



UNIVERSIDAD DE PAMPLONA

Facultad de Estudios a Distancia

Programas de Estudio a Distancia



Administración de Base de Datos

www.unipamplona.edu.co

Esperanza Paredes Hernández

Rectora

María Eugenia Velasco Espitia

Decana Facultad de Estudios a Distancia

Tabla de Contenido

Presentación
Introducción
Horizontes

UNIDAD 1: Fundamentación de las Bases de Datos

Descripción Temática

Horizontes

Núcleos Temáticos

Proceso de Información

1.1 DATOS E INFORMACIÓN

1.2 ¿QUÉ ES INFORMACIÓN?

1.3 MANEJO DE LA INFORMACIÓN

1.3.1 Como Recurso

1.3.2 Generada por Computadora

1.4 ¿CÓMO GENERAR LA INFORMACIÓN?

1.4.1 Características de la Información Útil

1.5 LAS CUATRO ETAPAS DEL PROCESO

1.6 SISTEMAS DE INFORMACIÓN EN LAS EMPRESAS

1.6.1 Sistemas de Información en Áreas Funcionales de Empresas

1.6.2 Sistemas de Información de Diferentes Sectores Empresariales

1.7 ADMINISTRACIÓN DE DATOS Y ADMINISTRACIÓN DE BASES DE DATOS

1.8 SISTEMA DE BASE DE DATOS

1.8.1 ¿Qué es un Sistema de Base de Datos?

1.8.2 Definición de Base de Datos

1.8.3 Componentes Principales

1.8.4 ¿Por qué Utilizar una Base de Datos?

1.8.5 Ventajas en el Uso de Bases de Datos

1.9 EL SISTEMA MANEJADOR DE BASES DE DATOS (DBMS)

1.9.1 El Administrador de La Base de Datos (DBA)

1.10 TIPOS DE MODELOS DE BASE DE DATOS

1.10.1 Modelo Jerárquico

1.10.2 El Modelo de Red

1.10.3 El Modelo Relacional

1.10.4 El Modelo Orientado Hacia Objetos

1.10.5 Algunas Bases de Datos

1.11 ¿QUÉ ES UNA BASE DE DATOS?

1.11.1 ¿Por Qué Utilizar una Base de Datos?

1.11.2 Ventajas del Enfoque de Bases de Datos

1.11.3 Ejemplo de Utilización de Base de Datos

Proceso de Comprensión y Análisis

Síntesis Creativa y Argumentativa

Solución de Problemas

Autoevaluación

Repaso Significativo

Bibliografía Sugerida

UNIDAD 2: Modelado de Datos

Descripción Temática

Horizontes

Núcleos Temáticos

Proceso de Información

2.1 DESCRIPCIÓN DEL CICLO DE RECEPCIÓN VENTA / EFECTIVO

2.1.1 De las Entidades a las Tablas

2.1.2 Establecer las Comprobaciones de Validación

2.2 DISEÑO CONCEPTUAL DE BASE DE DATOS

2.2.1 Realidad y Modelos

2.2.2 Modelos Conceptuales de Datos

Proceso de Comprensión y Análisis

Solución de Problemas

Síntesis Creativa y Argumentativa

Autoevaluación

Repaso Significativo

Bibliografía Sugerida

UNIDAD 3: Sistemas Relacionales

Descripción Temática

Horizontes

Núcleos Temáticos

Proceso de Información

3.1 CONCEPTOS FUNDAMENTALES

3.2 PROCESO DE NORMALIZACIÓN

3.2.1 Primera Forma Normal (1FN)

3.2.2 Segunda Forma Normal (2FN)

3.2.3 Tercera Forma Normal (3FN)

3.2.4 Cuarta Forma Normal (4FN)

3.2.5 Quinta Forma Normal (5FN)

3.3 INTEGRIDAD RELACIONAL

3.3.1 Tipo de Restricciones de Integridad

3.4 ALGEBRA Y CALCULO RELACIONAL

3.4.1 Unión

3.4.2 Intersección

3.4.3 Diferencia

3.4.4 Producto

3.4.5 Selección o Restricción

3.4.6 Proyección

3.4.7 Reunión

3.4.8 División

3.4.9 Asignación

Proceso de Comprensión y Análisis

Solución de Problemas

Síntesis Creativa y Argumentativa

Autoevaluación

Repaso Significativo

Bibliografía Sugerida

UNIDAD 4: Implementación de Bases de Datos Relacionales

Descripción Temática

Horizontes

Núcleos Temáticos

Proceso de Información

4.1 INTRODUCCIÓN A LA IMPLEMENTACIÓN RELACIONAL

4.2 TIPOS DE DATOS

4.2.1 Términos

4.3 DEFINICIÓN DE TABLAS

4.4 MANIPULACIÓN DE DATOS

4.5 DISEÑO DE SISTEMAS DE BASE DE DATOS RELACIONALES

4.6 MANEJO DE MOTOR DE BASES DE DATOS ACCESS

4.6.1 Definiciones Principales

4.6.2 Elementos de una Base de Datos de Ms Access

4.6.3 Manejo de Access

Proceso de Comprensión y Análisis

Solución de Problemas

Síntesis Creativa y Argumentativa

Autoevaluación

Repaso Significativo

Bibliografía Sugerida

Presentación

La educación superior se ha convertido hoy día en prioridad para el gobierno Nacional y para las universidades públicas, brindando oportunidades de superación y desarrollo personal y social, sin que la población tenga que abandonar su región para merecer de este servicio educativo; prueba de ello es el espíritu de las actuales políticas educativas que se refleja en el proyecto de decreto Estándares de Calidad en Programas Académicos de Educación Superior a Distancia de la Presidencia de la República, el cual define: "Que la Educación Superior a Distancia es aquella que se caracteriza por diseñar ambientes de aprendizaje en los cuales se hace uso de mediaciones pedagógicas que permiten crear una ruptura espacio temporal en las relaciones inmediatas entre la institución de Educación Superior y el estudiante, el profesor y el estudiante, y los estudiantes entre sí".

La Educación Superior a Distancia ofrece esta cobertura y oportunidad educativa ya que su modelo está pensado para satisfacer las necesidades de toda nuestra población, en especial de los sectores menos favorecidos y para quienes las oportunidades se ven disminuidas por su situación económica y social, con actividades flexibles acordes a las posibilidades de los estudiantes.

La Universidad de Pamplona gestora de la educación y promotora de llevar servicios con calidad a las diferentes regiones, y el Centro a Distancia de la Universidad de Pamplona, presentan los siguientes materiales de apoyo con los contenidos esperados para cada programa y les saluda como parte integral de nuestra comunidad universitaria e invita a su participación activa para trabajar en equipo en pro del aseguramiento de la calidad de la educación superior y el fortalecimiento permanente de nuestra Universidad, para contribuir colectivamente a la construcción del país que queremos; apuntando siempre hacia el cumplimiento de nuestra visión y misión como reza en el nuevo Estatuto Orgánico:

Misión: Formar profesionales integrales que sean agentes generadores de cambios, promotores de la paz, la dignidad humana y el desarrollo nacional.

Visión: La Universidad de Pamplona al finalizar la primera década del siglo XXI, deberá ser el primer centro de Educación Superior del Oriente Colombiano.

Maria Eugenia Velasco Espitia – Directora CEDUP

Introducción

Las necesidades de información de nuestra sociedad actual se dejan sentir de forma cada vez más imperiosa. El problema de la información está estrechamente relacionado con el desarrollo económico y social. La investigación, la planificación y la toma de decisiones exigen una información precisa, oportuna, coherente y adaptada a las necesidades específicas de cada usuario y de cada circunstancia.

Las bases de datos se desarrollan como respuesta al planteamiento de las organizaciones para la gestión de datos, buscando mejorar las prestaciones de los sistemas informáticos y aumentar su rendimiento.

El presente módulo llamado Base de Datos se estructura en cuatro unidades, la primera se titula Fundamentos de las Bases de Datos; la segunda, Modelado de Datos; la tercera, Sistemas Relacionales, y la cuarta, Implementación de Datos Relacionales, donde se pondrá en la práctica la información estudiada en el modulo, apoyados del motor de bases de datos Microsoft Access.

Horizontes

- Conocer las herramientas básicas necesarias para el diseño y desarrollo de Bases de Datos
- Adquirir destreza para el diseño e implementación de bases de datos relacionales
- Transformar un modelo de datos en un modelo de datos relacional.
- Explicar algunos elementos básicos de la estructura de un esquema de información SQL

UNIDAD 1

Fundamentación de las Bases de Datos

Descripción Temática

Las organizaciones han reconocido, desde hace mucho tiempo, la importancia de administrar recursos principales como mano de obra y las materias primas. La información se ha colocado en un lugar adecuado como recurso principal. Los tomadores de decisiones están comenzando a comprender que la información no es sólo un subproducto de la conducción, sino que a la vez alimenta a los negocios y puede ser el factor crítico para la determinación del éxito o fracaso de éstos.

En la presente unidad se exponen los conceptos básicos necesarios para empezar en el estudio de las bases datos, dado cada día este sistemas de información va acrecentando su importancia debido a los grandes beneficios y estupendas ventajas que genera su aplicación

Horizontes

- Conocer y comprender la importancia de la utilización de las bases de datos en el desarrollo de las actividades de la organización.
- Analizar los conceptos y objetivos que involucra un sistema de base de datos.
- Explicar como los cuatro componentes (hardware, software, los datos y las personas) se integran para constituir los sistemas de bases de datos actuales.
- Tener un conocimiento preliminar con algunas características comunes de los sistemas generales de bases de datos.

Núcleos Temáticos y Problemáticos

- Datos e Información
- ¿Qué es Información?
- Manejo de la Información
- ¿Cómo Generar la Información
- Las Cuatro Etapas del Proceso
- Sistemas de Información en las Empresas
- Administración de Datos y Administración de Bases de Datos
- Sistema de Base de Datos
- El Sistema Manejador de Bases de Datos (DBMS)
- Tipos de Modelos de Base de Datos

Proceso de Información

1.1 DATOS E INFORMACIÓN

Una computadora o sistema computacional es un conjunto de tecnologías capaz de efectuar cuatro operaciones:

- Aceptar entradas: reciben la información del mundo exterior.
- Producir salidas: dan información al mundo externo.
- Procesar información: llevan a cabo operaciones aritméticas y lógicas, toman decisiones con la información y un sentido a tal información.
- Almacenar información: mueven y almacenan la información de y hacia la memoria de la computadora y a los dispositivos de almacenamiento.

Al trabajar juntas estas tecnologías forman un sistema computacional, capaz de darle significado a la información del exterior.

Todo sistema de cómputo tiene componentes electrónicos y mecánicos. A estos componentes los conocemos como Hardware de la computadora.

Adicionalmente al hardware, es necesario otro componente básico llamado Software. A pesar de que el hardware tiene una gran capacidad de procesamiento

de información, no puede efectuar ninguna operación si no se le dan instrucciones para hacerlo. A estas instrucciones las conocemos con el nombre de Software.

Los términos “datos” e “información” no significan lo mismo. Dato es el plural en español de la palabra latina datum, que significa literalmente “algo dado”, o un hecho que puede tomar la forma de un número, un enunciado o una imagen. Los datos son la materia prima en la producción de información.

Por otra parte, la información son datos que adquieren significado dentro de un contexto. La información puede consistir en datos primarios o datos manipulados a través de la suma, la resta, la división o cualquier otra operación que conduzca a una mayor comprensión de una situación.

1.2 ¿QUÉ ES INFORMACIÓN?

Para aplicar las tecnologías de la información, es importante identificar en primera instancia el valor que representa la información para el individuo, para la sociedad, y, por ende para una empresa y país.

Desde nuestra gestación y hasta nuestra muerte, aprendemos a utilizar la información para conocernos a nosotros mismos y al medio ambiente que nos rodea. Con ello adquirimos habilidades y destrezas para transformarlas en conocimiento para nuestro desarrollo y sobrevivencia, y llega así no sólo a construir los pilares en los que se basa la operación diaria de una empresa sino también a perfilar la idiosincrasia de un país (estilos de vida, religiones, costumbres y herencias culturales).

Esta información, definida como un conjunto de datos seleccionados que se unen con el objeto de generar un conocimiento o comunicar un significado preciso, es apreciada de manera distinta por cada cultura.

Ejemplo: un ejecutivo de Dresdner Black comenta “todavía hay directivos que tardan en enterarse de cambios en los mercados, y muchas veces toman decisiones basados en datos que no están al día, aunque los sistemas tengan todas las posibilidades de mantenerse actualidades.

1.3 MANEJO DE LA INFORMACIÓN

1.3.1 Como Recurso

Para maximizar la utilidad de la información, un negocio la debe manejar correctamente tal como maneja los demás recursos. Los administradores necesitan comprender que hay costos asociados con la producción, distribución, seguridad, almacenamiento y recuperación de toda información.

Aunque la información se encuentra a nuestro alrededor ésta no es gratis, y su uso es estratégico para posicionar la competitividad de un negocio.

1.3.2 Generada por Computadora

La fácil disponibilidad de computadoras ha creado una explosión de información a través de la sociedad en general y de los negocios en particular. El manejo de información generada por computadora difiere en forma significativa del manejo de datos producidos manualmente. Por lo general, hay mayor cantidad de información de computadora a administrar. El costo de organizarla y mantenerla puede crecer a tasas alarmantes, y los usuarios frecuentemente la tratan menos escépticamente que la información obtenida por otras vías.

1.3 ¿CÓMO GENERAR LA INFORMACIÓN?

Si analizamos como ejemplo, el cálculo de los totales y promedios de los diferentes tipos de quejas o edades de los compradores podemos deducir que revela tendencias asociadas con los clientes. Estos cálculos son procesos. Un proceso es cualquier manipulación de datos, por lo general con el objetivo de producir información. Sin embargo, algunos procesos producen otro conjunto de datos provisionales.

Por tanto, mientras que los datos son materia prima, la información es una salida. Así como las materias primas se procesan en la producción para crear productos finales útiles los datos primarios se procesan en sistemas para obtener información final útil.

1.4.1 Características de la Información Útil

- Relevante: la información debe relacionarse con el problema por resolver. Por ejemplo, el número total de años de educación tal vez no sea importante para calificar a una persona que pide un trabajo. La información relevante es que la

Persona ha estudiado durante tantos años ingeniería mecánica y los años de experiencia. Además, la información debe presentarse de manera que resulte comprensible para una persona en un contexto específico.

- **Completa:** la información parcial a menudo es peor que la falta de información. Por ejemplo, los datos de mercadotecnia sobre los ingresos familiares pueden llevar a tomar decisiones incorrectas si no se acompañan con información vital sobre los hábitos de consumo de la población estudiada.
- **Precisa:** la información errónea puede conducir a decisiones desastrosas. Por ejemplo, un registro impreciso de la reacción de la penicilina de un paciente puede llevar a un doctor a causarle un daño cuando cree que lo está ayudando.
- **Actual:** a menudo las decisiones se basan en la información más reciente disponible, pero lo que ayer fue una realidad hoy tal vez ya no lo sea. Por ejemplo, optar una inversión a corto plazo y comprar una acción del día de hoy con base en los precios de accionarios de ayer será un error costoso si el precio de la acción ha aumentado.
- **Económica:** en un ambiente de negocios, el costo de obtener la información debe considerarse como un elemento de costo relacionado con cualquier decisión. Por ejemplo, es necesario investigar la demanda de un nuevo producto para reducir el riesgo del fracaso comercial, pero si la investigación de mercado es demasiado costosa, el costo por obtener la información puede disminuir la utilidad obtenida de las ventas.

1.4 LAS CUATRO ETAPAS DEL PROCESO

Todos los sistemas de información operan básicamente de la misma forma, incluyan una computadora o no. Sin embargo, la computadora proporciona un medio conveniente para ejecutar las cuatro operaciones principales de un sistema de información.

- Introducción de datos en el SI (entrada).
- Cambio y manipulación de los datos en el SI (procesamiento de datos).
- Obtención de datos fuera del SI (salida).
- Almacenamiento de datos e información (almacenamiento).

1.5 SISTEMAS DE INFORMACIÓN EN LAS EMPRESAS

Sin excepción cada función de una empresa en todo sector requiere sistemas de información. Desde las oficinas gubernamentales hasta las fábricas, desde la contabilidad hasta la mercadotecnia, los sistemas de información son herramientas clave en el procesamiento de transacciones, la toma de decisiones, la resolución de problemas y las operaciones de todas las organizaciones.

1.6.1 Sistemas de Información en Áreas Funcionales de Empresas

Cada tipo de SI sirve para diferentes propósitos en una organización, en lo que se conoce como "áreas funcionales de una empresa o negocio" (servicios internos que brindan apoyo al negocio principal de una organización). Las áreas funcionales de negocios incluyen, aunque no exclusivamente, contabilidad, finanzas, mercadotecnia y recursos humanos; estas áreas existen en la mayor parte de las compañías, de una forma u otra.

Contabilidad

En contabilidad los sistemas de información ayudan a registrar transacciones de negocios, producir informes financieros periódicos y crear otros requeridos por la ley, como los balances generales y los estados de pérdidas y ganancias. También ayudan a elaborar informes que no son obligatorios legalmente, pero que ayudan a los administradores a entender los cambios en las finanzas de una organización. Los SI de contabilidad poseen controles para asegurar apego a los estándares, como una partida doble (debe-haber).

Finanzas

Mientras que los sistemas de contabilidad se concentran en registrar e informar sobre cambios y estados financieros, el propósito de los sistemas financieros es facilitar la planeación financiera y transacciones de negocios. En finanzas, los sistemas de información ayudan a organizar presupuestos, manejar flujos de caja, analizar inversiones y tomar decisiones que podrían reducir los pagos de intereses y aumentar las ganancias de transacciones financieras.

Mercadotecnia

El propósito de la mercadotecnia es detectar las personas que poseen mayor probabilidad de comprar lo que vende la organización y promover los productos y servicios apropiados para esas personas. Por ejemplo, los sistemas de información de mercadotecnia ayudan a analizar la demanda de varios productos en diferentes

regiones y grupos de poblaciones, para comercializar con mayor exactitud el producto correcto con los clientes meta. Los SI de mercadotecnia proporcionan información que ayuda a la administración a decidir cuantos representantes de ventas asignar para productos específicos en áreas geográficas determinadas. Los sistemas identifican tendencias en la demanda de los productos y servicios de la empresa. También ayudan a responder preguntas del tipo "¿Cómo puede afectar a nuestras ganancias una campaña de publicidad?" la Web ha creado oportunidades excelentes para reunir datos de mercadotecnia, por un lado, y promover productos y servicios desplegando información sobre ellos, por el otro.

Por eso las organizaciones orientan gran parte de sus esfuerzos de mercadotecnia a través de sus SI vinculados con la Web.

Recursos Humanos

Los sistemas de información de recursos humanos ayudan principalmente a mantener registros y evaluaciones de los empleados. Cada organización debe mantener registros exactos de los empleados. Los sistemas de administración de recursos humanos llevan esos registros, que incluyen fotografías de los empleados y otros datos que deben usar otros sistemas, como una nomina. Los sistemas de evaluación proporcionan listas de verificación esenciales que los administradores pueden utilizar para evaluar a sus subordinados. Estos sistemas también ofrecen una utilería de calificación para cuantificar las fortalezas y debilidades de los trabajadores.

1.6.2 Sistemas de Información de Diferentes Sectores Empresariales

El análisis anterior clasifica los sistemas de información según la manera en que se usan; en esta sección se verán los sectores de negocios en que se aplican. Un sector empresarial está definido por el tipo general de actividad que desarrolla o por el tipo de organizaciones en que se da la actividad. De igual manera, los sistemas suelen adaptarse a necesidades específicas del sector en que se emplean.

Producción

Los sistemas de información se usan en toda la operación de producción, desde el control de inventario hasta el pago a proveedores; ayudan a asignar recursos, como personal, materias primas y tiempo, para optimizar la productividad. Los sistemas de control de inventarios ayudan a planear las cantidades óptimas de materias primas que se deben reponer, de modo que la empresa no pague demasiado por materiales que no se usaran durante mucho tiempo; mientras asegura que los materiales estén disponibles cuando se requieran. Las

operaciones de producción utilizan sistemas de información para procesar los pedidos de los clientes, realizar el control de calidad y preparar documentación de embarque.

En un ambiente global, fuertemente competitivo, mantener bajos los costos puede ser la diferencia entre el éxito y el fracaso de una organización productiva. Los sistemas de información para planeación de recursos juegan un papel vital en determinar cuál recurso usar, en qué lugar y en qué momento.

Si una maquina permanece sin trabajo durante algunos minutos, la empresa incurre en costos que no contribuyen a obtener ganancias. El problema empeora si la maquina es sólo una estación en la línea de producción que desarrolla una secuencia de operaciones; toda una línea de producción puede quedarse varada. Los sistemas de información ayudan a reducir al máximo estos contratiempos.

Ahora los almacenes son manejados por computadora. Los sistemas de información especializados avisan automáticamente cuando debe sustituirse un artículo. También lo hacen cuando un artículo se encuentra en cantidad económica de pedido (CEP), que es la cantidad suficiente de artículos antes de que ya no queden reservas, mientras que se reduce al máximo el valor de los productos del almacén y el costo de almacenamiento, entonces la mercancía se ordena de inmediato para evitar que se detenga la producción.

Los sistemas más modernos están conectados directamente con los sistemas del proveedor, de modo que este observa el inventario y envía los artículos que pronto harán falta, sin que los empleados del almacén tengan que esforzarse demasiado.

Los SI se utilizaban para administrar por separado cada área de planeación de una operación: planeación de inventario, compras, pagos, facturación, etc. En años recientes, el razonamiento de los sistemas ha llevado a muchos fabricantes y otras organizaciones a adoptar sistemas que satisfagan todas esas funciones; se les suele llamar Sistemas de Aplicaciones Empresariales. Un SI, formado por varios subsistemas, utiliza entradas como presupuestos de ventas y plazos de pago para ayudar a planear el manejo de recursos, hacer pagos y cobrar deudas.

Servicio

Los sistemas de información juegan un papel central en el sector de servicios porque a menudo son la espina dorsal de las organizaciones de servicios. Imagine una aerolínea sin un sistema de información; sería incapaz de reservar asientos para los pasajeros y fechar vuelos. Piense por un momento en lo que harían los bancos sin SI. Podríamos argumentar que lo único que "fabrican" los bancos es

información. Las entradas de los sistemas de información de los bancos incluyen el tipo de cuenta además de las fechas y el monto de los depósitos y retiros; su salida consta de estados, que muestran los intereses pagados o adeudados, saldos y otra información. La gente incluso hace retiros en efectivo, en la mayor parte de los casos, a través de sistemas de información llamados cajeros automáticos. Los bancos cada vez dan menos información en papel a los clientes. Ahora, los clientes reciben información por teléfono sobre sus cuentas directamente de sistemas de información por teléfono o mediante computadoras caseras. En realidad el fenómeno de "el banco en su casa" debe su crecimiento a los sistemas de información. En general, el sector de servicio ha aprovechado la TI para aumentar la productividad más que cualquier otro sector.

Venta al Menudeo

Como los SI permiten que una gran cadena de venta al menudeo se dirija como una sola tienda, estas cadenas han obtenido economías de escala significativas utilizando SI; es decir, pueden bajar costos gracias a su tamaño. Las cadenas de venta al menudeo invirtieron millones de dólares en sistemas de información en la década pasada. El propósito principal de estos sistemas es reducir al máximo los niveles bajos y altos de inventario en cualquiera de las tiendas de la cadena; estos se logran principalmente al vincular los sistemas de las tiendas para combinar los datos informativos de toda la cadena, y el conectar los sistemas de la cadena con los de los proveedores. Por ejemplo, las tiendas Wal-Mart y Kmart se vinculan con redes de comunicación vía satélite. La administración puede determinar rápidamente cuáles artículos se desplazan con mayor rapidez y cuáles no. Hay información disponible tienda por tienda, sobre una base regional o nacional.

Gracias a esta información las tiendas eliminan del inventario los artículos de lento desplazamiento y surten mayores cantidades de artículos populares para mejorar las ganancias. Debido a la disponibilidad de la información en línea el administrador de una tienda que se ha quedado sin un artículo determinado puede reponerse rápidamente usando las existencias de una tienda cercana. La creación de interfaces con las cajas registradoras y las bases de datos le permite a los administradores recibir información para la toma de decisiones diaria, más que semanal o mensual. Los expertos en venta al menudeo argumentan que Wal-Mart se convirtió en la cadena de tiendas más grande y eficiente del mundo porque utilizó SI innovadores.

Nuevos Negocios

El impacto de los SI en los negocios va más allá de la simple automatización. Muchas empresas adoptaron SI para proporcionar nuevos productos y servicios

que solo se volvieron económicos con el desarrollo de la tecnología de la información. Empresas de historiales de crédito, como Experian, Equifax y TransUnion, usan SI para registrar información importante de crédito sobre millones de usuarios de tarjetas de crédito y prestatarios.

Venden información a los bancos y otras instituciones financieras. Algunas aerolíneas venden servicios de sistemas de reservación a agencias de viajes.

Las compañías de paquetería proporcionan servicios de rastreo a sus clientes para localizar paquetes. Los anteriores son tres servicios que no estarían disponibles sin SI, e Internet ha hecho crecer miles de negocios pequeños y caseros como Amazon.com y eBay, crecieron rápidamente hasta convertirse en grandes empresas. Esas compañías son ejemplos clásicos de comercio electrónico (hacer negocios mediante redes de computadoras).

Gobierno

En las cuatro décadas pasadas las organizaciones gubernamentales y comerciales han instalado SI basados en computadoras para automatizar procesos y reemplazar el trabajo humano. El trabajo del gobierno depende mucho de los SI para recaudar impuestos, pagar servicios de seguridad social y comprar bienes y servicios a varios departamentos.

En 1999, 272 millones de personas vivían en Estados Unidos. Alrededor de 100 millones pagaban impuestos y seguridad social, el gobierno federal llevaba cientos de millones de registros que contienen detalles sobre los contribuyentes y los benefactores de la seguridad social. A partir de 1989 los contribuyentes podían hacer sus declaraciones electrónicamente y se estima que 20% lo hizo en 1999. Esto permite al Servicio de Recaudación Interna acreditar las cuentas bancarias de los contribuyentes en una semana y no en un periodo de tres a seis semanas, y también le ahorra millones de dólares en llenado y manejo de papel.

Con los SI las autoridades fiscales utilizan complejos programas para cruzar los archivos de los contribuyentes con los de otras autoridades federales, estatales y locales. La seguridad nacional y las agencias de beneficencia llevan registro de los contribuyentes y transfiere fondos a destinatarios seleccionados.

Los departamentos de defensa también dependen de los SI para planear la adquisición de equipamiento y de actividades de entrenamiento. Organizaciones económicas, como el departamento de Comercio y Trabajo, y los bancos centrales, utilizan SI para planear de modo que puedan asesorar a quienes toman decisiones sobre política económica. Las autoridades de inmigración llevan registros de las

personas que cruzan las fronteras nacionales. Los SI facilitan todas estas actividades y muchas más.

Internet proporciona una gran oportunidad para que las agencias gubernamentales mejoren el servicio mientras ahorran costos laborales. Por ejemplo, el Servicio de Recaudación interna de Estados Unidos proporciona todos los formularios fiscales disponibles en la Web. Es más, permite que la gente haga sus declaraciones de impuestos por Internet. El sitio Web de la Oficina de Marcas y Patentes de Estados Unidos le permite a cualquier persona el acceso a una enorme base de datos que contiene registros de todas las patentes registradas en Estados Unidos (cerca de 6 millones). Esto ahorra la enorme cantidad de tiempo que se perdía en búsqueda de patentes para determinar si un dispositivo o un método ya se había patentado, o para utilizar la tecnología de patentes existentes.

1.7 ADMINISTRACIÓN DE DATOS Y ADMINISTRACIÓN DE BASES DE DATOS

Existe una persona identificable con esta responsabilidad central sobre los datos. Ese individuo es el administrador de datos (abreviado a veces DA, data administrador). La información es uno de los activos más valiosos de la empresa, es indispensable contar con una persona, el administrador de datos que conozca la información y las necesidades de la empresa en este aspecto, en un nivel gerencial superior. Así, la labor del administrador de datos es decidir en primer término cuáles datos deben almacenarse en la base de datos, y establecer políticas para mantener y manejar los datos una vez almacenados en la base de datos.

Un ejemplo de esto podría ser una política para determinar quién puede realizar cuáles operaciones sobre cuáles datos y en qué circunstancias, es decir, una política de seguridad de la información.

Es importante señalar que el administrador de datos es un gerente, no un técnico (aunque ciertamente sí necesita apreciar las posibilidades de los sistemas de bases de datos en un nivel técnico). El técnico responsable de poner en práctica las decisiones del administrador de datos es el administrador de bases de datos (casi siempre abreviado DBA, database administrador).

Así, el DBA, a diferencia del administrador de datos, es un profesional en procesamiento de datos. La tarea del DBA es crear la base de datos en sí y poner en vigor los controles técnicos necesarios para apoyar las políticas dictadas por el administrador de datos. El DBA se encarga también de garantizar el funcionamiento adecuado del sistema y de proporcionar otros servicios de índole

técnica relacionados. El DBA cuenta por lo regular con un grupo de programadores de sistemas y otros asistentes técnicos, de modo que en la práctica será un equipo de varias personas el que desempeñe la función de DBA, no una sola. Por sencillez, conviene pensar en el DBA como un solo individuo.

1.7 SISTEMA DE BASE DE DATOS

Un sistema de base de datos no es más que un sistema para archivar en computador. La base de datos en sí puede considerarse como una especie de archivero electrónico o es un lugar donde se almacena un conjunto de archivos de datos computarizados.

Operaciones que puede realizar el usuario del sistema.

- Agregar archivos nuevos (vacíos) a la base de datos.
- Insertar datos nuevos en archivos ya existentes.
- Obtener datos de archivos ya existentes
- Actualizar datos en archivos ya existentes
- Borrar datos en archivos ya existentes y,
- Eliminar archivos ya existentes (vacíos o no) de la base de datos

1.8.1 ¿Qué es un Sistema de Base de Datos?

Es básicamente un sistema para almacenar en computador; o sea, es un sistema computarizado cuyo propósito general es mantener información y hacer que esté disponible cuando se solicite.

1.8.2 Definición de Base de Datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas. Por este motivo es importante conocer la forma en que están estructuradas las bases de datos y su manejo.

1.8.3 Componentes Principales

- Datos: los datos son la Base de Datos propiamente dicha.
- Hardware: el hardware se refiere a los dispositivos de almacenamiento en donde reside la base de datos, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.
- Software: está constituido por un conjunto de programas que se conoce como Sistema Manejador de Base de Datos (DMBS: Data Base Management System). Este sistema maneja todas las solicitudes formuladas por los usuarios a la base de datos.
- Usuarios. existen tres clases de usuarios relacionados con una Base de Datos:
 - El programador de aplicaciones, quien crea programas de aplicación que utilizan la base de datos.
 - El usuario final, quien accesa la Base de Datos por medio de un lenguaje de consulta o de programas de aplicación.
 - El administrador de la Base de Datos (DBA: Data Base Administrator), quien se encarga del control general del Sistema de Base de Datos.

1.8.4 ¿Por qué Utilizar una Base de Datos?

- Es compacta: no hacen falta archivos de papel que pudieran ocupar mucho espacio.
- Es rápida: la manipulación de los datos con mucha mayor velocidad
- Menos laborioso: se eliminan en gran parte el tedio de mantener archivos a mano.
- Es actual: se dispone en cualquier momento de información precisa y al día.

1.8.5 Ventajas en el Uso de Bases de Datos

- Globalización de la información: permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
- Eliminación de información redundante: Duplicada
- Eliminación de información inconsistente: si el sistema esta desarrollado a través de archivos convencionales, dicha cancelación deberá operarse tanto en el archivo de facturas del Sistema de Control de Cobranza como en el archivo de facturas del Sistema de Comisiones.
- Permite compartir información: varios sistemas o usuarios pueden utilizar una misma entidad.

- Permite mantener la integridad en la información: solo se almacena la información correcta.
- Independencia de datos: la independencia de datos implica un divorcio entre programas y datos; es decir, se pueden hacer cambios a la información que contiene la base de datos o tener acceso a la base de datos de diferente manera, sin hacer cambios en las aplicaciones o en los programas.

1.9 EL SISTEMA MANEJADOR DE BASES DE DATOS (DBMS)

El DBMS es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. Se compone de un lenguaje de definición de datos (DDL: Data Definition Language), de un Lenguaje de manipulación de datos (DML: Data Manipulation Language) y de un lenguaje de consulta (SQL: Structured Query Language).

El lenguaje de definición de datos (DDL) es utilizado para describir todas las estructuras de información y los programas que se usan para construir, actualizar e introducir la información que contiene una base de datos. El DDL contiene un diccionario de datos que se utiliza para almacenar y crear las definiciones de los datos, incluyendo localización, forma en que se almacenan y algunas otras características. Este lenguaje de datos debe permitir describir los datos y las estructuras de los archivos del sistema, especificando la forma en que serán agrupados en registros o divididos en campos. Una vez que se ha elaborado la definición de la base de datos, el DBMS se encarga de construir y generar las estructuras de información de manera automática.

El lenguaje de manipulación de datos (DML) es utilizado para escribir programas que crean, actualizan y extraen información de las bases de datos. A pesar de que el DBMS proporciona gran ayuda al programador, en ocasiones es necesario escribir programas para extraer datos dando respuesta a requisiciones especiales. El lenguaje de consulta (SQL) es empleado por el usuario para extraer información de la base de datos. Este lenguaje permite al usuario hacer requisiciones de datos sin tener que escribir un programa, usando instrucciones como el Select, el Project y el Join.

La secuencia conceptual de operaciones que ocurren para acceder cierta información que contiene una base de datos es la siguiente:

- El usuario solicita cierta información contenida en la base de datos.
- El DBMS intercepta este requerimiento y lo interpreta.

- El DBMS realiza las operaciones necesarias para acceder y/o actualizar la información solicitada.

Una de las ventajas del DBMS es que puede ser invocado desde programas de aplicación que pertenecen a sistemas transaccionales escritos en algún lenguaje de alto nivel, para la creación o actualización de las bases de datos, o bien para efectos de consulta a través de lenguajes propios que tienen las bases de datos o lenguajes de cuarta generación.

1.9.1 El Administrador de la Base de Datos (DBA)

El DBA es la persona encargada de definir y controlar las bases de datos corporativas, que además proporciona asesoría a los usuarios y ejecutivos que la requieran. Sus funciones incluyen:

- Apoyar y asesorar durante el proceso de adquisición del DBMS, tanto del paquete corporativo como los paquetes que servirán de herramienta para usuarios finales que deseen crear sus propias bases de datos.
- Definir la información que contendrán las bases de datos corporativas.
- Mantener la relación y comunicación estrecha con los especialistas del DBMS, que suelen laborar con el proveedor que vendió el paquete que maneja las bases de datos.
- Diseñar las estructuras de almacenamiento y estrategias de acceso a las bases de datos.
- Atender y servir como punto de enlace entre los usuarios de la organización, asegurando que las necesidades de información de los diferentes usuarios se encuentren contenidas en las bases de datos corporativas.
- Definir estándares y procedimientos para respaldar y recuperar la información que contienen las bases de datos.
- Proporcionar asesoría técnica a analistas y programadores que se encuentran desarrollando aplicaciones que crean y/o accesan las bases de datos.

Definamos tres términos:

Tabla

Fila = registros

Columna = campos

- Un campo almacenado es la unidad más pequeña almacenada que recibe un nombre.

- Un registro almacenado es un conjunto de campos almacenados, relacionados entre sí que cuentan con su propio nombre
- Un archivo almacenado es un conjunto (con nombre) de todas las ocurrencias de un tipo de registro almacenado

Ejemplo: Tabla Empleado

CODIGO-EMPLEADO	NOMBRE	CÉDULA	CARGO	SUELDO
Xxxxx	xxxx	xxxxx	xxxxx	xxxxx
Aaaa	aaaa	aaaa	aaaa	aaaa

1.10 TIPOS DE MODELOS DE BASE DE DATOS

Son las diferentes alternativas de organización y estructuración de la información contenida en una base de datos. Existen fundamentalmente tres alternativas disponibles para diseñarlas: el modelo jerárquico, el modelo de red y el modelo relacional. La diferencia reside en la manera en que los registros son ligados entre ellos.

1.10.1 Modelo Jerárquico

El modelo de datos jerárquico se usó mucho como modelo de datos, debido, principalmente, al anuncio que hizo la compañía IBM en 1968 de su producto de bases de datos llamado IMS (Information Management Systems), basado en un esquema jerárquico para la representación de la información.

La forma de esquematizar la información se realiza a través de representaciones jerárquicas o relaciones de padre/hijo, de manera similar a la estructura de un árbol. Así, el modelo jerárquico puede representar dos tipos de relaciones entre los datos: relaciones de uno a uno y relaciones de uno a muchos.

En el primer tipo se dice que existe una relación de uno a uno si el padre de la estructura de información tiene un solo hijo, y viceversa, si el hijo tiene un solo padre. En el segundo tipo se dice que la relación es de uno a muchos si el padre tiene más de un hijo, aunque cada hijo tenga un solo padre.

Inconvenientes del Modelo Jerárquico

A pesar de que la representación jerárquica es muy fácil de entender y comunicar, en la actualidad este modelo de representación de la información está dejando de

emplearse por muchas organizaciones debido a que presenta algunos inconvenientes, como por ejemplo:

Todo hijo tiene necesariamente un padre, lo cual dificulta dar de alta a algún hijo cuyos datos del padre se ignoran.

La representación de información donde se requieran relaciones de muchos a muchos tiene a complicarse, de tal forma que si un hijo llega a tener dos o más padres, la información de este hijo debe almacenarse en varios lugares diferentes de la base de datos, lo cual depende de cada uno de sus padres. Este caso se puede ver más claramente en una relación maestro-alumno, donde un maestro tiene varios alumnos, pero un alumno también tiene varios maestros, uno para cada clase. En este caso, si la información estuviera representada en forma jerárquica donde el padre es el maestro y el alumno es el hijo, la información del alumno tendría que duplicarse para cada uno de los maestros.

La situación que se describe en el párrafo anterior puede generar información incoherente, de tal manera que si se desea cambiar los datos generales de un alumno, se deberán recorrer todos los maestros que tienen bajo su estructura al alumno en cuestión y hacer los cambios a sus datos generales.

Otra dificultad que presenta el modelo jerárquico de representación de datos es respecto a las bajas, en este caso, si se desea dar de baja a un padre, ello necesariamente implicará dar de baja a todos y cada uno de los hijos que dependen de este padre.

1.10.2 El Modelo de Red

Este modelo de datos permite la representación de muchos a muchos, de tal forma que cualquier registro dentro de la base de datos puede tener varias ocurrencias superiores a él. No hay que olvidarse que ésta era una de las principales limitantes del modelo jerárquico de representación de datos y que la información de cada uno de los alumnos era repetida para cada uno de los maestros. El modelo de red evita esta redundancia en la información.

A pesar de que las desventajas descritas en el modelo jerárquico desaparecen con el diseño de red, la mayor parte de la complejidad y problemas de este modelo surge debido a la dificultad de manejar las conexiones o ligas entre los registros y sus correspondientes registros conectores.

Sus principales desventajas residen en la dificultad de crear y dar mantenimiento a la base de datos.

1.10.3 El Modelo Relacional

El modelo relacional para la representación de información de una base de datos se está empleando con más frecuencia en la práctica, debido a las ventajas que ofrece sobre los dos modelos anteriores, entre ellas, el rápido entendimiento por parte de usuarios que no tienen conocimientos profundos sobre sistemas de bases de datos.

En este modelo toda la información se representa a través de arreglos bidimensionales o tablas. Por lo general, el usuario de las bases de datos relacionales tiene conocimientos de las tablas que están definidas y su interacción con la información contenida en la base de datos se reduce a operaciones lógicas que se efectúan con las diferentes tablas. Estas operaciones básicas son:

- Seleccionar renglones de alguna tabla (Select)
- Seleccionar columnas de alguna tabla (Project).
- Unir o juntar información de varias tablas (Join)

El uso del modelo relacional se ilustra un ejemplo simple, en el que existen tres tablas definidas para la relación maestro-alumno. En un caso hipotético se desea hacer una consulta a la base de datos relacional formada por estas tablas. La consulta consiste en desplegar el nombre de todos los maestros que imparten clases al alumno llamado ANOMI. Para lograr lo anterior, se tiene que ejecutar la siguiente secuencia de operaciones con las tablas.

SELECT Calification WHERE A# = A1 GIVING Aux 1

Al hacer esto, en la tabla Aux1 aparecerá la información.

A continuación se llevará a cabo la secuencia siguiente:

JOIN Aux1and Maestro OVER M# GIVING Aux2

Finalmente:

PROJECT Aux2 OVER MNOM GIVING Final

Al ejecutar esta instrucción, en la tabla final aparecerá la información que corresponde al nombre de los maestros que imparten clases al alumno A1.

Es importante mencionar que la mayoría de los paquetes que manejan bases de datos disponibles en el mercado poseen las instrucciones Select, Project y Join con diferentes nombres y modalidades.

1.10.4 El Modelo Orientado hacia Objetos

Estos modelos son utilizados con los lenguajes orientados hacia objetos, donde éstos, además de guardar los datos y los procedimientos que los manipulan, contienen las relaciones con otras entidades. De aquí surgen los DBMS orientados hacia objetos (en inglés OODBMS: Object Oriented Data Base Management Sytem). Los OODBMS proveen interfaces gráficas para administrar el DBMS. Las bases de datos con orientación hacia objetos son muy eficaces en sistemas que usan componentes de multimedia, como los sistemas geográficos, así como en aplicaciones de CAD (Computer Arded Design) y CAM (Computer Arded Manufacturing).

1.10.5 Algunas Bases de Datos

SQL, ORACLE, DBASE, IV, FOXPRO, FOXBASE, PARADOS, ACCES, APPROACH.

1.11 ¿QUÉ ES UNA BASE DE DATOS?

Conviene llamar "persistentes" a los datos de una base de datos (aunque quizá en realidad no persistan mucho tiempo). Esto tiene por objeto sugerir que la información de una base de datos difiere de otros tipos de datos, más efímeros, como son los datos de entrada y de salida, las proposiciones de control, las colas de trabajo, los bloques de control de programas, los resultados intermedios y, en términos más generales, cualquier información cuya naturaleza sea hasta cierto punto transitoria. En seguida se explican los términos "datos de entrada" y "datos de salida":

- "Datos de entrada" se refiere a la información que entra al sistema por primera vez (casi siempre desde el teclado de una terminal, desde un lector de tarjetas o desde un dispositivo similar). Esta información podría dar pie a una modificación de los datos persistentes (podría convertirse en parte de estos últimos), pero en principio no forma parte de la base de datos propiamente dicha.
- De manera similar, "datos de salida" se refiere a mensajes y resultados que emanan de los sistemas (casi siempre impresos o presentados en la pantalla de una terminal). Una vez más, esta información podría derivarse de los datos persistentes, pero no se le considera en si como parte de la base de datos.

Una base de datos está constituida por cierto conjunto de datos persistentes utilizado por los sistemas de aplicaciones de una empresa determinada.

El término "empresa" tal y como se emplea en esta definición no es más que un término genérico usado por comodidad, aplicable a cualquier organización comercial, científica, técnica o de otro tipo con un grado razonable de autosuficiencia. Una empresa podría ser una sola persona (con una pequeña base de datos privada) o una corporación o entidad similar de gran tamaño (con una enorme base de datos compartida), o cualquier cosa entre estos extremos. He aquí algunos ejemplos:

- Una compañía manufacturera
- Un banco
- Un hospital
- Una universidad
- Una dependencia del gobierno

Toda empresa debe por fuerza mantener una gran cantidad de datos referentes a su operación. Estos son los "datos persistentes" de los cuales se habló antes. Las empresas recién mencionadas con toda seguridad incluirían entre sus datos persistentes los que siguen:

- Datos de productos
- Datos de cuentas
- Datos de pacientes
- Datos de estudiantes
- Datos de planificación

1.11.1 ¿Por qué Utilizar una Base de Datos?

Las ventajas de un sistema de base de datos sobre los métodos tradicionales de mantener registros en papel serán quizá más evidentes en estos ejemplos. He aquí algunas de ellas:

- Es compacto, no hacen falta archivos de papeles que pudieran ocupar mucho espacio.
- Es rápido, la máquina puede obtener y modificar datos con mucha mayor velocidad que un ser humano. Así es posible satisfacer con rapidez consultas de casos particulares, del momento, sin necesidad de búsquedas visuales o manuales que requieren mucho tiempo.

- Es menos laborioso, se elimina gran parte del tedio de mantener archivos a mano. Las tareas mecánicas siempre serán mejor realizadas por las máquinas.
- Es actual, se dispone en cualquier momento de información precisa y al día.

1.11.2 Ventajas del Enfoque de Bases de Datos

Esta sección concluye con la identificación de algunas de las ventajas específicas que emanan del concepto de control centralizado.

Es Posible Disminuir la Redundancia

En los sistemas sin bases de datos cada aplicación tiene sus propios archivos privados. Esto puede provocar considerable redundancia en los datos almacenados, con el consecuente desperdicio de espacio de almacenamiento.

Es Posible Evitar la Inconsistencia (hasta cierto punto)

En realidad, esto es un corolario del punto anterior. Vamos a suponer que un cierto dato acerca del mundo real —por ejemplo, el hecho de que el empleado E3 trabaja en el departamento D8— está representado por dos entradas distintas en la base de datos almacenada. Supongamos también que el DBMS no está consciente de esta duplicación, es decir, la redundancia no está controlada.

Es Posible Compartir los Datos

El compartimiento (sharing), implica no sólo que las aplicaciones ya existentes pueden compartir la información de la base de datos, sino también que se pueden desarrollar aplicaciones nuevas para trabajar con los mismos datos almacenados. Dicho de otro modo, es posible satisfacer las necesidades de información de las aplicaciones nuevas sin tener que almacenar datos adicionales.

Es Posible hacer Cumplir las Normas

Al tener un control centralizado de la base de datos, el DBA (siguiendo las indicaciones del administrador de datos) puede garantizar la observancia de todas las normas aplicables para la representación de los datos. Estas normas pueden ser de la empresa, de la instalación, del departamento, de la industria, nacionales o internacionales, o de todos estos tipos. La normalización de formatos de los datos almacenados es deseable sobre todo como apoyo para el intercambio de información, o migración de datos entre sistemas. Del mismo modo, las normas para nombrar y documentar los datos son muy convenientes como ayuda para el compartimiento y comprensibilidad de la información.

Es Posible Aplicar Restricciones de Seguridad

Al tener jurisdicción completa sobre la base de datos, el DBA:

- Puede asegurar que el acceso a la base de datos sea solo a través de los canales apropiados y, por tanto,
- Puede definir las verificaciones de seguridad por realizar cuando se intente acceder a información delicada (una vez más, siguiendo los lineamientos apropiados del administrador de datos).

Es factible establecer diferentes verificaciones para cada tipo de acceso (consulta, modificación, eliminación, etc, a cada elemento de información de la base de datos. Pero adviértase que sin este tipo de verificaciones la seguridad de la información podría estar de hecho en mayor peligro que en un sistema de archivos tradicionales (dispersos); es decir, la naturaleza centralizada de un sistema de base de datos en cierto modo requiere la implantación de un buen sistema de seguridad.

Es Posible Mantener la Integridad

El problema de la integridad radica en asegurar que la información de la base de datos sea correcta. La inconsistencia entre dos entradas que supuestamente representan el mismo "hecho" es un ejemplo de falta de integridad; por supuesto, ese problema en particular solo puede presentarse si existe redundancia en la base de datos almacenada. Sin embargo, aún cuando no haya redundancia, es evidente que la base de datos puede contener información errónea.

Es Posible Equilibrar Requerimientos Opuestos

Al conocer los requerimientos generales de la empresa, en contraste con los requerimientos de cualquier usuario individual, el DBA (como siempre bajo la dirección del administrador de datos) puede estructurar el sistema con miras a proporcionar un servicio general "óptimo para la empresa". Por ejemplo, es posible escoger una forma de representación de los datos almacenados con la cual las aplicaciones más importantes puedan tener un acceso rápido, aunque el funcionamiento de algunas otras aplicaciones sufra menoscabo.

1.11.3 Ejemplo de Utilización de Base de Datos

En esencia, un sistema de bases de datos no es más que un sistema para archivar en computador. La base de datos en sí puede considerarse como una especie de archivador electrónico; dicho de otra manera, es un lugar donde se almacena un conjunto de archivos de datos computadorizados. Al usuario del sistema se le

ADMINISTRACION DE BASE DE DATOS

brindarán recursos para realizar diversas operaciones sobre estos archivos, incluidas entre otras las siguientes:

- Agregar archivos nuevos (vacíos) a la base de datos;
- Insertar datos nuevos en archivos ya existentes;
- Obtener datos de archivos ya existentes;
- Actualizar datos en archivos ya existentes;
- Borrar datos en archivos ya existentes y
- Eliminar archivos ya existentes (vacíos o no) de la base de datos.

Como ilustración, se muestra en la Tabla 1 una base de datos muy pequeña que contiene un solo archivo, llamado CAVA, el cual a su vez guarda información referente al contenido de una cava de vinos. La Tabla 2 presenta un ejemplo de una operación de consulta de esa base de datos, junto con los datos (o, dicho de manera más precisa, el resultado, aunque en el contexto de bases de datos es costumbre referirse a los resultados también como datos) obtenidos de esa consulta.

La Tabla 3 proporciona ejemplos, todos bastante sencillos, de operaciones de inserción, actualización y eliminación sobre la base de datos de la cava de vinos.

En primer lugar, y por razones obvias, a los archivos de computador del tipo de CAVA del ejemplo con frecuencia se les denominan tablas en vez de archivos.

CÓDIGO	VINO	PRODUCTOR	AÑO	BOTELLAS	LISTO	COMENTARIOS
2	Chardonnay	Buena vista	88	1	91	Día de grac.
3	Chardonnay	Louis Martín	89	5	90	
6	Chardonnay	Chappellet	87	4	91	
11	Jo. Riesling	Jekel	89	10	92	
12	Jo. Riesling	Buena vista	87	1	92	Cosecha tar. Muy seco Robt. Flores
16	Jo. Riesling	Sattui	87	1	89	
21	Fume Blanc	Ch. St. Jean	88	4	91	
22	Fume Blanc	Robt. Mondavi	87	2	90	
25	Borgoña Blanc	Mirassou	86	6	89	Cosecha Aniversario
30	Gewurztramine	Buena Vista	87	3	90	
43	r	Robt. Mondavi	82	12	92	
50	Cab.	Mirassou	82	3	90	
51	Sauvignon	Ch. St. Jean	86	2	92	
64	Pinot Noir	MIrassou	84	9	92	
72	Pinot Noir	Robt. Mondavi	85	2	90	
	Zinfandel					
	Gamay					

Tabla 1: La Base de Datos de Cava de Vinos (Archivo CAVA)

Consulta

```
SELECT VINO, CÓDIGO, PRODUCTOR
FROM CAVA
WHERE LISTO = 91;
```

Resultado (impreso o presentado en la pantalla):

VINO	CÓDIGO	PRODUCTOR
Chardonnay	2	Buena vista
Chardonnay	6	Chapellet
Fume Blanc	21	Ch. St. Jean

Tabla 2: Ejemplo de Operaciones en la Base de Datos de Cava de Vinos

- Inserción de Datos Nuevos
INSERT
INTO CAVA
VALUES (53, "Pinot Noir", "Saintsbury", 87, 1, 93, "para navidad");
- Actualización de Datos ya Existentes
UPDATE CAVA
SET BOTELLAS = 4
WHERE ANAQ = 3;
- Eliminación de Datos ya Existentes
DELETE
FROM CAVA
WHERE ANAQ = 2;

Tabla 3: Ejemplos de INSERT, UPDATE y DELETE

En segundo lugar, puede considerarse que las filas de una tabla como ésta representan los registros del archivo (llamados a veces en forma explícita registros lógicos, con objeto de distinguirlos de otros tipos de registros que se analizarán más adelante). De manera similar, puede considerarse que las columnas representan los campos de esos registros lógicos. En este módulo se utilizarán los términos "fila" y "registro" en forma indistinta, y lo mismo se hará con las palabras "columna" y "campo".

En tercer lugar, las operaciones de selección (SELECT), inserción (INSERT), actualización (UPDATE) y eliminación (DELETE) presentadas anteriormente son en realidad ejemplos de proposiciones de un lenguaje de bases de datos llamado SQL ("Structured Query Language", lenguaje de consulta estructurada). Este lenguaje (cuyo nombre suele pronunciarse como la palabra inglesa "sequel") es el utilizado en gran numero de sistemas comerciales de bases de datos, incluyendo en particular el producto DB2; de hecho, SQL es el lenguaje estándar oficial para interactuar con sistemas de bases de datos, tales como el DB2, que adoptan el "enfoque relacional"

Proceso de Comprensión y Análisis

- ¿Cuáles son las operaciones que efectúa básicamente un sistema computacional?
- ¿Cuál es la diferencia entre los términos Dato e Información?
- ¿Por qué es conveniente utilizar una base de datos?
- ¿Cuál es el proceso que sigue un sistema de información?
- ¿cuáles razones hacen que un sistema de información sea primordial para una empresa?
- ¿Cuál es el propósito de un sistema de base de datos?

Solución de Problemas

- Mostrar los resultados de las siguiente operaciones de consulta en SQL de la base de datos de cava de vinos. Ver tabla 1.

— SELECT VINO, PRODUCTOR
FROM CAVA
WHERE CÓDIGO = 72;

— SELECT VINO, PRODUCTOR
FROM CAVA
WHERE AÑO > 87;

— SELECT CODIGO, VINO, AÑO
FROM CAVA
WHERE LISTO < 90;

- Mostrar los efectos de las siguientes operaciones de actualización en SQL de la base de datos de cava de vino. Ver tabla 1.
 - INSERT
INTO CAVA
VALUES (80, `merlot`, `Clos du Bois`, 88, 12, 92, `navidad`);
 - UPDATE CAVA
SET BOTELLAS = 5
WHERE CÓDIGO = 50;
 - UPDATE CAVA
SET BOTELLAS = BOTELLAS + 2
WHERE CÓDIGO = 50;
- Escribir proposiciones en SQL para realizar las siguientes operaciones en la base de datos de cava de vinos.
 - Obtener el código, nombre del vino y el número de botellas de todos los vinos de Mirassou.
 - Obtener el código, el nombre de todos los vinos con más de 5 botellas en existencias.
 - Obtener el código de todos los vinos tintos.
 - Añadir tres botellas al código 30.
 - Eliminar todo el Chardonnay de las existencias.
 - Añadir un registro para una caja nueva (12 BOTELLAS de Mirassou Chardonnay: código 5, año 88, listo 92, con "1er premio" como comentario)

Síntesis Creativa y Argumentativa

- ¿Cuáles son las características de la información útil? Dar un ejemplo de cada una.
- Dar un ejemplo de cada uno de los elementos que conforman una base de datos?
- ¿Por qué es conveniente utilizar una base de datos?

Autoevaluación

- ¿Qué es Hardware?
- ¿Qué es Software?
- ¿Qué significan los términos Datos e Información?
- ¿Qué es información?
- ¿Qué es una base de datos y cuáles son sus funciones principales?
- ¿Cuáles son las funciones del administrador de base de datos?
- ¿Qué es un sistema de base de datos?
- ¿Qué elementos componen una base de datos?
- ¿Qué es un dato de entrada?
- ¿Qué es un dato de salida?

Repaso Significativo

- ¿Qué ventajas presenta el uso de un sistema de base de datos?
- ¿Qué desventajas presenta el uso de un sistema de base de datos?
- Ampliar la información referente a DDL, DML y SQL
- ¿Cómo se define los modelos de datos jerárquico, en red, relacional, orientado hacia objetos?

Bibliografía Sugerida

C. J, Date. Introducción a los Sistemas de Bases de Datos. Volumen I. Quinta Edición. E.U.A. Adisson – Wesley Iberoamericana. 1990.

COHEN, Daniel y ASÍN, Enrique. Sistemas de Información para los Negocios. Un Enfoque de Toma de Decisiones. Tercera Edición. México. Mc-Graw Hill. 2001

GARY W, Hansen. Y JAMES V. Hansen. Diseño y Administración de Bases de Datos. Segunda Edición. España. Prentice Hall Inc. 1996.

OZ, Effy. Administración de Sistemas de Información. Segunda Edición. México. Thomson Editores. 2002

VILLAREAL, Sonia. Introducción a la Computación. Guía Práctica para el Aprendizaje de Paquetes. México. Mc-Graw Hill. 2000

UNIDAD 2: Modelado de Datos

Descripción Temática

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 descrito así "dados los requerimientos de información y proceso de 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).

En la presente unidad se estudiará la temática relacionada con el modelado de datos con el propósito de brindar algunas ideas concernientes a los pasos necesarios para diseñar una base de datos sencilla.

Se partirá con la explicación de conceptos fundamentales, para así, a medida que se avanza en el estudio de la unidad, lograr la comprensión de temas más profundos. Los contenidos a estudiar son: Descripción del Ciclo de Recepción Venta / Efectivo y Diseño Conceptual de Base de Datos; esta temática se apoya en conceptos fundamentales y ejemplos resueltos que se tendrán en cuenta al desarrollar los ejercicios que se proponen en la unidad.

Horizontes

- Comprender qué es un modelo de datos.
- Comprender la importancia de los modelos de datos en el desarrollo y uso de sistemas de información.
- Conocer las herramientas y técnicas utilizadas para el diseño de las bases de datos.

- Identificar las entidades que representen claramente los modelos de datos y las relaciones entre ellas.
- Implementar las entidades de los modelos de datos en forma organizada que representen con precisión su estructura.
- Comprender las reglas gobernantes de interrelación para definir los modelos de datos en SGBD.
- Establecer test de validación en el diseño e implementación de los modelos de datos que evite la incorporación de información incorrecta.
- Comprender un modelo orientado a objetos.

Núcleos Temáticos y Problemáticos

- Descripción del Ciclo de Recepción Venta / Efectivo
- Diseño Conceptual de Base de Datos

Proceso de Información

La investigación moderna sobre modelos de datos se ha centrado en los aspectos lógicos de las bases de datos y sobre los conceptos, herramientas y técnicas para el diseño de las mismas. Aspectos relativos a la implementación de los modelos, tales como velocidad de ejecución, concurrencia, integridad física y arquitecturas no son factores relevantes en el estadio de análisis de modelos de datos. La investigación más temprana sobre modelos de datos sí estaba más centrada en los aspectos de representación física. Cuando hablamos de modelos de datos clásicos, nos estamos refiriendo a la segunda de las generaciones de modelos de datos, la cual distingue cuatro etapas:

- Modelos de datos primitivos (orientados al fichero).
- Modelos de datos clásicos.
- Modelos de datos semánticos.
- Modelos de datos de propósito específico (orientados a la aplicación).

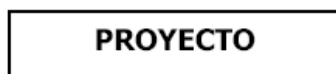
Los modelos de datos primitivos estaban absolutamente orientados al fichero: las entidades se representan en registros (divididos en campos, que representan sus propiedades), que se agrupan en ficheros. Las relaciones entre entidades son

únicamente aquellas que pueden ser representadas usando directorios, por ejemplo índices y listas invertidas. Un ejemplo de DBMS comercial de fichero, concretamente del tipo "lista invertida", es el CA-DATACOMB de Computer Associates International.

Los modelos de datos clásicos son tres: el jerárquico, el de red y el relacional.

2.1 DESCRIPCIÓN DEL CICLO DE RECEPCIÓN VENTA / EFECTIVO

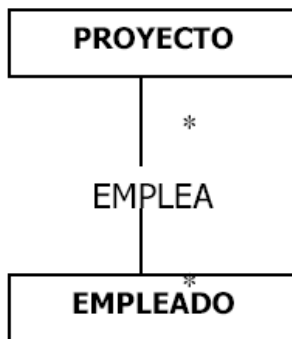
El primer paso para desarrollar una base de datos es crear un modelo de datos conceptual de sus acciones de negocios. Se necesitan las siguientes entidades para ilustrar el ciclo de recepción venta/efectivo: CLIENTE, PROYECTO, EMPLEADO, TIPO _ COMIDA y PAGO. En nuestro modelo de datos se representaran estas entidades con rectángulos. Por ejemplo, la entidad PROYECTO es:



La Figura 1 es un diagrama del modelo de datos que muestra cómo se relacionan entre sí las entidades.

¿Qué es lo que dice la Figura 1 sobre el ciclo de recepciones ventas /efectivo?

Veamos la interrelación entre la entidad PROYECTO y la entidad EMPLEADO.



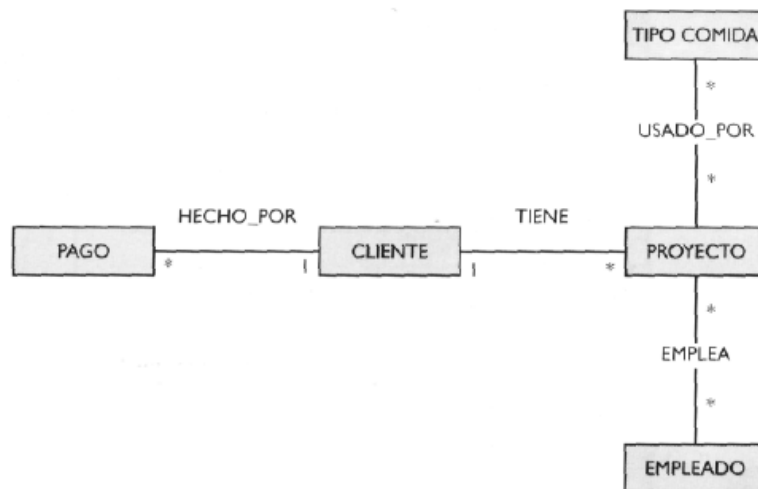


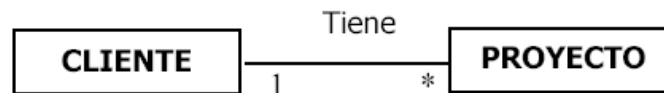
Figura 1: Modelo de Datos del Ciclo de Recepción Venta/Efectivo

La línea entre las entidades significa que éstas están relacionadas. El "*" y el "1" dicen cómo están relacionadas. La interrelación entre PROYECTO y EMPLEADO se lee:

- Cada PROYECTO puede emplear a uno o más (*) EMPLEADOS.
- Cada EMPLEADO puede trabajar en uno o más (*) PROYECTOS.

En otras palabras, cada empleado puede trabajar en varios proyectos y en cada proyecto pueden trabajar varios empleados.

Examinemos otra interrelación. Véase la interrelación entre la entidad CLIENTE y la entidad PROYECTO.



De nuevo la línea entre las dos entidades dice cómo están relacionadas. El "1" y el "*" dicen cómo están relacionadas. Esta interrelación se lee:

- Cada CLIENTE puede tener uno o mas (*) PROYECTOS.
- Cada PROYECTO puede tener solo un CLIENTE.

En otras palabras, cada proyecto de celebración puede ser solicitado por un solo cliente, pero cada cliente puede solicitar varios proyectos.

En la Figura 1 se pueden ver dos clases de interrelaciones. Estas son:

- Uno – a - muchos (1 a *), que es la que se muestra entre CLIENTE y PROYECTO, y entre CLIENTE y PAGO.
- Muchos – a - muchos (* a *) que es la que se muestra entre PROYECTO y EMPLEADO, y entre PROYECTO y TIPO COMIDA.

Existe también otro tipo de interrelación entre tablas que no se ha usado en este ejemplo. Esta es la interrelación uno – a - uno (1 a 1).

Ya se sabe cuales son las entidades involucradas y cómo estas están relacionadas entre sí. Estamos entonces preparados para comenzar a hacer corresponder el modelo con una implementación.

2.1.1 De las Entidades a las Tablas

Una tabla puede concebirse como una implementación de una entidad del modelo. En una tabla los datos se organizan en campos y registros (equivalentemente en columnas y filas). Por ejemplo, la tabla CLIENTE puede verse como algo parecido a:

Nombre	Dirección	Teléfono
Pedro Palacios	Clle 0 No. 5-29 La Flora	5786321
Ana María Flórez	Cra 1ª No. 7-25 El bosque	5762413
Valeria Acuña	Av. Sur No. 7-80 Jardines	5682146
Antonio Márquez	Clle 4ª No. 5ª-26 Arares	5798621

Nombre, Dirección y Teléfono son ejemplos de campos. El campo *Nombre* es una columna que contiene solamente nombres. Un registro (*record*) es una fila que contiene el nombre de un cliente, su dirección y su teléfono. Como se puede observar, la tabla CLIENTE contiene cuatro registros, uno para cada cliente, y tres campos, *Nombre, Dirección y Teléfono*. Los campos describen propiedades importantes de las entidades. Ahora se define esta tabla en un SGBD. Se necesita crear cinco tablas para los Servicios para Celebraciones. Estas son CLIENTE,

PROYECTO, EMPLEADO, TIPO COMIDA y PAGO. Se comenzará con la tabla CLIENTE.

Creación de las Tablas

SGBD: Servicio Celebraciones; se requieren crear 5 tablas:

- Cliente
- Proyecto
- Empleado
- Tipo comida
- Pago

Estructura de la Tabla Cliente

Nombre del campo	Tipo	Tamaño	clave
Nombre	A	25	*
Calle	A	25	
Ciudad	A	15	
Estado	A	2	
C.P	A	10	
Teléfono	A	14	

A*: Alfanumérico

Campo clave: Es un campo cuyo valor es único para cada registro, no hay dos registros con el mismo valor para ese campo.

Tabla Proyecto

NOMBRE DE CAMPO	TIPO	TAMAÑO	CLAVE
Proyect#	N		*
Fecha	D		
#Invitados	N		
Costo Est	\$		
Cant. Cargada	\$		

Tabla Empleado

NOMBRE DE CAMPO	TIPO	TAMAÑO	CLAVE
SS#	A	11	*
Nombre	A	25	

Tabla Tipo_Comida

NOMBRE DE CAMPO	TIPO	TAMAÑO	CLAVE
Nombre_Comida	A	25	*

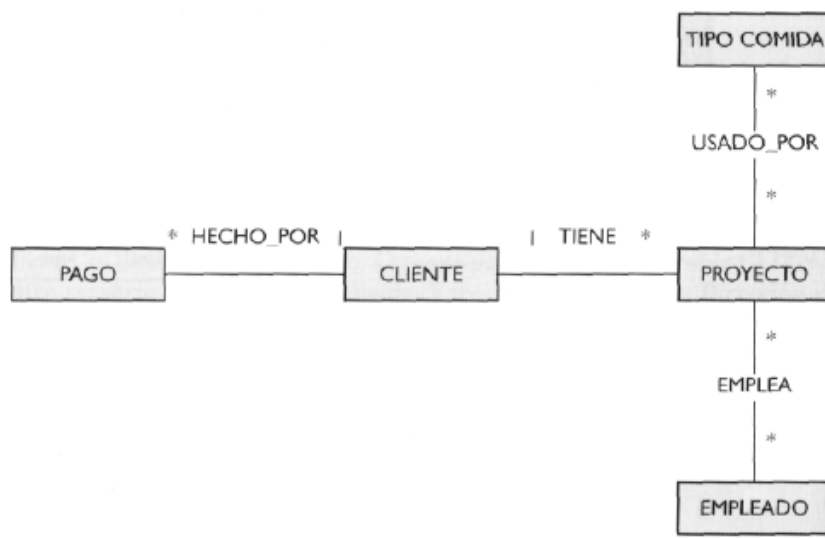
Tabla Pago

NOMBRE DE CAMPO	TIPO	TAMAÑO	CLAVE
Proyect#	N		*
Can. Recibida	\$		

Clases de Interrelación entre Tablas

- Uno – a – uno (1 a 1) no existe en el modelo de Servicio para Celebraciones
- Uno – a - muchos (1 a *) se muestra en las interrelaciones HECHO-PO TIENE.
- Muchos – a – muchos (* a *) se muestra en las interrelaciones USADO-PC EMPLEA.

Para estar seguros que estas interrelaciones están representadas correctamente la tabla, se deben seguir ciertas reglas, basadas en la interrelación en particular para cada tabla.



Hay tres reglas de tablas que gobiernan las interrelaciones entre ellas.

- Regla de Tabla 1: Si dos tablas tienen una interrelación uno – a - uno (1 a 1), entonces el campo clave de una de las tablas debe aparecer en la otra tabla. Ninguna de las interrelaciones de nuestro ejemplo se avienen a esta regla 1.
- Regla de Tabla 2: Si dos tablas tienen una interrelación uno – a – muchos (1 a *), entonces el campo clave de la tabla del (1) debe aparecer en la tabla del (*).

En el diagrama, dos interrelaciones admiten esta regla 2, las interrelaciones TIENE y HECHO-POR.

Para cumplir con la Regla de Tabla 2 para las tablas CLIENTE y PAGO, se pondrá *Nombre*, el campo clave de CLIENTE, en la tabla PAGO después del campo Proyec#. Puesto que *Nombre* es la clave de la tabla CLIENTE y se usa para conectarla a otras tablas, se le conoce como clave Foránea

Clave Foránea: es la misma ajena o externa, es un conjunto de atributos en una relación que constituye una clave en alguna otra (o posiblemente en la misma) relación; usada para indicar enlaces lógicos entre relaciones.

- Regla de Tabla 3: si dos tablas tienen una interrelación muchos – a - muchos (* a *), entonces debe crearse una nueva tabla que tenga los campos claves de las dos tablas.

En el diagrama, dos interrelaciones cumplen la Regla de Tabla 3, USADO-POR y EMPLEA. Para satisfacer la regla se deben Crear dos nuevas tablas. Primero se diseña con la que se enlazarán PROYECTO y TIPO_COMIDA. Esta tabla se llamara USADO_POR. Tendrá tres campos: Proyec# (primera clave), Nombre_Comida (segunda clave) y Cantidad. Note que entonces la tabla USADO_POR contiene los campos claves de PROYECTO y TIPO_COMIDA.

Tabla Usado Por

NOMBRE DE CAMPO	TIPO	TAMAÑO	CLAVE
Proyec#	N		*
Nombre_Comida	A	25	
Cantidad	N		

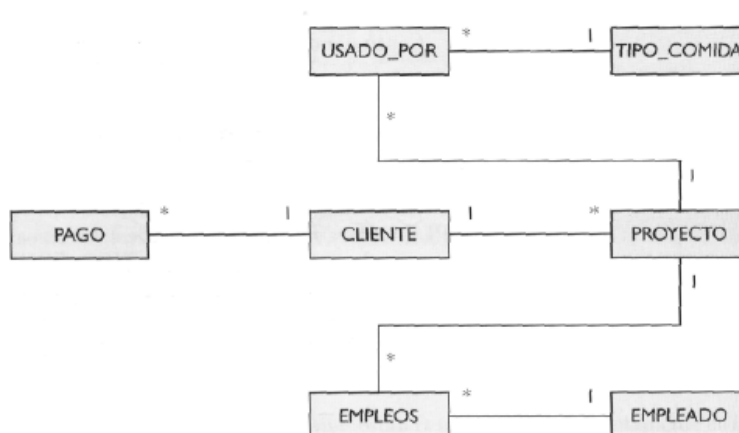
Ahora diseñaremos la tabla que enlaza PROYECTO y EMPLEADO. Esta tabla se llamará EMPLEA. Tendrá tres campos: *Proyec#* (primera clave), *SS#* (segunda

clave) y *Horas*. Usando los mismos pasos que en el caso anterior, se construye esta tabla EMPLEA.

Tabla Emplea

NOMBRE DE CAMPO	TIPO	TAMAÑO	CLAVE
Proyec#	N		*
SS#	A	11	*
Horas	N		

Ya están diseñadas todas las tablas y se refieren entre sí correctamente. Por "se refieren entre sí correctamente" se entiende que las tablas se relacionan de forma que representan con precisión la estructura de los Servicios para Celebraciones.



*Modelo de datos
versión modificada*

2.1.2 Establecer las Comprobaciones de Validación

La mejor manera de garantizar que los datos que se almacenen sean válidos es en primer lugar prevenir que no se puedan introducir datos incorrectos. Por esta razón es típico que los SGBD permitan imponer restricciones (test de validación) para cada campo en las tablas de una base de datos.

Los tipos de test de validación más comunes son:

- Campo obligatorio: se debe introducir un dato en este campo antes de poder guardar el registro.
- Mínimo: el menor valor que este campo puede aceptar.
- Máximo: el valor mayor que este campo puede aceptar.

- Por defecto: el valor que será puesto automáticamente en el campo por el manejador de base de datos, si el usuario omite el valor.
- Formato (*Picture*): una plantilla de los datos validos y el formato de este campo.

Con estos chequeos podemos estar seguros que el código postal, los números de teléfono y los números de seguridad social se introducen en la forma correcta; que los valores de tipo dinero (\$) están dentro de un rango razonable, y que los campos que sean claves siempre contendrán un dato. Las siguientes figuras ejemplifican como añadir los chequeos de validación a nuestras tablas.

Tabla Proyecto

Nombre de campo	Campo obligatorio	Mínimo
Proyec#	✓	
Costo Est.		
Cant. cargada		

Tabla Empleado

Nombre de campo	Campo obligatorio	Formato
SS#	✓	###_##_####

Tabla Tipo_Comida

Nombre de campo	Campo obligatorio
Nombre_Comida	✓

Tabla Pago

Nombre de campo	Campo obligatorio	Mínimo
Proyec#	✓	
Cant. recibida		0.00

Tabla Usado Por

Nombre de campo	Campo obligatorio	Mínimo	Formato
Proyec#	✓		
SS#	✓		###-##-####
Horas		0	

Tabla Emplea

Nombre de campo	Campo obligatorio	Mínimo	Máximo
Proyec#	✓		
Nombre_Comida	✓		
Cantidad		0	999

2.2 DISEÑO CONCEPTUAL DE BASE DE DATOS

2.2.1 Realidad y Modelos

¿Que es un Modelo?

Un modelo es una representación de la realidad que conserva solo los detalles relevantes. Por ejemplo, consideremos una transacción bancaria tal como un depósito en una cuenta corriente. Contaduría desea conservar ciertos detalles (numero de la cuenta, monto del deposito, tiempo, fecha, numero del cajero) e ignorar otros (las palabras que se van intercambiado durante la transacción, el numero de gente en el banco, número de personas que estaban esperando en la cola, la música que se estaba tocando en el audio local, las condiciones del clima fuera del banco, etc.). La realidad involucra un sin número de detalles, pero contaduría considerará a la mayoría de ellos irrelevantes para la transacción. De modo que un modelo, desde el punto de vista de Contaduría, conservará solo aquellos detalles que ésta considere relevante.

2.2.2 Modelos Conceptuales de Datos

Una metodología de modelado de datos que se estudiará y utilizará, podría llamarse modelo orientado a objetos porque considera la representación en la computadora de las entidades del mundo real como "objetos" que tienen su propia identidad y atributos y que participan en las relaciones, en lugar de la forma tradicional de considerarlas como registros en un sistema orientado a archivos.

Es generalmente, reconocido que las representaciones orientadas a objeto son más precisas para expresar la esencia lógica de las aplicaciones del mundo real que las representaciones basadas en registros. Por esta razón, esta metodología también puede llamarse semántica porque proporciona un medio poderoso para asociar los significados de las cosas en la realidad con las construcciones en el modelo. Desde principios de los setenta han sido propuestos varios modelos conceptuales y semánticos de datos. Se usará una metodología genérica de modelado, la cual tiene aspectos en común con las metodologías propuestas. Por simplicidad, a lo largo del libro se le llamará a este modelo, modelo conceptual de datos.

Modelo Semántico u Orientado a Objetos

Las bases de datos orientadas a objetos son el resultado de la convergencia de dos disciplinas de investigación: el modelado semántico de datos y los lenguajes orientados a objeto.

Estas disciplinas se desarrollaron de manera independiente, pero en los ochenta comenzaron a mezclarse con importantes implicaciones en el procesamiento de bases de datos.

El modelado semántico de datos fue originalmente desarrollado con el propósito de incrementar la efectividad y la precisión del diseño de bases de datos (Hull and King, 1987). Los métodos de modelado semántico fueron considerados apropiados para muchos problemas de usuario y podrían ser convertidos con facilidad a modelos con realizaciones basadas en registros tales como los modelos jerárquico, de redes y relacional.

El modelo de entidad-relación de Chen (E-R) ha sido el modelo semántico más popular y comúnmente se puede encontrar en libros sobre modelado conceptual de datos y diseño de bases de datos. Mientras que aquellos dedicados al modelado semántico de datos tienen que ver primariamente con las estructuras de los datos, los que desarrollan los lenguajes de programación orientados a objetos están más interesados en el comportamiento de los datos objetos. Esto es, están

buscando formas de manipular los datos que se centran en los datos y en las capacidades de manipulación del lenguaje (consulta, compute, actualización). La estructura de los datos pasa a un plano secundario.

Los elementos principales de un modelo conceptual de datos son los objetos y las relaciones. Los objetos se conciben a menudo como nombres, y las relaciones se ven como verbos. Aunque algunos modelos conceptuales de datos proporcionan algunas construcciones adicionales, los objetos y las relaciones son suficientemente poderosos para los problemas que se considerarán.

Objetos

Los objetos representan cosas que son importantes para los usuarios en el segmento de la realidad que queremos modelar. Ejemplos de objetos son las personas, los automóviles, los árboles, las maquinas lavaplatos, las casas, los martillos, los libros. Estos son objetos concretos. Objetos conceptuales son compañías, oficios, organizaciones, diseños de productos, transacciones de negocios y clasificaciones de los trabajos.

Proceso de Comprensión y Análisis

- Describir con sus propias palabras las diferencias entre una tabla y un campo.
- ¿Cómo se representa una tabla en el diagrama de un sistema de información?
- Con sus palabras definir qué es un Sistema de Información
- ¿Qué es un campo clave y qué objetivo persigue?

Solución de Problemas

Del modelo de datos que se presenta a continuación:

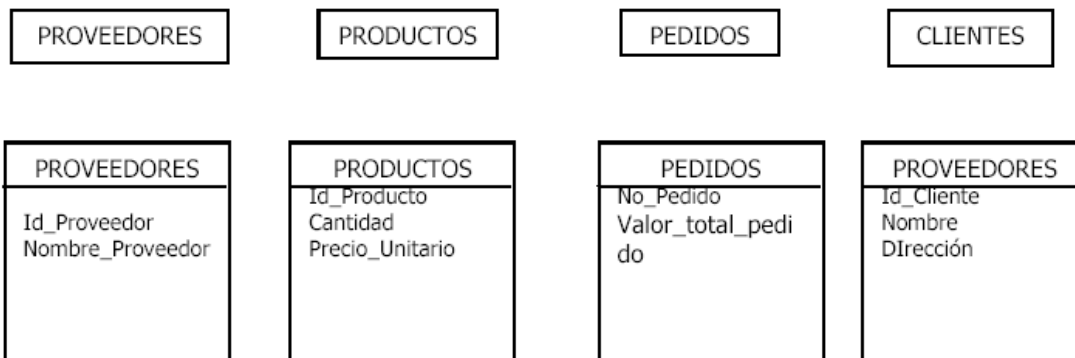
- Crear las tablas

Nombre de campo	Tipo	Tamaño	Clave
-----------------	------	--------	-------

- Aplicar las reglas de tablas que gobiernan las interrelaciones
- Mostrar el modelo de datos modificado
- Establecer las comprobaciones de validación que considere.

Ejemplo

Se tiene el siguiente modelo de datos



Síntesis Creativa y Argumentativa

- En un negocio de la región que proporcione bases de datos, desarrollar una pequeña aplicación de bases de datos. Diseñar y realizar dicha aplicación.
- Usando un ejemplo sencillo de un sistema de negocio, bosqueje un diagrama similar al ejemplo desarrollado en la unidad. Identificar las interrelaciones entre la tablas uno-a-uno, uno-a-muchos y muchos-a-muchos

Autoevaluación

- ¿Qué es un modelo?
- ¿Cómo define un objeto?
- ¿Qué es un campo y un registro?
- ¿Qué es una tabla?
- ¿Qué elementos componen una tabla?
- ¿Qué es un sistema de datos jerárquico, de red y relacional? ¿Qué diferencias se presentan entre ellos?
- ¿Qué es una entidad y qué características debe poseer para serlo?
- ¿Qué tipos de interrelación se pueden presentar entre las entidades? ¿Qué significa cada tipo?

- ¿Qué es una clave foránea? Dar un ejemplo.
- ¿Qué es un test de validación?. ¿Qué tipos hay?
- ¿Qué es un modelo orientado a objetos?

Repaso Significativo

- ¿Qué etapas distinguen la segunda generación de modelo de datos?
- Explicar por medio de ejemplos las reglas a tener en cuenta en la interrelación de tablas.

Bibliografía Sugerida

C. J, Date. Introducción a los Sistemas de Bases de Datos. Volumen I. Quinta Edición. E.U.A. Adisson – Wesley Iberoamericana. 1990.

GARY W, Hansen. Y JAMES V. Hansen. Diseño y Administración de Bases de Datos. Segunda Edición. España. Prentice Hall Inc. 1996.

UNIDAD 3: Sistemas Relacionales

Descripción Temática

Esta parte del modulo se centrará en el estudio del modelo de datos relacional y su uso como un modelo de diseño de implementación de bases de datos, además se define la construcción del modelo teniendo en cuenta el proceso de normalización para mostrarlo como un modelo relacional equivalente.

Horizontes

- Explicar los conceptos fundamentales del modelo relacional incluyendo relaciones, atributos, dominios, claves, claves foráneas, integridad de la entidad e integridad referencial.
- Demostrar como las relaciones pueden ser normalizadas.

Núcleos Temáticos y Problemáticos

- Conceptos Fundamentales
- Proceso de Normalización
- Integridad Relacional

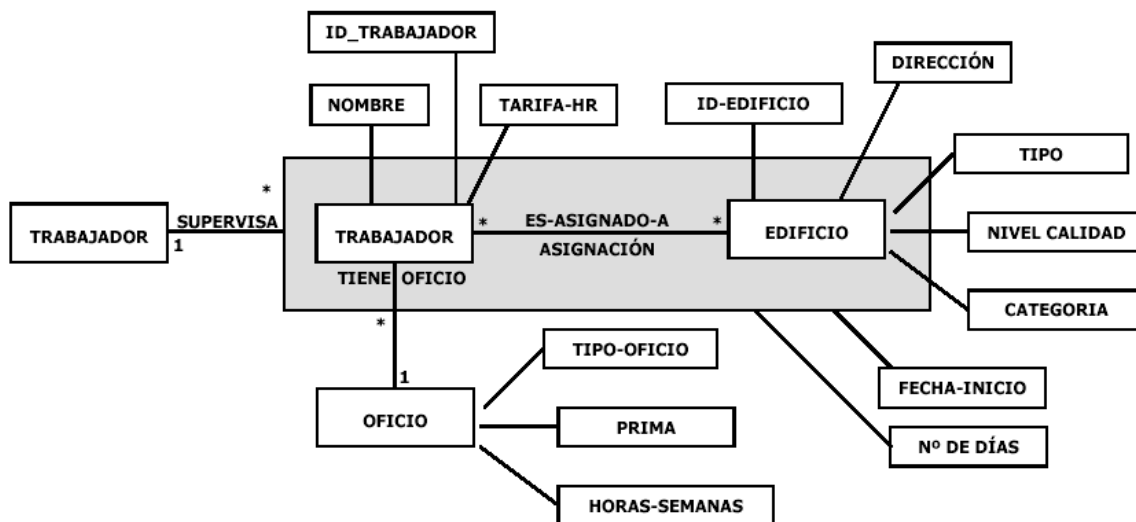
Proceso de Información

3.1 CONCEPTOS FUNDAMENTALES

Relaciones

El modelo de datos relacional organiza y representa los datos en forma de tablas o relaciones. Relación es un término que viene de la matemática y representa una simple tabla de dos dimensiones, consiste en filas y columnas de datos.

La figura siguiente muestra un modelo de datos conceptual que proporciona fundamentos para la base de datos relacional de una compañía.



Modelo de Datos Conceptual (Orientado a Objetos)

Objeto Asignación

Es una agregación, *es asignada a* entre trabajador y edificio. Cada una de estas organizaciones tienen 2 atributos.

- Fecha Inicio: fecha en que el trabajador es asignado a comenzar a trabajar en el edificio.
- Número de Días: Nº de días que el trabajador requiere para completar el trabajo en el edificio.

Interrelación Recursiva

Una interrelación que relaciona un objeto consigo mismo: Ejemplo: La relación supervisa.



Usando un proceso que se discutirá posteriormente, este modelo de datos puede ser convertido en un modelo de datos relacional. La figura muestra una relación con un ejemplo de los valores de los datos, la cual representa el conjunto de objetos TRABAJADOR, sus atributos y dos de sus relaciones. Cada columna en la relación, es un *atributo* de la relación. El nombre de la columna se llama "*nombre del Atributo*"

Este modelo de datos es orientado a objetos, puede ser convertido en un modelo de datos relacional.

- Muestra de relación con un ejemplo de los valores los datos, la cual representa el conjunto de objetos trabador.

TRABAJADOR ID_TRABAJADOR	NOMBRE	TARIFA_HR	TIPO DE OFICIO	ID: SUPV
1235	Francisco	2000	Electricista	1311
1412	Guillermo	1800	Fontanero	1520
2920	Gloria	2200	Carpintero	
3231	Marisol	1500	Albañil	
1520	Ricardo	1800	Fontanero	
1311	Claudia	2000	Electricista	
3001	Bernardo	1500	Albañil	3231

Atributos

FILAS
O
TUPLAS

Tabla Trabajador: Una parte de la relación trabajador.

- Muestra objeto trabajador, muestra sus atributos y dos de sus relaciones.

Atributo de la Relación

Una columna en una relación.

ID_trabajador Nombre Tarifa_AR Tipo de oficio ID_Supv	}	Atributo de la relación
---	---	-------------------------



Grado de Relación

Es el número de atributos de una relación. Ej. El grado de trabajador en cinco, dos atributos en una relación no pueden tener el mismo nombre.

Tupla

Una fila en una relación. No hay orden preestablecido de las filas o tuplas de una relación y 2 tuplas no tienen idéntico conjunto de valores.

Anotación para representar relaciones.

Trabajador (ID_Trabajador, Nombre, Tarifa_Ar, Tipo_de_Oficio, ID_supv)

Dominio de Atributo

Conjunto de valores que puede tomar un atributo. Dos dominios son idénticos sólo si tienen el mismo significado así:

- Nombre y tipo_de_oficio, poseen diferentes dominios aunque ambos dominios consisten en cadenas de caracteres.
- No es necesario que dos atributos con el mismo dominio tengan el mismo nombre. Ej: ID_supv, tiene el mismo dominio ID_Trabajador en ambos casos, el dominio consiste en los números de identificación del trabajador.

Valor Nulo

No es un espacio en blanco o cero, es simplemente un valor desconocido e inaplicable que puede ser reemplazado más tarde.

Clave

El conjunto mínimo de atributos que identifica unívocamente cada fila en una relación.

- Superclave: un conjunto de atributos que identifica unívocamente cada fila en una relación.
- Determinante Funcionalmente: determina unívocamente un valor o cada valor del atributo en una tupla.

- Clave Compuesta: una clave compuesta de más de un atributo.
- Clave Candidata: es una relación dada, puede que más de un conjunto de atributos puedan ser elegidos como clave. Ej: Es posible que el Nombre sea una clave candidata de la relación trabajador. Esto se asume que el nombre siempre fuera único.
- Clave Primaria: la clave candidata elegida como clave de la relación generalmente se usa el término clave para nombrar la clave primaria.

Claves Externas (Ajenas, Foreign) Foráneas

Un conjunto de atributos en una relación que constituyen una clave en alguna otra (o posiblemente la misma) relación; usada para indicar enlaces lógicos entre relaciones.

Se usan para vincular datos en una relación con datos en otra relación. Ejemplos: Tipo_de_oficio enlaza la relación trabajador con la relación oficio. *ID_Supv es una clave foránea en la relación trabajador que referencia la clave de su propia relación. Es ejemplo de clave externa recursiva.

- Clave Externa Recursiva: una clave que referencia su propia relación.
- Esquema de Base de Datos Relacional: un listado que muestra los nombres de las relaciones, los nombres de los atributos y claves foráneas.

Trabajador (ID_Trabajador, Nombre, Tarifa_HR, Tipo de oficio, ID_Supv)

Claves foráneas: Tipo_de_Oficio: Referencia a oficio

ID_Supv: Referencia a trabajador

Asignación: (ID_Trabajador, ID_edificio, Fecha_inicio, Num_días)

Claves foráneas: ID_Trabajador: Referencia a trabajador

ID_Edificio: Referencia a edificio.

Edificio(ID_Edificio, Dir_Edificio, Tipo, Nivel_Calidad, Calidad, categoría)

Oficio (Tipo_de_oficio, prima. Horas_por sem)

- Muestra de las relaciones en la base de datos de la compañía de construcción premier

Tabla Trabajador

ID_trabajador	Nombre	Tarifa_Hr	Tipo_oficio	ID_supervisor
1235	Francisco	2000	Electricista	1311
1412	Guillermo	1800	Fontanero	1520
2920	Gloria	2200	Carpintero	
3231	Marisol	1500	Albañil	
1520	Ricardo	1800	Fontanero	
1311	Claudia	2000	Electricista	
3001	Bernardo	1500	Albañil	3231

Tabla Asignación

ID_trabajador	ID_Edificio	Fecha_Inicio	Nº_Dias
1235	312	10/10	5
1412	312	01/10	10
1235	515	17/10	22
1412	460	08/12	18
1412	435	15/10	15
1412	515	11/05	8
1311	435	08/10	12

Tabla Edificio

ID_ Edificio	DIR_edificio	Tipo	Nivel_calidad	Categoría
312	Bloque A	Oficina	2	2
435	Bloque E	Comercio	1	1
515	Bloque C	Residencia	3	1
210	Bloque F	Oficina	3	1
111	Bloque H	Oficina	4	1
460	Bloque I	almacén	3	3

Tabla Oficio

Tipo-Oficio	Prima	Horas_sem
Fontanero	3.00	35
Electricista	3.50	37
Carpintero	2.33	40
Albañil	5.00	35

3.2 PROCESO DE NORMALIZACIÓN

Considere la siguiente tabla que combina parcialmente los datos de trabajador y de asignación.

Normalización

El proceso de conversión de una relación en una forma estándar.

Redundancia en los Datos

Repetición de datos en una base de datos. Con un pequeño análisis, se puede ver que la relación no está bien diseñada.

Tabla Trabajador

ID Trabajador	Nombre	Tipo de oficio	ID Supv	ID Edificio
1235	Francisco	Electricista	1311	312
1235	Francisco	Electricista	1311	525
1412	Guillermo	Fontanero		312
1412	Guillermo	Fontanero		460
1412	Guillermo	Fontanero		435
1412	Guillermo	Fontanero		515
1311	Claudia	Fontanero		435

Las 4 tuplas del trabajador 1412 repiten el mismo nombre y la información del tipo de oficio. Esta redundancia de los datos no solo ocupa espacio sino que conduce a perder integridad de los datos.

Integridad de los Datos

Consistencia de los datos en una base de datos. El problema surge porque un individuo puede estar trabajando en más de un edificio.

Anomalías de Actualización

Inconsistencia de los datos como resultado de datos redundantes y actualizaciones parciales. Ejemplo: Supongamos que el tipo de Oficio de Guillermo está erróneo y sólo la primera tupla está correcta, entonces se tendría una inconsistencia de tuplas.

Anomalías de Borrado

Suponga que Guillermo ha estado enfermo durante tres meses y todos los edificios asignados se terminaron.

Se decidió borrar todas las filas de la relación que contienen información de edificios terminados entonces ID_Trabajador, Nombre, Tipo de oficio, se perdería.

Pérdida no intencionada de datos debido a que se han borrado otros datos.

Anomalías de Inserción

Imposibilidad de adicionar datos en la base de datos a la ausencia de otros datos.
Ejemplo: Puede tenerse contratado un nuevo empleado llamado Adolfo que no se le haya asignado edificio alguno.

Estas anomalías son inconvenientes ¿Cómo se minimizan tales problemas?

Dividiendo la relación TRABAJADOR en dos relaciones TRABAJADOR y ASIGNACIÓN, como aparece en la tabla anterior. Esto es una solución intuitiva.

Ahora veremos un método más formal llamado descomposición para lograr el mismo resultado.

Descomposición

Es el proceso de dividir relaciones en múltiples relaciones para eliminar las anomalías y mantener la integridad de los datos. Para hacer esto se usan las formas Normales o reglas para relaciones estructuradas.

3.2.1 Primera Forma Normal (1FN)

Todos los valores de los atributos deben ser atómicos. Normalmente todos los esquemas que se encontraran estarán en 1FN.

- Valor Atómico: un valor que no es un conjunto de valores o un grupo repetitivo.
Ejemplo: de una tabla que no está en 1FN

Tabla Trabajador

ID_Trabajador	Nombre	Tipo_oficio	ID_Supv	ID_Edificio
1235	Francisco	Electricista	1311	{312,515
1235	Francisco			{312, 460
	Guillermo	Fontanero		435, 515
1311	Claudia	Electricista		435

Dependencias Funcionales

El valor de un atributo en una tupla determina el valor de otro atributo en la tupla.
Ej: En la Figura: en cada tupla, ID_Trabajador determina el Nombre y Tipo de Oficio.

FD: ID_Trabajador \rightarrow Nombre

FD: ID_Trabajador \rightarrow Tipo_oficio

Como A y B son atributos en la relación **R** entonces, FD: $A \rightarrow B$ significa que si cualesquiera dos tuplas en **R** tienen el mismo valor para su atributo A, deben tener el mismo valor para el atributo B. Esta definición se aplica también A y B son conjunto de columnas en lugar de columnas simples. La notación " \rightarrow " se lee "determina funcionalmente".

3.2.2 Segunda Forma Normal (2FN)

La 2FN y 3FN se ocupan de la relación entre los atributos claves y no claves. Los atributos no claves no pueden ser funcionalmente dependientes de una parte de la clave.

Una relación está en segunda forma normal (2FN) si no está en primera forma normal y el atributo no clave no es funcionalmente dependiente de una parte clave.

ASIGNACIÓN (ID_Trabajador, ID_Edificio, Fecha_Inicio, Nombre)

Tabla Trabajador

ID_Trabajador	ID_Edificio	Fecha_inicio	Nombre
1235	312	10/10	M. Faraday
1412	525	01/10	C. Nemo
1235	312	17/10	M. Faraday
1412	460	08/12	C. Nemo
1412	435	15/10	C. Nemo

Nombre, está determinado por ID_Trabajador y también es funcionalmente dependiente de una parte de la clave.

Conociendo ID_Trabajador, es suficiente para identificar el Nombre.

Cuando no está en 2FN, puede conducir a los problemas siguientes:

- El nombre del trabajador se repite en cada fila que se refiere a una asignación para ese trabajador.

- Si el nombre del trabajador cambia, cada fila que se refiera a una asignación de ese trabajador debe actualizarse. Esto, como se recordará, es una *anomalía de actualización*.
- Debido a esta redundancia, los datos podrían convertirse en inconsistentes, con distintas filas mostrando diferentes nombres para el mismo trabajador.
- Si al mismo tiempo no hay asignaciones para el trabajador, puede no haber filas en las cuales guardar su nombre.

Para resolver estos problemas, la relación puede descomponerse en los dos esquemas relacionales siguientes, los cuales están en 2FN:

ASIGNACIÓN: (ID_Trabajador, ID_Edificio, Fecha_Inicio)
Clave Externa: (ID_Trabajador referencia a trabajador.

TRABAJADOR: (ID_Trabajador, Nombre)

Por lo tanto 2FN reduce la redundancia e inconsistencia.

ID_Trabajador	ID_EDIFICIO	Fecha_Inicio
1235	312	10/10
1412	312	01/10
1235	515	17/10
1412	460	08/12
1412	435	15/10

3.2.3 Tercera Forma Normal (3FN)

Una relación está en 3FN si está también en la segunda forma normal y todos los campos que no sean la clave deben ser mutuamente independientes.

Considera la relación trabajador del ejemplo anterior:

FD: ID_Trabajador → Tipo_oficio

FD: ID_Trabajador → Prima

Son dependencias funcionales para esta relación debido a que el ID_Trabajador es una clave. Sin embargo,

FD: ID_Tipo Oficio → Prima

Es una dependencia funcional. Es claro que el criterio de 3FN se satisface para las dos primeras dependencias funcionales, pero el Tipo_Oficio no es una clave, por lo que el criterio de 3FN falla. Por lo tanto, la tabla trabajador no está en 3FN; sin embargo, esta tabla está en 2FN (eso se debe a que su clave consiste en solo un atributo), así, es posible para una relación estar en 2FN sin estar en 3FN.

Cuando no está en 3FN, puede conducir a los problemas siguientes:

- La prima para cada tipo de oficio se repite en las filas de cada empleado que tiene el mismo tipo de oficio. Esta redundancia de datos gasta espacio de memoria.
- Si la prima para el tipo de oficio cambia, tales filas deben ser actualizadas. Si una fila es borrada se puede perder el dato de la prima para un tipo de oficio. Por tanto la relación está sujeta a anomalías de actualización y borrado.
- Si hay actualmente empleados no permanentes para un tipo de oficio dado, puede no haber filas en las cuales guardar la prima del tipo de oficio. Esta es una anomalía de inserción.

Ahora se mostrará como aplicar el proceso de descomposición a relaciones que no están en 3FN.

Tabla Trabajador

ID_Trabajador	Tipo_Oficio	Prima
1235	Electricista	3.50
1412	Fontanero	3.00
1311	Electricista	3.50

Se comienza con el esquema relacional de la tabla trabajador. Forme una relación R1 moviendo de la tabla trabajador los atributos de la parte derecha. En el ejemplo, eso es PRIMA. Forme una nueva relación compuesta de los atributos de la parte derecha y de la parte izquierda estos son PRIMA y TIPO DE OFICIO, entonces:

R1 (ID_Trabajador, Tipo_Oficio)

Clave externa: Tipo_Oficio referencia a R2

Y

R2 (Tipo_Oficio, Prima)

Es decir, se deben crear dos tablas donde la clave principal de R2 se conecta con la clave externa de R1.

3.2.4 Cuarta Forma Normal (4FN)

Una relación está en cuarta forma normal (4FN) si está en 3FN y no tiene atributos multievaluados, debido a que el problema de las dependencias multievaluadas surge de los atributos multievaluados, se puede encontrar una solución poniendo todos los atributos multievaluados en relaciones formadas por ellos mismos, junto con la clave a la cual se aplican los valores de los atributos.

3.2.5 Quinta Forma Normal (5FN)

Una forma normal que elimina las dependencias de reunión (Join; también unión)

3.3 INTEGRIDAD RELACIONAL

Entidades y relaciones son una parte del modelado de datos. También se deben establecer reglas para que los datos sean correctos o plausibles. La probabilidad de los datos correctos es baja.

Ejemplo: se puede saber el número de empleados y productos, pero no, si los datos de ellos fueron bien introducidos.

El sistema puede sugerir cosas pero no corregir. Normalmente es caro notificar errores. Ningún sistema o diseñador de base de datos puede garantizar que los datos sean fidedignos, sólo que podría serlo a través de restricciones de integridad, algunos las llaman "reglas de negocio" (pero esto es más amplio no sólo se aplica a datos).

Ejemplo: seguridad es del administrador del sistema (negocio), no de datos.

3.3.1 Tipo de Restricciones de Integridad

- Dominios, Transiciones, Entidades: mantenimiento de integridad a las tablas individuales.
- Referencial: se mantienen las relaciones entre las tablas.
- De base de datos: gobiernan la base de datos como un todo.
- De transacciones: controlan la forma como se manipulan los datos dentro de una base de datos o entre múltiples base de datos.

Restricción de Integridad de Dominios

Dominio: conjunto de valores válidos de un campo.

- Restricción de dominios: regla que define esos valores válidos. Se puede definir más de una regla o restricción.

Ejemplo: contemplar valores desconocidos o inexistentes para el dominio.

Desconocido es diferente a *inexistente*.

Ejemplo:

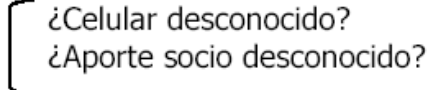
Desconocido	inexistente
No conozco a los vecinos	No tengo segundo nombre.

- Después de decidir si el dominio acepta esos valores, se debe decidir si el sistema los acepta.

Ejemplo: fecha transacción debe existir pero no la conocemos (desconocida pero existente).

- ¿Se puede guardar un valor desconocido? Puede que no valga la pena guardarlo.

Ejemplo:

Socios: 

En cualquiera se debe evitar añadir valores desconocidos.

- Definir los valores de un dominio lo más específicamente que se pueda.

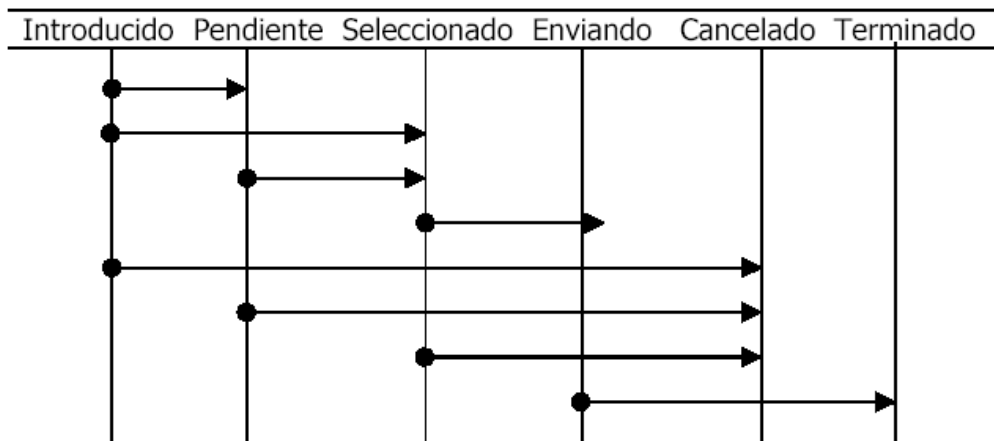
Ejemplo: fecha transacción no es sólo un conjunto de fechas, es el conjunto de fechas desde que el negocio se estableció hasta la actual y si se quiere, eliminar domingos, festivos, etc.

- Es importante captar restricciones y documentar.

Restricciones de Integridad de Transiciones

Definen los estados por los que un registro puede pasar válidamente.

Ejemplo: estados de un pedido.



Las restricciones de integridad de transiciones se usan para que el estado de un pedido nunca cambie de introducido a terminado sin haber pasado por los intermedios. O que un pedido cancelado cambie de estado.

Normalmente un estado es un atributo o campo, pero a veces son varios campos o relaciones.

Ejemplo: (cliente normal preferente: la restricción sería, el crédito del cliente se aumenta si lleva un año con la Empresa; el crédito se controla con un campo pero el tiempo de antigüedad no está en ningún sitio, se calculará de acuerdo con el pedido más antiguo que figure en "pedidos").

Restricciones de Integridad de Entidades

Aseguran integridad de tablas.

Nivel más simple: clave principal, la entidad debe identificarse de forma única.

Restricciones de Integridad Referencial

Mantienen y protegen vínculos (relaciones) entre tablas.

Las claves externas no pueden quedarse huérfanas. Ningún registro puede contener una clave externa que no corresponda con un registro de la tabla principal.

¿Cómo se crean entidades (tablas) huérfanas?

- Añadir registro a la tabla externa con una clave foránea que no corresponde a la tabla principal.
- La clave principal de la tabla principal cambia.
- Se elimina en la tabla principal el registro que hace referencia.

Si se mantiene integridad referencial, los tres casos que se manejan son los siguientes:

- Está prohibido
- No se da a menudo, pero si se permite, se debe asegurar que los cambios se realicen en las claves externas o foráneas. Utilizando el manejador de Bases de datos Access, se debe asegurar que en el SQL Server tenga actualización en cascada. Ejemplo: ¿si se elimina un cliente que pasa con sus pedidos?, debe haber una eliminación en cascada.

Restricciones de Integridad de la Base de Datos

Referencian más de una tabla.

Ejemplo: un cliente no puede ser “preferente” si no ha realizado una compra en los últimos doce meses.

- No confundir restricción de base de datos con proceso de trabajo.

Proceso de trabajo es algo que se hace con la base de datos como añadir pedidos, mientras que una restricción de la base de datos, es una regla acerca del contenido de la base de datos.

Reglas que definen tareas que se hacen por medio de la base de datos son restricciones de procesos de trabajo pero no de la base de datos.

Restricciones de Integridad de Transacciones

Gobiernan las formas en que se puede manipular la base de datos. Son sobre el procesamiento y por lo tanto por sí mismas no son parte del modelo de datos.

En el ejemplo de transferir dinero de una cuenta a otra, involucran muchos registros, tablas e incluso base de datos. Todas las operaciones sobre base de datos son transacciones.

Utilizando el manejador de Bases de datos Access, las opciones que tienen referencia a esta restricción son: Begin Transaction, Commit Transaction, Rollback Transact.

3.4 ALGEBRA Y CALCULO RELACIONAL

El álgebra relacional es un lenguaje procedimental para la manipulación de relaciones.

Procedimental: lenguaje que proporciona un método paso a paso para la solución de problemas

Calculo Relacional: un lenguaje no procedimental para la definición de soluciones a consultas.

No Procedimental: lenguaje que proporciona un medio para establecer que se desea en lugar de cómo hacerlo.

Las operaciones de álgebra relacional manipulan relaciones. Esto significa que estas operaciones usan una o dos relaciones existentes para crear una nueva relación. Esta nueva relación puede entonces usarse como entrada para una nueva operación.

El álgebra relacional consta de las siguientes nueve operaciones.

- Unión, Intersección, Diferencia, Producto: estas operaciones se toman de la Teoría de conjuntos de la matemática.
- Selección, Proyectar, Reunión, División: son nuevas, se aplican al modelo de datos relacional.
- Asignación: es la operación estándar de los lenguajes de computación de dar un valor a un nombre. En este caso la asignación se usa para dar un nombre a una nueva relación que se crea de relaciones existentes.

3.4.1 Unión

Operación del álgebra relacional que crea un conjunto unión de dos relaciones unión compatible. Unión (U) permite combinar los datos de dos relaciones:

Ejemplo:

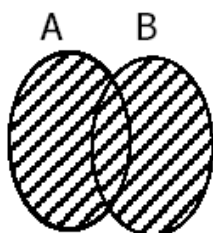
Tabla A: Alumnos > 20

Cod	Nom	Edad	Nac
A	LAURA	30	Col
B	LIDA	31	Per
C	SOFÍA	28	Ven

Tabla B: Alumnos Venezolanos

Cod	Nom	Edad	Nac
C	Sofía	28	Ven
D	José	18	Ven
E	Rafael	19	Ven

AUB



La unión debe ser compatible de dos o más relaciones que tienen columna equivalentes en número y dominio para que el resultado de la unión sea una relación.

$$C = A \cup B$$

ALUMNO := alumnos \geq 20 Alumnos_Venezolanos

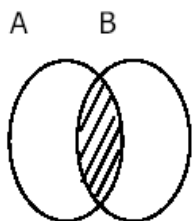
Cod	Nom	Edad	Nac
A	Laura	30	Col
B	Lida	31	Per
C	Sofía	28	Ven
D	José	18	Ven
E	Rafael	19	Ven

3.4.2 Intersección: (\cap)

Permite identificar las filas que son comunes a dos relaciones.

Ejemplo: $A \cap B$

C: $A \cap B$



Tomando la tabla A y B del ejemplo anterior, $A \cap B$ daría como resultado la tabla C.

Tabla C

Código	Nomb	Edad	Nac
C	Sofía	28	Ven

3.4.3 Diferencia

Se indica por un signo (-) permite identificar filas que están en una relación y no en otra.

C: = $A - B$ están en A y no están en B

Tabla C

Código	Nomb	Edad	Nac
A	Laura	30	Col
B	Lida	31	Per

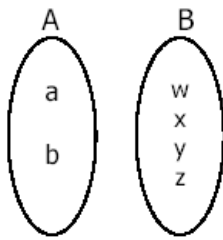
C: = $B - A$ Son los que están en B y no están en A

Tabla C

Código	Nomb	Edad	Nac
D	José	18	Ven
E	Rafael	19	Ven

3.4.4 Producto

Operación del álgebra relacional que crea el producto cartesiano de dos relaciones. Indica por el símbolo (*)



$A*B$

aw, ax, ay, az, bw, bx, by, bz

Es valiosa como un bloque para la construcción de una reunión (join) que es probablemente la operación más importante en el álgebra relacional.

Ejemplo: clientes y pedidos

Tabla A: Clientes

Cod_cli	Nombre
01	Pepito
02	Fulanito

Tabla B: Pedidos

Nro	Fecha	Cod_Cli
97	25/ab/2004	01
98	26/ab/2004	02
99	30/ab/2004	03
100	1/may/2004	04

Con relaciones 1 a Muchos (Especialmente)

$A * B$: cliente * pedidos

Tabla C: $A * B$

Cod_cliente	Nombre	Número	Fecha	Cod_Cliente
01	Pepito	97	25/ab/2004	01
01	Pepito	98	26/ab/2004	02
01	Pepito	99	30/ab/2004	01
01	Pepito	100	1/may/2004	02
02	Fulanito	97	25/ab/2004	01
02	Fulanito	98	26/ab/2004	02
02	Fulanito	99	30/ab/2004	01
02	Fulanito	100	1/may/2004	02

* *Inconsistencia*

3.4.5 Selección o Restricción: SELECT

Se usa para crear una relación a partir de otra relación, seleccionando sólo aquellas filas que satisfacen una condición específica:

Ejemplo:

Tabla Cliente

Cod_cliente	Nombre	Fecha de Ingreso	Nacionalidad	Dirección
1	Eduardo Pérez	25/ab/2004	Colombiano	Calle 5 # 3-56
2	José Velasco	26/ab/2004	Venezolano	Carrera 7 # 9-23
3	Ramiro Luna	30/ab/2004	Mexicano	Calle 0 # 9-12
4	Maria Mora	1/may/2004	Colombiano	Av. 5 # 78-34
5	Nicolás Ortiz	25/ab/2004	Colombiano	Calle 6# 9-56
6	Rosa Ceballos	26/ab/2004	Mexicano	Av. 16 # 23-12
7	Juan Becerra	30/ab/2004	Venezolano	Carrera 17# 5-76
8	Francisco Núñez	1/may/2004	Venezolano	Carrera 9 # 19-34

```
SELECT Cod_cli, Nom_Cli...  
FROM Cliente  
WHERE Nacionalidad = Colombiano
```

Resultado de la Selección

Cod_cliente	Nombre
1	Eduardo Pérez
4	Maria Mora
5	Nicolás Ortiz

3.4.6 Proyección

La operación proyectar puede pensarse como la eliminación de las columnas no deseadas. Operación del álgebra relacional que crea una relación borrando columnas de una relación existente.

Tabla Vendedor

Cod_vendedor	Nombre_vendedor	ID_jefe	Oficina	%_comisión
100	Alberto Jaimes	27	Cúcuta	10
200	José Ramírez	44	Pamplona	11
300	Ana Domínguez	44	Ocaña	9
400	Alfredo Daza	35	Bucaramanga	13
500	Teresa Gómez	27	Pamplona	10
600	Santiago Trujillo	44	Cúcuta	15
700	Ulises Colmenares	27	Bucaramanga	12
800	Fernando Perdomo	27	Ocaña	10

SELECT (Vendedor: %_comisión < 11) [Nombre_vendedor]

Resultado de la Proyección

Nombre_Vendedor
Alberto Jaimes
Ana Domínguez
Teresa Gómez
Fernando Perdomo

3.4.7 Reunión (JOIN)

Se usa para conectar datos a través de relaciones. Quizá es la función más importante en cualquier lenguaje de base de datos.

Existen varias versiones: la reunión natural (natural Join) que es la más importante, la reunión Theta (Theta join), la reunión Externa (Outer Join).

- Reunión Natural

Ejemplo: considérese la relación Venta.

Tabla Venta

Fecha	Cod_cliente	Cod_vendedor	cod_producto	cantidad
28/02	1	100	2241	200
12/02	2	300	2518	300
15/02	2	300	1035	150
19/02	1	500	2518	200
02/02	2	300	1035	200
05/02	3	100	2241	100
22/02	4	400	2518	150
14/02	3	100	2249	50
01/02	2	300	2249	75
04/02	2	300	2241	250

Esta relación almacena el cliente, el vendedor y el producto involucrado en una transacción de venta particular, incluyendo los **Cods** de los 3 elementos dados. Esta información permite hacer conexiones lógicas entre las relaciones, CLIENTE, VENDEDOR y PRODUCTO.

Ejemplo: se quieren conocer los nombres de los clientes que han hecho compras al vendedor 10. primero, se seleccionan aquellas ventas aplicadas solo al vendedor 100 y se ponen entonces en una relación que se llama Venta_100. esta da la siguiente relación.

Relación Venta_100

Fecha	Cod_Cliente	Cod_Vendedor	Cod_Producto	Cantidad
28/02	1	100	2241	200
05/02	3	100	2241	100
14/02	3	100	2249	50

Entonces se puede obtener información deseada reuniendo las relaciones Venta_100 y cliente. Esta operación procede como sigue:

- Se crea el producto Venta_100 y Cliente.
- Todas las filas de esta relación producto se eliminan, excepto aquellas en las cuales el cod_cliente de Venta_100 = cod_cliente de cliente, es decir, las inconsistencias. Observar que hay dos columnas cod_cliente en la relación y que en cada fila los valores de estas dos columnas son idénticos.
- Luego como quedan dos columnas ID_cliente con idéntica información entonces se eliminan.

Resultado del Producto Venta_100 y Cliente

Fecha	Cod_cliente	Cod_vendedor	cod_producto	Cant.	Cod_cliente	Nombre	Fecha de Ingreso	Nacionalidad	Dirección
28/02	1	100	2241	200	1	Eduardo Pérez	25/ab/2004	Colombiano	Calle 5 # 3-56
05/02	3	100	2241	100	3	Ramiro Luna	30/ab/2004	Mexicano	Calle 0 # 9-12
14/02	3	100	2249	50	3	Ramiro Luna	30/ab/2004	Mexicano	Calle 0 # 9-12

Resultado del Join Natural

Fecha	Cod_vendedor	cod_producto	Cant.	Cod_cliente	Nombre	Fecha de Ingreso	Nacionalidad	Dirección
28/02	100	2241	200	1	Eduardo Pérez	25/ab/2004	Colombiano	Calle 5 # 3-56
05/02	100	2241	100	3	Ramiro Luna	30/ab/2004	Mexicano	Calle 0 # 9-12
14/02	100	2249	50	3	Ramiro Luna	30/ab/2004	Mexicano	Calle 0 # 9-12

Join (A, B) se obtiene a través de tres pasos:

- Tomar el producto A y B la relación tendría dos columnas para cada C1...Cn
- Eliminar todas las filas del producto, excepto aquellas en las cuales las columnas C1,...Cn en A son iguales, respectivamente, a los valores de esas columnas en B.
- Proyectar una copia de las columnas C1,...Cn

Indica que si: A tiene K columnas, B tiene m columnas entonces la reunión natural, A y B tendrá $(K + m - n)$ donde n es el # de columnas que A y B tienen en común.

3.4.8 División

Operación del álgebra relacional que crea una relación, seleccionando las filas en una relación que se corresponden con todas las filas en otra relación.

Ejemplo: Listar los vendedores que han vendido todos los productos.

Nuestra Base de Datos contiene 4 productos: Cod: 1035, 2241, 2249 y 2518. Un vendedor satisface la consulta si él o ella han vendido cada uno de estos productos al menos una vez, en otras palabras, para cada uno de estos productos debe haber al menos una fila en VENTA que contenga el Cod_vendedor.

3.4.9 Asignación

Operación del álgebra relacional que da un nombre a una relación.

Tabla A := Select (Vendedor: %_comisión>11)

Tabla A se asigna el resultado de la operación de selección. El símbolo := significa "Es el nombre asignado a")

Proceso de Comprensión y Análisis

- Definir cada uno de los siguientes términos:
 - Relación
 - Atributo
 - Tupla
 - Dominio de atributo
 - Valor nulo
 - Clave
 - Reunión natural
- Describir las circunstancias en las que se usarían cada una de las operaciones siguientes del álgebra relacional:
 - Selección
 - Proyectar
 - Reunir
 - Asignación
 - Sustracción
 - Intersección
 - División

Solución de Problemas

- Usando el siguiente esquema relacional, indicar cuáles operaciones del álgebra relacional podrían usarse para responder las consultas dadas:

CLIENTE (ID_CLIENTE, NOMB_CLIENTE, RENTA_ANUAL)

EMBARQUE (EMBARQUE #, ID_CLIENTE, PESO, CAMIÓN_#, DESTINO)

- ¿Qué clientes tienen una renta anual que excede los cinco millones?
- ¿Cuál es el nombre del cliente 433?

- ¿Cuál es la ciudad destino del transporte ·# 3244?
- ¿Qué camiones han transportado paquetes con un peso por encima de las 100 libras?
- ¿Cuáles son los nombres de los clientes que han enviado paquetes a la ciudad de Medellín, Colombia?
- ¿A qué destinos han enviado paquetes los clientes con renta anual inferior a un millón?

Síntesis Creativa y Argumentativa

- Por medio de tres ejemplos señalar lo entendido por redundancia de datos.
- Señalar por medio de ejemplos algunas anomalías de borrado y de inserción.

Autoevaluación

- ¿Qué es una clave Externa?
- ¿Qué entiende por proceso de normalización?
- ¿En qué se basa la integridad de datos?
- Identificar el término que considere correspondiente a las siguientes definiciones:
 - Da un nombre a una relación.
 - Lista que define los atributos de una relación solución.
 - Lenguaje que brinda un método paso por paso para la solución de problemas.
 - Operación que conecta relaciones.
 - Relación que resulta de una operación de proyectar.
 - Expansión de la reunión natural que excluye todas las filas de ambas relaciones.
 - Crea el producto cartesiano de dos relaciones.
 - Crea el conjunto diferencia de dos relaciones unión-compatible.
 - Afirma la existencia de al menos una fila a la cual se la aplica una condición.
 - Crea el conjunto unión de dos relaciones unión-compatible.

Operación de álgebra relacional que usa una condición para mostrar las filas de una relación.

Un lenguaje no procedimental para la definición de soluciones a consultas.

(Los términos son los siguientes: diferencia, unión, procedimental, selección, cuantificador existencial, selección, join, producto, reunión externa, calculo relacional, asignación, proyección, proyección, lista resultado)

Repaso Significativo

- Señalar por medio de ejemplos las diferencias entre Superclave, Determinante Funcional, Clave Compuesta, Clave Candidata y Clave Primaria?
- ¿Qué mecanismos hay que tener en cuenta para eliminar las anomalías y mantener la integridad de los datos?

Bibliografía Sugerida

GARY W, Hansen. Y JAMES V. Hansen. Diseño y Administración de Bases de Datos. Segunda Edición. España. Prentice Hall Inc. 1996

UNIDAD 4

Implementación de Bases de Datos Relacionales con SQL

Descripción Temática

En la presente unidad se estudiarán las partes de los sistemas de gestión de bases de datos relacionales que tienen que ver con el lenguaje SQL, incluyendo tópicos del manejador de bases de datos ACCESS.

Horizontes

- Explicar los hechos básicos del desarrollo histórico de los sistemas de gestión de los sistemas de bases de datos relacionales.
- Definir un esquema relacional de base de datos en SQL
- Formular consultas de SQL de variada complejidad.
- Insertar, actualizar y borrar datos de una base de datos relacional a través de órdenes SQL.
- Observar algunos aspectos de cómo se escriben instrucciones SQL en un lenguaje de programación tradicional o en ACCESS.

Núcleos Temáticos y Problemáticos

- Introducción a la Implementación Relacional
- Tipos de Datos
- Definición de Tablas
- Manipulación de Datos
- Diseño de Sistemas de Base de Datos Relacionales
- Manejo de Motor de Bases de Datos Access

Proceso de Información

4.1 INTRODUCCIÓN A LA IMPLEMENTACIÓN RELACIONAL

Los lenguajes de definición de datos de SQL y ACCESS permiten definir tablas de bases de datos mediante la creación de tipos de datos definidos por el usuario, tablas, columnas y restricciones sobre todos estos. Los tipos de datos definidos por el usuario se desarrollan a partir de tipos básicos suministrados por el sistema y añadiendo ciertas restricciones sobre estos. Al definir las tablas se definen sus columnas con sus tipos de datos y sus restricciones. Las restricciones pueden ser reglas que restringen los valores que pueden estar en las columnas, éstas pueden indicar si se permiten valores nulos y pueden también garantizar la unicidad de los valores de una columna dentro de la tabla. Ellas también definen claves primarias o externas de una columna o múltiples columnas.

El desarrollo de las aplicaciones de los sistemas de bases de datos pueden tener lugar en un entorno que brinde capacidades bien establecidas en la interfaces gráficas del usuario. Usando ACCESS como ejemplo estudiamos la definición de aplicaciones donde se definen consultas y la presentación de los resultados de las consultas.

4.2 TIPOS DE DATOS

SQL define los siguientes tipos de datos:

Númericos Exactos

- Integer (enteros)
- Small Integer (enteros pequeños)
- Numeric (p, e) (numéricos)
- Decimal (p,e)

Para los dos últimos tipos de datos (numérico y decimal) se indica una precisión (p) y una escala (e). La precisión indica el total de números o dígitos en el número y la escala indica cuántos de éstos están a la derecha del punto decimal.

Númericos Aproximados

- Real
- Double precisión (doble precisión)

- Flota (flotante)

Estos tipos de datos se usan normalmente para cálculos científicos y de Ingeniería.

Cadenas de Caracteres

- Character (n) (carácter)
- Character varying (n) (character variable)

Los campos de character siempre almacenan n caracteres, aún cuando tengan que rellenar con blancos a la derecha para completar la longitud n. Los campos character varying sólo almacenan el número real de caracteres que se introdujeron (hasta un máximo de n).

Cadenas de Bits

- Bit (n)
- Bit varying (n)

Estos campos se usan para banderas u otras máscaras de bits para el control.

Fecha y Hora

- Date (fecha)
- Time (hora)
- Timestamp (sello de tiempo)
- Time con tiempo zonal
- Timestamp con tiempo zonal.

El tipo Date (fecha) se da en el orden año, mes, día con cuatro dígitos para el año. El Time se da en horas (0 a 23), minutos, segundos y décimas de segundo. El Timestamp es la fecha más la hora (date plus time).

Intervalos

- Year-month (año – mes)
- Day – time (día – hora)

Un intervalo es la diferencia entre dos fechas (año-mes) o entre dos horas (día-hora). Por ejemplo, entre diciembre 2004 y enero de 2006, el intervalo es un año y un mes.

4.2.1 Términos

Restricción: una regla que restringe los valores en una base de datos.

Valor por Defecto: un valor que se inserta automáticamente si el usuario no lo especifica en una entrada.

Definición de Dominio: un tipo de datos especializados definido dentro de un esquema y usado en las definiciones de columna.

4.3 DEFINICIÓN DE TABLAS

Las tablas se definen en tres pasos:

- Dar el nombre de la tabla
- Definir cada columna, posiblemente incluyendo restricciones de columna.
- Definir las restricciones de la tabla.

A continuación se da una definición de esquema para una base de datos.

```
CREATE SCHEMA COSNTRUCTORA – PREMIER
    AUTORIZATION Pedro_Pérez
    domain definitions
```

```
CREATE TABLE TRABAJADOR (
    ID_Trabajador          IDENTIFICADOR PRIMARY KEY.
    Nomb_Trabajador        CHARACTER (12),
    Tarifa_Hr              NUMERIC (5,2),
    Tipo-oficio            CHARACTER (8)
    ID_supervisor          NUMERIC (4),
    FOREIGN KEY ID_Supervisor REFERENTES TRABAJADOR
    ON DELETE SET NULL)
```

ID_ trabajador	Nomb_ trabajador	Tarifa_Hr	Tipo_ oficio	ID_ supervisor
1235	Francisco	2000	Electricista	1311
1412	Guillermo	1800	Fontanero	1520
2920	Gloria	2200	Carpintero	
3231	Marisol	1500	Albañil	
1520	Ricardo	1800	Fontanero	
1311	Claudia	2000	Electricista	
3001	Bernardo	1500	Albañil	3231

```
CREATE TABLE ASIGNACIÓN (
    ID_Trabajador          IDENTIFICADOR,
    ID_edificio            IDENTIFICADOR,
    Fecha_inicio          DATE,
    Nº_dias                INTERVAL DAY (3);
    PRIMARY KEY (ID_trabajador, ID_edificio),
    FOREIGN KEY ID_ trabajador REFERENCES TRABAJADOR
    ON DELETE CASCADE,
    FOREIGN KEY ID_ edificio REFERENCES EDIFICIO
    ON DELETE CASCADE)
```

ID_ trabajador	ID_ edificio	Fecha_ inicio	Nº_ dias
1235	312	10/10	5
1412	312	01/10	10
1235	515	17/10	22
2920	460	05/10	18
1412	460	08/12	18
2920	435	28/10	10
2920	210	10/11	15
3231	111	10/10	8
1412	435	15/10	15
1412	515	05/11	8
3231	312	24/10	20
1520	515	09/10	14
1311	435	08/10	12
1412	210	15/11	12
1412	111	01/12	4
3001	111	08/10	14
1311	460	23/10	24
1520	312	30/10	17
3001	210	27/10	14

```
CREATE TABLE EDIFICIO (
    ID_edificio            IDENTIFICADOR PRIMARY KEY,
    DIR_edificio           CHAR (12)
    TIPO                   CHAR (9) DEFAULT 'Oficina'
    CHECK (TIPO IN ('oficina', 'almacen', 'comercio', 'residencia')),
    Nivel_calidad          NUMERIC (1),
    Categoría              NUMERIC (1) DEFAULT 1
    CHECK (Categoría > 0 and Categoría < 4))
```

ID_edificio	DIR_edificio	TIPO	Nivel_calidad	Categoría
312	Calle 0 # 9-12	Oficina	2	2
435	Av. 5 # 78-34	Comercio	1	1
515	Calle 6# 9-56	Residencia	3	1
210	Av. 16 # 23-12	Oficina	3	1
111	Carrera 17# 5-76	Oficina	4	1
460	Calle 8B # 5-58	Almacen	3	3

4.4 MANIPULACIÓN DE DATOS

Consultas Simples: es una consulta que afecta a una sola tabla de la base de datos. Las consultas simples nos ayudan a ilustrar la estructura básica del SQL.

Ejemplo: consulta = ¿Quiénes son los fontaneros?

```
SELECT Nomb_Trabajador
FROM TRABAJADOR
WHERE OFICIO = "Fontanero"
Salida:      Nomb_Trabajador
           Guillermo
           Ricardo
```

La cláusula SELECT señala las columnas que se desean en la consulta.

La cláusula FROM lista las tablas que son preferidas por la consulta

La cláusula WHERE contiene una condición para seleccionar las filas de las tablas indicadas que se dan en la cláusula FROM

Ejemplos

- Consulta = relacione todos los datos de los edificios que sean oficinas.

```
SELECT*
FROM EDIFICIO
WHERE Tipo = 'Oficina'
El * en la cláusula SELECT significa la fila completa
```

Resultado

ID_edificio	DIR_edificio	TIPO	Nivel_calidad	Categoría
312	Calle 0 # 9-12	Oficina	2	2
210	Av. 16 # 23-12	Oficina	3	1
111	Carrera 17# 5-76	Oficina	4	1

- Consulta: ¿Cuál es la tarifa semanal de cada electricista?

```
SELECT Nomb_Trabajador, 'Tarifa semanal =', 40 * Tarifa_HR
FROM TRABAJADOR
WHERE Tipo = 'Electricista'
ORDER BY Nomb_Trabajador
```

Resultado

Nomb_ trabajador	Tarifa_Hr
Francisco	Tarifa semanal = 2000
Claudia	Tarifa semanal = 2000

- Consulta: ¿Quiénes tienen una tarifa por hora entre \$10 y \$12?

```
SELECT *
FROM TRABAJADOR
WHERE TARIFA_HR >= 1800 AND TARIFA_HR <= 2000
```

Resultados

ID_ trabajador	Nomb_ trabajador	Tarifa_Hr	Tipo_ oficio	ID_ supervisor
1235	Francisco	2000	Electricista	1311
1412	Guillermo	1800	Fontanero	1520
1520	Ricardo	1800	Fontanero	
1311	Claudia	2000	Electricista	

Uso de operadores de comparación conectores Boléanos =, <>, <, >, <=, >=
AND, OR, NOT

- Consulta: indique los fontaneros, albañiles y electricistas.

```
SELECT *
FROM TRABAJADOR
WHERE Oficio IN ('fontanero', 'albañil', 'electricista')
```

Resultado

ID_ trabajador	Nomb_ trabajador	Tarifa_Hr	Tipo_ oficio	ID_ supervisor
1235	Francisco	2000	Electricista	1311
1412	Guillermo	1800	Fontanero	1520
3231	Marisol	1500	Albañil	
1520	Ricardo	1800	Fontanero	
1311	Claudia	2000	Electricista	
3001	Bernardo	1500	Albañil	3231

- Consulta: encontrar todos los caracteres cuyo oficio comience con "Elec"

```
SELECT *
FROM TRABAJADOR
WHERE Oficio Like 'Elec%'
```

ID_ trabajador	Nomb_ trabajador	Tarifa_Hr	Tipo_ oficio	ID_ supervisor
2920	Gloria	2200	Carpintero	

Comodines % (tanto por ciento),y _ (Subrayado) equivale por un carácter.

- Consulta Multi – Tablas

— ¿Cuáles son los oficios de los trabajadores asignados al edificio 435?

Se encuentran en Trabajador y Asignación

```
SELECT Oficio
FROM TRABAJADOR, ASIGNACIÓN
WHERE TRABAJADOR. ID_trabajador = ASIGNACIÓN. ID_trabajador
AND ID_edificio = 435
```

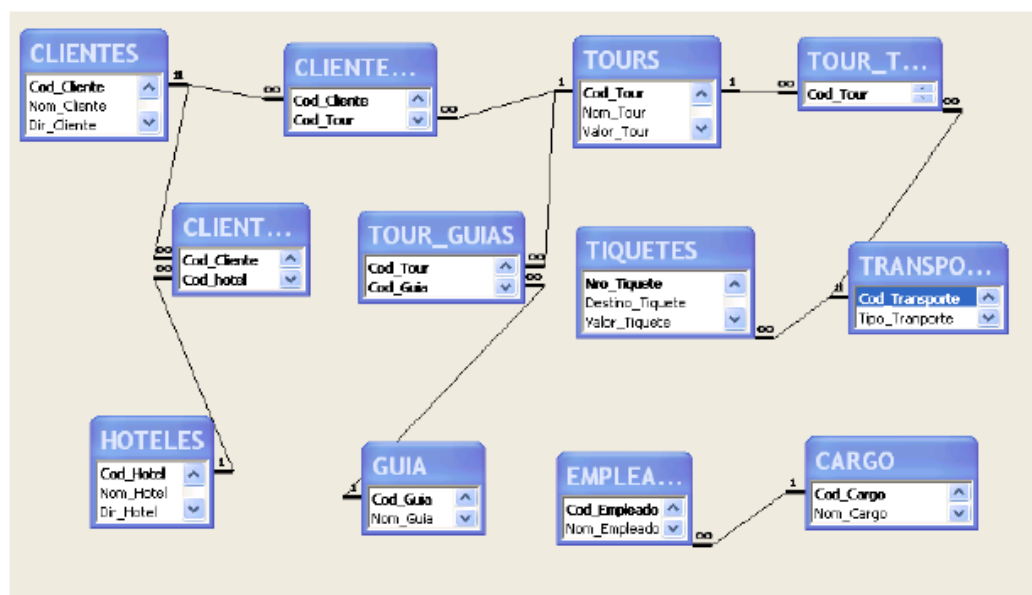
- Indique los nombres de los trabajadores asignado a Edificios que sen oficinas.

```
SELECT Nomb_trabajador
FROM TRABAJADOR, ASIGNACIÓN, EDIFICIO
WHERE TRABAJADOR. ID_trabajador = ASIGNACIÓN. ID_trabajador AND
      ASIGNACIÓN. ID_edificio = EDIFICIO. ID_Edificio AND
      Tipo = "Oficina".
```

4.5 DISEÑO DE SISTEMAS DE BASE DE DATOS RELACIONALES

Una agencia de viajes se especializa en vender Tuores (recorridos turísticos) y tiquetes (pasajes) de viaje a sus clientes. Los Toures (la Costa, zona cafetera, amazonas, guajira, santanderes, etc). Los Toures tienen guías turísticos que no son empleados de la agencia y se les paga lo que cobren. Los tiquetes son vendidos a la agencia de viajes por la empresa de transporte y la agencia de viajes se los vende a sus clientes. Los clientes son alojados en hoteles escogidos por la agencia de viajes. La agencia de viajes le paga a sus empleados de acuerdo al cargo que desempeñan (gerente, secretaria, vendedor, etc).

- Elaborar del diagrama entidad relación
- Crear tablas y relaciones



Modelo Entidad Relación de la Agencia de Viajes

4.6 MANEJO DE MOTOR DE BASES DE DATOS ACCESS

El MS ACCESS es una aplicación Windows que permite crear y administrar información almacenada en base de datos relacionales.

4.6.1 Definiciones Principales

Base de Datos: es una conjunto de información la cual ha sido organizada y presentada para servir a un propósito determinado.

Modelo Relacional: una base de datos de tipo relacional se muestra como una o más tablas rectangulares de filas y columnas.

Entidad: persona, lugar, objeto o evento de interés acerca del cual se recogen o procesan datos. Por ejemplo: pacientes, clientes, artículos son entidades de un hospital y de una tienda comercial respectivamente.

Dato: elemento individual de un campo, se identifica por un nombre y tiene un valor específico.

Campo: conjunto de datos de un mismo tipo. Por ejemplo: conjunto de nombres, conjunto de notas, conjunto de direcciones, etc.

Registro: conjunto de datos pertenecientes a una misma entidad. El registro consta de campos, cada campo tiene una longitud definida, por lo tanto los registros son de longitud fija.

Atributos de una entidad: cada entidad tiene características propias. Por ejemplo: la entidad de alumnos tiene las siguientes características: nombres, apellidos, edad, sexo, fecha de nacimiento, grado, dirección, teléfono, etc. a cada una de estas características o propiedades de la entidad se denomina atributo de la entidad.

4.6.2 Elementos de una Base de Datos de Ms Access

Tablas

Es un conjunto de datos dispuesto en una estructura de filas y columnas. En una tabla las filas se denominan registros y las columnas campos. En una tabla la primera fila contiene los nombres de campo. Cada campo contiene determinado tipo de datos y tiene una longitud expresada en el número de caracteres máximo del campo. Para crear una tabla será necesario definir su estructura:

- El nombre de la tabla
 - Los nombres de campo
 - Los tipos de datos de cada campo
 - Las propiedades o características de cada campo
 - El campo clave (clave principal)
- Consultas: es una pregunta específica acerca de los datos almacenados en la base de datos. Veamos algunos tipos de consultas:
 - Consulta de Selección: es una de las consultas más utilizadas. Con las consultas de selección podrá ver datos de las tablas, analizar dichos datos e incluso hacer cambios en ellos. Es posible ver datos procedentes de una tabla o bien agregar campos de varias tablas distintas.
 - Consulta de Tabla de Referencias Cruzadas: presenta los datos con títulos en las filas y columnas, como en una hoja de cálculo. Con una consulta de tabla de referencias cruzadas se puede resumir gran cantidad de información en un formato de fácil lectura.
 - Consulta de Actualización: permite cambiar los valores de un determinado campo en todos los registros.

Formularios

Son objetos en pantalla a través de los cuales podemos ingresar, mostrar y editar información de los registros de la tabla o consulta. Cada formulario está constituido por controles que permiten una mayor rapidez y menor posibilidad de error al momento de ingresar información.

Informes

Se pueden mostrar los registros agrupados y ordenados según múltiples claves de ordenamiento, los subtotales y totales por subgrupos y grupos.

Macros

Es una secuencia de instrucciones creadas para ejecutar un proceso. Cada instrucción origina una acción. Sirven para automatizar tareas rutinarias repetitivas.

Módulos

Son conjuntos de declaraciones, instrucciones almacenados como una unidad. Es decir es un programa en un lenguaje de programación como por ejemplo el Access Basic.

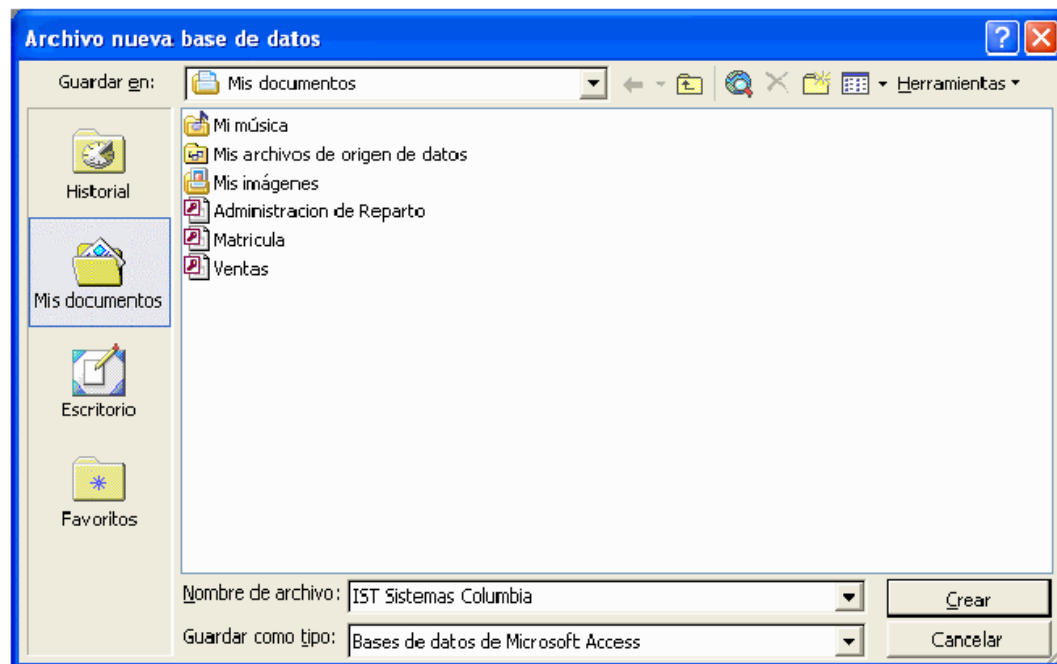
4.6.3 Manejo de Access

El ejemplo a continuación es un sistema de información para un instituto Superior Tecnológico donde se desarrollará una Base de Datos para la correcta manipulación de toda su información académica. En el cual se desea alcanzar algunos objetivos tales como:

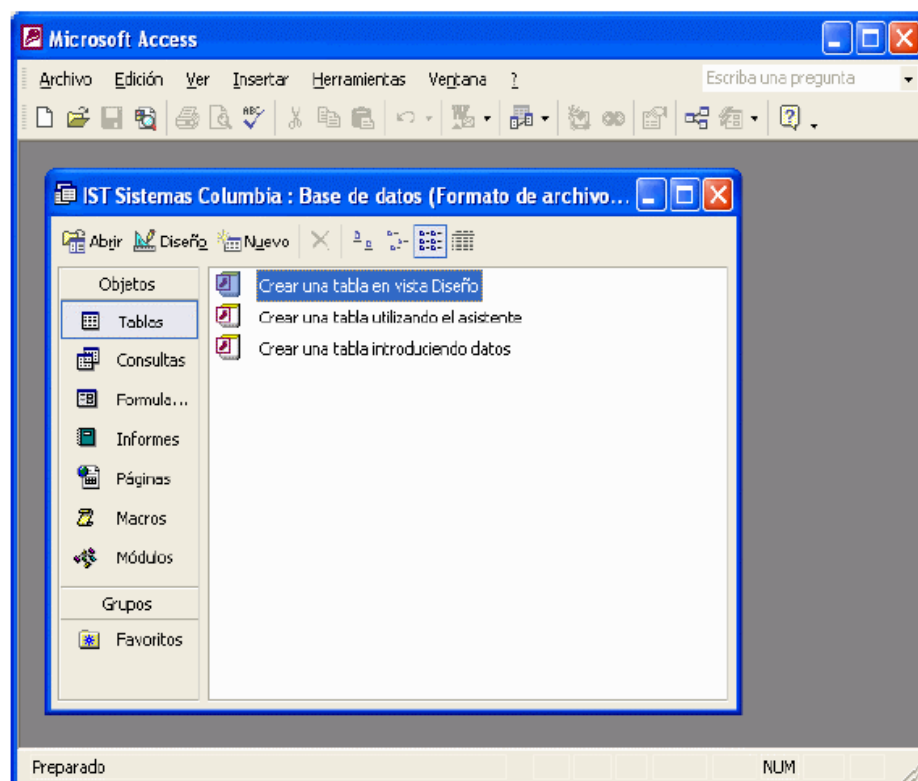
- Solución al problema de administración académica.
- Extraer datos de una realidad para reflejarlos en los campos de las tablas.
- Desarrollar consultas ya sean de selección o de tabla de referencias cruzadas, para hacer más fácil la visualización de datos específicos.

Ingreso a Ms Access 2000

- Haga clic en el botón Inicio de la Barra de Tareas – Programas.
- En el menú Programas – Luego Microsoft Access.
- En la ventana de MS Access, podemos crear una Nueva Base de Datos en blanco o abrir una base de datos existente.
- Después de este paso elegir Base de Datos en blanco, se mostrará la siguiente ventana de diálogo para guardar la nueva Base de Datos.
- En la Opción Guardar en: seleccione la ruta donde almacenará la base de datos.
- Ingrese un nombre para la base de datos, por ejemplo: IST Sistemas Columbia
- Al terminar, haga clic en el botón crear.

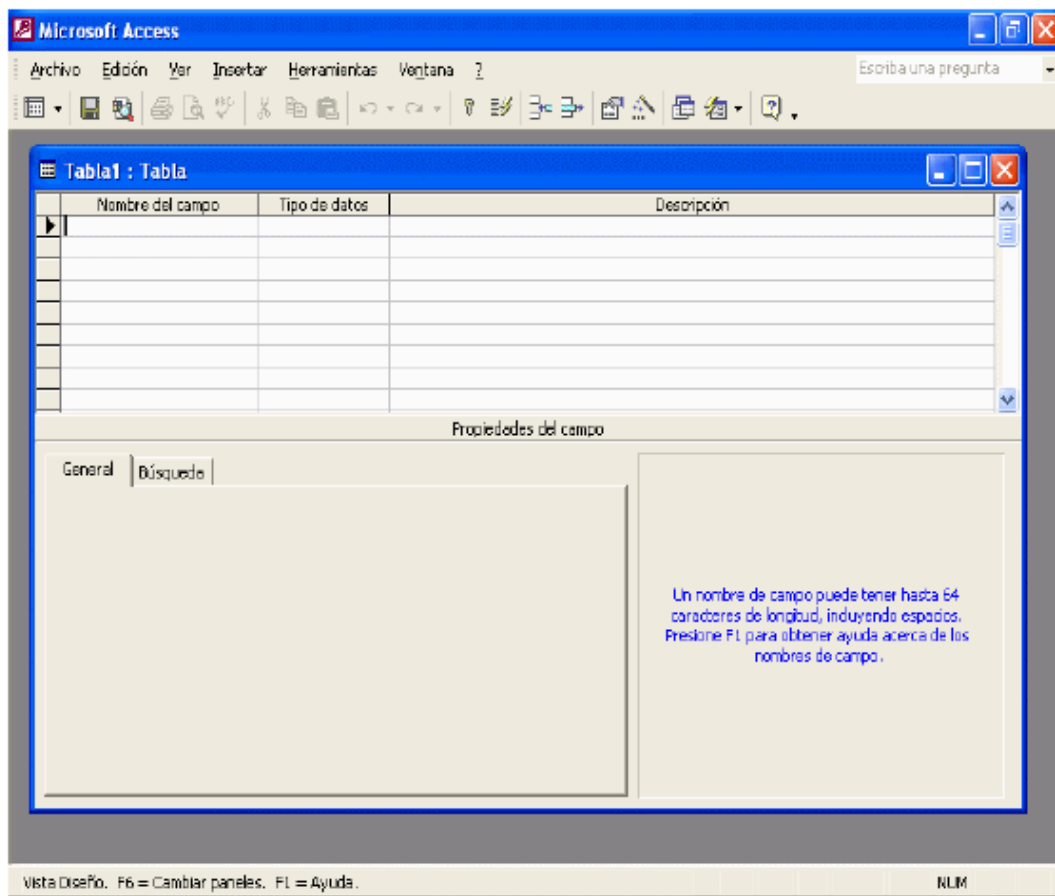


La Ventana de Ms Access



Creación de una Tabla

- En la ventana de base de datos, realice un clic en la ficha Tablas.
- Elija el botón "Diseño".
- Esto creará una tabla vacía en el modo de presentación de diseño.



Tipos de Datos

TIPO DE DATO	ALMACENA	TAMAÑO
Texto	Caracteres alfanuméricos	Hasta 255 bytes
Memo	Caracteres alfanuméricos	Hasta 64.000 bytes
Númérico	Valores Enteros o fraccionarios	1,2,4 u 8 bytes
Fecha/Hora	Fechas y Horas	8 bytes
Moneda	Valores de moneda	8 bytes
Autonumérico	Valor numérico de incremento automático	4 bytes

Si/No	Valores Booleanos (Verdadero o Falso)	1 byte
Objeto OLE	Imágenes o Gráficos	Hasta 1 Gb.

Tipo de Dato Numérico

VALOR DE CAMPO	RANGO	LUGARES DECIMALES	TAMAÑO
Byte	0 – 255	Ninguno	1 byte
Entero	-32.768 a 32.767	Ninguno	2 byte
Entero Largo	-2.147.486.648 a 2.147.486.647	Ninguno	4 byte
Simple	-3.4*10 a 3.4*10	7	4 byte
Doble	-1.797*10 a 1.797*10	15	8 byte

Propiedades del Campo

Es la apariencia que tiene los datos, evita la introducción incorrecta de los mismos, especifica valores predeterminados, acelera la búsqueda y la ordenación de la tabla mediante índices. Las propiedades de los campos se visualizan y se modifican individualmente para cada campo.

PROPIEDADES DEL CAMPO	PARA
Tamaño de campo	Ajusta el tamaño de un campo de tipo texto o limita el rango de valores permitidos en un campo numérico.
Máscara de entrada	Presentación de un formato de un campo para no escribirlos y los datos se ajusten a la máscara.
Reglas de validación	Permite limitar los datos introducidos en un campo.
Texto de validación	Texto que se muestra al infringir la regla de validación

Creación de una Base de Datos

- La Nueva Base de Datos a crear se llamará IST "Sistemas Columbia"
- Crear las siguientes tablas con sus respectivas claves principales y claves externas:

— Tabla: Ciclos

Campos	Tipo	Tamaño	Máscara Entrada
Cic_Codigo	Texto	6	LLL999
Cic_Nombre	Texto	13	
Cic_Nº Alumnos	Texto	Byte	99

Clave Principal: Cic_Codigo

Para la Tabla Ciclos se tiene los siguientes códigos:

Cic001	Primer Ciclo
Cic002	Segundo Ciclo
Cic003	Tercer Ciclo
Cic004	Cuarto Ciclo
Cic005	Quinto Ciclo
Cic006	Sexto Ciclo

— Tabla: Alumnos

Campos	Tipo	Tamaño	Máscara Entrada	Regla Validación
Alu_Codigo	Texto	6	LLL999	
Alu_Apellidos	Texto	50		
Alu_Nombres	Texto	20		
Alu_Sexo	Si/No			"Becado" o "Pagante"
Alu_Nº Matricula	Texto	3	999	

Cic_Codigo	Texto	6	LLL999	
Alu_FechaNacimiento	Fecha/Hora	Fecha corta		
Alu_Domicilio	Texto	50		
Alu_Condición	Texto	10		

Clave Principal : Alu_Codigo

Clave Externa : Cic_Codigo

— Tabla: Asignaturas

A continuación se va a crear la tabla asignaturas con sus claves principales por cada curso.

Campos	Tipo	Tamaño	Máscara Entrada
Asi_Codigo	Texto	6	LLL999
Cic_Codigo	Texto	6	LLL999
Asi_Nombre	Texto	50	
Asi_Duración	Texto	20	
Doc_Codigo	Texto	6	LLL999

Clave Principal: Asi_Codigo

Clave Externa: Cic_Codigo

Doc_Codigo

Asignaturas :

Asi_Codigo	Nombre Asignatura	Cic_Codigo	Doc_Codigo
Mat001	Matemática I	Cic001	Doc009
Apli001	Aplicativo I	Cic001	Doc002
Met001	Metodología Investigación	Cic001	Doc013
Adm001	Administración	Cic001	Doc011
Len001	Lenguaje Programación I	Cic001	Doc005
Ing001	Inglés I	Cic001	Doc002
Inf001	Informática Básica	Cic001	Doc012

ADMINISTRACION DE BASE DE DATOS

Fun001	Fundamentos Programación I	Cic001	Doc008
Mat002	Matemática II	Cic002	Doc009
Apli002	Aplicativo II	Cic002	Doc002
Len002	Lenguaje Programación II	Cic002	Doc005
Ing002	Inglés II	Cic002	Doc002
Eco002	Economía	Cic002	Doc014
Fun002	Fundamentos Programación II	Cic002	Doc008
Con002	Contabilidad I	Cic002	Doc001
Sop003	Sistemas Operativos I	Cic002	Doc006
Apli003	Aplicativo III	Cic003	Doc002
Len003	Lenguaje Programación III	Cic003	Doc005
Ing003	Inglés III	Cic003	Doc002
Est003	Estadística	Cic003	Doc014
Ama003	Análisis Matemático	Cic003	Doc009
Con003	Contabilidad II	Cic003	Doc001
Sop004	Sistemas Operativos II	Cic004	Doc006
Ans004	Análisis de Sistemas	Cic004	Doc006
Len004	Lenguaje Programación IV	Cic004	Doc005
Ing004	Inglés IV	Cic004	Doc002
Cpr004	Costos y Presupuestos	Cic004	Doc011
Mfi004	Matemática Financiera	Cic004	Doc009
Bda004	Base de Datos	Cic004	Doc007
Mbd005	Manejo de Base de Datos I	Cic005	Doc007
Red005	Redes (Novell, Netware)	Cic005	Doc003
Int005	Internet I (Interdev, JScript)	Cic005	Doc003
Dsi005	Diseño de Sistemas	Cic005	Doc006
Afi005	Administración Financiera	Cic005	Doc009
Mfp005	MS Front Page	Cic005	Doc003
Ing005	Inglés V	Cic005	Doc002
Len005	Lenguaje Programación V	Cic005	Doc005

Mbd006	Manejo de Base de Datos II	Cic006	Doc007
Int006	Internet II	Cic006	Doc003
Ecp006	Ensamblaje y Configuración Pc's	Cic006	Doc010
Len006	Lenguaje Programación VI	Cic006	Doc005
Ing006	Inglés VI	Cic006	Doc002
Mcr006	Macromedia Flash 5.0	Cic006	Doc004

— Tabla : Distritos

Campo	Tipo	Tamaño	Máscara Entrada
Dis_CodDistrito	Texto	3	L99
Dis_Nombre	Texto	20	

Clave Principal: Dis_CodDistrito

Para la Tabla Distritos se tiene los siguientes códigos:

Dis_CodDistrito	Dis_Nombre
L01	Los Olivos
L02	La Molina
L03	San Borja
L04	Chorrillos
L05	Barranco
L06	San Isidro
L07	Surco
L08	San Miguel
L09	Miraflores
L10	Lince
L11	Jesús María
L12	Sta. Anita
L13	Pueblo Libre
L14	Surquillo

— Tabla: Docentes

Campo	Tipo	Tamaño	Máscara Entrada
Doc_Codigo	Texto	6	LLL999
Doc_Apellidos y Nombres	Texto	50	
Doc_Domicilio	Texto	50	
Dis_CodDistrito	Texto	3	L99
Doc_Telefono	Texto	8	999-9999

Clave Principal: Doc_Codigo

Clave Externa: Dis_CodDistrito

Los códigos de los Docentes son:

Doc_Codigo	Doc_Apellidos y Nombres
Doc001	Vizarreta Cajo, Luis
Doc002	Zerpa Solano, Patricia
Doc003	Vives Romero, Carlos
Doc004	Tarazona Moreno, Mauro
Doc005	Sayán Lopez, Felix
Doc006	Saavedra Castillo, Claudio
Doc007	Romero Vargas, Ricardo
Doc008	Pizarro Alarcón, Domingo
Doc009	Pereyra Lozano, David
Doc010	Orozco Cruz, Eduardo
Doc011	Minaya Flores, Wilfredo
Doc012	Martínez Valdez, Emilio
Doc013	Guzmán Chavez, Julio
Doc014	Falcón Cáceres, Omar

— Tabla: Calificaciones

Campo	Tipo	Tamaño	Máscara Entrada
Alu_Codigo	Texto	6	LLL999
Cal_PParcial1	Número	Byte	99
Cal_PParcial2	Número	Byte	99
Cal_PParcial3	Número	Byte	99
Cal_PGeneral	Número	Byte	99
Cal_Aplazado	Si/No		

Clave Principal: Alu_Codigo

Luego de haber creado todas las tablas para esta Base de Datos se va a proceder a relacionar las tablas

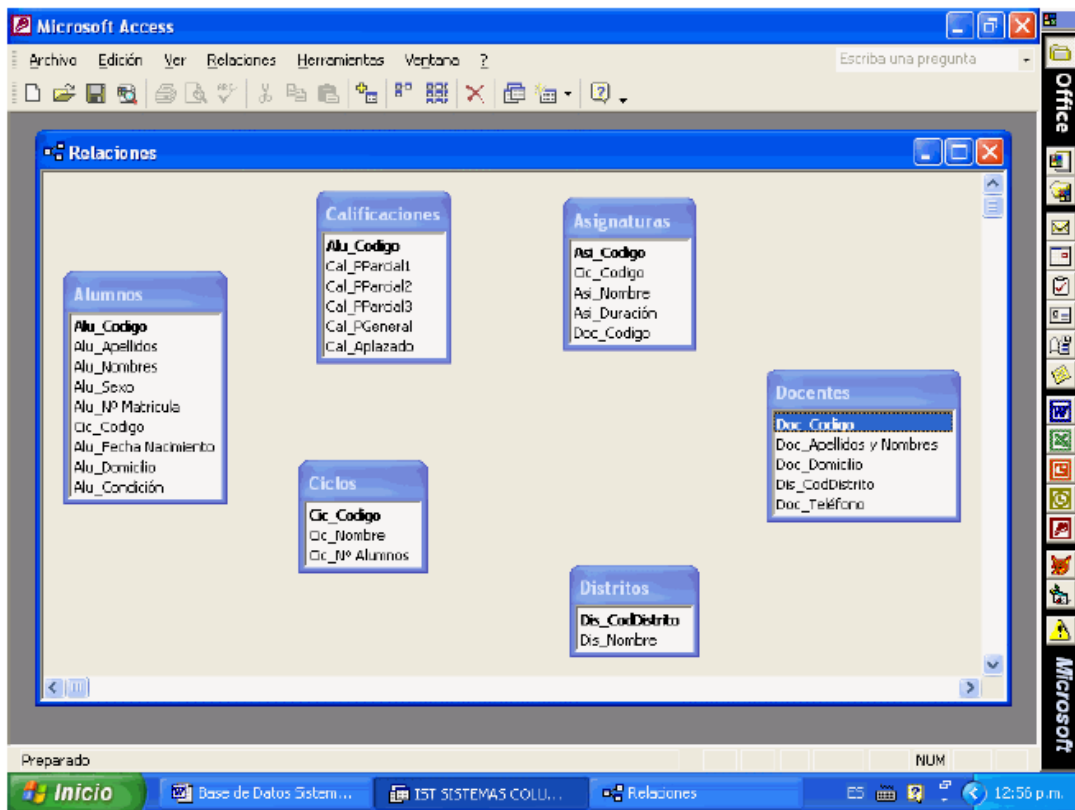
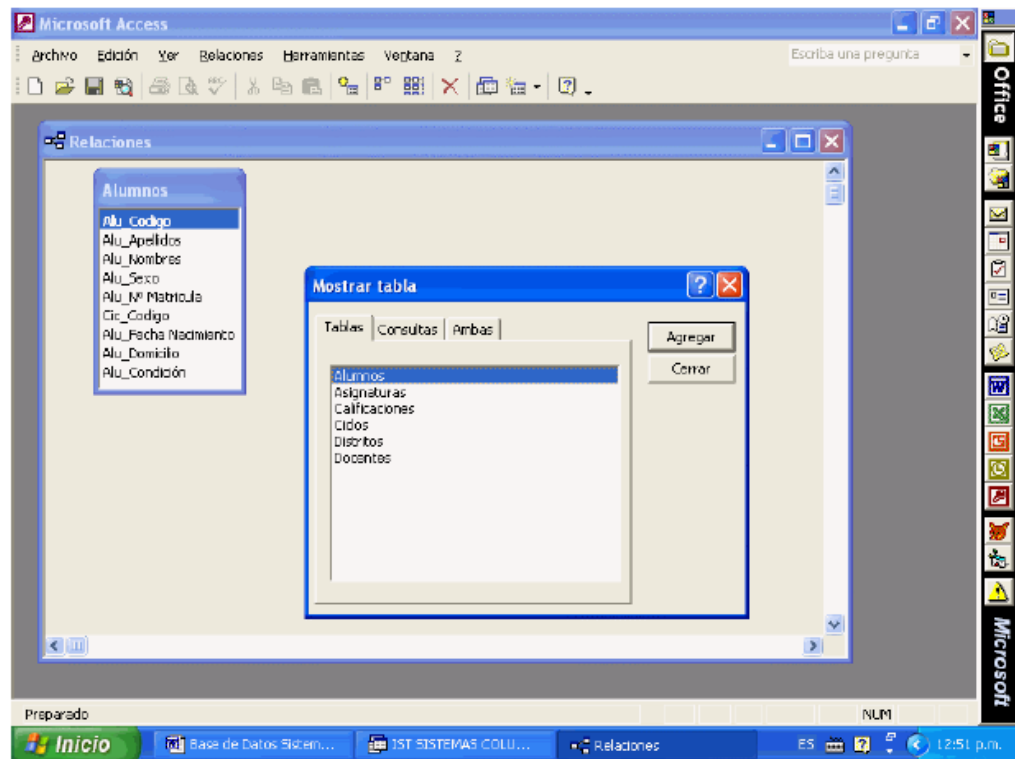
Relaciones

Se desean relacionar las tablas de la Base de Datos creada. Para que exista una relación entre tablas, éstas deben cumplir 2 condiciones:

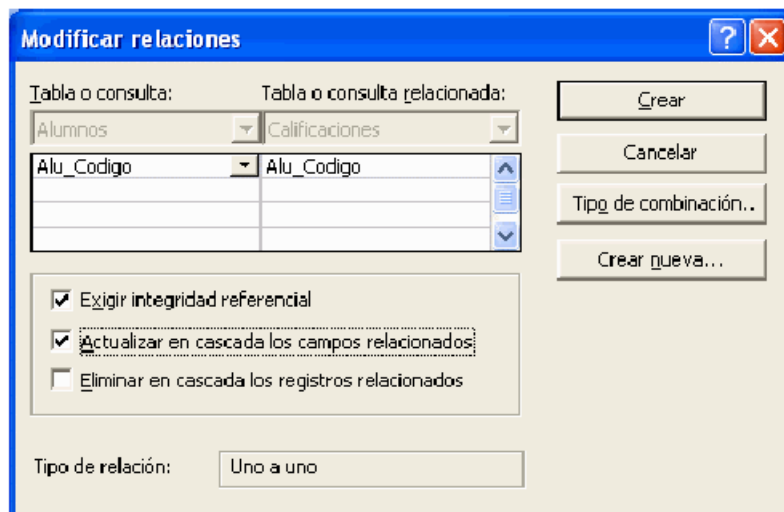
- De referencia los campos a relacionar en las dos tablas deben tener el mismo nombre.
- El contenido de los campos debe ser el mismo.

El procedimiento para establecer relaciones entre tablas se describe a continuación:

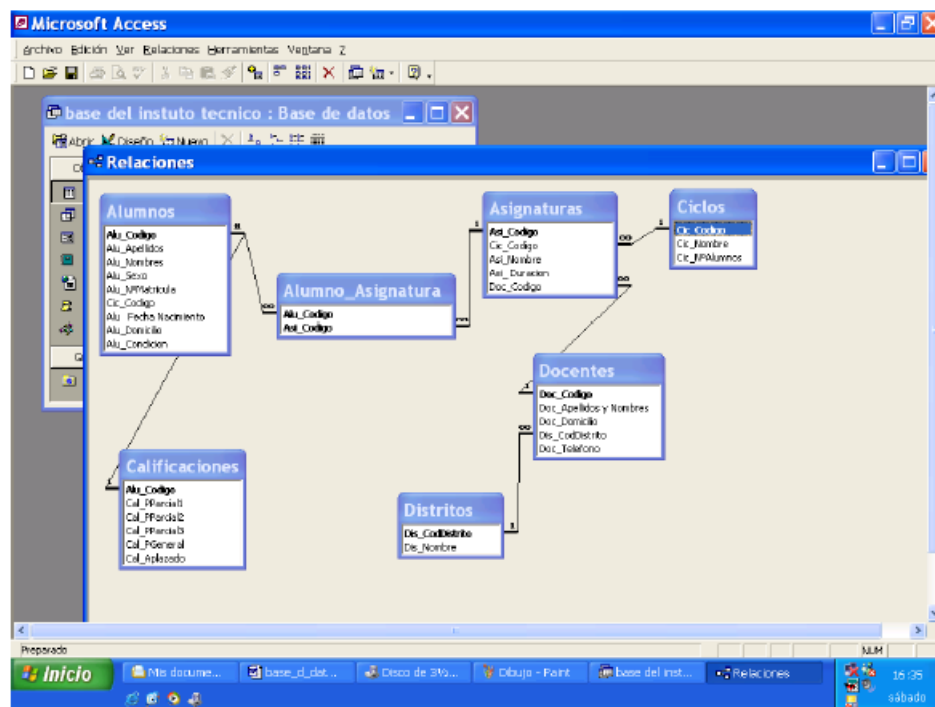
Elija menú Herramientas-Relaciones. En la pantalla se muestra la ventana Relaciones en la cual se irán agregando las estructuras de las tablas que desea relacionar desde la ventana Mostrar tabla. Una vez que haya agregado todas las tablas que desee, cierre la ventana Mostar tabla.



La ventana de relaciones presentará las tablas a relacionar como se observa en el siguiente gráfico. Con la ayuda del mouse arrastrar el campo a Relacionar desde la tabla Principal (en la cual el campo en común es clave principal) hacia el campo de la secundaria o relacionada tabla.



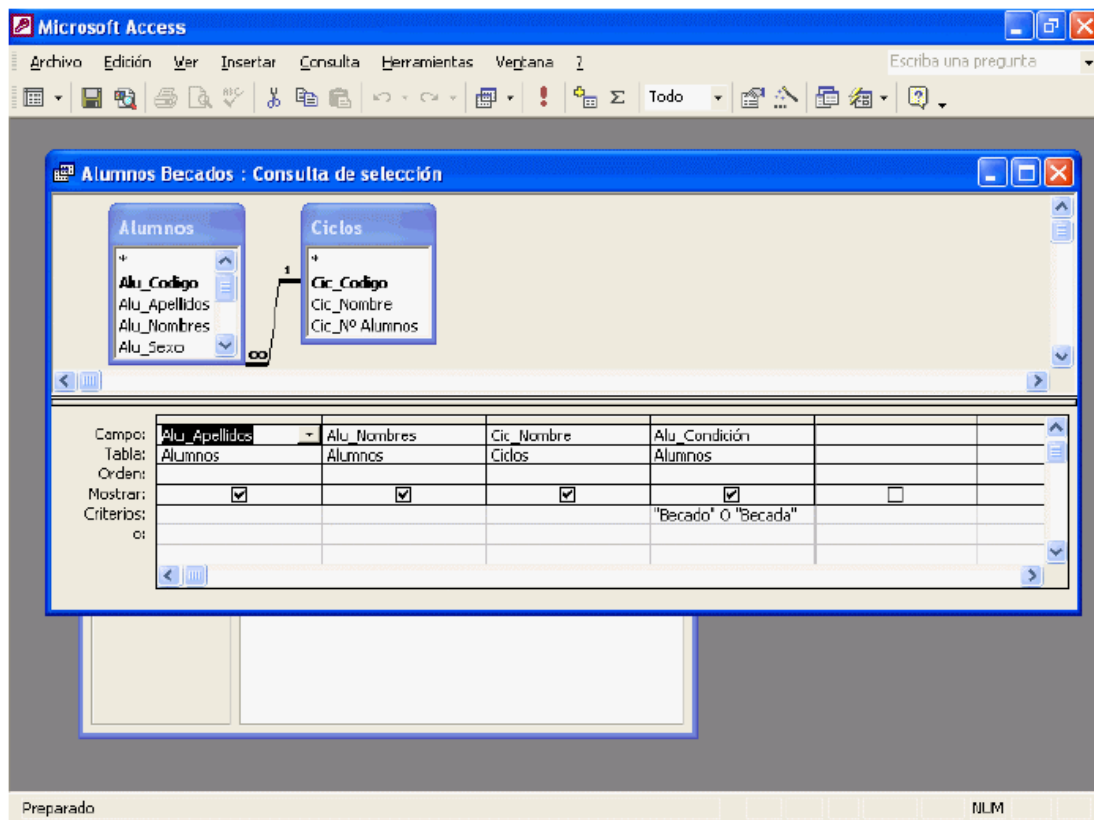
En la ventana de relaciones puede marcar la casilla verificación Exigir integridad referencial y Actualizar en cascada los registros relacionados.



Consultas de Selección

- Mostrar los apellidos y nombres de los alumnos becados por cada ciclo.
Desarrollo
 - Seleccionar el objeto de consulta
 - Dar clic en el botón Diseño
 - En la ventana mostrar tabla seleccionar la tabla Alumnos, Ciclos,
 - luego dar clic en el botón agregar
 - para finalizar clic en el botón cerrar

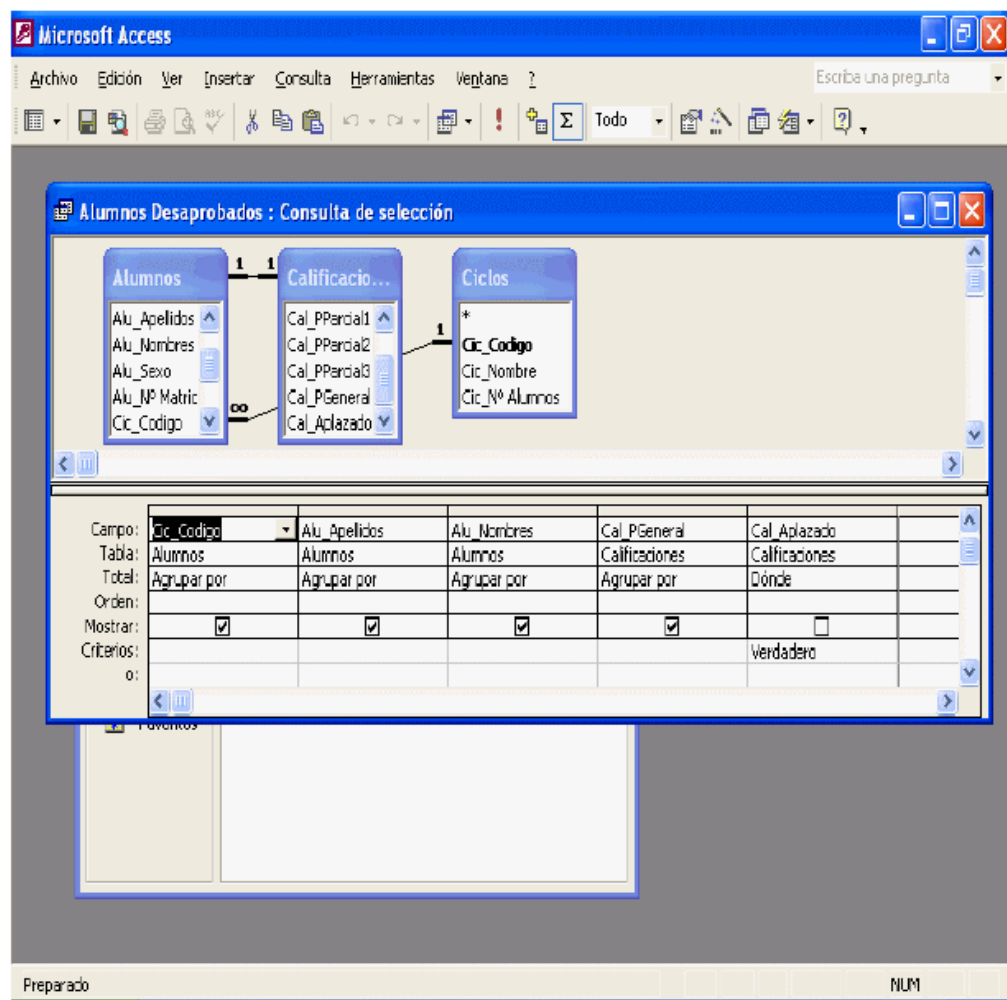
En la cuadrícula QBE se indicarán los siguientes argumentos:



Dar clic en el botón ejecutar, para ejecutar y visualizar la consulta.
Guardar la consulta con el nombre de Alumnos Becados

- Mostrar el código del ciclo, los apellidos y nombres de los alumnos y la calificación del promedio general para todos los alumnos desaprobados. Desarrollo:
 - Seleccionar el objeto de consulta
 - Dar clic en el botón Diseño
 - En la ventana mostrar tabla seleccionar la tabla Alumnos, Calificaciones y Ciclos; luego dar clic en el botón agregar para finalizar clic en el botón cerrar.

En la cuadrícula QBE se indicarán los siguientes argumentos:



Dar clic en el botón ejecutar , para ejecutar y visualizar la consulta.

Guardar la consulta con el nombre de Alumnos Desaprobados

Proceso de Comprensión y Análisis

- Definir y dar ejemplos de Números Exactos, Números Aproximados, Cadena de Caracteres, Cadena de Bits, Fecha y Hora, Intervalos.
- ¿A qué hace referencia la manipulación de datos?
- Definir cada una de las siguientes cláusulas:
 - SELECT
 - FROM
 - WHERE
 - ORDER BY
- ¿En qué situaciones es necesario el uso de operadores de comparación?. ¿Por qué se les conoce como conectores booleanos?
- ¿Cómo se define la estructura de una tabla?

Solución de Problemas

Desarrollar los ejercicios que se plantean a continuación:

- Un distribuidor de productos a supermercados realiza el siguiente informe:

PEDIDO	FECHA	ESTADO	PRODUCTOS	CLIENTE	DIRECCIÓN DE ENVIO	TELEF. CLIENTE	ENCARGADO DE DESPACHAR PEDIDO
2050	20/05/02	PENDIENTE	ARROZ LECHE JABÓN	SOBERANO	CALLE 9 N° 22-88	5689999	Luis Peña
2051	28/06/02	ENVIADO	SAL CAFÉ LECHE	DAY	CARRERA 5 N° 99-77	5686655	Lola Cardenas
2060	30/07/02	CANCELADO	FRIJOL JABÓN	LA CITA	DIAGONAL 21 N° 44-66	5685471	Lola Cardenas
2070	01/09/02	PENDIENTE	SAL ARROZ	SOBERANO	CALLE 9 N° 22-88	5689999	Lola Cardenas
2080	05/10/02	ENVIADO	CAFÉ FRIJOL	LA CITA	DIAGONAL 21 N° 44-66	5685471	Luis Peña

- Tomar el anterior informe como base para definir las tablas.
 - Elaborar el diagrama entidad-relación.
 - Elaborar los formularios necesarios para llenar datos.
 - Los datos mostrados deben ser digitados.
- En una ciudad existen varias instituciones sin ánimo de lucro que se dedican a ayudar a los más necesitados (damas rosadas, Club Rotario, Sociedad San Vicente de Paul, etc). Estas instituciones se han puesto de acuerdo para organizar su información y ser más eficientes y han decidido contratarlo a usted para que los ayude. Después de analizar la información usted ha sacado las siguientes conclusiones.

Todas las instituciones realizan la misma labor pero en sectores diferentes de la ciudad. En un sector solo actúa una institución y a una institución varios sectores. Cada institución tiene socios que no devengan ningún salario y que solo pueden pertenecer a una institución. Cada institución tiene sus propios empleados (secretaria, mensajero, aseador) a los que se les paga sin distinción el salario mínimo. Cada institución lleva a cabo varias actividades para recolectar dinero (rifas, bazares, bingos). Cada actividad tiene un costo y una ganancia. Cada socio dentro de la institución es responsable de una actividad. Los empleados no participan de las actividades. Los socios dan aportes fijos mensuales a la institución. Cada institución también cuenta con colaboradores en la ciudad que aportan una cuota fija mensual pero que no son socios y que no son responsables de actividades. Pueden ser colaboradores de varias instituciones a la vez. Las personas necesitadas de ayuda llena una solicitud con sus datos personales. Cada persona debe pertenecer a un solo sector. Cada institución ofrece ayudas o servicios a los necesitados (mercados, becas, pago de fórmulas médicas, vestuario, etc). Cada ayuda o servicio tiene un costo. Una persona puede recibir más de una ayuda.

- Definir tablas
- Elaborar un diagrama de entidad-relación
- Elaborar los formularios necesarios para llenar los datos.

Síntesis Creativa y Argumentativa

- ¿Qué importancia tienen las bases de datos en las organizaciones?
- Empleando un buscador en la Web, recopilar información sobre los Motores de Bases de Datos. ¿Por qué razón cree usted que en este modulo sólo nos referimos a Access.
- ¿Qué significa SQL?

Autoevaluación

- ¿Qué es una base de datos?
- ¿Qué es una tabla? ¿Cómo se definen las tablas?
- ¿Qué es un modelo relacional?
- ¿Qué es una restricción?
- ¿Qué es valor por defecto?
- ¿Qué se entiende por definición de dominio?
- ¿Qué es una consulta simple?
- ¿Qué es una consulta multi-tablas?
- ¿Qué es una entidad?
- ¿Qué es una relación?
- ¿Qué es un formulario?
- ¿Qué es un informe?
- ¿Qué es una Macro?
- ¿Qué es un modulo?

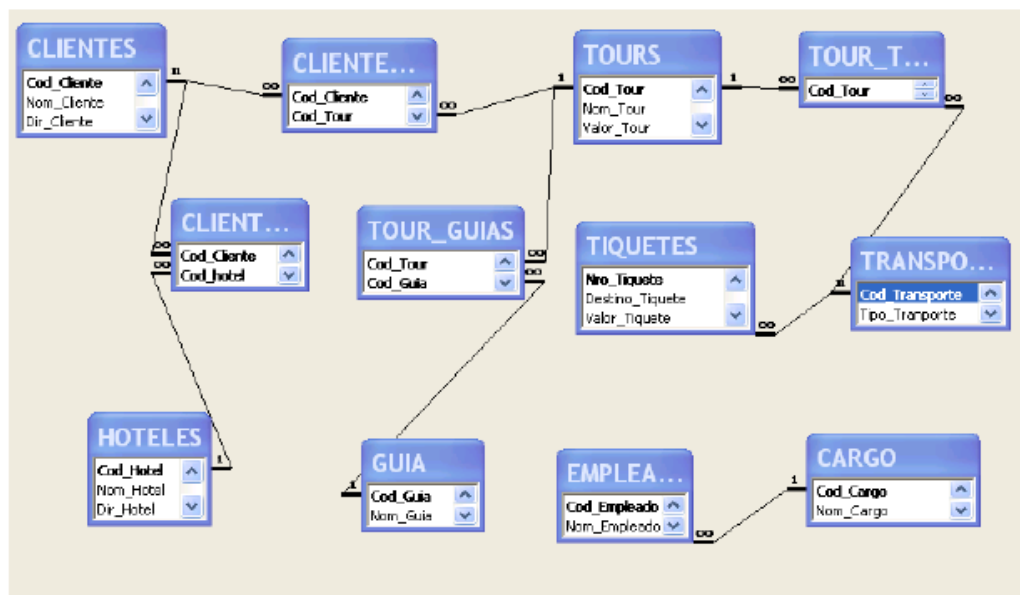
Repaso Significativo

- Teniendo en cuenta lo estudiado en la unidad, desarrollar los ítems que se plantean al final.

ALMACÉN DE PRODUCTOS AGRÍCOLAS

Un almacén de productos agrícolas vende los productos a sus clientes a través de pedidos. Al vender, el almacén emite facturas a sus clientes. Existen diferentes tipos de descuentos a las facturas (por pronto pago, por mes de promoción, por amistad, etc.). A una factura solo se le puede hacer un tipo de descuento. A los empleados del almacén se les paga un salario básico de acuerdo a su categoría (contador, servicios generales, mensajero, etc.) pero también se les pagan las horas extras trabajadas (horas extras diurnas normales, horas extras nocturnas normales, horas extras diurnos festivos, horas extras nocturnos festivos). Cada tipo de hora extra tiene un valor diferente. Los productos del almacén son suministrados por proveedores que no tienen ningún tipo de exclusividad. Los pedidos son recibidos por un empleado.

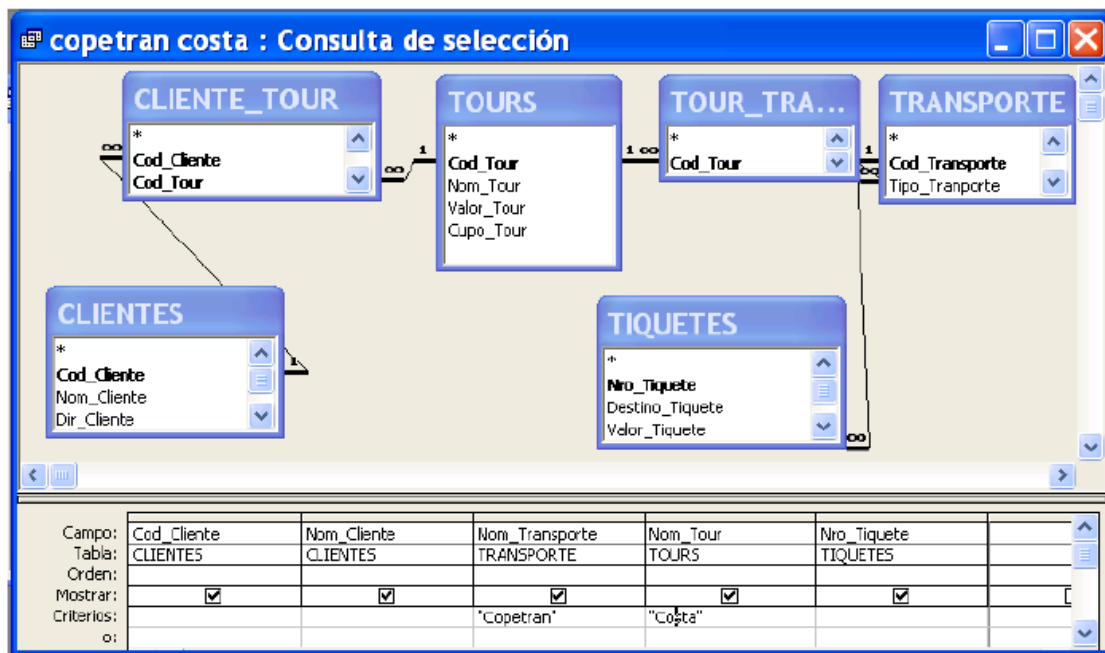
- Elaborar un diagrama entidad relación listo para crear tablas y relaciones en ACCES (es decir con las tablas unión y campos que se copian).
 - Crear tablas y relaciones en ACCESS.
 - Elaborar los formularios necesarios para llenar los datos.
 - Para este enunciado no considere al almacén como una "tabla".
- Siguiendo los pasos descritos anteriormente para el manejo del Access, el modelo entidad relación quedaría así:



- Se debe crear los formularios para el llenado de datos de cada una de las tablas.

Un ejemplo de una consultas sería: ¿Cuáles son los clientes que viajan en la empresa de transportes COPETRAN para la costa?

Observemos como quedaría el diseño de la consulta.



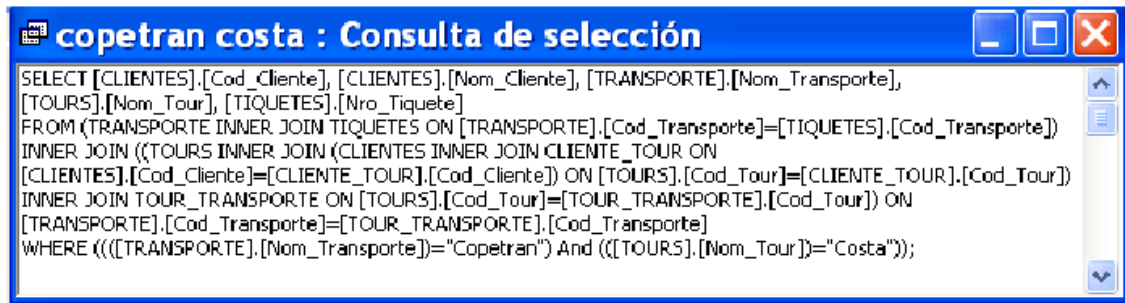
El resultado de la consulta se mostraría así:

El resultado de la consulta se muestra en una tabla con los siguientes datos:

	Cod_Cliente	Nom_Cliente	Nom_Transporte	Nom_Tour	Nro_Tiquete
▶	1001	Carlos Martinez	Copetran	Costa	5002

En la parte inferior, se indica: Registro: 1 de 1.

Si se quiere ver la consulta en SQL, ACCESS la muestra así:



- Cree los informes según la consulta que considere.

Bibliografía Sugerida

C. J, Date. Introducción a los Sistemas de Bases de Datos. Volumen I. Quinta Edición. E.U.A. Adisson – Wesley Iberoamericana. 1990.

GARY W, Hansen. Y JAMES V. Hansen. Diseño y Administración de Bases de Datos. Segunda Edición. España. Prentice Hall Inc. 1996.

BIBLIOGRAFÍA GENERAL

C. J, Date. Introducción a los Sistemas de Bases de Datos. Volumen I. Quinta Edición. E.U.A. Adisson – Wesley Iberoamericana. 1990.

COHEN, Daniel y ASÍN, Enrique. Sistemas de Información para los Negocios. Un Enfoque de Toma de Decisiones. Tercera Edición. México. Mc-Graw Hill. 2001

GARY W, Hansen. Y JAMES V. Hansen. Diseño y Administración de Bases de Datos. Segunda Edición. España. Prentice Hall Inc. 1996.

OZ, Effy. Administración de Sistemas de Información. Segunda Edición. México. Thomson Editores. 2002

VILLAREAL, Sonia. Introducción a la Computación. Guía Práctica para el Aprendizaje de Paquetes. México. Mc-Graw Hill. 2000