

Capítulo X

Normalización para las bases de datos relacionales

10.1 INTRODUCCIÓN

Cada relación consiste en varios atributos. Hasta ahora los atributos fueron agrupados en distintos capítulos del libro, esencialmente usando sentido común. Sin embargo, existen normas "formales" para determinar que una agrupación de atributos en una tabla sea mejor que otra.

Empezaremos recorriendo los distintos conceptos que debemos tener en cuenta para poder realizar buenos esquemas de relación con respecto a su agrupamiento en las distintas tablas que conforman la base de datos.

10.2 ANÁLISIS INFORMAL DE LAS RELACIONES

Como pautas informales de diseño encontramos:

- La relación que existe entre los valores de los atributos de una tupla.
- Evitar los valores nulos en las tuplas.

10.2.1 Significado de los atributos en una relación

Consideremos las tres tablas que se definieron en el capítulo II, Tabla 1: PROVEEDORES, Tabla 2: PRODUCTOS y la Tabla 3: PROV-PROD.

En la relación PROVEEDORES cada tupla representa un proveedor con los valores para su identificación, nombre y dirección. En la relación PRODUCTOS cada tupla representa un artículo con sus descripciones de número, nombre, precio, tamaño y ubicación. Estas dos relaciones representan entidades bien claras y definidas.

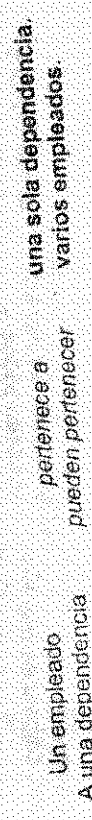
La tabla 3: PROV-PROD contiene un atributo que la relaciona con la tabla 1 (número de proveedor), otro atributo que la relaciona con la tabla 2 (número de artículo), y un tercer atributo que surge de dichas relaciones (cantidad de determinado producto provisto por determinado proveedor).

Quizás son éstos los casos más difíciles de organizar. Para ello, vamos a definir ciertas reglas para optimizar las relaciones:

- 1- Si la entidad A contiene tuplas que se relacionan de a una con más de una tupla de la entidad B (o sea una relación de uno a muchos), la relación co-

No hizo falta la creación de una tercer tabla, ya que la vinculación se produce cuando se agrega en la entidad B el atributo CODPROV(el cual es clave ajena referenciando a la entidad A).

2- Tenemos una tabla-relación que contiene los datos de los empleados de una empresa (entidad A). Por otro lado, tenemos la tabla que contiene los datos de las dependencias de esa empresa. Sabemos que un empleado no puede pertenecer a más de una dependencia. Esto me indica la relación:



Queda claro entonces que en la tabla EMPLEADOS (entidad A) habrá una "mano" (atributo de relación: NRODEP) que vincule esta tabla con la otra.

Entidad A: EMPLEADO

LEGAJO	NOMBRE	NRODEP
1010	Valdez, Juan	01
1012	Torres, José	02
1015	Bacchi, Nancy	03
1020	Campolongo, Bruno	01

Entidad B: DEPENDENCIA

NRO-DEP	NOMBRE-DEP
01	Sistemas
02	Contaduría
03	Personal
04	Contratos

En este caso tampoco hizo falta la creación de una tercera tabla, ya que la vinculación se produce cuando se agrega en la entidad A el atributo NRODEP (el cual es clave ajena referenciando a la entidad B).

respondiente a "muchos"(en este caso la entidad B) contendrá un atributo que la relacione con la otra.

2- Si la entidad A contiene más de una tupla que se relaciona con tuplas de la entidad B de a una (o sea una relación de muchos a uno), sucede lo mismo que en el caso anterior: en la relación de "muchos" (en este caso la entidad A) contendrá un atributo que la relacione con la otra.

3- Si la entidad A contiene más de una tupla que se relaciona con más de una tupla de la entidad B (o sea una relación de muchos a muchos), se justifica la creación de una tercer tabla que interrelacione las otras dos.

Ilustraremos estos tres casos de una manera simple y clara:

1- Tenemos una tabla-relación que contiene los datos de las provincias disponibles para residir en una convención laboral. Por otro lado, tenemos la tabla que contiene datos de becados que concurrirán a dicha convención. Es lógico, que cada becado sólo puede concurrir a una sola provincia para la convención. Por lo tanto:



Entidad A: PROVINCIAS

COD-PROV	DESCRIPCION
01	La Pampa
02	Corrientes
03	Neuquén
04	La Rioja

Entidad B: BECADOS

IDENTIF	NOMBRE	COD-PROV
01	Farina, Carmen	01
02	Farina, Olga	04
03	Sanchez, Andrés	02
04	Perez, Ana	01
05	Gómez, Andrea	02
06	Blanco, Ariel	02

3- Contamos con la entidad A: PACIENTES, que contiene más de una tupla que se relaciona con más de una tupla de la entidad B: MEDICAMENTOS (o sea una relación de muchos a muchos), se justifica la creación de una tercera tabla C: PAC-MED, que interrelacione las otras dos.

Esto me indica la relación:



Entidad A: PACIENTES

NUMPAC	NOMBRE	EDAD
21010	Valdez, Jorge	51
21012	Torres, José María	46
31015	Bacchi, Nancy	34
51020	Campolongo, Bruno	13
52340	Katounas, Ariel	21
67630	Banegas, Carolina	20
78932	Pérez, Leonardo	22

Entidad B: MEDICAMENTOS

NROMED	DESCRIPCION
01	Calmanite
02	Colirio
03	Tranquilizante
04	Cicatrizante
05	Espasmódico
06	Antiestrés
07	Antirreumático
08	Antihemorroidal
09	Sedante
10	Anticoagulante
11	Antiséptico

Interrelación C: PAC-MED

NUMPAC	NROMED	DOSISDIA
21010	01	2
21010	03	1
21012	02	5
21012	06	1
31015	07	2
31015	06	2
31015	01	2
51020	05	3
51020	01	2
67630	01	4
67630	09	1
78932	10	2

Vemos que el paciente 21010, Valdez Jorge, consume más de un medicamento (01, calmante y 03, tranquilizante). Y por ejemplo, el medicamento 01 calmante, es consumido por más de un paciente. Para organizar esto, debemos crear una tercera tabla, donde se produce la vinculación de ambas entidades.

10.2.2 Valores nulos de las tuplas

En una relación se agrupan determinada cantidad de atributos que a veces conforman una relación demasiado "grande". Si hubiera atributos que no se aplican a la mayoría de las tuplas de esta relación, aparecerán gran número de nulos. Por eso, evitaremos incluir en una relación base, atributos cuyos valores puedan ser nulos; y de existir, que sea en su minoría.

Por ejemplo:

Si sólo el 10 % de los alumnos trabajan, no se justifica incluir un atributo CUIT ó CUIL; más bien se podría crear una relación ALU-TRAB que contenga sólo tuplas con los alumnos que están empleados:

ALU-TRAB (LEGAJO, CUIT-CUIL);

10.3 DEPENDENCIA FUNCIONAL

Dada una relación S, el atributo X de S depende funcionalmente del atributo Y de S si y sólo si un solo valor X en R está asociado a cada valor Y en R (en cualquier momento dado). Los atributos X y Y pueden ser compuestos.

De esta manera describo la dependencia funcional muy matemáticamente, esto resulta para mi gusto, exacta pero poco clara. Por eso, voy a tratar de ser más específica con esta definición:

Un dominio es funcional completo dependiente si es dependiente de toda la clave pero no de ningún dominio clave en forma individual.

Cuando se explique con ejemplos las formas normales, podrán clarificar estos conceptos aplicados directamente en dichos ejercicios.

10.4 FORMAS NORMALES

En un principio, Edgar Codd propuso tres formas normales, las cuales se conocieron como primera (1FN), segunda (2FN) y tercera (3FN) formas normales. Luego Boyce y Codd propusieron una definición más estricta de 3FN, a la que se dio el nombre de Forma Normal de Boyce-Codd.

Todas estas formas se definen bajo las restricciones de las dependencias funcionales entre los atributos de una relación.

Existe también la 4FN y la 5FN, debidas a R. Fagin, pero las mismas se basan en dependencias multivaluadas y dependencias de reunión, a las cuales no suele llegarse, por lo que sólo se definirán.

Es importante destacar que no es necesario normalizar hasta la 5FN para considerar las relaciones de manera óptima. Podrán quedarse en formas inferiores por razones de rendimiento; esto quedará mejor entendido cuando se explique cada forma en detalle.

Bastará con llegar a poner las relaciones en tercera forma normal, lo cual supone una serie de ventajas:

- Se evitan anomalías en la inserción, borrado y modificación.
- Se facilita la extensión: si en un futuro se llevan a cabo ampliaciones, se tendrán menos cambios en la estructura de la base de datos y por lo tanto en los programas de aplicación.

La normalización de los datos consiste en descomponer las relaciones distribuyendo sus atributos en relaciones más pequeñas satisfaciendo un cierto con-

junto de restricciones. O sea, es el proceso de producción de grupos óptimos de atributos en las relaciones.

10.4.1 Primera forma normal (1FN)

Quiero aclarar que, decir que una relación está normalizada y decir que está normalizada en 1FN, es exactamente lo mismo. La otras formas normales son complementarias de ésta.

Se dice que una relación está en primera forma normal (1FN) si y sólo si todos sus dominios subyacentes contienen sólo valores atómicos.

Esto significa que cada atributo tienen un único valor para una ocurrencia de la entidad.

Dicho en otras palabras, se hace hincapié en la eliminación de grupos repetitivos.

Ejemplo :

EMPLEADO

LEGAJO	NOMBRE	COD-IDIOMA	NOMBRE-IDIOMA	NIVEL-IDIOMA	SECCIÓN
120	Patricia	01	Inglés	B	Cómputos
132	Juana	03	Portugués	A	Contable
132	Juana	04	Italiano	B	Contable
138	José	01	Inglés	C	Ventas
138	José	02	Francés	A	Ventas
144	Bruno	01	Inglés	A	Administ.

Los atributos referidos al "idioma" del empleado son los que provocan la repetición de los legajos. Entonces, eliminar los grupos repetitivos significa "llevarme a otra relación los atributos causantes de tal iteración".

Así quedan dos relaciones entidades dispuestas de la siguiente manera:

EMPLEADO (LEGAJO, NOMBRE, SECCIÓN)

Con los atributos propios del empleado.

IDIOMA (COD-IDIOMA, NOMBRE-IDIOMA)

Con los atributos propios del idioma.

Y una interrelación EMP-IDIOMA para indicar el NIVEL de cada empleado en cada idioma.

EMP-IDIOMA (LEGAJO, COD-IDIOMA, NIVEL-IDIOMA)

Los atributos subrayados indican las claves primarias, para identificar las tuplas unívocamente.

10.4.2 Segunda forma normal (2FN)

Una relación está en segunda forma normal (2FN) si y sólo si está en 1FN y todos los atributos no clave dependen por completo de la clave primaria (identificador); no existen dependencias parciales.

Ejemplo 1:

Volviendo al ejemplo anterior, analizamos la 2FN en la tabla EMP-IDIOMA que tiene la clave compuesta.

EMP-IDIOMA (LEGAJO, COD-IDIOMA, NIVEL-IDIOMA)

NIVEL-IDIOMA es un atributo no clave que tiene dependencia funcional completa (o plena) con toda la clave: el nivel del idioma depende del empleado que lo tiene y del idioma que sabe. Esto significa que esta relación está en 2FN.

Analizar la 2FN en las tablas que tienen claves primarias simples no tiene sentido, ya que siempre se va a cumplir las condiciones necesarias para la 2FN, o sea, que todos los atributos no-claves son dependientes funcional completos de la clave completa (es única).

Ejemplo 2:

Planteamos otro caso:

EMPLE (LEGAJO, COD-IDIOMA, NIV-IDIOMA, NIVEL-DE-ESTUDIO)

La relación se encuentra en 1FN porque la tupla se identifica en forma única, pero no se cumple la 2FN, ya que el atributo NIVEL-DE-ESTUDIO depende del empleado independientemente del idioma que sepa. Por lo tanto, ese atributo debe desaparecer de esta relación e incluirse en la relación entidad del empleado.

Las tablas correctas quedan:

EMPLEADO (LEGAJO, NOMBRE, SECCIÓN, NIVEL-DE-ESTUDIO)

Con los atributos propios del empleado.

IDIOMA (COD-IDIOMA, NOMBRE IDIOMA)

Con los atributos propios del idioma.

EMP-IDIOMA (LEGAJO, COD-IDIOMA, NIVEL-IDIOMA)

Con los atributos de la interrelación.

10.4.3 Tercera forma normal (3FN)

Una relación está en tercera forma normal (3FN) si y sólo si está en 2FN y los atributos no clave son mutuamente independientes entre sí. Esto significa que los valores de los atributos dependen sólo de la clave y no existen dependencias parciales con respecto a elementos que no sean clave.

Ejemplo:

EMPLEADO (LEGAJO, NOMBRE, NRO-SECCION, OFIC-SECCION)

Para la 3FN, además de estar en 1FN y 2FN ningún dominio no-clave debe mantener dependencia funcional completa con otro dominio no-clave.

NOMBRE y SECCION no tienen ninguna dependencia. NOMBRE y OFICINA-SECCION no tienen ninguna dependencia. NRO-SECCION y OFIC-SECCION dependen entre sí, ya que si cambia de sección, cambia de oficina. Por este caso, la relación no se encuentra en 3FN.

Se debe quitar de la tabla el atributo OFIC-SECCION, y armar otra donde no interfiera en la normalización de tercera forma.

EMPLEADO (LEGAJO, NOMBRE, NRO-SECCION)

SECCION (NRO-SECCION, OFICINA-SECCION)

Cabe señalar, que si la relación contiene un solo campo no-clave, el análisis de la 3FN sería innecesario (está automáticamente en 3FN). Por ejemplo:

IDIOMA (COD-IDIOMA, NOMBRE-IDIOMA)

Esta tabla está en 1FN, en 2FN y en 3FN respectivamente.

10.4.4 FORMA NORMAL DE BOYCE-CODD (FNBC)

Una relación está en forma Boyce-Codd si y sólo si todo determinante es una clave candidata.

Se define como determinante a cualquier atributo o grupo de atributos del cual depende funcionalmente completo algún otro atributo.

Ejemplo:

PROVEEDORES (NUMERO, NOMBRE, DOMICILIO, LOCALIDAD)

La relación PROVEEDORES está en 3FN y también está en BCNF ya que se cumple la condición de que los únicos determinantes son las claves candidatas. Los atributos NUMERO Y NOMBRE son claves candidatas (ya que todos los proveedores tienen números y nombres únicos).

10.5 FORMAS ESPECIALES (4FN Y 5FN)

La cuarta forma normal (4FN) consigue relaciones en las que no hay dependencias multivaluadas. Se dice que Y tienen dependencia multivaluada con respecto a X si a cada valor de X le corresponde un conjunto de valores de Y bien definido.

Esto significa que la dependencia funcional es un caso particular de la dependencia multivaluada, por lo que toda relación que esté en 4FN también estará en 3FN.

La quinta forma normal (5FN) se refiere a aquellos casos en los que la relación no pueda descomponerse en dos relaciones sin pérdida de información, pero pueda dividirse en tres o más.

Toda relación que esté en 5FN la está en 4FN; sin embargo, el procedimiento para pasar de una a otra es bastante confuso, por lo que la 5FN no suele nunca alcanzarse.

10.6 EJERCICIOS

10.6.1 ¿Por qué no se considera bueno tener muchos atributos nulos en una relación?

10.6.2. Comente las pautas informales para diseñar esquemas de relación.

10.6.3 ¿Qué es una dependencia funcional? ¿Quién las especifica?

10.6.4 ¿Por qué en general se considera buena a una relación que está en 3FN?

10.6.5 ¿Por qué la FNBC es mejor que la 3FN?

10.6.6 Tenemos los siguientes requerimientos para una base de datos universitaria con que se manejan las boletas de notas de los estudiantes. Para cada alumno consta su nombre, número de legajo, dirección, teléfono, otro teléfono opcional, fecha de nacimiento, sexo, departamento de carrera, departamento de especialidad.

Para cada departamento, figura su respectivo código con su nombre.

Para cada curso: el nombre del curso, su código, número de horas semanales, nivel, profesor a cargo, año.

Las boletas se conforman del número de legajo, curso, nota y fecha.

a) Diseñe un conjunto de relaciones normalizadas.

b) Lleve todas las relaciones hasta 2FN.

c) Especifique los atributos clave de cada relación.

10.6.7 (Ejercicio extraído de Análisis Estructurado de Sistemas, de Chris Gane y Trish Sarson, editorial "El Ateneo".)

Se trata de hacer una versión normalizada de la estructura que describe una compra de libros a una Compañía XZ.

Un pedido consiste en el nombre del cliente, la fecha del pedido, el ISBN (código internacional único) del libro pedido, el título, el autor, la cantidad que ha sido pedida, y el importe total del pedido para un libro determinado.

(No olvidar que "normalizada" es sinónimo de 1FN).

10.6.8 Tenemos el siguiente conjunto de datos que se va a grabar en una BD de personal de una Compañía:

- La compañía tiene un conjunto de departamentos.
- Cada departamento tiene un conjunto de empleados, un conjunto de proyectos y un conjunto de oficinas.
- Cada empleado tiene una historia de empleos. Para cada empleo, tiene una historia de salarios.
- Cada oficina tiene un conjunto de teléfonos.

La base de datos debe contener la siguiente información:

- Para cada departamento: Número de departamento (único), presupuesto de cada departamento y nro. de empleado del gerente del departamento (único).
- Para cada empleado: Nro. de empleado (único), nro. de proyecto actual, nro. de oficina y nro. de teléfono; además el título de cada trabajo que ha tenido el empleado, junto con la fecha y el salario para cada salario distinto recibido en ese trabajo.
- Para cada proyecto: Nro. de proyecto (único) y presupuesto del proyecto.
- Para cada oficina: Nro. de oficina (único), área en m² y nros. de todos los teléfonos de esa oficina.

Diseñar un conjunto apropiado de relaciones normalizadas hasta 3FN.

10.6.9 (Ejercicio extraído de *Análisis Estructurado de Sistemas*, de Chris Gane y Trish Sarson, editorial "El Ateneo".)
Supongamos que tenemos una relación para proveer la asignación de los empleados temporarios a los proyectos, como la siguiente:

PROY-ASIGN (NRO-EMP, TE, SUELDO-HORA, NRO-PROY, FECHA-FINAL)

- ¿Está en 1FN?
- ¿Está en 2FN?
- ¿Está en 3FN?

Capítulo XI

Álgebra relacional

11.1 DEFINICIÓN

El álgebra relacional es un conjunto de operaciones de alto nivel que sirven para manipular relaciones. Se puede, por ejemplo, seleccionar tuplas de relaciones individuales. El resultado de cada operación es una nueva relación.

Para el manejo de base de datos relacionales se define una cantidad de operadores. La intención fundamental del álgebra es ayudar a escribir expresiones.

La siguiente lista indica algunas posibles aplicaciones de tales expresiones:

- Definir los datos que se van a extraer como resultado de una recuperación.
- Definir los datos por insertar, modificar o eliminar como resultado de una operación de actualización.
- Definir datos virtuales, de instantánea, derechos de acceso, etc.

11.2 OPERACIONES

Las operaciones en cuestión son la unión, intersección, diferencia, producto cartesiano, - como operadores tradicionales - y la restricción (conocida también como selección), proyección, reunión (natural y theta) y división, como operadores relacionales específicos para base de datos relacionales.

La función de cada uno es la siguiente:

UNIÓN : Considerando dos relaciones, se construye una relación resultado, formada por todas las filas que aparecen en cualquiera de las dos relaciones especificadas.

INTERSECCIÓN: Considerando dos relaciones, se construye una relación formada por aquellas filas que aparezcan en las dos relaciones especificadas.

DIFERENCIA: Considerando dos relaciones, se construye una relación resultado formada por todas las filas de la primera relación **que no aparecen** en la segunda relación.

PRODUCTO: Considerando dos relaciones, se construye una relación que contiene **todas las combinaciones posibles de filas**, una de cada una de las dos relaciones.

RESTRICCIÓN: Extrae filas de una relación dada mediante una condición especificada (también llamada **SELECCIÓN**).

PROYECCIÓN: Extrae columnas de una relación dada.

REUNIÓN: Realiza las mismas acciones que el producto cartesiano, excepto que las dos filas participantes en una **combinación dada cumplen alguna condición especificada**.

DIVISIÓN: Trabaja con dos relaciones, una binaria y una unaria, y construye una relación formada por todos los valores de un atributo de la relación binaria que concuerdan **con todos los valores** en la relación unaria.

Recordamos que cada una de las operaciones es otra relación, y considerando que es un objeto del mismo tipo que los operandos (relaciones), el resultado de una operación puede convertirse en operando de otra.

Así, por ejemplo, será factible sacar la proyección de una unión o una reunión de dos restricciones.

Dicho de otra manera, es posible escribir expresiones relacionales anidadas.

Antes de avanzar en los conceptos, es necesario aclarar que en el álgebra relacional no es factible trabajar con atributos de igual nombre. Para solucionar esto introduciremos un nuevo operador, **RENAME**, cuyo propósito es en esencia cambiar el nombre de los atributos dentro de una relación.

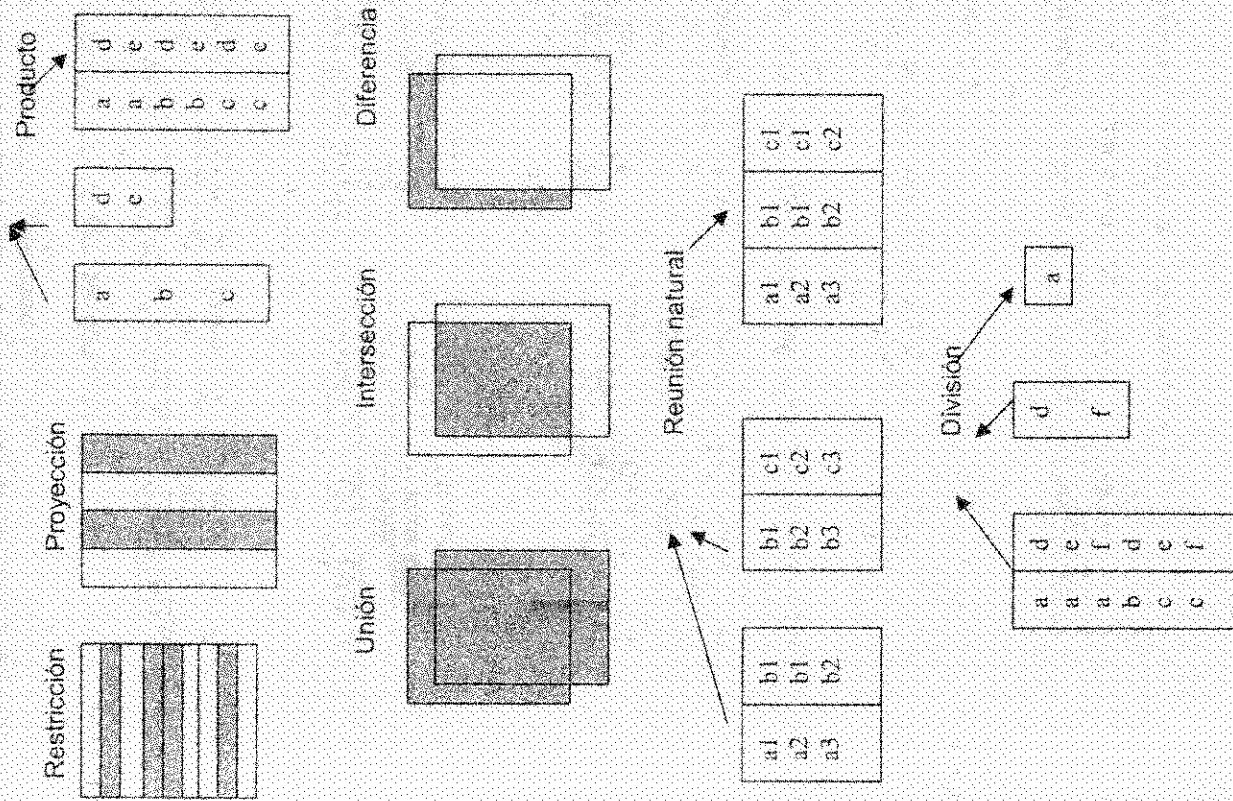
Por ejemplo:

PROVEEDORES RENAME LOCALIDAD AS PLOC

Sólo cambiará el nombre de **LOCALIDAD** a **SLOC**.

Ya estamos en condiciones de mencionar una sintaxis concreta para las operaciones del álgebra relacional.

Para cualquier expresión del álgebra relacional, existe una expresión de SQL equivalente en lo semántico.
La siguiente figura representa los diferentes operadores:



11.3 OPERACIONES TRADICIONALES

Ya se han definidos los siguientes operadores, pasaremos a explicar con un ejemplo ilustrativo cada uno de ellos.

- Unión: planteamos un ejemplo donde X:

PNRO	PNOMBRE	PRECIO	TAMAÑO	LOCALIDAD
001	Talco	5	Chico	Capital
002	Talco	7	Mediano	Capital

y Z:

PNRO	PNOMBRE	PRECIO	TAMAÑO	LOCALIDAD
001	Talco	5	Chico	Capital
003	Crema	8	Grande	Ramos Mejía
005	Esmalte	1,20	Normal	Chacarita

X UNION Z

PNRO	PNOMBRE	PRECIO	TAMAÑO	LOCALIDAD
001	Talco	5	Chico	Capital
002	Talco	7	Mediano	Capital
003	Crema	8	Grande	Ramos Mejía
005	Esmalte	1,20	Normal	Chacarita

Vemos que está formado por todas las tuplas pertenecientes a X y Z; elimina las tuplas repetidas.

- Intersección: con los mismos X y Z.

X INTERSECT Z

PNRO	PNOMBRE	PRECIO	TAMAÑO	LOCALIDAD
001	Talco	5	Chico	Capital

Sólo se muestran las tuplas que aparecen en X y Z.

- Diferencia: seguimos con X y Z.

X MINUS Z

PNRO	PNOMBRE	PRECIO	TAMAÑO	LOCALIDAD
002	Talco	7	Mediano	Capital

Z MINUS X

PNRO	PNOMBRE	PRECIO	TAMAÑO	LOCALIDAD
003	Crema	8	Grande	Ramos Mejía
005	Esmalte	1,20	Normal	Chacarita

Adviértase que X - Z dará diferente resultado que Z - X.

- Producto cartesiano: cambiando los datos iniciales para X y Z respectivamente.

NUMERO
101
102
103

PNRO
101
102
103

X TIMES Z

NUMERO	PNRO
101	001
101	002
101	003
102	001
102	002
102	003
103	001
103	002
103	003

11.4 OPERACIONES ESPECIALES

- **Restricción:** la expresión condicional en la cláusula WHERE de una restricción está formada por combinaciones booleanas arbitrarias de comparaciones simples.

Ejemplo: obtener un detalle completo de todos los proveedores que viven en Capital.

PROVEEDORES WHERE LOCALIDAD = 'Capital' ;

El resultado muestra las siguientes tuplas:

NUMERO	NOMBRE	DOMICILIO	LOCALIDAD
101	Gómez	Nazca 920	Capital
102	Pérez	Argerich 1030	Capital

- **Proyección:** El operador de proyección produce un subconjunto "vertical" de la relación dada y la eliminación de las tuplas repetidas dentro de los atributos seleccionados.

Ejemplo: obtener la localidad de todos los proveedores.

PROVEEDORES [LOCALIDAD]

LOCALIDAD
Capital
Ramos Mejía
Avellaneda

También se pueden combinar restricción y proyección:

Ejemplo: Obtener los números de proveedores que viven en Avellaneda

(PROVEEDORES WHERE LOCALIDAD = 'Avellaneda') [NUMERO]

NUMERO
104

- **Reunión natural:** tal como se ha definido, es tanto asociativa como conmutativa. Se usa la palabra JOIN para expresar la consigna. Se debe tener en cuenta que se está reuniendo dos relaciones especificadas con alguna condición de igualdad.

Ejemplo: Obtener todos los proveedores y productos que por lo menos se localicen en una localidad.

PROVEEDORES JOIN PRODUCTOS es equivalente a
PRODUCTOS JOIN PROVEEDORES

Numero	Nombre	Domicilio	Localidad	Phno	Prontibre	Precio	Tamaño
101	Gómez	Nazca 920	Capital	001	Talco	5	Chico
101	Gómez	Nazca 920	Capital	002	Talco	7	Mediano
102	Pérez	Argerich 1	Capital	001	Talco	5	Chico
102	Pérez	Argerich 1	Capital	002	Talco	7	Mediano
103	Vázquez	Sarmiento	Ramos M.	003	Crema	8	Grande
104	López	Alsina 720	Avellaneda	004	Cepillo	2	Grande

- **Reunión theta:** se adecua para aquellas ocasiones en las cuales necesitamos juntar dos relaciones con alguna condición diferente a la igualdad. La reunión theta de la relación A según el atributo X con la relación B según el atributo Y se define como:

(A TIMES B) WHERE X THETA Y

Los atributos X y Y deberán estar definidos sobre el mismo dominio, y la operación theta debe ser aplicable a ese dominio.

Ejemplo: se desea calcular la reunión mayor que de la relación PROVEEDORES según LOCALIDAD con respecto a la relación PRODUCTOS. Usaremos además el operador RENAME (cambio de nombre de atributo):

((PROVEEDOR RENAME LOCALIDAD AS PLOCALIDAD)
TIMES
PRODUCTOS)
WHERE PLOCALIDAD > LOCALIDAD

Nótese que la reunión theta siempre es equivalente a obtener el producto cartesiano de las dos relaciones (con modificaciones apropiadas de los nombres

de los atributos, si es necesario), y después realizar una restricción apropiada sobre el resultado.

En este caso particular del ejemplo, el resultado es:

Num	Nombre	Domicilio	Plocalidad	Ppro	Pprom	Pprecio	Tamaño	Localid
101	Gómez	Nazca 920	Capital	004	Cepillo	2	Grande	Avellan
102	Pérez	Argentin 1	Capital	004	Cepillo	2	Grande	Avellan
103	Vázquez	Sarmiento	Ramos M.	001	Talco	5	Chico	Capital
103	Vázquez	Sarmiento	Ramos M.	002	Talco	7	Median	Capital
103	Vázquez	Sarmiento	Ramos M.	004	Cepillo	2	Grande	Avellan
103	Vázquez	Sarmiento	Ramos M.	005	Esmalte	1,20	Normal	Chacarri

- **División:** El operador DIVIDEBY (dividir entre) es útil para consultas donde está implicada la palabra "todos" (por ejemplo, obtener los proveedores que suministran todos los productos)

Ejemplo: Si el dividendo es una tabla con algunas combinaciones de NUMERO con PNRO:

DIVIDENDO

NUMERO	PNRO
101	001
101	002
101	003
102	001
102	002
102	004
103	001
103	002
103	003
104	001
104	003
104	005

Y el divisor otra tabla con los siguientes datos:

DIVISOR

PNRO
001
002

En el resultado de: DIVIDENDO DIVIDEBY DIVISORD estoy averiguando qué proveedores suministran los productos 001 y 002:

NUMERO
101
102
103

- **Asignación relacional:** el propósito de esta operación es poder "recordar" el valor de alguna expresión algebraica. Cuando necesito obtener un solo valor como resultado se utilizará una variable para guardar el valor recogido.

Ejemplo 1: obtener el nombre del producto 001

R1 ← PRODUCTOS WHERE PNRO = 001 [PNOMBRE]

11.5 SINTAXIS PARA EL ÁLGEBRA RELACIONAL

Según la notación presentada en Sistemas de Bases de Datos - Conceptos Fundamentales -- segunda edición, Elmasri/Navathe:

Operación	(o RESTRINGIR)	Notación
SELECCIONAR	σ «condición de selección» (R)	
PROYECTAR	π «lista de atributos» (R)	
REUNIÓN	\bowtie (según el tipo de reunión)	
UNIÓN	$R_1 \cup R_2$	
INTERSECCIÓN	$R_1 \cap R_2$	
DIFERENCIA	$R_1 - R_2$	
PRODUCTO CARTESIANO	$R_1 \times R_2$	
DIVISIÓN	$R_1 \div R_2$	

Según la notación utilizada en Introducción a los Sistemas de Bases de Datos -- quinta edición, C.J.Date:

- 1) Presenta una gramática BNF (fórmulas de Backus Naur), que considero particularmente compleja y he decidido no trabajar con la misma.
- 2) Presenta una forma de oración similar al inglés ordinario, más fácil de entender, el cual vamos a adoptar como herramienta de trabajo.

11.6 UTILIZACIÓN DE OTRA NOTACIÓN

Ahora vamos a resolver las mismas operaciones con otra notación.

- **Restricción:** trabajamos con la siguiente relación:

EMPLEADO (NRO-EMP, NOM-EMP, DEP, SALARIO)

Ejemplo 1: seleccionar las tuplas de EMPLEADO que trabajan en la dependencia 1.

Se especifica: $\sigma_{DEP = 1}$ (EMPLEADO)

También se pueden combinar varias condiciones.

Ejemplo 2: seleccionar las tuplas de EMPLEADO que trabajan en la dependencia 2 y ganan más de 2000.

Se especifica: $\sigma_{DEP = 2 \text{ AND } SALARIO > 2000}$ (EMPLEADO)

- **Proyección:** seguimos con la misma relación.

Ejemplo 1: seleccionar el nombre y salario de todos los empleados.

Se especifica: $\pi_{NOM-EMP, SALARIO}$ (EMPLEADO)

Ejemplo 2: obtener el nombre y salario de todos los que trabajan en la dependencia 3.

Se especifica: $\pi_{NOM-EMP, SALARIO} (\sigma_{DEP = 3} (\text{EMPLEADO}))$

- **Unión:** continuamos con la misma relación.

Ejemplo: obtener los números de legajo de todos los empleados que trabajan en la dependencia 5 o que ganan menos de 1200.

Vamos a trabajar dejando cada resultado parcial guardado en tablas temporarias (simulando una asignación relacional)

EMP-DEP $\leftarrow \sigma_{DEP = 5}$ (EMPLEADO)

RES1 $\leftarrow \pi_{NRO-EMP}$ (EMP-DEP)

EMP-SAL $\leftarrow \sigma_{SALARIO < 1200}$ (EMPLEADO)

RES2 $\leftarrow \pi_{NRO-EMP}$ (EMP-SAL)

RESULTADO $\leftarrow RES1 \cup RES2$

- **Reunión natural:** Usamos distintos operadores de acuerdo con el tipo de reunión

= equirreunión (operador de comparación)

- reunión natural: es una equirreunión seguida de la eliminación de los atributos superfluos.

La definición de reunión natural exige que los dos atributos de reunión tengan el mismo nombre. De no ser así, primero se aplica el operador de cambio de nombre del atributo en cuestión.

Ejemplo: de acuerdo con las siguientes relaciones, obtener el nombre y la dirección de todos los empleados que trabajan para el departamento de "Investigación".

EMPLEADO (NRO-EMP, NOM-EMP, DIR, NDEP)

DPTO (NRO-DEP, DESCR)

DPTO-INVES $\leftarrow \sigma_{DESCR = \text{"Investigación"}}$ (DPTO)

EMP-DEP-INV $\leftarrow (DPTO-INVES \bowtie EMP)$ (EMP-DEP-INV)

RESULTADO $\leftarrow \pi_{NOM-EMP, DIR}$ (EMP-DEP-INV)

- **División**

Ejemplo: De acuerdo a las siguientes relaciones, obtener los nombres de los empleados que trabajan en todos los proyectos en los que trabaja "Bergo, Mercedes".

EMPLEADO (NRO-EMP, NOM-EMP, NDEP)

EMPL-TRAB (NUMEMP, NUMPROY)

MERCEDES $\leftarrow \sigma_{NOM-EMP = \text{"Bergo, Mercedes"}}$ (EMPLEADO)

NUM-MERC $\leftarrow \pi$ NUMPROY
(EMPL-TRAB * NUMEMP + NRO-EMP MERCEDES)

NUMRESPAR \leftarrow EMPL-TRAB / NUM-MERC

RESULTADO $\leftarrow \pi$ NOM-EMP (NUMRESPAR * EMPLEADO)

11.7 EJERCICIOS

Los ejercicios 8.7.1 al 8.7.7 están basados en ejemplos extraídos de Introducción a los Sistemas de bases de datos, Volumen I, Quinta edición, C.J. Date, Addison-Wesley Iberoamericana, 1993.

Los ejercicios presentados a continuación utilizan las relaciones: PROVEEDORES, PRODUCTOS Y PROV-PROD. Para facilitar su uso, se repiten sus respectivas estructuras:

PROVEEDORES (NUMERO, NOMBRE, DOMICILIO, LOCALIDAD)
PRODUCTOS (PNRO, PNOMBRE, PRECIO, TAMAÑO, LOCALIDAD)
PROV-PROD (NUMERO, PNRO, CANTIDAD)

Dar soluciones algebraicas a los siguientes ejercicios:

- 11.7.1 Obtener los detalles completos de todos los productos.
- 11.7.2 Obtener los detalles completos de todos los proveedores de Capital.
- 11.7.3 Obtener todos los envíos en los cuales la cantidad está entre 200 y 300 inclusive.
- 11.7.4 Obtener los números de los productos suministrados por algún proveedor de Avellaneda.
- 11.7.5 Obtener los números de los productos y localidades en los cuales la segunda letra del nombre de la localidad sea "a".
- 11.7.6 Obtener los precios de los productos provistos por el proveedor 102.
- 11.7.7 Construir una lista de todas las localidades en las cuales esté situado por lo menos un proveedor o un producto.

Plantear el siguiente ejercicio de acuerdo a las pautas que se consignan:

- 11.7.8 Una empresa de transporte tiene una flota de camiones, grúas, plumas, volquetes, transporte de contenedores y material de precisión. La empresa posee una amplia cartera de clientes (empresas a quienes les realiza sus traslados). Sus clientes se encuentran clasificados por diferentes rubros ej.: construcción, alimentación, automotriz, puertos, empresas químicas, en distintos puntos del país. Cada tipo de transporte puede atender cualquier servicio de clientes de su sector correspondiente a cualquier provincia.

- a) Obtener los códigos de grúas, modelo y patentes de las grúas que no operaran en el período analizado.
- b) Emitir una relación que obtenga para cada provincia los clientes que tiene.
- c) Obtener los diferentes rubros sobre los cuales se realizaron los traslados.

11.7.8 Nos manejamos con las siguientes tablas normalizadas:

PROV (COD-PROV, DESCR)
 CLIEN (COD-RUBRO, NCLIE, NOM, DOM, TE, COD-PROV)
 RUBROS (COD-RUBRO, DESCR)
 TRANSP (COD-TIPO, MODELO, PATENTE)
 FLOTILLA (COD-TIPO, DESCR)
 VIAJES (NVIAJE, PATENTE, NCLIE, COD-RUBRO, FECHA)

Capítulo XII

Proceso de diseño de bases de datos

12.1 INTRODUCCIÓN

Muchas organizaciones grandes opinan que la gestión de recursos de información (IRM) es decisiva para el éxito de la empresa.

Existen varias razones para avalar dichas opiniones, a saber:

- Varias funciones de las empresas están computarizadas, lo que aumenta la necesidad de contar con grandes volúmenes de datos permanentemente actualizados.

Si queremos encontrar un ejemplo de la vida cotidiana, podemos hacer mención a una memoria humana: está llena de datos (esa es la idea), de acuerdo a las necesidades se "accede a ellos" y se efectúa la actualización instantáneamente que se reciben los mismos.

- La complejidad de los datos y de las aplicaciones aumenta cada vez más, por lo que se hace necesario modelar y mantener vínculos (interrelaciones) complejos entre los datos.

Siguiendo con el ejemplo anterior, esto se refleja en la necesidad de "coordinar" las ideas y procesan los datos mediante interrelaciones que efectuamos constantemente.

Un motivo fundamental que justifica el cambio de sistemas con archivos convencionales a sistemas de bases de datos es el bajo costo de crear nuevas aplicaciones en comparación con el costo que insumiría la actualización de los sistemas con archivos tradicionales.

Hay que tener en cuenta dos aspectos imprescindibles que maneja el sistema relacional:

- 1) Bases de datos relacionales, o sea, relaciones presentadas simplemente como tablas.

- 2) Las operaciones de restricción (o selección), proyección, y reunión natural, sin requerir definiciones previas de rutas de acceso físico para realizar estas operaciones.

Además la disponibilidad de lenguajes de acceso a datos de alto nivel simplifica la tarea de escribir aplicaciones de bases de datos.

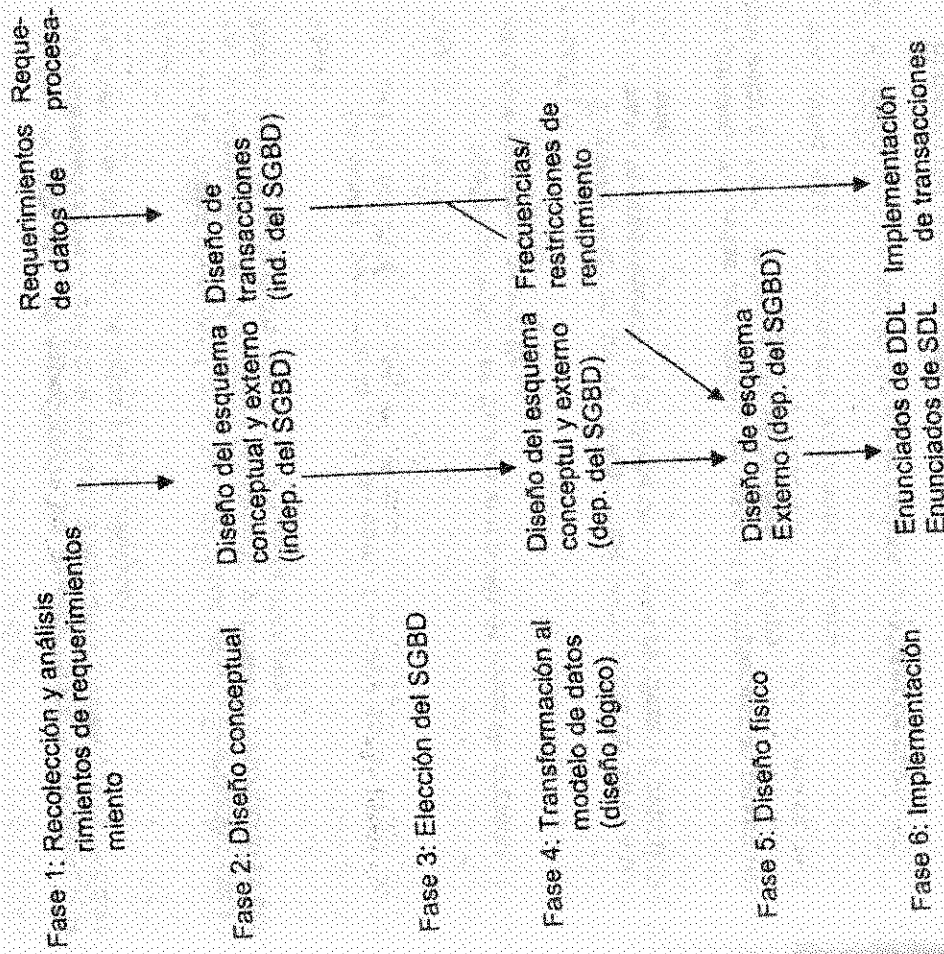
12.2 CICLO DE VIDA DE SISTEMA DE APLICACIÓN DE BD

Para analizar dicho punto, incluiremos ocho fases en este ciclo de vida:

(Esto es un análisis propuesto en el libro de Sistemas de Bases de Datos - Conceptos Fundamentales - Elmasri/Navethe, con la cual estoy completamente de acuerdo).

- 1) Definición del sistema: en esta etapa se definen el alcance del sistema de bases de datos, los usuarios que utilizarán la misma y sus aplicaciones generales y específicas.
- 2) Diseño: se confecciona un diseño completo lógico y físico del sistema de bases de datos en el sistema de gestión de bases de datos (SGBD) elegido.
- 3) Implementación: se escriben las definiciones correspondientes al esquema conceptual, externo e interno de la base de datos; se crean archivos de base de datos vacíos y se implementan las aplicaciones del software.
- 4) Carga o conversión de los datos: en algunos proyectos, la conversión de base de datos conlleva mucho más trabajo (y más planeación estratégica) que el desarrollo de programas. En otros casos, puede no haber existido una base de datos y esto da lugar a la carga completa de la misma.
- 5) Conversión de las aplicaciones: luego de esta fase se está en condiciones de efectuar la instalación.
- 6) Prueba y validación: se probará con datos reales en paralelo con el sistema existente, para no provocar, en caso de fallas, problemas en la organización existente.
- 7) Operación
- 8) Supervisión y mantenimiento

12.3 REPRESENTACIÓN DEL PROCESO DE DISEÑO DE BD



Cuadro extraído del libro **Sistemas de Bases de Datos - Conceptos Fundamentales** - Elmasri/Navethe, segunda edición, Addison-Wesley Iberoamericana

Fase 1:

Durante la recolección y análisis de requerimientos, los diseñadores mantienen entrevistas con los usuarios de la base de datos para poder interpretar y documentar sus necesidades en lo que respecta a requerimientos de información.

En paralelo con esto, se especificarán los requerimientos funcionales, o sea, las necesidades del usuario con respecto a las transacciones que se realizarán, incluyendo la obtención de datos y su actualización.

Fase 2:

El diseño conceptual de la base de datos es una descripción clara de los requerimientos de información del usuario, y contiene descripciones detalladas de los tipos de datos, los vínculos y las restricciones.

Fase 3:

Se debe implementar la base de datos diseñada en un SCBD comercial (se elige el más apropiado).

Fase 4:

El diseño conceptual se traduce al modelo de datos de implementación (es lo que llamamos diseño lógico).

Fase 5 y 6:

El diseño físico es donde se especifican estructuras de almacenamiento internas y la organización de archivos de la base de datos, y todo al fin, se implementa.

En paralelo con estas actividades, se diseñan e implementan programas de aplicación en forma de transacciones de base de datos.

Como vemos, se compone de dos actividades paralelas:

- 1) El diseño de contenido de datos y estructura de la base de datos.
- 2) El diseño del procesamiento de la base de datos y de las aplicaciones del software.

Las seis fases que describimos no tienen una razón lógica para que se ejecuten en ese orden.

El fundamento del proceso de diseño lo manejan las fases de diseño conceptual, lógico y físico (2, 4 y 5, respectivamente).

Para el diseño conceptual, generalmente se utiliza un modelo de datos como un ER (entidad-relación) o un EER (entidad-relación extendido). Para desarro-

llar este tema, esperamos el próximo capítulo, en donde se detallará las definiciones y distintas metodologías gráficas.

12.4 PAUTAS PARA EL DISEÑO FÍSICO

Los factores que se deben tener en cuenta para el diseño físico son:

- Análisis de las consultas y transacciones (en relación con su calidad).
- Análisis de las consultas y transacciones (con respecto a la frecuencia esperada de invocación).
- Análisis de las consultas y transacciones (con relación a las restricciones de tiempo sobre ellas).
- Análisis de la frecuencia esperada de las operaciones de actualización.

Basándonos en dichos factores influyentes, las pautas para el diseño físico de sistemas relacionales constan de:

- Aplicación de técnicas para optimizar las operaciones hechas sobre la base de datos.
- Prestar un cuidado especial en la organización de los archivos de la base de datos y la selección de los índices.

12.5 HERRAMIENTAS AUTOMATIZADAS DE DISEÑO

En general, las características que debe poseer una buena herramienta de diseño son:

- Una interfaz fácil de usar: debo manejarla en un entorno sin grandes complicaciones de uso.
- Componentes analíticos.
- Componentes heurísticos: esto significa componentes o reglas generales en forma de opiniones que sugieren procedimientos que se pueden seguir cuando no existen reglas de procedimiento invariables.

Dichas reglas son aproximadas y han sido generadas por un experto a través de años de experiencia.

Por ejemplo: si una sierra parece estar bien pero aún así no arranca, añádele la tensión de la cadena.

El uso de heurísticas tiende a la potencia y flexibilidad del sistema.

- Análisis de ventajas y desventajas.
- Exhibición de resultados del diseño.
- Verificación del buen diseño.

Capítulo XIII

Modelado semántico

13.1 INTRODUCCIÓN

Semántica es el estudio de significados. Por lo tanto, un "modelado semántico" sirve para representar el significado.

Cuando la actividad se realiza en el contexto del diseño de una base de datos, se conoce también con otros nombres, sobre todo para una aplicación específica (modelado de entidades/interrelaciones, modelado de entidades).

Se han propuesto varias metodologías de diseño por distintos autores. Analizaremos dos de dichas propuestas, el modelo de entidad/interrelación de Chen y el diagrama de entidad/relación (DER) de Martin, tratando en cada una de ellas, fundamentar la conveniencia o no de su uso y en qué casos se recomienda su aplicación (opinión puramente personal de quien les escribe).

13.2 ENFOQUE GENERAL

El enfoque general del problema del modelado semántico podemos caracterizarlo en pasos.

1. Identificación de un conjunto de conceptos semánticos.
Nos referimos a entidad, atributo, clase, subclase, interrelación, los que pasaremos a explicar con detalle más adelante.
2. Identificación de un conjunto de objetos simbólicos (formales) que puedan emplearse para representar los conceptos semánticos anteriores.
3. Desarrollo de un conjunto de reglas de integridad formales.
4. Desarrollo de un conjunto de operadores formales para manipular esos objetos formales.

Ahora nos detenemos en el punto 1 y 2 para ahondar en los conceptos semánticos y su respectiva simbología.

13.2.1 Conceptos semánticos

Entidad

Es el objeto de bases de datos más importante. La entidad representa objetos reales; es un objeto distinguible.

Ejemplo: proveedor, empleado, cliente, producto, etc.

Atributo

El atributo es una característica de una entidad. Es un elemento de información que describe a una entidad.

Ejemplo: nro. de proveedor, su nombre, dirección del cliente, tamaño del producto, etc.

Interrelación

Una entidad que sirve para conectar entre sí a otras dos o más entidades. Ejemplo: envío (proveedor-producto), grabación (composición-orquesta).

Subtipo

El tipo de entidad Y es un subtipo de entidad X si y sólo si todo Y es por fuerza un X.

Ejemplo: Empleado es un subtipo de persona.

a) Clase

Representa una colección de entidades similares (de un mismo tipo).

Ejemplo: todas las personas que trabajan en una compañía son miembros de la clase "empleados".

Cada clase tiene un nombre único en la base de datos.

Las entidades que pertenecen a determinada clase comparten los mismos atributos. Las entidades de una clase no tienen un orden preestablecido.

b) Subclase – superclase

La subclase es un subconjunto de una clase. La clase que sirve como "padre-raíz", se conoce como "superclase".

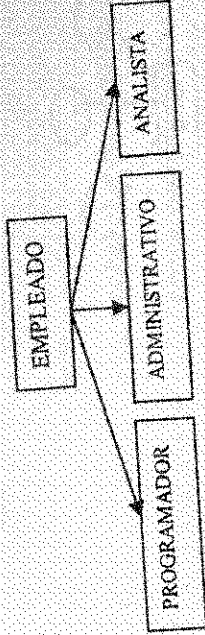
Para que estos conceptos queden bien claros, vamos a entablar una relación entre ellos:

Clase : define un tipo de entidad.
Subclase: define un subtipo de entidad.

Ejemplo:

EMPLEADO, es cualquier persona que trabaja en la compañía. PROGRAMADOR, especifica un tipo de empleado.

O sea, que todos los programadores pasan a ser un subtipo o subclase de empleado. Esto se ilustra con una gráfica de árbol:

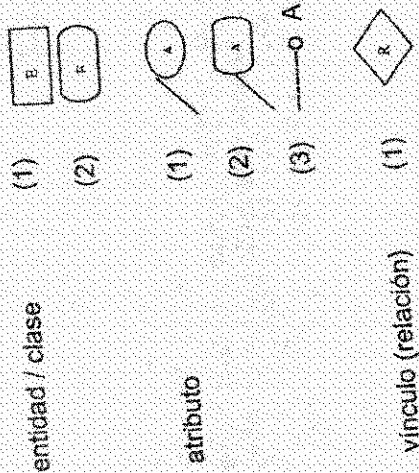


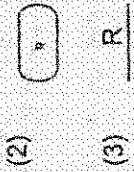
Fijense que tanto programador, como administrativo y analista son subtipos de empleados, los cuales "heredan" los atributos del supertipo.

13.2.2 Objetos simbólicos

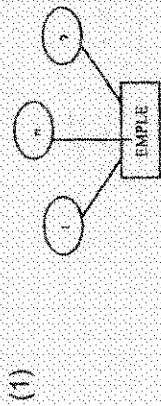
Vamos a presentar varias notaciones diagramáticas alternativas presentadas en el libro de Sistemas de Bases de Datos, Conceptos Fundamentales, de Elmasri/Navethe, 2da. edición, Addison-Wesley Iberoamericana.

a) Símbolos de tipo de:

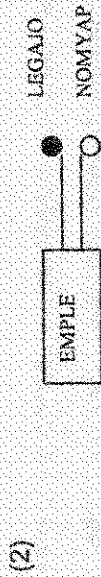




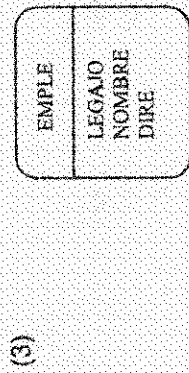
b) Representación de atributos



Este modelo lo estudiaremos con detalle en el modelo de entidad-relación de Chen.

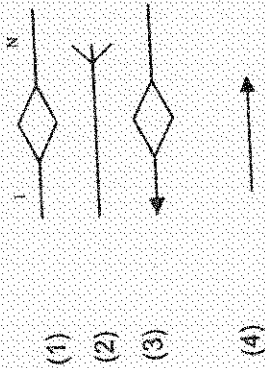


Con la simbología que tiene LEGAJO señala el atributo clave.



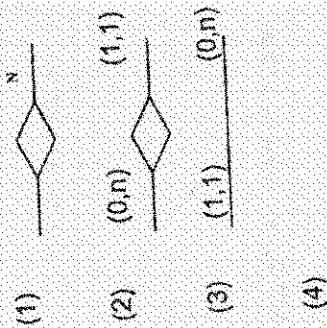
Esto corresponde a la metodología OOA (análisis orientado a objetos) y herramientas CASE (ingeniería de software asistida por computadora).

c) Cardinalidad



Esta última notación se emplea en el diagrama de Bachman.
La cardinalidad se refiere a dejar indicado para cada instancia de una entidad cuántos elementos le puede corresponder de la otra.

d) Restricciones (mínimo y máximo)

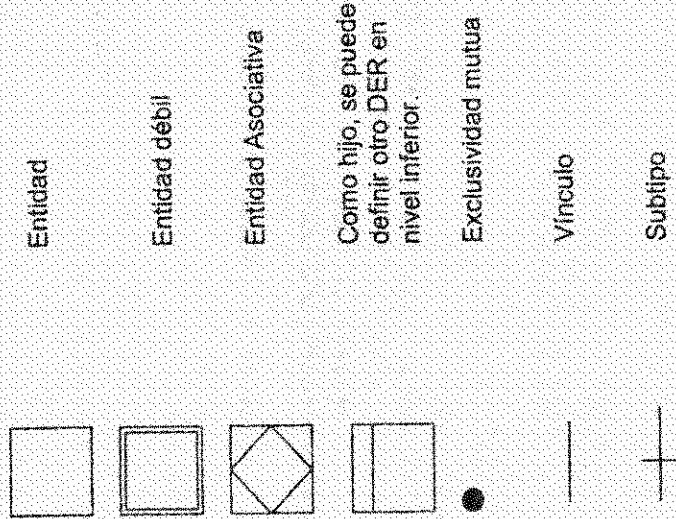


La última simbología pertenece a Martin, y es a mi entender, la forma más clara de indicar la cardinalidad con restricción.
En el siguiente ítem se ejemplificará con un diagrama completo. El trazado de la izquierda indica relación de 1, en cambio el de la derecha señala de 1 a muchos (se conocen como patas de gallo).

No existe una notación estándar.

13.3 DER (MARTIN)

La simbología que utiliza Martin es la siguiente:



La explicación de ENTIDAD (entity) no tiene ningún misterio de lo que había ahora hemos planteado.

La ENTIDAD DÉBIL (weak entity) se refiere a una entidad que sola no existe. Depende de una entidad fuerte.

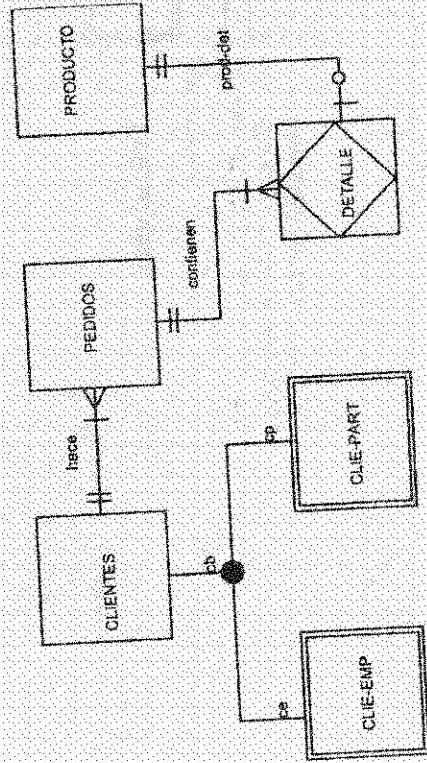
La ENTIDAD ASOCIATIVA vincula dos o más entidades.

SPLIT ENTITY es un símbolo que permite definir otro DER interior con un nivel inferior.

La EXCLUSIVIDAD MUTUA relaciona en una sola rama.

El diagrama de entidad relación que propone Martin sugiere una simbología clara y especialmente la asignación de cardinalidad.

El siguiente DER está realizado con la herramienta EASY CASE, un software muy práctico para diagramas funcionales y de datos.
Ejemplo:



La cardinalidad se explica accediendo a una instancia de cada entidad y analizar su relación con las otras. Por ejemplo:

El cliente X puede hacer 1 o más pedidos.
El pedido Z puede estar hecho sólo por un cliente.
De aquí la relación entre CLIENTES – PEDIDOS de uno a muchos. y PEDIDOS – CLIENTES de uno a uno.

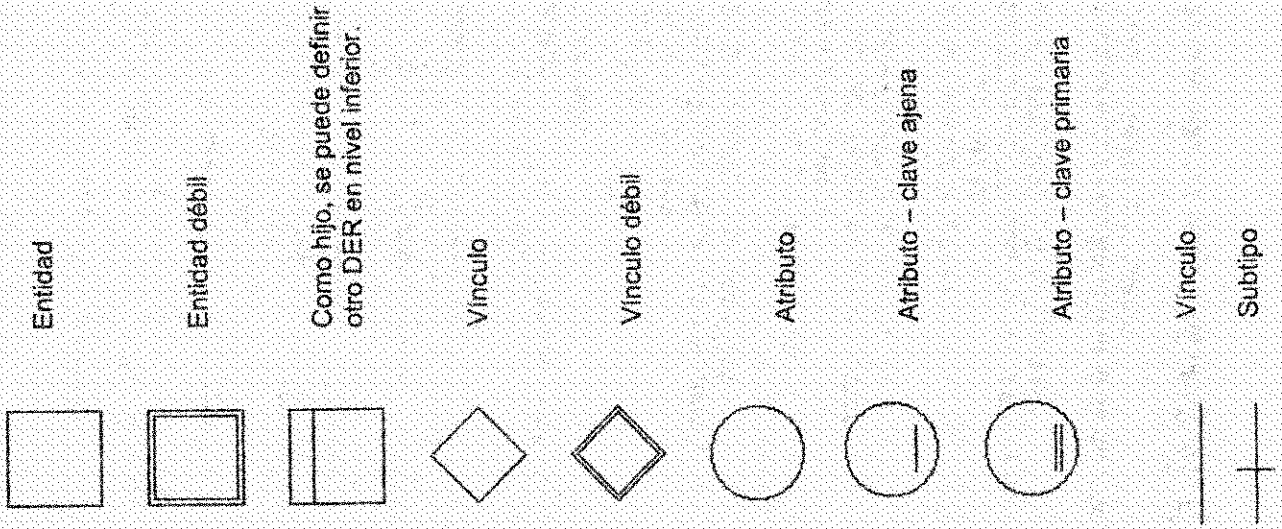
El pedido Y puede contener uno o varios renglones de detalle.
El renglón de detalle 3 pertenece a un solo pedido.
De aquí la relación entre PEDIDOS – DETALLE de uno a muchos. y DETALLE – PEDIDOS de uno a uno.

El producto A figura en ninguno o un renglón de detalle.
El renglón de detalle 2 contiene un solo producto.
De aquí la relación entre PRODUCTOS – DETALLE de cero a uno. y DETALLE – PRODUCTOS de uno a uno.

La exclusividad mutua que aparece debajo de CLIENTES trata de mostrar una clasificación de clientes de empresa y clientes particulares.

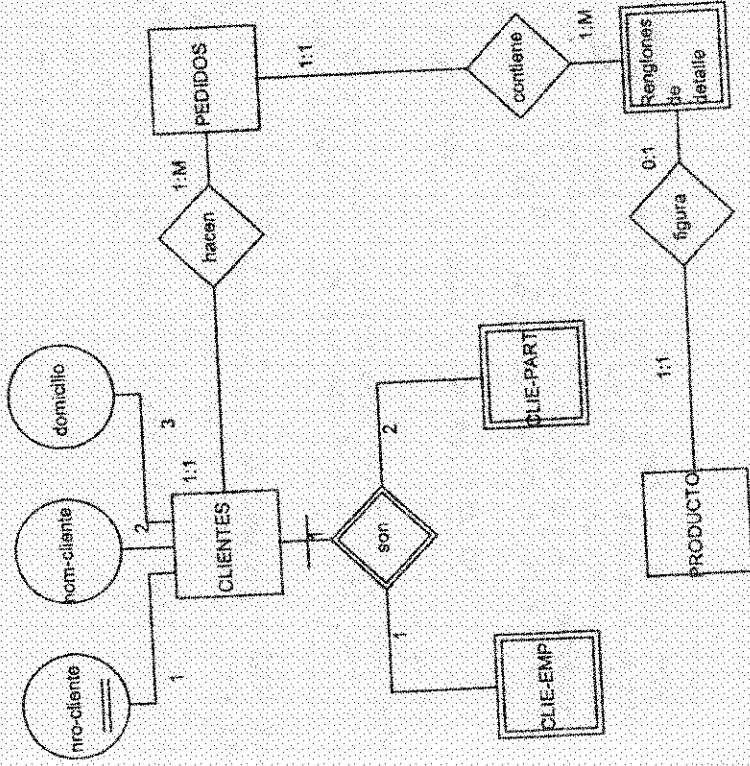
La entidad asociativa DETALLE no es más que una interrelación entre PEDIDOS Y PRODUCTOS.

13.4 MODELO ENTIDADES/INTERRELACIONES (CHEN)



Uno de los enfoques más conocidos es el de E/R: entity/relationship.

Ejemplo:



Entidad: Chen clasifica las entidades como cosas que se pueden identificar claramente y las divide en:

- Entidades regulares
- Entidades débiles

Una entidad débil es cuya existencia depende de otra entidad, o sea, no puede existir si no existe también la otra entidad.

Ejemplo: renglones de detalle existen porque está el pedido (sin él tampoco hay descripción).

Una entidad regular es una entidad que no es débil.

Propiedad: también conocidas como atributos. Todas las entidades de un tipo determinado tienen ciertas propiedades en común. Todos los clientes tienen un número, un nombre, un domicilio, etc. Se representan con círculos y el nombre doblemente subrayado indica la clave primaria. El subrayado simple señala la clave ajena.

Estas propiedades pueden ser:

- Simples (nom-clie)
- Compuestas
- Clave (es decir, únicas)
- Univaluadas o multivaluadas (es decir, se permiten grupos repetitivos).
- Base o derivadas (por ejemplo: cantidad total de una parte dada podría derivarse sumando las cantidades base de los envíos individuales de esa parte).

Interrelación: Chen la define como una vinculación entre entidades. Se dice que las entidades implicadas en una interrelación dada son los participantes de esa interrelación.

La cardinalidad se indica con número 0 o 1 y con la M, para muchos.

Subtipo: Por ejemplo: en el caso de clientes de empresa y clientes particulares son un subtipo del supertipo CLIENTES. Todas las propiedades de clientes se aplican de manera automática a los subtipos, pero lo contrario no se cumple. Cada uno de los tipos de entidad mostrados en una jerarquía de tipos corresponderá a una relación base. Cada una de estas relaciones incluirá atributos propios y subordinados por herencia.

Las jerarquías de tipos se conocen con varios nombres, entre ellos:

- Jerarquías de generalización
- Jerarquías de especialización

La generalización es el resultado de distinguir un tipo de objeto como más general.

ejemplo:



13.5 EJERCICIOS

13.5.1 Unir de acuerdo a la jerarquía de generalización los siguientes niveles zoológicos:

	Mamífero	Oso
Animal acuático		Llama
		Búfalo
		Ballena
	Pez	Murciélago
Animal volador		Caballito de mar
		Águila
	Ave	Flamenco
		Halcón
		Pingüino

13.5.2 Realizar un DER (Martin) para representar el ejercicio 11.7.8 del capítulo XI, en la página 109.

13.5.3 Realizar un Diagrama ER (Chen) para representar el ejercicio 11.7.8 del capítulo XI, en la página 109.

13.5.4 Para los ejercicios 10.6.6 y 10.6.8 del capítulo X, en la página 93, diseñar un diagrama de entidad relación (puede hacerlo con cualquiera de los dos propuestos).