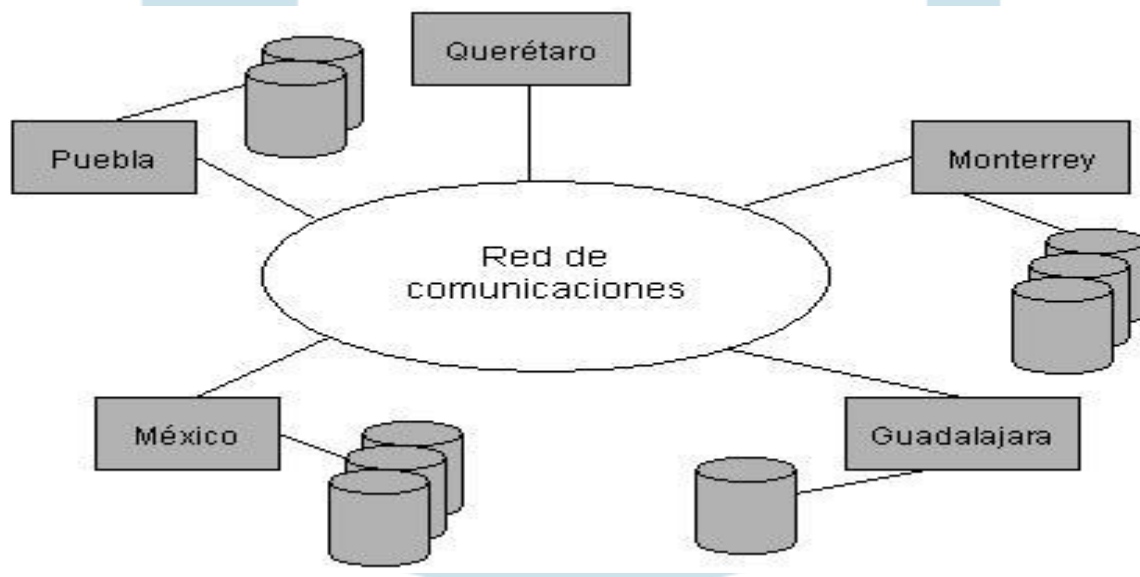


Figura 8 – Bases de datos distribuida en redes

c) Componentes

Los SGBD son paquetes de software muy complejos y sofisticados que deben proporcionar los servicios para el buen funcionamiento de la base de datos. No se puede generalizar sobre los elementos que componen un SGBD ya que varían mucho unos de otros. Sin embargo, es muy útil conocer sus componentes y cómo se relacionan cuando se trata de comprender lo que es un sistema de bases de datos.

Un SGBD tiene varios módulos, cada uno de los cuales realiza una función específica. El sistema operativo proporciona servicios básicos al SGBD, que es construido sobre él.

- El procesador de consultas es el componente principal de un SGBD, transforma las consultas en un conjunto de instrucciones de bajo nivel que se dirigen al gestor de la base de datos.
- El gestor de la base de datos es la interfase con los programas de aplicación y las consultas de los usuarios. El gestor de la base de datos acepta consultas y examina los esquemas externo y conceptual para determinar qué registros se requieren para satisfacer la petición. Entonces el gestor de la base de datos realiza una llamada al gestor de ficheros para ejecutar la petición.
- El gestor de ficheros maneja los ficheros en disco en donde se almacena la base de datos. Este gestor establece y mantiene la lista de estructuras e índices definidos en el esquema interno. Si se utilizan ficheros dispersos, llama a la función de dispersión para generar la dirección de los registros. Pero el gestor de ficheros no realiza directamente la entrada y salida de datos. Lo que hace es pasar la petición a los métodos de acceso del sistema operativo que se encargan de leer o escribir los datos en el buffer del sistema.
- El pre procesador del LMD convierte las sentencias del LMD embebidas en los programas de aplicación, en llamadas a funciones estándar escritas en el lenguaje anfitrión. El pre procesador del LMD debe trabajar con el procesador de consultas para generar el código apropiado.
- El compilador del LDD convierte las sentencias del LDD en un conjunto de tablas que contienen meta datos. Estas tablas se almacenan en el diccionario de datos.
- El gestor del diccionario controla los accesos al diccionario de datos y se encarga de mantenerlo. La mayoría de los componentes del SGBD acceden al diccionario de datos.

Los principales componentes del gestor de la base de datos son los siguientes:

- Control de autorización. Este módulo comprueba que el usuario tiene los permisos necesarios para llevar a cabo la operación que solicita.
- Procesador de comandos. Una vez que el sistema ha comprobado los permisos del usuario, se pasa el control al procesador de comandos.
- Control de la integridad. Cuando una operación cambia los datos de la base de datos, este módulo debe comprobar que la operación a realizar satisface todas las restricciones de integridad necesarias.
- Optimizador de consultas. Este módulo determina la estrategia óptima para la ejecución de las consultas.
- Gestor de transacciones. Este módulo realiza el procesamiento de las transacciones.
- Planificador (scheduler). Este módulo es el responsable de asegurar que las operaciones que se realizan concurrentemente sobre la base de datos tienen lugar sin conflictos.
- Gestor de recuperación. Este módulo garantiza que la base de datos permanece en un estado consistente en caso de que se produzca algún fallo.
- Gestor de buffers. Este módulo es el responsable de transferir los datos entre memoria principal y los dispositivos de almacenamiento secundario. A este módulo también se le denomina gestor de datos.

d) Servicios

Codd, el creador del modelo relacional, ha establecido una lista con los ocho servicios que debe ofrecer todo SGBD.

1. Un SGBD debe proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. Esta es la función fundamental de un SGBD y por supuesto, el SGBD debe ocultar al usuario la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento).

2. Un SGBD debe proporcionar un catálogo en el que se almacenen las descripciones de los datos y que sea accesible por los usuarios. Este catálogo es lo que se denomina diccionario de datos y contiene información que describe los datos de la base de datos (metadatos). Normalmente, un diccionario de datos almacena:

☐ ☐ Nombre, tipo y tamaño de los datos.

- Nombre de las relaciones entre los datos.
- Restricciones de integridad sobre los datos.
- Nombre de los usuarios autorizados a acceder a la base de datos.

- Esquema externo, conceptual e interno, y correspondencia entre los esquemas.
- Estadísticas de utilización, tales como la frecuencia de las transacciones y el número de accesos realizados a los objetos de la base de datos.

Algunos de los beneficios que reporta el diccionario de datos son los siguientes:

La información sobre los datos se puede almacenar de un modo centralizado. Esto ayuda a mantener el control sobre los datos, como un recurso que son.

- El significado de los datos se puede definir, lo que ayudará a los usuarios a entender el propósito de los mismos.
 - La comunicación se simplifica ya que se almacena el significado exacto. El diccionario de datos también puede identificar al usuario o usuarios que poseen los datos o que los acceden.
 - Las redundancias y las inconsistencias se pueden identificar más fácilmente ya que los datos están centralizados.
 - Se puede tener un historial de los cambios realizados sobre la base de datos.
 - El impacto que puede producir un cambio se puede determinar antes de que sea implementado, ya que el diccionario de datos mantiene información sobre cada tipo de dato, todas sus relaciones y todos sus usuarios.
 - Se puede hacer respetar la seguridad.
 - Se puede garantizar la integridad.
 - Se puede proporcionar información para auditorías.
3. Un SGBD debe proporcionar un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una transacción es un conjunto de acciones que cambian el contenido de la base de datos. Una transacción en el sistema informático de la empresa inmobiliaria sería dar de alta a un empleado o eliminar un inmueble. Una transacción un poco más complicada sería eliminar un empleado y reasignar sus inmuebles a otro empleado. En este caso hay que realizar varios cambios sobre la base de datos. Si la transacción falla durante su realización, por ejemplo porque falla el hardware, la base de datos quedará en un estado inconsistente. Algunos de los cambios se habrán hecho y otros no, por lo tanto, los cambios realizados deberán ser deshechos para devolver la base de datos a un estado consistente.
4. Un SGBD debe proporcionar un mecanismo que asegure que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente. Uno de los principales objetivos de los SGBD es el permitir que varios usuarios tengan acceso concurrente a los **datos que comparten**. El acceso

concurrente es relativamente fácil de gestionar si todos los usuarios se dedican a leer datos, ya que no pueden interferir unos con otros. Sin embargo, cuando dos o más usuarios están accediendo a la base de datos y al menos uno de ellos está actualizando datos, pueden interferir de modo que se produzcan inconsistencias en la base de datos. El SGBD se debe encargar de que estas interferencias no se produzcan en el acceso simultáneo.

5. Un SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos en caso de que ocurra algún suceso que la dañe. Como se ha comentado antes, cuando el sistema falla en medio de una transacción, la base de datos se debe devolver a un estado consistente. Este fallo puede ser a causa de un fallo en algún dispositivo hardware o un error del software, que hagan que el SGBD aborte, o puede ser a causa de que el usuario detecte un error durante la transacción y la aborte antes de que finalice. En todos estos casos, el SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos llevándola a un estado consistente.
6. Un SGBD debe proporcionar un mecanismo que garantice que sólo los usuarios autorizados pueden acceder a la base de datos. La protección debe ser contra accesos no autorizados, tanto intencionados como accidentales.
7. Un SGBD debe ser capaz de integrarse con algún software de comunicación. Muchos usuarios acceden a la base de datos desde terminales. En ocasiones estos terminales se encuentran conectados directamente a la máquina sobre la que funciona el SGBD. En otras ocasiones los terminales están en lugares remotos, por lo que la comunicación con la máquina que alberga al SGBD se debe hacer a través de una red. En cualquiera de los dos casos, el SGBD recibe peticiones en forma de mensajes y responde de modo similar. Todas estas transmisiones de mensajes las maneja el gestor de comunicaciones de datos. Aunque este gestor no forma parte del SGBD, es necesario que el SGBD se pueda integrar con él para que el sistema sea comercialmente viable.
8. Un SGBD debe proporcionar los medios necesarios para garantizar que tanto los datos de la base de datos, como los cambios que se realizan sobre estos datos, sigan ciertas reglas. La integridad de la base de datos requiere la validez y consistencia de los datos almacenados. Se puede considerar como otro modo de proteger la base de datos, pero además de tener que ver con la seguridad, tiene otras implicaciones. La integridad se ocupa de la calidad de los datos. Normalmente se expresa mediante restricciones, que son una serie de reglas que la base de datos no puede violar. Por ejemplo, se puede establecer la restricción de que cada empleado no puede tener asignados más de diez inmuebles. En este

caso sería deseable que el SGBD controlara que no se sobrepase este límite cada vez que se asigne un inmueble a un empleado.

Además, de estos ocho servicios, es razonable esperar que los SGBD proporcionen un par de servicios más:

1. Un SGBD debe permitir que se mantenga la independencia entre los programas y la estructura de la base de datos. La independencia de datos se alcanza mediante las vistas o subesquemas. La independencia de datos física es más fácil de alcanzar, de hecho hay varios tipos de cambios que se pueden realizar sobre la estructura física de la base de datos sin afectar a las vistas. Sin embargo, lograr una completa independencia de datos lógica es más difícil. Añadir una nueva entidad, un atributo o una relación puede ser sencillo, pero no es tan sencillo eliminarlos.
2. Un SGBD debe proporcionar una serie de herramientas que permitan administrar la base de datos de modo efectivo. Algunas herramientas trabajan a nivel externo, por lo que habrán sido producidas por el administrador de la base de datos. Las herramientas que trabajan a nivel interno deben ser proporcionadas por el distribuidor del SGBD. Algunas de ellas son:
 - Herramientas para importar y exportar datos.
 - Herramientas para monitorizar el uso y el funcionamiento de la base de datos.
 - Programas de análisis estadístico para examinar las prestaciones o las estadísticas de utilización.
 - Herramientas para reorganización de índices.
 - Herramientas para aprovechar el espacio dejado en el almacenamiento físico por los registros borrados y que consoliden el espacio liberado para reutilizarlo cuando sea necesario.

e) Usuarios

Los gestores de bases de datos, tienen para manejar diferentes tipos de usuario, de acuerdo rol que ejerce dentro de la organización, es así, como podemos hacer una clasificación de los tipos de usuarios de una base de datos en función de la forma en que interaccionan con el sistema.

- Administradores de la base de datos. Persona que tiene centralizado el control del sistema.

- Programadores de aplicaciones. Se trata de los profesionales que interactúan con el sistema a través de llamadas en DML, las cuales están incorporadas en un lenguaje anfitrión. A estos programas se les denominan programas de aplicación, como por ejemplo, los programas para la generación de cargos, abonos, transferencias de un sistema bancario. Como la sintaxis DML suele ser diferente de la sintaxis del lenguaje anfitrión, las llamadas en DML suelen ir precedidas de un carácter especial, de forma que se genere el código apropiado en el lenguaje anfitrión, lo cual se hace mediante un precompilador de DML, que convierte las sentencias DML en sentencias del lenguaje anfitrión. Una vez precompilado el programa, se compilaría mediante el compilador del lenguaje anfitrión, que generaría el código objeto apropiado.
- Usuarios sofisticados. Son los que interactúan con el sistema sin escribir programas, escribiendo las consultas en el lenguaje de consulta de la base de datos.
- Usuarios especializados. Se trata de usuarios sofisticados que crean aplicaciones de bases de datos especializadas para el procesamiento de la información.
- Usuarios ingenuos. Son los usuarios que interactúan con el sistema llamando a uno de los programas desarrollados por los programadores de aplicaciones.

Como primer tipo de usuario hemos descrito la figura del administrador, un usuario vital en el enfoque de bases de datos, y que tiene unas funciones que merecen ser estudiadas más detalladamente. Estas son:

- Definición del esquema conceptual. El esquema original de la base de datos se crea escribiendo un conjunto de definiciones que son traducidas por el compilador de DDL a un conjunto de metadatos que se guardan en el diccionario de datos.
- Definición del esquema físico. Se trata de definir las estructuras de almacenamiento y los métodos de acceso adecuados (especificación de los tipos de índices).
- Modificación del esquema y de la organización física. Si bien las modificaciones tanto del esquema de la base de datos como de la organización física no son demasiado habituales, éstas se realizan modificando el esquema conceptual y físico.
- Creación de permisos para el acceso a los datos. El administrador de la base de datos es el encargado de definir los permisos que autorizan a los usuarios a acceder a ciertas partes de la base de datos.
- Especificación de las restricciones de integridad. Estas restricciones se guardan en el diccionario de datos para ser consultado cada vez que se realice una actualización.

Lección 4: “Arquitectura de bases de datos”

Hay tres características importantes inherentes a los sistemas de bases de datos: la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos. En 1975, el comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles para los sistemas de bases de datos, que resulta muy útil a la hora de conseguir estas tres características.

El objetivo de la arquitectura de tres niveles es el de separar los programas de aplicación, de la base de datos física. En esta arquitectura, el esquema de una base de datos se define en tres niveles de abstracción distintos:

1. En el nivel interno se describe la estructura física de la base de datos mediante un esquema interno. Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.
2. En el nivel conceptual se describe la estructura de toda la base de datos para una comunidad de usuarios (todos los de una empresa u organización), mediante un esquema conceptual. Este esquema oculta los detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema.
3. En el nivel externo se describen varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinados y oculta a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.

La mayoría de los SGBD no distinguen del todo los tres niveles. Algunos incluyen detalles del nivel físico en el esquema conceptual. En casi todos los SGBD que se manejan vistas de usuario, los esquemas externos se especifican con el mismo modelo de datos que describe la información a nivel conceptual, aunque en algunos se pueden utilizar diferentes modelos de datos en el niveles conceptual y externos.

Hay que destacar que los tres esquemas no son más que descripciones de los mismos datos pero con distintos niveles de abstracción. Los únicos datos que existen realmente están a nivel físico, almacenados en un dispositivo como puede ser un disco. En un SGBD basado en la arquitectura de tres niveles, cada grupo de usuarios hace referencia exclusivamente a su propio esquema externo. Por lo tanto, el SGBD debe transformar cualquier petición expresada en términos de un

esquema externo a una petición expresada en términos del esquema conceptual, y luego, a una petición en el esquema interno, que se procesará sobre la base de datos almacenada. Si la petición es de una obtención (consulta) de datos, será preciso modificar el formato de la información extraída de la base de datos almacenada, para que coincida con la vista externa del usuario. El proceso de transformar peticiones y resultados de un nivel a otro se denomina correspondencia o transformación. Estas correspondencias pueden requerir bastante tiempo, por lo que algunos SGBD no cuentan con vistas externas.

La arquitectura de tres niveles es útil para explicar el concepto de independencia de datos que podemos definir como la capacidad para modificar el esquema en un nivel del sistema sin tener que modificar el esquema del nivel inmediato superior. Se pueden definir dos tipos de independencia de datos:

- La independencia lógica es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el esquema conceptual para ampliar la base de datos o para reducirla. Si, por ejemplo, se reduce la base de datos eliminando una entidad, los esquemas externos que no se refieran a ella no deberán verse afectados.
- La independencia física es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos). Por ejemplo, puede ser necesario reorganizar ciertos ficheros físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización de datos. Dado que la independencia física se refiere sólo a la separación entre las aplicaciones y las estructuras físicas de almacenamiento, es más fácil de conseguir que la independencia lógica.

En los SGBD que tienen la arquitectura de varios niveles es necesario ampliar el catálogo o diccionario, de modo que incluya información sobre cómo establecer la correspondencia entre las peticiones de los usuarios y los datos, entre los diversos niveles. El SGBD utiliza una serie de procedimientos adicionales para realizar estas correspondencias haciendo referencia a la información de correspondencia que se encuentra en el catálogo. La independencia de datos se consigue porque al modificarse el esquema en algún nivel, el esquema del nivel inmediato superior permanece sin cambios, sólo se modifica la correspondencia entre los dos niveles. No es preciso modificar los programas de aplicación que hacen referencia al esquema del nivel superior. Por lo tanto, la arquitectura de tres niveles puede facilitar la obtención de la verdadera independencia de datos, tanto física como lógica. Sin embargo, los dos niveles de correspondencia implican un gasto extra durante la ejecución de una consulta o de un programa, lo cual reduce la eficiencia del SGBD. Es por esto que muy pocos SGBD han implementado esta arquitectura completa.

Lección 5: “Planificación de proyectos de análisis, diseño e implementación de bases de datos”

Para la planificación de un proyecto de análisis, diseño e implementación de bases de datos, es necesario tener claro el área de la organización que se desea estudiar, pues éstas deben tener un contexto delimitado.

Una vez definido lo anterior, se debe planificar todo lo concerniente al levantamiento de información, pues en ésta, la participación del usuario es de vital importancia; lo mismo que la documentación de políticas y normas que tiene la organización, en el área bajo estudio.

Con respecto al análisis, este se realizará con la técnica Entidad Relación, con base al levantamiento de información. El diagrama resultante, debe estar avalado por el área usuaria, que en última instancia es el modelo lógico de datos.

En cuanto al diseño físico, es el resultado de pasar el modelo lógico de datos a tablas, con sus respectivos llaves principales y llaves foráneas.

Por último, la implementación del diseño físico se realizará en la herramienta MySQL (SGBD).

a) Levantamiento de información

Para desarrollar esta fase del proyecto, se recomienda que se recolecte la información a través de entrevistas y recolección de formatos que se manejen en el área.

- Para el caso de las entrevistas, se recomienda seguir los siguientes pasos:
 - Seleccionar los diferentes procesos que tiene el área bajo estudio.
 - Seleccionar las personas que mejor conocen los anteriores procesos.
 - Realizar con las personas anteriores, el cronograma para la realización de la entrevista.
 - Realizar la entrevista, ésta no debe durar más de dos horas, si esto ocurre, se debe realizar en otra sesión.
 - Editar la entrevista. Por lo general, la edición toma el doble de tiempo que duró la entrevista. Aquí, se debe quitar todos los saludos, palabras o frases que no tenga que ver con el tema.
 - Validación de la entrevista: Aquí, se envía al usuario el texto de la entrevista, con el fin de que verifique que lo que está escrito es lo correcto. Por lo general, los usuarios quitan y colocan cosas nuevas, esto debido a que a veces desean incluir mejoras; de ahí, que la validación es otra entrevista, pero con base en el documento editado.

- Los pasos 5 y 6 se repiten, tantas veces como sea necesario, hasta que el usuario le de el visto bueno definitivo al documento.
 -
- Para el caso de los formatos, se recomienda describir cada uno de ellos y colocar en que procesos interviene.

b) Análisis de la información

Para proyecto de bases de datos, el análisis consiste en realizar el modelo de datos, con base en el levantamiento de información. Es así, como en el texto se deben identificarse los conjuntos de datos, las relaciones y los atributos.

Una vez identificado y realizado el diagrama, este debe validarse con los usuarios del área.

c) Modelo Relacional

Con base al modelo lógico de datos, se aplican las reglas para pasar al modelo relacional o diseño físico, en otras palabras, las entidades, relaciones y atributos se pasan a tablas y campos.

d) Implementación

Una vez se tenga definida todas las tablas, con sus respectivos campos, llaves principales y llaves foráneas, se entra a definir en la herramienta de MySQL. Lo anterior se realiza con las sentencias DDL (Lenguaje de Definición de datos).

CAPÍTULO II: “FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE BASES DE DATOS RELACIONAL”

Lección 6: “Concepto de Modelo de Datos”

a) Definición de modelo y modelo de datos

Se entiende por modelo, el proceso de abstraer una realidad y tratar de modelarla a través de un gráfico, fórmula matemática, prototipo, etc. Para nuestro caso, entenderemos como modelo de datos, la abstracción del mundo real de la organización en término de datos y sus relaciones, plasmada en un gráfico llamado diagrama entidad-relación. Este diagrama, es un técnica de modelado conceptual (lógico) de datos, cuya bases teórica se encuentran en la teoría de conjuntos, relaciones y funciones de la ciencia de las matemáticas.

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información.

Con respecto al diseño de bases de datos, el modelado de datos puede ser escrito así (Brodie 1984:20): "dados los requerimientos de información y proceso e una aplicación de uso intensivo de datos (por ejemplo, un sistema de información), construir una representación de la aplicación que capture las propiedades estáticas y dinámicas requeridas para dar soporte a los procesos deseados (por ejemplo, transacciones y consultas). Además de capturar las necesidades dadas en el momento de la etapa de diseño, la representación debe ser capaz de dar cabida a eventuales futuros requerimientos".

Un modelo de datos es por tanto una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo. Conceptualmente, una aplicación puede ser caracterizada por:

- Propiedades estáticas: entidades (u objetos), propiedades (o atributos) de esas entidades, y relaciones entre esas entidades.
- Propiedades dinámicas: operaciones sobre entidades, sobre propiedades o relaciones entre operaciones.
- Reglas de integridad sobre las entidades y las operaciones (por ejemplo, transacciones).

Así, un modelo de datos se distingue de otro por el tratamiento que da a estas tres categorías. El resultado de un modelado de datos es una representación que tiene dos componentes: las propiedades estáticas se definen en un esquema y las

propiedades dinámicas se definen como especificaciones de transacciones, consultas e informes.

Un esquema consiste en una definición de todos los tipos de objetos de la aplicación, incluyendo sus atributos, relaciones y restricciones estáticas. Correspondientemente, existirá una reposición de información, la base de datos, que es una instancia del esquema. Un determinado tipo de procesos sólo necesita acceder a un subconjunto predeterminado de entidades definidas en un esquema, por lo que este tipo de procesos puede requerir sólo un subconjunto de las propiedades estáticas del esquema general. A este subconjunto de propiedades estáticas se le denomina sub esquema.

Una transacción consiste en diversas operaciones o acciones sobre las entidades de esquema o sub esquema. Una consulta se puede expresar como una expresión lógica sobre los objetos y relaciones definidos en el esquema; una consulta identifica un subconjunto de la base de datos. Las herramientas que se usan para realizar las operaciones de definición de las propiedades estáticas y dinámicas de la base de datos son los lenguajes de definición y manipulación de datos (DDL, DML), junto con los lenguajes de consulta (SQL).

- Modelo Entidad - Relación

El modelo E-R (Entidad-Relación) es un modelo de datos conceptual de alto nivel y que se suele utilizar bastante en el diseño de bases de datos Relacional. Se basa en una percepción del mundo real que consiste en un conjunto de objetos básicos denominados entidades y relaciones, y se desarrolló para facilitar el diseño de bases de datos.

El modelo E-R crea un modelo de la realidad que se asimila a la realidad que queremos modelar, y lo hace de forma que es independiente de la implementación posterior, ofreciendo un alto nivel de abstracción, y siendo una herramienta gráfica fácil de comprender.

El resultado del modelado E-R es un diagrama E-R que representa una estructura lógica general de la base de datos.

- Definición de Entidad y notación

Es una abstracción de un conjunto de cosas (objetos) del mundo real, las cuales tienen las mismas características y están sujeta a las mismas reglas. Una entidad válida, debe ser significativa para el alcance del análisis, debe tener más de una ocurrencia y cada ocurrencia debe ser ÚNICA e identificable. Ejemplos de entidad

es un conjunto de personas, vehículos, computadores, oficinas, facturas, créditos, etc. cabe resaltar que muchos de estos ejemplos con conjunto arbitrarios, solo creados por la imaginación de la mente humana, con el fin de organizar por categorías las cosas que lo rodean y poder tener mejor comprensión del mundo.

- Tipos de entidades:

Entidad fuerte o fundamental: es una entidad que se identifica por si sola, es decir, una o varias características (atributos) le garantiza UNICIDAD, por consiguiente no depende de otra entidad o entidades. Gráficamente, tenemos:



Figura 9 – Entidad fuerte

Entidad débil: es una entidad que no se puede identificar por si sola, es decir, no existe una o varias características de la entidad que le pueda garantizar UNICIDAD, por lo tanto, depende de otra entidad y su dependencia es fuerte. Es de anotar, que en estas entidades como mínimo, una de sus características (atributos), hace parte de su CLAVE PRIMARIA. Gráficamente, tenemos a la entidad Historia del empleado.

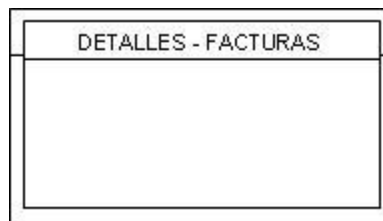


Figura 10 – Entidad débil

Entidad asociativa: es una entidad débil, pero, depende de DOS o mas entidades, con el fin de garantizar unicidad. Es de anotar, que esta es la única entidad que puede o no tener características propias (atributos). Gráficamente, tenemos a la entidad Detalles de Facturas.



Figura 11- Entidad asociativa

- Definición de Atributo

Es una abstracción de las características que poseen todas las instancias u ocurrencias de una entidad. Estas características deben cumplir con ciertas propiedades:

- **Incluir toda la información necesaria sobre la entidad.**
- **No deben colocarse atributos de otra entidad.**
- **Cada atributo debe referirse a una sola característica de la entidad.**
- **Cada atributo debe tomar sus valores independiente a los demás.**

Dominio: es un conjunto de valores que un atributo puede tomar. Para establecer los dominios se debe tener unas reglas de aceptación por parte de los responsables de la información y además citar los documentos que sustente las reglas acordadas. Estos pueden ser de diferentes tipos.

- **Listas: cuando se requiere que un atributo tome valores que no estan comprendidos en un rango.**
- **Rango: son los valores aceptables en un intervalo continuo.**

Atributo llave: es el conjunto de uno o más atributos, los cuales distinguen como única cada instancia u ocurrencia de una entidad.

- Definición de Relación y notación

Es la asociación entre dos o mas instancias del mismo o diferente tipo de entidad. Éstas, son relaciones simétricas, es decir, en doble sentido, de tal forma, que si una o varias instancias u ocurrencia de la entidad **A**, esta relacionada, con una o varias ocurrencias de la entidad **B**, también una o varias instancias u ocurrencia de la entidad **B**, esta relacionada, con una o varias ocurrencias de la entidad **A**. Por otro lado, pueden existir relaciones entre una o varias instancias u ocurrencia de la entidad **A**, con una o varias instancias u ocurrencia de la entidad **A**, es decir, con ella misma, lo que dá un subconjunto del conjunto de **A**, lo mismo puede ocurrir con **B**.

Para establecer relaciones, la forma como se hace es:

- a. ¿Cómo están las instancias u ocurrencias de la entidad "A" relacionada con la entidad "B"?
- b. Se deben utilizar frases verbales significativas para el negocio.
- c. Colocar nombre a las relaciones en ambos sentidos.

Muchas veces para no complejizar el diagrama, los nombre en las relaciones no se colocan, sin embargo se hace indispensable cuando existen dos o más relaciones entre las entidades ya sea la misma (A con A) ó diferentes (A con B).

- **Definición de Cardinalidad y notación**

Es el establecimiento de número de instancias u ocurrencias que se relacionan. Para el establecimiento de la cardinalidad en un sentido, obsérvese en la gráfica, que siempre parte del mínimo y llega al máximo. Veamos ahora, las cuatro formas cardinales que tienen las relaciones:

1) Relación Cero a Uno (0:1): Es la relación donde para Cero instancias u ocurrencias de A, hay máximo Una instancia u ocurrencia de B. La interpretación que tiene la relación con esta cardinalidad, es que puede existir instancias u ocurrencias en B, sin necesidad de que exista instancias y ocurrencias en A.

2) Relación Uno a Uno (1:1): Es la relación donde para Una instancias u ocurrencias de A, hay máximo Una instancia u ocurrencia de B. La interpretación que tiene la relación con esta cardinalidad, es que para que exista una instancias u ocurrencias en B, hay necesidad de que exista instancias y ocurrencias en A.

3) Relación Cero a Muchos (0:N): Es la relación donde para Cero instancias u ocurrencias de A, hay máximo Muchas instancia u ocurrencia de B. La interpretación que tiene la relación con esta cardinalidad, es que puede existir una o varias instancias u ocurrencias en B, sin necesidad de que existan instancias y ocurrencias en A.

4) Relación Uno a Mucho (1:N): Es la relación donde para Una instancias u ocurrencias de A, hay máximo Muchas instancia u ocurrencia de B. La interpretación que tiene la relación con esta cardinalidad, es que, para que exista una o varias instancias u ocurrencias en B, hay necesidad de que existan una instancias y ocurrencias en A.

Es importante aclarar, que la existencia de los mínimos, dependen de las políticas y normas que tiene la empresa, así por ejemplo, si la empresa determina que para incluir un empleado , debe existir el cargo para esto,

entonces se hace necesario que la relación de de cargo hacia empleados, como mínimo debe existir un cargo para poder inscribir empleados (1 - N).

Ahora miremos en forma gráfica como queda:

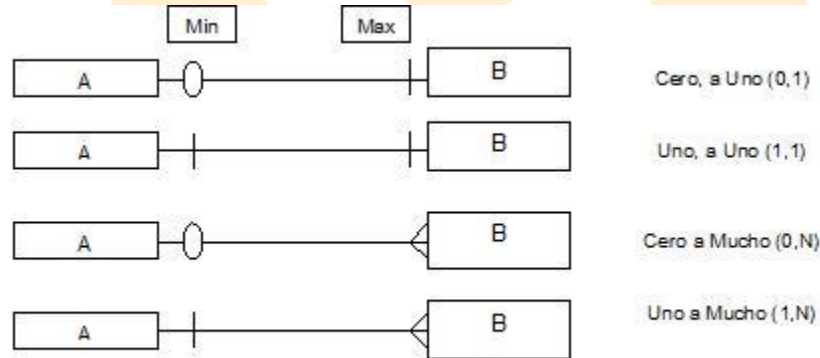


Figura 12 – Tipos de cardinalidad

Lecturas complementarias:

[Sistema de Bases de Datos: Diseño, Implementación y administración](#)

Lección 7: “Procedimiento para elaborar un Modelo”

a) Uso de la matriz de relación

Cuando se está iniciando a modelar conceptualmente los datos, una manera de realizarlo es a través de la técnica de la **Matriz de Relación**; esta técnica se basa en las teoría de conjunto, más específicamente, los conceptos de conjuntos y relaciones. Es así, que para empezar a trabajar una problemática (caso), se hace necesario que el estudiantes determine primero los conjuntos de datos (Entidades), las relaciones que existen entre los elementos (Instancias u ocurrencias) de los conjuntos de datos y el tipo de relación (Cardinalidad).

Con base en lo anterior, los pasos que se deben seguir para llenar la matriz de relación son los siguientes:

- Identificar en la problemática los sustantivos en plural (Conjuntos o Entidades).
- Colocar en las filas todas las entidades que encontró en el paso anterior.
- Colocar en las columnas todas las entidades, pero en el mismo orden que las colocó en las filas. Vale la pena aclarar, que esta organización es muy

importante, pues la diagonal, siempre debe ser la intersección entre los mismas entidades o conjunto.

- Llenar la diagonal, de tal forma, que se analice si existen relaciones entre las instancias u ocurrencias de las misma entidad (Subconjuntos); Sí no es así, entonces se coloca una **X**, si es así, entonces se entra a establecer el tipo de relación (Cardinalidad). Es bueno observar en el caso, que los verbos me identifican la relación y el tipo de relación que existen entre las instancias u ocurrencias de una o diferentes entidades.

Se empieza a llenar la matriz desde la primera fila, de izquierda a derechas, donde lo primero que se debe mirar es sí, existe o no relación; si existe, entonces se entra a determinar el tipo de relación, donde siempre de parte del **Mínimo** (Entidad Fila) y se llega al **Máximo** (Entidad Columna). Esta operación se continúa hasta llegar a la última fila. Se debe aclarar, que a partir del llenado de la segunda fila en adelante, las casillas que están trabajando y se encuentran debajo de la diagonal, el establecimiento de la relación se dá, sí y solo sí, la casilla contraria a la diagonal está llena. Lo anterior, debido a que las relaciones son en ambos sentidos, es decir simétricas; ya lo que resta es identificar el tipo de relación, es decir, la Cardinalidad. De modo gráfico se vería así:

ENTIDAD	Departamento	Cargo	Empleado
Departamento	X	X	1-N
Cargo	X	X	1-N
Empleado	1-1	1-1	X

Figura 13 – Matriz relación caso simple empleados

b) Diagrama Entidad Relación

Para realizar un diagrama entidad relación, partiendo de la matriz de relación, se debe tener en cuenta los siguientes pasos:

- ❖ Determinar en la matriz de relación cuál es la entidad que tiene mas relaciones Uno a Uno, ó Cero a Uno; o cuál es el nombre de la entidad, que mas identifica el caso. La entidad encontrada, se recomienda colocarla en toda la mitad.
- ❖ Comenzar desde la primera fila y empezar a ver con que entidades está relacionada (Entidad Columna); estas entidades deben colocarse cerca a ella,

hasta donde sea posible, y se va trazando la línea entre ellas o ella misma (Relación). Esta operación se repite hasta llegar a la última fila.

- ❖ Colocar las cardinalidades, de tal forma, que se empieza desde la primera fila y se mira el tipo de relación que se tiene (siempre está en un sentido) y se coloca en la notación que ya se vio anteriormente (Cardinalidad). Esta operación se repite hasta llegar a la última fila.

Nota: Los dos primeros pasos, es una organización de trabajo recomendada, sin embargo, a medida que se avanza en la resolución de casos, mucho de este procedimiento se va cambiando, de acuerdo a la destreza que las personas van adquiriendo y en última instancia, llegan a obviar el paso de la matriz de relación. Por ahora es recomendable, que para iniciar con el proceso de modelado, lo realicen desde la técnica de la matriz de relación.

- ❖ Colocar los atributos a cada una de las entidades, para esto, se recomienda leer de nuevo el caso e identificar los sustantivos en singular, es decir, una característica que ayude a identificar mas la entidad. ¿Cómo se sabe qué es una característica? Lo sabemos, porque éstas nos ayudan a conocer mejor acerca del objeto o sujeto que se está hablando (Entidad), de tal forma que muchas veces NO nos dicen que es, pero dada las características que nos dan, podemos categorizar, en última instancia podemos saber de que se está hablando.
- ❖ Establecer el atributo clave para cada entidad; se debe recordar, que éste puede tener uno o varios atributos de la entidad, de tal forma que le garantice unicidad, es decir, el atributo clave hace que cada instancia u ocurrencia de la entidad sean únicas e irrepetibles (principio de teoría de conjunto: Para que un conjunto sea válido, no debe haber elementos repetidos).
- ❖ Establecer el tipo de entidad, si es fuerte, débil o asociativa. Recordemos que una entidad fuerte es aquella, donde el atributo clave le garantiza unicidad; la entidad débil, es aquella que el atributo clave no le garantiza unicidad y para ello, debe apoyarse con otra entidad; lo mismo sucede con la asociativa, pero se apoya en dos o más entidades para garantizar unicidad.

Es pertinente aclarar dos cosas:

- a. La entidad debil siempre debe tener como mínimo un atributo clave.
- b. La entidad asociativa, es la única entidad que puede no tener atributos y/o atributo clave.

Lección 8: “Desarrollo Caso de Empleados”

- Enunciado Caso empleados - Caso Ejemplo

Para iniciar a modelar los datos, miremos el siguiente caso de empleados:

Se quiere crear una base de datos de personal para una compañía, con las siguientes características:

- **La compañía tiene un conjunto de departamentos.**
- **Cada departamento tiene un conjunto de empleados, un conjunto de proyectos y un conjunto de oficinas. Los proyectos tienen empleados asignados, y las oficinas empleados que laboran en ellas.**
- **Los cargos están codificados, de tal forma, que tienen un código único y una descripción del cargo.**
- **Cada empleado tiene una historia de cargos ocupados, en la cual se requiere saber la fecha en que ha ocupado éstos.**

La base de datos debe contener la siguiente información:

- **Por cada departamento, número de departamento (único), nombre, presupuesto anual.**
- **Por cada empleado, número de empleado (único), nombre, dirección y teléfono de residencia, departamento y proyecto al que esta asignado actualmente, número de oficina y cargo actual.**
- **Por cada proyecto, número de proyecto (único), nombre, presupuesto del proyecto y departamento al que pertenece.**
- **Por cada oficina, número de oficina (único), área en metros cuadrados, números de teléfono.**

Además se sabe lo siguiente:

- **Un empleado pertenece a máximo un departamento al tiempo.**
- **Un empleado está vinculado a máximo un proyecto al tiempo.**
- **Un empleado tiene máximo una oficina al tiempo.**
- **Un empleado desempeña máximo un cargo al tiempo.**
- **Una oficina pertenece a máximo un departamento al tiempo.**

Identificación de entidades y Matriz de relación

Observando el caso de empleados, encontramos los siguientes sustantivos (Entidades):

Departamentos, Empleados, Proyectos, Oficinas, Cargos e Historia de Cargos.

ENTIDAD	Dptos.	Empl.	Proy.	Ofic.	Cargos	Hist_Carg.
Dptos.	x	1-N	1-N	1-N	x	x
Empl.	1-1	x	1-1	1-1	1-1	1-N
Proy.	1-1	1-N	x	x	x	x
Ofic.	1-1	1-N	x	x	x	x
Cargos	x	1-N	x	x	x	1-N
Hist_Carg.	x	1-1	x	x	1-1	x

Figura 14 – Matriz relación caso empleados

En la matriz anterior, observemos las relaciones y cardinalidades de Historias de Cargos. Lo primero es que solo está relacionada con las entidades Empleados y Cargos. Miramos que un empleado puede tener como máximo varias historias de cargos, esto, porque una persona durante la vida laboral puede desempeñar varios cargos; y una historia de cargos le pertenece máximo a un empleado, esto debido a que cada vez que un empleado cambia de cargo, se genera una instancia u ocurrencia en historia de cargos. Por otro lado, con respecto a la relación que tiene con cargos, una Historia de Cargo, tiene máximo un Cargo, y un Cargo, puede encontrarse en varias instancias u ocurrencias de Historia de Cargos, esto debido a que pueden haber varios empleados con el mismo cargo.

Con respecto a las demás relaciones, son claras sus relaciones, de acuerdo al enunciado del caso.

- Paso matriz a diagrama E-R

Una vez que se tenga finalizada la Matriz de Relación, se procede de la siguiente forma:

- 1) Colocar en el centro la entidad que tiene mas relaciones 1-1 o la que mejor identifique el problema. Vemos que en la fila de la entidad EMPLEADOS, se tiene cuatro(4) relaciones 1-1, y una 1-N; las demás entidades tienen una(1) o máximo dos(2) y por lo tanto empleados debe ser la entidad central.
- 2) Continuar con las demás entidades, partiendo de la primera fila. Esta entidad, es la de DEPARTAMENTOS, vemos que tiene relación con las entidades

EMPLEADOS y OFICINAS; como la primera entidad que llevamos en el diagrama es empleados, luego colocamos en el extremo superior izquierdo la entidad de DEPARTAMENTOS. Seguimos con la entidad de PROYECTOS, y encontramos que esta tiene relación solo con DEPARTAMENTOS y EMPLEADOS, luego la colocamos al lado izquierdo de EMPLEADOS y abajo de DEPARTAMENTO. Luego encontramos la entidad de OFICINAS, esta solo está relacionada con DEPARTAMENTOS Y EMPLEADOS, por lo tanto la colocamos en el extremo superior derecho, cerca de éstas dos entidades. Seguimos con cargos, y encontramos que esta relacionada con EMPLEADOS e HISTORIA DE CARGOS, entonces la colocamos al lado derecho de EMPLEADOS. Por último, encontramos la entidad de HISTORIA DE CARGOS, la cual esta relacionada con EMPLEADOS y CARGOS, por lo tanto la colocamos debajo de estas dos entidades.

Todo el proceso anterior, es una recomendación de organización, con el fin de que cuando comencemos a establecer las relaciones, no nos queden líneas(Relaciones) cruzadas. Ver gráfico.

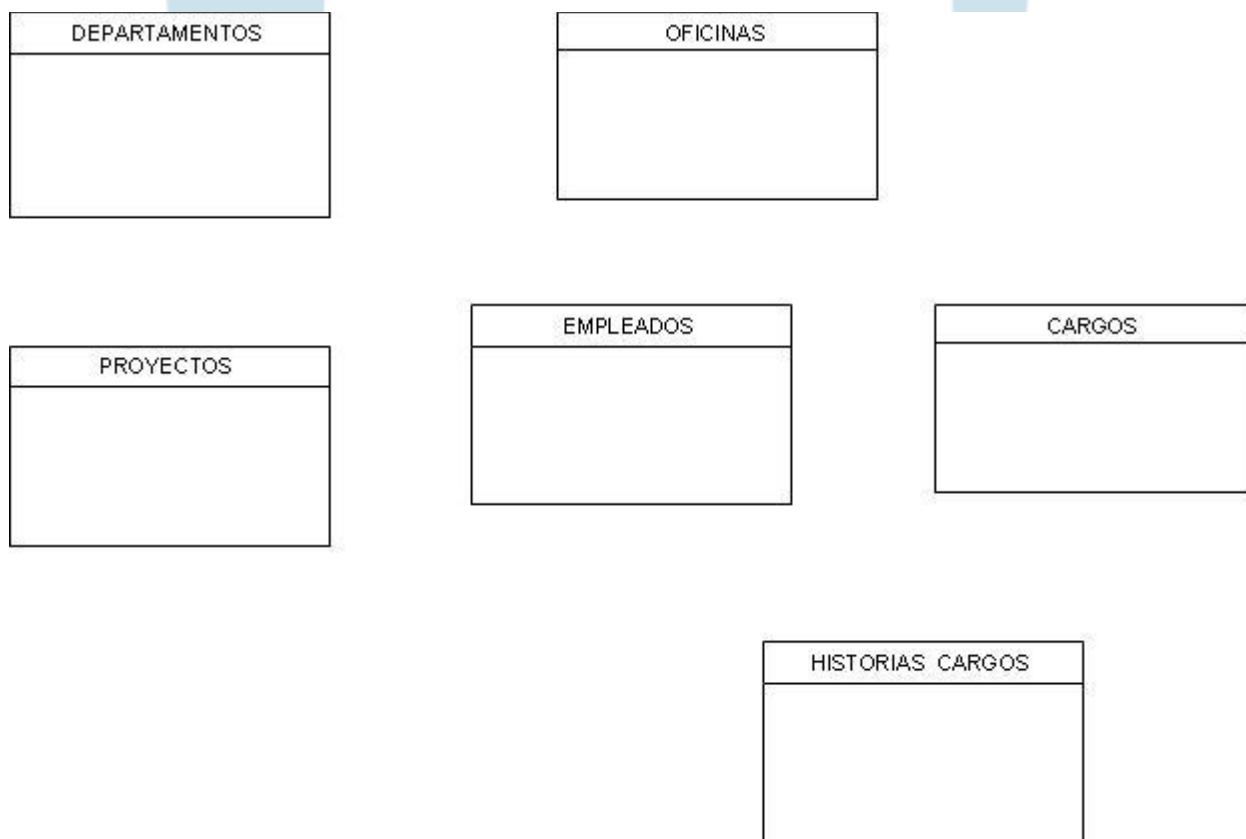


Figura 15 – DER- 1 caso empleados

3) Colocar las relaciones, partiendo de la primera fila, que es la entidad departamento; observamos que éste tiene relación con EMPLEADOS, PROYECTOS Y OFICINAS, por lo tanto, procedemos a pasar las respectivas líneas. Ver siguiente gráfico.

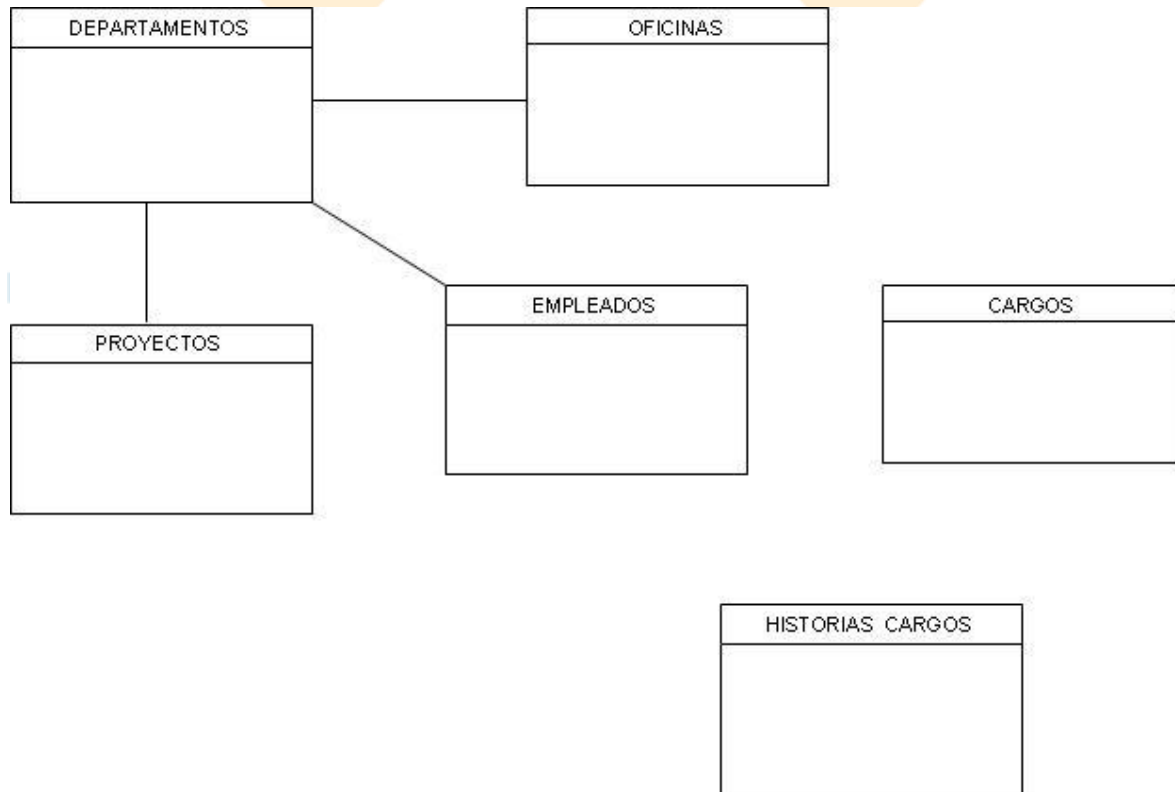


Figura 16 – DER- 2 caso empleados

Ahora procedemos a colocar la cardinalidad. Observamos que la primera casilla que encontramos es la de DEPARTAMENTOS con EMPLEADOS; para ellos se dice: Un DEPARTAMENTO, puede tener muchos EMPLEADOS. Lo mismo ocurre con PROYECTOS Y OFICINAS, un DEPARTAMENTO, puede tener muchos PROYECTOS y muchas OFICINAS. Nótese que siempre partimos del mínimo y llegamos al máximo. Veamos la gráfica siguiente:

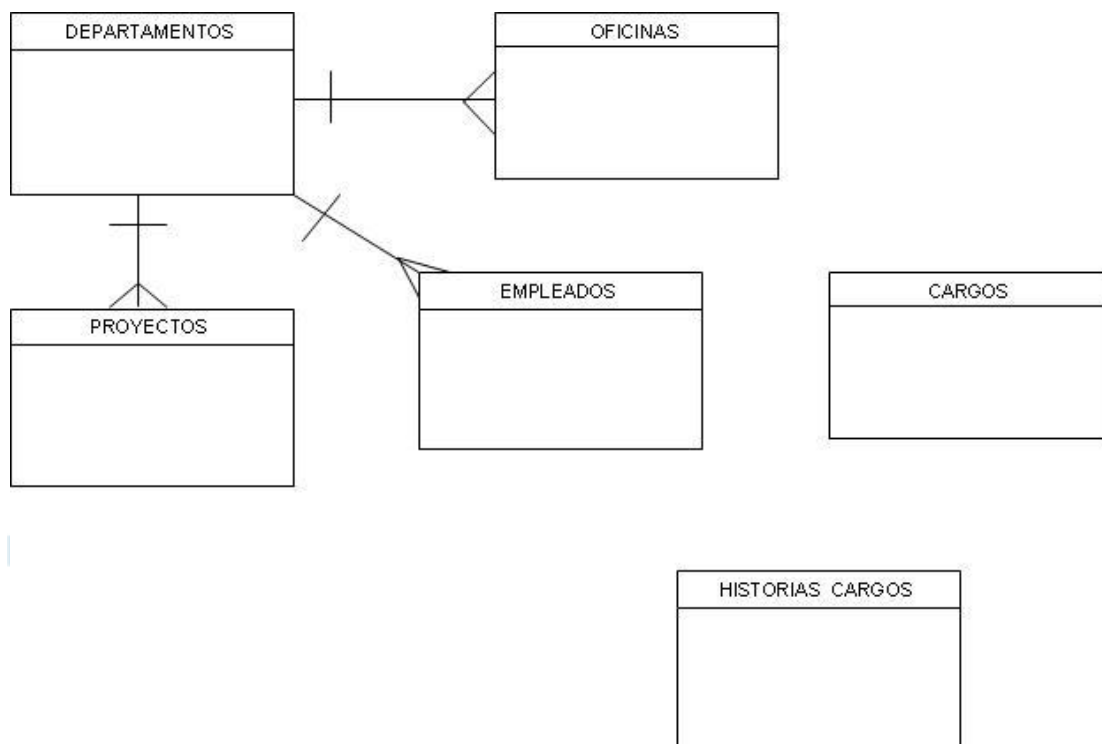


Figura 17 – DER- 3 caso empleados

El proceso anterior se sigue con las demás entidades que se encuentran en la fila, como son EMPLEADOS, PROYECTOS, OFICINAS, CARGOS e HISTORIAS DE CARGOS. Para lo cual tendríamos el siguiente diagrama:

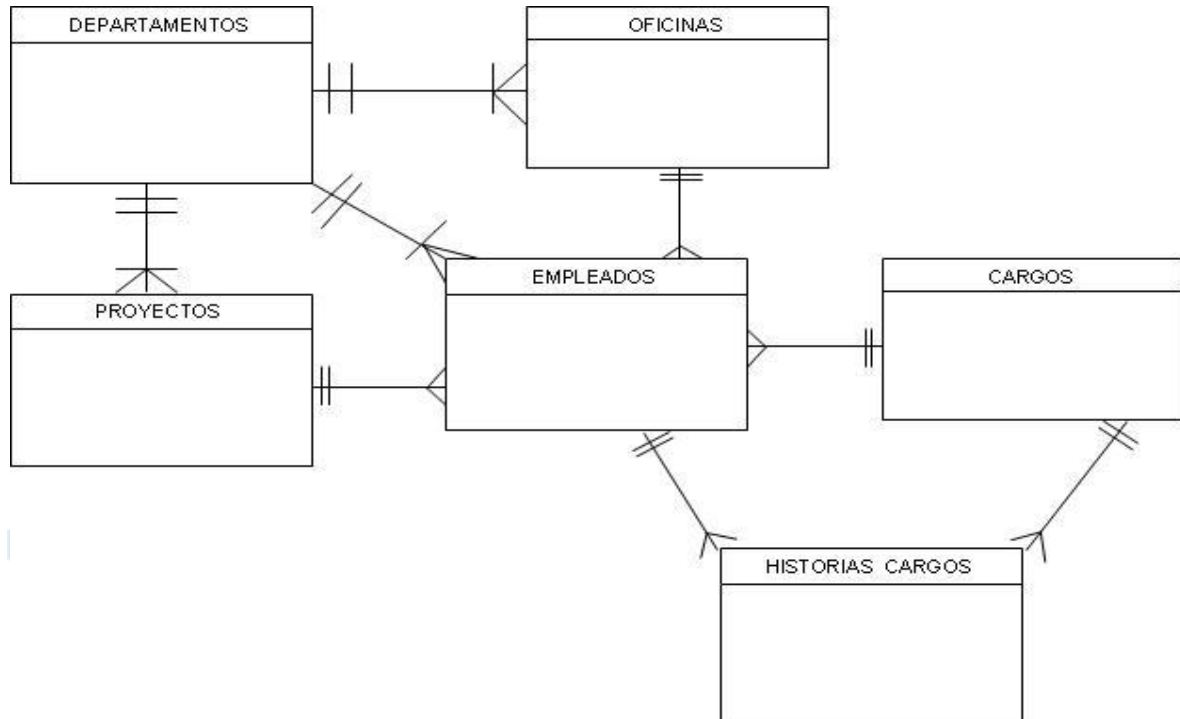


Figura 18 – DER- 4 caso empleados

- Selección de atributos y atributo clave

4) Colocar los atributos a cada una de las entidades; mirando el caso, vemos que los sustantivos en singular para cada entidad son los que aparecen en el siguiente Diagrama:

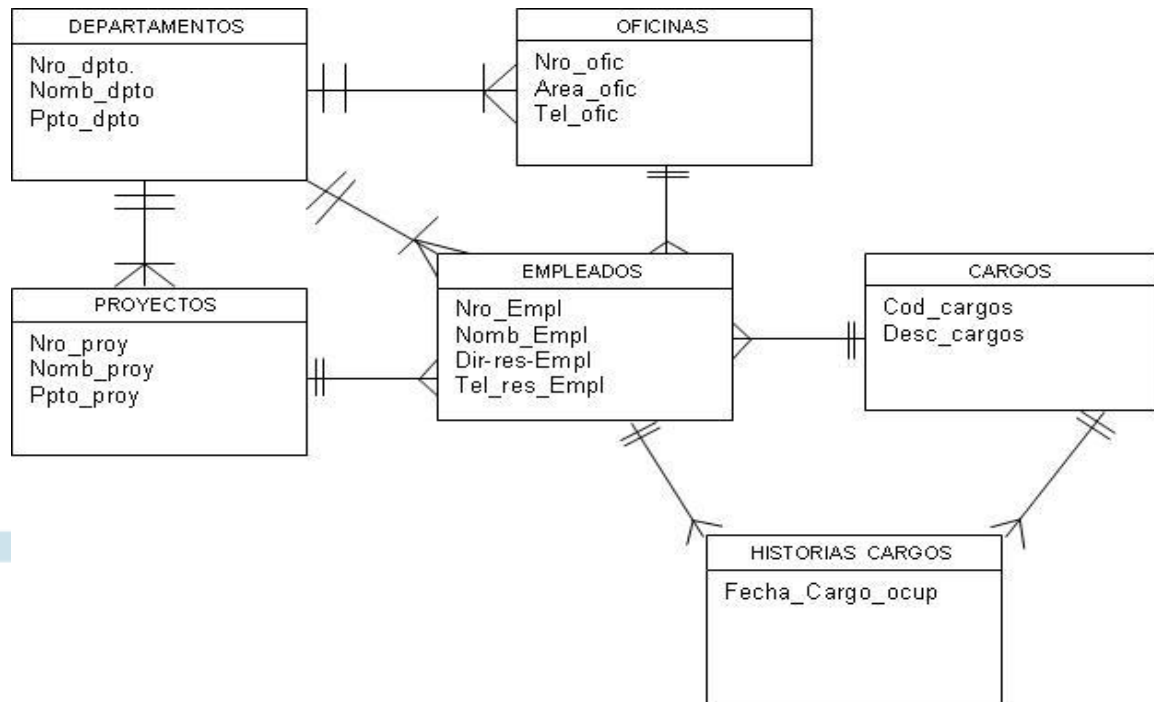


Figura 19 – DER- 5 caso empleados

5) Determinar el atributo clave. Si observamos en el caso y los atributos encontrados, observamos que en el caso, hay alguno de ellos que dicen que son únicos, es el caso de atributo número en DEPARTAMENTOS, EMPLEADOS, PROYECTOS y OFICINAS; para el caso de CARGOS, se ve el código, y por último, HISTORIAS DE SALARIOS solo tiene un atributo. En esta última entidad, el único atributo que puede ser clave es éste. Sin embargo analizaremos este caso para el siguiente paso. Teniendo en cuenta lo anterior, entonces nuestro diagrama quedaría así:

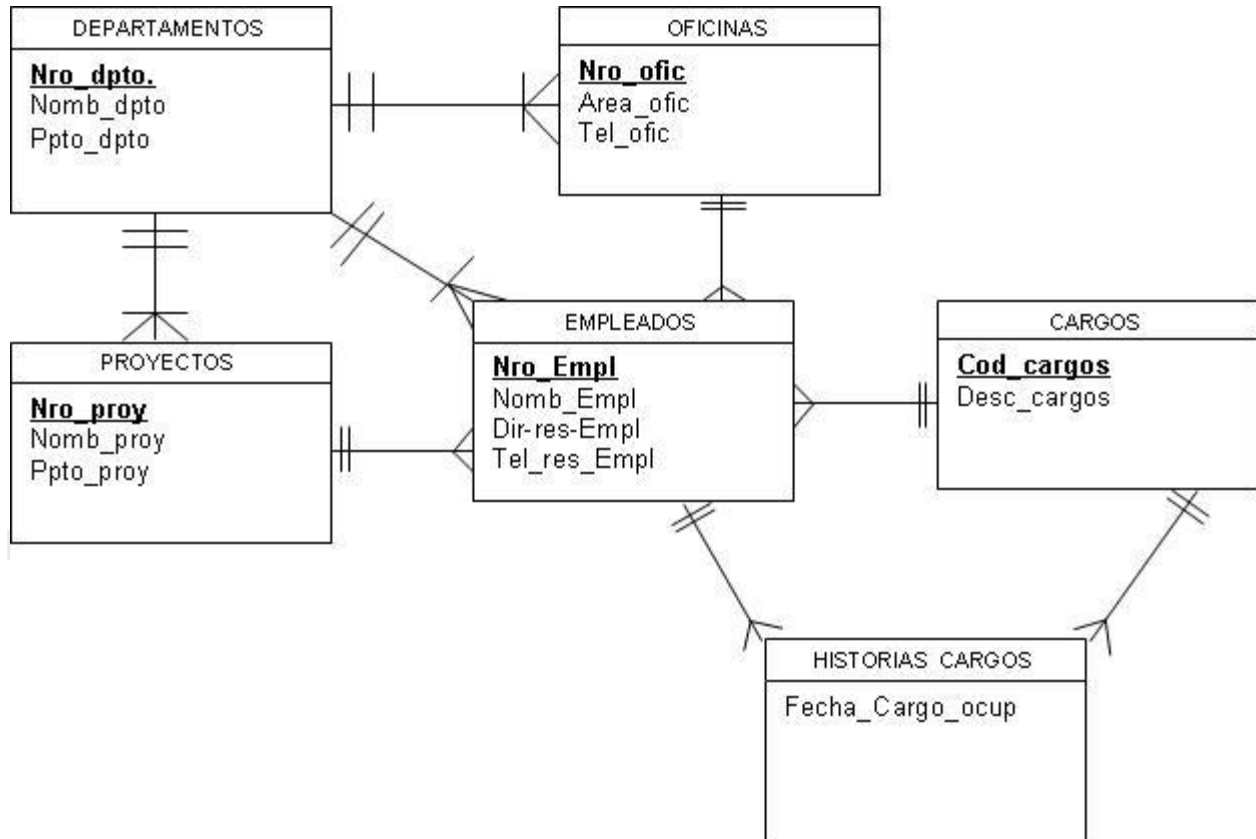


Figura 20 – DER- 6 caso empleados

- Especificación de tipos de entidades

6) Para finalizar nuestro diagrama, entraremos a determina el tipo de entidad, de acuerdo a esto, tenemos que las entidades, DEPARTAMENTOS, EMPLEADOS, PROYECTOS, OFICINAS y CARGOS son entidades fuerte, pues tienen un atributo clave que le garantiza unicidad, es decir, para cada conjunto de instancias y ocurrencias no hay repetición; como ejemplo, a los departamentos se les asignan un número para cada uno que tenga la compañía, de tal forma, que no hay dos departamentos con nombre diferentes y el mismo número. Lo anterior, sucede con todas las demás entidades.

Ahora , miremos el caso de la entidad HISTORIA DE CARGOS; ésta no tiene atributo clave, hasta ahora. Recordemos, que las entidades débiles siempre deben tener un atributo clave y que la entidades asociativas pueden o no tener atributos y/o atributo clave. Comenzaremos nuestro análisis colocando la fecha como atributo clave, pero observamos que este atributo no es suficiente para garantizar unicidad, pues en las misma fechas pueden haber mas de un empleado que se le cambio el cargo, entonces necesitaremos apoyarnos en otra entidad, para la cual esta

relacionada HISTORIA DE CARGOS; escojamos la entidad EMPLEADOS, de tal forma, que para una fecha dada, podemos decir que un mismo empleado no puede haber cambiado de cargo, esto nunca ocurre, puesto que es como decir que el empleado Pedro Perez, el días 9 de septiembre del 2008 estaba asignado a dos cargos, lo cual violaría la regla que dice que un empleado está asignado a un cargo al tiempo.

Con base en los anterior, podemos concluir, que HISTORIA DE CARGOS, es una entidad que depende solo de la entidad EMPLEADOS, por lo tanto es debil. De acuerdo a esto, el diagrama quedaría así:

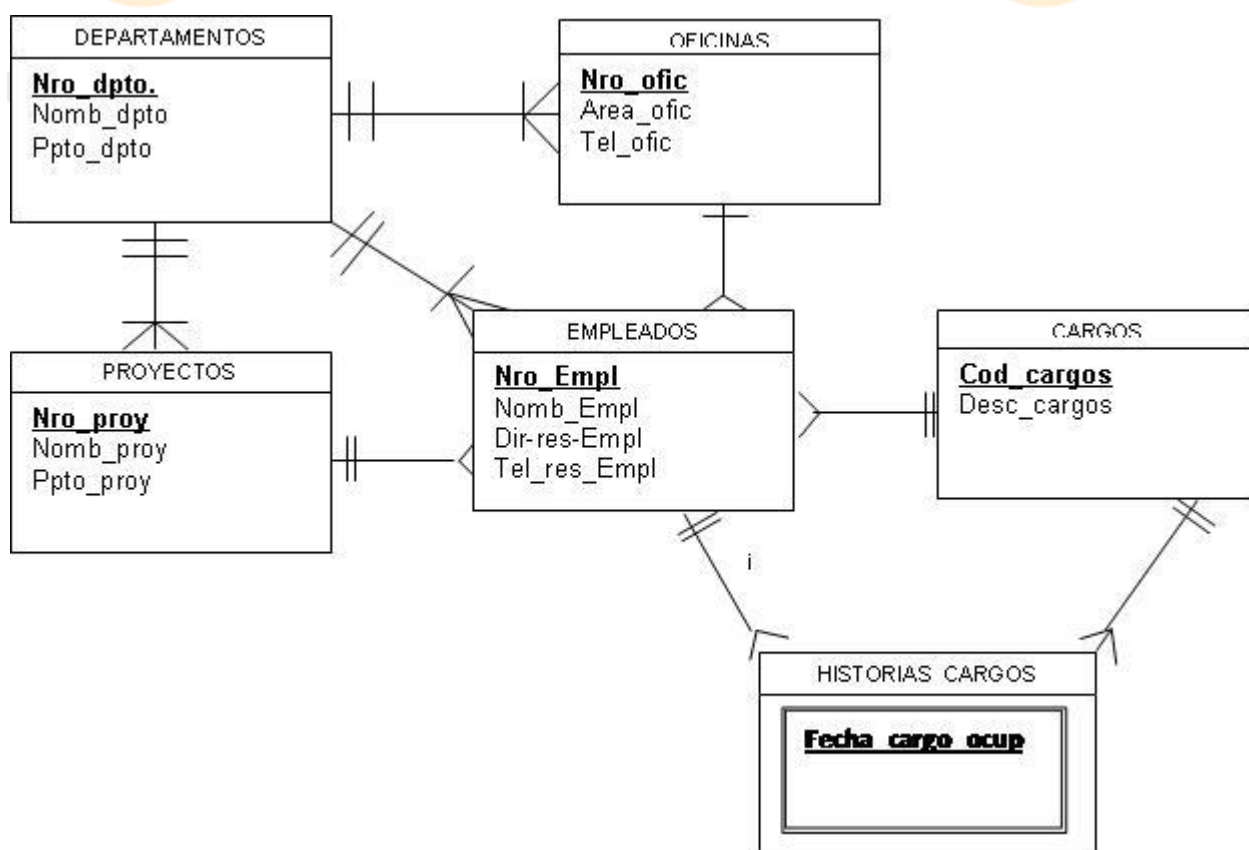


Figura 21 – DER- 7 caso empleados

En el diagrama anterior, observe que sobre la relación de EMPLEADOS e HISTORIAS DE CARGOS, hay una *i*, está, es la forma de identificar de qué entidad depende la entidad debil. Además, si se observan todos los atributos y atributo clave en cada uno de las entidades, no hay un atributo repetido y todos los atributos son características propias de cada entidad.

Es bueno aclarar, que la notación que empleamos aquí es la de James Martin, que es un ingeniero de software reconocido, el cual ha escrito muchos libros al respecto. Sin embargo, existen muchas notaciones, como es la de su creador Peter Chen.

Lección 9: “Análisis Caso de Cursos”

- Enunciado Caso

Una empresa que ofrece cursos de informática, esta interesada en crear una base de datos, para administrar todo lo referente a los estudiantes y los cursos. Usted ha sido contratado para que le modele la base de datos y para ello cuenta con la siguiente información:

Los cursos que se ofrecen son excell, access, visual fox pro, etc. Estos cursos tienen un código (único), descripción y un costo, además tienen una duración de 1 a 5 días. Los instructores que tienen, puede dictar uno o más cursos. María Torres y Juan Pérez son dos de los mejores instructores. Por cada Instructor, se lleva un registro manual que contiene cédula, nombre, apellidos y teléfonos. Los cursos dictados, se le asigna un solo instructor, con la respectiva fecha inicial y final, y hora inicial y final en que se dicta, número salón y descripción días de la semana. Los estudiantes pueden matricularse en uno o más cursos y para ello se lleva un número consecutivo de matrícula por curso dictado, la fecha, pago y forma de pago. Se mantiene información de todos los estudiantes sean que estén tomando curso o no en la actualidad para invitarlos cuando se abran cursos nuevos, además se tienen los datos de tipo de identificación, número de identificación, nombres y apellidos completos, entre otros. Las invitaciones se realizan vía internet o telefónicamente. Los diplomas se les envía directamente a su casa u oficina. Anualmente se les invita a una fiesta de fin de año donde hay rifas y concursos.

- Identificación de entidades y Matriz de relación

Las entidades que podemos observar en primera instancia son, de acuerdo al enunciado anterior:

CURSOS , pero los cursos pueden ser los OFERTADOS, es decir los que la institución tiene capacidad para ofrecer. Los otros cursos son los DICTADOS, que son los que la institución ya tienen programados para iniciar en una fecha determinada y el cual ya tiene un instructor asignado y estudiantes que se hayan matriculado. De esta forma, entonces tenemos que se generan dos entidades:

CURSOS OFERTADOS

CURSOS DICTADOS

Con respecto a los demás sustantivos, encontramos:

INSTRUCTORES

ESTUDIANTES

MATRICULAS

Realizando la matriz y haciendo el ejercicio de las relaciones tenemos:

ENTIDADES	CURSOS OFERTADOS	INSTRUCTORES	ESTUDIANTES	CURSOS DICTADOS	MATRÍCULAS
CURSOS OFERTADOS	X	X	X	1 - N Tener	X
INSTRUCTOR	X	X	X	1 - N Dictar	X
ESTUDIANTES	X	X	X	X	1 - N Tener
CURSOS DICTADOS	1 - 1 Pertenece	1 - 1 Tiene	X	X	1 - N Tener
MATRICULAS	X	X	1 - 1 Pertenece	1 - 1 Pertenece	X

Figura 22 – Matriz relación caso curso

De la matriz observemos las relaciones de las entidades de los cursos, tanto ofertados, como dictados, encontramos que:

Los cursos ofertados tienen relación solo con cursos dictados, debido a que los cursos ofertados, solo tiene el inventario de los cursos que ofrece la institución. Mientras que los cursos dictados tienen relación con todas las demás entidades, menos con estudiantes. Lo anterior debido, a que alrededor del curso dictado, se

asocia el instructor que lo dicta, las matriculas que realizan los estudiantes y el curso al cual pertenece.

Con respecto a la matrícula, vemos que está relacionada solo con estudiantes y los cursos dictados. En la primera relación, ya que un estudiante para tomar un curso debe matricularse en los cursos que ya se tienen programados, es decir, se conoce su fecha, horario e instructor que lo va a dictar. De esto se concluye por qué las entidad estudiantes y cursos dictados NO tienen relación, pues esta se realiza a través de la matrícula.

Paso matriz a diagrama E-R

Como primera medida, debemos colocar las entidades, de tal forma, que la entidad que mas relaciones 1 -1 tengan, o se identifique mas el problema se coloca en la mitad, así tenemos, que la entidad CURSOS DICTADOS, tiene mas relaciones 1-1, quedando inicialmente el diagrama entidad relación así:

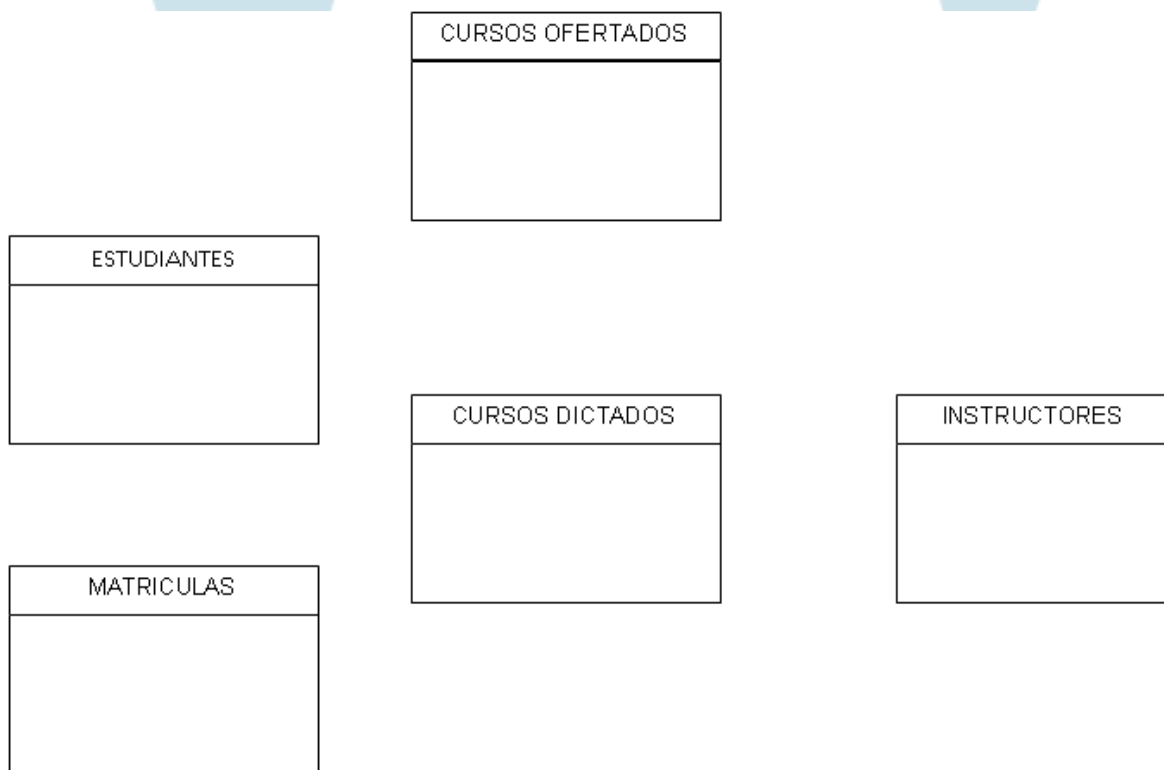


Figura 23 – DER-1 caso curso

El segundo paso sería establecer las relaciones, donde cursos dictados, estudiantes e instructor solo tienen una relación. Matrícula tiene dos relaciones y cursos dictados tiene tres relaciones.

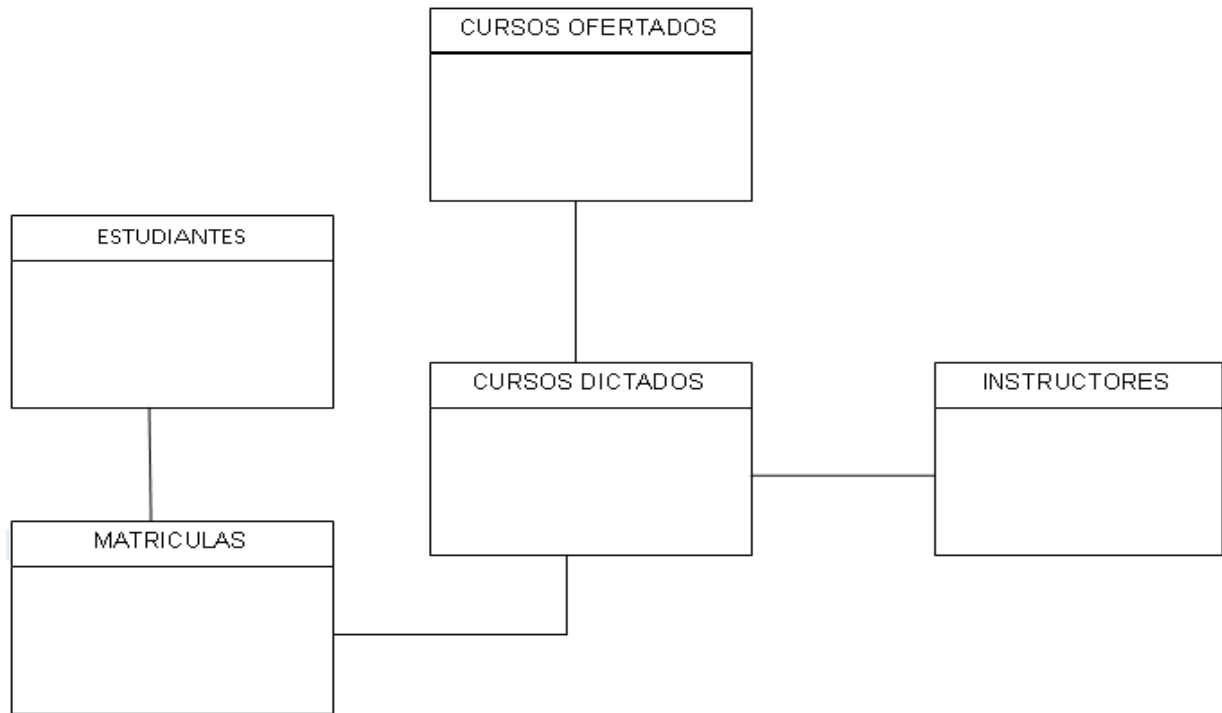


Figura 24 – DER-2 caso curso

Ahora miramos la cardinalidad, de tal forma que queda, que UN curso ofertado, puede haberse dictado muchas veces. Que un instructor puede haber dictado muchas veces varios cursos y que un estudiante puede haberse matriculado varias veces en los cursos. En el caso de los cursos dictados, encontramos que un curso dictado, solo pertenece a un curso ofertado; que un curso dictados solo puede ser dictado por un instructor y que en un curso dictado pueden haber matriculados muchos estudiantes. En este último es bueno hacer una aclaración, muchos estudiantes se preguntarán por qué no relacionar directamente cursos dictados con estudiantes y la respuesta tiene el siguiente análisis:

La relación da una cardinalidad de N-N, pues un curso dictado puede tener muchos estudiantes que lo han visto y UN estudiante puede haber tomado muchos cursos dictados, como las relaciones N-N, no es una función, es decir, NO son válidas para el diseño, entonces se debe destruir esta relación y aparecerá otra entidad, que puede ser asociativa o débil o fuerte, ésta última, sí es que se decide darle una identificación única: veamos esto último gráficamente:

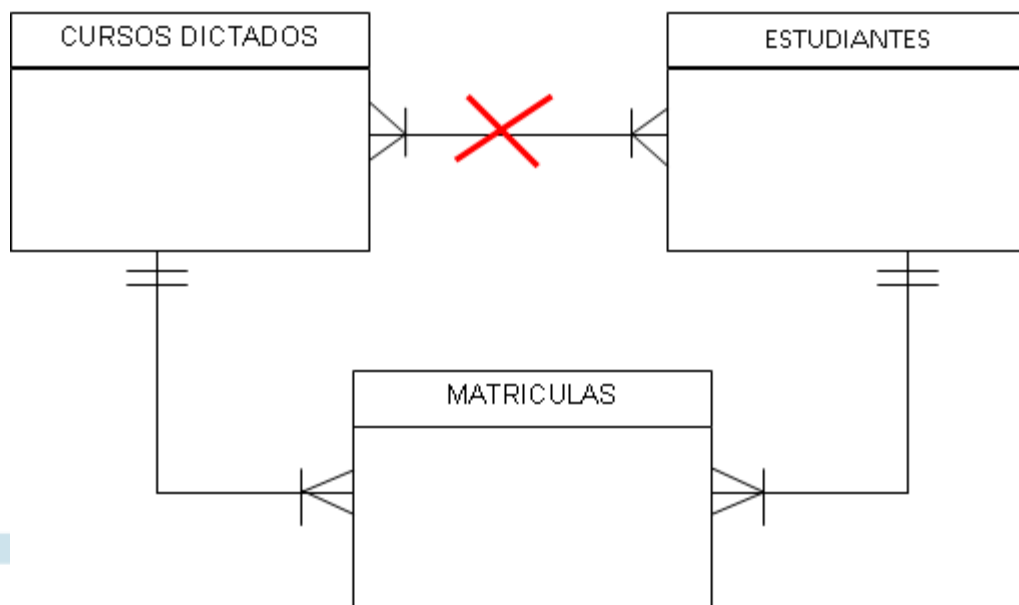


Figura 25 – DER-3 caso curso

En muchas ocasiones, no se tienen nombres creativos para la nueva entidad, dando por lo general nombres combinados de las dos entidades que intervienen Ej.: "ESTUDIANTES-CURSOS DICTADOS".

Con el anterior análisis, ya podemos entonces entrar a colocar las cardinalidades al diagrama.

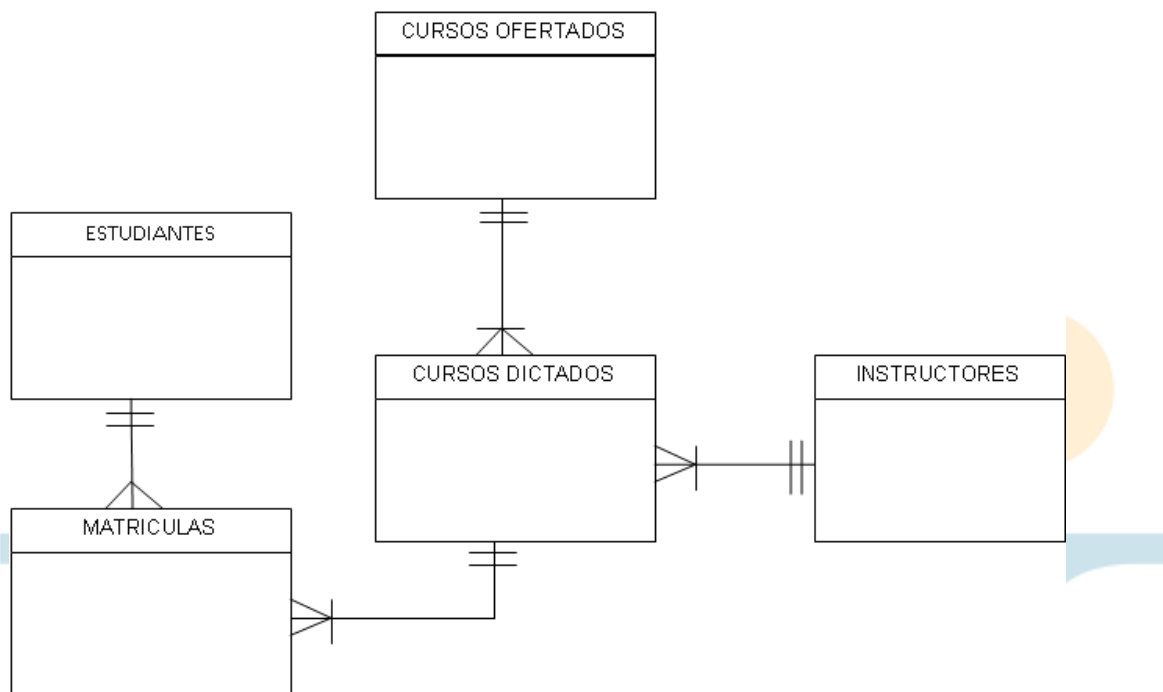


Figura 26 – DER-4 caso curso

Selección de atributos y atributo clave

En este paso, procederemos a colocar los atributos en las respectivas entidades, los cuales salen de identificar los sustantivos que se encuentran en singular en el caso. Veamos entonces como queda el diagrama:

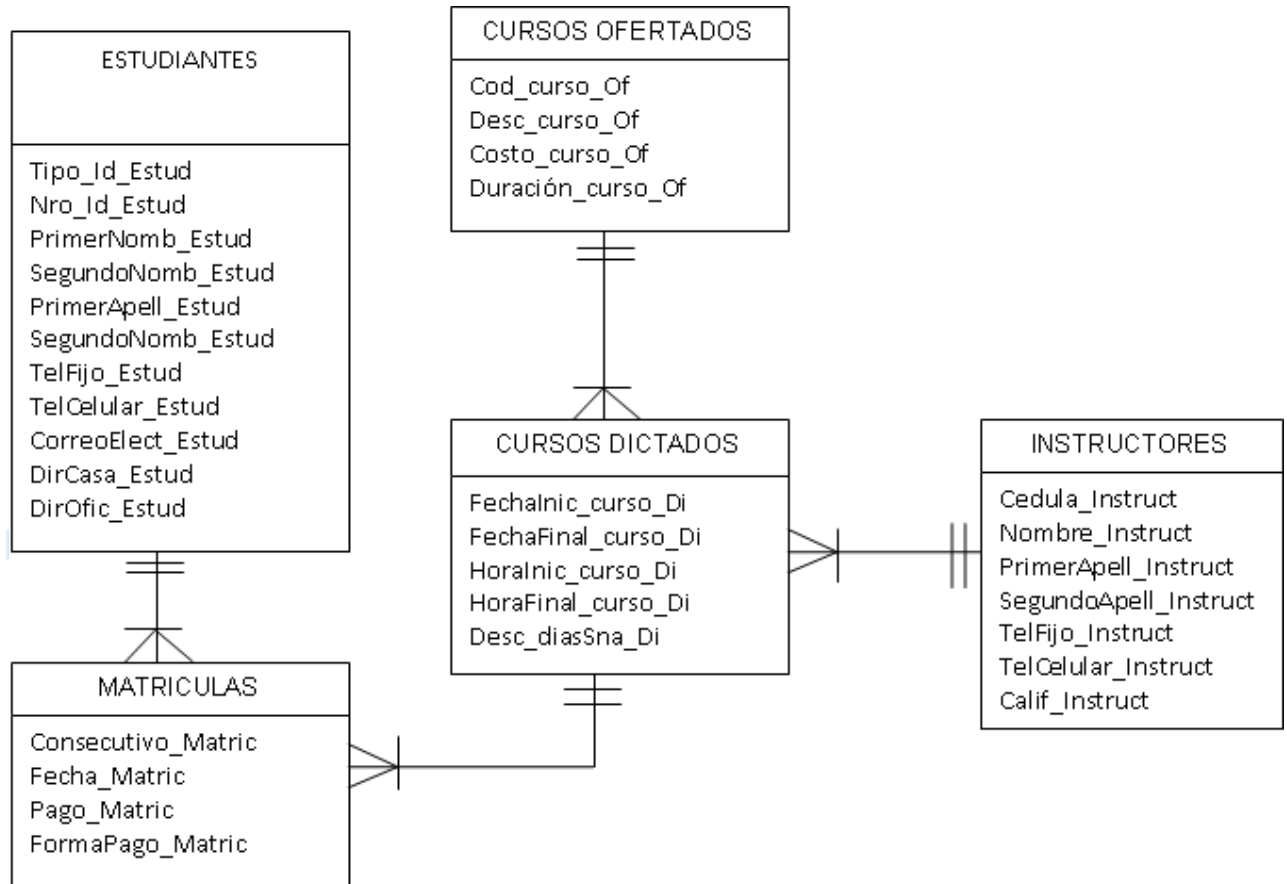


Figura 27 – DER-5 caso curso

Con base a los atributos anteriores, miremos entonces cuáles pueden ser atributos claves, para ello miremos en cada entidad:

1. En la Entidad de Cursos Ofertados, vemos que hay un **código**, el cuál en el problema dice que es único, luego entonces, este atributo es el atributo clave, pues nos garantiza que para cada curso va existir un unico código, es decir, nunca va a ver dos curso con un mismo código.
2. En la entidad de Instructores, vemos que la **cédula** es el atributo clave, pues no existen dos personas con el mismo número de cédula.
3. En la entidad de Estudiantes, vemos que el **Nro. de Id** es el atributo clave, pues no existen dos personas con el mismo número de identificación, ya sera cédula, o Tarjeta de Identidad o Cédula de Extranjería.
4. En la entidad Matrículas, vemos que el **Consecutivo** es el atributo clave, pues este es un número que jamás se repite.

5: Por último tenemos la entidad de cursos dictados, aquí vemos que no hay claridad de cuál debe ser el atributo, pues los que tiene, ninguno garantiza unicidad, ya que las fechas se pueden repetir, lo mismo que la hora. en estos casos, entonces nos enfrentamos a que la entidad NO es FUERTE. Lo anterior lo analizaremos en el próximo apartado.

De acuerdo a lo anterior, entonces tenemos el siguiente diagrama, donde los atributos en negrilla, son los atributos claves:

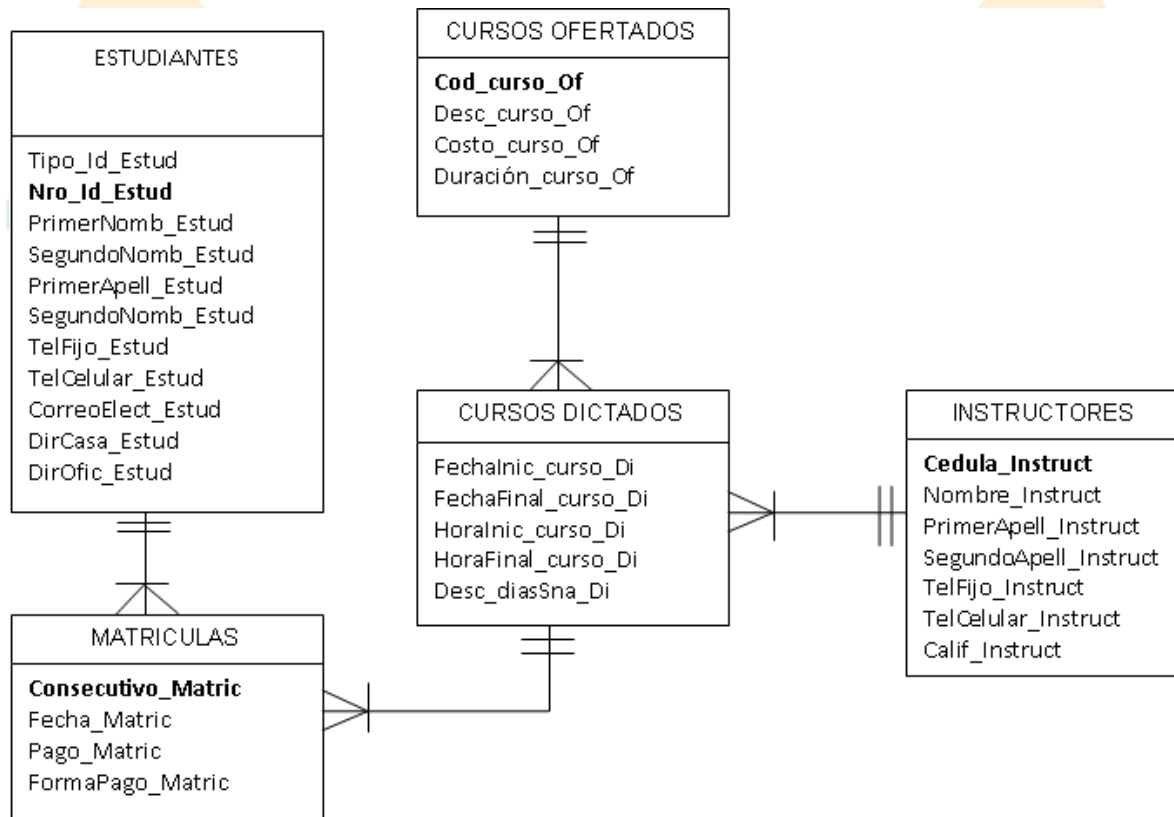


Figura 28 – DER-6 caso curso

Es de aclarar, que la entidad todas las entidades que NO sean ASOCIATIVAS, deben tener atributo clave, en el siguiente apartado, analizaremos el caso de la entidad CURSOS DICTADOS.

- Especificación de tipos de entidades

Para determinar los tipos de entidades en el caso de CURSOS, miramos con base en la especificación de los atributos claves realizados en el apartado anterior que las entidades CURSOS OFERTADOS, INSTRUCTORES, MATRÍCULAS Y ESTUDIANTES, tienen un atributo clave que las identifica totalmente, es decir,

que si tomamos una entidad de éstas y le borramos las relaciones que tienen con otras, las instancias u ocurrencias u objetos siguen existiendo, no hay necesidad de borrarlas.

Ahora miremos la entidad de CURSOS DICTADOS. La pregunta que nos debemos hacer es ¿De qué entidad o entidades, de las que está relacionada, se puede hacer depender? De tal forma que al combinar el atributo clave de dichas entidades, con uno o varios atributos de la entidad CURSOS DICTADOS, me pueda garantizar unicidad.

Para contestarnos esta pregunta, debemos revisar cuál o cuáles entidades (CURSOS OFERTADOS, INSTRUCTORES, ESTUDIANTES) es la más apropiada para ponerla a depender de ellas. Encontramos que, si la ponemos a depender de cursos ofertados y seleccionamos el atributo fecha inicial de cursos dictados, entonces se debe garantizar que nunca se va a dictar un mismo cursos, con fecha inicial igual, pero con horas diferentes. Si este caso se da, entonces deberíamos incluir también la hora inicial, de tal forma que se garantice que nunca va existir un mismo curso dictado con la misma fecha inicial y hora inicial. Lo anterior, se debe definir muy bien, pues una falla en esto, es una falla estructural en el diseño, puesto que es la definición del ATRIBUTO CLAVE.

Como ejemplo, supongamos que el curso ofertado de WORD, se va a dictar con fecha inicial de Agosto 15 de 2010, con hora inicial de 8 a.m. y que simultáneamente, exista otro curso WORD que se vaya a dictar con la misma fecha inicial (Agosto 15 de 2010), pero dictándolo a las 7 p.m. Si este es el caso, entonces podemos decidir que :

La entidad de CURSOS DICTADOS, es Débil con respecto a la entidad de CURSOS OFERTADOS y que los atributos claves son Fecha y Hora inicial del curso dictado. Veamos esto Gráficamente:

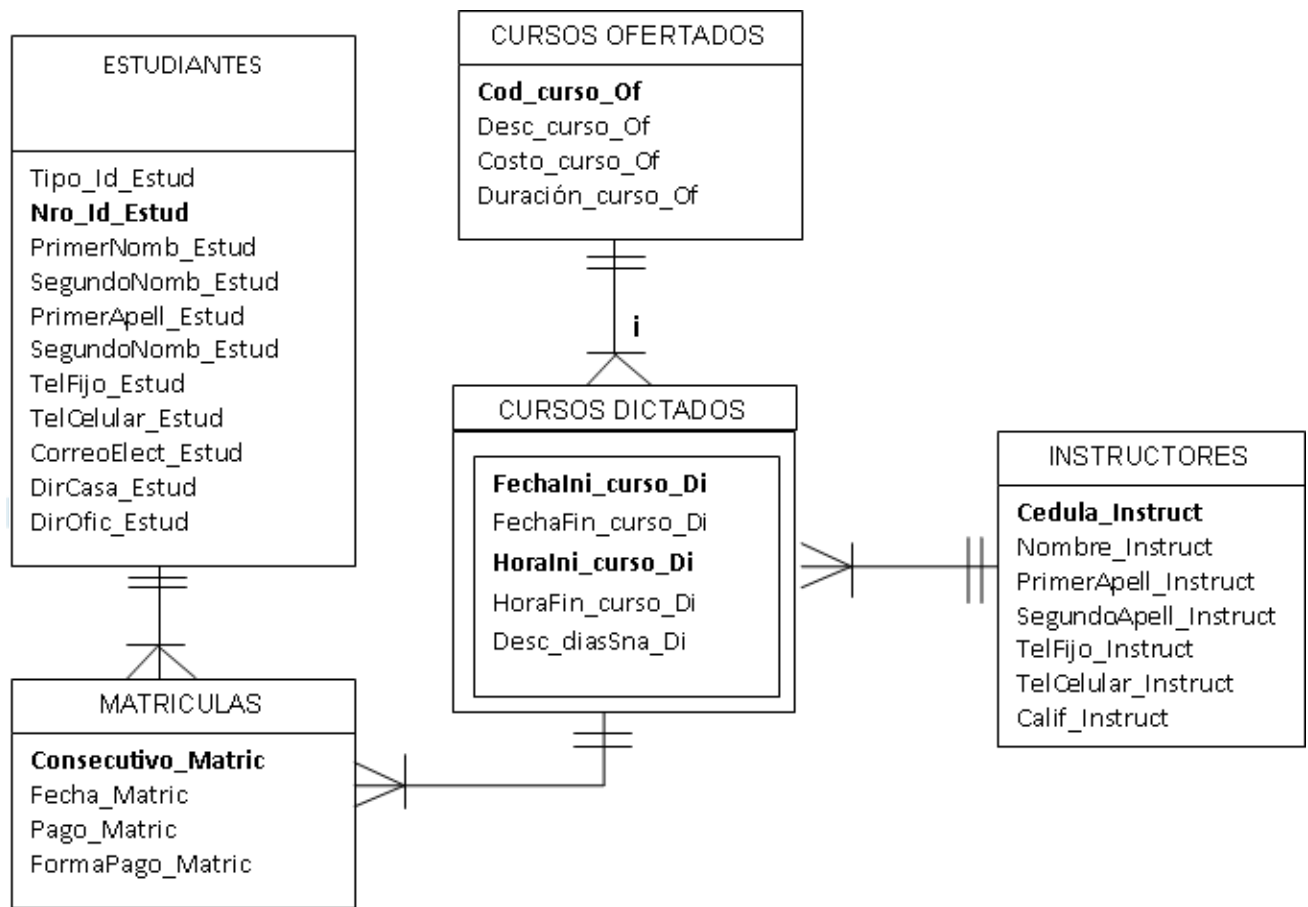


Figura 29 – DER-7 caso curso

Lección 10: “Análisis Caso de Órdenes de Pedidos”

Enunciado Caso Ordenes de Recibo

Se desea modelar una base de datos para recibo de órdenes, la cual debe contener la información de clientes, artículos y órdenes.

De los clientes se desea tener su número (único), dirección de envío (varias por cliente), Ingresos Anuales, límite de crédito y descuento.

De las ordenes se desea tener su número de la orden (único), número del cliente que la ordenó, la dirección de envío del cliente, fecha de orden. Además, todo el detalle de la orden, la cual contiene, los artículos pedidos con su cantidad ordenada y su cantidad pendiente, cada línea del detalle de la orden tienen un número de línea, que es un consecutivo.

De los artículos se desea tener su número (único), descripción del artículo, código de la planta que lo produce (varias) y cantidad almacenada en cada planta que lo produce.

Además se sabe que:

No existen dos clientes con una dirección de envío común

- Identificación de Entidades

Se desea modelar una base de datos para recibo de órdenes, la cual debe contener la información de clientes, artículos y órdenes.

De los clientes se desea tener su número (único), dirección de envío (varias por cliente), Ingresos Anuales, límite de crédito y descuento.

De las órdenes se desea tener su número de la orden (único), número del cliente que la ordenó, la dirección de envío del cliente, fecha de orden. Además, todos los detalles de las ordenes, las cuales contienen, los artículos pedidos con su cantidad ordenada y su cantidad pendiente; cada línea del detalle de las ordenes tienen un número de línea, que es un consecutivo.

De los artículos se desea tener su número (único), descripción del artículo, código de la planta que lo produce (varias) y cantidad almacenada en cada planta que lo produce.

Además se sabe que:

No existen dos clientes con una dirección de envío común

Observamos que en el caso anterior, hay tres sustantivos en plural que nos identifican de entrada tres entidades como son:

ORDENES

CLIENTES

ARTICULOS

Miremos entonces como queda la matriz de relación.

ENTIDADES	ORDENES	CLIENTES	ARTICULOS
ORDENES	X	1 -1 Pertenece	1 -N tener
CLIENTES	1 -N tener	X	X
ARTICULOS	1 -N estar	X	X

Figura 30 – Matriz relación caso órdenes

Con respecto a la diagonal, observamos que nos hay relación, esto debido a que no hay subconjuntos de ORDENES, CLIENTES ni ARTICULOS.

Ahora si analizamos la primera fila, encontramos que ORDENES tiene relación con CLIENTES y ARTICULOS, donde una orden pertenece máximo a un clientes, es decir, no puede haber ordenes que pertenezca a dos clientes; una orden puede tener muchos artículos, esto debido a que los clientes hacen su solicitud de artículos a través de una orden. Por lo tanto, si observamos en la segunda fila, el cruce entre CLIENTES y ARTICULOS, NO hay relación, debido a que el cliente pide sus artículos a través de las ORDENES. Siguiendo con la fila de CLIENTES, observamos que un cliente puede tener muchas ordenes, esto porque el cada vez que hace un pedido a la empresa, se le generará una orden.

Por último está la fila de artículos, está solo tiene relación con las ORDENES, de tal forma, que un artículo puede estar en varias ordenes, ya sea del mismo cliente o clientes diferentes.

- Paso Matriz Relación a Diagrama ER

Observando la matriz anterior, encontramos que la entidad central es ORDENES, pues es la que tiene mas relación uno a uno. También vemos que las otras dos entidades no se relacionan entre sí y por lo tanto graficamente tenemos:

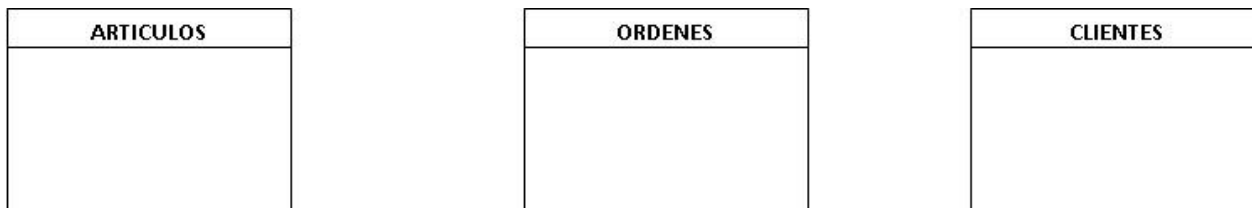


Figura 31 – DER-1 caso órdenes

Procedemos a colocar las relaciones, donde miramos que una orden puede tener muchos artículos y un artículo puede estar en muchas ordenes. Por otro lado, una orden pertenece solo a un cliente y un cliente puede tener varias ordenes. Miremos esto gráficamente.

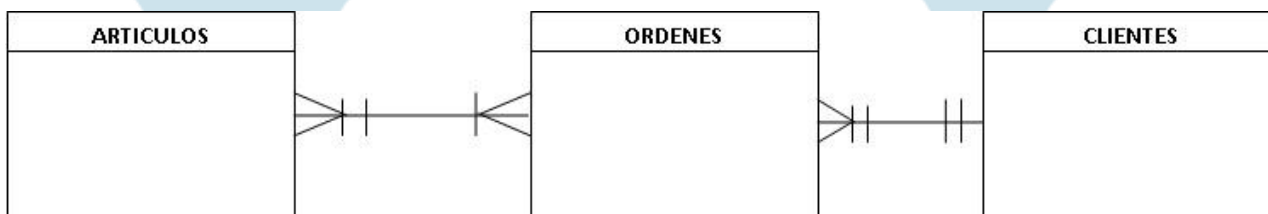


Figura 32 – DER-2 caso órdenes

Observamos en el gráfico anterior, que entre las entidades ARTICULOS y ORDENES, existe una relación Mucho a Mucho, como esto no es permitido, entonces procedemos a romperla, de tal forma que se crea una nueva entidad, que podemos llamar **DETALLES_ORDENES**. Miremos gráficamente esto:

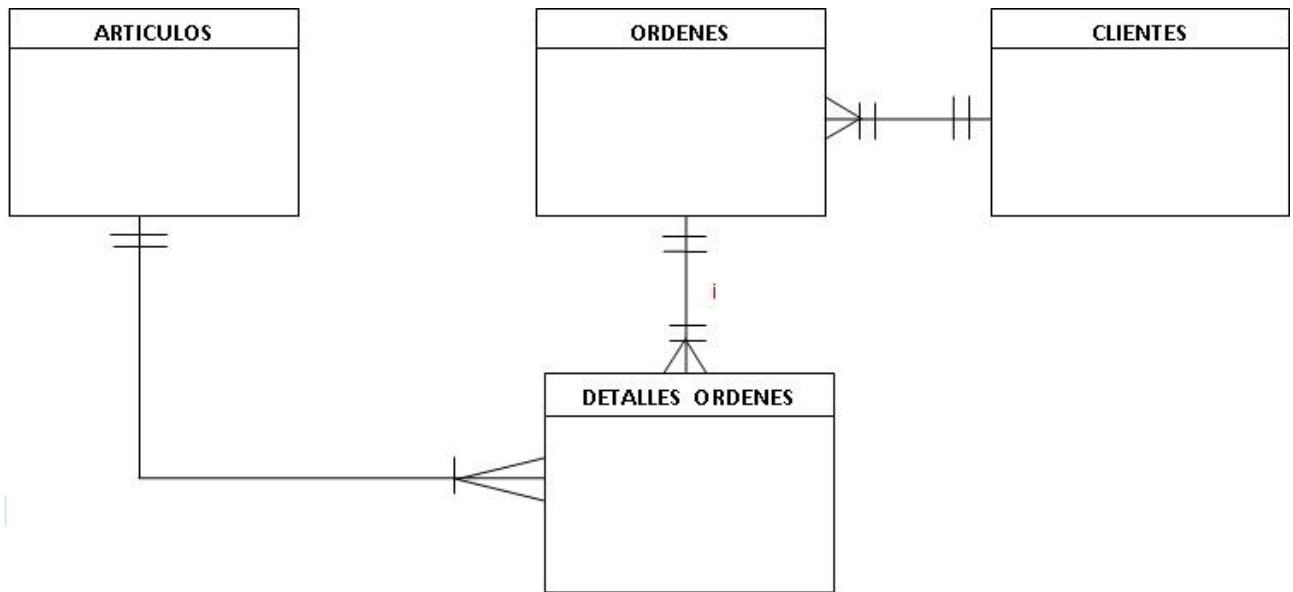


Figura 33 – DER-3 caso órdenes

Mirando el gráfico anterior, obsérvese que la cardinalidad se invierte, es decir, ya en las entidades ARTICULOS Y ORDENES, queda 1 a 1, mientras que en la nueva entidad (DETALLES_ ORDENES), QUE 1 a muchos. Otra observación, es que siempre los detalles, dependen del encabezado, en este caso ORDENES, por eso la notación de la **i**.

Selección Atributos, Atributos Claves y Tipos de Entidad

De acuerdo al caso, vamos a colocar los atributos a las entidades, para ello se debe volver a leer el caso de tal forma que a medida que vamos encontrando atributos los vamos colocando en la entidad que corresponda. Leamos de nuevo el caso

Enunciado caso

Se desea modelar una base de datos para recibo de órdenes, la cual debe contener la información de clientes, artículos y órdenes.

De los clientes se desea tener su número (único), dirección de envío (varias por cliente), nombre o razón social, Ingresos Anuales, límite de crédito y descuento.

De las órdenes se desea tener su número de la orden (único), número del cliente que la ordenó, la dirección de envío del cliente, fecha de orden. Además, todo el detalle de la orden, la cual contiene, los artículos pedidos con su cantidad

ordenada y su cantidad pendiente, cada línea del detalle de la orden tienen un número de línea, que es un consecutivo.

De los artículos se desea tener su número (único), descripción del artículo, código de la planta que lo produce(varias) y cantidad almacenada en cada planta que lo produce.

Además se sabe que:

No existen dos clientes con una dirección de envío común

Una planta puede producir varios artículos.

Identificación de atributos

Mirando el segundo párrafo encontramos que en la entidad clientes se desea tener guardada la información de número, dirección de envío y otros, pero observamos que en dirección de envío nos dicen que un cliente puede tener varias, por lo tanto, esta característica se vuelve entidad porque es un conjunto de DIRECCIONES_ENVÍO, y las demás características son atributos de la entidad cliente. Gráficamente tenemos:

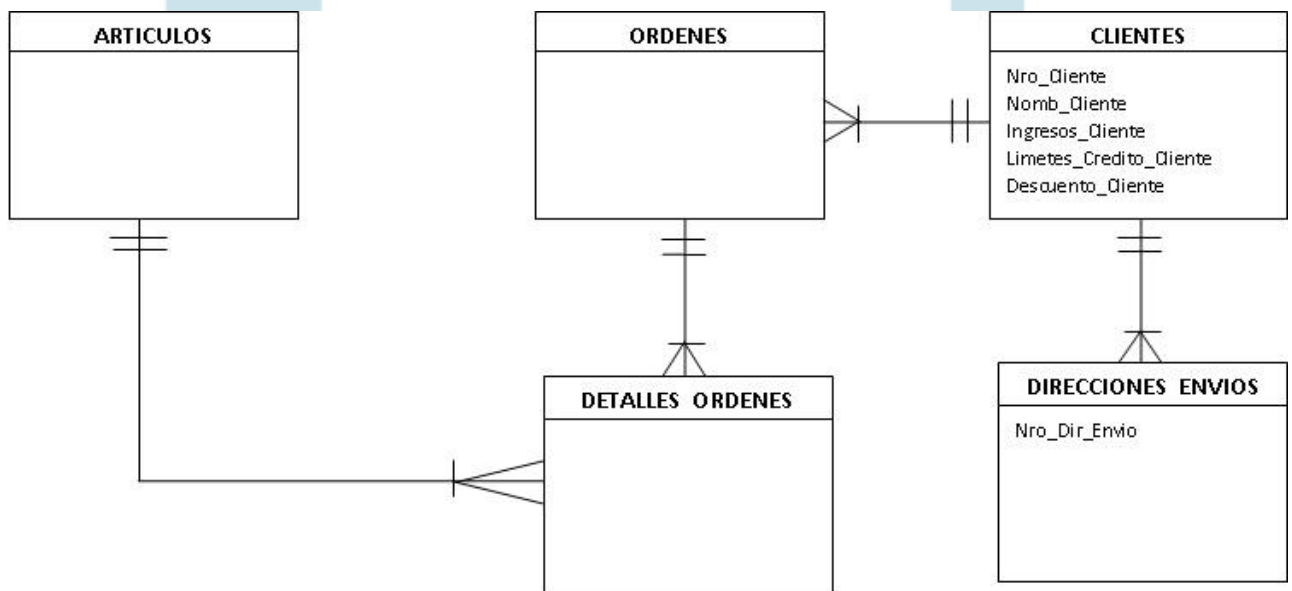


Figura 34 – DER-4 caso órdenes

Ahora miremos el tercer párrafo, donde se dice que las ORDENES, se desea tener un número único, el número del cliente que la ordenó, aquí este número NO debe colocarse como atributo de la entidad ordenes, debido a que es un atributo de la

entidad clientes y para esto se establece una relación entre las entidades ORDENES y CLIENTES, que ya está; en cuanto a la dirección de envío del cliente, tampoco debe ser un atributo de ORDENES, porque ya es atributo de DIRECCIONES_ENVIOS, por lo tanto se debe establecer una relación entre ORDENES y DIRECCIONES_ENVIOS, de tal forma que UNA orden solo puede tener una dirección de envío como máximos, mientras que UNA dirección de envíos puede estar en VARIAS ordenes, dependiendo de cuantas veces el cliente escogió esa dirección para que le hicieran llegar sus pedidos. Por último se tiene la fecha de la orden, que es un atributo de ÓRDENES. Gráficamente quedaría así:

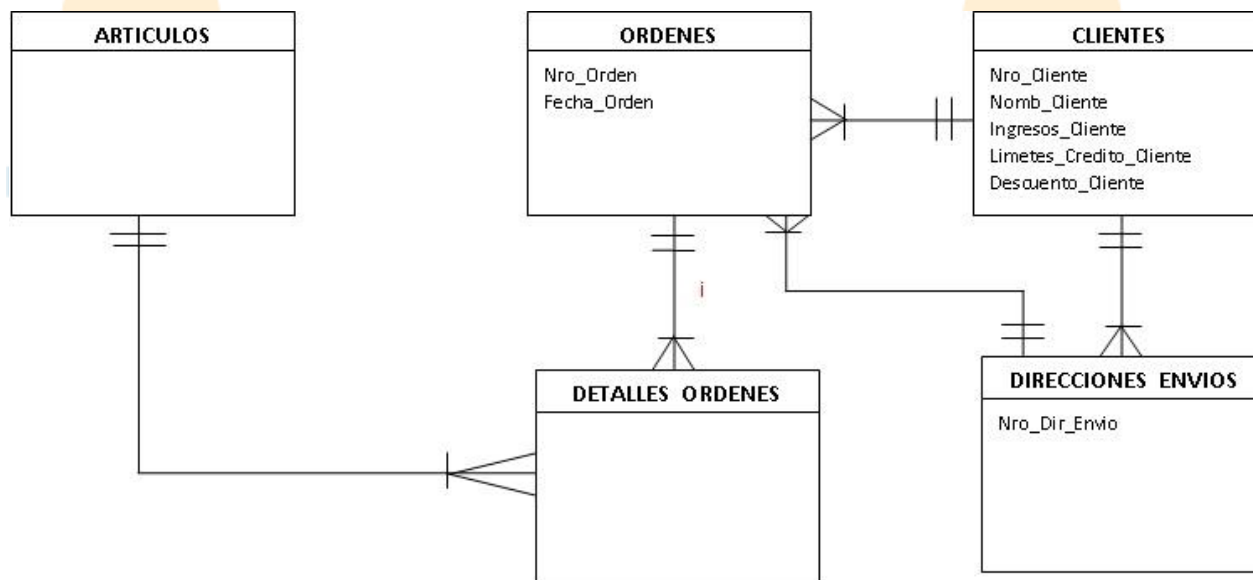


Figura 35 – DER-5 caso órdenes

Siguiendo con el tercer párrafo, encontramos el detalle de la orden que es la entidad DETALLES_ORDENES, y en esta se guarda el artículo, con su respectiva cantidad pedida y cantidad pendiente. Sin embargo, el artículo no es atributo, porque ya es una entidad y por lo tanto se establece una relación entre DETALLES_ORDENES y ARTICULOS, el cual ya se encuentra. Por lo tanto, quedan los atributos de la cantidad pedida y pendiente. Por último, se dice que el detalle, tiene un número de línea que es un consecutivo, el cual es otro atributo. Gráficamente queda así:

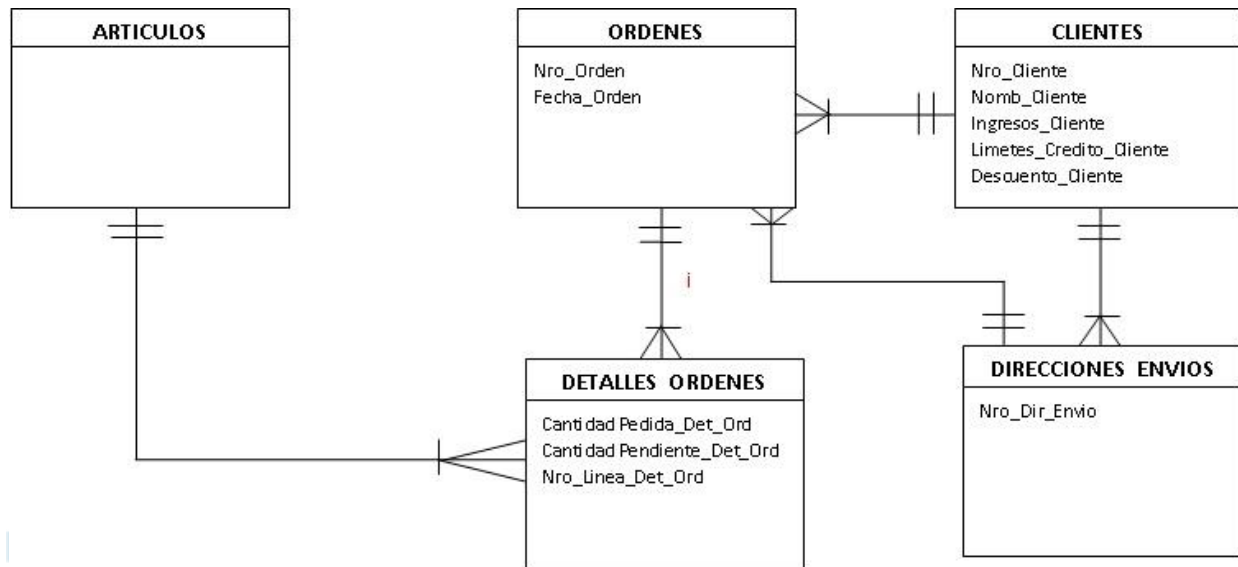


Figura 36 – DER-6 caso órdenes

Por último tenemos que de ARTICULOS se dese tener la información de número, descripción, código de planta que lo produce, que pueden ser varias; aquí entonces tenemos, que los primeros dos son atributos de la entidad ARTICULOS, pero el último, como son varias plantas que pueden producir un mismo artículo se hace necesario crear una entidad PLANTAS y uno de los atributos de ésta sería el código planta. Entonces tenemos, que UN artículo puede ser producido por MUCHAS plantas, y que una planta puede producir VARIOS artículos. Gráficamente tenemos:

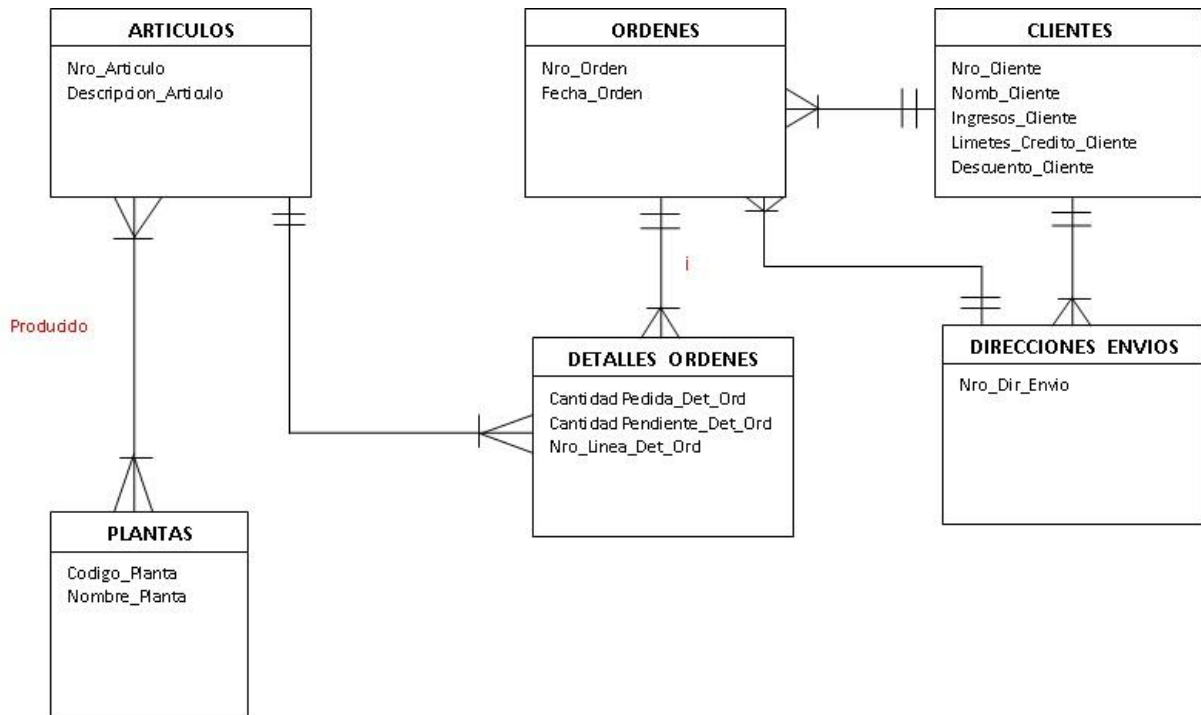


Figura 37 – DER-7 caso órdenes

En lo anterior observamos, que se nos presenta la misma situación de ORDENES y ARTÍCULOS, que hay una relación MUCHA a MUCHA entre ARTICULOS Y PLANTAS. Por lo tanto, debemos destruir esta relación, de tal forma que aparece una nueva entidad que se puede llamar ARTICULOS_PLANTAS, pero que en última instancia lo que se desea es guardar es la cantidad almacenada de cada artículo en la planta. Luego entonces tenemos, que si somos creativos, esta entidad la podemos llamar ALMACENAMIENTOS. Por lo tanto la cantidad almacenada, es un atributo de ésta última entidad. Gráficamente tenemos:

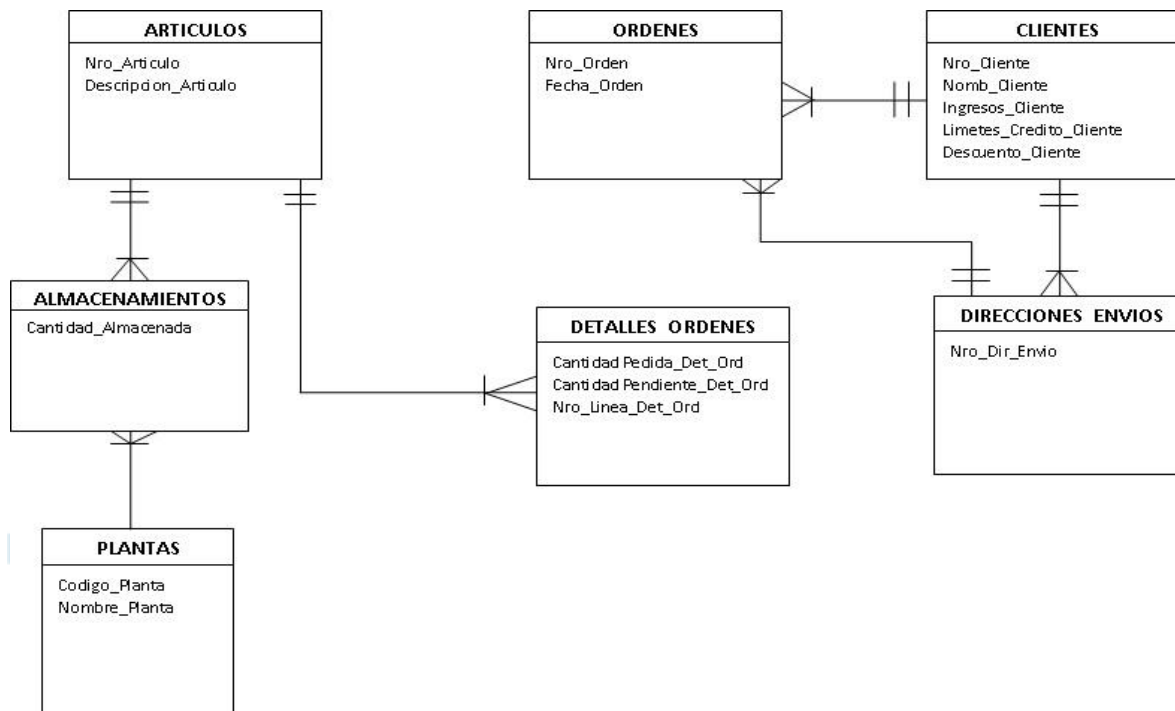


Figura 38 – DER-8 caso órdenes

Observando todo lo anterior, vemos que en entidad PLANTAS, aparece un atributo como es el *Nombre_Planta*, esto debido, a que cada vez que tengamos este tipo de entidades donde tenga un código, necesariamente debe haber un nombre o descripción, según se desee.

Definición de Atributos Claves y Tipos de Entidad

Vemos que la entidades CLIENTES, ARTÍCULOS, ORDENES Y PLANTAS, tiene un número único o un código, que son los atributos claves de estas entidades y son respectivamente: ***Nro_Cliente***, ***Nro_Articulo***, ***Nro_Orden*** y ***Codigo_Planta***. Por lo tanto son entidades FUERTES.

En cuanto a la entidad DIRECCIONES_ENVIOS, podemos decir que el único atributo que tiene es el atributo clave, por cuanto en el caso se garantiza que dos clientes no tienen nunca una misma dirección. Además asumimos aquí que las direcciones son a nivel municipal y no departamental, no nacional. Si fuera así, entonces debíamos pensar en otra entidad como es municipios, relacionarla con dirección y hacer ésta última débil con respecto a municipios, pues en un municipio nunca hay direcciones repetidas. Para no complejizar el problema, entonces asumanos que es local y por lo tanto el atributo clave es ***Nro_Dir_Envio***.

En la entidad DETALLES_ORDENES, se dice que tiene un número de línea, que es un consecutivo, por lo tanto no se repite, pero dentro del detalle de la misma orden, es decir, que se puede encontrar número de líneas iguales, pero para diferentes ÓRDENES, nunca en la misma orden. En este caso el atributo clave, no es suficiente para garantizar unicidad, por lo que hay que apoyarse en la entidad ÓRDENES. Así que aunque el atributo clave **Nro_Linea_Det_Ord**, no garantice esto, si es necesarios definirlo, pues cuando una entidad es **DEBIL** con respecto a una sola entidad, se necesita definir un atributo clave, que al combinarlo con el atributo clave de la entidad que depende, en este caso ORDENES, la combinación es única.

Por último tenemos la entidad ALMACENAMIENTO, observamos que tiene solo un atributo que es la cantidad almacenada, y estos tipos de atributos nunca son atributos claves, por lo tanto, debemos recurrir a apoyarnos en las entidades externas con que está relacionada, que en este caso es ARTICULOS y PLANTAS, donde, entonces la entidad ALMACENAMIENTO es una entidad **ASOCIATIVA** y este tipo de entidad es la única que puede tener o no tener atributos.

Con base en el análisis anterior, gráficamente tenemos:

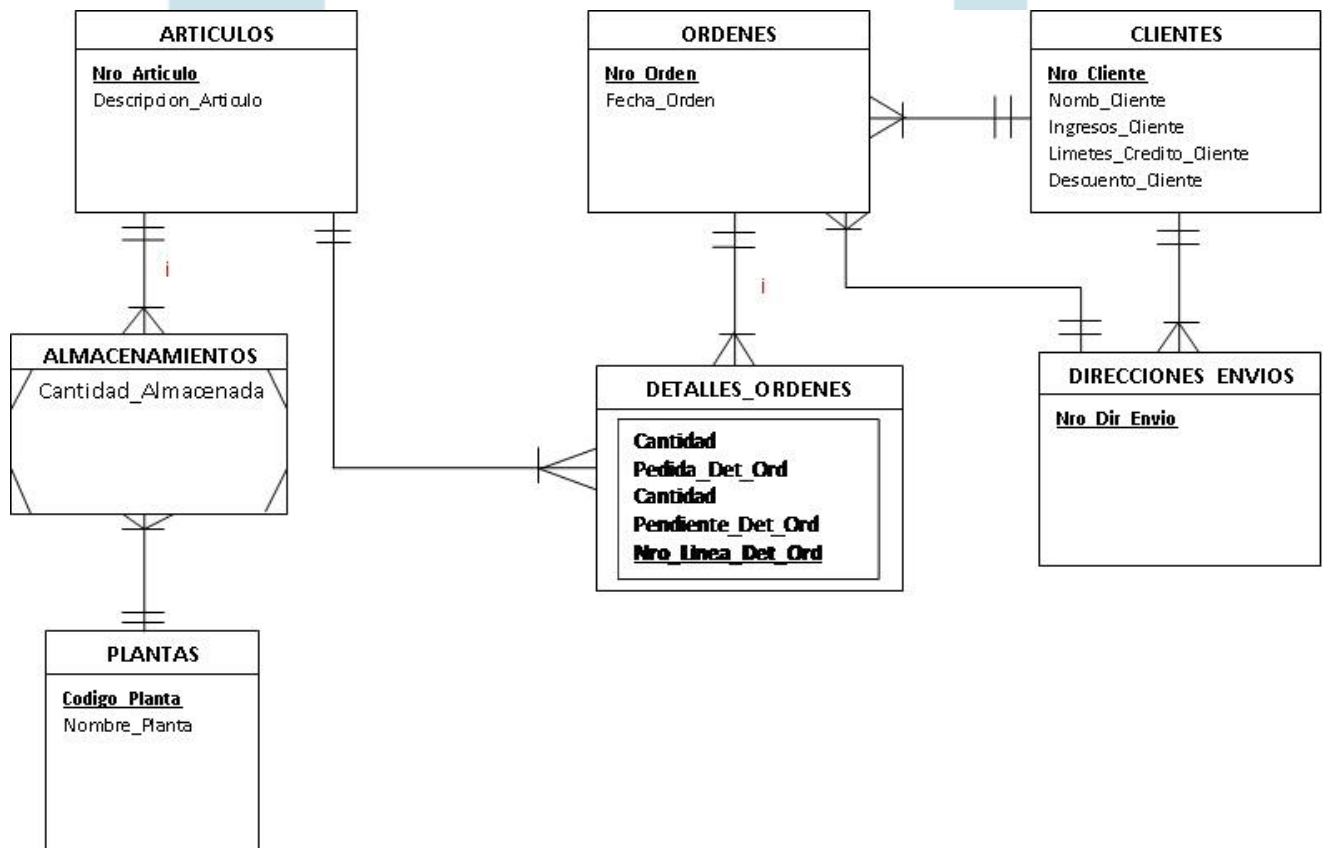


Figura 39 – DER-9 caso órdenes

CAPÍTULO 3: “MODELO DE DATOS BASADOS EN FORMATOS”

Cuando nos encontramos en el mundo de las organizaciones, nos encontramos con muchos formatos que manejan éstas, de tal forma, que cuando vamos a realizar el levantamiento de información, nos encontramos con muchos tipos de éstos y muchas veces no sabemos, como a partir de ellos, modelar lógicamente las organizaciones en términos de sus datos. De allí, que se haga necesario en forma general, reconocer los diferentes tipos y como podemos identificar las entidades, las relaciones y los atributos.


Lección 11: “Tipos de Formatos”

1) Formato de **un solo cuerpo**: el cual consiste en que todos los tipos de datos que contienen, no hay repetición de éstos. En forma gráfica se vería así:

INDUSTRIA ALIMENTICIA NENE ACTUALIZACIÓN DATOS BÁSICOS PROVEEDOR	
Fecha Diligenciamiento:	Nro:
Tipo Id.: <input type="checkbox"/> C.C <input type="checkbox"/> C.E <input type="checkbox"/> NIT	Nro. Id.: _____
Nombre Completo o Razón Social: _____	
Dir. Oficina: _____	Municipio: _____ Dpto.: _____
Correo electrónico: _____	
Teléfono Fijo 1: _____	Teléfono Fijo2: _____ Teléfono Fijo 3: _____
Nro. Celular 1: _____	Nro. Celular 1: _____ Nro. Celular 1: _____
Datos Persona Contacto	
Tipo Id.: <input type="checkbox"/> C.C <input type="checkbox"/> C.E	Nro. Id.: _____
Nombre Completo: _____	
Nro. Teléfono Celular: _____	
Correo Electrónico: _____	
Dir. Oficina: _____	
<div style="display: flex; justify-content: space-between;"> <div> Diligencia Nombre Cargo </div> <div> Firma </div> </div>	

Figura 40 – Formato de un solo cuerpo

2) Formato de un **Dos cuerpos**: el cual consiste en que contienen u cuerpo con tipos de datos no repetidos y otro cuerpo con tipos de datos repetidos. En forma gráfica se vería así:



INDUSTRIA ALIMENTICIA NENE
 SOLICITU VACACIONES EMPLEA DOS

Fecha Diligenciamiento:
 Nombre Dependencia Solicita:

Nro:

Tipo Vacaciones	Nro. Id. Empleado	Nombres completos Empleado	Fecha Causada de Vacaciones	Fecha Inicial Disfrute Vacaciones	Fecha Final Disfrute Vacaciones	Cantidad días Vacaciones

OBSERVACIONES: _____

Nombre Jefe Dependencia _____
 Cargo _____

Firma _____

Figura 41 – Formato de dos cuerpo

Observando el formato anterior, vemos que la columnas tipo de vacaciones, número de identificación del empleado, etc que se encuentran en esa misma fila, son tipos de datos que se repiten, aunque, cuando se llene el formato los datos son deferentes, es decir, empleados diferentes.

3) Formato de un **Varios cuerpos**: el cual consiste en que puede contener uno o varios cuerpo con tipos de datos no repetidos y uno o varios cuerpos con tipos de datos repetidos. En forma gráfica se vería así:

BANCO PAGAME SI PUEDES SOLICITUD DE CRÉDITO									
Fecha Diligenciamiento: _____					Nro.: _____				
Oficina: _____			Municipio: _____			Dpto.: _____			
Datos Persona Contacto									
Tipo Id.	C.C.	C.E.	NIT	Nro. Id.: _____	Lugar Exped.: _____	Fecha: _____			
1er. Nombre : _____					2do. Nombre : _____				
1er. Apellido : _____					2do. Apellido : _____				
Nro. Teléfono Celular: _____					Correo Electrónico: _____				
Direcciones:									
Tipo Dirección	Número Dirección	Municipio	Dpto.						
Residencia									
Oficina									
Familiar Allegado									
Referencias Comerciales									
Nombre completo		Teléfono Fijo		Teléfono Celular		Municipio			
Ingresos									
Rubro		valor							
Salarios									
Arriendos									
Rendimientos Financieros									
Acciones									
Egresos									
Rubro		valor							
Arriendo									
Gastos Varios									
Créditos Financieros									
Hipotecas									
Firma Nombre Cliente: _____					Firma Nombre Empleado: _____				

Figura 42 – Formato de varios cuerpos

Podemos observar en el formato anterior, que las direcciones, referencias comerciales, ingresos y egresos, son cuatro cuerpos que tienen tipos de datos repetidos, mientras que existen dos cuerpos con tipos de datos no repetidos como son los dos primeros y último, donde están las firmas.

Todos los formatos anteriores, son ejemplos de los muchos que se puede encontrar en el mundo de las organizaciones.

Lección 12: “Procedimiento para elaborar Modelos basados en formatos”

De acuerdo a los formatos vistos en la lección anterior, encontramos básicamente que existen tres tipos, para cada uno de ellos existen formas de arrancar estipuladas inicialmente cuál es el número de ENTIDADES según el tipo y recorriendo todos los tipos de datos haciendo preguntas. Empezamos entonces por el número de entidades iniciales:

a) Identificación de Entidades Iniciales

1) Formato de un solo cuerpo

BOLETIN ALIMENTACION
ACTUALIZA CON DATOS DE SUS PROVEEDORES

Fecha diligenciamiento: _____ No:

Tipo ID: ☐ CC ☐ CE ☐ NT No. ID: _____

Nombre Completo o Razón Social: _____ Municipio: _____ Depto: _____

Córeo electrónico: _____

Teléfono Fijo 1: _____ Teléfono Fijo 2: _____ Teléfono Fijo 3: _____

Celular 1: _____ Celular 2: _____ Celular 3: _____

Datos Persona Contacto

Tipo ID: ☐ CC ☐ CE ☐ NT No. ID: _____

Nombre Completo: _____

Celular: _____

Córeo Electrónico: _____

Dr. Oficio: _____

Diligencia: _____ Firma: _____

Entidades

Explicación

DATOS BASICOS PROVEEDORES

De entrada suponemos, que inicialmente todos los tipos de datos son atributos de la entidad datos básicos proveedores, es decir, partimos de una sola entidad y cuyo nombre por lo general es el nombre del formato.

Figura 43 – Entidades iniciales formato un solo cuerpo

1) Formato de dos cuerpos

BOLETIN ALIMENTACION
SOLICITUDES VACACIONES EMPLEADOS

Fecha diligenciamiento: _____ No:

Nombre Dependiente: Solicita:

Tipo Vacaciones	No. ID Empleado	Nombre completo Empleado	Fecha causada de Vacaciones	Fecha Inicial Deseada Vacaciones	Fecha Final Deseada Vacaciones	Cantidad días Vacaciones

Observaciones: _____

Nombre Jefe Dependiente: _____ Firma: _____

Entidades

Explicación

ENCABEZADO

SOLICITUDES VACACIONES
EMPLEADOS



DETALLES

SOLICITUDES VACACIONES
EMPLEADOS

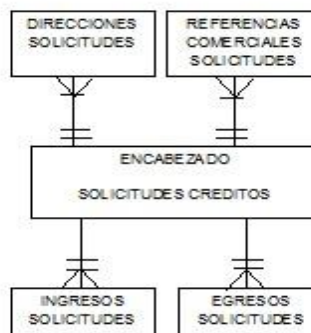
De entrada suponemos, que inicialmente se tiene dos entidades; la primera, el encabezado, del cuerpo con tipos de datos que no se repiten y la segundo con datos que se repiten. La relación que establece es de uno a muchos, pues un encabezado puede tener muchos detalles.

Figura 44 – Entidades iniciales formato dos cuerpos

1) Formato de varios cuerpos



Entidades



Explicación

De entrada suponemos, que inicialmente se tiene la entidad del encabezado y el número de entidades según la cantidad de cuerpos que hayan con tipos de datos repetidos, en nuestro caso son cuatro(4). Todas éstas, están relacionadas con el encabezado de uno a muchos, como si fuera cada una un detalle.

Figura 45 – Entidades iniciales formato varios cuerpos

b) Asignación de Atributos

Para cada formato iniciamos en el primer tipo de datos que nos encontramos y nos hacemos la siguiente pregunta:

¿Es este tipo de datos **atributo** o **entidad**?

La respuesta a la anterior pregunta, nos invita a pensar si es posible que el tipo de dato pueda en un momento dado representar un conjunto o si solo es una característica de la entidad en la que estamos inicialmente ubicados. Un ejemplo de esto sería:

Miremos los tipos de datos **fecha de diligenciamiento** y **Teléfonos Fijos 1, Teléfonos Fijo 2, Teléfonos Fijo 3**, del primer formato. Observemos que con el tipo de datos **fecha diligenciamiento**, no tienen sentido crear un conjunto de fechas, pues si lo creamos, la entidad **fechas diligenciamientos**, solo tendría una instancia u ocurrencia o registro por cada dato básico del proveedor. En el caso de los tipos de datos **Teléfonos Fijos**, vemos que en el formato hablan de 3, es decir, que es un tipo de datos que se repite, por lo tanto se pueden pensar en tener un conjunto de **Teléfonos Fijos** y por lo tanto se crearía la entidad **Teléfonos Fijos** el cual estaría relacionado uno a muchos con la entidad del **Encabezados Datos Básicos Proveedores**. Veamos la situación anterior gráficamente:



Figura 46 – Asignación de atributos

El procedimiento anterior, se repite para cada uno de los cuerpos que tenga el formato, hasta que se finaliza con el último tipo de datos. Obsérvese que en este proceso, el establecimiento de relaciones, por lo general es de uno a mucho, sin embargo habrá casos en que sea de uno a uno. cuando los tipos de datos, son demasiados especiales y se desea guardar un historial de ellos. Un ejemplo podría ser el historial de fechas de actualizaciones de datos, para lo cual, entonces se hace necesario crear una entidad para guardar las fechas de diligenciamiento.

c) Identificación de Atributos Claves y Tipo de Entidades

Estos dos pasos se hacen de forma similar como se hizo en el capítulo anterior, cuando se partió de un Estudio de Caso.

Lección 13: “Análisis formato Datos Básico Proveedores”

Vamos a hacer un análisis del formato donde los proveedores actualizan la información básica, recordemos el formato:

INDUSTRIA ALIMENTICIA NENE ACTUALIZACIÓN DATOS BÁSICOS PROVEEDOR	
Fecha Diligenciamiento:	Nro:
Tipo Id.: <input type="checkbox"/> C.C <input type="checkbox"/> C.E <input type="checkbox"/> NIT Nro. Id.: _____	
Nombre Completo o Razón Social: _____	
Dir. Oficina: _____	Municipio: _____ Dpto.: _____
Correo electrónico: _____	
Teléfono Fijo 1: _____	Teléfono Fijo 2: _____ Teléfono Fijo 3: _____
Nro. Celular 1: _____	Nro. Celular 1: _____ Nro. Celular 1: _____
Datos Persona Contacto	
Tipo Id.: <input type="checkbox"/> C.C <input type="checkbox"/> C.E Nro. Id.: _____	
Nombre Completo: _____	
Nro. Teléfono Celular: _____	
Correo Electrónico: _____	
Dir. Oficina: _____	
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> Diligencia Nombre Cargo </div> <div style="width: 45%; text-align: right;"> Firma </div> </div>	

Figura 47 – Formato actualización datos proveedores

Descripción del Formato.

Lo primero que debemos hacer es conocer de que se trata cada tipo de datos, para ello debemos tener una entrevista con la persona en la organización encargada del proceso del trámite de actualización de datos básicos. Supongamos , que la explicación que nos dio para cada tipo de tados es el siguiente:

La **fecha de diligenciamiento**, es la fecha en que el proveedor hace entrega del formato diligenciado, cuando el proveedor es nuevo, debe llenar todos los datos, pero cuando es antiguo, el actualiza los datos que considera pertinente, es decir que hayan cambiado.

El **número**, es un consecutivo que el formato tienen, esto es mas para efecto de control, para que en un momento dado, que el proveedor diga que hay alguna información de datos básicos mal, nos podamos remitir al formato original y verificar qué ocasionó el error.

El **Tipo Id.**, es el tipo de identificación, la cuál puede ser Cédula de Ciudadanía, Cédula de Extranjería o Nit. Por lo tanto el **Nro. de Id.**, es el número de cédula o nit, según el caso y este número no se repite, es decir, es un número único de identificación.

El **nombre completo o razón social**, es el nombre completo, si es persona natural, y razón social, si es persona jurídica. El primero, con el número de cédula se obtiene el nombre y en la segunda con el certificado de la cámara de comercio de la respectiva ciudad donde se encuentra registrada la empresa del proveedor.

La **Dr. Oficina**, es la dirección completa de la oficina, con su respectivo **municipio** y **departamento** donde se encuentra. Cuando se trata de una empresa con muchas sucursales, les solicitamos colocar la dirección de la oficina central, donde se realizan todos los trámites administrativos del proveedor. Es de anotar, que nuestros proveedores son todos a nivel nacional.

Los tres **Teléfonos Fijos**, son los teléfonos que se encuentran en la misma dirección de la oficina, como mínimo deben tener uno. Lo anterior, debido a política de la empresa, ya que es una forma de asegurarnos de la existencia real del proveedor.

Los tres **Nro. Celular**, son los números de teléfonos móviles que el proveedor cuenta. En este punto, le recomendamos que si tienen operadores móviles diferentes, nos coloquen mejor un número de teléfono por cada operador.

A todos los proveedores, se les exige que nos dé una persona contacto, ésta necesariamente es una persona natural. Los datos que se les pide de la persona contacto son:

El **Tipo Id.**, que puede ser cédula de ciudadanía o extranjería, ya que solo son personas naturales; el **Nro de Id.**, luego es el número de la cédula.

El **nombre completo**, el cual tiene nombres y apellidos.

el **Nro. Teléfono celular**, es el número del teléfono celular, aquí exigimos que el contacto cuente con un teléfono celular, no fijo.

El **Correo Electrónico**, que es la dirección de su correo.

La **Dirección Oficina**, está se pide, porque existen muchos proveedores que pueden tener varias sucursales y por lo tanto, se requiere saber en qué oficina se encuentra la persona contacto.

Por último, el formato debe llevar el **nombre**, el **cargo** y la firma de quién lo diligenció, por parte del proveedor. Por lo general debe ser el representante legal, que siempre es una persona natural.

Procedimiento

1) Identificación de Entidades Iniciales

Como este es un formato un solo cuerpo, la entidad inicial que sale es **Datos Básicos Proveedores**, graficamente tenemos:

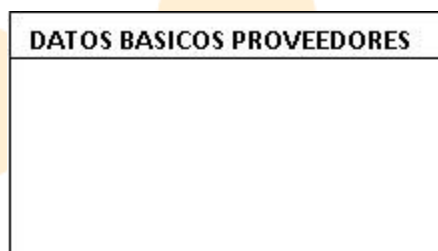


Figura 48 – Identificación entidades caso proveedores

2) Asignación de Atributos

Como se dijo, en la lección anterior, se debe seleccionar tipo de datos por tipo de datos y preguntarnos ¿Es Atributo o Entidad? así que procederemos a hacerlo para cada uno de los tipos de datos que se encuentran en el formato.

- **fecha de diligenciamiento**, observemos que no tienen sentido crear un conjunto de fechas, pues si lo creamos, la entidad **fechas diligenciamientos**, solo tendría una instancia u ocurrencia o registro por cada dato básico del proveedor. Por lo tanto, queda mejor definido como un atributo de la entidad **Datos Básicos Proveedores**.
- El **número**, es un atributo de la entidad de **Datos Básicos Proveedores**, porque no tiene sentido crear un conjunto de números consecutivos, conceptualmente no es significativo.
- El **Tipo Id.**, lo manejamos como un atributo, porque a pesar de que existen varios tipos, estos solo son tres y no vale la pena, crear una entidad que solo va a contener tres instancias u ocurrencias u objetos. Por lo tanto queda como atributo de la entidad **Datos Básicos Proveedores**.
- El **Nro de Id.**, es un atributo de la entidad **Datos Básicos Proveedores**, y es la característica mas importante, porque es la que identifica al proveedor.
- El **nombre completo o razón social**, y **Dir. Oficina**, son atributos de la entidad de **Datos Básicos Proveedores**, pues es un atributos que está relacionado con el Nro. de Id. del proveedor. Además, en el caso de la dirección de oficina, solo se pide una al proveedor.
- El **municipio** y el **departamento**, se puede pensar en tenerlos como entidades, pues en las organizaciones, a veces es necesarios generar estadísticas geográficas, y un parámetro importante son los municipios y/o departamentos. Por lo tanto, se hace necesario tener codificado tanto el municipio como el departamento. Entonces, crearemos entidades llamadas **Municipios y Departamentos**, donde cada entidad de está tiene como mínimo dos atributos **Codigo_Munic y Nombre_Munic**, así mismo para departamentos.
- **Teléfonos Fijos**, vemos que en el formato hay para colocar 3, es decir, que es un tipo de datos que se repite para cada proveedor, por lo tanto se pueden pensar en

tener un conjunto de **Teléfonos Fijos** y por lo tanto se crearía la entidad **Teléfonos Fijos** el cual estaría relacionada uno a mucho con la entidad de **Datos Básicos Proveedores**.

- El **Nro. Teléfono celular**, es el mismo análisis que el teléfono fijo, pero en vez de crear otra entidad, creamos una entidad de **Teléfonos Proveedores**, y allí guardamos tanto los teléfonos fijos, como los celulares, de tal forma que no hay necesidad de crear una entidad solo para teléfonos fijos.

Miremos gráficamente como queda todo el análisis realizado hasta aquí:

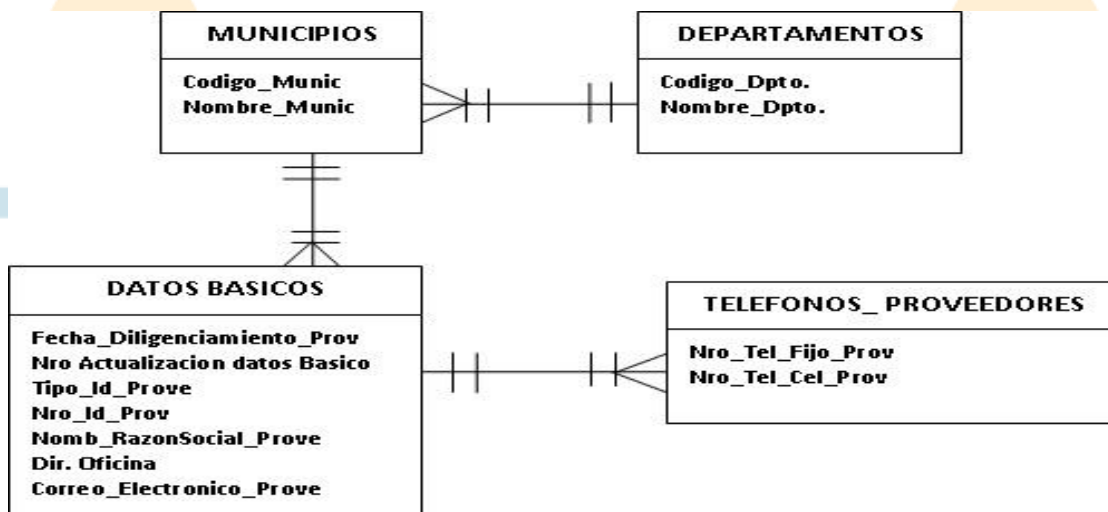


Figura 49 – Asignación de atributos caso proveedores

De acuerdo al análisis anterior, observamos que partimos de una sola entidad que se llama **Datos Básicos Proveedores**, y aparecieron Tres entidades mas, **Municipios**, **Departamentos** y **Teléfonos proveedores**.

Ahora seguimos con los datos de la persona contacto, los tipos de datos del contacto son propios de esta persona natural y por lo tanto debe crease un conjunto de éstos y relacionarlos con la entidad de **Datos Básicos Proveedores** y el diseño puede cumplir, hacia futuro, si un proveedor quiera tener varios contactos, de tal forma que saldrá una nueva entidad llamada **Contactos Proveedores**. Los atributos de está entidad sería similar a la que se hizo anteriormente. Gráficamente quedará así:

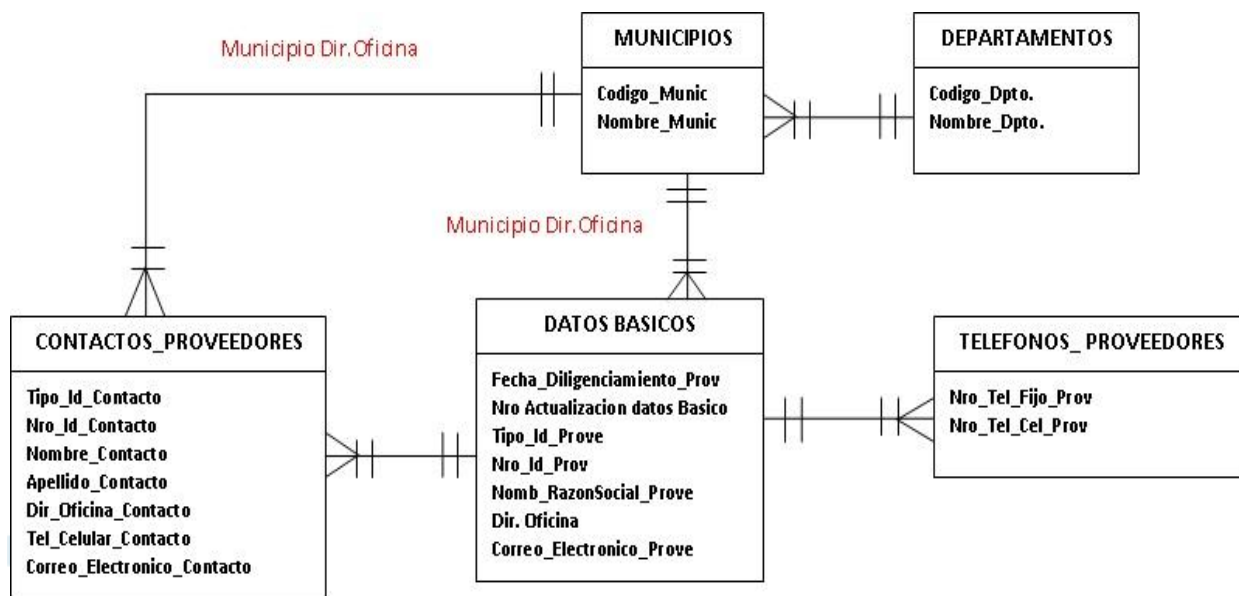


Figura 50 – Identificación de mas entidades y atributos

3) Identificación de Atributos Claves y Tipo de Entidad

En las entidades de **Municipios** y **Departamentos**, observamos que cada uno de ellos tiene un código, de tal forma, que cuando en una organización entran a codificar los datos, estos no se repiten, es decir, no puede existir dos municipios o departamentos que se llamen de forma diferentes y con el mismo código. Por lo tanto el atributo clave de estas dos entidades son: **Codigo_Munic** y **Codigo_Dpto**.

En la entidades de **Datos Basicos Proveedores** y **Contactos Proveedores**, observamos que cada uno de ellos tiene Numero de Identificación, de tal forma, que en cada uno de ellos no se pueden tener personas jurídicas y naturales con el mismo nombre, pero con el mismo número de identificación. Por lo tanto el atributo clave de estas dos entidades son: **Nro_Id.Prove** y **Nro_Id_Contacto**.

En la entidad de **Teléfonos Proveedores**, los teléfonos fijos, se colocan con todos los indicativos, de tal forma que a nivel nacional nunca se repiten, y los teléfonos celulares nunca se repite el número. Si colocamos un solo atributo llamado **Nro_Telenfono_Prove**, este nunca se va a repetir porque nunca van a suceder que un número fijo coincida con un número celular. Por lo tanto, esta entidad solo va a tener un atributo y es a su vez es el atributo clave.

Todas las situaciones anteriores, nos lleva a que todas las entidades tienen un atributo clave que las identifica totalmente y por lo tanto **todas las entidades son FUERTES**. Veamos graficamente esto:

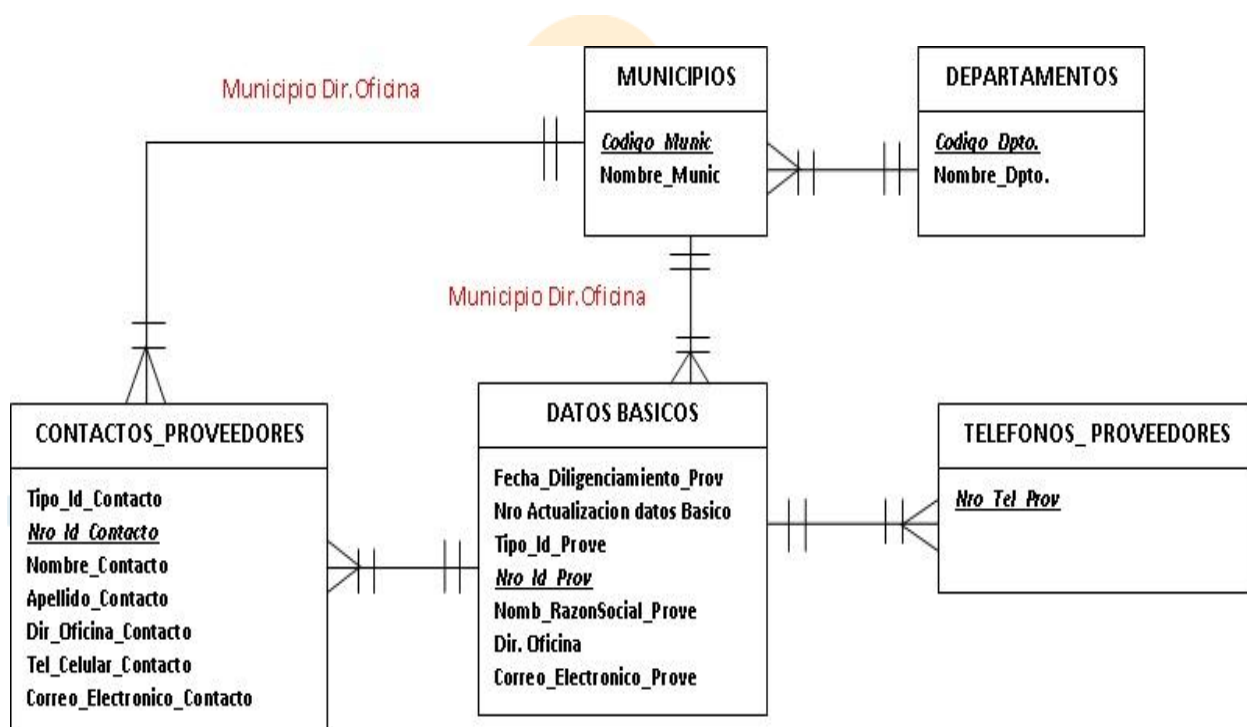


Figura 51 – Identificación de atributos claves y tipo de entidades caso proveedores

En conclusión, cuando se aborda este tipo de problemáticas dentro una organización, se debe tener permanente contacto con la o las personas que conocen a fondo el proceso que emplean el formato, con el fin de tener criterios válidos para tomar decisiones de diseño en el modelo de datos.

Lección 14: “Análisis formato Vacaciones Empleados”

Descripción formato

INDUSTRIA ALIMENTICIA NENE
SOLICITU VACACIONES EMPLEADOS

Fecha Diligenciamiento:
Nombre Dependencia Solicita:

Nro:

Tipo Vacaciones	Nro. Id. Empleado	Nombres completos Empleado	Fecha Causada de Vacaciones	Fecha Inicial Disfrute Vacaciones	Fecha Final Disfrute Vacaciones	Cantidad días Vacaciones

OBSERVACIONES:

Nombre Jefe Dependencia
Cargo

Firma

Figura 52 – Descripción formato solicitud vacaciones

En entrevista, con el Asistente de Recursos Humanos, nos comentó que este formato es diligenciado por cada dependencia de la empresa, una vez al mes, si tiene personal programado para darle vacaciones, y lo diligencia a dos copia, la original para Talento Humano y la copia para la dependencia, con firma de recibido una vez sea entregada, donde consta de los siguientes tipos de datos:

Nro. es un consecutivo que se le da al formato con el fin de control, es decir, si hay necesidad de recurrir al original en caso de errores que se presenté, entonces queda mas fácil buscar por el consecutivo.

Fecha Diligenciamiento Es la fecha en que la Dependencia entrega el formato a Talento Humano.

Nombre Dependencia Solicita Es el nombre de la dependencia; en TH tenemos codificadas las dependencias, pero el usuario solo coloca el nombre.

A continuación, el funcionario de la dependencia debe relacionar todos los empleados que van a salir a vacaciones, para ellos deben llenar:

Tipo Vacaciones, las cuales pueden ser de tres tipos **Ord**: Ordinarias, **Colect**: Colectivas o **Susp**: Suspendidas; ésta última, cuando el empleado va entrar a disfrutar vacaciones, porque fueron suspendidas sus vacaciones ordinarias anteriormente.

Nro. Id. Empleado, Se colocó el número de identificación del empleado, por política en la empresa solo se trabaja con personal que hayan cumplido la mayoría de edad.

Nombre Completo Empleado, es el primer nombre y el primer apellido del empleado.

Fechas Causadas Vacaciones, esta es la fecha inicial y final de período del año que trabajó y por lo que ya tiene ganado el derecho al disfrute de las vacaciones.

Fecha Disfrute vacaciones, esta es la fecha inicial en que sale y la fecha final en que entrar del período de disfrute de sus vacaciones.

Es de aclarar, que cuando las vacaciones son de tipos ordinarias o suspendidas la fecha de disfrute debe ser posterior a la fecha causada. Sólo, cuando las vacaciones son colectivas, puede que sea lo contrarios, es decir, que la fecha de disfrute sea menor a la fecha causada.

Cantidad días Vacaciones, es la resta que se hace de la fecha de disfrute final a la fecha de disfrute inicial.

Al final del formato se encuentran unos datos de control, en caso de presentarse problemas.

Observaciones, Allí se puede consignar alguna explicación de por qué sacar a disfrutar de vacaciones a un empleado que todavía no tiene derecho.

Nombre Jefe Dependencia, es el nombre del jefe de la dependencia; es obligatorio que el jefe de la dependencia sea el que se responsabilice de los datos consignados en el formato.

Cargo, el nombre del cargo del jefe de la dependencia; los cargos están codificados en Talentos Humano.

Firma, Por último el jefe de la dependencia debe firmar el formato diligenciado.

Procedimiento

1) Identificación de las entidades iniciales

Como el anterior formato es de dos cuerpos, tenemos que inicialmente se tienen dos entidades que son VACACIONES_EMPLEADOS y DETALLES_VACACIONES. Gráficamente tenemos:

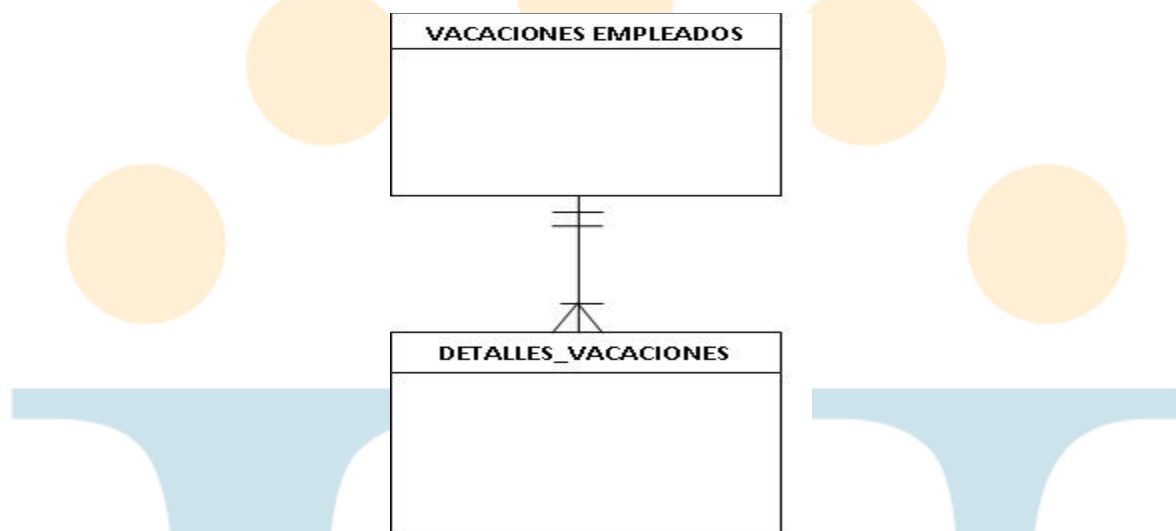


Figura 53 – Identificación de entidades centrales caso solicitud vacaciones

2) Asignación de Atributos

De acuerdo a la descripción de los tipos de datos del formato, vemos que los tipos de datos **Nro.** y **Fecha Diligenciamiento**, son atributos de la entidad VACACIONES EMPLEADOS.

El nombre de la dependencia, están codificados, por lo tanto no es un atributo de la entidad VACACIONES EMPLEADOS, sino una entidad llamada DEPENDENCIAS y cuyos atributos como mínimo son código dependencia y nombre dependencia. Por consiguiente, estas dos entidades están relacionadas y podemos decir, que UN formato de VACACIONES DE EMPLEADOS puede haber sido diligenciado por UNA sola dependencia a la vez, pero que UNA dependencia puede haber diligenciado MUCHOS formatos.

Entonces miremos el análisis anterior, como queda gráficamente.

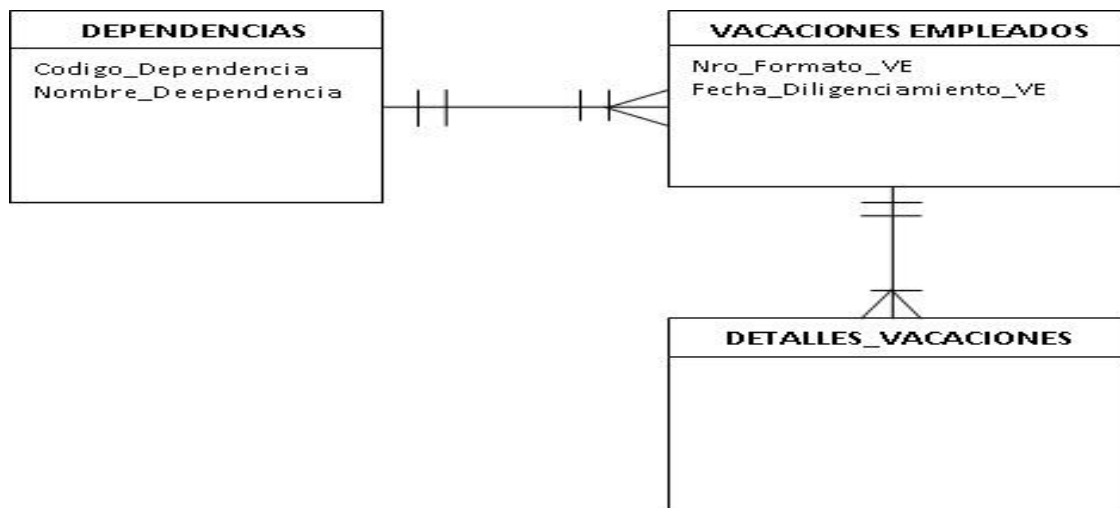


Figura 54 – Asignación de atributos caso solicitud vacaciones

Ahora vemos que los tipos de datos **Tipo de Vacaciones**, **Nro. Id. Empleado**, hasta, **Cantidad días vacaciones**, si son atributos, pertenecen a la segunda entidad DETALLES_VACACIONES. Analicemos entonces, estos atributos.

El tipo de Vacaciones, como son apenas tres tipos, es recomendable tratarlo como atributo y no como entidad, pues tenemos la certeza que esta entidad no va a tener mas de tres ocurrencias, o instancias u objetos. Por lo tanto **Tipos Vacaciones**, es un atributo de la entidad DETALLE_VACACIONES.

El Nro. Ide. Empleados y Nombres Completos, debe convertirse en una entidad llamada EMPLEADOS y cuyos atributos de esta entidad sería Nro. Id. Empleado, Nombre Empleado y Apellido Empleado, como mínimo. Como en el caso de las DEPENDENCIAS y VACACIONES_EMPLEADOS, las entidades EMPLEADOS y DETALLES_VACACIONES, están relacionadas y podemos decir, que UN empleado puede estar relacionado VARIAS en detalles vacaciones, debido a que cada vez que salga a vacaciones se creará una línea de detalle, y que UN detalle vacaciones solo puede tener UN empleado, pues si nos damos cuenta en el formato, es imposible colocar dos empleados en una misma línea de detalle.

Con respecto a los tipos de datos de las fechas y la cantidad de días, estos son atributos de la entidad DETALLES_VACACIONES, ya que las fechas, en ciertos casos excepcionales, podrían ser una entidad, como cuando se está configurando un calendario especial en la organización, pero por lo general las fechas y las cantidades son atributos y así las trataremos en este caso.

Todo el análisis anterior, gráficamente lo podemos ver así:

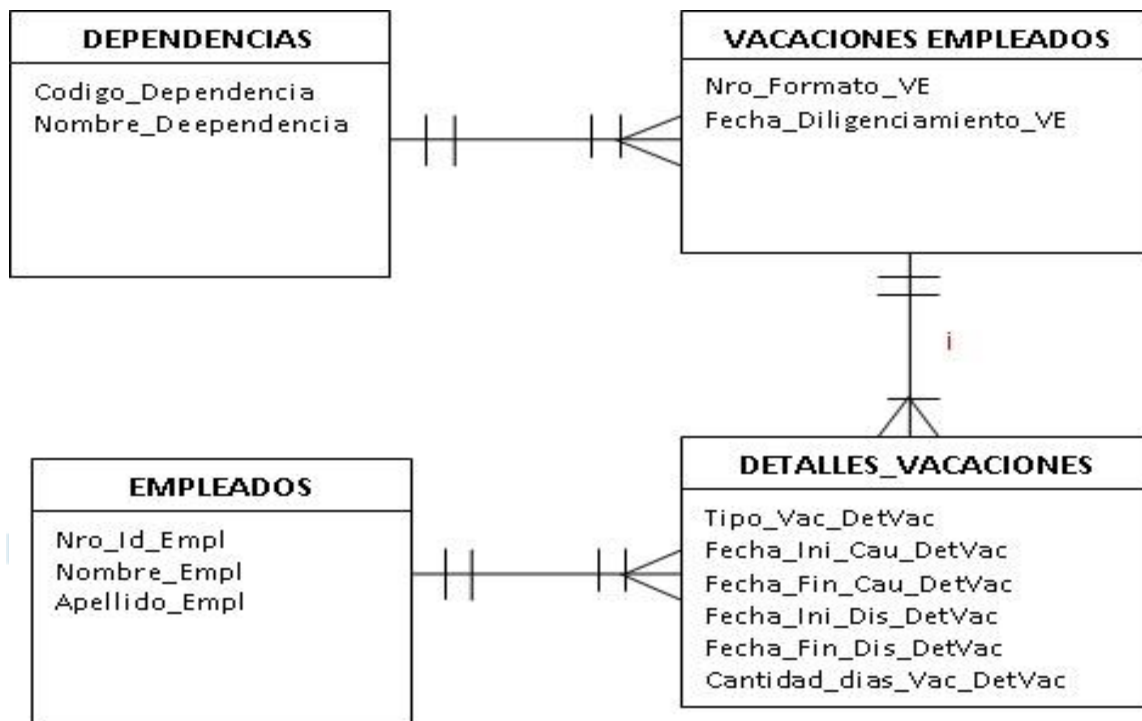


Figura 55 – Identificación de otras entidades y atributos caso solicitud vacaciones

Por último tenemos los tipos de datos observaciones, nombre jefe dependencia, cargo y firma. Como se puede observar en el diseño del formato, estos tipos hacen parte a la entidad VACACIONES_EMPLEADOS. Luego, el tipo de datos observaciones es un atributo y NO una entidad, pues en ninguna parte de la explicación del formato, se dijo que estaban codificadas.

Con respecto al nombre del jefe de la dependencia, observemos que es OTRO empleado, pero con una jerarquía mayor, por lo tanto, debemos entonces entablar un relación entre EMPLEADOS y VACACIONES_ EMPLEADOS, llamada "Jefe Dependencia". Otro aspecto, es que aquí se hace necesario conocer entonces, a qué dependencia pertenece el empleado, por lo que, debemos relacionar las entidades DEPENDENCIAS y EMPLEADOS. Por último la Firma, si se desea guardar, entonces sería un atributo de la entidad EMPLEADOS.

Gráficamente, quedaría la situación anterior así:

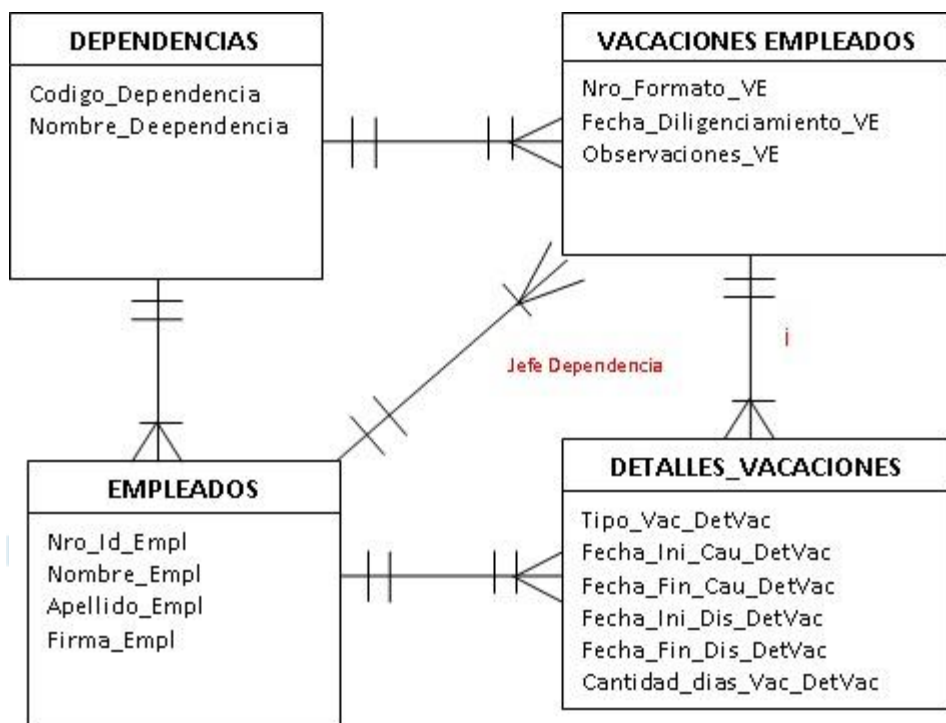


Figura 56 – Identificación de relaciones caso solicitud vacaciones

Es bueno llamar la atención con respecto a los nombres que se le dan a los atributos y entidades, obsérvese que los nombres de los atributos son SINGULARES, mientras los nombres de las entidades son PLURALES, esto con el fin de marcar la diferencia semántica entre lo que es una característica y un conjunto.

3) Identificación de Atributos Claves y Tipo de Entidad

En la entidad **DEPENDENCIAS**, existe un atributo que es el **Codigo_Dependencia**, ya habíamos dicho en casos anteriores, que los códigos son únicos, pues no tiene sentido colocarle a un mismo código, dos nombres diferentes, por lo tanto este código es el atributo clave de la entidad **DEPENDENCIAS** y ésta a su vez es una entidad FUERTE.

En la entidad **EMPLEADOS**, existe un atributo que es el **Nro_Id_Empl**. El empleado es una persona natural, por lo tanto ya se había analizado en el caso de la lección anterior, que este no se repite, es decir, no hay dos personas con cédulas iguales. Por lo tanto el número de identificación del empleado es el atributo clave y por consiguiente la entidad **EMPLEADOS** es FUERTE.

En la entidad VACACIONES_EMPLEADOS, hay un atributo que es el **Nro_Formato_VE**, que es un consecutivo, estos no se repiten y por lo tanto sería el atributo clave de esta entidad que guarda todos los datos de los formatos de vacaciones que se diligencian en la organización. Por consiguiente la entidad VACACIONES_EMPLEADOS, es una entidad FUERTE.

En la entidad DETALLES_VACACIONES, no existen atributos, como consecutivos o códigos, que pudieran servir de atributo clave; surgen entonces la sospecha, que esta entidad en primera instancia es débil, para lo cual nos apoyaríamos en su encabezado, que en este caso es VACACIONES_EMPLEADOS; si está fuera la situación, entonces miremos si combinando el número de formato de ésta entidad, con alguna fecha, puede haber repeticiones, seleccionemos la fecha_Ini_Cau, y observamos que se puede repetir, ya que pueden haber empleados que su fecha inicial causada, sea igual, por lo tanto NO nos garantiza unicidad. Entonces podemos pensar en hacerla también débil con respecto a la entidad EMPLEADOS, pues la combinación del número de formato, con la fecha inicial causada y el número de empleado, si nos garantiza la no repetición de esta situación, puesto que es imposible que en el mismo formato, para un mismo empleado se relacione la misma fecha inicial causada de sus vacaciones. Así pues, del análisis anterior, concluimos que la entidad DETALLES_VACACIONES, es una entidad ASOCIATIVA con respecto a las entidades VACACIONES_EMPLEADOS y EMPLEADOS. Gráficamente, quedaría la situación anterior así:

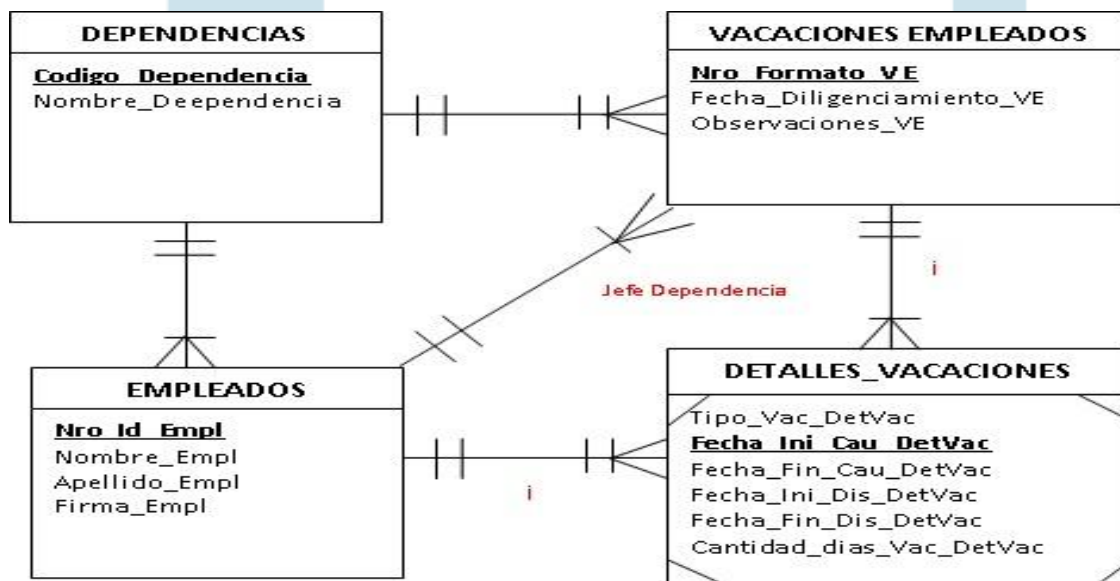


Figura 57 – Identificación de atributos claves y tipo de entidades centrales caso solicitud vacaciones

De esta forma completamos el modelo de datos lógico, del formato de dos cuerpo, que partió de dos entidades y quedó con cuatro.

Lección 15: “Análisis formato Créditos Bancarios”

Descripción del formato

BANCO PAGAME SI PUEDES SOLICITUD DE CRÉDITO				
Fecha Diligenciamiento: _____			Nro.: _____	
Oficina: _____		Municipio: _____		Dpto.: _____
Datos Persona Solicitante				
Tipo Id.	C.C.	C.E.	NIT	Nro. Id.: _____
			Lugar Exped.: _____	Fecha: _____
1er. Nombre : _____		2do. Nombre : _____		
1er. Apellido : _____		2 do. Apellido: _____		
Nro. Teléfono Celular: _____		Correo Electrónico: _____		
Direcciones:				
Tipo Dirección	Número Dirección	Municipio	Dpto.	
Residencia				
Oficina				
Familiar Allegado				
Referencias Comerciales				
Nombre completo	Teléfono Fijo	Teléfono Celular	Municipio	
Ingresos				
Rubro	valor			
Salarios				
Arriendos				
Rendimientos Financieros				
Acciones				
Egresos				
Rubro	valor			
Arriendo				
Gastos Varios				
Créditos Financieros				
Hipotecas				
Firma Nombre Cliente: _____			Firma Nombre Empleado: _____	

Figura 58 – Descripción formato caso crédito bancario

Nro. es un consecutivo que se le da al formato con el fin de control, es decir, si hay necesidad de recurrir al original en caso de errores que se presenté, entonces queda mas fácil buscar por el consecutivo.

Fecha Diligenciamiento Es la fecha en que el usuario entrega el formato en la oficina del banco.

Oficina, es el nombre de la oficina, el banco tiene codificada las oficinas.

Municipio, es el nombre del municipio donde se encuentra ubicada la oficina, el banco tiene codificada los municipios.

Departamento, es el nombre del departamento donde se encuentra ubicado el municipio, el banco tiene codificada los departamentos.

En los datos personales del solicitante tenemos:

El **Tipo Id.**, es el tipo de identificación, la cuál puede ser Cédula de Ciudadanía, Cédula de Extranjería o Nit. Por lo tanto el **Nro. de Id.**, es el número de cédula o nit, según el caso y este número no se repite, es decir, es un número único de identificación, además se les pide el municipio y la fecha en donde fue expedido el documento.

El **Primer nombre**, es el primer nombre del solicitante o la razón social.

El **Segundo nombre**, es el segundo nombre del solicitante, solo para personas naturales, sí lo tiene. Lo mismo sucede con el **Primer apellido y Segundo apellido**.

Lo mismo sucede

El **Nro. Teléfono celular**, es el número del teléfono celular.

El **Correo Electrónico**, que es la dirección de su correo.

A los solicitantes se les pide tres tipos de **direcciones**, dos de ellas son obligatoria, sí es persona natural, como son la **Dirección Residencia y Dirección Familiar allegado**. Mientras que si es persona jurídica, solo se le pide la **dirección de la oficina**. Es de anotar, que si una persona natural trabaja, debe dar la dirección de oficina. Todas las direcciones anteriores debe tener el **Municipio y el Departamento** de las direcciones. Una cosa importante, es que nos gusta tener un historial de direcciones, sobre todo la de familiares allegados y oficina, para que en un momento dado que se requiera localizar por morosidad en la deuda, podamos contar con gente o lugares, que puedan ayudar a localizarlo.

En cuanto a las **Referencias Comerciales**, se les pide como mínimo una, y debe tener el **Nombre completo**, que es la razón social, pues las referencias comerciales son personas jurídicas; **Teléfono fijo, Teléfono celular y el Municipio del teléfono fijo**.

En cuanto a los **Ingresos**, esto son los montos percibidos mensualmente, si el solicitante es una persona natural, entonces se le solicita lo que se recibe por **Salario**, si es que la persona trabaja, o los ingresos por contratos o ventas, si es que es independiente o persona jurídica; **Arriendos**, en el caso que se tenga una propiedad que tiene alquilada, entonces el canon de arrendamiento que se percibe; **Rendimientos financieros**, son los ingresos por intereses generados que percibe por cuentas de ahorro y/o certificados a término fijo; Por último,

Acciones, son los rendimientos que percibe por acciones en sociedades anónimas.

En cuanto a los **Egresos**, esto son los montos erogados mensualmente, por **Arriendos**, es el canon de arrendamiento que se paga mensualmente en caso de no vivir o funcionar en propiedad privada; **Gastos varios**, es la suma de todas las erogaciones necesarias para el funcionamiento de la empresa, si es persona natural o para la subsistencia, si es persona natural; **Créditos Financieros**, son las cuotas mensuales que se deben pagar por préstamos de consumo o productividad, realizado a entidades financieras. Por último tenemos las **Hipotecas**, que es la cuota mensual que se paga por préstamos de bienes inmuebles.

Por último, tanto la persona solicitante, como el funcionario del banco que atiende al usuario, debe colocar el nombre y firmar la solicitud. En el caso que el solicitante sea una persona jurídica, es el representante legal o el director administrativo o financiero de la organización que debe firmarla. En el caso del empleado, en el banco tenemos codificado todos los empleados, donde el número de identificación de ellos, hace de código, además del nombre, el apellido, la dirección, teléfono, etc.

Procedimiento

1) Identificación de las entidades iniciales

En este formato, en principio pudieramos pensar en entidades tales como: SOLCITUDES_CREDITOS, DATOS_BASICOS_CLIENTES, DIRECCIONES_CLIENTES, INGRESOS, EGRESOS y REFERENCIAS_COMERCIALES. Sin embargo detengámonos en la entidades DIRECCIONES, INGRESOS Y EGRESOS; éstas pueden convertirse en atributos de la entidad solicitudes, pues a pesar de haber varias direcciones, son de diferentes tipos, lo mismo sucede con los ingresos y egresos; la otra pregunta que se haría para poder tomar esta decisión, es si los tipos de datos que se encuentran en cada uno de ellos, son muy consultados, si es así, podríamos pensar en dejarlos en la entidad solicitudes, pues es menos costoso en tiempo la consulta. Por ahora podíamos pensar que los datos anteriores son muy consultados y los dejaremos como parte de la entidad de solicitudes. Siendo así, entonces tenemos que inicialmente podemos pensar en las entidades : SOLCITUDES_CREDITOS, REFERENCIAS_COMERCIALES y DATOS_BASICOS_CLIENTES. Con respecto a las relaciones, vemos que UNA solicitud puede tener MUCHAS referencias comerciales, pero a su vez UNA referencia comercial, puede ser referenciada por VARIAS solicitudes; con respecto a los datos básicos, vemos que UNA solicitud solo le pertenece a UN cliente,

mientras que UN cliente puede tener VARIAS solicitudes, según la cantidad de créditos que haya tramitado. Gráficamente tenemos:

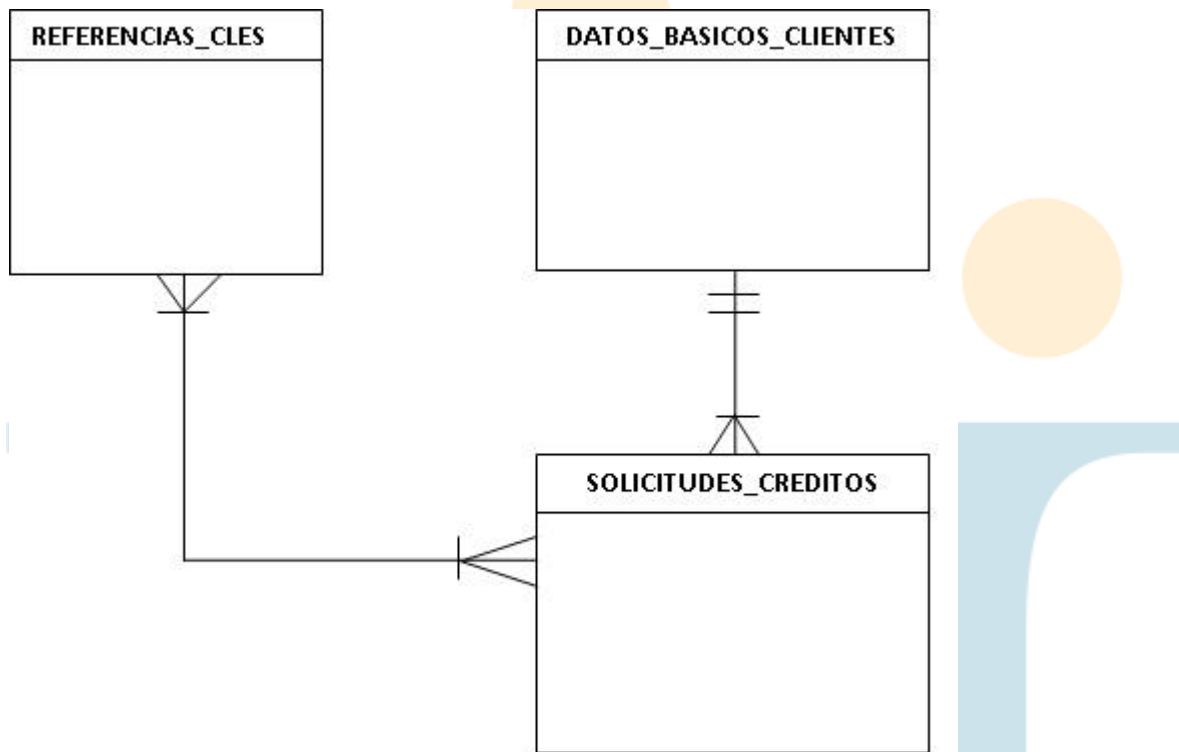


Figura 59 – Identificación entidades centrales caso crédito bancario - 1

Como podemos ver, hay una relación VARIOS a VARIOS o N-N y esto no es permitido, por lo tanto destruimos esta relación y aparece una nueva entidad, que la podemos llamar REF_CLES_SOLIC, luego gráficamente quedaría así:

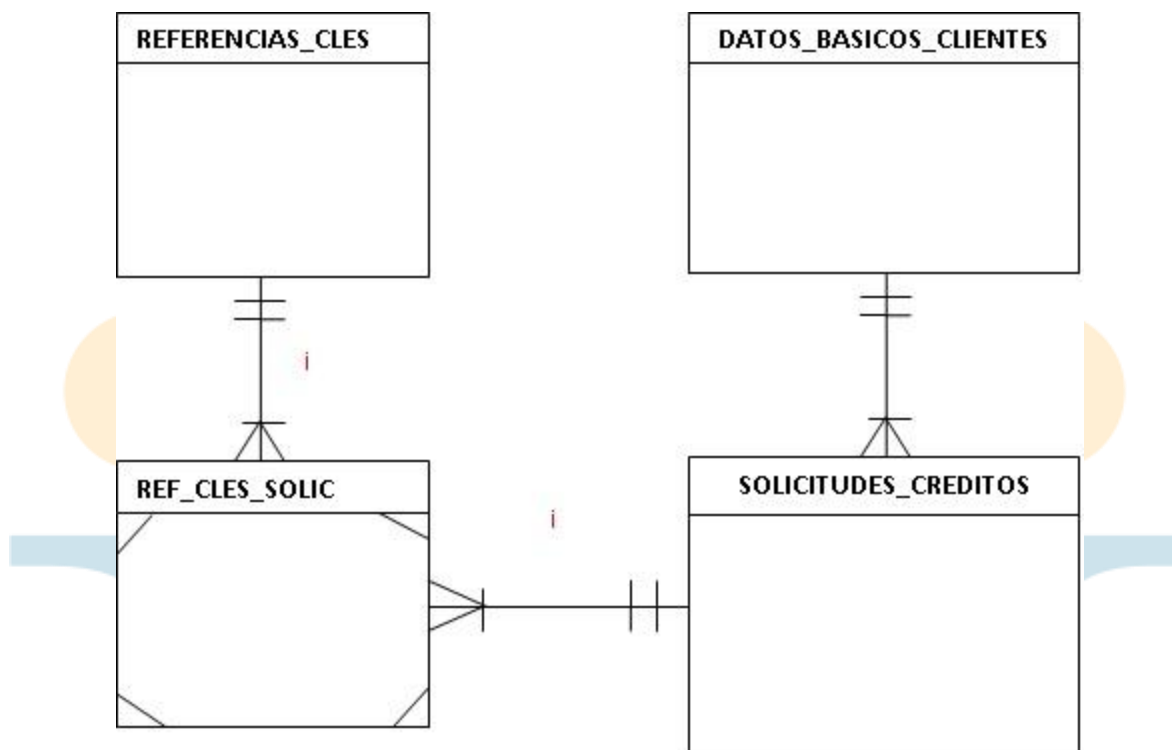


Figura 60 – Identificación entidades centrales caso crédito bancario - 2

2) Asignación de Atributos

Procediendo entonces a ver los diferentes tipos de datos, encontramos que:

La fecha de diligenciamiento y le Número son atributos de la entidad SOLICITUDES_CREDITOS. El tipo de datos Oficina, no es un atributo, sino una entidad, ya que éstas están codificadas; por lo tanto queda el entidad OFICINAS con los atributos Código_Ofic., Nombre_Ofic y dirección. Por consiguiente se establece una relación donde UNA oficina puede tener VARIAS solicitudes, mientras que UNA solicitud solo pertenece a UNA oficina. El mismo análisis anterior, lo debemos hacer para municipios y departamentos, estos están codificados y por lo tanto son entidades MUNICIPIOS y DEPARTAMENTOS: en cuanto a las relaciones, municipio se encuentra relacionado con la entidad OFICINAS, UN municipio puede tener VARIAS oficinas, mientras UNA oficina solo se encuentra ubicada en un municipio. Por otro lado, la entidad DEPARTAMENTOS, esta relacionado con la entidad MUNICIPIOS, de tal forma que UN departamento puede tener varios municipios y UN municipio solo pertenece a un departamento, lo anterior sale de la organización política que tiene el país de Colombia. Gráficamente tenemos:

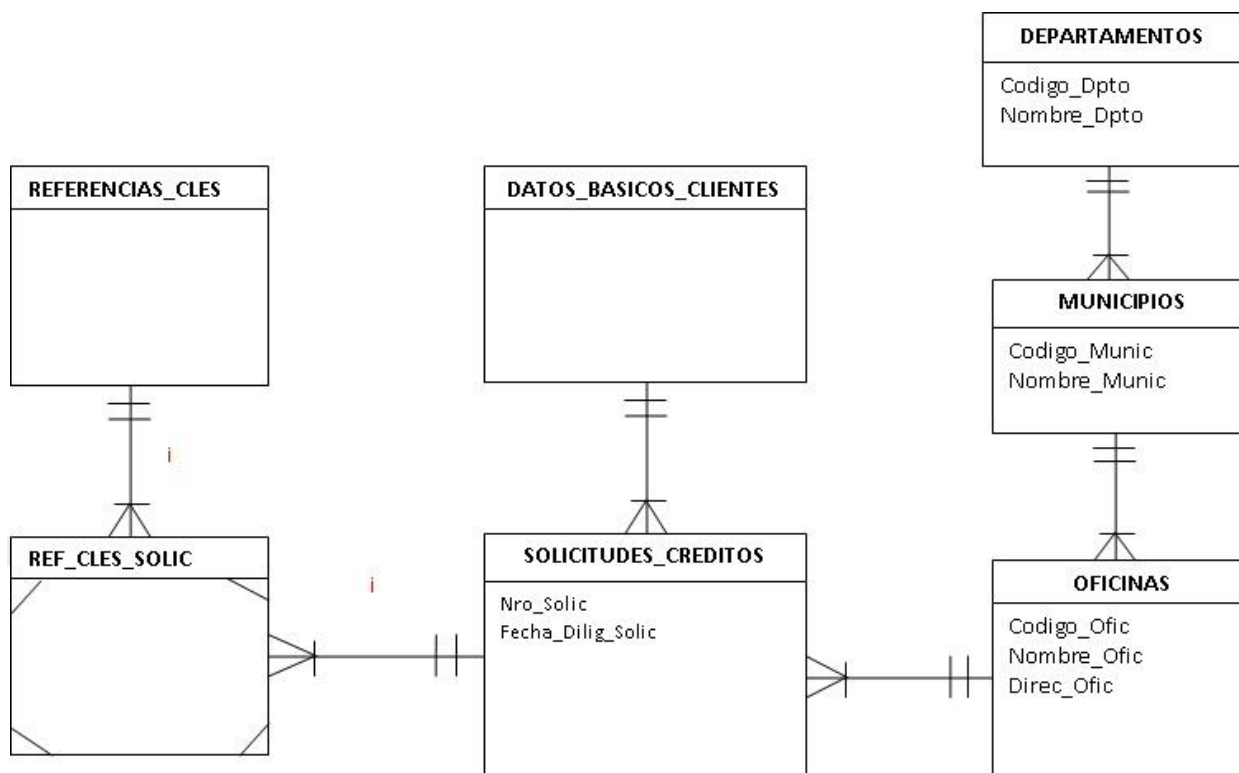


Figura 61 – Identificación atributos caso crédito bancario

Con respecto a los tipos de los datos básicos, encontramos que el tipo de identificación, el número de identificación, lugar y fecha de expedición, los nombre, los apellidos, el número teléfono celular y la dirección de correo electrónico, son atributos de la entidad **DATOS_BÁSICOS_CLIENTES**.

Con respecto a la dirección de residencia, oficina y familiar allegado, unas pueden hacer parte de la entidad **SOLICITUDES_CRÉDITOS**, como es la de oficina y familiar allegado, pues se requiere tener un historial de este tipo de datos y que mejor que guardarlo en cada solicitud, que es cuando necesariamente un cliente actualiza todas las direcciones. En cuanto a la direccion de residencia, esta puede estar en la entidad de **DATOS_BASICOS_CLIENTES**, pues es un dato muy consultado y además solo sirve la última dirección de residencia reportada.

Con respecto a los tipos de datos de ingresos y egresos, según el análisis hecho anterior, estos son atributos de la entidad **SOLICITUDES_CREDITOS**, donde se deben guardar: salarios recibidos, arriendos recibidos, Rendimientos financieros recibidos y dividendos recibidos: por otro lado egresos en canon de arrendamiento, gastos varios, cuota mensual créditos financieros y cuota mensual hipoteca.

Con respecto a las referencias comerciales, de acuerdo a la explicación dada y al análisis realizado con anterioridad, éstas son personas jurídicas, por lo tanto tiene un número de identificación, que es el NIT, que no se repite, la razón social, los teléfonos y el municipio donde se encuentra. Con respecto al municipio, como ya existe la entidad MUNICIPIOS, entonces no se pone como atributo de la entidad REFERENCIAS_COMERCIALES, sino que se establece una relación entre ellas, de tal forma que UN municipio puede tener MUCHAS empresas, mientras que UNA empresa solo puede estar en UN MUNICIPIO. Veamos gráficamente todas las situaciones anteriores:

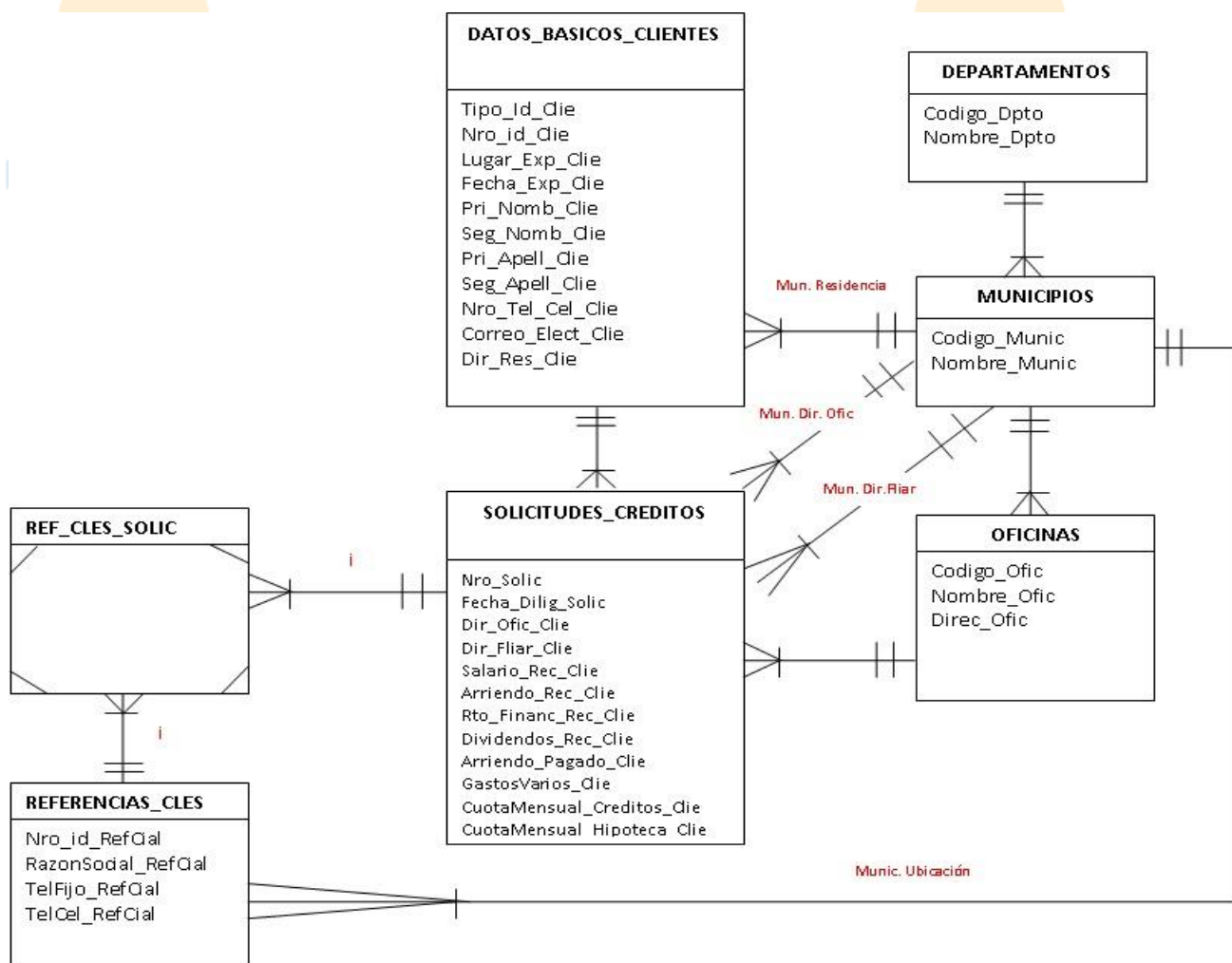


Figura 62 – Identificación de otras entidades y atributos caso crédito bancario - 1

Por último tenemos el nombre y la firma del empleado; en este caso vemos que el empleado no es un atributo de la entidad solicitudes, pues los empleados se encuentran codificados, luego entonces tenemos que crear la entidad EMPLEADOS, cuyos atributos sería, número identificación de empleados, nombre

y apellidos, dirección, teléfono. Por lo tanto, la entidad EMPLEADOS y SOLICITUDES_CREDITOS, se encuentra relacionada, donde UN empleado puede firmar VARIAS solicitudes y UNA solicitud solo puede ser firmada por UN empleado. En el caso del nombre y la firma del solicitante, podemos colocar como atributo el número de identificación y el nombre completo del representante legal, en la entidad de SOLICITUDES_CREDITOS, con el fin de saber cuál era el representante legal en el momento de solicitar el crédito en el evento de clientes jurídicos. Gráficamente, entonces tenemos:

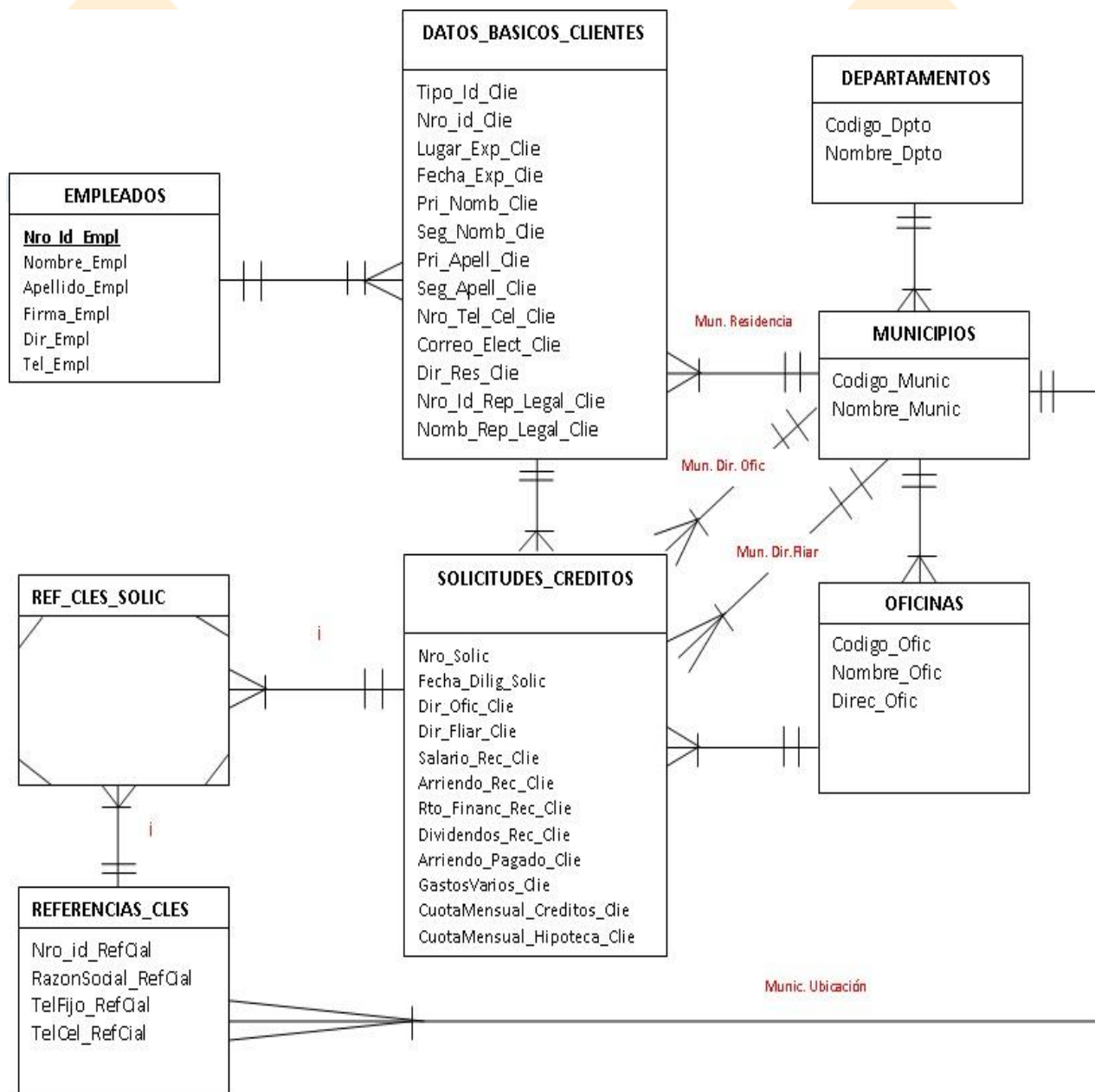


Figura 62 – Identificación de otras entidades y atributos caso crédito bancario - 2

3) Identificación de Atributos Claves y Tipo de Entidad

Observando el diagrama anterior, vemos que las entidades DEPARTAMENTOS, MUNICIPIOS y OFICINAS tienen un código, y ya en análisis de anteriores casos, anotábamos que los códigos son único y por lo tanto este sería el atributo clave en cada una de estas entidades, quedando a su vez en entidades FUERTES. Por otro lado las entidades EMPLEADOS, DATOS_BASICOS_CLIENTES y REFERENCIAS_COMERCIALES, tienen un número de identificación, que igual al caso anterior, no se repiten y se convierten en el atributo clave de cada uno de ellos por lo tanto son entidades FUERTES. En cuanto a la entidad SOLICITUDES_CREDITOS, vemos que tiene un número que es un consecutivo y que no se repite, por lo tanto, el atributo clave es este número. Por último tenemos la entidad REF_CIAL_SOLIC, que no tiene atributo, pero es ASOCIATIVA, y ya habíamos dicho en casos anteriores que este tipo de entidad es la única que puede o no tener atributos. Gráficamente tenemos:

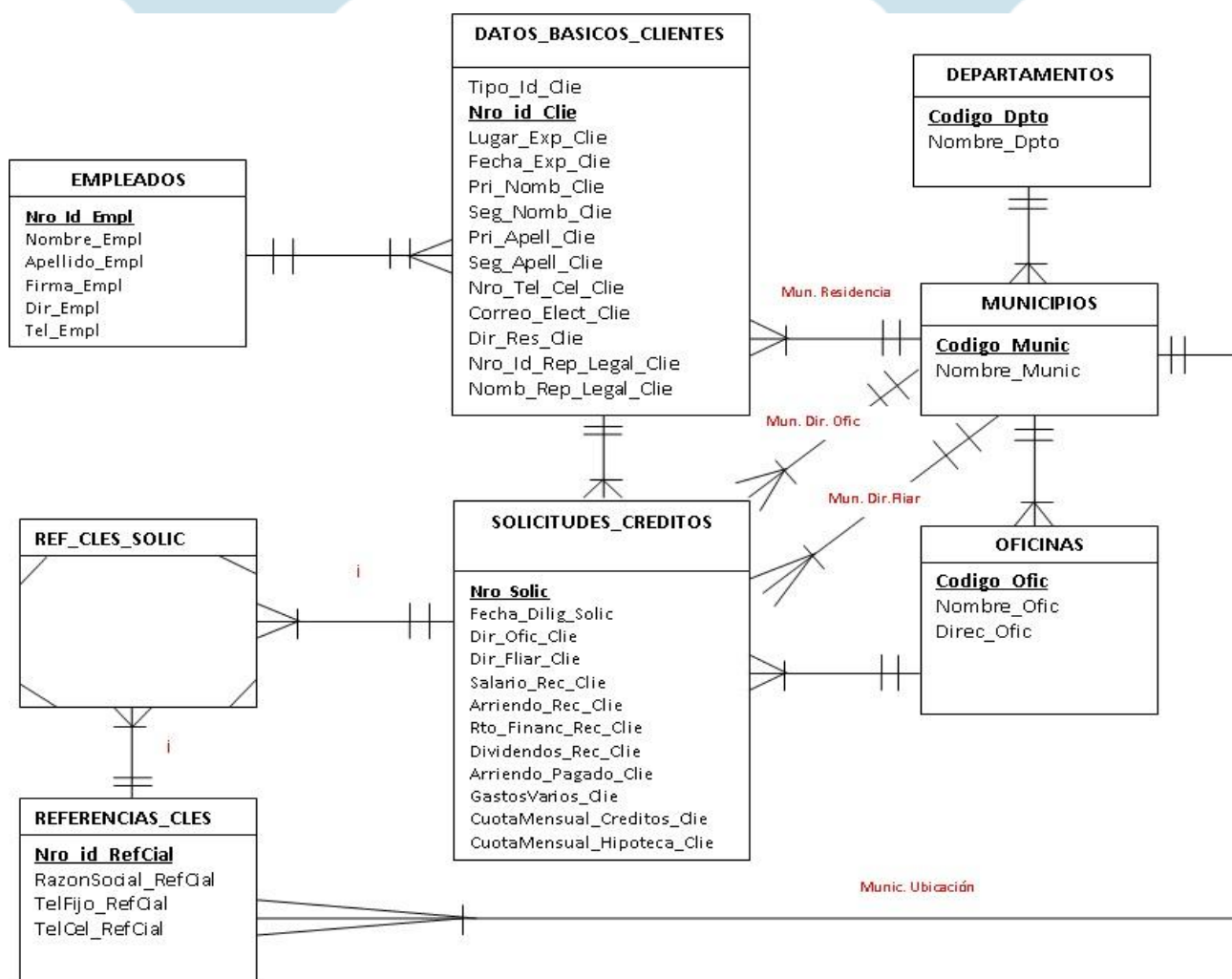


Figura 63 – Identificación de atributos claves y tipos de entidad caso crédito bancario

- Conclusiones

Podemos concluir varios aspectos con respecto a todos los casos realizados en la unidad I:

a) Los atributos en los casos no se repiten, esto es, NO PUEDE haber un atributo con el mismo nombre aunque esten en diferentes entidades.

b) Como los atributos son sustantivos, pero singular, estos pueden llegar a ser entidades, cuando se convierten en un conjunto. Lo mismo sucede con las entidades, pueden llegar a ser atributos, si es que la situación lo amerita.

c) Un atributo puede convertirse en relación, cuando establecemos que el atributo debe ser una entidad y por lo tanto debe relacionarse esta nueva entidad con la entidad a la que supuestamente pertenecía el atributo.

BIBLIOGRAFÍA

1. EFFY-OZ, Administración de sistemas de información. Pag.30
2. KENDALL & KENDALL. Análisis y diseño de sistemas. Pag.40
3. C.J. DATE. Introducción a los sistemas de bases de datos. Par. 5 ver
4. Ibidem.
5. SILBERSCHATZ, KNORT,SUDARSHAN. Fundamentos de bases de datos. Pag. 1.
5. NAVATHE, B.C. Diseño Conceptual de Bases de Datos. Un enfoque de Entidades – Interrelaciones. Editorial Addison Wesley/Dias de Santo. 1994. Estados Unidos.

CIBERGRAFÍA

Modelado de Entidad – Relación

http://books.google.com/books?hl=es&lr=&id=B_UVi51RDY4C&oi=fnd&pg=PA103&dq=modelo+de+datos+pata+de+gallo&ots=NfsqTHqOba&sig=oFlz_Fv5VsuhHGCaS7CPjWwyjrY#v=onepage&q=modelo%20de%20datos%20pata%20de%20gallo&f=false

ACTIVIDADES DE AUTOEVALUACIÓN DE LA UNIDAD

1. Analizar el siguiente estudio de caso y realizar el Diagrama Ent-Rel:

La organización HAY-FESTIVAL, lo ha contratado a usted para que le organice todos los datos de la planeación de los festivales que realiza a nivel nacional.

La organización cuenta con una serie de festivales, los cuales se encuentran codificados y además tienen su nombre, duración, tipo de arte (cine, pintura, etc.), municipio donde se realiza y mes de realización. Los tipos de arte (cine, pintura, etc.), se encuentran codificados.

Cada vez que se realiza un festival, esto son llamados certámenes, los cuales tienen un número único, temática, fecha inicio y fecha final.

También se tienen las personas que van a participar en cada uno de los certámenes y de cada una de ellas se tiene el tipo de identificación, el número de identificación, nombres, apellidos, nacionalidad y tipo de arte.

Para cada certamen se tiene una programación de los diferentes eventos y a cada uno de ellos se le asigna un número consecutivo con su respectiva fecha, hora, duración, participante, lugar, promedio de evaluación de los jurados y tipo de evento. Los lugares y los tipos de eventos, se encuentran codificados. De cada uno de ellos se requiere tener código y nombre.

Los datos que se quieren guarda del jurado es igual al de los participantes. Cada vez que un jurado califica a un participante, esto queda registrado y de ello se guarda el certamen, el evento, calificación y una observación.

2. Dado el siguiente formato, realice el modelo entidad relación.

El banco “Paga si puedes”, está interesada en que usted le modele el formato de créditos. Adicional al formato se tiene la siguiente información:

- Un municipio puede tener varias sucursales, pero una sucursal solo le pertenece a un municipio.
- Un empleado solo está adscrito a una sucursal y una sucursal tiene varios empleados.
- Una referencias tanto personal como familiar puede tenerla varios clientes.
- Un cliente puede tener varias solicitudes de crédito.
- El banco tiene codificado los municipios.
- Los tipos de documentos puede ser CC, cédula de ciudadanía o cédula de extranjería.



BANCO “PAGA SI PUEDES”

SOLICITUDO DE CREDITO

No. _____

Monto Solicitado: \$ _____ Fecha: AAAA/MM/DD Municipio: _____

Sucursal: _____ Empleado: _____

INFORMACIÓN CLIENTE

Información básica

Nro. Documento _____ Tipo: CC/CE Lugar Exp.: _____

1er Nombre: _____ 2do Nombre: _____

1er Apellido: _____ 2do Apellido: _____

Dir. Residencia: _____ Tel.: _____ Munic: _____

Dir. Oficina: _____ Tel.: _____ Munic: _____

Nro. Celular: _____

Información financiera

Ingresos mensuales: \$ _____ Egresos Mensuales: \$ _____

Referencia Familiar

Nro. Documento _____ Tipo: CC/CE

Nombres: _____ Apellidos: _____ Tel.: _____ Parent.: _____

Nro. Documento _____ Tipo: CC/CE

Nombres: _____ Apellidos: _____ Tel.: _____ Parent.: _____

Referencia Personal

Nro. Documento _____ Tipo: CC/CE

Nombres: _____ Apellidos: _____ Tel.: _____

Nro. Documento _____ Tipo: CC/CE

Nombres: _____ Apellidos: _____ Tel.: _____

Firma Cliente

Firma Empleado

UNIDAD 2.

MODELO RELACIONAL

En la unidad anterior, se veía la forma cómo se analiza y modela lógicamente una base de datos relacional fundamentada en la técnica Entidad - Relación, que fue creada por Peter Chen y cuya primera publicación fue en 1976. En esta unidad nos detendremos a profundizar sobre el diseño físico también llamado Modelo Relacional, por su creador Edgar F. Cood, quien en 1970 habló sobre este modelo.

Es bueno aclarar que la técnica Entidad - Relación, es eso, una técnica que da como resultado un modelo conceptual de las bases de datos, o a veces llamado “Modelo Lógico” mientras que las técnicas de Normalización nos dan un diseño físico de las bases de datos, es decir, queda listo para comenzar a trabajar en cualquier herramienta de bases de datos.

Si se analiza los párrafos anteriores, se observa que se habla de Modelo, Técnica y Herramientas, es bueno aclarar estos conceptos pues difieren en que el Modelo es una abstracción, la técnica es la forma cómo voy hacer el modelo y la herramienta es en qué se apoya para hacerlo. Siendo así, vemos que el Modelo es la parte cognitiva del sujeto que modela, la técnica es la forma como evidencia la parte cognitiva el sujeto y la herramienta es la forma de apoyar la técnica. Por lo tanto, de nada sirve, tener un conocimiento profundo de la herramienta, si no se tiene técnica o peor aún, si no se sabe modelar. Esto es lo que precisamente pasa en el mundo de las tecnologías de información y comunicación a los profesionales de esta disciplina, son tantas las herramientas, que le restamos importancia a la tarea del modelado, privilegiando el uso de éstas.

Por último es bueno aclarar que en la implementación de una base de datos hay tres pasos fundamentales, desde el punto de vista de quien desarrolla Sistemas de Información, apoyadas en éstas:

- 1) El análisis y el modelo lógico, que se vió en la Unidad I.
- 2) El modelo relacional o diseño físico, que es el que abarcaremos en esta unidad.
- 3) La implementación del modelo relacional o diseño físico, que abarcaremos en la Unidad III.

UNIDAD 2

Nombre de la Unidad	MODELO RELACIONAL
<p>Introducción</p>	<p>En el mundo de las organizaciones se tiene y manipula una gran cantidad de datos, que utilizando los métodos, las técnicas y las herramientas adecuadas, configuran la materia prima de la información.</p> <p>En la unidad anterior, nos centramos sobre los dos primeros aspectos. En esta unidad continuaremos profundizando sobre el segundo aspecto e inspeccionaremos otra forma de diseñar las bases de datos relacionales, la cual genera un modelo relacional. También se verá como este modelo relacional, se manipula.</p> <p>Por lo tanto, en esta unidad se presentará en el primer capítulo las reglas para pasar de un modelo lógico de datos a un modelo relacional, este último muchas veces llamado diseño físico. En el segundo capítulo abordaremos la forma de consultar los datos a un modelo relacional, a través del álgebra relacional; y en el último capítulo se verá cómo diseñar una base de datos relacional, a partir de la aplicación de las formas normales, donde directamente resulta el modelo relacional.</p> <p>Sin lugar a dudas, el estudio de las bases de datos relacionales es un tema interesante y de permanencia tecnológica, el cual es la base para estudios o aplicaciones en sistemas más complejos.</p>
<p>Justificación</p>	<p>El tema de modelo relacional es importante porque en primera instancia, se hace necesario tenerlo para implementarlo posteriormente en una herramienta y lo segundo para saber el lenguaje nativo que originó el SQL y como inicialmente se diseñaban las bases de datos sin la técnica entidad relación.</p>
<p>Intencionalidades Formativas</p>	<ul style="list-style-type: none"> • Conocer cómo se pasa un modelo lógico de datos, basado en la técnica entidad-relación, a un modelo relacional, basado en tablas y campos. • Potenciar en el estudiante la capacidad de comprensión y aprehensión de los operadores del álgebra relacional. • Desarrollar en el estudiante la habilidad para diseñar una base de datos relacional, aplicando los conceptos de normalización.
<p>Denominación de los capítulos</p>	<ul style="list-style-type: none"> • Diseño modelo relacional de datos • Álgebra relacional • Normalización

CAPÍTULO 4: “DISEÑO MODELO RELACIONAL DE DATOS”

Lección 16: “Concepto del Modelo Relacional”

Para entrar a hablar del modelo relacional o diseño físico, vamos hacer un paralelo con el modelo conceptual así:

Modelo Lógico o Modelo conceptual	Modelo Relacional o Diseño Físico	Observaciones
Entidad	Tabla	La entidad en el modelo conceptual se convierte en Tabla en el diseño físico
Relación	Llaves foráneas	Las relaciones en el modelo conceptual se convierten en llaves foráneas de la tabla, que a su vez es una columna.
Atributo	Columnas	Los atributos en el modelo conceptual se convierten en columnas de la tabla.
Atributo Clave	Llave primaria	El atributo clave se convierte en PARTE de la llave primaria de la tabla, que a su vez es una columna.
Instancias u Ocurrencias u Objetos	Filas o Tuplas	Las instancias del modelo conceptual se convierten en las filas de la tabla.

Tabla 1 – Contrastación entre Modelo lógico y Modelo relacional

De la tabla anterior se puede sacar varias conclusiones:

- 1) En el modelo conceptual, se habla de ENTIDAD, RELACIÓN Y ATRIBUTO, mientras que en el diseño físico, se habla de TABLA y COLUMNAS, porque en última instancia el atributo clave, es un atributo de la entidad y por lo tanto una columna en la tabla.
- 2) Las instancias en el modelo conceptual, que son a su vez los elementos del conjunto, se traducen en filas. Por lo tanto, una tabla que contenga filas y columnas tiende a parecerse a una hoja de Cálculo, pero donde las diferentes columnas están relacionadas y todas las filas deben tener las mismas propiedades.

En la literatura del modelo relacional, se tienen unas definiciones más formales y las cuales tomaremos dos autores, de los muchos que han escrito sobre el tema. El autor C.J. Date, que es uno de las personas que ha hecho un libro muy completo sobre la temática de Bases de Datos, que trabajó durante muchos años con IMB en el desarrollo de las tecnologías de bases de datos e Irene Luque y Otros, que crearon un libro en la Universidad de Córdoba en Colombia y que trata las temáticas con un lenguaje mas intuitivo, pero sin perder de vista el significado del concepto.

De acuerdo a C.J. Date, tenemos los siguientes términos:

Modelo Relacional: "Es una forma de ver los datos; es decir, es una receta para representar los datos (a saber mediante tablas) y la receta para manipular esas representaciones (...). En términos mas precisos, el modelo relacional se ocupa de tres aspectos de los datos: su estructura, su integridad y su manipulación" (Date, C., 1993).

Relación: Corresponde a lo que hasta ahora se ha llamado una tabla.

Tuplas: Corresponde a una fila de esa tabla y un atributo de esa columna, donde el número de tuplas se denomina cardinalidad y el numero de columnas se denomina grado.

Clave primaria: Es un identificador único para la tabla, es decir, una columna o una combinación de éstas, con las restricciones que nunca existen dos filas de la tabla con el mismo valor de esa columna o la combinación de éstas.

Dominio: Es una colección de valores, de los cuales una o mas columnas obtienen sus valores reales.

De acuerdo a Luque, Irene y Otros, maneja los siguientes términos:

Tabla: es una matriz rectangular que puede ser descrita de forma simple matemáticamente y posee las siguientes propiedades:

Cada entrada de la tabla, es decir, cada elemento de la matriz rectangular, representan a un ítem de datos elemental.

Una tabla es homogénea por columnas, es decir, todos los ítems de datos elementales de una columna (En todas las filas) son de la misma clase y, por tanto, están definidos en el mismo dominio de datos y representan una misma propiedad o característica en el dominio del problema.

Cada columna de la tabla tiene asignado un nombre único en el conjunto de columnas de esa tabla, aunque pueden existir tablas diferentes con columnas de igual nombre.

Para una tabla todas las filas son diferentes, no se admiten filas duplicadas.

Tanto las filas como las columnas pueden ser consideradas en cualquier secuencia sin afectar, por ello, ni al contenido de la información, ni a la representación semántica de la misma.

Es bueno aclarar que la segunda autora y otros, hacen una analogía de los términos de tablas como una relación, fila como una tupla y columna como un dominio, igual como lo utiliza C.J.Date, que se apega más a la terminología propia del modelo relacional. Sin embargo, a pesar de esto, semánticamente no riñen, sino que los segundos autores aclaran o hacen más intuitiva su interpretación. De allí, que ellos concluyen definiendo que una base de datos relacional es un conjunto de relaciones (tablas).

De acuerdo a los términos anteriores y a la tabla que inicialmente hace un contraste entre el modelo conceptual (lógico o semántico) y el modelo relacional (diseño físico); es bueno realizar varias observaciones debido a que en la literatura se encuentran que los diferentes autores manejan indistintamente ciertos términos así:

- 1) Atributos con campos ó columnas; pero en nuestro caso si estamos hablando del modelo conceptual o lógico nos vamos a referir siempre como atributo o característica y para el modelo relacional columna o campo. De esta forma, en el primer modelo se habla de atributo clave en el segundo modelo de campo clave.
- 2) Instancias u ocurrencias u elementos, con filas o tuplas; en nuestro caso si estamos trabajando con el modelo lógico nos vamos a referir a instancias u ocurrencias y para el modelo relacional como filas. Con frecuencia en las diferentes literaturas las instancias o filas las llaman entidad.
- 3) Las relaciones con tablas, estos dos conceptos difieren para los dos modelos. En el modelo conceptual se habla de relaciones con sus respectivas cardinalidades, como función y no se maneja el concepto de tabla; mientras que en el modelo relacional son homogéneos; nosotros utilizaremos aquí el término de tabla, en aras de que no haya confusión.

Por lo tanto desde la perspectiva del modelo relacional, gráficamente tenemos:

Bases de Datos Empleado

Empleados

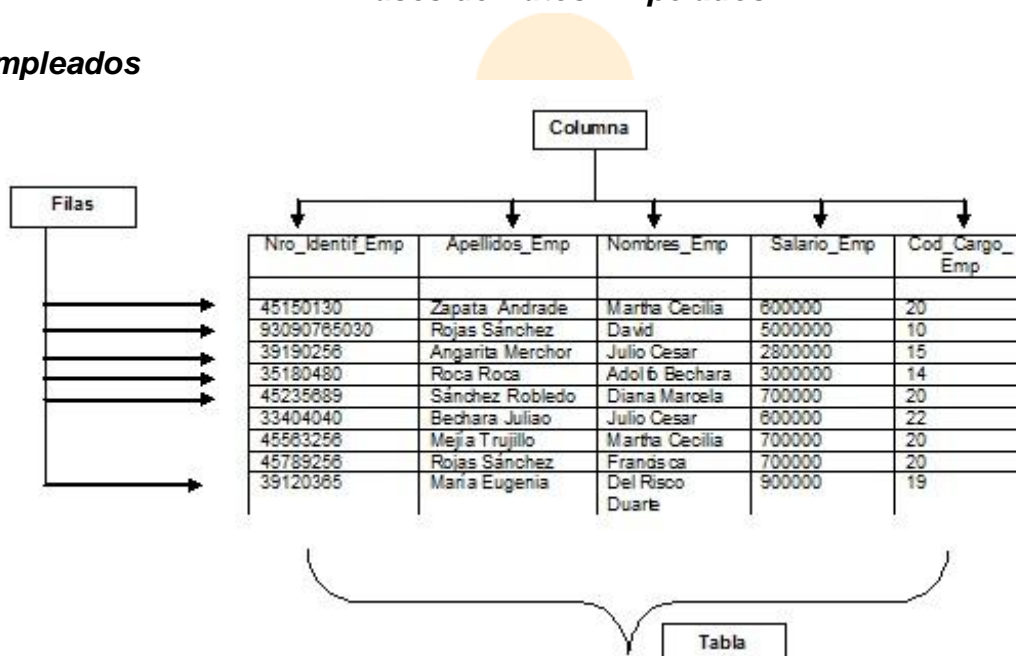


Figura 64 – (a) Tabla empleados

Cargos

Cod_Cargo	Nomb_Cargo
20	Secretaria
19	Auxiliar Contable
15	Subgerente Financiero
22	Auxiliar compras
10	Gerente General
14	Subgerente Producción

Figura 64 – (b) Tabla Cargos

En esta pequeña bases de datos se tiene dos tablas (Empleados y Cargos). En la primera se observa que las columnas son Nro. de identificación, apellidos, etc. y en la segunda es el código del cargo y el nombre. En los diferentes valores que toman las columnas de la tabla empleados, se ve también, que la única columna que no se repite es el Nro. de Identificación del Empleado, en las demás algunas se repiten; por lo tanto se ve que este sería el **campo clave**. En la tabla de cargos se ve que los valores de la columna que no se repite es el código del cargo, por lo tanto en esta tabla el campo clave sería éste. Pero también se debe observar el

campo clave de la tabla cargos con la columna de cargo_Emp de la tabla de empleados son iguales, esto es lo que se llama que la columna cargo_Emp es una **Llave Foránea** y es así como se establecen las relaciones entre las tablas en una base de datos relacional, esto también es lo que se llama la **redundancia controlada**, es decir, aparentemente se repiten los datos, pero es la única forma de establecer la relación de una tabla a otra.

Por último, miremos dos conceptos que aunque ahora no se tendrán en cuenta, mas adelante sí. El término Intención y Extensión. El primero se refiere a la cantidad de columnas que tiene una tabla y esto con el tiempo varía muy poco, tanto que se considera estático, y por lo tanto se dice que es la parte Estructural de la base de datos; el segundo se refiere a la cantidad de filas que tienen en un momento dado la tabla, este sí es variable, pues cambia en el tiempo, es decir, cada vez que se inserta o borra una fila, la extensión varía. Así de esta forma encontramos, que en la tabla EMPLEADOS, la intención es cinco (5) y la extensión es nueve (9); mientras en la tabla CARGOS, la intención es dos (2) y la extensión seis (6).

Lección 17: “Paso de un modelo lógico de datos a Modelo Relacional”

De acuerdo a los conceptos vistos, la pregunta que se debe hacer es ¿Cómo paso un modelo conceptual o lógico a un modelo relacional o diseño físico?

Para contestar esta pregunta, existen unas reglas muy sencillas que se deben entrar a aplicar:

- 1) Toda entidad es una tabla.
- 2) Todos los atributos de una entidad, son columnas o campos de la tabla. El **atributo clave** es campo clave.
- 3) Toda Relación Uno a Varios (1-N), la tabla hija hereda de la tabla padre el **campo clave**. Si la relación es fuerte (lleva i), entonces el campo clave heredado es campo clave del hijo.

Ejemplo: Suponiendo que se tiene el siguiente modelo.

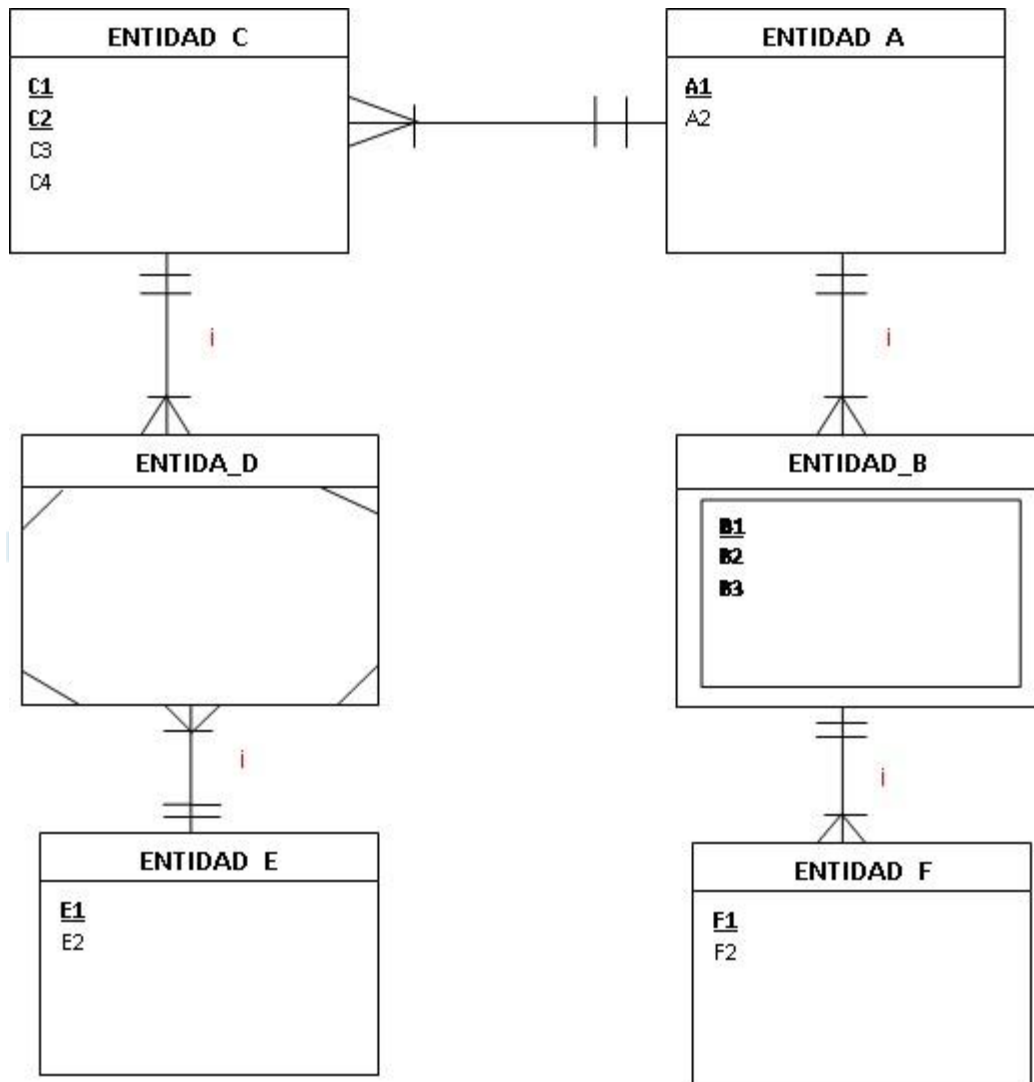


Figura 65 – Diagrama ER ficticio

Antes de empezar a aplicar las reglas anteriores, se debe tener en cuenta las siguientes recomendaciones:

- 1) Empezar a relacionar las entidades que son padres, es decir, aquellas que no tienen pata de gallina como son la ENTIDAD_A y la ENTIDAD_E.
- 2) Después se relacionan las entidades que son hijas, tienen pata de gallina con las entidades que ya fueron descritas en el punto anterior: En este caso, observamos que la ENTIDAD_B y la ENTIDAD_C tienen relación con la ENTIDAD_A.
- 3) Y el proceso se continúa hasta finalizar. Así tenemos que las entidades que faltan son hijas de entidades que ya están relacionadas; en este ejemplo la

ENTIDAD_D y la ENTIDAD_F, están relacionadas, la primera con la ENTIDAD_C y la ENTIDAD_E; la segunda con la ENTIDAD_B.

De acuerdo a lo anterior y a la aplicación de la primera regla "Toda entidad es una tabla", quedaría así:

ENTIDAD_A (

ENTIDAD_E (

ENTIDAD_B (

ENTIDAD_C (

ENTIDAD_D (

ENTIDAD_F (

Obsérvese que se abre un paréntesis, eso es con el fin de comenzar a aplicar la segunda regla "Todo atributo es un campo de la tabla y si es atributo clave, también es campo clave". Luego entonces quedaría así:

ENTIDAD_A (A1, A2)

ENTIDAD_E (E1, E2)

ENTIDAD_B (B1, B2, B3

ENTIDAD_C (C1, C2, C3, C4

ENTIDAD_D (

ENTIDAD_F (F1, F2

En las tablas anteriores, se observa que la ENTIDAD_A y la ENTIDAD_E se le cerró el paréntesis, esto debido a que son entidades que no heredan de nadie. Haciendo analogía con un árbol, son la raíz del árbol, son solo padres. Sin embargo las demás quedan con el paréntesis abierto pues las demás, en ciertas relaciones, son todas hijas de alguien.

Ahora se continúa aplicando la tercera regla "Toda relación uno a varios, la entidad hija hereda el campo clave de la tabla padre; si la relación es fuerte, entonces la hereda como campo clave también" Miremos como quedaría:

Como la tablas ENTIDAD_A y la ENTIDAD_E, ya están finalizadas, entonces seguimos con nuestro análisis con la tabla ENTIDAD_B, donde vemos que esta

entidad es hija de la ENTIDAD_A y el campo clave de la tabla ENTIDAD_A es **A1**, pero además, como la relación es fuerte (tiene i), entonces este campo clave es también campo clave de la tabla ENTIDAD_D. Gráficamente se tiene:

ENTIDAD_A (**A1**, A2)

ENTIDAD_E (**E1**, E2)

ENTIDAD_B (**B1**, B2, B3, **A1**)

Se sigue con la tabla ENTIDAD_C, se observa que esta entidad es hija de la ENTIDAD_A, por lo tanto hereda el campo clave **A1**, pero en este caso NO es parte del campo clave. Gráficamente se tiene:

ENTIDAD_A (**A1**, A2)

ENTIDAD_E (**E1**, E2)

ENTIDAD_B (**B1**, B2, B3, **A1**)

ENTIDAD_C (**C1**, **C2**, C3, C4, A1)

Se continúa con la tabla ENTIDAD_D, ésta es una entidad asociativa, por lo tanto es débil de mínimo dos entidades que en este caso es la ENTIDAD_C y la ENTIDAD_E. Aplicando la regla encontramos que hereda el campo clave **C1** y **C2** de la primera tabla y el campo clave **E1** de la segunda tabla, pero como también las relaciones que tiene con las dos entidades son fuertes (i), entonces estos campos claves a su vez son campos claves de la tabla hija. Gráficamente se tiene:

ENTIDAD_A (**A1**, A2)

ENTIDAD_E (**E1**, E2)

ENTIDAD_B (**B1**, B2, B3, **A1**)

ENTIDAD_C (**C1**, **C2**, C3, C4, A1)

ENTIDAD_D (**C1**, **C2**, **E1**)

En la tabla ENTIDAD_D, observe que era una entidad sin atributos y que al aplicar las primera dos reglas quedaban sin campos la tabla, pero ahora se observa aquí, que ya tiene tres campos y adicionalmente son campos claves; estos nos recuerda lo que se dijo la unidad I, que la única entidad que podía tener o no tener atributos era la asociativa, pero las TABLAS todas tienen que tener campos y como mínimo el CAMPO CLAVE, que en el caso de esta entidad se cumple plenamente.

Por último se entra a mirar la tabla ENTIDAD_F, esta entidad es hija de la ENTIDAD_B, por lo tanto hereda el campo clave **B1** y **A1**. Nótese que en la tercera regla se colocó en negrilla **campo clave**, para denotar que NO ES el atributo clave, porque si fuera así, entonces la ENTIDAD_F solo heredaría B1 y esto NO es correcto. Además como la relación NO es fuerte, entonces lo hereda solo como un campo mas. Tenemos entonces

ENTIDAD_A (**A1**, A2)

ENTIDAD_E (**E1**, E2)

ENTIDAD_B (**B1**, B2, B3, **A1**)

ENTIDAD_C (**C1**, **C2**, C3, C4, A1)

ENTIDAD_D (**C1**, **C2**, **E1**)

ENTIDAD_F (**F1**, F2, B1, A1)

En el ejemplo anterior, podemos sacar varias conclusiones:

- 1) Todas las relaciones Uno a Varios (1-N), se convierten en las tablas hijas, en un campo que se llama **LLAVE FORÁNEAS**.
- 2) Cuando las relaciones son fuertes, la **LLAVE FORÁNEAS**, hace parte del **campo clave** de la tabla.
- 3) La diferencia que existe entre atributo clave y campo clave, es que en el segundo pueden haber **LLAVE FORÁNEAS**.

Para finalizar miremos la literatura como define el concepto de **LLAVE FORÁNEAS**:

Segun C.J. Date, "(...) es una columna de una tabla cuyos valores están restringidos a los valores de alguna otra clave primaria". Teniendo en cuenta el ejemplo miramos la llave foránea A1 en la tabla ENTIDAD_C, sus valores están limitados a los valores que tenga el campo clave A1 de la tabla ENTIDAD_A, en otras palabras, NO puede existir un valor de A1 en la tabla ENTIDAD_C, que no se encuentre en A1 de la tabla ENTIDAD_A. Si miramos el ejemplo de la lección anterior, bases de datos EMPLEADOS, obsérvese que en la tabla de EMPLEADOS, los valores del campo Cargo_Emp ESTÁN en los valores del campo Codigo_cargo de la tabla de CARGOS. En última instancia, a través de los campos claves y las llaves foráneas, es por donde el modelo relacional GARANTIZA LA INTEGRIDAD DE LOS DATOS en la base de datos.

Lección 18: “Paso Modelo lógico Datos Básico Proveedores a Modelo Relacional”

Con base al diagrama Entidad - Relación, es decir, el modelo conceptual de este caso que se desarrolló en la unidad anterior, se entrará a realizar el paso al modelo relacional.

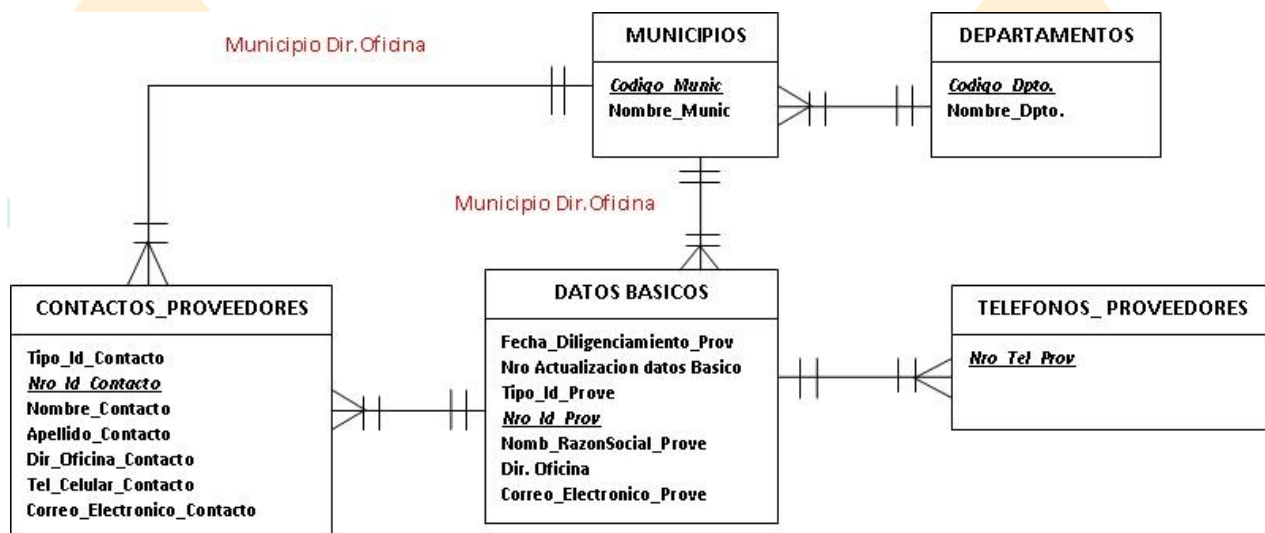


Figura 66 – Modelo lógico proveedores

Aplicación de las reglas

1) Toda Entidad es una Tabla.

acogiendo la recomendación de enunciar las tablas de acuerdo a si son padres o hijas, se observa en el diagrama, que la entidad que es solo padre, es decir, NO tienen pata de gallina, es DEPARTAMENTOS, por lo tanto está debe ser la primera tabla a relacionar. Seguimos con la entidad de MUNICIPIOS, pues esta es hija solo de DEPARTAMENTOS, que ya está en la lista. Se sigue con la entidad DATOS_BASICOS, pues es hija solo de MUNICIPIOS. Por último seguimos con las entidades CONTACTOS_PREOVEEDORES y TELEFONOS_PROVEEDORES, pues los padres de ellas ya están relacionados. Gráficamente se tiene:

DEPARTAMENTOS (

MUNICIPIOS (

DATOS_BASICOS (

CONTACTOS_PROVEEDORES (

TELEFONOS_PROVEEDORES (

2) Todo atributo es campo de la tabla. El atributo clave también es campo clave.

DEPARTAMENTOS (Codigo Dpto, Nombre_Dpto)

MUNICIPIOS (Codigo Munic, Nombre_Munic

DATOS_BASICOS (Fecha_Diligenciamiento_Prov, Nro_Actualizacion_DB,
 Tipo_Id_Prov, Nro Id Prov, Nomb_RazonSocial_Prov, Dir_Oficina,
 Correo_Electronico_Prov

CONTACTOS_PROVEEDORES (Tipo_Id_Contacto, Nro Id Contacto,
 Nombre_Contacto, Apellido_Contacto, Dir_Oficina_contacto,
 Tel_Celular_Contacto, Correo_Electronico_Contacto

TELEFONOS_PROVEEDORES (Nro_Tel_Prov

3) Toda Relación Uno a Varios (1-N)

Comenzamos con la relación que existe entre las entidades DEPARTAMENTOS y MUNICIPIOS, vemos que es de Uno a Varios (1_N), donde la tabla MUNICIPIOS hereda de la tabla padre DEPARTAMENTOS, el campo clave de su padre, que es Codigo Dpto. Además, como NO es una relación fuerte (i), entonces el campo heredado NO hace parte del campo clave de MUNICIPIOS, y vemos que la tabla de MUNICIPIOS no hereda de nadie mas. Gráficamente se tiene:

DEPARTAMENTOS (Codigo Dpto, Nombre_Dpto)

MUNICIPIOS (Codigo Munic, Nombre_Munic, Codigo_Dpto)

Se sigue con la relación que tiene la entidad DATOS_BASICOS y MUNICIPIOS, se observa que la primera es hija de esta última y la relación NO es fuerte, por lo tanto el campo clave Codigo Munic, es heredado por DATOS_BASICOS, como un campo NO clave y el cual puede tener como nombre Cod_Mun_DirOfic, dado que la relación está dada con el nombre de código del municipio de la dirección de la oficina. Las otras relaciones, no se tienen en cuenta para el análisis de

herencias, pues aquí la entidad DATOS_BASICOS, se está analizando en las relaciones donde solamente es HIJA no padre. Gráficamente se tiene:

DEPARTAMENTOS (Codigo Dpto, Nombre_Dpto)

MUNICIPIOS (Codigo Munic, Nombre_Munic, *Codigo_Dpto*)

DATOS_BASICOS (Fecha_Diligenciamiento_Prov, Nro_Actualizacion_DB, Tipo_Id_Prov, Nro Id Prov, Nomb_RazonSocial_Prov, Dir_Oficina, Correo_Electronico_Prov, *Cod_Mun_DirOfic*)

Por último se analizan las dos últimas tablas, puestos que entre estas dos no hay relaciones, y las relaciones que tienen establecidas con las demás entidades, ya están completamente definidas. Luego entonces si se ve a la entidad CONTACTOS_PROVEEDORES, miramos que tiene relación con la entidad MUNICIPIOS y DATOS_BASICOS, que las relaciones NO son fuertes y por lo tanto la tabla CONTACTOS_PROVEEDORES hereda de las dos tablas sus campos claves Codigo Munic y Nro Id Prov respectivamente como campo NO clave. Al igual que el análisis realizado en la entidad de DATOS_BASICOS, encontramos que el código municipio es de la dirección de la oficina del proveedor, por lo tanto podemos llamar a este campo *Cod_Mun_DirOfic*. Gráficamente se tiene:

DEPARTAMENTOS (Codigo Dpto, Nombre_Dpto)

MUNICIPIOS (Codigo Munic, Nombre_Munic, *Codigo_Dpto*)

DATOS_BASICOS (Fecha_Diligenciamiento_Prov, Nro_Actualizacion_DB, Tipo_Id_Prov, Nro Id Prov, Nomb_RazonSocial_Prov, Dir_Oficina, Correo_Electronico_Prov, *Cod_Mun_DirOfic*)

CONTACTOS_PROVEEDORES (Tipo_Id_Contacto, Nro Id Contacto, Nombre_Contacto, Apellido_Contacto, Dir_Oficina_contacto, Tel_Celular_Contacto, Correo_Electronico_Contacto, *Cod_Mun_DirOfic*, *Nro_Id_Prov*)

Con respecto a la entidad TELEFONOS_PROVEEDORES, solo tiene una relación con la entidad DATOS_BASICOS; luego entonces la tabla TELEFONOS_PROVEEDORES hereda de la tabla DATOS_BASICOS, el campo

clave **Nro Id Prov**, de ésta última y como la relación NO es fuerte, entonces como campo NO clave. Gráficamente se tiene:

TELEFONOS_PROVEEDORES (**Nro_Tel_Prov**, Nro_Id_Prov)

De esta forma se llega al final de la aplicación de todas las reglas, donde el resultado es un MODELO RELACIONAL o DISEÑO FÍSICO, en el cual solo tenemos las tablas y los campos; estos últimos que pueden ser solo campos, campos claves o llaves foráneas. De esta forma, vemos como en el modelo relacional, la definición de las relaciones que se dan en el modelo conceptual, son establecidas a través de los campos claves y las llaves foráneas. Miremos las tablas anteriores de forma gráfica.

BASES DE DATOS: Datos Básicos Proveedores



Figura 67 – Modelo relacional proveedores

Lección 19: “Paso Modelo lógico Órdenes de Pedidos a Modelo Relacional”

Recordemos como quedó el digrama Entidad - Relación, es decir, el modelo conceptual de este caso:

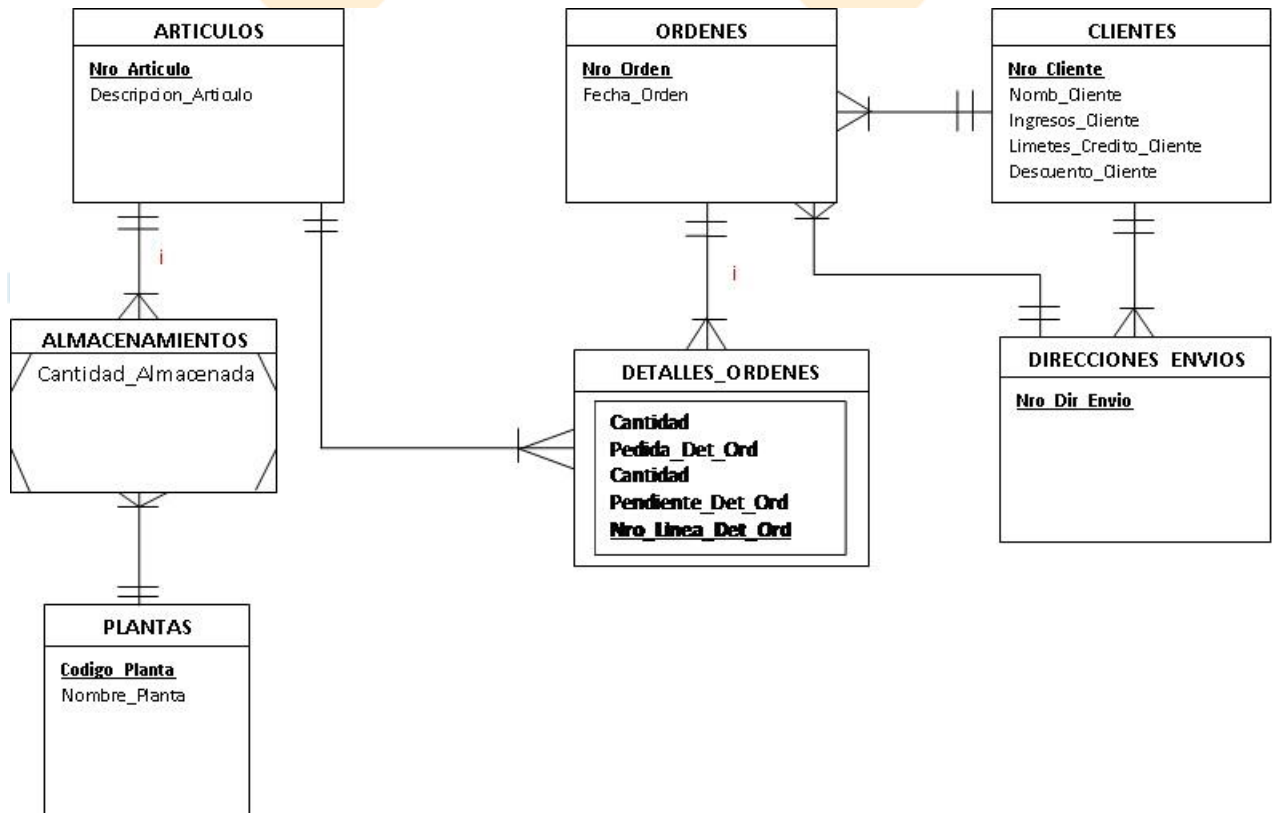


Figura 68 – Modelo lógico órdenes de pedidos

Aplicación de las reglas

1) Toda Entidad es una Tabla.

Acogiendo la recomendación de enunciar las tablas de acuerdo a si son padres o hijas, se observa en el diagrama que las entidades que solo son padres, es decir, NO tienen pata de gallina, es ARTICULOS, CLIENTES y PLANTAS por lo tanto estas deben ser las primeras tablas a relacionar. Seguimos con las entidades de DIRECCIONES_ENVÍOS Y ALMACENAMIENTOS, pues estas son hijas de CLIENTES y, ARTICULOS con PLANTAS, respectivamente, que ya están en la lista. Sigue la entidad ORDENES que dependen de CLIENTES y DIRECCIONES_ENVÍOS que ya están en la lista. Por último se tiene la entidad DETALLES_ORDENES, pues es hija de ORDENES y ARTÍCULOS. Gráficamente se tiene:

ARTICULOS(

CLIENTES(

PLANTAS (

DIRECCIONES-ENVIOS (

ALMACENAMIENTOS (

ORDENES (

DETALLES_ORDENES (

2) Todo atributo es campo de la tabla. El atributo clave también es campo clave.

ARTICULOS(**Nro_Articulo**, Descripción_Articulo

CLIENTES(**Nro_Cliente**,Nombre_Cliente,Ingresos_Cliente,Limite_Creditos_Cliente, Descuento_Cliente

PLANTAS (**Cod_Planta**, Nombre_Planta

DIRECCIONES-ENVIOS (**Nro_Dir Envio**

ALMACENAMIENTOS (Cantidad_Almacenada

ORDENES (**Nro Orden**, Fecha_Orden

DETALLES_ORDENES (Cant_Pedida_DetOrd, Cant_Pend_DetOrd,
Nro linea DetOrd

3) Toda Relación Uno a Varios (1-N)

Se inicia con las relaciones que tienen Artículos, Clientes y Plantas. Estas tres entidades son todas padres, por lo tanto NO heredan de nadie. Gráficamente se tiene:

ARTICULOS(**Nro_Articulo**, Descripción_Articulo)

CLIENTES(Nro_Cliente,Nombre_Cliente,Ingresos_Cliente,Limite_Creditos_Cliente, Descuento_Cliente)

PLANTAS (Cod_Planta, Nombre_Planta)

Ahora se sigue con la entidad de ALMACENAMIENTOS, que tiene dos relaciones, una con ARTICULOS y otra con PLANTAS, luego hereda de sus padres el campo clave. Pero como la relación que tiene con las dos entidades es FUERTE, entonces es heredada como parte de su campo clave. Gráficamente se tiene:

ARTICULOS(Nro_Articulo, Descripción_Articulo)

CLIENTES(Nro_Cliente,Nombre_Cliente,Ingresos_Cliente,Limite_Creditos_Cliente, Descuento_Cliente)

PLANTAS (Cod_Planta, Nombre_Planta)

ALMACENAMIENTOS (Cantidad_Almacenada, Nro_Articulo, Cod_Planta)

En cuanto a la entidad DIRECCIONES_ENVIOS, se observa que tiene solo una relación, donde es hija, con la entidad CLIENTES, luego hereda de su padres el campo clave. Gráficamente se tiene:

DIRECCIONES-ENVIOS (Nro_Dir Envio, Nro_Cliente)

Lección 20: “Paso Modelo lógico Créditos Bancarios a Modelo Relacional”

Teniendo en cuenta el diagrama Entidad - Relación, es decir, el modelo conceptual de este caso:

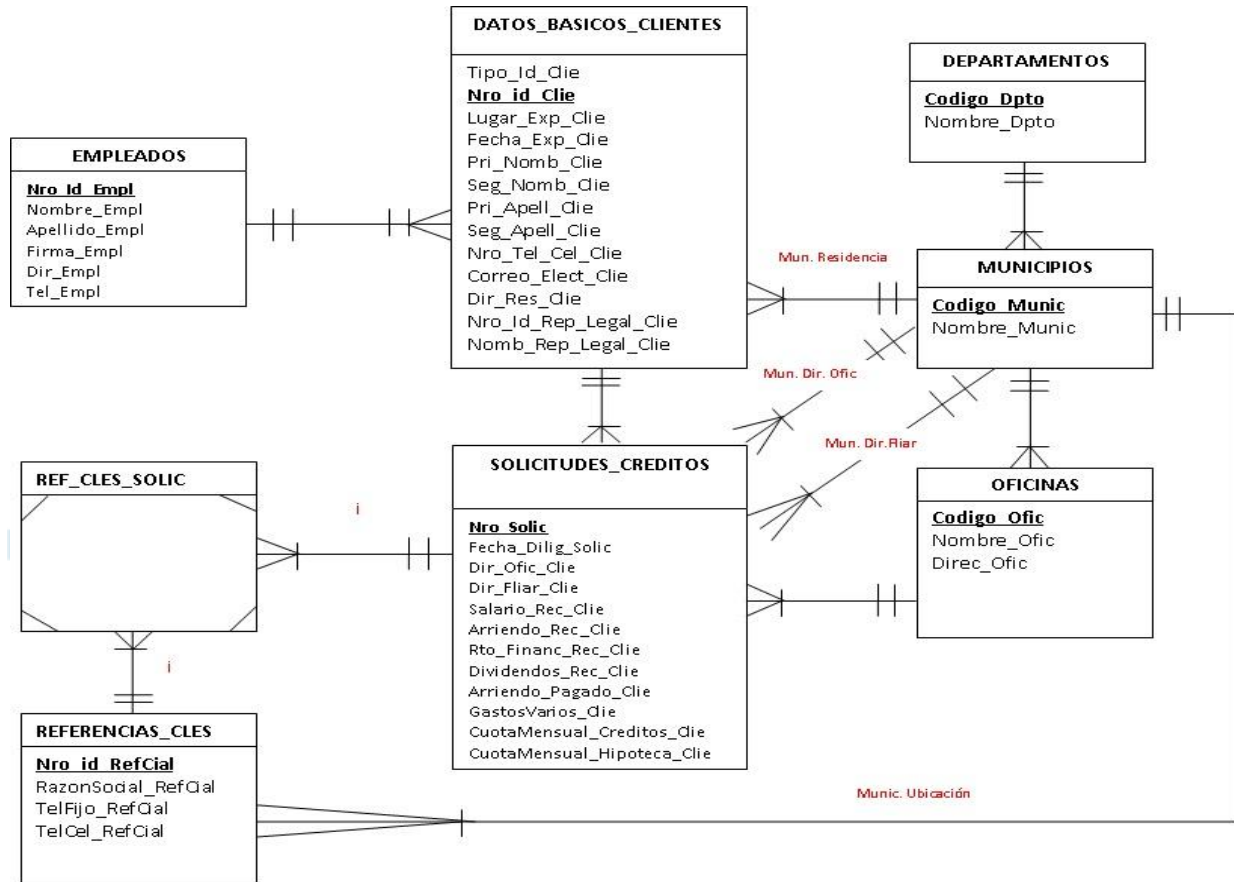


Figura 69 – Modelo lógico solicitud crédito

Aplicación de las reglas

1) Toda Entidad es una Tabla.

En este ejemplo se observa que las entidades EMPLEADOS y DEPARTAMENTOS, son solo padres, por lo tanto van de primero. Le sigue la entidad Municipios, que es hija de Departamento. Después va la entidad DATOS_BASICOS_CLIENTES, que es hija de la entidad empleados y municipios; seguida de OFICINAS y REFERENCIAS_CLES, que ambas son hijas de la entidad municipios. Por último tenemos las SOLICITUDES_DREDITOS y REF_CLES_SOLIC, donde la primera es hija de las entidades datos básicos de los clientes y oficinas y municipios y la segunda es hija de las entidades solicitudes de créditos y referencias comerciales. Gráficamente se tiene:

EMPLEADOS (

DEPARTAMENTOS (

MUNICIPIOS (

OFICINAS (

DATOS_BASICOS_CLIENTES (

REFERENCIAS COMERCIALES(

SOLICITUDES_CREDITOS (

REF_CLES_SOLIC (

2) Todo atributo es campo de la tabla. El atributo clave también es campo clave.

EMPLEADOS (**Nro_id_empl**, nombre_Empl, ...

DEPARTAMENTOS (**Codigo_Dpto**, Nombre_Dpto

MUNICIPIOS (**Codigo_Munic**, Nombre_Munic,

OFICINAS (**Codigo_Oifc**, Nombre_Ofic, ...

DATOS_BASICOS_CLIENTES (Tipo_Id_clie, **Nro_Id_Clie**,

REFERENCIAS COMERCIALES(**Nro_Id_RefCial**,

SOLICITUDES_CREDITOS (**Nro_Solic**,

REF_CLES_SOLIC (

En la última tabla, se puede observar que no hay campos, esto es solo posible en este puntos, porque la entidad no tiene atributos. Los puntos suspensivos son los demas atributos de la entidades, por efecto de abreviación, lo colocamos así.

3) Toda Relación Uno a Varios (1-N)

Se tiene que las dos primeras tablas no heredan, por lo tanto se cierra el paréntesis. La tabla municipios hereda de la tabla Departamentos su campo clave **Codigo_Dpto**, pero como NO es una relación fuerte, entonces la hereda como un atributo mas. La tabla oficina hereda solo de municipios su campo clave **Codigo_Munic**, pero como la relación NO es fuerte, entonces la hereda como un campo mas. La tabla de datos básicos de los clientes, hereda de la tabla empleados el campo clave **Nro_id_empl**, y municipios dos veces el campo clave **Codigo_Munic**, pero como en una misma tabla no se puede tener dos campos

con el mismo nombre, entonces los se debe acompañar con el nombre de la relación, de tal forma que queda **Codigo_Munic_Ofic** y **Codigo_Munic_Res**; además la hereda como un atributo mas para las tres relaciones, pues éstas no son fuertes. Gráficamente se tiene:

EMPLEADOS (**Nro_id_empl**, nombre_Empl, ...)

DEPARTAMENTOS (**Codigo_Dpto**, Nombre_Dpto)

MUNICIPIOS (**Codigo_Munic**, Nombre_Munic,, Codigo_Dpto)

OFICINAS (**Codigo_Ofic**, Nombre_Ofic, ..., Codigo_Munic)

DATOS_BASICOS_CLIENTES (Tipo_Id_clie, **Nro_Id_Clie**,, Nro_Id_Empl, Codigo_Munic_Ofic, Codigo_Munic_Ofic)

Las tablas referencias comerciales y solicitudes de créditos se dejan como ejercicio.

Por último se analiza la tabla REF_CLES_SOLIC, esta tabla hereda de las tablas de referencias comerciales y solicitudes de crédito el campo clave de cada una de ellas, como son **Nro_Id_RefCial** y **Nro_Solic**, estas son heredadas como campos claves, pues ambas relaciones son fuertes. Gráficamente se tiene:

EMPLEADOS (**Nro_id_empl**, nombre_Empl, ...)

DEPARTAMENTOS (**Codigo_Dpto**, Nombre_Dpto)

MUNICIPIOS (**Codigo_Munic**, Nombre_Munic,, Codigo_Dpto)

OFICINAS (**Codigo_Ofic**, Nombre_Ofic, ..., Codigo_Munic)

DATOS_BASICOS_CLIENTES (Tipo_Id_clie, **Nro_Id_Clie**,, Nro_Id_Empl, Codigo_Munic_Ofic, Codigo_Munic_Ofic)

REFERENCIAS COMERCIALES(**Nro_Id_RefCial**,

SOLICITUDES_CREDITOS (**Nro_Solic**,

REF_CLES_SOLIC (**Nro_Id_RefCial**, **Nro_Solic**)

CAPÍTULO 5: “ÁLGEBRA RELACIONAL”

Lección 21: “Conceptos de Álgebra Relacional”

Según Date, el álgebra relacional consiste en un conjunto de operadores de alto nivel que se aplican sobre relaciones, Codd definió ocho operadores. De acuerdo a los conceptos que hemos venido tratando, tomaremos la palabra *relaciones* como sinónimo de "Tablas". Así tenemos, que cada uno de los operadores cogen de entrada una o dos tablas y generan una nueva tabla. Es bueno precisar que las tablas deben estar completamente normalizadas. Claro está, que si el modelo entidad relación fue bien realizado y luego de aplicar las reglas para pasarlo al diseño relacional, se hicieron correctamente, no se va a tener problemas, porque como mínimo el diseño se encuentran en **Tercera Forma Normal**, que es esencial para garantizar la **Integridad** de las bases de datos para que los operadores funcionen correctamente.

Por otro lado, existe otro concepto, que tienden a confundirse con el *Álgebra Relacional* y es el concepto de *Cálculo Relacional*. Según Date, mientras el primero ofrece un conjunto explícito de operadores, que en la práctica sirven para expresar la forma de construir una tabla (Resultado de la consulta en tabla de salida) a partir de otras (Manipulación de tablas de entrada); el cálculo sólo ofrece una notación para formular la definición de las consultas deseadas.

Por lo tanto, ahora se profundiza más en los conceptos anteriores.

A. Integridad de entidades, integridad referencial y claves externas.

La restricción de integridad de entidades (Filas de la Tabla) establece que ningún valor de clave primaria puede ser nulo. Esto se debe a que el valor de la clave primaria sirve para identificar las tuplas (Filas) de una relación (Tabla), y si la clave primaria puede tener valores nulos, no podríamos identificar algunas de ellas.

La restricción de integridad referencial se especifica entre dos relaciones (Tablas), y establece que en una tupla (Fila) de una relación que haga referencia a otra relación, deberá referirse a una tupla (Fila) existente en dicha relación. Por ejemplo, el campo nombreSuc de empleados indica la sucursal en que trabaja un empleado, y su valor deberá coincidir con el valor de nombreSuc en alguna tupla (fila) de la relación (Tabla) sucursales.

Para completar la definición de integridad referencial, debemos definir el concepto de clave externa (CE); Un conjunto de campos CE del esquema de la relación R1(Tabla1) es una clave externa de R1 si satisface estas condiciones:

1. Los campos de CE tienen el mismo dominio que los campos de la clave primaria CP de otro esquema de relación R2 (Tabla2); se dice que los campos CE hacen referencia o se refieren a la relación R2.

2. Un valor de CE en una tupla (fila) t1 del estado actual de R1 es el valor de CP en alguna tupla (Fila) t2 del estado actual de R2, o bien es nulo. Si no es nulo, miremos que la tupla t1 hace referencia a la tupla t2. R1 será la relación referenciante, y R2 la relación referenciada.

En una base de datos con muchas relaciones (tablas), suele haber muchas restricciones de integridad referencial, ya que surgen de las relaciones representadas entre las diferentes filas representadas por los esquemas de relación. Una clave externa también puede hacer referencia a su propia relación (caso de supervisor y empleado). Las restricciones de integridad referencial pueden representarse gráficamente trazando un arco desde la clave externa hacia la relación a la que hace referencia, con la flecha apuntando hacia el campo o campos referenciados. Deben expresarse en el esquema de la base de datos, para que se mantengan automáticamente.

Las restricciones vistas no incluyen otra clase de restricciones llamadas restricciones de integridad semántica, tales como "un alumno no puede estar matriculado en más de 80 créditos", que necesitan un lenguaje de especificación de restricciones de propósito general. Para ello, se usan disparadores o triggers y aserciones. También hay otras restricciones llamadas restricciones de transición, para tratar con cambios de estado de la base de datos ("el sueldo de un empleado sólo puede incrementarse"). Estas restricciones se especifican con reglas de actividad y disparadores.

Teniendo en cuenta lo anterior, solo realizando un buen diseño **conceptual** y físico, podemos garantizar que las operaciones del álgebra relacional funcionen **BIEN**. De lo contrario, se reduce a una simple **colección de tablas**, donde la mayor fortaleza de la bases datos se desperdiciará, como es, la **facilidad de manipulación de los datos**.

Lección 22: “Operaciones Básica”

Las operaciones básicas en el álgebra relacional son cinco (5) y solo basta con ellas, para realizar todas las consultas que uno desea realizar a una base de datos relacional. Miremos mas detalladamente cada una de ellas.

1. Operación Selección

Esta operación consiste en seleccionar las filas de una tabla según una condición. Aquí, la tabla donde se guardan los resultados quedan con los mismos campos, pero las filas pueden ser iguales o menos, según el caso. Veamos gráficamente.

Tabla 1

A	B	C
a1	B1	C1
a2	B2	C2
a3	B3	C3
a1	B3	C3

Figura 70 – Tabla 1, operación selección

Ej: Seleccionar las filas cuyo valor de C es igual a "c3" de la tabla 1.

Tabla 2 = SELECT (Tabla 1/C="c3")

Gráficamente se tiene que el resultado es

Tabla 2

A	B	C
a3	b3	C3
a1	b3	C3

Figura 71 – Tabla 2, operación Selección

2. Operación Proyección

Esta operación consiste en proyectar ciertas columnas de la tabla. Aquí, la tabla donde se guardan los resultados quedan con las mismas filas, pero con menores columnas. En caso de que quede con las mismas columnas, entonces fue una operación sin sentido. Veamos gráficamente.

Tabla 1

A	B	C
---	---	---

a1	b1	c1
a2	b2	c2
a3	b3	c3
a1	b3	c3

Figura 72 – Tabla 1, operación Proyección

Ej: seleccionar las columnas A y C de la tabla 1.

Tabla 2= PROYECT (Tabla 1/A,C)

Tabla 2

A	C
a1	c1
a2	c2
a3	c3
a1	c3

Figura 73 – Tabla 2, operación Proyección

La utilidad de esta operación radica en que muchas de las operaciones donde intervienen dos tablas, se requieren que tengan las mismas columnas.

3. Operación Unión

Esta operación consiste en unir las filas de dos tablas. Aquí la tabla donde se guardan los resultados queda con máximo la suma de las filas de las dos tablas, en caso de haber filas repetidas, coloca solo una de ellas. Por otro lado, queda con las mismas columnas. Veamos gráficamente:

Tabla 1

A	B	C
a1	b1	c1
a2	b2	c2

a3	b3	c3
a1	b3	c3

Figura 74 – Tabla 1, operación Unión

Tabla 2

		C
a2	b1	c2
a2	b2	c2
a3	b3	c3
a1	b1	c4

Figura 75 – Tabla 2, operación Unión

Tabla 3 = Tabla 1 UNION Tabla 2

Tabla 3

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3
a1	b3	c3
a2	b1	c2
a1	b1	c4

Figura 76 – Tabla 3, operación Unión

Obsérvese que el resultado de la tabla tres dio, todas las filas de la tabla 1(4) mas dos (2) filas de la tabla 2, esto porque la filas 2 y 3 de la tabla 2 ya se encontraban en la tabla 1. También se debe observar que las columnas de la tabla 1 y la tabla 2, son las mismas (A,B y C).

4. Operación Diferencia

Esta operación coloca las filas de la primera tabla que no se encuentran en la segunda tabla. Aquí, la tabla donde se guardan los resultados queda con máximo las filas de la primera tabla, así como con las mismas columnas. Veamos gráficamente:

Tabla 1

A	B	C
a1	B1	c1
a2	B2	c2
a3	B3	c3
a1	B3	c3

Figura 77 – Tabla 1, operación Diferencia

Tabla 2

A	B	C
a2	B1	C2
a2	B2	C2
a3	B3	C3
a1	B1	C4

Figura 78 – Tabla 2, operación Diferencia

Tabla 3 = Tabla 1 MINUS Tabla 2

Tabla 3

A	B	C
a1	B1	C1
a1	B3	C3

Figura 79 – Tabla 3, operación Diferencia

Como se puede observar el resultado son las filas 1 y 4 de la tabla 1, pues éstas no se encuentran en la tabla 2, mientras que las filas 2 y 3 de la tabla 1 se encuentran en la tabla 2 y por eso no las saca en la tabla 3, es decir, la tabla de resultado.

Cabe resaltar, que en las dos operaciones anteriores (Unión y Diferencia), las columnas de las tablas que intervienen, deben ser iguales.

5. Operación Producto

Esta operación consiste en realizar un plano cartesiano entre las filas de las tablas que intervienen, de tal forma que el total de las filas de la tabla resultante es el resultado de multiplicar el número de filas de la primera tabla con el número de filas de la segunda tabla. Aquí, las columnas de las tablas que intervienen **NO TIENEN** que ser iguales. Veamos gráficamente:

Tabla 1

A	B	C
a1	B1	C1
a2	B2	C2
a3	B3	C3

Figura 80 – Tabla 1, operación Producto

Tabla 2

A	D	E
a2	D1	E2
a2	D2	E2

Figura 81 – Tabla 2, operación Producto

Tabla 3 = Tabla 1 PRODUCT Tabla 2

Tabla 3

A	B	C	A	D	E
a1	B1	c1	a2	d1	e2
a1	B1	c1	a2	d2	e2
a2	B2	c2	a2	d1	e2
a2	B2	c2	a2	d2	e2
a3	B3	c3	a2	d1	e2
a3	B3	c3	a2	d2	e2

Figura 82 – Tabla 3, operación Producto

Como se puede ver, en la tabla tres, el resultado da seis (6) filas (3x2) y seis (6) columnas (3+3).

Lección 23: “Operaciones Avanzadas”

Estas operaciones, son el resultado de combinar operaciones básicas, sin embargo, dada la utilidad que tienen, se han definido para facilitar el proceso. Estos son tres:

1. Operación de Intersección

Esta operación consiste en seleccionar las filas que están en las dos tablas que intervienen, de tal forma que el total de filas resultantes no debe ser mayor, al total de filas de la tabla que tiene menor número de filas. Aquí, las columnas de las tablas que intervienen deben ser iguales. Veamos gráficamente:

Tabla 1

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3
a1	b3	c3

Figura 83 – Tabla 1, operación Intersección

Tabla 2

A	B	C
a2	b1	c2
a2	b2	c2
a3	b3	c3
a1	b1	c4

Figura 84 – Tabla 2, operación Intersección

Tabla 3 = Tabla 1 INTERSECT Tabla 2

Tabla 3

A	B	C
a2	b2	c2
a3	b3	c3

Figura 85 – Tabla 3, operación Intersección

Obsérvese, que las dos filas resultantes en la tabla 3, están tanto en la tabla 1, como en la tabla 2.

2. Operación Reunión Natural

Esta operación saca las filas con los valores de la columna o columnas que son iguales en ambas tablas. Aquí, las columnas de las tablas no tienen que ser iguales, pero debe existir al menos una columna igual. Veamos gráficamente:

Tabla 1

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

Figura 86 – Tabla 1, operación Reunión Natural

Tabla 2

A	D	E
a2	d1	e2
a2	d2	e2
a3	d3	e3

Figura 87 – Tabla 2, operación Reunión Natural

Tabla 3 = Tabla 1 JOIN Tabla 2

Tabla 3

A	B	C	D	E
a2	b2	c2	d1	e2
a2	b2	c2	d2	e2
a3	b3	c3	d3	e3

Figura 88 – Tabla 3, operación Reunión Natural

Obsérvese, que la columna que es igual en ambas tablas es la A, entonces la comparación se realiza por ella, dando como resultado tres (3) filas.

3. Operación División

Esta operación saca la columna o columnas de la primera tabla, que no se encuentran en la segunda tabla, pero las filas de la primera tabla deben estar relacionadas, en su TOTALIDAD, con las filas de la segunda tabla. Aquí, es conveniente aclarar que las columnas de la segunda tabla debe ser un subconjunto de las columnas de la primera tabla. Gráficamente se tiene:

Tabla 1

A	B
a1	b1
a1	b2
a1	b3
a2	b2
a2	b3

Figura 89 – Tabla 1, operación División

Tabla 2

B
b1
b2
b3

Figura 90 – Tabla 2, operación División

Tabla 3 = Tabla 1 DIVISION Tabla 2

Tabla 3

A
a1

Figura 91 – Tabla 3, operación División

Obsérvese, que las columnas A , no se encuentran en la tabla 2, y con respecto a la columna B, el valor a1 es el único que esta relacionado totalmente con los valores del campo B de la tabla 2 (b1, b2 y b3) , mientras a2 NO, pues solo está relacionado con b2 y b3, haciéndole falta b1.

Como conclusión de las dos últimas lecciones, podemos decir que una de las grandes ventajas que tienen las bases de datos relacionales, es que se pudo hallar una forma matemática para manipular los datos.

Lección 24: “Análisis de ejecución de operaciones Caso Reservas”

Dada las siguientes tablas, el Gerente de la agencia de viajes desea que usted le facilite la siguiente información:

- 1) Generar toda la información de los hoteles que en este momento tienen inscritos.
- 2) Generar toda la información de los hoteles que se encuentran en el municipio cuyo código es el 02.
- 3) Generar toda la información de los hoteles, el nombre del municipio y que se encuentran en el municipio cuyo código es el 02.
- 4) Generar el número de reserva, la fecha inicial y el nombre del hotel
- 5) Generar todos los nombres de hoteles y Huéspedes, que tienen reservas.

Reservas

Rnum	Fecha_Ini	Dias duracion	Nro_Id_Hues	Cod_Hot	Nro_Hab	Fec_Res
001	23-09-10	3	39153037	H01	2-12	30-03-10
002	01-10-11	2	72044926	H02	3-01	01-01-10
003	13-08-11	2	72982340	H01	4-02	10-03-10
005	06-12-11	5	70425365	H03	1-30	06-06-10

Hoteles Huespedes

Cod_Hot	Hnombre	Cod_Mun
H01	Caribe	02
H02	Hilton	01
H03	Las Americas	02
H04	Nutibara	03
H05	Caribian	04

Nro_Id_Hues	Nombre_Hues
39153037	María Perez
72044926	Pedro Acosta
72982340	Jose Ortíz
45922444	Ana Nuñez
70425365	Rafael Leconte

Municipios

Cod_Mun	Nombre Municipio
01	Bogotá
02	Cartagena
03	Medellín
04	Barranquilla

Figura 92 – Modelo relacional caso Reserva

Siempre que se quiera dar respuesta a una solicitud de información, por medio del álgebra relacional, se recomiendan tener en cuenta tres aspectos:

- Sacar los campos que se están solicitando
- Mirar cuáles son las condiciones o restricciones solicitadas.
- Tratar de aplicar las operaciones de selección y restricción primero con el fin de disminuir registros y columna, para que las aplicación de las demás operaciones sea más rápida.
- Aplicar las demás operaciones.

Desarrollo

- Generar toda la información de los hoteles que en este momento tienen inscritos.
 - la información que nos están pidiendo es código hotel, nombre hotel y código municipio.
 - No hay ninguna restricción.
 - Aplicar la operación PROYECT, para seleccionar todos los campos de la tabla HOTELES.

Tabla 1= PROYECT (Hoteles/Cod_hot,Hnombre,Cod_Mun)

Gráficamente se tiene:

Tabla 1

Cod_Hot	Hnombre	Cod_Mun
H01	Caribe	02
H02	Hilton	01
H03	Las Americas	02
H04	Nutibara	03
H05	Caribian	04

Figura 93 – Tabla 1, consulta 1 caso reservas

2. Generar toda la información de los hoteles que se encuentran en el municipio cuyo código es el 02.

Esta consulta es similar al anterior, pero tiene una condición.

- a) la información que se está pidiendo es el código del hotel, nombre del hotel y código del municipio.
- b) La condición dada, es que se quieren los hoteles que pertenecen al municipio cuyo código es el 02.
- c) Se debe aplicar la condición, y esta se hace a través del operador SELECT, pues está nos disminuye filas.

Tabla 1 = SELECT (Hoteles/Cod_Mun=02)

Gráficamente se tiene:

Tabla 1

Cod_Hot	Hnombre	Cod_Mun
H01	Caribe	02
H03	Las Américas	02

Figura 94 – Tabla 1, consulta 2 caso reservas

Como la operación anterior solo disminuye fila y NO columnas, luego entonces se deja así, pues es la información que está pidiendo el gerente.

3. Generar toda la información de los hoteles, el nombre del municipio y que se encuentran en el municipio cuyo código es el 02.

Esta consulta es parecida al anterior, pero adicionalmente nos están pidiendo el nombre del municipio, como este no está en la tablas de Hoteles, entonces necesitaremos consultar la tabla de municipios.

- a) Están pidiendo de la tabla hoteles, Código del hotel, nombre del hotel y código municipio. En la tabla municipios necesitamos tomar nombre municipios.
- b) La restricción necesitan solo los hoteles que estén en el municipio cuyo código es el 02. Por lo tanto, lo primero que se aplica es la operación SELECT.

Tabla 1 = SELECT(Hoteles/Cod_Mun=02)

Gráficamente se tiene:

Tabla 1

Cod_Hot	Hnombre	Cod_Mun
H01	Caribe	02
H03	Las Americas	02

Figura 95 – Tabla 1, consulta 3 caso reservas

Ahora se necesita hacer una operación de selección en la tabla Municipios, cuyo código también sean el 02

Tabla 2 = SELECT (Municipios/Cod_Mun=02)

Gráficamente se tiene:

Tabla 2

Cod_Mun	Nombre Municipio
02	Cartagena

Figura 96 – Tabla 2, consulta 3 caso reservas

- c) A hora se necesita hacer un producto entre las dos tablas resultantes Tabla 1 y Tabla 2

Tabla 3 = Tabla 1 PRODUCT Tabla 2

Tabla 3

Cod_Hot	Hnombre	Cod_Mun	Nombre_Municipio
H01	Caribe	02	Cartagena
H03	Las Americas	02	Cartagena

Figura 97 – Tabla 3, consulta 3 caso reservas

Obsérvese que en este ejercicio se aplica primero los operadores UNARIOS, en este caso SELECT, con el fin de restringir el dominio y que cuando se apliquen las operaciones BINARIAS, en este caso PRODUCT, se ejecute en forma más eficiente esta última operación.

La consulta anterior se puede hacer utilizando operadores avanzados, para lo cual en dos pasos se genera la misma información, así:

Tabla 1 = SELECT (Municipios/Cod_Mun=02)

Gráficamente se tiene:

Tabla 2

Cod_Mun	Nombre Municipio
02	Cartagena

Figura 98 – Tabla 1, consulta 3 caso reservas -1

A hora se requiere hacer un JOIN entre las tablas Tabla 1 y Hoteles.

Tabla 2 = Tabla 1 JOIN Hoteles

Tabla 2

Cod_Hot	Hnombre	Cod_Mun	Nombre_Municipio
H01	Caribe	02	Cartagena
H03	Las Americas	02	Cartagena

Figura 99 – Tabla 2, consulta 3 caso reservas - 2

Obsérvese que el resultado es exactamente igual.

4. Generar el número de reserva, la fecha inicial y el nombre del hotel

Como se vio anteriormente, cuando se utilizan los operadores avanzados se reducen los pasos para generar consultas. Cuando una consulta requiere ser sacada de dos tablas, se hace indispensable que ellas se encuentren relacionadas

entre sí, pero sino es así, entonces hay que observar que relaciones *indirectas* tiene de tal forma que las podamos relacionar, sin esta condición, es imposible generar información *válida*. Ahora veamos el desarrollo de la consulta.

- a) Nos están pidiendo de la tabla *hoteles*, nombre del hotel. En la tabla *reservas* el número de la reserva y la fecha inicial.
- b) en esta consulta no hay restricción.

Si se compara la tabla *reservas* y *hoteles*, se observa que entre las dos hay una relación que se plantea a través del *código del hotel*. Luego entonces aquí se puede aplicar el operador JOIN. Sin embargo, antes de hacer esto, se debe mirar que la tabla de *reservas* tiene mucho mas información de la que están pidiendo, luego entonces, se debe pensar en reducir la tabla *reservas* a los campos estrictamente necesarios para poder aplicar el operador JOIN. Para ello, se utiliza primero la operación PROYECT, que genera solo las columnas que se necesita, así:

Tabla 1 = PROYECT (Reservas/ Rnum,Fecha_Ini,Cod_Hot)

Gráficamente queda así

Tabla 1

Rnum	Fecha_Ini	Cod_Hot
001	23-09-06	H01
002	01-10-06	H02
003	13-08-06	H01
005	06-12-06	H03

Figura 100 – Tabla 1, consulta 4 caso reservas

Ahora se aplica el operador JOIN entre las tablas (Tabla1 y Hoteles)

Tabla 2 = Tabla 1 JOIN Hoteles

Gráficamente se tiene:

Tabla 2

Rnum	Fecha_Ini	Cod_Hot	Nombre_Municipio
001	23-09-10	H01	Caribe
002	01-10-11	H02	Hilton
003	13-08-11	H01	Caribe
005	06-12-11	H03	Las Américas

Figura 101 – Tabla 2, consulta 4 caso reservas

Obsérvese que solo queda pendiente remover la columna código hotel, pues esta NO la piden en la consulta. Para lo cual se utiliza nuevamente el operador de PROYECT.

Tabla 3 = Proyect (tabla 2/Rnum,Fecha_Ini,Nombre_Municipio)

Tabla 3

Rnum	Fecha_Ini	Nombre_Municipio
001	23-09-10	Caribe
002	01-10-11	Hilton
003	13-08-11	Caribe
005	06-12-11	Las Américas

Figura 102 – Tabla 3, consulta 4 caso reservas

5. Generar todos los nombres de hoteles y Huéspedes, que tienen reservas.

En esta consulta, se pide los nombres de los hoteles que están en la tabla de hoteles y los nombres de los huéspedes que están en la tabla de huéspedes, pero mirando la tabla de hoteles y huéspedes, entre ellas no hay relación directa. Luego entonces se debe pensar como relacionarla, y para ellos es necesario mirar las relaciones indirectas. Obsérvese entonces, que la tabla de reservas es una tabla común a hoteles y huéspedes, por lo tanto es a través de ésta que se debe diseñar la consulta.

Están pidiendo de la tabla *hoteles*, nombre del hotel. En la tabla *Huéspedes* el nombre del huésped.

- a) La restricción es que necesitan los nombres de los huéspedes que tienen reservas, luego entonces todos los huéspedes que aparecen en la tabla de reservas, tienen reservas.

Se inicia haciendo la proyección de los campos código hotel y número de identificación del huésped de la tabla de reservas, así:

Tabla 1 = PROYECT(Reservas/Cod_Hot,Nro_Id_Hues)

Gráficamente se tiene:

Tabla 1

Nro_Id_Hues	Cod_Hot
39153037	H01
72044926	H02
72982340	H01
70425365	H03

Figura 103 – Tabla 1, consulta 5 caso reservas

En esta tabla se encuentran los códigos de los huéspedes que tienen reservas, con su respectivo código de hotel, en el que reservó.

Ahora se aplica la operación JOIN entre la tabla 1 y la de Huéspedes, para tener el nombre de éste, así:

Tabla 2 = Tabla 1 JOIN Huéspedes

Gráficamente se tiene:

Tabla 2

Nro_Id_Hues	Cod_Hot	Nombre_Hues
39153037	H01	María Pérez
72044926	H02	Pedro Acosta
72982340	H01	José Ortiz
70425365	H03	Rafael Leconte

Figura 104 – Tabla 2, consulta 5 caso reservas

Ya lo único que hace falta es tener el nombre del hotel, para ello aplicamos nuevamente la operación JOIN entre la tabla 2 y Hoteles.

Tabla 3 = Tabla 2 JOIN Hoteles

Gráficamente se tiene:

Tabla 3

Nro_Id_Hues	Cod_Hot	Nombre_Hues	Hnombre
39153037	H01	María Pérez	Caribe
72044926	H02	Pedro Acosta	Hilton
72982340	H01	José Ortiz	Caribe
70425365	H03	Rafael Leconte	Las Americas

Figura 105 – Tabla 3, consulta 5 caso reservas

Por último, se debe generar la información que están solicitando, como es el Nombre del huésped y el nombre del Hotel. Para ello, se utiliza el operador de proyección.

Tabla 4 = PROYECT(Tabla 3/Nombre_Hues,Hnombre)

Gráficamente se tiene:

Tabla 4

Nombre_Hues	Hnombre
María Pérez	Caribe
Pedro Acosta	Hilton
José Ortiz	Caribe
Rafael Leconte	Las Americas

Figura 106 – Tabla 4, consulta 5 caso reservas

Lección 25: “Análisis de Ejecución de operaciones Caso de Certámenes”

Dada las siguientes tablas, el Gerente de una organización de Certámenes nacionales e internacionales, desea que usted le genere la siguiente información:

CERTAMENES

Cnum	Fecha_Ini	Fecha_Fin	Cnombre
100	23-03-06	05-04-06	Cine culto
101	01-01-04	05-01-04	Pintura Infantil
102	13-03-05	13-03-05	Cine junior
103	06-06-04	10-06-04	Musica rock

JURADOS

Nro_Id_Ju	Jnombre
101010	Daniel Rojas
202020	Blanca Lemaitre
303030	María Sanchez
70425	Rafael leconte

PARTICIPANTES

Nro_Id_Part	Nombre_Partici	Pais
39153	María Perez	Argentina
72044	Pedro Acosta	USA
72982	Jose Ortíz	Colombia
45922	Ana Nuñez	Francia
70425	Rafael Leconte	España

EVENTOS

Enum	Cnum	Fecha_eve	Hora_eve	Duración Hrs	Nro_Id_Partc	Evaluación	Nro_Id_Ju
1416	101	23-03-06	8 p.m	2	39153037	4	101010
1425	102	01-01-04	6 p.m	3	72044926	4.5	202020
2579	102	13-11-08	8 p.m	3	39153037	null	303030
1417	103	01-12-08	6 p.m	2	72044926	null	303030
1525	101	13-12-05	8 p.m	2	72982340	3	101010
2861	102	14-11-08	6 p.m	4	45922444	null	

Figura 107 – Modelo relacional caso certámenes

1. Se requiere saber toda la información de los Certámenes que están evaluados.
2. Se requiere saber el numero, la fecha de eventos y el nombre del certamen, que no han sido evaluados.
3. Se requiere saber el nombre del participante y la evaluación que tuvo en los eventos que participó.

Desarrollo

1. Se quiere saber toda la información de los certámenes que están evaluados.
 - a. la información que están pidiendo son todos los campos de la tabla de CERTÁMENES, es decir, el número, la fecha inicial, la fecha final y el nombre.
 - b. En cuanto a la restricción, dicen que estén evaluado los certámenes y en la tabla que podemos averiguar esto, es en la de EVENTOS, en el campo Evaluación.
 - c. De acuerdo a lo anterior, se debe aplicar primero una operación SELECT sobre la tabla EVENTOS, para saber cuáles son los certámenes que ya se encuentran evaluados, así:

Tabla 1= SELECT (EVENTOS/Evaluacion<> null)

Gráficamente se tiene:

Tabla 1

Enum	Cnum	Fecha_eve	Hora_eve	Dura_Hrs	Nro_Id_Partc	Evaluación	Nro_Id_Ju
1416	101	23-03-06	8 p.m	2	39153037	4	101010
1425	102	01-01-04	6 p.m	3	72044926	4.5	202020
1525	101	13-12-05	8 p.m	2	72982340	3	101010

Figura 108 – Tabla 1, consulta 1, caso certámenes

Luego se debe aplicar la operación de proyección del campo Cnum, pues estos son los certámenes que están evaluados, así

Tabla 2= PROYECT (Tabla 1/Cnum)

Tabla 2

Cnum
101
102
101

Figura 109 – Tabla 2, consulta 1, caso certámenes

- a) Ahora se aplica la operación de Reunión Natural (JOIN) entre la tabla 2 y la tabla de CERTÁMENES, para generar toda la información de los certámenes que están evaluados, así:

Tabla 3 = Tabla 2 JOIN CERTÁMENES

Tabla 3

Fecha_Ini	Fecha_Fin	Cnombre
01-01-04	05-01-04	Pintura Infantil
13-03-05	13-03-05	Cine junior
06-06-04	10-06-04	Musica rock

Figura 110 – Tabla 3, consulta 1, caso certámenes

2. Se requiere saber el número del evento, la fecha del evento y el nombre del certamen, que no han sido evaluados.

- a. la información que están pidiendo son el número del evento, la fecha del evento y el nombre del certamen. Se observa que los dos primeros campos se encuentran en la tabla de EVENTOS, mientras que el tercero se encuentra en la tabla CERTÁMENES.
- b. En cuanto a la restricción, dicen que se requieren los eventos que no hayan sido evaluados y éstos se observan en la tabla de EVENTOS, en el campo de Evaluación.

De acuerdo a lo anterior, se debe aplicar primero la operación SELECT sobre la tabla EVENTOS, para saber cuáles son los certámenes que NO se encuentran evaluados, así:

Tabla 1= SELECT (EVENTOS/Evaluacion= null)

Gráficamente se tiene:

Tabla 1

Enum	Cnum	Fecha_eve	Hora_eve	Duración Hrs	Nro_Id_Partc	Evaluación	Nro_Id_Ju
2579	102	13-11-08	8 p.m	3	39153037	null	303030
1417	103	01-12-08	6 p.m	2	72044926	null	303030
2861	102	14-11-08	6 p.m	4	45922444	null	

Figura 111 – Tabla 1, consulta 2, caso certámenes

Ahora de la Tabla 1, se debe sacar los campos número del evento y la fecha del evento. Además se debe sacar el número del certamen, pues se va a requerir más

adelante para generar la información del nombre del certamen, ya que en esta tabla no se encuentra. Luego se tiene:

Tabla 2 = PROYECT (Tabla 1/ Enum, Fecha_Eve, Cnum)

Tabla 2

Enum	Cnum	Fecha_eve
2579	102	13-11-08
1417	103	01-12-08
2861	102	14-11-08

Figura 112 – Tabla 2, consulta 2, caso certámenes

- b. En este momento se tiene casi toda la información que se pide, pero falta el nombre del certamen, como este campo se encuentra en la tabla de CERTÁMENES, entonces debemos hacer una Reunión Natural de ésta tabla con la tabla 2. Sin embargo es recomendable que primero se haga una proyección de la tabla CERTÁMENES de los campos Cnum y Cnombre, con el fin de quitar los campos que NO interesan, así:

Tabla 3 = PROYECT (CERTÁMENES/Cnum,Cnombre)

Tabla 3

Cnum	Cnombre
100	Cine culto
101	Pintura Infantil
102	Cine junior
103	Musica rock

Figura 113 – Tabla 3, consulta 2, caso certámenes

Ahora, se hace la Reunión natural entre la Tabla 2 y la Tabla 3. La cual queda así:

Tabla 4 = Tabla 2 JOIN Tabla 3

Tabla 4

Enum	Cnum	Fecha_eve	Cnombre
2579	102	13-11-08	Cine junior
1417	103	01-12-08	Musica rock
2861	102	14-11-08	Cine junior

Figura 114 – Tabla 4, consulta 2, caso certámenes

Por último, como NO pide el número del certamen, entonces se debe realizar una proyección de la tabla 4, con los campos que importan así:

Tabla 5 = PROYECT (Tabla 4/Enum,Fecha_Eve,Cnombre=

Tabla 5

Enum	Fecha_eve	Cnombre
2579	13-11-08	Cine junior
1417	01-12-08	Musica rock
2861	14-11-08	Cine junior

Figura 115 – Tabla 5, consulta 2, caso certámenes

3. Se requiere saber el nombre del participante y la evaluación que obtuvo en los eventos que se presentó.

Este puntos, se deja como ejercicio.

CAPÍTULO 6: “NORMALIZACIÓN”

Lección 26: “Conceptos de Normalización”

La normalización es una técnica que sirve para diseñar las bases de datos relacionales. Esta técnica se utiliza cuando NO se quiere utilizar la técnica Entidad-Relación.

La normalización parte de un concepto fundamental como es el de *Dependencia Funcional*.

Según Irene y otros en su libro La define así:

"Dada una tabla T, se dice que el campo T.y que pertenece a T, es funcionalmente dependiente de otro campo T.x que pertenece a T, y se expresa de la forma T.x-àT.y ,si y solo si, cada valor de T.x tiene asociado a él exactamente un valor en T.y para cualquier extensión de la tabla T." .

La anterior definición esta tomada del libro "bases de datos, desde Chen hasta Codd" de los autores Irene Luque Ruiz y otros de la Universidad de Córdoba. En ella se ha cambiado la palabra Relación por Tabla, con fines pedagógicos, de tal forma que para los estudiantes sea mas claro.

Tomando el primer ejemplo se toma la tabla de Empleados y se puede observar un ejemplo de T.x y T.y. Solo se cojera tres campos.

Nro_Empl	Nomb_Empl	Dir_Res_Empl
101010	Laura Jaramillo	D. 22 No 30-40
202020	Stella Salgado	Cra. 22 No 10-10
303030	Jose Jaramillo	D. 22 No 30-40
404040	Laura Jaramillo	C. 1 No 10-10

Figura 116 – Tabla empleados, normalización

Se toma como T.x a Nro_Empl y a T.y a Nomb_Empl y Dir_Res_Empl. Se observa que para un T.x (101010) siempre va a tener el mismo T.y (Laura Jaramillo); pero si se toma un T.y (Laura Jaramillo) no siempre va a tener el mismo T.x, en este caso (101010 o 404040). Dichos en otras palabras, los T.x, en una tabla, siempre es el Campo o Campos claves, mientras que los T.y son los campos que no hacen parte de la clave.

Teniendo claro el concepto de dependencia funcional, se pasa ahora a conceptualizar las diferentes formas normales (FN) en las que puede estar las instancias o filas de una tabla:

Lección 27: “Formas Normales”

1. Primera Forma Normal (1FN)

Una tabla esta en primera forma normal, si y solo si, no existe campos multievaluados. Visto este concepto gráficamente tenemos:

A	B	C	D	E
---	---	---	---	---

Supongamos que D, es un campo que para el mismo A, puede tener tres valores, entonces se debe separar de tal forma que la tabla queda así:

A	B	C	D1	D2	D3	E
---	---	---	----	----	----	---

Estos casos, se presentan muchos cuando se tienen campos Fechas, a veces se define un solo campo fecha, pero dependiendo del estado de A, entonces puede ser Fecha de Ingreso o Fecha de Retiro o Fecha de vacaciones, lo cual es un error de diseño, se debe pues, definir los tres tipos de fecha.

Hay otro caso, que es cuando un campo está compuesto por varios subcampos, por ejemplo la dirección (Calle, Carrera, Numero, Cardinalidad, Edificio, Apto, etc). Sin embargo, ya existen motores de bases de datos que permiten este tipo de campos, pero como un tipo de datos, lo mismo para manejar en un campo como tipo vector, días de semana, meses de año. Donde de alguna manera se garantiza no exceder las posiciones dadas.

2. Segunda Forma Normal (2FN)

Una tabla esta en segunda forma normal, si y solo si, esta en 1FN, y todos los campos no primos, es decir que no son campos claves (T.y), dependen funcionalmente en su totalidad del campo primo, es decir, del campo clave (T.x). Por convención se denota en negrilla, los campos que son claves.

A	B	C	D	E
---	---	---	---	---

Se supone que en este ejemplo, los campos B y C dependen funcionalmente por completo del campo A, que el campo E depende funcionalmente de D y, que el campo D no depende funcionalmente de A. entonces la tabla debe partirse así:

A	B	C
D	E	

De tal forma que en la segunda tabla D pasa a ser el T.x y E el T.y. Este tipo de situaciones se presentan, cuando hay dos tabla que NO tienen relación directa.

3. Tercera Forma Normal (3FN)

Una tabla esta en 3FN, si y solo si, está en 2FN y ningún campo no primo (T.y) depende funcionalmente en forma transitiva del campo primo (campo clave) T.x.

A	B	C	D	E
---	---	---	---	---

Se supone que los campos B, C y D dependen funcionalmente de A; y que E depende funcionalmente de D. Entonces vemos, que E depende funcionalmente de A, en forma transitiva. Para solucionar este problema de diseño, partimos la tabla en dos así:

A	B	C	D
D	E		

Se observa que D, aparece en las dos tablas. De acuerdo a los visto hasta ahora, se dice que D es una llave foránea de la primera tabla. Un ejemplo practico de esto podría ser el Código del cargo (D) y el Nombre del cargo (E), se sabe que el nombre del cargo depende funcionalmente del código del cargo, pero adicionalmente se requiere saber el código del cargo en la tabla de empleados, suponiendo que la primera tabla es la de empleados.

La situación anterior se presenta cuando hay relación entre dos tablas.

4. Consideraciones de diseño

Se dice, que para que un diseño de bases de datos sea aceptable, este debe estar como mínimo en Tercera Forma Normal. Sin embargo, cuando se aprende la técnica Entidad Relación, vista en la unidad I, da como resultado un diseño de bases de datos, como mínimo en 3FN. Es el analista de sistemas, el que preferirá que técnica seleccionar y con cual se siente más cómodo. Sin embargo en la actualidad hay muchas herramientas bajo la técnica Entidad-Relación, no así, para aplicar reglas de normalización.

Lección 28: “Análisis de Normalización Caso Matrícula”

Una institución de Educación superior, lo ha contratado para el diseño de una base de datos relacional. Actualmente, la Institución cuenta con una hoja de cálculo, donde tienen toda la información de las matrículas de los estudiantes por asignatura. El diseño de la hoja de cálculo es la siguiente:

MATRICULAS

Nro_Asig	Nro_Id_Est	Fecha_Mat	Nomb_Asig	Nro_Cred	Apell_Est	Nomb_Est	Dir_Est	Tel_Est	Nro_Aula	Ub_Aula	Grupo
----------	------------	-----------	-----------	----------	-----------	----------	---------	---------	----------	---------	-------

La problemática que existe con el diseño de la hoja de cálculo, es que muchas veces se encuentran para un mismo estudiante, nombres diferentes o apellidos, lo mismo sucede con la asignatura, para un mismo número de asignatura se encuentran nombres diferentes. Cada fila de la hoja de cálculo es única por número de asignatura y número identificación estudiantes.

Para dar solución al problema anterior, el ingeniero pregunta que le expliquen cada uno de los campos de la hoja de cálculo. Por lo tanto proceden a darle la siguiente información.

Nro_Asig: Es el número de la asignatura, el cual es único para cada asignatura.

Nro_Id_Est: Es el número de identificación del estudiante, el cual es único para cada estudiante.

Fecha_Mat: Es la fecha en que el estudiantes matriculó la asignatura.

Nom_Asig: Es el nombre de la asignatura

Nro_Cred: Es el número de créditos con el que cuenta la asignatura.

Apell_Est: Son los apellidos del estudiante.

Nom_Est: Son los nombres del estudiante.

Dir_Est: Dirección residencia del estudiante.

Tel_Est: Es el teléfono del estudiante.

Nro_Aula: Es el número de aula y es único en toda la Institución

Ub_Aula: Es la ubicación que tiene el aula en la institución

Grupo: Es una letra de los grupos que existe en una misma asignatura, cuando es necesario abrir varios para una misma asignatura. Por lo tanto, este grupo es único dentro de la asignatura, de tal modo que un estudiante cuando matricula una asignatura, se le asigna también el grupo.

Por otro lado, en una aula, puede estar asignados varios grupos de una o distinta asignatura. Pero un grupo de una asignatura determinada, solo se le asigna una aula.

Solución:

Lo primero que se debe establecer en el caso anterior son las dependencias funcionales que existen entre el X y el Y. Para este caso la X la conforma el Nro_Asig y Nro_Id_Est.; esto debido a que dicen que ellos llevan un registro único de matrícula por estos dos campos. Los Y, serían los demás campos.

Si se observan los campos Apell_Est, Nom_Est, Dir_Est y Tel_Est; éstos no dependen funcionalmente del X, sino de una parte de éste, como es el Nro_Id_Est. Lo cual da, que estos campos Y, dependen transitivamente de una parte de la X y por lo tanto se debe crear una nueva relación, que se puede llamar ESTUDIANTES y pasar los campos Y a la nueva Relación, así:

MATRICULAS

Nro_Asig	Nro_Id_Est	Fecha_Mat	Nomb_Asig	Nro_Cred	Nro_Aula	Ub_Aula	Grupo
----------	------------	-----------	-----------	----------	----------	---------	-------

ESTUDIANTES

Nro_Id_Est	Apell_Est	Nomb_Est	Dir_Est	Tel_Est
------------	-----------	----------	---------	---------

Aquí se observa que se aplicó la Tercera Forma Normal.

Ahora se puede mirar que los campos Nomb_Asig y Nro_Cred.; no dependen funcionalmente del X, sino de una parte de éste, como es el Nro_Asig. Lo cual da, que estos campos Y, dependen transitivamente de una parte de la X y por lo tanto se debe crear una nueva relación, que se puede llamar ASIGNATURAS y pasar los campos Y a la nueva Relación, así:

MATRICULAS

Nro_Asig	Nro_Id_Est	Fecha_Mat	Nro_Aula	Ub_Aula	Grupo
----------	------------	-----------	----------	---------	-------

ESTUDIANTES

Nro_Id_Est	Apell_Est	Nomb_Est	Dir_Est	Tel_Est
------------	-----------	----------	---------	---------

ASIGNATURAS

Nro_Asig	Nomb_Asig	Nro_Cred
----------	-----------	----------

Aquí se vuelve aplicar la 3FN.

Nótese, que en los dos análisis anteriores, el X de las tablas Estudiantes (Nro_Id_Est) y Asignaturas (Nro_Asig), siguen perteneciendo a la tabla de matrículas, puesto que ellas están relacionadas con dicha tabla.

Siguiendo con el caso, se observa que el campo Nro_Aula, no depende funcionalmente de X. Por lo tanto debe crearse otra relación, que se puede llamar AULAS. Por otro lado, el campo Ub_Aula, depende funcionalmente únicamente del campo de Nro_aula y por lo tanto debe salir de la Tabla de MATRICULAS y hacer parte de la Tabla de AULAS, así:

MATRICULAS

Nro_Asig	Nro_Id_Est	Fecha_Mat	Grupo
----------	------------	-----------	-------

ESTUDIANTES

Nro_Id_Est	Apell_Est	Nomb_Est	Dir_Est	Tel_Est
------------	-----------	----------	---------	---------

ASIGNATURAS

Nro_Asig	Nomb_Asig	Nro_Cred
----------	-----------	----------

AULAS

Nro_Aula	Ub_Aula
----------	---------

Como el campo Ub_Aula, depende funcionalmente del campo Nro_Aula, entonces éste pasa a ser el X, y Ub_Aula el Y de la tabla AULAS. En este caso se aplicó la segunda forma normal (2FN), pues no hay relación directa entre las tablas AULAS y MATRICULAS.

Con el panorama anterior se observa que los Y que quedan en la tabla MATRICULAS, dependen funcionalmente de X. Sin embargo el caso plantea que pueden existir para una asignatura varios grupos y que los grupos son único dentro de cada asignatura. Además de que se le asigna un aula a un grupo de una determinada asignatura, de tal forma, que los estudiantes saben cuál es el aula en la que se dicta. Por lo tanto, se observa que hay una relación entre la asignatura, el grupo y el aula, de tal forma, que se debe crear una tabla donde las relacione, así:

ASIGNATURAS_GRUPOS_AULAS

Nro_Asig	Grupo	Nro_Aula
----------	-------	----------

Se observa que el X de esta tabla es Nro_Asig y Grupo, esto porque nos dice que una asignatura puede tener varios grupos, pero que el grupo es único dentro de una misma asignatura. El Y queda el Nro_Aula, debido a que a cada grupo de asignatura se le asigna un aula. Por lo tanto, la tabla Asignatura_Grupos_Aulas, es una tabla que está relacionada con las tablas Asignaturas y Aulas.

Aquí se aplica todo el concepto de Dependencia Funcional y 3FN.

Lección 29: “Análisis de Normalización Caso Mensajería”

En una empresa de mensajería intermunicipal, lleva un registro de todas las recepciones y envíos. Para ellos tienen en una hoja de cálculo la siguiente información.

REGISTROS_MENSAJERÍAS

Nro_Reg	Mun_Rem	Fecha_Env	Nomb_Rem	Nro_Id_Rem	Nro_Id_Des	Nomb_Des	Dir_Rem	Tel_Rem	Dir_Des	Tel_Des	Mun_Des	Cod_Mun_Rem	Desc_encom
Valor_Encom			Cod_Mun_Des										

La explicación dada de los diferentes campos fueron:

Nro_Reg: Es un consecutivo que ellos le asignan cada vez que ingresan un registro y este consecutivo es único para cada envío.

Mun_Rem: Es el nombre del municipio del remitente.

Fecha_Env: Es la fecha en que el remitente se acercó a la oficina a realizar la operación de envío.

Nom_Rem: Es el nombre y apellido completo de la persona que realiza la operación de envío, también se le llama cliente o remitente.

Nro Id Rem: Es el número de identificación del remitente.

Nom_Des: Es el nombre y apellido completo de la persona a quien va dirigida la encomienda, también se llama Cliente o Destinatario.

Nro Id Des: Es el número de identificación del Destinatario.

Dir_Rem: Dirección del remitente

Tel_Rem: Telefono remitente.

Dir_Des: Dirección del destinatario

Tel_Des: Telefono destinatario.

Mun_Des: Es el nombre del municipio del destinatario.

Cod_Mun_Rem: Código municipio del remitente.

Cod_Mun_Des: Código municipio del destinatario.

Desc_Encom: Es una descripción de la encomienda que se está enviando.

Valor_Encom: Es el costo que debe pagar el remitente por el envío de la encomienda.

Los problemas de información que tiene la empresa llevando la información en esta forma, es no poder generar estadísticas, ni tampoco tener acceso rápido a una determinada encomienda, cuando se presentan reclamaciones por parte de los clientes. Además, que muchas veces para un mismo código de municipio o un mismo número de identificación, existe nombre de municipio o personas diferentes.

Por lo tanto, solicitan que le diseñe una base de datos relacional, donde se jefe le piden que normalice la estructura anterior de datos.

Solución:

Lo primero que se debe establecer en el caso anterior son las dependencias funcionales que existen entre el X y el Y. Para este caso la X la conforma el Nro_Reg; esto debido a que dicen que ellos llevan un consecutivo por cada envío. Los Y, serían los demás campos.

Si se observan los campos Nom_Rem, Dir_Rem y Tel_Rem; éstos no dependen funcionalmente del X, sino de del campo Nro_Id_Rem. Lo cual da, que estos campos Y, dependen transitivamente de X y por lo tanto se debe crear una nueva relación, que se puede llamar REMITENTES y pasar los campos Y a la nueva Relación, así:

REGISTROS_MENSAJERÍAS

Nro_Reg	Mun_Rem	Fecha_Env	Nro_Id_Rem	Nro_Id_Des	Nomb_Des	Dir_Des	Tel_Des	Mun_Des	Cod_Mun_Rem	Desc_Encom
---------	---------	-----------	------------	------------	----------	---------	---------	---------	-------------	------------

REMITENTES

Nro_Id_Rem	Nomb_Rem	Dir_Rem	Tel_Rem
------------	----------	---------	---------

En la operación anterior, aplicamos la 3FN

Si se observa en la tabla REGISTROS_MENSAJERÍA que quedo, los campos Nomb-Des, Dir_Des y Tel_Des, sucede lo mismo que lo anterior. Por lo tanto se debe crear una nueva tabla con estos datos así:

REGISTROS_MENSAJERÍAS

Nro_Reg	Mun_Rem	Fecha_Env	Nro_Id_Rem	Nro_Id_Des	Mun_Des	Cod_Mun_Rem	Desc_Encom	Valor_Encom	Cod_Mun_Des
---------	---------	-----------	------------	------------	---------	-------------	------------	-------------	-------------

REMITENTES

Nro_Id_Rem	Nomb_Rem	Dir_Rem	Tel_Rem
------------	----------	---------	---------

DESTINATARIOS

Nro_Id_Des	Nomb_Des	Dir_Des	Tel_Des
------------	----------	---------	---------

El análisis anterior realizado a la campos de los remitentes y destinatarios, se puede aplicar igual a los de municipios remitentes y destinatarios, de tal forma, que el campo descripción del municipio depende funcionalmente del código del municipio. Pero adicionalmente se puede hacer otro análisis y es crear un tabla municipios con su código y su descripción y relacionarla con las tablas de remitentes y destinatarios y estos campos salen totalmente de la tabla de REGISTROS MENSAJERÍA. Gráficamente se tiene:

REGISTROS_MENSAJERÍAS

Nro_Reg	Fecha_Env	Nro_Id_Rem	Nro_Id_Des	Desc_Encom	Valor_Encom
---------	-----------	------------	------------	------------	-------------

REMITENTES

Nro_Id_Rem	Nomb_Rem	Dir_Rem	Tel_Rem	Cod_Mun
------------	----------	---------	---------	---------

DESTINATARIOS

Nro_Id_Des	Nomb_Des	Dir_Des	Tel_Des	Cod_Mun
------------	----------	---------	---------	---------

MUNICIPIOS

Cod_Mun	Des_Mun
---------	---------

Aquí se aplicaron dos reglas la 2FN, con la tabla de Registros de Mensajería y la 3FN, con las tablas Remitentes y Destinatarios.

Lección 30: “Análisis de Normalización Caso Biblioteca”

En una biblioteca de una institución educativa, se lleva un registro de todos préstamos de libro. Para ellos tienen en una hoja de cálculo la siguiente información.

PRESTAMOS_LIBROS

Nro_Pres	Fecha_Pres	Nomb_Est	Nro_Id_Est	Id_Empl	Nomb_Empl	Nomb_Prog	Tel_Est	Dias_Pres	Fecha_Dev	Dias_Ret	Cod_Lib	Tit_
----------	------------	----------	------------	---------	-----------	-----------	---------	-----------	-----------	----------	---------	------

La explicación dada de los diferentes campos fueron:

Nro_Pres: Es un consecutivo que ellos le asignan cada vez que un estudiante presta un libro.

Fecha_Pres: Es la fecha en que el estudiante prestó el libro.

Nom_Est: Es el nombre y apellido completo del estudiante que hace el préstamo

Nro_Id_Est: Es el número de identificación del estudiante.

Nom_Emp: Es el nombre y apellido completo del empleado de la biblioteca, quien atiende el préstamo del libro.

Id_Emp: Es el número de identificación del empleado.

Nom_Prog: Nombre del programa que está adscrito el estudiante.

Tel_Est: Telefono estudiante.

Dias_prest: Número de días que se le presta el libro al estudiante.

Fecha_Dev: Fecha en la cual el estudiante debe devolver el libro a la biblioteca.

Dia_Ret: Dias de retraso de la entrega del libro

Cod_Lib: Código del libro.

Tit_Libro: Titulo del libro.

Editorial: Editorial donde fue impreso el libro.

Valor_San: es el valor de la sanción que debe pagar el estudiante por un día de retraso

Los problemas de información que tiene la empresa llevando la información en esta forma, es no poder generar estadísticas, ni tampoco tener acceso rápido a una determinado préstamo, cuando se presentan reclamaciones por parte de los estudiantes. Además, que muchas veces para un mismo código de libro o un mismo número de identificación, existe títulos de libros diferentes o nombres de personas diferentes para un mismo código o número de identificación, respectivamente.

Por lo tanto, solicitan que le diseñe una base de datos relacional, donde el jefe de biblioteca le piden que normalice la estructura anterior de datos.

Solución:

Lo primero que se debe establecer en el caso anterior son las dependencias funcionales que existen entre el X y el Y. Para este caso la X la conforma el Nro_Pret; esto debido a que nos dicen que ellos llevan un consecutivo por cada préstamo. Los Y, serían los demás campos.

Si se observan los campos Nom_Est y Tel_Est; éstos no dependen funcionalmente del X, sino de del campo Nro_Id_Est. Lo cual dá, que estos campos Y, dependen transitivamente de X y por lo tanto se debe crear una nueva relación, que se puede llamar ESTUDIANTES y pasar los campos Y a la nueva Relación, así:

PRESTAMOS_LIBROS

Nro_Pres	Fecha_Pres	Nro_Id_Est	Id_Empl	Nomb_Empl	Nomb_Prog	Dias_Pres	Fecha_Dev	Dias_Ret	Cod_Lib	Tit_Lib	Editorial	Valor_Sanc
----------	------------	------------	---------	-----------	-----------	-----------	-----------	----------	---------	---------	-----------	------------

ESTUDIANTES

Nro_Id_Est	Nomb_Est	Tel_Est
------------	----------	---------

En la operación anterior, aplicamos la 3FN

Se deja como ejercicio, continuar con la normalización de la tabla de PRESTAMOS_LIBROS.

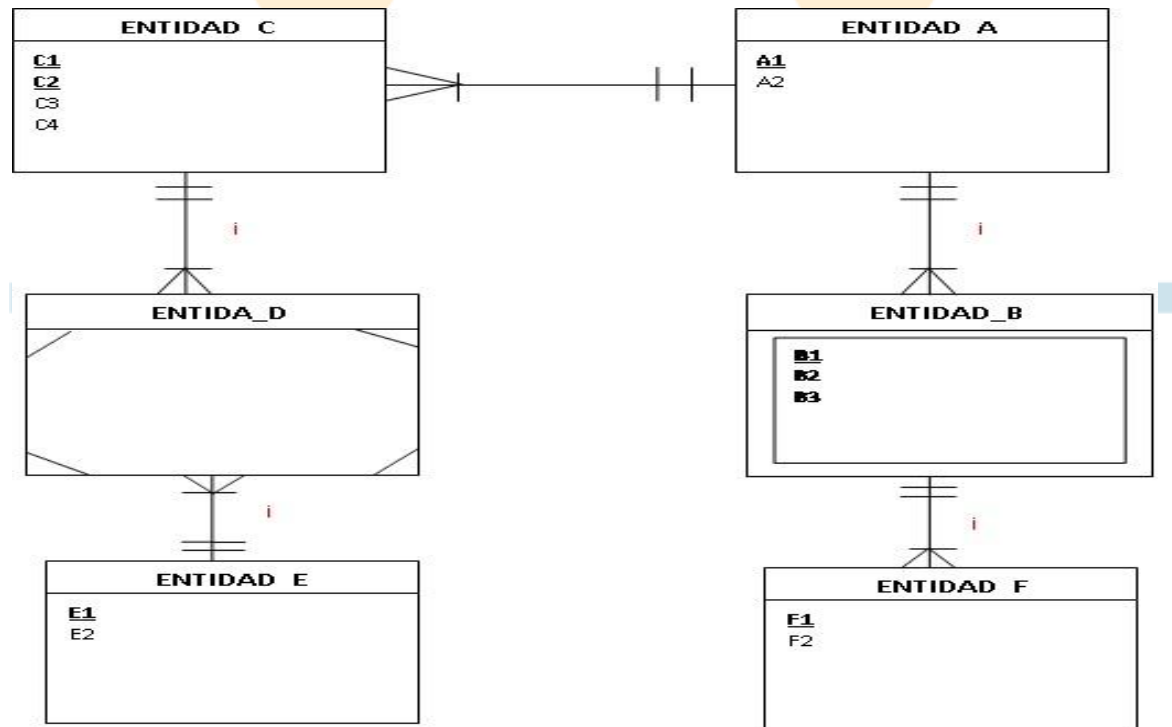
BIBLIOGRAFÍA

C.J. DATE. Introducción a los sistemas de bases de datos.

I. LUQUE Introducción a las bases de datos. Desde Cood hasta Chen. Univ. De Cordoba.

ACTIVIDADES DE AUTOEVALUACIÓN DE LA UNIDAD

1. Dado el siguiente modelo lógico de datos, realizar el modelo relacional aplicando las reglas de paso de un modelo a otro.



2. De acuerdo a la estructura dada, realice las siguientes consultas basados en operadores de álgebra relacional.

Esquema de la base de datos “LABORATORIOS”

ESTUDIANTES (IdEst, Nombre, Grupo)

PRACTICAS (IdPractica, Curso, Fecha)

REALIZA (IdEst, IdPractica, Nota)

Enunciados de consulta:

- Nombre de los estudiantes que aprobaron todas las prácticas del curso Bases de Datos.
 - Nombre de los estudiantes que realizaron todas las prácticas del curso Bases de Datos.
 - Nombre de los estudiantes que han realizado prácticas de Bases de Datos y de Física.
 - Nombre de los estudiantes que sólo han realizado prácticas de Física.
 - Nombre de los estudiantes que han realizado por lo menos una práctica de Bases de Datos, de Física y de Algoritmos.
 - Nombre de los estudiantes que pertenecen al grupo 10 del curso Algoritmos.
 - Nombre de los estudiantes que no han aprobado ninguna práctica.
3. Dada la siguiente estructura datos, aplicar las reglas de normalización para que quede mínimo en tercera forma normal.

PRESTAMO_LIBROS

Nro_Pres	Fecha_Pres	Nro_Id_Est	Id_Empl	Nomb_Empl	Nomb_Prog	Dias_Pres	Fecha_Dev	Dias_Ret	Cod_Lib	Tit_Lib	Editorial	Valor_Sanc
----------	------------	------------	---------	-----------	-----------	-----------	-----------	----------	---------	---------	-----------	------------

UNIDAD 3.

“LENGUAJE ESTANDAR DE CONSULTA Y HERRAMIENTAS”

Nombre de la Unidad	“LENGUAJE ESTANDAR DE CONSULTA Y HERRAMIENTAS”
Introducción	<p>En la unidad anterior se estudio las operaciones del álgebra relacional imprescindibles para entender las consultas que se pueden realizar en una base de datos relacional. En general, el álgebra relacional está clasificada como un lenguaje de consulta formal procedimental, en el que el usuario indica cómo (en qué orden) se debe especificar las operaciones de la consulta para obtener el resultado deseado. En este tema se va a estudiar el lenguaje de consulta estructurado (SQL), basado en álgebra relacional. Los Sistemas Gestores de Bases de Datos Relacional (SGBDR) comerciales cuentan con una interfaz de lenguaje declarativo de alto nivel en el que se especifica la consulta en SQL (Lenguaje de Consulta Estructurado, del inglés Structured Query Language o QBE, y el propio SGBD es el que se encarga de realizar las optimizaciones necesarias para ejecutar la consulta.</p> <p>SQL fue diseñado e implementado por IBM Research y se ha convertido en un estándar para las bases de datos relacionales. Además, es posible que los programadores de aplicaciones escriban sentencias SQL en sus aplicaciones de programas que utilizan bases de datos relacionales para el almacenamiento de datos, dando lugar a lo que se conoce como SQL embebido. SQL es un lenguaje de consulta completo, y se puede utilizar como Lenguaje de Definición de Datos (DDL) o como Lenguaje de Manipulación de Datos (DML), ya que cuenta con sentencias para Definición de datos y Manipulación de datos (Consulta y Actualización). Además, SQL cuenta con un mecanismo para la definición de vistas de la base de datos, creación y eliminación de índices y para la incorporación de sentencias SQL en lenguajes de programación de propósito general.</p>

	<p>Cuando se utilizan las sentencias SQL, aquí la terminología utilizada es TABLAS, COLUMNAS Y FILAS. Donde, la tabla en el álgebra relacional la llaman Relación y la fila la llaman Tupla, las columnas se siguen llamando igual.</p>
Justificación	<p>El lenguaje de manipulación de datos, es un tema de vital importancia en las bases de datos relacional, pues es por medio de este lenguaje que se puede estructura y manipula los datos, ya sea en forma interactiva o embebida. Cada día, los motores de bases de datos agregan nuevas funcionalidades. Sin embargo, todos debe tener y sostener un estándar mínimo.</p>
Intencionalidades Formativas	<ul style="list-style-type: none"> • Fundamentar las sentencias para estructura la base de datos. • Fundamentar las sentencias para manipular los datos en la base de datos. • Potenciar en el estudiante la de análisis y solución a solicitudes de información.
Denominación de los capítulos	<ul style="list-style-type: none"> • Lenguaje de definición de datos • Lenguaje de manipulación de datos • Herramienta MySql

UNIDAD 3.

“LENGUAJE ESTANDAR DE CONSULTA Y HERRAMIENTAS”

Hasta ahora hemos visto en las unidades anteriores, como se realiza el modelo lógico de datos y el modelo relacional. El diseño de este último, se puede realizar pasando por un modelo lógico de datos, aplicando la técnica entidad relación o aplicando directamente las reglas de normalización. También se vió el lenguaje primitivo para la manipulación de datos, esto último fue lo que animo a Cood a crear toda una filosofía de estructuración de datos, de tal forma, que la creación, actualización, borrado y consulta, se hiciera de una forma controlada y fácil.

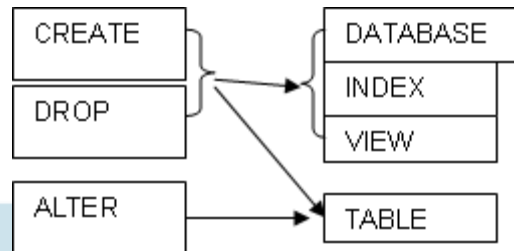
Con el paso del tiempo, se creó un lenguaje NO procedimental, pero basado en operaciones de álgebra relacional, donde solo la preocupación del usuario era realizar sentencias declarativas para manipular los datos, pero no intervenía en el CÓMO. A esto se le conoció como SQL y fue cuando se inició la cuarta generación (4GL). De tal forma, que hoy por hoy, todos los Sistemas Gestores de Bases de Datos Relacional, debe tener incorporado el SQL, con las sentencias estándar de la industria.

Con más de treinta años, las tecnologías de las bases de datos relacionales han ido incorporando la teoría que le dio inicio y que hasta la presente, muchos fabricantes de esta herramienta, han ido incorporando más facilidades. Con esto último se debe tener especial cuidado porque se alejan del estándar y cuando se quiera migrar de un fabricante a otro, se puede tener problemas de incompatibilidad. Es por eso que en esta unidad, abarcaremos solo las sentencias estándar que se encuentran en cualquier herramienta de bases de datos relacional.

Para el desarrollo de esta unidad, tomaremos el modelo relacional de reservas, que se encuentra en el anexo No 1

CAPÍTULO 7: “LENGUAJE DE DEFINICIÓN DE DATOS”

El lenguaje de Definición de Datos, es el encargado de DEFINIR toda la estructura de una base de datos relacional. Para esto se apoya en tres instrucciones básicas como son el CREATE, el ALTER y el DROP. Los elementos de las bases de datos que se le puede aplicar estas instrucciones son bases de datos (DATABASE), tablas (TABLE), índices (INDEX) y vistas (VIEW). Miremos esquemáticamente lo anterior:



Obsérvese que las bases de datos, índices y vistas solo pueden ser creadas (CREATE) o borradas (DROP), mientras que las tablas, pueden también, ser modificadas, mediante la instrucción ALTER.

Lección 31: “Creación de Bases de Datos y Tablas”

La convención que se va a utilizar para cualquier instrucción es que las PALABRAS RESERVADAS se colocan en letra azul. Ahora, miremos en forma general como sería la instrucción de creación de cualquier elemento:

NOMB_INSTRUCCION ELEMENTO Nomb_Elemento

Tomando el ejemplo de Reservas de Hoteles, que se colocó al principio de esta unidad. Se procede así:

1. Creación de una base de datos

Para el ejemplo llamaremos a la bases de datos Reservas_Hoteles

CREATE DATABASE Reservas_Hoteles;

2. Creación de una tabla

```
CREATE TABLE Municipios
```

```
(Cod_Mun INT NOT NULL,
```

```
Nombre_Municipio CHAR(30) NOT NULL,
```

```
PRIMARY KEY (Cod_Mun));
```

```
CREATE TABLE Hoteles
```

```
(Cod_Hot CHAR(4) NOT NULL,
```

```
Hnombre CHAR(30) NOT NULL,
```

```
Cod_Mun INT,
```

```
PRIMARY KEY (Cod_Hot),
```

```
FOREIGN KEY (Cod_Mun) REFERENCES Municipios (Cod_Mun));
```

Haciendo un análisis hasta aquí, se puede observar que cuando finaliza una sentencia, se coloca punto y coma (;). Claro que muchos motores de bases de datos, ya no ponen problema en esto, sino que si la sentencia está bien escrita la ejecuta. Otro punto es que en la definición de la llave foránea, hubo necesidad de cualificar el campo, debido a que el nombre recibido por código municipio es igual en ambas tablas (Municipios y Hoteles).

```
CREATE TABLE Huespedes
```

```
(Nro_Id_Hues INT NOT NULL,
```

```
Nombre_Hues CHAR(50) NOT NULL,
```

```
PRIMARY KEY (Nro_Id_Hues));
```

```
CREATE TABLE Reservas
```

```
(Rnum INT NOT NULL,
```

```
Fecha_Ini DATE,
```

Dias_Duracion INT,
Cod_Hot CHAR(4),
Nro_Id_Hues INT,
Nro_Hab CHAR(4),
Fecha_Res DATE,
PRIMARY KEY (Rnum),
FOREIGN KEY (Cod_Hot) REFERENCES Hoteles (Cod_Hot),
FOREIGN KEY (Nro_Id_Hues) REFERENCES Huespedes (Nro_Id_Hues));

Una aspecto a tener en cuenta con la definición de las llaves foráneas, es que la columna a la que se hace referencia, debe llamarse igual como se haya definido en la Tabla., así se observa que, Nro_Id_Hues que se encuentra entre paréntesis se llama exactamente igual a como se definió en la tabla de Huespedes. además obsérvese que se cualificó, debido a que tiene el mismo nombre en la tabla de Reservas y por lo tanto se debe evitarse las ambigüedades.

Lección 32: “Creación de Índices”

Existen muchas definiciones de índices:

Es un arreglo ordenado de claves e indicadores (Rob, 2004). Donde la clave de índice es el campo por donde se el índice a la tabla.

Permite el acceso de forma rápida a los renglones por medio del valor de campo indizado (Kroenke, 2003). Donde los renglones, son las filas de la tabla.

Los índices son los caminos que al motor de las bases de datos se le dan para que los encuentre de forma mas eficiente los datos, de tal forma, que escoja el camino más rápido para dar respuesta a una petición de un usuario. Un criterio fundamental para diseñar los índices, es la frecuencia de consulta de un dato o grupo de datos en una tabla y que no es llave primaria y el tamaño de la tabla. Sin embargo, hay que tener en cuenta que si bien se gana rapidez en la consulta, mientras más índices tenga una tabla, los procesos de inserción, borrado y

modificación son mas demorado. De allí, que el administrador de bases de datos requiera mirar la forma como se ejecutó una consulta, una vez que crea un índice, con el fin de observar si el índice fue utilizado por el gestor de consulta, pues si este no fue utilizado, debe ser borrado, ya que no tiene sentido de acuerdo a lo dicho al inicio de este párrafo. Otro criterio para crear índices, es cuando se desean que el valor de un campo NO se repita, es el caso cuando una tabla tiene varias llaves candidatas a ser llave primaria; aquellas que no se seleccionan, entonces se les puede crear un índice, para garantizar tambien la unicidad por este campo.

Todo índice se crea *solo* sobre *una* tabla, y este puede ser de un campo o la combinación de varios, según como se requiere. De acuerdo al ejemplo que se está trabajando, supóngase que en la tabla de reservas, se consulta mucho por el número del Huesped; si se observa la llave primaria de esa tabla es el número de la reserva, el cual los huéspedes olvidan con facilidad y por lo tanto se consulta mucho por el número de identificación de él. Así que se creará un índice por este campo.

```
CREATE INDEX Nombre_Indice ON Nomb_Tabla(Nombre_Campos1,  
Nombre_Campo2, ....);
```

```
CREATE INDEX Huesped ON Reservas(Nro_Id_Hues);
```

Otro índice que se puede crear es el de hoteles en la tabla reservas, pues también las reservas son consultados frecuentemente por este campo.

```
CREATE INDEX Hoteles ON Reservas(Cod_Hot);
```

De todas maneras, es bueno advertir, que como diseñadores de bases de datos, la creación de índices se deja mejor a opción del administrador de la base de datos, pues es él único que tiene la certeza del desempeño óptimo de la base de datos.

Lectura recomendada tomada del libro:

Introducción a las bases de datos: el modelo relacional. Escrito por Olga Pons Capote

[Haga Clic aqui para leer.](#)

Lección 33: “Creación de Vistas”

Las vistas son tablas virtuales, estas sirven para dar seguridad de acceso a las bases de datos. Las vistas solo existen, en el momento que se están ejecutando, es decir, de acuerdo al diseño de esta, los datos se cargan y están en ellas mientras se esté usando.

A diferencia de los índices, las vistas pueden diseñarse con base en una o varias tablas. A través de las vistas se puede actualizar datos en la base de datos real. Ahora se creará una vista donde se tiene el número de identificación y el nombre del huésped, el número de reservas que tiene con su hotel y la fecha. Luego quedaría así:

```
CREATE VIEW RESERVA_HUESPEDES ON
```

```
(SELECT Nro_Id_Hues, Nomb_Hues, Rnum, Cod_hot
```

```
FROM RESERVAS, HUESPEDES
```

```
WHERE RESERVAS.Nro_Id_Hues= HUESPEDES.Nro_Id_Hues);
```

1) Para que se puedan hacer inserciones de filas a la tabla , a partir de una vista. La vista debe contener como mínimo la llave primaria de la tabla y solo debe diseñarse sobre esta tabla.

2) Para realizar borrados o actualizaciones sobre las filas de una tabla, a partir de una vista. Debe cumplir con todo los parámetros válidos de borrado o actualización, para que no haya anomalía de borrado, por ejemplo, dejar a hijos huérfanos.

Lectura recomendada, tomada del libro Introducción a las bases de datos de C. J. Date.

[Haga Click aqui para leer.](#)

Lección 34: “Modificación de Tablas”

La modificación de la tablas, desde el punto de vista de la estructura de las bases de datos, es de las operaciones mas delicadas que puede haber, ya que si la modificación de la estructura se realiza sin que la tabla tenga datos, no hay ningún problema, pero si se realiza teniendo datos la tablas, hay muchos motores que tienen restricciones como las siguientes:

1. No deja modificar ningún tipo de datos de los campos existentes.
2. Si deja modificar un tipo de datos en algún campo existente, esto solo los deja si los datos son numéricos o fechas a Alfanumérico o Caracter, lo contrario, por lo general no le permite.
3. Si el datos a modificar hacen parte de la llave primaria, no lo permite.
4. Si se desea quitar un campo que pertenece a la llave primaria, no lo permite.

Sin embargo, todo lo que sea adiciones de campos, lo permite.

La pregunta que en un momento dado, se puede hacer es: Cómo adicionar un campo a la llave primaria o cambiar su tipo de datos? La respuesta a esta pregunta, es que no se puede hacer desde la sentencia ALTER con las tablas llenas. Por lo tanto, el administrador de la base de datos debe hacer unos procesos previos, tales como, sacar copia de la base de datos, borrar todas las filas de la tabla o tablas que estén relacionadas con la tabla a modificar, realizar la modificación y volver a cargar los datos de la copia a la tabla o tablas modificadas. Otra pregunta que surge es: Qué pasa con las aplicaciones que utiliza la tabla o las tablas relacionadas con ésta? Para ellos el programador, debe entrar a realizar todas las modificaciones en las sentencias de consultas, donde intervienen estas tablas, pues la condición de reunión que se basa en los campos de las llaves primarias, cambian necesariamente por la adición de un campo a la llave primaria.

Haciendo un análisis de la situación anterior, se puede observar que realizar cualquier tipo de modificación a un campo que pertenece o se agrega a la llave primaria, es compleja y trae mucho traumatismo, además de lo delicado de los procesos que se deben realizar esto. Por eso, es importante realizar un buen análisis y diseño, antes de empezar a implementar cualquier base de datos.

Ahora se muestra las sentencias para realizar tres tipos de modificaciones:

1. Adicionar la ciudad a la tabla de HUESPEDES.

```
ALTER TABLE HUESPEDES ADD Ciudad CHAR (30);
```

2. Borrar el número de habitación a las RESERVAS.

```
ALTER TABLE RESERVAS DELL Nto_hab;
```

3. Modificar el tipo de datos días de duración en las RESERVAS.

```
ALTER TABLE RESERVAS MOD Dias_Duracion CHAR (15);
```

Lección 35: “Borrados de Bases de Datos, Tablas, Índices y Vistas”

Para el borrado de cualquier elemento de la estructura de una base de datos, incluyendo ella misma, existe la sentencia **DROP**, al igual que las modificaciones, existen motores que restringen o se aseguran antes que esto es lo que quiere realizar el usuario. Obsérvese algunas de ellas:

1. Cuando se desea borrar todos los elementos de una base de datos, hay motores que no lo permiten si ésta está llena. Otros, pregunta sobre si realmente lo desea hacer.
2. Cuando se desea borrar solo tablas, sucede lo mismo que el caso anterior.
3. Cuando se desea borrar un índice o una vista, algunos motores se asegura con un pregunta, si desean efectuar dicha operación. No tiene restricción si la base de datos tiene o no tiene datos, ya que el índice y la vista, se puede rehacer sin ningún problema.

La forma como se borran los diferentes elementos son:

1. Borrado de una base de datos

DROP DATABASE RESERVAS_HOTELES;

2. Borrado de una tabla

DROP TABLE MUNICIPIOS;

3. Borrado de un índice

DROP INDEX HUESPEDES_Nro_Id_Hues;

4. Borrado de una vista

DROP VIEW VISTA_RESERVAS;

CAPÍTULO 8: “**LENGUAJE DE MANIPULACIÓN DE DATOS**”

El lenguaje de manipulación de datos, es el que permite adicionar, borrar, modifica y consultar los datos, en la base de datos. Este lenguaje puede ser utilizado en dos forma: Embebido e Interactivo. El primero, es cuando se utilizan las sentencias dentro de un lenguaje procedimental (C, Pascal, Cobol, Java, Visual Basic etc.) y el segundo es cuando a través de la interface de motor de la base de datos, se utilizan las sentencia en línea, es decir, inmediatamente se ejecuta.

Lectura Recomendada tomada del libro Introducción a las bases de datos de Ma. Victoria Nevado Cabello.

[Dar clic aqui para leer](#)

Lección 36: “**Inserción de Filas**”

La operación de inserción, es la que permite crear los datos en las bases de datos. Para garantizar el éxito de esta operación, es necesarios tener en cuentas estas situaciones:

1. La sentencia, solo permite insertar datos en UNA sola tabla.
2. Como mínimo, se debe insertar los datos de los campos que conforman la LLAVE PRIMARIA.
3. Si existen campos, que no permiten valores NULOS, estos deben necesariamente, insertales valores.
4. Si la tabla, donde se van a insertar los datos, contiene LLAVES FORÁNEAS, debe garantizarse que al introducir un datos, este ya este creado en la tabla padre. Por lo tanto, es conveniente que al realizar inserciones en este tipo de tablas, lo primero que se debe hacer es insertar en las tablas que no tienen LLAVES FORANEAS, es decir, que son solo padres.

Cuando se insertan datos a una base de datos, se puede hacer de dos formas:

1. Insertando datos Fila por Fila

Se inserta la primera fila de la tabla de MUNICIPIOS y la segunda fila de la tabla de HOTELES.

INSERT INTO MUNICIPIOS

(Cod_Mun,Nombre_municipio)

VALUES(01,'Bogotá');

INSERT INTO HOTELES

(Cod_Hot,Hnombre,Cod_Mun)

VALUES('H02','Hilton',01);

Observese en las instrucciones anteriores, que como son los primeros registros, antes de ingresar datos a la tabla HOTELES, debe haberse creado ingresado datos en la tabla de MUNICIPIOS. Sin embargo, si se desea llenar la tabla de HOTELES, sin llenar la de MUNICIPIOS, ha debido declarar en la estructura de tabla de HOTELES, que el campo Cod_Mun puede tener valores NULOS. De tal forma que si esta es la situación, entonces lo anterior puede quedar así:

INSERT INTO HUESPEDES

(Nro_Id_Hues,Nombre_Hues)

VALUES(39153037,'Maria Pérez');

Otra cosa a tener en cuenta, es que la sentencia INSERT, puede o no especificar el nombre de los campos así:

INSERT INTO MUNICIPIOS

VALUES(01,'Bogotá');

Sin embargo, se debe tener mucho cuidado con esta instrucción, pues cuando se usa así, el programador debe tener claro la POSICIÓN que tiene cada campo. Por lo tanto, aunque técnicamente se pueda realizar, no se recomienda para usarlo en

tablas con varios campos y cuando se utiliza en forma embebida en un lenguaje de programación.

2. Insertando datos en varias filas

Cuando se utiliza la sentencia INSERT para agregar datos en varias filas, necesariamente se debe tomar con base en otra tabla, que tenga los campos estructurados de la misma forma de la tabla donde se va a insertar. Un ejemplo sería cargar una tabla temporal solo con los datos de número de reservas, número del huésped y fecha de reservas, cuya fecha inicial de reserva mayor a julio de 2011. Lo anterior quedaría así:

INSERT

INTO Temporal (Rnum, Nro_Id_Hues, Fecha_Ini)

SELECT Rnum, Nro_Id_Hues, Fecha_Ini

FROM RESERVAS

WHERE Fecha_Ini > '01/07/2011';

Es de aclarar, que la tabla Temporal, se ha debido crear previamente, con la sentencia CREATE.

Lección 37: “Actualización de Filas”

La sentencia para actualizar se llama **UPDATE**. Cuando se ejecuta esta sentencia en su mínima expresión, se actualizan todas las filas. Obsérvese el siguiente ejemplo:

Actualizar el nombre del hotel “Caribbean”, por “Palacé”

UPDATE HOTELES

SET Hnombre = 'Palacé'

WHERE Hnombre = 'Caribbean';

Aquí es conveniente hacer varias precisiones. La cláusula WHERE, es opcional; si esta cláusula no se coloca, entonces todas las filas quedan con nombre “Palacé”, cosa que no es común que ocurra. En cambio, con la cláusula WHERE, solo cambia el que tiene el nombre de ‘Caribbean’, siendo así las cosas, podemos decir que la cláusula WHERE en la práctica, no es tan opcional.

Lección 38: “Borrado de Filas”

La clausula para borrar filas es **DELETE**. Cuando se ejecuta esta sentencia en su mínima expresión, se borran todas las filas. Obsérvese el siguiente ejemplo:

Vamos a borrar todas las filas de la tabla de reservas cuya fecha de inicio son del año 2010.

```
DELETE FROM HOTELES
```

```
WHERE CIUDAD = ‘Armenia’;
```

Aquí vale la pena hacer la misma aclaración que hicimos con UPDATE, y es que la cláusula WHERE es opcional. De tal forma, que si no se coloca, entonces BORRA TODOS LOS DATOS DE LA TABLA, pero no la estructura.

Lección 39: “Consultas Basicas”

Para mirar toda la potencialidad de la sentencia SELECT, hemos organizado las consultas de menor a mayor grado de dificultad, por cuestiones pedagógicas. Comenzaremos con consultas sobre una sola tabla hasta finalizar con subconsultas, donde intervienen más de una tabla. Miremos la sentencia general

```
SELECT Campo1[,Campo 2, .....]
```

```
FROM Tabla 1[,Tabla 2, .....]
```

```
WHERE Condición 1 [AND Condición 2][OR Condición 2][.....]
```

```
ORDER BY Campo 1 [,Campo 2, ....]
```

```
GROUP BY Campo 1 [,Campo 2, ....]
```

```
HAVING Condición de Grupo [AND Condición 2][OR Condición 2][.....]
```

1. Consultas simples

- a. Consulta de un determinado campo. Consultar el código, nombre y código municipios de todos los hoteles.

```
SELECT Cod_Hot, Hnombre, Cod_Mun  
FROM HOTELES;
```

- b. Consultas de todos los campos. Consultar todos los datos de los hoteles.

```
SELECT *  
FROM HOTELES;
```

- c. Consultas con DISTINCT. Consultar el código de los hoteles que han tenido reservas.

```
SELECT DISTINCT Cod_Hot  
FROM RESERVAS;
```

Si no se coloca la palabra **DISTINCT**, saldría dos veces el código H01.

- d. Consultas de valores calculados. Consultar el nombre de los hoteles con sus respectivas tarifas, aumentadas en un 10%.

```
SELECT Hnombre, "Tarifa Aumentada en 10%",Tarifa_Noche*1.1  
FROM HOTELES;
```

- e. Consultas con campos cualificados. Consultar el código y el nombre de los municipios.

```
SELECT MUNICIPIOS.Cod_Mun,MUNICIPIOS.Nomb_Mun  
FROM MUNICIPIOS;
```

- f. Consultas con condición. Consultar el código del hotel que han reservados en el año 2011

```
SELECT DISTINCT Cod_Hot
FROM RESERVAS
WHERE Fecha_Ini='01/01/2011' AND Fecha_Fin='31/12/2011';
```

- g. Consultas con ordenamiento. Consultar el código del hotel y la fecha de reservas, organizado por código hotel.

```
SELECT DISTINCT Cod_Hot, Fec_Res
FROM RESERVAS
ORDER BY Cod_Hot;
```

2. Consultas de Reunión

- a. Reunión simple. Consultar el nombre de los hoteles que han tenido reservas

```
SELECT DISTINCT Hnombre
FROM HOTELES,RESERVAS
WHERE HOTELES.Cod_Hot=RESERVAS.Cod_Res;
```

Siempre que se tiene en una consulta la intervención de mas de una tabla, necesariamente debe haber Condiciones de Reunión. Para esto, el número de condiciones de reunión siempre van hacer igual al numero de las tablas que intervienen (n), menos uno (1). En este caso, se tienen dos tablas, menos uno, entonces debe haber una condición de reunión. Esto NO implica, que puedan haber más condiciones, pero NO de reunión.

También se puede observar, que es necesario cualificar los campos, ya que tanto en la tabla de HOTELES, como de RESERVAS el nombre del código del hotel es idéntico.

Por último, se debe utilizar el **DISTINCT** para que no salgan repetidos el nombre del hotel. Si desea puede quitar el **DISTINCT** y observe el resultado.

- b. Reunión con condición. Consultar el nombre del hotel, el nombre del municipio donde está ubicado con sus respectivas fechas de reserva, solo para las reservas del año 2011.

```
SELECT Hnombre, Nomb_Mun
FROM HOTELES, MUNICIPIOS, RESERVAS
WHERE HOTELES.Cod_Hot=RESERVAS. Cod_Hot AND
      HOTELES. Cod_Mun =MUNICIPIOS. Cod_Mun AND
      Fec_Res>='01/01/2011' AND Fec_Res<='31/12/2011';
```

En esta consulta se observan que intervienen tres tablas, por lo tanto debe haber dos consultas de reunión. También se hace necesario cualificar los campos pues el nombre del código del hotel es el mismo que en la tabla de reservas y lo mismo sucede con el código municipio, pero en la tabla de hoteles. Se debe tener en cuenta que cuando se tienen más de una condición de reunión, están deben estar unidas por el operador **AND**, lo mismo que cuando se tienen otras condiciones, que a veces se hace necesario colocarlas entre paréntesis, si la condición utiliza el operador **OR**.

- c. Reunión de una tabla consigo misma. Consultar las parejas de códigos del hotel, que se encuentran en la misma ciudad.

```
SELECT PRIMERA.Cod_Hot,SEGUNDA. Cod_Hot
FROM HOTELES PRIMERA, HOTELES SEGUNDA
WHERE PRIMERA.Cod_Hot<SEGUNDA.Cod_Hot AND
      PRIMERA. Cod_Mun =SEGUNDA. Cod_Mun;
```

Nota: Obsérvese que el JOIN, es decir, la IGUALACIÓN, se está realizando por el campo de código municipio y no por el campo llave código hotel. Además, se utilizan alias en el nombre de las tablas, con el fin de poder combinar la tabla consigo misma.

3. Consultas con Funciones Agregadas.

COUNT(*) :	Cuenta los registros de una tabla
COUNTO (Campo):	Cuenta los registros por un campo determinado. Si el campo tiene un valor nulo, no lo cuenta.
SUM (Campo):	Suma todos los valores del campo. No suma nulos. el campo debe ser numérico.
AVG (Campo):	Promedia el campo. No tiene en cuenta los nulos y el campo debe ser numérico.
MAX (Campo):	Selecciona el valor máximo del campo. No cuenta el valor nulo.
MIN (Campo):	Selecciona el valor mínimo del campo. No cuenta el valor nulo.

a. Consultas para contar filas. Consultar el número total de hoteles.

```
SELECT COUNT (*)  
FROM HOTELES;
```

b. Consultas para contar final con condición. Obtener la cantidad total de hoteles que se encuentran en el municipio H01.

```
SELECT COUNT(Cod_Hot)
FROM HOTELES
WHERE Cod_Mun='H01' ;
```

- c. Consultas con agrupación. Obtener el número total de reservas por hotel

```
SELECT Cod_Hot,COUNT(Nro_Res)
FROM RESERVAS
GROUP BY Cod_Hot;
```

En la consulta anterior, cuando tenemos en la cláusula **SELECT** campos acompañados con funciones agregadas, *siempre debe agruparse por estos campos*, o si no, sale error en la consulta y NO se ejecuta.

- d. Consultas con agrupación y reunión de tablas. Consultar el nombre del hotel con su respectivo número total de reservas.

```
SELECT Hnombre,COUNT(Nro_Res)
FROM HOTELES,RESERVAS
WHERE HOTELES.Cod_Hot=RESERVAS.Cod_Hot
GROUP BY Hnombre;
```

En esta consulta, tenemos el mismo caso de la cláusula **SELECT**. Campos acompañados con funciones agregadas, *siempre debe agruparse por ellos*, o si no, sale error en la consulta y NO se ejecuta.

- e. Consultas con condición de agrupación. Consultar el nombre del hotel, con su respectiva cantidad total de días reservados; solos aquellos hoteles que tienen más de dos días reservados.

```
SELECT Hnombre,SUM(Dias_Dur)
FROM HOTELES,RESERVAS
WHERE HOTELES.Cod_Hot=RESERVAS.Cod_Hot
GROUP BY Hnombre
HAVING SUM(Dias_Dur)>2;
```

En esta consulta, tenemos el mismo caso de la clausula **SELECT**. Campos acompañados con funciones agregadas, *siempre debe agruparse por ellos*, o si no, sale error en la consulta y NO se ejecuta.

Aquí la clausula **HAVING** se ejecuta después de que el motor ha hecho el agrupamiento y sumado los días de duración, es decir, esta condición es de grupo, no por fila como la **WHERE** .

- f. Consultas con condición y agrupamiento varios. Consultar el nombre del hotel, y el nombre de los municipios, con su respectiva cantidad de reservas. Solo los hoteles del municipio de Cartagena o Bogotá, y que tengan más de 2 reservas.

```
SELECT Hnombre,Nomb_Mun,SUM(Dias_Dur)
FROM HOTELES,MUNICIPIOS,RESERVAS
WHERE HOTELES.Cod_Hot=RESERVAS.Cod_Hot AND
      HOTELES.Cod_Mun=MUNICIPIOS.Con_Mun AND
      (MUNICIPIOS.Nomb_Mun='Cartagena' OR
      MUNICIPIOS.Nomb_Mun='Bogotá')
```


GROUP BY Hnombre, Nomb_Mun

HAVING SUM(Dias_Dur)>1;

En esta consulta, tenemos el mismo caso de la cláusula **SELECT**. Campos acompañados con funciones agregadas, *siempre debe agruparse por ellos*, o si no, sale error en la consulta y NO se ejecuta.

Aquí la cláusula **HAVING** se ejecuta después de que el motor ha hecho el agrupamiento y sumado los días de duración, es decir, esta condición es de grupo, no por fila como la **WHERE**.

Lección 40: “Consultas Avanzadas”

- a. Consultas con el operador LIKE. Consultar los hoteles cuyo nombre empiezan con la letra ‘A’.

SELECT *

FROM HOTELES

WHERE Hnombre **LIKE** 'A%';

- b. Consultar los nombres de los hoteles cuya segunda letra sea ‘l’.

SELECT *

FROM HOTOLES

WHERE Hnombre **LIKE** '_A%';

- c. Consultar los nombres de los Municipios cuya última letra sea igual a ‘E’.

SELECT *

FROM MUNICIPIOS

```
WHERE Nomb_Mun LIKE '%E' ;
```

d. Consultar los proveedores que tengan en el nombre la letra 'S'.

```
SELECT *  
FROM MUNICIPIOS  
WHERE Nomb_Mun LIKE '%t%' ;
```

e. Subconsultas. Consultar los hoteles que no han tenido reservas.

```
SELECT *  
FROM HOTELES  
WHERE Cod_Hot NOT IN  
    (SELECT Cod_Hot  
     FROM RESERVAS) ;
```

Nota: Esta consulta se puede realizar también utilizando el EXISTS

```
SELECT *  
FROM HOTELES  
WHERE NOT EXISTS  
    (SELECT Cod_Hot  
     FROM RESERVAS  
     WHERE RESERVAS.Cod_Hot=HOTELES. Cod_Hot)
```

6.2. OTRAS OPERACIONES

- a. Crear tablas temporales con INSERT INTO. Crear una tabla temporal llamada TEMP_RESERVAS, que contenga el nombre del huésped, hotel y la cantidad total de días de reservados.

```
SELECT Nombre_Hues,Hnombre SUM(Dias_Duracion) AS TOT_DIAS  
INTO TEMP_RESERVAS  
FROM HUESPEDES, RESERVAS, HOTELES  
WHERE HUESPEDES.Nro_Id_Hues=RESERVAS. Nro_Id_Hues AND  
HOTELES.Cod_Hot=RESEVAS. Cod_Hot  
GROUP BY Nombre_Hues,Hnombre;
```

- b. Crear una vista con CREATE VIEW. Crear una vista con todas las partes y su respectivo número total de proyectos que han suministrado y cantidad total suministrada.

```
CREATE VIEW VISTA_PARTES AS  
(SELECT PARTE,COUNT(YNRO),SUM(CANT)  
FROM PARTES,SUMINISTROS  
WHERE PARTES.PNRO=SUMINISTROS.PNRO  
GROUP BY PARTE);
```

- c. Consultar sobre una tabla temporal. Consultar la mayor cantidad suministrada

```
SELECT MAX(TOT_CANT)  
FROM TEMP_PROV;
```

Anexo No 1.

Modelo relacional Caso Reserva.

Reservas

Rnum	Fecha_Ini	Dias_Dur	Nro_Id_Hues	Cod_Hot	Nro_Hab	Fec_Res
001	23-09-10	3	39153037	H01	2-12	30-03-10
002	01-10-11	2	72044926	H02	3-01	01-01-10
003	13-08-11	2	72982340	H01	4-02	10-03-10
005	06-12-11	5	70425365	H03	1-30	06-06-10

Hoteles

Cod_Hot	Hnombre	Cod_Mun	Tarifa_Noche (Pesos)
H01	Caribe	02	200000
H02	Hilton	01	180000
H03	Las Americas	02	210000
H04	Nutibara	03	150000
H05	Caribbean	04	120000

Municipios

Cod_Mun	Nomb_Mun
01	Bogotá
02	Cartagena
03	Medellín
04	Barranquilla

Huéspedes

Nro_Id_Hues	Nombre_Hues
39153037	María Perez
72044926	Pedro Acosta
72982340	Jose Ortíz
45922444	Ana Nuñez
70425365	Rafael Leconte

CAPÍTULO 9: “HERRAMIENTA MySQL”

Lección 41: “Generalidades de la Herramienta”

MySQL AB es una empresa comercial fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación. MySQL AB se estableció originalmente en Suecia por David Axmark, Allan Larsson, y Michael “Monty” Widenius

MySQL, el sistema de gestión de bases de datos SQL Open Source más popular, lo desarrolla, distribuye y soporta MySQL AB.

Es un sistema de gestión de bases de datos relacionales. Es software Open Source. El servidor de base de datos MySQL es muy rápido, fiable y fácil de usar. Desarrollado originalmente para tratar grandes bases de datos mucho más rápido que soluciones existentes. Trabaja en entornos cliente/servidor o incrustados.

Lección 42: “Creación de Bases de Datos y Tablas”

[Ver documento](#)

Lección 43: “Inserción, Actualización y Borrado de Filas”

[Ver documento](#)

Lección 44: “Consultas Básicas”

[Ver documento](#)

Lección 45: “Consultas Avanzada”

[Ver documento](#)

ACTIVIDADES DE AUTOEVALUACIÓN DE LA UNIDAD

De acuerdo al siguiente esquema, realizar las siguientes consultas utilizando álgebra relacional con el siguiente esquema de bases de datos.

Esquema de la base de datos “**LABORATORIOS**”

ESTUDIANTES (IdEst, Nombre, Grupo)

PRACTICAS (IdPractica, Curso, Fecha)

REALIZA (IdEst, IdPractica, Nota)

Enunciados de consulta:

- Nombre de los estudiantes que aprobaron todas las prácticas del curso Bases de Datos.
- Nombre de los estudiantes que realizaron todas las prácticas del curso Bases de Datos.
- Nombre de los estudiantes que han realizado prácticas de Bases de Datos y de Física.
- Nombre de los estudiantes que sólo han realizado prácticas de Física.
- Nombre de los estudiantes que han realizado por lo menos una práctica de Bases de Datos, de Física y de Algoritmos.
- Nombre de los estudiantes que pertenecen al grupo 10 del curso Algoritmos.
- Nombre de los estudiantes que no han aprobado ninguna práctica.
- Listado de prácticas junto con el grupo al que pertenecen, en una fecha específica.
- Listado de estudiantes de todos los grupos de Física.
- Nombre de los estudiantes que están inscritos en un único curso.

FUENTES DOCUMENTALES

BIBLIOGRAFÍA

- Silberstchatz y otros, Fundamentos de bases de datos, vol. I. Editorial Mc Graw Hill. 1998.
- C.J. Date, "Introducción a los sistemas de bases de datos", vol I. Madrid: Addison Wesley, 1993.
- Effy Oz, "Administration de sistemas de información", Madrid: Prentice Hall, 2001.
- Kendal y Kendal, "Análisis y diseño de sistemas,", Mexico: Prentice Hall, 2006.
- RUIZ L.,Irene y otros. "Bases de datos, desde chen hasta Codd con Oracle". Alfaomega Ra-Ma,2003.
- PRESSMAN, James. "Ingeniería de Software, un enfoque práctico". McGraw – Hill. 2003.
- Castaño, A., Piattini M. "Fundamentos y modelo de bases de datos", Ed. Alfaomega S.A. 1998.
- Rodriguez M. "Bases de datos". Editorial Mc Graw Hill. 1992.
- Figeredo, J.A. "Módulo de "Bases de Datos I" de la UNAD, 1ª. Edición. 2005.

CIBERGRAFÍA

- http://dis.unal.edu.co/profesores/eleon/cursos/BD/presentaciones/teo3_modelo_er.pdf
- <http://www.calasanz-pereira.edu.co/index.php/Informatica/entidadrelacion.html>
- <http://www.desarrolloweb.com/articulos/modelo-entidad-relacion.html>
- <http://basdatos.tripod.com/ejercicios.html>
- <http://www.mitecnologico.com/Main/DiagramasEntidadRelacionER>
- <http://ict.udlap.mx/people/carlos/is341/bases03.html>
- <http://www3.uji.es/~mmarques/f47/apun/node43.html>
- <http://www.slideshare.net/guestd19144b/modelo-relacional-1128750>
- <http://www.uazuay.edu.ec/analisis/EI%20modelo%20relacional.pdf>
- http://www.tejedoresdelweb.com/wiki/images/a/a5/Basesdatos_teo5_modelo_relacional.pdf
- <http://usuarios.lycos.es/cursosgbd/UD3.htm>
- <http://www.programacion.com/bbdd/tutorial/modrel/>
- <http://usuarios.lycos.es/cursosgbd/UD3.htm>
- <http://www3.uji.es/~mmarques/f47/apun/node43.html>
- http://pisis.unalmed.edu.co/cursos/material/3004590/1/ejer_alg.pdf
- http://dis.unal.edu.co/profesores/eleon/cursos/BD/presentaciones/teo6_algebra_relacional.pdf
- <http://algebrarelacional.awardspace.com/Algebra%20Relacional.htm>
- http://www.unalmed.edu.co/~mstabare/Algebra_Rel.htm
- <http://www.scribd.com/doc/2450884/Algebra-Relacional>
- <http://cnx.org/content/m18351/latest/>

- www.postgresql.org
- www.programacion.com
- www.lawebdelprogramador.com
- <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiplehtml/x332.html>
- <http://www.tramullas.com/nautica/documatica/2-7.html>
- http://atenea.udistrital.edu.co/profesores/jdimate/basedatos1/tema3_1.htm
- http://dis.um.es/~barzana/Informatica/IAGP/IAGP_BD_Relacional.html

AUTOR

María Dora Alba Sánchez Gomez

Ingeniera de Sistemas Universidad E.A.F.I.T.

Magistra en administración de empresa, Universidad del Norte

Electronic mail: mariadora.sanchez@unad.edu.co

Teléfono Oficina: (034) 2910273 Medellín

ANEXOS

Reservas

Rnum	Fecha_Ini	Dias duracion	Nro_Id_Hues	Cod_Hot	Nro_Hab	Fec_Res
001	23-09-10	3	39153037	H01	2-12	30-03-10
002	01-10-11	2	72044926	H02	3-01	01-01-10
003	13-08-11	2	72982340	H01	4-02	10-03-10
005	06-12-11	5	70425365	H03	1-30	06-06-10

Hoteles

Cod_Hot	Hnombre	Cod_Mun
H01	Caribe	02
H02	Hilton	01
H03	Las Americas	02
H04	Nutibara	03
H05	Caribian	04

Municipios

Cod_Mun	Nombre Municipio
01	Bogotá
02	Cartagena
03	Medellín
04	Barranquilla

Huéspedes

Nro_Id_Hues	Nombre_Hues
39153037	María Perez
72044926	Pedro Acosta
72982340	Jose Ortíz
45922444	Ana Nuñez
70425365	Rafael Leconte