

DISEÑO DE BASE DE DATOS RELACIONALES



CEPSUNI – 2021

LIMA - PERU

DISEÑO DE BASE DE DATOS RELACIONALES

Derechos Reservados © 2021 CEPS-UNI

Primera Edición

LIMA - PERÚ

Presentación

En el desarrollo de las aplicaciones que sirven para manipular las bases de datos nos encontramos con bases de datos que ya han sido creadas en algún DBMS en particular, luego el programador debe conocer su estructura y las reglas que juegan en dichas bases de datos para a continuación según las necesidades del usuario diseñar la aplicación más adecuada.

El conocimiento de cómo se llegó a diseñar la base de datos se hace importante porque nos permitirá conocer los criterios que se tuvieron en cuenta para llegar a dicho modelo, esto hará que nuestra aplicación tenga mayor consistencia y más aún se puedan detectar algunos errores de diseño que tal vez no se tomaron en cuenta el proceso del modelamiento.

Este curso, tiene como objetivo principal entonces adquirir las técnicas adecuadas para diseñar e implementar, a partir de una realidad concreta, una base de datos en un servidor determinado y con el DBMS (SQL Server).

Es importante añadir, que tan importante como conocer las técnicas adecuadas y utilizar el lenguaje y simbología formal que se usa en todo diseño de Ingeniería, también es igual de importante conocer las Reglas de Negocio de la realidad en la que se va a implementar el diseño. Estas llamadas reglas de negocio se deben encontrar en la documentación que se encuentra en esta realidad, pero en muchas ocasiones estas reglas no están claras o no se especifican totalmente; El diseñador debe por lo tanto investigar estas reglas no solo con la documentación, sino entrevistando a las personas que interactúan con la realidad. Para ello el diseñador debe usar un lenguaje adecuado con las personas que intervienen en dicha realidad, ya que realizar preguntas en un lenguaje netamente técnico les llevara a confusiones que distorsionaran al final el modelo a desarrollar. Para ello el lenguaje de comunicación con los usuarios debe ser claro y usando la terminología que ellos puedan manejar o entender.

Julio Enrique Flores Manco
INSTRUCTOR

Índice

CAPÍTULO 1 SISTEMAS DE INFORMACION.....	7
DISEÑO DE BASE DE DATOS Y FASES DE DISEÑO	7
SISTEMAS DE INFORMACIÓN	9
DATOS E INFORMACIÓN	9
<i>Datos</i>	9
<i>Información</i>	9
LA INFORMACIÓN Y LA TOMA DE DECISIONES.....	10
CARACTERÍSTICAS DE LA INFORMACIÓN	10
COMPONENTES DE UN SISTEMA DE INFORMACIÓN.....	11
TIPOS DE SISTEMAS DE INFORMACIÓN	12
<i>S.I. Transaccionales (OLTP)</i>	12
<i>Sistemas para soporte de decisiones (DSS)</i>	12
<i>Sistemas Estadísticos o Científicos</i>	12
<i>Sistemas Mixtos</i>	13
<i>Sistemas de información empresarial</i>	13
BASE DE DATOS	14
<i>Necesidad de una Base de Datos</i>	14
<i>Beneficios de una base de datos:</i>	15
<i>El Diseño de la Base de Datos</i>	16
<i>Ciclo de Vida del Diseño de Base de Datos</i>	17
CAPÍTULO 2 DISEÑO CONCEPTUAL.....	18
MODELO DE DATOS	20
ENTIDAD.....	22
<i>Concepto de Entidad</i>	22
<i>Requisitos Para La Existencia De Una Entidad</i>	22
INSTANCIAS Y ATRIBUTOS.....	23
ATRIBUTO IDENTIFICADOR.	24
REPRESENTACION DE LAS ENTIDADES Y SUS ATRIBUTOS.....	24
CAPÍTULO 3 RELACIONES.....	26
DESCUBRIR LAS RELACIONES MAS RELEVANTES EN UNA REALIDAD.....	26
CARDINALIDAD DE UNA RELACIÓN	27
REPRESENTACION DE LAS REGLAS NEGOCIO EN LAS CARDINALIDES DE UNA RELACION.....	28
EL MC Y LAS REGLAS DE NEGOCIO	29
VERIFICACION DE LAS POSIBLES CARDINALIDADES DE UNA RELACION	31
CAPÍTULO 4 DISEÑO LÓGICO	32
TECNOLOGÍAS PRINCIPALES.....	32
LA TECNOLOGÍA DEL MODELO RELACIONAL.....	32

TABLA, FILA, REGISTRO O TUPLA	33
<i>Tabla</i>	33
COLUMNAS (ATRIBUTOS). DOMINIO	33
<i>Columna, campo o atributo</i>	33
<i>Dominio</i>	33
CONCEPTO DE CLAVE PRIMARIA PK (PRIMARY KEY)	34
<i>Criterios para establecer la PK</i>	34
CONCEPTO DE CLAVE FORÁNEA FK (FOREING KEY)	35
CAPÍTULO 5 MODELO LÓGICO RELACIONAL	36
TRANSFORMACIÓN DEL MODELO CONCEPTUAL EN EL MODELO LÓGICO CORRESPONDIENTE.	36
REGLAS DE TRANSFORMACIÓN SEGÚN LAS CARDINALIDADES DEL MODELO CONCEPTUAL	36
RELACIÓN NO IDENTIFICADORA	37
RELACIÓN IDENTIFICADORA	38
ENTIDAD DEPENDIENTE	38
ENTIDAD INDEPENDIENTE	39
CAPÍTULO 6 RELACIONES ADICIONALES	40
RELACIONES MULTIPLES ENTRE DOS ENTIDADES	40
RELACIONES RECURSIVAS	43
RELACIONES TERNARIAS	44
RELACIONES DE SUPERTIPOS Y SUBTIPOS	45
CAPÍTULO 7 NORMALIZACIÓN.....	48
MODELO RELACIONAL DE CODD	48
PRIMERA FORMA NORMAL (1FN)	49
SEGUNDA FORMA NORMAL (2FN)	54
TERCERA FORMA NORMAL (3FN)	59
ANOMALIAS ADICIONALES. ANOMALÍA DE LA DEPENDENCIA CONDICIONAL (VALOR NULL)	63
CAPÍTULO 8 DISEÑO FÍSICO	65
DBMS, DATA BASE MANAGEMENT SYSTEM	65
DBMS'S MÁS COMERCIALES.....	67
USO DE SQLSERVER Y CREACIÓN DE UNA BASE DE DATOS.	69
<i>Creación de una Base de datos</i>	69
CREACIÓN DE UN ESPACIO DE BASE DE DATOS	71
TIPOS DE DATOS MÁS USADOS EN SQL SERVER	73
<i>Char(n)</i>	73
<i>Varchar(N)</i>	73
<i>Integer</i>	73
<i>float</i>	73
<i>Tinyint</i>	73
<i>Decimal (p, s)</i>	74

<i>Money:</i>	74
<i>Datetime:</i>	74
USO DE LA INTERFAZ GRÁFICA DEL SQL SERVER PARA LA CREACIÓN DE UNA BASE DE DATOS.	75
<i>Creación de una Base de datos</i>	75
CREACIÓN DE LA BASE DE DATOS CORRESPONDIENTE A ESTE MODELO FÍSICO.	76
<i>Ejercicio-de- Practica:</i>	96
LABORATORIOS	¡ERROR! MARCADOR NO DEFINIDO.
LABORATORIO 1	97
<i>Proyecto 1 – Clínica “Santo Tomas”</i>	97
<i>Proyecto 2 – Envío de Paquetes</i>	98
<i>Proyecto 3 – Alquiler de Películas</i>	99
<i>Proyecto 4 – Empresa de Transporte de pasajeros</i>	100
<i>Proyecto 5 – Venta de automóviles</i>	101
<i>Proyecto 6 – Venta de Flores</i>	102
<i>Proyecto 7 – Biblioteca</i>	104
LABORATORIO 2	105
<i>Proyecto1 – Alquiler de coches</i>	105
<i>Proyecto 2 – Compañía-Departamentos-Proyectos</i>	107
<i>Proyecto 3 – Agencia Aeroespacial</i>	109
<i>Proyecto 4 – Casos de corrupción</i>	111
<i>Proyecto 5 - Joyería</i>	112
<i>Proyecto 6 – Liga de Futbol</i>	114
LABORATORIO 3	115
<i>Proyecto 1 de Normalización. - Ficha de Matricula</i>	115
<i>Proyecto 2 – Biblioteca Municipal.</i>	117
<i>Proyecto 3 – Tienda de artículos deportivos</i>	118
<i>Proyecto 4 – Lavandería</i>	119
<i>Proyecto 5 – Aeropuertos-Vuelos</i>	121
<i>Proyecto 6 - Gastos Por Cobrar</i>	124
<i>Proyecto 7 – Ventas Vendedores por zonas.</i>	127
LABORATORIO 4	128
<i>Proyecto 1 - Sistema De Líneas Telefónicas</i>	128
<i>Proyecto 2- Planilla De Empleados</i>	130
<i>Proyecto 3 - Fabricación De Insumos De Oficina</i>	132
CURSOS RELACIONADOS	137

Capítulo 1

SISTEMAS DE INFORMACION

DISEÑO DE BASE DE DATOS Y FASES DE DISEÑO



Las bases de datos son exactamente lo que dice su nombre, un almacén de información que se administra mediante un motor o gestor de base de datos y un lenguaje de base de datos.

En el desarrollo de las Aplicaciones que sirven para manipular las Bases de datos nos encontramos con Bases de Datos que ya han sido creadas en algún DBMS en particular, luego el programador debe conocer su estructura y las reglas que juegan en dichas B.D. para a continuación según las necesidades del usuario diseñar la aplicación más adecuada.

El conocimiento de cómo se llegó a diseñar la B.D. se hace importante porque nos permitirá conocer los criterios que se tuvieron en cuenta para llegar a dicho modelo, esto hará que nuestra aplicación tenga mayor consistencia y más aún se puedan detectar algunos errores de diseño que tal vez no se tomaron en cuenta el proceso del modelamiento.

Este curso, tiene como objetivo principal entonces adquirir las técnicas adecuadas para diseñar e implementar, a partir de una realidad concreta, una base de datos en un servidor determinado y con el DBMS (SQL Server).

Es importante añadir, que tan importante como conocer las técnicas adecuadas y utilizar el lenguaje y simbología formal que se usa en todo diseño de Ingeniería, también es igual de importante conocer las Reglas de Negocio de la realidad en la que se va a implementar el diseño. Estas llamadas reglas de negocio se deben encontrar

en la documentación que se encuentra en esta realidad, pero en muchas ocasiones estas reglas no están claras o no se especifican totalmente; El diseñador debe por lo tanto investigar estas reglas no solo con la documentación, sino entrevistando a las personas que interactúan con la realidad. Para ello el diseñador debe usar un lenguaje adecuado con las personas que intervienen en dicha realidad, ya que realizar preguntas en un lenguaje netamente técnico les llevara a confusiones que distorsionaran al final el modelo a desarrollar. Para ello el lenguaje de comunicación con los usuarios debe ser claro y usando la terminología que ellos puedan manejar o entender.

Fases del diseño

- **Diseño conceptual:**

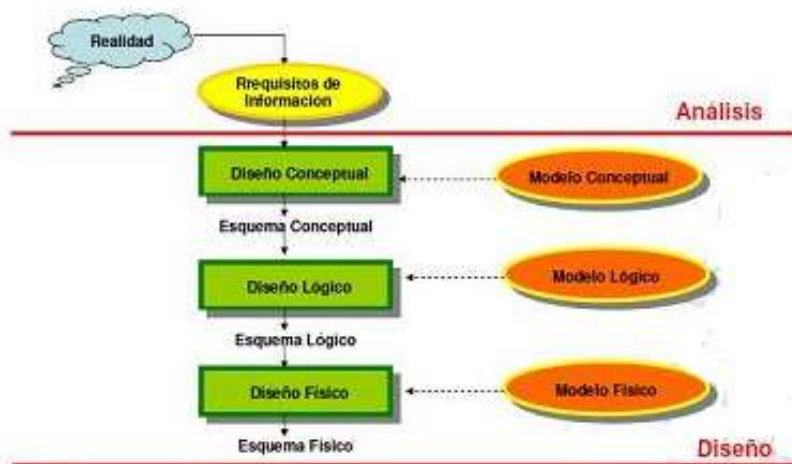
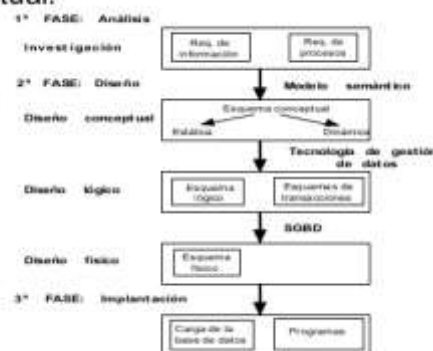
- Esquema conceptual.

- **Diseño lógico:**

- Esquema lógico.

- **Diseño físico:**

- Esquema físico.



SISTEMAS DE INFORMACIÓN

Un Sistema de Información, es aquél que permite recopilar, administrar y manipular un conjunto de datos que conforman la información necesaria para que los estamentos ejecutivos de una organización puedan realizar una toma de decisiones de la forma más adecuada. Es aquel conjunto ordenado de elementos (no necesariamente computacionales) que permiten manipular toda aquella información necesaria para implementar aspectos específicos de la toma de decisiones de elementos que interactúan para realizar un conjunto de funciones específicas.



DATOS E INFORMACIÓN

Datos

Representación formal de hechos, conceptos o instrucciones adecuada para su comunicación, interpretación y procesamiento por seres humanos o medios automáticos.

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

Un Dato es un valor específico, por ejemplo "15", el cual, por sí sólo, no entrega ningún conocimiento útil.

Información

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

LA INFORMACIÓN Y LA TOMA DE DECISIONES

Las funciones administrativas, así como el arte de la toma de decisiones tienen como materia prima fundamental la información. El grado de incertidumbre en la toma de decisiones depende de la información que se posea. En consecuencia, la información otorga ventajas estratégicas a la organización en el mercado frente a los competidores.

CARACTERÍSTICAS DE LA INFORMACIÓN

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

Calidad

Cuanto más exacta sea la información mejor será su calidad y los usuarios recurrirán a ella con más confianza cuando tomen una decisión.

Oportunidad

La información debe estar siempre disponible, para la acción correcta y la toma de decisiones en el momento que se le necesite.

Cantidad

Los administradores casi nunca toman decisiones correctas y acertadas si la información no es acertada, oportuna y suficiente; la información debe ser la suficiente para cubrir el espectro del problema en la toma de decisiones.

Relevancia

Debe darse solo la información relevante para la toma de decisiones; toda la información proporcionada debe ser de utilidad; los datos no deben ser de relleno.

COMPONENTES DE UN SISTEMA DE INFORMACIÓN

Un Sistema de información es un sistema, automatizado o manual, que engloba a personas, máquinas y/o métodos organizados para recopilar, procesar, transmitir datos que representan información.

Implica la infraestructura, organización, personal y componentes para la recopilación, procesamiento, almacenamiento, transmisión, visualización, diseminación y organización de información.

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

Lo que se exige al S.I.

- ✓ Obtención de ventajas estratégicas para la organización.
- ✓ Lugar correcto, tiempo correcto y forma correcta.
- ✓ Reducción de costos.
- ✓ Proveer funciones de automatización para tareas críticas.
- ✓ Soporte a un número creciente de usuarios.
- ✓ Nuevas aplicaciones en función a aplicaciones existentes.
- ✓ Proveer soporte para periodos largos de tiempo.

TIPOS DE SISTEMAS DE INFORMACIÓN

S.I. Transaccionales (OLTP)

(On Line Transaction Process) Están orientados a cubrir la operatividad de la Organización, cumplir con cada uno de los procesos que la empresa realiza como objetivo de negocio. Por ejemplo, el Sistema de ventas.

Características:

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

Sistemas para soporte de decisiones (DSS)

(Decision Support System) Deben cubrir la demanda de información para la toma de decisiones estos sistemas se orientan al nivel estratégico de la organización y al táctico en un menor grado;

Características:

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

Sistemas Estadísticos o Científicos

Apoyan a los procesos de investigación y explotación de sistemas complejos

Características:

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

Sistemas Mixtos

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

Sistemas de información empresarial

Los sistemas de información empresarial constituyen el conjunto de recursos de la empresa que sirven como soporte para el proceso básico de captación, transformación y comunicación de la información.

Un sistema de información debe ser eficaz y eficiente. Es eficaz si facilita la información necesaria, y es eficiente si lo realiza con los menores recursos posibles.

Factores que favorecen el desarrollo de sistemas de información:

- ✓ Complejidad de los sistemas actuales (p.ej. competencia).
- ✓ Capacidad de los ordenadores.

Ventajas

- ✓ Económicas (ahorro de costes, aumento de productividad).
- ✓ No económicas (valor percibido).

BASE DE DATOS

Una base de datos es un conjunto de datos organizados de manera tal que se facilita el almacenamiento, la actualización, y la consulta de los datos. Una base de datos relacional está organizada como un conjunto de filas y columnas relacionadas.

Es una colección sistematizada de datos, que puede ser usada como fuente de información única para toda la organización.

Colección organizada y sistematizada de datos de interés de la organización a la cual accedemos y explotamos para obtener información de utilidad para el manejo de la organización.



El **objetivo primario** de una base de datos es almacenar datos de interés para la organización. Además de:

- ✓ Proveer como fuente única de información a toda la organización.
- ✓ Mantener seguridad y privacidad de la información.
- ✓ Tener a la información como activo de la organización

Necesidad de una Base de Datos

El desarrollo de Sistemas de Información trae consigo una nueva necesidad de tener la información almacenada en fuentes de datos uniformes para un acceso a la base de datos uniformes y homogéneos en lo posible.

Por otra parte, la concepción de la información como un activo de gran valor para la organización hace que nos apoyemos en ésta para salvaguardarla y explotarla al máximo.

Beneficios de una base de datos:

- ✓ Reducción de la redundancia de datos.
- ✓ Se evita la inconsistencia en los datos.
- ✓ Los datos pueden ser compartidos por las áreas relacionadas de la organización.
- ✓ Permite la estandarización de los datos.
- ✓ Permite aplicar restricciones de seguridad.
- ✓ Permite controlar la integridad de los datos.

El planeamiento de la base de datos ofrece las siguientes ventajas:

- ✓ Permite entender la información con la que cuenta la organización como un recurso valioso para sus actividades diarias.
- ✓ Permite establecer los requerimientos de información de los usuarios, y la manera de satisfacerlos.
- ✓ Permite administrar eficientemente los recursos de información.
- ✓ Permite establecer planes de acción para lograr los objetivos de la organización.
- ✓ Ayuda a desarrollar una base de datos que permita cumplir con los requerimientos de la organización

El Diseño de la Base de Datos

Una base de datos correctamente diseñada permite obtener acceso a información exacta y actualizada. Puesto que un diseño correcto es esencial para lograr los objetivos fijados para la base de datos, parece lógico emplear el tiempo que sea necesario en aprender los principios de un buen diseño ya que, en ese caso, es mucho más probable que la base de datos termine adaptándose a sus necesidades y pueda modificarse fácilmente.

El Diseño de una Base de datos, también conocido como Modelamiento de datos, el proceso de abstracción de la realidad (modelo) en función de los conceptos (datos que la componen)

Se realiza de acuerdo a la interpretación de la realidad que se está observando

Es descrito en función de la interpretación de los conceptos (semántica)

Debe ser consistente con el modelamiento de procesos de negocio

Características que debe cumplir un Modelo Optimo

Necesidad de Modelar los Datos

Identificar los requerimientos del usuario

Incrementar la productividad

Mantener la consistencia de los datos

Optimizar la performance

Permitir el diseño de información integrada

Ciclo de Vida del Diseño de Base de Datos



Capítulo 2

DISEÑO CONCEPTUAL

El diseño conceptual permite describir, de un modo totalmente independiente de la implementación, los datos que el usuario quiere recoger en el sistema. Dependiendo de la cantidad de información que se desee representar, tendremos aplicaciones más o menos orientadas a los datos. Así, por ejemplo, la gestión de una biblioteca es una aplicación pura de Bases de Datos (en adelante BD) ya que prácticamente toda la funcionalidad del sistema se centra en el mantenimiento de los datos (introducir un libro, prestar un libro, etc.).

Existen, sin embargo, otras aplicaciones, como por ejemplo un sistema de control de navegación aérea, en las que los datos son algo secundario. Podemos decir que, en general, los datos son el núcleo de todo SI orientado a la gestión.

El desarrollo estructurado de software, a diferencia de lo que ocurre en el desarrollo orientado a objetos, mantiene una clara separación entre los datos y las funciones del sistema. Por ello, es necesario disponer, en cada una de las etapas del proceso de desarrollo, de técnicas específicas para la especificación de los datos, que serán diferentes de las técnicas orientadas a la especificación de las funciones o procesos.

El modelado conceptual es una actividad que se realiza en la etapa de análisis, paralelamente al modelado de procesos o funciones. Su objetivo, como ya hemos dicho, es captar toda la información del mundo real que se desea representar en el mundo informático. En este proceso es importante abstraer los detalles sin importancia y representar tan sólo aquella información que sea relevante.

En este punto no nos interesa el cómo ni donde se va a implementar el sistema. De hecho, dependiendo del tipo de sistema (más o menos orientado a los datos), del volumen de información, de los requisitos de eficiencia, etc. se podrán utilizar distintos mecanismos de persistencia de los datos: Sistemas de Bases de Datos, Sistemas de Ficheros, etc.

En esta etapa interesa recoger la máxima cantidad de información posible, por lo necesitamos una técnica que cumpla los siguientes requisitos:

- ✓ Ser independiente de los modelos o lenguajes de implementación
- ✓ Tener una capacidad semántica alta
- ✓ Ser lo más cercana posible al usuario

Aunque existen diversas técnicas, utilizaremos el modelo E/R porque además de cumplir los requisitos anteriores es la técnica de modelado conceptual universalmente aceptada para el desarrollo estructurado.

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976. El modelo Entidad-Relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Originalmente, el modelo entidad-relación sólo incluía tres clases de objetos: entidades, interrelaciones y atributos. Más tarde, se añadieron al modelo básico un conjunto de constructores que ayudaron a recoger una mayor semántica del universo de discurso (los atributos compuestos, cardinalidades máximas y mínimas, las jerarquías de generalización, dependencia en existencia y en identificación, etc.) y este nuevo modelo es lo que se ha denominado modelo E/R extendido.

El Modelo E/R ha tenido una gran difusión en la comunidad informática dedicada a las BD, prueba de ello es que ha sido el modelo más extendido en las herramientas CASE de ayuda al diseño de BD.

MODELO DE DATOS

Un modelo puede definirse como la construcción mental a partir de la realidad en la que se reproducen los principales componentes y relaciones del segmento de la realidad analizada. Éste es, efectivamente, el significado de modelo en las ciencias empíricas en las que, a fin de estudiar el comportamiento de una determinada parcela de la realidad, se crea un modelo de ésta. Dicho modelo habrá de ser isomórfico respecto a la realidad que representa y más simple que ésta, destacando sus principales características, o aquellas que son relevantes para un determinado interés de estudio. Sin embargo, ésta no es la única acepción del término modelo, al que podemos asignar dos significados:

- ✓ Por un lado, el modelo entendido como una reproducción simplificada de la realidad; este es el caso, ya expuesto, de las ciencias empíricas, en las que se definen modelos de comportamiento simplificados de la parcela del mundo real que es objeto de estudio;
- ✓ Y, por otro lado, el modelo entendido como la realidad propiamente dicha; piénsese, por ejemplo, en un pintor, quien reproduce en lienzo a sus modelos; en este segundo caso, el modelo no es la representación del mundo real, sino que constituye el mundo real en sí mismo: es un modelo a copiar o a simular.

Esta segunda acepción de modelo es la que corresponde a la lógica matemática, donde la representación recibe el nombre de teoría y lo representado el de modelo, mientras que la primera corresponde, como ya hemos visto, al concepto de modelo en las ciencias empíricas.

En algunos casos, como en el caso de los modelos de datos, el concepto de modelo

Responde simultáneamente a estas dos acepciones. Quizá resulte esclarecedor el ejemplo del arquitecto, para quien una maqueta es un modelo a copiar para la construcción de un nuevo edificio. La maqueta sería, en un primer momento, la realidad puesto que el edificio aún no existe. El arquitecto obtiene un nuevo edificio tomando dicha maqueta como modelo. A partir de este momento, el edificio constituye la realidad, mientras que la maqueta puede considerarse una copia del mismo.

En el ámbito de las bases de datos es muy común la utilización del término modelo de datos, y existen diferentes definiciones del mismo en la literatura. Así, por ejemplo, Dittrich define modelo de datos como "un conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con: estructuras y tipos de datos, operaciones y restricciones".

Un modelo de datos permite representar una parcela de información del mundo real de especial interés, lo que habitualmente se denomina universo del discurso o, en términos, de Dittrich mini-mundo. La representación del universo del discurso se concibe en dos niveles: el de la información en sí misma y el de las estructuras que hacen posible la representación de tal información. Estos dos niveles dan lugar, en el ámbito de las bases de datos, a la distinción entre esquema y base de datos, conceptos que Dittrich define como sigue: "La descripción específica de un determinado mini-mundo en términos de un modelo de datos se denomina esquema (o esquema de datos) del mini-mundo. La colección de datos que representan la información acerca del mini-mundo constituya la base de datos".

Teniendo presente que esta terminología puede variar y que empleamos ésta por ser la más extendida en el mundo de los "datos" en el que se centra la presente unidad, definimos modelo de datos como un:

Conjunto de conceptos, reglas y convenciones que permiten describir y manipular los datos de la parcela del mundo real que constituye nuestro universo del discurso

ENTIDAD

En el Modelo Entidad – Relación (E/R), tal como fue propuesto por Chen, se distinguen los siguientes elementos:

- Entidad
- Interrelación
- Dominio
- Atributo.

Concepto de Entidad

El mundo real se compone de una serie de objetos (reales o abstractos) acerca de los cuales queremos obtener y representar información. En general dichos objetos podrán agruparse en conjuntos de acuerdo a unas características comunes a todos aquellos que forman parte de un mismo conjunto. Denominamos entidad a la abstracción que permite representar aquellos objetos del mundo real que comparten una serie de características comunes.

Las entidades son los objetos principales sobre los que debe recogerse información y generalmente denotan personas, lugares, cosas o eventos de interés. Las entidades aparecerán reflejadas en el enunciado habitualmente como nombres. A cada una de las posibles ocurrencias (cada persona, lugar, cosa o evento concreto) de la entidad se denomina Instancia o ejemplar.

Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Requisitos Para La Existencia De Una Entidad

Uno de los problemas que existirán en el diseño E/R es la decisión de si un determinado objeto o concepto se modela como un tipo de entidad o no. Por ejemplo, el color es habitualmente una propiedad de una entidad (como es el caso del color de un coche), pero en una fábrica de pinturas probablemente sería apropiado modelar el color como una entidad con sus propias propiedades.

Por esta razón, algunos autores han intentado precisar el concepto de entidad. Así, TARDIEU (1979) proponen tres reglas generales que debe cumplir una entidad:

- ✓ tiene que tener existencia propia
- ✓ cada instancia de un tipo de entidad debe poder distinguirse de las demás
- ✓ todos los ejemplares de un tipo de entidad deben tener las mismas propiedades.

En conclusión. Una Entidad para existir como tal, debe estar compuesta por varios elementos (Instancias) y Además Cada elemento debe tener varias propiedades (atributos)

INSTANCIAS Y ATRIBUTOS

Entonces una Instancia es un caso particular de todos los elementos que conforman una Entidad.

Atributos

Los atributos se utilizan para detallar las entidades asignándoles propiedades descriptivas tales como nombre, color y peso. Los atributos son cada una de las propiedades o características que tiene un tipo de entidad o un tipo de interrelación.

La existencia de un atributo está ligada a la del correspondiente tipo (de entidad o de interrelación).

Existen dos tipos de atributos: identificadores y descriptores. Los primeros se utilizan para distinguir de manera única cada una de las ocurrencias de una entidad (distinguiéndose entre identificadores principales e identificadores alternativos), mientras que los descriptores se utilizan para describir una ocurrencia de entidad. No sólo es posible especificar atributos en las entidades sino también en las interrelaciones (en este caso sólo tiene sentido hablar de atributos descriptores y no de identificadores)

ATRIBUTO IDENTIFICADOR.

Entre todos los atributos de un tipo de entidad han de existir uno o varios conjuntos de atributos (simples y/o compuestos) que identifiquen unívocamente cada una de las instancias de ese tipo de entidad.

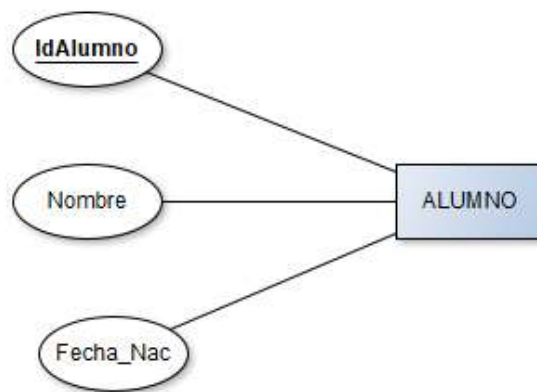
Cada uno de estos conjuntos de atributos se denomina Identificador Candidato (IC). Todo IC debe cumplir la condición de ser unívoco y mínimo (Criterio de Unicidad y Criterio de Minimidad).

Cuando un IC es compuesto, el número de los atributos que lo componen debe ser mínimo, en el sentido de que la eliminación de cualquiera de ellos le haría perder su carácter de identificador.

Entre los IC se elige uno como Identificador Principal (IP) y el resto serán Identificadores Alternativos (IA).

REPRESENTACION DE LAS ENTIDADES Y SUS ATRIBUTOS

En la notación los atributos de una entidad se representan gráficamente así:



El atributo identificador o atributo clave se muestra subrayado. En el Modelo Conceptual se deben considerar solo los atributos más relevantes.

Atributos Opcionales vs Obligatorios:

Opcionales: Por otro lado, puede obligarse a un atributo de un tipo de entidad a que tome, como mínimo, un valor del (o de los) dominio(s) subyacente(s) para cada ejemplar de entidad, es decir, el valor de ese atributo es obligatorio (no puede ser nulo) para toda instancia de la entidad

Existen dos **categorías** de entidades:

Regulares o fuertes

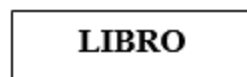
Son aquellas que existen por sí mismas y que la existencia de una instancia o ejemplar en la entidad no depende de la existencia de otras instancias en otra entidad. La representación gráfica dentro del diagrama es mediante un rectángulo.

Débiles

Son aquellas entidades en las que se hace necesaria la existencia de instancias de otras entidades distintas para que puedan existir instancias en esta entidad.

La representación gráfica dentro del diagrama es mediante un rectángulo con las esquinas ovaladas.

Un ejemplo de entidad débil sería EJEMPLAR, ya que la existencia de una instancia depende de la existencia del LIBRO, y, por tanto, la desaparición de un determinado libro de la base de datos hace que desaparezcan todas las instancias de dicho libro. Su representación como entidades sería la siguiente:



Capítulo 3 RELACIONES

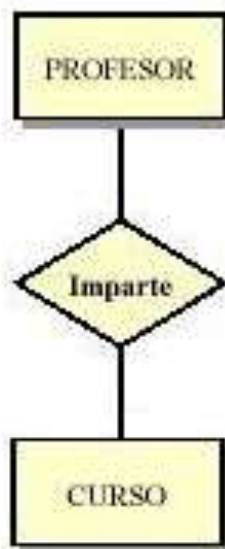
Se entiende por relación una asociación, vinculación o correspondencia entre entidades. Distinguiremos entre el tipo de relación o estructura genérica que describe un conjunto de relaciones, y cada relación, es decir, cada uno de las instancias concretas.

Un ejemplo de relación sería: imparte es un tipo de relación que vincula los tipos de entidad PROFESOR y CURSO; una instancia del tipo de relación imparte es la vinculación entre el profesor “Carrasco” y el curso “Modelamiento de Datos”.

DESCUBRIR LAS RELACIONES MAS RELEVANTES EN UNA REALIDAD.

En una interrelación podemos encontrar lo siguientes elementos:

Nombre: identificador único en el esquema.



CARDINALIDAD DE UNA RELACIÓN

La Cardinalidad establece cuántas instancias de una entidad están relacionadas con una instancia de la otra entidad.

Según su cardinalidad, las relaciones pueden ser:

- ✓ De uno a uno
- ✓ De uno a muchos
- ✓ De muchos a muchos

Ejemplos de la cardinalidad

Cardinalidad	Representación
De uno a cero, uno o muchos	
De uno a uno o muchos	
De uno a cero o uno	
De muchos a muchos	

REPRESENTACION DE LAS REGLAS NEGOCIO EN LAS CARDINALIDADES DE UNA RELACION

La Cardinalidad de una relación permite representar entonces la cantidad de instancias de una entidad que se relacionan con otra entidad y esa cuantificación reflejara como se comunican las instancias entre las entidades. El resultado está sujeto a las reglas de la realidad que permiten esa comunicación entre las entidades,

Ventajas del Diseño Conceptual

Luego podemos observar los siguientes beneficios del Diseño Conceptual:

- ✓ Herramienta de comunicación para audiencias no técnicas y técnicas
- ✓ Permite la Implementación de múltiples DBMS's
- ✓ Las bases de datos y aplicaciones son más adaptables a los cambios
- ✓ Las bases de datos y aplicaciones representan mejor a las necesidades de los negocios y usuarios finales
- ✓ Disminuye horas de diseño inicial y posterior mantenimiento.

EL MC Y LAS REGLAS DE NEGOCIO

EJERCICIO 1 de Modelo Conceptual

Se desea diseñar una base de datos que guarde la información de las reservas de una empresa dedicada al alquiler de automóviles. Los supuestos semánticos son los siguientes:

Un determinado cliente puede tener en un momento dado varias reservas.

Una reserva la realiza un único cliente, pero puede involucrar a varios coches.

Es importante registrar la fecha de comienzo de la reserva y la de terminación.

Todo coche tiene siempre asignado un número determinado de garaje, que no puede cambiar.

Cada reserva se realiza en una determinada agencia.

En la base de datos pueden existir clientes que no hayan hecho ninguna reserva.

Todas las entidades tienen una clave alfanumérica que las identifica unívocamente.

Se pide realizar el diseño del modelo Conceptual e indicar aquellos supuestos que no se han considerado.

Solución:

Primero se identifican las Entidades existentes:

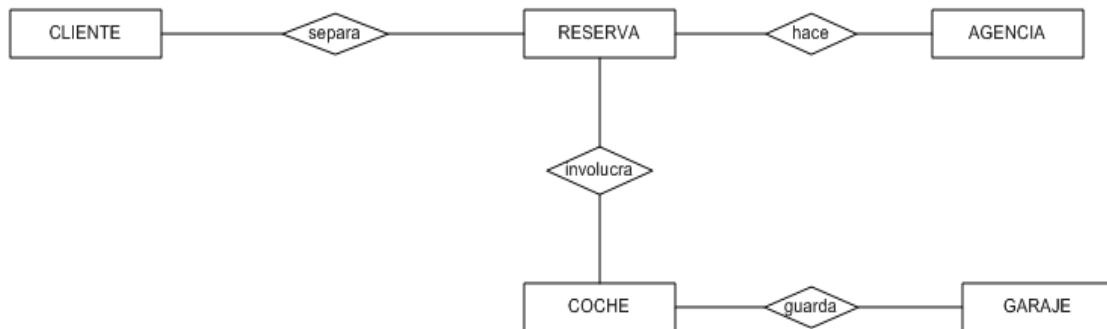
- ✓ CLIENTE
- ✓ RESERVA
- ✓ AGENCIA
- ✓ COCHE
- ✓ GARAJE

Se verifican si son entidades si cumplen con:

- ✓ tener varias Instancias
- ✓ tener varios atributos

Luego se grafican las entidades en un diagrama, relacionando las entidades según las reglas de negocios y la forma en que se comunican las entidades unas con otras.

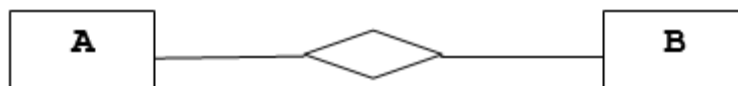
Por ejemplo, un cliente se relaciona con un coche por que este lo a usar, pero antes de llevarse el coche, primero tiene que separar una reserva, luego la relación más inmediata de la entidad Cliente es con la entidad Reserva.



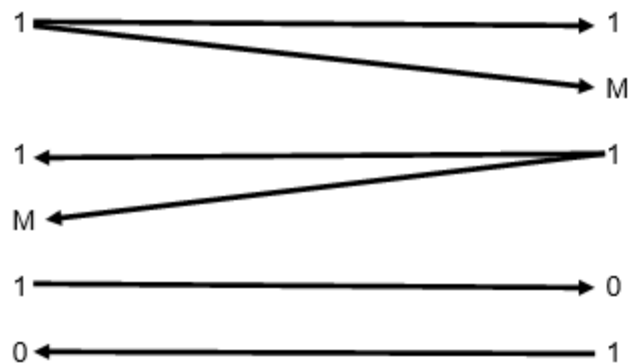
VERIFICACION DE LAS POSIBLES CARDINALIDADES DE UNA RELACION

Luego debemos encontrar las cardinalidades de cada relación verificando las siguientes posibilidades.

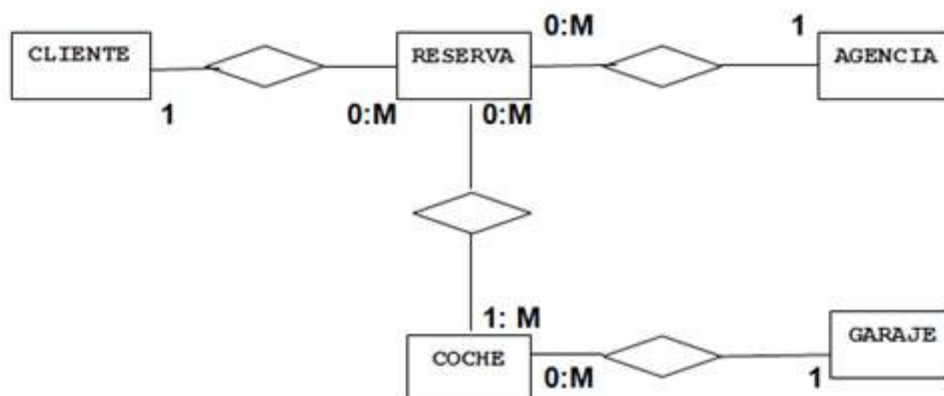
Suponiendo que tenemos dos entidades A y B:



Se verifican todas las posibilidades de A hacia B y de B hacia A



Luego el modelo con sus cardinalidades incluidas seria:



Capítulo 4 DISEÑO LÓGICO

TECNOLOGÍAS PRINCIPALES

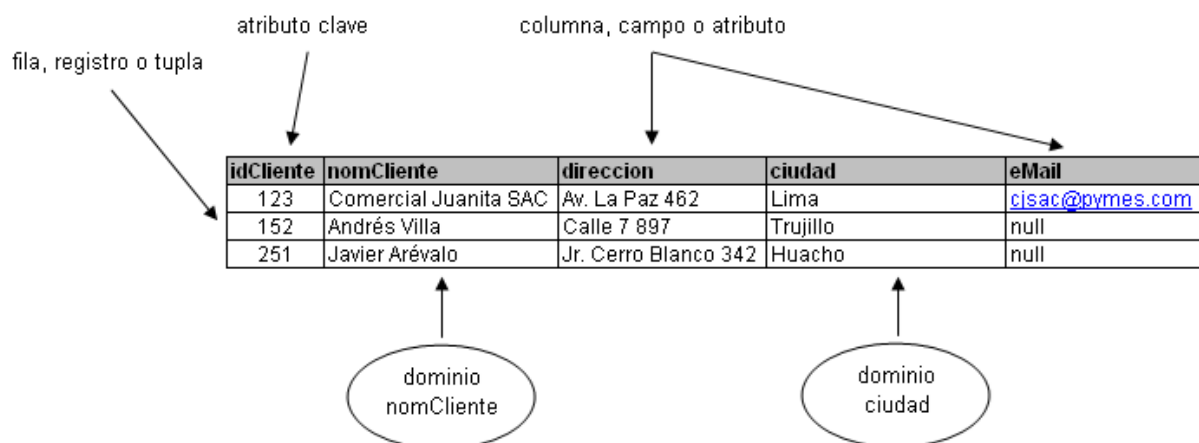
El diseño lógico parte del esquema conceptual y se obtiene como resultado un esquema lógico de base de datos. Un esquema lógico es una descripción de la estructura de la base de datos que puede procesar el software de DBMS. Un modelo lógico es un lenguaje usado para especificar esquemas lógicos. Los modelos lógicos más utilizados son:

- ✓ Redes
- ✓ Jerárquico
- ✓ Relacional

LA TECNOLOGÍA DEL MODELO RELACIONAL

En el modelo relacional los datos tienen una estructura de registros de longitud fija con un número fijo de campos o atributos. Los datos se representan en formato de filas y columnas.

La base del modelo relacional es la tabla o relación, un ordenamiento rectangular de filas y columnas de los valores de los datos.



TABLA, FILA, REGISTRO O TUPLA

Tabla

La estructura fundamental del modelo relacional es la tabla bidimensional constituida por filas (tuplas) y columnas (atributos). Las relaciones representan las entidades que se consideran interesantes en la base de datos. Cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representan las propiedades de la entidad.

Por ejemplo, si en la base de datos se tienen que representar personas, podrá definirse una relación llamada "Personas", cuyos atributos describen las características de las personas. Cada tupla de la relación "Personas" representará una persona concreta.

Por ejemplo, la relación: Personas (DNI, nombre, apellido, sexo, estadoCivil, fechaNacimiento) es apenas una definición de la estructura de la tabla, es decir su nombre y la lista de atributos que la componen. Si esta estructura se puebla con datos, entonces tendremos una lista de valores individuales para cada tupla, atributo por atributo.

Las tuplas en una relación son un conjunto en el sentido matemático del término, es decir una colección no ordenada de elementos diferentes. Para distinguir una tupla de otra, se recurre al concepto de "llave primaria", o sea un atributo o conjunto de atributos que permiten identificar unívocamente una tupla en una relación (en el ejemplo, el atributo DNI cumple con esta función).

COLUMNAS (ATRIBUTOS). DOMINIO

Columna, campo o atributo

Una relación o tabla contienen un conjunto de atributos que también se conocen como columna o campo. Cada atributo debe estar definido sobre un dominio

Dominio

Cada atributo de una relación se caracteriza por un nombre y por un dominio. El dominio indica qué valores pueden ser asumidos por una columna de la relación. A menudo un dominio se define a través de la declaración de un tipo para el atributo (por ejemplo, diciendo que es una cadena de diez caracteres), pero también es posible definir dominios más complejos y precisos.

Por ejemplo, para el atributo "sexo" de nuestra relación "Personas" podemos definir un dominio por el cual los únicos valores válidos son 'M' y 'F'; o bien para el atributo "fechaNacimiento" podremos definir un dominio por el que se consideren válidas sólo las fechas de nacimiento después del uno de enero de 1960, si en nuestra base de datos no está previsto que haya personas con fecha de nacimiento anterior a esa.

En forma general podemos definir cuatro tipos de dominio para el Modelo Lógico:

- Texto
- Numero
- Fecha
- Otros

Como podemos observar estos dominios son muy genéricos, es decir no se entra en muchos detalles todavía, un dominio tipo numero puede ser un entero grande o pequeño real o dinero, este detalle no interesa todavía en el modelo Lógico. Posteriormente se entrará en un detalle más específico del tipo de dato en el Modelo Físico.

CONCEPTO DE CLAVE PRIMARIA PK (PRIMARY KEY)

La clave primaria sirve para identificar unívocamente a cada instancia de la entidad, y puede estar formada por uno o más atributos. Cuando una clave está formada por más de un atributo se la conoce como clave compuesta. A la clave primaria se le conoce también como PK (Primary Key). En ocasiones, para una entidad dada, más de un atributo o conjunto de atributos puede ser seleccionado como clave. Estos atributos o conjuntos de atributos se conocen como claves candidatas.

Criterios para establecer la PK

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

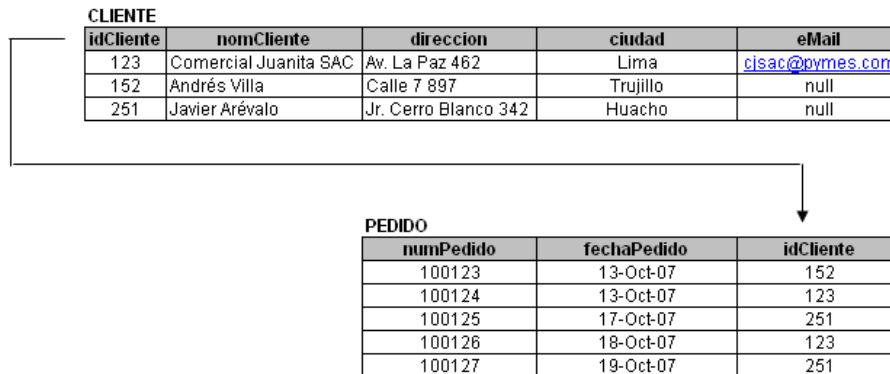
Criterio de Unicidad. Los atributos seleccionados como PK deben identificar unívocamente a una instancia de la entidad (no puede existir dos instancias con un mismo valor para la PK).

Criterio de Minimidad. El número de atributos que se selecciona como la clave primaria (PK) debe ser el mínimo.

Las claves que solo cumplen con el criterio de unicidad, son solo claves candidatas.

CONCEPTO DE CLAVE FORÁNEA FK (FOREING KEY)

La clave foránea, también denominada FK (Foreign Key) es el atributo de una entidad que la relaciona con otra entidad a través de la clave primaria de ésta. La clave foránea es un atributo que es clave primaria en la otra entidad.



Como se observa en el diagrama, el atributo IdCliente es la clave primaria de la entidad CLIENTE, y se ha definido como la clave foránea de la entidad PEDIDO.

Para que dos entidades se “relacionen” deben tener datos en común; esto es, atributos en común. Los atributos relacionados no necesitan tener el mismo nombre. Es suficiente que sus valores posibles correspondan al mismo dominio.

Capítulo 5

MODELO LÓGICO RELACIONAL

TRANSFORMACIÓN DEL MODELO CONCEPTUAL EN EL MODELO LÓGICO CORRESPONDIENTE.

Para obtener el esquema lógico de base de datos, en primer lugar, se debe tener a la mano el esquema conceptual de base de datos.

En segundo lugar, se debe seleccionar el modelo lógico a utilizar. En el curso utilizamos el modelo lógico Relacional explicado anteriormente.

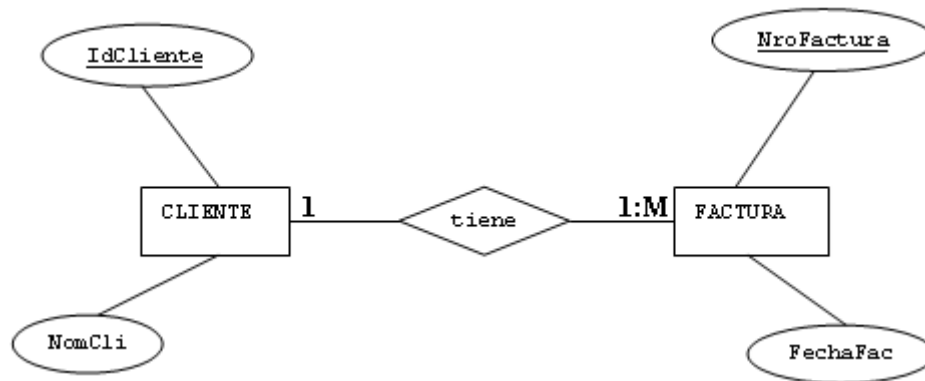
Finalmente se aplican las reglas de transformación que a continuación se detallan.

REGLAS DE TRANSFORMACIÓN SEGÚN LAS CARDINALIDADES DEL MODELO CONCEPTUAL

1. Toda entidad se convierte en una tabla o relación que toma el nombre de la entidad. Los atributos de la entidad serán las columnas de la tabla y el atributo identificador principal será la clave primaria. A menos que se indique lo contrario los atributos no identificadores podrán tomar valores nulos.
2. Las interrelaciones 1:N o 1:1 se transforman propagando el atributo identificador principal de la entidad que tiene cardinalidad máxima 1 a la que tiene cardinalidad máxima N. Si la relación fuese 1:1 la propagación de clave podría hacerse en cualquier sentido. El atributo propagado es una clave ajena que referencia a la tabla con cardinalidad máxima de 1.
3. Las interrelaciones N:M se transforman en una tabla cuya clave primaria será la concatenación de los atributos principales de las entidades que se asocia; estos atributos serán claves ajenas que referencian a las respectivas tablas donde son claves primarias. Los atributos de la interrelación serán columnas de la tabla.

RELACIÓN NO IDENTIFICADORA

Ejemplos de Transformación



Transformando al Modelo Lógico:



Los atributos identificadores se transforman en Claves Primarias (PK)

La relación de cardinalidad de uno a muchos se transforma en una

Relación No Identificadora, y se grafica con línea discontinua.

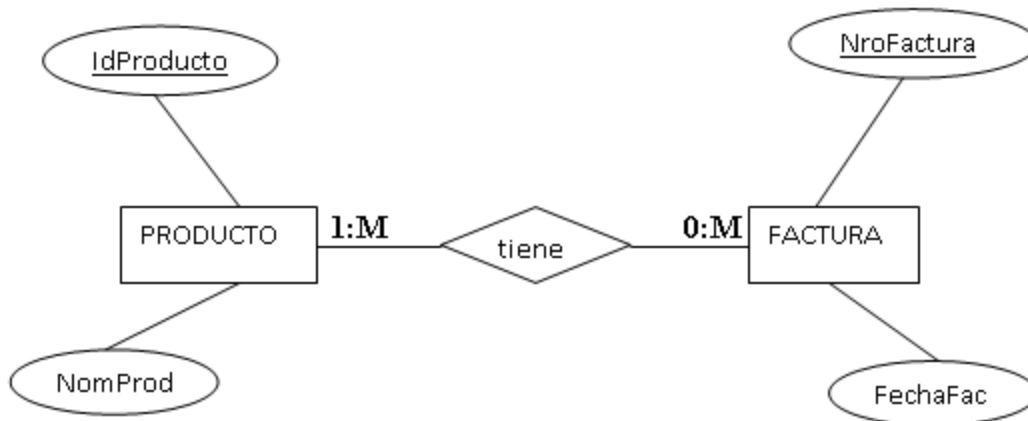
La entidad que tiene el lado de la cardinalidad 1 es la que tiene la PK en la relación; se le denomina Entidad Padre.

La Entidad Padre entonces le transmite su PK como Clave Foránea a la otra Entidad con Cardinalidad de Muchos. A esta entidad se le denomina Entidad Hija. (para esta relación).

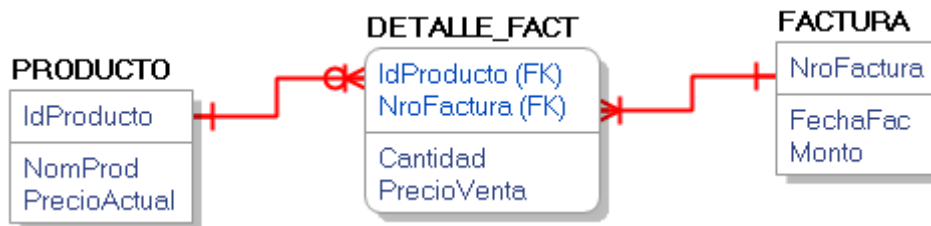
Luego para el ejemplo CLIENTE es Padre y FACTURA es Hija

RELACIÓN IDENTIFICADORA

Otro ejemplo de transformación



Transformando al Modelo Lógico correspondiente:



La relación de cardinalidad de muchos a muchos se transforma en una Relación Identificadora.

Entre las dos entidades iniciales se grafica una entidad intermedia cuya PK está conformada por las PKs de las otras dos.

La PK de la nueva entidad intermedia se compone por las PK's de las otras dos, las cuales pasan a la entidad intermedia como Claves Foráneas (FK)

La relación Identificadora se grafica con línea continua.

ENTIDAD DEPENDIENTE

Entidad Dependiente (débil) Tiene la PK formada por claves Foráneas p.e. DETALLE_FACT; su existencia depende de FACTURA y PRODUCTO.

Se dibuja con las esquinas ovaladas.

ENTIDAD INDEPENDIENTE

Entidad Independiente (fuerte) Tiene la PK formada por atributos que no son foráneos.
Tiene existencia propia.

Ejercicio 1 de Transformación

Se tiene una tienda que tiene una serie de clientes que compran y luego se le prepara una factura que indica los productos que se han vendido en una determinada venta que queda registrada como una factura.

De los clientes se requieren guardar datos como su código, nombre, dirección.

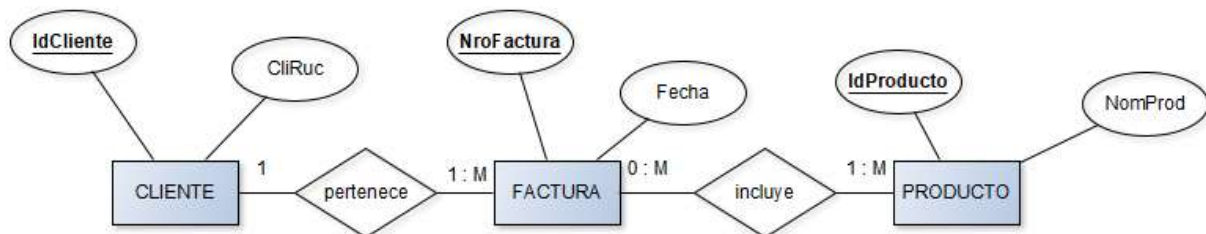
Cada factura generada tiene un número, una fecha y un monto.

Cada producto también tiene un código que los distingue, también se debe conocer su nombre, Stock.

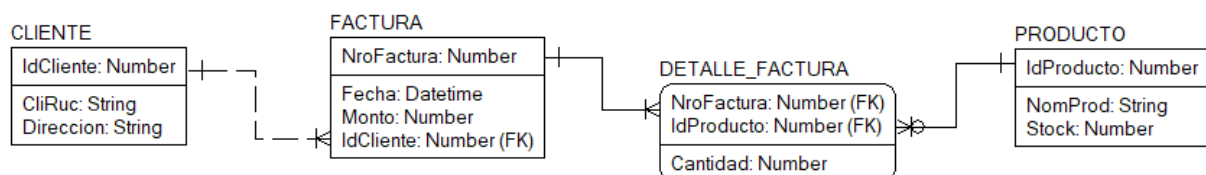
Construir el Modelo Conceptual correspondiente a esa realidad. Luego transformar el Modelo Lógico obtenido a un Modelo Lógico Relacional, usando correctamente las reglas de transformación para cada caso.

Solución:

Modelo Conceptual:



Transformación al Modelo Lógico:



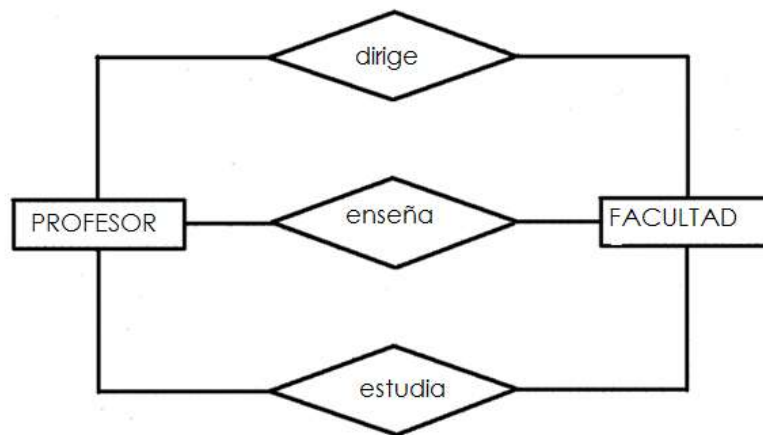
Capítulo 6

RELACIONES ADICIONALES

RELACIONES MULTIPLES ENTRE DOS ENTIDADES

Se puede dar el caso de más de una relación entre dos entidades, todo depende de las reglas de negocio existentes.

También pueden existir relaciones múltiples entre las mismas entidades, como se ejemplifica a continuación:



Relaciones Múltiples entre las Mismas Entidades

En el caso de una Universidad.

Un Profesor puede enseñar en una Facultad.

El que dirige a la facultad es el decano, pero para elegir un decano por ley, debe ser un Profesor que cumpla con ciertos requisitos, y que además se ha elegido por votación. Luego, un Profesor también puedes dirigir a la Facultad.

Un Profesor de una Facultad, podría estar estudiando una maestría en la misma Facultad, por lo tanto, tendríamos una tercera relación entre Profesor y Facultad.

Desarrollar el siguiente caso.

Ejercicio

Una empresa de transporte de pasajeros requiere se le diseñe una Base de Datos que cumpla con los siguientes requerimientos:

La empresa se dedica al transporte interprovincial de pasajeros entre las principales ciudades del país.

Todas sus rutas son directas; es decir, que partiendo de la ciudad de origen se dirigen sin escalas a la ciudad destino.

Se desea guardar información de las ciudades, para que los usuarios puedan conocer las características de las mismas.

Cada ruta tiene una ciudad de origen, una hora de salida programada, una hora de salida real, una ciudad destino, una hora de llegada programada, una hora de llegada real.

Para cada ruta se tiene un precio determinado.

Las unidades de transporte se han adquirido de distintos fabricantes y tienen distinta capacidad e pasajeros. Hay unidades para 30 pasajeros, unidades para 40 pasajeros, etc.

Se desea guardar información de los fabricantes de los buses para mantener un correcto funcionamiento de las unidades adquiridas.

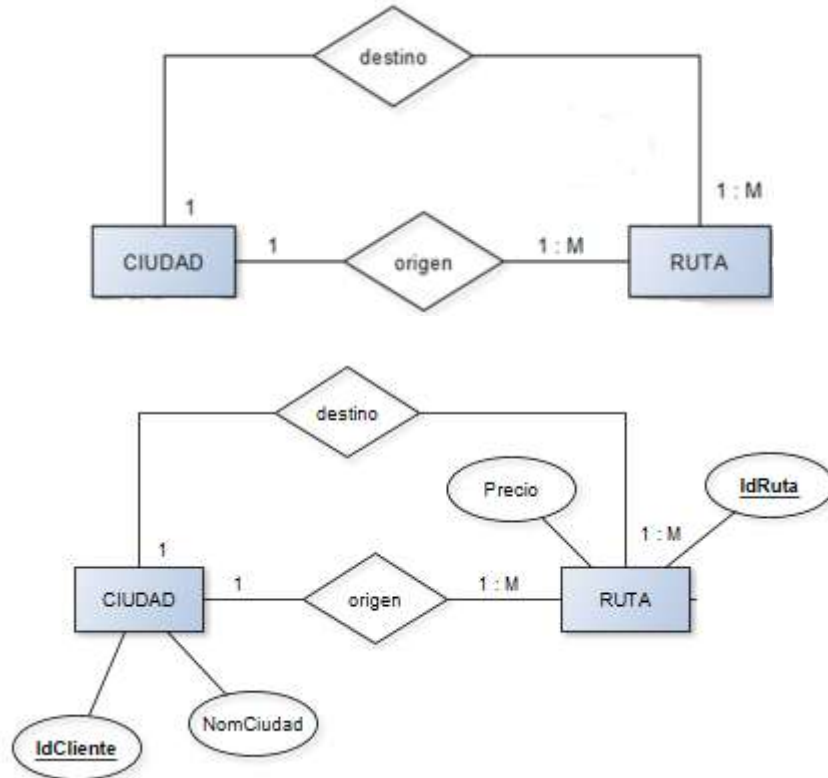
Para cada viaje se asignan 2 o más choferes. Además, en cada viaje se tiene a bordo a una o más azafatas para la atención a los pasajeros.

Se consideran a los choferes y azafatas como tripulantes en general. De ellos, se requiere almacenar sus datos personales.

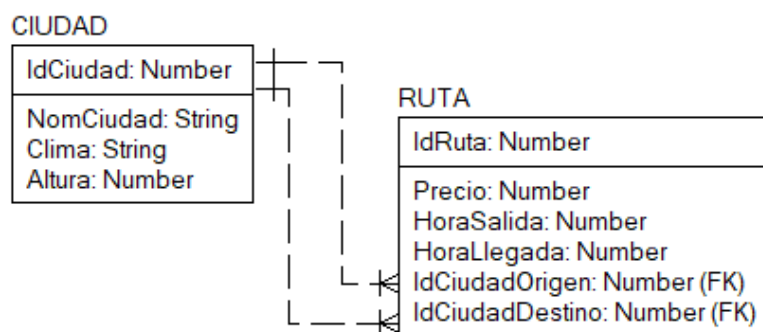
Presentar el modelo de datos conceptual. Transformar la Modelo Lógico correspondiente.

Solución:

En el caso de CIUDAD con RUTA se presenta una relación DOBLE



Al transformar la relación. La Clave Primaria (IdCiudad) migrara dos veces a la entidad hija RUTA. Es por eso que se le debe cambiar el nombre a la clave Foránea migrada.



Terminar de desarrollar el resto del Ejercicio.

RELACIONES RECURSIVAS

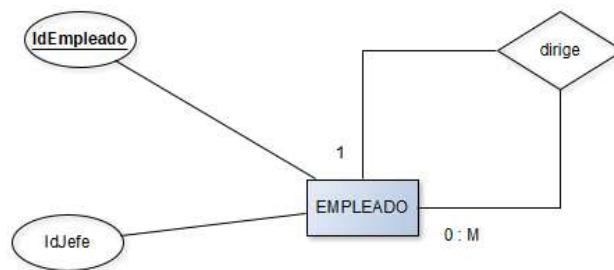
Relaciona una entidad consigo misma.

La línea de relación parte de la entidad y regresa a la misma.

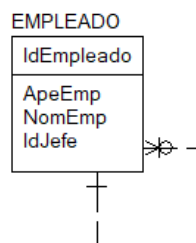
Ejemplo: empleados que pueden ser jefes de otros empleados.

En este caso de los empleados, existe un jefe que es a su vez un empleado, pero que tiene a su cargo a otros empleados, por lo tanto, la relación se da en la misma entidad. A este tipo de relación también se le conoce como relación HOOK (gancho).

El Modelo Conceptual sería:



El Modelo Lógico correspondiente es:

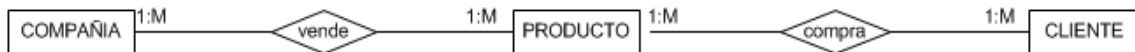


IdEmpleado	ApeEmp	NomEp	IdJefe
1	Javier	Valle	
2	Pedro	Gomez	1
3	Rosa	Vilchez	1
4	Carlos	Perez	2
5	Maria	Soto	2
6	Nancy	Orbegozo	2
7	David	Herrera	3
8	Kevin	Quispe	3
9	Luisa	Zapata	3
10	Fidel	Mandujano	3

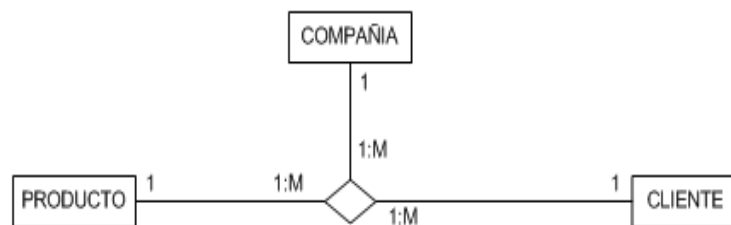
RELACIONES TERNARIAS

Supongamos el caso de varias COMPAÑÍAS que venden varios PRODUCTOS a varios CLIENTES. Esta situación se puede representar mediante dos relaciones binarias, tal como se muestra en el diagrama siguiente.

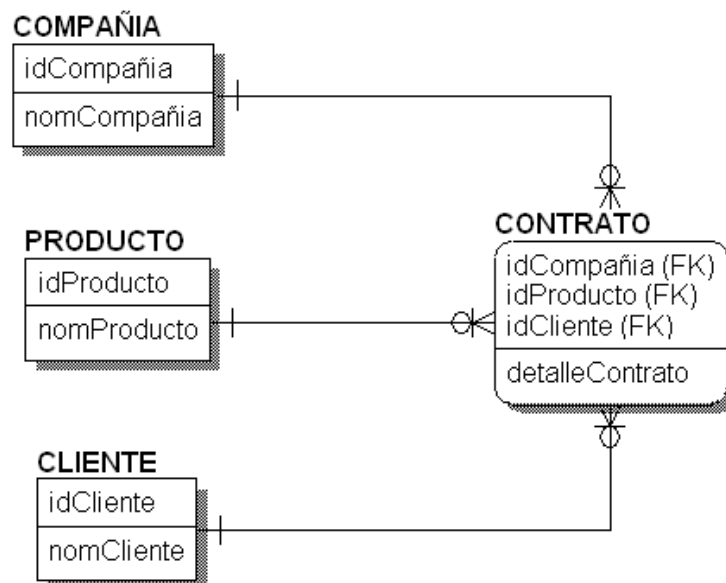
En el Modelo Conceptual tendríamos:



Y su transformación en el Modelo Lógico sería:



Y su equivalente en el Modelo Lógico sería:



RELACIONES DE SUPERTIPOS Y SUBTIPOS

Un subtipo o sub-entidad es un subconjunto de otra entidad. Consideremos el modelo de datos de una entidad financiera que tiene distintos tipos de CUENTAs:

Cuenta de ahorros: número de la cuenta de ahorros, fecha de apertura, fecha de revisión, balance, tasa de interés, interés ganado.

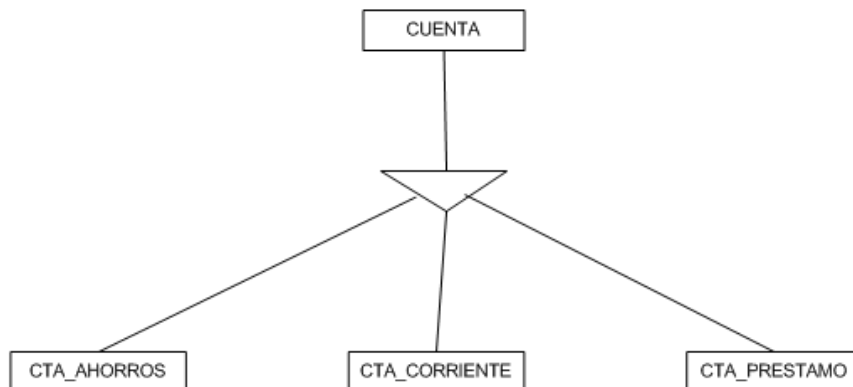
Cuenta corriente: número de la cuenta corriente, fecha de apertura, fecha de revisión, balance, balance disponible, cargo por cuenta corriente.

Cuenta de préstamo: número de la cuenta de préstamo, fecha de apertura, fecha de revisión, monto del préstamo, tasa de interés, balance actual.

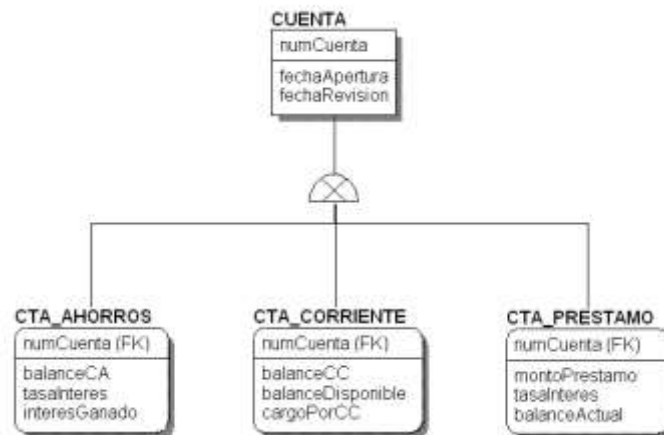
Observe que las distintas cuentas tienen atributos comunes que se pueden agrupar en un supertipo o entidad generalizada, creándose de esta manera una estructura jerárquica de entidades.

Cada una de las cuentas y los atributos que son exclusivos de ella formarían un subtipo o sub-entidad.

Su representación en el Modelo Conceptual sería el siguiente:



Al transformar al Modelo Lógico tendríamos:



Ejercicio 1 de Relación Ternaria

La empresa de desarrollo de software ABC.NET tiene en la actualidad 5 proyectos en ejecución y 4 proyectos por iniciar. Cada uno de los proyectos incluye software, equipos y repuestos (hardware) que son abastecidos por 8 empresas mayoristas. El gerente general desea que se elabore un reporte donde se indique la cantidad de productos que cada proveedor abasteció a cada uno de los proyectos. Dibujar el diagrama E/R.

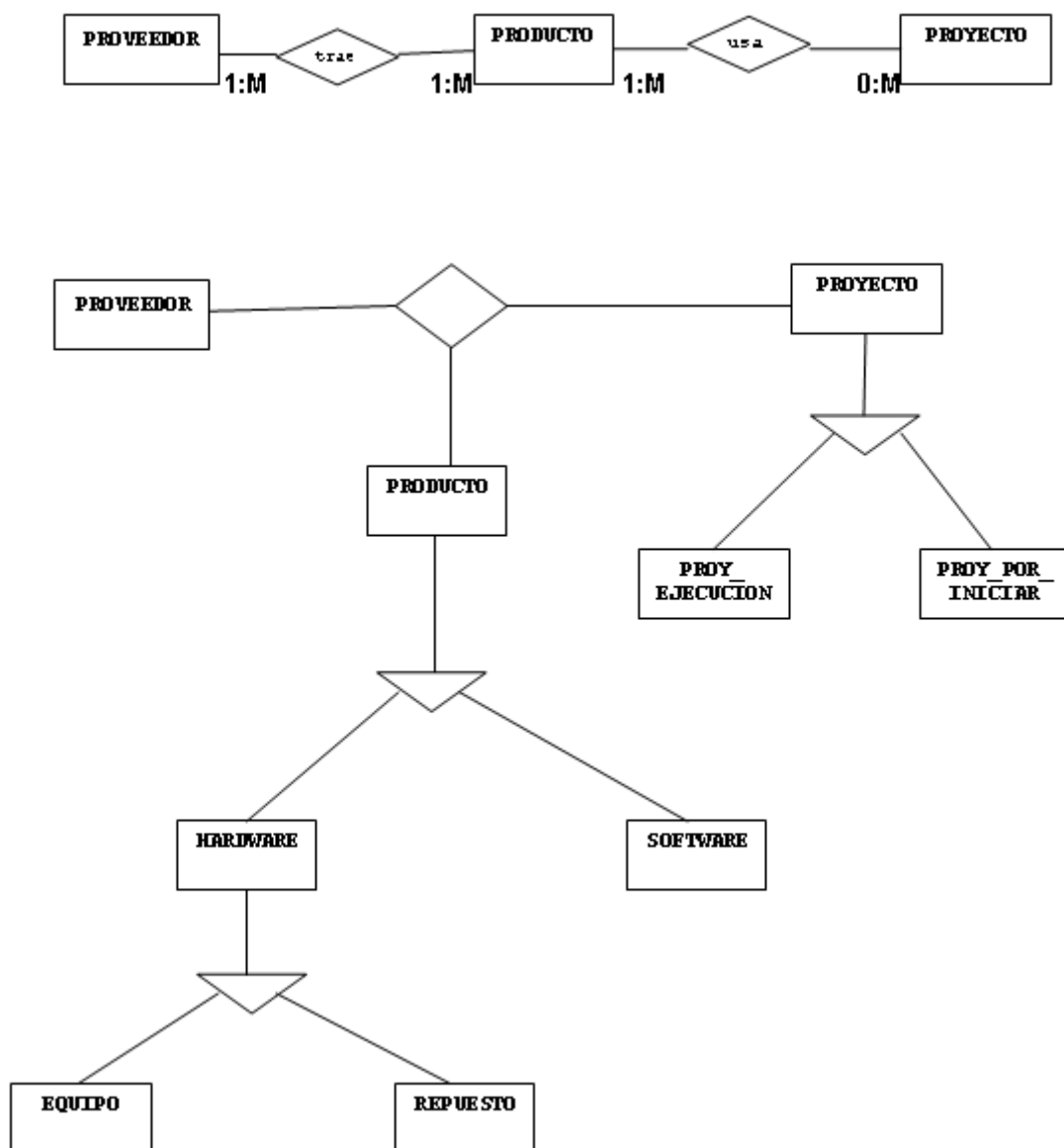
Diseñar el Modelo Conceptual y el Modelo Lógico.

Solución:

Las Entidades involucradas son:

- PROYECTO
- EQUIPO
- PROVEEDOR

El Modelo Conceptual sería



Capítulo 7

NORMALIZACIÓN

MODELO RELACIONAL DE CODD

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

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

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

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

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

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

La presentación formal de la teoría de la Normalización exige un bagaje matemático, con sus correspondientes algoritmos, y es solo indispensable en una formación académica universitaria. No pretendo desarrollar esta formalización, sino más bien presentar estas ideas de una manera intuitiva para que los lectores a quienes va

dirigido este trabajo, que por lo general son personas que quieren llegar directo “al grano”, puedan comprender esta teoría de manera muy simple y práctica.

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

PRIMERA FORMA NORMAL (1FN)

Una entidad se encuentra en la primera forma normal si todos sus atributos son simples (no descomponibles), y si no tiene grupos repetitivos.

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

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

- ✓ Identificar la clave primaria en la entidad compleja
- ✓ Identificar los grupos repetitivos en la entidad compleja.
- ✓ Rescribir la entidad compleja original sin los grupos repetitivos.
- ✓ Crear una segunda entidad con los grupos repetitivos
- ✓ Añadir la clave de la primera entidad (la original) a la segunda entidad.

Ejercicio 1 - Caso práctico

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

BOLETA DE VENTA					
Boleta Nro 1356					
Cliente: ALFASA			Código. C201		
Vendedor: Marcelo V. Ricardo			Código del vendedor: V14		
Fecha:.....02/05/2012					
Cod	Producto	Unidad	Can	Precio Unidad	Sub Total
P104	Arroz del Norte	Saco	2	130.00	260.00
P089	Leche Gloria	Caja	3	120.00	360.00
P133	Aceite El Sol	Caja	5	90.00	450.00
Total de venta:				1070.00	

Esta es solo una boleta de tanta de las que existen en esta realidad. Existen luego varias instancias de boletas. También observamos que las boletas tienen varios atributos, Luego podemos concluir que todas las boletas se pueden representar como una entidad llamada BOLETA.

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

La representación formal de la Entidad es un rectángulo con la Clave Primaria en la parte superior y todos los demás atributos abajo

BOLETA	
NroBol	
IdCli	
Cliente	
NomVen	
IdVend	
Fecha	
IdProd	
Producto	
Unidad	
Cantidad	
TotalVenta	

Para poder mostrar el proceso de Normalización de forma más adecuada, vamos a inclinar a la entidad “echándola” hacia la izquierda.

NroBol	IdCli	Cliente	NomVen	IdVend	Fecha	IdProd	Producto	Unidad	Cantidad	TotalVenta
--------	-------	---------	--------	--------	-------	--------	----------	--------	----------	------------

Ahora vamos representar a la ENTIDAD como un “Tren de atributos”

NroBol	IdCli	Cliente	NomVen	IdVend	Fecha	IdProd	Producto	Unidad	Cantidad	TotalVenta
--------	-------	---------	--------	--------	-------	--------	----------	--------	----------	------------


La denominaremos Entidad Compleja, debido a las complejidades o anomalías que contiene, las cuales serán solucionadas por el proceso de normalización que vamos a desarrollar a continuación.

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

Identificar los grupos repetitivos en la entidad compleja.

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

Grupo Repetitivo



NroBol	IdCli	Cliente	NomVend	IdVend	Fecha	Cod	Producto	Unidad	Can	PU	SubTotal	TotalVenta
1356	C201	ALFASA	Marcelo Ricardo	V14	02/05/12	P104	Amondillo Norte	Saca	2	1300.	260.	1070.00
1357	C104	OMEGA	Jaime Vidal	V02	03/05/12	P018	Jabon ROLL	docena	3	17.0	41.0	680.00

Como se puede observar los atributos que forman parte del grupo repetitivo son desde Codd hasta Subtotal que son los atributos que tienen varios datos para una misma celda. Por ejemplo, en la celda que corresponde a la columna del Codd en la primera fila debe estar el dato P104, pero también en esta misma fila deberían estar los datos P089 y P133, ya que corresponden a la misma Boleta o sea a la misma instancia; estos datos no se podrían ubicar más abajo, en la siguiente fila, porque estos lugares corresponden a otra fila o a otra instancia. Luego estos tres datos deben ubicarse en la misma celda yuxtaponiéndose entre ellos; por lo tanto, los datos de este atributo no permiten representar a la entidad en un plano sino más bien en una representación espacial.; luego este atributo se dice forma parte del grupo repetitivo.

Lo anterior también se aplica para los atributos Producto, Unidad, Can, PU, y SubTotal

Rescribir la entidad compleja original sin los grupos repetitivos.

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

NroBol	IdCli	Cliente	NomVend	IdVend	Fecha	TotalVenta
---------------	-------	---------	---------	--------	-------	------------

Esta entidad ahora está ahora en Primera Forma Normal (1FN) ya que se le han extraídos los atributos anómalos del Grupo repetitivo; se puede representar en dos dimensiones sin ningún problema.

Crear una segunda entidad con los grupos repetitivos y añadir la clave de la primera entidad (la original) a la segunda entidad. Al pasar la PK a otra entidad, pasa como Clave Foránea (FK).

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

Entidad “B”.

NroBol(FK)	Codd	Producto	Unidad	Can	PU	SubTotal
------------	------	----------	--------	-----	----	----------

Esta entidad también está en 1FN porque si quisiéramos representarla como una tabla se puede sin ningún problema, como lo mostramos a continuación:

NroBol(FK)	Codd	Producto	Unidad	Can	PU	SubTotal
1356	P104	Arroz del No	Saco	2	130.00	260.00
1356	P089	Leche Gloria	Caja	3	120.00	360.00
1356	P133	Aceite El Sol	Caja	5	90.00	450.00
1357	P104	Arroz del No	Saco	4	135.00	540.00
...

Como el NroBol es ahora FK se puede repetir sin ningún problema en esta nueva entidad B.

Como se observa es ahora Plan, luego ya está también en 1FN.

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

A

NroBol	IdCli	Cliente	NomVend	IdVend	Fecha	TotalVenta
---------------	-------	---------	---------	--------	-------	------------

B

NroBol(FK)	Codd	Producto	Unidad	Can	PU	SubTotal
------------	------	----------	--------	-----	----	----------

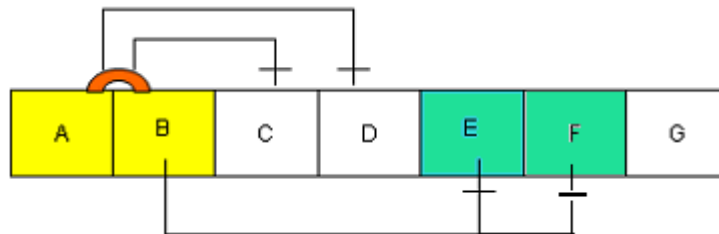
SEGUNDA FORMA NORMAL (2FN)

Para pasar una entidad a la segunda forma normal, el primer requisito es que ya se encuentre en primera forma normal; luego se deben eliminar los atributos que tengan Dependencias parciales.

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

Dependencia Parcial.

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



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

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

Asumamos también que el atributo F le sucede lo mismo, también se puede identificar tan solo conociendo el atributo B, sin necesidad de conocer el atributo A. Luego se puede afirmar que el atributo F también tiene una dependencia funcional parcial de la PK.

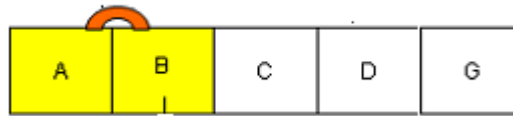
Para pasar una Entidad a la Segunda Forma Normal, las condiciones son:

La Entidad ya debe estar en 1FN

La Entidad no debe tener atributos con Dependencia Parcial

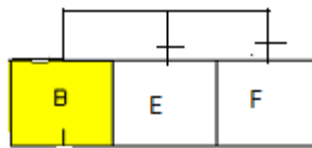
Luego Para normalizar, primeros sacamos las anomalías:

La Entidad quedaría así:



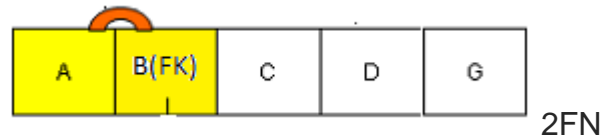
Como en esta entidad resultante ya se eliminaron las dependencias parciales, podemos afirmar ya que esta en Segunda Forma Normal (2FN)

Con los atributos anómalos (los que sacamos por tener dependencia parcial) y el atributo que los identifico (en este caso B, formamos una nueva entidad:

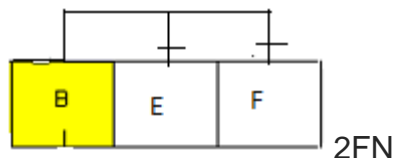


En esta nueva entidad como el atributo B es que identifica a todos los demás atributos de la entidad, se convierte en la PK de esta última entidad, quedando entonces en la entidad anterior el atributo B como Clave Foránea (FK). Esta FK se relaciona con la PK de la anterior entidad; luego tenemos dos entidades resultantes, pero que tienen como nexo a la PK y a la FK.

B1:



B2:



En nuestro caso ejemplo, tenemos dos entidades A y B resultantes que se encuentran en Primera Forma Normal; luego son candidatas a aplicarles la Segunda Forma Normal.

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

atributo que se identifique con la PK, ya que al depender funcionalmente de la PK dependerá de toda, no de una parte de ella.

A

NroBol	IdCli	Cliente	NomVend	IdVend	Fecha	TotalVenta
---------------	-------	---------	---------	--------	-------	------------

2FN

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

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

NroBol(FK)	Codd	Producto	Unidad	Can	PU	SubTotal
1356	P104	Arroz del No	Saco	2	130.00	260.00
1356	P089	Leche Gloria	Caja	3	120.00	360.00
1356	P133	Aceite El Sol	Caja	5	90.00	450.00
1357	P104	Arroz del No	Saco	4	135.00	540.00
...

Aquí se puede observar que el NroBol se puede repetir en diferentes instancias de la entidad B, esto se debe a que es una Clave Foránea (FK).

Pero con solo este atributo no se puede identificar plenamente a una instancia en particular de esta entidad.

Por ejemplo, si tuviéramos el NroBol 1356 tendríamos tres casos que cumplen con este valor. Luego, se debe recurrir a otro atributo adicional para formar la PK.

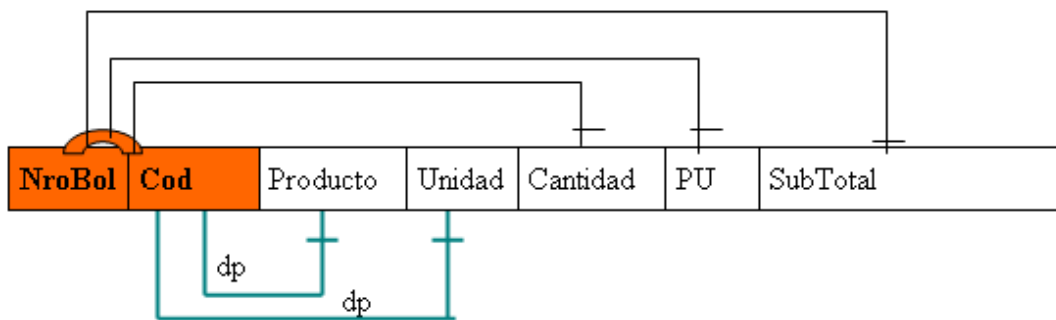
Un posible candidato podría ser el atributo Producto; Si tenemos el NroBol 356 y el Producto Arroz del Norte, identificamos inmediatamente a la primera fila de la tabla.

La cuarta fila no podría ser por pertenecer a otra boleta (1357). Por lo tanto, la combinación NroBol + Producto cumple con el criterio de Unicidad.

Otra posible combinación sería NroBol + idProd, y cómo podemos observar también cumple con el primer criterio de Minimidad. Pero además es más pequeña que la primera candidata; Por lo tanto, también cumple con el criterio de Minimidad.

Finalmente elegimos a la combinación NroBol + idProd como la PK de la entidad B:

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

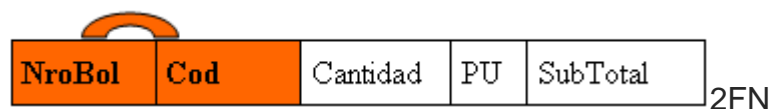


En el grafico se puede observar que los atributos Producto (Nombre del producto) y Unidad (unidad en que se comercializa el producto), se pueden identificar solo con conocer el Codd (el código del producto debe ser único por producto).

Luego no necesitan al toda la PK para ser identificados, por lo tanto, tienen dependencia funcional parcial de la PK.

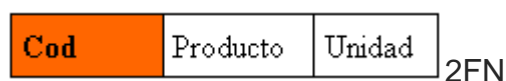
Para pasar a la 2FN se deben eliminar los atributos que originen dependencias parciales:

Entidad B1



Y con los atributos eliminados tenemos:

Entidad B2



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

Hasta este momento, en este ejemplo, tenemos tres entidades resultantes en

2FN: A, B1 y B2.

Entidad A

NroBol	IdCli	Cliente	NomVend	IdVend	Fecha	TotalVenta
---------------	-------	---------	---------	--------	-------	------------

2FN

Entidad B1

NroBol	Cod	Cantidad	PU	SubTotal
---------------	------------	----------	----	----------

2FN

Entidad B2

Cod	Producto	Unidad
------------	----------	--------

2FN

TERCERA FORMA NORMAL (3FN)

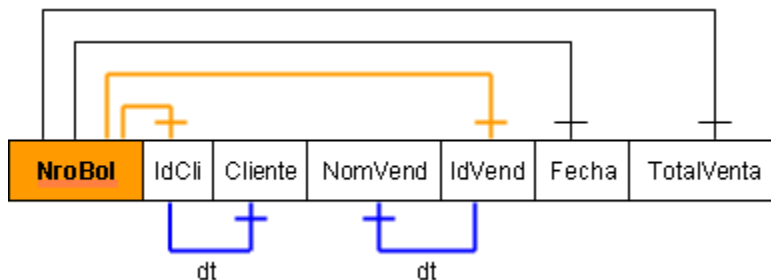
Una entidad se encuentra en la Tercera Forma Normal si ya está en la 2FN, y además no tiene atributos con dependencia Transitiva; es decir, atributos que se pueden obtener a partir de otros.

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

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

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

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



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

Del mismo modo se puede afirmar que el atributo NomVend tiene otra dependencia transitiva de la PK.

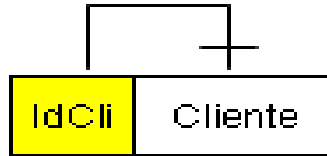
A continuación, procedemos a eliminar estas dependencias transitivas de la entidad A y obtenemos la siguiente entidad resultante:

Entidad A1

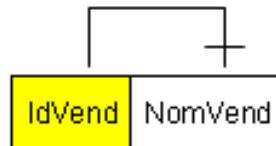
NroBol	IdCli	IdVend	Fecha	TotalVenta
---------------	-------	--------	-------	------------

A continuación, con los atributos extraídos formamos otras entidades:

Entidad A2



Entidad A3



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

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

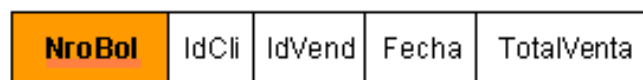
Cuando se tiene, que las entidades resultantes solo tienen dependencias directas y totales (obsérvese las líneas de dependencia), podemos afirmar que el proceso de normalización ha terminado.

Como resultado final entonces tenemos en este caso cinco entidades:

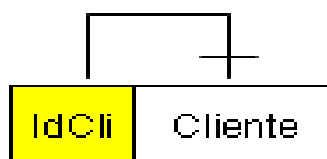
A1 A2 A3 B1 B2

A continuación, se debe poner nombres más específicos, que reflejen su esencia, (en singular), a estas entidades resultantes:

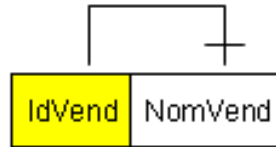
A1 BOLETA



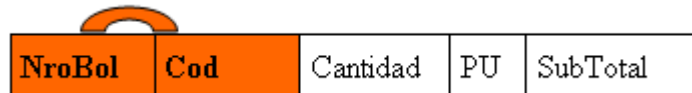
A2 CLIENTE



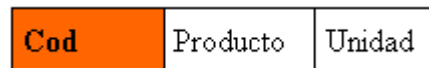
A3 VENDEDOR



B1 DETALLEBOLETA

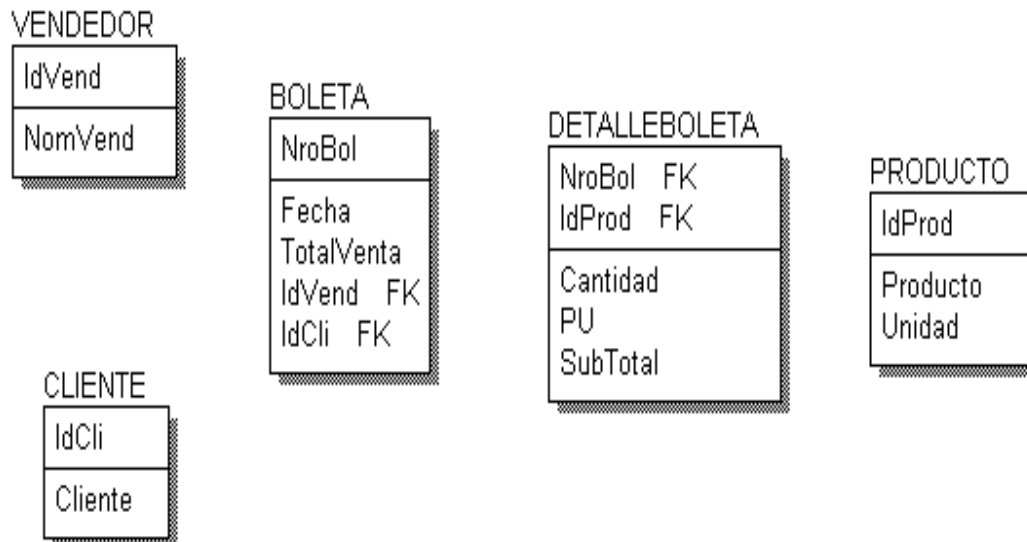


B2 PRODUCTO



El siguiente paso es construir el Diagrama Entidad Relación con estas entidades resultantes del proceso de normalización.

Las entidades resultantes, otra vez regresan a su forma original (Paradas). Primero se dibujan las entidades con sus atributos indicado en la parte superior del cuadrilátero que representa a cada entidad a la PK en un recuadro. Las claves foráneas se deben indicar para establecer luego las relaciones. El diagrama debe mostrar lo siguiente:



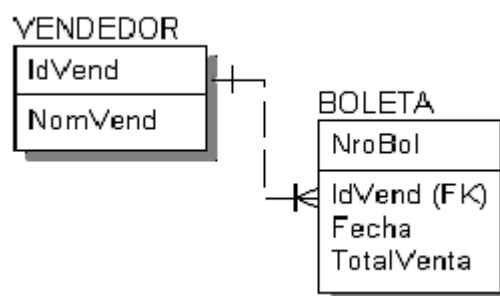
Luego hay que definir el tipo de relación a establecer entre las entidades; por ejemplo, vamos determinar el tipo de relación que existe entre la entidad VENDEDOR y la entidad BOLETA, que como ya se había determinado en el proceso de la Normalización, el nexa entre estas dos entidades es el atributo IdVend.

Se observa que el atributo nexa es PK en la entidad VENDEDOR y, a su vez es FK en la entidad BOLETA; por esta razón, para esta relación en particular VENDEDOR es la Entidad Padre, y BOLETA es la Entidad Hija.

Además, para determinar una instancia en particular en la entidad Hija no se necesita conocer IdVend (al vendedor); luego podemos especificar un número de boleta, por ejemplo 356 y ya podemos esa boleta; no hubo necesidad de especificar el vendedor (IdVend).

Además, si solo tuviéramos como dato el código del vendedor (IdVend) no podríamos particularizar una boleta, pues pueden existir muchas boletas relacionadas con un mismo vendedor. Por lo tanto, podemos afirmar que la relación existente entre VENDEDOR y BOLETA es una RELACIÓN NO IDENTIFICADORA.

A continuación, trazamos la línea de relación entre las dos entidades:



La línea de relación parte de la entidad Padre, señalando a la PK con Cardinalidad uno, y llegando a la entidad Hija con Cardinalidad uno o muchos. Esta Cardinalidad inicial podrá variarse luego verificando las reglas de negocio de la realidad que estamos modelando.

Nótese que, en una relación identificadora, la entidad Hija tiene a la FK como un atributo común más.

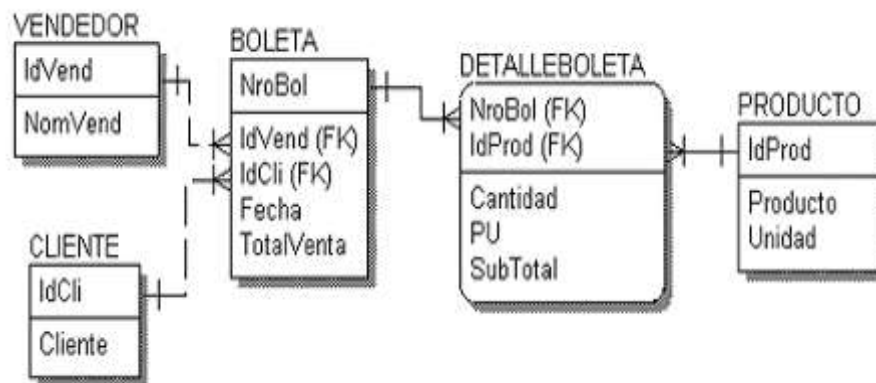
En el caso de la relación entre CLIENTE y BOLETA también se cumple algo similar, y luego se determina que la relación existente entre estas dos entidades también es una RELACION NO IDENTIFICADORA.

Con respecto a la relación entre BOLETA y DETALLEBOLETA,

El atributo que sirve de nexa entre las entidades en el atributo NroBol podemos anotar que para identificar una instancia de DETALLEBOLETA se necesita conocer un número de boleta (NroBol), si este dato será imposible ubicar una instancia en DETALLEBOLETA. Luego la relación existente entre estas dos entidades en una

RELACION IDENTIFICADORA porque con la PK de BOLETA podremos identificar alguna instancia que represente a algún detalle de la venta de un Producto. Luego se hace también necesario conocer el producto que se vendió en dicha boleta (IdProd). De esto último entonces también se infiere que la relación existente entre PRODUCTO y DETALLEBOLETA es también una RELACIÓN IDENTIFICADORA.

Finalmente llegaremos al siguiente Diagrama Entidad Relación (E-R):



Podemos señalar que para la Relación Identificadora la FK en la entidad Hija figura como parte de la PK.

Podemos también afirmar que la entidad DETALLEBOLETA tiene instancias que solo podrán existir si existen Boletas y Productos, luego esta entidad es una Entidad DEPENDIENTE o también conocida como DEBIL.

En cambio, las otras entidades no necesitan de atributos foráneos para identificar a cada una de sus respectivas instancias. Luego podemos afirmar que son Entidades INDEPENDIENTES o también conocidas como FUERTES.

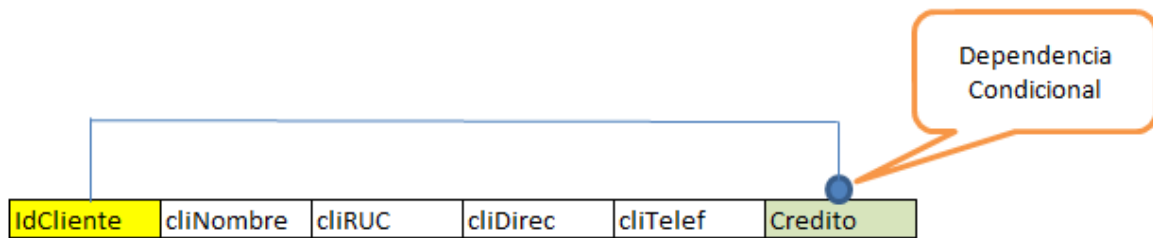
ANOMALIAS ADICIONALES. ANOMALÍA DE LA DEPENDENCIA CONDICIONAL (VALOR NULL)

Se establece cuando en un campo de una entidad para algunas instancias no existen datos, o sea tienen valores nulos.

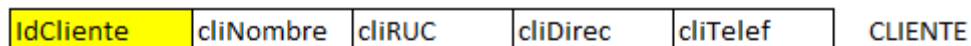
Por ejemplo, existen algunos clientes que no tiene crédito en un grupo de clientes.

Luego en el atributo Credito de la entidad cliente existen algunas instancias que en el atributo Credito no tienen valor alguno, (Tienen nulos).

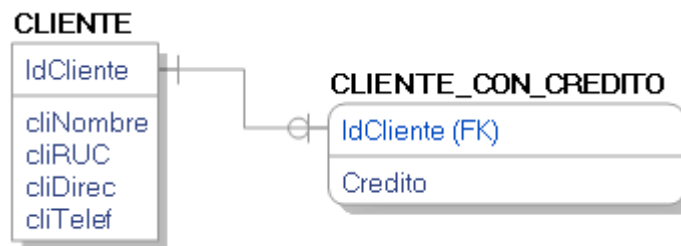
Luego en el gráfico de la entidad CLIENTE tendríamos la siguiente figura:



Si se quiere eliminar esta dependencia condicional en la entidad CLIENTE, debemos separar los atributos con dependencia Condicional y con una copia de la PK para mantener la referencia entre las entidades resultantes.



Llevando estas entidades al modelo Lógico:



Se observa que este tipo particular de relación la cardinalidad siempre será de uno a ninguno o uno.

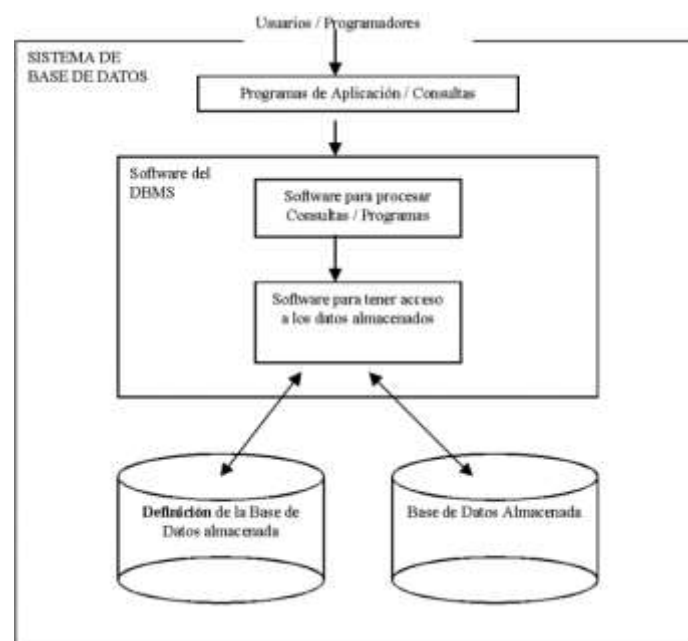
Capítulo 8

DISEÑO FÍSICO

El modelo físico es un modelo que representa la realidad en la implementación y por lo tanto es dependiente de la plataforma que se use. Se necesita definir por lo tanto el DBMS con el que se va a implementar finalmente la Base de Datos. En nuestro caso vamos a usar el SQL Server de Microsoft.

El Modelo Físico se usa entonces para plasmar la solución a nivel físico, en el caso de una base de datos, acá se tendrá que modelar de acuerdo al motor de base de datos que uses, por ejemplo, acá se tiene que colocar el tipo de datos de los conceptos lógicamente relacionados en los modelos anteriores (ejemplo: Char, String, Int, var, etc.)

DBMS, DATA BASE MANAGEMENT SYSTEM



Data Base Management System. Son las siglas en Ingles para los Sistemas de Gestión o Administración de Bases de Datos.

Es una aplicación que permite a los usuarios definir, crear y mantener una Base de Datos, además de proporcionar un acceso controlado a la misma.

Es una aplicación que interacciona con los usuarios de los programas de aplicación y la base de datos.

Sistema de Administración de Bases de Datos (DBMS).

Se conoce como Sistema de Administración de Base de Datos, o Sistema de Gestión de Base de Datos, o DataBase Management System (DBMS) al conjunto de hardware y software utilizado para almacenar y administrar una base de datos. En el mercado tenemos una variedad de DBMS's, desde aquellos diseñados para ser utilizados en computadoras personales hasta los creados para ser utilizados en entornos distribuidos:

- Microsoft Access
- MySQL
- PostgreSQL
- Microsoft SQL Server
- Sybase System
- DB2 Universal Database
- Oracle Database

Razones para usar un DBMS

Un DBMS evita dos grandes problemas que se presentan en los sistemas convencionales de almacenamiento de datos:

- Redundancia de los datos: cuando cada área de una organización maneja su propio almacenamiento de datos, la data suele duplicarse en cada una de las áreas que tienen relación entre sí.
- Inconsistencias en los datos: como consecuencia de la duplicidad de los datos, la actualización de un dato puede requerir que esta operación se ejecute en varios lugares. Si esto no se hace con cuidado, necesariamente se va a presentar inconsistencia en los datos, ya que algunos estarán actualizados y otros no.

Arquitectura de un DBMS.

Un DBMS se puede dividir en tres niveles:

- Nivel externo
- Nivel conceptual
- Nivel interno

Nivel externo

Es el nivel del SABD más cercano al usuario por lo que se le conoce también como la vista del usuario. El nivel externo define cómo el usuario ve los datos almacenados por el SABD.

Nivel conceptual

Es el nivel que define la base de datos tal como es y cómo la ve el DBA (Database Administrator). Este nivel define las relaciones entre los datos.

Nivel interno

Es el nivel que tiene que ver con el almacenamiento físico de los datos. Define cómo se guardan los datos en el medio físico.

DBMS'S MÁS COMERCIALES

A continuación, presentamos algunos de los DBMS's más comerciales.

ORACLE®

Oracle: Uno de los más reconocidos a nivel mundial como de los mejores DBMS's. Multiplataforma, confiable y seguro. Su administración puede ser un poco difícil, pero nada que no se pueda resolver.



DB2: Heredero de System R de IBM. Es un gran DBMS que ofrece una seguridad casi impenetrable cuando se instala sobre una máquina AS/400



Informix

Informix: Otra opción de IBM para el mundo empresarial que necesita un DBMS sencillo y confiable.



MS SQL Server: Es la opción de Microsoft para los usuarios de servidores NT o 2k. Posee muchos asistentes y herramientas de administración, aunque maneja el estándar a su antojo y tiene uno de los lenguajes de procedimientos más difíciles para programar.



SyBase: Un DBMS con bastantes años en el mercado, tiene 3 versiones para ajustarse a las necesidades reales de cada empresa.



D3: Un nuevo DBMS que pretende abrirse camino con características novedosas

InterBase®

InterBase: Pequeño y sin hacer mucho ruido está metido en varias organizaciones importantes como: Nokia, US Army, Compañías Telefónicas de Canadá y Francia, etc. Es fácil de instalar y usar. Seguro y Robusto. Soporte Técnico tanto con Borland como con otras compañías especializadas.



Progress: Otro DBMS con años en el mercado, con una nueva cara y un nuevo espíritu. Está enfocado a los grandes negocios.

El DBMS Microsoft SQLServer

En el desarrollo de este curso usaremos el DBMS DE Microsoft; SQL Server

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son El Transac SQL (T-SQL) y ANSI SQL.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL.

USO DE SQLSERVER y CREACIÓN DE UNA BASE DE DATOS.



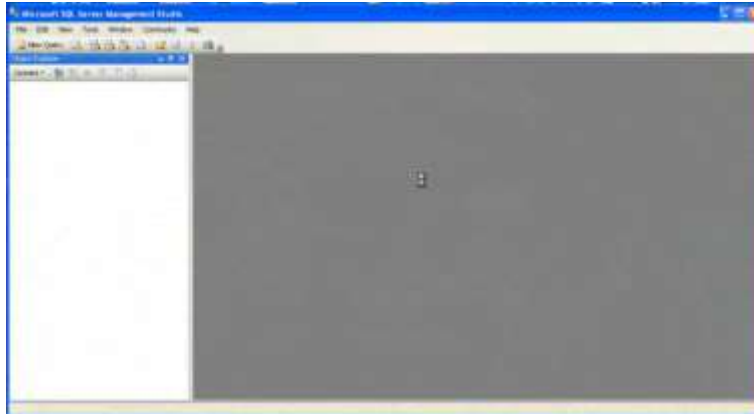
Creación de una Base de datos.

Hacer click en el icono de acceso directo SQLServer Management Studio



Aparece la ventana de Ingreso





Y luego nos pide elegir el tipo de autenticación con el que se va a ingresar

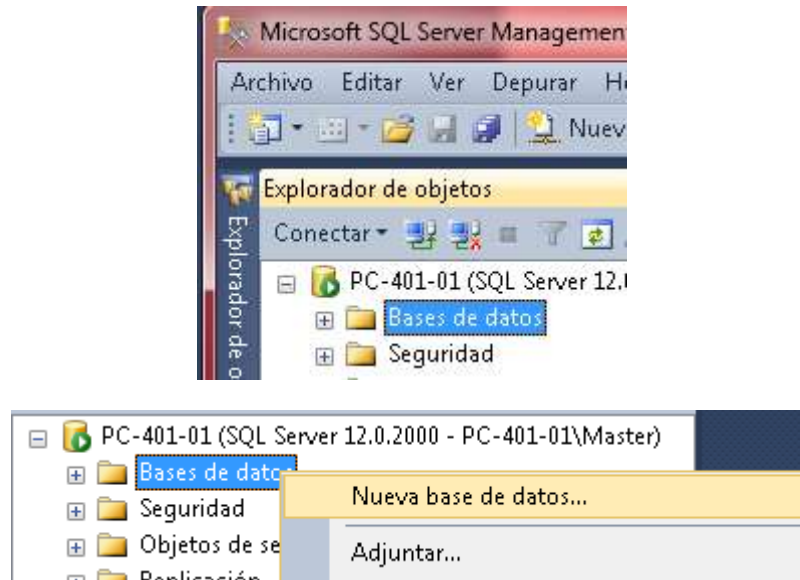


En este caso elegimos la Autenticación Windows (por defecto)

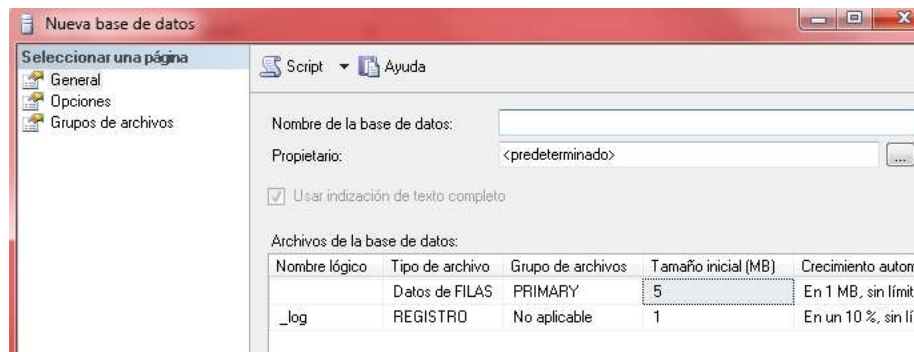


CREACIÓN DE UN ESPACIO DE BASE DE DATOS

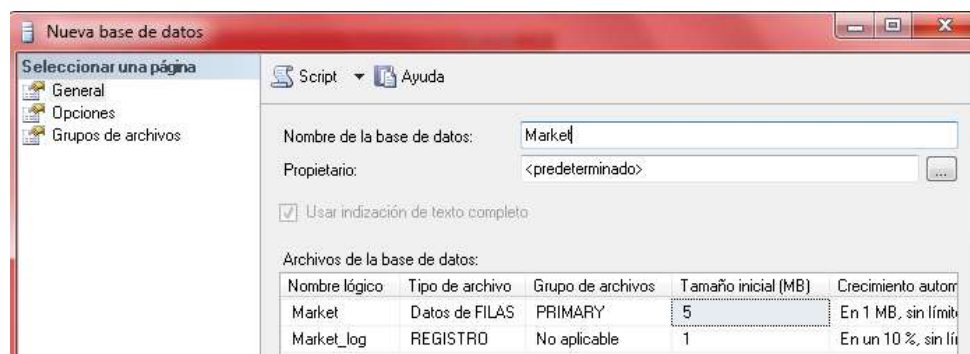
Para crear una nueva Base de Datos Nos ubicamos en el Explorador de Objetos de la izquierda y hacemos click con derecha en la carpeta de Databases

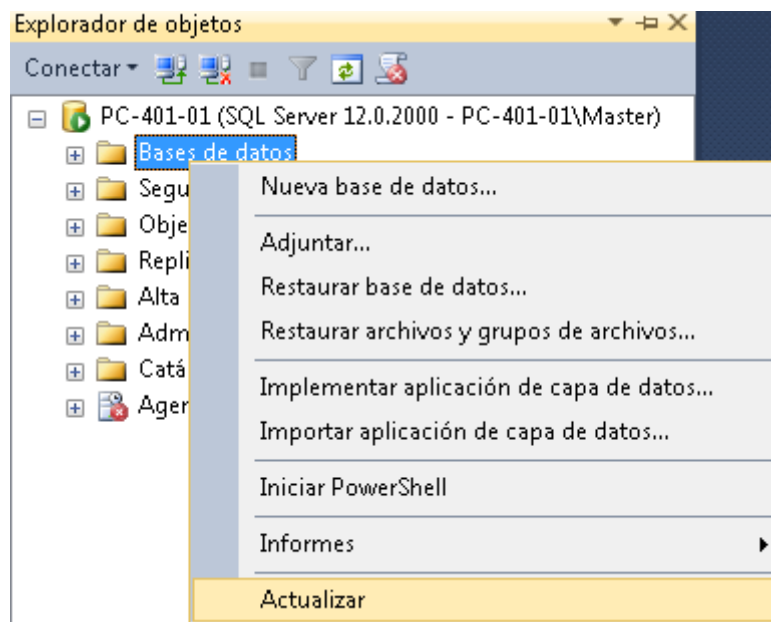


Elegimos (Nueva Base de datos)

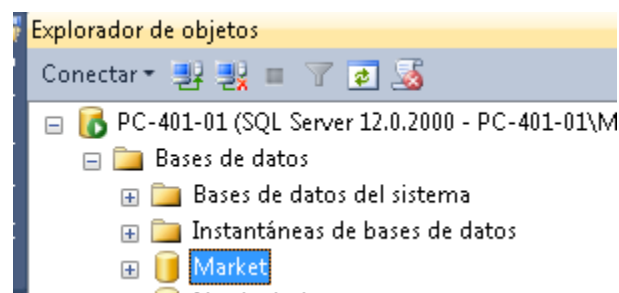


En la ventana de New Database digitamos el nombre de la nueva Base de datos que vamos a crear, en este caso Market





Para verificar la Base de datos nueva hacemos click con derecha en Databases y elegimos (Actualizar)



Y ya podemos visualizar la nueva Base de datos que hemos creado (Market).

TIPOS DE DATOS MÁS USADOS EN SQL SERVER

Char(n).

Almacena n caracteres en formato ASCII, un byte por cada letra. Cuando almacenamos datos en el tipo char, siempre se utilizan los n caracteres indicados, incluso si la entrada de datos es inferior. Por ejemplo, si en un char (5), guardamos el valor 'A', se almacena 'A ', ocupando los cinco bytes.

Varchar(N)

Se usa para almacenar cadenas de caracteres. Una cadena es una secuencia de caracteres. Se coloca entre comillas (simples); ejemplo: 'Hola', 'Juan Pérez'. El tipo "varchar" define una cadena de longitud variable en la cual determinamos el máximo de caracteres entre paréntesis. Puede guardar hasta 8000 caracteres. Por ejemplo, para almacenar cadenas de hasta 30 caracteres, definimos un campo de tipo varchar(30), es decir, entre paréntesis, junto al nombre del campo colocamos la longitud.

Si asignamos una cadena de caracteres de mayor longitud que la definida, la cadena no se carga, aparece un mensaje indicando tal situación y la sentencia no se ejecuta.

Por ejemplo, si definimos un campo de tipo varchar(10) e intentamos asignarle la cadena 'Buenas tardes', aparece un mensaje de error y la sentencia no se ejecuta.

Integer

Se usa para guardar valores numéricos enteros, de -2000000000 a 2000000000 aprox. Definimos campos de este tipo cuando queremos representar, por ejemplo, cantidades.

float:

Se usa para almacenar valores numéricos con decimales. Se utiliza como separador el punto (.). Definimos campos de este tipo para precios, por ejemplo.

Tinyint:

Una columna o variable de tipo tinyint puede almacenar el rango de valores de 0 a 255.

Decimal (p, s).

Una columna de tipo decimal puede almacenar datos numéricos decimales sin redondear. Donde p es la precisión (número total de los dígitos) y s la escala (número de valores decimales)

Money:

Almacena valores numéricos monetarios de -263 a 263-1, con una precisión de hasta diez milésimas de la unidad monetaria.

Datetime:

Almacena fechas con una precisión de milisegundo. Debe usarse para fechas muy específicas. El formato que presenta es de fecha y hora (0:00:00). Ocupa 8 bytes

Existen por supuesto más tipos, pero solo mencionaremos por ahora los que necesitaremos para los ejemplos que se van a desarrollar.

Antes de crear una tabla debemos pensar en sus campos y optar por el tipo de dato adecuado para cada uno de ellos.

Por ejemplo, si en un campo almacenaremos números enteros, el tipo "float" sería una mala elección; si vamos a guardar precios, el tipo "float" es más adecuado, no así "integer" que no tiene decimales. Otro ejemplo, si en un campo vamos a guardar un número telefónico o un número de documento, usamos "varchar", no "integer" porque si bien son dígitos, con ellos no realizamos operaciones matemáticas.

En el desarrollo de las Aplicaciones que sirven para manipular las Bases de datos nos encontramos con Bases de Datos

USO DE LA INTERFACE GRÁFICA DEL SQL SERVER PARA LA CREACIÓN DE UNA BASE DE DATOS.

Creación de una Base de datos



Modelo Físico de ejemplo



Creación de la Base de datos correspondiente a este Modelo Físico.

Hacer click en el icono de acceso directo SQLServer Management Studio que puede estar en el escritorio



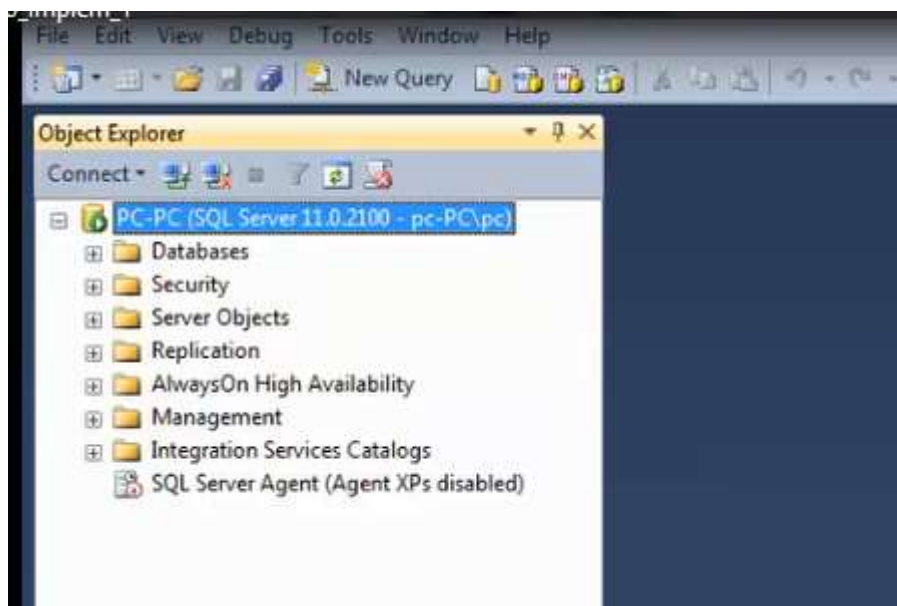
O en la Barra de Herramientas en la parte inferior:

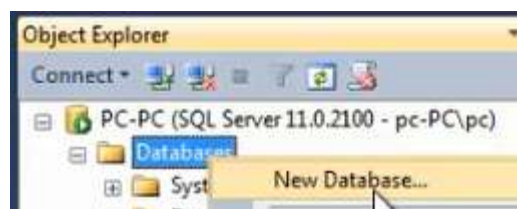
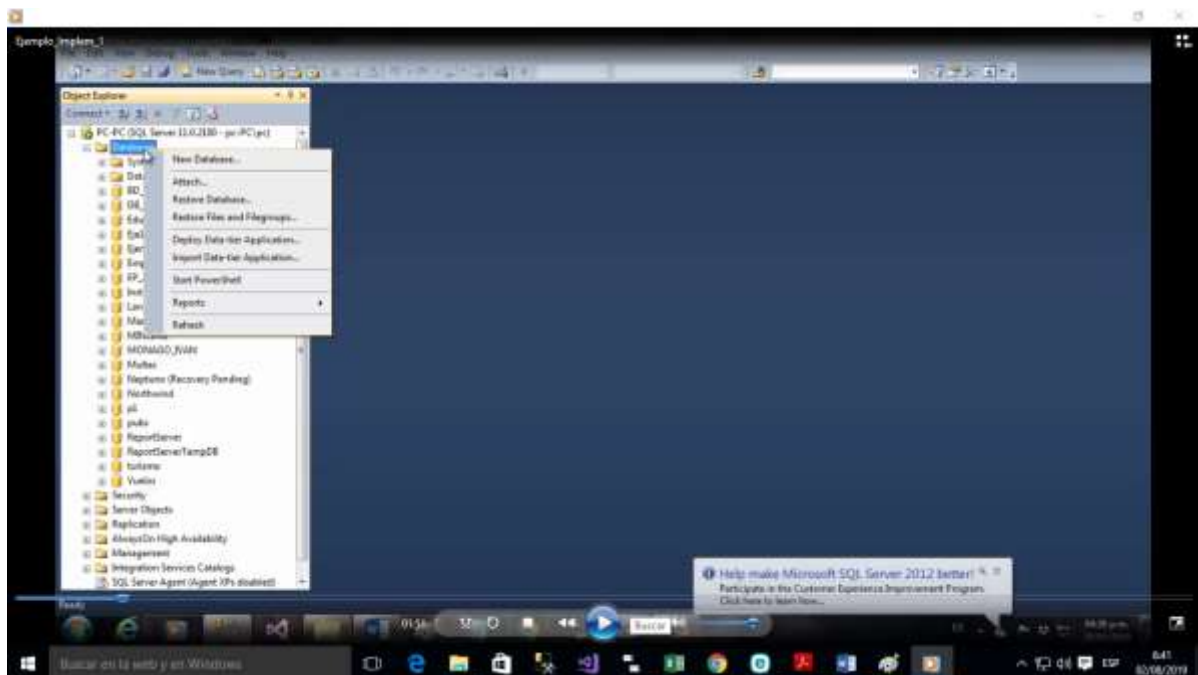


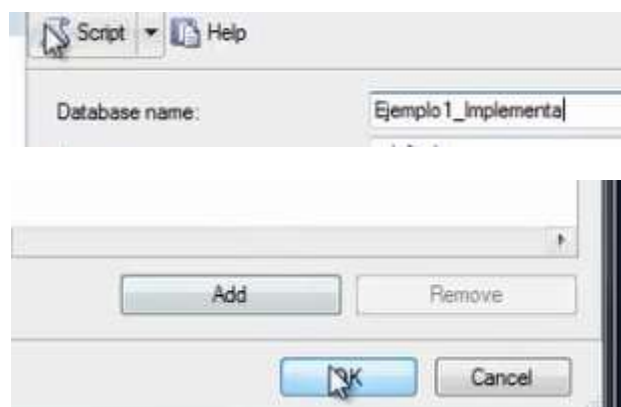
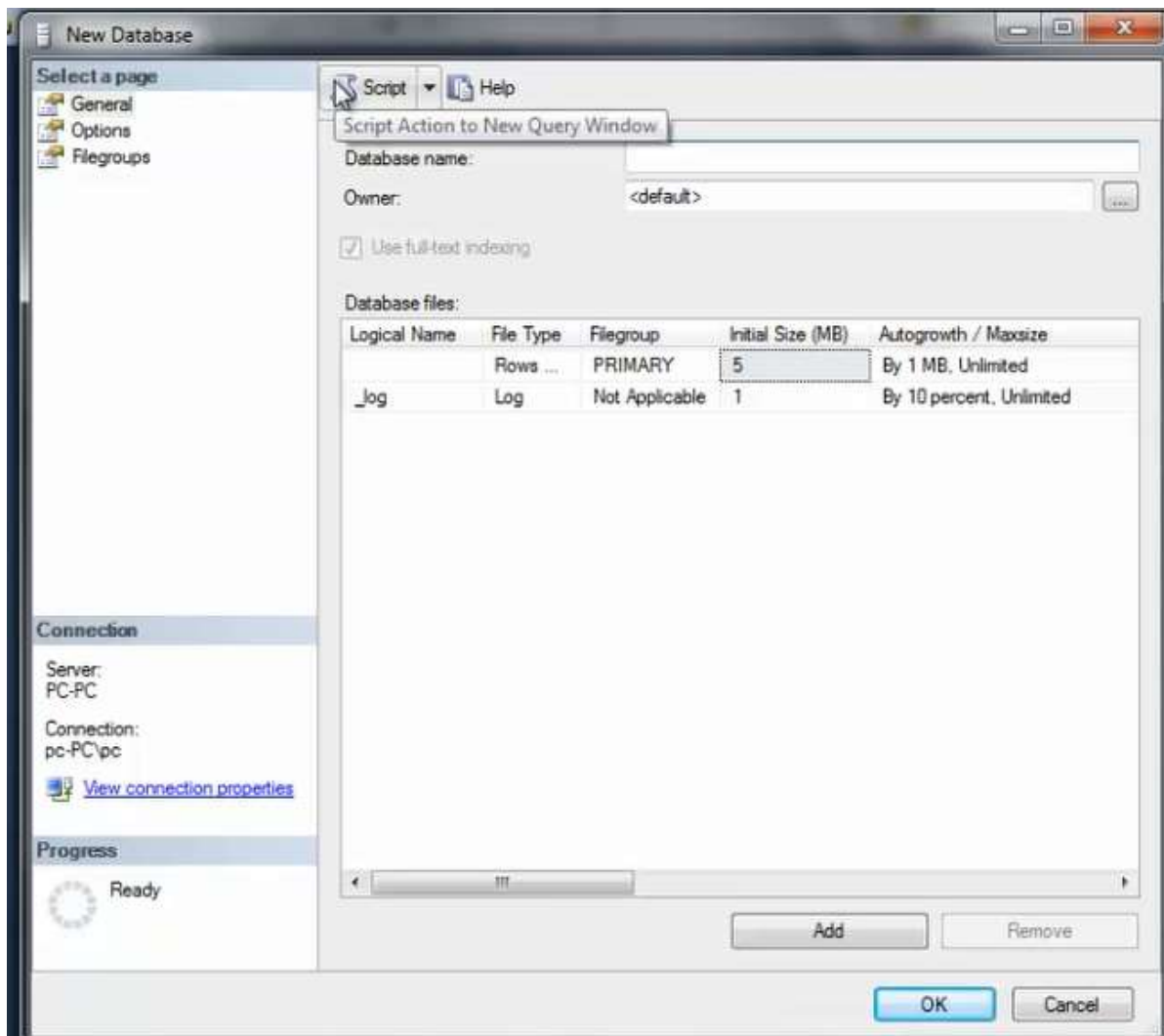
Luego aparece la ventana de identificación de usuario:

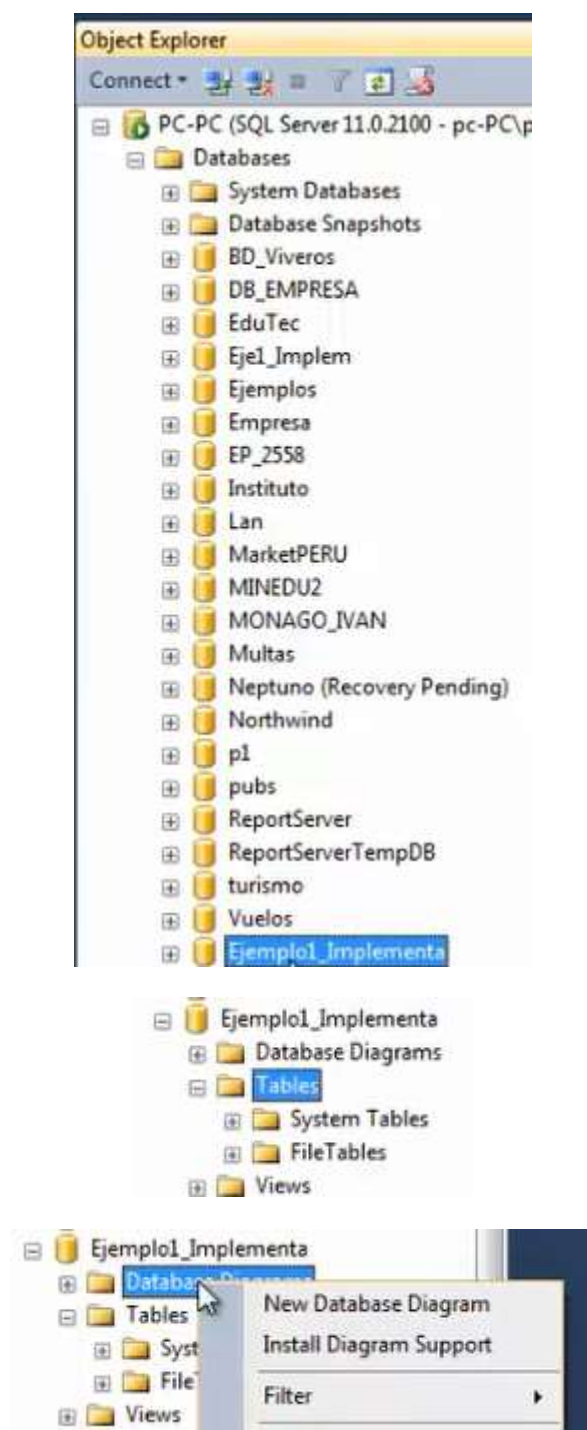


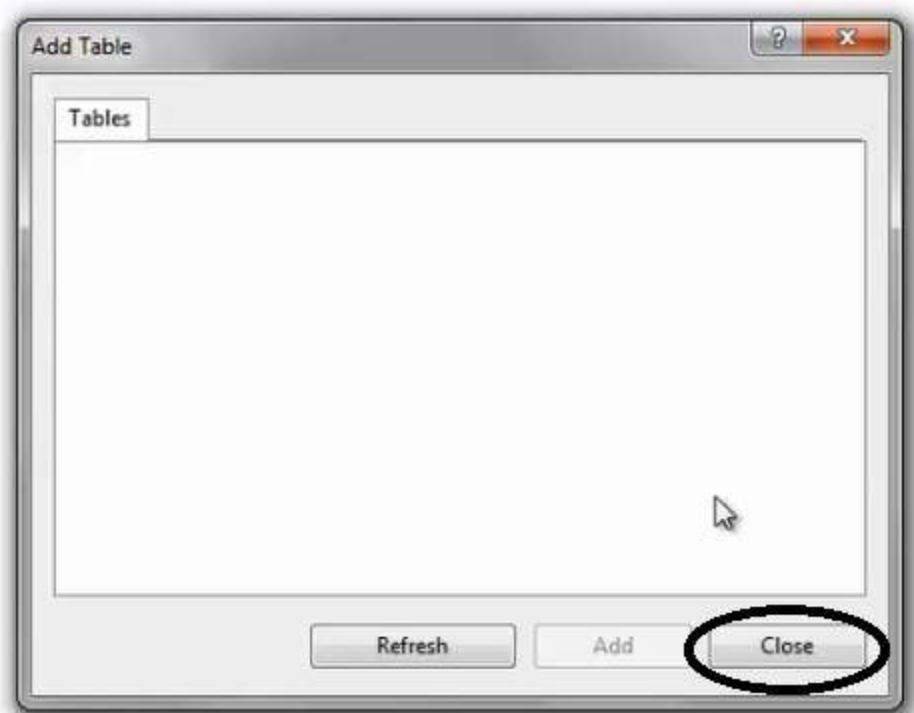
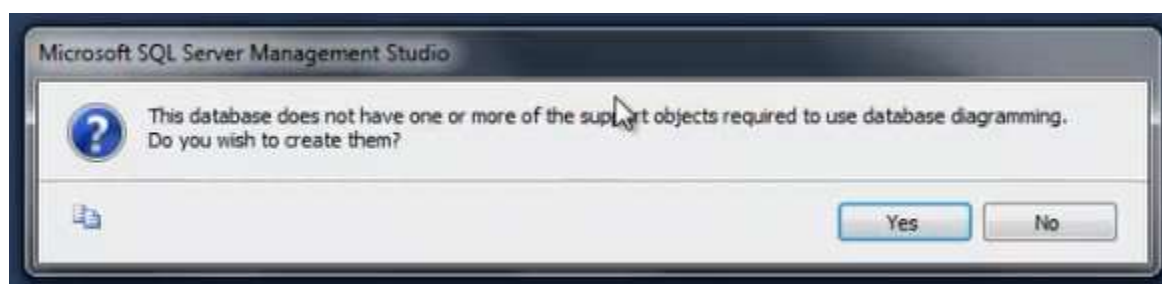
Hacemos click en Connect (Conectar)

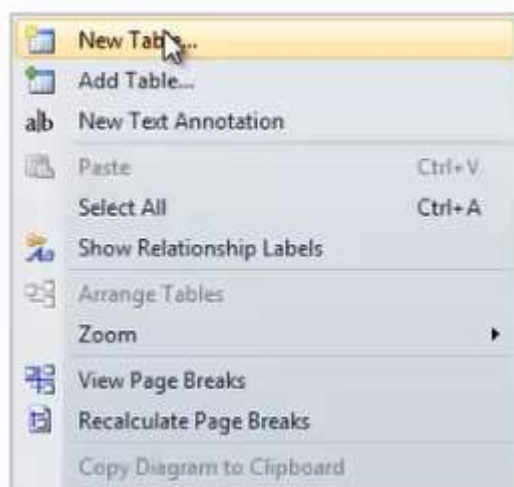
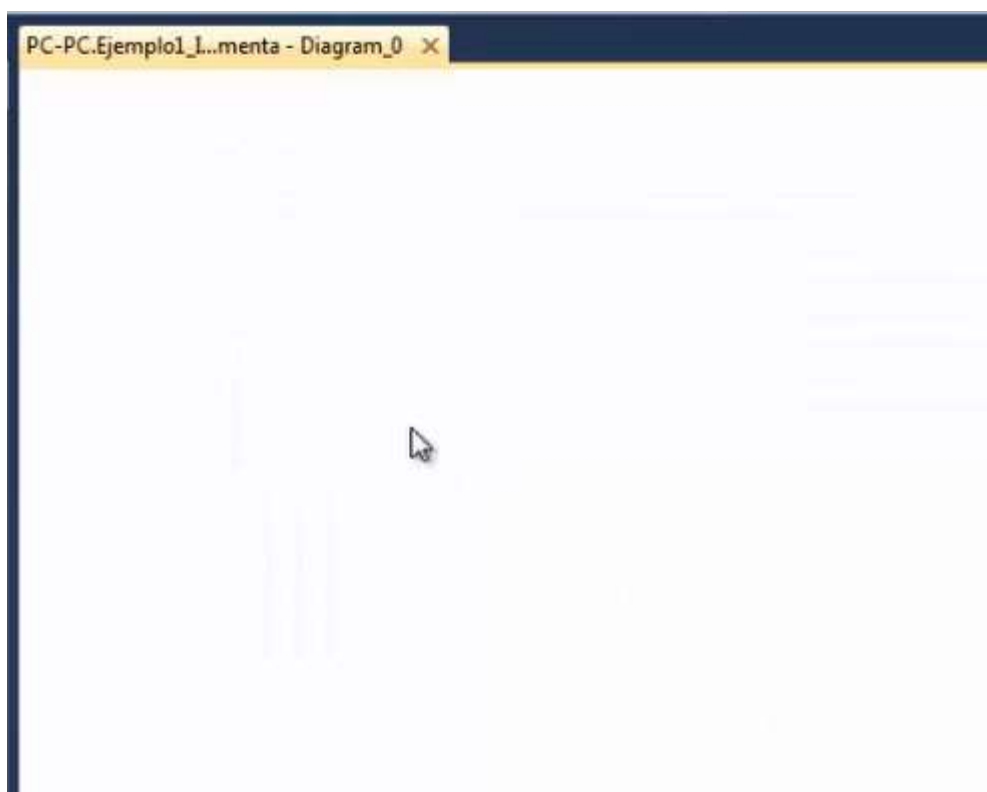


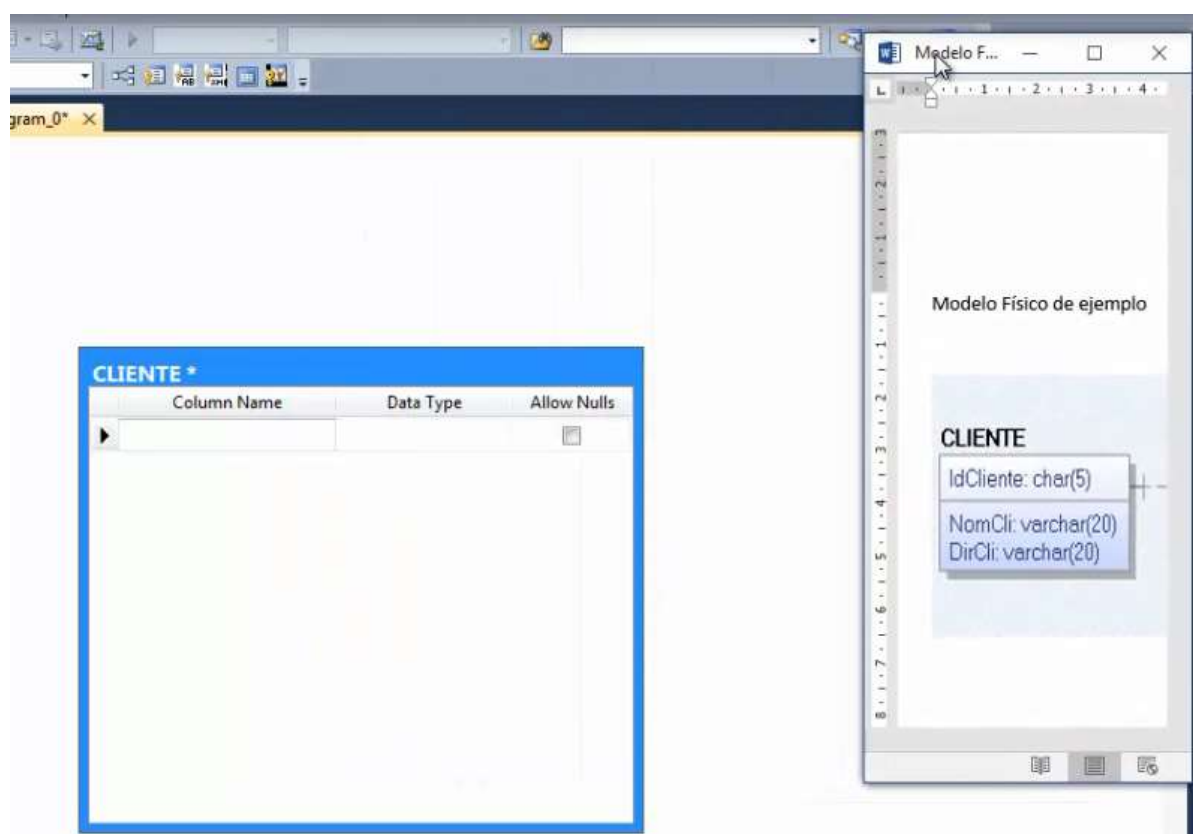












CLIENTE *

Column Name	Data Type	Allow Nulls
► IdCliente	nchar(10)	<input checked="" type="checkbox"/>
	binary(50)	<input type="checkbox"/>
	bit	<input type="checkbox"/>
	char(10)	<input type="checkbox"/>
	date	<input type="checkbox"/>
	datetime	<input type="checkbox"/>
	datetime2(7)	<input type="checkbox"/>
	datetimeoffset(7)	<input type="checkbox"/>
	decimal(18, 0)	<input type="checkbox"/>

CLIENTE *

Column Name	Data Type	Allow Nulls
► IdCliente	char(5)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

CLIENTE *

Column Name	Data Type	Allow Nulls
► IdCliente	char(5)	<input type="checkbox"/>
		<input type="checkbox"/>

CLIENTE *

Column Name	Data Type	Allow Nulls
IdCliente	char(5)	<input type="checkbox"/>
►		<input type="checkbox"/>

CLIENTE *

Column Name	Data Type	Allow Nulls
IdCliente	char(5)	<input type="checkbox"/>
NomCli	varchar(20)	<input type="checkbox"/>
► DirCli	varchar(20)	<input type="checkbox"/>
		<input type="checkbox"/>

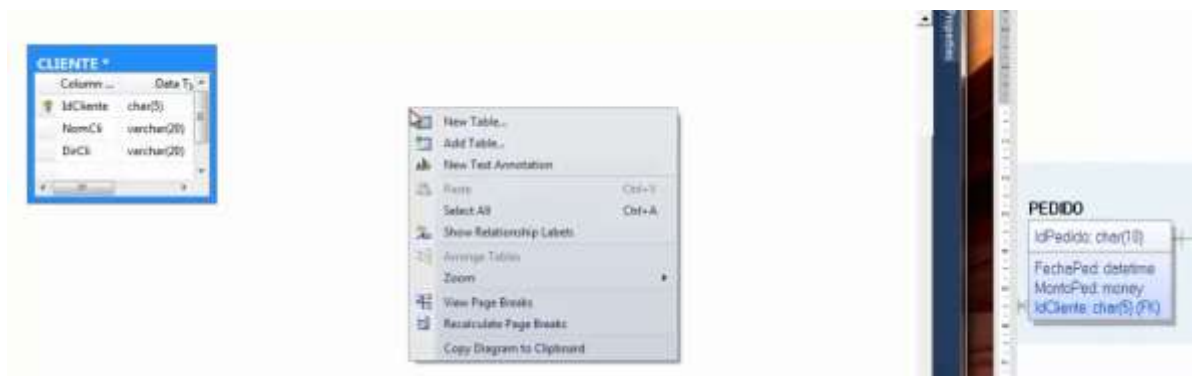
CLIENTE

IdCliente: char(5)
NomCli: varchar(20)
DirCli: varchar(20)



CLIENTE *			
	Column Name	Data Type	Allow Nulls
	IdCliente	char(5)	<input type="checkbox"/>
	NomCli	varchar(20)	<input type="checkbox"/>
	DirCli	varchar(20)	<input type="checkbox"/>

Ahora continuamos con la Tabla PEDIDO de acuerdo al Modelo Físico



De manera similar creamos los campos:

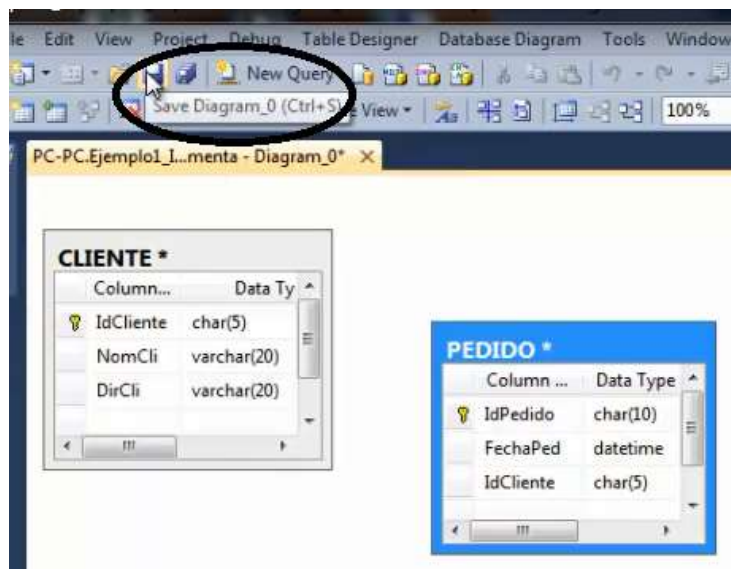


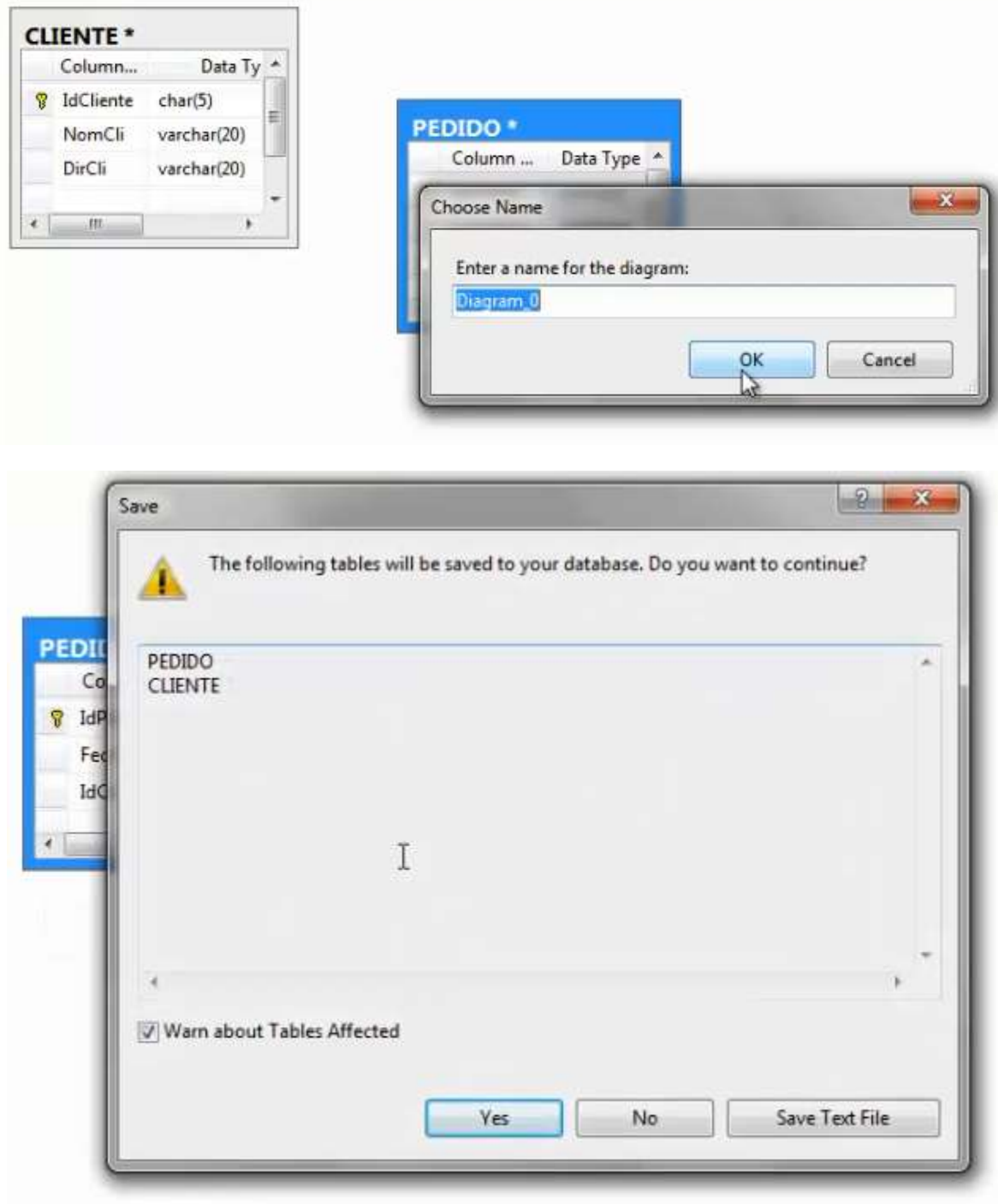
Redimensionamos y ubicamos la Tabla PEDIDO



Ahora vamos a ingresar la Tabla DETALLE_PEDIDO.

Hay que ir grabando de rato en rato, para que no se vayan a perder los cambios que vamos haciendo en la Base datos



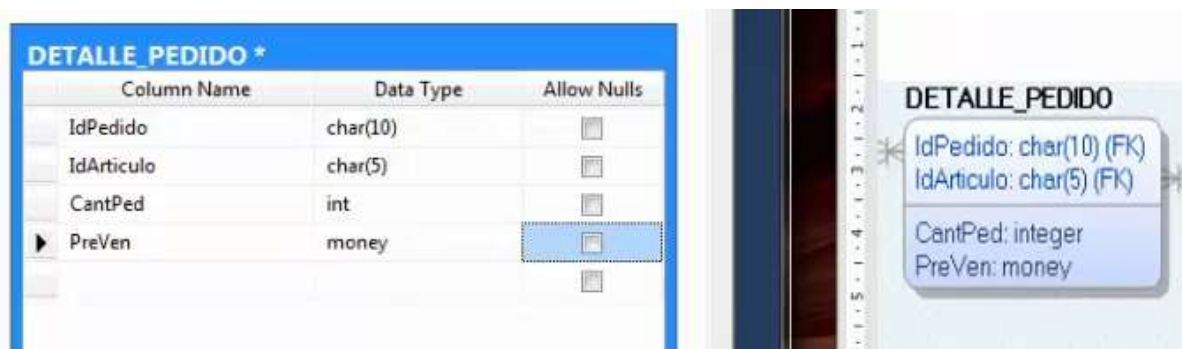


Aceptamos con el botón Yes

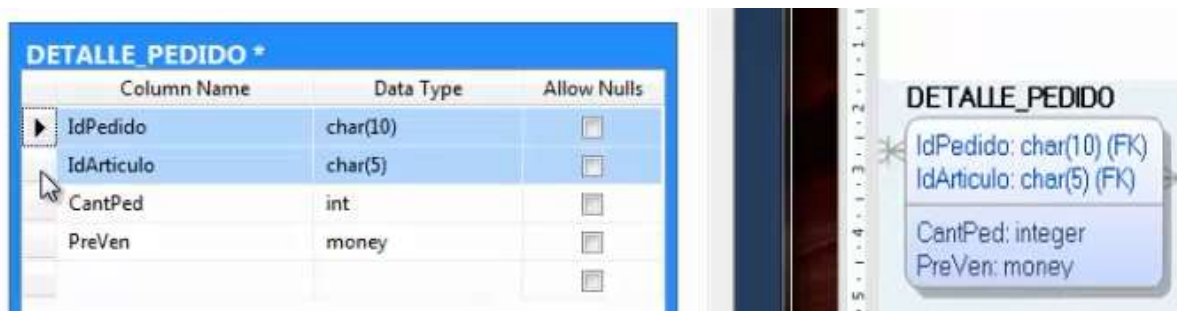
Ahora ingresamos la tabla DETALLE_PEDIDO



Luego ingresamos los campos



Como la PK es compuesta seleccionamos los dos primeros campos a la vez para definir la PK compuesta:



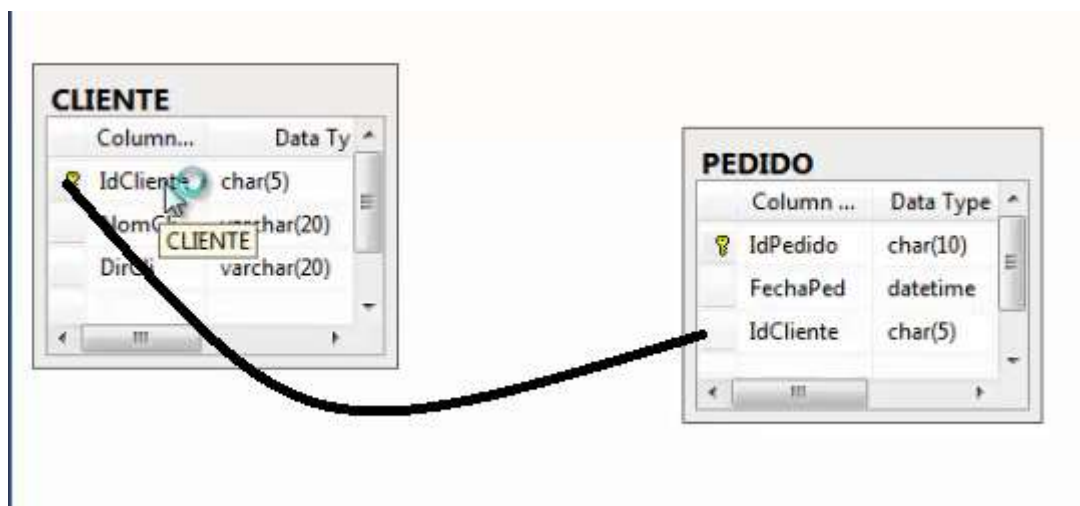
DETALLE_PEDIDO *			
Column Name	Data Type	Allow Nulls	
IdPedido	char(10)	<input type="checkbox"/>	
IdArticulo	char(5)	<input type="checkbox"/>	
CantPed	int	<input type="checkbox"/>	
PreVen	money	<input type="checkbox"/>	

Queda así definida la PK compuesta por los campos IdPedido + IdArticulo

De forma análoga seguimos agregando las Tablas que faltan, y tenemos entonces:



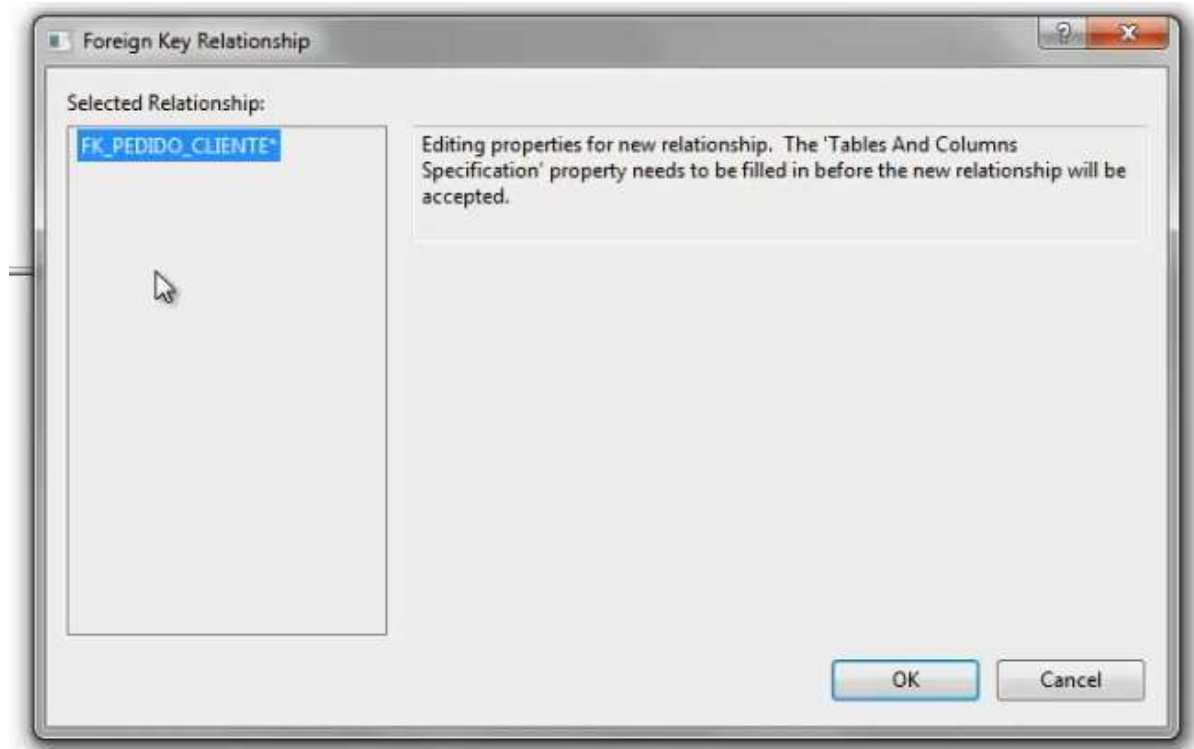
Relacionando las Tablas:



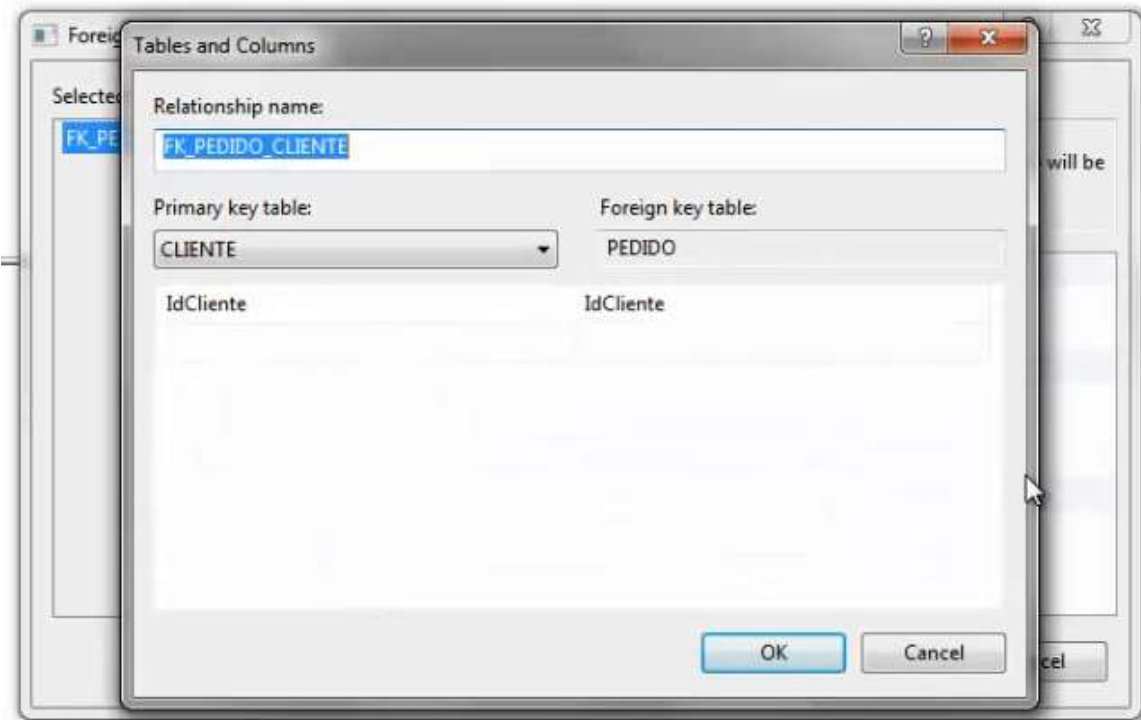
Para relacionar CLIENTE con PEDIDO, hacemos un arrastre con el Mouse desde la PK de la Tabla CLIENTE hasta el campo IdCliente de la Tabla PEDIO, para que este se convierta en Clave Foránea (FK).

Luego obtenemos la relación como sigue:

Aparece la ventana

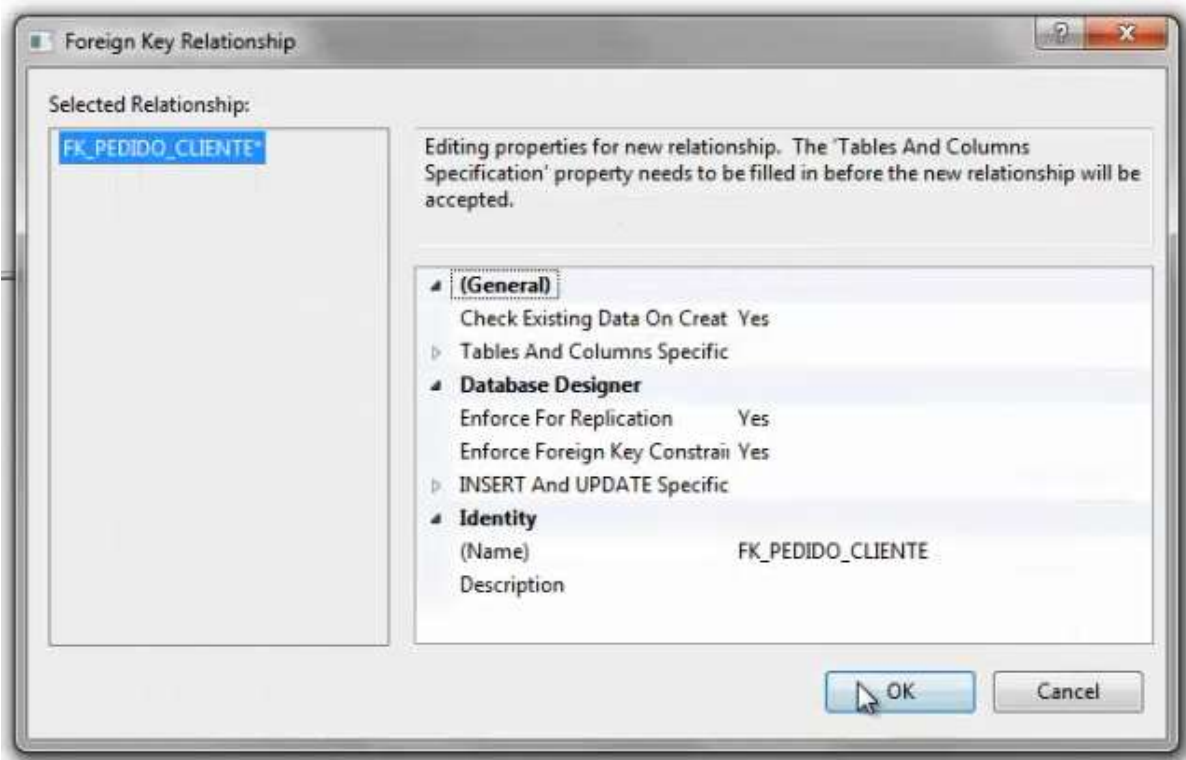


Hacemos click en OK, y Luego aparece la Siguiete Ventana:

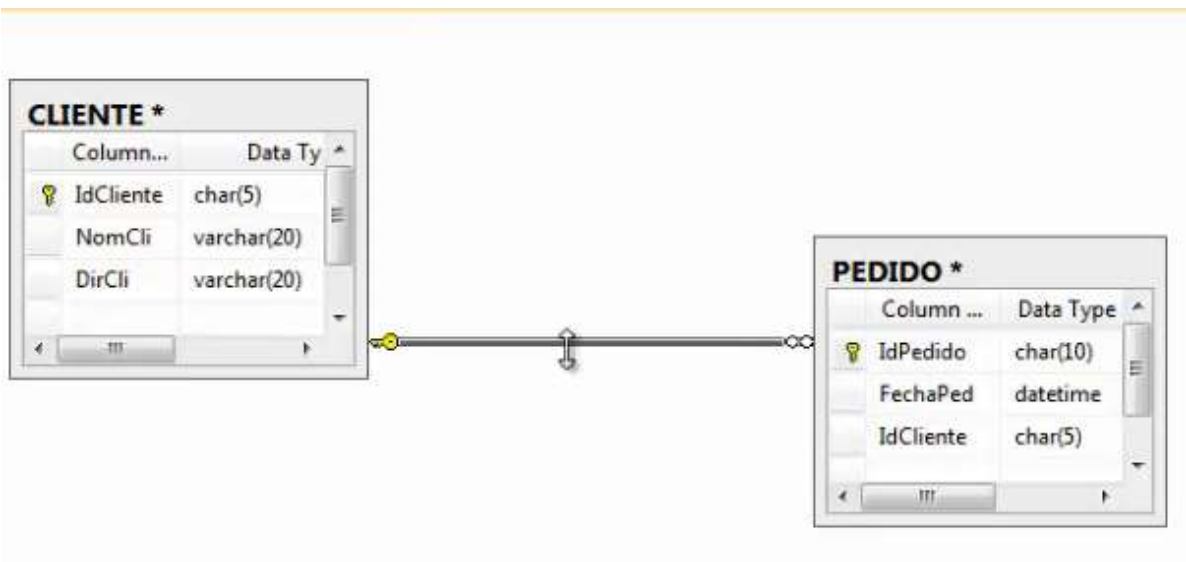


Se verifica que partimos de la Tabla CLIENTE desde la PK IdCliente y hemos llegado al campo IdCliente de la Tabla PEDIDO para que se convierta en Clave Foránea (FK) para finalmente establecer la relación.

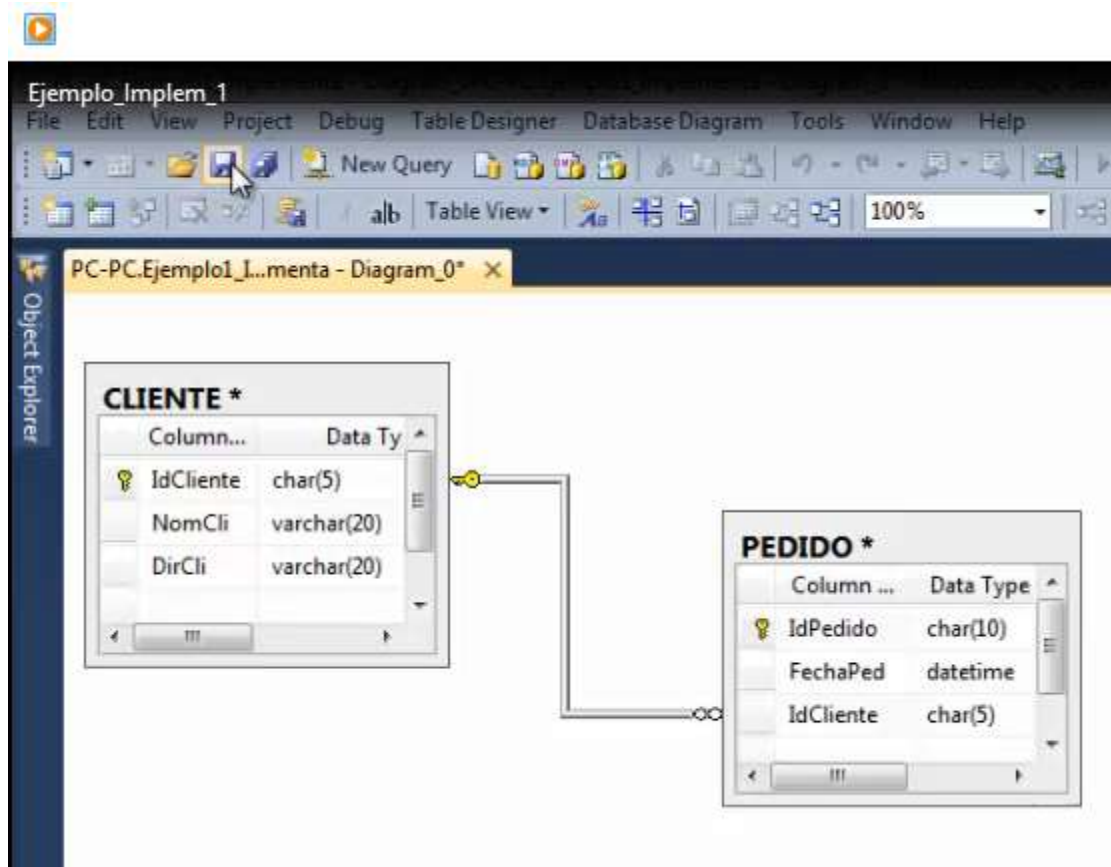
Hacemos Click en OK



Esta ventana nos confirma que ya se estableció la relación



La relación ya está establecida, pero falta alinear bien la relación para que se apunte bien a la PK y a la FK.

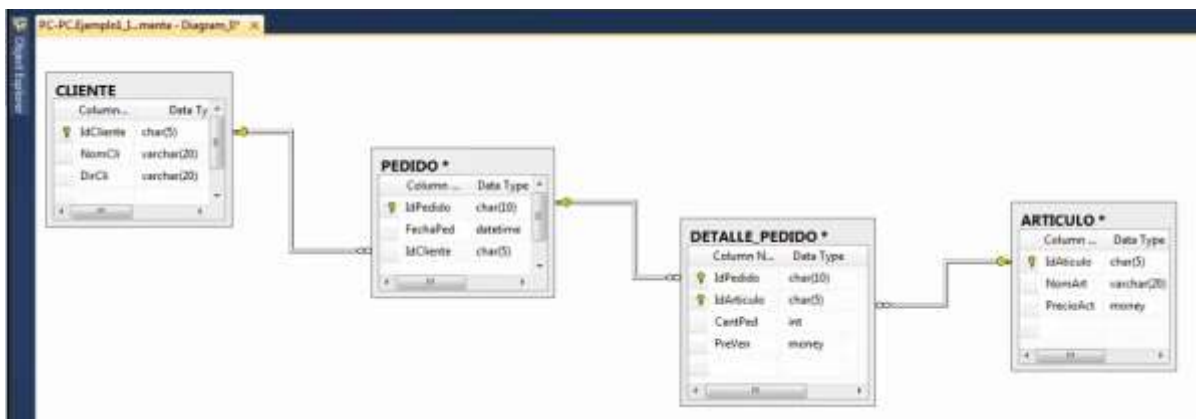


Y guardamos los cambios

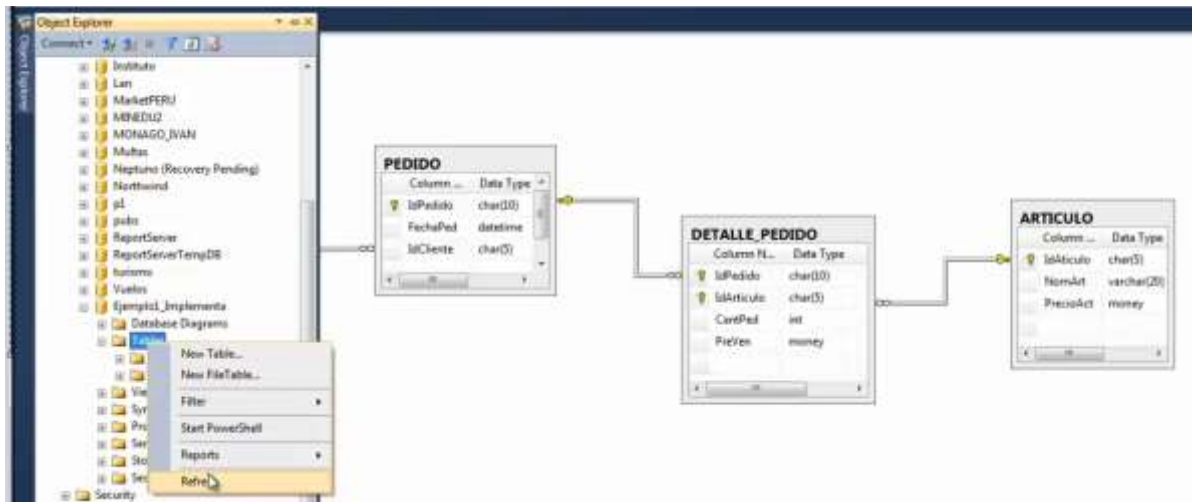


De manera similar terminamos las otras relaciones

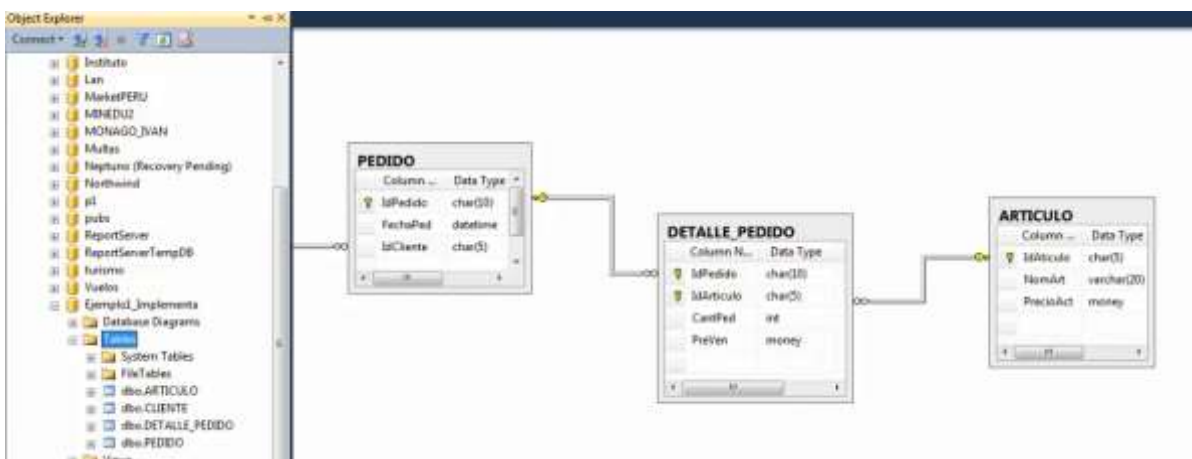
Luego tenemos el diagrama



Vemos el Explorador de Objetos de la parte izquierda y refrescamos la Carpeta de Tablas para que se actualice la imagen y se muestren la Tablas que hemos creado.



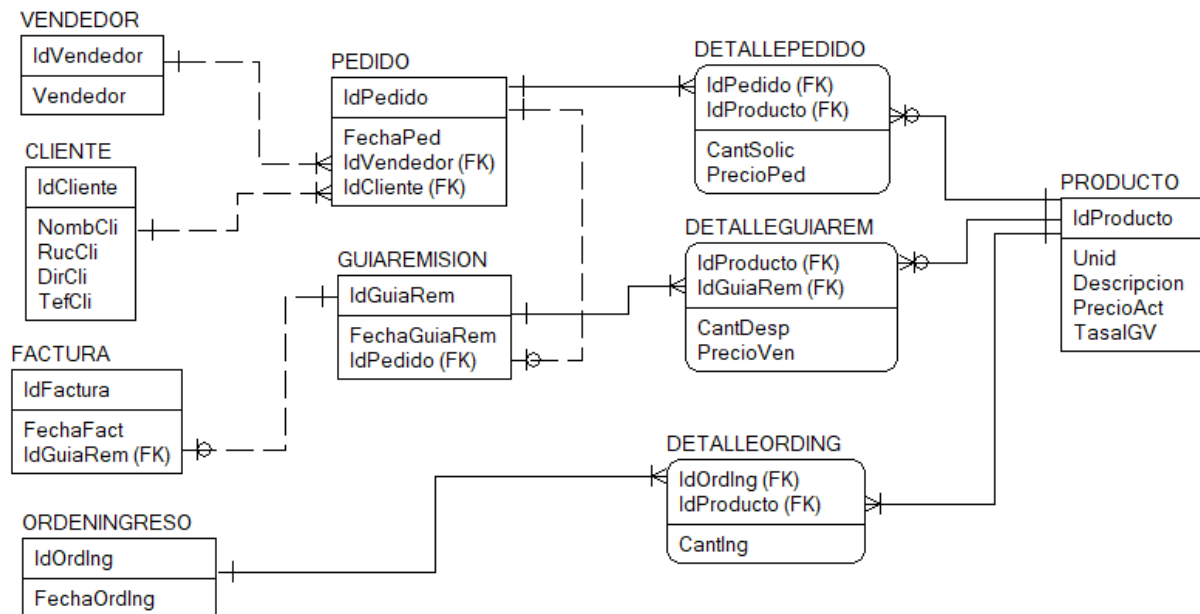
Finalmente, ya tenemos la Base de Datos implementada en el servidor a partir de un Modelo Físico del cual partimos.



No olvidar ir grabando de rato en rato para que no se pierdan los cambios que vamos introduciendo al Diagrama de la Base de Datos.

Ejercicio-de- Practica:

En el caso del ejemplo del Proyecto3 de Normalización, el modelo Lógico se debe convertir al modelo físico correspondiente e implementar la base de datos en SQL Server



PRACTICAS CALIFICADA

PRACTICA CALIFICADA 1

Proyecto 1 – Clínica “Santo Tomas”



A partir de la siguiente realidad propuesta, diseñar el modelo conceptual correspondiente:

La clínica “SANTO TOMAS” necesita llevar un control informatizado de su gestión de pacientes y médicos.

De cada paciente se desea guardar el código, nombre, apellidos, dirección, población, provincia, código postal, teléfono y fecha de nacimiento.

De cada médico se desea guardar el código, nombre, apellidos, teléfono y especialidad.

Se desea llevar el control de cada uno de los ingresos que el paciente hace en el hospital.

Cada ingreso que realiza el paciente queda registrado en la base de datos. De cada ingreso se guarda el código de ingreso (que se incrementará automáticamente cada vez que el paciente realice un ingreso), el número de habitación y cama en la que el paciente realiza el ingreso y la fecha de ingreso.

Un médico puede atender varios ingresos, pero el ingreso de un paciente solo puede ser atendido por un único médico. Un paciente puede realizar varios ingresos en la clínica.

Proyecto 2 – Envío de Paquetes



A partir del siguiente enunciado se desea realizar el modelo conceptual correspondiente.

Se desea informatizar la gestión de una empresa de transportes que reparte paquetes por todo el País. Los encargados de llevar los paquetes son los camioneros, de los que se quiere guardar el DNI, nombre, teléfono, dirección, salario y población en la que vive. De los paquetes transportados interesa conocer el código de paquete, descripción, destinatario y dirección del destinatario. Un camionero distribuye muchos paquetes, y un paquete sólo puede ser distribuido por un camionero.

De las provincias a las que llegan los paquetes interesa guardar el código de provincia y el nombre. Un paquete sólo puede llegar a una provincia. Sin embargo, a una provincia pueden llegar varios paquetes.

De los camiones que llevan los camioneros, interesa conocer la matrícula, modelo, tipo y potencia. Un camionero puede conducir diferentes camiones en fechas diferentes, y un camión puede ser conducido por varios camioneros.

Proyecto 3 – Alquiler de Películas



La cadena de Video-Clubs Glob-Gusters ha decidido, para mejorar su servicio, emplear una base de datos para almacenar la información referente a las películas que ofrece en alquiler. Esta información es la siguiente:

Una película se caracteriza por su título, nacionalidad, productora y fecha (p.e., “Quo Vadis”, “Estados Unidos”, “M.G.M.”, 1955).

En una película pueden participar varios actores (nombre, nacionalidad, sexo) algunos de ellos como actores principales.

Una película está dirigida por un director (nombre, nacionalidad).

De cada película se dispone de uno o varios ejemplares diferenciados por un número de ejemplar y caracterizados por su estado de conservación.

Un ejemplar se puede encontrar alquilado a algún cliente (nombre, dirección, teléfono). Se desea almacenar la fecha de comienzo del alquiler y la de devolución.

Cada socio puede alquilar como máximo 4 ejemplares.

Un socio tiene que ser avalado por otro socio, que responda de él en caso de tener problemas en el alquiler.

Proyecto 4 – Empresa de Transporte de pasajeros



Una empresa de transporte de pasajeros requiere se le diseñe una Base de Datos que cumpla con los siguientes requerimientos:

La empresa se dedica al transporte interprovincial de pasajeros entre las principales ciudades del país.

Todas sus rutas son directas; es decir, que partiendo de la ciudad de origen se dirigen sin escalas a la ciudad destino.

Cada ruta tiene una ciudad de origen, una hora de salida programada, una hora de salida real, una ciudad destino, una hora de llegada programada, una hora de llegada real. Para cada ruta se tiene dos tipos de pasajes: Estándar (asiento pulman) y Premium (asiento-cama).

Las unidades de transporte se han adquirido de distintos fabricantes y tienen distinta capacidad e pasajeros. Hay unidades para 30 pasajeros, unidades para 40 pasajeros, etc.

Para cada viaje se asignan 2 choferes. Además, en cada viaje se tiene a bordo a una azafata para la atención a los pasajeros.

En los controles de salida y entrada de las ciudades se debe entregar una relación de pasajeros que van en el viaje, así como de la tripulación.

La BD debe permitir el control de la venta de pasajes, de las unidades de transporte utilizadas, de la tripulación asignada, y del viaje. Presentar el modelo de datos conceptual.

Proyecto 5 – Venta de automóviles



A partir del siguiente supuesto diseñar el modelo conceptual:

Se desea diseñar una base de datos para almacenar y gestionar la información empleada por una empresa dedicada a la venta de automóviles, teniendo en cuenta los siguientes aspectos:

La empresa dispone de una serie de coches para su venta. Se necesita conocer la matrícula, marca y modelo, el color y el precio de venta de cada coche.

Los datos que interesa conocer de cada cliente son el RUC, nombre, dirección, ciudad y número de teléfono: además, los clientes se diferencian por un código interno de la empresa que se incrementa automáticamente cuando un cliente se da de alta en ella. Un cliente puede comprar tantos coches como desee a la empresa. Un coche determinado solo puede ser comprado por un único cliente.

El concesionario también se encarga de llevar a cabo las revisiones que se realizan a cada coche. Cada revisión tiene asociado un código que se incrementa automáticamente por cada revisión que se haga. De cada revisión se desea saber si se ha hecho cambio de filtro, si se ha hecho cambio de aceite, si se ha hecho cambio de frenos u otros. Los coches pueden pasar varias revisiones en el concesionario.

Proyecto 6 – Venta de Flores



Una empresa que se dedica a la floristería desea ampliar su negocio y realizar ventas a través de Internet.

Para cumplir con este objetivo se necesita crear una base de datos.

Cada pedido incluye un número de pedido, la fecha de venta, el importe total y una lista con las flores solicitadas y en qué cantidad.

Las flores se identifican mediante un código, su nombre y el precio de venta.

Las flores pertenecen a una especie determinada.

Para cada especie, se almacena el nombre, la época de floración, la estación de plantación, el tipo de suelo apropiado y el tiempo de recomendado.

Cada pedido es atendido por un vendedor, de los cuales se necesita guardar sus datos como Apellidos, nombres, dirección, dni y teléfono.

Cada pedido es atendido por un solo vendedor.

Un vendedor puede haber atendido muchos pedidos, pero en el caso de uno nuevo, podría ser que no haya atendido ningún pedido aún.

Cada pedido es luego enviado a la dirección que indique el cliente; estos envíos tienen un costo que se le incluye también al cliente.

Estos envíos los realizan otras empresas dedicadas a estas labores. Se necesita guardar datos de cada empresa de envíos como su nombre, teléfono y dirección.

Crear el Modelo Conceptual que represente esta realidad.

Los Pedidos corresponden a los clientes que los solicitan. Un pedido solo le corresponde a un solo cliente, pero un cliente puede realizar muchos pedidos. Se desea guardar información de los clientes como sus Apellidos, Nombre, Dirección, teléfono, etc.

Proyecto 7 – Biblioteca



Se requiere diseñar una base de datos que guarde la información los préstamos de libros que se realizan en una biblioteca; las reglas son la siguientes:

Los autores tienen un código que los identifica unívocamente, también tienen nombres y apellidos.

Un libro puede tener un ejemplar o varios ejemplares, cada ejemplar tiene un código que los identifica.

Un Lector puede realizar muchos prestamos (en diferentes fechas)

Pueden existir lectores que no han realizado ningún préstamo aún.

Se deben guardar datos del lector como código, Apellidos, nombres, dni, dirección y teléfono.

En el préstamo se debe registrar el ejemplar que se presta, la fecha del préstamo y la fecha de devolución.

Un préstamo de un ejemplar de un libro lo realiza un solo lector.

Un Ejemplar puede prestarse muchas veces, pero podría existir un ejemplar que no haya sido prestado aún.

Cada libro tiene un código que los identifica, un título, cantidad de páginas.

Un Autor puede escribir un libro o muchos libros, y un libro puede tener un autor o varios autores, No se registran libros sin autor ni autores sin libros relacionados.

PRACTICA CALIFICADA 2

Proyecto1 – Alquiler de coches



Se desea diseñar una base de datos que guarde la información de las reservas de una empresa dedicada al alquiler de automóviles. Los supuestos semánticos son los siguientes:

Un determinado cliente puede tener en un momento dado varias reservas.

Una reserva la realiza un único cliente, pero puede involucrar a varios coches.

Es importante registrar la fecha de comienzo de la reserva y la de terminación.

Todo coche tiene siempre asignado un número determinado de garaje, que no puede cambiar.

Cada reserva se realiza en una determinada agencia.

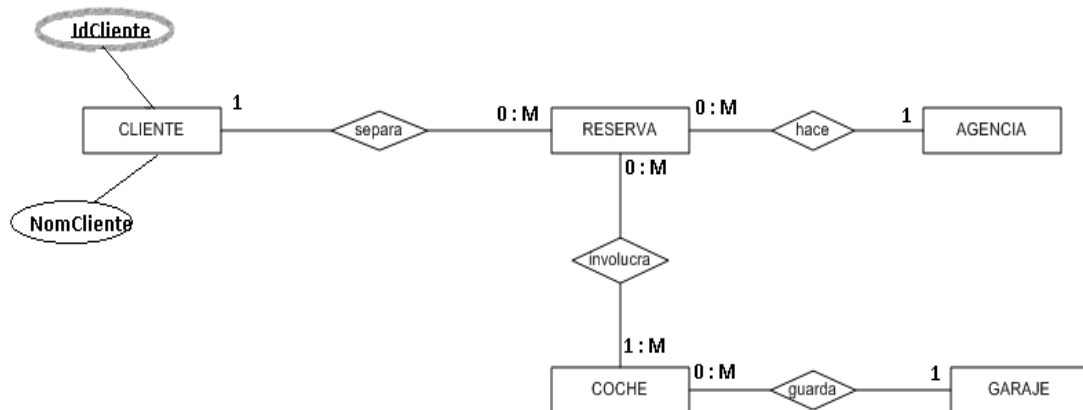
En la base de datos pueden existir clientes que no hayan hecho ninguna reserva.

Todas las entidades tienen una clave alfanumérica que las identifica unívocamente.

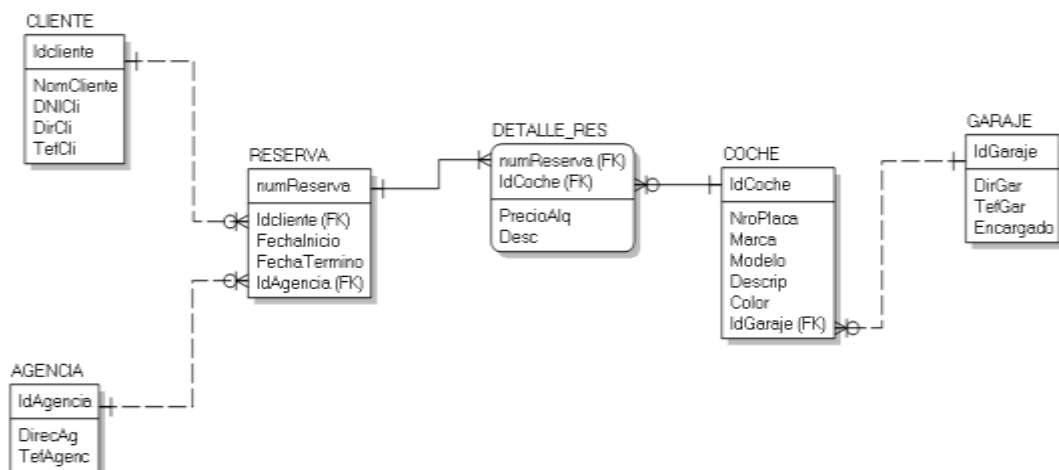
Se pide realizar el diseño del modelo E/R e indicar aquellos supuestos que no se han considerado.

Solución:

Modelo Conceptual



Transformado el Modelo Conceptual al Modelo Lógico correspondiente:



Proyecto 2 – Compañía-Departamentos-Proyectos



La base de datos COMPAÑIA debe almacenar información de los empleados, departamentos y proyectos de una empresa, de acuerdo con los siguientes requisitos:

La compañía está organizada en departamentos. Cada departamento tiene un nombre único, un número único y un empleado que la dirige y estamos interesados en guardar la fecha en que dicho empleado comenzó a dirigir el departamento.

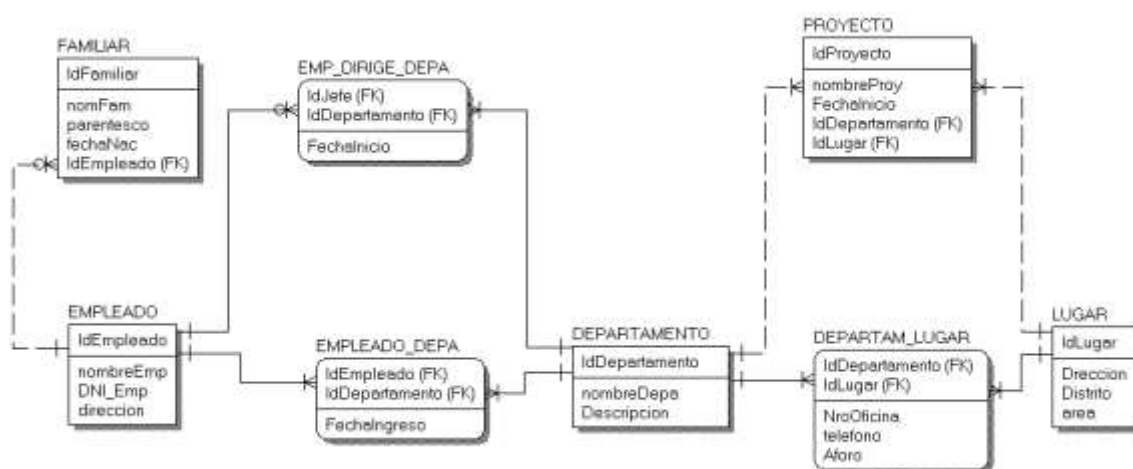
Un departamento puede estar distribuido en varios lugares.

Cada departamento controla un cierto número de proyectos, cada uno de los cuales tiene un nombre y un número únicos, y se realiza en un sólo lugar.

Se almacena el nombre, número de la Seguridad Social, dirección, salario, sexo y fecha de nacimiento de cada empleado. Todo empleado está asignado a un Departamento, pero puede trabajar en varios proyectos que no tienen por qué ser del mismo departamento. Nos interesa saber el número de horas que un empleado trabaja en cada proyecto a los que está asignado.

También se quiere guardar la relación de las cargas familiares de cada empleado para administrar el seguro que poseen. Almacenaremos el nombre, sexo y fecha de nacimiento de cada una de las cargas familiares y su parentesco con el empleado.

Solución:



Proyecto 3 – Agencia Aeroespacial



Diseñar un modelo que cumpla los siguientes requisitos de una agencia Aeroespacial.

Queremos reflejar toda la información referente a sus empleados técnicos, astronautas, científicos y familiares que dependen de éstos.

Los astronautas participan en misiones en un periodo de tiempo determinado, pero a lo sumo en un máximo de diez misiones.

Cada misión es propuesta por un científico en una fecha determinada, aunque hay algunas de estas misiones que constan de otras propuestas de otros científicos.

Un científico puede proponer una misión y puede viajar, o no, en dicha misión también

Estas misiones tienen por objetivo el estudio de uno o varios planetas.

Es por ello que interesa la información sobre los planetas, así como a qué galaxia y sistema solar pertenecen y qué satélites tienen y la distancia al sol del sistema medida en años luz.

La agencia dispone de una flota de naves espaciales de distintos tipos que realizan vuelos entre dos fechas en cada una de las misiones. Estas naves son manejadas por tripulantes espaciales de dos tipos, astronautas y robots.

En la misión necesariamente tiene que participar Técnicos, Astronautas, Científicos y Robots (uno o más).

Un Familiar solo depende de un tripulante (técnico, astronauta o científico)

Los Robots no tienen familiares.

Un tripulante (técnico, astronauta o científico) puede tener uno a varios familiares a su cargo que van en el viaje

Una vez finalizada la misión se almacenan sus conclusiones y fecha de finalización.

Los tripulantes se asignan al viaje. Y una sola nave se asigna a una misión.

Una nave puede ser reutilizada en varias misiones.

Implementar el Modelo Conceptual y Lógico correspondientes.

Proyecto 4 – Casos de corrupción



Una determinada organización después de una larga labor de investigación, ha logrado obtener una valiosa información acerca de numerosos casos de corrupción que se están dando en un determinado País. Para tenerla ordenada y accesible va a preparar una base de datos que contenga toso estos datos.

De cada caso se corrupción se desea saber su código, nombre por el que se le conoce, una breve descripción y una estimación de los millones que se han desviado (dinero).

Cada caso es investigado por un Juez, del que se desea tener su nombre, dirección, fecha de nacimiento y fecha que comenzó a ejercer. Un Juez solo puede investigar uno o más casos, pero un caso solo lo investiga un solo Juez.

En cada caso hay una serie de ciudadanos implicados, cada uno de ellos con un cargo principal, de estos ciudadanos se necesita conocer sus datos principales (Dni, Apellidos, Nombres, dirección y patrimonio. Un ciudadano puede estar involucrado en un solo caso, o en varios casos. En la lista de ciudadanos solo figuran los que esta implicados.

Los ciudadanos pueden o, no pertenecer a un partido político. A veces, un ciudadano, podría además tener algún cargo en dicho partido. De cada Partido se necesita conocer su nombre, dirección y teléfono principal.

Cada caso de corrupción es descubierto por un periódico, Se necesita conocer el nombre, dirección y tirada. Un periódico puede descubrir muchos casos, pero un caso solo lo descubre un solo periódico. Cada periódico puede tener, o no, una afinidad con un partido político, y por otro lado un partido político, puede tener afinidad con varios periódicos, o con ninguno.

Crear el Modelo Conceptual que represente esta realidad.

Transformar el Modelo anterior a su Modelo Lógico relacional correspondiente.

Proyecto 5 - Joyería



La joyería Accla, actualmente crece en su preferencia y es muy reconocida en Lima, no solo por los productos de calidad que ofrece, sino también por estar a la vanguardia con sus novedosos diseños en joyería, relojes y por la gran variedad de adornos de plata para el hogar.

Se ofrecen productos los cuales tiene un nombre, un precio y un código.

Los productos pertenecen a una colección determinada. Las colecciones también tienen un código y un nombre.

Un producto solo pertenece a una colección y una colección puede tener uno o varios productos. Los productos provienen de un proveedor, y un proveedor puede suministrar varios productos, pero un producto solo proviene de uno solo proveedor.

Un producto puede estar compuesto por uno o varios materiales principales (oro, plata, etc.).

Un material puede estar vinculado a uno o varios productos en un porcentaje determinado.

Un producto se puede vender una o varias veces siempre y cuando haya el stock suficiente.

Una venta es realizada por un solo cliente, pudiendo este cliente tener muchas ventas.

Cada venta es atendida por un solo vendedor; del vendedor es necesario guardar sus datos más esenciales.

Los clientes también poseen datos que son necesarios guardar.

Una venta se realiza en una de las sucursales que tiene la joyería distribuidas por todo el País.

Las sucursales tienen características que son necesarias almacenar.

Par mayor seguridad de los clientes cada producto que se vende tiene un seguro, que está afiliado a una compañía de seguros.

Una Compañía de seguros tiene a su cargo uno varios productos, y producto solo está afiliada a una sola compañía aseguradora. Se debe almacenar las características de cada compañía de seguros.

Crear el Modelo Conceptual que represente esta realidad.

Transformar el Modelo anterior a su Modelo Lógico relacional correspondiente.

Proyecto 6 – Liga de Fútbol



Una liga de fútbol profesional, ha decidido informatizar sus instalaciones creando una base de datos para guardar la información de los partidos que se juegan en la liga.

Se desea guardar en primer lugar los datos de los jugadores. De cada jugador se quiere guardar el nombre, fecha de nacimiento y posición en la que juega (portero, defensa, centrocampista...). Cada jugador tiene un código de jugador que lo identifica de manera única.

De cada uno de los equipos de la liga es necesario registrar el nombre del equipo, nombre del estadio en el que juega, el aforo que tiene, el año de fundación del equipo y la ciudad de la que es el equipo. Cada equipo también tiene un código que lo identifica de manera única. Un jugador solo puede pertenecer a un único equipo.

De cada partido que los equipos de la liga juegan hay que registrar la fecha en la que se juega el partido, los goles que ha metido el equipo de casa y los goles que ha metido el equipo de fuera. Cada partido tendrá un código numérico para identificar el partido.

También se quiere llevar un recuento de los goles que hay en cada partido. Se quiere almacenar el minuto en el que se realizar el gol y la descripción del gol. Un partido tiene varios goles y un jugador puede meter varios goles en un partido.

Por último, se quiere almacenar, en la base de datos, los datos de los presidentes de los equipos de fútbol (DNI, nombre, apellidos, fecha de nacimiento, equipo del que es presidente y año en el que fue elegido presidente). Un equipo de fútbol tan sólo puede tener un presidente, y una persona sólo puede ser presidente de un equipo de la liga. Crear el Modelo Lógico relacional que represente esta realidad.

PRACTICA CALIFICADA 3

Proyecto 1 de Normalización. - Ficha de Matricula



Normalizar estructuras complejas de datos.

Establecer las dependencias funcionales entre los atributos.

Identificar las claves primarias y foráneas.

Construir diagramas E-R

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

FICHA DE MARTICULA			Nº 20023
Fecha:	24/09/2015		
Carrera Técnica:	Programador en Bases de Datos	Código:	C
Modulo:	C04		
Alumno:	Córdova Tello, Javier	Código:	99409
Mes Acd.	CodCurso	Nombre del Curso	
3	VB2	Visual Basic Nivel II	
3	SQ2	SQL Server Nivel II	
2	SQ1	SQL Server Nivel I	
Observaciones:			

Del documento obtener las entidades que conformaran la estructura compleja.

Identificar la clave primaria.

Reconocer los Grupos Repetitivos. Utilizar la técnica desarrollada en clase.

Aplique la primera Forma Normal.

En las entidades resultantes reconocer las dependencias parciales. (Dibujar las líneas de dependencia)

Aplicar la Segunda Forma Normal

En las entidades resultantes reconocer las dependencias transitivas.

Aplicar la Tercera Forma Normal.

Con las entidades resultantes dibuje el Diagrama E-R correspondiente.

Identificar los Atributos que serán las Claves Principales y los Atributos que serán Claves Foráneas.

Trazar las líneas de relación entre las entidades partiendo desde la PK en la entidad Padre hacia la FK en la entidad Hija

Con el diagrama E-R del papel se debe construir el Modelo Lógico

Establecer las cardinalidades correspondientes a las relaciones.

Proyecto 2 – Biblioteca Municipal.



A partir del siguiente documento Normalizar y construir el Diagrama E-R, El Modelo Lógico, el Modelo Físico. Implementar la BD en SQL Server

Biblioteca Municipal de Los Cipreses Registro de préstamo de libros				Nro. de Préstamo : 40289	
Cód. de Lector 501		Nombre de Lector : Ríos Pérez, Juan		Teléfono : 5447291	
Cod Libro	Nro Copia	Título	Autor	Cod Edit	Nombre Editorial
1005	3	Estadística	Murray Spiegel	901	McGraw Hill
1007	5	Visual Basic. Net	E. Petroustos	803	Anaya
Cantidad de Libros prestados : 2			Fecha de Préstamo 15/03/2009		

1. Descubrir los atributos y presentar la Entidad Compleja.
2. Normalizar hasta la 3FN.
3. Construir el diagrama E-R.

En base al diag. Anterior construir el Modelo Lógico

Considerar un Precio de referencia para cada libro; también se quiere tener información sobre los autores, luego se pide considera un código de Autor y un País de origen de cada autor.

Un Lector puede prestase varios libros en un mismo préstamo, pero no puede llevarse Más de una copia por libro; es decir no puede llevarse más de un ejemplar por libro.

Proyecto 3 – Tienda de artículos deportivos



Dado el siguiente documento correspondiente a una muestra de algunas órdenes de compra de una organización comercial:

Id orden	Fecha	Id cliente	Nom cliente	Ciudad	Num_art	nom_art	cant	Precio
2301	23/02/15	101	Martin	Lima	3786	Red	3	35,00
2301	23/02/15	101	Martin	Lima	4011	Raqueta	6	65,00
2301	23/02/15	101	Martin	Lima	9132	Paq-3	8	4,75
2302	25/02/15	107	Herman	Trujillo	5794	Paq-6	4	5,00
2303	27/02/15	110	Pedro	Arequipa	4011	Raqueta	2	65,00
2303	27/02/15	110	Pedro	Arequipa	3141	Funda	2	10,00

1. Normalizar estructuras complejas de datos hasta la Tercera forma normal (3FN)
2. Establecer las dependencias funcionales entre los atributos.
3. Identificar las claves primarias y foráneas.
4. Construir diagramas Entidad Relación.

Proyecto 4 – Lavandería



Juan es un agente viajero que vende aspiradoras domésticas, las que se publicitan como que son lo suficientemente potentes como para aspirar gatos pequeños. Después de haber librado a varios ciudadanos de un pequeño poblado de sus queridos felinos, se ve forzado a retirarse rápidamente a otra ciudad.

Lo primero que hace al llegar es registrarse en un pequeño hotel y abrir su equipaje.

Obviamente tiene mucha ropa que debe ser lavada, y se dirige a la lavandería, llevando quince camisas blancas y un saco de pana.

La empleada de la lavandería lo recibe y le pregunta si es nuevo en el pueblo. Le pide sus datos: nombre, apellido, y el número telefónico del hotel donde está registrado. Anota que hay que lavar 15 camisas, y un saco que debe ser lavado en seco. También hace una nota acerca de las extrañas manchas rojas y del abundante pelo de gato que tiene adherido el saco.

Juan recibe una nota fechada y numerada con el precio indicado en una esquina, y la empleada le promete que su ropa estará lista al día siguiente, después de las 5 de la tarde.

Analizando los datos que figuran en la nota de pedido de Juan, proceda a su Normalización.

Nombre del campo	Valor
Recibo	1376
Nombre	Juan
Apellido	Jiménez
Teléfono	555-4567
Fecha de recepción	28/01/2002
Fecha de entrega	30/01/2002
Hora de entrega	5:00 p.m.
Tipo de prenda	Camisa de hombre
Tipo de servicio	Lavandería
Cantidad	15
Precio unitario	S/ 5.00
Tipo de prenda	Saco sport
Tipo de servicio	Lavado en seco
Cantidad	1
Precio unitario	S/ 15.00
Instrucciones especiales	Mancha, pelos de gato

Se requiere que cada cliente tenga un código que lo identifique; asimismo se debe incluir un código para cada tipo de prenda., las instrucciones especiales son para un tipo de prenda en un recibo en particular. Algunas prendas puede que no tengan instrucción especial alguna.

Normalizar hasta la Tercera Forma Normal (3FN). Construir el Modelo Lógico.

Proyecto 5 – Aeropuertos-Vuelos



Se tiene el caso de un aeropuerto donde se realizan vuelos a nivel nacional e Internacional. Un aviador, puede estar asignado a uno o muchos vuelos; y en un vuelo se asigna uno o varios aviadores. Cuando a un aviador se le asigna un vuelo se le asigna también el cargo que va a tener como responsabilidad en dicho vuelo.

Nro. de Vuelo 1473
Fecha: 25/10/2018
Cantidad de pasajeros: 45
Origen: Lima Destino: Arequipa
Datos del Avión:
Código: 278 Matricula: WP56AX99 Modelo: FD132
Fecha de Fabricación: 25/01/2015
Aviadores

Código	Apellidos y Nombres	Cargo en el vuelo	Nro. teléfono
17	Gonzales Castro, Manuel	Capitán	992255718
29	Bustamante Rivera, Frank	Copiloto 1	984567199

Lista de Pasajeros:

ID	Apellidos y Nombres	dni/NroP	FechaNac	Teléfono	Nacionalidad
104	Javier Loayza	092346789	15/03/1986	967123517	Perú
202	María Campuzano	092347601	25/11/1989	313936991	Perú
025	Steven Brown	A007B2345	01/12/1977	252773993	USA
-----		-----	-----	-----	

En base este documento realizar un diseño de datos, verificando las Formas normales.

Implementar el Modelo Lógico.

Un Pasajero puede viajar muchas veces, para ser considerado como pasajero debe haber viajado por lo menos una vez.

Solución:

NroVuelo	Grupo repetitivo 1										Grupo repetitivo 2									
	CantPasaj	Fecha	Origen	Destino	IdAvión	MatAvion	ModAvión	FechaFabr	IdAviador	ApeAviador	NomAviador	CargoAviador	TelfAviador	IdPasaj	ApePasaj	NomPasaj	DocPasaj	FechaNacPj	TelfPasaj	NacionPasaj

Se separan los dos grupos repetitivos

A	NroVuelo	CantPasaj	Fecha	Origen	Destino	IdAvión	MatAvion	ModAvión	FechaFabr	1FN	2FN	
B	NroVuelo	IdAviador	ApeAviador	NomAviador	CargoAviador	TelfAviador				1FN		
C	NroVuelo	IdPasaj	ApePasaj	NomPasaj	DocPasaj	FechaNacP	TelfPasaj	NacionPasaj		1FN		
B	NroVuelo (FK)	IdAviador	ApeAviador	NomAviador	CargoAviador	TelfAviador						
	B1	NroVuelo	IdAviador (FK)	CargoAviador						1FN	2FN	3FN
	B2	IdAviador	ApeAviador	NomAviador	TelfAviador					1FN	2FN	3FN
C	NroVuelo (FK)	IdPasaj	ApePasaj	NomPasaj	DocPasaj	FechaNacP	TelfPasaj	NacionPasaj	AsientoPasaj			
	C1	NroVuelo	IdPasaj (FK)	AsientoPasaj						1FN	2FN	3FN
	C2	IdPasaj	ApePasaj	NomPasaj	DocPasaj	FechaNacP	TelfPasaj	NacionPasaj		1FN	2FN	3FN
A	NroVuelo	CantPasaj	Fecha	Origen	Destino	IdAvión	MatAvion	ModAvión	FechaFabr			
	A1	NroVuelo	CantPasaj	Fecha	Origen	Destino	IdAvión			1FN	2FN	3FN
	A2	IdAvión	MatAvion	ModAvión	FechaFabr					1FN	2FN	3FN

Finalmente tenemos:

A1 VUELO	NroVuelo	CantPasaj	Fecha	Origen	Destino	IdAvión	
A2 AVION	IdAvión	MatAvion	ModAvión	FechaFabr			
B1 CARGO_AVIADOR	NroVuelo (FK)	IdAviador (FK)	CargoAviador				
B2 AVIADOR	IdAviador	ApeAviador	NomAviador	TelfAviador			
C1 ASIENTO_PASAJERO	NroVuelo (FK)	IdPasaj (FK)	AsientoPasaj				
C2 PASAJERO	IdPasaj	ApePasaj	NomPasaj	DocPasaj	FechaNacPj	TelfPasaj	NacionPasaj

Dibujando las entidades en forma Vertical (paradas)

VUELO	AVION	CARGO_AVIADOR	AVIADOR	ASIENTO_PASAJERO	PASAJERO
NroVuelo	IdAvión	NroVuelo (FK)	IdAviador	NroVuelo (FK)	IdPasaj (FK)
CantPasaj	MatAvión	IdPasaj (FK)	ApeAviador	IdPasaj (FK)	ApePasaj
Fecha	ModAvión	CargoAviador	NomAviador	AsientoPasaj	NomPasaj
Origen	FechaFabr		TelfAviador		DocPasaj
Destino					FechaNacPj
IdAvión (FK)					TelfPasaj
					NacionPasaj

Proyecto 6 - Gastos Por Cobrar



En el siguiente documento se presentan los gastos por cobrar de una cadena de edificios. El Documento tiene el código NumGPCM (Numero de Gastos por cobrar mensual). Cada edificio tiene dirección y su código (E1, E2, etc.)

NumGPCM: 1003

GASTOS POR COBRAR AL 10/06/2010							
	Piso	Dpto.	Titular	Tef	Gastos Comunes	Gastos Extras	Gastos Totales
Edificio "Neptuno"	2	A	Villa, Juan	641-1315	300.00	250.00	550.00
	3	B	Veliz, Carlos	641-5712	400.00	300.00	700.00
Av. Javier Prado 1479	3	C	Sanchez, Patricia	641-3712	300.00	250.00	550.00
San Isidro	TOTAL EDIFICIO.....				1000.00	800.00	1800.00
(E1)					*****	*****	*****
					*****	*****	*****
Edificio "Jupiter"	1	1	Lopez, Daniel	731-1921	150.00	50.00	200.00
Av. Arenales 689	2	5	Lopez, Daniel	731-2525	200.00	50.00	250.00
Lince	3	5	Katz, Diana	731-1616	200.00	50.00	250.00
(E2)	3	2	Trigo, Pedro	731-4214	150.00	50.00	200.00
	TOTAL EDIFICIO.....				700.00	200.00	900.00
					*****	*****	*****
	TOTAL GENERAL A COBRAR				11700.00	10000.00	21700.00

Se pide elaborar:

La Entidad Compleja y Normalizar hasta la 3FN

Dibujar el diagrama de entidades y sus relaciones.

Pasar este diagrama al Modelo Lógico

Solución:

Identificamos la Entidad Compleja con los atributos que se describen en el documento; y a continuación encontramos los atributos que forman parte del Grupo Repetitivo:

NumGPCM	IdEdificio	NomEdif	DirEdif	Piso	Dpto	Titular	TefDpto	GCdpto	GEdpto	GTdpto	GCedif	Geedif	Gtedif	T_GC_mes	T_GE_mes	T_GT_mes
---------	------------	---------	---------	------	------	---------	---------	--------	--------	--------	--------	--------	--------	----------	----------	----------

1FN

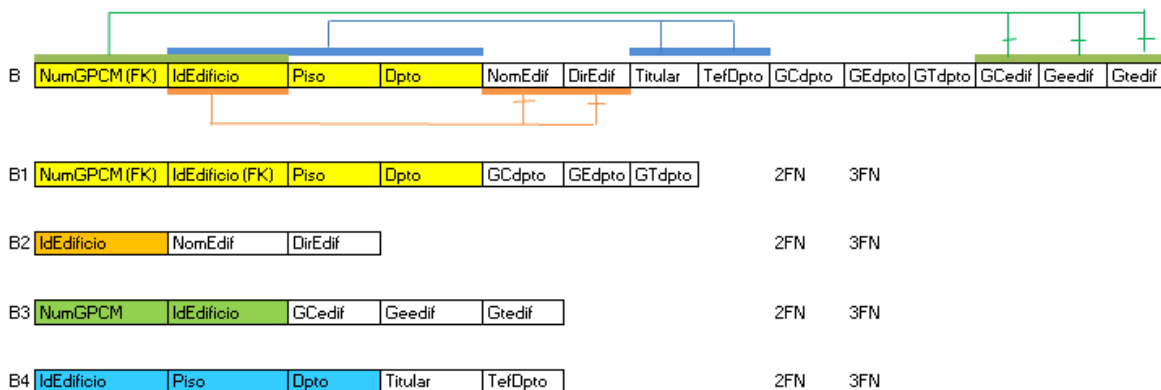
Luego para pasar a la Primera Forma Normal separamos el Grupo repetitivo en otra entidad aparte llevándonos una copia de la PK para mantener el vínculo entre las entidades resultantes:

A	NumGPCM	T_GC_mes	T_GE_mes	T_GT_mes	1FN	2FN	3FN							
B	NumGPCM(FK)	IdEdificio	NomEdif	DirEdif	Piso	Dpto	Titular	TefDpto	GCdpto	GEdpto	GTdpto	GCedif	Geedif	Gtedif

2FN

Para pasar a la Segunda Forma Norma identificamos los atributos con Dependencia Parcial de la Clave Primaria.

Nótese que la entidad A ya se encuentra en 2FN porque su PK es Simple (está formada por un solo atributo).

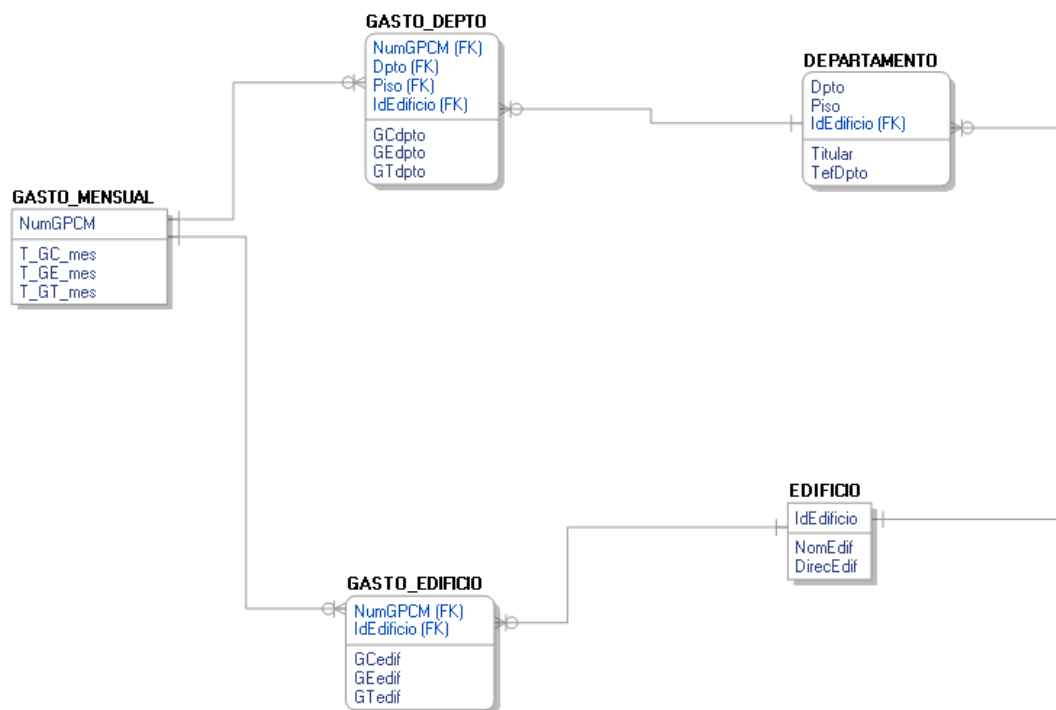


3FN

Para pasar a la tercera forma normal verificamos las dependencias transitivas de atributos con respecto de su clave primaria y separamos estos atributos en otra entidad llevándonos una copia del atributo del cual dependen transitivamente de la PK.

A	NumGPCM	T_GC_mes	T_GE_mes	T_GT_mes				GASTO_MENSUAL
B1	NumGPCM (FK)	IdEdificio (FK)	Piso	Dpto	GCdpto	GEdpto	GTdpto	GASTO_DEPTO
B2	IdEdificio	NomEdif	DirEdif					EDIFICIO
B3	NumGPCM (FK)	IdEdificio (FK)	GCedif	Geedif	Gtedif			GASTO_EDIFICIO
B4	IdEdificio	Piso	Dpto	Titular	TefDpto			DEPARTAMENTO

Finalmente representamos estas entidades resultantes en un Diagrama Entidad-Relación y luego implementamos el Modelo Lógico correspondiente.



Proyecto 7 – Ventas Vendedores por zonas.



En el siguiente documento se presenta un listado de Ventas totales por zona y vendedor.

Listado Nro: 1359

Periodo: Febrero 2010

	ZONA 1	ZONA 2	ZONA 3	ZONA 4	
VENDEDOR: RUIZ, DANIEL (V103)					
Línea Automotores (L1)	5,00	1,50	-	1,50	
Línea Electrodomésticos (L2)	100,00	20,00	10,00	5,00	
Línea Industrias (L3)	300,00	50,00	30,00	60,00	
=====					
TOTALES VENDEDOR :	405,00	71,50	40,00	66,50	583,00
VENDEDOR: MARTINEZ, JULIO (V116)					
Línea Microcomputadoras(L4)	50,00	10,00	-	-	
Línea Minicomputadoras(L5)	100,00	50,00	-	10,00	
Línea Mainframes(L6)	-	-	-	500,00	
=====					
TOTALES VENDEDOR :	150,00	60,00	-	510,00	720,00
.....					
.....					
					TOTAL GENERAL 1.860,00

Los Vendedores tienen un código que los identifica (V103, V102, etc.). Cada ZONA corresponde a un grupo de ciudades del País. Las líneas tienen códigos como L1, L2, L3, etc. Se pide elaborar:

La entidad compleja. Normalizar hasta la 3FN

Dibujar el diagrama de Entidades con sus relaciones. Pasar al Modelo Lógico.

PRACTICA CALIFICADA 4

Proyecto 1 - Sistema De Líneas Telefónicas

Se tiene una organización que cuenta con un sistema de líneas conectadas a una central telefónica

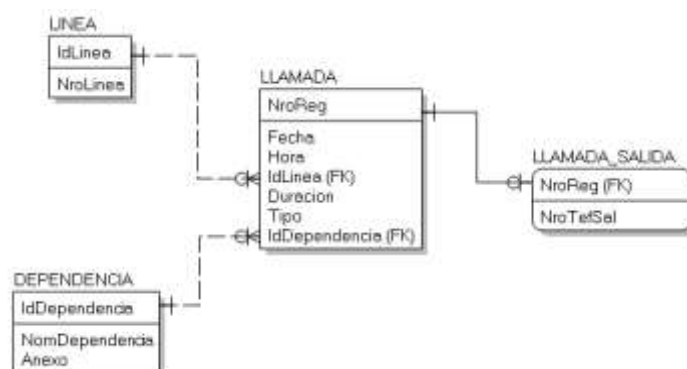


Se tiene entonces a continuación una muestra de las llamadas realizadas desde las diferentes dependencias de la organización

Nro.Reg	Fecha	Hora	Duración	Línea	Tipo	Nro.Telef.
Anexo 101 Dirección General						
1008	14.02.10	10:00 a.m.	00:10:30	1 - 2244650	Salida	4829935
1012	18.02.10	06:00 p.m.	00:15:45	2 - 2225060	Entrada	
1025	20.02.10	11:00 a.m.	00:25:25	2 - 2225060	Salida	4722233
----	----	----	----	----	----	----
----	----	----	----	----	----	----
Anexo 102 Departamento de Logística						
1020	20.02.10	11:00 a.m.	00:01:15	3 - 2231515	Salida	4464064
----	----	----	----	----	----	----
----	----	----	----	----	----	----
Anexo 116 Departamento de Contabilidad						
----	----	----	----	----	----	----

Cada Dependencia solo dispone de un anexo para recibir o hacer llamadas.

Solución:



Proyecto 2- Planilla De Empleados



Se tiene el siguiente documento correspondiente a la Planilla de Empleados de una empresa determinada. Obtener la Estructura compleja correspondiente y luego aplicar las técnicas de Normalización.

Planilla de empleados del mes de enero del 2020 N° 200-01								
Cod	Apellidos y Nombres	Sueldo	SNP	AFP	NombAFP	Pensión	Salud	Vivienda Total
Dependencia 100 Logística								
95001	Coronel Castillo, Gustav	2500	S			250	250	125 1875
95010	Flores Ramírez, Julio	2000	N	INT	Integra	100	200	100 1600
96014	Marcelo Villalobos, Ricar	1000	N	HOR	Horizonte	100	100	50 750
...
...
Dependencia 900 Contabilidad								
96601	Matsukawa Maeda, Sergi	2500	N	HOR	Horizonte	250	250	125 1875
97098	Carrasco Salas, Javier	2000	N	ITG	Integra	100	200	100 1600
94073	Pérez Castro, Carmen	2500	S			250	250	125 1875
...
...
Totales		xxxx				xxx	xxxx	xxxx xxxx

Pasos a realizar en el laboratorio:

Del documento obtener las entidades que conformaran la estructura compleja.

Identificar la clave primaria.

Aplique la primera Forma Normal.

En las entidades resultantes reconocer las dependencias parciales

Aplicar la Segunda Forma Normal

Aplicar la Tercera Forma Normal.

Identifique las dependencias condicionales

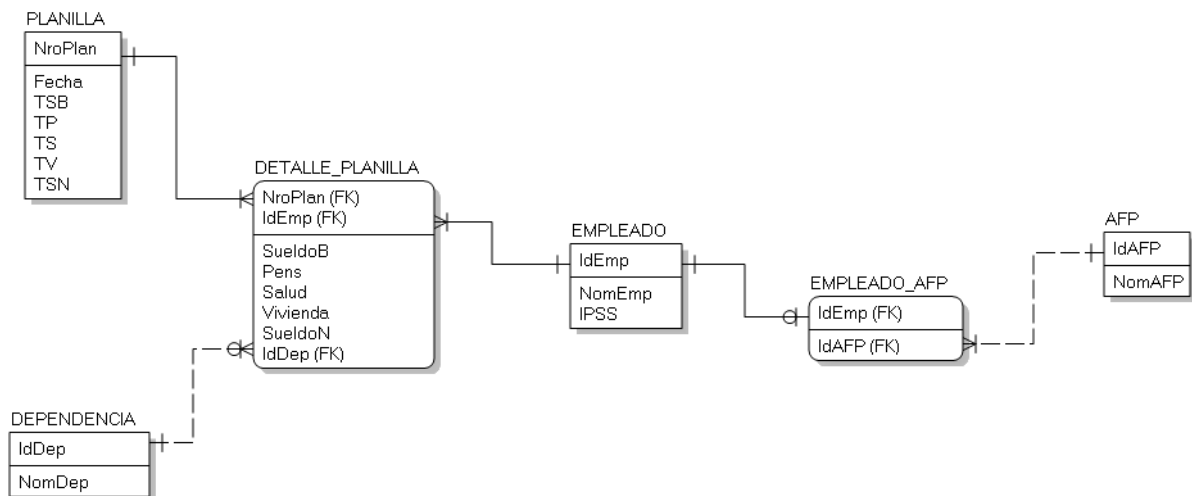
Identificar las de pendencias condicionales y separarlas en otra entidad.

Con las entidades resultantes dibuje el Diagrama E-R correspondiente.

Con el diagrama E-R del papel se debe construir el Modelo Lógico.

Establecer las cardinalidades correspondientes a las relaciones.

Solución:



Proyecto 3 - Fabricación De Insumos De Oficina

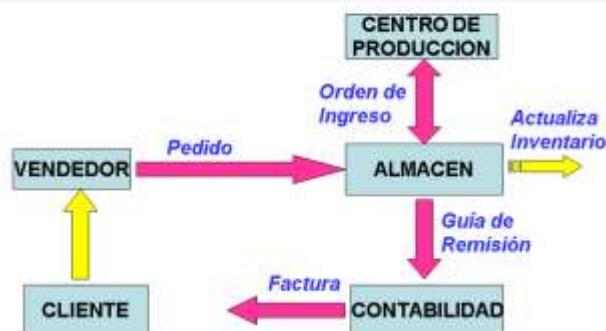


Se tiene el caso de una Empresa de fabricación de insumos de oficina y útiles de escritorio. El centro de producción se encuentra en Santa Anita; El departamento de comercialización y marketing se encuentra en San Isidro; Los Almacenes se encuentran en Lince y El Departamento de Contabilidad y Gerencia General se ubica en Miraflores.

Se envía equipos a los diferentes locales con el fin de recaudar los datos necesarios para realizar un modelo de datos y luego implementar una base de datos en SQL Server.

Los equipos han reportado unas muestras de los documentos de Pedidos, Guías de Remisión, Facturas y de las Boletas de Ingreso a Almacén.

Proceso de Comercialización de un Producto



Solución

GUIA DE PEDIDOS

Numero Pedido	Fecha	Cod.Vend	Nombre Vend	Cliente				Artículo				
				Nombre Cli.	RUC Cli	Dirección Cli	Teléfono Cli	Códart.	Cantidad	Unidad med.	Nombre Artículo	P u
P001	14/08/2010	V004	RUIZ	ALVA	23456789	Av. Tacna 333	4234567	A004	1	CIENTO	Lapicero	20
P001	14/08/2010	V004	RUIZ	ALVA	23456789	Av. Tacna 333	4234567	A001	10	DOCENA	Cuaderno	10
P001	14/08/2010	V004	RUIZ	ALVA	23456789	Av. Tacna 333	4234567	A003	5	RESMA	Papel Bond A4	50
P002	15/08/2010	V001	CALLE	BELTRAN	87654123	Jr Trujillo 214	5566789	A002	2	DOCENA	Borrador	4
P002	15/08/2010	V001	CALLE	BELTRAN	87654123	Jr Trujillo 214	5566789	A004	4	CIENTO	Lapicero	20
P002	15/08/2010	V001	CALLE	BELTRAN	87654123	Jr Trujillo 214	5566789	A003	5	RESMA	Papel Bond A4	50

Entidad Compleja

IdPedido	FechaPed	IdVend	Vendedor	IdCliente	NomCli	RUCCli	DirCli	TefCli	IdProd	Cant	Unid	Descrip	Precio
----------	----------	--------	----------	-----------	--------	--------	--------	--------	--------	------	------	---------	--------

Después de Normalizar

A1

IdPedido	FechaPed	IdVend	IdCliente
----------	----------	--------	-----------

A3

IdCliente	NomCli	RUCCli	DirCli	TefCli
-----------	--------	--------	--------	--------

A2

IdVendedor	Vendedor
------------	----------

B1

IdPedido	IdProd	Cant	Precio
----------	--------	------	--------

B2

IdProd	Descrip	Precio
--------	---------	--------

GUIA DE REMISION

Numero Rem.	Fecha	Número Pedido	Cod.Vend	Nombre Vendedor	Cliente				Artículo				
					Nombre Cli.	RUC Cli	Dirección Cli	Telf Cli	CódArt.	Cant	Unidad med.	Nombre Artículo	P u
R101	15/08/2010	P001	V004	RUIZ	ALVA	23456789	Av. Tacna 333	4234567	A004	1	CIENTO	Lapicero	20.00
R101	15/08/2010	P001	V004	RUIZ	ALVA	23456789	Av. Tacna 333	4234567	A001	10	DOCENA	Cuaderno	10.00
R101	15/08/2010	P001	V004	RUIZ	ALVA	23456789	Av. Tacna 333	4234567	A003	5	RESMA	Papel Bond A4	50.00
R104	16/08/2010	P002	V001	CALLE	BELTRAN	87654123	Jr Trujillo 214	5566789	A002	2	DOCENA	Borrador	4.00
R104	16/08/2010	P002	V001	CALLE	BELTRAN	87654123	Jr Trujillo 214	5566789	A004	4	CIENTO	Lapicero	20.00
R104	16/08/2010	P002	V001	CALLE	BELTRAN	87654123	Jr Trujillo 214	5566789	A003	5	RESMA	Papel Bond A4	50.00

Entidad Compleja

IdGuiaRem	FechaGuiaRem	IdPedido	IdVend	Vendedor	IdCliente	NomCli	RUCCLI	DirCli	TelCli	IdProd	Cant	Unid	Descrip	Precio
-----------	--------------	----------	--------	----------	-----------	--------	--------	--------	--------	--------	------	------	---------	--------

Después de Normalizar

C1

IdGuiaRem	FechaGuiaRem	IdPedido	IdVend	IdCliente
-----------	--------------	----------	--------	-----------

Ya están referidas en A1

C3

IdCliente	NomCli	RUCCLI	DirCli	TelCli
-----------	--------	--------	--------	--------

C2

IdVendedor	Vendedor
------------	----------

D1

IdGuiaRem	IdProd	Cant	Precio
-----------	--------	------	--------

D2

IdProd	Descrip	Precio
--------	---------	--------

FACTURA

Número		Num		Cliente				Artículo				Monto	Total	Total	Total	
				Nombre Cli	RUC Cli	Dirección Cli	Teléf. Cli	Cód art.	Cant	Unidad med.	Nombre Artículo					Prec un
Factura	Fecha	Rem	Cod.Ve													
F00234	15/08/2010	R101	V004	ALVA	23456789	Av. Tacna 333	4234567	A004	1	CIENTO	Lapicero	20,00	20,00	370,00	66,60	436,60
F00234	15/08/2010	R101	V004	ALVA	23456789	Av. Tacna 333	4234567	A001	10	DOCENA	Cuaderno	10,00	100,00			
F00234	15/08/2010	R101	V004	ALVA	23456789	Av. Tacna 333	4234567	A003	5	RESMA	Papel Bond A4	50,00	250,00			
F00235	16/08/2010	R104	V001	BELTRAN	87654123	Jr Trujillo 214	5566789	A002	2	DOCENA	Borrador	4,00	8,00			
F00235	16/08/2010	R104	V001	BELTRAN	87654123	Jr Trujillo 214	5566789	A004	4	CIENTO	Lapicero	20,00	80,00			
F00235	16/08/2010	R104	V001	BELTRAN	87654123	Jr Trujillo 214	5566789	A003	5	RESMA	Papel Bond A4	50,00	250,00	338,00	60,84	398,84

Entidad Compleja

IdFacta	FechaFact	IdGuiaRem	IdVend	Vendedor	IdCliente	NomCli	RUCCLI	DirCli	TelCli	IdProd	Cant	Unid	Descrip	Precio	Monto	MontoV	MontoF
---------	-----------	-----------	--------	----------	-----------	--------	--------	--------	--------	--------	------	------	---------	--------	-------	--------	--------

Después de Normalizar

E1

IdFacta	FechaFact	IdGuiaRem	IdVend	IdCliente	MontoV	MontoF	MontoFact
---------	-----------	-----------	--------	-----------	--------	--------	-----------

Atributos redundantes

E3

IdCliente	NomCli	RUCCLI	DirCli	TelCli
-----------	--------	--------	--------	--------

E2

IdVendedor	Vendedor
------------	----------

F1

IdFacta	IdProd	Cant	Precio	Monto
---------	--------	------	--------	-------

Atributo redundante

F2

IdProd	Descrip	Precio
--------	---------	--------

De las reglas de Negocio se conocen:

Monto = Precio * Monto
MontoV = Sum(Monto)
MontoF = Sum(Monto * IGV Prod)
MontoFactura = MontoV + MontoF

**BOLETA DE
INGRESO DE ALMACEN**

Número de Ingreso	Fecha	Artículo			
		Cod.Art.	Nombre Art.	Cantidad	Unidad
I0984	01/06/2010	A004	Lapicero	100	CIENTO
I0984	01/06/2010	A001	Cuaderno	60	DOCENA
I0984	01/06/2010	A003	Papel Bond A4	50	RESMA

Entidad Compleja

IdOrding	FechaOrding	IdProd	Cant	Unid	Descrip
----------	-------------	--------	------	------	---------

G1

IdOrding	FechaOrding
----------	-------------

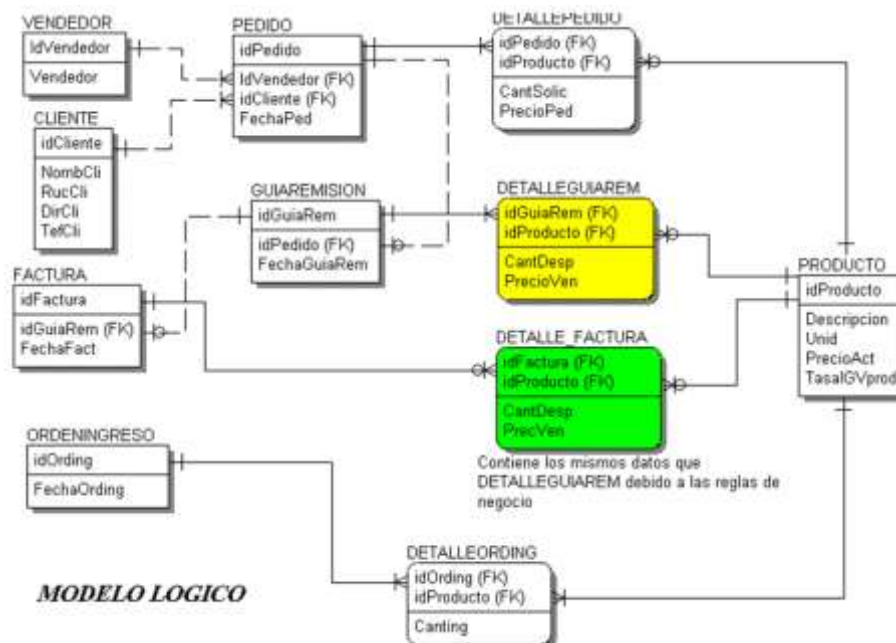
H1

IdOrding	IdProd	Cant
----------	--------	------

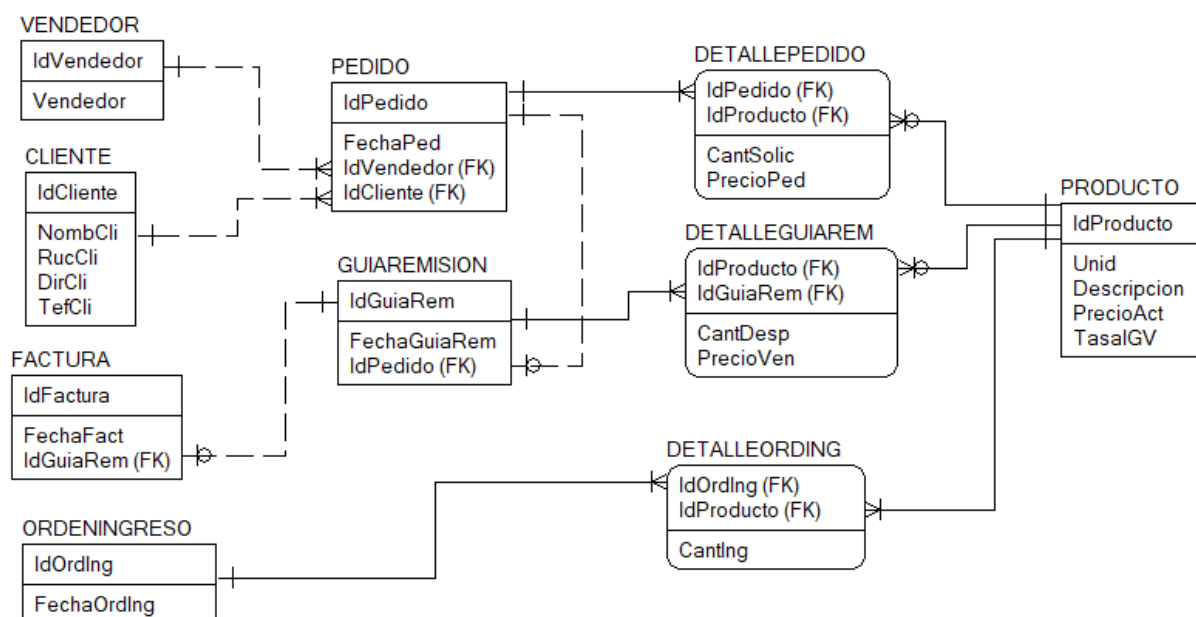
H2

IdProd	Descrip	Unid
--------	---------	------

Con las entidades resultantes creamos un solo diagrama donde convergen todas las Entidades. Se han asignado nombres a cada Entidad



Se elimina la Entidad Redundante y obtenemos el Diagrama E-R Final



CURSOS RELACIONADOS

Para contactarte con algún representante del CEPS-UNI tienes la siguiente URL:

<https://www.ceps.uni.edu.pe/contacto/>

A continuación, tienes la lista de cursos relaciones que te pueden interesar:



