



Base de Datos Teoría

ÍNDICE

Presentación	5
Red de contenidos	6
Unidad de Aprendizaje 1	
SEMANA 1 : Introducción al curso. Conceptos básicos	7
SEMANA 2 : Arquitectura de un sistema de bases de datos	23
Unidad de Aprendizaje 2	
SEMANA 3 : Modelamiento conceptual	33
SEMANA 4 : Diagrama Entidad-Relación Parte 1	57
SEMANA 5 : Diagrama Entidad-Relación Parte 2	63
Unidad de Aprendizaje 3	
SEMANA 6 : Álgebra relacional	71
SEMANA 7 : Examen Parcial	
SEMANA 8 : Álgebra relacional	91
Unidad de Aprendizaje 4	
SEMANA 9 : Modelo relacional. Normalización	97
SEMANA 10 : Normalización	111
SEMANA 11 : Normalización	129
SEMANA 12 : Normalización	137
Unidad de Aprendizaje 5	
SEMANA 13 : Obtención del modelo lógico-global de los datos a partir del DER	143
SEMANA 14 : Obtención del modelo lógico-global de los datos a partir del DER	151
SEMANA 15 : Metodología para el diseño de bases de datos	157
Unidad de Aprendizaje 6	
SEMANA 16 : Recuperación, concurrencia y seguridad	171
SEMANA 17 : Examen Final de teoría	

PRESENTACIÓN

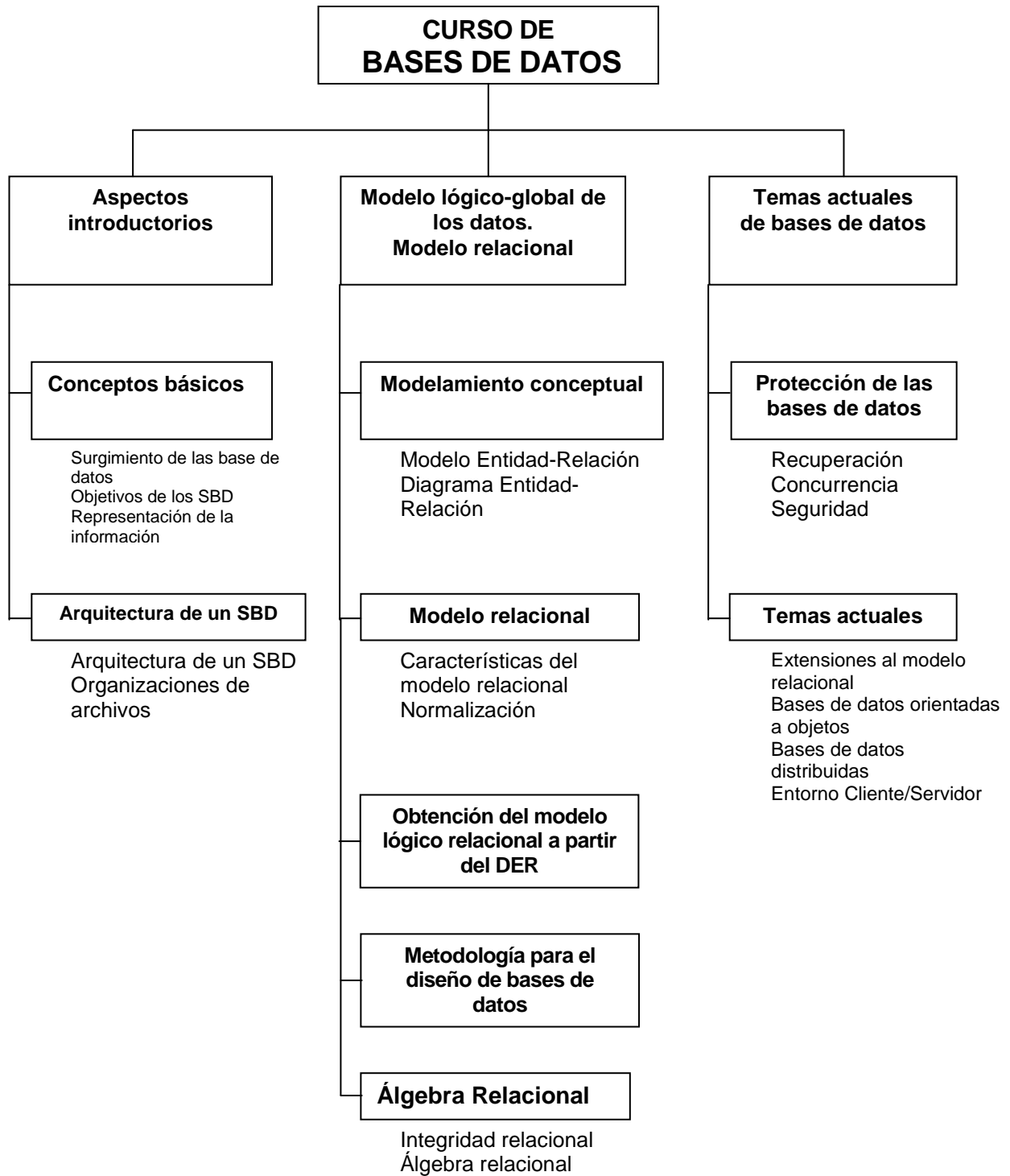
El presente manual está diseñado para que sea utilizado como texto de trabajo en el curso de Bases de Datos. Por esto, se encuentra estructurado de acuerdo a la organización de este curso.

Cada sesión contempla, entre otros aspectos, los objetivos que se persiguen y los contenidos que se abordan en la misma, lo cual permite que el alumno pueda tener una idea clara de los aspectos que se van a tratar y del alcance de éstos en el curso.

En el desarrollo de los contenidos, las explicaciones están ilustradas con ejemplos y figuras que facilitan la comprensión. Algunas sesiones están completamente dedicadas a la ejercitación de los contenidos impartidos, de modo que los alumnos puedan alcanzar habilidades en la aplicación de las técnicas estudiadas.

Al final de cada sesión se añade una Autoevaluación, en la cual se proponen ejercicios para que los estudiantes los realicen independientemente, fuera de clases, de manera que puedan aumentar sus habilidades y comprobar sus conocimientos. Para recordar brinda, de modo resumido, las conclusiones más importantes acerca de los contenidos tratados.

RED DE CONTENIDOS



UNIDAD DE APRENDIZAJE 1
SEMANA 1

INTRODUCCIÓN AL CURSO. CONCEPTOS BÁSICOS

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos describirán los componentes de una base de datos a partir de casos propuestos por el profesor, tomados de situaciones cotidianas y de ejemplos producidos individualmente.

TEMARIO

Surgimiento histórico de las bases de datos integradas

Objetivos de los sistemas de base de datos (SBD)

Representación de la información

ACTIVIDADES PROPUESTAS

1. Explicaciones por parte del facilitador combinadas con ejemplos.
2. Se efectúan preguntas de comprobación dirigidas a los alumnos sobre los temas tratados.

1. SURGIMIENTO HISTÓRICO DE LAS BASES DE DATOS

Al estudiar el desarrollo del procesamiento automatizado de datos, en lo que se refiere al aseguramiento técnico, se habla de diferentes generaciones.

Desde el punto de vista del aseguramiento matemático y, en particular, del aseguramiento de programas, algunos autores reconocen 3 generaciones:

- Solución de tareas aisladas.
- Integración de tareas aisladas en sistemas particulares.
- Integración de sistemas particulares en sistemas automatizados de dirección.

Este proceso de integración ocurre paralelamente, aunque no simultáneamente, en dos esferas:

a. Integración de los programas.

Ha estado facilitada por el uso de lenguajes de programación cada vez más sofisticados y de redactores que permiten el acoplamiento de módulos escritos en lenguajes diferentes.

b. Integración de los datos.

En la integración de los datos se han producido tres categorías de técnicas para su manipulación:

1. Sistemas orientados a los dispositivos.

Programas y archivos que son diseñados y empleados de acuerdo a las características físicas de la unidad central y los periféricos. Cada programa está altamente interconectado con sus archivos, por lo que la integración de datos de diferentes sistemas es imposible prácticamente.

2. Sistemas orientados a los archivos.

La lógica de los programas depende de las técnicas de organización de los archivos (secuencial, directo, etc.). Cada usuario organiza su archivo de acuerdo con sus necesidades y las relaciones entre los elementos se establecen a través de los programas de aplicación.

Esta forma de trabajo implica redundancia de datos, que trae aparejada mayor gasto de memoria y complica las operaciones de actualización (modificar un dato donde quiera que aparezca). Esto aumenta el tiempo de tratamiento y atenta contra la integridad de la información.

Cuando se habla de integridad se está haciendo referencia a que, en todo momento, los datos almacenados estén correctos en correspondencia con la realidad.

Además, en la vida real se establecen relaciones entre los objetos que son muy difíciles de representar u obtener a partir de sistemas tradicionales de archivos. Por ejemplo, si se tiene información sobre trabajadores y

estudiantes de una facultad, las aplicaciones requeridas van a definir la manera de organizar y estructurar los archivos.

Si se desea obtener datos como:

- Promedio de las calificaciones de cada estudiante
- Listado de estudiantes por grupo
- Categoría científica y docente de cada profesor
- Salario de cada profesor

Resulta adecuado establecer dos archivos: uno de profesores y uno de estudiantes.

¿Qué ocurre si se quiere establecer vínculos entre los profesores y estudiantes? Por ejemplo, si se desea obtener:

- Los estudiantes de un profesor
- Los profesores de un estudiante

Se estructuraría un archivo de profesores y estudiantes que resolvería algunas demandas, pero sería ineficiente para otras.

Entonces, ¿es posible representar de manera eficiente, utilizando los medios de cómputo, los casos o procesos¹ de la realidad objetiva, aunque sea, por supuesto, de forma esquemática, pero en la que se establezcan determinados vínculos entre los elementos u objetos que forman parte de esos procesos o casos?

Veremos que esto es posible hacerlo a través de la utilización de bases de datos (BD) y de los sistemas de gestión de bases de datos (SGBD) que dirigen su manipulación.

Se tiene entonces la tercera categoría:

3. Sistemas orientados a base de datos

En los que hay una débil interdependencia entre los programas de aplicación y la organización física de los datos.

Ahora bien, ¿qué es una **base de datos**? Aunque existen distintas formas de definir una base de datos, la siguiente puede considerarse una definición adecuada:

Definición de base de datos: Conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. Es decir que, que una base de datos puede considerarse una colección de datos variables en el tiempo.

El software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde

¹ A los casos o procesos también se le denomina proceso de negocio.

diferentes puntos de vista y a la vez, se denomina **sistema de gestión de bases de datos** (SGBD).

Es importante diferenciar los términos base de datos y SGBD.

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, de tal forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

Los programas de aplicación operan sobre los datos almacenados en la base utilizando las facilidades que brindan los SGBD, los que, en la mayoría de los casos, poseen lenguajes especiales de manipulación de la información que facilitan el trabajo de los usuarios.

2. OBJETIVOS DE LOS SBD

Existen muchas formas de organizar las bases de datos, pero hay un conjunto de objetivos generales que deben cumplir todos los SGBD, de modo que faciliten el proceso de diseño de aplicaciones y que los tratamientos sean más eficientes y rápidos, dando la mayor flexibilidad posible a los usuarios.

Los objetivos fundamentales de los SBD son:

2.1 Independencia de los datos y los programas de aplicación

Ya vimos que, con archivos tradicionales, la lógica de la aplicación contempla la organización de los archivos y el método de acceso. Por ejemplo, si por razones de eficiencia se utiliza un archivo secuencial indexado, el programa de aplicación debe considerar la existencia de los índices y la secuencia del archivo. Entonces, es imposible modificar la estructura de almacenamiento o la estrategia de acceso sin afectar el programa de aplicación (naturalmente, lo que se afecta en el programa son las partes de éste que tratan los archivos, lo que es ajeno al problema real que el programa de aplicación necesita resolver). En un SBD sería indeseable la existencia de aplicaciones y datos dependientes entre sí, por dos razones fundamentales:

1. Diferentes aplicaciones necesitarán diferentes aspectos de los mismos datos (por ejemplo, puede requerirse la representación decimal o binaria).
2. Se debe poder modificar la estructura de almacenamiento o el método de acceso según los cambios en el caso o proceso de la realidad sin necesidad de modificar los programas de aplicación (también para buscar mayor eficiencia).

La independencia de los datos se define como: la inmunidad de las aplicaciones a los cambios en la estructura de almacenamiento y en la estrategia de acceso y esto constituye el objetivo fundamental de los SBD.

2.2 Minimización de la redundancia

Habíamos visto cómo, con los archivos tradicionales, se produce redundancia de la información. Uno de los objetivos de los SBD es minimizar la redundancia de los datos. Se dice disminuir la redundancia, no eliminarla, pues, aunque se definen las bases de datos como no redundantes, en realidad existe redundancia en un grado no significativo para disminuir el tiempo de acceso a los datos o para simplificar el método de direccionamiento. Lo que se trata de lograr es la eliminación de la redundancia superflua.

2.3 Integración y sincronización de las bases de datos

La **integración** consiste en garantizar una respuesta a los requerimientos de diferentes aspectos de los mismos datos por diferentes usuarios, de forma que, aunque el sistema almacene la información con cierta estructura y cierto tipo de representación, debe garantizar entregar al programa de aplicación los datos que solicita y en la forma en que lo solicita.

Está vinculada a la **sincronización**, que consiste en la necesidad de garantizar el acceso múltiple y simultáneo a la base de datos, de modo que los datos puedan ser compartidos por diferentes usuarios a la vez. Están relacionadas, ya que lo usual es que diferentes usuarios trabajen con diferentes enfoques y requieran los mismos datos, pero desde diferentes puntos de vista.

2.4 Integridad de los datos

Consiste en garantizar la no contradicción entre los datos almacenados, de modo que, en cualquier momento del tiempo, los datos almacenados sean correctos, es decir, que no se detecte inconsistencia entre los datos. Está relacionada con la minimización de la redundancia, ya que es más fácil garantizar la integridad si se elimina la redundancia.

2.5 Seguridad y recuperación

Seguridad (también llamada protección): garantiza el acceso autorizado a los datos, la forma de interrumpir cualquier intento de acceso no autorizado, ya sea por error del usuario o por mala intención.

Recuperación: permite que el sistema de bases de datos disponga de métodos que garanticen la restauración de las bases de datos al producirse alguna falla técnica, interrupción de la energía eléctrica, etc.

2.6 Facilidad de manipulación de la información

Los usuarios de una base de datos pueden acceder a ella con solicitudes para resolver muchos problemas diferentes. El SBD debe contar con la capacidad de una búsqueda rápida por diferentes criterios, debe permitir que los usuarios planteen sus demandas de una forma simple, aislándolo de las complejidades del tratamiento de los archivos y del direccionamiento de los

datos. Los SBD actuales brindan lenguajes de alto nivel, con diferentes grados de facilidad para el usuario no programador, que garantizan este objetivo, los llamados sublenguajes de datos.

2.7 Control centralizado

Uno de los objetivos más importantes de los SBD es garantizar el control centralizado de la información. Permite controlar, de manera sistemática y única, los datos que se almacenan en la base de datos, así como el acceso a ella.

Lo anterior implica que debe existir una persona, o conjunto de personas, que tenga la responsabilidad de los datos operacionales: el administrador de la base de datos, que puede considerarse parte integrante del SBD y cuyas funciones se abordarán en la siguiente sesión.

Existen otros objetivos que deben cumplir los SBD que, en muchos casos, dependen de las condiciones o requerimientos específicos de utilización del sistema.

3. REPRESENTACIÓN DE LA INFORMACIÓN

En el proceso y construcción de todo sistema informativo automatizado, el diseño de la base de datos ocupa un lugar importante, a tal punto que ésta puede verse como un proceso relativamente independiente dentro del diseño del sistema y compuesto por una serie de etapas. Es por ello que resulta de interés el estudio de los problemas relacionados con el diseño de las bases de datos y el modelamiento de la información.

3.1 Niveles de abstracción referidos a la información

Cuando se habla de información, se hace referencia, de forma general, a tres niveles diferentes de abstracción, tendiéndose a saltar de uno a otro sin establecer una advertencia previa.

3.1.1 Nivel del mundo real

El primero de estos niveles es el del MUNDO REAL, en el que existen entidades u objetos, que no son más que cosas o elementos que existen y están bien diferenciados entre sí, que poseen propiedades y entre los cuales se establecen relaciones. Por ejemplo, una silla es una entidad u objeto, un automóvil, un empleado, un profesor, un estudiante, que son cosas concretas, también lo son; pero también puede ser algo no tangible, como un suceso cualquiera, una cuenta de ahorro, o un concepto abstracto.

Entre las propiedades que caracterizan a una entidad u objeto pudieran encontrarse el color, el valor monetario, el nombre, etc.

De las relaciones entre las entidades u objetos hablaremos más adelante.

La determinación de cierta entidad u objeto correspondiente a un caso o proceso, está muy relacionada con el nivel de abstracción en el que se esté realizando el análisis. Así, por ejemplo, si se estudia el comportamiento de un insecto específico en determinadas condiciones climáticas, las propiedades y relaciones que interesan son de un cierto tipo; sin embargo, si se estuviera realizando un estudio de las diferentes especies de insectos, entonces serían otros los objetos a definir, así como las propiedades que los caracterizarían y las relaciones que se establecerían. Si se estuviera analizando todo el reino animal, serían también otros los objetos a definir, con sus características y propiedades.

3.1.2 Nivel del dominio de las ideas

El segundo nivel es el del **DOMINIO DE LAS IDEAS**. Es en el que se decide la información que debe existir en la base de datos sobre un caso o proceso del mundo real, o sea, qué información debe almacenarse. En este nivel es donde realmente se define el contenido informativo que representará al caso, proceso o ente de la realidad objetiva que se está analizando. De modo que, en este nivel, se definen cuáles objetos y qué propiedades de éstos son representativas y sobre los cuales es necesario almacenar información.

En este nivel es donde se trabaja con los conceptos más importantes del modelo de datos, que establecen la relación entre el mundo real y la información almacenada físicamente en la base de datos:

Campo o atributo: es la unidad menor de información sobre un objeto (almacenada en la base de datos) y representa una propiedad de un objeto (por ejemplo, el color). Sin embargo, hay que distinguir entre el **nombre o tipo del atributo** y el **valor del atributo**, ya que un nombre de atributo puede tomar diferentes valores sobre un cierto conjunto que se denomina **dominio**. A un valor de un atributo determinado o definido en el dominio dado, en un cierto momento del tiempo, se le denomina **ocurrencia** del atributo.

Ejemplo:

Atributo	Color	Cat_Doc
Dominio	{Azul, Rojo, Verde,...}	{PT, PA, A, I}
Ocurrencia	Rojo	A

Ahora bien, una colección identificable de campos asociados es un **artículo o registro** y representa un objeto con sus propiedades. Una vez más, es imprescindible distinguir entre **nombre o tipo de artículo y ocurrencia de artículo**. Una **ocurrencia de artículo o tupla**, consiste en un grupo de ocurrencias de campos relacionados, representando una asociación entre ellos. Por ejemplo, tenemos un artículo correspondiente al objeto profesor, en un caso o proceso de la realidad que pretenda representar el comportamiento de una Facultad. El nombre o tipo de artículo puede ser PROFESOR, que esté formado por los siguientes tipos de campos o atributos:

DNI_PROF : número de DNI del profesor
 NOM_PROF : nombre del profesor

CAT_DOC : categoría docente del profesor
DPTO : departamento docente al que pertenece el profesor

Una ocurrencia de este artículo puede ser:

12801731 Hernández, Juan PA Computación.

Un archivo o archivos pueden ser definidos como un conjunto de ocurrencias de un mismo tipo de artículo.

En la práctica, a menudo interesan las colecciones o conjuntos de objetos similares, necesitándose almacenar la información de las mismas propiedades para cada uno de ellos, por ejemplo, el conjunto de profesores de la Facultad.

Entonces, una **base de datos** contendrá muchas ocurrencias de cada uno de los tipos de artículos, lo que implica que la base de datos, por supuesto, también contendrá muchas ocurrencias de los distintos tipos de atributos.

Uno de los momentos cruciales en el diseño de un caso de la realidad objetiva que se concreta en una base de datos es, precisamente, la selección de los conjuntos de objetos y sus propiedades.

Además, existe otro concepto muy importante en este nivel, que es el concepto de **llave o clave**: un atributo o conjunto de atributos de un artículo que define que cada ocurrencia de artículo de la base de datos sea única. En principio, cada artículo tiene una llave, ya que se tiene como hipótesis que cada elemento u ocurrencia del artículo es diferente de las demás. Por ejemplo, el DNI del trabajador puede constituir la llave del artículo trabajador.

3.1.3 Nivel de los datos

El tercer nivel es el de los datos propiamente dichos, representados mediante cadenas de caracteres o de bits.

En este nivel es necesario tener en cuenta la diferencia entre tipo de dato y valor del dato. El tipo de dato corresponde a un atributo o tipo de atributo, que está asociado a un tipo de artículo correspondiente, mientras que, el valor, corresponde a una ocurrencia del atributo. Sin embargo, una colección de bits o caracteres que representa un único valor de datos y que puede existir independientemente de cualquier información que se almacena, adquiere significado sólo cuando se le asocia a un tipo de atributo. Se puede, por ejemplo, almacenar permanentemente los valores ROJO, AZUL, VERDE, etc. y asociarlo en un momento determinado a un tipo de atributo a través de los valores que toma, representando una ocurrencia en una tupla.

3.2 Relaciones de correspondencia

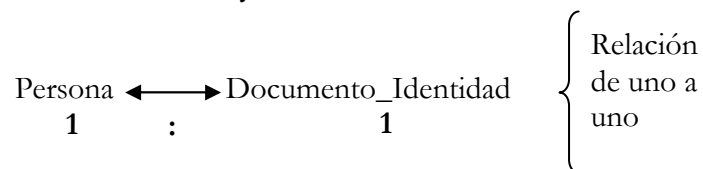
Es importante notar que, en general, habrá asociaciones o relaciones enlazando las entidades básicas.

Estos enlaces se pueden establecer entre diferentes objetos o tipos de artículos o entre un mismo tipo de artículo. Por ejemplo, se puede tener una relación entre **dos tipos de objetos**: SUMINISTRADOR y PRODUCTO, de modo que, un suministrador puede suministrar muchos productos y que un producto puede ser suministrado por muchos suministradores y se conoce, además, la CANTIDAD de cada producto que suministra un suministrador dado. Otro ejemplo pudiera ser con el artículo PERSONA, sobre el que se pudiera representar la relación "SER MADRE DE", que no es más que una relación que se establece entre elementos de **un mismo tipo de artículo**.

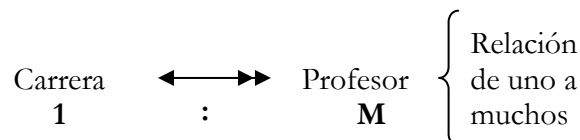
Es necesario profundizar acerca de los diferentes tipos de relaciones que pueden ocurrir en la práctica.

Es necesario establecer la correspondencia que existe entre los datos. Esta relación puede ser simple o compleja.

Por relación **simple** se entiende una correspondencia biunívoca (de uno a uno) entre las ocurrencias de los objetos, o sea, de los artículos. Si, por ejemplo, los objetos o entidades son Documento_Identidad y Persona, la correspondencia entre ellos es simple: a cada persona le corresponde un documento de identidad y viceversa.



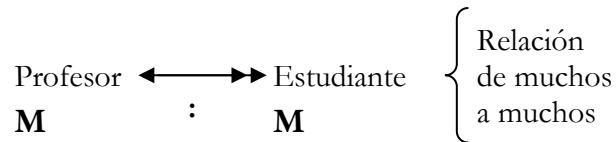
Si las entidades son Profesor y Carrera, la relación es más complicada, porque en cada carrera docente trabajan varios profesores. La terminología usual expresa que la correspondencia de profesor a carrera es simple, ya que cada profesor es miembro de una única carrera, mientras que la correspondencia de carrera a profesor es **compleja**, pues cada carrera tiene, por lo general, muchos profesores.



Hay cuatro tipos de relaciones posibles entre dos tipos de artículos A y B: La correspondencia de A a B puede ser simple y la recíproca compleja. La correspondencia de A a B puede ser compleja y la recíproca simple. Ambas correspondencias pueden ser complejas o ambas pueden ser simples.



Un ejemplo donde ambas correspondencias son complejas, lo es la relación que se establece entre PROFESOR y ESTUDIANTE por la impartición de clases, ya que un profesor puede impartir clases a varios estudiantes, pero, a su vez, un estudiante puede recibir clases de varios profesores:



Las relaciones pueden tener diferentes características:

- Aunque la mayoría de las relaciones asocian dos tipos de entidades, éste no es siempre el caso. Por ejemplo, PROFESOR_HORARIO_ESTUDIANTE. Esto podría representar el hecho de que un profesor imparte clases a una cierta hora a un cierto estudiante. Esto no es lo mismo que la combinación PROFESOR_HORARIO y HORARIO_ESTUDIANTE, ya que la información de que "el profesor P5 imparte clases en el horario H1 al estudiante E4" dice más que la combinación "el profesor P5 imparte clases en el horario H1" y "el estudiante E4 recibe clases en el horario H1"
- Las relaciones pueden establecerse entre un mismo tipo de entidad. Por ejemplo, una asociación entre un profesor y otro puede venir dada por el hecho de que un profesor sea el jefe de otros profesores. A este tipo de relación frecuentemente se le llama relación recursiva
- Es importante señalar que una asociación entre entidades puede ser considerada en sí como una entidad, ya que una relación se puede ver como un objeto bien diferenciado sobre el cual se desea almacenar información.
- Entonces, un modelo de datos no es más que la representación de un caso de la realidad objetiva a través de los objetos, sus propiedades y las relaciones que se establecen entre ellos.

3.3 Ejemplo integrador

Caso: La biblioteca

En una biblioteca se desea diseñar la base de datos para el control de los préstamos de libros. De cada libro se conoce el código que lo identifica, su título y la cantidad de páginas que tiene.

Un libro se clasifica por una materia y por una materia se clasifican muchos libros. De cada materia se conoce el código que la identifica y su nombre. Los libros tienen muchos ejemplares pero un ejemplar lo es de un solo libro. De cada ejemplar se sabe su código y su estado de conservación.

Asimismo, un libro se les puede prestar a muchos usuarios y a un usuario se le pueden prestar muchos ejemplares; del usuario se conoce su DNI, nombre y apellido paterno su dirección y su ocupación

- Se determinan las entidades:
Libro

Materia
Ejemplar
Usuario

- Se determinan los atributos de cada entidad:
Libro = código que lo identifica, su título y la cantidad de páginas que tiene

Materia = código que la identifica y nombre

Ejemplar = código y estado de conservación

Usuario = DNI, nombre y apellido paterno, dirección y ocupación

- Nivel de datos:
Valores que tendrá cada atributo de una entidad

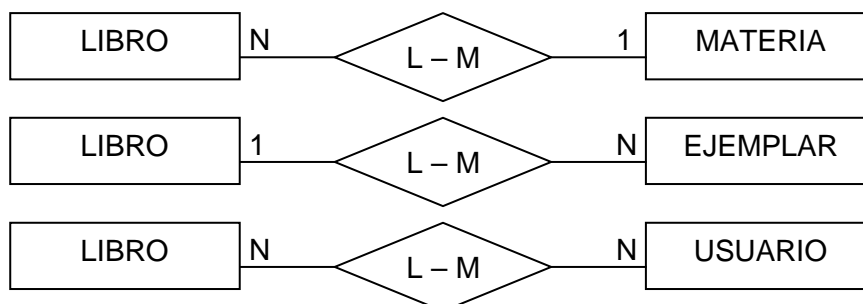
ENTIDAD: LIBRO		
CÓDIGO	TÍTULO	CANTIDAD DE PÁGINAS
L0001	BASE DE DATOS RELACIONALES	500
L0002	SISTEMAS OPERATIVOS MODERNOS	600

ENTIDAD: MATERIA	
CÓDIGO	NOMBRE
M0001	BASE DE DATOS
M0002	SISTEMAS OPERATIVOS

ENTIDAD: EJEMPLAR	
CÓDIGO	ESTADO DE CONSERVACIÓN
E001	BUENO
E002	MALO

ENTIDAD: USUARIO				
DNI	NOMBRE	APELLIDO PATERNO	DIRECCIÓN	OCUPACIÓN
45874125	JUAN	PEREZ	LA MAR 123	DOCTOR
48525641	PAULO	GUILLEN	LA JOTA 666	ABOGADO

- Relaciones de correspondencia



4. AUTOEVALUACIÓN

- a. Defina los conceptos de **base de datos** y de **sistema de gestión de bases de datos**.
- b. ¿Cuál es el principal objetivo de los SBD? Explique en qué consiste.
- c. ¿Es la minimización de la redundancia un objetivo de los SBD? Explique su respuesta.
- d. Mencione 10 entidades que usted conozca. Defina algunos atributos para cada una y decida cuál puede ser la llave primaria en cada caso.
- e. Mencione 10 relaciones que sea capaz de reconocer. Defina el tipo de éstas y algunos atributos para cada una, siempre que sea posible.
- f. Aplique la terminología explicada en clases, correspondiente al nivel del dominio de las ideas, y describa las relaciones de correspondencia existentes en los siguientes casos de la realidad objetiva. Para ello, represente en un gráfico, de acuerdo con su criterio, las entidades y sus atributos, indicando los atributos que constituyen las llaves primarias, las relaciones y el tipo de éstas, así como sus atributos.
 - f.1 Para el control de las escuelas primarias en la ciudad de Lima se tiene la siguiente información:
 De cada escuela, un número que la identifica, su nombre y su dirección. De cada aula, el número que la identifica, la cantidad de asientos que tiene y el piso en que se encuentra situada. De cada grupo de clases, un identificador del grupo, el grado escolar del grupo y la cantidad de alumnos que tiene. De cada maestro, su DNI, su nombre, su sexo y el año en que se graduó. De cada alumno, su número de expediente, su nombre, su sexo y su fecha de nacimiento.
 Una escuela tiene muchas aulas y muchos grupos de clases, pero cada aula pertenece a una escuela y lo mismo sucede con cada grupo.
 Un grupo siempre recibe clases en la misma aula y una aula pertenece a un solo grupo.
 En un grupo imparte clases un maestro y éste sólo imparte clases en un grupo. En cada grupo de clases hay muchos alumnos, pero un alumno forma parte de un solo grupo.
 - f.2 En un centro de información científica (biblioteca) se desea controlar la utilización de la bibliografía que en él existe.
 En el centro existen varias salas. De cada sala se conoce el número que la identifica, especialidad y cantidad de empleados.
 En cada sala están disponibles revistas y libros. De cada revista se conoce el código que la identifica, nombre, fecha de publicación y país de procedencia. De cada libro se conoce su código, título, editorial y país de procedencia. Cada libro y revista existente sólo se encuentra en una sala. Las revistas sólo pueden ser consultadas en las salas, sin embargo, los libros pueden ser solicitados en préstamo por los usuarios, llevándose el control de dichos préstamos. Un libro puede ser prestado a varios usuarios (durante la existencia del libro) y un usuario puede solicitar varios libros. De cada usuario se sabe su DNI, nombre, distrito en que reside y departamento en que trabaja. Para cada libro se conoce la fecha de inicio de un préstamo realizado a determinado usuario.

- f.3 En los centros de beneficio agrícola se les aplica diversos tratamientos a los tubérculos.

De cada tipo de tubérculo (papa, yuca, camote, etc.), se conoce el código que lo identifica, su descripción y la variedad a la que pertenece. Cada tipo de tubérculo puede ser enviado a distintos centros de beneficio y en un centro de beneficio pueden ser tratados distintos tipos de tubérculos. De cada centro de beneficio se sabe el código, que lo identifica, su dirección y la cantidad de trabajadores que tiene. Se sabe, para cada centro de beneficio, el plan (en kilogramos) de atención a un tipo de tubérculo.















En un centro de beneficio se aplican diferentes tratamientos y un tratamiento puede ser aplicado en diferentes centros de beneficio. Se conoce la cantidad de un tipo de tubérculo (en kilogramos) tratado en un centro de beneficio según un tratamiento dado. De cada tratamiento, se conoce el código que lo identifica, su nombre y el objetivo que persigue.


- f.4 En un hospital se desea controlar la actividad asistencial que se brinda en las consultas de la Sala de Emergencias.

En la Sala de Emergencias brindan sus servicios los médicos organizados en equipos. A un equipo pertenecen varios médicos y un médico pertenece a un equipo. De cada equipo se conoce el código que lo identifica, el nombre del jefe del equipo y la periodicidad con la que le corresponde hacer guardia al equipo. De cada médico se conoce su DNI, nombre, especialidad y categoría. Los pacientes que llegan al Cuerpo de Guardia pueden ser atendidos por varios médicos (si sus síntomas indican la necesidad de que varios especialistas lo asistan) y un médico atiende a muchos pacientes. De cada paciente se sabe su DNI, nombre, edad, sexo y ocupación. Se sabe el tiempo dedicado por el médico a la atención de un determinado paciente, así como el diagnóstico que le hizo y el tratamiento que le indicó.

Los médicos en la Sala de Emergencias pueden utilizar en su labor asistencial diferentes medios de diagnóstico (Rayos X, análisis, etc.) y un medio de diagnóstico puede ser empleado por muchos médicos. De cada medio de diagnóstico se conoce el código que lo identifica, su descripción y el costo por unidad. Para cada médico se conoce la cantidad de veces que ha ordenado la aplicación de un medio de diagnóstico dado.

Resumen

-  El procesamiento automatizado de datos ha pasado por diferentes etapas en su desarrollo hasta llegar a la actual, en la que se emplean bases de datos para el almacenamiento de la información.
-  El principal objetivo de los SBD es garantizar la independencia de los datos respecto a los programas de aplicación.
-  Otro objetivo muy importante de los SBD es la minimización de la redundancia.
-  **Entidad:** objeto del cual se describen ciertas características.
-  **Campo o atributo:** es la unidad menor de información sobre un objeto (almacenada en la base de datos) y representa una propiedad de un objeto.
-  Un atributo puede tomar diferentes valores sobre un cierto conjunto que se denomina **dominio**.
-  **Dominio:** rango de valores posibles de un atributo.
-  A un valor de un atributo definido en el dominio dado, en un cierto momento del tiempo, se le denomina **ocurrencia** del atributo.
-  Un **artículo o registro** es una colección identificable de campos asociados y representa un objeto con sus propiedades.
-  Una **ocurrencia de artículo o tupla** consiste en un grupo de ocurrencias de campos relacionados, representando una asociación entre ellos.
-  Un **archivo o archivos** es un conjunto de ocurrencias de un mismo tipo de artículo.
-  Una **base de datos** está formada por múltiples archivos.
-  Existen asociaciones o **relaciones** enlazando las entidades, que pueden tener o no atributos. Pueden establecerse sobre la misma entidad o sobre entidades diferentes. En una relación puede participar cualquier cantidad de entidades.
-  Las relaciones pueden ser de uno a uno (**1 : 1**), de uno a muchos (**1 : m**) y de muchos a muchos (**m : m**).

-  Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

✎ <http://sistemas.itlp.edu.mx/tutoriales/basedat1/>

En esta página web hallará algunos conceptos complementarios a los mostrados en el manual sobre la introducción a la base de datos.

✎ <http://sistemas.itlp.edu.mx/tutoriales/basedat1/>

Es esta página web encontrará definiciones complementarias sobre los objetivos de las base de datos.

✎ <http://sistemas.itlp.edu.mx/tutoriales/basedat1/>

Es esta página web encontrará ejercicios sobre la representación de la información.

**UNIDAD DE
APRENDIZAJE****1****SEMANA****2**

ARQUITECTURA DE UN SISTEMA DE BASES DE DATOS

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos determinarán los elementos de una base de datos y sus interrelaciones.

TEMARIO

Arquitectura de un sistema de bases de datos (SBD)

Organizaciones de archivos y el nivel interno de la arquitectura

ACTIVIDADES PROPUESTAS

1. Explicaciones por parte del facilitador combinadas con ejemplos.
2. Se efectúan preguntas de comprobación dirigidas a los alumnos sobre los temas tratados.

1. ARQUITECTURA DE UN SBD

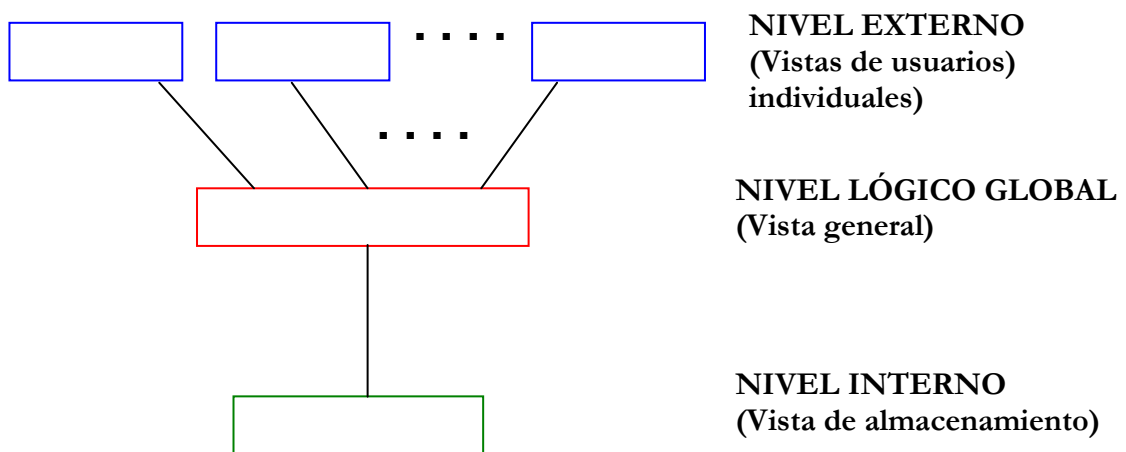
Presentaremos a continuación la arquitectura de un SBD, aunque no podemos asegurar que cualquier SBD se corresponda exactamente con ella. Sin embargo, esta arquitectura se corresponde suficientemente bien con un gran número de sistemas. Además, está de acuerdo con la arquitectura propuesta por el grupo ANSI/SPARC.

La arquitectura se divide en tres niveles generales: interno, lógico global y externo.

El nivel interno es el más cercano al almacenamiento físico, o sea, es el relacionado con la forma en que los datos están realmente almacenados.

El nivel externo es el más cercano a los usuarios, o sea, es el relacionado con la forma en que los datos son vistos por cada usuario individualmente.

El nivel lógico global es un nivel intermedio entre los dos anteriores.



Existirán varias "vistas externas" diferentes, siendo cada una representación más o menos abstracta de alguna porción de la base de datos total y existirá únicamente una "vista general", consistente en una representación también abstracta de la base de datos en su totalidad. Igualmente, existirá una única "vista interna" que representa a la base de datos completa, tal y como está realmente almacenada.

A continuación estudiaremos con mayor detalle cada uno de los niveles de la arquitectura vista anteriormente y la forma en que ellos interactúan.

1.1 El nivel externo

Es el nivel del usuario individual, donde un usuario puede ser un programador de aplicación o un usuario final con cualquier grado de sofisticación. Cada usuario tiene un lenguaje a su disposición:

- Para el programador, ese lenguaje puede ser un lenguaje de programación convencional, tal como Pascal, o un lenguaje de programación específico de un sistema, tal como el FoxPro.

- Para el usuario final, el lenguaje puede ser un lenguaje de consulta (interrogaciones, query) o un lenguaje de propósito especial, quizás basado en sistemas de menús y ventanas y construido para satisfacer los requerimientos de un usuario, encontrándose soportado por algún programa de aplicación en línea.

Es importante señalar que todo lenguaje debe incluir un sublenguaje de datos, o sea, un subconjunto del lenguaje que trata específicamente con los objetos de la base de datos y sus operaciones. Se dice que el sublenguaje de datos (DSL) está embebido dentro del correspondiente lenguaje huésped. Este lenguaje huésped es el encargado de asegurar otras facilidades ajenas a la base de datos, tales como variables locales, operaciones de cálculo, lógica if-then-else, etc. Un sistema dado, puede soportar múltiples lenguajes huésped y múltiples sublenguajes de datos.

En principio, cualquier sublenguaje de datos es realmente una combinación de, al menos, dos lenguajes subordinados: un lenguaje de definición de datos (DDL), el cual garantiza la definición o descripción de los objetos de la base de datos, y un lenguaje de manipulación de datos (DML), el que garantiza la manipulación o procesamiento de esos objetos.

Ya se ha indicado que un usuario individual estará generalmente interesado sólo en cierta porción de la base de datos completa. Aún más, la vista de esa porción será generalmente abstracta cuando se compara con la forma en que los datos están físicamente almacenados. El término definido por el comité *ANSI/SPARC* para una vista de un usuario es vista externa, la cual es el contenido de la base de datos tal y como es vista por un usuario en particular. O sea, para ese usuario, la vista externa es la base de datos.

En general, una vista externa consiste en múltiples ocurrencias de múltiples tipos de artículos externos. Un artículo externo no es necesariamente igual a un artículo almacenado.

El sublenguaje de datos del usuario se define en términos de artículos externos; por ejemplo, una operación del DML que permite recuperar artículos, generará una ocurrencia de artículos externos y no una ocurrencia de artículos almacenados.

Cada vista externa se define mediante un esquema externo, consistente, básicamente, en definiciones de cada uno de los diferentes tipos de artículos externos en esa vista. El esquema externo se escribe usando la porción del DDL del sublenguaje de datos del usuario; además, tiene que existir una definición de la correspondencia entre el esquema externo y el esquema lógico global.

1.2 El nivel lógico global

La vista lógica es una representación del contenido informativo total de la base de datos. Es una forma abstracta, en comparación con la forma en que los datos están almacenados físicamente. Esta vista puede ser muy diferente de la forma en la que los datos son vistos por un usuario en particular. La vista lógica pretende ser una vista de los datos tal como son, en lugar de cómo los usuarios están forzados a verlos por las restricciones, digamos, de un lenguaje particular o de un determinado hardware que utilicen.

La vista lógica consiste en múltiples ocurrencias de múltiples tipos de artículos lógicos. Por ejemplo, puede ser una colección de ocurrencias de artículos de departamentos, más una colección de ocurrencia de artículos de empleados, etc. Un artículo lógico no es necesariamente igual a un artículo externo ni a un artículo almacenado.

La vista lógica se define mediante el esquema lógico que incluye las definiciones de cada uno de los diferentes tipos de artículos lógicos. El esquema lógico se describe usando otro lenguaje de definición de datos: el DDL lógico. Si se desea lograr la independencia de los datos, entonces las definiciones del DDL lógico no deben comprender ninguna consideración sobre la estructura de almacenamiento ni la estrategia de acceso. Ellas tienen que ser definiciones sólo referentes al contenido informativo.

Si el esquema lógico logra verdaderamente la independencia de los datos, entonces los esquemas externos que se definen sobre el esquema lógico lograrán también, necesariamente, la independencia de los datos.

La vista lógica es entonces una vista del contenido total de la base de datos y el esquema lógico es una definición de esa vista. Sin embargo, el esquema lógico no es simplemente un conjunto de definiciones como las que se encuentran, por ejemplo, en un programa Pascal. Las definiciones en el esquema lógico deben incluir una gran cantidad de aspectos adicionales, tales como los chequeos de protección y los chequeos de integridad.

En la mayoría de los sistemas actuales, el esquema lógico es realmente sólo un poco más que la simple unión de todos los esquemas externos individuales, posiblemente con la adición de algunos chequeos simples de protección e integridad. Sin embargo, está claro que los sistemas del futuro soportarán un nivel lógico mucho más sofisticado, que permita también describir la forma en que se usan los datos, como fluyen de un punto a otro, para qué se usan en cada punto, a qué controles son sometidos, etc.

1.3 El nivel interno

La vista interna es una representación de bajo nivel de la base de datos completa, que consiste en múltiples ocurrencias de múltiples tipos de artículos internos.

"Artículo interno" es el término definido por ANSI/SPARC para la construcción que hasta ahora hemos llamado artículo almacenado. La vista interna está entonces aún a un paso del nivel físico, ya que ella no opera en términos de artículos físicos (también llamados páginas o bloques) ni con consideraciones específicas de los equipos, tales como tamaños de sectores o pistas. Básicamente, la vista interna asume un espacio de dirección lineal infinita. Los detalles de cómo se hace corresponder ese espacio con el almacenamiento físico son muy específicos de un sistema y deliberadamente se omitieron de la arquitectura.

La vista interna se describe mediante el esquema interno, el cual no sólo define los diferentes tipos de artículos almacenados, sino que también especifica los índices que existen, la representación de los campos almacenados, la secuencia física en que están los artículos almacenados,

etc. El esquema interno se describe usando otro lenguaje de definición de datos: el DDL interno.

1.4 Correspondencias entre los niveles de la arquitectura

En el esquema presentado de la arquitectura de un SBD, se observan los niveles de correspondencias, una entre los niveles externo y lógico global y otra entre los niveles lógico global e interno.

La correspondencia lógica/interna especifica la forma en que los artículos y campos lógicos se representan en el nivel interno. Si se cambia la estructura de la vista interna, o sea, si se hace un cambio en el esquema interno, entonces la correspondencia lógica/interna tiene también que cambiar en consecuencia, de modo que el esquema lógico permanezca invariable. En otras palabras, los efectos de estos cambios deben ser aislados por debajo del nivel lógico para que se mantenga la independencia de los datos.

Existe también una correspondencia externo/lógica entre cada vista externa particular y la vista lógica. Las diferencias que pueden existir entre estos dos niveles son similares a las que pueden existir entre las vistas lógica y la interna. Por ejemplo, los campos pueden tener diferente tipos de datos, se pueden cambiar los nombres de artículos y campos, múltiples campos lógicos pueden ser combinados en un único campo externo, etc. Puede existir al mismo tiempo cualquier cantidad de vistas externas; cualquier cantidad de usuarios puede compartir una vista externa dada; las diferentes vistas externas se pueden solapar. Algunos sistemas permiten la definición de una vista externa a partir de otra (mediante una correspondencia externa/externa); esta característica es útil cuando varias vistas externas están estrechamente relacionadas entre sí.

1.5 El administrador de la base de datos (DBA)

Por último, en el esquema aparece el administrador de la base de datos, quien es la persona o grupo de personas responsable del control total de todo el sistema. Entre las **tareas del administrador de la base de datos** está:

- decidir el contenido informativo de la base de datos
- decidir la estructura de almacenamiento y la estrategia de acceso
- garantizar el enlace con los usuarios
- definir los chequeos de autorización y procedimientos de validación
- definir la estrategia de reorganización de la base de datos para aumentar la eficiencia del sistema

El SGBD interactúa con cada uno de los niveles y las correspondencias entre ellos.

2. ORGANIZACIONES DE ARCHIVOS Y EL NIVEL INTERNO DE LA ARQUITECTURA

Físicamente, se puede decir que las bases de datos se almacenan siguiendo diferentes organizaciones de archivos. Cada una de éstas tiene distintas características de desempeño, ninguna se puede decir que sea óptima para todas las aplicaciones, sino que se decide emplear una u otra en dependencia de la aplicación.

A continuación presentaremos, a grandes rasgos, las principales características de las más usadas organizaciones de archivos, de modo que puedan tener ideas de cómo se pueden organizar los datos en estos archivos para lograr determinados objetivos. Pero, antes de ello, abordaremos algunas definiciones y características relativas a los archivos.

Empezaremos diciendo que la utilización de los archivos se debe a dos causas fundamentales:

- Manejo de mucha información
- Almacenamiento de información permanente

Ya que el trabajo en memoria interna es muy rápido, pero es un recurso relativamente escaso y caro, y la memoria periférica es más barata, aunque más lenta. Además, influye en esto, el hecho de que el contenido de la memoria central se pierde al cesar el fluido eléctrico, lo que hace aconsejable almacenar en soportes externos los grandes volúmenes de información.

2.1 Operaciones fundamentales sobre archivos

Las operaciones fundamentales que se realizan sobre archivos pueden colocarse en dos grandes grupos:

2.1.1 Acceso

Se entiende el acceso a un archivo como la forma por la cual es posible tener conocimiento de la información contenida en los respectivos registros, o sea, el modo como se pueden leer los registros del archivo, las ocurrencias de estos registros almacenadas en él.

Las diferentes formas de acceder un archivo pueden agruparse en dos tipos:

- **En secuencia o secuencial:** se caracteriza por el hecho de que, en la lectura del archivo, es obligatorio que, a continuación del tratamiento del registro de orden n , se traten los de orden $n+1$, $n+2$, ... hasta un límite determinado. Puede realizarse desde el inicio del archivo o a partir de un registro n .
- **Aleatorio o directo:** se caracteriza porque se lee, y consecuentemente se trata, cualquier registro en cualquier orden (a través de la llave o una transformación de ésta). No puede realizarse en periféricos de acceso secuencial.

2.1.2 Actualización

La mayoría de los archivos deben ser actualizados con el decursar del tiempo para que se ajusten a la realidad. La actualización de un archivo se realiza mediante tres operaciones fundamentales:

- **Alta:** se crea un nuevo registro en el archivo.
- **Baja:** se elimina un registro. Puede ser lógica o física.
- **Modificación:** se cambia alguna información en el registro.

Para realizar las actualizaciones es importante validar los datos que se van a actualizar.

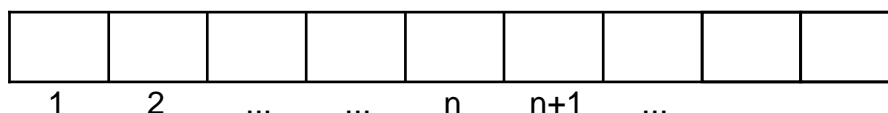
2.2 Organización de los archivos

Es necesario indicar que la bibliografía existente al respecto asume diferentes posiciones respecto a la terminología a emplear. En muchos casos las definiciones son ambiguas y hay poco acuerdo entre los textos que tratan este tema.

A pesar de esto, en la actualidad, las organizaciones generalmente aceptadas son:

2.2.1 Organización Secuencial

Es la forma más sencilla para almacenar los registros de un archivo, uno después de otro, a continuación del otro. Todos los registros se almacenan por su posición: uno es el primero, el siguiente es el segundo y así sucesivamente. Es la más vieja forma de organizar un archivo y fue empleada para las cintas magnéticas desde los inicios de la computación.



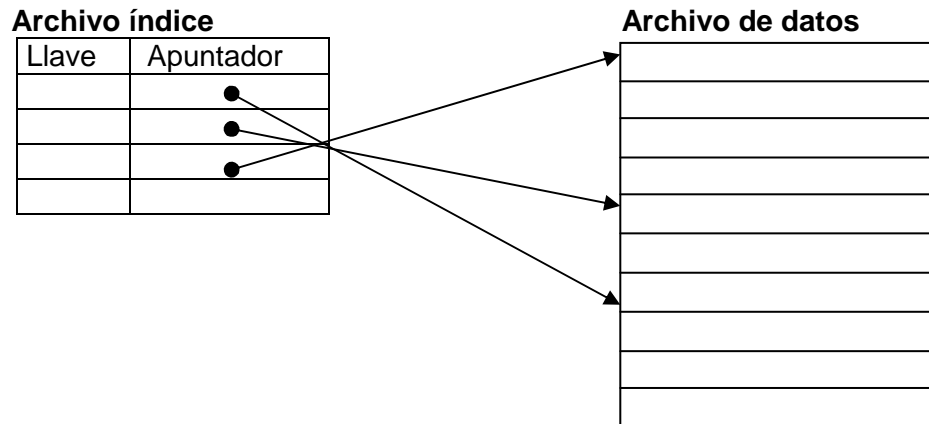
Ejemplo:

Archivo PROFESOR, en el que están las ocurrencias de profesores una a continuación de la otra:

María Álvarez	Av. Javier Prado Oeste #568	9461 5378
Felipe González	Av. Salaverry #3540	9931 2095
José Martínez	Av. Angamos Este #653	9345 7598
Juan Benavides	Prolongación Primavera #342	9621 5337
...

2.2.2 Organización Indizada

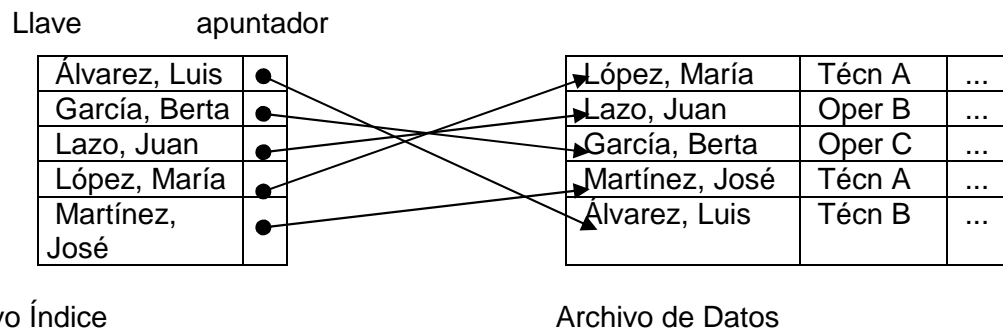
Los registros almacenados se accesan a través de un índice. La forma básica de un índice incluye una llave de registro y la dirección de almacenamiento para éste. Para encontrar un registro específico, se rastrea primero el índice y, al encontrar la dirección, se accesa directamente el registro.



Entonces, un **índice** es un archivo auxiliar que se utiliza para acceder los registros de otro archivo, que llamaremos archivo principal o de datos, por el valor de un dato o conjunto de datos, que es la **llave o clave de indización**.

A los registros del índice se les llama **entradas**. Cada entrada corresponde a un valor o intervalo de valores de la llave y es el padre cuyos hijos son los registros del archivo principal en los cuales la llave toma el valor, o un valor del intervalo de valores, correspondiente a la entrada en cuestión.

En la siguiente figura se muestra un archivo índice muy simple en el que se tiene la llave y un puntero al dato en el archivo principal.



Para un mismo archivo principal puede haber varios índices, correspondientes a diferentes claves de indización.

Si la llave de indización para un archivo índice es una llave primaria, entonces al índice se le llama **primario**. Si la llave de indización es algún atributo que no identifica unívocamente cada ocurrencia, se dice que el índice es **secundario**. En este caso, para un valor de la llave existen, en general, varios registros que tienen ese valor en el atributo correspondiente. Por ejemplo, en un archivo de PRODUCTOS el índice por número de producto (NUMPROD) sería primario y un índice por tipo de producto (TIPOPROD) sería secundario.

En muchos sistemas informativos los índices secundarios son numerosos.

Se dice que un índice es **denso** si tiene una entrada para cada valor de la clave que aparece, por lo menos, en algún registro del archivo principal, lo que no quiere decir que tenga una entrada para cada uno de estos registros. El índice primario es

generalmente un índice denso, es decir, contiene todos los valores de la clave primaria.

Por ejemplo, en la figura anterior, el índice es denso, porque existe una entrada en el índice para cada valor diferente de la llave de indización. Para el caso del archivo de datos de la figura anterior, el siguiente índice que considera como llave el cargo del trabajador también es denso pues, de igual forma, existe una entrada en el índice **para cada valor diferente** de la llave de indización

Archivo índice






Oper B
Oper C
Técn A
Técn B

Un índice **no denso** necesita que la clave sea campo de ordenamiento del archivo de datos, pues así es posible acceder los registros, cuyo valor de la llave no aparece en ninguna entrada, para lo cual se accesa el registro apuntado por la entrada con un valor de la clave más próximo por defecto (por ejemplo) al del registro buscado y, luego, se va recorriendo secuencialmente el archivo principal en orden creciente de los valores de la llave, desde aquel registro, hasta encontrar el registro buscado, si existe, o hasta encontrar uno con valor más alto de la llave, en cuyo caso no existe. Este es el caso de la organización secuencial indizada.

3. AUTOEVALUACIÓN

- Mencione los niveles en que se divide la arquitectura de un SBD y las características fundamentales de cada uno de ellos.
- Explique cómo se establece la correspondencia entre los distintos niveles de la arquitectura de un SBD y cómo se logra garantizar la independencia de los datos respecto a los programas de aplicación.
- Mencione tres de las funciones principales que realiza un administrador de bases de datos.
- Explique cuáles son las principales operaciones que se realizan sobre los archivos y los distintos tipos de cada una de ellas.
- Describa, a grandes rasgos, los tres tipos de organización de archivos más empleados.
- Defina un archivo índice. ¿Cuáles estructuras de archivos índice conoce? Explique las características de cada una.
- ¿Cuál es la relación del nivel interno de la arquitectura de un SBD con los archivos y las organizaciones de éstos?

Resumen

-  La arquitectura de un SBD está compuesta por tres niveles: externo, lógico global e interno. El nivel lógico global de los datos representa el contenido informativo total de la base de datos y, desde el punto de vista del diseñador, es el más importante.
-  Existe una correspondencia entre el nivel externo y el lógico global; y entre el lógico global y el interno. Para garantizar la independencia de los datos respecto a los programas de aplicación, cualquier cambio en el nivel interno debe reflejarse adecuadamente en la correspondencia interna/lógico global, para no afectar el nivel lógico global.
-  Las principales operaciones que se realizan sobre los archivos son: acceso, que puede ser secuencial o directo, y actualización, que se refiere a las altas, bajas y modificaciones de la información.
-  El nivel interno de la arquitectura de un SBD puede considerarse estructurada por archivos con distintos modos de organización.
-  Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

 <http://www3.uji.es/~mmarques/f47/apun/node33.html>

En esta página web hallará algunos conceptos complementarios a los mostrados en el manual sobre la arquitectura de una base de datos.

 <http://www.mitecnologico.com/Main/EstructuraArchivosOrganizacionSecuencialIndexado>

En esta página web encontrará definiciones complementarias al almacenamiento secuencial indexado.

**UNIDAD DE
APRENDIZAJE****2****SEMANA****3**

MODELAMIENTO CONCEPTUAL

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos diseñan el diagrama entidad relación (DER) de un proceso de negocio a partir de casos planteados por el profesor, relacionando las características del modelo conceptual con el diseño de una base de datos.

TEMARIO

Características del modelo conceptual

El modelo entidad-relación

ACTIVIDADES PROPUESTAS

1. Explicaciones por parte del facilitador combinadas con ejemplos.
2. Se efectúan preguntas de comprobación dirigidas a los alumnos sobre los temas tratados.
3. Se plantea un caso para identificar los componentes de un Diagrama Entidad Relación y se construye el modelo correspondiente.

1. CARACTERÍSTICAS DEL MODELO CONCEPTUAL

El proceso de diseño de la base de datos transita a través de una serie de pasos en los cuales se va avanzando de un nivel de abstracción menor a otro más profundo, mediante la elaboración de una sucesión de modelos. En los últimos años se ha generalizado la concepción del diseño de las base de datos propuestas por el grupo *ANSI/SPARC*, la cual constituye, al mismo tiempo, una arquitectura para los SBD, tal y como la acabamos de estudiar.

Hemos visto en esta arquitectura que cada nivel de la misma es una cierta forma de representación abstracta de la información y una de las funciones más importantes del SGBD consiste precisamente en permitirle al usuario la interacción con los datos en estos términos abstractos, en lugar de tenerlo que hacer directamente con la forma en que esos datos están físicamente almacenados. Es por ello que, al acometerse la tarea de diseño de una base de datos, la atención se debe centrar en el aspecto lógico de la información, ya que los detalles relacionados con el almacenamiento físico son parte de todo SGBD comercial que se utilice, y por tanto, no pueden ser modificados.

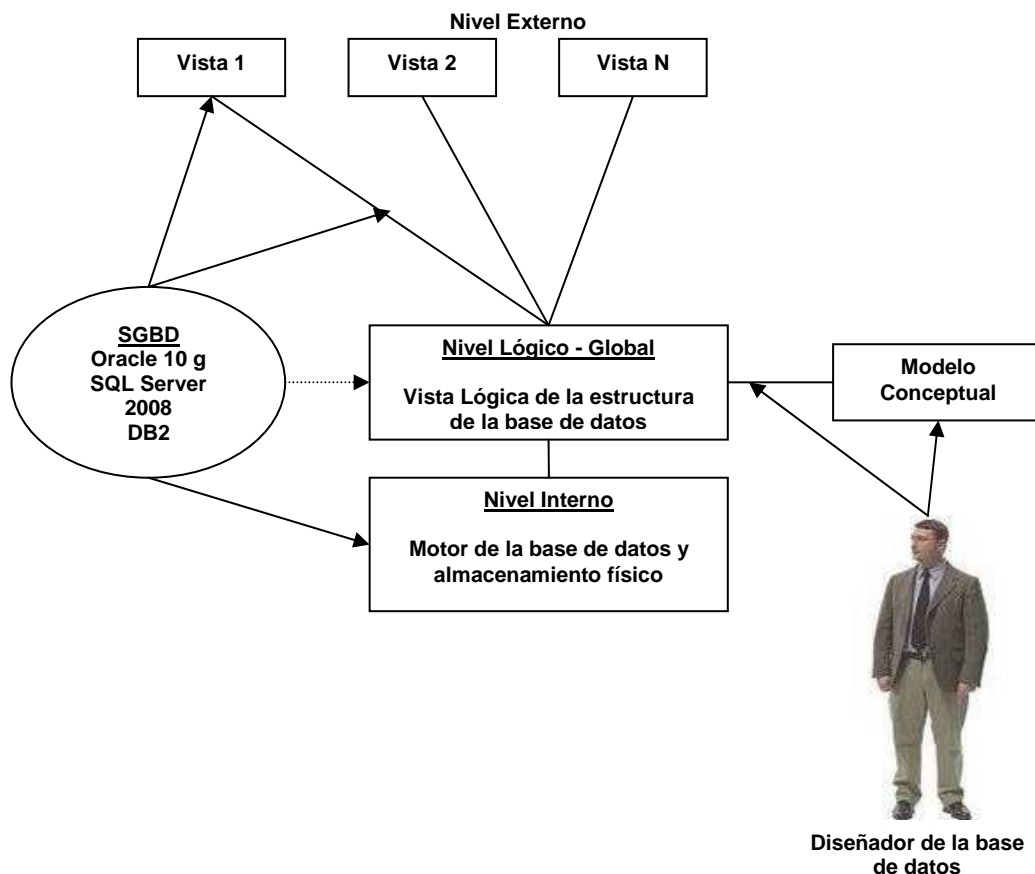
Los SGBD existentes utilizan diferentes modelos de datos para la representación en el nivel lógico global. Son comunes a todos ellos las siguientes características:

- La representación de la información se basa en el uso de determinadas estructuras de datos que poseen una capacidad descriptiva limitada; sólo diferencian un rasgo semántico: el tipo de proyección (1:1, 1:n, n:m).
- Utilizan una terminología que no es familiar al usuario del sistema, por lo que dificultan la comunicación usuario - diseñador.

Además, cada uno de estos modelos está vinculado con un tipo particular de SGBD.

Por todo ello, es necesario tratar con otro tipo de modelo cuando se aborda el problema del diseño de la base de datos, el cual debe superar los problemas anteriores y constituye un nivel de abstracción intermedio entre la realidad informativa y el nivel lógico global de la arquitectura. A este nuevo tipo de modelo se le denomina modelo conceptual. O sea, el modelo conceptual se define exteriormente al SGBD, realizándose, manualmente o con la ayuda de algún sistema automatizado, la transformación entre el modelo conceptual y el lógico global.

A continuación se mostrará los respectivos niveles:



El proceso de modelamiento conceptual es denominado también modelamiento semántico, ya que con estos modelos se pretende reflejar en mayor medida la semántica o significado de los datos y sus interrelaciones.

2. EL MODELO ENTIDAD-RELACIÓN

Este modelo (presentado mediante un diagrama y denominado DER) fue propuesto en 1976 y ha encontrado una amplia aceptación como instrumento para modelar el mundo real en el proceso de diseño de las bases de datos. El modelo opera con los conceptos de entidad y relación que estudiamos anteriormente.

Las ocurrencias de entidades se clasifican en distintas entidades E_i , tales como "empleado", "departamento", etc. Existirá un predicado asociado con cada entidad que permitirá comparar si una ocurrencia arbitraria pertenece a una entidad dada. Las ocurrencias pueden pertenecer a más de una entidad, o sea, las entidades no son mutuamente disjuntas. Por ejemplo: Una ocurrencia de la entidad "mujeres" también pertenece a la entidad "persona".

Una relación es una relación matemática entre n entidades.

$$\{ (e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n \}$$

y cada elemento de esa relación es una ocurrencia de relación (e_1, e_2, \dots, e_n) , donde las E_i y e_i no tienen que ser necesariamente diferentes. El rol de una entidad en una relación expresa la función que desempeña dicha entidad en la relación. En la relación "matrimonio" definida entre ocurrencias de la entidad "persona", o sea,

$$\text{"matrimonio"} = \{(e_1, e_2) \mid e_1 \in \text{"persona"}, e_2 \in \text{"persona"}\},$$

el primer elemento en el tuplo puede aparecer en el rol de "esposo" y el segundo, en el rol de "esposa".

Información adicional sobre una entidad (además de los predicados y las relaciones) se obtiene mediante los atributos asociados con la entidad. Ejemplos de valores que pueden tomar los atributos son: "rojo", "3", "Juan", etc. y ellos se clasifican en dominios mutuamente disjuntos, tales como "color", "edad", "nombre", etc.

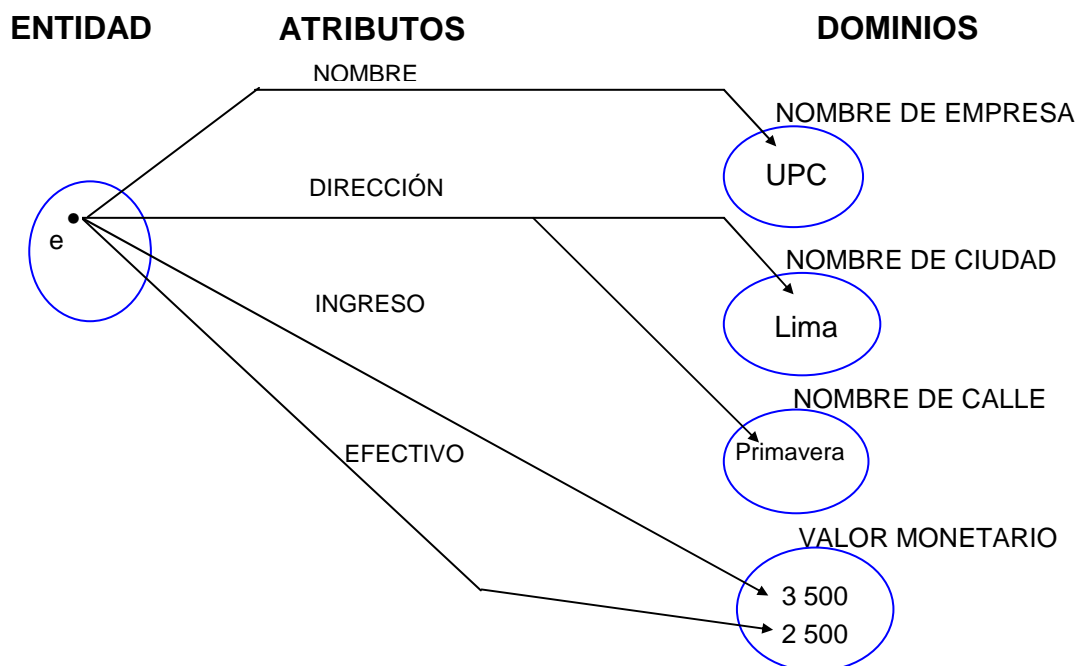
Un valor de un dominio puede ser equivalente a otro valor en un dominio diferente. Por ejemplo, "100" en el dominio "centímetros" es equivalente a "1" en el dominio "metros".

Un atributo se define en el modelo entidad – relación, como una función matemática que establece una correspondencia desde una entidad o relación hacia un dominio o un producto cartesiano de dominios:

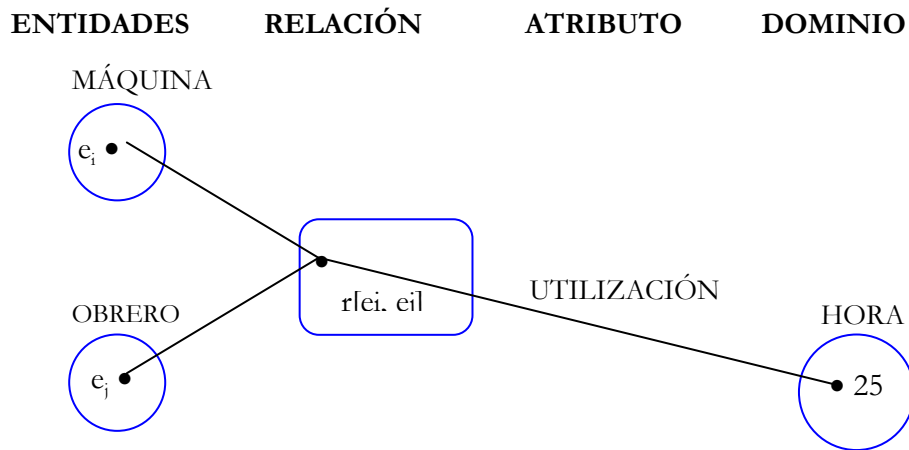
$$\text{atrib}_1: E_i \rightarrow D_{i1} \times D_{i2} \times \dots \times D_{in}$$

$$\text{atrib}_2: R_i \rightarrow D_{i1} \times D_{i2} \times \dots \times D_{in}$$

En la figura siguiente se muestran los atributos definidos para la entidad EMPRESA. El atributo NOMBRE hace corresponder a las ocurrencias de empresa con elementos del dominio NOMBRE DE EMPRESA. El atributo DIRECCIÓN establece una correspondencia desde la entidad EMPRESA hacia el par de dominios NOMBRE DE CIUDAD, NOMBRE DE CALLE. INGRESO Y EFECTIVO establecen ambos una correspondencia desde la entidad EMPRESA hacia el dominio VALOR MONETARIO. Nótese que un atributo se define siempre como una función, por lo que siempre hace corresponder a una ocurrencia dada con un único valor de una tupla, pues se define un producto cartesiano de dominios.



Las relaciones pueden también tener atributos. En la figura siguiente, el atributo UTILIZACIÓN define el número de horas que un obrero específico e_j usa una máquina e_i y constituye un atributo de la relación correspondiente. Él no es ni un atributo del OBRERO ni de la MÁQUINA, ya que su significado depende de la relación entre ellos dos.



Es importante destacar las siguientes características de los atributos en este modelo:

- Los atributos sólo son correspondencias funcionales. Así, por ejemplo, si tenemos la entidad AUTOMÓVIL y el atributo COLOR, el hecho de que un auto pueda tener más de un color no se puede representar como un atributo en este modelo.
- El único hecho que puede ser registrado sobre los valores en este modelo es su pertenencia a un dominio. Si se desea representar otra propiedad, el atributo asociado tiene que ser convertido en una entidad. Por ejemplo, si queremos registrar la longitud de onda de cada color no podemos hacerlo en el modelo entidad - relación, sino convirtiendo el atributo COLOR en una entidad.

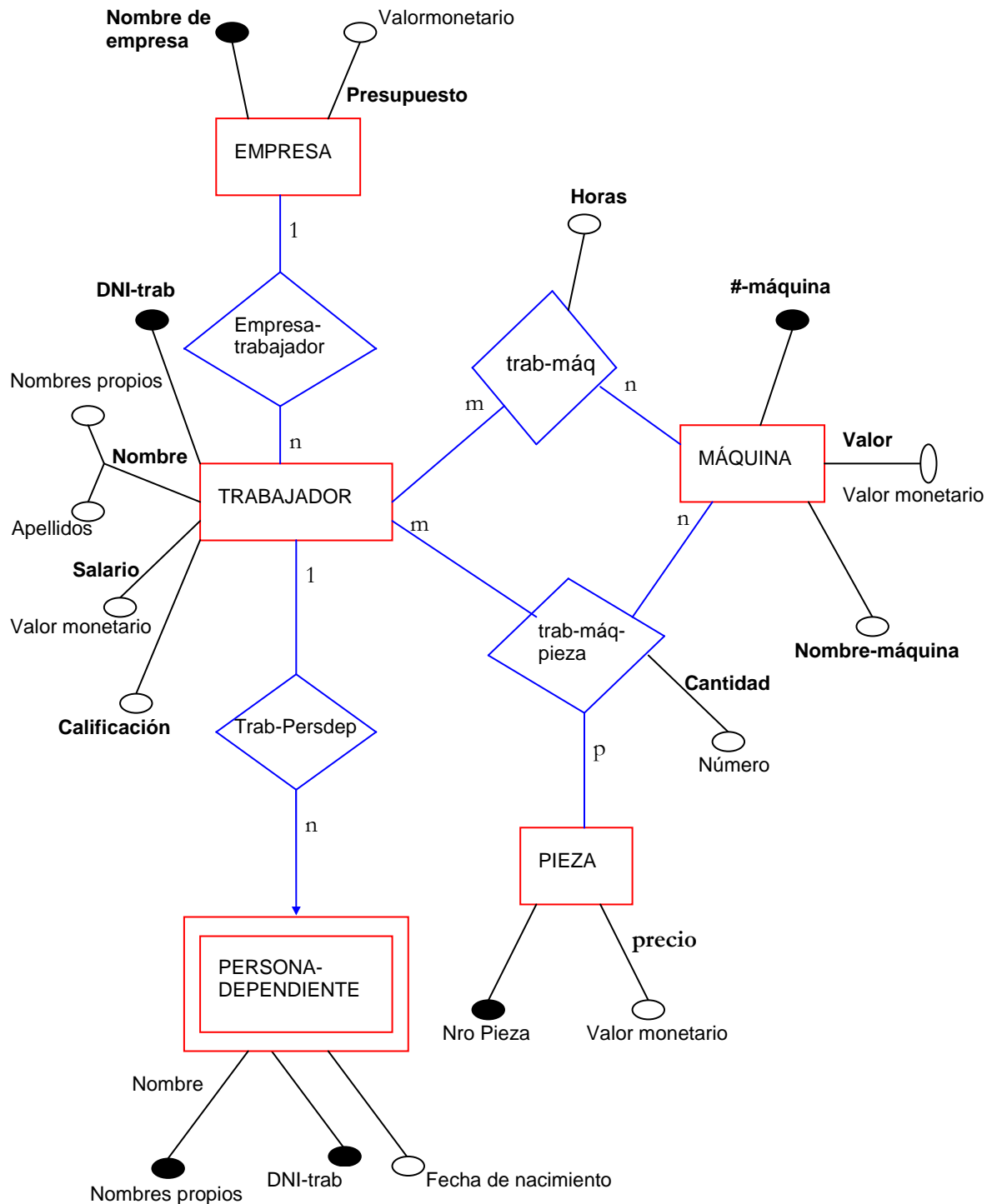
El modelo entidad - relación tiene asociada una representación gráfica denominada **Diagrama Entidad-Relación (DER)**.

En un DER, cada entidad se representa mediante un rectángulo, cada relación mediante un rombo y cada dominio mediante un círculo. Mediante líneas se conectan las entidades con las relaciones, igual que las entidades con los dominios, representando a los atributos.

Los atributos llaves de las entidades se representan mediante un círculo relleno.

En ocasiones, una entidad no puede ser identificada únicamente por el valor de sus propios atributos. En estos casos, se utilizan conjuntamente las relaciones con los atributos para lograr la requerida identificación unívoca. Estas entidades reciben el nombre de entidades débiles y se representan en el DER con un doble rectángulo. El modelo entidad - relación restringe las relaciones a usar para identificar las entidades débiles a relaciones binarias de, a lo sumo, 1:n. Así, por ejemplo, una ocurrencia de **trabajador** puede tener n ocurrencias **persona-dependiente**

asociadas, donde, además, la existencia de una ocurrencia en la segunda entidad depende de la existencia de una ocurrencia que le corresponda en la primera entidad. Por ejemplo, en el modelo que se representa en el DER de la figura, habrá personas dependientes de un trabajador sólo si ese trabajador existe. Para indicar esa dependencia en la existencia se usa una saeta en el DER. La llave de una entidad débil se forma combinando la llave de la entidad regular que la determina con algún otro atributo o conjunto de atributos de la débil, que definan unívocamente cada entidad débil asociada a una entidad regular dada (una entidad se denomina regular si no es débil).



Para cada relación se determina su tipo (simple o complejo) y en el DER se escribe el tipo de correspondencia. Por ejemplo, una empresa puede tener varios (**n**) trabajadores asociados y un trabajador pertenece a una sola (**1**) empresa. En la relación trab-máq-pieza, un trabajador puede trabajar en **n** máquinas, produciendo **p** piezas, una pieza puede ser producida por **m** trabajadores en **n** máquinas y en una máquina pueden trabajar **m** trabajadores produciendo **p** piezas. Aquí, **m**, **n** y **p** no identifican un número específico, sino solamente el tipo de correspondencia que se establece en la relación (muchos).

Aunque en el modelo entidad - relación se define que la llave de una relación es la combinación de las llaves de todas las entidades asociadas, es conveniente, desde ahora, analizar más profundamente esto.

En una relación de muchos a muchos (**n:m**), efectivamente, la llave de la relación está formada por **las llaves de las entidades que participan** en la relación, pues, como a cada ocurrencia de una de las entidades le corresponden varias ocurrencias de la otra entidad y viceversa, es preciso utilizar la identificación de cada una de las entidades que participan en la relación para referirse a una sola ocurrencia de cada una de ellas y, así, referirse a una ocurrencia de la relación. Por ejemplo, en la relación **trab-máq** la llave será **DNI-trab, #-máquina**.

Pero en una relación de muchos a uno (**m:1**), la llave de la relación es **la llave de la entidad del extremo muchos (m)**, pues a cada ocurrencia de esa entidad le corresponde sólo una ocurrencia de la entidad del otro extremo, por lo que, con la llave de la entidad del extremo muchos está perfectamente determinada también una única ocurrencia de la entidad del extremo 1. Por ejemplo, en la relación **Empresa-trabajador** la llave será **DNI-trab**.

De modo similar, en una relación de uno a uno (**1:1**) la llave de la relación está formada por **la llave de cualquiera de las dos entidades que participan**, pues a una ocurrencia de una de ellas le corresponde sólo una ocurrencia de la otra y viceversa, por lo que, con la llave de una de las entidades está perfectamente determinada también una única ocurrencia de la otra entidad. Por ejemplo, si en un centro de trabajo un trabajador, que se identifica con **dnitrab**, es jefe de un piso, que se identifica con **númpiso**, y un piso tiene un jefe, es decir, que la relación es de 1:1, entonces la llave de la relación puede ser o **dnitrab** o **númpiso**.

Una entidad se puede relacionar consigo misma. A estas relaciones se les llama, usualmente, recursivas o cíclicas.

Es posible extender la capacidad semántica del modelo entidad - relación aplicando sobre sus objetos básicos (entidad y relación) diferentes operaciones, tales como:

- **Generalización:** Permite formar una nueva entidad, mediante la unión de otras entidades. El proceso inverso se denomina **especialización** y divide una entidad en cierto número de otras entidades.
- **Agregación:** Construye una nueva entidad sobre la base de una relación.

A las entidades, relaciones y conjuntos definidos hasta ahora les llamaremos tipos básicos para distinguirlos de los nuevos tipos de datos que se obtendrán con las operaciones anteriores.

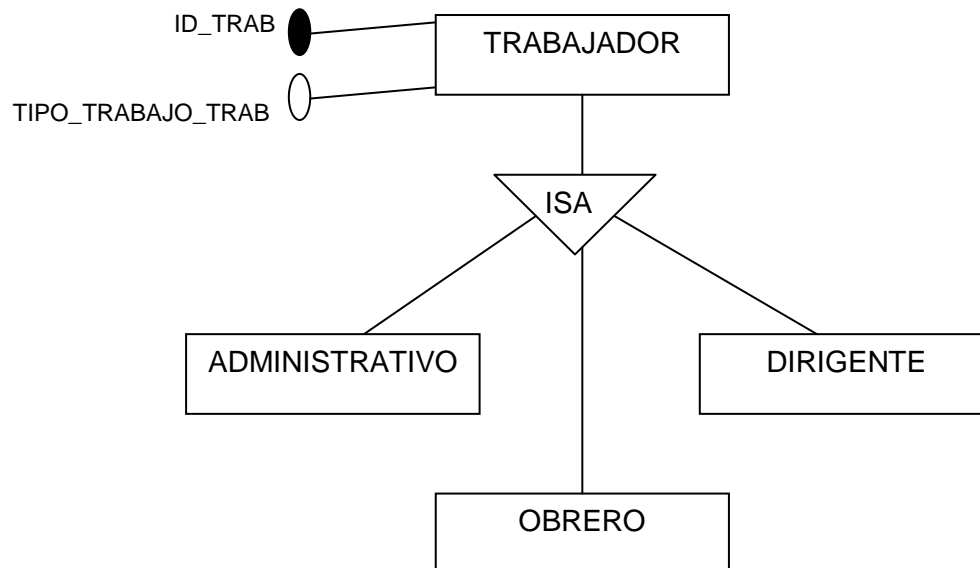
Veamos cada una de las operaciones:

2.1 Generalización / Especialización

Si T_1, T_2, \dots, T_n son entidades (que pueden a su vez ser resultado de una generalización), la generalización define una nueva entidad T con el siguiente significado:

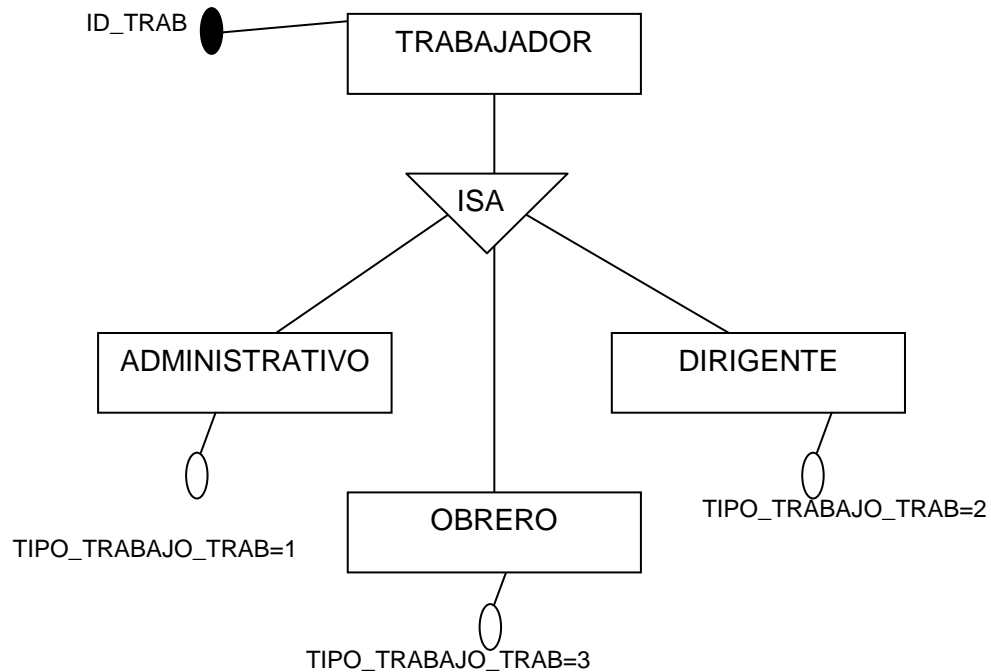
$$T = \{t \mid t \in T_i, 1 \leq i \leq n\}$$

o sea, para cada ocurrencia t en T existe, al menos, un conjunto T_i que contiene a esa ocurrencia. Por ejemplo, en el DER anterior, puede ser necesario distinguir los trabajadores de una empresa de acuerdo a su ocupación como obreros, dirigentes y administrativos. Esto no puede ser representado en el modelo que está representado en el DER de la figura anterior y sólo a través del hecho de que la entidad "obrero", por ejemplo, es siempre (o sea, en todo momento) un subconjunto de la entidad "trabajador", podemos deducir cierta clase de dependencia entre los dos tipos. Usando la generalización podemos obtener un nuevo diagrama como se muestra parcialmente en la figura siguiente:



Nótese que hemos introducido un nuevo atributo para la entidad trabajador. Este atributo nos permite distinguir entre los miembros de diferentes clases de trabajadores.

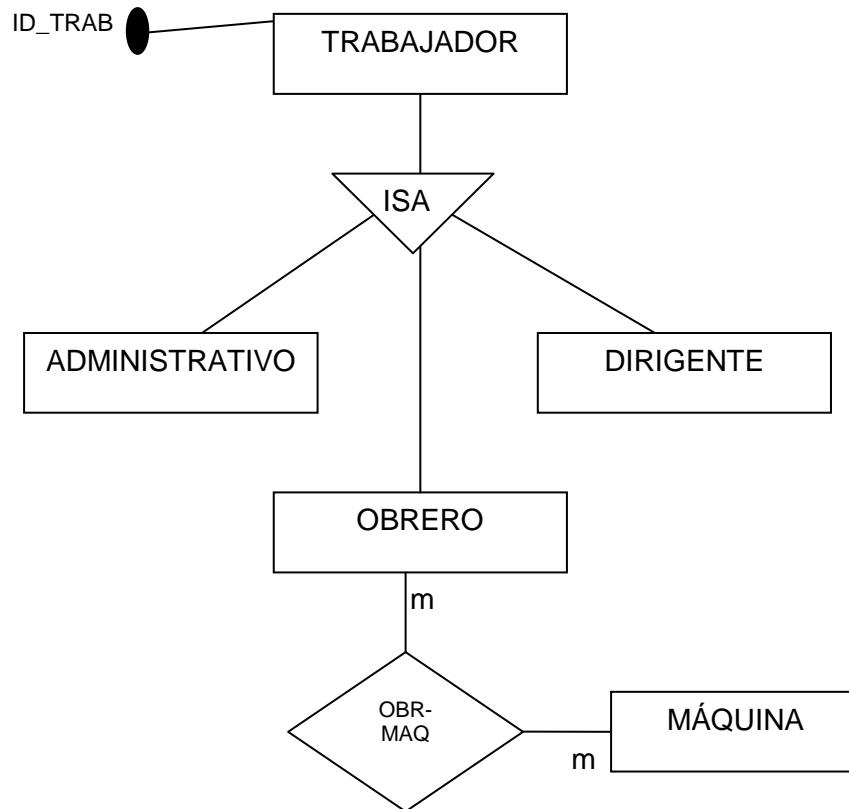
Si tenemos una entidad **TRABAJADOR** y queremos usar la operación de Especialización como inversa a la generalización, tenemos que especificar "roles" en el modelo, o sea, reglas que definan cuándo una ocurrencia de **TRABAJADOR** pertenece a uno u otro componente de la entidad. Entonces la representación de esta operación en el DER se generaliza como se muestra en la figura siguiente:



Si para cada ocurrencia de la entidad Trabajador nosotros podemos siempre deducir a cuál entidad componente pertenece usando alguna propiedad ya representada, entonces no es necesario introducir un nuevo atributo Tipo de Trabajo.

Las reglas que definen la especialización de una entidad se denominan "caracterizaciones". Por ejemplo, Tipo de Trabajo = 1 es la caracterización de la entidad Administrativo dentro de la entidad Trabajador.

En una Generalización / Especialización, los atributos y relaciones de la entidad "generalizada" son heredados por las entidades componentes (entidades especializadas). La llave de una especialización es la llave de la generalización. Además, se pueden definir nuevos atributos y relaciones para cada entidad especializada. Por ejemplo, la relación Obrero-Máquina se define ahora sólo para la entidad especializada Obrero, componente de la entidad generalizada Trabajador:



Si bien es cierto que, según lo visto anteriormente, las operaciones de Generalización y Especialización pueden denotarse de modo diferente, no es menos cierto que, con la notación empleada para la generalización, pueden expresarse las entidades generalizadas y especializadas perfectamente y es ésta la empleada normalmente.

Sí es importante agregar algo más a lo visto hasta ahora para poder expresar las siguientes situaciones que se presentan:

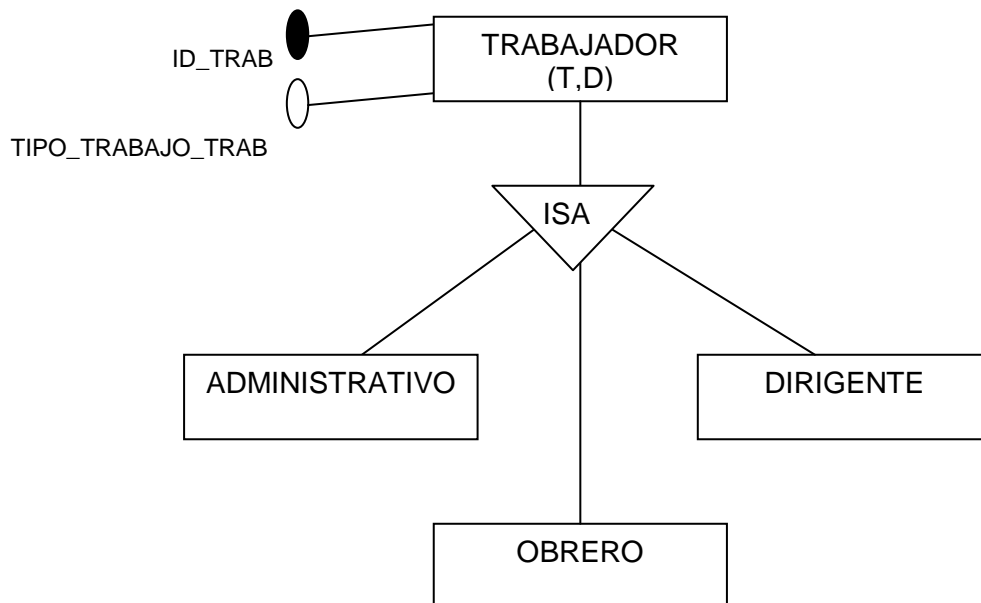
Las ocurrencias de las especializaciones pueden abarcar o no el universo de las ocurrencias de la generalización, es decir, la totalidad de las ocurrencias de la generalización puede o no estar contenidas en alguna o algunas de las especializaciones. Por lo tanto, las especializaciones pueden ser totales (T) o parciales (P).

Una ocurrencia de la generalizada puede o no estar en más de un conjunto T_i , o lo que es lo mismo, la intersección entre algunos de los conjuntos T_i puede o no ser vacía. Es decir, las especializaciones pueden ser solapadas (S) o disjuntas (D).

Es por ello que, en el DER, se añade en cada generalización, entre paréntesis, la especificación:

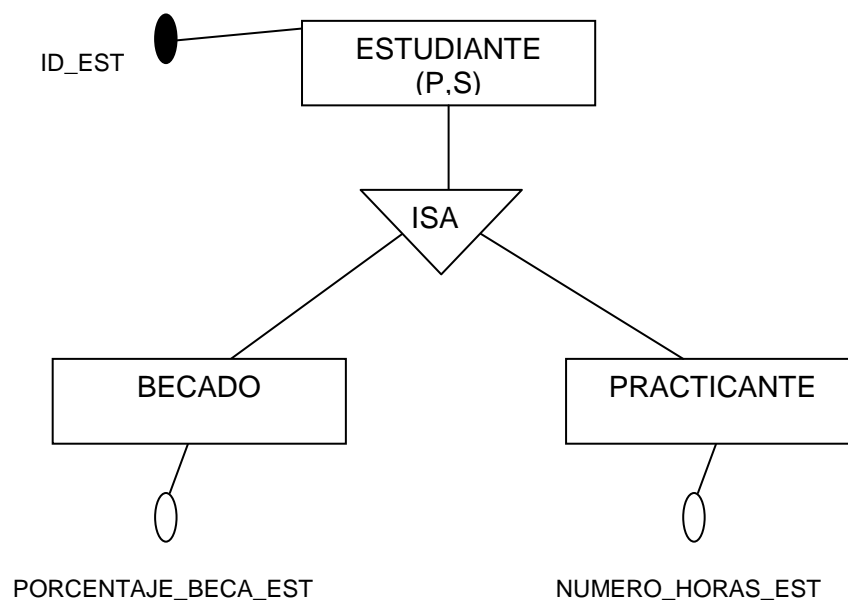
- (T, S): indicando que la especialización realizada es total y solapada
- (T, D): indicando que la especialización realizada es total y disjunta
- (P, S): indicando que la especialización realizada es parcial y solapada
- (P, D): indicando que la especialización realizada es parcial y disjunta

Entonces, el ejemplo visto anteriormente quedaría:



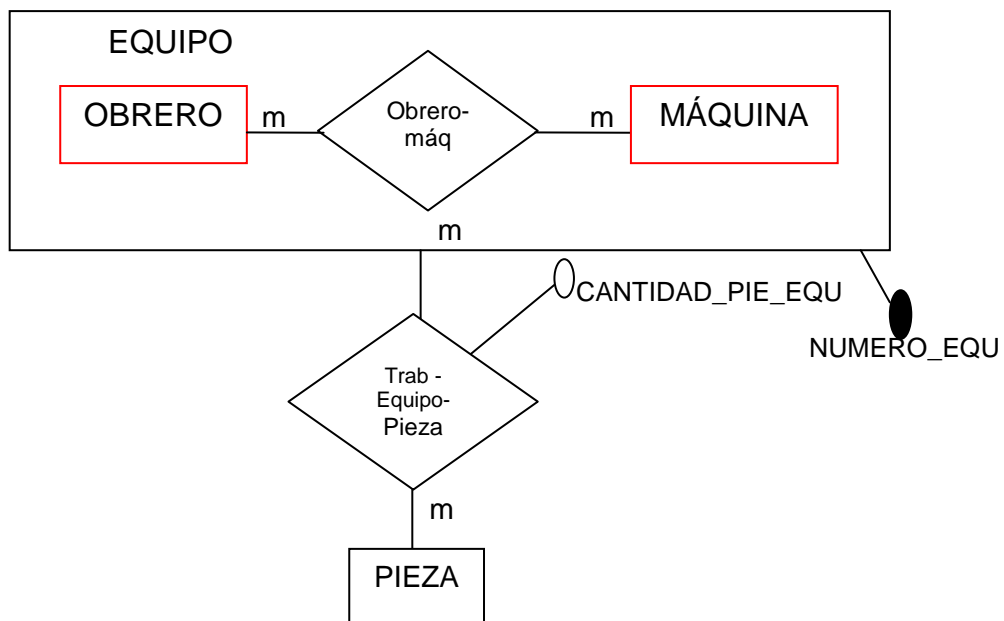
T (total): ya que todo trabajador en el ejemplo es administrativo o dirigente u obrero y **D** (disjunto): pues un trabajador pertenece sólo a una de las especializaciones.

Otro ejemplo de Generalización/Especialización podría ser el caso de Estudiante, Practicante y Becado. Un Practicante es un caso especial de Estudiante. Lo mismo ocurre con Becado. Pero un Practicante también puede ser Becado. Hay muchos Estudiantes que no son Practicantes ni Becados. Obviando los atributos en el DER, esta situación se representaría del modo siguiente:



2.2 Agregación

Obsérvese en el ejemplo que representa la situación de la producción en las empresas, que la relación Trab-Máq-Pieza representa la idea de que una actividad en la empresa se describe en términos de "un obrero en alguna máquina produce una pieza dada en alguna cantidad específica". Sin embargo, la misma situación puede ser vista de forma algo diferente. En la empresa, las máquinas pueden estar asignadas a los obreros y estos "equipos", producir piezas en cierta cantidad. En el modelo entidad - relación original esta situación no hubiera podido ser modelada correctamente, ya que una relación no puede relacionarse con otra relación o entidad. Con la operación de Agregación esta situación se resuelve fácilmente, tal y como se muestra en la figura siguiente:



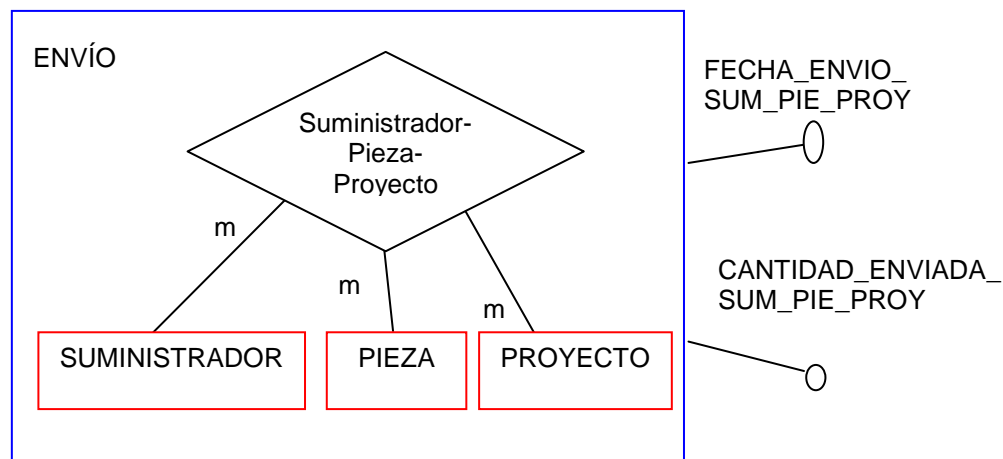
La agregación se define de la siguiente forma:

Si T_1, T_2, \dots, T_n son entidades, la operación define una nueva entidad T con el significado siguiente:

$$T = \{t \mid \exists t_1, t_2, \dots, t_n (t_1 \in T_1 \wedge t_2 \in T_2 \dots \wedge t_n \in T_n \wedge (t_1, t_2, \dots, t_n) = t)\}$$

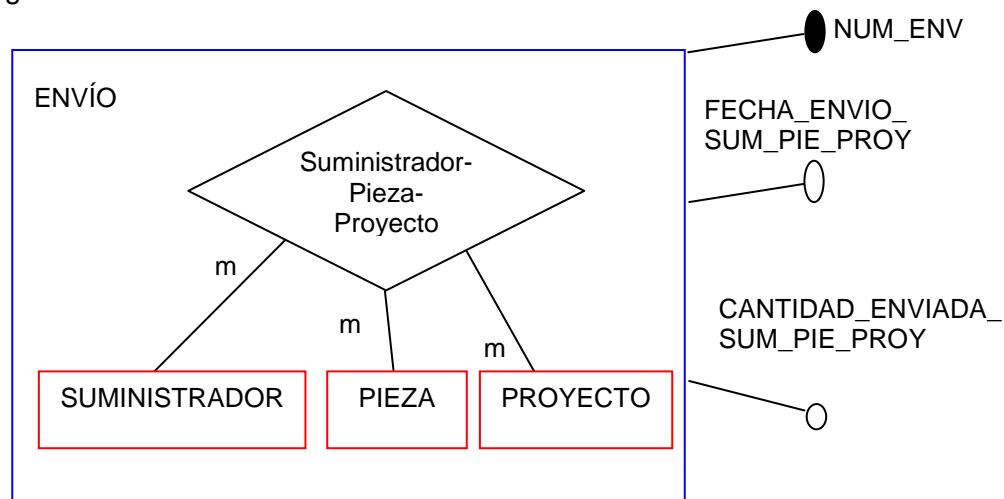
O sea, las nuevas ocurrencias se forman como tuplas de ocurrencias de las entidades componentes. Para que la operación tenga sentido, las entidades T_1, T_2, \dots, T_n tienen que formar parte en alguna relación común y esa relación siempre será incluida en la representación de la entidad generada (entidad agregada).

A la nueva entidad se le pueden asignar atributos. También puede tomar parte en cualquier relación. Otro ejemplo de Agregación se muestra a continuación:



La nueva entidad ENVÍO se define como una agregación de tres entidades: Suministrador, Pieza y Proyecto con los nuevos atributos Fecha del Envío y Cantidad Enviada. Hay una diferencia importante entre estos dos atributos: está claro que la Fecha del Envío no puede pertenecer a ninguna de las entidades componentes, sin embargo, la Cantidad Enviada se refiere claramente a las piezas. Diremos entonces, que la Cantidad Enviada es una "caracterización" de la entidad PIEZA con respecto al ENVÍO.

La llave de una entidad agregada es la llave de la relación que la origina, excepto en el caso en que se defina un identificador (llave) para ella. En el ejemplo anterior del ENVÍO, la llave de la entidad agregada ENVÍO está formada por la llave del suministrador, más la llave de la pieza, más la llave del proyecto. Pero la situación del ejemplo podría tener la variante que se muestra en la siguiente figura, donde se ha definido un número de envío (**NUM_ENV**) que identifica al envío y que, por lo tanto, pasa a ser la llave de la entidad agregada:

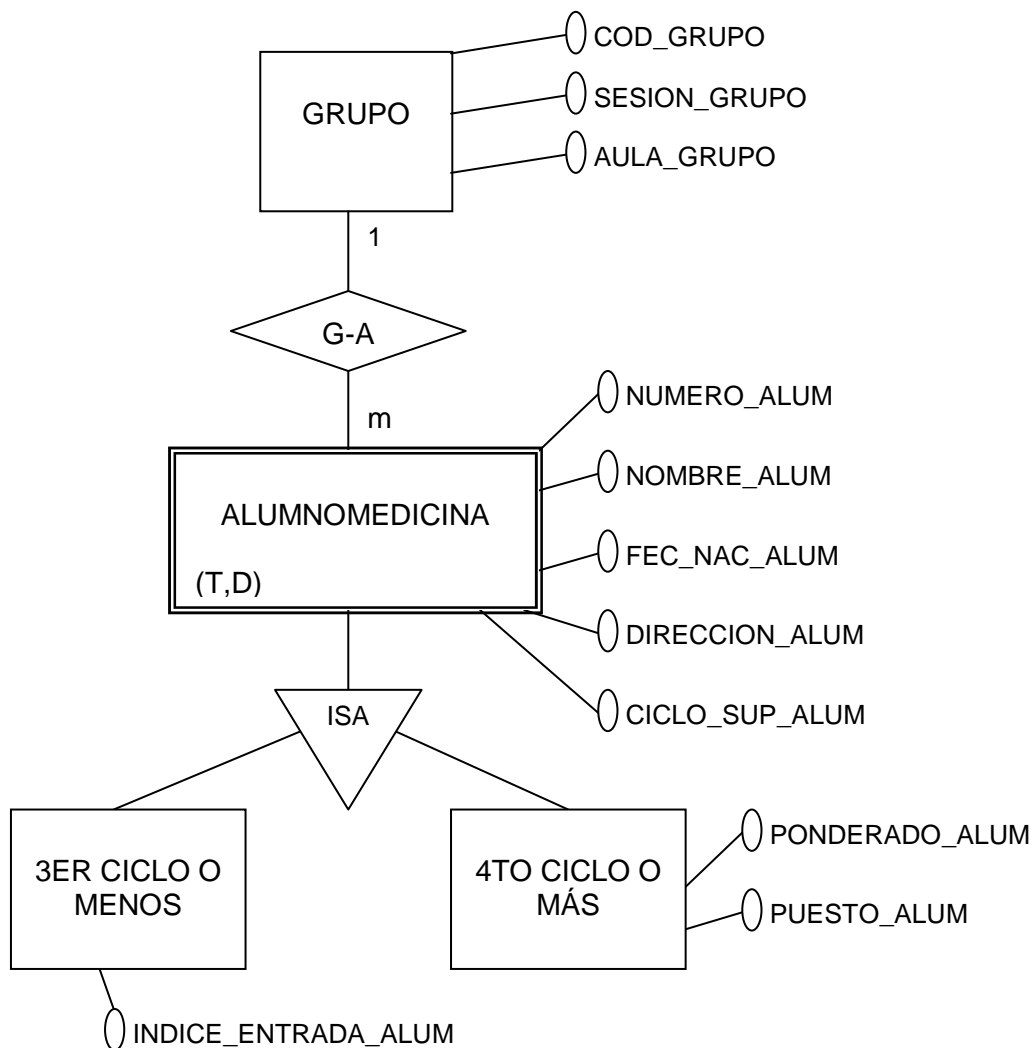


Para el modelo entidad - relación, incluyendo las dos operaciones estudiadas, pueden plantearse una serie de restricciones de integridad:

- Al aplicar la generalización/especialización, una entidad puede pertenecer a una jerarquía de diferentes entidades. Por ejemplo, las entidades PERSONA, TRABAJADOR, OBRERO forman una jerarquía de entidades, sucesivamente más especializadas. Entonces, una entidad existente en un nivel dado, tiene que existir en todos los niveles superiores. De forma inversa, si una entidad se elimina de un conjunto en un nivel dado, debe ser eliminada también en los niveles más bajos.
- La agregación constituye una entidad agregada sobre la base de una relación, por lo que dicha entidad se comportará de forma similar a como se comporta la relación. Entonces, para que una ocurrencia de la agregación exista, deben existir las ocurrencias de todas las entidades que toman parte en la relación. Lo inverso no tiene que ocurrir necesariamente, ya que, por ejemplo, en el caso visto del ENVÍO, pueden existir suministradores que no abastezcan a ningún proyecto, sino que se registran como tales porque en determinado momento pudieran estar activos. Desde luego, si la política de la organización es tal que un suministrador se considera como tal sólo si realmente suministra piezas a algún proyecto, entonces la existencia de, al menos, una ocurrencia de la entidad agregada ENVÍO para un suministrador es indispensable para la existencia de la ocurrencia de ese suministrador en la entidad SUMINISTRADOR.

Es importante señalar que en un DER pueden combinarse todos los elementos y operaciones explicadas anteriormente para representar adecuadamente un caso de la realidad, de modo que, por ejemplo, en una agregación puede participar una entidad débil o una generalizada; una especialización puede ser, a su vez, la generalización de otras especializaciones; una entidad débil puede ser, a su vez, una generalizada o ser débil de una generalizada, etc.

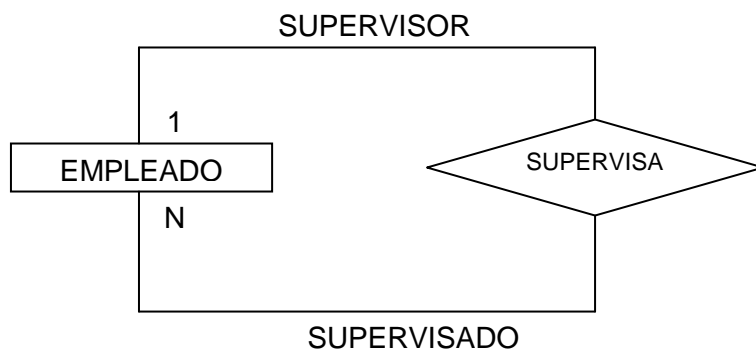
A manera de ejemplo, se muestra a continuación un DER que representa al alumno de Medicina como entidad débil de grupo, ya que el alumno se identifica por un número consecutivo dentro del grupo al que pertenece y que, por lo tanto, puede repetirse en distintos grupos, por lo que, para identificar a un alumno, es preciso decir “el alumno número n del grupo código g”. A la vez, la entidad *alumnomedicina* es una generalización que tiene como especializaciones a los alumnos de 3er. ciclo o menos y a los alumnos de 4to. ciclo o más.



2.3 Relación Recursiva

Relación donde la entidad se relaciona consigo mismo.

Las multiplicidades se colocan respecto a los nombres de rol.



2.3 Ejemplo integrador

Caso: La Liga de Surco

La Liga de Surco requiere controlar la constitución de los diferentes equipos deportivos del distrito y de esta manera programar torneos que les permitan mejorar su calidad deportiva. Para ello, ha decidido crear una base de datos.

La liga cuenta con diferentes clubes de los cuales se tiene su nombre, fecha de creación, dirección y número de locales. Los clubes tienen distintos tipos de jugadores contratados. De los jugadores se conoce su código, el cual se puede repetir para diferentes clubes, los nombres y apellidos, dirección, sexo y fecha de nacimiento, entre otros datos. Cabe mencionar que un jugador es capitán de otros jugadores, lo cual es importante determinar para capacitarlo en cursos de liderazgo y *coaching* deportivo.

Asimismo, la liga tiene empleados de dos tipos: Administrativos y Técnicos. De los empleados se almacena un código, los nombres y apellidos, dirección, sexo, fecha de nacimiento y teléfono fijo y celular. Es importante mencionar que para los empleados de tipo Administrativos se almacena su nivel (Pregrado o Postgrado) y en el caso de los Técnicos la especialidad deportiva (fútbol, voleibol, natación, etc.)

La liga asigna un Técnico un grupo de jugadores y estos pueden tener diferentes Técnicos durante la etapa de jugadores, lo cual constituye un Equipo; de este se almacena la categoría (de acuerdo a la fecha de nacimiento del jugador, como Sub-15, etc.) y la disciplina.

Los empleados Administrativos elaboran varios Contratos de los cuales se guarda el número, la fecha de inicio y fin, entre otros datos. Los Contratos son confeccionados para los Técnicos.

Finalmente, la liga programa a los equipos en diferentes torneos para que eleven su nivel deportivo controlando la cantidad de participaciones que tiene un determinado equipo. Del torneo se registra el nombre del torneo, las fechas de inicio y fin, así como la disciplina correspondiente.

1. En base al caso planteado se pide que construya el Diagrama Entidad Relación (DER).

Solución:

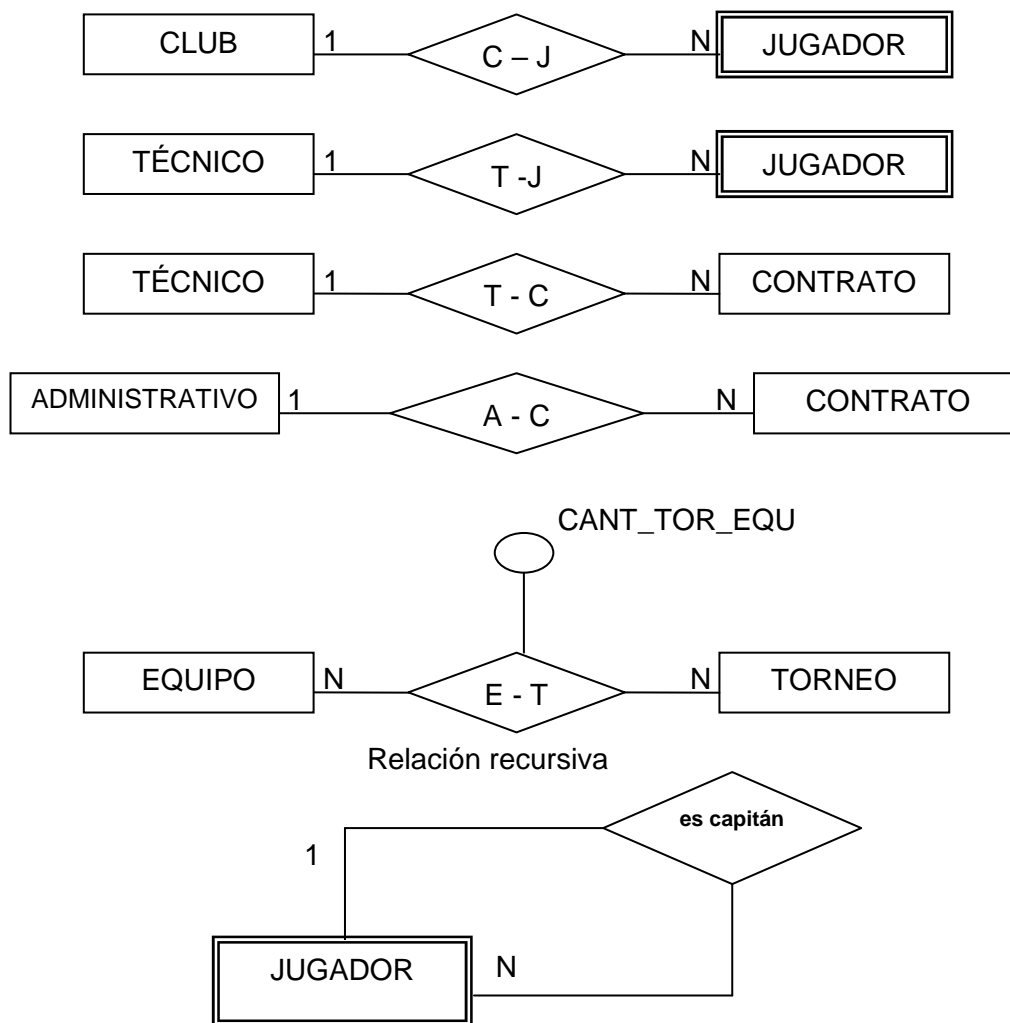
- a. Se determinan las entidades fuertes

- Club
- Empleado
- Técnico
- Administrativo
- Contrato
- Equipo
- Torneo

- b. Se determina la entidad débil

- Jugador

c. Se identifican las relaciones



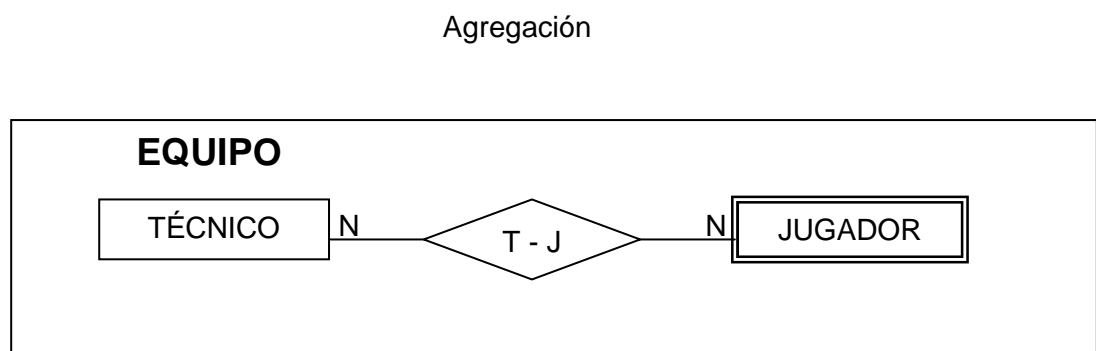
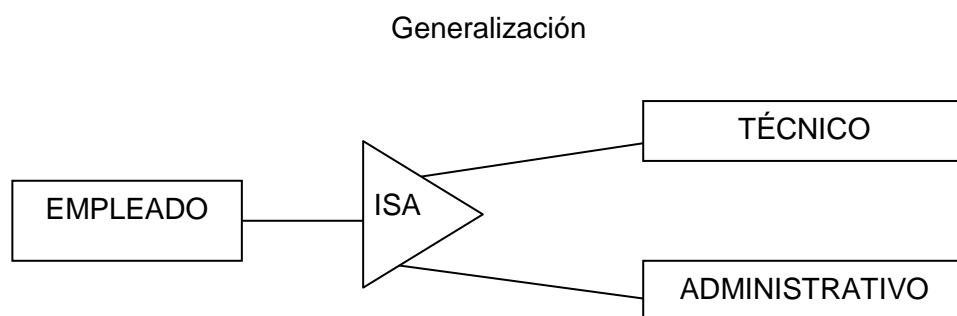
d. Se identifican los atributos y se asocian a las entidades y relaciones

ENTIDAD	ATRIBUTO
CLUB	COD_CLUB, NOM_CLUB, FEC_CRE_CLUB, DIR_CLUB, NRO_LOCAL_CLUB
JUGADOR	COD_JUG, NOM_JUG, APE_PAT_JUG, APE_MAT_JUG, DIR_JUG, SEX_JUG, FEC_NAC_JUG
EMPLEADO	COD_EMP, NOM_EMP, APE_PAT_EMP, APE_MAT_EMP, DIR_EMP, SEX_EMP, FEC_NAC_EMP, TEL_FIJO_EMP, TEL_CEL_EMP
TÉCNICO	ESP_TEC_EMP
ADMINISTRATIVO	NIV_ADM_EMP
EQUIPO	CAT_EQU, DISC_DEP_EQU
CONTRATO	NUM_CON, FEC_INI_CON, FEC_TER_CON
TORNEO	COD_TOR, NOM_TOR, FEC_INI_TOR, FEC_TER_TOR, DISC_DEP_TOR

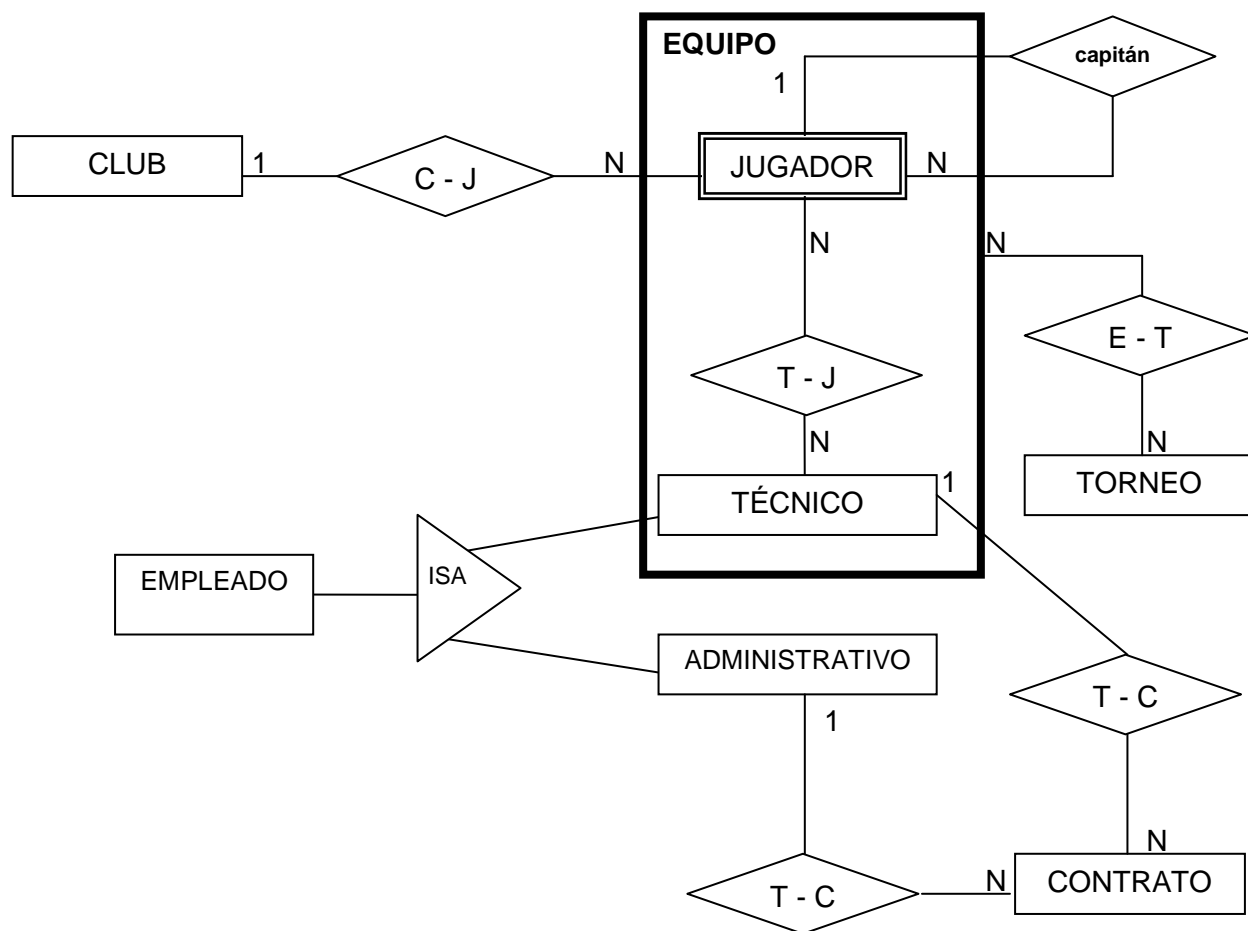
- e. Se determinan los identificadores o claves (simples o compuestas) de cada entidad

ENTIDAD	ATRIBUTO IDENTIFICADOR		
CLUB	COD_CLUB		
JUGADOR	COD_CLUB	COD_JUG	
EMPLEADO	COD_EMP		
EQUIPO	COD_CLUB	COD_JUG	COD_EMP
CONTRATO	NUM_CON		
TORNEO	COD_TOR		

- f. Se determinan, si las hubiese, las jerarquías de generalización y agregaciones.



g. Se dibuja el Diagrama Entidad-Relación (DER)

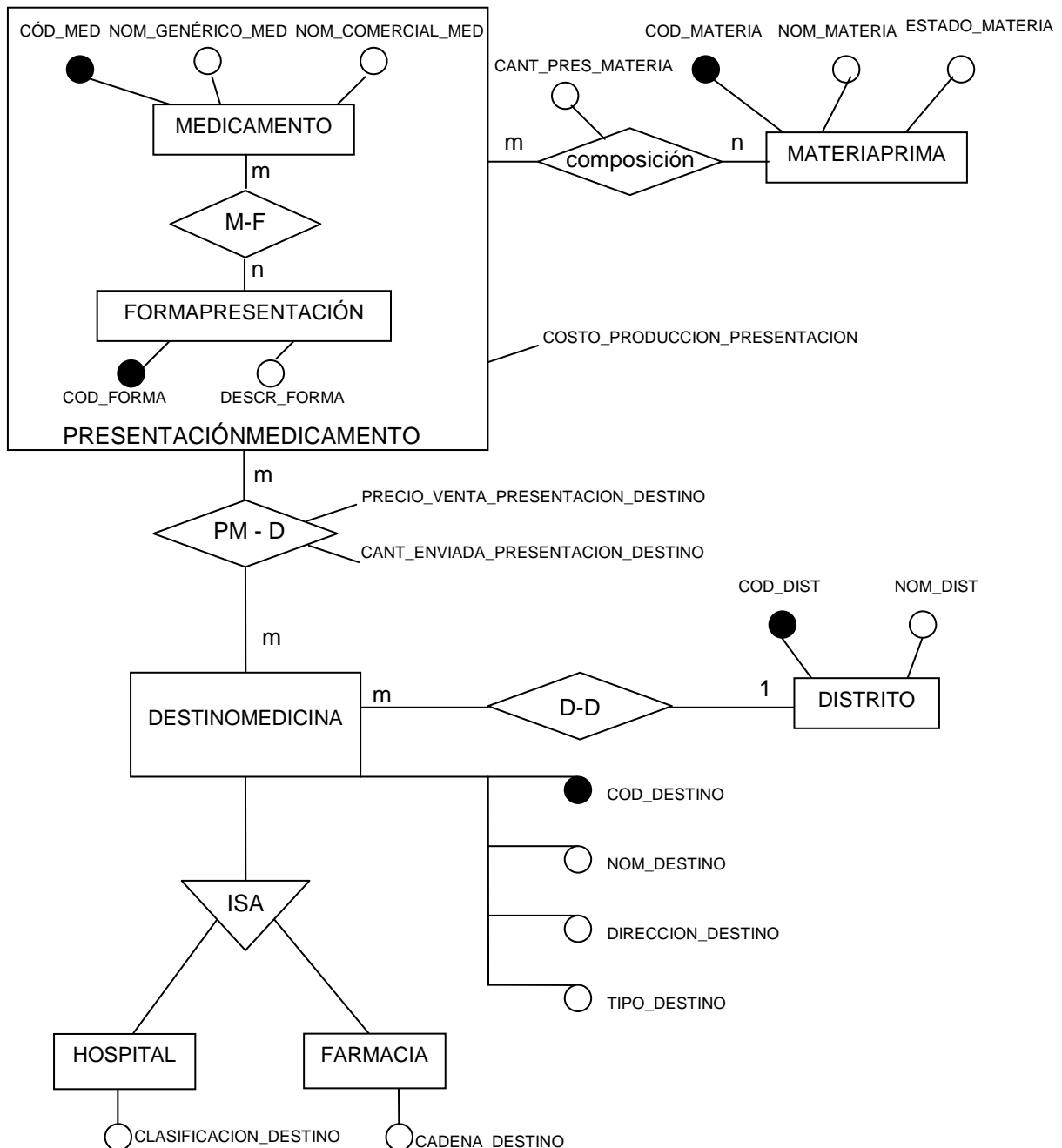


NOTA: en el DER se deberá agregar los atributos e identificadores de cada entidad, así como en la relación de EQUIPO – TORNEO (E - T) el atributo cantidad de torneos en que participa un equipo (CANT_TOR_EQU), como se vio anteriormente.

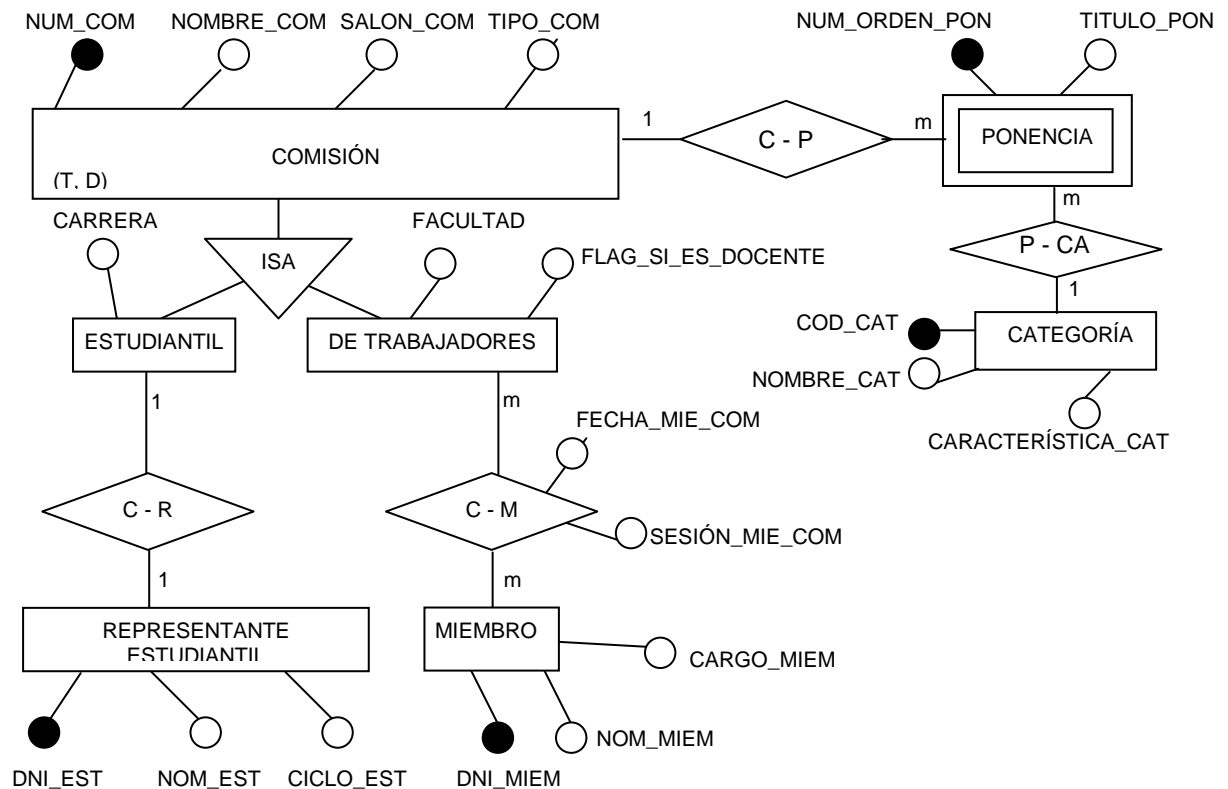
3. AUTOEVALUACIÓN

a. Describa con palabras los casos que representan los siguientes Diagramas Entidad-Relación. Especifique en su descripción los atributos y las llaves de las entidades y las relaciones, así como los tipos de cada relación.

a.1 El caso que se representa en el siguiente DER se refiere a la producción de medicamentos en distintas formas de presentación (cápsulas, ampollas, etc.) y su distribución a los distintos destinos que tienen estos medicamentos.



a.2 El siguiente DER representa el caso de la organización de un evento científico en una universidad, en el que participan estudiantes y trabajadores (que pueden ser docentes y no docentes).



b. Confeccione el DER que represente los siguientes casos:

b.1 En un organismo se reciben productos que son importados de diferentes países, de modo que un producto puede importarse de varios países y de un país se pueden importar distintos productos. Un producto se importa de un país en cierta cantidad. Es necesario controlar las cantidades que se importan de cada país y el valor de las importaciones. Las propiedades de los productos son: número, que lo identifica, nombre, unidad de medida, peso y precio unitario. Las propiedades de los países son: número, que lo identifica, nombre, zona geográfica y tipo de moneda.

b.2 La Biblioteca Nacional del Perú desea efectuar el control de préstamos de los libros y cubículos a sus diferentes usuarios, para lo cual lo ha contratado a fin de que diseñe y cree una base de datos.

Los libros y cubículos son considerados "Recursos" de la biblioteca de los cuales se tiene un código que los identifica. Los libros son prestados a través de una forma denominada Solicitud de Préstamo, de tal manera que un tipo de libro puede ser pedido en varias solicitudes de préstamo, dado que se controla la cantidad de libros existentes (*stock*). De los libros se almacena su nombre, edición, fecha de edición, cantidad, autor y un breve resumen del libro y de los cubículos se registra su capacidad y cantidad de

equipos con que cuenta. Para llevar el control de los libros prestados se registra la fecha de inicio y fin del préstamo, así como el correspondiente control de la devolución.

La biblioteca cuenta con dos (2) tipos de empleados: contratados y practicantes. De los empleados se guarda el código, nombre, apellidos, sexo, dirección, fecha de nacimiento, documento de identidad y el tipo del trabajador. Una solicitud de préstamo es atendida únicamente por un empleado del tipo contratado, dado que debe dar su autorización mediante la firma de dicho documento. De la solicitud de préstamo se almacena el número de la misma, la fecha de solicitud y datos de los usuarios, libros y empleados.

Sin embargo, los Pedidos de los Cubículos se efectúan a través de la Internet generándose un número único para su identificación, siendo tramitado exclusivamente por empleados del tipo practicantes (por no requerir una firma física) para aquellos usuarios que lo requieran. De los Pedidos de Cubículos se registra la fecha del préstamo, el turno solicitado y su correspondiente aprobación. Es importante mencionar que de los empleados Contratados se almacena la fecha de inicio del contrato y de los Practicantes las fechas de inicio y fin de las prácticas.








Asimismo, los usuarios pueden ser de dos tipos: alumnos y profesores, los cuales son de diferentes institutos o colegios, y pueden generar varias Solicitudes de Préstamo para diferentes libros o Pedidos de Cubículos. De los usuarios se almacena su código, nombre, apellidos, dirección, teléfono de casa y documento de identidad.

Finalmente, la biblioteca aplica sanciones basadas en el tiempo que excedió la entrega de uno o varios libros. Las sanciones son de tres tipos:

- Definitiva, por haber perdido uno o más libros.
- Parcial, cuya duración es de un mes de suspensión, por haber excedido la fecha máxima del préstamo en una semana.
- Inicial, cuya duración es de una semana, por haber excedido la fecha máxima del préstamo.

De las sanciones se guarda el tipo de la sanción, fecha inicio, fecha término.

Resumen

-  El Modelo Entidad-Relación permite representar lógicamente un caso.
-  En el Diagrama Entidad-Relación, una entidad se representa mediante un rectángulo, una relación mediante un rombo, un atributo mediante un círculo. Todos estos elementos deben aparecer debidamente identificados por medio de un nombre. Los atributos que constituyen la llave de una entidad, deben tener el círculo relleno.
-  Una entidad cuyos atributos no sean suficientes para identificarla se denomina débil y su llave está formada por algún o algunos de sus atributos más la llave de la entidad que le da origen. Se representa con un doble rectángulo y con la relación entre ella y la entidad que le da origen. Esta relación es de, a lo sumo, muchos (por el extremo de la débil) a uno (por el extremo de la entidad que la origina).
-  La llave de una relación de m:n está formada por la llave de las entidades que participan en la relación. La llave de una relación de 1:n está formada por la llave de la entidad del extremo muchos. La llave de una relación de 1:1 está formada por la llave de cualquiera de las entidades que participan.
-  En una Generalización/Especialización, la entidad Generalizada describe las características generales o comunes que son aplicables a todas las especializaciones. Las especializaciones, como casos especiales de la generalización, sólo contemplan sus propiedades particulares. La llave de cada Especialización es la misma de la Generalización.
-  Una Agregación es el resultado de considerar una relación como una entidad. Los atributos de la relación pasan a ser atributos de la entidad agregada. La llave de la entidad agregada es la llave de la relación que la originó, excepto en el caso en que se defina especialmente un identificador para la agregación, pasando entonces a ser la llave el identificador.
-  Todos estos elementos pueden combinarse en un DER para representar adecuadamente un . Por ejemplo: una entidad débil puede ser, a su vez, una generalización que tenga sus especializaciones; una generalización puede tener especializaciones que, a su vez, puedan ser generalizaciones de otras especializaciones; entre las entidades que participan en una agregación puede haber una entidad débil, etc.

📖 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

🔗 http://basesdatos.uc3m.es/fileadmin/Docencia/DBD/Curso0607/Teoria/MODELO_ER.pdf

En esta página web hallará conceptos complementarios sobre el modelo conceptual, así como ejemplos didácticos asociados al Diagrama Entidad Relación.

🔗 <http://sistemas.itlp.edu.mx/tutoriales/basedat1/>

En esta página web complementará los conceptos sobre el Diagrama Entidad Relación.

🔗 <http://www.dcc.uchile.cl/~cguierr/cursos/BD/g1.pdf>

Es esta página web encontrará una guía de ejercicios resueltos del Diagrama Entidad Relación.

**UNIDAD DE
APRENDIZAJE****2****SEMANA****4**

DIAGRAMA ENTIDAD RELACIÓN PARTE 1

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos diseñan el diagrama entidad relación (DER) de un proceso de negocio a partir de casos planteados por el profesor, relacionando las características del modelo conceptual con el diseño de una base de datos.

TEMARIO

Ejercicios propuestos

ACTIVIDADES PROPUESTAS

1. Se propone la discusión de la solución del ejercicio 2 de la Autoevaluación de la semana 3.
2. Se realizan algunos ejercicios, a manera de ejemplo, por parte del facilitador, en colaboración con los alumnos.
3. Se propone la solución de ejercicios por parte de los alumnos con la orientación del facilitador.

1. DIAGRAMA ENTIDAD-RELACIÓN. EJERCITACIÓN

Diseñe el diagrama Entidad-Relación que represente cada una de los siguientes casos:

Ejercicio #1

Represente, mediante un DER, las relaciones de maternidad, paternidad y matrimonio existente entre hombres y mujeres (emplee sólo las entidades HOMBRE y MUJER para establecer dichas relaciones).

Ejercicio #2

Se desea diseñar una base de datos sobre el movimiento mercantil de un organismo en un año. En el organismo existen mercancías de las que se conoce su código, nombre y unidad de medida. Las mercancías proceden de diferentes países de los que se sabe nombre y tipo de moneda. Para la transportación de las mercancías existen diversas formas, cada una de las cuales se caracteriza por su tipo (barco, avión, tren, etc.) y tarifa. Para cada mercancía de diferentes países existen diferentes formas de transportación; para cada país existen diferentes mercancías que son transportadas en diferentes formas de transportación; y una forma de transportación puede serlo de diferentes mercancías de diferentes países. Una mercancía procedente de un país transportada de una forma dada constituye un embarque y para éste se conoce su fecha de arribo y cantidad.

Un embarque se distribuye entre diferentes almacenes y en un almacén se tienen diferentes embarques, cada uno en cierta cantidad. De cada almacén se tiene su código y dirección. Un almacén distribuye los productos entre diferentes empresas y cada empresa recibe productos de diferentes almacenes. Una empresa se caracteriza por su número, nombre y rama económica; a su vez, las empresas establecen relaciones contractuales entre sí. Entre dos empresas dadas sólo se puede establecer un contrato anual. De cada contrato se conoce su número, valor y fecha de vencimiento.

Ejercicio #3

En una empresa de taxis de Lima existe un conjunto de taxis que brindan servicio a los hospitales de dicha ciudad. En la empresa trabajan choferes, cada uno de los cuales se caracteriza por su DNI, nombre y años de servicio. En la empresa hay numerosos taxis, de los que se conoce la placa, el número del motor, la marca y el modelo. Un taxi puede ser conducido por diferentes choferes (en distintos momentos), pero un chofer siempre conduce el mismo taxi. Se conoce la cantidad de kilómetros totales recorridos por un chofer en su correspondiente taxi. La empresa brinda servicios a varios hospitales. De cada hospital se conoce su nombre, tipo y dirección. Un chofer le brinda servicios (realiza viajes) a distintos hospitales y a un hospital le brindan servicios distintos choferes. Se sabe la cantidad de viajes realizados por cada chofer a cada hospital.

Ejercicio #4

En una empresa de la industria mecánica se fabrican distintos tipos de piezas. Para cada una se conoce su código, que la identifica, descripción y peso unitario. Sobre cada pieza se realizan distintas operaciones (corte, fresado, etc.) acerca de las cuales se conoce su código y nombre. En su proceso de fabricación, una pieza pasa por diferentes equipos que pueden realizar diferentes operaciones, una operación puede ser realizada en diferentes equipos a diferentes piezas y en un equipo se realizan diferentes operaciones sobre diferentes piezas. De cada equipo se conoce su nombre, modelo y país de procedencia. Una operación realizada sobre una pieza en un equipo dado constituye una norma que se identifica por un código y tiene asociados una descripción y un tiempo de ejecución.

En la empresa trabajan obreros de dos tipos distintos: directos y auxiliares. Un obrero directo está vinculado con el cumplimiento de diferentes normas y una norma puede ser aplicada a distintos obreros directos. Un obrero directo cumple una norma dada en un determinado por ciento. Un obrero auxiliar realiza el mantenimiento de diferentes equipos y un equipo dado siempre es atendido por un mismo obrero auxiliar. Para cada equipo atendido por un obrero auxiliar dado se conoce el tipo de mantenimiento efectuado y la última fecha en que se realizó. Todos los obreros de la fábrica están caracterizados por un número, su nombre, calificación laboral y categoría ocupacional (directo o auxiliar).

Ejercicio #5

El Instituto de Aeronáutica del Perú cuenta con información sobre las diferentes compañías de aviación que existen en el mundo. De cada compañía se conoce su nombre (Cubana, Copa, AeroContinente, etc.), su volumen anual de ventas y un código que la identifica. Cada compañía puede estar representada en diferentes países y en un país pueden estar representadas diversas compañías. De cada país se conoce su código, nombre, idioma y área de moneda.

También se sabe que por vía aérea se realizan diferentes viajes. Cada viaje posee un código, un lugar de origen, un destino y una cantidad de kilómetros a recorrer. Además, se conoce que existen distintos tipos de aviones. Cada tipo de avión se identifica por un nombre (IL-62, DC-10, etc.) y posee un consumo de gasolina por kilómetro y una cierta cantidad de asientos. Un mismo viaje puede ser realizado por distintos tipos de aviones y un tipo de avión puede ser utilizado en diversos viajes. Un viaje realizado por un tipo de avión constituye un vuelo y para cada vuelo se conoce su tiempo de duración.

Una compañía de aviación realiza muchos vuelos y un mismo vuelo puede ser cubierto por diversas compañías. Para cada vuelo que ofrece una compañía se conoce el precio del pasaje.

Ejercicio #6

Se desea controlar la actividad de reparación de radios en los talleres dedicados a esta tarea. Cada usuario puede llevar diferentes radios para ser reparados, aunque un radio sólo pertenece a un usuario. Para cada radio se conoce el número de la solicitud de reparación, que lo identifica, tipo de radio, descripción y tipo de rotura (**sencillas** o **complejas**). De cada usuario se conoce su DNI, nombre y dirección. Cada técnico de reparaciones pertenece a un taller y tiene un código que lo identifica dentro de su taller, por lo que cada código puede repetirse para diferentes talleres, y tiene, además, un nombre, una categoría (**A**, **B**, **C**) y un salario. En un

taller laboran muchos técnicos de reparaciones. De cada taller se conoce su código, que lo identifica, nombre y dirección. Los radios con roturas **complejas** sólo podrán ser atendidos por técnicos de categoría **A**, mientras que los radios con roturas **sencillas** pueden ser atendidos por cualquier técnico. En cualquier caso, un técnico puede reparar muchos radios, pero un radio es reparado por un solo técnico.

Ejercicio #7

En un taller de mantenimiento de equipos de computación se desea controlar los servicios brindados. En el taller laboran varios trabajadores. De cada uno se conoce su DNI, su nombre, categoría laboral y tipo de mantenimiento (**Tipo1**, **Tipo2**) que se encarga de realizar a los equipos. El mantenimiento de los equipos de computación se ejecuta por órdenes de servicio que son presentadas por las empresas usuarias de este taller. Una empresa usuaria puede presentar muchas órdenes de servicio y cada orden de servicio es presentada por una empresa usuaria. De cada orden de servicio se conoce su número, que la identifica dentro de la empresa que la presenta, por lo que puede repetirse ese número para diferentes empresas, y la fecha en que se presenta. De cada empresa usuaria se sabe el código que la identifica, su nombre y la rama económica a la que pertenece.

Una orden de servicio puede contener la solicitud de mantenimiento de varios equipos y puede solicitarse el mantenimiento de un equipo en diferentes órdenes de servicio. De cada equipo se sabe el código que lo identifica, su descripción, su marca y su tipo (sólo se toman en cuenta dos tipos: **Computadoras** y **Otros**). El mantenimiento de las **Computadoras** sólo puede ser realizado por los trabajadores que realizan mantenimientos **Tipo1** y el mantenimiento de los **Otros** equipos sólo puede realizarse por trabajadores encargados de hacer los mantenimientos **Tipo2**. En ambos casos, el mantenimiento de un equipo es realizado por un solo trabajador, aunque un trabajador puede realizar el mantenimiento de varios equipos.

Ejercicio #8

Se desea controlar la programación de los itinerarios a recorrer por las locomotoras en una estación de ferrocarriles. En la estación existen varias locomotoras, de las que se conoce el código que la identifica y los años que lleva prestando servicios.

Existen diversas marcas de locomotora; una locomotora es de una marca y una marca puede serlo de varias locomotoras. De cada marca se tiene el código que la identifica, el consumo de combustible promedio por hora y la velocidad máxima que puede alcanzar. Una locomotora realiza diversos itinerarios, pero un itinerario lo realiza una sola locomotora. De cada itinerario se conoce el número que lo identifica para una locomotora dada, pero que puede repetirse para distintas locomotoras (es decir, la locomotora A puede tener un itinerario 2 y la locomotora B puede tener también un itinerario 2), los kilómetros a recorrer, el destino final y el tipo de itinerario (**Diurno** o **Nocturno**; un itinerario es diurno o es nocturno y todos los itinerarios son de uno de estos dos tipos).

Las locomotoras son conducidas por maquinistas. Un maquinista (cualquiera) puede conducir varias locomotoras y una locomotora puede ser conducida por muchos maquinistas y se conoce la cantidad de kilómetros recorrido por un maquinista en una locomotora. De cada maquinista se conoce su DNI, que lo identifica, su nombre y si tiene más de 10 años de antigüedad como maquinista o

no. Los itinerarios nocturnos sólo los pueden realizar los maquinistas con más de 10 años de antigüedad y los maquinistas con más de 10 años de antigüedad sólo realizan itinerarios nocturnos. Los itinerarios diurnos sólo los realizan los maquinistas con 10 o menos años de antigüedad y los maquinistas con 10 o menos años de antigüedad sólo realizan itinerarios diurnos. En ambos casos ocurre que un maquinista realiza muchos itinerarios y que un itinerario es realizado por muchos maquinistas.

**UNIDAD DE
APRENDIZAJE****2****SEMANA****5**

DIAGRAMA ENTIDAD RELACIÓN PARTE 2

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos diseñan el diagrama entidad relación (DER) de un proceso de negocio a partir de casos planteados por el profesor, relacionando las características del modelo conceptual con el diseño de una base de datos.

TEMARIO

Ejercicios propuestos

ACTIVIDADES PROPUESTAS

1. Solución de casos prácticos asesorados por el facilitador del curso y la participación directa de los alumnos.

CASO DE APLICACIÓN 1: EMPRESA “200 MILLAS”

La empresa “200 Millas” se dedica a la extracción y fabricación de productos derivados de la rica fauna marítima de nuestro litoral. Tiene su sede en el Callao y desde allí realiza las actividades de producción y distribución de sus productos a sus diferentes clientes. Los productos son tanto para el consumo humano (Conservas) como también para la industria, como por ejemplo fertilizantes, insumos químicos. etc., y a los largo de sus 5 años de funcionamiento tiene una gran acogida debido a la calidad de productos que fabrica, así como los precios con los cuales los comercializa.

En el último año, el Ing. Juan Quiroz, gerente general de “200 millas”, ha decidido que para seguir con el ritmo de crecimiento de la empresa y convertirla en un corto plazo en una empresa líder dentro del ámbito donde se desarrolla, se necesita ser cada vez mas competitivos, lo que implica tener un mejor control sobre las materias primas, el proceso productivo, la distribución de los productos, su comercialización, etc. Se ha elaborado para ello una cartera de proyectos, de los cuales algunos están en su fase de análisis y otros en la de desarrollo.

Dentro de estos últimos se encuentra el Sistema de Control del Proceso Productivo, el cual tiene un modulo denominado “Seguimiento de Extracción de Materia Prima”. Este modulo controla, como su nombre lo indica, las diferentes actividades desde la extracción de la materia prima hasta la fabricación de los productos finales. El modulo permitirá efectuar un adecuado seguimiento y registro de toda la información pertinente. El responsable del proyecto es el Sr. analista Rubén Díaz, quien actualmente se encuentra en la etapa del diseño de una Base de Datos la cual permita el adecuado registro de toda la información derivada de las actividades propias del modulo.

En sus constantes visitas y entrevistas con la Ing. Inés Tang, supervisora del area de Producción, Ella ha manifestado lo siguiente:

La empresa posee una flota de aproximadamente 100 embarcaciones de distinta capacidad y tipo. Es importante para nosotros almacenar la información de cada una de las embarcaciones, las cuales están debidamente codificadas. Es necesario registrar también el nombre, fecha de compra, tipo de embarcación y capacidad de almacenamiento.

En las diferentes embarcaciones se encuentran trabajando personal debidamente capacitado en la ubicación y extracción del pescado. Es de nuestra incumbencia mantener un catálogo actualizado de dicho personal, en el cual se registren el código, apellido paterno, apellido materno, nombres, fecha de nacimiento, fecha de ingreso, libreta electoral, cargo, haber básico y fecha de ultima capacitación. Un trabajador solo está asignado a una embarcación y para llevar el manejo de una embarcación se requiere la participación de varios trabajadores.

Las actividades de pesca son de gran importancia dentro del proceso productivo. Nosotros las llamamos faenas de pesca. Las faenas de pesca son ejecutadas por una determinada embarcación, por lo que es lógico pensar que una embarcación ha ejecutado varias faenas de pesca. Cada faena tiene un número que nos permite identificarla y distinguirla de otras. Es necesario mantener información de cada faena almacenando su número, fecha de inicio, fecha de fin, zona del mar donde se ejecuto, así como la embarcación que la ejecutó.

Para un mejor control de las especies marinas que extraemos de nuestro litoral se maneja un catálogo denominado Especies, en el cual se detalla la información mas relevante de las materias primas (pescados) que permiten la elaboración de nuestros productos. En este catálogo se registra información como el código del pescado,

nombre del pescado, tipo de pescado y su cotización en el mercado (por tonelada métrica).

Se sabe que una faena de pesca puede recoger una o mas especies, por ejemplo, la Faena "F093" recogió atún, sardina y anchoveta y, como es lógico deducir, una especie puede ser recogida en varias faenas.

Por ltimo es importante almacenar la información acerca de los diferentes productos que se fabrican en "200 Millas ". Para tal efecto se maneja una relación con la información pertinente de cada producto, como su código, descripción, tipo, precio de venta y unidad de medida. Para la fabricación de un determinado producto se emplean una o mas especies y una especie sirve para la preparación de diferentes productos. Finalmente necesitamos una base de datos que nos permita almacenar la información descrita anteriormente así como satisfacer necesidades puntuales como:

- Qué barco ejecutó una determinada faena
- Qué especies y en que cantidad se recogió en una determinada faena
- Qué especies y en que cantidad se necesitan para la elaboración de un determinado producto
- Qué empleados laboran en un barco, entre otros requerimientos de información.

La elaboración de una adecuada estructura de base de datos garantizará la satisfacción de las necesidades establecidas"

Tras el manifiesto de la Srta. Tang, el Sr. Díaz debe de elaborar la base de datos que permita el soporte de manejo de información para el módulo a desarrollar.

Asuma Ud. el papel del Sr. Díaz y elabore los modelos genéricos y detallados , así como las consideraciones finales necesarias para la implementación de la base de datos en cuestión.

CASO DE APLICACIÓN 2: COMERCIAL SANTA ÚRSULA

La empresa Comercial “Santa Ursula” está ubicada en La Victoria y se dedica a la comercialización de diversos productos para seguridad industrial, útiles de oficina, insumos de cómputo y de limpieza en general.

Se ha encomendado al Ing. Juan Mendoza la elaboración de una base de datos que permita controlar las transacciones más importantes de la empresa, de tal forma que pueda dar soporte de datos a las diferentes aplicaciones que en el futuro se desean implementar.

Se ha decidido iniciar el proyecto con el Módulo de Ventas y Abastecimiento por lo que se requiere de identificar las principales entidades que son de interés en estas áreas. Tras una serie de reuniones con los usuarios y sus colaboradores, el Ing. Mendoza ha llegado a las siguientes conclusiones:

Cada producto posee un código que los distingue además de su descripción, precio actual, unidad de medida, línea (industrial, oficina, cómputo o limpieza) , stock actual, stock mínimo y un campo que permite saber si el producto es o no importado.

Los clientes a los que se atienden son todas empresas de las cuales se registra su código, razón social, dirección, teléfono, RUC, *e_mail* y el nombre del contacto. Los clientes pueden generar varias facturas, pero cada una de estas sale a nombre de un determinado cliente.

Los vendedores son los que generan las ventas y se les registra en cada factura que se genera. Operan en varios distritos, pero en un distrito solo se asigna a un vendedor. Existen también vendedores de mostrador, que no se asignan a ningún distrito y trabajan en la sede principal. Se debe registrar el código del vendedor, así como su nombre, apellido, fecha de ingreso, haber básico, y su tipo (estable o contratado).

Las transacciones de venta que se realizan con estas empresas se registran a través de facturas, en las cuales a su vez pueden contener uno o más productos. A su vez, un producto se comercializa en varias facturas. Las facturas tienen un Número de factura, fecha de emisión, un estado (pendiente, cancelada o anulada) y el porcentaje de IGV de la factura. Se sabe que un mismo cliente puede generar una o más facturas y cada factura es dirigida a un cliente. Cada factura sale a nombre del vendedor que origino la transacción, por lo que es lógico pensar que un vendedor ha originado varias facturas.

También se maneja información sobre proveedores. Están registrados en un catálogo donde figuran su código, razón social, dirección, teléfono, RUC y *e_mail*. Además, como siempre tengo que estar con productos en stock, manejo la posibilidad de que un proveedor me abastezca de uno o más productos y viceversa, los cuales me lo abastecen a diferentes precios.

Las transacciones de compra se registran en órdenes de compra, en las cuales se solicita el abastecimiento de uno o más productos. Por supuesto, cada vez que se necesita de un producto, este se solicita mediante la orden de compra respectiva. Se maneja un número de orden de compra, fecha de emisión y su estado (pendiente, atendida o rechazada). Un proveedor puede atender una o mas ordenes de compra y cada orden se emite a un proveedor.

Como deseo ubicar estratégicamente a mi clientela, vendedores y proveedores, tengo un catálogo de distritos donde almaceno el código del distrito y su nombre. En un distrito ubico a uno o más clientes y proveedores y cada uno de ellos se ubica en un distrito. Un vendedor se asigna a varios distritos y en un distrito opera un solo vendedor.

La Base de Datos a desarrollar debe permitir realizar, entre otras, las siguientes consultas:

- El volumen de Facturas por cliente
- El artículo más solicitado por distrito
- Qué artículos se han comercializado en cada factura, en que cantidad y precio
- Qué alternativas tengo cuando un proveedor no tiene el producto que se necesita
- Cuál es el vendedor que mas facturas a generado
- Cuál es la línea de productos que más demanda tiene

Asuma el papel del Ing. Mendoza y elabore un modelo entidad relación genérico y detallado para el caso antes descrito.

CASO DE APLICACIÓN 3: CASO: DISTRIBUIDORA “MINERVA”

Estamos seguros que hoy es su día de suerte, acaba de ser contratado por la empresa Distribuidora “Minerva”, a fin de diseñar su base de datos. El Gerente General le ha dado la siguiente información:

La empresa cuenta con personal en la área de ventas de los cuales se desea almacenar su código, nombre, apellido paterno, apellido materno, sueldo, fecha de ingreso, estado y tipo de personal. En la empresa se cuenta con tres tipos de personal de ventas: Los vendedores de campo, los supervisores y los vendedores de oficina. Adicionalmente a los datos ya registrados, de los vendedores de campo se desea registrar el monto de su comisión por ventas, de los supervisores las zonas que tienen a su cargo para supervisar, y de los vendedores de oficina el numero anexo telefónico en el cual pueden ser ubicados; se debe de tener en cuenta que los supervisores tienen a su cargo un grupo de vendedores de campo pero un vendedor de campo solo reporta a un supervisor.

Esta empresa tiene un catálogo amplio de productos los cuales los clasifica en industriales y domésticos; de los productos en general se requiere almacenar su código, descripción, precio de compra, precio de venta, lugar de procedencia y tipo; sólo de los productos industriales se requiere saber su rubro.

Los productos que comercializa esta distribuidora son proveídos por diversas empresas proveedoras de las mismas que se requiriere almacenar su código, razón social y dirección. Cabe indicar, que un mismo producto puede ser proveído por diferentes empresas proveedoras. Un requerimiento específico del Gerente General fue saber la situación actual de cada uno de los productos proveídos por cada proveedor.

Eventualmente la empresa desarrolla promociones de ventas, las mismas que son realizadas por un determinado vendedor de campo, de un determinado producto en un determinado distrito; de estas promociones se desea almacenar su fecha de inicio y fecha de fin. De los distritos se desea registrar su código y la descripción de mismo.

En base al caso planteado se pide que:

1. Presente el diagrama entidad – relación (entidades, generalizaciones o agregaciones y relaciones).

2. Justifique el diagrama entidad – relación, explicando cada una de las relaciones, generalizaciones o especializaciones, agregaciones, entidades fuertes y débiles, y recursividades que puedan haber sido consideradas en la pregunta 1.
3. Indique para cada entidad el campo clave, consideradas en la pregunta 1.
4. Indique si las siguientes acciones se pueden realizar y justifique su respuesta:
 - a. El gerente de ventas desea eliminar de la base de datos los vendedores de campo, cuya comisión supera su sueldo pues se ha establecido como política que ningún vendedor de campo gane más de 1,500 nuevos soles.
 - b. ¿Es posible que el gerente de logística actualice información de los proveedores que distribuyen productos industriales?
 - c. A los vendedores de Surco se les ha pedido que ingresen el lugar de procedencia de los productos que ofrecen.

CASO DE APLICACIÓN 3: “CYBER VENTAS”

La empresa Cyber Construcción se dedica a la construcción de diferentes Obras que son parte de Proyectos sociales, en los cuales se emplean Máquinas de propiedad de la empresa o alquiladas a terceros. En cada proyecto, se asignan diferentes tipos de Personal como Jefes de Proyecto y los empleados (obreros).

La empresa cuenta con una serie de Máquinas de las cuales se almacena el código, fecha de compra, tipo y horas trabajadas. Las Máquinas se dividen en Propias y Alquiladas.

Cuando una obra no puede ser atendida con maquinaria de la empresa se contrata los servicios de un tercero, al cual se le denomina Team Externo. Un Team Externo se conforma por Máquinas del tipo Alquiladas que cumplen una Función y son proporcionadas por un Proveedor.

Un Proveedor provee muchas Máquinas Alquiladas que cumplen una determinada Función. Una Función es cumplida por muchas Máquinas Alquiladas que son proporcionadas por un Proveedor. Una Máquina alquilada es provista por un proveedor y cumple una determinada Función.

De los Team Externos que se conforman se almacena un código autogenerado y una fecha en la que se conforman. De los proveedores se almacena su RUC, razón social, dirección y teléfono. Para la Función que cumplen las Máquinas Alquiladas se cuenta con un código de la misma.

La atención de una Obra puede ser de varias Máquinas propias pero estas son asignadas a una determinada Obra por un período de tiempo. Sin embargo, algunas Obras requieren de uno o varios Team Externo dependiendo de la magnitud de la Obra. De las Obras se almacena un número autogenerado, la fecha de inicio y término, y el monto de la misma.

Las Obras que se realizan son parte de Proyectos, de tal forma que un Proyecto puede involucrar varias Obras. Para los Proyectos se almacena un número, fecha de inicio y término y el número de etapas.

A cada Proyecto se le asignan varios trabajadores (que en la empresa se les denomina Personal) de los cuales se almacena su código, DNI, nombres, apellidos, dirección, teléfono domiciliario, fecha de ingreso a la empresa y su cargo.

Es importante precisar, que un empleado (Personal) puede ser el Jefe de otros empleados, al que se le considera Jefe del Proyecto.

En base al caso planteado se pide que:

1. Presente el diagrama entidad – relación (entidades, generalizaciones o agregaciones y relaciones).
2. Justifique el diagrama entidad – relación, explicando cada una de las relaciones, generalizaciones o especializaciones, agregaciones, entidades fuertes y débiles, y recursividades que puedan haber sido consideradas en la pregunta 1.
3. Indique para cada entidad el campo clave, consideradas en la pregunta 1.
4. Indique si las siguientes acciones se pueden realizar y justifique su respuesta:
 - a. ¿Es posible actualizar los datos de un Proveedor sin afectar la composición del Team Externo?
 - b. El Gerente de Logística desea ingresar las características de nuevas Máquinas del Tipo Alquiladas.
 - c. Debido a que la nueva política de la empresa es no conservar información pasada, el Gerente de Logística ha dispuesto que Sistemas elimine todos los Team Externos que no han sido considerados desde el año anterior para una obra.



ÁLGEBRA RELACIONAL

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos elaboran consultas para la obtención de información a partir de ejercicios propuestos por el profesor empleando los operadores lógicos y las operaciones del álgebra relacional.

TEMARIO

Integridad relacional

Operaciones del álgebra relacional

Operaciones tradicionales de la teoría de conjuntos

Operaciones relacionales especiales

ACTIVIDADES PROPUESTAS

1. Explicaciones por parte del facilitador combinadas con ejemplos.
2. Se efectúan preguntas de comprobación dirigidas a los alumnos sobre los temas tratados.

1. INTEGRIDAD RELACIONAL

Todo modelo de datos debe abarcar tres elementos:

- La parte estructural
- La parte de integridad
- La parte manipulativa

¿En qué consiste la parte estructural del modelo relacional?

Esto es algo que ya conocemos de las clases anteriores y se refiere a que la única estructura usada en el Modelo Relacional² es la relación de grado n compuesta de atributos y tuplas, donde los atributos están definidos sobre dominios y cada valor del dominio es atómico³. Además, las relaciones deben satisfacer los requerimientos de la segunda y tercera forma normal (2FN y 3FN)⁴ del proceso de normalización de fuente de datos. Estas dos formas de normalización exigen la dependencia de una clave primaria y la dependencia completa de una clave compuesta, así como la transitividad de las relaciones mediante la clave foránea (FK).

La parte de integridad del modelo relacional está compuesta por dos reglas:

- a. **Regla de integridad de entidades (de llaves primarias):** Los valores de la llave primaria en una relación no pueden estar indefinidos.
- b. **Regla de integridad referencial (de llaves extranjeras):** Donde quiera que aparezca un atributo (o combinación de ellos) **Y** en una relación **R2** cuyos valores se correspondan con la llave primaria **X** en una relación **R1**, entonces cada valor de **Y** en **R2** tiene que ser:
 1. ser igual al valor de la llave primaria **X** en alguna tupla de **R1**

o bien,

2. estar indefinido

o sea, que la regla de integridad referencial establece que si una tupla **t2** referencia a una tupla **t1**, entonces **t1** tiene que existir.

Ejemplos:

En el modelo **Suministrador-Producto**, los atributos **S** y **P** de SP se refieren a **SNUM** y **PNUM** respectivamente. En este caso, **S** tiene que tomar, de acuerdo a la variante a. de la regla de integridad referencial, alguno de los valores que tome **SNUM** y, de forma similar, **P** tiene que tomar alguno de los valores que tome **PNUM**. Esto es perfectamente natural, ya que no tiene sentido tener almacenado en SP un suministro de un suministrador que no exista ni un suministro de un producto que no exista.

² Modelo Relacional, está basado en tablas y relaciones de uno a muchos, muchos a uno o uno a uno. Este modelo es empleado para construir la estructura de una base de datos en un gestor de base de datos como *SQL Server 2005* o *2008*.

³ Un valor atómico es único y que no se puede dividir.

⁴ La segunda y tercera forma normal del proceso de normalización de datos se estudiará en las sesiones de la semana 4.

SUMINISTRADOR

<u>SNUM</u>	SNOM	TIPO	DIST
S1	PÉREZ	30	SAN ISIDRO
S2	RAMOS	10	SURCO
S3	ARENAS	20	SAN ISIDRO
S4	VALLE	20	LINCE
S5	LÓPEZ	15	LINCE

SP

<u>S</u>	<u>P</u>	CANT
S1	P1	3
S1	P2	2
S1	P3	4
S1	P4	2
S1	P5	1
S1	P6	1
S2	P1	3
S2	P2	4
S3	P3	4
S3	P5	2
S4	P2	2
S4	P4	3
S4	P5	4

PRODUCTO

<u>PNUM</u>	PNOM	PRECIO	PESO
P1	CLAVO	0.10	12
P2	TUERCA	0.15	17
P3	MARTILO	3.50	80
P4	TORNILLO	0.20	10
P5	ALICATE	2.00	50
P6	SERRUCHO	4.00	90

Aunque la regla de integridad referencial establece que el atributo Y puede estar indefinido, en el caso de este ejemplo, no sería posible que **S** o **P** estuvieran indefinidos, porque **S** y **P** constituyen la llave primaria de la relación SP y, de acuerdo con la regla de integridad de entidades, todos los atributos que forman una llave primaria tienen que tener valor definido.

Veamos a continuación otro ejemplo:

Supongamos que las siguientes dos tablas describen los datos de los profesores y de los departamentos docentes en los que trabajan los profesores:

profesor (dniprof, nomprof, **cdpto**)

departamento (cdpto, nomdpto)

Profesor		
<u>dniprof</u>	nomprof	cdpto
45682457	Cesar	C001
45123585	Juana	C001
58424741	Rocío	C002
65478542	Carlos	C001

Departamento	
<u>cdpto</u>	Nomdpto
C001	Ciencias
C002	Informática

cdpto en la tabla profesor se refiere al atributo cdpto en la tabla departamento, que constituye la llave primaria de la relación departamento. En todas las tuplas de la tabla profesor, el atributo **cdpto** tiene que tomar alguno de los valores que están definidos para cdpto en la tabla departamento, si es que tiene algún valor. Pero en este caso sí puede ocurrir, además, que **cdpto** en la tabla profesor, esté indefinido, lo cual tiene sentido, ya que es posible que aún no se haya determinado el departamento en que trabajará el profesor correspondiente.

2. OPERACIONES DEL ÁLGEBRA RELACIONAL

La parte manipulativa del modelo de datos relacional establece un conjunto de operaciones que conforman el álgebra relacional.

En realidad, existen dos variantes para la parte manipulativa del modelo relacional: el álgebra relacional y el cálculo relacional, aunque sólo estudiaremos el álgebra. Ambos son equivalentes en el sentido de que todo lo que se puede expresar con el álgebra se puede expresar con el cálculo y viceversa. La diferencia radica en la forma de expresar lo que se desea recuperar. El álgebra es procedural y el cálculo es no procedural. El álgebra indica **cómo** se debe hacer una recuperación, mientras que el cálculo indica sólo **qué** se quiere obtener, sin decir qué operaciones se deben realizar.

Cada operador del álgebra relacional toma una o dos relaciones como entrada y produce una nueva relación como salida. Originalmente se definen ocho operadores:

1. Las operaciones tradicionales de la teoría de conjuntos: **unión, intersección, diferencia y producto cartesiano.**
2. Las operaciones relacionales especiales: **selección, proyección, concatenación y división.**

Para simplificar la discusión de cada una de estas operaciones, asumiremos lo siguiente:

1. El orden de izquierda a derecha de los atributos dentro de una relación es significativo.
2. Un atributo de una relación puede tener siempre un nombre calificado, o sea, si **R** es un nombre de relación y **a** es un nombre de atributo dentro de **R**, entonces éste puede ser referenciado con el nombre calificado **R.a**. Del mismo modo, si **t** es una tupla de la relación **R**, **t.a** se refiere al valor que toma el atributo **a** en la tupla **t**.

3. OPERACIONES TRADICIONALES DE LA TEORÍA DE CONJUNTOS

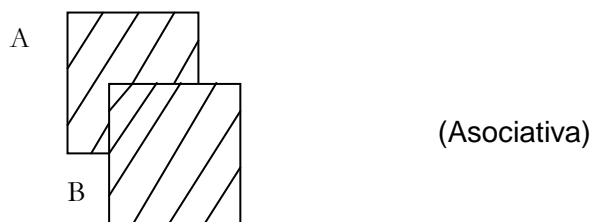
Veamos brevemente las operaciones tradicionales de la teoría de conjuntos.

Ante todo, indicaremos que, para las operaciones de unión, intersección y diferencia debe ocurrir que las relaciones A y B cumplan lo siguiente:

- tienen el mismo grado (sea n)
- y el i -ésimo atributo (donde $1 \leq i \leq n$) de cada una está definido sobre el mismo dominio

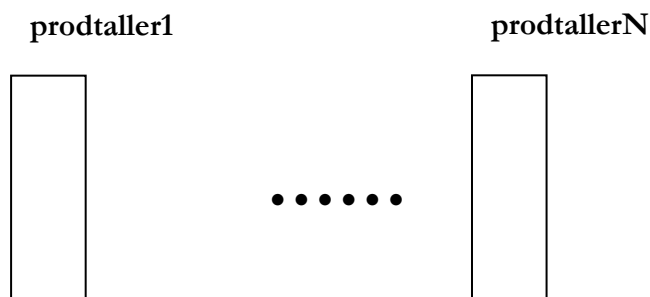
3.1 Unión

A UNIÓN B es el conjunto de las tuplas que pertenecen a A o a B o a ambas



Ejemplo:

Supongamos una empresa que posee diferentes talleres (taller1, taller2, ..., tallerN) en los que se producen ciertos productos. Los productos que se producen en un taller pueden o no producirse también en otro taller. En cada taller se tiene una base de datos que, entre otras tablas, tiene una relación o tabla en la que se almacenan los datos de los productos que se producen en el taller (**prodtaller1**, **prodtaller2**, ..., **prodtallerN**). Todas estas tablas tienen los mismos atributos, en el mismo orden. A continuación se muestra una figura que representa estas relaciones:



Si se desea recuperar en una tabla **todos los productos que se producen en la empresa**, se aplicaría la operación de unión (que puede denotarse **UNIÓN** o \cup) de la siguiente forma:

prodtaller1 \cup \cup **prodtallerN**

Ejemplo práctico

Se cuenta con las siguientes relaciones:

Tabaco 1			
Nombre	Licencia	Hoja	Nic
Camel	R.J. Reynolds	Turca	1.1
Marlboro	Philips Morris	Sin especificar	0.9

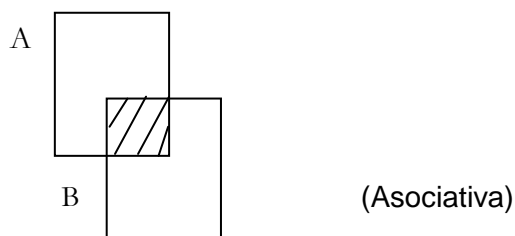
Tabaco 2			
Nombre	Licencia	Hoja	Nic
Ducados	Tabacalera	Sin especificar	1.1
Fortuna	Tabacalera	Sin especificar	1.0
Camel	R.J. Reynolds	Turca	1.1

Operación: Tabaco 1 \cup Tabaco 2

Tabaco 1 \cup Tabaco 2			
Nombre	Licencia	Hoja	Nic
Camel	R.J. Reynolds	Turca	1.1
Marlboro	Philips Morris	Sin especificar	0.9
Ducados	Tabacalera	Sin especificar	1.1
Fortuna	Tabacalera	Sin especificar	1.0

3.2 Intersección

A INTERSECCIÓN B es el conjunto de todas las tuplas que pertenecen a A y a B



Ejemplo:

Para la situación del ejemplo anterior, supongamos que se desea obtener, en una tabla, **los productos que se producen en todos los talleres**. Entonces se aplicaría la operación de intersección (que puede denotarse **INTERSECCIÓN** o \cap) de la siguiente manera:

$$\text{prod taller1} \cap \dots \cap \text{prod tallerN}$$

Ejemplo práctico

Se cuenta con las siguientes relaciones:

Tabaco 1			
Nombre	Licencia	Hoja	Nic
Camel	R.J. Reynolds	Turca	1.1
Marlboro	Philips Morris	Sin especificar	0.9

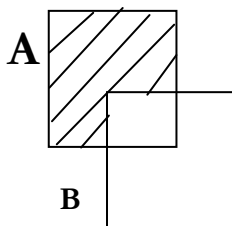
Tabaco 2			
Nombre	Licencia	Hoja	Nic
Ducados	Tabacalera	Sin especificar	1.1
Fortuna	Tabacalera	Sin especificar	1.0
Camel	R.J. Reynolds	Turca	1.1

Operación: Tabaco 1 \cap Tabaco 2

Tabaco 1 \cap Tabaco 2			
Nombre	Licencia	Hoja	Nic
Camel	R.J. Reynolds	Turca	1.1

3.3 Diferencia

A MENOS B es el conjunto de todas las tuplas que pertenecen a A y no pertenecen a B



Ejemplo:

Siguiendo con el mismo caso de los ejemplos anteriores, supongamos que se desea obtener, en una tabla, **los productos que se producen en el taller 1 y que no se producen en el taller 2**. En esta situación se aplicaría la operación diferencia (que puede denotarse **MENOS** o **-**) de la siguiente forma:

prodtaller1 - prodtaller2

Ejemplo práctico

Se cuenta con las siguientes relaciones

Tabaco 1			
Nombre	Licencia	Hoja	Nic
Camel	R.J. Reynolds	Turca	1.1
Marlboro	Philips Morris	Sin especificar	0.9

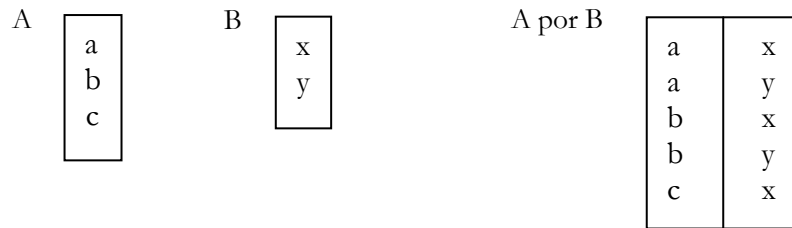
Tabaco 2			
Nombre	Licencia	Hoja	Nic
Ducados	Tabacalera	Sin especificar	1.1
Fortuna	Tabacalera	Sin especificar	1.0
Camel	R.J. Reynolds	Turca	1.1

Operación: Tabaco 1 - Tabaco 2

Tabaco 1 - Tabaco 2			
Nombre	Licencia	Hoja	Nic
Marlboro	Philips Morris	Sin especificar	0.9

3.4 Producto cartesiano

A POR B es el conjunto de tuplas t tales que t es la concatenación de una tupla $a \in A$ y una tupla $b \in B$. En el caso de la operación producto cartesiano, A y B no tienen que ser del mismo grado y sus respectivos atributos no tienen que estar definidos en el mismo dominio.

Ejemplo:

Las operaciones de unión, intersección y producto son asociativas, no así la diferencia, tal y como se indicó anteriormente.

Ejemplo práctico

Se cuenta con las siguientes relaciones

Tabaco 1			
Nombre	Licencia	Hoja	Nic
Camel	R.J. Reynolds	Turca	1.1
Marlboro	Philips Morris	Sin especificar	0.9

Estancos		
Propietario	Calle	Teléfono
La Pajarita	El Nido, 5	2765578
El Clavel	El Jardín,23	4448765

Operación: Tabaco 1 X Estancos

Tabaco 1 X Estancos						
Nombre	Licencia	Hoja	Nic	Propietario	Calle	Teléfono
Camel	R.J. Reynolds	Turca	1.1	La Pajarita	El Nido, 5	2765578
Camel	R.J. Reynolds	Turca	1.1	El Clavel	El Jardín,23	4448765
Marlboro	Philips Morris	Sin especificar	0.9	La Pajarita	El Nido, 5	2765578
Marlboro	Philips Morris	Sin especificar	0.9	El Clavel	El Jardín,23	4448765

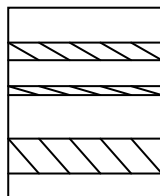
4. OPERACIONES RELACIONALES ESPECIALES**4.1 Selección**

Sea θ cualquier operador de comparación ($<$, \leq , $=$, \neq , \geq , $>$). Entonces:

$$R_{R.X \theta R.Y}$$

es el conjunto de todas las tuplas t de R tales que la comparación $t.X \theta t.Y$ toma valor cierto, donde los atributos X y Y deben estar definidos en el mismo dominio y la operación θ debe tener sentido en ese dominio. En lugar de X o Y se puede especificar una constante, por ejemplo: $R_{R.X \theta \text{ constante}}$

La operación de selección obtiene un subconjunto "horizontal" de una relación dada, o sea, el subconjunto formado por aquellas tuplas de la relación que satisfacen una comparación dada.



Las comparaciones θ pueden ser combinadas con los operadores lógicos

AND (\wedge), **OR** (\vee) y **NOT** (\neg).

Entonces:

- $R_{C1 \text{ AND } C2}$ es equivalente a R_{C1} INTERSECCIÓN R_{C2}
- $R_{C1 \text{ OR } C2}$ es equivalente a R_{C1} UNIÓN R_{C2}
- $R_{\text{NOT } C}$ es equivalente a R MENOS R_C

En la selección, **R** puede ser una relación o una expresión entre paréntesis.

Ejemplo:

SUMINISTRADOR

<u>SNUM</u>	SNOM	TIPO	DIST
S1	PÉREZ	30	SAN ISIDRO
S2	RAMOS	10	SURCO
S3	ARENAS	20	SAN ISIDRO
S4	VALLE	20	LINCE
S5	LÓPEZ	15	LINCE

Supongamos que se desea obtener una tabla con los datos de los suministradores que radican en el distrito SAN ISIDRO. En este caso se plantearía la siguiente selección:

SUMINISTRADOR_{DIST = 'SAN ISIDRO'}

y la tabla resultante sería:

<u>SNUM</u>	SNOM	TIPO	DIST
S1	PÉREZ	30	SAN ISIDRO
S3	ARENAS	20	SAN ISIDRO

Ejemplo práctico

Se cuenta con la siguiente relación:

Tabaco 2			
Nombre	Licencia	Hoja	Nic
Ducados	Tabacalera	Sin especificar	1.1
Fortuna	Tabacalera	Sin especificar	1.0
Camel	R.J. Reynolds	Turca	1.1

Consulta: R Licencia='Tabacalera' and Nic>1 (Tabaco2)

Nombre	Licencia	Hoja	Nic
Ducados	Tabacalera	Sin especificar	1.1

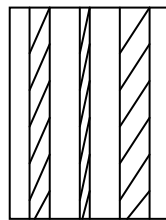
4.2 Proyección

La proyección de una relación R sobre los atributos X, Y, ..., Z:

R [X, Y, ..., Z]

es el conjunto de todas las tuplas (x, y, ..., z) tales que existe una tupla **t** en R que tiene el valor **x** en el atributo **X**, el valor **y** en el atributo **Y**, ... y el valor **z** en el atributo **Z**.

Así, la proyección obtiene un subconjunto "vertical" de una relación dada, o sea, el subconjunto obtenido seleccionando los atributos especificados, en determinado orden, de izquierda a derecha, y eliminando, si es necesario, las tuplas duplicadas que podrían formarse al escogerse sólo los atributos seleccionados.



Ya que consideramos significativo el orden de los atributos dentro de una relación, la proyección brinda la forma de reordenar los atributos de una tabla, si esto se desea.

En la proyección, R puede ser una relación o una expresión entre paréntesis.

Ningún atributo puede ser especificado más de una vez en la lista de atributos de la proyección. Omitir dicha lista es equivalente a especificar todos los atributos en su correspondiente orden de izquierda a derecha, o sea, dicha proyección sería idéntica a la relación dada. Se eliminan las tuplas repetidas.

Ejemplos:

SUMINISTRADOR

<u>SNUM</u>	SNOM	TIPO	DIST
S1	PÉREZ	30	SAN ISIDRO
S2	RAMOS	10	SURCO
S3	ARENAS	20	SAN ISIDRO
S4	VALLE	20	LINCE
S5	LÓPEZ	15	LINCE

Supongamos que se desea obtener una tabla en que sólo aparezcan los atributos DIST y TIPO, en ese orden, que no es el mismo orden en que aparecen esos atributos en la relación SUMINISTRADOR. Entonces se plantearía la siguiente proyección sobre la tabla SUMINISTRADOR:

SUMINISTRADOR [DIST, TIPO]

y la tabla resultante sería:

DIST	TIPO
SAN ISIDRO	30
SURCO	10
SAN ISIDRO	20
LINCE	20
LINCE	15

Veamos otro ejemplo de proyección en el que se destaca el hecho de que se eliminan las filas iguales.

SP

<u>S</u>	<u>P</u>	CANT
S1	P1	3
S1	P2	2
S1	P3	4
S1	P4	2
S1	P5	1
S1	P6	1
S2	P1	3
S2	P2	4
S3	P3	4
S3	P5	2
S4	P2	2
S4	P4	3
S4	P5	4

Supongamos que se desea obtener en una tabla los números de los suministradores que suministran productos (pueden existir suministradores que no estén suministrando productos). Para satisfacer esta demanda, se haría la proyección de la tabla SP sobre el atributo S del siguiente modo:

SP [S]

y la tabla resultante sería:

<u>S</u>
S1
S2
S3
S4

Ejemplo práctico

Se cuenta con la siguiente relación:

Tabaco 2			
Nombre	Licencia	Hoja	Nic
Ducados	Tabacalera	Sin especificar	1.1
Fortuna	Tabacalera	Sin especificar	1.0
Camel	R.J. Reynolds	Turca	1.1

Consulta: R Nombre, Licencia (Tabaco2)

Nombre	Licencia
Ducados	Tabacalera
Fortuna	Tabacalera
Camel	R.J. Reynolds

4.3 Concatenación (join)

Sea θ cualquier operador de comparación (al igual que en la selección).

La **concatenación- θ** de una relación **A** sobre el atributo **X** con una relación **B** sobre el atributo **Y**, es el conjunto de todas las tuplas **t** tales que **t** es la concatenación de una tupla **a** \in **A** y una tupla **b** \in **B** y para las que el predicado **a.x** θ **b.y** toma valor **cierto**.

Debe cumplirse que los atributos **a.x** y **b.y** estén definidos sobre el mismo dominio y que la operación θ tenga sentido en él.

Si θ es la comparación **=**, la operación se denomina "equijoin". Entonces, el resultado de un equijoin tiene que incluir dos atributos idénticos. Si uno de ellos se elimina, entonces el resultado se denomina concatenación natural (join natural) o simplemente concatenación (**JOIN**, que puede denotarse también *)

Así, por ejemplo:

SUMINISTRADOR * SP sobre los atributos **SNUM** en SUMINISTRADOR y **S** en SP, que están definidos sobre el mismo dominio, equivale a:

((SUMINISTRADOR **POR** SP) SUMINISTRADOR.SNUM=SP.S)
[SUMINISTRADOR.SNUM, SUMINISTRADOR.SNOM,
SUMINISTRADOR.TIPO, SUMINISTRADOR.DIST, SP.P, SP.CANT]

Supondremos, para mayor facilidad en la definición de la operación, que la concatenación se realiza según el último atributo de la primera relación y el primero de la segunda, para no tener que especificar los atributos que se comparan.

Ejemplo:

	s	x		x	z		s	x	z
A	s1	x1	B	x1	z1	A JOIN B	s1	x1	z1
	s2	x1		x2	z2		s2	x1	z1
	s3	x2		x3	z3		s3	x2	z2
				x2	z4		s3	x2	z4

Donde **s** y **z** representan, en general, varios atributos y **x**, el atributo que se compara.

De acuerdo a lo planteado anteriormente acerca de que la concatenación se realiza según el último atributo de la primera relación y el primero de la segunda, entonces puede ser necesario plantear las correspondientes proyecciones antes de realizar el JOIN con el objetivo de reordenar los atributos convenientemente. Por ejemplo, en el caso anterior de SUMINISTRADOR * SP debía plantearse:

SUMINISTRADOR [SNOM, TIPO, DIST, SNUM] * SP

La operación de concatenación es asociativa.

Ejemplo práctico

Se cuenta con las siguientes relaciones:

Tabaco 3			
Nombre	Licencia	Hoja	Nic
Camel	R.J. Reynolds	Turca	1.1
Marlboro	Philips Morris	Sin especificar	0.9
Rex	Tabacalera	Canaria	0.9

Descripción		
Hoja	Clase	Color
Turca	Normal	Rubio
Turca	Light	Rubio
Holandesa	Normal	Rubio
Canaria	UltraLight	Negro

Consulta: Tabaco3 JOIN DESCRIPCION

Nombre	Licencia	Hoja	Nic	Clase	Color
Camel	R.J. Reynolds	Turca	1.1	Normal	Rubio
Camel	R.J. Reynolds	Turca	1.1	Light	Rubio
Rex	Tabacalera	Canaria	0.9	UltraLight	Negro

4.4 División

La operación de división divide una relación dividiendo A de grado $m+n$ entre una relación divisor B de grado n, produciendo una relación cociente de grado m. El atributo $m+i$ de A y el atributo i -ésimo de B ($i = 1, 2, \dots, n$) deben estar definidos sobre el mismo dominio.

Si consideramos los primeros m atributos de A como un único atributo compuesto **X** y los últimos n atributos como un único atributo compuesto **Y**, la relación A puede ser vista como un conjunto de pares de valores (**x**, **y**). Similarmente, la relación B puede ser vista como un conjunto de valores únicos (**y**). Entonces el resultado de dividir A por B, o sea, el resultado de evaluar la expresión **A ENTRE B**, es la relación C con un único atributo compuesto **X**, tal que cada valor **x** de C.X aparece como un valor de A.X y el par de valores (**x**, **y**) aparece en A para todos los valores **y** que aparecen en B

A ENTRE B es el conjunto de las tuplas **t** tales que para todas las tuplas **b** \in

B, la tupla **t.b** \in **A**

Ejemplo:

	X	Y			
A	a	x			
	a	y			
	a	z			
	b	x			
	c	y			

	Y				
B	x	z			

	C (A entre B)				
	a				

La división tiene la misma prioridad que la concatenación.

De las operaciones relacionales especiales, la selección es la de mayor prioridad, seguida de la proyección y, por último, la concatenación y la división. Como es usual, es posible usar los paréntesis para combinar dichas operaciones alterando sus prioridades.

Prioridades

+	↑	Selección
	↑	Proyección
-	↓	Concatenación y División

Como se ha visto, hemos definido el álgebra relacional como un conjunto de 8 operaciones, aunque este conjunto no es mínimo en el sentido de que algunas operaciones definidas pueden ser obtenidas a partir de otras. En realidad, de las 8 operaciones, sólo 5 son primitivas (o sea que no pueden definirse a partir de otras). Estas son:

- Unión
- Diferencia
- Producto
- Selección
- Proyección

Como vimos, estas operaciones permiten realizar la recuperación de datos, aunque éste no es su único objetivo pues, en general, el álgebra permite la escritura de expresiones que pueden ser utilizadas con diversos fines y no sólo para la recuperación, por ejemplo, para definir derechos de acceso, restricciones de integridad, etc.

Por último, diremos que las expresiones algebraicas pueden ser manipuladas para encontrar otras equivalentes, pero que sean más eficientes para realizar las recuperaciones, por lo que el álgebra puede ser utilizada como base para el proceso de optimización de las consultas de información a las bases de datos. El proceso de optimización de las consultas lo realiza el SGBD.

A manera de ilustración, veamos el siguiente ejemplo en el que se plantean dos expresiones en las que se recuperan los datos de los suministradores que suministran el producto cuyo número es 'P1', así como los datos del suministro correspondiente de ese producto:

La expresión del álgebra relacional

$(\text{SUMINISTRADOR} [\text{SNOM}, \text{TIPO}, \text{DIST}, \text{SNUM}] * \text{SP})_{P='P1'}$

es equivalente a:

$\text{SUMINISTRADOR} [\text{SNOM}, \text{TIPO}, \text{DIST}, \text{SNUM}] * \text{SP}_{P='P1'}$

Sin embargo, la segunda expresión es más eficiente, pues, en la primera, se hace primero el join de SUMINISTRADOR con TODAS las tuplas de SP y después es que se seleccionan las filas en que el producto es 'P1', lo que hace que se tengan que concatenar muchas tuplas que no son necesarias para la recuperación que se desea hacer. Por el contrario, en la segunda expresión se hace primero la selección de las filas de SP en que se suministra 'P1' y, después, se hace el join de esas filas ya escogidas con SUMINISTRADOR, por lo que se concatenan, en general, menos tuplas, lo que produce un ahorro considerable de tiempo.

NOTA.- La sintaxis a emplear para las consultas en álgebra relacional debe ser:

**R <Nombre de la Relación u operación entre Relaciones a efectuar>
<Atributos a mostrar> <Condición>**

R significa la nueva relación que se va a generar, la misma que en otras fuentes de conocimiento (libros) se podría denotar como SELECT o **TT**

Ejemplo práctico

Se cuenta con las siguientes relaciones:

Tabaco 1 X Estancos						
Nombre	Licencia	Hoja	Nic	Propietario	Calle	Teléfono
Camel	R.J. Reynolds	Turca	1.1	La Pajarita	El Nido, 5	2765578
Camel	R.J. Reynolds	Turca	1.1	El Clavel	El Jardín, 23	4448765
Marlboro	Philips Morris	Sin especificar	0.9	La Pajarita	El Nido, 5	2765578
Marlboro	Philips Morris	Sin especificar	0.9	El Clavel	El Jardín, 23	4448765

Distribución			
Propietario	Calle	Teléfono	Licencia
La Pajarita	El Nido, 5	2765578	R.J. Reynolds
El Clavel	El Jardín 23	4448765	Philips Morris

Consulta: Tabaco1-Estancos DIVISIÓN Distribución

Nombre	Hoja	Nic
Camel	Turca	1.1
Marlboro	Sin especificar	0.9

5. AUTOEVALUACIÓN

- a. Los estudiantes que estudian en cierto centro de Educación Superior realizan trabajos de investigación. Un estudiante participa en un solo trabajo de investigación y en un trabajo de investigación participan muchos estudiantes. Para representar esta situación se tienen las siguientes tablas:

estudiante (cest, nomest, sec, ntrab)

trabajo (ntrab, nomtrab, tipo)

donde:

cest: código que identifica al estudiante

nomest: nombre del estudiante

sec: sección a la que pertenece el estudiante

ntrab: número que identifica el trabajo de investigación

nomtrab: nombre del trabajo de investigación

tipo: tipo del trabajo de investigación (referativo, proyecto, desarrollo, etc.)

Suponga que las ocurrencias que se desea almacenar en ambas tablas son las siguientes:

ESTUDIANTE

cest	Nomest	sec	ntrab
C001	José Pérez	3A	002
C105	María González	3C	010
C045	Marta López	4A	
C203	Juan Martínez	3A	003
C009	Rossana Mesa	4B	001
	Luis Gutiérrez	3D	
C104	Daniel García	5A	003
C307	Lisette Rodríguez	5A	002
C503	Raúl Valdés	3D	020

TRABAJO

ntrab	nomtrab	tipo
001	Las Bases de Datos Distribuidas	REF
002	Sist. para el Control de Planillas	DE S
003	Prototipos de Sist. Informáticos	PR O
004	Sist. para el Control de Boletas	DE S
	Modelamiento Semántico	REF

Analice estos valores y defina si hay algunos que infringen las reglas de integridad del modelo relacional. En caso afirmativo, diga cuál regla se infringe y por qué.

b. Sobre el modelo Suministrador-Producto:

suministrador (snum, snom, tipo, dist)

producto (pnum, pnom, precio, peso)

sp (s, p, cant)

b.1 Describa las recuperaciones que se realizan mediante las siguientes expresiones del álgebra relacional:

1) $sp_{s='S3'}[p]$

2) $(sp_{p='P5'}[s] * suministrador)[tipo]$

3) $(suministrador[snum]_{dist='LINCE'} * sp[s, p] * producto)[pnom]$

b.2 Realizar las siguientes recuperaciones:

1) Obtener los números de los productos suministrados por el suministrador 'S2'

2) Obtener los nombres de los productos suministrados por el suministrador 'S2'

Resumen


- 📖 La integridad del modelo relacional la rigen dos reglas: la de integridad de entidades y la de integridad referencial. La primera exige que los atributos que componen una llave primaria tienen que estar definidos. La segunda, que los atributos que hacen referencia a una llave primaria de otra relación tienen que tomar uno de los valores definidos para esa llave primaria o estar indefinidos.
- 📖 El resultado de aplicar cualquier operación del álgebra relacional es siempre una nueva relación.
- 📖 El álgebra relacional incluye las operaciones Unión, Intersección, Diferencia y Producto Cartesiano que se heredan de la Teoría de Conjuntos.
- 📖 Para el álgebra relacional se definen operaciones especiales: Selección, Proyección, Concatenación (Join) y División.
- 📖 La selección permite escoger filas que cumplan una cierta condición. Una condición está formada por la comparación de dos campos o de un campo y un atributo. Pueden emplearse condiciones más complejas, que se obtienen combinando condiciones simples mediante los operadores lógicos **and**, **or** y **not**.
- 📖 La proyección permite seleccionar columnas en un cierto orden.
- 📖 El join permite combinar informaciones que se encuentran en diferentes tablas. Las informaciones de dos tablas se combinan en un join a partir de la comparación, por el operador $=$, del último atributo de la primera tabla con el primero de la segunda. La relación resultante del join está formada por la concatenación de aquellas tuplas de ambas tablas para las que se cumpla que la comparación de los dos atributos antes mencionados es cierta.
- 📖 La división de **A ENTRE B** es el conjunto de las tuplas **t** tales que para todas las tuplas **b** \in **B**, la tupla **t.b** \in **A**.
- 📖 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

🔗 <http://www.oei.eui.upm.es/Asignaturas/BD/BD/docbd/tema/algebra.pdf>

En esta página web hallará conceptos complementarios sobre álgebra relacional.

🔗 [http://www.dsic.upv.es/~mapastor/BDA/BDA_tema2\(ejer.%20AR\)_2008.pdf](http://www.dsic.upv.es/~mapastor/BDA/BDA_tema2(ejer.%20AR)_2008.pdf)

En esta página web obtendrá ejercicios propuestos.

 http://www.kybele.etsii.urjc.es/docencia/DBD/2008-2009/Material/%5BDBD-2006-07%5DEjercicios_AR_2.pdf

En esta página web obtendrá mas ejercicios propuestos.

**UNIDAD DE
APRENDIZAJE****3****SEMANA****8**

ÁLGEBRA RELACIONAL

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos elaboran consultas para la obtención de información a partir de ejercicios propuestos por el profesor empleando los operadores lógicos y las operaciones del álgebra relacional

TEMARIO

Ejercicios propuestos

ACTIVIDADES PROPUESTAS

1. Se realizan ejercicios, a manera de ejemplo, por parte del facilitador, en colaboración con los alumnos.
2. Los alumnos realizan ejercicios con la orientación del facilitador.

1. ÁLGEBRA RELACIONAL. EJERCITACIÓN

Ejercicio # 1

Realizar, utilizando el álgebra relacional, las siguientes recuperaciones sobre el modelo Suministrador-Producto:

Modelo Suministrador-Producto: **suministrador** (snum, snom, tipo, dist)
 producto (pnum, pnom, precio, peso)
 sp (s, p, cant)

- Obtenga los números de los suministradores que suministran el producto 'P3'.
- Obtenga los nombres de los suministradores que suministran el producto 'P3'.
- Obtenga los números de los suministradores que sirven algún producto cuyo precio sea menor de s/ 10.00.
- Obtenga los nombres de los suministradores que suministran, al menos, un producto de precio s/ 0.10.
- Obtenga el nombre y el distrito en que radica cada suministrador que suministra el producto Martillo o el producto Alicata.
- Para aquellos suministradores que radican en el distrito 'San Isidro', obtenga su nombre, el número de los productos que suministra y la cantidad en que suministra cada uno de esos productos.

Ejercicio # 2

Se tienen las siguientes relaciones sobre el trabajo científico de los estudiantes de un Centro de Educación Superior (CES):

estud (nest, nomes, año, espec, prom)
 (contiene los datos de todos los estudiantes del CES)

nest:	# del estudiante
nomes:	nombre del estudiante
año:	año académico
espec:	especialidad
prom:	promedio

prof (nprof, dpto, nompr, categ)
 (contiene los datos de todos los profesores del CES)

nprof:	# del profesor
dpto:	departamento
nompr:	nombre del profesor.
categ:	categoría docente.

trab-cien (nes, npr)
 (contiene las relaciones estud-prof con respecto a la atención al trabajo científico)

nes: # del estudiante
npr: # del profesor

y se desea resolver, mediante el álgebra relacional, las siguientes recuperaciones:

- a. Para cada estudiante, obtenga su número y el departamento de cada profesor que lo ha atendido.
- b. Obtenga los nombres de los profesores que atienden algún estudiante de la especialidad Informática.
- c. Obtenga los nombres de los estudiantes que no están incorporados al trabajo científico.
- d. Obtenga los números de aquellos estudiantes que han sido atendidos en su trabajo científico por el profesor Juan Pérez.
- e. Obtenga los nombres de los estudiantes de la especialidad Informática que hayan sido atendidos en su trabajo científico por profesores que tengan categoría docente 'PA' y que pertenezcan al departamento de Computación.

Ejercicio # 3

Se cuenta con una base de datos sobre microcomputadoras, profesores y alumnos.

De cada entidad se conoce:

Para las microcomputadoras: # serie, marca, modelo.

Para los profesores: DNI del profesor, nombre, categoría y departamento.

Para los estudiantes: DNI del estudiante, nombre, especialidad y sesión de clases

Un profesor puede trabajar en varias micros y viceversa (en cada una cierta cantidad de horas)

Un estudiante puede trabajar en varias micros y viceversa (en cada una cierta cantidad de horas)

Un profesor atiende a varios estudiantes y un estudiante es atendido por varios profesores, una cierta cantidad de horas semanales.

1. Diseñe el modelo de datos.
2. Utilizando el álgebra relacional, obtenga lo siguiente:
 - a. Los nombres de los profesores del departamento "Informática" que atienden a estudiantes del grupo 3A.
 - b. La especialidad de los estudiantes que trabajan en microcomputadoras marca COMPAQ más de cinco horas semanales.
 - c. Los nombres de los estudiantes de la especialidad "Electrónica" que son atendidos por profesores del departamento "Electrónica", los cuales (los profesores) trabajen, al menos, 10 horas en alguna microcomputadora.
 - d. Los # serie de las microcomputadoras donde han trabajado estudiantes de la especialidad "Industrial" y también profesores del departamento "OCT".

CONSULTAS CON MAYOR NIVEL DE COMPLEJIDAD

Ejercicio # 4

Se cuenta con relaciones que contienen datos de infracciones, propietarios de vehículos y otros datos adicionales, tal como se muestra a continuación:

TB_VEHICULO				TB_PAPELETA			
Num_Placa	Año	Cod_Marca	Cod_Propietario	Num_Papeleta	Fec_Papeleta	Cod_Infraccion	Num_Placa
BN-5478	2007	M03	E0003	P0001	15/08/01	T02	DF-8723
CI-5239	1990	M01	E0002	P0002	25/08/02	T04	JK-8877
DF-4598	2002	M04	E0010	P0003	16/05/00	T08	WQ-8956
DF-8723	1990	M03	E0009	P0004	15/03/01	T03	DW-1245
DG-5899	1997	M02	E0003	P0005	21/08/00	T06	SQ-5688
DW-1245	2000	M04	E0001	P0006	10/08/01	T06	PQ-8720
DW-8842	1996	M03	E0003	P0007	5/08/02	T07	FB-5477
FB-5477	1996	M02	E0001	P0008	16/12/00	T01	CI-5239
FI-8574	1998	M01	E0002	P0009	16/04/01	T04	JK-8877
FS-5541	2008	M03	E0002	P0010	21/01/00	T09	HS-4522
FX-5682	1999	M04	E0006	P0011	25/03/01	T11	DF-8723
GH-8622	2001	M01	E0006	P0012	22/08/02	T12	FB-5477
HJ-5499	2008	M02	E0002	P0013	12/04/00	T09	RY-7841
HJ-9644	1999	M01	E0004	P0014	15/08/02	T01	DF-8723
HS-4522	2008	M02	E0001	P0015	11/08/00	T08	WQ-8956
HU-9951	2001	M03	E0009	P0016	15/06/01	T04	GH-8622
JA-4578	1999	M06	E0001	P0017	25/07/02	T08	DW-1245
JH-8955	2000	M04	E0005	P0018	16/05/02	T05	JK-8877
JK-8877	2007	M03	E0003	P0019	17/09/01	T05	JH-8955
JN-5477	1997	M01	E0004	P0020	16/10/00	T03	DF-8723

TB_MARCA		TB_DISTRITO	
Cod_Marca	Des_Marca	Cod_Distrito	Nombre del Distrito
M01	TOYOTA	D01	Surco
M02	NISSAN	D02	Jesús María
M03	VOLKSWAGEN	D03	San Isidro
M04	MERCEDES BENZ	D04	La Molina
M05	HONDA	D05	San Miguel
M06	MITSUBISHI	D06	Miraflores
M07	DAEWOO	D07	Barranco
M08	PEUGEOT	D08	Chorrillos
M09	CHEVROLET	D09	San Borja
M10	FORD	D10	Lince

TB_PROPIETARIO				
Cod_Propietario	Nom_Propietario	Ape_Propietario	Dir_Propietario	Cod_Distrito
E0001	JUAN RUBEN	SOLANO SOSA	AV . LAS ROSAS 156	D02
E0002	ANA MARIA	PRIETO MAMANI	JR. PUMACAHUA 378	D01
E0003	CARMEN ROSA	ROJAS MARAVI	JR. LOS ALAMOS 671	D04
E0004	MONICA MARIA	LEON ARIAS	PJE. LOS PIRINEOS 455	D04
E0005	JUAN ALBERTO	GALLARDO VARGAS	AV ANGAMOS ESTE 340	D05
E0006	INES MARITZA	MONTOYA RIMACHI	AV. AREQUIPA 2899	D02
E0007	JUJAN JOSE	PEREDA RIOS	JR. PRISMA 210	D01
E0008	ANA ROSA	LOO PEREZ	AV. JAVIER PRADO ESTE 266	D01
E0009	CESAR ALBERTO	SILVA ROMERO	JR. MONTES DE OCA 891	D03
E0010	RUBEN ANGEL	URRUTIA VEGA	JR. EL CARMEN 677	D04

1. Muestre las siguientes consultas:


- a. Seleccione los nombres y apellidos de propietarios que posean vehículos cuya descripción de marca sea Toyota o Ford.

Solución:

R ((TB_VEHÍCULOS JOIN TB_PROPIETARIOS) JOIN TB_MARCA)
(NOM_PROPIETARIO, APE_PROPIETARIO) MARCA='TOYOTA' OR MARCA='FORD'


- b. Seleccione las placas vehiculares, descripción de la marca y Dirección de sus propietarios, de vehículos con papeletas impuestas
- c. Seleccione los nombres y apellidos de propietarios de vehículos del año 1999 o 2008.
- d. Seleccione las papeletas impuestas a vehículos cuyos propietarios residan en La Molina.
- e. Seleccione las placas, apellidos del propietario, dirección y nombre del distrito del propietario de aquellos vehículos que tengan papeletas por la infracción "T05"
- f. Muestre los datos de todos los distritos donde NO RESIDEN PROPIETARIOS.
- g. Muestre las marcas de los vehículos que no han sido adquiridas por un usuario.

Resumen

 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

 <http://www3.uji.es/~mmarques/f47/apun/node33.html>

En esta página web hallará algunos conceptos complementarios a los mostrados en el manual sobre la arquitectura de una base de datos.

 https://www.unoweb-s.uji.es/0203/IS24/1044437011/documentos/ej_algebra.pdf

En esta página web encontrará ejercicios propuestos.

 http://www3.uji.es/~aliaga/e44/prob_alg_calc_sol.pdf

En esta página web encontrará ejercicios resueltos de álgebra relacional.

**UNIDAD DE
APRENDIZAJE****4****SEMANA****9**

MODELO RELACIONAL. NORMALIZACIÓN

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos organizan datos no estructurados y los integran al diseño de la base de datos a partir de documentos comerciales propuestos por el profesor utilizando la teoría de la Normalización hasta la Primera Forma normal (1FN).

TEMARIO

Modelo Relacional

Normalización

Primera Forma Normal

ACTIVIDADES PROPUESTAS

1. Explicaciones por parte del facilitador combinadas con ejemplos.
2. Se efectúan preguntas de comprobación dirigidas a los alumnos sobre los temas tratados.

1. MODELO RELACIONAL

Uno de los modelos matemáticos más importantes y actuales para la representación de las bases de datos, es el enfoque relacional.

Se basa en la teoría matemática de las relaciones, suministrándose por ello una fundamentación teórica que permite aplicar todos los resultados de dicha teoría a problemas tales como el diseño de sublenguajes de datos y otros.

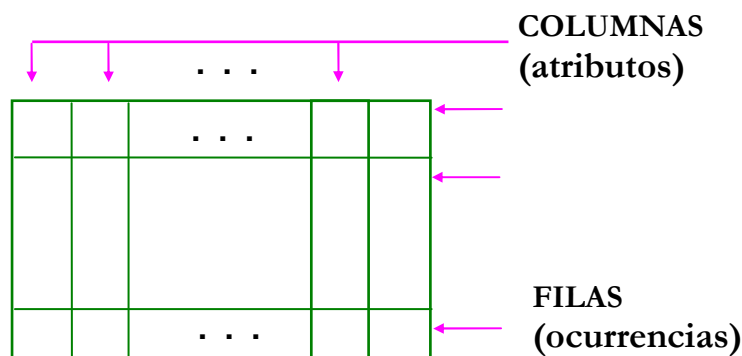
El término relación se puede definir matemáticamente como sigue:

Definición: Relación

Dados los conjuntos D_1, D_2, \dots, D_n (no necesariamente distintos), R es una relación sobre esos n conjuntos si está constituida por un conjunto de n -tuplos ordenados d_1, d_2, \dots, d_n tales que $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$.

Los conjuntos D_1, D_2, \dots, D_n se llaman dominios de R y n constituye el grado de la relación. La cantidad de tuplas constituye la cardinalidad (tipo de relación de correspondencia 1-1, 1-n, n-1 y n-n)

Es conveniente representar una relación como una tabla bidimensional donde cada fila representa un n -tuplo.



En el modelo relacional, tanto los objetos o entidades, como las relaciones que se establecen entre ellos, se representan a través de "tablas", que en la terminología relacional se denominan relaciones.

Cada relación está compuesta de filas (las ocurrencias de los objetos) y se les denomina, en la terminología relacional, como tuplos, tuplas o uplos, uplas (en realidad, n -tuplos, pero en muchos casos se suprime la n cuando no existe posibilidad de confusión).

También la relación está compuesta por columnas (los atributos o campos que toman valores en sus respectivos dominios)

Es importante lo siguiente:

1. No hay dos filas (tuplas) iguales.
 2. El orden de las filas no es significativo.
- (1 y 2 se deben a que la relación es un conjunto)

Siendo rigurosos, el orden de las columnas sí es significativo, pues representa el orden de los dominios implicados, pero como siempre nos referimos a una columna por su nombre y nunca por su posición relativa:

3. El orden de las columnas no es significativo.
4. Cada valor dentro de la relación (cada valor de un atributo) es un dato atómico (o elemental), es decir, no descomponible; por ejemplo: un número, una cadena de caracteres. En otras palabras, en cada posición fila, columna existe un solo valor, nunca un conjunto de valores.

Una relación que satisface este último punto se denomina "normalizada", aunque veremos más adelante que, en realidad, lo que ocurre es que está en Primera Forma Normal.

La teoría de la normalización se basa en la necesidad de encontrar una representación del conjunto de relaciones que en el proceso de actualización sea más adecuada. Llevar una relación no normalizada a normalizada es muy simple. Existen diferentes niveles de normalización que se llaman formas normales que veremos más adelante.

Ejemplo:

Veamos cómo nuestro ejemplo de suministrador y producto se puede representar fácil y claramente mediante el modelo relacional.

Los atributos de estas dos entidades son:

Suministrador: Número, que lo identifica, Nombre, Tipo y Distrito donde radica.

Producto: Número, que lo identifica, Nombre, Precio unitario y Peso.

Además, un suministrador puede suministrar muchos productos y un producto puede ser suministrado por varios suministradores. Se conoce la cantidad de un determinado producto que suministra un suministrador dado.

SUMINISTRADOR

SNUM	SNOM	TIPO	DIST
S1	PÉREZ	30	SAN ISIDRO
S2	RAMOS	10	SURCO
S3	ARENAS	20	SAN ISIDRO
S4	VALLE	20	LINCE
S5	LÓPEZ	15	LINCE

PRODUCTO

<u>PNUM</u>	PNOM	PRECIO	PESO
P1	CLAVO	0.10	12
P2	TUERCA	0.15	17
P3	MARTILO	3.50	80
P4	TORNILLO	0.20	10
P5	ALICATE	2.00	50
P6	SERRUCHO	4.00	90

SP

<u>SNUM</u>	<u>PNUM</u>	CANT
S1	P1	3
S1	P2	2
S1	P3	4
S1	P4	2
S1	P5	1
S1	P6	1
S2	P1	3
S2	P2	4
S3	P3	4
S3	P5	2
S4	P2	2
S4	P4	3
S4	P5	4

La representación en el modelo relacional es más simple que con el modelo jerárquico y el modelo reticular, ya que con tres (3) tablas se tiene todo el modelo representado.

En el modelo relacional, el resultado de una demanda es también una relación y las demandas simétricas (en el sentido de ser una la inversa de la otra; por ejemplo, recuperar los números de los suministradores que suministran el producto 'P4' y recuperar los números de los productos que suministra el suministrador 'S2') requieren operaciones simétricas.

Las diversas formas de expresar las recuperaciones dan lugar a los lenguajes relacionales cuyas formas más representativas son:

- Álgebra relacional (basado en las operaciones del álgebra de relaciones)
- Cálculo relacional (basado en el cálculo de predicados)

Ventajas del modelo relacional:

- Una de las principales ventajas es su simplicidad, pues el usuario formula sus demandas en términos del contenido informativo de la base de datos sin tener que atender a las complejidades de la realización del sistema, lo que implica gran independencia de los datos.
- La información se maneja en forma de tablas, lo que constituye una manera familiar de representarla.

- Al igual que en el modelo reticular, si se tienen relaciones normalizadas, no surgen dificultades grandes en la actualización.

Veamos en el modelo del SUMINISTRADOR-PRODUCTO presentado anteriormente, un ejemplo de cada tipo de operación de actualización:

Creación: Añadir un producto P7. Se agrega la nueva ocurrencia en la tabla PRODUCTO. Es posible hacerlo aunque ningún suministrador lo suministre.

Eliminación: Se puede eliminar el suministrador S1 sin perder el producto P6, a pesar de que es el único suministrador que lo suministra.

Modificación: Se puede cambiar el precio del producto P2 sin necesidad de búsquedas adicionales ni posibilidad de inconsistencias.

No obstante, veremos que el proceso de normalización no es suficiente hasta el punto aquí visto.

Desventajas: Se dice que la fundamental consiste en la dificultad de lograr productividad adecuada de los sistemas, ya que no se emplean los medios técnicos idóneos, tales como las memorias asociativas, siendo necesario simular este proceso, pero, en realidad, la eficiencia y productividad de los sistemas actuales resultan realmente muy satisfactorias.

Ejemplos de SGBD relacionales:

Query By Example (QBE)(IBM)

dBase, FoxPro, Informix, Oracle, SQL Server

2. NORMALIZACIÓN

La teoría de la normalización se ha desarrollado para obtener estructuras de datos eficientes que eviten las anomalías de actualización. El concepto de normalización fue introducido por E.D. Codd y fue pensado para aplicarse a sistemas relacionales. Sin embargo, tiene aplicaciones más amplias.

La normalización es la expresión formal del modo de realizar un buen diseño. Provee los medios necesarios para describir la estructura lógica de los datos en un sistema de información.

Ventajas de la normalización:

- Evita anomalías en la actualización.
- Mejora la independencia de los datos, permitiendo realizar extensiones de la base de datos, afectando muy poco, o nada, a los programas de aplicación existentes que accedan la base de datos.

La normalización involucra varias fases que se realizan en orden. La realización de la 2da. fase supone que se ha concluido la 1ra. y así sucesivamente. Tras completar cada fase se dice que la relación está en:

Primera Forma Normal (1FN)

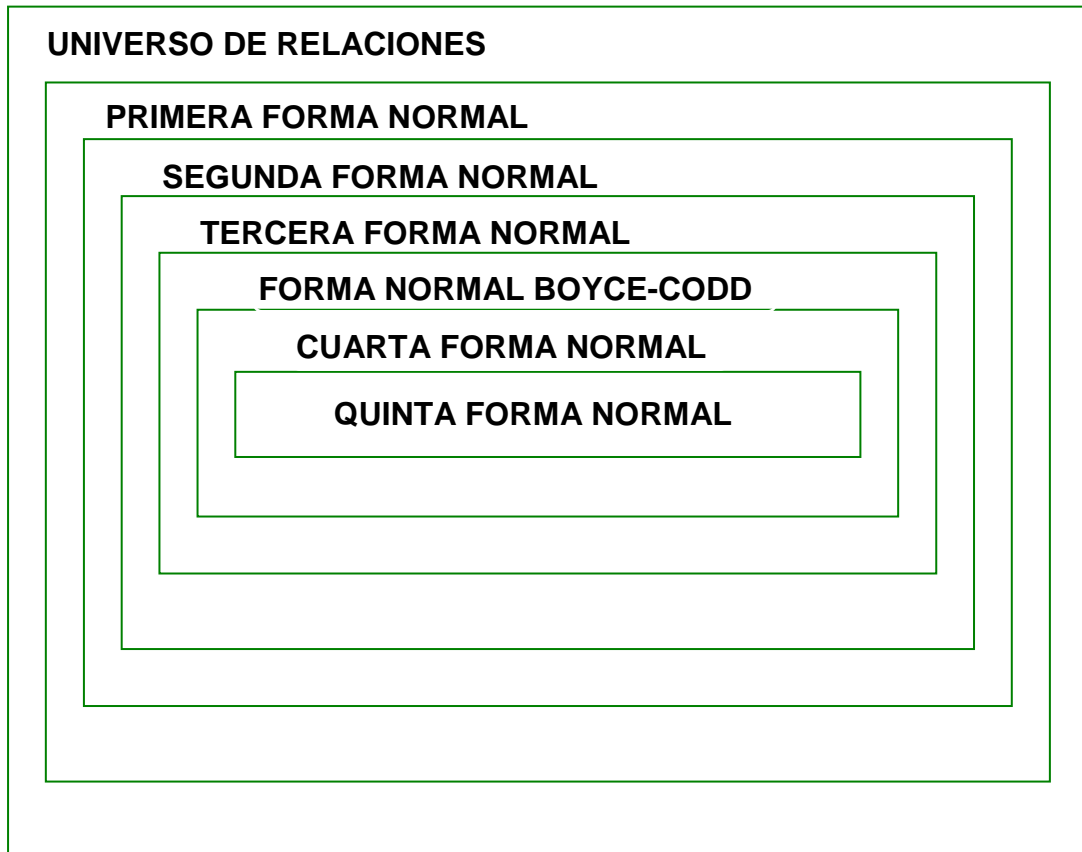
Segunda Forma Normal (2FN)

Tercera Forma Normal (3FN)

Forma Normal de *Boyce-Codd* (FNBC)

Existen, además, la Cuarta (4FN) y la Quinta (5FN) Formas Normales.

Las relaciones en 1FN son un subconjunto del universo de todas las relaciones posibles. Las relaciones en 2FN son un subconjunto de las que están en 1FN y así sucesivamente, como se muestra en la siguiente figura:



3. PRIMERA FORMA NORMAL (1FN)

Para las explicaciones de los contenidos correspondientes a las formas normales de la 1FN hasta la 3FN, desarrollaremos un ejemplo que consiste en el diseño de la base de datos para la automatización del control de los pedidos de productos y que se presenta a continuación. Supongamos que los modelos para pedir los productos sean como se muestra en la siguiente figura:

FECHA: 25/4/2002	PEDIDO NRO. :		123456	
	PROVEEDOR NRO. :		75621	
	NOMBRE PROVEEDOR:		JUAN PÉREZ	
	DIRECCIÓN PROVEEDOR:		AV. LOMAS 123	
DETALLE DEL PEDIDO				
NÚMERO PRODUCTO	DESCRIPCIÓN	PRECIO UNITARIO	CANTIDAD	TOTAL
969715	TELEVISOR	600.00	1	600.00
439124	ANTENA	20.00	10	200.00
439126	ESPIGA	10.00	10	100.00
IMPORTE TOTAL				900.00

PASO 1: se lista los atributos y se determina la llave de toda la relación

- a. El análisis de este modelo de pedido de productos muestra que los atributos que se listan a continuación son de interés:

<u>NUM_PED:</u>	número del pedido
FEC_PED:	fecha en que se realiza el pedido
NUM_PROV:	número del proveedor
NOM_PROV:	nombre del proveedor
DIR_PROV:	dirección del proveedor
NUM_PROD:	número del producto
DES_PROD:	descripción del producto
PR_UN_PROD:	precio unitario del producto
CANT_PROD_PED:	cantidad de unidades del producto que se solicita
PR_PROD_PED:	precio a pagar por concepto de ese producto; corresponde a la columna TOTAL
PR_PED:	precio a pagar por todo el pedido; corresponde al IMPORTE TOTAL

- b. El número de pedido es único y se utiliza para referirse a un pedido, por tanto, se **usará como clave (llave)**.

Nota: esta es una suposición inicial, que luego se discutirá en la siguiente sesión.

- c. Se determina los subconjuntos posibles.
- NUM_PROD: número del producto
 - DES_PROD: descripción del producto
 - PR_UN_PROD: precio unitario del producto
 - CANT_PROD_PED: cantidad de unidades del producto que se solicita
 - PR_PROD_PED: precio a pagar por concepto de ese producto;
corresponde a la columna TOTAL

- d. Se determina el atributo del cual dependen los demás atributos total o parcialmente, en este caso es NUM_PROD, para crear la clave compuesta.

- e. Luego, se determina la relación resultante:

PEDIDO (NUM_PED, FEC_PED, NUM_PROV, NOM_PROV, DIR_PROV,
NUM_PROD, DES_PROD, PR_UN_PROD, CANT_PROD_PED,
PR_PROD_PED, PR_PED)

Definición: Primera Forma Normal (1FN)

- **La definición formal de 1FN es:** una relación está en 1FN si cumple la propiedad de que sus dominios no tienen elementos que, a su vez, sean conjuntos.

Los valores que puede tomar un atributo de una relación son los elementos contenidos en su correspondiente dominio.

Si se permitiera que un elemento del dominio de un atributo fuera un conjunto, entonces dicho atributo pudiera tomar como valor ese conjunto de valores. Eso implicaría que en una posición fila, columna habría un conjunto de valores y no un único valor, como hemos visto que debe ocurrir en el modelo relacional.

- **Otra definición:** toda relación normalizada, o sea, con valores atómicos de los atributos, está en 1FN.
- **Otra definición:** una relación está en 1FN si no incluye ningún grupo repetitivo. (Un grupo repetitivo es un atributo que contiene un conjunto de valores y no un único valor)

Se puede observar que la relación **PEDIDO** contiene 5 atributos repetitivos:

- NUM_PROD
- DES_PROD
- PR_UN_PROD
- CANT_PROD_PED
- PR_PROD_PED

Ya que un pedido puede contener más de una línea de pedido y, por lo tanto, puede contener varios números de producto (NUM_PROD), varias descripciones de producto (DES_PROD), varios precios unitarios (PR_UN_PROD), varias cantidades (CANT_PROD_PED) y varios precios por concepto del producto (PR_PROD_PED).

Esta situación acarrea problemas de actualización, pues habría que decidir la cantidad máxima de líneas de pedido de productos que se permitiría en un pedido, dado que los campos de la tabla PEDIDO deben tener un tamaño dado y, entonces, serían capaces de almacenar sólo una determinada cantidad máxima de valores cada uno de ellos.

Esto sería agregar una limitación en el modelo pues no tiene ese comportamiento en la realidad. Además, en general, se desaprovecharía memoria, dado que si se decide, por ejemplo, que se va a permitir hasta 20 líneas de pedido en cada pedido, habría que definir tamaños de campos para los grupos repetitivos que permitieran almacenar esa cantidad de valores. Entonces, si en un pedido se solicitan menos de 20 productos, lo cual puede ser muy frecuente, no se utilizaría parte del espacio reservado para cada campo.

PASO 2: determinar las relaciones de grupos repetidos de los que no lo son

Hay que eliminar esos grupos repetitivos para que la relación esté en 1FN. Para ello, se crea:

1. Una relación para los campos que sean únicos, es decir, se dejan en la relación original sólo los atributos que no son repetitivos:

PEDIDO (NUM_PED, FEC_PED, NUM_PROV, NOM_PROV, DIR_PROV, PR_PED)

2. Se crea una relación para los grupos repetitivos. Además, se crea una llave compuesta formada por la llave primaria de la relación original (NUM_PED) y el atributo del cual dependen los demás atributos repetidos total o parcialmente, en este caso es NUM_PROD⁵:

PED-PROD (NUM_PED, NUM_PROD, DES_PROD, PR_UN_PROD, CANT_PROD_PED, PR_PROD_PED)

PASO 3: determinación de la llave de cada relación

Ambas tienen como llave, o parte de la llave, a NUM_PED. Pero en **PED-PROD** es necesario la llave compuesta para identificar los productos individuales pues, por ejemplo, CANT_PROD_PED se refiere a la cantidad en que se pide **un determinado producto en un pedido dado** (el producto puede solicitarse en otros pedidos en diferentes cantidades y en el pedido se pueden estar pidiendo otros productos en diferentes cantidades) y, por lo tanto, para identificar una cantidad dada es necesario referirse al pedido y al producto correspondientes.

⁵ NUM_PROD fue definido como atributo clave en el paso 1 de la Primera Forma Normal.

Ahora estas nuevas dos relaciones en 1FN modelan el que nos ocupa. Los problemas de actualización mencionados anteriormente quedan resueltos con este nuevo modelo.

En lugar de tener varios valores en cada campo de acuerdo a la cantidad de líneas de pedido, tal y como ocurría en la tabla **PEDIDO** original, se tienen varias ocurrencias en la tabla **PED-PROD**, una por cada producto que se solicita en el pedido. Esto permite que se soliciten tantos productos como se desee en cada pedido, pues sólo significa agregar una nueva ocurrencia en la relación **PED-PROD** por cada producto solicitado.

Sin embargo, este modelo en 1FN tiene aún problemas de actualización, como se muestra en las siguientes operaciones:

Creación: la información sobre un nuevo producto no se puede insertar si no hay un pedido que lo incluya.

Eliminación: eliminar una línea de pedido que sea la única que pida un producto implica perder la información del producto.

Modificación: por cada línea de pedido en la que se solicite determinado producto se tiene una ocurrencia en **PED-PROD**, que repite la información sobre éste. Si cambia algún atributo del producto, entonces es necesario hacer muchas actualizaciones.

Entonces será necesario aplicar formas normales más fuertes a este modelo para eliminar los problemas de actualización que presenta, como veremos en la próxima sesión.

4. AUTOEVALUACIÓN

Ejercicio 1:

En un laboratorio productor de medicamentos se tiene una relación para almacenar la información acerca de la composición de cada medicamento, de modo de saber cada compuesto que contiene el medicamento y la cantidad de éste necesario para su elaboración:

medicamento (cód_med, nom_med, presentación, nom_comp, cant_unid_comp)

por lo que una ocurrencia de esta relación podría ser, por ejemplo:

<u>cod_med</u>	nom_med	presentacion	nom_comp	cant_unid_comp
M-0456	ComplejoB	grageas	Vit. B1-Vit. B2-Vit. B12-Vit. B6	25-16-18-20

- Analice los inconvenientes que tiene este diseño
- ¿Está la relación **medicamento** en 1FN? Explique su respuesta.

Ejercicio 2:

En un centro de estudios se tiene diseñada una relación **estudiante** para controlar los resultados docentes de éstos de la siguiente forma:

estudiante (DNI_{est}, nombre_{est}, edad_{est}, sexo_{est}, nombre_{curso}, calificación)

por lo que una ocurrencia de esta relación podría ser, por ejemplo:

<u>DNI_est</u>	nombre_est	Fec_nac -est	sexo_est	nombre_curso	calificacion_curso
45378902	José García Pérez	18	M	Matemática- Física-Inglés	17-19-18

- Analice los inconvenientes que tiene este diseño
- ¿Está la relación **estudiante** en 1FN? Si no lo está, aplique la primera forma normal convenientemente y analice si subsisten problemas de actualización en el nuevo modelo obtenido

Resumen

📖 En el modelo relacional, tanto las entidades como las relaciones se representan como relaciones (tablas). Las ocurrencias (de las entidades o de las relaciones) se almacenan como filas y las columnas son los atributos (de las entidades o de las relaciones).

📖 Para toda tabla del modelo relacional se cumple lo siguiente:

- 1) No hay dos filas (tuplas) iguales.
- 2) El orden de las filas no es significativo.
- 3) El orden de las columnas no es significativo.
- 4) Cada valor de un atributo es un dato atómico (o elemental).

📖 La normalización es la expresión formal del modo de realizar un buen diseño. Con la aplicación de la teoría de la normalización se evitan los problemas de actualización.

📖 La normalización está constituida por pasos sucesivos:

- Primera Forma Normal (1FN)
- Segunda Forma Normal (2FN)
- Tercera Forma Normal (3FN)
- Forma Normal de *Boyce-Codd* (FNBC)

Para que una relación esté en 2FN tiene que estar en 1FN, para que esté en 3FN tiene que estar en 2FN y así sucesivamente.

📖 Una relación está en 1FN si no incluye ningún grupo repetitivo.

📖 Para transformar una relación que no está en 1FN en otras que sí lo estén y representen mejor el , se deben eliminar los grupos repetitivos, para lo cual:

1. Se dejan en la relación original sólo los atributos que no son repetitivos.
2. Se extraen en una nueva relación los grupos repetitivos, escogiendo adecuadamente la llave primaria, que siempre estará formada por la llave de la relación original más uno o varios de los atributos repetitivos que permitan distinguir cada valor diferente en dichos campos.

📖 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

🖱 <http://sistemas.itlp.edu.mx/tutoriales/basedat1/>

En esta página web hallará conceptos complementarios a los mostrados en el manual sobre la el Modelo Relacional, así como ejemplos didácticos de su aplicación en los lenguajes de programación de base de datos.

🖱 <http://www.mitecnologico.com/Main/ModeloErYNormalizacion>

Es esta página web encontrará definiciones complementarias a la normalización de datos.

🖱 http://www-gist.det.uvigo.es/~martin/bd/ejemplo_norm.pdf

En esta página web se muestra la normalización de una fuente de datos paso a paso.

**UNIDAD DE
APRENDIZAJE****4****SEMANA****10**

NORMALIZACIÓN

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos organizan datos no estructurados y los integran al diseño de la base de datos a partir de documentos comerciales propuestos por el profesor utilizando la teoría de la Normalización hasta la Tercera Forma normal (3FN)

TEMARIO

Dependencia funcional y otros conceptos

Segunda Forma Normal (2FN)

Tercera Forma Normal (3FN)

Forma Normal de Boyce-Codd (FNBC)

Cuarta Forma Normal (4FN)

Quinta Forma Normal (5FN)

ACTIVIDADES PROPUESTAS

1. Explicaciones por parte del facilitador combinadas con ejemplos.
2. Se efectúan preguntas de comprobación dirigidas a los alumnos sobre los temas tratados.

1. DEPENDENCIA FUNCIONAL Y OTROS CONCEPTOS

Antes de pasar a la 2FN, es necesario abordar algunas definiciones previas que son imprescindibles para explicar las siguientes formas normales:

1.1 Dependencia funcional

Definición: **Dependencia Funcional (DF)**

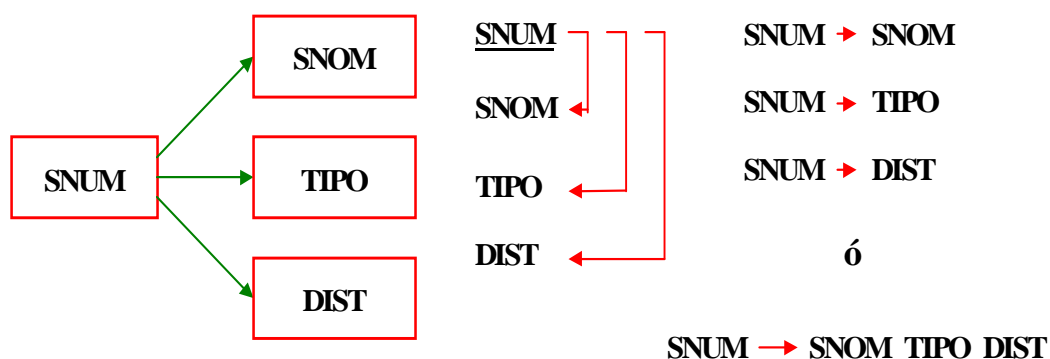
Dada una relación R, se dice que el atributo **Y** de R es funcionalmente dependiente del atributo **X** de R, si y sólo si, cada valor **X** en R tiene asociado a él, precisamente, un valor de **Y** en R en cualquier momento del tiempo.

Y depende funcionalmente de X se denota como:

$$X \longrightarrow Y$$

Ejemplo en la relación SUMINISTRADOR:

SNOM, TIPO y DIST son funcionalmente dependientes cada uno de SNUM, ya que para un valor de SNUM existe un valor correspondiente de SNOM, TIPO y DIST.



Estas cuatro (4) representaciones son formas de mostrar las dependencias funcionales.

El reconocimiento de las dependencias funcionales es una parte esencial de la comprensión de la semántica, del significado del dato. El hecho de que **DIST** dependa funcionalmente de **SNUM** significa que cada suministrador está situado en un distrito.

La noción de dependencia funcional puede ser extendida hasta cubrir el caso en que X, Y o ambos atributos sean compuestos.

1.2 Dependencia funcional completa

Definición: **Dependencia funcional completa**

El atributo **Y** es funcionalmente y completamente dependiente del atributo **X**, si es funcionalmente dependiente de **X** y no es funcionalmente dependiente de algún subconjunto de **X**.

En algunos textos lo representan como: $X \Rightarrow Y$

Ejemplo en la relación SP:

El atributo **CANT** es funcionalmente dependiente del par de atributos (**SNUM= número del suministrador –proveedor- y PNUM= número del producto**). Por ello, se podría denotar CANT como CANT_PRO_SUM, que en el diccionario de datos se detallaría como la cantidad de productos provista por un suministrador.

Esta última explicación muestra que la cantidad tiene dependencia funcional completa tanto del suministrador como del producto (Entidades Fuertes)



1.3 Llave

Al hablar de entidad en el modelo entidad - relación, se asumió que existe una llave para cada entidad, formada por un conjunto de atributos que definen de forma única la entidad. Un concepto análogo se define para las relaciones o tablas en el modelo relacional.

Definición: Llave

Sean:

R una relación con atributos **A1, A2, ..., An** y
X un subconjunto de **A1, A2, ..., An**

Se dice que **X** es una llave de **R** si:

1. $X \rightarrow A1, A2, \dots, An$
 o sea, todos los atributos de la relación dependen funcionalmente de **X**
2. $\nexists Y \subset X \mid Y \rightarrow A1, A2, \dots, An$

Lo planteado en el punto 2. es una condición de minimalidad que no se requería para el concepto de llave en el modelo entidad - relación.

Una relación puede tener varias llaves. Una de ellas se nombra "**llave primaria**" (la que se escoja para trabajar) y las restantes se nombran "**llaves candidatas**".

Una **superllave** será cualquier superconjunto de una llave. Entonces, una llave es un caso especial de superllave.

1.3.1 Procedimiento para hallar las llaves candidatas de una relación

Supongamos que se quiere encontrar las llaves candidatas de una relación $R(A,B,C,D,E)$ con las siguientes dependencias funcionales:

1. $A \rightarrow B$
2. $BC \rightarrow D$
3. $AB \rightarrow E$

Para comenzar, se parte de que no existen más llaves que dependencias funcionales, pues el concepto de llave incluye la existencia de dependencia funcional. Se analiza, por tanto, cada una de las DF presentes en la relación, añadiendo los atributos que sean imprescindibles en la parte izquierda para lograr determinar a todos los atributos de la relación. El conjunto de atributos así formado debe ser mínimo.

Luego se analiza cada uno de esos conjuntos mínimos, de forma que, si alguno es un superconjunto de otro, ya no es llave, sino superllave. Pueden resultar varias llaves candidatas.

En el ejemplo:

1. $A \rightarrow B$ $AC \rightarrow A B E C D$ \implies AC es llave
2. $BC \rightarrow D$ $BCA \rightarrow B C D A E$ \implies BCA es superllave
3. $AB \rightarrow E$ $ABC \rightarrow A B E C D$ \implies ABC es superllave

La única llave es AC. No hay ninguna otra llave candidata, pues en las otras DF se obtiene el mismo conjunto ABC y este conjunto contiene a AC.

Ejemplo:

Sea la relación **R** (ciudad, calle, código postal). Para abreviar, $R(C, A, P)$ donde se tiene que:

- **Una calle en una ciudad tiene un código postal:** $CA \rightarrow P$
- El código postal tiene una estructura tal que su valor determina la ciudad: $P \rightarrow C$
- Pero en una ciudad, varias calles pueden tener el mismo código, por lo que no se cumple $P \rightarrow A$.

Entonces, analizando cada DF se tiene:

- $CA \rightarrow P$, pero CA se determina a sí mismo también: $CA \rightarrow CA$, por lo tanto, $CA \rightarrow CAP$, es decir, que CA determina a todos los atributos de R y, como no existe ningún subconjunto de CA que, a su vez, determine a todos los atributos de la relación, se puede concluir que **AC** es llave
- $P \rightarrow C$ y no determina a A , pero agregando A en el lado izquierdo: $PA \rightarrow CA$ y es obvio que $PA \rightarrow PA$ también, entonces $PA \rightarrow CAP$, y no existe ningún subconjunto de PA que, a su vez, determine a todos los atributos de la relación, por lo tanto, **PA** también es llave

Definición: **Atributo llave**

Un atributo **A_i** de **R** es un atributo llave si él es (o es miembro de) una llave (candidata o primaria).

Definición: **Atributo no llave**

Un atributo **A_j** de **R** es un atributo no llave si él no es miembro de ninguna llave candidata.

Ejemplo: en el caso de la relación **R(C, A, P)** vista anteriormente, **C, A y P** son todos atributos llaves.

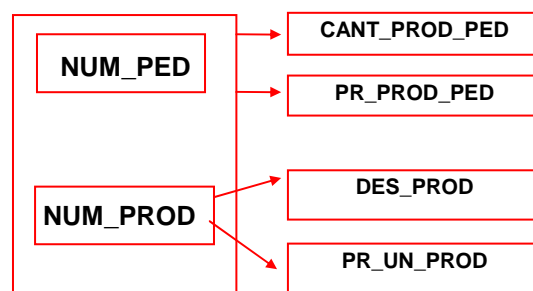
2. SEGUNDA FORMA NORMAL (2FN)

Definición: **Segunda Forma Normal**

Una relación **R** se dice que está en 2FN si está en 1FN y si, y sólo si, los atributos no llaves (ni primarias, ni candidatas) de **R**, si los hubiese, son funcional y completamente dependientes de la llave primaria de **R**.

Entonces, **se aplica sólo a relaciones con llaves compuestas**, pues no es posible que en una relación cuya llave primaria sea simple (compuesta por un solo atributo) haya atributos que dependan de parte de la llave primaria. Una relación que esté en 1FN y que tenga una llave primaria simple, está en 2FN.

Continuando con el ejemplo de los pedidos de productos, habíamos visto que en la relación PED-PROD subsistían problemas de actualización. Analicemos las Dependencia funcionales que existen en dicha relación:



Esta relación no está en 2FN, pues **DES_PROD** y **PR_UN_PROD** no dependen funcional y completamente de la llave (NUM_PED, NUM_PROD).

Paso único: se determina si existen relaciones con clave compuesta. Si **NO** las hay, las relaciones obtenidas en la Primera Forma Normal se encuentran en Segunda Forma Normal. De lo contrario, se efectúa lo siguiente:

1. Se crea una relación para todos los atributos que dependen funcional y completamente de la llave (y los atributos que no se analizan por ser atributos llaves, pertenecientes a claves candidatas).

PED-PROD (NUM_PED, NUM_PROD, CANT_PROD_PED, PR_PROD_PED)

2. Se crea una relación para los atributos que dependan de cada parte (subconjunto) de la llave. La llave de la relación así formada será la parte (subconjunto) de la llave primaria de la cual dependen los atributos.

PRODUCTO (NUM_PROD, DES_PROD, PR_UN_PROD)

Los problemas planteados en la 1FN se resuelven con la 2FN. Veamos:

Creación: se puede insertar la información sobre un producto aunque no haya un pedido que lo solicite.

Eliminación: se puede eliminar una línea de pedido y no se pierde la información sobre el producto, aunque sea el único pedido que pide ese producto.

Modificación: si cambia un atributo del producto, sólo hay que cambiarlo en un lugar. Se elimina redundancia.

Pero aún tenemos problemas en este caso que son similares a los vistos, pero con la relación **PEDIDO** y, específicamente, cuando se trata de insertar, eliminar o modificar la información de proveedores:

Creación: no podemos insertar la información de un proveedor, a menos que haya un pedido para él.

Eliminación: se perderá la información sobre un proveedor al borrar un pedido que era el único que se le hacía a ese proveedor.

Modificación: para cambiar información sobre un proveedor, hay que recorrer todos los pedidos de ese proveedor. Hay redundancia.

3. TERCERA FORMA NORMAL (3FN)

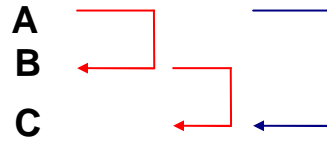
Definición: Tercera Forma Normal

Una relación R está en 3FN si está en 2FN y si, y sólo si, los atributos no llaves son independientes de cualquier otro atributo no llave primaria.

Esto es lo mismo que decir que se deben eliminar las dependencias transitivas de atributos no llaves respecto a la llave primaria, estando ya la relación en 2FN.

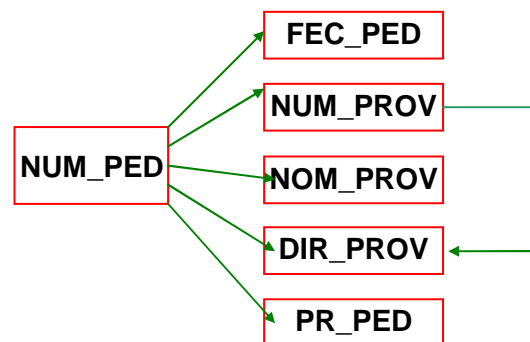
Definición: Dependencia transitiva

Sean **A**, **B** y **C** conjuntos de atributos de una relación **R**. Si **B** es dependiente funcionalmente de **A** y **C** lo es de **B**, entonces **C** depende transitivamente de **A**.



Este paso se ejecuta examinando todas las relaciones para ver si hay atributos no llaves que dependan unos de otros. Si se encuentran, se forma una nueva relación para ellos.

Analicemos las dependencias funcionales que existen en la relación **PEDIDO**:



Paso 1: se remueven los atributos que no dependen de la llave

1. Se crea una relación para los atributos no llaves que no dependen transitivamente de la llave primaria (y los atributos que no se analizan por ser atributos llaves, pertenecientes a claves candidatas).

PEDIDO (NUM_PED, FEC_PED, NUM_PROV, PR_PED)

Paso 2: se remueven los atributos que dependen de la llave

2. Se crea una relación para los atributos no llaves que dependen transitivamente de la llave primaria a través de otro atributo o conjunto de atributos no llave primaria (que no son parte de la llave primaria.) La llave primaria de la relación así formada será el atributo o conjunto de atributos a través de los cuales existe la dependencia transitiva.

PROVEEDOR (NUM_PROV, NOM_PROV, DIR_PROV)

Es necesario analizar las otras relaciones. Puede comprobarse que en las otras relaciones no hay dependencia entre atributos no llaves, por lo que están en 3FN.

Entonces el modelo de datos relacional en 3FN que representa el de los pedidos de productos está formado por las siguientes relaciones:

PEDIDO (NUM_PED, FEC_PED, PR_PED, NUM_PROV)
PED-PROD (NUM_PED, NUM_PROD, CANT_PROD_PED, PR_PROD_PED)
PRODUCTO (NUM_PROD, DES_PROD, PR_UN_PROD)
PROVEEDOR (NUM_PROV, NOM_PROV, DIR_PROV)

La 3FN ha eliminado los problemas asociados con la información sobre el proveedor en la 2FN. Veamos:

Creación: se puede insertar la información de un proveedor, aunque no haya un pedido para él.

Eliminación: al borrar un pedido que era el único que se le hacía a un proveedor, no se perderá la información sobre el proveedor

Modificación: la información sobre un proveedor está en una sola ocurrencia, por lo que, para cambiar cierta información de éste, sólo hay que hacerlo en dicha ocurrencia.

Ya en esta etapa se puede optimizar la 3FN. Puede haber relaciones “degeneradas” que contengan sólo la clave y que la información que aportan esté considerada en otra relación, por lo que se pueden eliminar. Puede que varias relaciones tengan la misma clave, por lo que se pueden combinar en una sola, siempre que el resultado sea lógico y tenga sentido.

Los analistas y diseñadores con experiencia producen relaciones en 3FN casi sin saber o preocuparse de esto y es que utilizan el sentido común y la experiencia para escribir relaciones normalizadas. No obstante, no siempre la intuición es suficiente y la metodología para normalizar las bases de datos se convierte en una herramienta imprescindible, que garantiza un diseño idóneo de los datos.

Merece la pena destacar en este momento algunas otras cuestiones que, aunque no están relacionadas directamente con la teoría de la normalización, sí propician también la realización de un buen diseño de la base de datos:

1. En el ejemplo, se consideran atributos calculables, o también llamados secundarios, que no deben aparecer en el modelo lógico de la base de datos, como, por ejemplo, **PR_PROD_PED** y **PR_PED**. La modificación del precio unitario de un producto (**PR_UN_PROD**), que se logró que apareciera como una única ocurrencia de ese atributo, gracias a la aplicación de la 2FN, implicaría que hubiera que modificar el valor de los atributos calculables **PR_PROD_PED** y **PR_PED** en todas aquellas ocurrencias relacionadas con el producto cuyo precio se modifica, donde quiera que aparecieran, lo cual trae problemas de actualización, que son, precisamente, los que se tratan de evitar con la normalización. Por ello, en el modelo lógico, no deben aparecer atributos calculables. (Se incluyeron en el ejemplo para poder realizar la presente explicación).
2. Siempre que en una relación se escoja correctamente la llave primaria, esa relación ya está en 1FN, debido a la propia definición de llave. En este ejemplo, inicialmente, se partió de trabajar con el atributo **NUM_PED** como llave primaria cuando, en realidad, no lo es. La llave primaria sería el par de

atributos **NUM_PED**, **NUM_PROD**, que son los que garantizan que cada ocurrencia de atributo tenga un solo valor. (Se partió de esa llave no correcta para poder aplicar la 1FN).

Aún en el caso de relaciones que estén en 3FN pueden existir problemas y habrán de ser resueltos con las siguientes formas normales:

4. FORMA NORMAL DE BOYCE-CODD (FNBC)

Antes de explicar la FNBC, es preciso definir el concepto de determinante.

Definición: **Determinante**

Un determinante es cualquier atributo o conjunto de atributos del cual depende funcional y completamente cualquier otro atributo. O sea, la parte izquierda de la implicación cuando la dependencia funcional es completa.

Ejemplo en la relación SUMINISTRADOR:

SNUM es un determinante.

La definición de la 3FN puede resultar inadecuada en el caso de una relación donde ocurre lo siguiente:

1. La relación tiene varias llaves candidatas, donde
2. esas llaves candidatas son compuestas y
3. esas llaves candidatas se solapan (o sea, tienen, al menos, un atributo común)

Es decir, para una relación donde no tengan lugar las tres condiciones anteriores, la FNBC es idéntica a la 3FN. Esas tres condiciones son necesarias, pero no suficientes, para que la relación no esté en FNBC.

Definición: **Forma Normal de Boyce-Codd (FNBC)**

Una relación R está en FNBC si, y sólo si, cada determinante es una superllave (candidata o primaria).

Obsérvese que se habla en términos de llaves candidatas y no sólo de la llave primaria, ya que una llave es un caso especial de superllave y la llave puede ser candidata o primaria.

Además, la definición de FNBC es conceptualmente más simple, aunque es una FN más "fuerte". Una relación que esté en FNBC está también en 1FN, 2FN y 3FN.

Por ejemplo, la relación

PRODUCTO (NUM_PROD, NOM_PROD, PRE_PROD, PESO_PROD)

con las DF : NUM_PROD \rightarrow NOM_PROD, PRE_PROD, PESO_PROD
 NOM_PROD \rightarrow NUM_PROD, PRE_PROD, PESO_PROD

es decir, suponiendo que NUM_PROD y NOM_PROD sean llaves candidatas, está en FNBC, ya que, en todas las DF que existen, los determinantes son llaves candidatas y, por tanto, superllaves de **PRODUCTO**.

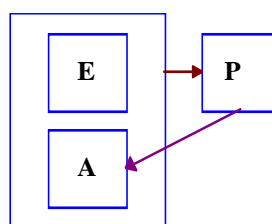
Analicemos otro ejemplo:

Sea la relación **EAP** (Estudiante, Asignatura, Profesor). Para simplificar: **EAP** (E, A, P)

donde una tupla significa que un estudiante **E** recibe la asignatura **A** impartida por el profesor **P** y en la cual se cumple que:

- para cada asignatura, cada estudiante tiene un solo profesor.
- cada profesor imparte sólo una asignatura.
- cada asignatura es impartida por varios profesores.

O sea, $EA \rightarrow P$
 $P \rightarrow A$
 y no existe DF de P con respecto a A



EA es una llave candidata de la relación pues $EA \rightarrow EA$ y $EA \rightarrow P$, por lo que $EA \rightarrow EAP$ y no existe ningún subconjunto de EA que también determine a todos los atributos de la relación. Por otra parte, $P \rightarrow A$ y como $P \rightarrow P$, entonces $P \rightarrow AP$ y, agregando E en el lado izquierdo, se tiene que $EP \rightarrow EAP$, por lo que EP es una llave candidata de la relación, ya que determina a todos los atributos (E, P, A) y no existe ningún subconjunto de EP que también determine a todos los atributos de la relación. Ambas llaves (EA y EP) son compuestas y se solapan, por lo que debe analizarse la FNBC.

Esta relación está en 3FN, pero no en FNBC, ya que el determinante P no es llave de la relación, es decir, existe un determinante que no es llave candidata de la relación.

Supongamos las siguientes ocurrencias de esta relación **EAP**:

EAP

E	A	P
Pérez	Matemática	prof. Blanco
Pérez	Física	prof. Valdés
Rodríguez	Matemática	prof. Blanco
Rodríguez	Física	prof. Hernández

Esta relación presenta anomalías de actualización. Por ejemplo:

Eliminación: Si queremos eliminar la información de que Rodríguez estudia Física, perdemos la información de que el profesor Hernández imparte Física.

Estos problemas pueden eliminarse sustituyendo la relación original **EAP** por dos relaciones, quedando las siguientes columnas:

EP (E, P) y **PA** (P, A)

O sea, se separa la DF problemática ($P \rightarrow A$) en una relación independiente, donde el determinante será la llave (P), y se forma otra relación, donde dicho determinante P participará como atributo, pero en la cual no puede participar el atributo determinado en la dependencia funcional conflictiva (A).

Otro ejemplo:

Sea la relación **R** (ciudad, calle, código postal), que ya presentamos anteriormente. Para abreviar, $R(C, A, P)$

donde se tiene que:

Una calle en una ciudad tiene un código postal: $CA \rightarrow P$

- El código postal tiene una estructura tal que su valor determina la ciudad: $P \rightarrow C$
- Pero en una ciudad, varias calles pueden tener el mismo código, por lo que no se cumple $P \rightarrow A$.

Ya habíamos visto que esta relación tenía a **CA** y **PA** como llaves candidatas.

Esta relación no está en FNBC, ya que existe un determinante (P) que no es superllave.

La solución será dividir **R** en

R1 (P, C) y **R2** (A, P)

Otro ejemplo:

Sea la relación **EXAM** (E, A, L) donde:

E: estudiante

A: asignatura

L: lugar en el escalafón alcanzado por un estudiante en una asignatura y en la que se supone que 2 estudiantes no pueden alcanzar el mismo lugar en una asignatura.

Entonces:



Como se ve, existen dos llaves candidatas, **EA** y **AL**, las cuales se solapan, pero no existen otros determinantes que no sean esas llaves candidatas (que son casos especiales de superllaves), por lo que la relación **EXAM** está en FNBC. Este es un ejemplo de que las tres condiciones (que existan varias llaves, compuestas y que

se solapen) son necesarias, pero no suficientes para que una relación no esté en FNBC.

Por último, es necesario destacar que la descomposición de una relación para obtener la FNBC puede implicar que se pierdan dependencias funcionales. Por ejemplo, en la relación **EAP** donde **EA** → **P** y **P** → **A** al descomponerse en

EP (E, P) y **PA** (P, A)

se pierde la DF **EA** → **P**, ya que esta DF no puede ser deducida a partir de las que existen en **EP** y **PA** (que sólo es **P** → **A**), lo cual implica que ambas relaciones **EP** y **PA** no pueden ser actualizadas "independientemente", sino que una actualización en un lugar conlleva un chequeo de actualización en el otro lugar (para añadir un profesor a un estudiante hay que chequear la materia que el profesor imparte y hay que chequear si el estudiante ya tiene un profesor de esa materia y en ese caso no se puede añadir).

5. CUARTA FORMA NORMAL

Un esquema de relaciones **R** está en 4FN con respecto a un conjunto **D** de dependencias funcionales y de valores múltiples sí, para todas las dependencias de valores múltiples en **D** de la forma **X** → **Y**, donde **X** ≤ **R** y **Y** ≤ **R**, se cumple por lo menos una de estas condiciones:

- **X** → **Y** es una dependencia de valores múltiples trivial.
- * **X** es una superllave del esquema **R**.

Para entender mejor aún esto consideremos una afinidad (tabla) llamada estudiante que contiene los siguientes atributos: Clave, Especialidad y Curso, tal y como se demuestra en la siguiente figura:

Clave	Especialidad	Curso
S01	Sistemas	Natación
S01	Bioquímica	Danza
S01	Sistemas	Natación
B01	Bioquímica	Guitarra
C03	Civil	Natación

Suponemos que los estudiantes pueden inscribirse en varias especialidades y en diversos cursos. El estudiante con clave S01 tiene su especialidad en sistemas y Bioquímica y toma los cursos de Natación y danza, el estudiante B01 tiene la especialidad en Bioquímica y toma el curso de Guitarra, el estudiante con clave C03 tiene la especialidad de Civil y toma el curso de natación.

En esta tabla o relación no existe dependencia funcional porque los estudiantes pueden tener distintas especialidades, un valor único de clave puede poseer muchos valores de especialidades al igual que de valores de cursos. Por lo tanto existe dependencia de valores múltiples. Este tipo de dependencias produce redundancia de datos, como se puede apreciar en la tabla anterior, en donde la clave S01 tiene tres registros para mantener la serie de datos en forma independiente lo cual ocasiona que al realizarse una actualización se requiera de demasiadas operaciones para tal fin.

Existe una dependencia de valores múltiples cuando una afinidad tiene por lo menos tres atributos, dos de los cuales poseen valores múltiples y sus valores dependen solo del tercer atributo, en otras palabras en la afinidad R (A,B,C) existe una dependencia de valores múltiples si A determina valores múltiples de B, A determina valores múltiples de C, y B y C son independientes entre sí.

En la tabla anterior, Clave determina valores múltiples de especialidad y clave determina valores múltiples de curso, pero especialidad y curso son independientes entre sí.

Las dependencias de valores múltiples se definen de la siguiente manera:

- Clave \rightarrow Especialidad y Clave \rightarrow Curso: esto se lee "Clave multidetermina a Especialidad, y clave multidetermina a Curso"

Para eliminar la redundancia de los datos, se deben eliminar las dependencias de valores múltiples. Esto se logra construyendo dos (2) tablas, donde cada una almacena datos para solamente uno de los atributos de valores múltiples.

Para el ejemplo planteado, las tablas correspondientes son:

Tabla Eespecialidad

Clave	Especialidad
S01	Sistemas
B01	Bioquímica
C03	Civil

Tabla Ecurso

Clave	Curso
S01	Natación
S01	Danza
B01	Guitarra
C03	Natación

6. QUINTA FORMA NORMAL

Un esquema de relaciones R está en 5FN con respecto a un conjunto D de dependencias funcionales, de valores múltiples y de producto, si para todas las dependencias de productos en D se cumple por lo menos una de estas condiciones:

- $(R_1, R_2, R_3, \dots, R_n)$ es una dependencia de producto trivial.
- Toda R_i es una superllave de R.

La quinta forma normal se refiere a dependencias que son extrañas. Tiene que ver con tablas que pueden dividirse en subtablas pero que no pueden reconstruirse.

7. AUTOEVALUACIÓN

- a. En centro de estudios, se tiene diseñada una relación estudiante para controlar los resultados docentes de éstos de la siguiente forma:

estudiante (DNI_est, nombre_est, fec_nac_est, sexo_est, cod_curso, nombre_curso, calif_curso)

por lo que una ocurrencia de esta relación podría ser, por ejemplo:

DNI_est	nombre_est	fec_nac_est	sexo_est	cod_curso	nombre_curso	calif_curso
45378902	José García Pérez	25/07/1966	M	03-07-09	Matemática-Física-Inglés	17-19-18

Analice los inconvenientes que tiene este diseño y aplique las formas normales convenientemente para mejorarlo.

- b. La dirección de recursos humanos de una universidad, desea controlar la información sobre los docentes, de los cuales se conoce:

-DNI
-nombre
-categoría docente
-antigüedad
-salario

Cada docente posee una categoría determinada, pero una categoría puede serlo de muchos docentes. Cada docente posee una antigüedad determinada, pero una antigüedad puede serlo de muchos docentes. Para una categoría docente dada y una antigüedad determinada, el salario siempre es igual.

- 1) Determine las dependencias funcionales suponiendo que inicialmente todos los datos se encuentran agrupados en una relación.
- 2) Determine las llaves candidatas y señale la llave primaria.
- 3) Diseñe el modelo de datos relacional normalizando las relaciones hasta la 3FN. Deben aparecer todas las relaciones que se vayan obteniendo en cada paso de normalización.

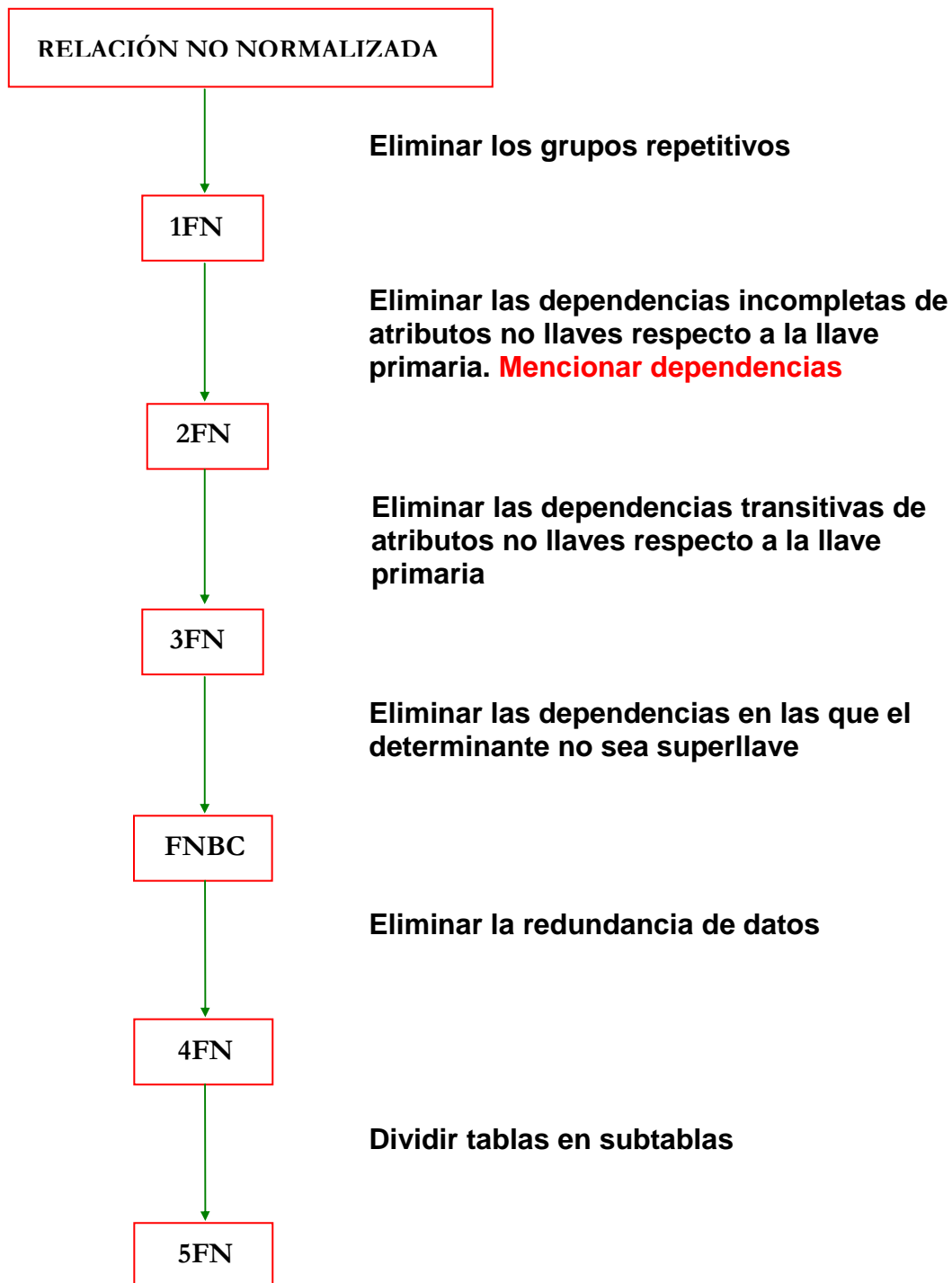
- c. Determine las dependencias funcionales que están presentes en el siguiente :


En una videoteca se desea diseñar una base de datos para el control de los préstamos de películas. De cada película se conoce su código, que la identifica, su título y la duración que tiene. Una película es de un género y un género puede serlo de muchas películas. De cada género se conoce su código, que lo identifica y su nombre. De cada película se tienen muchas copias, pero una copia lo es de una sola película. De cada copia se sabe su código, que la identifica, la marca del cassette en que está grabada y su estado de conservación. Una copia se les puede prestar a muchos usuarios y a un usuario se le pueden prestar muchas copias. No obstante, se sabe que a un determinado usuario, en una fecha dada, se le presta una sola copia. También se sabe que una determinada copia, en una fecha dada, se le presta a un solo usuario. De cada usuario se conoce su DNI, que lo identifica, su nombre, su dirección y su ocupación.

Resumen

- 📖 Dada una relación R , se dice que el atributo Y de R es **funcionalmente dependiente** del atributo X de R , si y sólo si, cada valor X en R tiene asociado a él, precisamente, un valor de Y en R en cualquier momento del tiempo.
- 📖 El atributo Y es **funcionalmente y completamente dependiente** del atributo X , si es funcionalmente dependiente de X y no es funcionalmente dependiente de algún subconjunto de X .
- 📖 Se dice que X es una **llave** de R si:
 1. $X \rightarrow A_1, A_2, \dots, A_n$
o sea, todos los atributos de la relación dependen funcionalmente de X
 2. $\nexists Y \subset X \mid Y \rightarrow A_1, A_2, \dots, A_n$
- 📖 Una relación R se dice que está en **2FN** si está en 1FN y si, y sólo si, los atributos no llaves (ni primarias, ni candidatas) de R , si los hubiese, son funcional y completamente dependientes de la llave primaria de R .
- 📖 Una relación R está en 3FN si está en 2FN y si, y sólo si, los atributos no llaves son independientes de cualquier otro atributo no llave primaria. Esto es lo mismo que decir que se deben eliminar las dependencias transitivas de atributos no llaves respecto a la llave primaria, estando ya la relación en 2FN.
- 📖 Un **determinante** es cualquier atributo o conjunto de atributos del cual depende funcional y completamente cualquier otro atributo. O sea, la parte izquierda de la implicación cuando la dependencia funcional es completa.
- 📖 Una relación que tenga varias llaves candidatas, que sean compuestas y que se solapen cumple con las tres condiciones necesarias, pero no suficientes, para que la relación no esté en FNBC.
- 📖 Una relación R está en FNBC si, y sólo si, cada determinante es una superllave (candidata o primaria).

De modo resumido, se puede decir que los pasos que se deben dar en el proceso de normalización son:



 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

 <http://sistemas.itlp.edu.mx/tutoriales/basedat1/>

En esta página web se muestra conceptos complementarios de cada paso de la normalización de datos.

 http://www-gist.det.uvigo.es/~martin/bd/ejemplo_norm.pdf

En esta página web se muestra la normalización de una fuente de datos paso a paso.

**UNIDAD DE
APRENDIZAJE****4****SEMANA****11**

NORMALIZACIÓN

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos organizan datos no estructurados y los integran al diseño de la base de datos a partir de documentos comerciales propuestos utilizando la teoría de la Normalización hasta la Tercera Forma normal (3FN).

TEMARIO

Ejercicios propuestos

ACTIVIDADES PROPUESTAS

1. Se realiza un ejemplo que abarque todos los contenidos a ejercitar por parte del facilitador, en colaboración con los alumnos.
2. Se desarrollan de ejercicios sencillos por parte de los alumnos con la orientación del facilitador.

1. NORMALIZACIÓN. EJEMPLO

Para abordar un problema en el que pretendemos realizar el diseño lógico de la base de datos a través de la aplicación de la teoría de la normalización, debemos organizar nuestro análisis, por lo que proponemos a continuación los pasos se deben dar para ello:

1.1 Pasos a dar para la aplicación de la normalización

- Determinar las DF presentes en el
- Representar en una única relación todos los atributos que participan, y son significativos, en el
- Determinar las llaves candidatas y seleccionar la llave primaria. Indicar la llave primaria en la relación que representa el . (Al haber escogido bien la llave, se parte de una relación que está en 1FN)
- Aplicar la 2FN
- Aplicar la 3FN
- Analizar las relaciones obtenidas para optimizarlas (unir las que tenga sentido unir; eliminar las que no aporten información)

El primero y el segundo paso se pueden intercambiar de orden o, incluso, irse haciendo simultáneamente, pero deben realizarse antes de los siguientes.

Es importante emplear nombres semánticos para las relaciones y los atributos, que permitan captar el significado del modelo lo más claramente posible

1.2 Ejemplo de normalización hasta la 3FN

A continuación desarrollaremos, detalladamente, un ejemplo de aplicación de la normalización:

Se desea diseñar una base de datos para controlar la disponibilidad de materiales de construcción. De cada proveedor de materiales se conoce el código (**cprov**) que lo identifica, su nombre (**nomprov**) y el distrito en que radica (**dist**). De cada material se sabe el código (**cmat**) que lo identifica, su descripción (**desc**), la unidad de medida que se aplica al material (**um**) y el precio por unidad de medida (**precio**). Para guardar estos materiales hasta su posterior distribución, existen diversos almacenes. De cada almacén se conoce el código (**calm**) que lo identifica, su dirección (**diralm**) y la capacidad de almacenaje (**capac**). Un proveedor puede suministrar varios materiales y un material puede ser suministrado por diferentes proveedores. Se sabe que un material suministrado por un proveedor está en un solo almacén y, además, se sabe qué cantidad de un material suministrado por un proveedor se encuentra en el almacén (**cantmat**). En un almacén se guardan distintos materiales y pueden existir varios almacenes donde se guarde un mismo material.

1.2.1 Determinar las DF

Veamos las dependencias funcionales que se derivan de esta situación:

Como **cprov** identifica al proveedor:

a. $cprov \rightarrow nomprov \text{ dist}$

Como **cmat** identifica al material:

b. $cmat \rightarrow desc \text{ um } \text{ precio}$

Como **calm** identifica al almacén:

c. $calm \rightarrow diralm \text{ capac}$

d. $\begin{array}{cccccccccccc} & & & 1 & & 2 & & 3 & & 4 & & 5 & & 6 & & 7 & & 8 \\ cprov & cmat & \rightarrow & nomprov & dist & desc & um & precio & calm & diralm \\ & & & capac & cantmat & & & & & & & & & & & & & \\ & & & & & & 9 & & & & & & & & & & & \end{array}$

Se incluyen en esta dependencia funcional los atributos:

1 y 2: por lo explicado para **a.**

3, 4 y 5: por lo explicado para **b.**

6: porque, dado un proveedor y un material, se conoce en qué almacén se guarda ese material suministrado por ese proveedor.

7 y 8: porque 6 se incluye y entonces vale lo explicado para **c.**

9: porque, dado un proveedor y un material, se conoce también en qué cantidad se guarda en el almacén correspondiente.

(Debe destacarse que no se acostumbra a agregar atributos tales como los señalados como 1, 2, 3, 4, 5, 7 y 8 en esta dependencia funcional **d.**, pues se deduce del hecho de que existen las DF **a.**, **b.** y **c.** y no es preciso señalarlo explícitamente. En este ejemplo se agregan para dejar más claras las DF. En realidad, basta con señalar en esta DF **d.** lo siguiente:

e. $cprov \text{ cmat} \rightarrow calm \text{ cantmat}$).

1.2.2 Representar en una relación todos los atributos

R (cprov, nomprov, dist, cmat, desc, um, precio, calm, diralm, capac, cantmat)

1.2.3 Determinar las llaves candidatas y seleccionar la primaria

Para ello, debe analizarse cada dependencia funcional. Resulta conveniente empezar el análisis por aquellas en que se determinan más atributos en la parte derecha. Comenzaremos entonces por la DF **d.**:

Como puede apreciarse, **cprov cmat** determinan a todos los atributos de la relación. Los subconjuntos de este conjunto de atributos **cprov cmat** son **cprov** (por un lado) y **cmat** (por el otro). Ninguno de los subconjuntos determina, a su vez, a todos los atributos de la relación, como puede comprobarse en las DF **a.** y **b.** Por lo tanto, **cprov cmat** es una llave candidata de la relación **R**.

Es preciso seguir analizando las restantes DF. Veamos la DF **a.**:

a. $cprov \rightarrow nomprov \text{ dist}$

Es necesario agregar atributos en la parte izquierda para lograr determinar a todos los atributos de la relación. Se escogen para agregar, atributos que aparecen, a su vez, en la parte izquierda de las DF, pues son los que ayudan a determinar a los otros.

Probemos agregando el atributo **cmat**. Rápidamente nos daremos cuenta de que estamos en el caso de la DF **d**. que ya analizamos.

Probemos agregando el atributo **calm**:

$cprov \text{ calm} \rightarrow nomprov \text{ dist diralm capac}$

Como se puede ver, no se tienen los atributos del material y sería preciso agregar el atributo **cmat** en el lado izquierdo:

$cprov \text{ calm cmat} \rightarrow nomprov \text{ dist diralm capac desc um precio}$

y, aunque se podría decir que esos tres atributos determinan a todos los de la relación, pues se podría añadir **cantmat** en el lado derecho, ya que

$cprov \text{ cmat} \rightarrow cantmat$

ocurre que **cprov calm cmat** es una **superllave**, pues contiene a **cprov cmat**, que ya sabemos que es una llave candidata.

Análisis similares a éste hay que realizarlos con todas las DF, pero se van a obviar aquí. Baste decir que la única llave candidata que es posible hallar en este caso es **cprov cmat** y, por lo tanto, es la llave primaria:

R (cprov, nomprov, dist, cmat, desc, um, precio, calm, diralm, capac, cantmat)

1.2.4 Aplicar la 2FN

La relación **R** no está en 2FN, pues existen dependencias funcionales incompletas de atributos no llaves respecto a la llave primaria, según se aprecia en las dependencias funcionales **a.** y **b.**, por lo que se crean las relaciones siguientes:

I. **proveedor** (cprov, nomprov, dist)

II. **material** (cmat, desc, um, precio)

y la relación original queda de la siguiente forma:

R (cprov, cmat, calm, diralm, capac, cantmat)

1.2.5 Aplicar la 3FN

Las relaciones anteriores I. y II. están en 3FN, pero la relación restante de la original, no, ya que existe dependencia transitiva de atributos no llaves respecto a la llave primaria pues:

cprov cmat \rightarrow calm y
calm \rightarrow diralm capac

por lo que se separa la siguiente relación:

III. almacén (calm, diralm, capac)

y la relación **R** anterior ahora queda así (le llamaremos **suministro**, pues ya es posible asignarle un nombre adecuado):

suministro (cprov, cmat, calm, cantmat)

Entonces, las 4 tablas resultantes son:

proveedor (cprov, nomprov, dist)
material (cmat, desc, um, precio)
almacén (calm, diralm, capac)
suministro (cprov, cmat, calm, cantmat)

1.2.6 Analizar las relaciones obtenidas

En este caso, no hay relaciones que tengan igual llave ni relaciones que estén constituidas sólo por la llave y cuya información se obtenga ya en otra relación, por lo que el modelo lógico representado por estas cuatro tablas es satisfactorio.

2. NORMALIZACIÓN. EJERCICIOS

En los ejercicios que se presentan a continuación:

- Determine las dependencias funcionales suponiendo que inicialmente todos los datos se encuentran agrupados en una relación.
- Determine las llaves candidatas y señale la llave primaria.
- Diseñe el modelo de datos relacional normalizando las relaciones hasta la 3FN. Deben aparecer todas las relaciones que se vayan obteniendo en cada paso de normalización.

Ejercicio #1

Se desea controlar la actividad de una empresa de Proyectos. Para ello se cuenta con la siguiente información:

De cada trabajador:

- DNI
- nombre
- salario

De cada proyecto:

- código
- fecha de terminación

Además, se conoce las horas en plan de trabajo que cada trabajador dedica a cada proyecto. Se sabe que un trabajador puede laborar en varios proyectos y en un proyecto participan varios trabajadores. Cada proyecto tiene una fecha de

terminación y cada trabajador tiene un nombre y un salario, aunque un mismo salario o nombre puede serlo de varios trabajadores.

Ejercicio #2

En una tienda se desea controlar las ventas y se conoce:

Código de cada mercancía, que la identifica.

Descripción de cada mercancía.

País de procedencia de cada mercancía.

Moneda de cada país.

Precio de venta de cada mercancía de una calidad dada.

Una mercancía tiene varias calidades (1ra., 2da., 3ra., etc.) y una calidad puede serlo de varias mercancías. Una mercancía procede de un país y de un país proceden varias mercancías. Una mercancía tiene una descripción, pero una descripción puede serlo de mercancías con diferentes códigos.

Ejercicio #3

En una empresa editorial de libros se desea controlar las publicaciones que se realizan. Para ello se tiene la siguiente información:

De cada autor se conoce su código, que lo identifica, nombre y dirección.

Para cada libro se conoce su título, género literario y año en que fue escrito. Un libro puede ser escrito por varios autores y un autor puede escribir varios libros. Un libro puede tener varias ediciones y por cada edición de un libro, cada autor recibe un determinado pago. De cada edición de un libro se conoce su fecha y cantidad de ejemplares. (Considere que los títulos de los libros no se repiten).

3. AUTOEVALUACIÓN

En los ejercicios que se presentan a continuación:

- Determine las dependencias funcionales suponiendo que inicialmente todos los datos se encuentran agrupados en una relación.
- Determine las llaves candidatas y señale la llave primaria.
- Diseñe el modelo de datos relacional normalizando las relaciones hasta la 3FN. Deben aparecer todas las relaciones que se vayan obteniendo en cada paso de normalización.

- En una empresa de productos de acero se desea controlar la producción total mensual y anual y la producción mensual por talleres de cada producto, para lo cual cuenta con la siguiente información:

código de cada producto, que lo identifica.

descripción de cada producto.

peso de cada producto.

producción mensual total de cada producto.

producción mensual por taller de cada producto.

número de cada taller, que lo identifica.

Se sabe que un producto puede ser producido por diferentes talleres y en diferentes meses y un taller produce cada mes varios productos, aunque no todos los meses se producen los mismos productos en los mismos talleres. Considere que la información se almacena para un año.

- En una agencia bancaria se desea realizar el control de las cuentas de ahorro personales que en ella existen. Para ello se tiene la siguiente información:

que identifica la cuenta.

de identidad del propietario de la cuenta (sólo un propietario por cuenta).

nombre de cada propietario.

dirección de cada propietario.

Una cuenta tiene sólo un propietario pero cada ciudadano puede ser propietario de varias cuentas. En una fecha dada, para una cuenta determinada, se conoce el saldo que queda en esa cuenta.

- En un Hospital de Lima se desea controlar la atención que se brinda a los pacientes en consultas externas. Un paciente puede requerir atención médica en varias especialidades y en una especialidad se atienden muchos pacientes. De cada paciente se conoce su DNI, su nombre, (que puede repetirse en diferentes pacientes), su edad y su sexo. De cada especialidad se tiene un código, que la identifica, su descripción y la cantidad de médicos de esa especialidad que laboran en consultas externas. Un médico atiende sólo una especialidad, aunque en una especialidad laboran muchos médicos. Un médico atiende a muchos pacientes y un paciente es atendido por varios médicos. De cada médico se sabe su DNI, su nombre, (que puede coincidir con el de otro médico), el año de su graduación y el grado que tiene como médico (especialista de 1er. grado, de 2do. grado, etc.). Se conoce para un paciente, atendido por un médico, el diagnóstico y el tratamiento indicado.
- Se desea diseñar una base de datos para controlar la disponibilidad de materiales de construcción. De cada proveedor de materiales se conoce su código (cprov), que lo identifica, su nombre (nomprov) y el distrito en que radica (dist). De cada material se sabe su código (cmat), que lo identifica, su descripción (desc), la unidad de medida que se aplica al material (um) y el precio por unidad de medida (precio). Para guardar estos materiales hasta su posterior distribución existen diversos almacenes. De cada almacén se conoce su código (calm), que lo identifica, su dirección (diralm) y la capacidad de almacenaje (capac). Un proveedor puede suministrar varios materiales y un material puede ser suministrado por diferentes proveedores. Se sabe que un material suministrado por un proveedor está en un solo almacén y, además, se sabe qué cantidad de un material suministrado por un proveedor se encuentra en el almacén (cantmat). En un almacén se guardan distintos materiales y pueden existir varios almacenes donde se guarde un mismo material. Se sabe también que un proveedor en un almacén sólo guarda un material.

**UNIDAD DE
APRENDIZAJE****4****SEMANA****12**

NORMALIZACIÓN

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos organizan datos no estructurados y los integran al diseño de la base de datos a partir de documentos comerciales propuestos utilizando la teoría de la Normalización hasta la Tercera Forma normal (3FN).

TEMARIO

Ejercicios Propuestos

ACTIVIDADES PROPUESTAS

1. Se realizan ejercicios complejos por parte de los alumnos con la orientación del facilitador

1. NORMALIZACIÓN. EJERCITACIÓN

En los ejercicios que se presentan a continuación:

- a) Determine las dependencias funcionales suponiendo que inicialmente todos los datos se encuentran agrupados en una relación.
- b) Determine las llaves candidatas y señale la llave primaria.
- c) Diseñe el modelo de datos relacional normalizando las relaciones hasta la 3FN. Deben aparecer todas las relaciones que se vayan obteniendo en cada paso de normalización.

Ejercicio #1

En un Hospital de Lima se desea controlar la atención que se brinda a los pacientes en consultas externas. Un paciente puede requerir atención médica en varias especialidades y en una especialidad se atienden muchos pacientes. De cada paciente se conoce su DNI, su nombre, (que puede repetirse en diferentes pacientes), su edad y su sexo. De cada especialidad se tiene el código que la identifica, su descripción y la cantidad de médicos de esa especialidad que laboran en consultas externas. Un médico atiende a muchos pacientes y un paciente es atendido por varios médicos. De cada médico se sabe su DNI, su nombre, (que puede coincidir con el de otro médico), el año de su graduación y el grado que tiene como médico (especialista de 1er. grado, de 2do. grado, etc.). Se conoce para un paciente, atendido por un médico, el diagnóstico y el tratamiento indicado, así como la especialidad en que fue tratado. Se sabe también que, en una especialidad, a un paciente lo atiende un solo médico.

Ejercicio #2

En una biblioteca se desea diseñar una base de datos para el control de los préstamos de libros. De cada libro se conoce el código que lo identifica, su título y la cantidad de páginas que tiene. Un libro se clasifica por una materia y por una materia se clasifican muchos libros. De cada materia se conoce el código que la identifica y su nombre. De cada libro se tienen muchos ejemplares, pero un ejemplar lo es de un solo libro. De cada ejemplar se sabe el código que lo identifica, y su estado de conservación. Un ejemplar se le puede prestar a muchos usuarios y a un usuario se le pueden prestar muchos ejemplares. No obstante, se sabe que a un determinado usuario, en una fecha dada, se le presta un solo ejemplar. También se sabe que un determinado ejemplar, en una fecha dada, se le presta a un solo usuario. De cada usuario se conoce su DNI, su nombre, su dirección y su ocupación.

Ejercicio #3

En un taller de mantenimiento de equipos de computación se desea diseñar una base de datos para el control de los servicios brindados. En el taller laboran varios trabajadores. De cada uno se conoce su DNI, su nombre y categoría laboral. De cada tipo de mantenimiento se sabe el código que lo identifica, su descripción y la frecuencia con que debe realizarse. Un trabajador puede realizar distintos tipos de mantenimiento y un tipo de mantenimiento puede ser ejecutado por diversos trabajadores. El mantenimiento de los equipos de computación se ejecuta por órdenes de servicio. De cada orden de servicio se conoce el número que la identifica, la empresa que la solicita y la fecha en que se solicita. Una orden de servicio puede contener la solicitud de mantenimiento de varios tipos de equipos y puede solicitarse el mantenimiento de un tipo de equipo en diferentes órdenes de

servicio. De cada tipo de equipo se sabe el código que lo identifica, su descripción y fabricante. A cada tipo de equipo se le pueden realizar diferentes tipos de mantenimiento y un mismo tipo de mantenimiento se le puede realizar a diferentes tipos de equipos, pero se sabe que, en una orden de servicio dada, para un tipo de equipo, sólo se solicita ejecutar un único tipo de mantenimiento y también se conoce que un solo trabajador lo ejecuta. Además, en una orden de servicio dada, para un tipo de equipo, se sabe la cantidad (de equipos) que se solicita mantener. En una orden de servicio se pueden solicitar distintos tipos de mantenimiento y un tipo de mantenimiento se puede solicitar en distintas órdenes de servicio, pero está establecido que, en una orden de servicio, un tipo de mantenimiento dado, sólo se solicita para un único tipo de equipo.

2. AUTOEVALUACIÓN

En los formularios que se presentan a continuación:

- a. Normalice cada documento.
 - b. Genere el modelo relacional correspondiente
- 1) El siguiente documento muestra un análisis de precios y requerimientos tanto de materiales, personal, así como de los equipos y herramientas necesarios para realizar una obra.

PRESUPUESTO DE OBRA				
NRO PRESUPUESTO		FECHA		
CLIENTE		ATENDIDO POR		
DIRECCION		DISTRITO		
TLF		CONTACTO		
MATERIALES				
MATERIAL	UNIDAD	CANTIDAD	PRECIO UNITARIO	SUBTOTAL
PERSONAL				
CARGO DEL PERSONAL	UNIDAD	CANTIDAD	PRECIO UNITARIO	SUBTOTAL
EQUIPOS Y HERRAMIENTAS				
EQUIPO/HERRAMIENTA	UNIDAD	CANTIDAD	PRECIO UNITARIO	SUBTOTAL
UNIDADES VALIDAS : U1: UNI U2:ROLL U3: GLB U4: GLN U5: DIA U6: HR				
TOTAL PRESUPUESTO				

- 2) El siguiente documento representa la labor realiza por los empleados de Confecciones “HILOS DE ORO”. Permite controlar el trabajo realizado al registrar la cantidad de los tipos de prendas producidas por día, por ejemplo: el registro de las prendas programadas, prendas confeccionadas o prendas dañadas por día. Asimismo, se registra el tipo de hora (ingreso o salida) y se especifica la hora por día.

Por último, se registran las ocurrencias que se han suscitado durante la semana especificada.

TARJETA DE CONTROL								
CODIGO DEL EMPLEADO :				SUPERVISOR :				
EMPLEADO :								
SEMANA TERMINADA :								
CONTROL DE TRABAJO								
TIPO DE PRENDA	DOM	LUNES	MART	MIÉRC	JUEVES	VIER	SÁB	
Total prendas a pagar								
CONTROL DE TIEMPO								
ITEM	DESCRIPCION DEL TIPO HORA	DOM	LUNES	MART	MIÉRC	JUEVES	VIER	SÁB
Total horas a pagar								
OCURRENCIAS DE LA SEMANA								
item	OCURRENCIA					DIA	HORA	

- 3) El siguiente documento registra a un vehículo para su mantenimiento y/o reparación. Al ingresar un vehículo se hace un inventario de los accesorios con los que cuenta, luego el mecánico que atiende registra el diagnóstico y/o reparación que realiza. Por último, en el presente documento se registran las ocurrencias indicando el turno y fecha.

MANTENIMIENTO DE VEHÍCULOS

NRO. MANTENIMIENTO	FECHA REGISTRO
EMPLEADO	PLACA
MARCA	MODELO
HORA DE ENTRADA	HORA SALIDA
FECHA SALIDA	

INVENTARIO

CODIGO ACCESORIO	ACCESORIO	ESTADO
------------------	-----------	--------

DIAGNOSTICO/REPARACION

CODIGO	DIAGNOSTICO/REPARACION	MECANICO
--------	------------------------	----------

* Anotar todas las ocurrencias

OCURRENCIAS

ITEM	OCURRENCIA	TURNO	FECHA
------	------------	-------	-------

TURNO : M: MAÑANA T : TARDE N: NOCHE



OBTENCIÓN DEL MODELO LÓGICO-GLOBAL DE LOS DATOS A PARTIR DEL DER

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos construyen el modelo de datos a partir del DER obtenido en el caso propuesto y del MER resultante de organizar datos no estructurados, explicando la metodología de diseño de base de datos utilizada.

TEMARIO

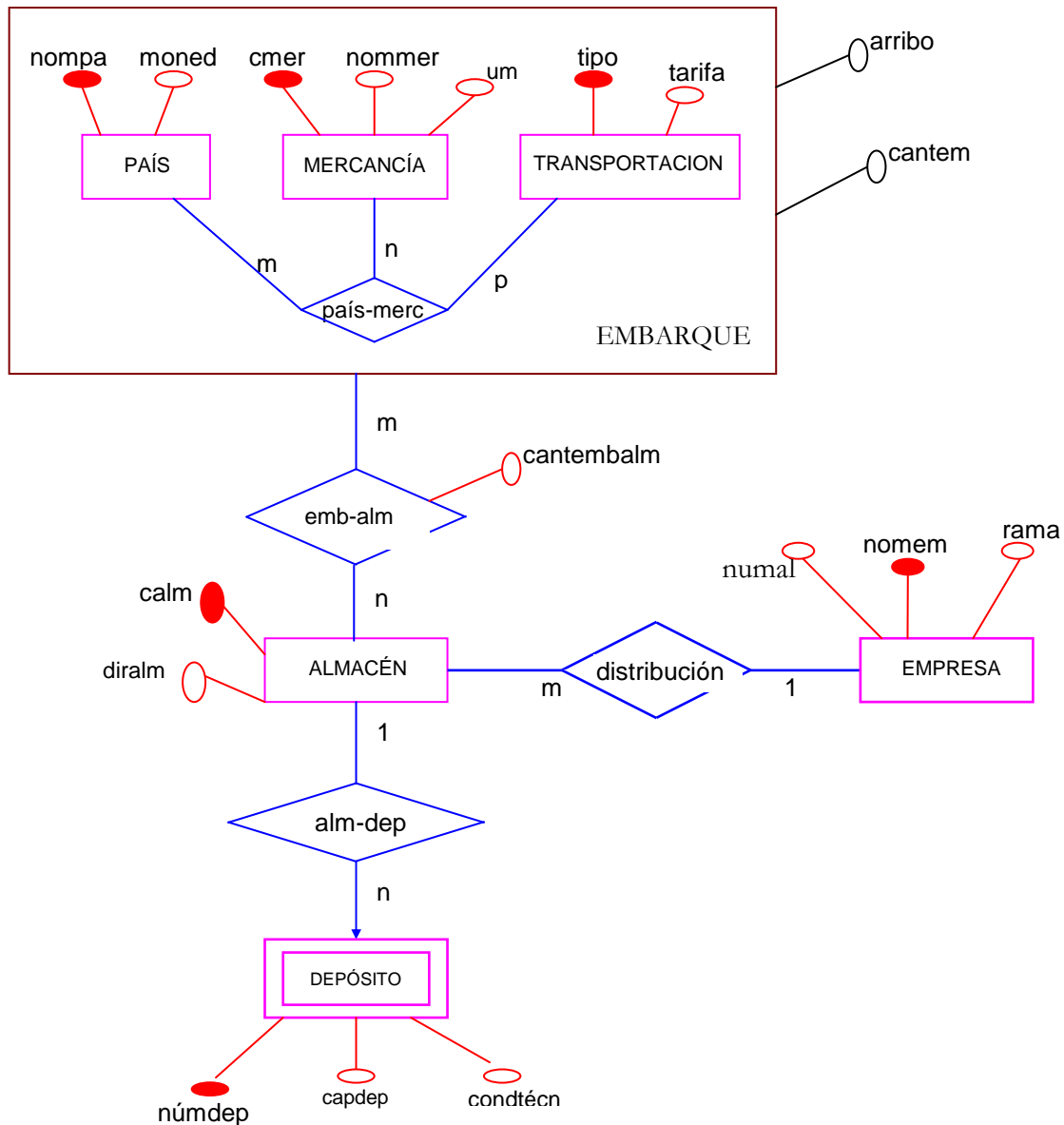
Obtención de Modelo Relacional a partir del DER

ACTIVIDADES PROPUESTAS

1. Explicaciones por parte del facilitador combinadas con ejemplos y preguntas de comprobación dirigidas a los alumnos.
2. Se realizan ejercicios por parte de los alumnos con la orientación del facilitador.

1. OBTENCIÓN DEL MODELO RELACIONAL A PARTIR DEL DER

Para obtener el modelo lógico global de los datos, según el enfoque relacional a partir del DER, se sigue un procedimiento que iremos describiendo paso a paso y aplicándolo, así mismo, al ejemplo siguiente:



En el DER anterior se representa el del movimiento mercantil de un organismo. En el organismo existen mercancías de las que se conoce su código, nombre y unidad de medida. Las mercancías proceden de diferentes países de los que se sabe nombre y tipo de moneda. Para la transportación de las mercancías existen diversas formas, cada una de las cuales se caracteriza por su tipo (barco, avión, tren, etc.) y tarifa. Para cada mercancía de diferentes países existen diferentes formas de transportación; para cada país existen diferentes mercancías que son transportadas en diferentes formas de transportación; y una forma de transportación puede serlo de diferentes mercancías de diferentes países. Una mercancía procedente de un país transportada de una forma dada constituye un embarque y para éste se conoce su fecha de arribo y cantidad.

Un embarque se distribuye entre diferentes almacenes y en un almacén se tienen diferentes embarques, cada uno en cierta cantidad. De cada almacén se tiene su código y dirección. Un almacén envía sus productos a una sola empresa y cada empresa recibe productos de diferentes almacenes. Una empresa se caracteriza por su número, nombre y rama económica.

Cada almacén tiene distintos depósitos subordinados. De cada depósito se conoce su número (que se puede repetir en diferentes almacenes), capacidad y condiciones técnicas.

1.1 Representar las entidades regulares

Representar cada entidad regular en una tabla relacional.

- a. país (nompa, moneda)
- b. mercancía (cmerc, nommerc, um)
- c. transportación (tipo, tarifa)
- d. almacén (calm, diralm)
- e. empresa (nútemp, nomemp, rama)

1.2 Representar las entidades agregadas

Representar en una tabla relacional cada entidad agregada con sus correspondientes atributos (entre ellos un identificador si fue definido) y, las llaves de las entidades que forman la agregación. La llave primaria de esta tabla es la llave de la relación que da origen a la agregación, si no hay un identificador definido; pero si lo hay, entonces la llave será dicho identificador

- f. embarque (nompa, cmerc, tipo, arribo, cantemb)

Nótese que la llave estaría formada por las llaves de las 3 entidades regulares que intervienen en la agregación.

Pero podía haberse definido un identificador para la entidad embarque. (Supóngase añadido en el DER un atributo de la entidad embarque, que sería su llave: **idemb**). Entonces se añadiría como atributo llave en la agregación y los 3 atributos nompa, cmerc y tipo permanecerían en la relación, pero no como llave:

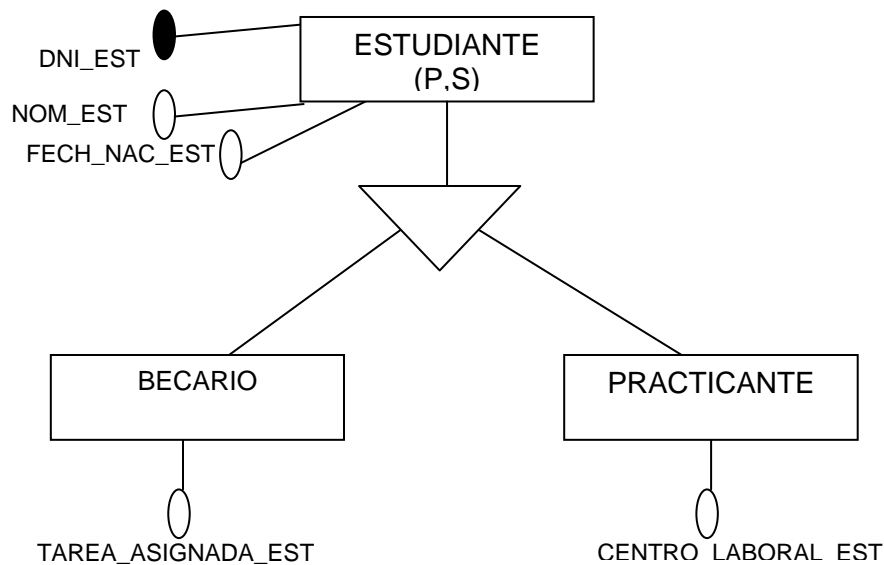
- f. embarque (**idemb**, nompa, cmerc, tipo, arribo, cantemb)

1.3 Representar las entidades generalizadas y especializadas

Representar cada entidad generalizada en una tabla que contendrá sus atributos (sólo los de la generalizada) y, entre ellos, la llave.

Representar cada entidad especializada en una tabla que contendrá, como llave primaria, la llave de la generalización y los atributos propios sólo de la especialización.

Como este ejemplo no incluye ninguna generalización /especialización, analicemos cómo se llevaría a tablas el siguiente DER, en que se tiene la entidad generalizada estudiante y los casos especiales, becario y practicante



estudiante (**DNI_EST**, NOM_EST, FECH_NAC_EST)
 becario (**DNI_EST**, TAREA_ASIGNADA_EST)
 practicante (**DNI_EST**, CENTRO_LABORAL_EST)

1.4 Representar las relaciones de m:m

Representar en una tabla relacional cada relación de m:m, incluyendo las llaves de las entidades relacionadas y los atributos de la relación, si los hubiese. La llave primaria de esta tabla estará formada por las llaves de las entidades que participan en la relación.

g. emb-alm (nompa, cmerc, tipo, calm, cantembalm)

Esta relación sería de esta forma sin considerar la existencia del idemb. En el caso de que éste estuviera definido quedaría:

g'. emb-alm (idemb, calm, cantembalm)

ya que entonces la llave de la entidad embarque sería idemb.

1.5 Representar las relaciones de 1:m

Para cada relación de 1:m, añadir la llave de la entidad del extremo "1" como un nuevo atributo no llave a la entidad del extremo "m" y los atributos de la relación, si existen.

En nuestro ejemplo, se da la relación

amacén ↔ empresa

por lo que la llave de la entidad empresa (númemp) debe añadirse como atributo no llave en la tabla que representa la entidad almacén. Esto hace que

la relación **d.**, obtenida en el paso 1.1 quede modificada de la siguiente manera:

d. almacén (calm, diralm, númemp)

Al agregarse el atributo númemp en la relación almacén se sabe entonces a qué empresa abastece el almacén. Está claro que muchas ocurrencias de almacén pueden tener el mismo valor en el atributo númemp, pues una empresa recibe mercancías de varios almacenes, en general.

1.6 Representar las entidades débiles

Representar cada entidad débil en una tabla relacional que contendrá la llave de la entidad determinante y los atributos de la entidad débil. La llave de la entidad débil estará formada por la llave de la entidad determinante y uno o varios atributos de la débil que permitan distinguir cada ocurrencia.

h. depósito (calm, númdep, capdep, condtecn)

Estas son las tablas relacionales que se derivan del DER del ejemplo. Ahora es preciso analizar cada una de las relaciones para asegurarse que están normalizadas hasta la 3FN.

En este caso, puede verse que todas las relaciones cumplen con la 3FN, por lo que no es preciso hacer nada más.

2. EJERCITACIÓN

Obtenga el modelo lógico-global de los datos a partir del DER que representa cada uno de los siguientes s (los DER correspondientes a estos ejercicios fueron realizados en la sesión 4 o indicados en la Autoevaluación correspondiente):

Ejercicio #1

En una empresa de la industria mecánica se fabrican distintos tipos de piezas. Para cada una se conoce su código, que la identifica, descripción y peso unitario. Sobre cada pieza se realizan distintas operaciones (corte, fresado, etc.) acerca de las cuales se conoce su código y nombre. En su proceso de fabricación, una pieza pasa por diferentes equipos que pueden realizar diferentes operaciones, una operación puede ser realizada en diferentes equipos a diferentes piezas y en un equipo se realizan diferentes operaciones sobre diferentes piezas. De cada equipo se conoce su nombre, modelo y país de procedencia. Una operación realizada sobre una pieza en un equipo dado constituye una norma que se identifica por un código y tiene asociados una descripción y un tiempo de ejecución.

En la empresa trabajan obreros de dos tipos distintos: directos y auxiliares. Un obrero directo está vinculado con el cumplimiento de diferentes normas y una norma puede ser aplicada a distintos obreros directos. Un obrero directo cumple una norma dada en un determinado por ciento. Un obrero auxiliar realiza el mantenimiento de diferentes equipos y un equipo dado siempre es atendido por un

mismo obrero auxiliar. Para cada equipo atendido por un obrero auxiliar dado se conoce el tipo de mantenimiento efectuado y la última fecha en que se realizó. Todos los obreros de la fábrica están caracterizados por un número, su nombre, calificación laboral y categoría ocupacional (directo o auxiliar).

Ejercicio #2

Se desea controlar la actividad de reparación de radios en los talleres dedicados a esta tarea. Cada usuario puede llevar diferentes radios para ser reparados, aunque un radio sólo pertenece a un usuario. Para cada radio se conoce el número de la solicitud de reparación que lo identifica, tipo de radio, descripción y tipo de rotura (**sencillas** o **complejas**). De cada usuario se conoce su DNI, nombre y dirección. Cada técnico de reparaciones pertenece a un taller y tiene un código que lo identifica dentro de su taller, por lo que cada código puede repetirse para diferentes talleres, y tiene, además, un nombre, una categoría (**A, B, C**) y un salario. En un taller laboran muchos técnicos de reparaciones. De cada taller se conoce el código que lo identifica, nombre y dirección. Los radios con roturas **complejas** sólo podrán ser atendidos por técnicos de categoría **A**, mientras que los radios con roturas **sencillas** pueden ser atendidos por cualquier técnico. En cualquier caso, un técnico puede reparar muchos radios, pero un radio es reparado por un solo técnico.

3.- AUTOEVALUACIÓN

Obtenga el modelo lógico-global de los datos a partir del DER que representa cada uno de los siguientes s (los DER correspondientes a estos ejercicios fueron realizados en la sesión 4 o indicados en la Autoevaluación correspondiente):








- 1) En un organismo se reciben productos que son importados de diferentes países. Es necesario controlar las cantidades que se importan de cada país y el valor de las importaciones. Las propiedades de los productos son número, nombre, unidad de medida, peso y precio unitario. Las propiedades de los países son número, nombre, zona geográfica y tipo de moneda. Existe relación comercial con varios países de donde se importan varios productos en ciertas cantidades.
- 2) En una empresa de taxis de Lima existe un conjunto de taxis que brindan servicio a los hospitales de dicha ciudad. En la empresa trabajan choferes, cada uno de los cuales se caracteriza por su DNI, nombre y años de servicio. En la empresa hay numerosos taxis, de los que se conoce la placa, el número del motor, la marca y el modelo. Un taxi puede ser conducido por diferentes choferes (en distintos momentos), pero un chofer siempre conduce el mismo taxi. Se conoce la cantidad de kilómetros totales recorridos por un chofer en su correspondiente taxi. La empresa brinda servicios a varios hospitales. De cada hospital se conoce su nombre, tipo y dirección. Un chofer le brinda servicios (realiza viajes) a distintos hospitales y a un hospital le brindan servicios distintos choferes. Se sabe la cantidad de viajes realizados por cada chofer a cada hospital.
- 3) El Instituto de Aeronáutica del Perú cuenta con información sobre las diferentes compañías de aviación que existen en el mundo. De cada compañía se conoce su nombre (Cubana, Copa, AeroContinente, etc.), su volumen anual de ventas y un código que la identifica. Cada compañía puede estar representada en diferentes países y en un país pueden estar representadas

diversas compañías. De cada país se conoce su código, nombre, idioma y área de moneda.

También se sabe que por vía aérea se realizan diferentes viajes. Cada viaje posee un código, un lugar de origen, un destino y una cantidad de kilómetros a recorrer. Además, se conoce que existen distintos tipos de aviones. Cada tipo de avión se identifica por un nombre (IL-62, DC-10, etc.) y posee un consumo de gasolina por kilómetro y una cierta cantidad de asientos. Un mismo viaje puede ser realizado por distintos tipos de aviones y un tipo de avión puede ser utilizado en diversos viajes. Un viaje realizado por un tipo de avión constituye un vuelo y para cada vuelo se conoce su tiempo de duración.

Una compañía de aviación realiza muchos vuelos y un mismo vuelo puede ser cubierto por diversas compañías. Para cada vuelo que ofrece una compañía se conoce el precio del pasaje.

Resumen

-  Representar cada entidad regular en una tabla relacional.
-  Representar en una tabla relacional cada entidad agregada con sus correspondientes atributos (entre ellos un identificador si fue definido) y, las llaves de las entidades que forman la agregación. La llave primaria es la llave de la relación que da origen a la agregación, si no hay un identificador definido; pero si lo hay, entonces será dicho identificador.
-  Representar cada entidad generalizada en una tabla que contendrá sus atributos (sólo los de la generalizada) y, entre ellos, la llave. Representar cada entidad especializada en una tabla que contendrá, como llave primaria, la llave de la generalización y los atributos propios sólo de la especialización.
-  Representar en una tabla relacional cada relación de m:m, incluyendo las llaves de las entidades relacionadas y los atributos de la relación, si los hubiese. La llave primaria de esta tabla estará formada por las llaves de las entidades que participan en la relación.
-  Para cada relación de 1:m, añadir la llave de la entidad del extremo "1" como un nuevo atributo no llave a la entidad del extremo "m" y los atributos de la relación, si existen.
-  Representar cada entidad débil en una tabla relacional que contendrá la llave de la entidad determinante y los atributos de la entidad débil. La llave de la entidad débil estará formada por la llave de la entidad determinante y uno o varios atributos de la débil que permitan distinguir cada ocurrencia.
-  Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

 <http://ict.udlap.mx/people/carlos/is341/bases03.html>

En esta página web encontrará conceptos adicionales y ejemplos de cómo se transforma paso a paso un Diagrama Entidad Relación a un Modelo Relacional.

**UNIDAD DE
APRENDIZAJE****5****SEMANA****14**

OBTENCIÓN DEL MODELO LÓGICO-GLOBAL DE LOS DATOS A PARTIR DEL DER

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos construyen el modelo de datos a partir del DER obtenido en el caso propuesto y del MER resultante de organizar datos no estructurados, explicando la metodología de diseño de base de datos utilizada.

TEMARIO

Obtención del Modelo Relacional a partir del DER

ACTIVIDADES PROPUESTAS

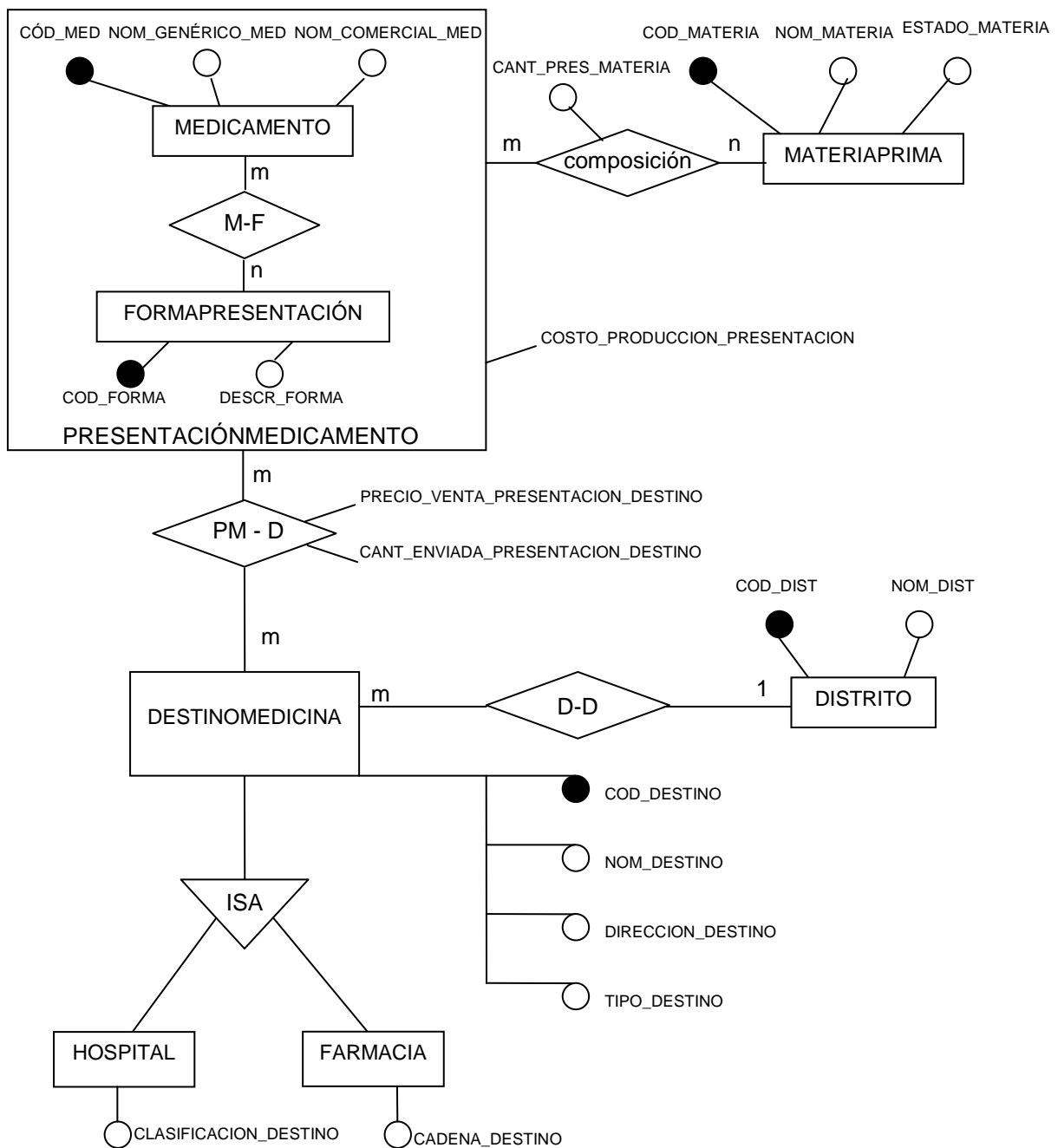
1. Se realizan ejercicios por parte de los alumnos con la orientación del facilitador.

1. OBTENCIÓN DEL MODELO RELACIONAL A PARTIR DEL DER

EJERCITACIÓN

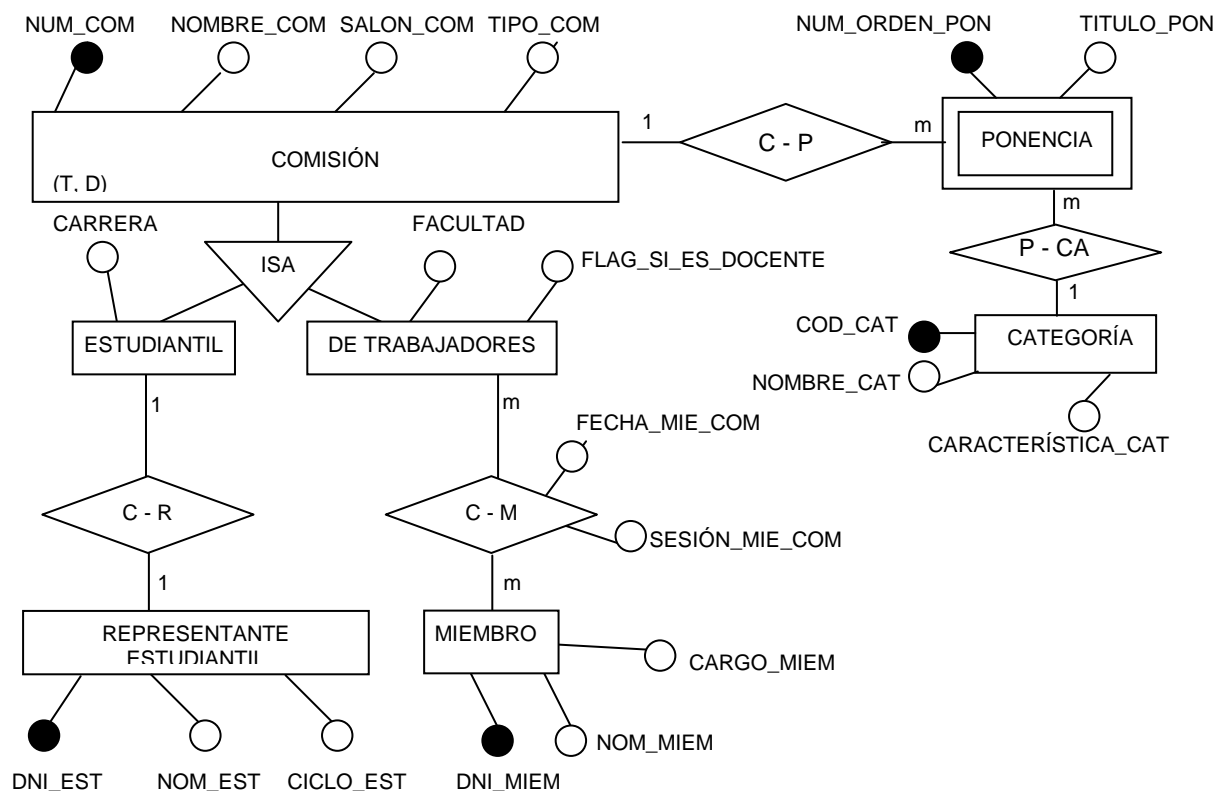
Ejercicio #1

Diseñe el modelo de datos relacional a partir del siguiente Diagrama Entidad-Relación que representa el de la producción de medicamentos en distintas formas de presentación (cápsulas, ampollas, etc.) y su distribución a los distintos destinos que tienen estos medicamentos. El atributo tipodest indica si el destino es una farmacia o un hospital.



Ejercicio #2

Diseñe el modelo de datos relacional a partir del siguiente Diagrama Entidad-Relación que representa la organización de un evento científico en una universidad, en el que participan estudiantes y trabajadores (que pueden ser docentes y no docentes).



2. AUTOEVALUACIÓN







a) Obtenga el modelo lógico-global de los datos a partir del DER que representa cada uno de los siguientes s (los DER correspondientes a estos ejercicios fueron realizados en la sesión 4 o indicados en la Autoevaluación correspondiente):

- 1.1.1. En un taller de mantenimiento de equipos de computación se desea controlar los servicios brindados. En el taller laboran varios trabajadores. De cada uno se conoce su DNI, su nombre, categoría laboral y tipo de mantenimiento (**Tipo1**, **Tipo2**) que se encarga de realizar a los equipos. El mantenimiento de los equipos de computación se ejecuta por órdenes de servicio que son presentadas por las empresas usuarias de este taller. Una empresa usuaria puede

presentar muchas órdenes de servicio y cada orden de servicio es presentada por una empresa usuaria. De cada orden de servicio se conoce el número que la identifica dentro de la empresa que la presenta, por lo que puede repetirse ese número para diferentes empresas, y la fecha en que se presenta. De cada empresa usuaria se sabe el código que la identifica, su nombre y la rama económica a la que pertenece. Una orden de servicio puede contener la solicitud de mantenimiento de varios equipos y puede solicitarse el mantenimiento de un equipo en diferentes órdenes de servicio. De cada equipo se sabe el código que lo identifica, su descripción, su marca y su tipo (sólo se toman en cuenta dos tipos: **Computadoras** y **Otros**). El mantenimiento de las **Computadoras** sólo puede ser realizado por los trabajadores que realizan mantenimientos **Tipo1** y el mantenimiento de los **Otros** equipos sólo puede realizarse por trabajadores encargados de hacer los mantenimientos **Tipo2**. En ambos casos, el mantenimiento de un equipo es realizado por un solo trabajador, aunque un trabajador puede realizar el mantenimiento de varios equipos.

- 1.1.2. Se desea controlar la programación de los itinerarios a recorrer por las locomotoras en una estación de ferrocarriles. En la estación existen varias locomotoras, de las que se conoce el código que la identifica y años que lleva prestando servicios. Existen diversas marcas de locomotora; una locomotora es de una marca y una marca puede serlo de varias locomotoras. De cada marca se tiene el código que la identifica, el consumo de combustible promedio por hora y la velocidad máxima que puede alcanzar. Una locomotora realiza diversos itinerarios, pero un itinerario lo realiza una sola locomotora. De cada itinerario se conoce el número que lo identifica para una locomotora dada, pero que puede repetirse para distintas locomotoras (es decir, la locomotora A puede tener un itinerario 2 y la locomotora B puede tener también un itinerario 2), los kilómetros a recorrer, el destino final y el tipo de itinerario (**Diurno** o **Nocturno**; un itinerario es diurno o es nocturno y todos los itinerarios son de uno de estos dos tipos). Las locomotoras son conducidas por maquinistas. Un maquinista (cualquiera) puede conducir varias locomotoras y una locomotora puede ser conducida por muchos maquinistas y se conoce la cantidad de kilómetros recorrido por un maquinista en una locomotora. De cada maquinista se conoce su DNI, su nombre y si tiene más de 10 años de antigüedad como maquinista o no. Los itinerarios nocturnos sólo los pueden realizar los maquinistas con más de 10 años de antigüedad y los maquinistas con más de 10 años de antigüedad sólo realizan itinerarios nocturnos. Los itinerarios diurnos sólo los realizan los maquinistas con 10 o menos años de antigüedad y los maquinistas con 10 o menos años de antigüedad sólo realizan itinerarios diurnos. En ambos casos ocurre que un maquinista realiza muchos itinerarios y que un itinerario es realizado por muchos maquinistas.

Resumen

-  Es necesario usar la variedad formal en situaciones que así lo establezcan como reuniones de trabajo, entrevistas con autoridades o presentaciones orales, siempre dentro de un contexto riguroso.
-  Al representar una entidad agregada, ya se está representando la relación que la origina, por lo que ésta, aún en el caso en que sea de m:n, no se representa aparte.
-  En el caso de que una entidad A sea débil de otra entidad débil B, debe tenerse en cuenta esto para señalar adecuadamente la llave primaria. La llave de la entidad débil A es la llave de la entidad débil B (que contiene, a su vez, la llave de la entidad que le dio origen) más él o los atributos correspondientes de la entidad A.
-  Si una generalización es también una entidad débil, su llave, y la de sus especializaciones, está dada por su condición de débil, es decir, debe incluir la llave de la entidad que le dio origen.
-  No es preciso seguir el orden en que han descrito los pasos para la obtención del modelo relacional a partir del DER en la sesión 11, sino que puede hacerse según el criterio del diseñador, siempre y cuando se apliquen todos los pasos, según corresponda, adecuadamente.
-  Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

 <http://ict.udlap.mx/people/carlos/is341/bases03.html>

En esta página web encontrará conceptos adicionales y ejemplos de cómo se transforma paso a paso un Diagrama Entidad Relación a un Modelo Relacional.



METODOLOGÍA PARA EL DISEÑO DE BASES DE DATOS

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos construyen el modelo de datos a partir del DER obtenido en el caso propuesto y del MER resultante de organizar datos no estructurados, explicando la metodología de diseño de base de datos utilizada.

TEMARIO

Metodología para el diseño de base de datos

ACTIVIDADES PROPUESTAS

1. Explicaciones por parte del facilitador combinadas con ejemplos.
2. Se efectúan preguntas de comprobación dirigidas a los alumnos sobre los temas tratados.

1. METODOLOGÍA PARA EL DISEÑO DE BASES DE DATOS

La metodología para el diseño de la base de datos, consta de los siguientes pasos:

- Determinación de entidades y atributos
- Normalización de entidades
- Determinación de relaciones (DER)
- Obtención del modelo lógico global de los datos
- Diseño físico de la base de datos

Cuando se va a realizar el diseño de la base de datos para un sistema dado, es necesario determinar los datos que se deben tomar en cuenta y las dependencias funcionales existentes entre ellos.

Rigurosamente, esto se obtiene luego de realizada la etapa de análisis del sistema y partiendo de lo obtenido en ésta, pero trataremos de dar indicaciones que permitan lograr un buen diseño de la base de datos a partir de lo que conocemos.

La situación a partir de la cual se discutirá la metodología para diseñar la base de datos es la siguiente:

En un hospital Clínico Quirúrgico se desea automatizar el control de la actividad quirúrgica. En el hospital se tienen varios quirófanos y se realizan intervenciones quirúrgicas en diferentes especialidades. Las intervenciones quirúrgicas pueden ser electivas (planificadas) o urgentes (no planificadas). Se tiene la planificación de las operaciones quirúrgicas a realizar cada día. Las operaciones planificadas pueden ser suspendidas por distintas causas. Para cada operación realizada (haya sido planificada o no), se confecciona un informe operatorio. En cada operación realizada se emplean materiales cuyo gasto se controla.

Las salidas o reportes que se quieren obtener con el sistema automatizado son las siguientes:

PROGRAMACIÓN QUIRÚRGICA					Para el día: ___ de _____ del 20__ .			
Paciente	Hist Clín.	Edad	Sexo	Diagnóstico Pre-Oper.	Intervenc. Indicada	Hora	Salón	Cirujano

OPERACIONES SUSPENDIDAS		En el día: de _____ del 20 _____.	
Paciente	Historia Clínica	Edad	Causa

INFORME OPERATORIO		Fecha de Operación:	
		Hora Comienzo	Hora Término
Paciente	Historia Clínica	Edad	
Diagnóstico Clínico: _____ Operación Realizada: _____ Diagnóstico Operatorio: _____ _____			
Situación Final del Paciente: Satisfactoria <input type="checkbox"/> No Satisfactoria <input type="checkbox"/> Fallecido <input type="checkbox"/>			
Especialidad	Tipo de Operación Realizada Electiva <input type="checkbox"/> Urgente <input type="checkbox"/>		
Cirujano: _____			

RESUMEN DE ACTIVIDADES QUIRÚRGICAS	Mes:
Cantidad de Operaciones Electivas:	
Cantidad de Operaciones Urgentes:	
Especialidad	Cantidad de Operaciones

GASTO DE MATERIALES DE OPERACIÓN						En el día: de de 19 _____		
Paciente	Hist. Clín.	Edad	Hora Com.	Patología	Cód. Mat.	Descr. Mater.	Cantidad	Costo

Para comenzar a estudiar la metodología de diseño de la base de datos, partiremos de una premisa: lo que se obtiene con un sistema automatizado son sus salidas, es decir, el resultado que se desea obtener, lo que llega al usuario, son justamente las salidas o reportes del sistema y para ello se desarrolla; por lo tanto, lo que no sea necesario para obtener las salidas del sistema, no tiene por qué estar almacenado en la base de datos. En fin, lo que debe almacenarse en la base de datos es lo que es imprescindible para obtener las salidas.

Es por esta razón que, para el diseño de la base de datos, se parte del análisis de las salidas que se desea obtener con el sistema.

1.1 Determinación de entidades y atributos

Para cada salida:

- Consultar su formato
- Determinar datos que se calculen u obtengan a partir de otros e ir sustituyéndolos hasta llegar a los primarios
- Determinar la existencia de archivos con información normativa y/o de consulta
- Cada salida y cada archivo, tomarlos como entidades y relacionar (listar) sus atributos

En el ejemplo, representando cada una de las salidas por una relación, se tienen las siguientes relaciones o entidades iniciales:

PQ(Fecha, NomPac, HC, Ed, Sexo, DiagPre, Interv, Hora, Salón, Cirujano)

OS(Fecha, NomPac, HC, Ed, Causa)

IO(Fecha, HoraCom, HoraTerm, NomPac, HC, Ed, DiagClín, Operac, DiagOper, SitPac, Espec, TipoOpReal, CirReal)

RAQ (Mes, Elect, Urg, Espec, Cant)

GMO (Fecha, NomPac, HC, Ed, HoraCom, Patología, CódMat, DescMat, Cant, Costo)

Analizando cada dato se ve que:

En **PQ**:

Hora se calcula según el tiempo promedio de demora para cada operación, por lo que es necesario un archivo normativo:

⇒ **TPO** (Operación, Tiempo Prom)

Salón se decide teniendo en cuenta la distribución de los salones, que en este caso, es por día de la semana:

⇒ **DS** (Salón, DíaSem, Espec)

En **OS**:

Todos son datos primarios

En **IO**:

Todos son datos primarios

En **RAQ**:

Elect, *Urg* y *Cant* son datos que se calculan a partir de contar en **IO**, por lo que la relación **RAQ** se puede eliminar completamente para el análisis, pues al ser calculables todos los datos contenidos en ella, no aporta información nueva a almacenar en la base de datos.

En **GMO**:

Costo = cant x precio, por lo que se sustituye costo por precio (ya cant está en el modelo)

Esto implica la necesidad de que exista un archivo de consulta MATERIAL (puede salir en entrevistas a usuarios)

1.2 Normalización de entidades

Para cada relación obtenida en el paso 1:

- Determinar las DF
- Normalizar cada entidad
- Fusionar, de ser lógico, las entidades normalizadas que tengan la misma llave

Entonces, en el ejemplo, tendríamos:

PQ (Fecha, NomPac, HC, Ed, Sexo, DiagPre, Interv, Hora, Salón, Cirujano)

Normalizando:

Paciente (HC, NomPac, Ed, Sexo)

PQ (Fecha, HC, DiagPre, Interv, Hora, Salón, Cirujano)

OS (Fecha, NomPac, HC, Ed, Causa)

Normalizando:

Paciente1 (HC, NomPac, Ed)

OS (Fecha, HC, Causa)

IO (Fecha, HoraCom, HoraTerm, NomPac, HC, Ed, DiagClín, Operac, DiagOper, SitPac, Espec, TipoOpReal, CirReal)

Normalizando:

2FN:

Paciente2 (HC, NomPac, Ed)

PlanOperac (Fecha, HC, DiagClín, Operac, Espec)

IO (Fecha, HoraCom, HC, HoraTerm, DiagOper, SitPac, TipoOpReal, CirReal, OperReal)

3FN:

Paciente2 (HC, NomPac, Ed)

PlanOperac (Fecha, HC, DiagClín, Operac)

ClasifOperac (Operac, Espec)

IO (Fecha, HoraCom, HC, HoraTerm, DiagOper, SitPac, TipoOpReal, CirReal, OperReal)

RAQ (ya está eliminada, como se vio antes, para este análisis)

GMO (Fecha, NomPac, HC, Ed, HoraCom, Patología, CódMat, DescMat, Cant, Precio) (Precio en lugar de Costo)

Normalizando:

Paciente3 (HC, NomPac, Ed)

Diagnóstico (Fecha, HC, Patología)

Material (CódMat, DescMat, Precio)

GMO (Fecha, HC, HoraCom, CódMat, Cant)

Además, se tienen los dos archivos normativos o de consulta:

TPO (Operación, TiempoProm)

DS (Salón, DíaSem, Espec)

Fusionando las entidades normalizadas con igual llave, siempre que sea lógico, se tiene:

De **Paciente**, **Paciente1**, **Paciente2** y **Paciente3**:

Paciente (HC, NomPac, Ed, Sexo)

De **PQ**, **OS**, **PlanOperac** y **Diagnóstico**:

PlanOperac está contenida en **PQ** y **Diagnóstico** está contenida en **PQ** también. Tener en cuenta que *DiagPre* = *DiagClín* = *Patología* son alias diferentes para indicar un mismo atributo, que tiene el mismo dominio.

Pero **OS**, a pesar de tener igual llave, está representando otra entidad, la **Operación Suspendida**; semánticamente, no representa lo mismo que una **Operación Planificada** y, por lo tanto, no tiene sentido unirlos. Es más, si se hiciera, todas las operaciones tendrían el atributo Causa, y en la mayoría de las ocurrencias, este atributo estaría vacío, pues sólo son algunas operaciones las que se suspenden. Por lo tanto, de **PQ**, **OS**, **PlanOperac** y **Diagnóstico** quedarían:

PQ (Fecha, HC, DiagPre, Interv, Hora, Salón, Cirujano)

OS (Fecha, HC, Causa)

De **ClasifOperac** y **TPO**:

Intervención (Operación, TiempoProm, Espec)

De **DS**:

DS (Salón, DíaSem, Espec)

De **IO**:

IO (Fecha, HoraCom, HC, HoraTerm, DiagOper, SitPac, TipoOpReal,
CirReal, OperReal)

De **Material**:

Material (CódMat, DescMat, Precio)

De **GMO**:

GMO (Fecha, HC, HoraCom, CódMat, Cant)

Por lo que, en este paso, finalmente, quedarían las siguientes relaciones:

Paciente (HC, NomPac, Ed, Sexo)

PQ (Fecha, HC, DiagPre, Interv, Hora, Salón, Cirujano)

OS (Fecha, HC, Causa)

Intervención (Operación, TiempoProm, Espec)

DS (Salón, DíaSem, Espec)

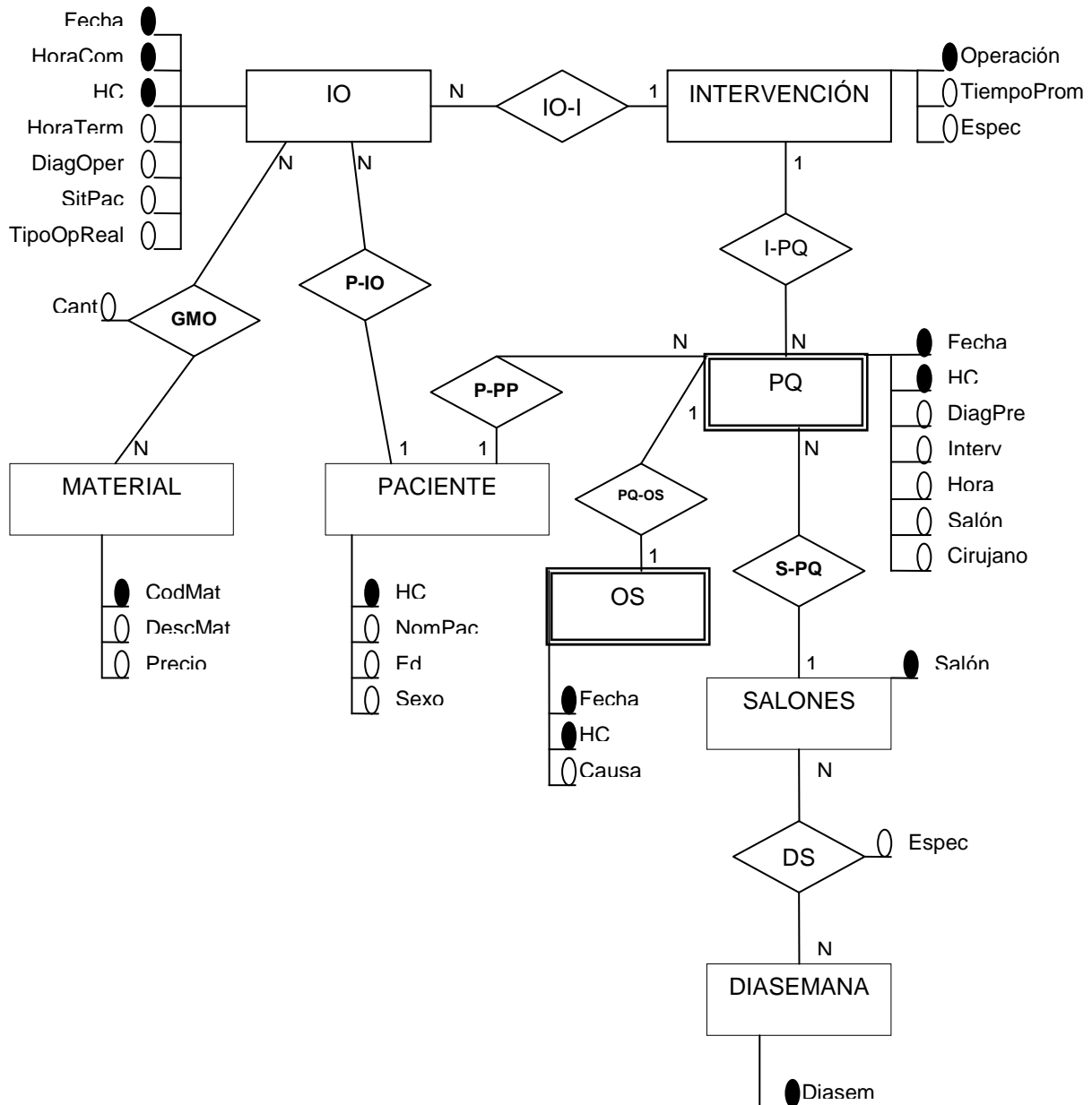
IO (Fecha, HoraCom, HC, HoraTerm, DiagOper, SitPac, TipoOpReal,
CirReal, OperReal)

Material (CódMat, DescMat, Precio)

GMO (Fecha, HC, HoraCom, CódMat, Cant)

1.3 Determinación de relaciones (DER)

- Analizar cada entidad con las restantes para determinar si existe relación, determinando tipo de proyección
- Si hay alguna entidad cuya llave es compuesta, de modo que resulta que esa llave es la combinación de las llaves de otras entidades, es porque representa una relación entre ellas de m: n
- Si hay alguna entidad cuya llave es compuesta y un subconjunto de los atributos que la forma no es llave de otra entidad, es porque es una entidad débil
- Si en una entidad aparece, como atributo no llave, la llave de otra entidad es porque existe entre ellas una relación de m:1
- Analizar entidades agregadas, entidades generalizadas/especializadas y sus relaciones con otras
- Ir confeccionando el DER
- Fusionar, de ser lógico, las entidades normalizadas que tengan la misma llave



En este paso se puede comprobar si lo que se ha realizado es correcto o, si surge alguna contradicción, se puede completar el modelo con nuevos elementos que se hayan podido obtener en intercambios con los usuarios, ya que con el DER se tiene un modelo menos abstracto, más cercano al . El analista puede proponer nuevas posibilidades a los usuarios y agregarlas en caso de acordarse esto, etc.

1.4 Obtención del modelo lógico global de los datos

Aquí se deben seguir los pasos para llevar del DER al modelo lógico global de los datos.

Si en el paso anterior no se modifica el DER obtenido, entonces las tablas que se obtienen en este paso, en general, son las mismas que las

obtenidas en el segundo paso (Normalizar las entidades). Supongamos que, en nuestro ejemplo, éste es el caso. Entonces:

Diseño lógico:

Paciente (HC, NomPac, Ed, Sexo)

Intervención (operación, TiempoProm, Espec)

Material (CódMat, DescMat, Precio)

PQ (Fecha, HC, DiagPre, Interv, Hora, Salón, Cirujano)

IO (Fecha, HoraCom, HC, HoraTerm, DiagOper, SitPac, TipoOpReal, CirReal, OperReal)

OS (Fecha, HC, Causa)

DS (Salón, DíaSem, Espec)

GMO (Fecha, HC, HoraCom, CódMat, Cant)

1.5 Diseño físico de la base de datos

En este paso hay que tener en cuenta:

- Aplicaciones a realizar sobre los datos
- Características particulares del SGBD

Ejemplos de estos dos aspectos a tener en cuenta:

- considerar la inclusión de campos que sean secundarios debido a la cantidad de veces que habría que calcularlos en una determinada aplicación
- separar un archivo en varios debido a la cantidad de campos y lo que se accesa cada vez
- archivos índices
- archivos para reportes

2. AUTOEVALUACIÓN

- a) Suponiendo que, en cada caso, las tablas que se presentan han sido las únicas obtenidas después de concluirse el segundo paso de la metodología para el diseño de bases de datos (normalización), obtenga el diagrama entidad-relación correspondiente:

1.1.1. **niño** (idniño, nomniño, fechnac, sexo)

vacuna (códvac, nombrevac, paísproced)

vacunación (idniño, códvac, fechavacunación)

1.1.2. **cigarro** (nombrecigarro, calidad, preciounidad, códempresa)

empresaproductora (códempresa, nomempresa, añosdefundada)

1.1.3. **trabajopresentadoaconcurso** (códtrab, título

trabajopremiado (códtrab, premioobtenido, fechapublicación)

trabajodestacado (códtrab, lugaralcanzado)

(tipotrab indica si el trabajo presentado a concurso fue premiado, destacado o no tuvo ninguna distinción)

- 1.1.4. **producto** (códprod, nomprod, precio)
empresa (númemp, nmempresa, rama)
producción (númemp, códprod, cantidadproducida)
almacén (códalm, diralm, capacidad)
distribución (númemp, códprod, códalm, cantidadrecibida, fecharecepción)
- 1.1.5. **grupo** (númgrupo, ubicación)
alumno (númgrupo, númlista, nombrealumno, fechnac)
- 1.1.6. **compañíaaviación** (nomcía, sedeprincipal, añosdefundada)
viaje (códviaje, origen, destino, cantidadkm)
vuelo (nomcía, códviaje, precio, frecuencia)
- 1.1.7. **centrocultural** (códcent, nomcent, dircent, tipocent)
teatro (códcent, año fundación, directorteatro)
museo (códcent, manifestaciónartística, horario)
salaexposiciones (códcent, distrito)
- (tipocent indica si el centro cultural es un teatro, un museo o una sala de exposiciones)
- 1.1.8. **periódico** (códperiód, nomperiód, país)
noticia (códperiód, fecha, númnoticia, título, textonoticia)
- 1.1.9. **medicamento** (códmed, nomgenérico, nomcomercial)
formadepresentación (códforma, descripforma)
medicamento_forma (códmed, códforma)
presentacióndelmedicamento (códmed, códforma, costoproducción)
materiaprima (códmat, nommat, estadofísico)
composición (códmed, códforma, códmat, cantidad)
- 1.1.10. **temainvestigación** (códtema, nomtema, ramacientífica, DNIprofesor)
profesor (DNIprofesor, nomprofesor, categoríaprofesor)

Resumen

- 📖 La metodología para el diseño de la base de datos, consta de los siguientes pasos:
 - Determinación de entidades y atributos
 - Normalización de entidades
 - Determinación de relaciones (DER)
 - Obtención del modelo lógico global de los datos
 - Diseño físico de la base de datos

- 📖 Para la determinación de entidades y atributos, para cada salida, se hace lo siguiente:
 - Consultar su formato
 - Determinar datos que se calculen u obtengan a partir de otros e ir sustituyéndolos hasta llegar a los primarios
 - Determinar la existencia de archivos con información normativa y/o de consulta
 - Cada salida y cada archivo, tomarlos como entidades y relacionar (listar) sus atributos en una tabla

- 📖 En la normalización de entidades, para cada tabla definida, hay que:
 - Determinar las DF
 - Normalizar cada entidad
 - Fusionar, de ser lógico, las entidades normalizadas que tengan la misma llave


- 📖 Determinación de relaciones (DER)
 - Analizar cada entidad con las restantes para determinar si existe relación, determinando tipo de proyección.
 - Si hay alguna entidad cuya llave es compuesta, de modo que resulta que esa llave es la combinación de las llaves de otras entidades, es porque representa una relación entre ellas de m: n
 - Si hay alguna entidad cuya llave es compuesta y un subconjunto de los atributos que la forma no es llave de otra entidad, es porque es una entidad débil
 - Si en una entidad aparece, como atributo no llave, la llave de otra entidad es porque existe entre ellas una relación de m:1
 - Analizar entidades agregadas, entidades generalizadas/especializadas y sus relaciones con otras
 - Ir confeccionando el DER
 - Fusionar, de ser lógico, las entidades normalizadas que tengan la misma llave

- 📖 Obtención del modelo lógico global de los datos: se realiza de acuerdo a lo estudiado en la sesión 11

- 📖 Diseño físico de la base de datos

En este paso hay que tener en cuenta:

- Aplicaciones a realizar sobre los datos
- Características particulares del SGBD

 Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

 <http://ict.udlap.mx/people/carlos/is341/bases03.html>

En esta página web encontrará conceptos adicionales y ejemplos de cómo se transforma paso a paso un Diagrama Entidad Relación a un Modelo Relacional.

**UNIDAD DE
APRENDIZAJE****6****SEMANA****16**

RECUPERACIÓN, CONCURRENCIA Y SEGURIDAD

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al final de la unidad, los alumnos señalan las características de las bases de datos actuales a partir de casos expuestos por el profesor, aplicando los conceptos de seguridad de base de datos y almacén de grandes volúmenes de datos.

TEMARIO

Recuperación

Concurrencia

Seguridad

Otros temas de interés sobre base de datos

ACTIVIDADES PROPUESTAS

1. Explicaciones por parte del facilitador combinadas con ejemplos.
2. Se efectúan preguntas de comprobación dirigidas a los alumnos sobre los temas tratados

1. RECUPERACIÓN

Los problemas de recuperación y concurrencia están muy relacionados con el concepto de transacción, por lo que es preciso conocer éste en primer lugar.

Definición: **Transacción**

Una transacción es una unidad lógica de trabajo.

Ejemplo:

En el modelo **suministrador-producto**, cambiar el número del suministrador **SX** por el número **SY**.

Para el usuario, esto es una única, indivisible operación. Sin embargo, esto implica realizar dos actualizaciones en la base de datos: actualizar la ocurrencia correspondiente en **suministrador** y todas las correspondientes en **sp**.

Probablemente, durante el tiempo en que se realicen estas dos actualizaciones, la base de datos no estará siempre consistente.

Entonces, una **transacción** no es necesariamente una única operación sobre la base de datos, sino que es, en general, una secuencia de varias operaciones sobre el repositorio de datos que transforman un estado consistente de este en otro, sin que se preserve necesariamente la consistencia en los momentos intermedios.

Para nuestro ejemplo, no puede permitirse que se realice una de las actualizaciones y la otra no, pues la base de datos quedaría inconsistente. Sería preciso garantizar que ambas actualizaciones se realizaran, pero esto no se puede lograr, ya que siempre existe la posibilidad de que algo funcione mal. Por ejemplo, puede ocurrir una falla del sistema tal como falta de fluido eléctrico, entre ambas actualizaciones.

Pero un sistema que realice el procesamiento de transacciones provee métodos que son lo más cercano posible a esa garantía. Específicamente, garantiza que si una transacción T realiza ciertas actualizaciones y ocurre una falla antes de que la T se termine, entonces esas actualizaciones se eliminan (undo). Entonces, una T se ejecuta totalmente o se cancela completamente como si nunca se hubiera realizado. De esta forma, una secuencia de operaciones no indivisibles puede parecer como si lo fuera.

Para ello, los SGBD utilizan dos operaciones que señalan el final exitoso de una T: **TERMINADA** o bien el final no exitoso de una T: **CANCELADA**, en cuyo caso las actualizaciones hechas hasta el momento deben ser canceladas.

El SGBD mantiene un "diario" en cinta magnética o disco en el que registra detalles de todas las operaciones de actualización, en particular, los valores anterior y posterior del artículo actualizado. Entonces, si es necesario cancelar determinada actualización, el sistema usa la entrada correspondiente en el "diario" para restaurar el artículo con el valor previo a la actualización, una vez que el sistema comience nuevamente a trabajar.

Para una transacción que haya sido señalada por el sistema como TERMINADA, el SGBD tiene que garantizar que sus actualizaciones queden establecidas en la base de datos, aún si el sistema falla justamente en el próximo instante al que se haya dado como TERMINADA, y aún cuando las actualizaciones no hayan sido físicamente escritas en la base de datos (ellas podían estar aún esperando en un *buffer* en la memoria principal y por tanto, perderse en el momento de la falla). Incluso si esto pasa, el procedimiento de reinicio del sistema debe incluir estas actualizaciones en la base de datos. Esto es posible hacerlo, ya que los valores a ser escritos pueden conocerse examinando las entradas del "diario". Por supuesto, que el diario debe ser físicamente escrito antes de que se dé la señal de TERMINADA para una T dada.

Un sistema puede sufrir diferentes tipos de fallas y debe estar preparado para recuperarse cuando ocurre cualquiera de ellas.

1.1 Tipos de fallas

1.1.1 Falla local

Por ejemplo, la ocurrencia de un "overflow" dentro de una única T. Una falla local afecta sólo la T en la cual ocurre y para ella puede aplicarse lo que hemos visto hasta el momento.

1.1.2 Falla del sistema

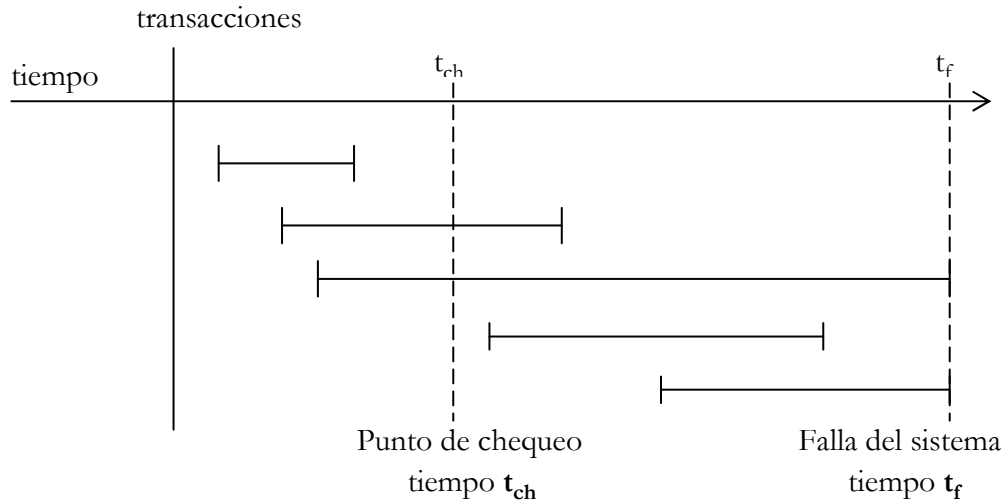
Por ejemplo, falla del fluido eléctrico. Afecta todas las T que se estaban desarrollando en el momento, pero no daña físicamente la base de datos.

Lo principal aquí es que el contenido de la memoria se pierde (en particular, los buffers de la base de datos). No se conoce el estado de las T que se estaban desarrollando y, por tanto, deben ser canceladas cuando el sistema se reinicia. Además, deben completarse las T que fueron terminadas pero cuyas actualizaciones no fueron escritas físicamente en la base de datos.

El sistema debe conocer cuáles T cancelar y cuáles completar cuando se reinicia. Para ello, cada cierto intervalo (típicamente cuando cierto número de entradas se han escrito en el "diario") el sistema automáticamente toma un "punto de chequeo". Esto implica:

- a. Escribir físicamente el contenido de todos los *buffers* de la base de datos en la base de datos almacenada en el dispositivo correspondiente.
- b. Escribir físicamente un artículo especial de punto de chequeo en el "diario". Este artículo contiene una lista de todas la T que se estaban desarrollando en el momento en que se tomó el punto de chequeo.

En el siguiente gráfico se representan todas las situaciones en que puede estar una transacción en el momento de una falla del sistema:



En este gráfico, se representa lo siguiente:

Ocurrió una falla del sistema en el tiempo t_f .

El más reciente punto de chequeo antes de t_f se tomó en el tiempo t_{ch} .

Las T del tipo T1 son las terminadas antes del tiempo t_{ch} .

Las T del tipo T2 empiezan antes de t_{ch} y terminan antes de t_f .

Las T del tipo T3 empiezan antes de t_{ch} pero no terminan antes de t_f .

Las T del tipo T4 empiezan después de t_{ch} y se terminan antes de t_f .

Las T del tipo T5 empiezan después de t_{ch} pero no se completan antes de t_f .

Cuando el sistema se reinicie, las transacciones del tipo T3 y T5 deben ser canceladas y las del tipo T2 y T4, completadas. Las del tipo T1 no se considerarán en el procedimiento de reinicio del sistema, ya que sus actualizaciones fueron físicamente escritas en la base de datos en el tiempo t_{ch} .

En el momento del reinicio, el sistema ejecuta el siguiente procedimiento para identificar las transacciones a cancelar y a completar:

		CANCELAR	COMPLETAR
1	Comienza con 2 listas de transacciones: lista de T a CANCELAR y lista de T a COMPLETAR. Incluye en la lista CANCELAR todas las T en desarrollo en el tiempo t_{ch} y la lista COMPLETAR se deja vacía.	T2 T3	
2	Busca hacia adelante en el "diario", empezando desde el artículo de punto de chequeo.		
3	Si se encuentra una entrada de "inicio de T", se añade T a la lista de CANCELAR.	T2, T3 T4 T5	
4	Si se encuentra una entrada de "TERMINADA", se mueve T de la lista CANCELAR a la lista COMPLETAR. T5	T3 T5	T2 T4
5	Cuando se alcanza el final del diario, las listas CANCELAR y COMPLETAR identifican, respectivamente, las T a cancelar (T3 y T5), y las T a completar (T2 y T4)	T3 T5	T2 T4

El sistema ahora trabaja hacia atrás a través del "diario", cancelando las T de la lista CANCELAR y completando las T de la lista COMPLETAR. Cuando toda esta actividad de recuperación se termina es que el sistema está listo para aceptar nuevos trabajos.

1.1.3 Falla de los medios

Por ejemplo, una falla de la cabeza del disco. Causa daño a la base de datos o a alguna porción de ella y afecta, al menos, aquellas transacciones que estaban usando tal porción en el momento de la falla.

Recuperarse de este tipo de falla significa volver a cargar (restaurar) la base de datos a partir de una copia de ésta y entonces usar el "diario" para rehacer las T que fueron terminadas desde que la copia de la base de datos fue tomada. (No hay que cancelar las T que se estaban desarrollando en el momento de la falla ya que de todos modos las actualizaciones de esas T serían eliminadas y por ello no se toman en cuenta en esta recuperación).

Entonces el sistema tendrá un utilitario para hacer volcados (*backups*) y restauraciones de la base de datos cuando esto sea necesario según se demande.

2. CONCURRENCIA

La mayoría de los SGBD son de múltiples usuarios. Es decir, permiten cualquier número de T accedendo la misma base de datos al mismo tiempo. En estos sistemas, se necesita algún mecanismo de control de concurrencia como, por ejemplo, el mecanismo de bloqueo o cierre, que es uno de los más empleados en la práctica.

Hay distintos casos en que puede haber problemas, es decir, en los que una T correctamente definida puede, no obstante, producir un resultado incorrecto debido a la interferencia de otra T (en el caso de que no haya un mecanismo de control adecuado).

Veamos uno de estos casos a manera de ilustración:

2.1 Actualización perdida

<u>Transacción A</u>	<u>Tiempo</u>	<u>Transacción B</u>
Recuperar R	t1	
	t2	Recuperar R
Actualizar R	t3	
	t4	Actualizar R

A recupera el artículo R en el tiempo **t1** y **B**, el mismo artículo en el tiempo **t2**.

A actualiza **R** en **t3** y **B** actualiza el mismo artículo en **t4**, de acuerdo al valor encontrado en **t1**.

Entonces, la actualización hecha por **A** se pierde en el tiempo **t4**, pues **B** sobrescribe el artículo sin haberla considerado.

2.2 Cierre o bloqueo

La idea básica del método de cierre es simple: cuando una T necesita que algún objeto (típicamente un artículo) no cambie de forma impredecible mientras que ella lo está tratando, T adquiere un cierre sobre tal objeto. El efecto es que otras T "queden fuera", que no se les permita acceder el artículo y, por tanto, se previene que éste sea cambiado por esas T.

En realidad, existen diferentes tipos de cierre o bloqueo, pero asumiremos 2:

2.2.1 Tipos de cierres o bloqueos

- Cierre o bloqueo exclusivo (X)

Si **A** adquiere un cierre **X** sobre el artículo **R**, entonces una solicitud de una transacción **B** de obtener un cierre de cualquier tipo sobre **R** causará que **B** vaya a un estado de espera (hasta que el cierre de **A** sea liberado).

- Cierre o bloqueo compartido (**S**)

Si **A** adquiere un cierre **S** sobre **R**,

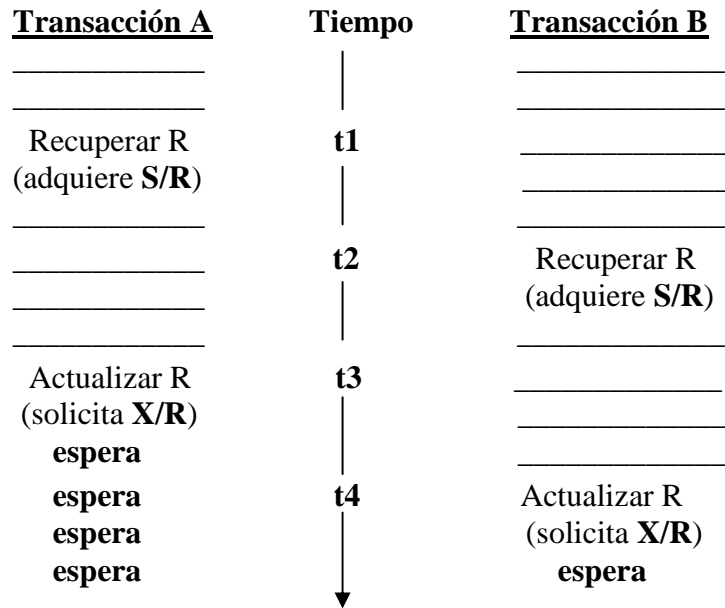
- una solicitud de **B** de obtener un **X** sobre **R** causará que **B** pase a un estado de espera (hasta que el cierre de **A** sea liberado).
- una solicitud de **B** de obtener un **S** sobre **R** será satisfecha, es decir, **B** obtendrá también un cierre **S** sobre **R**.

Esto puede representarse por medio de una matriz de compatibilidad (por completitud se incluye el caso de no cierre representado por "-").

A \ B	X	S	-
X	No	No	Sí
S	No	Sí	Sí
-	Sí	Sí	Sí

En los sistemas más modernos, las solicitudes de cierre por las **T** son implícitas. Cuando una **T** recupera un artículo, automáticamente adquiere un **S** sobre el artículo. Cuando una **T** actualiza un artículo, automáticamente adquiere un **X** sobre el artículo (si ya tenía un **S**, éste es promovido a **X**).

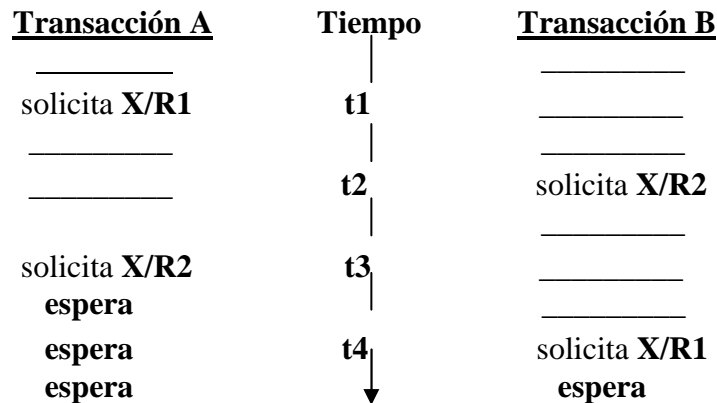
Veamos cómo este mecanismo resuelve el problema de la actualización perdida.



A no pasa a actualizar, pues **B** tiene un **S** sobre **R**, por lo que no se pierde una actualización; pero **B** tampoco puede proceder, así que el cierre resuelve el problema de la actualización perdida creando un nuevo problema, llamado "abrazo fatal" (deadlock).

2.3 Abrazo fatal

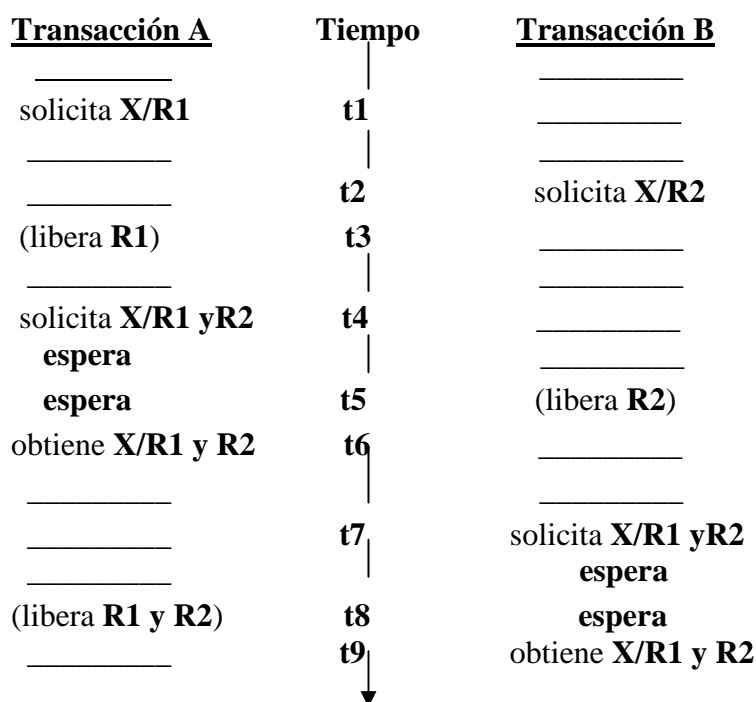
Una versión más generalizada del problema del abrazo fatal se muestra a continuación:



El abrazo fatal es una situación en la que dos o más T están simultáneamente en un estado de espera, cada una esperando porque una de las otras libere un recurso que ella necesita para proseguir.

Si un lazo fatal ocurre, el sistema debe detectarlo y romperlo. Esto último se realiza seleccionando una de las T involucradas en el abrazo fatal como "víctima" y deshaciendo lo realizado por ella y, por lo tanto, liberando sus cierres, lo que permite a otra T proseguir. Muchos sistemas reinician automáticamente dicha T, asumiendo que las condiciones causantes del abrazo fatal no ocurrirán.

No obstante, es preferible evitar la ocurrencia de un abrazo fatal que recuperarse de él. Lo más usual para evitarlo es no permitir a una transacción solicitar un recurso, (por ejemplo, un artículo) si ya tiene asignado alguno. Una transacción deber solicitar de una vez todos los recursos que requiere para una operación dada y, antes, deberá liberar todos los recursos sobre los que tenía acceso permitido.



3. SEGURIDAD

El término seguridad se refiere a asegurar que los usuarios tengan autorización o no para realizar lo que están tratando de hacer.

La unidad de datos para el propósito de su seguridad puede variar desde una colección de tablas hasta un valor de dato específico en una (fila, columna) en una tabla. Un usuario tendrá diferentes derechos de acceso o autoridades sobre distintos objetos. Por ejemplo, autoridad para recuperar sobre una tabla, para recuperar y actualizar sobre otra, etc. También, diferentes usuarios podrán tener diferentes autoridades sobre los mismos objetos.

Existen diferentes mecanismos que se utilizan para garantizar la seguridad:

3.1 Vistas

Este mecanismo es utilizado para "esconder" datos a usuarios no autorizados.

Por ejemplo, permitir a un usuario el acceso a los artículos **suministrador**, pero sólo a los suministradores del distrito SAN ISIDRO.

3.2 Subsistema de autorización

Permite a usuarios que tienen determinados derechos, selectiva y dinámicamente otorgar esos derechos a otros usuarios y revocarlos si se desea.

Por supuesto, la decisión de qué derechos se otorgan a cada usuario es una decisión más bien política, que se sale de la jurisdicción del SGBD. Éste debe garantizar que se respeten esos derechos. Para ello:

- Esas decisiones deben informarse al SGBD y éste debe "recordarlas", almacenándolas en el catálogo en forma de "restricciones de autorización".
- Existe un medio de chequear cada solicitud contra las "restricciones de autorización".
- Para decidir las restricciones aplicables a una solicitud, el sistema tiene que reconocer qué usuario la realiza. Para ello, el usuario tiene que suministrar su identificación (ID) para decir quién es él y también una palabra de pase o clave (password) para probar que él es quien él dice que es. Esa palabra de pase sólo es conocida por el sistema y por el usuario legítimo.

Pueden usarse diferentes procedimientos para la palabra de pase:

- a. Por medio de una función **F** que se asigna al usuario:

El sistema envía un número aleatorio **x**. El usuario debe calcular $F(x)=y$ y envía ese valor **y** al sistema (el usuario conoce **F**). El sistema confirma la transformación verificando que el resultado es **y**. (No es fácil deducir **F**, aún conociendo **x** y **y**, aún para $F(x)=(\text{suma dígitos en posiciones impares})^2+(\text{hora del día})$)

- b. Método del polinomio

$$x_i = a_{x_i-1} + b_{x_i-2}$$

Los sistemas reconocen un amplio rango de diferentes autoridades, como por ejemplo, recuperar, actualizar, eliminar, insertar.

El subsistema de autorización incluye la designación de un usuario privilegiado que es el administrador del sistema que está autorizado a realizar cualquier operación permitida por el sistema. Inicialmente, sólo él es el usuario que puede hacer cualquier cosa incluyendo otorgar derechos a otros usuarios.

Supongamos que el administrador del sistema otorga el derecho a crear algún objeto (sea una tabla) a otro usuario **U** y que **U**, efectivamente, la crea. El usuario **U** automáticamente recibe todos los derechos sobre ese objeto, incluyendo el de otorgar tales derechos a otros usuarios.

Por supuesto, en este mecanismo los derechos que se otorgan tienen que tener sentido. Por ejemplo, **U** tiene autoridad para RECUPERAR en la tabla **T** y **U** crea una vista **V** basada en **T**, entonces **U** no recibe autoridad para ACTUALIZAR **V**.

Si un usuario U1 otorga alguna autoridad a un usuario U2, consecuentemente puede revocar tal autoridad.

No se puede asumir que el sistema de seguridad en un SGBD es perfecto. Siempre es posible que alguien pueda infiltrarse burlando los controles.

Por ello, cuando los datos son muy sensitivos, se puede realizar una auditoría que analice todas las operaciones realizadas sobre los datos a partir de un archivo o base de datos especial en el cual el sistema almacena información acerca de todas las operaciones realizadas por el usuario, como por ejemplo:

operación (ejemplo: ACTUALIZAR)
terminal desde la que la operación fue invocada
usuario que la invocó
fecha y hora de la operación
BD, tabla, artículo y **campo** afectado
valor viejo del campo
valor nuevo del campo

En algunos sistemas este archivo o base de datos está integrado con el "diario" utilizado para la recuperación del sistema.

3.3 Cifrado de los datos

Otro nivel de seguridad se obtiene por el cifrado de los datos. Los datos se almacenan físicamente y son transmitidos a través de las líneas de comunicación en forma cifrada, de manera que alguien que trate de acceder los datos de forma ilegal obtenga una ininteligible colección de bits.

Algunos sistemas permiten usar, o no, en forma opcional los mecanismos de seguridad.

4. REQUERIMIENTOS ACTUALES

Como hemos visto anteriormente, el modelo relacional no es el único modelo de datos que ha existido. Mencionamos que las primeras bases de datos respondían al modelo jerárquico y luego al modelo reticular. Pero el modelo relacional no es último, es decir, el desarrollo ha continuado y en esta sesión exploraremos algunas direcciones de la evolución actual de la tecnología de base de datos.

Los SBD surgieron con el propósito de manipular grandes volúmenes de información. Tan es así que las primeras bases de datos (jerárquicas) surgieron del desarrollo de los sistemas de gestión de archivos iniciales. Las aplicaciones de base de datos de todos estos años (incluyendo las bases de datos relacionales) han tenido como características comunes las siguientes:

- Uniformidad: Datos estructurados de manera similar, de igual tamaño (en bytes).
- Orientación en registros: Registros de longitud fija.
- Datos (relativamente) pequeños.
- Campos atómicos: No hay una estructura en ellos (1FN).
- Transacciones cortas: transacciones de poca duración, sin que haya interacción humana durante su ejecución.

- Esquema de las base de datos estático: Cambia con poca frecuencia, cambios sencillos.

En los últimos años, la tecnología de base de datos se ha adaptado a aplicaciones que ya no son del ámbito del procesamiento de datos “tradicional” (aplicaciones de “gestión”). Estas nuevas aplicaciones difieren en sus características de las mencionadas anteriormente y entre estas aplicaciones se encuentran:

- **CAD (Computer Aided Design)**

La base de datos debe almacenar datos de diseño de ingeniería que son, entre otros, los componentes del objeto de diseño, versiones diferentes de los diseños, etc.

- **CAM (Computer Aided Manufacturing)**

En un servicio de ayuda a la producción de componentes o máquinas, donde se monitorea el ciclo de producción. Es necesario almacenar el estado de los objetos y la historia de dichos estados en el proceso de manufactura.

- **CASE (Computer Aided Software Engineering)**

Los datos a almacenar serán:

Códigos, dependencias entre módulos, definiciones de variables y versiones del sistema que se desarrolla.

- **Base de datos de Multimedia**

Inclusión de datos de audio, video, etc.(llamados *BLOBs*, o sea grandes objetos binarios)

- **Sistema de información de oficinas**

Las aplicaciones de la ofimática incluyen guardar y recuperar documentos, calendarios, objetos de datos de imágenes y voz.

Estas nuevas aplicaciones no se consideraron en la década de los 70 y parte de los 80, cuando se desarrollaron muchos de los SGBD que han existido. Pero, en la actualidad, pueden llevarse a la práctica gracias al desarrollo de los medios técnicos.

Y esas nuevas aplicaciones tienen nuevos requisitos, como son:

- **Objetos complejos**

Datos vistos como un simple objeto en el mundo real, pero que, a su vez, contienen otros objetos.

- **Datos de comportamiento**

Cómo deben los objetos responder a las órdenes que reciben. Esta información puede almacenarse como parte de la base de datos, (ejemplo: los métodos en los sistemas de bases de datos orientadas a objetos (SBDOO))

- **Metaconocimiento o “Reglas del negocio”**

Reglas generales acerca de la aplicación que son tan importantes como los datos mismos.

Por ejemplo: “en un sistema bancario, las cuentas con saldo mayor de 1000 soles pagan 5% de intereses; para saldos menores no hay pago de intereses”.

- **Transacciones de larga duración**

Las aplicaciones como las de CAD y CASE implican gran interacción humana con los datos. Los conflictos de bloqueo son mayores y más serios.

El modelo relacional tiene insuficiencias en las áreas antes mencionadas y ante esta situación hay dos variantes de solución:

- Extender o ampliar el modelo relacional
- Desecharlo y sustituirlo por algo nuevo.

Vamos a analizar la primera variante:

5. EXTENSIONES DEL MODELO RELACIONAL

Se han propuesto diferentes extensiones, algunas más recientes y vinculadas a los problemas antes mencionados, y otras no tan recientes, entre ellas están:

- **Reglas para llaves extranjeras**

Enunciadas por Date, se refieren a cómo garantizar la Integridad Referencial.

- **Nuevos operadores algebraicos**

Entre éstos se encuentra la reunión externa (*Outer Join*)

- **M. Relacional Extendido (RM/T)**

Propuesto por *Codd* en 1979. Tiene características similares al modelo entidad - relación ya que también intenta abordar el problema del modelamiento semántico.

- **Relaciones Anidadas**

También llamadas relaciones NF^2 : **Non First Normal Form**, ya que se desecha el requerimiento de que las relaciones estén normalizadas, o sea, que cumplan la 1FN. Esto permite entonces que el valor de un atributo pueda ser, a su vez, otra relación (anidada) y facilita la representación de objetos complejos.

- **Sistemas basados en la lógica**

También llamadas base de datos deductivas, SGBD expertos, etc. Como su nombre indica, se basan en la lógica (específicamente, en la **Lógica de Predicados de Primer Orden**). En una base de datos clásica el contenido está constituido por hechos particulares, mientras que en una base de datos deductiva se almacenan hechos y reglas que permiten obtener nuevos hechos y reglas.

- **Procedimientos almacenados, Disparadores o Gatillos (*triggers*) y Reglas**

Muchos SGBD relacionales actuales permiten que se mantenga, junto con los datos, **procedimientos almacenados**, que podrán ser ejecutados, explícitamente, por los programas de aplicaciones que accedieron la base de datos. También hay SGBD que tienen un sistema de disparadores o *triggers*. Estos **triggers** definen un conjunto de reglas que gobiernan bajo qué circunstancias debe ser ejecutado un procedimiento almacenado. O sea, que estos disparadores permiten que, automáticamente, se ejecute un código determinado como respuesta a alguna acción, por ejemplo, una actividad de la base de datos que ocasione que se cumplan las reglas definidas.

Todo esto permite construir cierta “inteligencia” de la base de datos.

6. BASES DE DATOS DISTRIBUIDAS

El surgimiento de los sistemas de bases de datos distribuidas (SBDD) ha estado propiciado, entre otros aspectos, por:

- La introducción de las redes de computadoras
- La expansión territorial de las empresas
- El aumento de la cantidad de usuarios no informáticos

Su desarrollo está íntimamente ligado al enfoque Cliente/Servidor, como veremos más adelante.

Los SBDD surgen a partir de dos **vías**:

- **SBD centralizados:**

En estos sistemas se presentaban dificultades en la transportación de datos y por el alargamiento de los plazos de entrega. Como soluciones a esos dos problemas se comenzaron a usar terminales no inteligentes con una computadora central y, poco a poco, esas terminales fueron adquiriendo más funciones y capacidad de procesamiento y de almacenamiento de pequeñas base de datos, hasta lograr que su interconexión diera paso a las BDD.

- **base de datos sobre micro y minicomputadoras**

La existencia de base de datos sobre micro y minicomputadoras y su posterior interconexión llevan a la descentralización de los datos.

Ambas vías permitieron desconcentrar los medios técnicos y distribuir funciones y datos, lo que dio lugar al surgimiento de las BDD.

Al definir en qué consiste un sistema de bases de datos distribuidos, es preciso mencionar las siguientes características:

- Está compuesto por varios nodos conectados por canales de comunicación
- Cada nodo puede procesar programas y almacenar datos
- Todos los nodos no tienen que tener igual configuración
- Los nodos son equivalentes y deciden, de manera autónoma, si procesan una tarea o la pasan a otro nodo
- El sistema es transparente para los usuarios, en el sentido de que el usuario no tiene que conocer dónde están los datos que solicita y, ni siquiera, conocer de la existencia de otros nodos

Un **sistema distribuido** puede verse como cualquier sistema que comprenda múltiples nodos interconectados en algún tipo de red de comunicación, en el cual un usuario (usuario final o programador de aplicaciones) en cualquier nodo puede acceder datos almacenados en cualquier nodo.

Cada nodo puede a su vez ser visto como un SBD con su administrador de base de datos, terminales y usuarios, CPU, memoria y SGBD.

Existen diferentes **objetivos** que los SBDD deben cumplir y, aunque diferentes autores plantean objetivos un tanto diferentes, pueden mencionarse, de manera general, los siguientes:

- Compartir los datos entre los nodos

Un dato situado en un nodo puede ser accesado desde cualquier nodo.

- Almacenar los datos en los lugares de su uso frecuente

Esto evita costos de transmisión y reduce tiempos de respuesta.

- Accesos transparentes a la ubicación del dato

Un usuario, desde un nodo, solicita un dato sin conocer la ubicación de éste. Para el usuario es como si el sistema no fuera distribuido.

- Crecimiento

El sistema debe permitir agregar nuevos nodos y aumentar la base de datos en cada nodo.

- Confiabilidad y disponibilidad

Los nodos restantes pueden continuar trabajando aunque fallen uno o más nodos. Los datos se replican, por lo que, en general, aunque falle un nodo es posible acceder el dato en algún otro nodo.

- Fragmentación de datos

El sistema soporta la fragmentación de los datos. Un objeto lógico se divide en fragmentos. Un fragmento es cualquier subrelación arbitraria que es derivable

de la relación original mediante selección y proyección, excepto que en el caso de la proyección se debe preservar la llave primaria de la relación original o algún identificador de la tupla.

- Transparencia de la fragmentación

Debe resaltarse la facilidad del modelo relacional para la fragmentación y su transparencia.

- Réplicas de los datos

El sistema debe ser capaz de permitir la existencia de datos replicados y que ello sea transparente para el usuario. Trae la gran ventaja de la confiabilidad y disponibilidad del sistema, pero tiene como desventaja que hay que actualizar todas las réplicas. Si hay fragmentación, la unidad de réplica es el fragmento, no la relación lógica completa.

Los SGBDD externamente trabajan como un SGBD, con sus mismas funciones y otras adicionales, tales como:

- Descripción de la BDD

Define el esquema global que contemple cada uno de las base de datos locales (ubicación física y descripción lógica)

- Manipulación de la BDD

Posibilidad de obtener datos que se encuentren en cualquier nodo. Para ello, un requerimiento global se descompone en subrequerimientos locales que se dirigen a los nodos, de forma que, para acceder a un dato, el SGBDD sigue los siguientes pasos:

- determinar si el dato es local o no
- si no es local, determinar su ubicación y rutas de acceso
- enviar subrequerimientos a los nodos
- recibir respuestas en una respuesta global

Como un requerimiento se descompone en varios subrequerimientos que son enviados a diferentes nodos, entonces éstos, en general, trabajan en paralelo resolviendo esos subrequerimientos, por lo que se agilizan las consultas.

Todo esto debe ser transparente al usuario.

La distribución de los datos en un SBDD se puede analizar desde dos puntos de vista: según la redundancia y según la fragmentación.

Según la **redundancia** que posea la BDD, o sea, según la duplicidad de los datos, pueden estar distribuidos según tres métodos:

- Cada base de datos de un nodo tiene una copia completa de la base de datos total. Es poco práctica, por el tamaño de la base de datos y por la dificultad que representa la actualización de tanta información al existir una redundancia total.
- Cada base de datos contiene datos que no están en ningún otro nodo, o sea, que hay cero redundancia. En este caso, sería difícil decidir, si se requiere un

dato con igual frecuencia en dos lugares, dónde ponerlo. Además, si un nodo falla, esto implica la caída de todo el sistema.

- Cada base de datos tiene subconjuntos de datos almacenados al mismo tiempo en otros nodos, es decir, hay redundancia parcial.

Según la **fragmentación** de las relaciones también pueden usarse tres métodos diferentes de distribución de los datos:

- Obteniendo fragmentos del esquema global por medio de proyecciones, es decir, obteniendo una fragmentación **vertical**. Por ejemplo, podrían distribuirse los segmentos en los nodos del siguiente modo, donde cada R_i representa un fragmento de una cierta relación R obtenido por proyección:

nodo 1	nodo 2	nodo 3
R1	R3	R1
R2	R4	R2
		R3

- Distribuyendo las ocurrencias de una base de datos entre los nodos, es decir, obteniendo fragmentos por medio de selecciones o, lo que es lo mismo, una fragmentación horizontal. Por ejemplo, si se tiene una relación llamada **Trabajador** con los campos (*DNl_{trab}, nombre, salario, distrito*), se podrían obtener fragmentos de modo que cada uno esté formado por la selección de los trabajadores de un distrito y distribuir estos fragmentos en los distintos nodos, pudiendo replicarse cada uno en más de un nodo.
- Combinación de las dos anteriores

Los SBDD brindan varias ventajas, entre ellas:

- Uso compartido de los datos de todos los nodos
- Mayor confiabilidad y disponibilidad de datos
- Agilización del procesamiento de consultas
- Representan mejor la estructura de las empresas modernas, que tienen dependencias geográficamente dispersas

7. ENTORNO CLIENTE / SERVIDOR

Recientemente este término Cliente/Servidor (C/S) se ha incorporado al argot de la Informática. La computación C/S se puede describir como un amplio paradigma de la computación mediante el cual los procesadores centralizados de años atrás están siendo reemplazados por redes de computadores más pequeñas y menos caras (LANs).

Con esta filosofía las empresas comerciales están interconectando sus microcomputadoras en LANs en las que la norma es la computación C/S. De esta forma, el poder de procesamiento de una colección de sistemas de cómputo es usado en un problema de negocios.

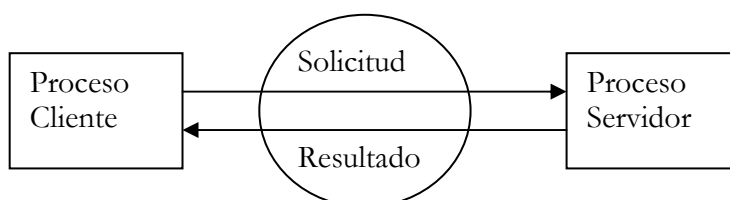
La **computación C/S** se usa para describir aplicaciones donde dos o más procesos lógicos separados cooperan para resolver una tarea. Los procesos se llaman cliente y servidor. Pero es necesario aclarar que tanto a los procesos o aplicaciones como a las computadoras donde éstos se ejecutan se les llama cliente y servidor.

Lo que determina si el procesamiento es C/S o no es el diseño lógico y no la distribución física. Ambos procesos pueden estar en el mismo equipo de cómputo y aún ser C/S. No obstante, normalmente, sobre todo en lo relativo a base de datos, se considera que el cliente y el servidor están en computadoras diferentes.

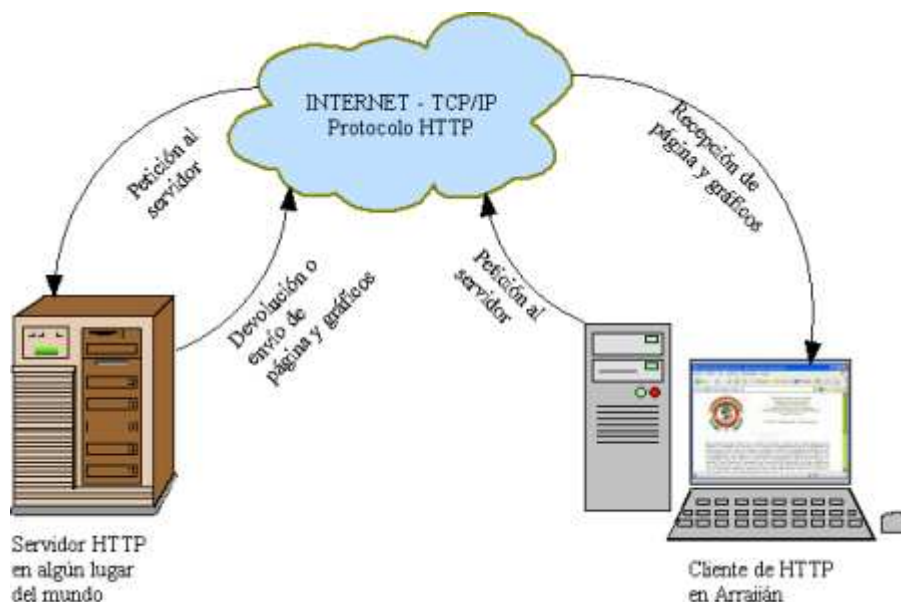
En una aplicación con esta filosofía C/S, un componente de la aplicación que se ejecuta en un sistema de cómputo (cliente) solicita un servicio a un componente que se ejecuta en otro sistema de cómputo (servidor).

En una aplicación C/S hay 3 componentes físicos:

1. Proceso **cliente**: solicita los servicios
2. Proceso **servidor**: satisface las solicitudes de servicios del cliente.
3. **Red** de comunicación.



Red de comunicación



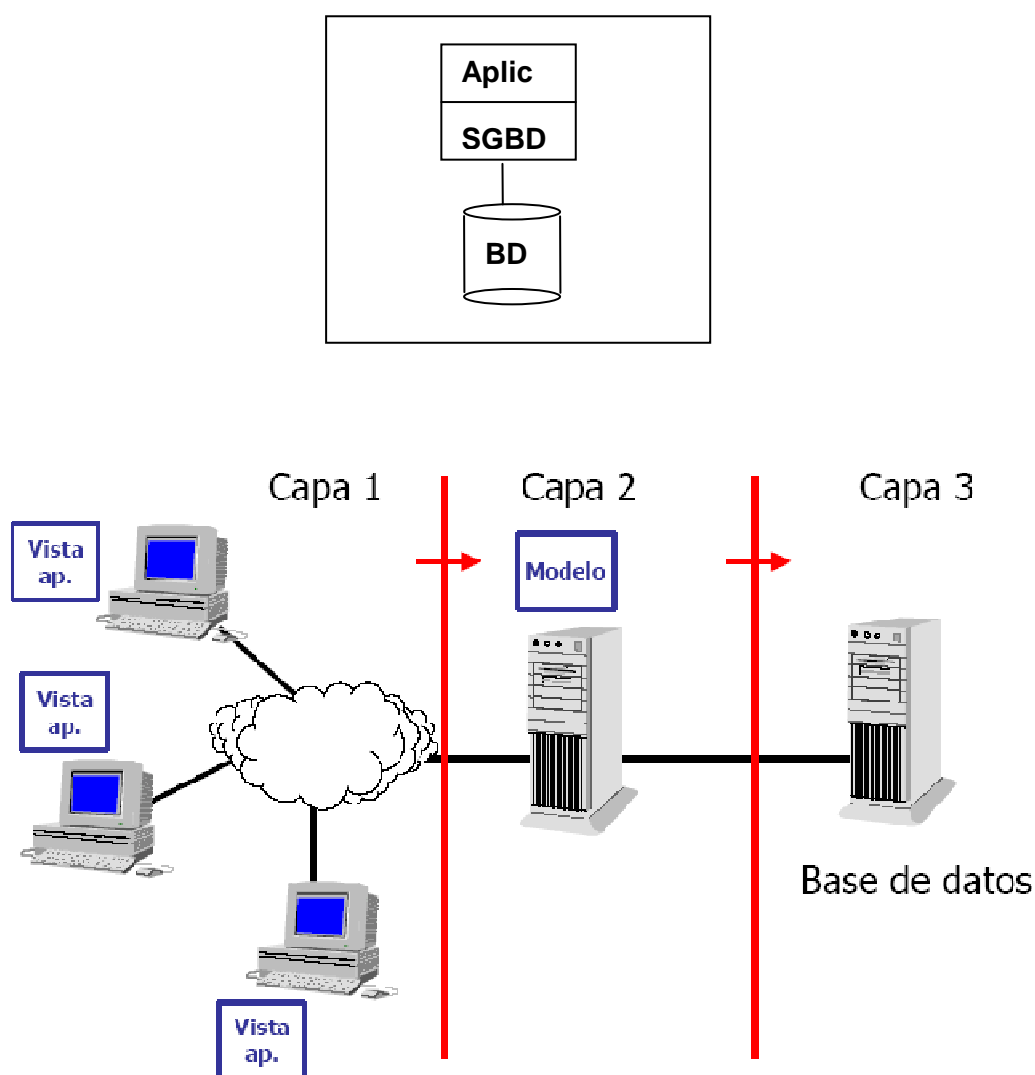
Cliente – Servidor
mediante Internet

El paradigma C/S permite a muchos componentes cliente acceder a un servicio común proporcionado por un servidor y también permite a un cliente acceder a muchos componentes servidores diferentes. Este amplio paradigma puede ser adoptado para crear una variedad de formas de computación distribuida.

Los componentes de una aplicación C/S son: la aplicación, el SGBD y la base de datos. Dependiendo de dónde residan estos componentes, se obtienen diferentes configuraciones de base de datos C/S. Entre estas configuraciones están:

- **Base de datos centralizada**

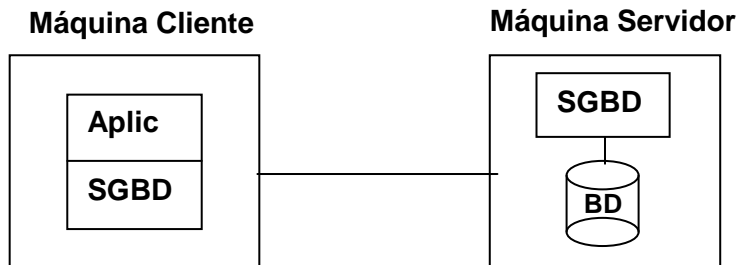
Todo se encuentra almacenado en un equipo informático, o sea, tanto la aplicación, como el SGBD y la base de datos residen en el mismo equipo de cómputo.



Base de datos centralizada en una arquitectura de tres (3) capas: Cliente-Reglas de Negocio-Base de Datos

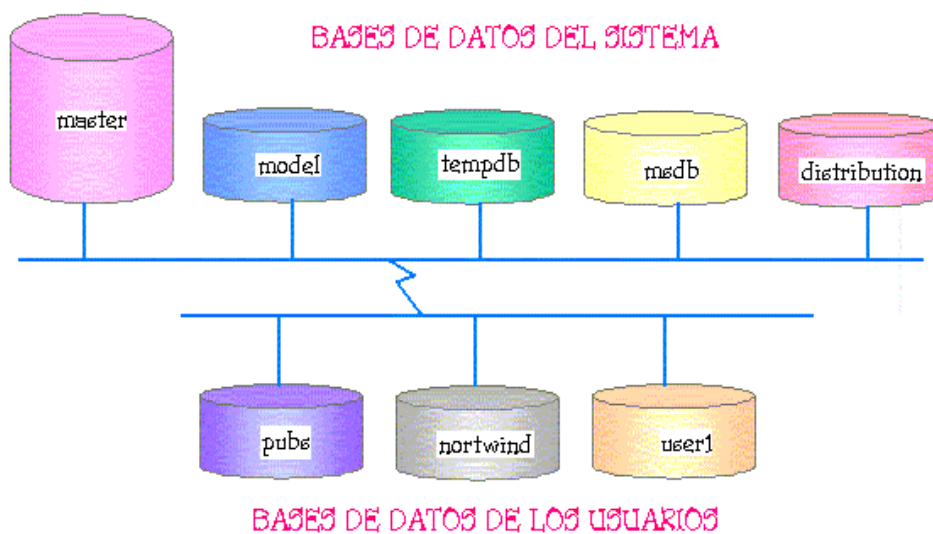
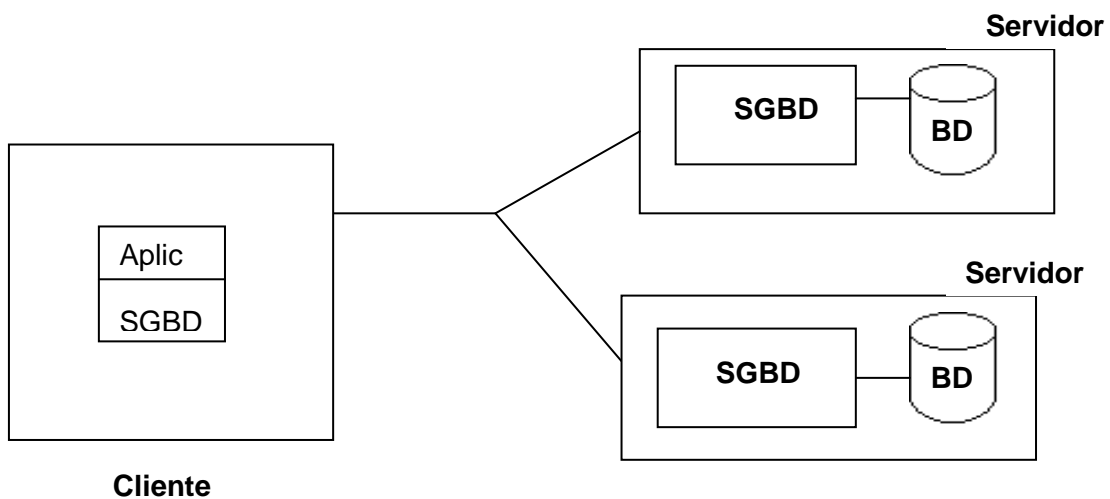
- **Base de datos Cliente/Servidor**

La base de datos reside en un equipo de cómputo diferente de la que se ejecuta la aplicación, y el SGBD es dividido entre los dos (2) equipos de cómputo.



- **Base de datos distribuida**

Tal y como fue visto anteriormente, la base de datos (BD) puede residir en más de un equipo informático.



9.- AUTOEVALUACIÓN

- a) Defina los siguientes conceptos:
- 1.1.1. Recuperación
 - 1.1.2. Transacción
 - 1.1.3. Falla local
 - 1.1.4. Falla del sistema
 - 1.1.5. Falla de los medios
 - 1.1.6. Punto de chequeo
 - 1.1.7. Volcado o copia de seguridad
 - 1.1.8. Concurrencia
 - 1.1.9. Cierre o bloqueo
 - 1.1.10. Seguridad
- b) Explique cómo se recupera un SGBD ante una falla local, ante una falla del sistema y ante una falla de los medios.
- c) ¿Qué tipos de cierres o bloqueo usted conoce? Explique cada uno de ellos.
- d) Explique en qué consisten los siguientes problemas que pueden presentarse por la ejecución concurrente de las transacciones y cómo se resuelven:
- a. Actualización perdida
 - b. Abrazo fatal
- e) Explique los mecanismos que usted conoce para garantizar la seguridad en un SGBD.
- f) Diga si son verdaderas o falsas las siguientes afirmaciones. Justifique todas sus respuestas.
- 1) Cuando varias transacciones accedan concurrentemente un mismo dato con el propósito de modificarlo puede producirse un "abrazo fatal" y, para evitar este, se utiliza la técnica de cierre o bloqueo exclusivo y la devolución de los recursos asignados antes de solicitar otros nuevos.
 - 2) La concurrencia en un sistema de bases de datos consiste en garantizar el acceso múltiple y simultáneo a la base de datos, de modo que los datos puedan ser compartidos por diferentes usuarios a la vez.
 - 3) Para controlar la concurrencia en los sistemas de bases de datos, el mecanismo de cierre o bloqueo permite que, si una transacción tiene un cierre compartido sobre un registro, otra transacción pueda adquirir un cierre exclusivo sobre ese mismo registro.
 - 4) Con el objetivo de garantizar la recuperación de la base de datos basta con guardar un diario de las operaciones realizadas sobre los datos almacenados, ya que dicho diario sustituye a los vaciados (o volcados o

backups) de la información almacenada, por lo que éstos no se consideran necesarios.

- 5) Una transacción debería ser una operación indivisible pero, como esto no se puede garantizar, el sistema de gestión de bases de datos tiene que brindar métodos que permitan deshacer una transacción que comenzó pero que no concluyó antes del momento de una falla del sistema.
- 6) Uno de los mecanismos de seguridad utilizados es el acceso restringido a los datos a través de vistas.
- 7) Para garantizar la seguridad de las bases de datos no basta con realizar la identificación de cada usuario, sino que también es necesario controlar las operaciones a realizar sobre determinados datos.

g) Defina los siguientes conceptos:

- 1) Procedimiento almacenado
- 2) Disparador o trigger
- 3) Identidad del objeto
- 4) Persistencia
- 5) Base de datos orientada a objetos
- 6) Base de datos distribuida
- 7) Computación Cliente/Servidor

Resumen

- 📖 Una transacción es una secuencia de operaciones sobre la base de datos que transforman un estado consistente de la base de datos en otro, sin que se preserve necesariamente la consistencia en los momentos intermedios. Una transacción se ejecuta totalmente o se cancela completamente.
- 📖 Recuperación es la capacidad de un SGBD para recuperarse ante una falla. Un SGBD permite la concurrencia si se ejecutan varias transacciones simultáneamente. Seguridad, se refiere a garantizar que no se realicen accesos no autorizados a los datos.
- 📖 El SGBD mantiene un "diario" en cinta magnética o disco en el que registra detalles de todas las operaciones de actualización, en particular, los valores anterior y posterior del artículo actualizado.
- 📖 Existen tres tipos de fallas en un SGBD que maneja transacciones:
 - Local: afecta sólo la T en la cual ocurre.
 - Del sistema: afecta todas las T en desarrollo, pero no daña físicamente la base de datos.
 - De los medios: causa daño físico a la base de datos o a alguna porción de ella.
- 📖 Para la recuperación de un SGBD se usan diferentes mecanismos, entre ellos: el diario, los puntos de chequeo y los volcados o copias de seguridad (*backups*) de la base de datos.
- 📖 Uno de los mecanismos usados para garantizar la sincronización de las transacciones concurrentes es el cierre o bloqueo. Existe varios tipos de cierres, entre ellos: cierre exclusivo y cierre compartido. Si una transacción A adquiere un cierre compartido sobre un recurso R, una transacción B puede adquirir también uno compartido sobre ese recurso, pero no uno exclusivo, hasta que A lo libere. Si una transacción A adquiere un cierre exclusivo sobre un recurso R, una transacción B no puede adquirir ningún cierre sobre ese recurso, hasta que A lo libere.
- 📖 El abrazo fatal es una situación en la que dos o más T están simultáneamente en un estado de espera, cada una esperando porque una de las otras libere un recurso que ella necesita para proseguir. Para salir de un abrazo fatal se selecciona una de las T involucradas como "víctima" y se deshace lo realizado por ella, lo que hace que se liberen sus recursos y que otra T pueda proseguir.
- 📖 Para evitar la ocurrencia de un abrazo fatal se puede exigir que una transacción solicite de una vez todos los recursos que requiere para una operación dada pero, antes, deberá liberar todos los recursos sobre los que tenía acceso permitido.

📖 Para garantizar la seguridad se emplean diferentes mecanismos, entre ellos: las vistas, el subsistema de autorización y el cifrado de los datos.

📖 Se le han hecho diferentes extensiones al modelo relacional, algunas de ellas para lograr satisfacer los nuevos requisitos que imponen las actuales aplicaciones, entre ellas:

- Reglas para llaves extranjeras
- Nuevos operadores algebraicos
- **M. Relacional Extendido (RM/T)**
- Relaciones Anidadas
- Sistemas basados en la lógica
- Procedimientos almacenados, Disparadores o Gatillos (triggers) y Reglas

📖 Al definir en qué consiste un sistema de bases de datos distribuidos, es preciso mencionar las siguientes características:

- Está compuesto por varios nodos conectados por canales de comunicación
- Cada nodo puede procesar programas y almacenar datos
- Todos los nodos no tienen que tener igual configuración
- Los nodos son equivalentes y deciden autónomamente si procesan una tarea o la pasan a otro nodo
- El sistema es transparente para los usuarios

Un **sistema distribuido** puede verse como cualquier sistema que comprenda múltiples nodos interconectados en algún tipo de red de comunicación, en el cual un usuario (usuario final o programador de aplicaciones) en cualquier nodo puede acceder datos almacenados en cualquier nodo.

📖 La distribución de los datos en un SBDD, según la **redundancia**, puede ser de tres tipos:

- Redundancia total
- Cero redundancia
- Redundancia parcial

📖 La distribución de los datos en un SBDD, según la **fragmentación**, puede ser de tres tipos:

- Vertical, mediante proyecciones.
- Horizontal, mediante selecciones.
- Combinación de ambas

📖 La **computación C/S** se usa para describir aplicaciones donde dos o más procesos lógicos separados cooperan para resolver una tarea. Los procesos se llaman cliente y servidor. En una aplicación con esta filosofía C/S, un componente de la aplicación que se ejecuta en un sistema de cómputo (cliente) solicita un servicio a un componente que se ejecuta en otro sistema de cómputo (servidor).

Los componentes de una aplicación C/S son: la aplicación, el SGBD y la base de datos. Dependiendo de dónde residan estos componentes, se obtienen diferentes configuraciones de base de datos C/S, entre ellas las denominadas son centralizada, cliente/servidor y distribuida.

Si desea saber más acerca de estos temas, puede consultar las siguientes páginas.

http://ocw.uc3m.es/informatica/disenio-y-administracion-de-bases-de-datos/teoria/Tema4_4%28Administracion_Recuperacion%29.pdf

En esta página web hallará conceptos complementarios sobre la recuperación de los datos.

<http://ict.udlap.mx/people/carlos/is341/bases10.html>
<http://www.lsi.us.es/docencia/get.php?id=3162>

En estas páginas web obtendrá conceptos complementarios sobre la concurrencia a una base de datos.

[http://msdn.microsoft.com/es-es/library/ms187648\(SQL.90\).aspx](http://msdn.microsoft.com/es-es/library/ms187648(SQL.90).aspx)

Es esta página web encontrará las consideraciones de seguridad para bases de datos y aplicaciones de bases de datos, según Microsoft.

<http://mit.ocw.universia.net/15.565J/NR/rdonlyres/Sloan-School-of-Management/15-565JIntegrating-eSystems---Global-Information-SystemsSpring20/72F605CB-7CD4-45D8-942B-4E6DFC474D35/0/lecture13.pdf>

Es esta página web encontrará información complementaria sobre base de datos centralizadas y distribuidas, así como su problemática asociada.

<http://www.oei.eui.upm.es/Asignaturas/BD/DYOB/DISTRIBUIDAS.pdf>

Es esta página web encontrará conceptos complementarios de bases de datos distribuidas.